



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SUPERVISIÓN, CONTROL Y ADQUISICIÓN DE DATOS (SCADA) PARA UN PROTOTIPO A ESCALA DE PARQUE EÓLICO

AUTOR: JULIO YUSTAS TALAMANTES

TUTORA: SOLEDAD INMACULADA BERNAL PÉREZ

Curso Académico: 2019-20

Agradecimientos

“En primer lugar, me gustaría agradecer a mi tutora, la Dra. Soledad Inmaculada Bernal Pérez, la oportunidad brindada con este maravilloso proyecto y su inagotable esfuerzo durante todos estos meses de trabajo.

A mi familia y amigos, ya que, sin su cariño y aliento en los momentos difíciles jamás lo hubiera logrado; esto también es vuestro.

A las personas que están, a las que estuvieron; a las personas que me acompañan y a los que desgraciadamente ya se fueron.

Para finalizar, no me agradaría despedirme sin dar las gracias a todos los maestros que me han acompañado y guiado durante toda mi trayectoria como estudiante; gracias por todo el conocimiento que me habéis regalado y sobre todo, gracias por educarme y hacerme mejor persona.

Mamá, Papá, Verónica, Diana, ¡aiios! esto es más vuestro que mío, ojalá os podáis sentir orgullosos y lo disfrutéis.”

“ Strive for perfection in everything you do.
Take the best that exists and make it better.
When it does not exist, design it.”

- Sir Henry Royce

Resumen

El veloz avance tecnológico durante las últimas décadas ha generado un crecimiento sin precedentes en las nuevas formas de generación de energía y más concretamente en las energías de origen renovable.

La imperiosa necesidad de quemar etapas en el desarrollo tecnológico de las energías renovables hasta alcanzar su plena disponibilidad en el mercado, ha provocado que se dediquen muchos esfuerzos y recursos a la investigación con la finalidad de optimizar estas nuevas vías.

En este escenario, la correcta monitorización de las centrales de generación energética se antoja fundamental para analizar posibles errores y evitar la utilización de recursos de forma innecesaria. El presente Trabajo de Fin de Grado tiene como objetivo el diseño e implementación de un sistema de supervisión, control y adquisición de datos (SCADA) para un prototipo a escala de parque eólico.

Para ello se ha generado un código completo, utilizando el software de programación de LabVIEW; incluido un sistema de identificación para garantizar la seguridad del acceso a la información. Desde este sistema, el cliente tendrá la posibilidad de realizar diversas tareas: control de un parque en tiempo real, simulación, consulta de parámetros en la base de datos, realización de maniobras sobre los equipos de protección y consulta del archivo de errores entre otros.

Palabras Clave: SCADA, renovables, control, eólica, software TFG

Resum

El veloç avanç tecnològic durant les últimes dècades ha generat un creixement sense precedents en les noves formes de generació d'energia i més concretament en les energies d'origen renovable.

La imperiosa necessitat de cremar etapes en el desenvolupament tecnològic de les energies renovables fins a aconseguir la seua plena disponibilitat en el mercat, ha provocat que es dediquen molts esforços i recursos a la investigació amb la finalitat d'optimitzar aquestes noves vies.

En aquesta situació, el correcte monitoratge de les centrals de generació d'energia s'antulla fonamental per a analitzar possibles errors i evitar la utilització de recursos de manera innecessària. El present Treball de Fi de Grau té com objectiu el disseny i implementació d'un sistema de supervisió, control i adquisició de dades (SCADA) per un prototip a escala de parc eòlic.

Per a això s'ha generat un codi complet, utilitzant el codi de programació de LabVIEW; inclòs un sistema d'identificació per a garantir la seguretat de l'accés a la informació. Des d'aquest sistema, el client tindrà la possibilitat de fer diverses tasques: control d'un parc en temps real, simulació, consulta de paràmetres a la base de dades, realització de maniobres sobre els equips de protecció i consulta de l'arxiu d'errors entre d'altres.

Paraules clau: SCADA, renovables, control, eòlica, software

Abstract

The rapid technological progress during the last decades has generated an unprecedented growth in new forms of energy generation and more specifically in energies of renewable sources.

The imperative need to burn out stages in the technological development of renewable energies until they are fully available on the market, has led to a great deal of effort and resources being devoted to research in order to optimise these new ways.

In this scenario, the correct monitoring of the new power generation plants appears to be fundamental for analysing possible errors and avoiding the use of resources in an unnecessary way. This Thesis aims to design and implement a supervision, control, and data acquisition system (SCADA) for a wind farm scale prototype.

For this purpose, a complete code has been generated, using LabVIEW's programming software; including an identification system to guarantee the security of the access to the information. From this system, the user will have the possibility to perform various tasks: control of a wind farm in real time, simulation, query of parameters in the database, carrying out manoeuvres on the protective equipment and query of the error file among others.

Keywords: SCADA, renewable, control, wind energy, software

Índice general

Resumen	III
Índice general	XIII
I Memoria	1
I.1 Introducción	3
I.1.1 Objeto del proyecto	5
I.1.2 Alcance	6
I.1.3 Antecedentes	7
I.1.3.1 Antecedentes: Breve historia de la energía eólica	7
I.1.3.2 Antecedentes: Evolución de los sistemas SCADA	7
I.1.4 Justificación	9
I.1.4.1 Personal	9
I.1.4.2 Académica	9
I.1.4.3 Industrial	10
I.1.5 Motivación	10
I.1.6 Normativa aplicada	10
I.1.7 El proyecto PROMOTioN	12
I.1.8 Objetivos de PROMOTioN	13
I.1.9 Socios	13
I.1.10 Organización del proyecto	14
I.1.11 Epítome del contenido del documento	15
I.2 Sistema SCADA: Análisis de alternativas	17
I.2.1 ¿Qué es un sistema SCADA?	19
I.2.2 Ejemplos de sistemas SCADA en la industria	20
I.2.3 Análisis de alternativas	21

I.3 Simulación de señales	23
I.3.1 Tipos de señales a simular	25
I.3.2 Software empleado	26
I.3.3 Protocolos de envíos de señal	28
I.3.4 Base de datos	30
I.4 Configuración y programación	33
I.4.1 Configuración física	35
I.4.2 Configuración digital	35
I.4.3 Requisitos del sistema	38
I.4.4 Estructura del programa	39
I.4.5 Control y medida	40
I.4.6 Elementos adicionales	41
I.5 Solución del problema	43
I.5.1 Introducción	45
I.5.2 Diseño	46
I.5.3 Implementación	48
I.5.3.1 Fase de inicialización	48
I.5.3.2 Selección de modo	57
I.5.3.3 Interfaz visual	62
I.5.4 Código en MATLAB	84
I.5.4.1 Simulador	84
I.5.4.2 Banco de pruebas	85
I.5.4.3 Breaker	85
I.5.5 Posibles mejoras	85
I.6 Verificación y validación	87
I.6.1 Verificación vs. Validación	89
I.6.2 Técnicas	89
I.6.3 Proceso de depuración	89
I.6.4 Listado de fallos	91
I.6.5 Pruebas funcionales: caja negra	92
I.6.6 Pruebas de integración incremental	92
I.7 Resultados finales, conclusiones, visión futura y ampliaciones	95
I.7.1 Resultados finales y conclusiones	97
I.7.2 Visión futura y ampliaciones	98
II Estudio económico	101
II.1 Planificación temporal	103
II.1.1 Diagrama de Gantt	104

II.2 Presupuesto del proyecto	109
II.2.1 Presupuesto del proyecto	111
II.2.2 Precios descompuestos	111
II.2.3 Presupuesto de ejecución material por capítulos	117
II.2.4 Presupuesto de ejecución por contrata	118
III Pliego de condiciones técnicas y administrativas	119
III.1 Pliego de condiciones técnicas y administrativas	121
III.1.1 Entidad que inicia el procedimiento	123
III.1.2 Objeto del proyecto	123
III.1.3 Antecedentes y situación actual	123
III.1.4 Alcance del proyecto	123
III.1.5 Descripción de las tareas a realizar	124
III.1.6 Compatibilidad	124
III.1.7 Entorno tecnológico	124
III.1.7.1 Entorno de programación MATLAB	124
III.1.7.2 Entorno de usuario MATLAB	125
III.1.7.3 Entorno de programación LabVIEW	125
III.1.7.4 Entorno de usuario LabVIEW	126
III.1.7.5 Entorno de Wireshark	127
III.1.8 Metodología	127
III.1.9 Mecanismos de seguimiento y control	128
III.1.10 Documentación	128
III.1.11 Idioma	128
IV Planos	129
IV.1 Listado de planos	131
IV.2 Planos	133
V Manuales de usuario	163
V.1 Manual de usuario (Castellano)	165
V.2 Manual de usuari (Valencià)	191
V.3 User Manual (English)	217
V.4 Benutzerhanbuch (Deutsch)	243

VI	Manuales de programador	269
VI.1	Manual de programador (Castellano)	271
VI.2	Manual de programador (Valencià)	307
VI.3	Programmer's Manual (English)	343
VI.4	Programmiererhanbuch (Deutsch)	379
VII	Anexos	415
VII.1	Breve historia de la energía eólica	417
VII.1.1	Breve història de la energia eólica	419
VII.2	Contexto energético	429
VII.2.1	Situación energética	431
VII.2.1.1	Situación energética: Nivel mundial	431
VII.2.1.2	Situación energética: Nivel europeo	433
VII.2.1.3	Situación energética: Nivel nacional.	435
VII.2.2	Problema energético	436
VII.2.2.1	La sostenibilidad energética	436
VII.2.2.2	El Cambio Climático: Un problema Real	437
VII.2.3	Energías renovables.	438
VII.2.3.1	Definición	438
VII.2.3.2	Clasificación	438
VII.2.3.3	Papel en el problema energético	438
VII.2.4	Energías Eólica	439
VII.2.4.1	¿Qué es la energía eólica y como se produce?	439
VII.2.4.2	Determinación del emplazamiento de un aerogenerador.	440
VII.2.4.3	Usos de la energía eólica.	442
VII.2.4.4	Clasificación	442
VII.2.4.5	Aerogenerador	444
VII.2.4.6	Ventajas e inconvenientes de la energía eólica.	446
	Bibliografía	447

Índice de figuras

I.1.1.	Objetivos definidos por el proyecto europeo PROMOTioN	6
I.1.2.	Lugar que ocupa este trabajo dentro del proyecto general. Fuente: PROMOTioN	6
I.1.3.	Evolución de los sistemas SCADA (infografía)	8
I.1.4.	Lugar que ocupa este trabajo dentro del proyecto general	14
I.2.1.	Diagrama de flujo de un sistema SCADA habitual	19
I.2.2.	Front view de SpaceX. Güncelleme 03/07/2020	20
I.3.1.	Esquema de comunicación	25
I.3.2.	Esquema de comunicación sistema SCADA (parte 1/2)	26
I.3.3.	Esquema de comunicación sistema SCADA (parte 2/2)	27
I.3.4.	Captura entorno de programación de simulink (MATLAB)	27
I.3.5.	Captura entorno de programación de LabVIEW	28
I.3.6.	Captura entorno de Wireshark	29
I.3.7.	Esquema de envío protocolos TCP y UDP	29
I.3.8.	Recorte envío Simulink - LabVIEW vía UDP	30
I.3.9.	Parámetros del envío UDP	30
I.3.10.	Protocolo UDP en LabVIEW	31
I.3.11.	Bases de datos	31
I.4.1.	Recortes de las gráficas generadas por MATLAB	36
I.4.2.	Esquema general de comunicación entre códigos	37
I.4.3.	Recortes proceso de obtención IP	38

I.4.4.	Recortes de los foros de los diferentes softwares	39
I.4.5.	Recorte de la recepción de datos vía UDP por parte de Simulink	40
I.4.6.	Recorte de las "probe" en LabVIEW	41
I.4.7.	Recorte del correo electrónico de confirmación	41
I.5.1.	Kostal monitoring portal,Kostal 10/08/2020	45
I.5.2.	Diagrama de flujo simplificado	47
I.5.3.	Recorte de la "front view" del inicio de sesión de usuario	48
I.5.4.	Infografía elementos operables	49
I.5.5.	Recorte del "block diagram" del inicio de sesión de usuario	50
I.5.6.	Componentes del "block diagram" del inicio de sesión de usuario	51
I.5.7.	Recorte de la "front view" del registro de usuario	52
I.5.8.	Recorte del correo de confirmación	52
I.5.9.	Infografía elementos operables	53
I.5.10.	Recorte del "block diagram" de registro de usuario	54
I.5.11.	Componentes del "block diagram"	55
I.5.12.	Componentes del "block diagram"	56
I.5.13.	Recorte de la "front view" del menú	57
I.5.14.	Infografía elementos operables	57
I.5.15.	Recorte del "block diagram" del menú	59
I.5.16.	Componentes del "block diagram"	60
I.5.17.	Recorte de la "front view" (1/2)	62
I.5.18.	Recorte de la "front view" (2/2)	62
I.5.19.	Infografía elementos operables (1/3)	63
I.5.20.	Infografía elementos operables (2/3)	63
I.5.21.	Infografía elementos operables (3/3)	64
I.5.22.	Recorte del "block diagram" (1/2)	65
I.5.23.	Recorte del "block diagram" (2/2)	65
I.5.24.	Componentes del "block diagram"	66
I.5.25.	Recorte de la "front view"	68
I.5.26.	Infografía elementos operables	68
I.5.27.	Recorte del "block diagram"	69
I.5.28.	Componentes del "block diagram"	70

I.5.29.	Recorte de la "front view"	71
I.5.30.	Infografía elementos operables	71
I.5.31.	Recorte del "block diagram"	72
I.5.32.	Componentes del "block diagram"	72
I.5.33.	Recorte de la "front view"	74
I.5.34.	Infografía elementos operables	74
I.5.35.	Recorte del "block diagram"	75
I.5.36.	Componentes del "block diagram"	75
I.5.37.	Recorte de la "front view"	77
I.5.38.	Infografía elementos operables	77
I.5.39.	Recorte del "block diagram"	78
I.5.40.	Componentes del "block diagram"	79
I.5.41.	Recorte de la "front view"	80
I.5.42.	Infografía elementos operables (1/2)	80
I.5.43.	Infografía elementos operables (2/2)	81
I.5.44.	Recorte del "block diagram"	82
I.5.45.	Componentes del "block diagram"	83
I.5.46.	Recorte del código de simulador	84
I.5.47.	Recorte del código de banco de pruebas	85
I.5.48.	Recorte del código de elementos de protección	85
I.6.1.	Esquema de las técnicas de comprobación	90
I.6.2.	Recorte de LabVIEW	90
I.6.3.	Recorte creación ejecutable	91
I.6.4.	Flujograma corrección errores	91
I.6.5.	Esquema de las pruebas de integración	93
II.2.1.	Amortizaciones	111
VII.1.1.	Boceto de turbina realizado por Brush(windstory)	419
VII.1.2.	Diseño de aerogenerador ideado por la Cour (windstory)	420
VII.2.1.	Evolución poblacional	431
VII.2.2.	Infografía sobre el consumo y la demanda global	432

VII.2.3.	Recortes de la infografía energética publicada por BP	432
VII.2.4.	EU-28 Energy Flow 2000. Fuente: Eurostat, May 2019	433
VII.2.5.	EU-28 Energy Flow 2017. Fuente: Eurostat, May 2019	433
VII.2.6.	Consumo energético primario. Fuente: EU energy in figures. Statistical pocketbook 2019	434
VII.2.7.	Total Primary Products	434
VII.2.8.	Energía eólica en Europa	434
VII.2.9.	Procedencia de la energía en España. Fuente: Eurostat, May 2019	435
VII.2.10.	Dependencia exterior de España. Fuente: Eurostat, May 2019	435
VII.2.11.	Objetivos de energías renovables en España. Fuente: Eurostat-RES SHARES, March 2019	436
VII.2.12.	Principales factores del denominado como problema energético	436
VII.2.13.	Pilares para alcanzar la sostenibilidad energética	437
VII.2.14.	PIB vs. % de electrificación	437
VII.2.15.	Esquema de las energías renovables y su procedencia	438
VII.2.16.	Ejemplo de impacto ambiental para autoconsumo fotovoltaico de 2 kW. Fuente: Prácticas personales en Solats SL	439
VII.2.17.	Ciclo eólico. Fuente: <i>Energía Eólica: Guía Técnica</i>	440
VII.2.18.	Potencia eólica	440
VII.2.19.	Ejemplo de rosa de los vientos. Fuente: <i>Energía Eólica: Guía Técnica</i>	441
VII.2.20.	Ejemplo de la influencia de la orografía. Fuente: <i>Energía Eólica: Guía Técnica</i>	441
VII.2.21.	Influencia de los obstáculos y separación necesaria. Fuente: <i>Energía Eólica: Guía Técnica</i>	441
VII.2.22.	Recorte del "Global Wind Atlas". Atlas 21/06/2020	442
VII.2.23.	Boceto de ambos tipos de aerogeneradores Gate 14/06/2020	443
VII.2.24.	Aerogeneradores a barlovento y a sotavento. Fuente: <i>Cuaderno de aplicaciones técnicas n.12: Plantas eólicas</i>	443
VII.2.25.	Diferencias entre los aerogeneradores de 2 y 3 palas. Fuente: <i>Cuaderno de aplicaciones técnicas n.12: Plantas eólicas</i>	443
VII.2.26.	Velocidad específica	444
VII.2.27.	Infografía en función de la velocidad específica y el coeficiente de potencia. Fuente: <i>Cuaderno de aplicaciones técnicas n.12: Plantas eólicas</i>	444
VII.2.28.	Principio de funcionamiento de una turbina eólicas	445
VII.2.29.	Comparativa generación de ruido de un generador frente a otras acciones. Fuente: <i>Cuaderno de aplicaciones técnicas n.12: Plantas eólicas</i>	446

Índice de tablas

I.2.1.	Tabla comparativa de los sistema SCADA posibles para este proyecto	22
VII.2.1.	Ventajas e inconvenientes de la energía eólica	446

Parte I

Memoria

Capítulo I.1

Introducción:

En este capítulo se describe el objeto del proyecto, la motivación tanto académica como personal del mismo y su alcance profesional en relación a todo aquello estudiado en el Grado de Ingeniería en Tecnologías Industriales.

A su vez, se realizará un repaso sobre el proyecto en cuestión PROMOTioN y se detallarán las tareas realizadas por parte de la Universitat Politècnica de València.

Índice

I.1.1 Objeto del proyecto.	5
I.1.2 Alcance	6
I.1.3 Antecedentes.	7
I.1.3.1 Antecedentes: Breve historia de la energía eólica	7
I.1.3.2 Antecedentes: Evolución de los sistema SCADA	7
I.1.4 Justificación	9
I.1.4.1 Personal	9
I.1.4.2 Académica	9
I.1.4.3 Industrial	10
I.1.5 Motivación	10
I.1.6 Normativa aplicada	10
I.1.7 El proyecto PROMOTioN.	12
I.1.8 Objetivos de PROMOTioN.	13
I.1.9 Socios	13
I.1.10 Organización del proyecto	14
I.1.11 Epítome del contenido del documento.	15

I.1.1 Objeto del proyecto

La finalidad última del presente trabajo de fin de grado en Ingeniería en Tecnologías Industriales de la Universitat Politècnica de València consiste en la creación de un entorno visual, atractivo y sencillo mediante un sistema SCADA para la correcta monitorización de parques eólicos marinos.

Este trabajo forma parte del proyecto europeo dentro del horizonte 2020 PROgress on Meshed HVDC Offshore¹ Transmission Networks , un proyecto cuyo diseño es el desbloqueo de todo el potencial eólico marino en el continente europeo; aportando soluciones a las barreras técnicas, legales, normativas, económicas y financieras que presentan este tipo de planes.

Los propósitos de este proyecto se pueden observar de forma más gráfica en la figura I.1.1, en ella se representan los pilares de la investigación. Se trata de conseguir un equilibrio real entre las diferentes variables: económica, social, jurídica y tecnológica; económicamente los proyectos de generación de energía deben de ser realizables y el precio de venta del kwh ha de ser accesible para el usuario; esto lleva a la segunda dimensión, la social, donde se debe respetar no solo su economía sino también las tradiciones y rasgos propios de los habitantes de cada zona. Estos dos pilares no tendrían sentido alguno sin las otras dos patas de la mesa: el avance tecnológico, fundamental para alcanzar la madurez de estas nuevas fuentes energéticas y la dimensión legal imprescindible para conseguir un marco jurídico que avale estos proyectos.

El proyecto plantea así avanzar en el desarrollo tecnológico de una red europea mallada en continua (meshed HVDC grid) que permita aprovechar los recursos eólicos marinos disponibles. Los principales retos a los que se enfrenta el proyecto son la interoperabilidad de convertidores, así como diseño de estrategias y dispositivos de protección (DC breakers).

Una de las áreas de investigación del proyecto implica el desarrollo de controles en aerogeneradores y parques eólicos marinos.

Para la validación de los controles diseñados se dispone de herramientas para la simulación en tiempo real que permitan la integración de equipos de protección comercial. En el Instituto de Automática e Informática Industrial (ai2) se dispone de dicho simulador, siendo necesaria una herramienta que permita integrar señales, monitorizando los ensayos de validación, registro de datos y gestión de toda la información. El objeto del proyecto es la creación de ese entorno que posibilite dicha monitorización.

Con este fin se definen los siguientes objetivos principales:

- Conexión del simulador virtual y/o el prototipo a escala con el sistema SCADA mediante los protocolos TCP² IP y UDP³ .
- Monitorización del parque eólico (tensiones, potencias, etc.)
- Desarrollo de un sistema SCADA y, por consiguiente, de una interfaz de usuario.
- Creación de una base de datos para la representación y consulta del histórico de datos.
- Configuración de los ensayos a realizar (selección de estrategias de control, generación de faltas, etc).

El soporte documental del proyecto lo constituye la presente memoria que incluye referencia a la normativa aplicable, una valoración económica para analizar su viabilidad y calcular el coste del

proyecto, la creación de un manual de usuario y otro de programador y un vídeo explicativo del mismo. 4

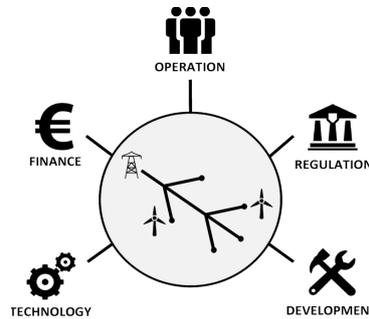


Figura I.1.1: Objetivos definidos por el proyecto europeo PROMOTiON

I.1.2 Alcance

En este caso, el trabajo de fin de grado expuesto se centra en el desarrollo de una interfaz de usuario que satisfaga las necesidades existentes y que proporcione, tanto al usuario como al equipo técnico, una herramienta capaz de interpretar los datos y una plataforma sencilla de comunicación con el simulador.

La actual situación de alarma sanitaria nacional provoca que sea imposible trabajar en el laboratorio de la universidad como se había planteado inicialmente.

En resumen, el alcance de este proyecto engloba desde el preacondicionamiento y tratado de las señales provenientes del simulador y/o prototipo a escala hasta su representación, pasando por la detección de errores o el control del interruptor de protección "breaker".

Es destacable a su vez, que dicha interfaz de usuario o programa dispondrá de la capacidad de trabajar como un propio simulador en sí; permitiendo al usuario generar señales manualmente y comprobar posibles errores en el aerogenerador. Todo lo anteriormente expuesto vendrá comprendido dentro de una interfaz totalmente inteligente, con menús de ayuda, registro e identificación, entre otros. El lugar que ocupa el presente trabajo se puede observar de forma gráfica en la figura I.1.2.

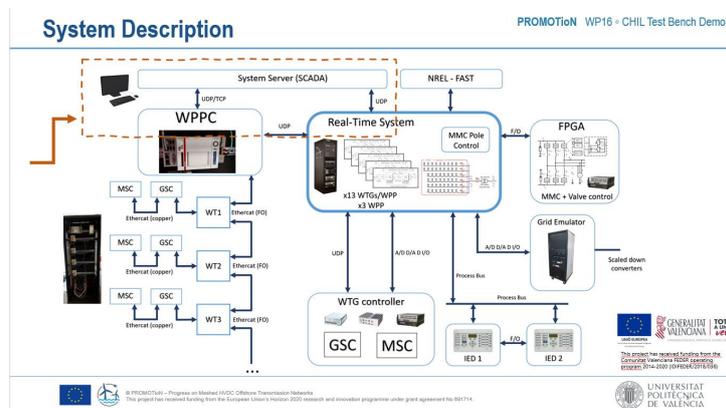


Figura I.1.2: Lugar que ocupa este trabajo dentro del proyecto general. Fuente: PROMOTiON

I.1.3 Antecedentes

A priori no resulta sencillo acotar los precedentes del presente trabajo. Sería necesario mencionar la generación de energía eólica, el control de aerogeneradores, la monitorización de procesos o las estrategias de protección de las instalaciones eléctricas. Se han seleccionado las que más influyen en la motivación de proyecto y en la solución adoptada, hablando de generación de energía eólica y sistemas SCADA.

Las bases documentales en las que se ha basado la redacción del TFG han sido los informes previos realizados por el grupo de investigación de la UPV participante en el proyecto PROMOTioN y la información disponible respecto al código de simulación en MATLAB que, como se observará, resultará de gran utilidad a lo largo de este documento.

I.1.3.1 Antecedentes: Breve historia de la energía eólica

La humanidad ha aprovechado la energía del viento desde que existen registros, convirtiendo así a la energía eólica en una de las fuentes de energía más antiguas que se conocen. Su evolución, a pesar de ser algo lenta se ha desarrollado de forma continuada a lo largo de los años hasta su explosión en el último siglo.

Para la realización de este trabajo se han consultado diversos libros acerca de la historia de esta apasionante obra ingenieril, redactando a su vez un breve resumen que finalmente ha sido imposible incluir debido a la estipulación de una extensión máxima de la memoria.

A pesar de ello, se ha decidido incluirlo en los anexos para que el lector pueda disfrutar de ella si lo desea.

I.1.3.2 Antecedentes: Evolución de los sistema SCADA

Antes de la introducción en el mercado laboral de los sistemas SCADA, las empresas dependían exclusivamente de la mano de obra para monitorear y controlar su maquinaria. El aumento de los salarios y la expansión e industrialización hicieron que estos sistemas de trabajo quedaran pronto obsoletos.

En los años 50, las empresas estadounidenses comenzaron a utilizar ordenadores en su cadena de producción. De la mano de los avances en las tecnologías de la información, surgieron lo que hoy en día se conoce como sistemas de control remoto o sistemas SCADA.

A esta vorágine tecnológica ayudó también la descentralización (deslocalización) y la necesidad de seguir controlando la industria desde un lugar alejado. La evolución de los sistemas SCADA transcurrió bastante pareja a la de las redes de comunicación, avanzando de la mano en muchos aspectos debido a su dependencia intrínseca. Las primeras comunicaciones, por lo tanto, se realizaron en el interior de industrias mediante cables Ethernet; gracias a la evolución de la tecnología, estos cables se sustituirían décadas después por protocolos de conexión mas sofisticados como el UDP. Los expertos dividen la historia de los sistemas SCADA en tres etapas evolutivas:

I.1.3.2.1 Sistemas SCADA monolíticos

Los sistemas SCADA eran sistemas independientes de cualquier otro, sin ningún tipo de conectividad, con protocolos de conexión muy ajustados que no admitían ninguna funcionalidad más allá que la de control.

I.1.3.2.2 Sistemas SCADA distribuido

En esta etapa se empezaría a utilizar la tecnología de redes de área local (LAN), para distribuir el procesamiento a través de múltiples sistemas. Esta distribución, proporcionaría más potencia de procesamiento para el conjunto del sistema comparada con la de la anterior etapa.

Esta mejora en la potencia fue acompañada, a su vez, de una mejora en la fiabilidad ya que, si una estación HDMI fallaba, se podría utilizar otra paralela.

I.1.3.2.3 Sistema SCADA en red

El auge de Internet ha permitido el traslado de estos sistemas a la nube. Los estándares abiertos eliminan algunas de las limitaciones de las versiones anteriores.

Además, es durante esta “generación” cuando empezaron a originarse los primeros negocios de desarrollo de hardware y software en relación con el monitoreo.

La mejora más importante de los sistemas SCADA en esta última generación es el empleo de protocolos WAN que permite la conexión a grandes distancias, una gran capacidad de supervivencia ante “desastres” y mejoras en la confiabilidad general.

Todos los esquemas de estas etapas se pueden observar en la figura I.1.3.

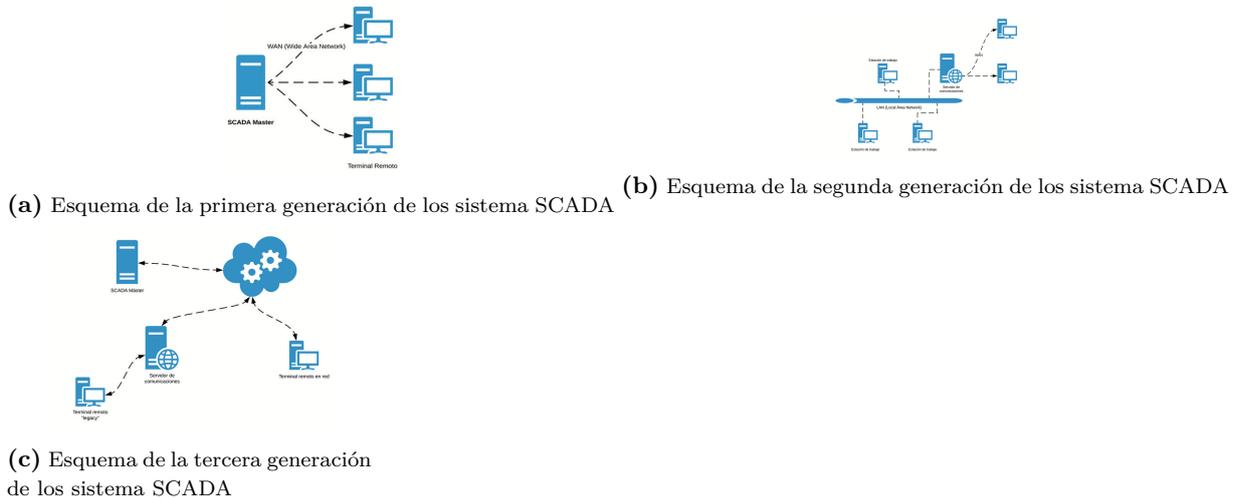


Figura I.1.3: Evolución de los sistemas SCADA (infografía)

I.1.4 Justificación

I.1.4.1 *Personal*

Para este primer apartado, abandonaré el estilo impersonal característico de este tipo de escritos durante unas breves líneas.

Unos de mis primeros recuerdos al pensar en mi niñez, es hablar con mi padre sobre qué eran aquellos extraños "paneles solares" que vendían a la gente, y cómo eran éstos capaces de encender la bombilla de la habitación de sus dueños.

Sería pocos años después, cuando el pequeño Julio decidió sustituir definitivamente sus populares y caros "Action Man" por aquellos juguetes curiosos que traía su padre de cada uno de sus viajes.

Mi ya, por aquel entonces, estrecho vínculo con la tecnología se reforzaría con el paso de los años. Recuerdo con mucho cariño a mis maestros de tecnología industrial en el instituto que fueron capaces de transmitirme su amor por esta profesión.

Es posible, que hoy en día, sumidos en esta vorágine de consumismo y globalización, nos hayamos olvidado del verdadero significado de la palabra ingeniería, o simplemente, peque de romanticismo ingenuo. Pero fueron los motivos que me llevaron a estudiar esta carrera y permítanme que los defienda.

Personalmente creo que este proyecto cumple con la finalidad última de todo ingeniero: idear, construir o programar algo, cuyo objetivo final sea mejorar la calidad de vida del consumidor y contribuir al bienestar general.

Todo esto acompañado de mis otras dos pasiones en este mundo científico: la programación y las energías renovables, convierte este proyecto en un auténtico regalo para mí.

I.1.4.2 *Académica*

Colegialmente hablamos de un trabajo sumamente completo, donde se abordan casi en su totalidad todos los aspectos clave del grado en "Ingeniería en Tecnologías Industriales".

En primer lugar, la generación de energía, así como su posterior tratamiento y sus métodos de protección; conceptos estudiados en todas las materias de electricidad y electrónica a lo largo de la carrera. En segundo lugar, el tratamiento de señales ondulatorias, donde ha sido imprescindible recurrir a los fundamentos físicos adquiridos en los primeros años del grado.

Por otro lado, este proyecto lleva innatos muchos conceptos de programación, en este caso se ha intentado ampliar los conocimientos de las asignaturas de *Informática e Informática Industrial* con otros conocimientos de forma totalmente autodidacta.

Por último, pero no por eso menos importante, se ha requerido en gran medida de la asignatura de *Proyectos* a la hora de realizar la memoria técnica, los presupuestos y el pliego de condiciones.

I.1.4.3 Industrial

Como se ha analizado anteriormente, el papel de la energía eólica resulta imprescindible para el correcto desarrollo energético mundial. En este sentido, se hace fundamental explotar al máximo los recursos eólicos de la zona, lo que muchas veces hace inevitable tener que instalar los parques en zonas marinas a unos cuantos kilómetros de la costa.

En este sentido, es ineludible mantener un control correcto y fluido respecto a todas y cada una de las variables del parque, con el objetivo de minimizar al máximo los desplazamientos a la zona debido a la dificultad técnica y el coste económico que esto conllevaría.

Este trabajo pretende dar solución a este problema, proporcionando una interfaz sencilla y completa sobre el estado del parque y permitiendo al usuario llevar el control en todo momento de este.

I.1.5 Motivación

La motivación principal para la realización del presente trabajo de final de grado es la consolidación de los conocimientos y técnicas adquiridas a lo largo de la titulación de Ingeniería en Tecnologías Industriales, impartida en la Universidad Politècnica de València.

La ocasión de compaginar un caso real, como el que ofrece este proyecto, con los conceptos aprendidos durante estos cuatro años de carrera, resulta esencial para el correcto desarrollo de un ingeniero y su integración en el mercado profesional.

Se trata una oportunidad perfecta para ponderar, valorar, consolidar y recordar muchos conocimientos adquiridos durante este tiempo y construir de esta manera, y después de muchos años de estudio, el primer puente real entre el mundo estudiantil y el laboral.

I.1.6 Normativa aplicada

La correcta realización de todo proyecto conlleva el conocimiento y la aplicación de la normativa vigente al mismo. El presente trabajo de fin de grado ha sido realizado tanto en el departamento de ingeniería eléctrica como en el Instituto de Automática e Informática industrial de la UPV (ai2). Los documentos que aparecen a continuación, emitidos por entidades estatales, estipulan las normativas aplicables:

Ley 31/1995, de 8 de noviembre, de prevención de riesgos laborales. La presente ley promueve la seguridad y la salud de los trabajadores mediante la aplicación de medidas de prevención para los riesgos derivados del trabajo.

- Publicado en: «BOE» núm. 269, de 10 de noviembre de 1995, páginas 32590 a 32611 (22 págs.)
- Departamento: Jefatura del Estado
- Referencia: BOE-A-1995-24292

Directiva 2009/28/CE, de 23 de abril, relativa al fomento del uso de energía procedente de fuentes renovables, establece que cada Estado miembro elaborará un Plan de Acción Nacional en

materia de Energías Renovables (PANER) para conseguir los objetivos nacionales fijados en la propia Directiva.

- Publicado en: Diario Oficial de la Unión Europea.
- Departamento: el Parlamento Europeo y el Consejo de la Unión Europea.

Directiva 2009/147/CE, de 30 de noviembre de 2009 relativa a la conservación de las aves silvestres.

- Publicado en: Diario Oficial de la Unión Europea.
- Departamento: el Parlamento Europeo y el Consejo de la Unión Europea.

Real Decreto 2177/2004, de 12 de noviembre, por el que se modifica el Real Decreto 1215/1997, de 18 de julio, por el que se establecen las disposiciones mínimas de seguridad y salud para la utilización por los trabajadores de los equipos de trabajo, en materia de trabajos temporales en altura.

- Publicado en: «BOE» núm. 274, de 13 de noviembre de 2004, páginas 37486 a 37489 (4 págs.)
- Departamento: Ministerio de la Presidencia
- Referencia: BOE-A-2004-19311

Real Decreto 1614/2010, de 7 de diciembre, por el que se regulan y modifican determinados aspectos relativos a la actividad de producción de energía eléctrica a partir de tecnologías solar termoeléctrica y eólica.

- Publicado en: «BOE» núm. 298, de 8 de diciembre de 2010, páginas 101853 a 101859 (7 págs.)
- Departamento: Ministerio de Industria, Turismo y Comercio
- Referencia: BOE-A-2010-18915

Real Decreto 947/2015, de 16 de octubre, por el que se establece una convocatoria para el otorgamiento del régimen retributivo específico a nuevas instalaciones de producción de energía eléctrica a partir de biomasa en el sistema eléctrico peninsular y para instalaciones de tecnología eólica.

- Publicado en: «BOE» núm. 249, de 17 de octubre de 2015, páginas 97340 a 97342 (3 págs.)
- Departamento: Ministerio de Industria, Energía y Turismo
- Referencia: BOE-A-2015-11200

Ley 37/2003, de 17 de noviembre, del Ruido.

- Publicado en: «BOE» núm. 276, de 18 de noviembre de 2003, páginas 40494 a 40505 (12 págs.)

- Departamento: Jefatura del Estado
- Referencia: BOE-A-2003-20976

Ley 4/1989, de 27 de marzo, de Conservación de los Espacios Naturales y de la Flora y Fauna Silvestres.

- Publicado en: «BOE» núm. 74, de 28 de marzo de 1989, páginas 8262 a 8269 (8 págs.)
- Departamento: Jefatura del Estado
- Referencia: BOE-A-1989-6881

Ley 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.

- Publicado en: «BOE» núm. 294, de 06/12/2018.
- Departamento: Jefatura del Estado
- Referencia: BOE-A-2018-16673

ISO/IEC 25000, System and Software Quality Requirements and Evaluation.

I.1.7 El proyecto PROMOTioN

PROMOTioN gira en torno al establecimiento de una red integrada que conecte eficiente y eficazmente estaciones eólicas marítimas en el mar del Norte entre sí y con el continente a través de líneas de corriente continua de alta tensión (HVDC) para contribuir a la evolución del mercado eléctrico europeo hacia fuentes de obtención de energía limpias y sostenibles.

El plan consta de los siguientes pasos intermedios y componentes:

- La planificación y el diseño de las infraestructuras
- Los acuerdos entre operadores y fabricantes/productores
- La consolidación de manifestaciones tecnológicas HVDC, a gran escala y rentables: las principales son la instauración e inicio de actividad en los sistemas de protección y control de las redes apoyados por múltiples proveedores, las pruebas de conmutadores con aislamiento de gas sostenidas a largo plazo y la experimentación con interruptores de circuito a plena potencia
- El desarrollo de los elementos innovadores que conformarán la red
- La creación de un marco financiero que posibilite su implantación
- Y por último, pero de gran relevancia para el asentamiento de la idea en el mundo real, la formulación de regulaciones para sentar las bases legales del proyecto, complejas debido a la cooperación multilateral del sector público y privado de varios países europeos.

I.1.8 Objetivos de PROMOTioN

Para completar exitosamente este proyecto, se deben salvar varias barreras y desafíos, ante los altos costes de este tipo de infraestructuras, la insuficiente experiencia con su conservación, monitorización y vigilancia y la problemática legislativa a la hora de instalar parques eólicos marinos emplazados entre fronteras: a nivel tecnológico, hace falta conseguir que sus componentes esenciales resulten rentables y garanticen fiabilidad, al mismo tiempo que se asegura el adecuado funcionamiento de las conexiones HVDC. Esto se sostendrá sobre los pertinentes marcos financiero, legal para su puesta en marcha y acuerdos de mercado para vincular a fabricantes, promotores y usuarios de la red.

Con estos prerrequisitos asentados, los objetivos pretenden orientar al diverso conjunto de participantes hacia metas comunes.

- De nuevo, se parte de la base tecnológica, construyendo a gran escala sistemas de control y protección generales, modelos de convertidores de energía eólica efectivos, conmutadores HVDC con aislamiento de gas y protecciones.
- Desde allí, se define una hoja de ruta de despliegue e implementación progresiva hasta 2050, a la que se suman propuestas concretas para insertar este trabajo en el marco legal comunitario.
- A través de estas líneas de acción, se pretende llevar a cabo controles, protección y gestión de la red interoperables y seguros; trabajando siempre hacia la estandarización y la interoperabilidad tecnológica para un uso accesible y sencillo para todos los países y agentes participantes y beneficiarios.

I.1.9 Socios

Para la consecución de lo que se plantea en el punto anterior, trabajan conjuntamente 33 colaboradores de 11 países, desde puntos de vista de investigación hasta la implantación final de los nuevos parques eólicos y las redes que los interconectarán.

Participan fabricantes de aerogeneradores, convertidores y sistemas de protección, operadores de redes eléctricas, universidades y entidades de certificación: desde los grupos suecos privados ABB y SCiBreak y la empresa estatal Svenska Kraftnät hasta la corporación pública irlandesa Eirgrid, pasando por Mitsubishi Electric Europe emplazada en Países Bajos, Grid Solutions en Reino Unido, el operador francés RTE, Tennet B.V desde Países Bajos y Alemania, Siemens AG desde Alemania, T&D Europe desde Bélgica, el grupo italiano Prysmian, la danesa Energinet y la escocesa Scottish Hydro Electric Transmission plc; 8 países están involucrados solamente en la vertiente eléctrica y energética del proyecto.

Por supuesto, dada la necesaria combinación con los aerogeneradores, toman parte también la compañía energética noruega Equinor, Orsted Windpower y la joint venture MHI Vestas Offshore Wind A/S desde Dinamarca.

A la hora de compaginar y conducir esta labor coordinadamente, agencias asesoras guían el trabajo cooperativo técnico: la neerlandesa DNV GL, la Deutsche Windguard y la fundación Offshore-Windenergie alemanas, la asociación inglesa The Carbon Trust y la asesoría belga TRACTEBEL. Mientras, prestigiosas universidades y centros de investigación especializados como la KU Leuven

en Bélgica, el Instituto Real de Tecnología KTH en Suecia, el Supergrid Instituto francés, las Universidades de Aberdeen y Strathclyde en Reino Unido, TU Delft y la Universidad de Groningen en Países Bajos, la Universidad Técnica danesa, la Universidad RWTH y la FGH alemanas, la Escuela de Regulación de Florencia (Italia) y la Universitat Politècnica de València aportan el background académico necesario.

I.1.10 Organización del proyecto

En la figura I.1.4 se pueden observar todos los paquetes de trabajo en los que se divide el proyecto; concretamente, la UPV se centra en los bloques de trabajo WP3 y WP16:

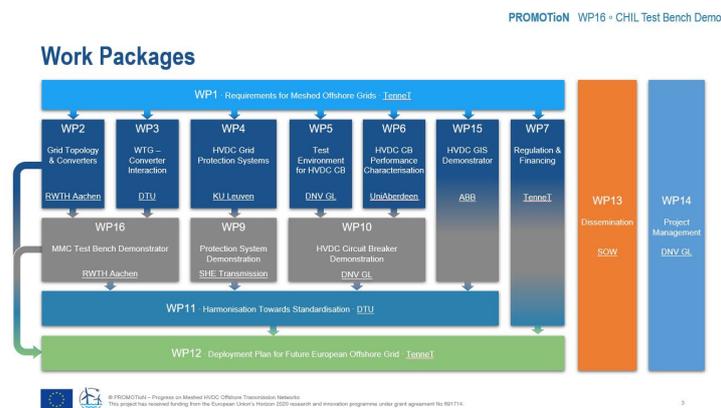


Figura I.1.4: Lugar que ocupa este trabajo dentro del proyecto general

WP3 - Self-start and black-start operation:

Lo que se pretende aquí es utilizar convertidores de la red eléctrica con capacidad de arranque automático, sin necesidad de modificaciones en el hardware de las turbinas eólicas. Para ello, se tienen en cuenta 3 escenarios posibles, todos ellos black-start y alto voltaje. El primero se trata de corriente alterna conectada con WPP, el segundo emplea corriente continua e incorpora un transformador marino y el tercero no cuenta con este paso intermedio.

Estos ensayos requieren el cumplimiento de ciertos requerimientos establecidos en el proyecto y se han aplicado a un PLC de Vestas, con resultados satisfactorios. Se fijaron los siguientes 4 objetivos a alcanzar:

- La definición de requisitos funcionales de los parques eólicos offshore
- El desarrollo casos de estudio y algoritmos de control
- La evaluación del código
- La publicación de resultados

Finalmente, se ha certificado su funcionamiento trabajando de forma normal, en casos de faltas o errores y con servicios auxiliares. Gracias a esto, se allana el camino para validar CHiL en tiempo real, eje vertebrador del bloque WP16, que validará el diseño ejecutado en el WP3.

En este subapartado, han participado 4 universidades (la Universidad Técnica de Dinamarca, la de Strathclyde, la de Aachen y la FGH) en la vertiente de investigación, con los equipos

de fabricantes de conexiones HVDC como Energinet, ABB o Siemens y empresas puramente dedicadas a manufacturar aerogeneradores: MHI Vestas Offshore Wind, Orsted y Equinor; todos ellos coordinados mediante la asesoría de DNV-GL.

WP16 - MMC Test Bench Demonstrator:

El objetivo de este paquete es validar (demostrar) el control y la operación/operatividad de redes eléctricas HVDC empleando PHiL y CHiL mediante simuladores en tiempo real para incrementar la confianza en la viabilidad de MTDC.

Trabajan conjuntamente la RWTH de Aachen, la Universidad Técnica de Dinamarca y el Instituto Real de Tecnología de Estocolmo bajo la coordinación de DNV-GL y Dong Energy; que completan el resto de tareas asignadas.

El procedimiento se abre definiendo y especificando casos de prueba para el nuevo banco de ensayo (16.1), preparando medidas para instalar el sistema MMC (16.2) y configurándolo eléctrica y mecánicamente (16.3). Es entonces cuando comienza la simulación de tests básicos (16.4), la implementación de un método analítico para examinar fenómenos de resonancia armónica entre varios convertidores (16.5) y la demostración de de casos de prueba definidos en cuanto a su interoperabilidad, esquemas de control y protección (16.6). Por último, se conduce un estudio del impacto sobre los requerimientos para redes malladas HVDC offshore (16.7).

La Universitat Politècnica realiza el primer punto como preparativo, pero su aportación comienza con el punto 16.5; sobresaliendo su trabajo en el 6.

MS 79 - Real time CHIL set up: En esta subtarea, la UPV se encarga de construir un prototipo demostrador RT para el CHiL de una turbina eólica, aportando una descripción técnica completa del sistema, ya preparado para su implementación y ensayo.

I.1.11 Epítome del contenido del documento

La memoria del presente documento se iniciará con una primera toma de contacto con los sistemas de monitorización; se definirá el concepto de sistema SCADA y se analizarán sus utilidades a nivel industrial, para posteriormente realizar un análisis de las posibles alternativas existentes.

Este apartado proseguirá con un profundo estudio de los requisitos del proyecto, seguido de un análisis acerca de la simulación de las señales y de la programación y configuración del software seleccionado.

Todo lo anterior permitirá comprender a la perfección el siguiente punto y columna vertebral de la memoria, la solución adoptada.

Para finalizar, se realizará una verificación y validación de dicha solución, se redactarán los resultados y conclusiones y se concluirá con un apartado de visión futura y ampliaciones donde se introducirán algunas de las ideas aún no adoptadas en el software.

Indicar que las fuentes de las figuras utilizadas son múltiples y variadas; en el caso de no indicar su procedencia significará que son de creación propia, dicha afirmación se cumplirá a lo largo de todo el documento.

Capítulo I.2

Sistema SCADA: Análisis de alternativas

La finalidad de este capítulo es familiarizar al lector con el concepto de sistema SCADA (Supervisory Control And Data Acquisition) y sus utilidades.

Se comenzará con una primera toma de contacto y una descripción acerca de este tipo de sistemas para, posteriormente, proseguir con algunos ejemplos de su empleo a gran escala en industrias punteras alrededor de todo el mundo.

Para finalizar con el capítulo se realizará un estudio de alternativas en la que diferentes programas se compararán con la finalidad de obtener el más adecuado para este proyecto.

Índice

I.2.1 ¿Qué es un sistema SCADA?	19
I.2.2 Ejemplos de sistemas SCADA en la industria	20
I.2.3 Análisis de alternativas	21

I.2.1 ¿Qué es un sistema SCADA?

El sistema SCADA o Sistema de Supervisión, Control y Adquisición de Datos como es conocido en castellano, es un sistema que combina elementos de hardware y software permitiendo entre otros aspectos: controlar procesos industriales, monitorear, recopilar información, generar bases de datos independientes e interactuar con actuadores.

Este tipo de sistemas son cruciales en muchas organizaciones para mantener la eficiencia y optimizar procesos.

Los sistemas SCADA están diseñados para recopilar datos, transferirlos a un ordenador central, almacenarlo y proporcionar al usuario de una interfaz. Su línea de flujo empieza en los controladores y sensores lógicos programables (PCL), dichos sensores permiten controlar variables industriales como: velocidad, temperatura o intensidad.

La información generada por estos controladores es posteriormente enrutada hacia una computadora central con el software SCADA el cual procesa los datos siguiendo el código creado por el programador figura I.2.1.

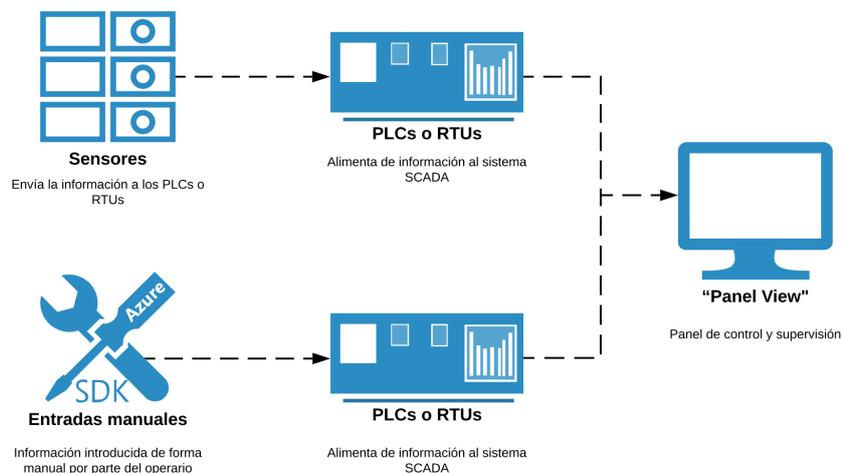


Figura I.2.1: Diagrama de flujo de un sistema SCADA habitual

Las ventajas de este tipo de sistemas para la industria y la investigación científica son prácticamente innumerables; algunas de ellas son:

- La capacidad del sistema SCADA para operar en situaciones críticas.
- La posibilidad de generar alarmas y archivar datos que de otra forma sería imposible.
- El potencial para poder ofrecer a los operadores una interfaz clara y sencilla.
- La retroalimentación en tiempo real.

- Un ahorro económico significativo.
- Su escalabilidad.

Estas dos últimas razones son las que probablemente más pesen a la hora de decantarse por la utilización de estos esquemas; además proporciona otras ventajas en segundo plano como pueden ser un aumento de la productividad, un incremento de la calidad y seguridad y una disminución de los tiempos de producción.

I.2.2 Ejemplos de sistemas SCADA en la industria

SpaceX y el famoso "abort"

SpaceX es una empresa estadounidense de transporte aeroespacial fundada en 2002 por Elon Musk el multimillonario CEO de empresas como PayPal o Tesla.

En su particular "carrera espacial", SpaceX requería de un sistema que les permitiera a sus ingenieros monitorizar todas las variables de sus aeronaves en tiempo real desde la sala de control situada en estados unidos.

Durante el proceso de ingeniería, cuando la programación estaba a punto de finalizar, comenzó a circular por redes sociales la foto de la figura I.2.2; en dicha infografía se puede identificar con claridad el popular símbolo de "abort" del programa SCADA "LabView (el mismo que se utilizará a lo largo de este proyecto) en la esquina superior izquierda del "front panel"; un detalle que paso desapercibido por parte de los ingenieros de la empresa americana.



Figura I.2.2: Front view de SpaceX. Güncelleme 03/07/2020

Uso de sistemas de Foxboro en Kimberly Clark Costa Rica, planta Belén

Kimberly Clark es una empresa dedicada a la elaboración de productos reciclados a base de papel. Los procesos de obtención del producto final son complejos lo que llevó a la empresa a automatizar y monitorizar los procesos.

Para la realización del proyecto costarricense se utilizó un sistema SCADA que permite a los operadores de la planta tener un control completo de todos los procesos que se están realizando.

Algunos otros ámbitos de aplicación son: control de señales de tráfico, domótica o producción eléctrica.

I.2.3 Análisis de alternativas

El reciente auge en el ámbito industrial de los sistemas SCADA han provocado la aparición de numerosas opciones de software, mejorando la competitividad entre ellas y colaborando a la mejora continua del conjunto.

A pesar de haber tantas opciones, es común que, a la hora de realizar un proyecto complejo, el programador utilice más de una; en este sentido han trabajado muchas compañías proveedoras de software en los últimos años con el objetivo de mejorar la simbiosis entre sus respectivas plataformas.

Para la realización de este proyecto se han tomado como posibles alternativas: MatLAB de Matworks, Phyton de Python Software Foundation, C++, Ignition de Inductive Automation y LabVIEW de National Instruments; se han estudiado y comparado minuciosamente y finalmente se han utilizado la primer y la última de ellas.

Para efectuar el estudio de alternativas se analizarán diferentes aspectos: coste, facilidad de programación, nivel de profundidad del lenguaje y capacidad para crear una interfaz usuario atractiva.

En cuanto al coste, se pueden separar los programas anteriores en dos grupos: los "freeware" o gratuitos y los de pago. En el primer cuadrante se puede encontrar a Phyton y C++ mientras que en el segundo están MatLAB, inductive automation ignition y LabVIEW. La licencia anual de programas como LabVIEW puede oscilar entre los 400€y los 3000€, anuales dependiendo del paquete. Dicho conveniente económico no supone, por fortuna, un agravio a la hora de realizar este proyecto puesto que se cuenta con la licencia gratuita por parte de la Universitat Politècnica de València.

A pesar del elevado coste, los programas de pago y más en concreto LabVIEW ofrece una facilidad de programación increíble, alejándose del tradicional código de programación lineal y apostando por los recursos gráficos. Este aspecto ayuda mucho ya que permite tener una curva de aprendizaje rápida para programar tareas básicas pero comunes y muy potentes.

La experiencia de C++ y MATLAB han permitido que con ellos se pueda realizar casi cualquier cosa utilizando pocos recursos de computación. Por su parte LabVIEW es capaz de conectarse a la mayoría de los dispositivos presentes en el mercado.

En relación al último aspecto, todos tienen potencial para generar interfaces de usuario agradables, aunque el tiempo de programación se reduce bastante si se utiliza LabVIEW en lugar de C++ por ejemplo.

Como se ha dicho anteriormente, para la realización de este proyecto se han utilizado tanto MATLAB como LabVIEW; el primero de ellos puesto que era el software de nacimiento del simulador con el que se contaba al inicio de este trabajo y el segundo porque permitía una comunicación casi perfecta con el simulador además de ser mucho más intuitivo y visual tanto para el programador como para el usuario.

<i>Tabla comparativa de las diferentes tecnologías</i>					
<i>Características</i>	<i>LabVIEW</i>	<i>C++</i>	<i>MATLAB</i>	<i>Ignition</i>	<i>Python</i>
<i>Precio de la licencia</i>		<i>Gratuita</i>			<i>Gratuita</i>
<i>Facilidad de programación</i>	<i>Muy alta</i>	<i>Muy baja</i>	<i>Media</i>	<i>Media</i>	<i>Media</i>
<i>Profundidad del lenguaje</i>	<i>Elevada</i>	<i>Muy elevada</i>	<i>Muy elevada</i>	<i>Media - Baja</i>	<i>Media</i>
<i>Resultado atractivo</i>	<i>Si</i>	<i>Si (pero muchas horas)</i>	<i>Si (pero muchas horas)</i>	<i>Si</i>	<i>Si</i>

Tabla I.2.1: Tabla comparativa de los sistema SCADA posibles para este proyecto

Al inicio de la programación se utilizó durante un tiempo inductive automation ignition pero los problemas originados y la posibilidad de contar con el estupendo curso ofrecido por Guillermo Prados (Partner and Manager en MeasureIT Systems SL) inclinó la balanza hacia LabVIEW. Por último, se adjunta una tabla comparativa que resume lo anteriormente expuesto.

Capítulo I.3

Simulación de señales

Todo sistema ingenieril está plagado de parámetros controlables gracias, en parte, a la gran variedad de sensores presentes hoy en día en el mercado. En este sentido, se presenta fundamental el procesado, envío y tratamiento de estas señales para su correcta comprensión posteriormente.

A lo largo del presente capítulo se enumerarán los tipos de señales tanto recibidas como enviadas por el programa, se describirán los protocolos de envío de señal utilizados y se analizarán los tratamientos que han requerido.

A su vez, se hará una breve descripción del software empleado, poniendo especial atención a aquellos puntos más relevantes a la hora de la realización del presente trabajo de fin de grado.

Índice

I.3.1 Tipos de señales a simular	25
I.3.2 Software empleado	26
I.3.3 Protocolos de envíos de señal	28
I.3.4 Base de datos	30

I.3.1 Tipos de señales a simular

Los principales parámetros a controlar en un aerogenerador se pueden dividir en dos grandes grupos según su tipología.

Por un lado, se disponen las variables de tipo lógico o binario cuya unidad solo puede tomar dos posibles estados, tradicionalmente etiquetados como 0 y 1 o abierto y cerrado.

En el programa se encontrarán este tipo de datos en casi cualquier "front view⁴" ya que son necesarios para la utilización de las estructuras de casos. Además, se presentarán a la hora de controlar el "breaker" o interruptor automático, debido a que éste solo puede presentar dos posibles estados: abierto y cerrado o lo que es lo mismo, apagado y encendido.

Por otra parte, se tienen las variables de tipo "float" o variables numéricas, que son aquellas que pueden tomar cualquier valor numérico y cuya precisión se establecerá en tres decimales a lo largo de este proyecto. En este grupo se pueden encontrar la mayor parte de las variables de un aerogenerador: tensión, corriente, potencia activa, potencia reactiva, velocidad del viento, etc.

De todas las variables expuestas anteriormente, en el proyecto se trabajará con las cuatro primeras mediante una comunicación bidireccional, es decir, se recibirán datos por parte del simulador o prototipo a escala hacia la computadora donde se encuentra el sistema SCADA y viceversa.

En las siguientes figuras se muestra el flujo de datos a nivel general I.3.1 y a nivel particular I.3.2 y I.3.3; la sustitución del denominado "Real Time Simulator" por el simulador en MATLAB es debido a la imposibilidad de acudir al laboratorio de forma presencial.

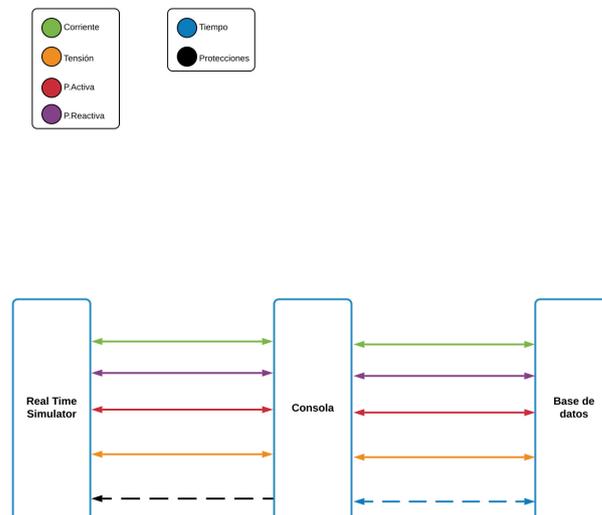


Figura I.3.1: Esquema de comunicación

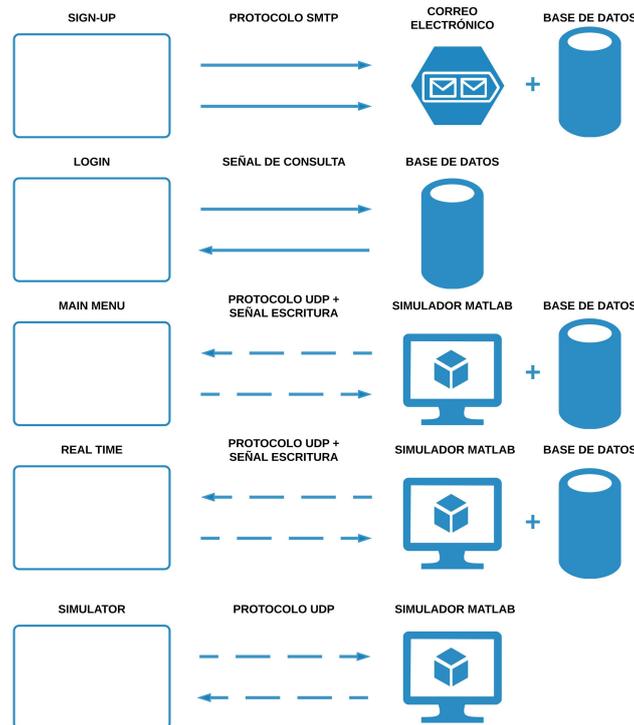


Figura I.3.2: Esquema de comunicación sistema SCADA (parte 1/2)

I.3.2 Software empleado

La crisis sanitaria que ha azotado todo el mundo debida al virus COVID-19 ha impedido que se pudiera trabajar en el laboratorio, siendo necesaria la sustitución del prototipo a escala por un simulador a partir de lenguaje de MATLAB.

MATLAB, según su propia definición, es una plataforma de programación diseñada específicamente para ingenieros y científicos. Esta aplicación permite: analizar datos, desarrollar algoritmos y crear modelos y aplicaciones.

Será esta última característica la que se explotará para la construcción del presente trabajo; esto se realizará a través de simulink, un entorno de programación visual que funciona sobre el entorno de programación de MATLAB.

Para la realización del programa se ha utilizado LabVIEW, "Laboratory Virtual Instruments Engineering and Workbench", es al igual que Simulink, un software de diseño gráfico utilizado para simular, controlar y automatizar procesos industriales realizado por la compañía norteamericana National Instruments.

LabVIEW proporciona una poderosa plataforma para la realización de una amplia variedad de aplicaciones. No solo se puede emplear para el control de equipos industriales o la creación de bases de datos sino también para proyectos médicos o biólogos.

Por último, pero no menos importante, se ha manejado Wireshark. Wireshark es un analizador de protocolos utilizado para realizar análisis y solucionar problemas de comunicación.

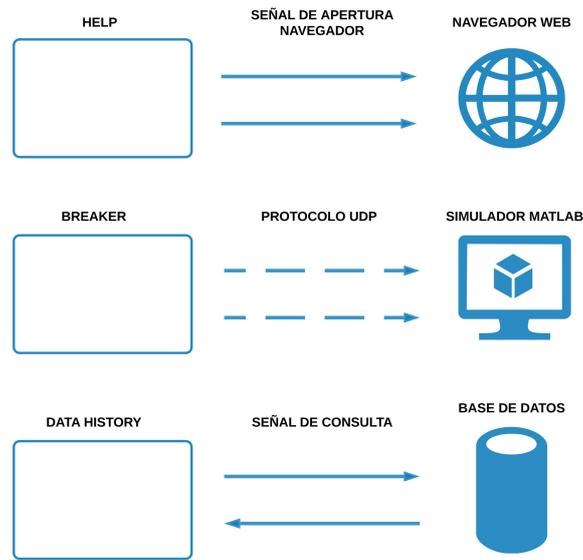


Figura I.3.3: Esquema de comunicación sistema SCADA (parte 2/2)

A lo largo de este proyecto se ha utilizado para garantizar y comprobar las conexiones vía UDP entre dos computadoras: la computadora "host" del sistema SCADA y la computadora que realizaba la función de simulador.

En las figuras I.3.4, I.3.5 y I.3.6, situada bajo, se puede encontrar una captura de pantalla del entorno visual de estos tres programas mencionados.

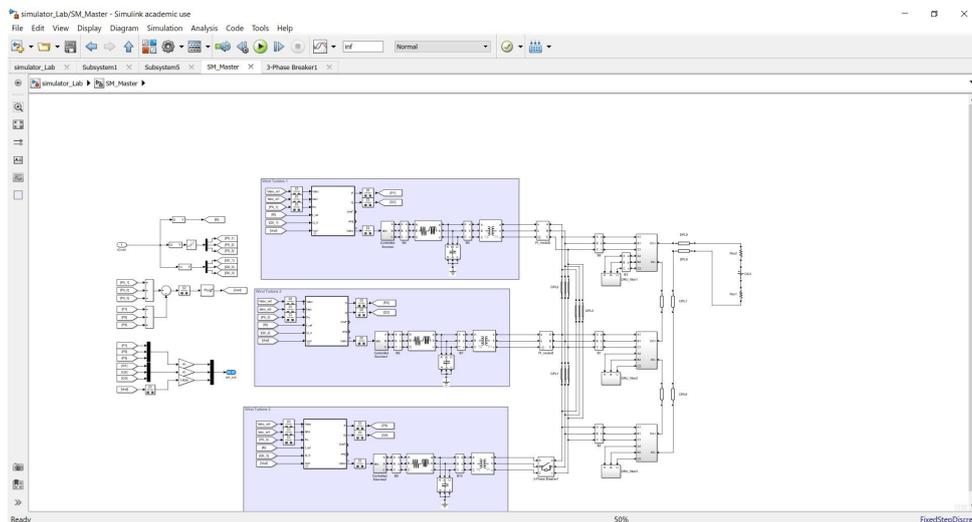


Figura I.3.4: Captura entorno de programación de simulink (MATLAB)

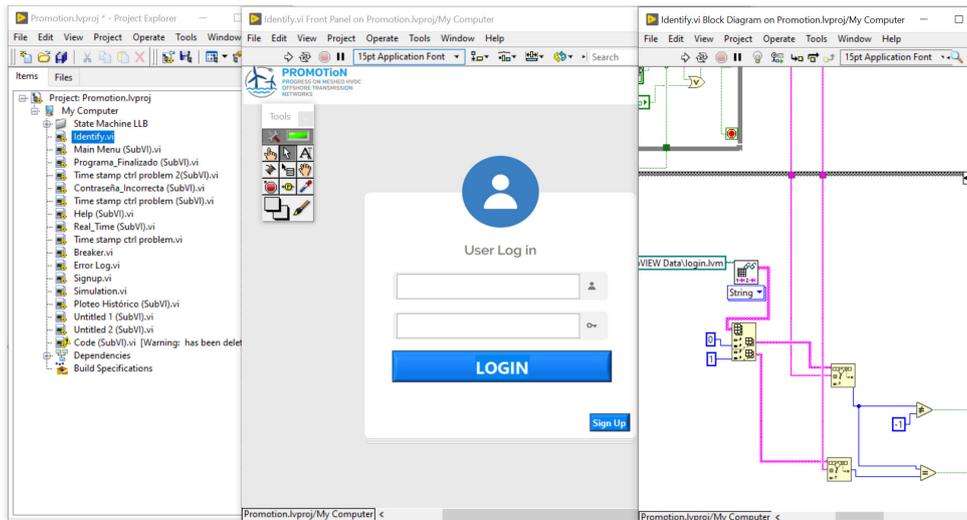


Figura I.3.5: Captura entorno de programación de LabVIEW

I.3.3 Protocolos de envíos de señal

El envío de señales entre el prototipo a escala y la computadora central o, en su defecto, entre ambos computadores es, probablemente, la piedra angular del presente proyecto.

Gran parte del éxito de este trabajo recae en haber conseguido realizar un envío estable, veloz, fiable y sólido, ya que, en el caso de fallar dicha comunicación, el programa carecería de utilidad alguna.

Desde el inicio del proyecto, se han contado con varias alternativas para el protocolo de envío de señales: la utilización de un servidor OPC y el protocolo TCP y el finalmente utilizado protocolo UDP.

OPC es una tecnología de comunicación con una arquitectura de cliente y servidor. Una aplicación actúa de servidor (la computadora que ejecuta el código de MATLAB) proporcionando datos y otra actúa como cliente leyéndolos o manipulándolos (el ordenador con el sistema SCADA).

El protocolo TCP es un estándar que define cómo establecer y mantener una comunicación de red a través de la cual los programas de aplicación pueden intercambiar datos entre sí.

Por su parte, el protocolo UDP es un sistema de envío para aplicaciones especialmente sensibles al tiempo. Acelera las comunicaciones al no requerir de un "handshake", evitando la necesidad de aceptar la comunicación entre ambos equipos.

El protocolo UDP no tiene la capacidad de verificación de errores de la que si dispone el protocolo TCP y se utiliza mejor cuando no es necesaria la verificación de errores y la velocidad es un requerimiento importante.

En cuanto a su empleo, el protocolo UDP es utilizado comúnmente en comunicaciones urgentes dónde se prefiere descartar un paquete de datos antes que esperar.

En la figura I.3.7 se puede observar los protocolos de envío TCP y UDP. El protocolo de envío TCP requiere de un handshake entre ambas computadoras, es decir, necesita la confirmación del receptor para que el emisor pueda transmitir datos; para ello utiliza el SYN⁵ y el ACK⁶.

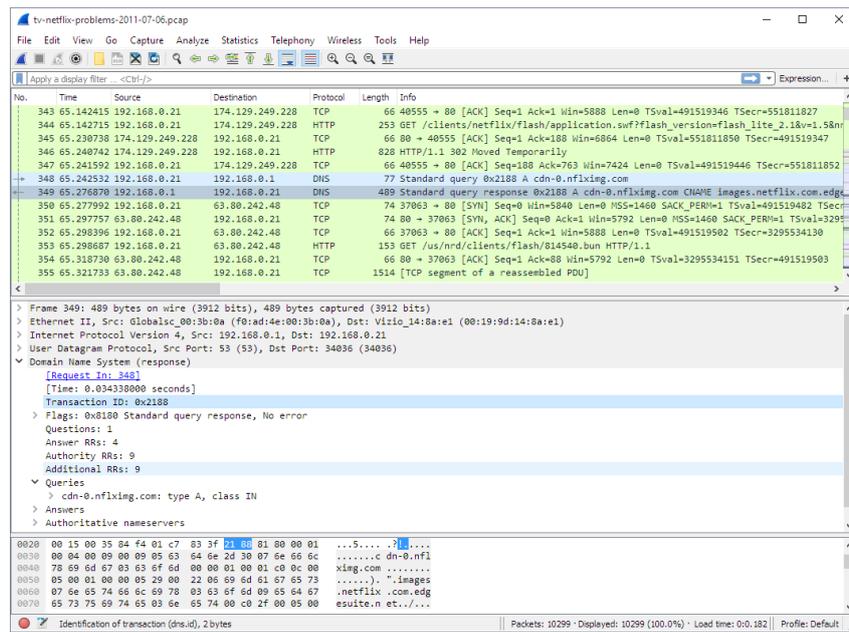


Figura I.3.6: Captura entorno de Wireshark

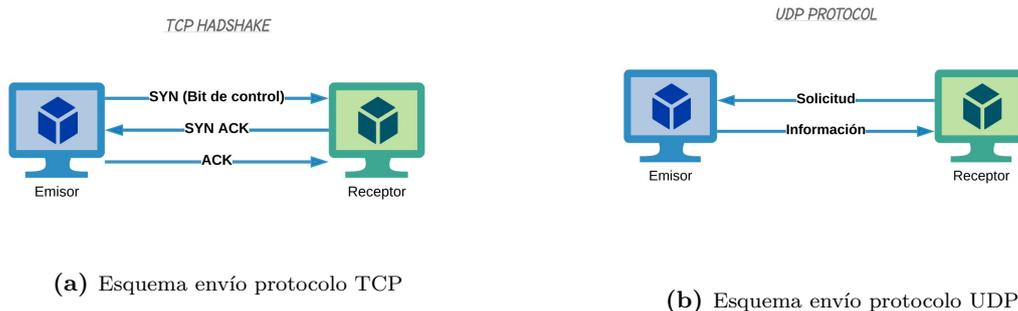


Figura I.3.7: Esquema de envío protocolos TCP y UDP

Por otro lado, en el protocolo UDP solo es necesaria la solicitud por parte del receptor para comenzar a enviar información, incrementando así la velocidad de la comunicación.

En el programa de Simulink este envío se realizará siguiendo el esquema de la figura I.3.8 ; es decir, primero se aplicará una ganancia para no perder precisión, posteriormente se realizará una codificación ASCII para transformar el valor numérico en un dato de tipo "string" o texto y posteriormente se realizará el envío.

Por su parte en LabVIEW se realizará una apertura del recibo vía protocolo UDP, en esta apertura se seleccionará el puerto a utilizar (figura I.3.10a); posteriormente, se determinará el tamaño mínimo y el tamaño máximo del mensaje que se espera recibir.

El bloque de la figura I.3.10b es el encargado de recibir el mensaje, si dicho proceso se realiza con éxito, devolverá una dirección IP y el mensaje en formato "string", si, por el contrario, no se puede realizar dicha acción, devolverá un error.

Por último, el bloque de la figura I.3.10c es el encargado de finalizar esta conexión vía UDP y notificar los errores en caso de que hubiese.

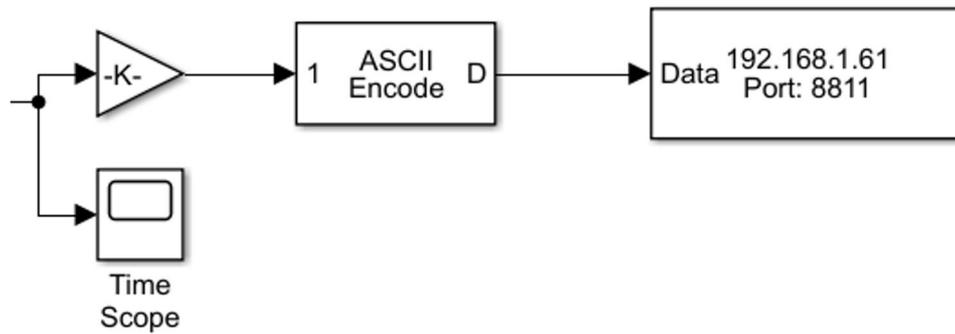


Figura I.3.8: Recorte envío Simulink - LabVIEW vía UDP

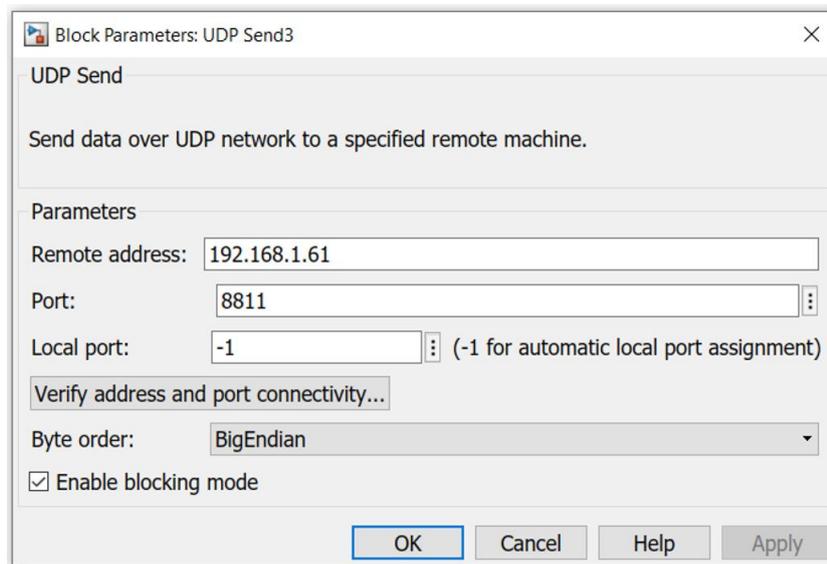


Figura I.3.9: Parámetros del envío UDP

I.3.4 Base de datos

La correcta administración de las bases de datos es un elemento fundamental en todo proyecto de tipo SCADA debido, entre otros aspectos, al gran volumen de datos con los que se trabaja en este tipo de proyectos.

En este trabajo se ha apostado por simplificarlo al máximo utilizando documentos de texto ".lvm" y archivos Excel ya que son los que se leen con mayor velocidad por parte de LabVIEW, además su extendida utilización permite una fácil simbiosis con muchos programas de gestión de datos figura I.3.11.

A pesar de esto, y como se propondrá en el capítulo de "Visión futura y posibles ampliaciones", se podría sustituir el sistema empleado por una base de datos creada con el propio código de LabVIEW o mediante una base de datos de tipo SQL.

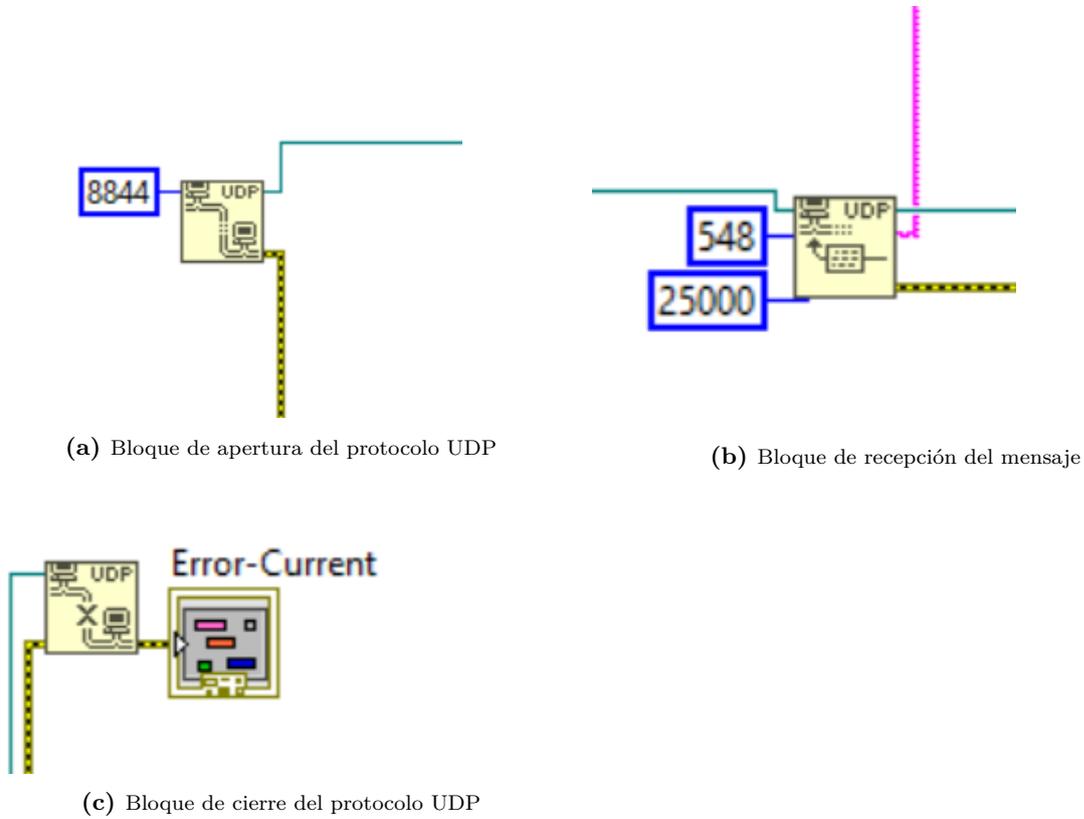
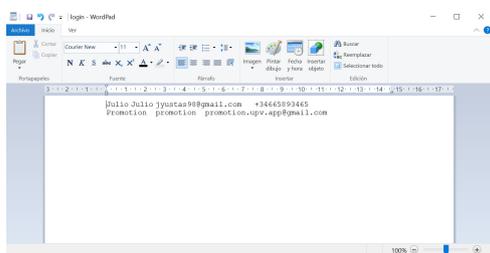
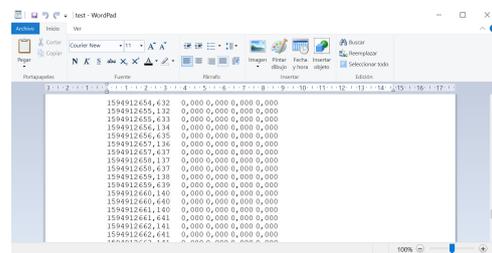


Figura I.3.10: Protocolo UDP en LabVIEW



(a) Base de datos del registro de credenciales



(b) Base de datos de la información recogida del simulador

Figura I.3.11: Bases de datos

Capítulo I.4

Configuración y programación

En el presente capítulo se iniciará la descripción en profundidad de la fase experimental del proyecto que se extenderá hasta el estudio económico y los anexos.

A lo largo de estos puntos se podrá encontrar toda la información relacionada con la configuración física y digital del simulador a utilizar, además de las consideraciones previas antes de programar, la estructura (columna vertebral) del programa, los elementos de control y medida y los elementos adicionales.

Índice

I.4.1 Configuración física	35
I.4.2 Configuración digital	35
I.4.3 Requisitos del sistema	38
I.4.4 Estructura del programa	39
I.4.5 Control y medida	40
I.4.6 Elementos adicionales	41

I.4.1 Configuración física

La planificación inicial del proyecto contaba con un paso más allá del simulador digital del parque eólico realizado con MATLAB; dicho paso consistía en probar el programa en un prototipo a escala situado en el ai2 dentro de la ciudad de la innovación en la Universitat Politècnica de València (España).

A pesar de mantener dicho objetivo hasta el último momento, al final ha resultado imposible realizar esta prueba debido a la crisis sanitaria mundial sufrida durante el primer cuatrimestre del año 2020.

Este hecho ha provocado que el sistema físico se redujera a dos computadoras enviándose datos entre sí, dicho aspecto se explicará con mayor profundidad en el siguiente apartado.

I.4.2 Configuración digital

Como se ha descrito anteriormente, para la ejecución del presente trabajo de fin de grado se ha requerido fundamentalmente de tres tecnologías: MATLAB (con su extensión Simulink), LabVIEW y Wireshark.

Todas ellas han sido explotadas bajo Windows 10, uno de los sistemas operativos desarrollados por Microsoft.

Se diferenciarán dos computadoras diferentes, que a partir de este momento se conocerán como ordenador A o simulador y ordenador B o SCADA, utilizando cada uno de ellos un programa diferente.

En el ordenador A o simulador, se instalará MATLAB en su versión MATLABR2017b. En esta computadora se desarrollará el programa "simulador.slx", "bancodepruebas.slx" y "bancodepruebasbreaker.slx" a partir de dos modificaciones realizadas sobre el software ya desarrollado por parte de la Universitat Politècnica de València para este mismo proyecto.

A su vez, se incluirán diversas "scopes" o pantallas de ploteo (figura I.4.1) que permitirán monitorizar en todo momento el software con el objetivo de comprobar que la generación de datos se está realizando de forma correcta y que dichos datos coinciden con aquellos recibidos por la computadora B.

Por último, el simulador tendrá instalado también Wireshark que ayudará a estudiar las señales recibidas como enviadas por el ordenador B y permitirá encontrar posibles errores que se pudiesen originar durante el funcionamiento del programa.

En el ordenador B o SCADA, se instalará LabVIEW (32-bit) English junto a sus extensiones: LabVIEW Control Design and Simulation Module (32-bit), LabVIEW Database Connectivity Toolkit (32-bit), LabVIEW Report Generation Toolkit (32-bit) y LabVIEW Runtime (32-bit).

En dicho software se desarrollará el proyecto "promotion.lvproj" con sus respectivos .vi y .subvi que acabarán transformándose a un archivo .exe para su correcta distribución.

Al igual que en el anterior ordenador, se instalará también el programa wireshark para analizar el envío y la recepción de señales.

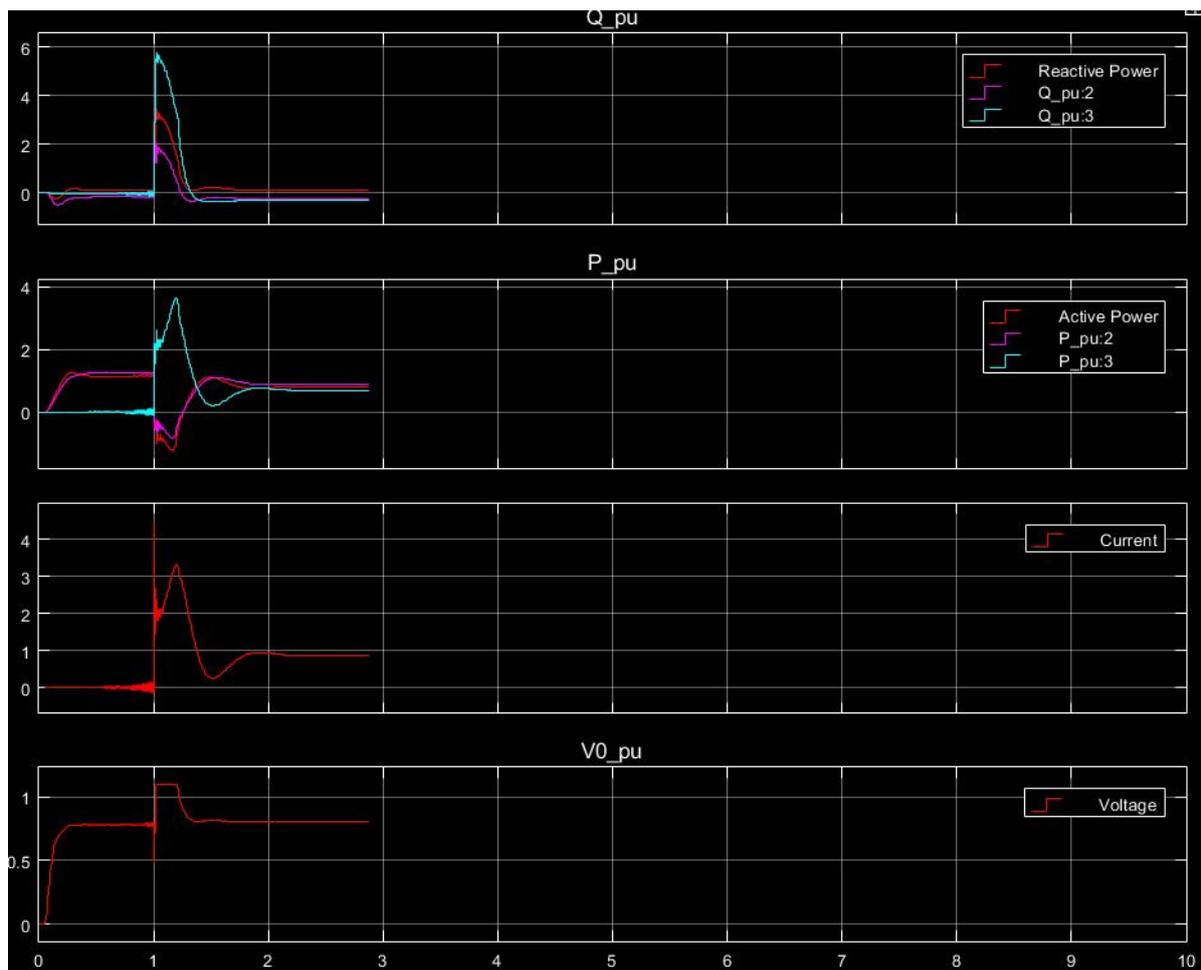


Figura I.4.1: Recortes de las gráficas generadas por MATLAB

Para finalizar, hay que destacar que, para que el conjunto funcione de una forma óptima, ambas computadoras deberán trabajar bajo la misma red WIFI. Una vez comprobado este aspecto, se averiguará la dirección IP de ambas máquinas pues habrá que modificarlo tanto en "simulador.slx", "bancodepruebas.slx" y "bancodepruebasbreaker.slx" como promotion.lvproj; dicha modificación quedará reflejada en el Manual de Programador presente en el Anexo de este mismo Trabajo de Final de Grado (figura I.4.2).

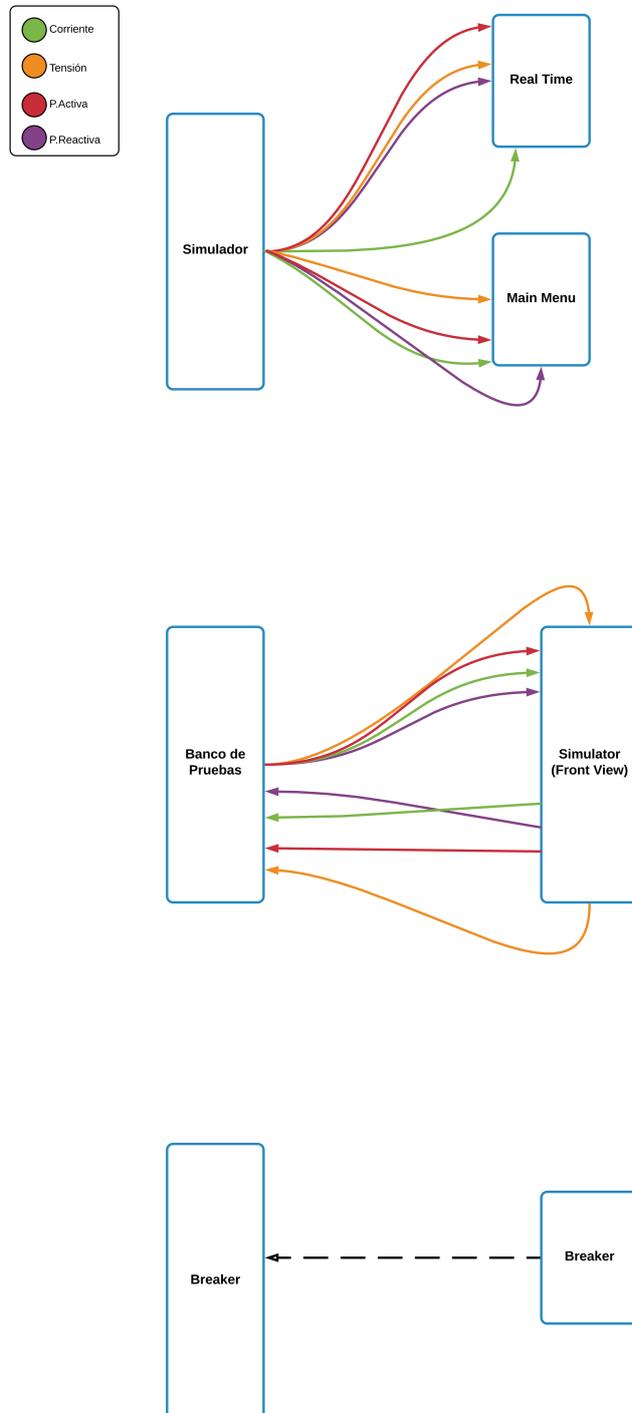
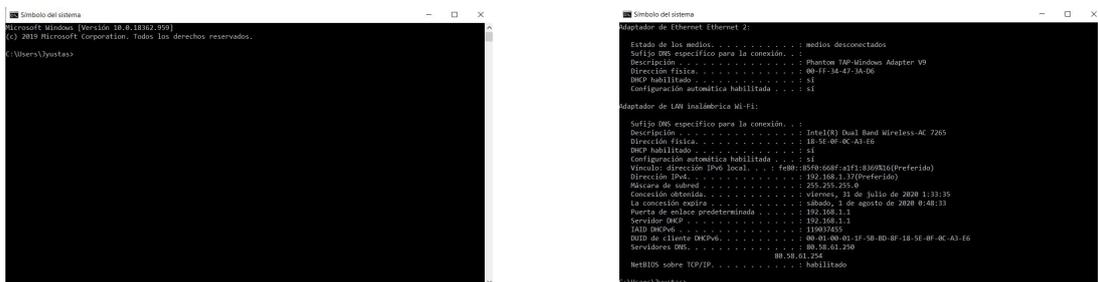


Figura I.4.2: Esquema general de comunicación entre códigos

En dicho manual se detallará también cuales son los puertos utilizados para el envío y recepción de cada señal por si se requiriese a la hora de utilizar los programas o en una posible implementación del programa en otras máquinas.

Los pasos a seguir para averiguar la dirección IP son sencillos (figura I.4.3):

1. Abrir la consola de comandos cmd (se puede encontrar tecleando "cmd" en el buscador de Windows).
2. Ejecutar el comando "ipconfig/all"
3. Localizar la dirección IP ("Dirección IPv4").



(a) Recorte del proceso de obtención de la dirección IP (1/2) (b) Recorte del proceso de obtención de la dirección IP (2/2)

Figura I.4.3: Recortes proceso de obtención IP

I.4.3 Requisitos del sistema

El primer aspecto a tener en cuenta si se desea reproducir el presente trabajo es disponer de dos computadoras diferentes, conectadas a la misma red wifi y con los programas anteriormente comentados instalados.

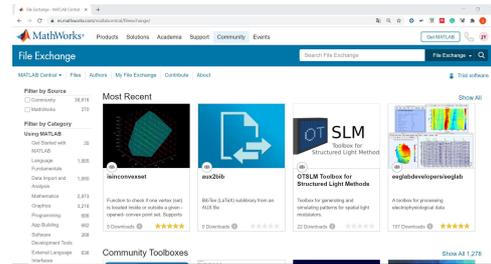
En segundo lugar, se tendrá que averiguar las direcciones IP y realizar los cambios pertinentes de acuerdo con el Manual de Programador y Manual de Usuario presentes en el Anejo de este documento.

Se deberá disponer de periféricos de entrada de tipo ratón y teclado, o similar, en ambas computadoras para hacer uso de forma correcta de todos los programas; además de sendos monitores para poder visualizar los avances en ambos programas.

Cambiando de punto, a lo largo de esta programación se ha tomado como apoyo los foros y comunidades creadas por las propias empresas a la hora de resolver dudas y de generar conocimiento de forma autodidacta (figura I.4.4).



(a) Recorte del foro de National Instruments



(b) Recorte del repositorio de MATLAB

Figura I.4.4: Recortes de los foros de los diferentes softwares

I.4.4 Estructura del programa

Se debe analizar por separado las estructuras de ambos programas: por una parte, el simulador, desarrollado con lenguaje de Simulink y por otra parte el sistema SCADA programado con la tecnología de National Instruments.

Para el desarrollo del programa de MATLAB se contó desde un inicio con una base que simulaba tres aerogeneradores y consistía, a grandes rasgos, en dos grandes pantallas, la primera "SC Console" que realiza la función de monitor y la segunda "SM Master" que es el simulador en sí.

Dentro de estas dos hay numerosas subpestañas que permiten al futuro programador navegar de una forma intuitiva por el mismo. A partir de esta plantilla, se implementó la comunicación vía UDP, esto se ha realizado modificando la representación simple mediante un "scope" por un diagrama de bloques que consiste en tres partes: aplicar una ganancia, codifica a ASCII y envía. Esta última ecuación se ha aplicado a todas y cada una de las variables a controlar: tensión, corriente, potencia activa y potencia reactiva.

En el caso del banco de pruebas, se dispone además unas entradas que provienen del programa SCADA, para ello se han modificado los parámetros iniciales por una secuencia de bloques con el objetivo de "simular" un procesado de señal por parte del prototipo a escala, pero en sentido inverso: primero se recibe y posteriormente se decodifica.

En el caso de LabVIEW, se han desarrollado diferentes .vi de las cuales nacen otras, denominadas como .subvi. Este árbol se ha generado para intentar optimizar el funcionamiento del programa, pensando a su vez en posibles futuras ampliaciones.

El funcionamiento de este será analizado con más profundidad en el siguiente punto cuando se habla de la interfaz visual del sistema SCADA.

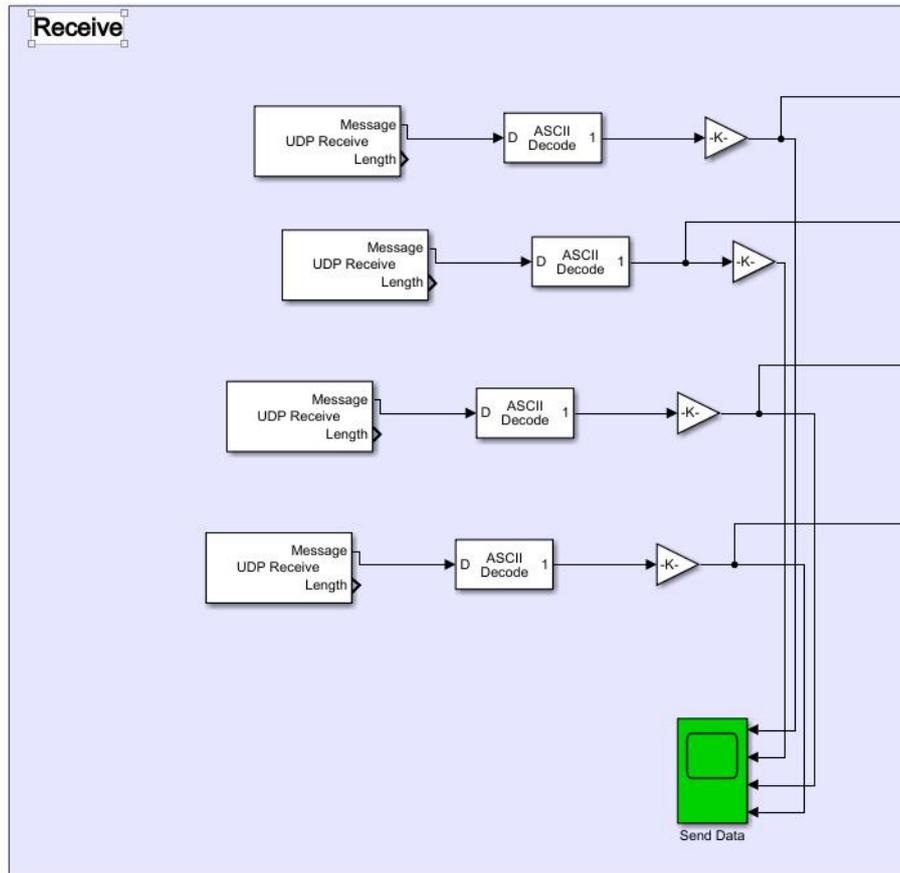


Figura I.4.5: Recorte de la recepción de datos vía UDP por parte de Simulink

I.4.5 Control y medida

Para el control de los datos en Simulink, como se ha señalado anteriormente, se ha dispuesto de una serie de gráficas que permiten contrastar los datos con aquellos proyectados por el sistema SCADA.

Por su parte, el programa SCADA cuenta con la posibilidad tanto de ver las gráficas finales, como de poner "probetas" a lo largo de la circuitería virtual del diagrama de bloques, este aspecto de LabVIEW permite ver cómo van evolucionando los datos a lo largo de nuestro "block diagram" y ayuda a ahorrarnos mucho tiempo y esfuerzo a la hora de detectar errores en la programación.

En la figura I.4.6 se pueden observar unos ejemplos de "probetas" para una .vi programada mediante LabVIEW que recibe valores a través de un protocolo UDP y los procesa hasta almacenarlos en una base de datos.

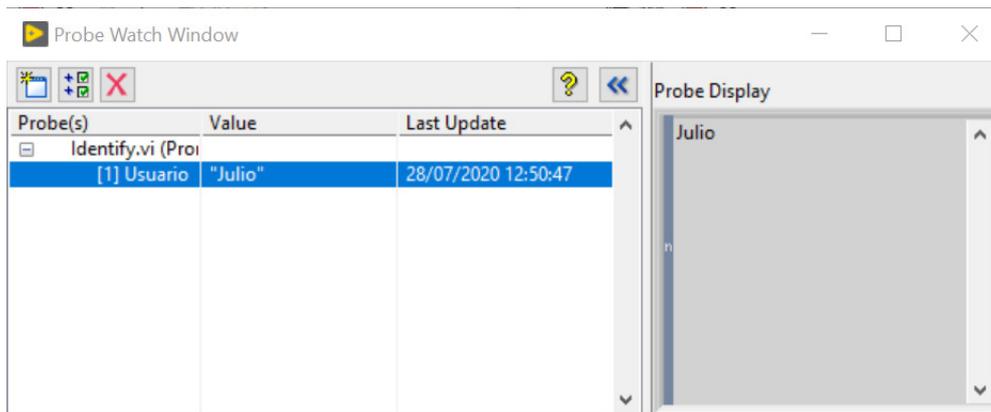


Figura I.4.6: Recorte de las "probe" en LabVIEW

I.4.6 Elementos adicionales

Para la configuración del programa se ha utilizado también en momentos puntuales software como: Photoshop o Gimp; con el objetivo de crear una interfaz de usuario más atractiva y visualmente ergonómica. Además de las requeridas para hacer esquemas y redactar: Canva, Texmaker, TeXnicCenter y Lucidchart.

Asimismo, se han desarrollado dos bases de datos en formato .txt para que el usuario, a través del propio programa SCADA, pueda almacenar y/o consultar toda la información que desee sobre el registro de usuarios y los datos recopilados por el sistema.

Por último, se ha empleado el servidor SMTP de Gmail en el .vi de "sign-up" para enviar un correo de confirmación a todos los usuarios que se registren; creando a su vez un correo para el remitente: promotion.upv.app@gmail.com (figura I.4.7).

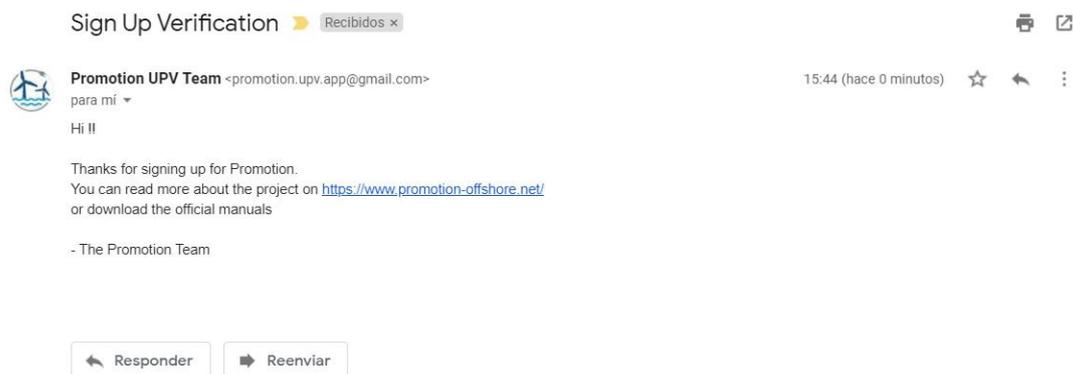


Figura I.4.7: Recorte del correo electrónico de confirmación

Capítulo I.5

Solución del problema

A lo largo de la presente sección se abordará la concepción de la solución para el problema planteado inicialmente.

Con la finalidad de mejorar la comprensión del programa, el capítulo comenzará con una pequeña toma de contacto con el software en forma de "boceto", es decir, se analizará la solución tal y como se concibió, de lo general a lo específico, siguiendo el esquema mental propio de un arquitecto de software.

Una vez elaborado el esquema general, el trabajo se adentrará en el "corazón" del programa, se describirá el entorno visible por el usuario, identificando todas las funcionalidades de este; el funcionamiento del programa de forma teórica, y su aplicación práctica con lenguaje MATLAB y LabVIEW.

Por último, pero no por esto menos importante, se describirán posibles propuestas de mejora; éstas, junto con el manual de programador adjuntado en el Anexo, ofrecerá al lector cimientos para extender las competencias del presente programa con la mayor facilidad posible.

Índice

I.5.1	Introducción	45
I.5.2	Diseño	46
I.5.3	Implementación	48
I.5.3.1	Fase de inicialización	48
I.5.3.2	Selección de modo	57
I.5.3.3	Interfaz visual	62
I.5.4	Código en MATLAB	84
I.5.4.1	Simulador	84
I.5.4.2	Banco de pruebas	85
I.5.4.3	Breaker	85
I.5.5	Posibles mejoras	85

I.5.1 Introducción

La monitorización de todo parque energético es una parte fundamental a la hora de desarrollar cualquier proyecto energético. Este aspecto se potencia aún más cuando se habla de huertos fotovoltaicos o parques eólicos con tecnología en proceso de crecimiento.

En esta tesitura, y con objeto de servir como apoyo para el proyecto de investigación desarrollado por la Universitat Politècnica de València como colaboración dentro del proyecto europeo PROMOTioN, se ha programado un software con todas las competencias necesarias para controlar el simulador físico situado en el Instituto de Automática Industrial dentro de la Ciudad de la Innovación.

Dada la inviabilidad técnica provocada por la crisis sanitaria generada por el virus COVID-19, dicho simulador ha tenido que ser sustituido por una versión digital desarrollada en MATLAB; intentado, en todo momento fomentar la simbiosis entre ambos campos.

Las funciones iniciales del programa, tal y como se indica en los anexos de este trabajo son:

- Recibir datos de tensión, corriente, potencia activa y potencia reactiva.
- Ser capaz de parar el simulador, mediante la creación de una especie de interruptor automático virtual.
- Permitir al usuario realizar ensayos contra el simulador, introduciendo datos de tensión, potencia activa y potencia reactiva y plotearlos posteriormente.
- Consultar el histórico de datos, creando a su vez una base de datos propia.

Para cumplir con las mencionadas características, se empezó analizando programas de comportamiento similar; por ejemplo, los sistemas de monitorización desarrollados por las empresas de inversores fotovoltaicos como KOSTAL I.5.1.

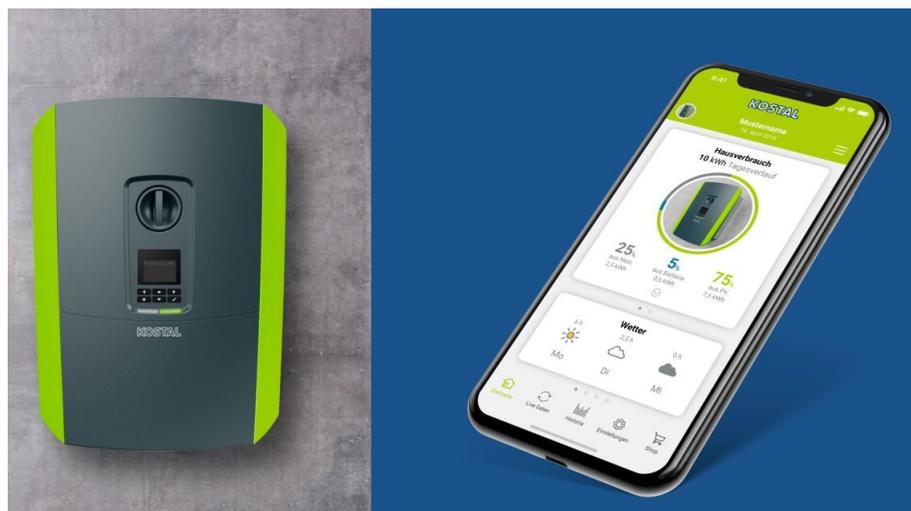


Figura I.5.1: Kostal monitoring portal, Kostal 10/08/2020

La experiencia en este tipo de programas, tanto en el software del fabricante como en el propio programa de empresa, fue adquirida durante mi estancia en prácticas en la empresa SOLATS

S.L. En este caso, se analizaba la potencia generada por pequeñas instalaciones fotovoltaicas y la energía consumida y vertida a la red eléctrica.

A pesar de las evidentes diferencias entre ambos trabajos, en aquel momento se contaba con la ayuda de otro ingeniero y de un programador para "construir" un programa mucho más sencillo, ambos proyectos comparten la misma esencia.

I.5.2 Diseño

Como se ha comentado anteriormente, para el desarrollo de este proyecto se han tomado como ejemplo algunos de los programas desarrollados por los propios fabricantes de los equipos energéticos.

En estos programas, existen tres zonas claramente diferenciadas que se repiten de forma frecuente: la primera de ellas es la zona de identificación/ registro de usuarios, que en el caso de las instalaciones fotovoltaicas suponía un paso fundamental, pues cada credencial debía dar acceso a una única información. La segunda era una interfaz de control donde el usuario tenía el poder de consultar varios datos: información de la instalación, de los equipos instalados, de la empresa contratista, etc. La última zona, presente en el 100 % de los programas analizados era la zona de consulta de datos, tanto en tiempo real como el histórico de datos.

Una vez realizado el análisis, se realizó un pequeño boceto en el que se vislumbraban a "grosso modo" las características del programa a construir. En la figura I.5.2, se puede analizar una recreación de aquel "brainstorming". Dentro de este diagrama, las casillas negras representan interfaces pensadas desde un inicio mientras que las de una tonalidad más verdosa representan aquellas que fueron añadidas posteriormente, bien por necesidad o bien por ampliar el potencial del software de control.

Dicho esquema será ampliado en el último apartado de este punto cuando se especifiquen algunas de las propuestas de mejora para el programa; el resto de las propuestas de mejora, con un carácter más general, estarán recogidas en el punto de "Visión Futura y Ampliaciones".

En el boceto se pueden identificar varios componentes:

- Ventana de identificación o "Log-In": es la ventana inicial del programa, pide al usuario sus credenciales, condición "sine qua non" para acceder a todo el potencial de este software.
- Ventana de registro o "Sign-Up": es una ventana que se accede a través de la ventana de identificación y permite a los usuarios no registrados hacerlo, almacenando sus credenciales en la base de datos login.txt y enviando un correo de confirmación.
- Menú principal o "Main menu": se podría decir que esta ventana es el corazón del programa, es el punto al que se volverá siempre que se desee cambiar de información visualizada y tiene contacto con casi todas las "front views" del programa.
- Ventana de datos a tiempo real o "Real Time": como su propio nombre indica, proyecta los datos recibidos de tensión corriente y potencias activa y reactiva.
- Ventana de histórico de datos o "Data History": es un conjunto de ventanas que permiten al usuario elegir la fecha exacta, con precisión de décimas de segundo y proporcionando al usuario los datos más próximos a la fecha de inicio y fin seleccionada.

- Ventana de pruebas o "Simulator": se conecta con el bando de pruebas, actuando de una manera muy parecida a la ventana de "Real Time" con la diferencia de que en este caso es el usuario el que decide que datos va a recibir el sistema.
- Ventana de ayuda o "Help": es la ventana que recoge todos los datos relativos al programa, sus creadores y el proyecto madre de este.
- Ventana de interruptor automático o "Breaker": permite controlar la conexión / desconexión del programa en todo momento.
- Ventana de errores o "Error Log": recoge los errores que se puedan originar en el resto de las ventanas, mostrando una luz verde o roja, según salte el error o no y en su caso, la codificación del mismo y una breve descripción. Para conocer más acerca de los errores, en su apartado, habrá que consultar el manual de usuario o el manual de programador.

Todos estos componentes se analizarán con más profundidad justo a continuación en los respectivos apartados dedicados en exclusiva para cada uno de éstos I.5.2.

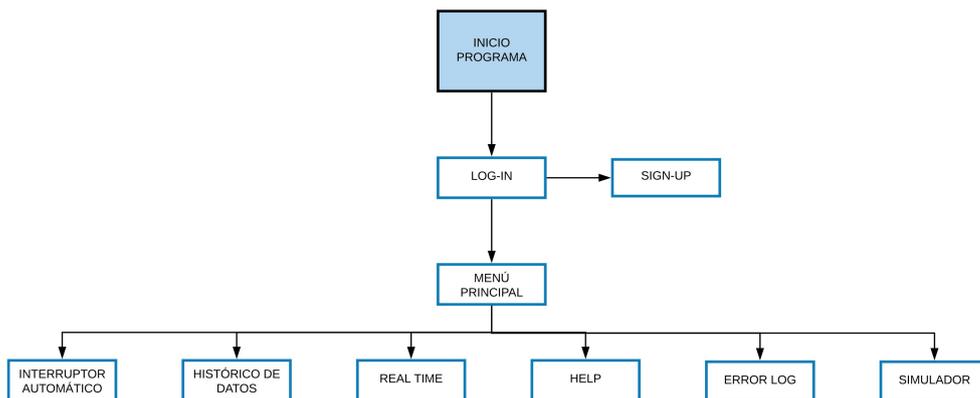


Figura I.5.2: Diagrama de flujo simplificado

I.5.3 Implementación

I.5.3.1 Fase de inicialización

La fase de iniciación del programa consta de dos ventanas claramente diferenciadas: la pestaña de identificación o "log-in" y la pestaña de registro o "sign-up".

La función global de esta fase es "dar la bienvenida" al usuario al programa, permitiéndole identificarse en el sistema, o en su defecto registrarse en la base de datos de operadores autorizados.

A continuación, se analizará cada pestaña más detalladamente, dividiendo el estudio en diferentes fases:

- Captura del interfaz visible por el usuario (entorno gráfico).
- Breve descripción de las características de la pestaña.
- Identificación de los componentes operables por el usuario y su funcionalidad.
- Captura del "block diagram" (entorno de programación).
- Breve descripción de las propiedades del diagrama de bloques.
- Identificación de los componentes programables por el usuario y funcionalidad.

I.5.3.1.1 Log-In Front View

Entorno gráfico

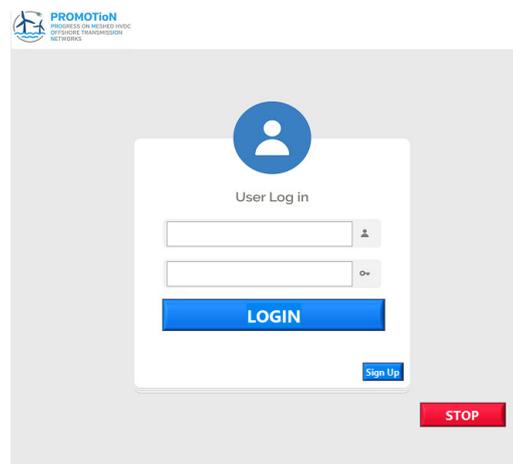


Figura I.5.3: Recorte de la "front view" del inicio de sesión de usuario

Descripción

Esta ventana permite al usuario identificarse mediante un nombre de usuario y una contraseña. Dichas credenciales pueden haber sido registradas por el mismo operario o venir proporcionadas por el programador mediante una escritura directa sobre la base de datos utilizada por el software.

Como se ha mencionado, se ofrece la posibilidad de que el usuario registre sus datos manualmente. Para ello, debe pulsar sobre el botón "sign-up" localizado en la esquina inferior derecha de la ventana.

Elementos operables

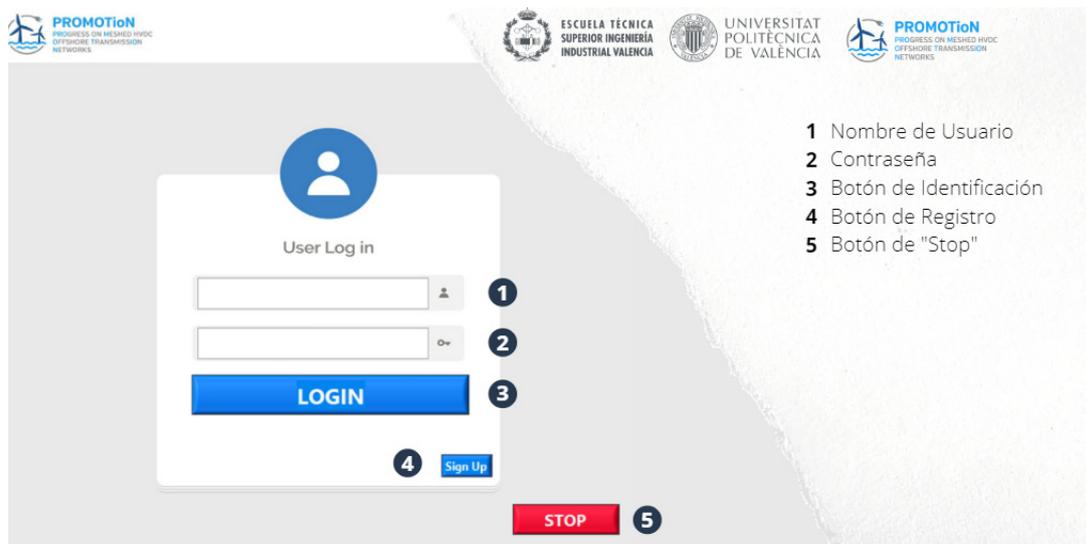


Figura I.5.4: Infografía elementos operables

1. **Nombre de usuario:** el usuario deberá de rellenar la casilla con su nombre de usuario.
2. **Contraseña:** este es el lugar donde introducir la contraseña para la credencial de usuario anteriormente completada. Esta introducción de texto se encuentra codificada, es decir, por motivos de seguridad, el usuario del programa no podrá ver lo que está escribiendo.
3. **Botón de identificación:** una vez completados los dos campos anteriores, habrá que clicar sobre este botón. En el caso de que las credenciales sean válidas, se avanzará hasta el menú principal, autorizando al usuario hacer uso del potencial completo del programa; si, por el contrario, los datos introducidos fueran incorrectos, el programa generará un mensaje de error, notificando el problema.
4. **Botón de registro:** este mando permite al cliente sin credenciales acceder a la pestaña de registro donde puede registrarse en la base de datos del programa para su posterior identificación.
5. **Clavija de "Stop":** si el cliente pulsa este botón, el programa finalizará de inmediato y se generará un mensaje donde se confirma al usuario que este cierre se ha producido con éxito.

Entorno de programación

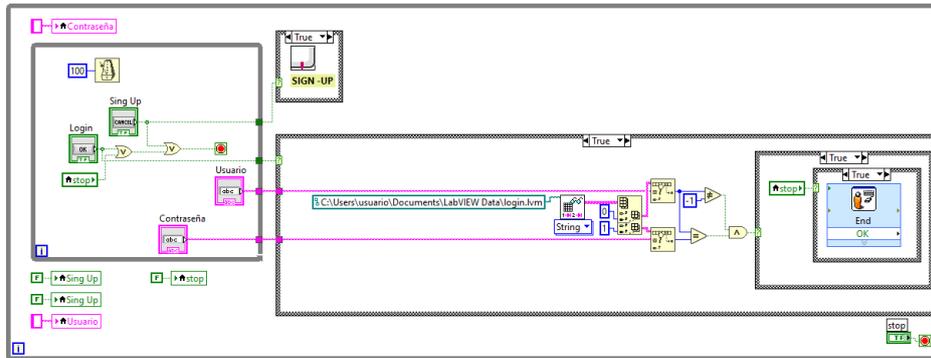


Figura I.5.5: Recorte del "block diagram" del inicio de sesión de usuario

Descripción

Se presenta ahora el diagrama de bloques o programación del entorno gráfico para la identificación de usuario.

El presente código se sitúa en el interior de un bucle "while" cuya función es repetir el programa, siempre y cuando no se valide la opción que finaliza el bucle de repetición, es decir, botón de "stop" pulsado, lo cual generaría un mensaje de cierre y finalizaría el programa; a la vez que reinicia las variables y los campos de relleno.

Dentro de este bucle se encuentra otro bucle "while" con las casillas para introducir las credenciales: nombre de usuario y contraseña, y las clavijas ("login" y "sign-up"). Serán estas dos últimas las encargadas de permitir el avance hasta los siguientes bloques.

Sendos bloques son condicionales: el primero de ellos ejecutará la pestaña de registro en el caso de que la validación del bucle "while" anterior se haya hecho mediante el pulsado del botón "sign-up".

El segundo condicional se ejecuta cuando se pulsa el botón de "login"; en este momento el programa abre la base de datos "login.lvm" y se produce una consulta línea a línea del nombre de usuario y contraseña. Dependiendo de si los datos introducidos se han encontrado o no, el programa autorizará al usuario pasar a la pestaña de "main menu" o le emitirá un mensaje de error y le instará a continuar probando, lo que devolvería al cliente al estado inicial.

Componentes

1. **Bucle "while"**: tiene la función de repetir el proceso hasta que se verifique alguna de las condiciones de finalización: se abra la pestaña de registro, se pulse el botón de cierre del programa o se identifique el usuario correctamente.
2. **Reinicio de variables**: en este punto, el código reinicia las variables cada vez que se inicia el programa o el cliente se identifica de forma errónea.

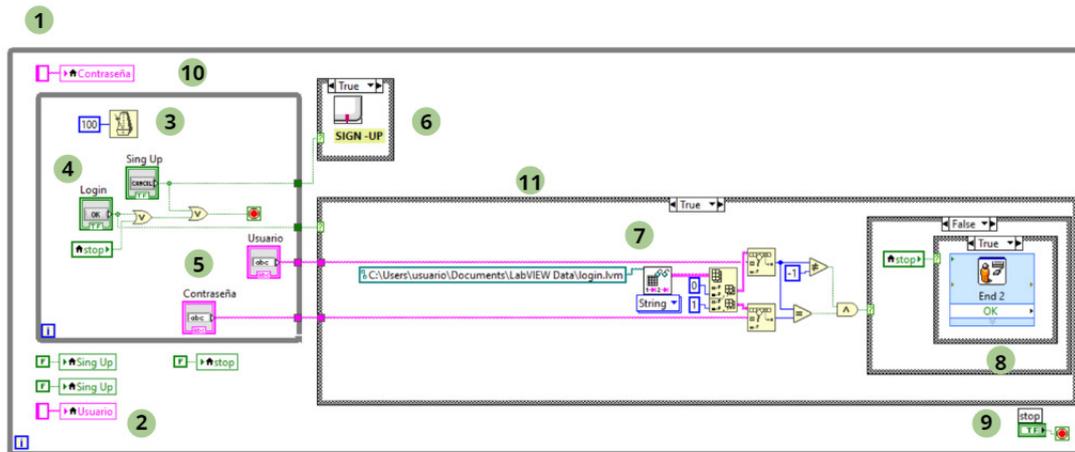


Figura I.5.6: Componentes del "block diagram" del inicio de sesión de usuario

3. **Temporizador:** marca el tiempo que tarda el programa en ejecutar la siguiente repetición del bucle "while".
4. **Clavijas de "Log-In" y "Sign-Up":** son las encargadas de continuar con el proceso, dando paso a la apertura de una u otra pestaña según cuál de ambas sea la tecleada.
5. **Campos de relleno:** posibilitan al usuario introducir sus credenciales con el objetivo de que el programa las valide y le dé acceso.
6. **Estructura Sign-Up:** esta estructura de casos da paso a la apertura de la ventana de "sign-up" en el caso de que ese botón sea el seleccionado, pero si no lo es, no hará nada.
7. **Búsqueda en la base de datos:** el código abrirá la base de datos "login.lvm" y contrastará los datos introducidos por el usuario con los allí presentes. Si coinciden, se concederá acceso para trabajar con la versión completa del programa, si, por el contrario, no se encontraran, generará un "prompt" notificando al usuario que se ha producido un error en el proceso de verificación.
8. **Acciones:** esta serie de estructuras de casos darán paso a una u otra acción, dependiendo de si se cumplen o no las condiciones requeridas.
9. **Botón de "stop":** finalizará el programa en cualquier momento, sacando al código del bucle "while".
10. **Bucle "while" de relleno de credenciales:** se repite siempre y cuando el usuario no haya rellenado sus credenciales y pulsado el botón de "log-in", clicado la clavija de "sign-up" o la de "stop".
11. **Estructura de casos:** se ejecutará siempre y cuando se haya pulsado sobre el botón de "log-in", en ese momento se dará paso a las acciones descritas en el punto 7.

I.5.3.1.2 Sing-Up Front View

Entorno gráfico

Figura I.5.7: Recorte de la "front view" del registro de usuario

Descripción

La pestaña de "sign-up" habilita al usuario a darse de alta dentro de la base de datos del sistema, guardando sus datos en el fichero "login.lvm". Esto se llevará a cabo mediante la introducción de los siguientes datos: nombre de usuario, teléfono móvil (optativo), correo electrónico y contraseña.

El programa dará el visto bueno si se cumplen las siguientes condiciones:

- El correo electrónico y el nombre de usuario no han sido utilizados anteriormente.
- Todos los campos han sido completados (con la excepción del teléfono móvil).
- Las contraseñas utilizadas coinciden.

En el momento que se validen todas las condiciones, el software incluirá sus credenciales en la base de datos y enviará un correo electrónico semejante al que aparece en la figura I.5.8 donde se confirma que el registro se ha realizado con éxito.

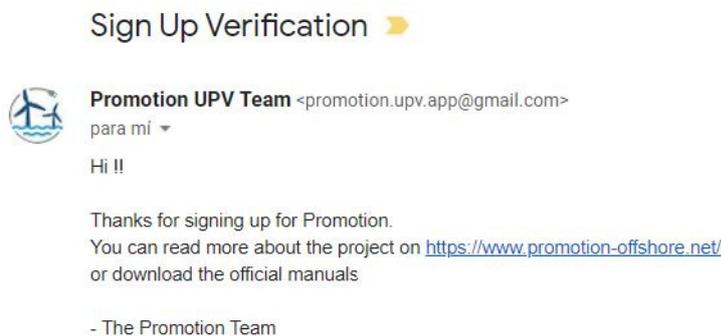


Figura I.5.8: Recorte del correo de confirmación

Elementos operables

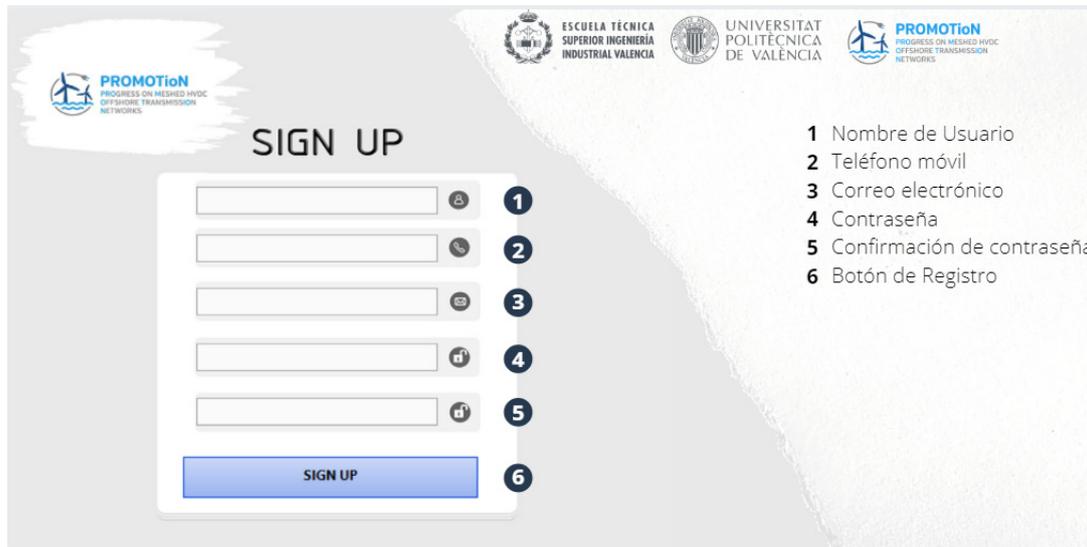


Figura I.5.9: Infografía elementos operables

1. **Nombre de usuario:** el usuario deberá de rellenar la casilla con su nombre de usuario.
2. **Teléfono móvil:** el cliente tendrá la posibilidad de incluir, o no, su número de teléfono dentro de la información recogida por el sistema.
3. **Correo electrónico:** el usuario deberá facilitar una dirección de correo no utilizada previamente en este mismo software.
4. **Contraseña:** campo reservado para la introducción de la contraseña personal de cada usuario, igual que el de la ventana de "log-in" se encuentra codificado por motivos de seguridad.
5. **Confirmación de contraseña:** sirve para verificar que no ha existido un error en la introducción anterior de la contraseña; los "password" introducidos en el punto 4 y 5 han de coincidir.
6. **Botón de registro:** el operario deberá de pulsar sobre este botón una vez crea que todos los campos han sido completados correctamente. En este punto se dará paso al programa para que continúe con su proceso de verificación: comprobar requerimientos y enviar un correo de confirmación.

Entorno de programación

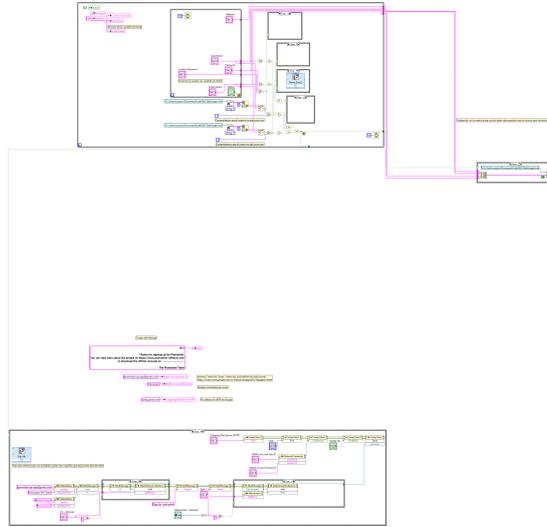


Figura I.5.10: Recorte del "block diagram" de registro de usuario

Descripción

En los recortes anteriores se aprecia el código de programación para la ventana de registro de usuario.

El primer componente a destacar es un gran bucle "while" dentro del cual se engloban varios elementos:

- Un bucle de la misma tipología cuyo objetivo es repetir el programa tantas veces como sea necesario hasta que se rellenen los campos de registro.
- Un reinicio de las variables a completar, esto se realizará al inicio de la ventana y en el caso de que se tenga que repetir el registro por ser incorrecto.
- La comprobación de que las diferentes condiciones para que el registro se haya realizado de forma exitosa se cumplen: todos los campos completados y el usuario y correo electrónico no utilizados previamente.
- Además, se puede encontrar un timing y sendas condiciones para la finalización de ambos bucles.

Continuando en el exterior de la mencionada estructura, se encuentra una estructura condicional que tiene la función de almacenar los parámetros en la base de datos "login.lvm", en el caso de que se hayan cumplido todas las condiciones anteriores.

Por último, se puede observar un inicio de una serie de variables de tipo "string", texto, utilizadas posteriormente en la estructura que se encuentra justo debajo y cuya función es enviar un correo de confirmación al usuario que se acaba de registrar.

En caso de que todo lo mencionado anteriormente se haya llevado a cabo de forma exitosa, el programa generará un mensaje de confirmación al usuario.

Componentes

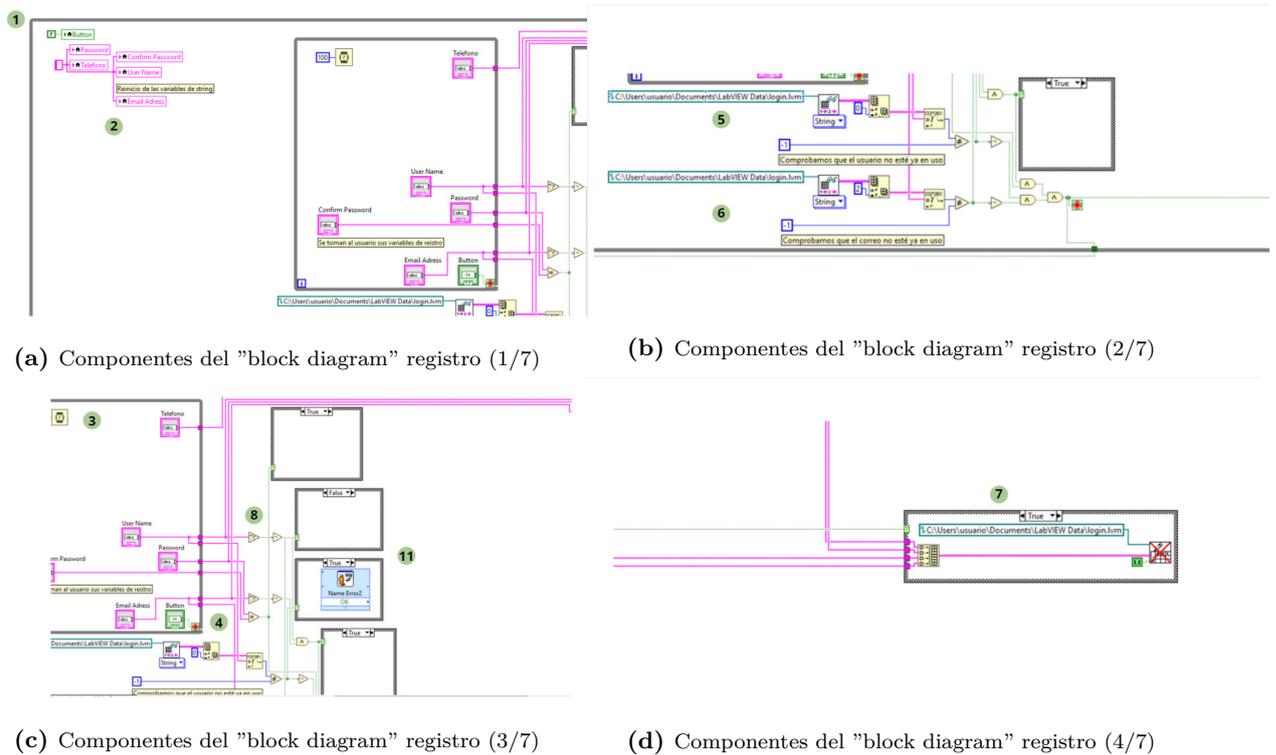
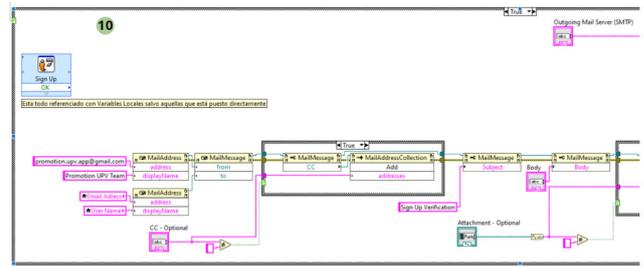


Figura I.5.11: Componentes del "block diagram"

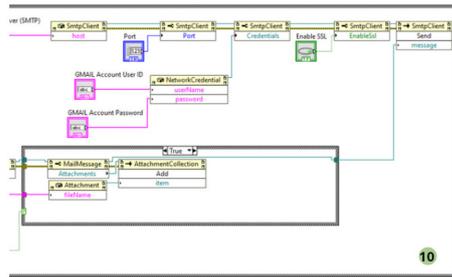
1. **Bucle "while"**: tiene la función de repetir el proceso hasta que se verifiquen todas las condiciones para la finalización.
2. **Reinicio de variables**: en este punto el código reinicia las variables cada vez que se inicia el programa o el cliente se registra de forma errónea.
3. **Temporizador**: marca el tiempo que tarda el programa en ejecutar la siguiente repetición del bucle "while".
4. **Clavija de "Sign-Up"**: es la encargada de continuar con el proceso, dando paso a que el programa valide las condiciones de registro.
5. **Nombre de usuario**: este pequeño bloque de código ya ha sido utilizado en la anterior ventana de identificación y tiene como objetivo buscar en la base de datos para determinar si el nombre de usuario ha sido utilizado previamente o no.
6. **Correo electrónico**: este código trabaja de forma completamente análoga al anterior (punto 5).
7. **Escritura en la base de datos**: el código abrirá la base de datos "login.lvm" y redactará sobre ella las nuevas credenciales.
8. **Condicionales**: esta serie de bloques booleanos permitirán seguir avanzando en el código según se cumplan, o no, las condiciones anteriormente expuestas.
9. **Preparación del mensaje**: inicia las variables necesarias para el envío del mensaje al usuario, cumplimentando cada una con una constante ya programada: correo electrónico del remitente, servidor de correo electrónico y el cuerpo del mensaje.



(a) Componentes del "block diagram" registro (5/7)



(b) Componentes del "block diagram" registro (6/7)



(c) Componentes del "block diagram" registro (7/7)

Figura I.5.12: Componentes del "block diagram"

10. **Estructura de envío:** se ejecuta solo si se cumplen todas las condiciones anteriores y su función es hacer llegar el mensaje al usuario lo antes posible; además genera un mensaje de confirmación una vez se ha realizado todo con éxito.
11. **Mensajes de error:** se ejecutará uno u otro dependiendo de la condición incumplida durante el registro de usuario.

I.5.3.2 Selección de modo

La fase de selección de modo del programa consta de la ventana de "Main Menu".

I.5.3.2.1 Main Menu Front View

Entorno gráfico

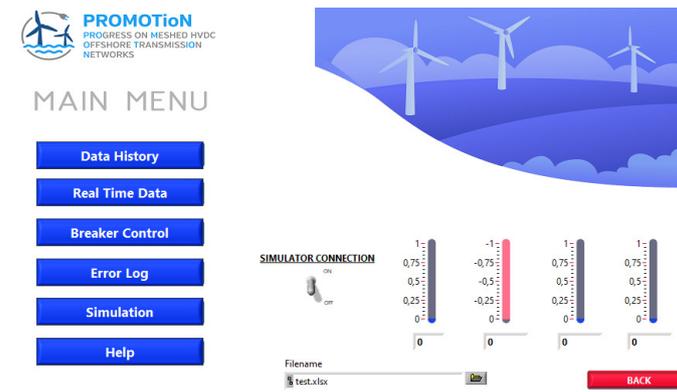


Figura I.5.13: Recorte de la "front view" del menú

Descripción

La presente ventana permite al usuario elegir entre las diferentes funcionalidades del programa: ver datos en tiempo real, consultar el histórico de datos, entrar al simulador, controlar el interruptor automático, consultar los errores y abrir la pestaña de información del programa.

Además, se incluye una breve información sobre los datos en tiempo real, aunque esta funcionalidad se explicará con más detalle en el punto dedicado a la ventana de datos en tiempo real.

Elementos operables

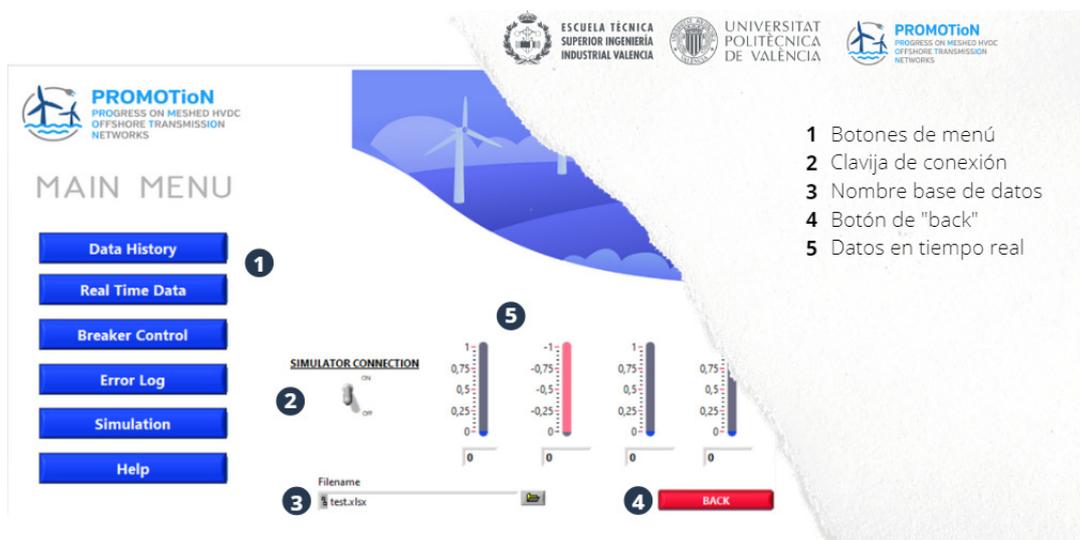


Figura I.5.14: Infografía elementos operables

1. **Botones de menú:** permiten al usuario elegir a qué pestaña de todas las disponibles desea acceder: ver datos en tiempo real, consultar el histórico de datos, entrar al simulador, controlar el interruptor automático, consultar los errores y abrir la pestaña de información del programa.
2. **Clavija de conexión:** habilita al programa a conectarse contra el simulador (o el prototipo a escala) y empezar a recibir datos vía UDP en tiempo real.
3. **Nombre para la base de datos:** posibilita al usuario elegir el nombre del fichero en el que se almacenarán los datos que se reciban en tiempo real. Se recomienda no cambiar el nombre definido por defecto ya que, de hacerlo, también habría que modificar ciertas partes del código.
4. **Botón de "back":** de ser pulsado, retrocedería al usuario hacia la ventana de registro. Puede ser utilizado en el caso de que se quiera finalizar el programa y/o cambiar las credenciales con los que se han accedido.
5. **Datos a tiempo real:** esta serie de indicadores permiten al usuario recibir y consultar los datos que se están generando a tiempo real sin necesidad de abrir la pestaña específica de "real time".

Entorno de programación

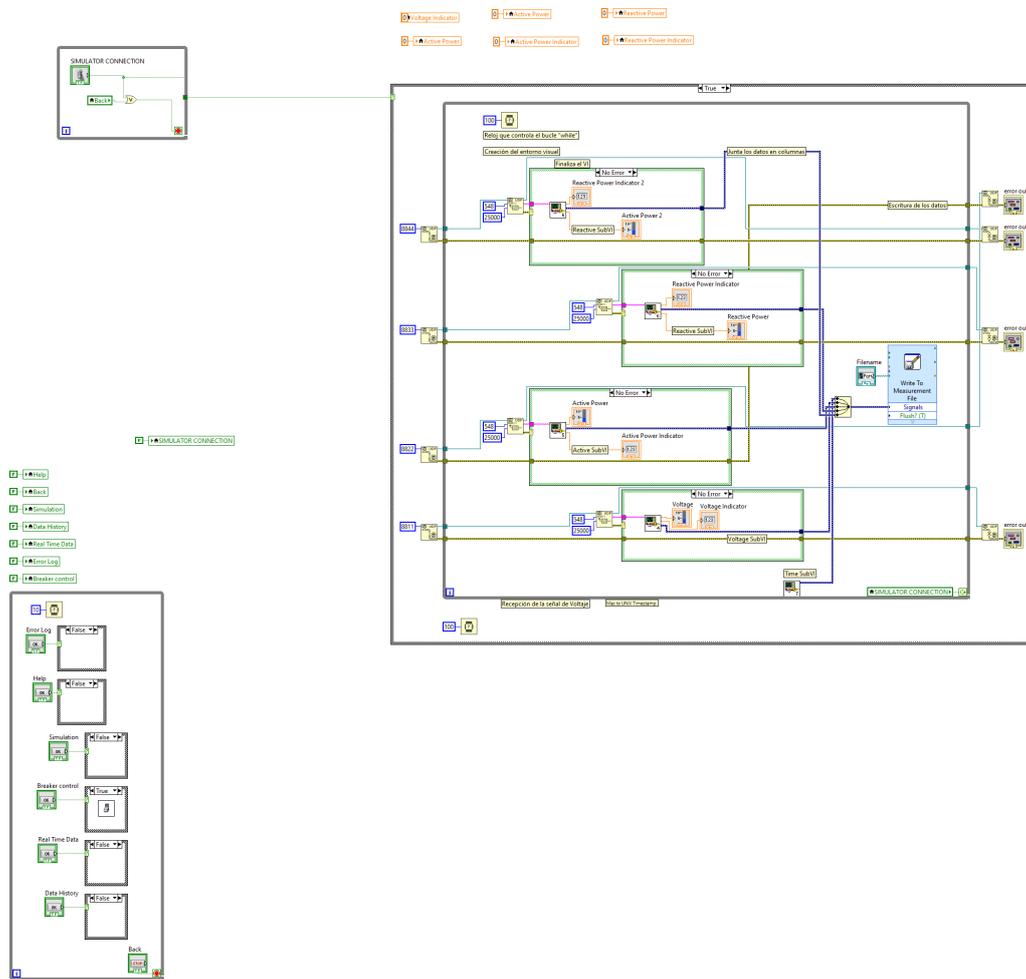


Figura I.5.15: Recorte del "block diagram" del menú

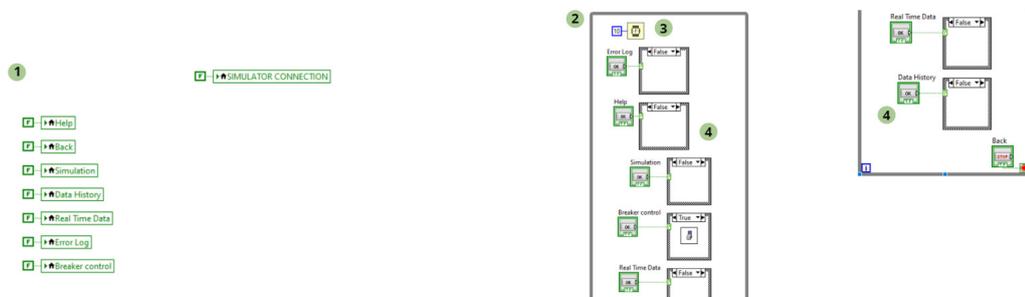
Descripción

En este "block diagram" se encuentran dos elementos clave que actúan de forma simultánea y totalmente independiente:

Un primer bucle "while" que repite el código desde el inicio del programa a la vez que se reinician las variables utilizadas en el mismo; este bucle se repite constantemente incluso si se valida alguna de las estructuras condicionales que se localizan dentro del mismo. La única forma de finalizarlo es tecleando sobre el botón de "back".

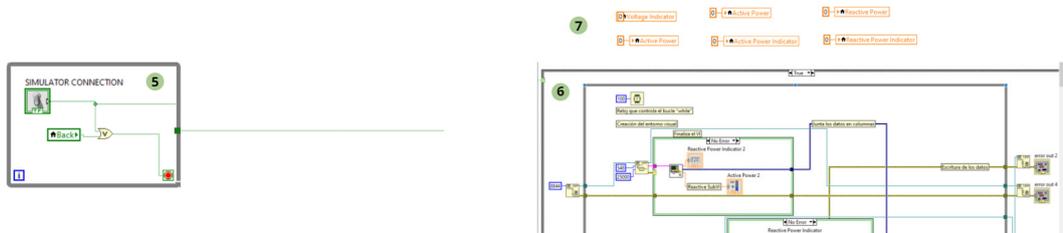
Otro bucle "while" dentro de una estructura condicional que se habilita una vez se conecta la clavija de recepción de datos en tiempo real; dentro de este bucle se tiene, repetido tantas veces como datos se desee recibir, el proceso de apertura, lectura y cierre del protocolo UDP, así como el tratado de la información para poder representarla de forma correcta en la "front view".

Componentes



(a) Componentes del "block diagram" del menú (1/4)

(b) Componentes del "block diagram" del menú (2/4)



(c) Componentes del "block diagram" del menú (3/4)

(d) Componentes del "block diagram" del menú (4/4)

Figura I.5.16: Componentes del "block diagram"

1. **Reinicio de variables:** en este punto el código reinicia las variables cada vez que se inicia el programa.
2. **Bucle "while":** tiene la función de repetir el proceso de forma continuada incluso si se verifica alguna de las estructuras condicionales, hasta que se pulse el botón de "back".
3. **Temporizador:** marca el tiempo que tarda el programa en ejecutar la siguiente repetición del bucle "while".
4. **Clavijas:** permiten al usuario elegir a qué pestaña entre todas las disponibles desea acceder: ver datos en tiempo real, consultar el histórico de datos, entrar al simulador, controlar el

interruptor automático, consultar los errores y abrir la pestaña de información del programa. En el caso de ser ejecutadas, abrirán su correspondiente ventana asociada.

5. **Bucle de inicio de recepción de datos:** este bucle finaliza en dos casos: si se pulsa el botón de "back" o si se activa la clavija de recepción de datos, en cuyo caso también se activaría la estructura condicional de recepción vía UDP explicada en el punto 6.
6. **Estructura de recepción:** es un bucle "while" dentro de una estructura condicional, este bucle aloja a los componentes necesarios para que se pueda realizar un envío mediante protocolo UDP; no se explicará este código con más precisión por el momento, ya que se dispone de un punto dedicado expresamente a ello.
7. **Escritura en la base de datos:** el código abrirá la base de datos "login.lvm" y redactará sobre ella las nuevas credenciales.
8. **Reinicio de variables UDP:** en este punto, el código reinicia las variables cada vez que se inicia el programa.

I.5.3.3 Interfaz visual

I.5.3.3.1 Data History Front View

Entorno gráfico

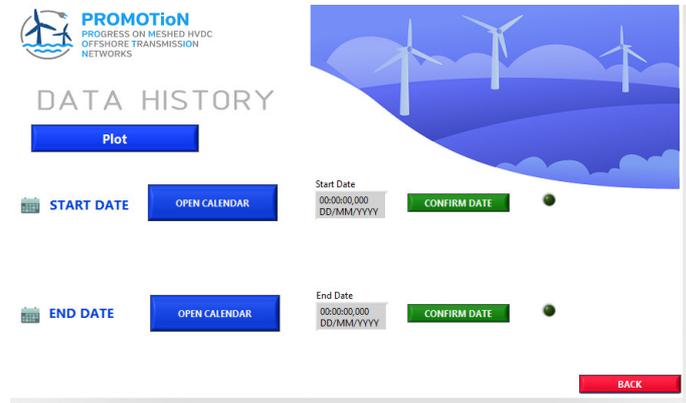


Figura I.5.17: Recorte de la "front view" (1/2)

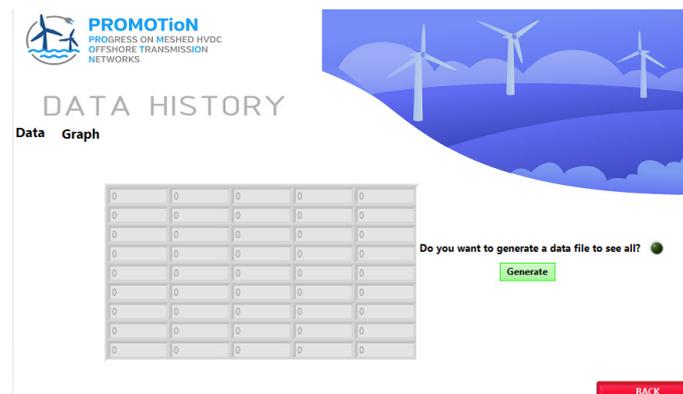


Figura I.5.18: Recorte de la "front view" (2/2)

Descripción

Esta pantalla ofrece al usuario la posibilidad de consultar los datos recibidos por parte del simulador en un intervalo temporal exacto. Para ofrecer estas funcionalidades, la pestaña dispone de dos botones que abren el calendario de inicio y fin del intervalo y dos botones de confirmación con su respectivo LED.

Una vez seleccionado el intervalo, el usuario deberá pulsar el botón de plotear, es ese mismo momento se abrirá una subpestaña cuya función será encontrar los datos dentro de este intervalo consultando la base de datos.

Cuando ya se hayan encontrado, se proyectarán en forma de tabla y gráficamente, ofreciendo al usuario la disponibilidad de descargar estos datos en un fichero.

Elementos operables



Figura I.5.19: Infografía elementos operables (1/3)

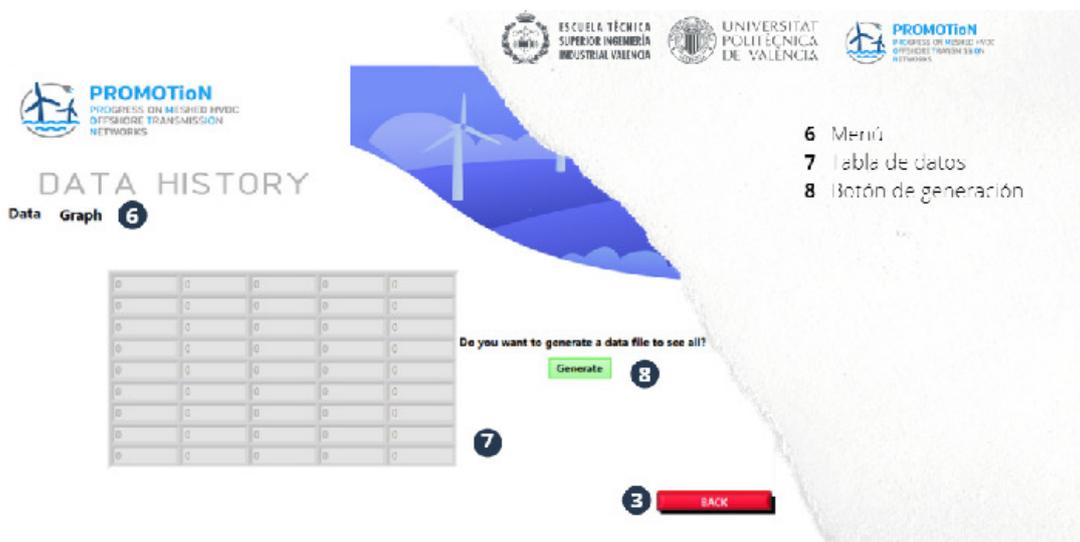


Figura I.5.20: Infografía elementos operables (2/3)

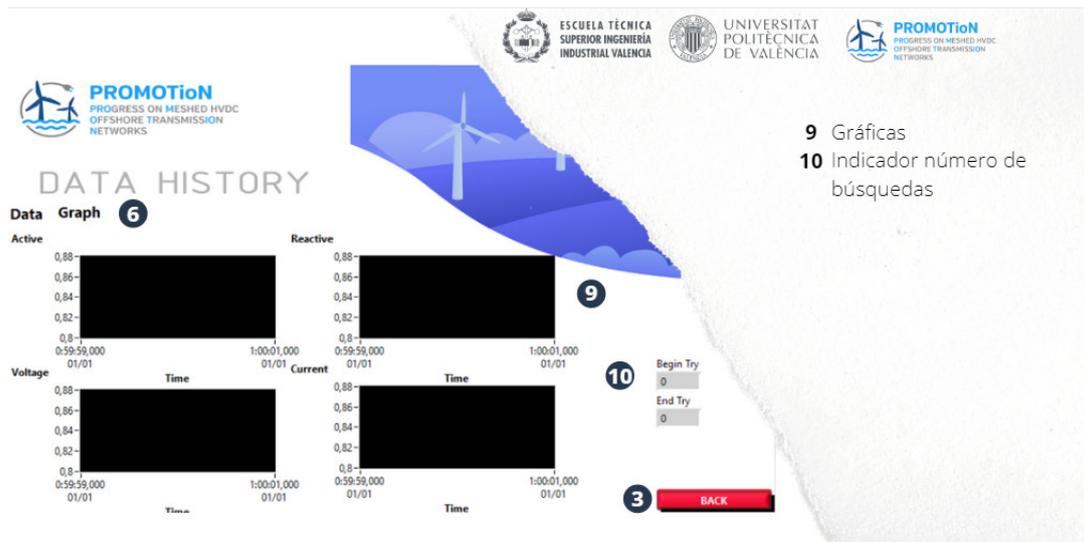


Figura I.5.21: Infografía elementos operables (3/3)

1. **Calendario:** abre la subpestaña que se puede observar en el punto 5.
2. **Clavija confirmación:** confirma la fecha seleccionada en el calendario y enciende el LED de la derecha.
3. **Botón de "back":** de ser pulsado, retrocedería al usuario hacia la ventana de menú principal; puede ser utilizado en el caso de que se desee acceder a otra pestaña del software.
4. **Botón de ploteo:** una vez pulsado, abre la subventana dedicada a encontrar y representar los datos presentes en el intervalo de tiempo seleccionado.
5. **Subpestaña de calendario:** da la posibilidad al usuario de elegir la fecha exacta con una precisión de décimas de segundo. Este aspecto se consigue trabajando con el tiempo en Unix Time Stamp, es decir, con los segundos exactos que han pasado desde el 1 de enero de 1970.
6. **Botones de menú:** permiten al usuario elegir a que pestaña de todas las disponibles desea acceder.
7. **Tabla de valores:** representa los datos encontrados que cumplan las condiciones impuestas por el usuario.
8. **Botón de generación:** ofrece la posibilidad de descargar los valores encontrados en un fichero de texto.
9. **Gráficas:** se representan los valores encontrados para: corriente, intensidad, potencia activa y potencia reactiva.
10. **Número de búsquedas:** la mayoría de las veces en la base de datos no se encuentra la fecha exacta elegida, por ello el programa realizará búsquedas de precisión de un segundo para encontrar la fecha más cercana. En este punto se indican el número de intentos necesarios para encontrar el primer valor.

Entorno de programación

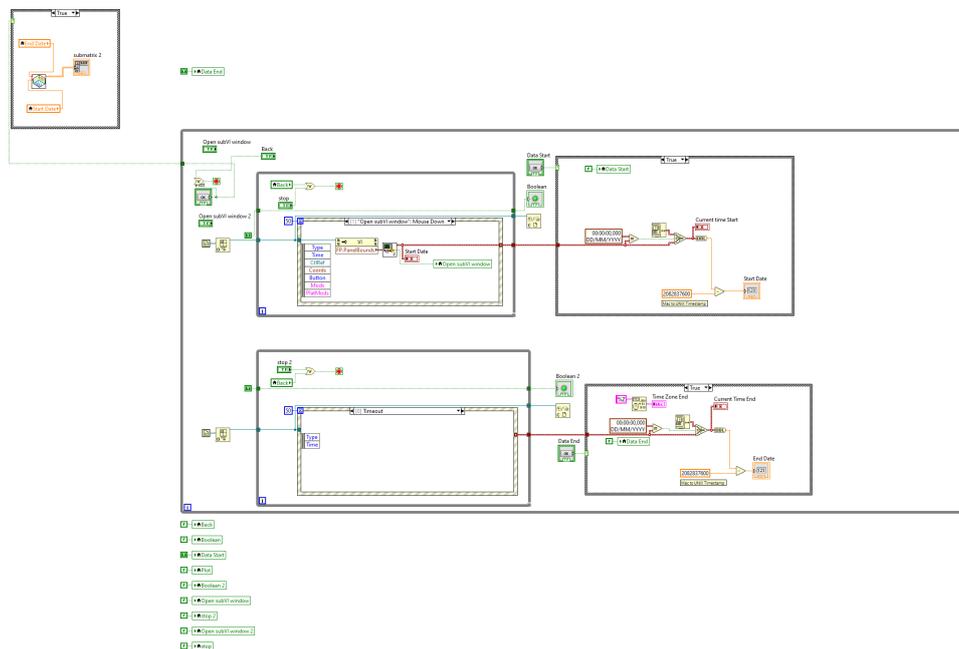


Figura I.5.22: Recorte del "block diagram" (1/2)

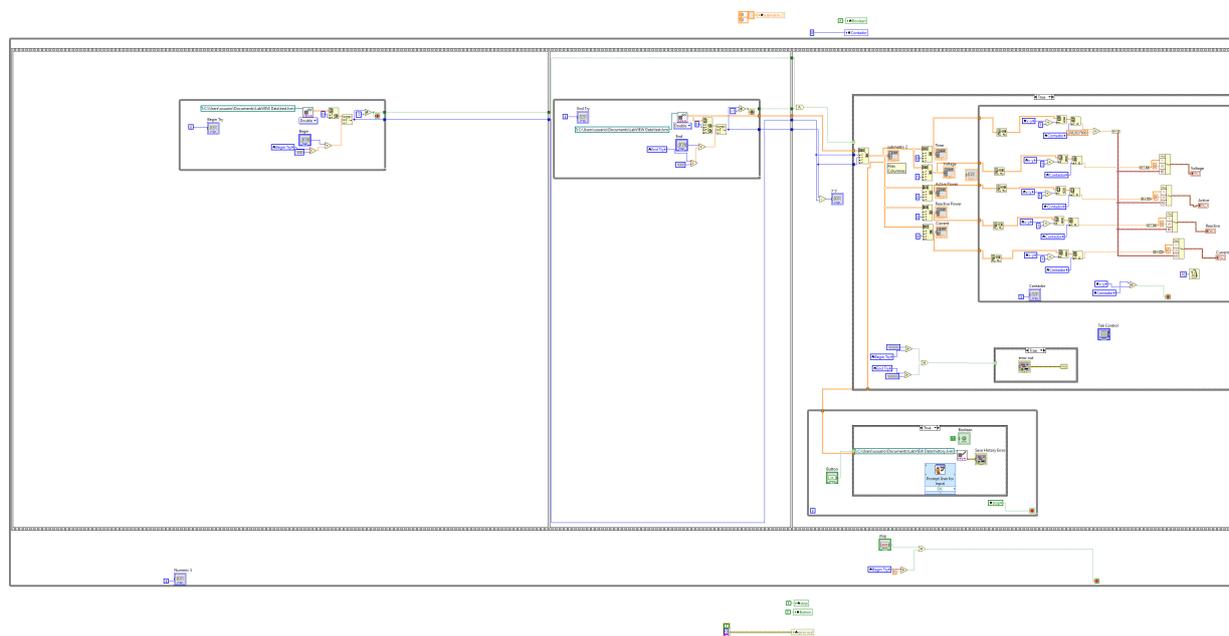


Figura I.5.23: Recorte del "block diagram" (2/2)

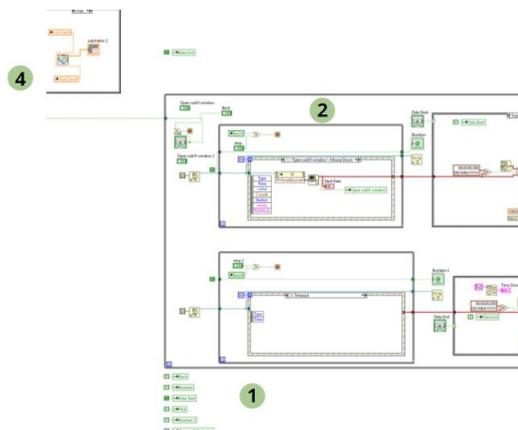
Descripción

Esta funcionalidad del programa esta compuesta por dos subpestañas: la primera de ellas permite seleccionar la fecha y la segunda representa los datos.

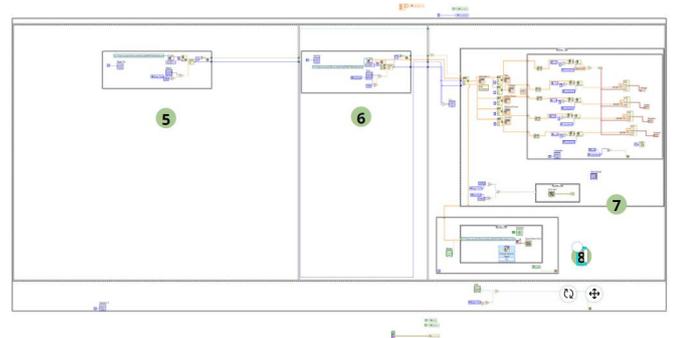
Para la selección de fecha se dispone de un gran bucle "while" dentro del cual se encuentran sendos bloques condicionales que se ejecutarán cuando se seleccione la fecha y se confirme. El gran bucle "while" finalizará cuando se cumplan estos dos parámetros y el usuario pulse sobre el botón de plotear.

En ese momento el programa abrirá la otra pestaña, esta segunda ventana está constituida por una secuencia: dos bucles "while", donde el software busca los datos más cercanos y dos últimos bucles "while", el primero formado por bloques de tratamiento de matrices para el ploteo y el segundo con un condicional para el almacenaje de los datos ploteados. Todo esto dentro de un gran bucle "while" que se ejecutará hasta que el usuario indique lo contrario.

Componentes



(a) Componentes del "block diagram" (1/2)



(b) Componentes del "block diagram" (2/2)

Figura I.5.24: Componentes del "block diagram"

1. **Reinicio de variables:** en este punto el código reinicia las variables cada vez que se inicia el programa.
2. **Bucle "while":** tiene la función de repetir el proceso de forma continuada incluso si se verifica alguna de las estructuras condicionales, hasta que se pulse el botón de "back" o las condiciones indicadas para su finalización.
3. **Estructuras condicionales:** tienen la función de abrir el calendario para permitir al usuario seleccionar la fecha deseada y confirmar esta misma; una vez validada la envían a la segunda pestaña para que pueda ser utilizada.
4. **Apertura de la segunda pestaña:** abre el segundo vi, donde el programa busca y plotea los datos que se sitúen dentro del intervalo temporal seleccionado.
5. **Búsqueda de inicio:** va buscando en la base de datos (hacia delante) hasta que encuentre el primer dato dentro del intervalo, si no se encuentra ninguno cercano, se comunicará al usuario. El grado de libertad para buscar datos puede ser modificado por el programador.

6. **Búsqueda de fin:** va buscando en la base de datos (hacia delante) hasta que encuentre el dato más cercano a la fecha de finalización, si no se encuentra ninguno cercano, se comunicará al usuario. El grado de libertad para buscar datos puede ser, análogamente al anterior, modificado por el programador.
7. **Ploteo:** esta serie de bloques reúne todos los elementos necesarios para el tratamiento de las matrices y el ploteo de los datos.
8. **Salvado de datos:** en el caso de que el usuario decida guardar los datos ploteados, será este bucle el que se ejecutará para permitir dicha funcionalidad.

I.5.3.3.2 Real Time Front View

Entorno gráfico

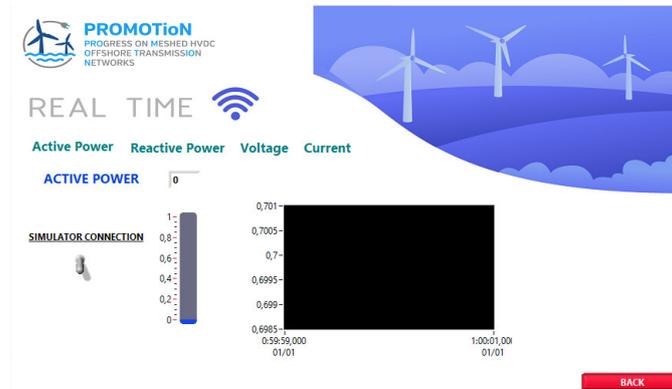


Figura I.5.25: Recorte de la "front view"

Descripción

La pestaña de recepción de datos en tiempo real ofrece la posibilidad al usuario de navegar entre los distintos parámetros y observar los datos recibidos mediante distintos tipos de indicadores: numéricos, gráficos y barras numéricas.

Como en el resto de las pestañas, se dispone de un botón de retroceso que permite al usuario volver a la pestaña de menú principal.

Elementos operables

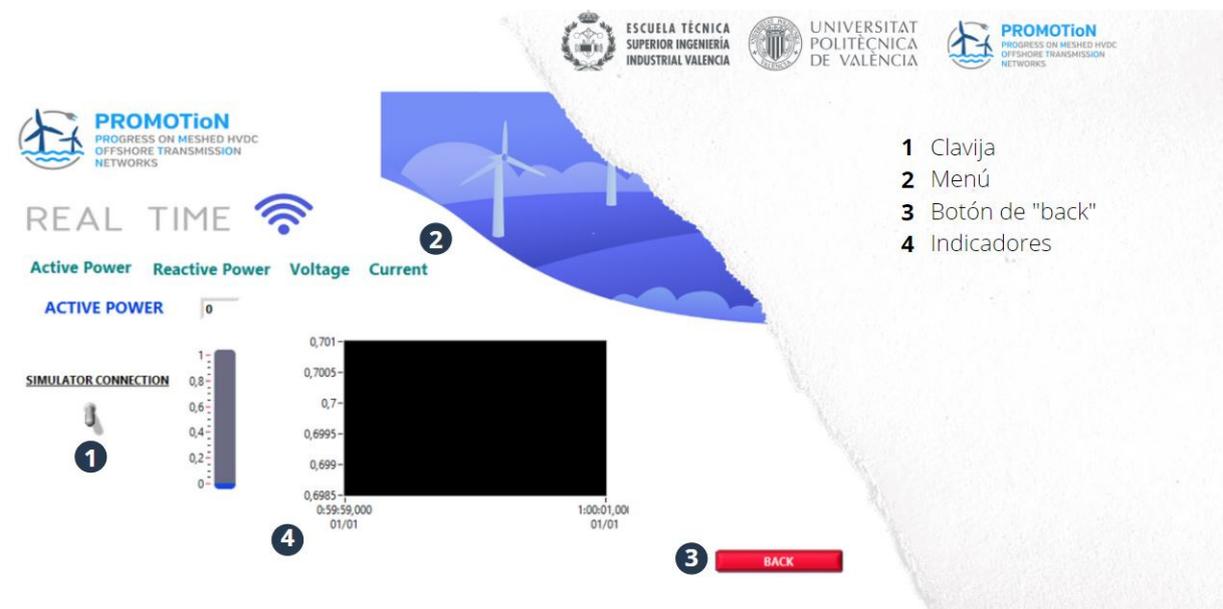


Figura I.5.26: Infografía elementos operables

1. **Clavija de conexión:** habilita al programa a conectarse contra el simulador (o el prototipo a escala) y empezar a recibir datos vía UDP en tiempo real.

2. **Botones de menú:** permiten al usuario elegir a qué pestaña de todas las disponibles desea acceder.
3. **Indicadores:** habilita al usuario a visualizar los datos recibidos.
4. **Botón de "back":** de ser pulsado, retrocedería al usuario hacia la ventana de menú principal; puede ser utilizado en el caso de que se desee acceder a otra pestaña del software.

Entorno de programación

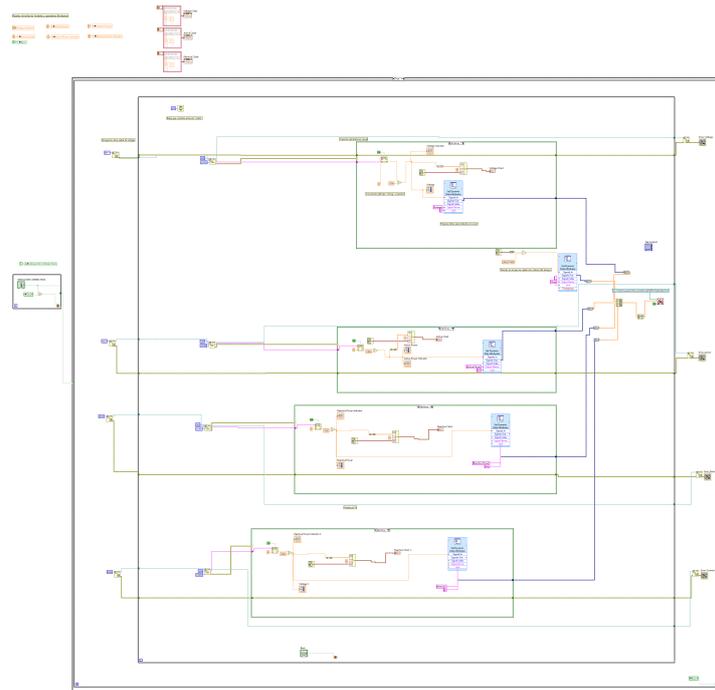


Figura I.5.27: Recorte del "block diagram"

Descripción

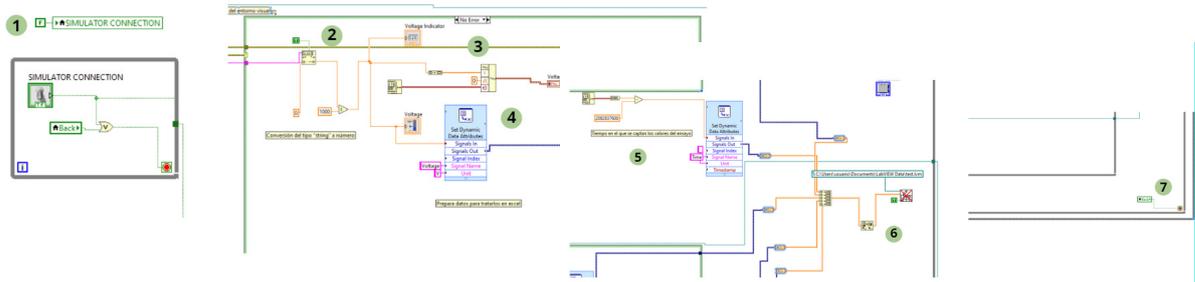
El entorno de programación de esta ventana cuenta con un bucle "while" dentro del cual se halla la clavija de conexión con el simulador.

Una vez conectada, se validará la estructura de casos. Dentro de esta se dispone otro bucle "while" que ejecuta el conjunto de bloques de recepción de datos de forma continuada hasta que se pulse el botón de retroceso.

La recepción de datos consta de un bloque de apertura del protocolo de envío UDP donde se asigna el puerto local a utilizar, un bloque de lectura que recibe la señal, una serie de bloques de tratamiento de señal y un bloque de cierre del protocolo. Como en el resto de las ventanas se observan también unos indicadores de errores que serán utilizados por la pestaña de errores.

Componentes

1. **Bucle de inicio de recepción de datos:** este bucle finaliza en dos casos: si se pulsa el botón de "back", o si se activa la clavija de recepción de datos, en cuyo caso también se pondría en marcha la estructura condicional de recepción vía UDP.



(a) Componentes del "block diagram" (1/2)

(b) Componentes del "block diagram" (2/2)

Figura I.5.28: Componentes del "block diagram"

2. **Recepción de datos:** este conjunto de bloques tiene el objetivo de recibir los datos, modificarlos de tipo "string" a tipo numérico y restar la ganancia aplicada antes de proceder al envío.
3. **Indicadores:** permiten al usuario visualizar gráficamente los datos recibidos.
4. **Preparación de datos para almacenar:** prepara los datos recibidos para poder ser almacenados posteriormente en la base de datos.
5. **Tiempo:** administra la constante de tiempo para ser almacenada correctamente.
6. **Escritura de datos:** escribe los datos en la base de datos "test.lvm".
7. **Botón de "back":** finaliza el bucle "while" cerrando a su vez la pestaña.

I.5.3.3.3 Breaker Control Front View

Entorno gráfico

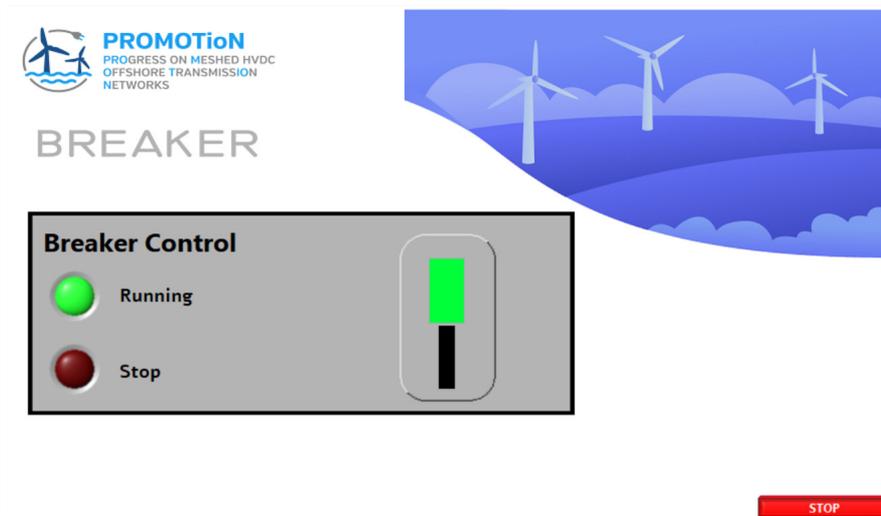


Figura I.5.29: Recorte de la "front view"

Descripción

Esta ventana permite al usuario obtener el control total del interruptor automático del simulador o prototipo a escala; pulsando tan solo un botón podrá parar en cualquier momento el trabajo de la otra máquina remota.

Elementos operables

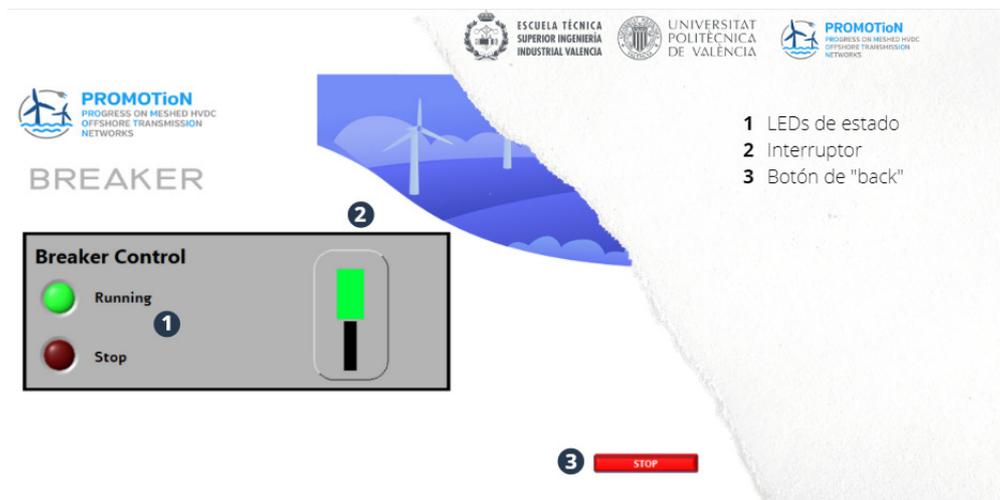


Figura I.5.30: Infografía elementos operables

1. **LEDs de estado:** indican el estado del interruptor automático, abierto si está encendido el LED rojo y cerrado si lo está el de color verde.
2. **Interruptor:** habilita al usuario a conectar o desconectar la circuitería digital de forma remota.

3. **Botón de "back"**: de ser pulsado, retrocedería al usuario hacia la ventana de menú principal; puede ser utilizado en el caso de que se desee acceder a otra pestaña del software.

Entorno de programación

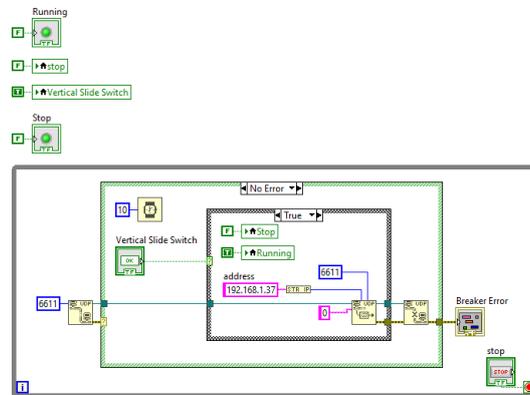


Figura I.5.31: Recorte del "block diagram"

Descripción

El código de programación de esta ventana cuenta con un reinicio de las variables (incluidos los dos LED) y un bucle "while" que, de forma análoga a los anteriores, se reproduce hasta que se valida la condición de salida.

Dentro de este bucle se encuentra un envío vía protocolo UDP; primero se abre el protocolo y se asigna un puerto (en este caso el 6611). Si este proceso se realiza sin problemas, se validará la estructura condicional. Dentro de ella se halla el interruptor, y según su posición, se abrirá una u otra parte de la estructura condicional, enviando un 1 si se ha abierto el circuito o un 0 si, por el contrario, está cerrado.

En el interior de esta última estructura se dispone el envío que se realizará después de introducir el puerto de destino y la dirección (la cual habrá que modificar de dato numérico a "string").

Componentes

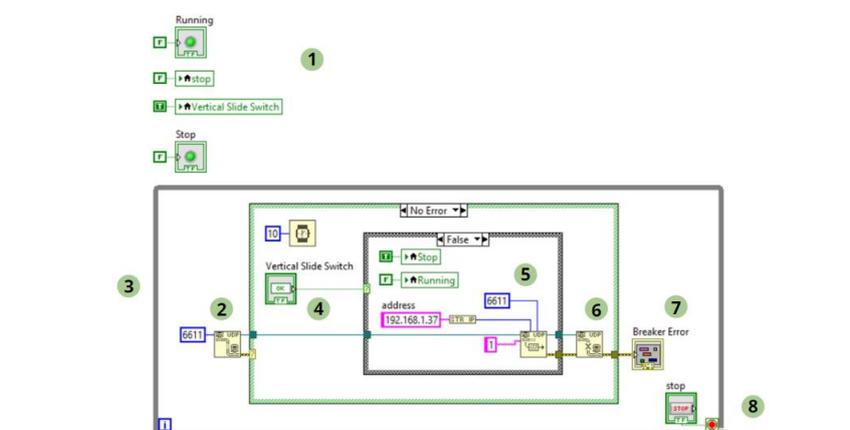


Figura I.5.32: Componentes del "block diagram"

1. **Reinicio de variables**: en este punto el código reinicia las variables cada vez que se inicia el programa.

2. **Apertura envío UDP:** se inicia el protocolo indicando el puerto local a utilizar.
3. **Bucle "while":** tiene la función de repetir el proceso de forma continuada incluso si se verifica alguna de las estructuras condicionales, hasta que se pulse el botón de "back".
4. **Interruptor:** su posición dictamina la apertura de una parte u otra del código dentro de la estructura condicional.
5. **Estructura condicional:** establece los parámetros de los LED según la casuística y envía el mensaje al otro operador de la comunicación. Este envío se realizará vía la función de envío mediante protocolo UDP, se indicará la dirección IP y el puerto destino; así como el mensaje a enviar.
6. **Cierre del protocolo UDP:** como su propio nombre indica, su función es cerrar la comunicación vía UDP. Se puede deducir más acerca de esta función en la figura ??.
7. **Error:** indica un posible error en la comunicación; este indicador será utilizado por la pestaña de "Error Log".
8. **Botón de "back":** finaliza el bucle "while" cerrando a su vez la pestaña.

I.5.3.3.4 Error Log Front View

Entorno gráfico

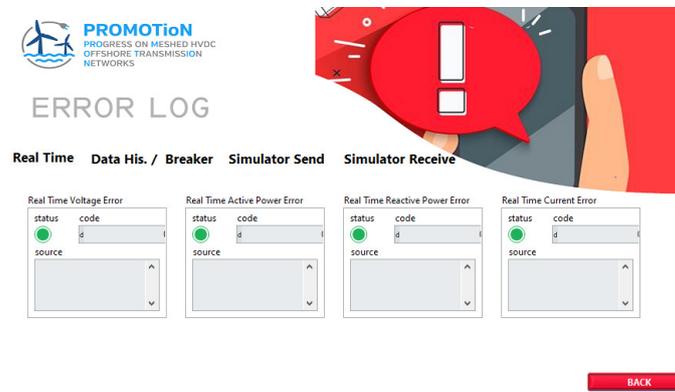


Figura I.5.33: Recorte de la "front view"

Descripción

La ventana en cuestión permite al usuario del programa consultar los errores que pudieran derivarse del empleo de cualquiera de las otras pestañas del programa: "Real Time", "Data History", "Breaker" y "Simulation".

Cuando exista algún error, este se notificará encendiendo el LED de "status" en rojo e indicando el código y una breve explicación acerca del fallo en cuestión para que el usuario o, en su defecto, el programador pudiesen solventarlo.

Se incluye, además, un "tap control" para facilitar la navegación y la visualización de esta "front view".

Elementos operables



Figura I.5.34: Infografía elementos operables

1. **Menú:** este "tap control" permite al usuario navegar entre los distintos tipos de errores con tan solo un "clic", sin necesidad de cambiar de pestaña de trabajo y a máxima velocidad.
2. **Pantalla de error:** existe una pantalla de error para cada posible error que se pudiera dar a lo largo de la ejecución del programa. Las pantallas constan de tres elementos: el "status", que es un LED que puede adoptar dos colores (verde y rojo), el código del error y una breve descripción del mismo.

Entorno de programación

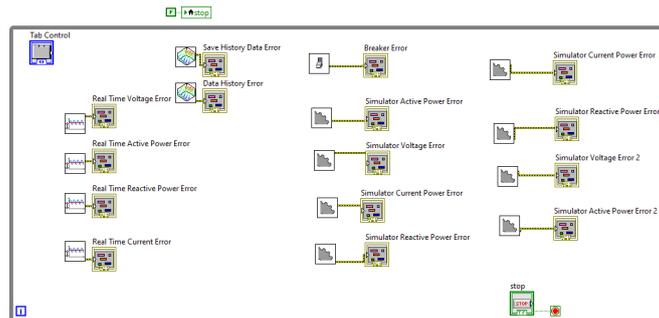


Figura I.5.35: Recorte del "block diagram"

Descripción

El entorno de programación de la pestaña de errores no es más que una especie de "repositorio" donde se agrupa toda la información sobre errores presente en el resto de las pestañas, y que se ha hecho invisible por motivos de comodidad.

Por lo tanto, se dispondrá de un bucle "while" que se repetirá mientras no se cumpla la condición de salida y, dentro de él, se encontrarán las llamadas a los errores de las diferentes ventanas junto a un indicador gráfico para cada una de ellas.

Componentes

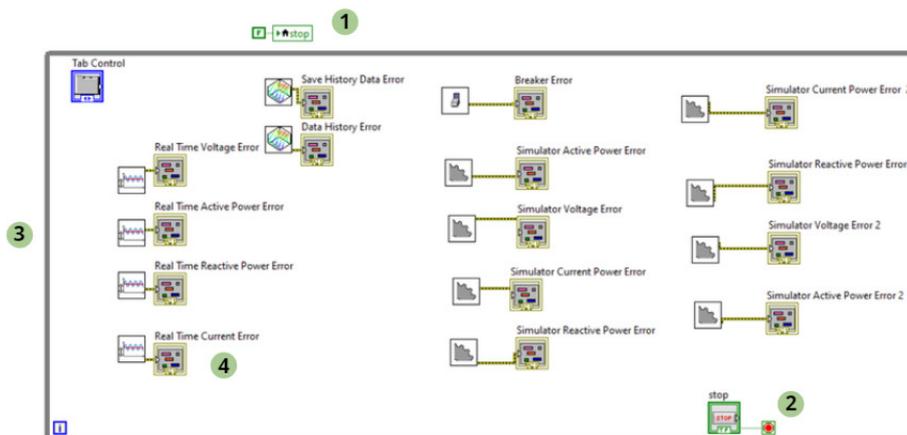


Figura I.5.36: Componentes del "block diagram"

1. **Reinicio de variables:** en este punto el código reinicia las variables cada vez que se inicia el programa.

2. **Botón de "back"**:finalizará la ventana en cualquier momento, sacando al código del bucle "while".
3. **Bucle "while"**: su función es repetir el programa hasta que se pulse el botón de retroceso, en ese momento devolvería al usuario a la ventana de "Main Menu".
4. **Llamadas a los vi**: llama a cada pestaña del programa para consultar el estado de sus errores y representarlos.

I.5.3.3.5 Help Front View

Entorno gráfico



Figura I.5.37: Recorte de la "front view"

Descripción

Esta ventana transmite al usuario más información acerca del programa: programadores, entidades colaboradoras, etc.

Se compone, a su vez, de dos botones: el primero de ellos abre una pestaña que conduce al usuario a la página web del proyecto PROMOTioN, mientras que el segundo abre el enlace del repositorio donde el cliente puede descargar el manual de usuario del programa.

Elementos operables



Figura I.5.38: Infografía elementos operables

1. **Clavija de información del proyecto:** abre un enlace en el navegador que lleva al usuario hasta la página web del proyecto matriz del que nace este Trabajo de Final de Grado: <https://www.promotion-offshore.net/>.
2. **Clavija de manual de usuario:** de forma completamente análoga a la clavija anterior, abre un enlace dentro del navegador que conduce al usuario hasta el repositorio creado para

este programa, donde podrá descargar el "Manual de Usuario" en cualquiera de sus múltiples idiomas.

3. **Botón de "back"**: de ser pulsado, retrocedería al usuario hasta la ventana de menú "Main Menu"; puede ser utilizado en el caso de que se desee cambiar de funcionalidad.
4. **Información sobre los creadores**: recoge la información esencial relacionada con los creadores del programa.
5. **Información sobre los colaboradores**: recoge la información esencial relacionada con las entidades colaboradoras.

Entorno de programación

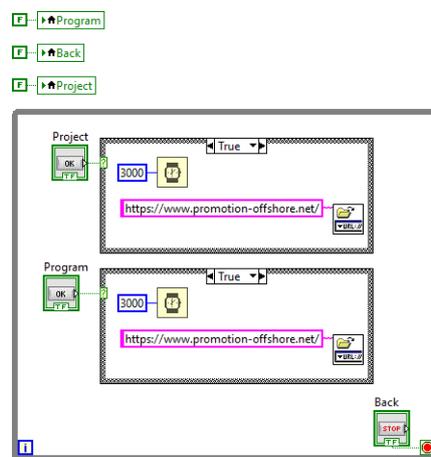


Figura I.5.39: Recorte del "block diagram"

Descripción

Este es, probablemente, el código más sencillo que se encontrará dentro del software presentado. Cuenta con un reinicio de variables que actúa de forma semejante a los anteriores y un bucle "while".

Este bucle se repetirá siempre y cuando el botón de retroceso no esté pulsado; dentro del "while" el usuario dispone de dos opciones: consultar la web del proyecto o acceder al repositorio del programa.

La temporización de estos bloques condicionales se realiza debido a que la acción mecánica de las clavijas es un "switch until released" y de no existir esta temporización, se abrirían muchas pestañas de búsqueda.

Componentes

1. **Reinicio de variables**: en este punto el código reinicia las variables cada vez que se inicia el programa.
2. **Clavijas**: permiten al usuario elegir si desea consultar la web del proyecto o el repositorio del programa; dichas clavijas validan las estructuras condicionales mencionadas en el punto 4.

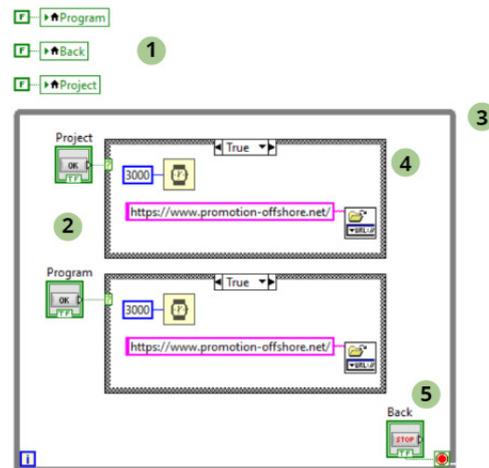


Figura I.5.40: Componentes del "block diagram"

3. **Bucle "while":** su función es repetir el programa hasta que se pulse el botón de retroceso, en ese momento devolvería al usuario a la ventana de "Main Menu".
4. **Estructuras condicionales:** de validarse, abrirían la url correspondiente a cada una.
5. **Botón de "back":** finalizará la ventana en cualquier momento, sacando al código del bucle "while".

I.5.3.3.6 Simulation Front View

Entorno gráfico

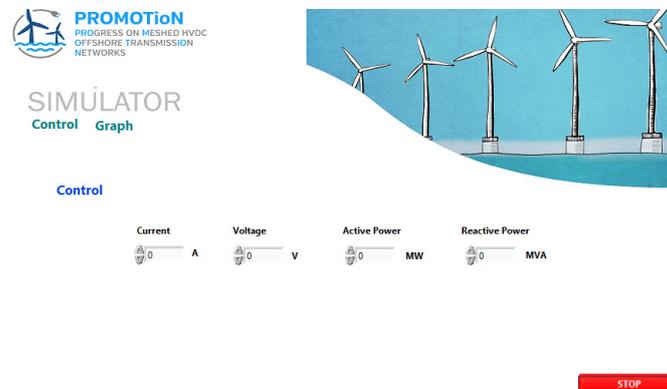


Figura I.5.41: Recorte de la "front view"

Descripción

Esta ventana permite al usuario realizar una simulación contra el prototipo a escala o, en su defecto, contra el archivo de banco de pruebas generado con código de programación de MATLAB.

El usuario tiene la posibilidad de enviar cualquiera de los siguientes parámetros: tensión, intensidad, potencia activa y potencia reactiva. El envío se realizará mediante protocolo UDP.

El programa tiene también la capacidad de recibir los datos del propio banco de pruebas y ofrecer al usuario la posibilidad de plotear uno de ellos o todos.

Elementos operables

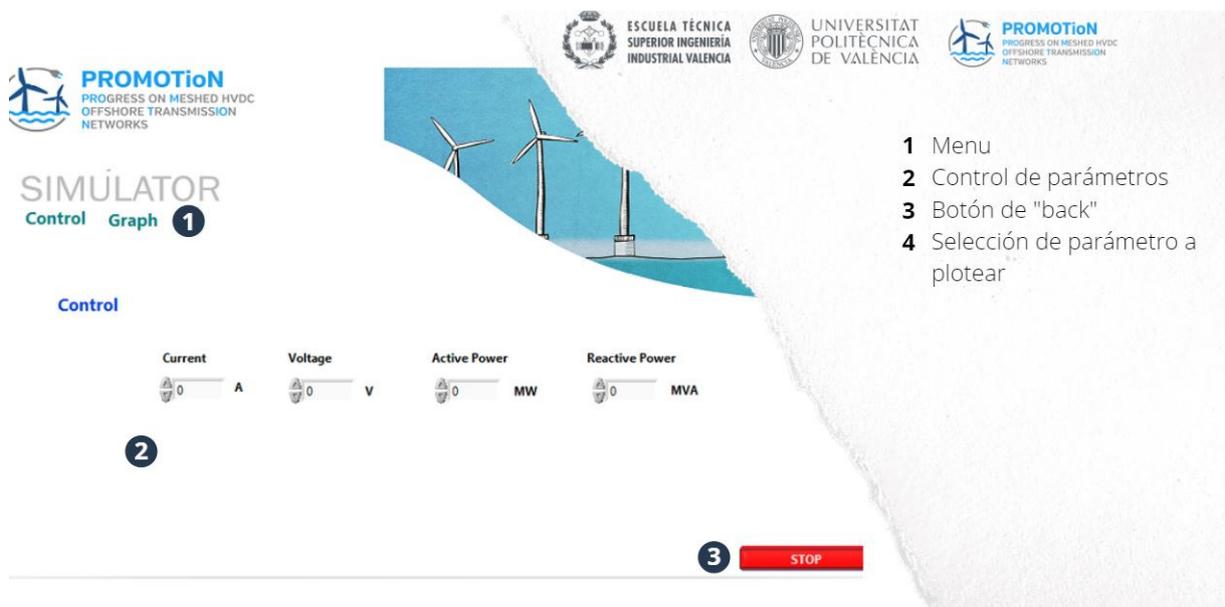


Figura I.5.42: Infografía elementos operables (1/2)

1. **Botones de menú:** permiten al usuario elegir a qué pestaña de todas las disponibles desea acceder.

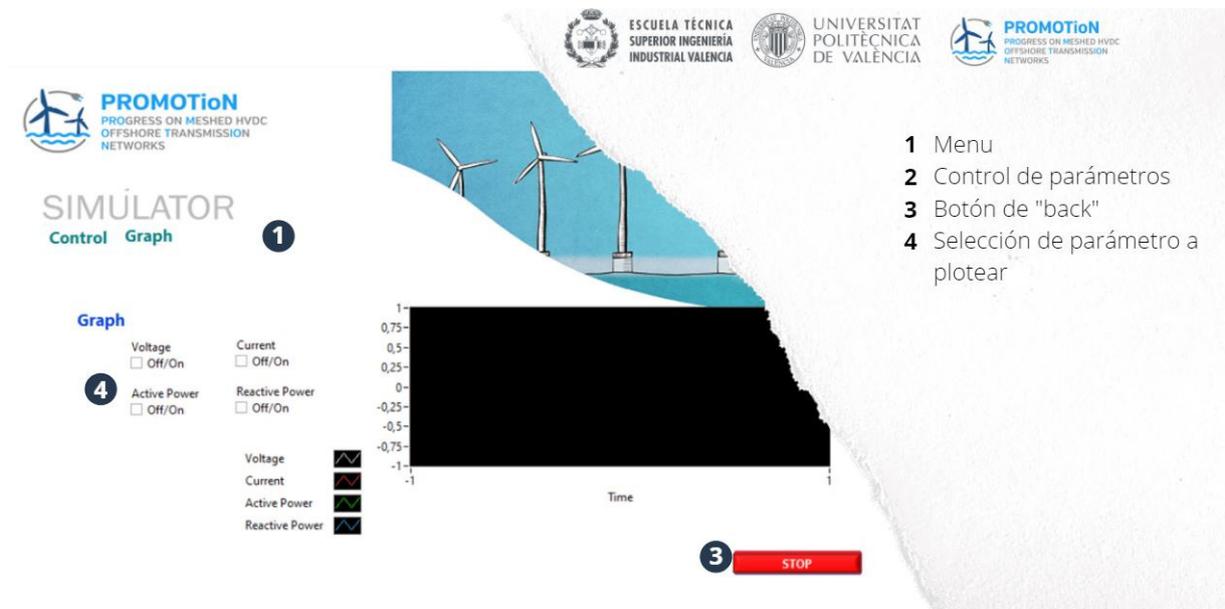


Figura I.5.43: Infografía elementos operables (2/2)

2. **Control de parámetros:** habilita al usuario a introducir los datos que desee para cada parámetro de control.
3. **Botón de "back":** de ser pulsado, retrocedería al usuario hasta la ventana de registro; puede ser utilizado en el caso de que se quiera finalizar el programa y/o cambiar las credenciales con las que se ha accedido.
4. **Selección de parámetro a plotear:** el usuario podrá elegir que datos quiere que se ploteen; justo bajo de la gráfica se dispone una leyenda que especifica el color asignado a cada variable.

Entorno de programación

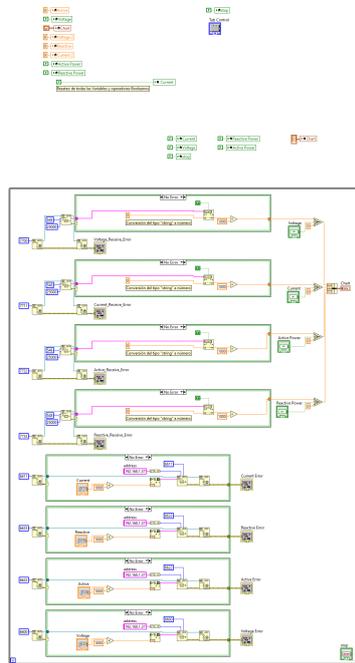


Figura I.5.44: Recorte del "block diagram"

Descripción

Este "block diagram" reúne aspectos ya utilizados en las pestañas de "breaker" y de "real time". Se trata de un gran bucle "while", dentro del cual se desarrollan dos tareas simultáneas: el envío de datos y la recepción de datos vía UDP. Para ello, se seguirá el esquema típico de "apertura, envío y cierre" o "apertura, recepción y cierre", según el caso.

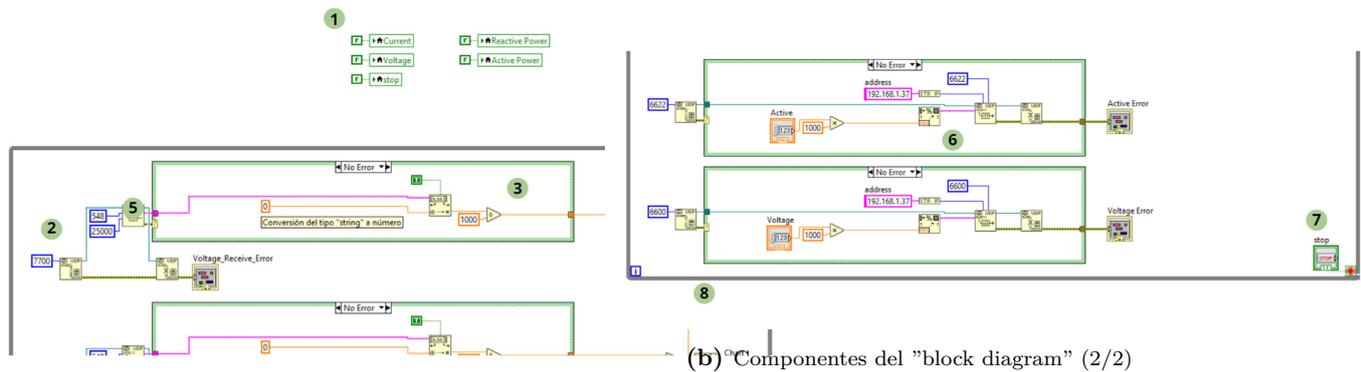
Tanto en el envío como en la recepción se dispone de unos indicadores de error, utilizado por la página de "Error Log"; por su parte, la recepción continúa con un sistema de ploteo.

Se ha empleado un operador booleano para ofrecer al usuario la posibilidad de elegir si se plotea o no la variable en cuestión, acompañado de una gráfica para visualizarlas los mismos.

Al igual que en el resto de las pestañas, una clavija de retroceso deja abierta para el usuario la posibilidad de volver al menú principal en todo momento

Componentes

1. **Reinicio de variables:** en este punto el código reinicia las variables cada vez que se inicia el programa.
2. **Apertura envío UDP:** se inicia el protocolo indicando el puerto local a utilizar.
3. **Recepción de datos:** este conjunto de bloques tiene el objetivo de recibir los datos, modificarlos de tipo "string" a tipo numérico y restar la ganancia aplicada antes de proceder al



(a) Componentes del "block diagram" (1/2)

Figura I.5.45: Componentes del "block diagram"

envío. Las ganancias durante todo el programa son de 1000 o 0.001 según sean de envío o recepción.

4. **Clavijas:** su posición dictamina la apertura de una parte u otra del código dentro de la estructura condicional.
5. **Chart:** reúne todas las variables que el usuario ha decidido representar y las plotea en la gráfica.
6. **Estructura de envío:** tiene una estructura muy parecida a la de recepción, se abre el protocolo UDP asignando un puerto local, se especifica la dirección IP y el puerto destino y se le aplica la ganancia a la variable antes de convertirla a "string" y enviarla.
7. **Botón de "back":** finaliza el bucle "while", cerrando a su vez la pestaña.
8. **Bucle "while":** tiene la función de repetir el proceso de forma continuada incluso si se verifica alguna de las estructuras condicionales, hasta que se pulse el botón de "back".

I.5.4 Código en MATLAB

En MATLAB se dispone de tres códigos diferentes:

I.5.4.1 *Simulador*

El simulador se basa en un programa creado para la investigación realizada por parte de la UPV, en el que se han realizado una serie de modificaciones para que los datos que produce puedan ser enviados al sistema SCADA programado.

En la figura I.5.46 se pueden observar las modificaciones realizadas en este código:

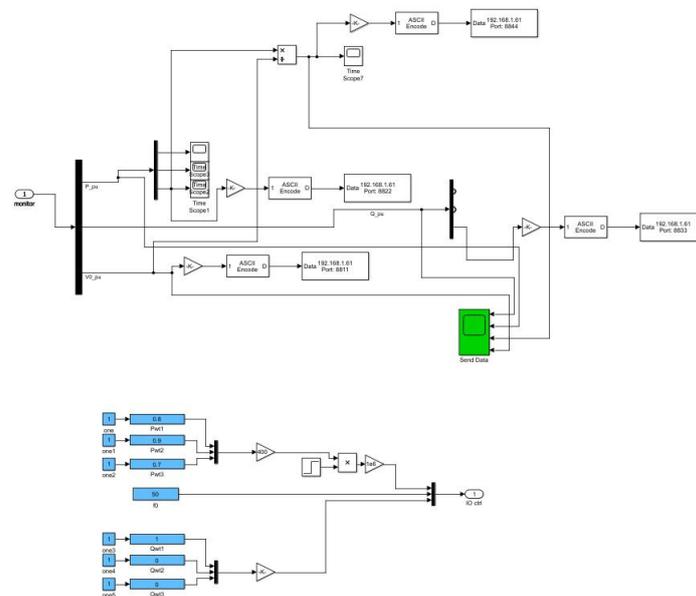


Figura I.5.46: Recorte del código de simulador

I.5.4.2 Banco de pruebas

Banco de Pruebas es un código creado de cero, su propósito es leer los datos recibidos por parte del sistema SCADA, realizar un pequeño tratamiento (simulando un sistema real) y enviarlos.

A lo largo del código, se le añaden una serie de gráficas para comprobar su correcto funcionamiento en todo momento.

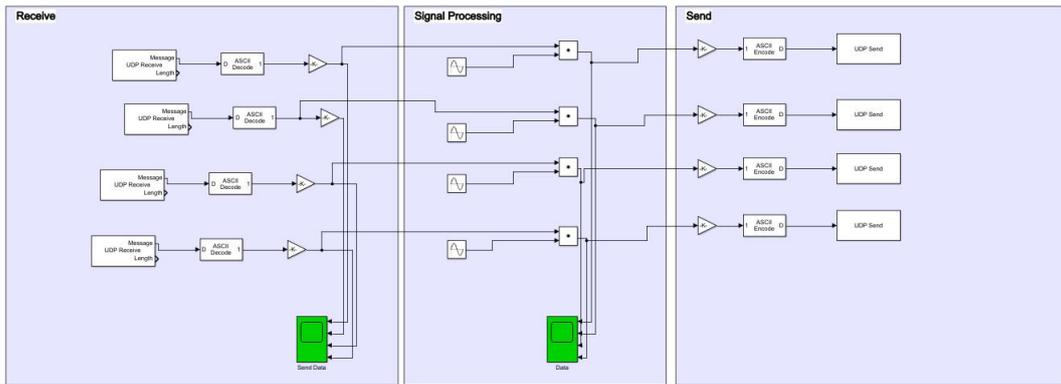


Figura I.5.47: Recorte del código de banco de pruebas

I.5.4.3 Breaker

Este es un código simple que tiene como objetivo visualizar un cambio de dato recibido por parte del sistema SCADA. Trabaja como una simplificación de lo que sería un interruptor automático.

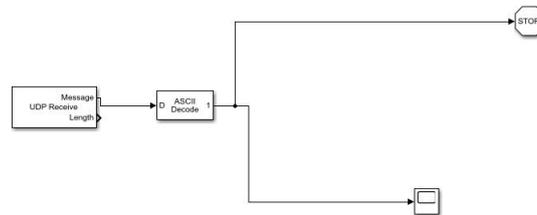


Figura I.5.48: Recorte del código de elementos de protección

I.5.5 Posibles mejoras

Algunas de las posibles mejoras a implementar en el programa se detallan a continuación:

- Generar un proceso de confirmación junto al envío de correo cuando el usuario se registra en el programa.
- Enviar un SMS al usuario en el caso de que exista algún error grave en el sistema.
- Generar una base de datos con la información enviada y recibida por la ventana de simulación.
- Introducir el vídeo explicativo del programa dentro de la pestaña de ayuda.

Capítulo I.6

Verificación y validación

El proceso de verificación y validación del programa es, con total seguridad, uno de los apartados que más dificultad entraña.

Se entiende como verificación y validación el conjunto de procesos de comprobación y análisis que aseguran que el software que se desarrolla está acorde a su especificación y cumple las necesidades de los clientes.

A lo largo del presente punto se explicará la diferencia entre verificación y validación, se expondrán las técnicas empleadas y se defenderá la validez final del producto.

Índice

I.6.1 Verificación vs. Validación	89
I.6.2 Técnicas	89
I.6.3 Proceso de depuración	89
I.6.4 Listado de fallos	91
I.6.5 Pruebas funcionales: caja negra	92
I.6.6 Pruebas de integración incremental	92

I.6.1 Verificación vs. Validación

Aunque muy semejantes a primera vista, estos dos términos no significan exactamente lo mismo cuando se acompañan de la palabra software.

En 1979, Theobald Boehm expresó la diferencia entre ambos vocablos:

- **Verificación:** el objetivo de la verificación es comprobar que el software está de acuerdo con las especificaciones requeridas inicialmente. Para ello, se debe comprobar que el programa cumple con los requerimientos funcionales y no funcionales demandados.
- **Validación:** Boehm definía este proceso como algo más general, entendiendo que un software es válido cuando cumple con las expectativas del cliente.

Un ejemplo para lo anteriormente expuesto podría ser una máquina de compra de comida, en su proceso de verificación se tendrían que cumplir aspectos como tiempo de respuesta, temperatura interior, etc. Por otro lado, si se habla de validación, probablemente los requerimientos del cliente sean más generales, como que la máquina no se equivoque de producto o que devuelva el cambio correctamente.

I.6.2 Técnicas

Dentro del proceso de verificación y validación se encuentran las siguientes técnicas de comprobación:

1. **Inspecciones de software:** donde se analiza y comprueba el funcionamiento del sistema, el código fuente de este o los diagramas de diseño. Esta es una técnica de validación estática y se ejecuta sobre todas las etapas del proceso de desarrollo.
2. **Pruebas de software:** es una técnica dinámica de verificación, ya que requiere disponer de un prototipo ejecutable del sistema. Su función reside en implementar ciertos datos y analizar cómo respondería el programa. En el caso del presente programa, esta técnica ha sido muy utilizada sobre todo en los códigos que requerían de base de datos, utilizándose una técnica de "prueba y error" y adaptando las bases de datos a las necesidades del programa.

Las flechas indican las fases del proceso donde se utilizan las técnicas. Las inspecciones de software pueden realizarse en todas las etapas, mientras que las técnicas de pruebas solo pueden ser utilizadas con un prototipo ya en mano I.6.1.

I.6.3 Proceso de depuración

La depuración es el proceso de eliminación de los errores que se descubren durante las fases de prueba. Es un proceso complicado porque no siempre los errores se detectan cerca del punto donde se originaron.

Para evitar dicho inconveniente, LabVIEW permite ir introduciendo "probe" en los diagramas de bloques de cada ventana, esto permite ir rastreando el error de una forma infinitamente más sencilla. En la figura I.6.2 se recoge un ejemplo de esta técnica.

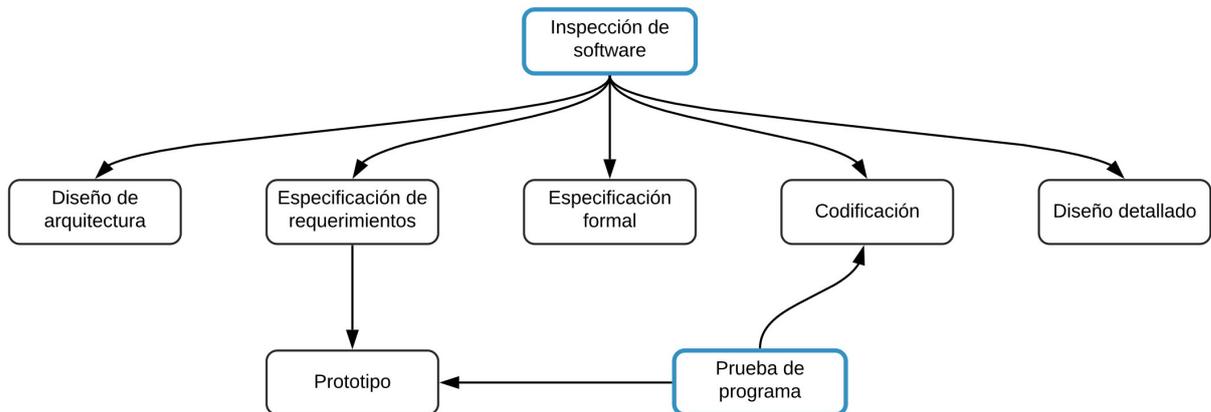


Figura I.6.1: Esquema de las técnicas de comprobación

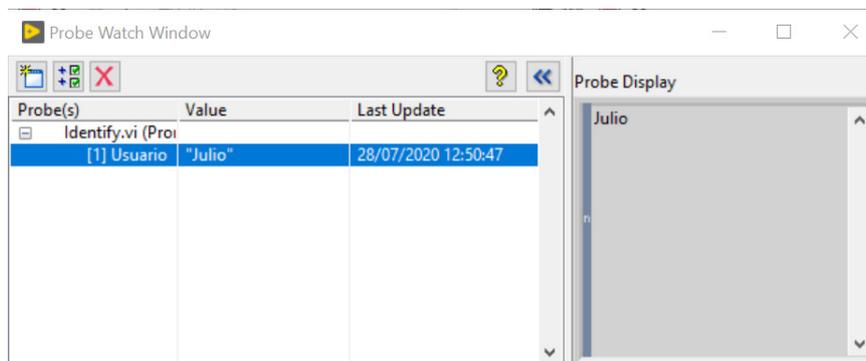


Figura I.6.2: Recorte de LabVIEW

Las herramientas de depuración forman habitualmente parte de las herramientas de apoyo al lenguaje y que sirven de base al compilador. A la hora de crear un ejecutable con LabVIEW, National Instruments incorpora su propia depuradora. Esto facilita de forma notable el trabajo del programador.

Después de que se descubra el origen del fallo en el programa, este debe corregirse y entonces reevaluar el sistema. Esto implica repetir de nuevo las pruebas anteriores (pruebas de regresión), como se puede observar en la figura I.6.4.

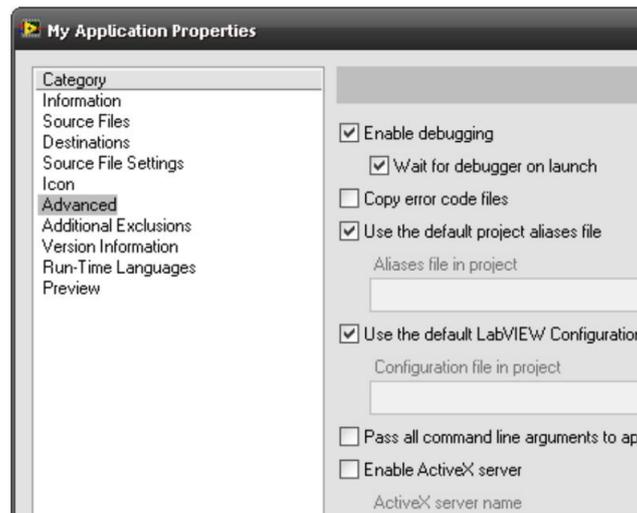


Figura I.6.3: Recorte creación ejecutable

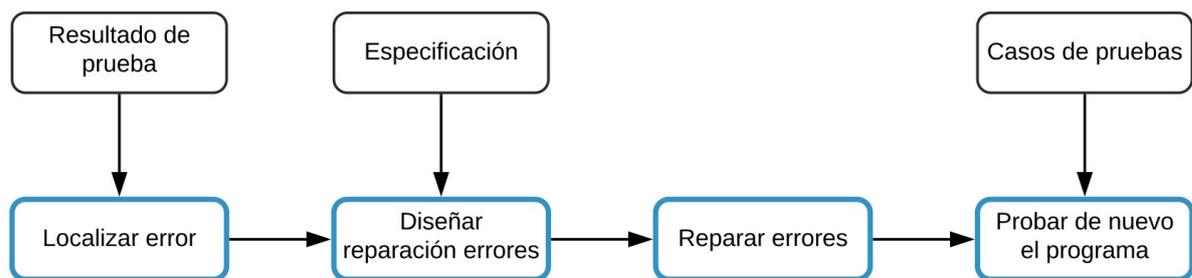


Figura I.6.4: Flujograma corrección errores

I.6.4 Listado de fallos

El proceso de inspección siempre se realiza utilizando una lista de los errores comunes de los programadores. Esta se somete a discusión por el personal con experiencia y se actualiza frecuentemente según se vaya teniendo experiencia.

En este caso la lista se compone de los siguientes puntos:

- **Fallos de datos:**

- ¿Las variables se inician antes de que se utilicen los valores?
- ¿Las variables no se reinician a tiempo?
- ¿Todas las variables y/o constantes están nombradas?
- ¿Se realiza correctamente la decodificación del mensaje?
- ¿Se agrupan correctamente las cadenas de valores?
- ¿Se utiliza el formato correcto para cada variable?

- **Fallos de control:**

- ¿Es correcta cada decisión condicional?
- ¿Todos los ciclos finalizan o tienen la posibilidad de hacerlo?
- ¿Se han completado todos los casos en los bloques "case"?
- ¿Existe código no alcanzable?
- ¿El bucle "while" tiene la posibilidad de finalizar?

- **Fallos de entrada y salida:**

- ¿Se utilizan todas las variables de entrada?
- ¿Se representan y/o archivan todas las variables de salida?

- **Fallos de la gestión del almacenamiento:**

- ¿Se asigna correctamente el espacio?
- ¿Se libera correctamente la memoria?

Todas estas preguntas se han evaluado de forma repetida a lo largo del trabajo de programación para pulir el resultado final.

I.6.5 Pruebas funcionales: caja negra

Otra de las pruebas utilizadas ha sido la denominada como "caja negra". Las pruebas funcionales o de caja negra son una estrategia para seleccionar las pruebas de fallos basándose en las especificaciones de los componentes y programas, para ello, se introducen datos que se sabe que van a causar error en el sistema para analizar su comportamiento.

I.6.6 Pruebas de integración incremental

Estas han sido el último tipo de pruebas implementadas a lo largo del desarrollo del presente trabajo de fin de grado. Consisten en analizar la respuesta de grupos de módulos interconectados a fin de detectar fallos en la interacción entre ellos.

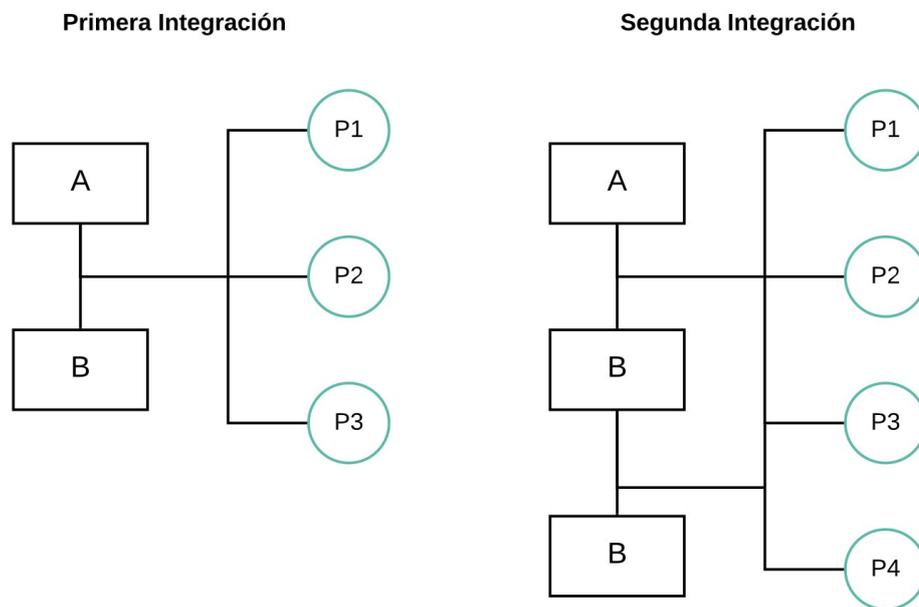


Figura I.6.5: Esquema de las pruebas de integración

Para llevarlo a cabo, se divide el proceso en secuencias de pruebas, por ejemplo: "Log-In" + "Sign-Up"; "Log-In" + "Sign-Up" + "Main Menu"; etc. Este proceso se puede comprobar de forma gráfica en la figura I.6.5

Capítulo I.7

Resultados finales, conclusiones, visión futura y ampliaciones

Las mejoras en casi cualquier ámbito científico son prácticamente infinitas; en este sentido, la mejora continua como ciencia se plantea como una herramienta fundamental para que cualquier software no se convierta en un código obsoleto y estéril.

El caso del sistema desarrollado a lo largo de estas páginas no es una excepción. Durante el presente capítulo se expondrán algunas sugerencias para introducir avances y perfeccionar las competencias ya programadas en el código y otros conceptos para futuras ampliaciones del software.

A lo largo de este capítulo se presentarán también los resultados finales del proyecto y se realizará una recapitulación de las conclusiones alcanzadas durante la ejecución del mismo.

Índice

I.7.1 Resultados finales y conclusiones.	97
I.7.2 Visión futura y ampliaciones.	98

I.7.1 Resultados finales y conclusiones

En el actual Trabajo de Fin de Grado se ha programado un sistema SCADA para la monitorización de un prototipo de parque eólico a escala, así como la adquisición de datos y la notificación de errores de este.

El mencionado sistema ha sido programado mediante el lenguaje de National Instruments (LabVIEW). Con el objetivo de crear un sistema completo, intuitivo y atractivo para el usuario; el código se ha dividido en diferentes pestañas: la primera ventana que se encontrará en el programa será la pestaña de identificación, la cual permitirá acceder a todo el potencial del sistema SCADA o navegar hasta otra ventana mediante la cual el usuario se podrá registrar.

El potencial completo del programa reúne todas las funcionalidades recogidas en el pliego de condiciones, así como aspectos nuevos que se han ido agregando durante el proceso de programación. Se encontrará una ventana de consulta de base de datos, donde se podrá examinar cualquier dato almacenado previamente; una pestaña de "breaker" o interruptor automático, que permitirá finalizar el trabajo del simulador en cualquier momento; una pestaña para consultar los datos a tiempo real; una ventana de archivo de errores; un simulador y una pestaña de información acerca del programa.

La crisis sanitaria que sobrevino durante el segundo trimestre del año 2020 y la consecuente incapacidad de poder trabajar de forma presencial en la Universitat Politècnica de València ha convertido trabajar contra el prototipo a escala en un imposible. Para suplir dicha carencia, se han generado tres códigos en MATLAB que cumplieran con la función de simulador y se ha programado el código en base a ellos, intentando pensar en todo momento en una futura simbiosis con el sistema real.

A lo largo de este trabajo se ha llegado a diversas conclusiones:

- La monitorización de un parque de generación de energía se erige, hoy en día, como una de las piedras angulares para todo proyecto energético, permitiendo evitar muchos problemas durante el funcionamiento de la planta, posibilitando al personal encargado de su operación el comportamiento del sistema ante eventuales incidencias.
- El grado de robustez de un código es fundamental, lejos de lo atractivo que parezca o de lo completo que sea, la capacidad de trabajar largas horas sin que aparezcan errores o "bugs" es lo que hace de un código un verdadero programa preparado para salir al mercado.
- La importancia de una conexión sin "handshake" previo, que habilite al programa a recibir y/o enviar datos sin la necesidad de recibir una confirmación por parte del otro computador y/o simulador. En sistemas donde la velocidad es clave, es preferible recibir algún dato erróneo a perder un paquete de datos entero.
- La envergadura de las energías renovables en pos de conseguir un futuro más sostenible energéticamente. Este punto puede sonar algo tópico, pero solo hace falta repasar los apartados de "Situación energética" y "problema energético" presentes en este mismo trabajo para darse cuenta de este hecho.
- La necesidad de compartir datos y conocimientos en la red, así como la importancia de la creación y el mantenimiento de foros de discusión e intercambio de archivos. Esta opinión

puede sonar algo extraña, pero el presente trabajo de fin de grado no habría sido posible de no ser por estas comunidades de gran ayuda a la hora de resolver problemas y dudas.

- La importancia de generar una base de datos accesible en todo momento, así como de un sistema de consulta robusto para poder explorar datos siempre que se desee.
- Y, por último, pero no por esto menos importante, la relevancia del papel investigador de las universidades, quienes juegan muchas veces un rol protagonista en nuevos avances tecnológicos y son, en muchas ocasiones, poco reconocidas socialmente.

I.7.2 Visión futura y ampliaciones

Para empezar, sería conveniente recordar que el software presentado en este documento es el primer prototipo para la monitorización de los ensayos realizados por parte de la Universitat Politècnica de València, con objeto de su investigación en los paquetes WP-3 Y WP-16 para el proyecto europeo PROMOTioN.

Todo esto, junto a la imposibilidad de utilizar el prototipo a escala previsto inicialmente debido a la crisis sanitaria derivada de la COVID-19, ha hecho de este programa lo que se conoce como una versión "betha" del software final, a falta de añadir algunas funcionalidades y reconfigurar los parámetros para trabajar contra el prototipo en vez de contra otra computadora.

A pesar de ello, este programa tiene como objetivo convertirse en el software referente para el control de los ensayos realizados en la UPV y los cimientos para cualquier trabajo posterior de la misma tipología sirviendo como una base amplia y estable donde comenzar a construir el programa definitivo.

A lo largo de este documento se ha conseguido dar solución a la totalidad de problemas planteados al inicio: establecimiento de una conexión estable y veloz sin requerir un "handshake", creación de una base de datos y de una pestaña de consulta de histórico, programación de una ventana que permita al operario controlar el sistema de protección de la maquinaria, un simulador y un archivo de errores, todo ello dentro de un entorno atractivo para el usuario. Además se ha generado un sistema de identificación acompañado de un módulo de registro con su respectiva base de datos que tiene la funcionalidad de enviar un mensaje al usuario confirmando su registro en esta; asimismo se ha generado una ventana de ayuda donde se recoge toda la información relativa al proyecto y los manuales creados para el mismo (tanto de programador como de usuario).

Dado que las principales mejoras posibles ya han sido expuestas y analizadas en el apartado de "Solución del problema", se emplearán las siguientes líneas para expresar posibles ampliaciones para el código generado:

La primera extensión sería la creación de una pestaña de control de parque donde observar en conjunto todos los aerogeneradores presentes en la instalación y sus datos y variables más destacadas en tiempo real, sobre un mapa o dibujo simplificado. Se plantearía como una ventana de inicio, en la cual el usuario pudiera elegir entre ejercer un control general sobre el parque o una monitorización de una turbina eólica específica, clicando sobre ella para obtener más información.

La segunda sugerencia consiste en generar una aplicación móvil desde la que acceder a los parámetros de la instalación de la misma forma que desde un ordenador; este software serviría

exclusivamente como medio de consulta y explotaría la idea de mejora expuesta en el apartado de "Solución del problema", que trataba la posibilidad de trabajar con el apoyo de una base de datos en la nube.

Este planteamiento permitiría convertir este sistema SCADA en un programa válido para su utilización en autoconsumo o para pequeños consumidores, debido a la creciente fama tecnológica adquirida por estos sistemas de control y su facilidad de descarga y uso.

La tercera propuesta de ampliación tiene que ver con el idioma, más concretamente con la posibilidad de elegirlo. Para ello, sería necesario configurar un menú en el cual el usuario pudiera seleccionar el idioma con el que deseara trabajar (preferiblemente en la pestaña de identificación), y modificar la presentación de todo el código en consonancia.

La cuarta y última ampliación que se plantea en este escrito es la creación de una comunidad donde los usuarios pudieran compartir los datos de sus instalaciones, y la generación de una extensión dentro del propio software que permitiese comparar en tiempo real los datos recogidos en las distintas plantas eólicas.

Parte II

Estudio económico

Capítulo II.1

Planificación temporal

La planificación temporal es clave en cualquier proyecto de ingeniería.

Dictaminar correctamente el tiempo dedicado a cada tarea se plantea fundamental a la hora de cumplir con los plazos de entrega de los documentos.

Asimismo, es un documento esencial una vez finalizado el trabajo, para analizar en qué puntos se puede mejorar o qué tareas han supuesto mayor retraso y/o dedicación de recursos para así asignarlos de manera más eficiente a continuación.

II.1.1 Diagrama de Gantt

En este proyecto en cuestión podemos identificar tres fases: una primera fase de análisis de alternativas y elaboración del código del programa, una segunda etapa en la que se realiza la implementación y la última, que engloba todos los aspectos relacionados con reuniones, planificación y redacción. Se puede observar un distanciamiento (retraso) a lo largo de toda la fase uno debido a que se empezó utilizando otro software de programación, lo que supuso una pérdida significativa de tiempo y recursos.

Diagrama de Gantt

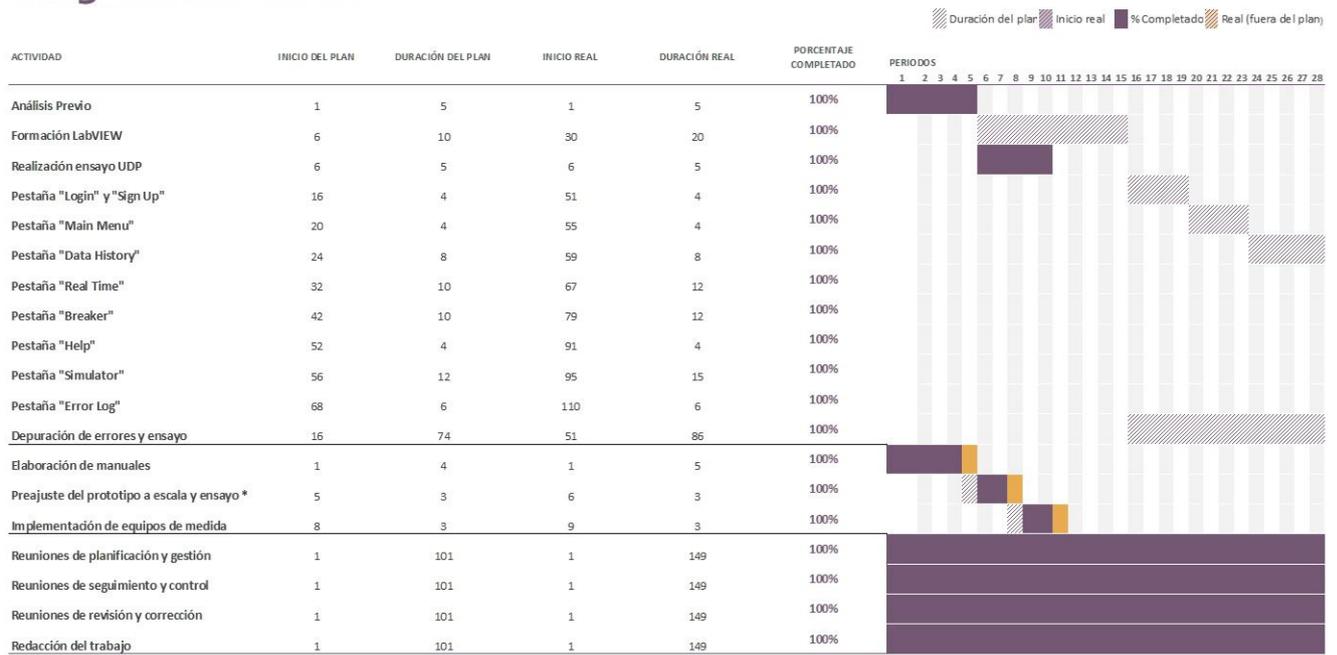


Diagrama de Gantt

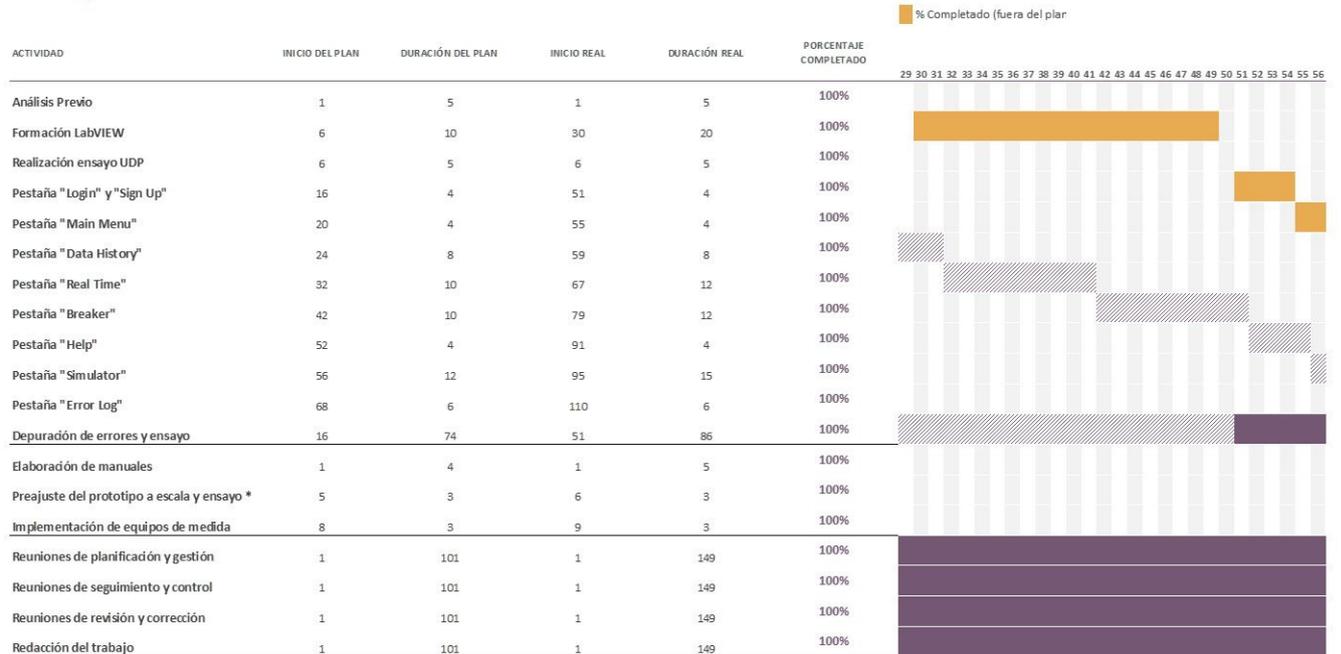


Diagrama de Gantt

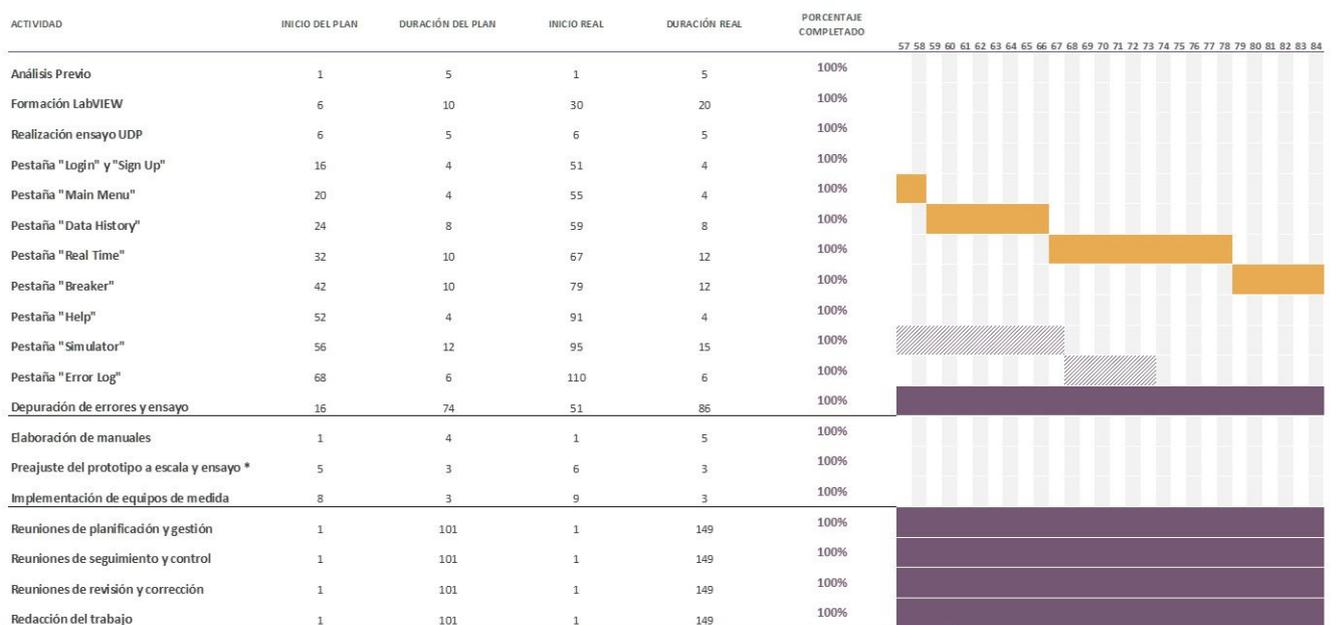


Diagrama de Gantt

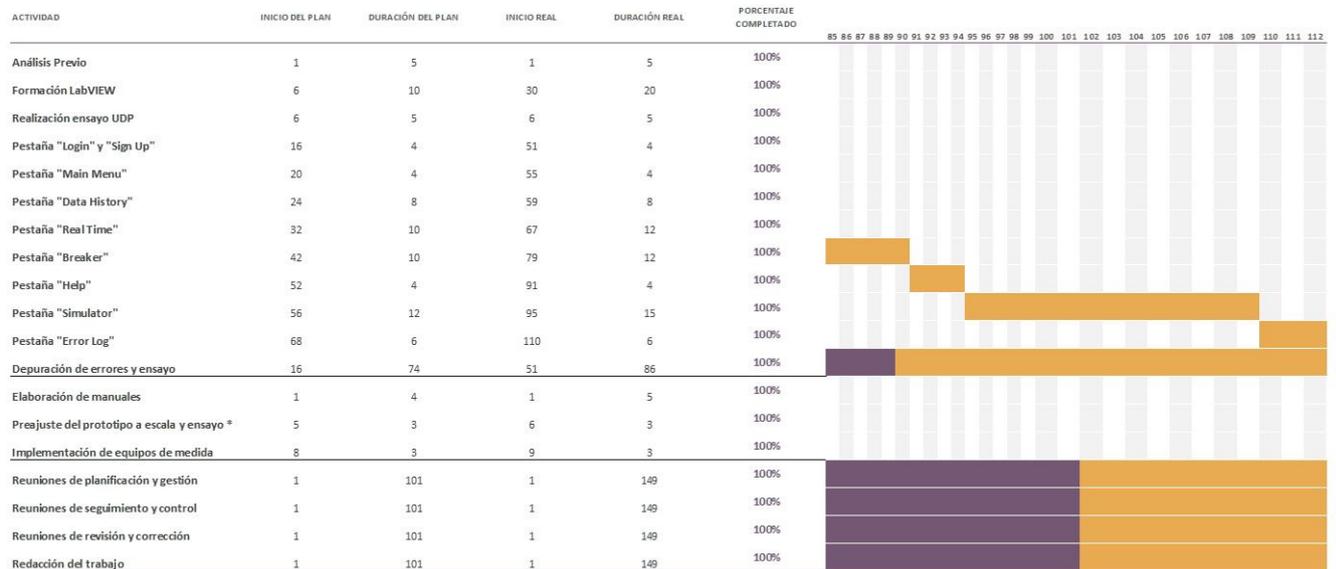


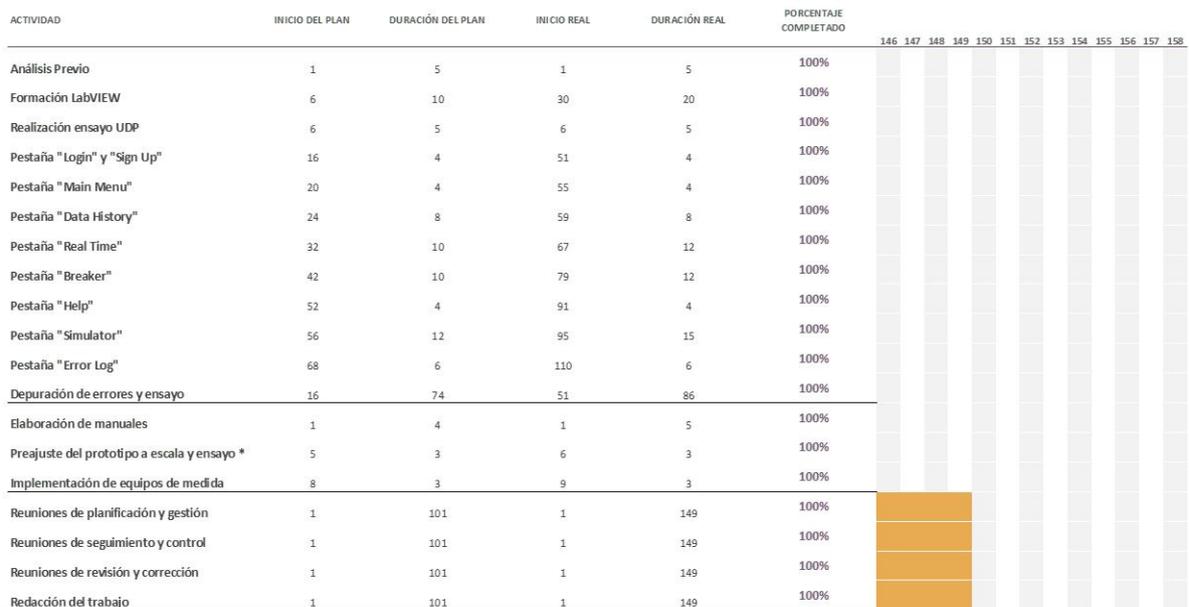
Diagrama de Gantt



Diagrama de Gantt



Diagrama de Gantt



Capítulo II.2

Presupuesto del proyecto

En el siguiente apartado se procede a desglosar el presupuesto necesario para realizar el presente trabajo de investigación. No será un presupuesto al uso, como se acostumbra a ver en los trabajos de esta naturaleza debido a que, para su realización no serán necesarias grandes cantidades de material.

Índice

II.2.1 Presupuesto del proyecto	111
II.2.2 Precios descompuestos	111
II.2.3 Presupuesto de ejecución material por capítulos	117
II.2.4 Presupuesto de ejecución por contrata	118

II.2.1 Presupuesto del proyecto

Las actividades realizadas se han dividido en varios capítulos y subcapítulos para facilitar su comprensión e interpretación, con el objetivo de evaluar la viabilidad económica del proyecto y servir como ejemplo para proyectos futuros de la misma naturaleza.

Los precios unitarios de los programas, materiales y horas de trabajo se han obtenido de fuentes fiables. Por su parte, el cálculo del coste de los equipos se ha realizado a partir de la amortización de estos, con un periodo de vida útil de 10 años. Para realizar esta operación se ha utilizado la fórmula:

$$A = \frac{t * C}{P}$$

Figura II.2.1: Amortizaciones

Dónde A es la amortización en euros (€), t es el tiempo en horas (h), C es el costo del equipo en euros (€) y P es el periodo de vida útil en horas (h).

II.2.2 Precios descompuestos

CAPÍTULO 1. Estudio de alternativas y trabajos previos

Subcapítulo 1.1. Análisis de alternativas						
Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)	
P.II	hr	Ingeniero Industrial	10	15,600	156,00	
L.LAB	ud	(Libro) LabVIEW Graphical Programming	1	52,580	52,58	
L.SIM	ud	(Libro) I. rápida a Matlab y Simulink para ciencia e ingeniería	1	8,550	8,55	
R.ORD	hr	Ordenador de sobremesa	8	0,040	0,32	
R.MON	hr	Monitor de sobremesa	8	0,017	0,14	
R.PER	hr	Periféricos	8	0,001	0,01	
R.LAM	hr	Lámpara de sobremesa	6	0,002	0,01	
				Costes Directos Complementarios (2%)	217,606	4,35
					221,96	

Subcapítulo 1.2. Formación en LabVIEW						
Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)	
C.LAB	ud	Curso LabVIEW (MeasureIT)	1	0	0,00	
P.II	hr	Ingeniero Industrial	16	15,600	249,60	
R.ORD	hr	Ordenador de sobremesa	16	0,040	0,64	
R.MON	hr	Monitor de sobremesa	16	0,017	0,27	
R.PER	hr	Periféricos	16	0,001	0,02	
R.LAM	hr	Lámpara de sobremesa	10	0,002	0,02	
				Costes Directos Complementarios (2%)	250,548	5,01
					255,56	

Diseño e implementación de un sistema de supervisión, control y adquisición de datos (SCADA)
para un prototipo a escala de parque eólico

Subcapítulo 1.3. Distribución temporal					
Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	5	15,600	78,00
L.SL	ud	Licencia Slack	1	0,000	0,00
R.ORD	hr	Ordenador de sobremesa	5	0,040	0,20
R.MON	hr	Monitor de sobremesa	5	0,017	0,09
R.PER	hr	Periféricos	5	0,001	0,01
R.LAM	hr	Lámpara de sobremesa	3	0,002	0,01
		Costes Directos Complementarios (2%)		78,296	1,57
					79,86

Subcapítulo 1.4. Licencias					
Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
L.MAT	hr	Licencia MATLAB	100	0,4	40,00
L.LAB	hr	Licencia LabVIEW	211	0,700	147,70
L.C	hr	Licencia C++	10	0,000	0,00
L.MO	hr	Licencia Microsoft Office	40	0,075	3,00
L.LA	hr	Licencia Latex	80	0,000	0,00
L.W10	hr	Licencia Windows 10	500	0,005	2,50
L.LC	ud	Licencia LucidChart	1	7,950	7,95
		Costes Directos Complementarios (2%)		193,2	3,86
					205,01

CAPÍTULO 2. Primeros ensayos de comunicación vía UDP

Subcapítulo 2.1. Realización del ensayo					
Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	8	15,600	124,80
R.TP	ud	TP-Link TL-SF1005D - Conmutador Fast Ethernet con 5 Puertos	1	7,600	7,60
R.POR	hr	Portatil personal	8	0,083	0,66
R.ORD	hr	Ordenador de sobremesa	8	0,040	0,32
R.MON	hr	Monitor de sobremesa	8	0,017	0,14
R.PER	hr	Periféricos	8	0,001	0,01
R.LAM	hr	Lámpara de sobremesa	6	0,002	0,01
		Costes Directos Complementarios (2%)		133,54	2,67
					136,21

CAPÍTULO 3. Programación

Subcapítulo 3.1. Pestaña "Login" y "Sign Up"

Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	8	15,600	124,80
R.POR	hr	Portatil personal	8	0,083	0,66
R.LAM	hr	Lámpara de sobremesa	6	0,002	0,01
		Costes Directos Complementarios (2%)		125,476	2,51
					127,99

Subcapítulo 3.2. Pestaña "Main Menu"

Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	8	15,600	124,80
R.POR	hr	Portatil personal	8	0,083	0,66
R.LAM	hr	Lámpara de sobremesa	6	0,002	0,01
		Costes Directos Complementarios (2%)		125,476	2,51
					127,99

Subcapítulo 3.3. Pestaña "Data History"

Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	15	15,600	234,00
R.POR	hr	Portatil personal	15	0,083	1,25
R.LAM	hr	Lámpara de sobremesa	12	0,002	0,02
		Costes Directos Complementarios (2%)		235,269	4,71
					239,97

Subcapítulo 3.4. Pestaña "Real Time"

Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	30	15,600	468,00
R.POR	hr	Portatil personal	30	0,083	2,49
R.ORD	hr	Ordenador de sobremesa	30	0,040	1,20
R.MON	hr	Monitor de sobremesa	30	0,017	0,51
R.PER	hr	Periféricos	30	0,001	0,03
R.LAM	hr	Lámpara de sobremesa	20	0,002	0,04
		Costes Directos Complementarios (2%)		472,27	9,45
					481,72

Subcapítulo 3.5. Pestaña "Breaker"

Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	30	15,600	468,00
R.POR	hr	Portatil personal	30	0,083	2,49
R.ORD	hr	Ordenador de sobremesa	30	0,040	1,20
R.MON	hr	Monitor de sobremesa	30	0,017	0,51
R.PER	hr	Periféricos	30	0,001	0,03
R.LAM	hr	Lámpara de sobremesa	20	0,002	0,04
		Costes Directos Complementarios (2%)		472,27	9,45
					481,72

Subcapítulo 3.6. Pestaña "Help"

Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	10	15,600	156,00
R.POR	hr	Portatil personal	10	0,083	0,83
R.LAM	hr	Lámpara de sobremesa	10	0,002	0,02
		Costes Directos Complementarios (2%)		156,85	3,14
					159,99

Diseño e implementación de un sistema de supervisión, control y adquisición de datos (SCADA)
para un prototipo a escala de parque eólico

Subcapítulo 3.7. Pestaña "Simulator"

Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	40	15,600	624,00
R.POR	hr	Portatil personal	40	0,083	3,32
R.ORD	hr	Ordenador de sobremesa	40	0,040	1,60
R.MON	hr	Monitor de sobremesa	40	0,017	0,68
R.PER	hr	Periféricos	40	0,001	0,04
R.LAM	hr	Lámpara de sobremesa	35	0,002	0,07
		Costes Directos Complementarios (2%)		629,71	12,59
					642,30

Subcapítulo 3.8. Pestaña "Error Log"

Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	20	15,600	312,00
R.POR	hr	Portatil personal	20	0,083	1,66
		Costes Directos Complementarios (2%)		313,66	6,27
					319,93

CAPÍTULO 4. Depuración de errores y ensayo

Subcapítulo 4.1. Depuración de errores y ensayo

Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	50	15,600	780,00
R.POR	hr	Portatil personal	50	0,083	4,15
R.ORD	hr	Ordenador de sobremesa	50	0,040	2,00
R.MON	hr	Monitor de sobremesa	50	0,017	0,85
R.PER	hr	Periféricos	50	0,001	0,05
R.LAM	hr	Lámpara de sobremesa	40	0,002	0,08
		Costes Directos Complementarios (2%)		787,13	15,74
					802,87

Subcapítulo 4.2. Elaboración de manuales

Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	20	15,600	312,00
R.POR	hr	Portatil personal	20	0,083	1,66
R.LAM	hr	Lámpara de sobremesa	10	0,002	0,02
		Costes Directos Complementarios (2%)		313,68	6,27
					319,95

CAPÍTULO 5. Ensayo en banco de laboratorio (no realizado debido a la crisis sanitaria)

Subcapítulo 5.1. Preajuste del prototipo a escala y ensayo *

Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	16	15,600	249,60
P.DP	hr	Directora responsable del proyecto	10	24,000	240,00
		Costes Directos Complementarios (2%)		489,6	9,79
					499,39

CAPÍTULO 6. Implementación industrial

Subcapítulo 6.1. Implementación de equipos de medida y software (*)					
Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
R.UC	ud	Universal current transducer	1	159,700	159,70
R.CP	ud	PLC de control de potencia	1	39,560	39,56
R.BTR	ud	Interruptor automático trifásico	1	382,290	382,29
R.OI	ud	Ordenador	1	500,000	500,00
R.PI	ud	Monitor	1	250,000	250,00
R.PP	ud	Periféricos	1	17,000	17,00
P.II	hr	Ingeniero Industrial	16	15,600	249,60
P.TE	hr	Técnico electricista	30	8,838	265,14
		Costes Directos Complementarios (2%)		1863,29	37,27
					1900,56

Precio por aerogenerador que tenga el parque

Subcapítulo 6.2. Mantenimiento					
Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.TE	hr	Técnico electricista	10	8,838	88,38
		Costes Directos Complementarios (2%)		88,38	1,77
					90,15

CAPÍTULO 7. Reuniones

Subcapítulo 7.1. Reuniones de planificación y gestión					
Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	5	15,600	78,00
P.DP	hr	Directora responsable del proyecto	5	24,000	120,00
		Costes Directos Complementarios (2%)		198	3,96
					201,96

Subcapítulo 7.2. Reuniones de seguimiento y control					
Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	10	15,600	156,00
P.DP	hr	Directora responsable del proyecto	10	24,000	240,00
P.PI	hr	Personal docente de investigación	2	15,000	30,00
		Costes Directos Complementarios (2%)		426,00	8,52
					434,52

Subcapítulo 7.3. Reuniones de revisión y corrección					
Código	Unidad	Descripción	Cantidad	Precio unitario (€)	Importe (€)
P.II	hr	Ingeniero Industrial	20	15,600	312,00
P.DP	hr	Directora responsable del proyecto	10	24,000	240,00
		Costes Directos Complementarios (2%)		552,00	11,04
					563,04

II.2.3 Presupuesto de ejecución material por capítulos

Capítulo 1. Estudio de alternativas y trabajos previos		
Código	Descripción	Importe (€)
1.1	Análisis de alternativas	221,96
1.2	Formación en LabVIEW	255,56
1.3	Distribución temporal	79,86
1.4	Licencias	197,06
		754,44

Capítulo 2. Primeros ensayos de comunicación vía UDP		
Código	Descripción	Importe (€)
2.1	Realización del ensayo	136,21
		136,21

Capítulo 3. Programación		
Código	Descripción	Importe (€)
3.1	Pestaña "Login" y "Sign Up"	127,99
3.2	Pestaña "Main Menu"	127,99
3.3	Pestaña "Data History"	239,97
3.4	Pestaña "Real Time"	481,72
3.5	Pestaña "Breaker"	481,72
3.6	Pestaña "Help"	159,99
3.7	Pestaña "Simulator"	642,30
3.8	Pestaña "Error Log"	319,93
		2581,61

Capítulo 4. Depuración de errores y ensayo		
Código	Descripción	Importe (€)
4.1	Depuración de errores y ensayo	802,87
4.2	Elaboración de manuales	319,95
		1122,82

Capítulo 5. Ensayo en banco de laboratorio		
Código	Descripción	Importe (€)
5.1	Preajuste de prototipo a escala y ensayo	499,39
		499,39

Capítulo 6. Implementación industrial		
Código	Descripción	Importe (€)
6.1	Implantación de equipos de medida y software	1900,56
6.2	Mantenimiento	90,15
		1990,71

Capítulo 7. Reuniones		
Código	Descripción	Importe (€)
7.1	Reuniones de planificación y gestión	201,96
7.2	Reuniones de seguimiento y control	434,52
7.3	Reuniones de revisión y corrección	563,04
		1199,52

II.2.4 Presupuesto de ejecución por contrata

Código	Descripción	Importe (€)
Capítulo 1	Estudio de alternativas y trabajos previos	754,44
Capítulo 2	Primeros ensayos de comunicación vía UDP	136,21
Capítulo 3	Programación	2581,61
Capítulo 4	Depuración de errores y ensayo	1122,82
Capítulo 5	Ensayo en banco de laboratorio	499,39
Capítulo 6	Implementación industrial	1990,71
Capítulo 7	Reuniones	1199,52
Presupuesto de ejecución material		8284,70
Gastos generales y cargos fiscales (7%)		579,93
Beneficio Industrial (10%)		828,47
Suma		9693,10
IVA (21%)		2035,55
Presupuesto de ejecución por contrata		11728,65

El presupuesto total asciende a la cantidad de **once mil setecientos veintiocho con sesenta y cinco euros**.

Los capítulos con * se refieren a tareas que no se han podido realizar debido a la crisis sanitaria pero forman parte del proyecto final.

Parte III

Pliego de condiciones técnicas y
administrativas

Capítulo III.1

Pliego de condiciones técnicas y administrativas

El Pliego de condiciones es el documento del proyecto que regula las relaciones entre el Promotor, y las empresas contratistas, en el cual se recogen los derechos, obligaciones y responsabilidades entre ellos.

Los pliegos de condiciones, recogen las especificaciones técnicas definidas, tanto en los planos, como en la memoria del proyecto, especificando las características técnicas de los materiales, los medios auxiliares a emplear durante el transcurso de las obras, así como los equipos necesarios para llevar a cabo las unidades de obra definidas en el proyecto. Guisado 02/08/2020

Índice

III.1.1 Entidad que inicia el procedimiento	123
III.1.2 Objeto del proyecto	123
III.1.3 Antecedentes y situación actual	123
III.1.4 Alcance del proyecto	123
III.1.5 Descripción de las tareas a realizar	124
III.1.6 Compatibilidad.	124
III.1.7 Entorno tecnológico	124
III.1.7.1 Entorno de programación MATLAB.	124
III.1.7.2 Entorno de usuario MATLAB.	125
III.1.7.3 Entorno de programación LabVIEW	125
III.1.7.4 Entorno de usuario LabVIEW	126
III.1.7.5 Entorno de Wireshark	127
III.1.8 Metodología	127
III.1.9 Mecanismos de seguimiento y control	128
III.1.10 Documentación.	128
III.1.11 Idioma	128

III.1.1 Entidad que inicia el procedimiento

La Universidad Politècnica de València para PROMOTioN, un proyecto financiado con fondos de la Unión Europea.

III.1.2 Objeto del proyecto

El objeto principal del proyecto es el desarrollo, diseño e implementación de un sistema de supervisión y adquisición de datos que permita controlar y monitorizar un simulador en tiempo real de parque eólico marino. Dicho sistema vendrá soportado por alguna solución software existente adaptada a requisitos específicos del simulador.

La conexión de señales atenderá a los protocolos TCP/IP y UDP y se desarrollará una interfaz completa con el usuario.

El sistema quedará abierto a posibles ampliaciones en sus funcionalidades.

III.1.3 Antecedentes y situación actual

Como punto de partida se cuenta con un simulador en tiempo real donde se implementan modelos de parques eólicos marinos y se comprueba su comportamiento en régimen de funcionamiento normal, maniobras y faltas. Para este simulador se requiere diseñar e implementar un interfaz de usuario para mostrar los resultados de los ensayos realizados, establecer límites que provoquen la activación de alarmas, así como interactuar con el sistema para la construcción de los distintos escenarios.

III.1.4 Alcance del proyecto

El sistema SCADA en cuestión debe abarcar las siguientes funcionalidades:

- Representación de datos históricos, permitiendo al usuario seleccionar el periodo que desea visualizar.
- Configuración ensayos seleccionando los parámetros de entrada (tensiones, corrientes y potencias).
- Archivar errores y generar alarmas de faltas.
- Generar informes a partir de los datos recogidos en la base de datos.
- Monitorización del sistema en tiempo real (tensiones, corrientes y potencias).

III.1.5 Descripción de las tareas a realizar

El adjudicatario será el encargado de coordinar los distintos módulos desde el punto de vista técnico y organizativo, de forma que:

- Disponga de la visión global del avance del trabajo y la situación de cada subproyecto en el que pueda ser dividido.
- Sea el interlocutor único con la Universidad Politècnica de València y los departamentos asociados.

III.1.6 Compatibilidad

El software propuesto debe ser compatible con el sistema de comunicación de la Universidad Politècnica de València y con la infraestructura disponible en el Instituto Universitario de Automática e Informática Industrial, así como con el prototipo a escala propuesto.

III.1.7 Entorno tecnológico

III.1.7.1 Entorno de programación MATLAB

Matlab es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, macOS y GNU/Linux. Matlab es capaz de aprovechar los recursos disponibles de hardware, tales como aceleradoras gráficas, tarjetas de sonido y pantallas de color, así como las propiedades de multitarea.

Matlab es una aplicación de 32 o 64 bits. Los requisitos mínimos del sistema para Windows son los siguientes:

Sistemas operativos:

- Windows 10 (versión 1709 o superior)
- Windows 7 Service Pack 1
- Windows Server 2019
- Windows Server 2016

Procesadores:

- Mínimo: cualquier procesador Intel o AMD x86-64
- Recomendado: Cualquier procesador Intel o AMD x86-64 con cuatro núcleos lógicos y soporte de conjunto de instrucciones AVX2 Disco
- Mínimo: 3 GB de espacio en el disco duro solo para MATLAB, 5-8 GB para una instalación típica
- Recomendado: se recomienda un SSD

Una instalación completa de todos los productos MathWorks puede ocupar hasta 31 GB de espacio en disco.

RAM:

- Mínimo: 4 GB
- Recomendado: 8 GB

Para Polyspace, se recomiendan 4 GB por núcleo.

Gráficos:

No se requiere una tarjeta gráfica específica. Se recomienda una tarjeta gráfica acelerada por hardware compatible con OpenGL 3.3 con memoria de GPU de 1 GB. La aceleración de GPU con Parallel Computing Toolbox requiere una GPU que tenga una capacidad de cálculo 3.0 o superior. Para obtener más información, consulte Soporte de GPU por versión.

Otros elementos:

- Ratón para Windows.
- Adaptador gráfico de 8 bits y pantalla para más de 256 colores.
- Impresora.
- Tarjeta de sonido.

III.1.7.2 Entorno de usuario MATLAB

Los requerimientos para el usuario son los mismos que para el programador.

III.1.7.3 Entorno de programación LabVIEW

LabVIEW es una plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico pensado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido.

LabVIEW es una aplicación de 32 o 64 bits. Los requisitos mínimos del sistema para Windows son los siguientes:

Sistemas operativos:

- Windows 10/8.1/8/7 SP1 (32- and 64-bit)
- Windows Server 2012 R2 (64-bit)
- Windows Server 2008 R2 SP1 (64-bit)

Procesadores:

- Pentium 4M (or equivalent) or later (32-bit)
- Pentium 4 G1 (or equivalent) or later (64-bit)

Disco:

5 GB mínimo. **RAM:** 1 GB mínimo.

Gráficos:

1024 x 768 pixels

Otros elementos:

- Ratón para Windows.
- Adaptador gráfico de 8 bits y pantalla para más de 256 colores.
- Impresora.
- Tarjeta de sonido.

III.1.7.4 Entorno de usuario LabVIEW

A diferencia de Matlab, en LabVIEW si que se modifican ligeramente los requisitos técnicos después de generar el archivo .exe. Éstos son los siguientes:

Sistemas operativos:

- Windows 10/8.1/8/7 SP1 (32- and 64-bit)
- Windows Server 2012 R2 (64-bit)
- Windows Server 2008 R2 SP1 (64-bit)

Procesadores:

- Pentium 4M (or equivalent) or later (32-bit)
- Pentium 4 G1 (or equivalent) or later (64-bit)

Disco:

620 MB mínimo.

RAM:

256 MB mínimo.

Gráficos:

1024 x 768 pixels

Otros elementos:

- Ratón para Windows.
- Adaptador gráfico de 8 bits y pantalla para más de 256 colores.
- Impresora.
- Tarjeta de sonido.

III.1.7.5 Entorno de Wireshark

Sistemas operativos:

- Windows XP Home
- Windows XP Pro
- Windows XP Tablet PC
- Windows XP Media Center
- Windows Server 2003
- Windows Vista
- Windows 2008
- Windows 7
- Windows 2008 R2

Tarjetas de Red:

Para el caso de las Ethernet cualquier tarjeta que soporte Windows. Para el caso de las tarjetas inalámbricas las 802.11.

Disco:

- 75MB disponibles en Disco. (Depende del número de paquetes que se vayan a almacenar en disco). RAM
- 128MB (depende del número de paquetes se vayan a capturar).

Gráficos:

1280x1024 pixeles.

Otros elementos:

- Ratón para Windows.
- Adaptador gráfico de 8 bits y pantalla para más de 256 colores.
- Impresora.
- Tarjeta de sonido.

III.1.8 Metodología

Se exigirá la aplicación de métodos ágiles con el objetivo de lograr:

- Entregas frecuentes de software.
- Adopción de modificaciones solicitadas por el cliente.
- Integración continua de proyectos.
- Controles de calidad.

III.1.9 Mecanismos de seguimiento y control

El seguimiento del proyecto se realizará mediante dos tipos de vías:

Reuniones de seguimiento: con periodicidad semanal se celebrarán reuniones de seguimiento y revisiones técnicas, en ellas participarán el jefe de proyecto la dra. Soledad Inmaculada Bernal Pérez y los miembros del equipo de trabajo.

Correos electrónicos: con periodicidad aleatoria y dónde se consultarán dudas y se resolverán cuestiones sobre el programa en desarrollo. A su vez, toda la documentación generada durante el presente proceso será almacenada en la nube mediante "Dropbox".

III.1.10 Documentación

Para la validación del proyecto, será necesario presentar los siguientes entregables:

- Memoria técnica del trabajo.
- Presupuesto.
- Manuales de usuario.
- Manuales de programador.

III.1.11 Idioma

Todos los contenidos y componentes desarrollados (los elementos de entrada/salida) se han de presentar de tal forma que, en cualquier momento, se pueda optar por el idioma castellano, valenciano o inglés.

Parte IV

Planos

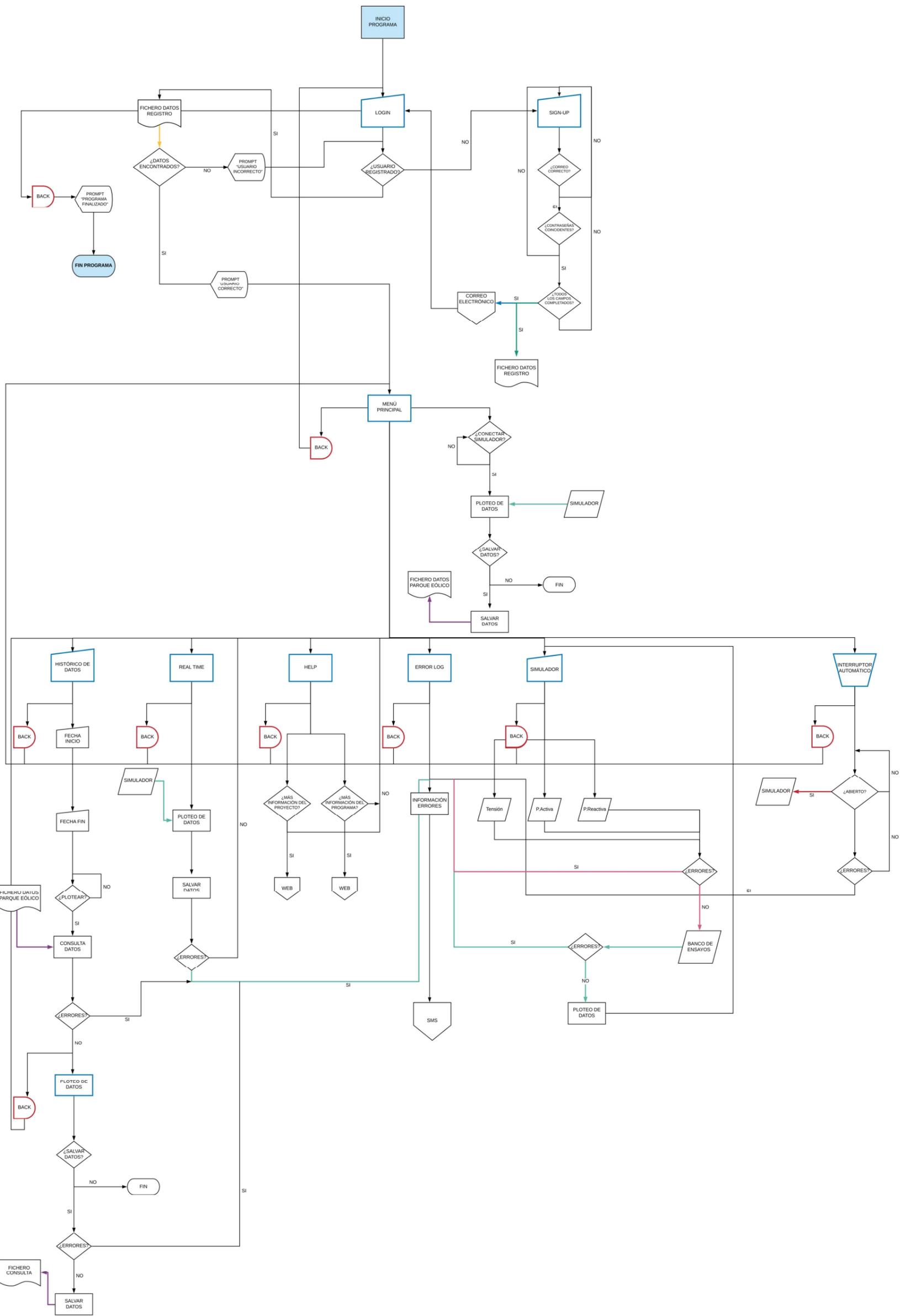
Capítulo IV.1

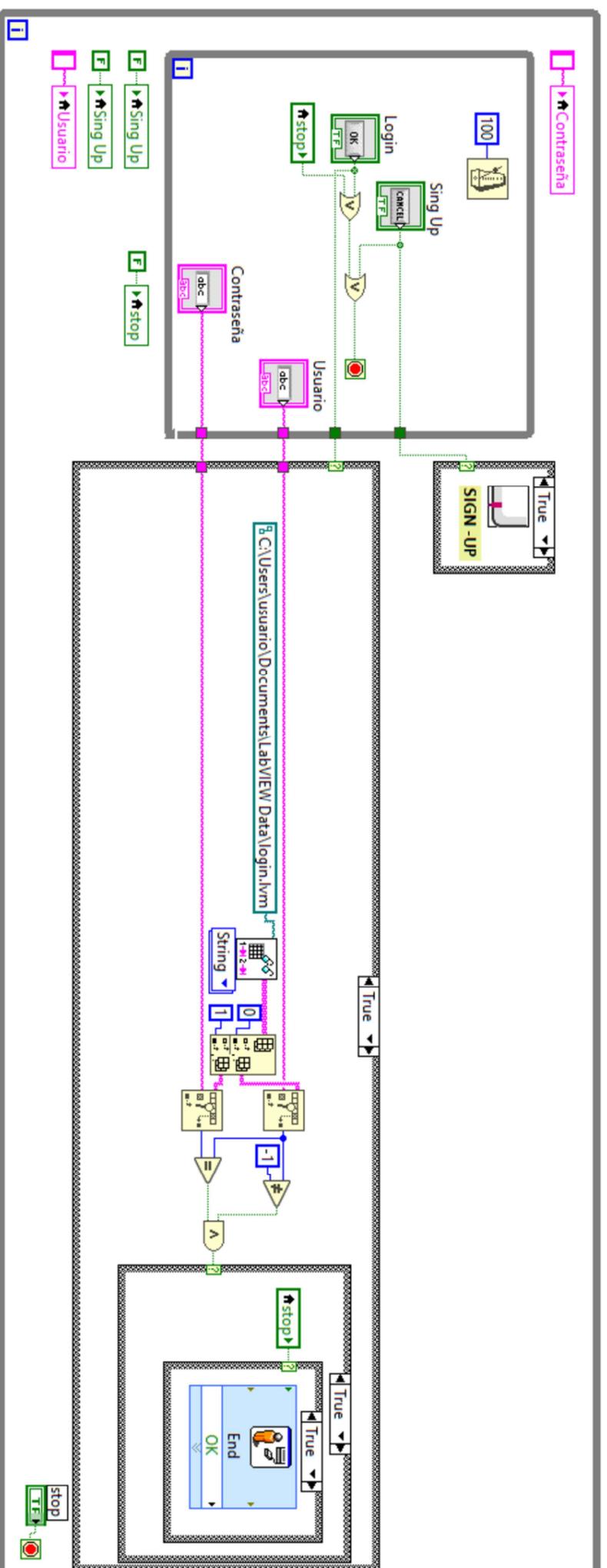
Listado de planos

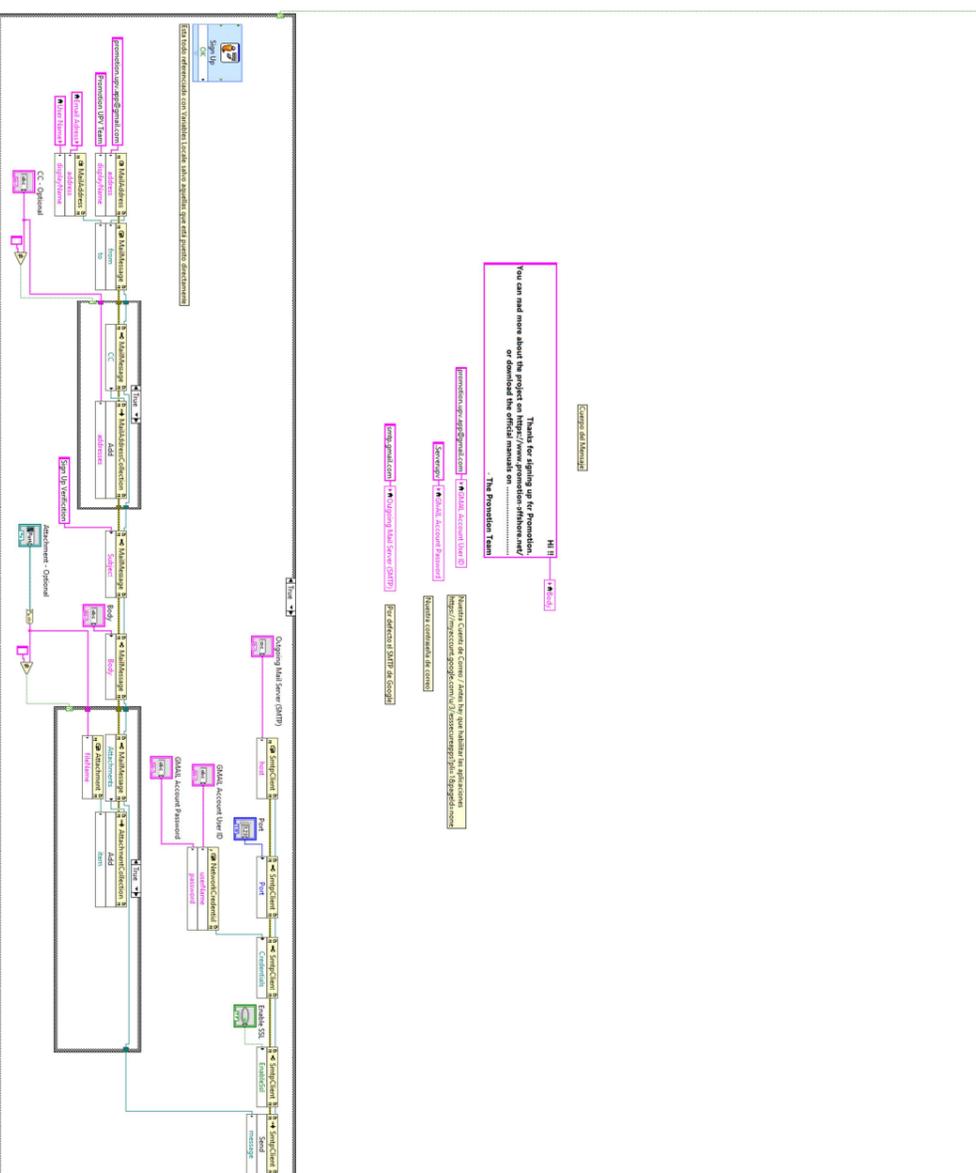
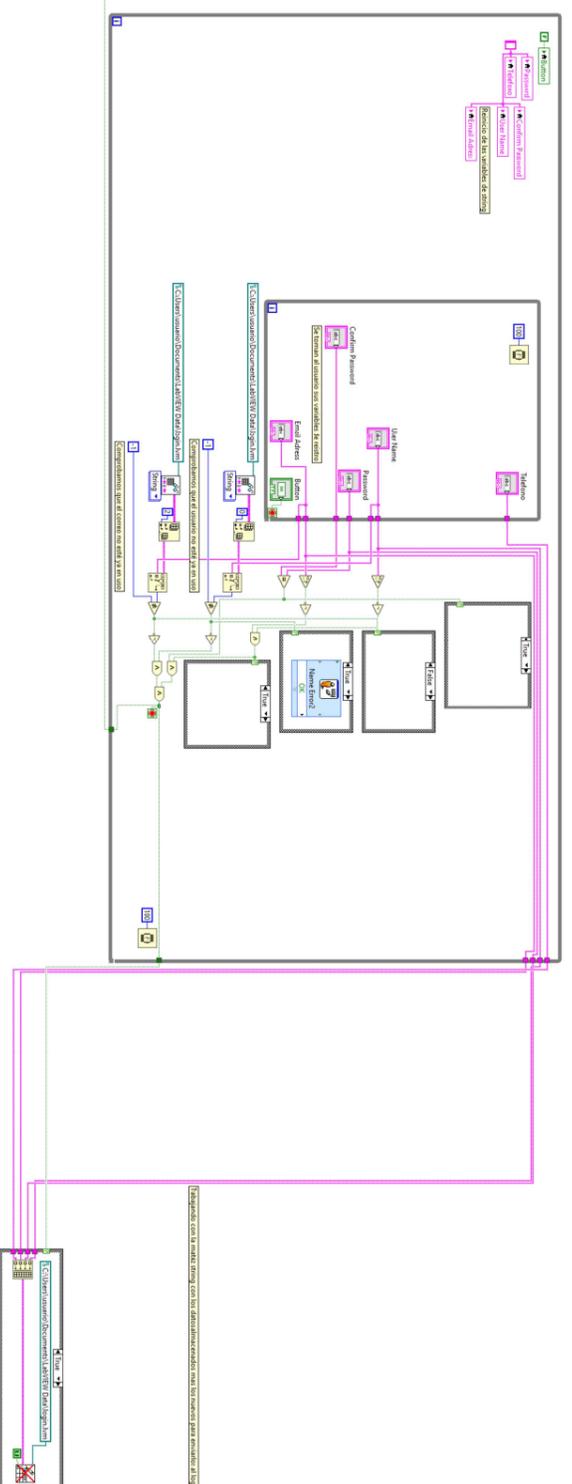
Número de plano	Nombre
1	<i>Diagrama de flujo general del software</i>
2	<i>Código de programación para ventana de "Log-in"</i>
3	<i>Código de programación para ventana de "Sign-Up"</i>
4	<i>Código de programación para ventana de "Main Menu"</i>
5	<i>Código de programación para ventana de "Real Time"</i>
6	<i>Código de programación para ventana de "Data History"</i>
7	<i>Código de programación para ventana de "Data History (Plotting)"</i>
8	<i>Código de programación para ventana de "Simulator"</i>
9	<i>Código de programación para ventana de "Error Log"</i>
10	<i>Código de programación para ventana de "Breaker"</i>
11	<i>Código de programación para ventana de "Help"</i>
12	<i>Código de programación MATLAB "Simulator"</i>
13	<i>Código de programación MATLAB "Simulator (SM_Master)"</i>
14	<i>Código de programación MATLAB "Simulator (SC_Console)"</i>
15	<i>Código de programación MATLAB "Banco de Pruebas"</i>
16	<i>Código de programación MATLAB "Breaker"</i>
17	<i>"Log-In" Interfaz de usuario</i>
18	<i>"Sign-Up" Interfaz de usuario</i>
19	<i>"Main Menu" Interfaz de usuario</i>
20	<i>"Real Time" Interfaz de usuario</i>
21	<i>"Simulator" Interfaz de usuario 1/2</i>
22	<i>"Simulator" Interfaz de usuario 2/2</i>
23	<i>"Data History" Interfaz de usuario 1/3</i>
24	<i>"Data History" Interfaz de usuario 2/3</i>
25	<i>"Data History" Interfaz de usuario 3/3</i>
26	<i>Calendario</i>
27	<i>"Error Log" Interfaz de usuario</i>
28	<i>"Breaker" Interfaz de usuario</i>
29	<i>"Help" Interfaz de usuario</i>

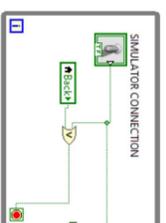
Capítulo IV.2

Planos

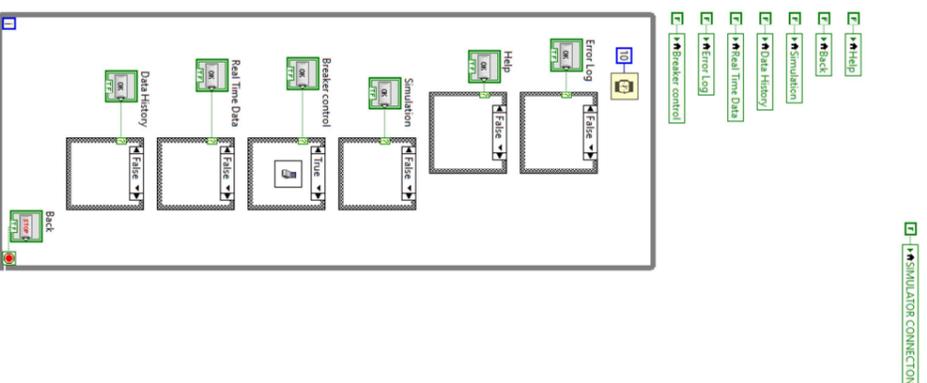
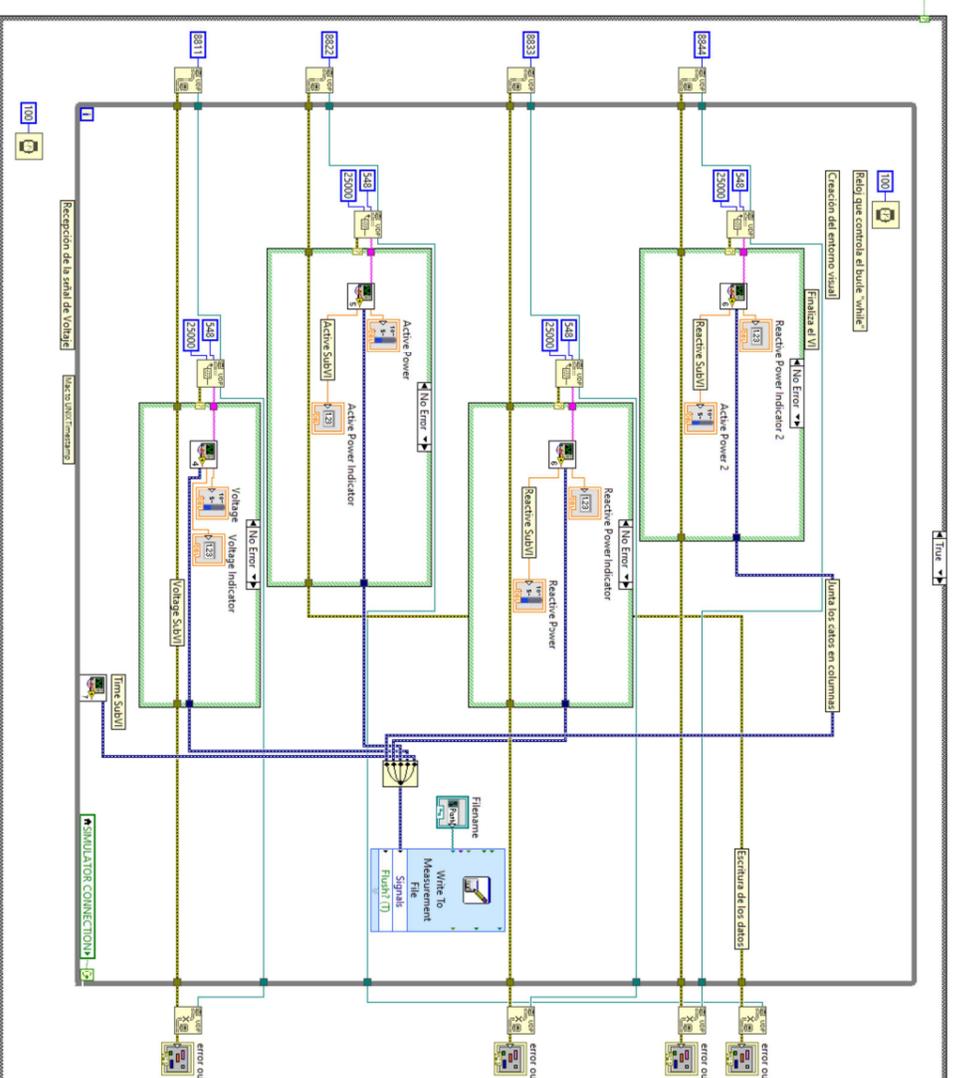


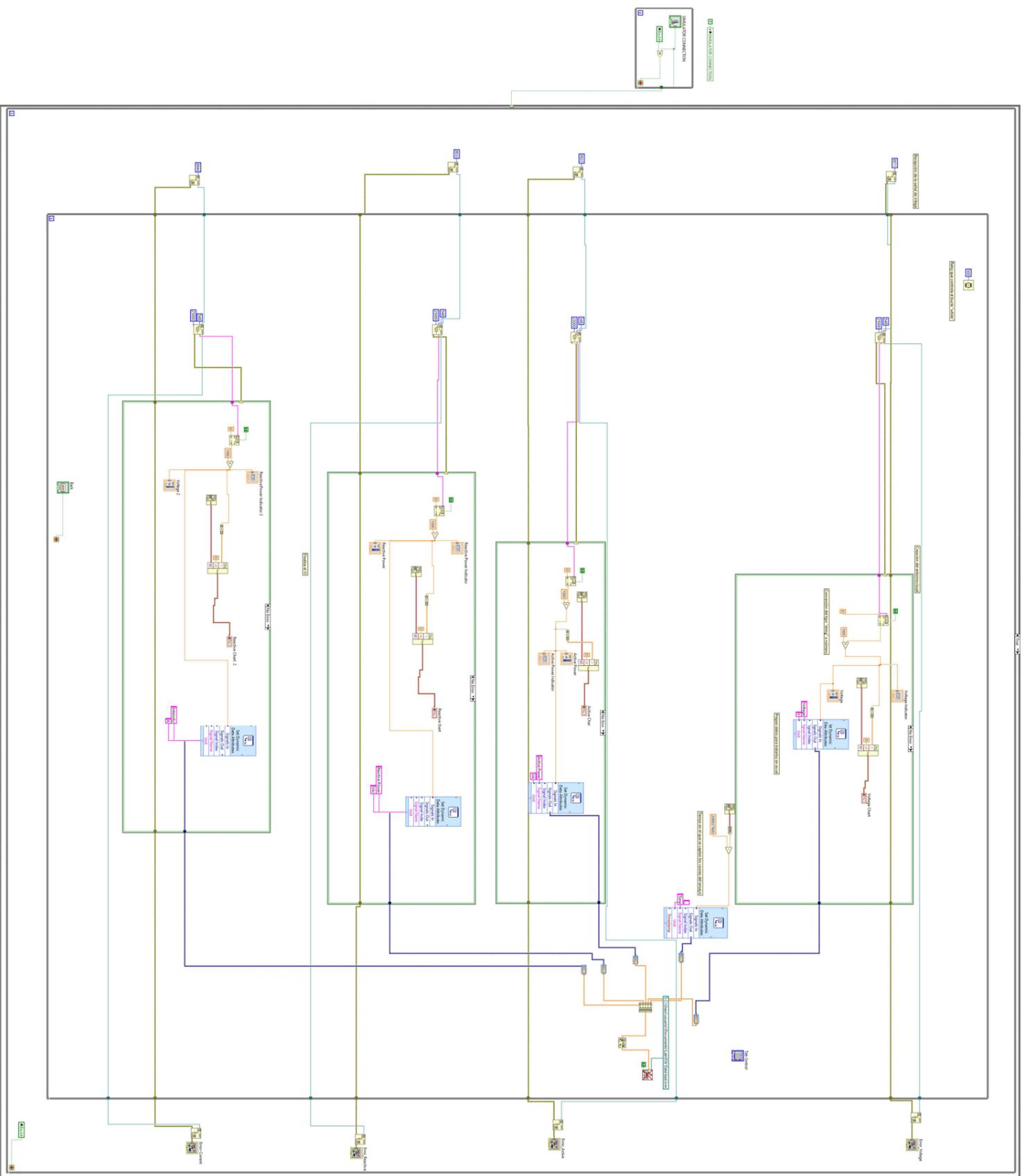


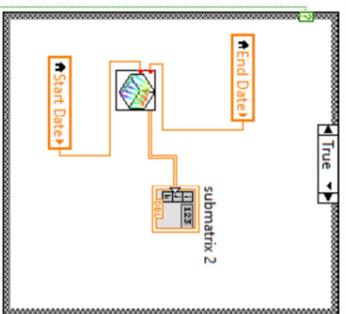




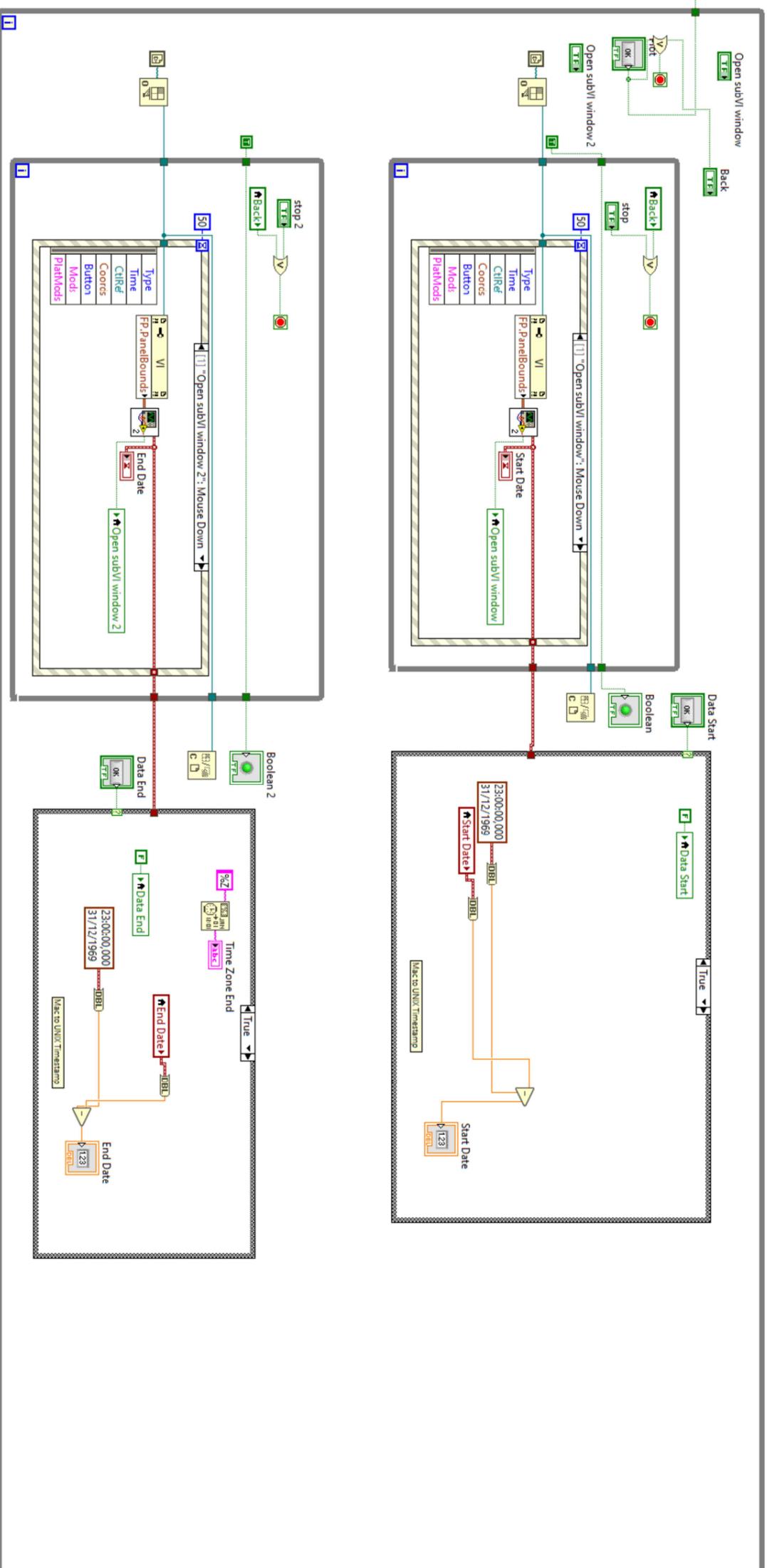
- Voltage Indicator
- Active Power
- Reactive Power
- Active Power Indicator
- Reactive Power Indicator







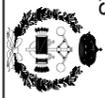
→ Data End



- Back
- Boolean
- Data Start
- Plot
- Boolean 2
- Open subVI window
- stop 2
- Open subVI window 2
- stop



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR DE INGENIERÍA
INDUSTRIAL VALENCIA

TRABAJO FINAL DE GRADO EN INGENIERIA EN TECNOLOGIAS INDUSTRIALES

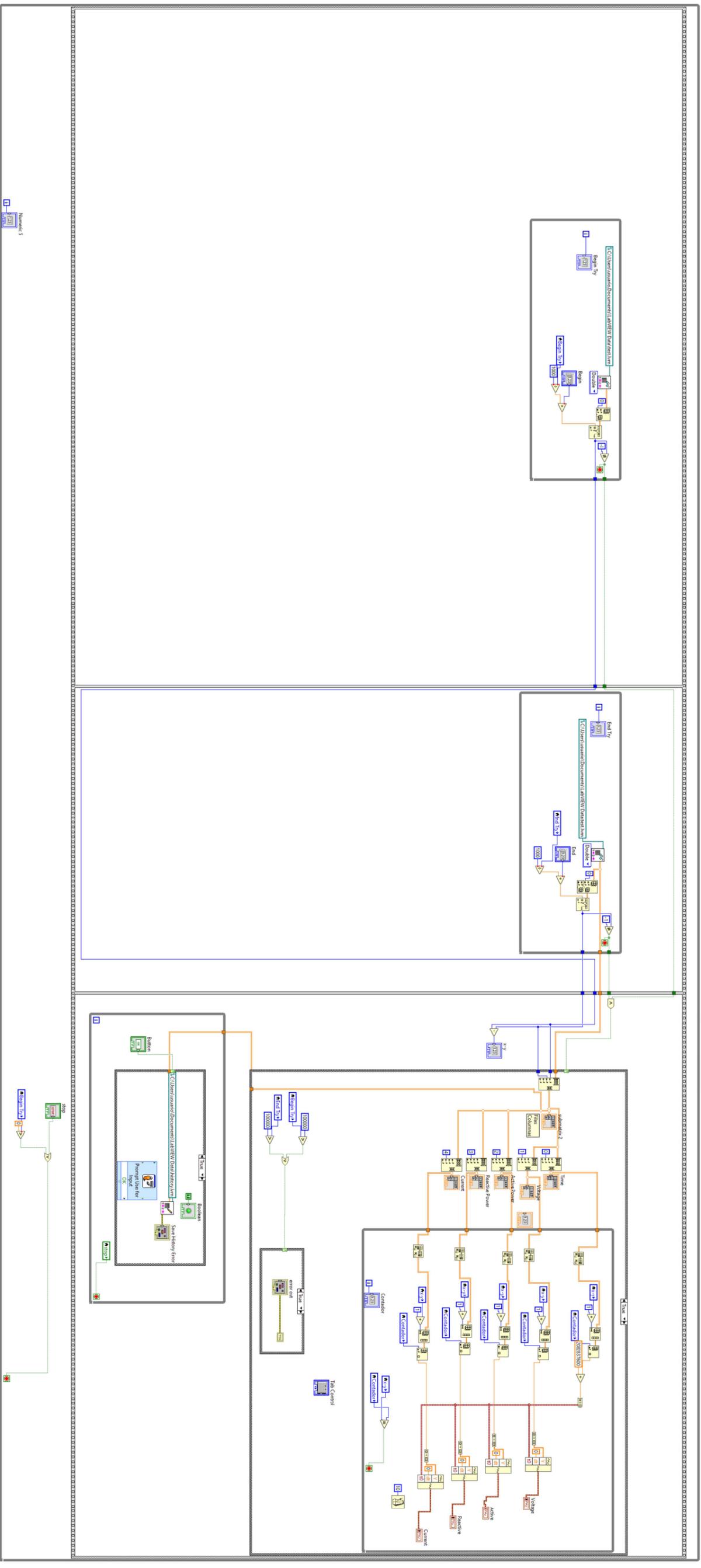
Proyecto: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SUPERVISIÓN, CONTROL Y ADQUISICIÓN DE DATOS (SCADA) PARA UN PROTOTIPO A ESCALA DE PARQUE EÓLICO

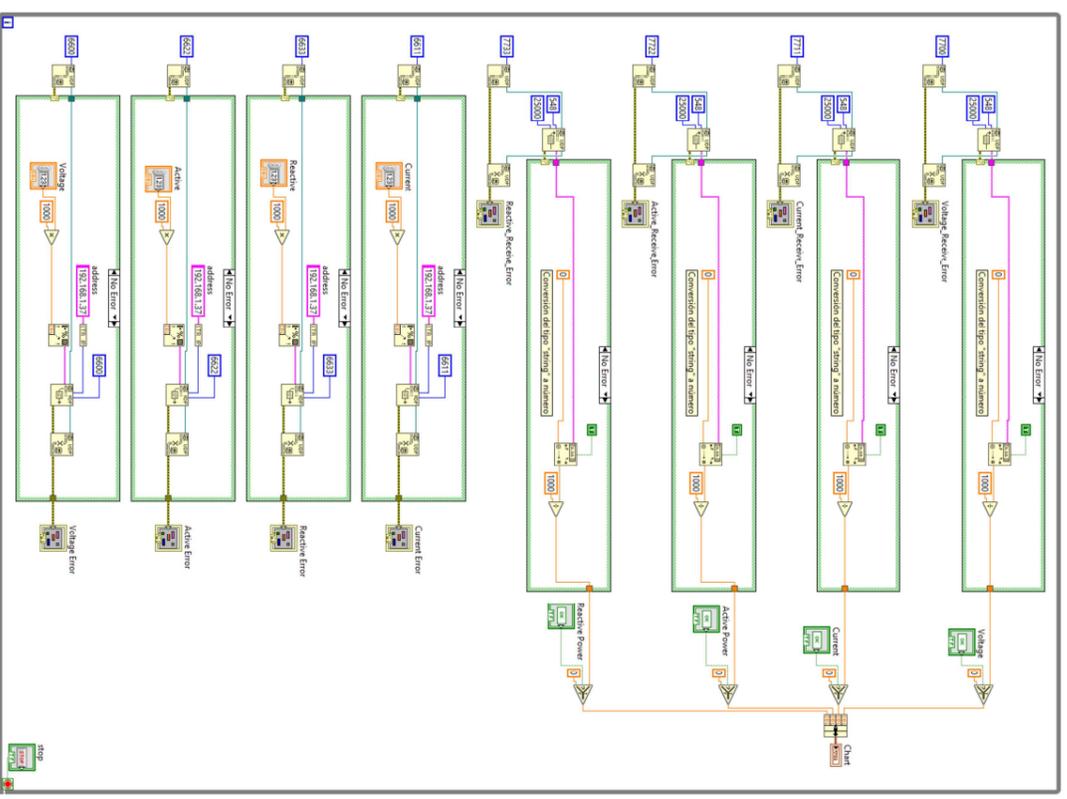
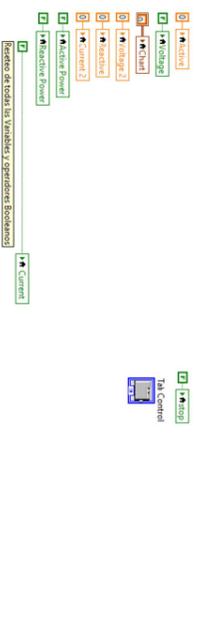
Plano: Código de programación para ventana de "Data History (Plotting)"
Autor: Julio Yustas Talamantes

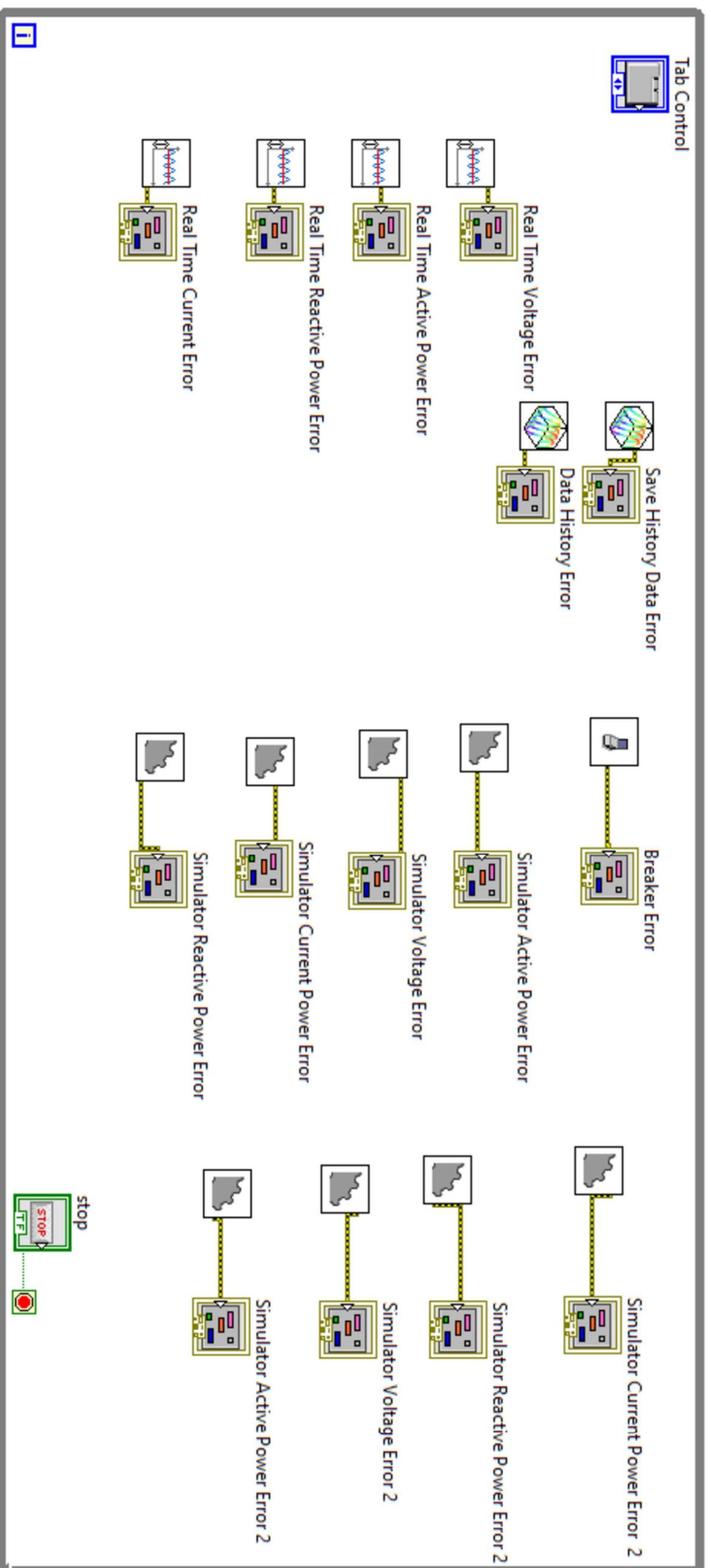
Fecha: Agosto 2020
Escala:

Nº Plano:

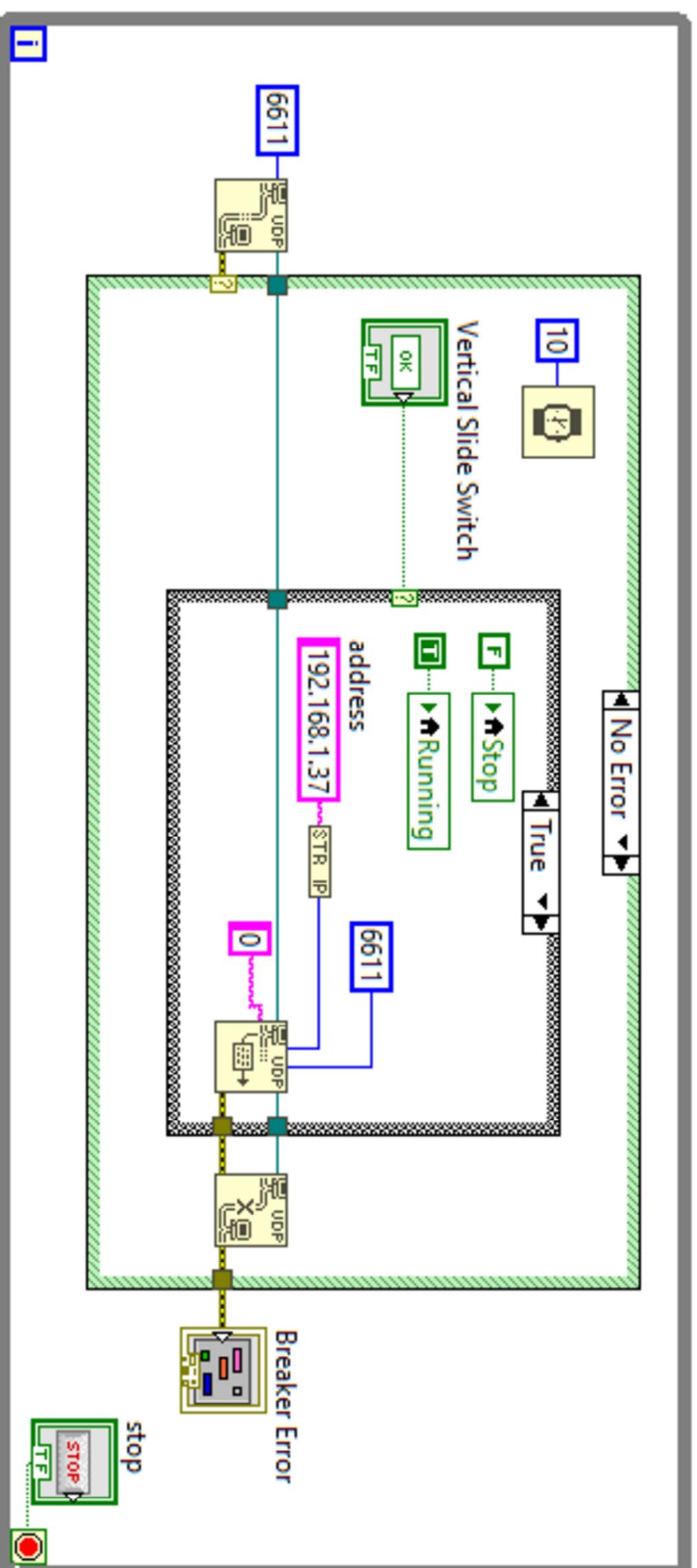
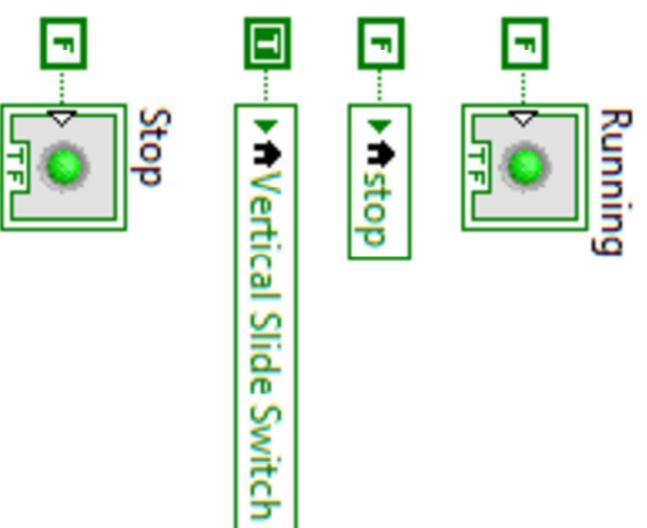
7

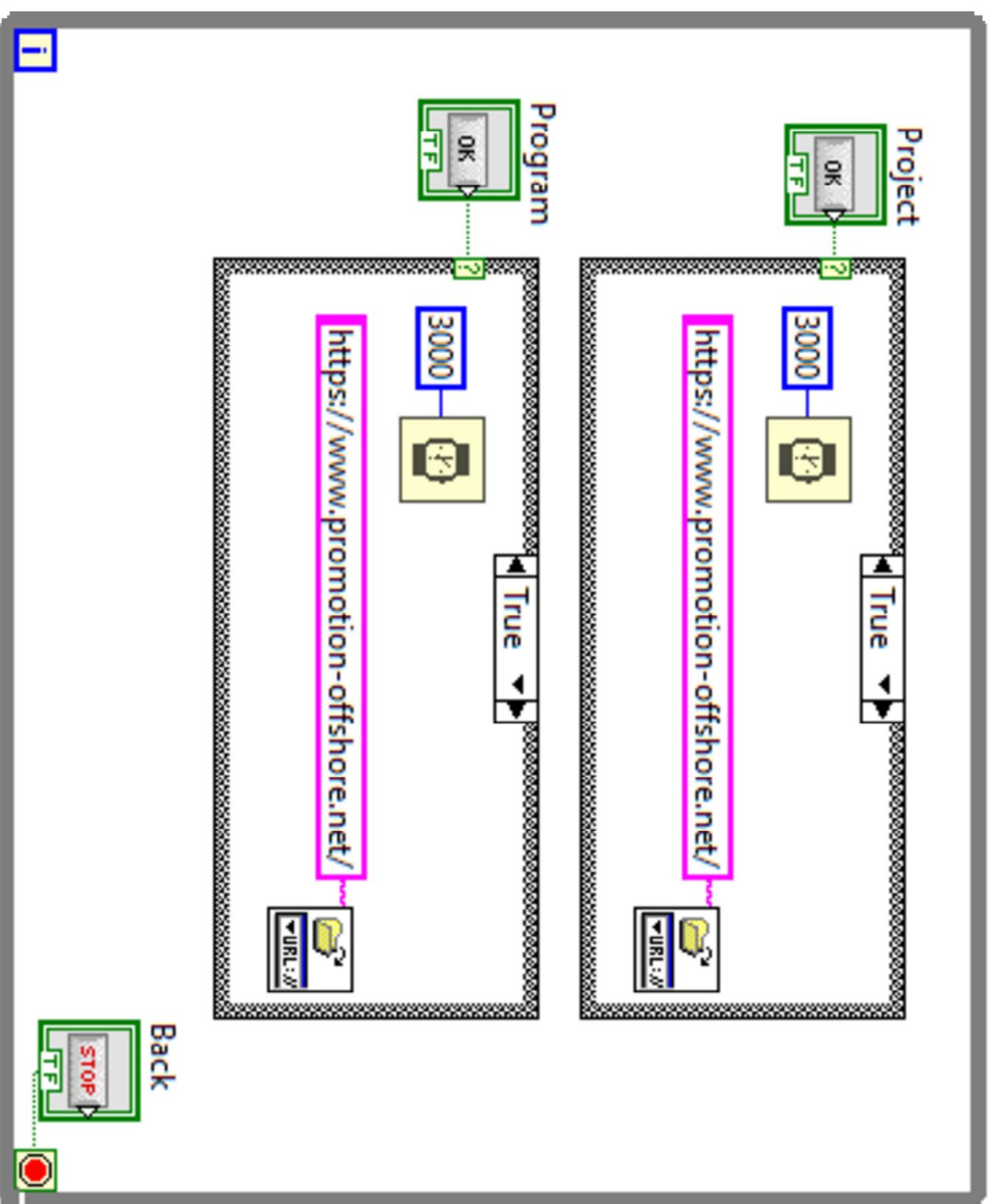
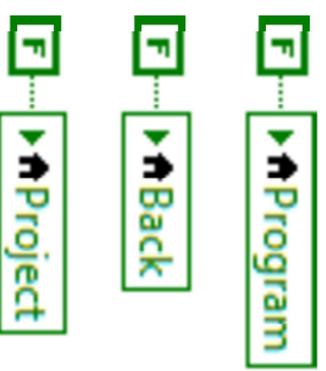






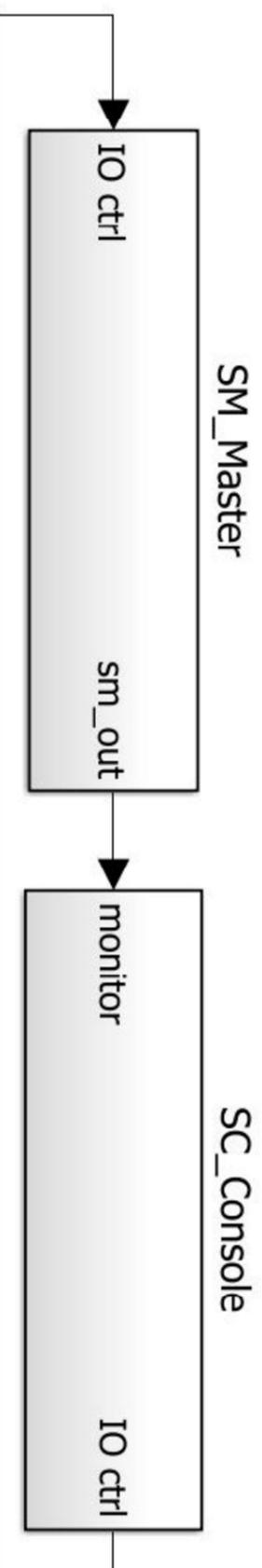
F... stop

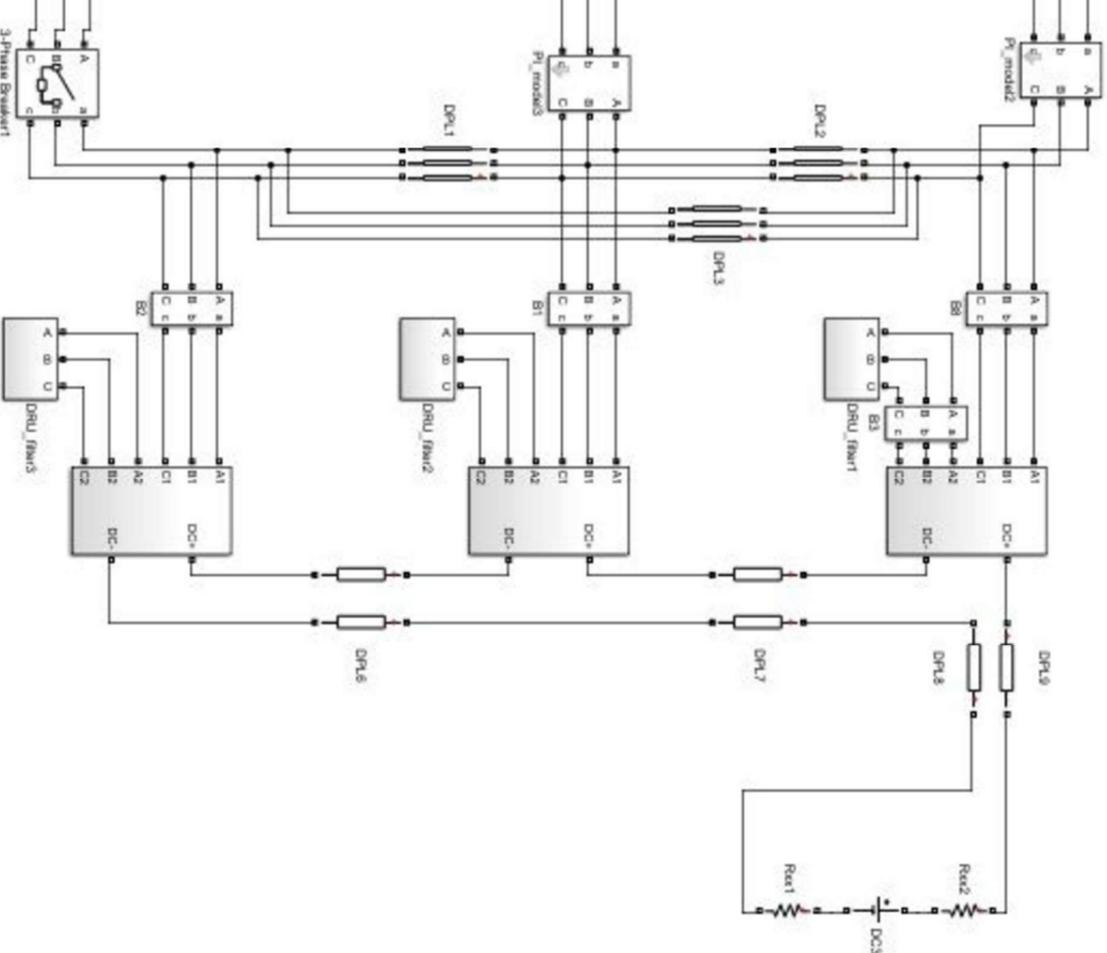
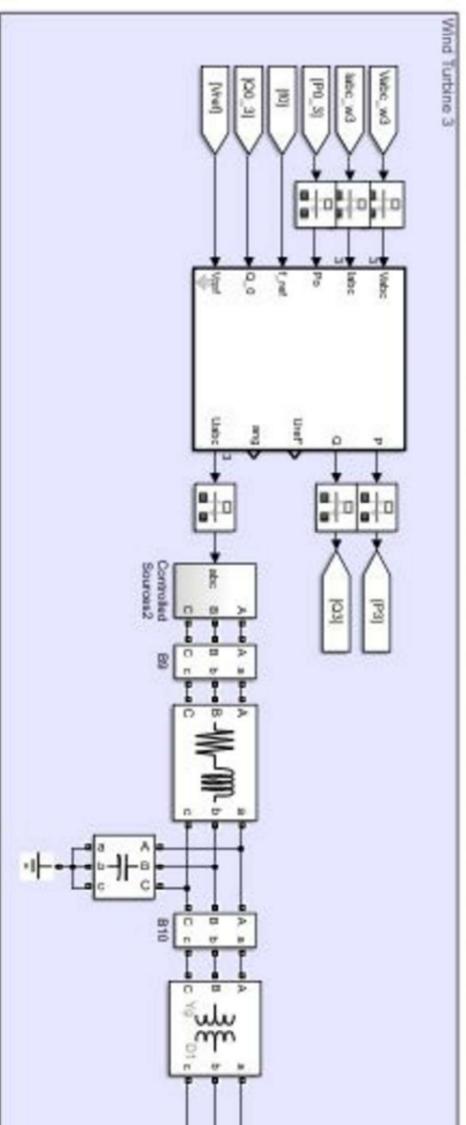
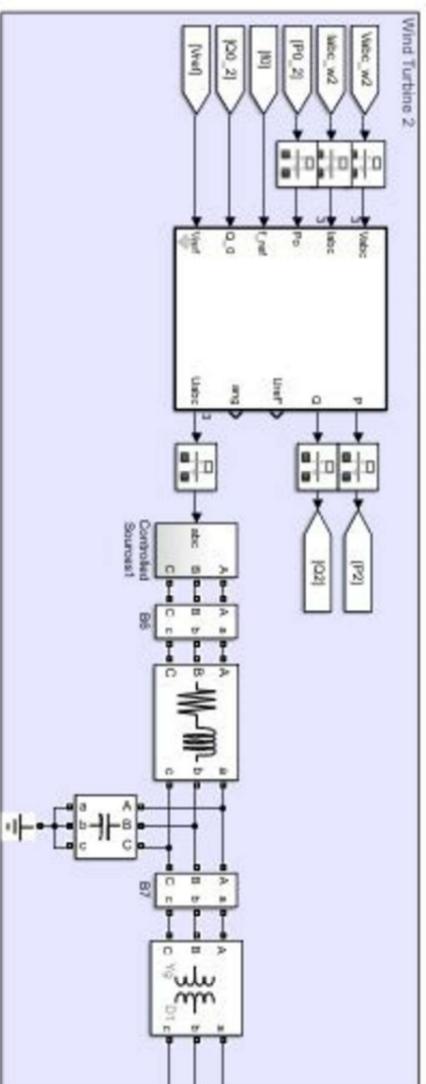
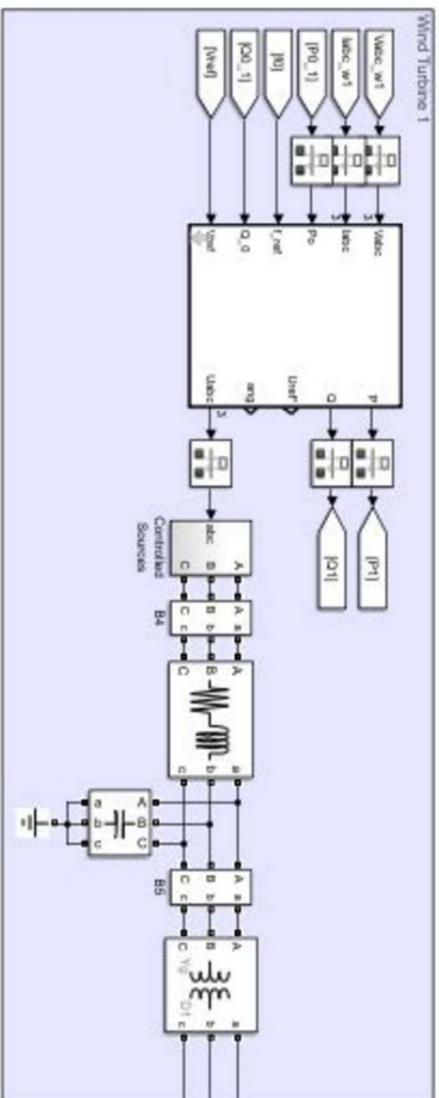
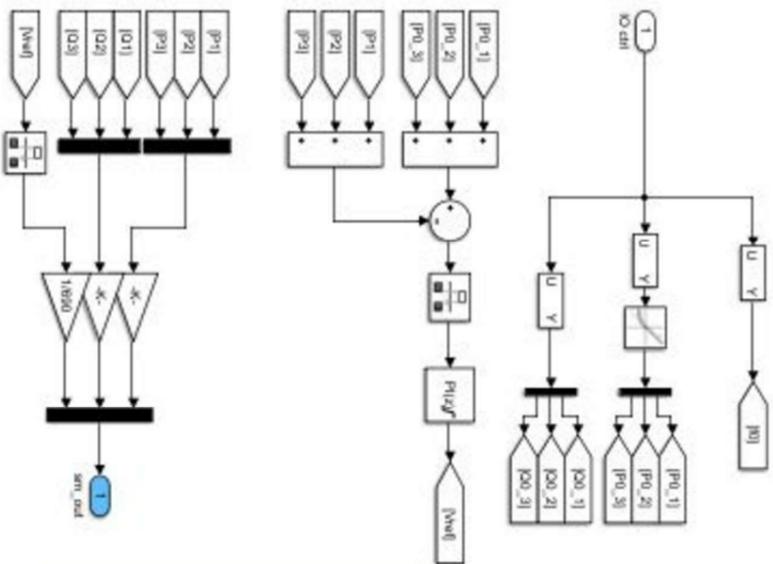


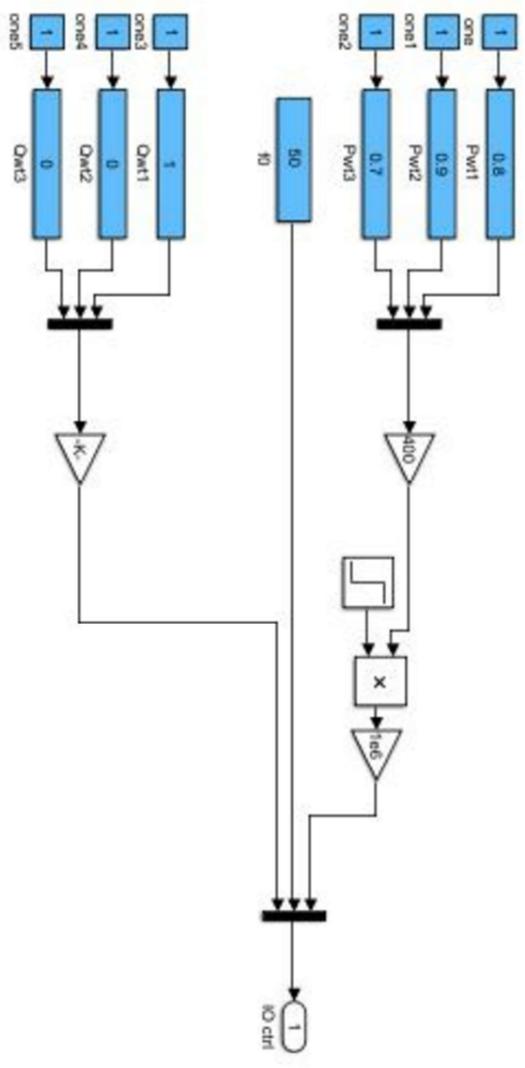
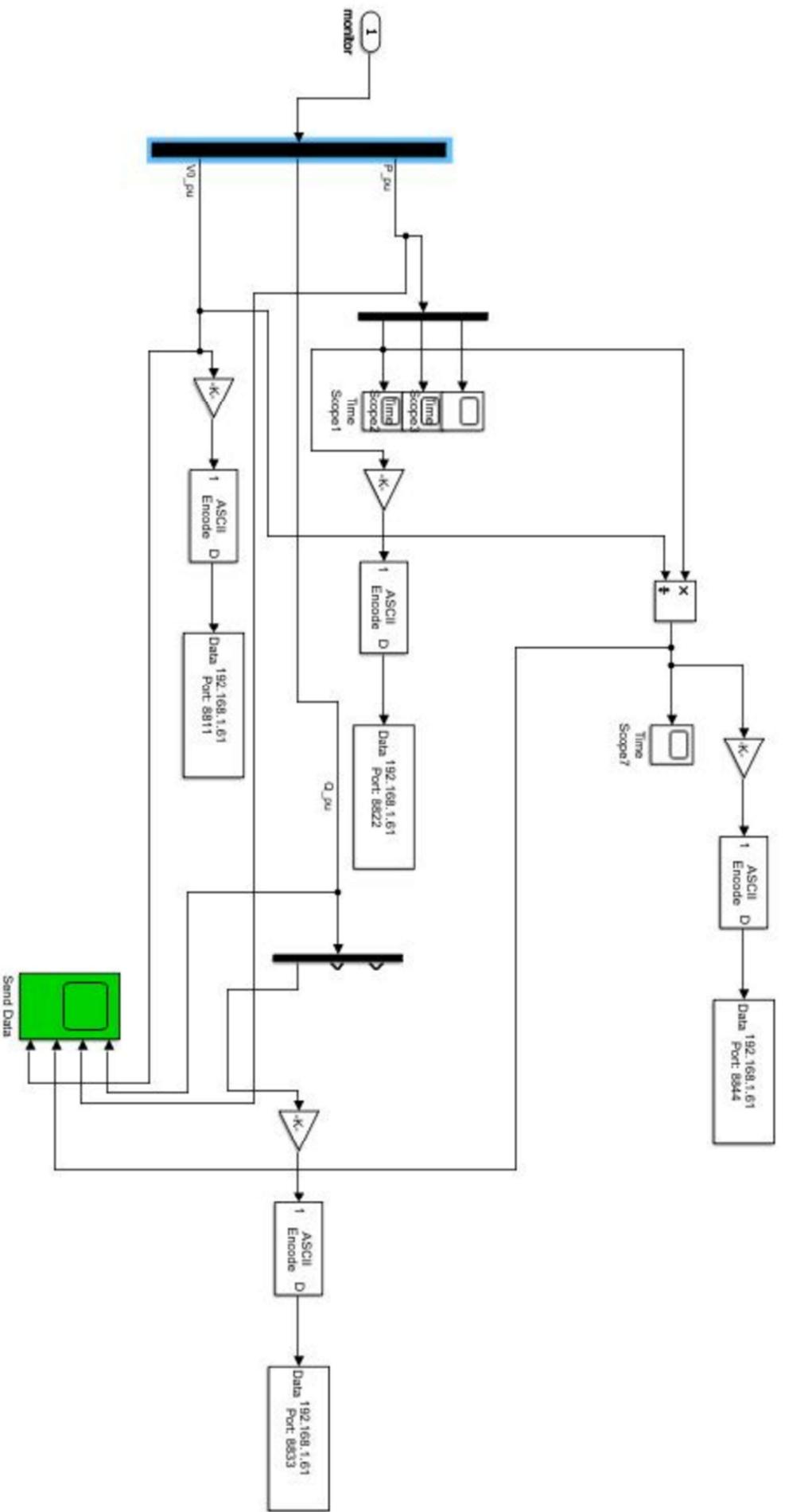


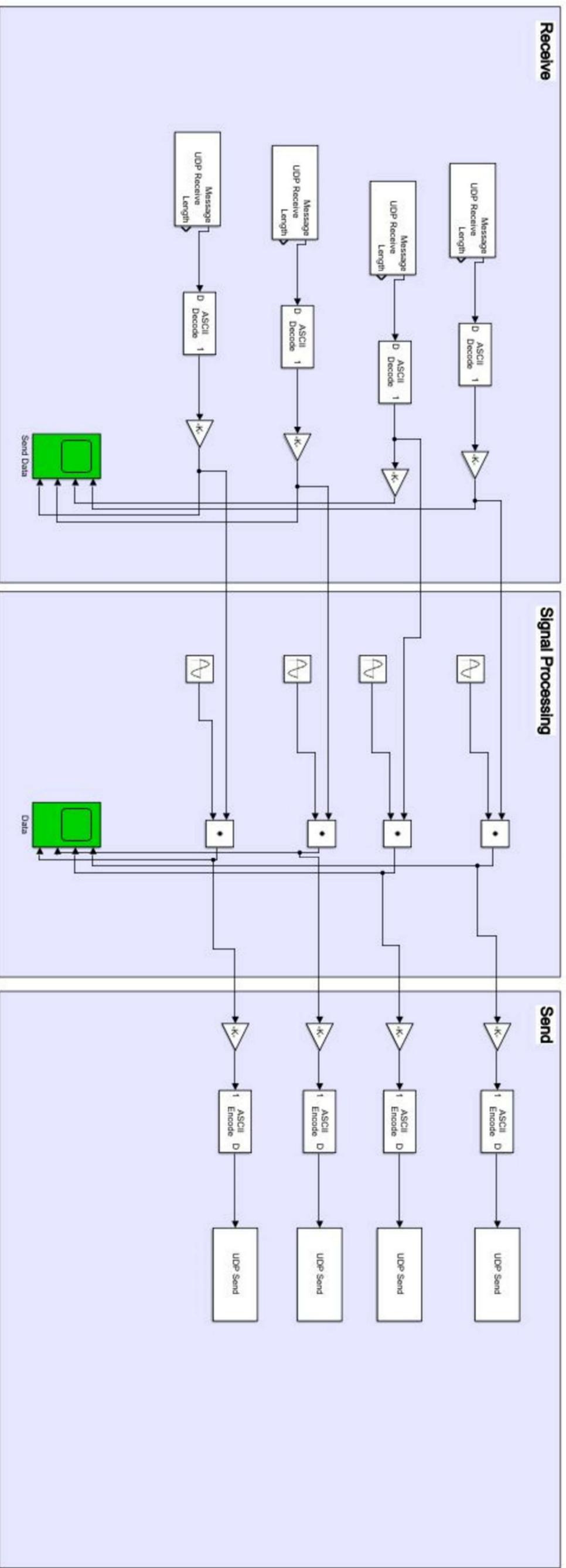
Discrete
5e-05 s.

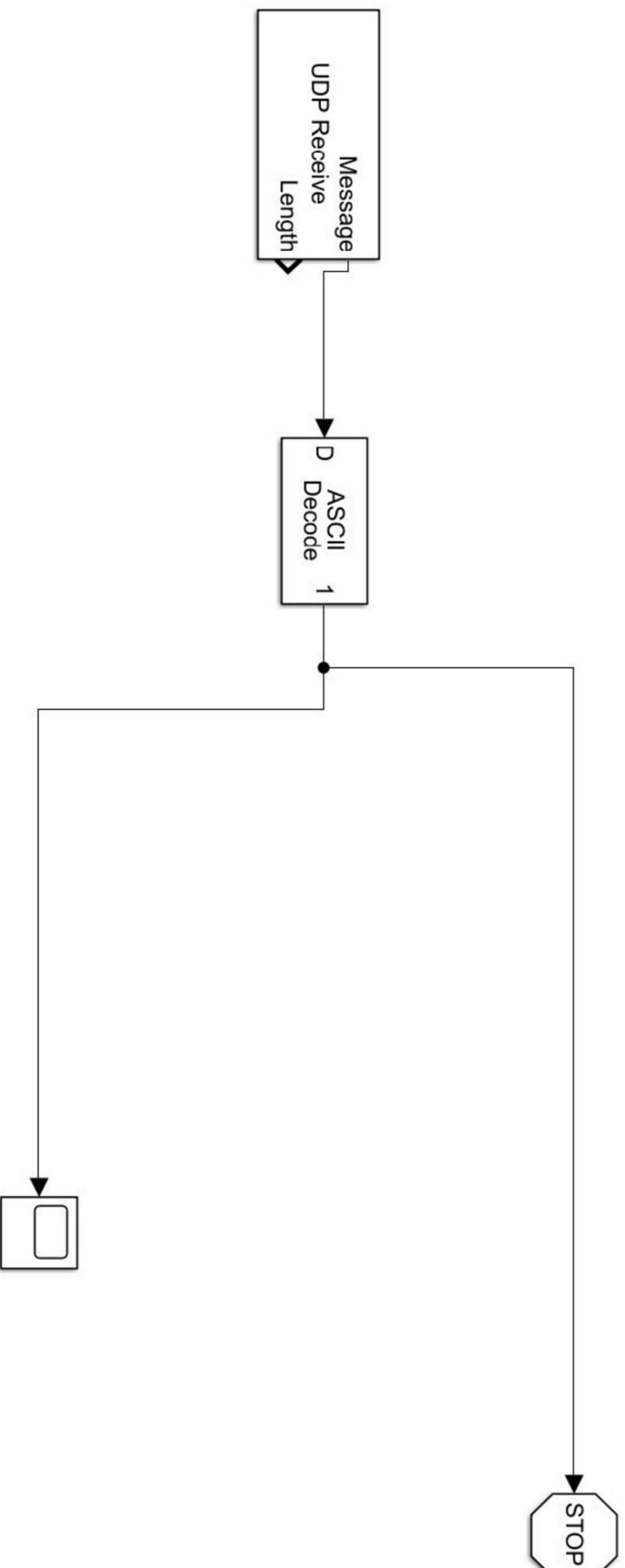
SIMULADOR
- InitFcn. Ts=50e-6; Ts2=250e-6;
Model Initialization

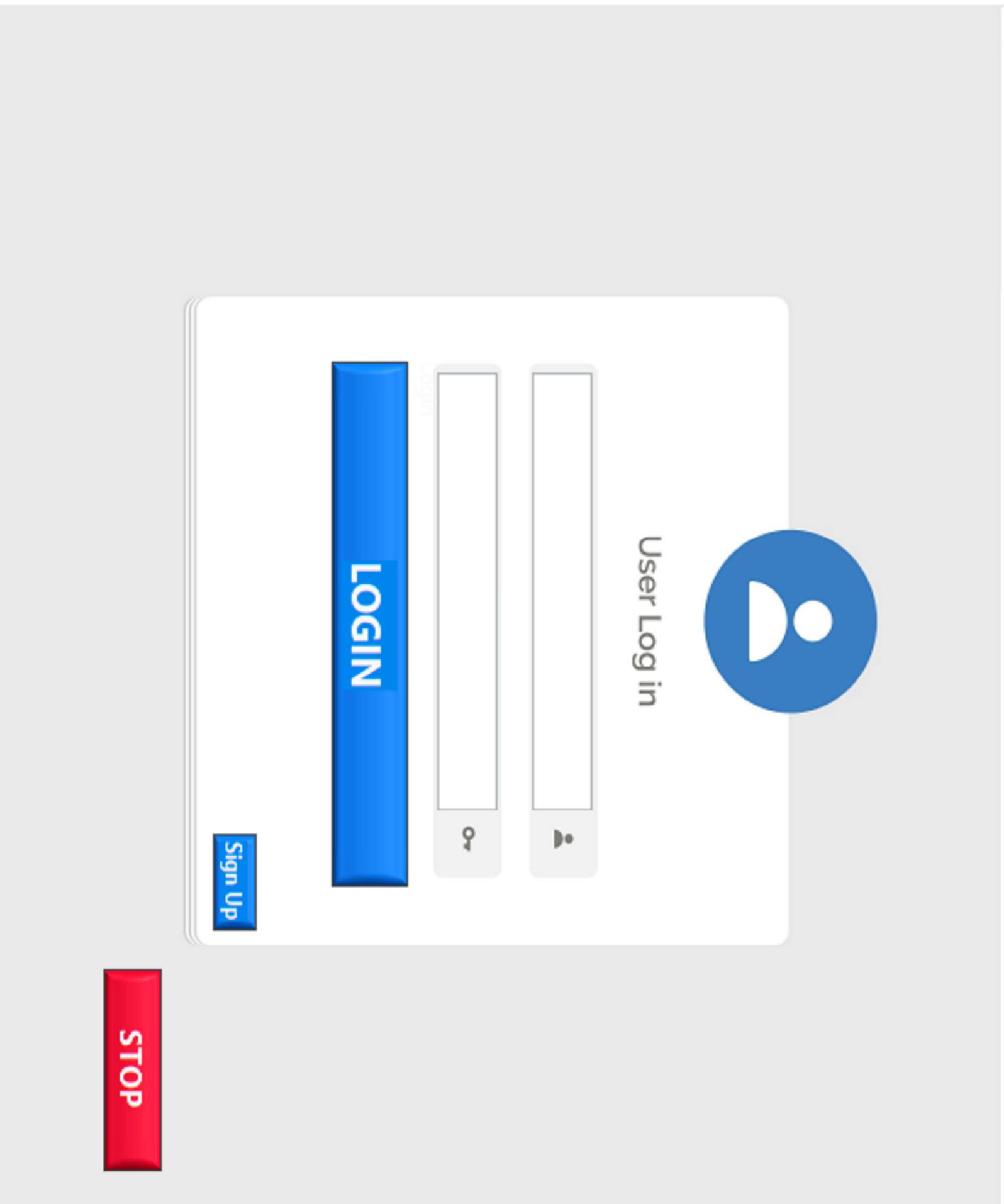














PROMOTION
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS

SIGN UP

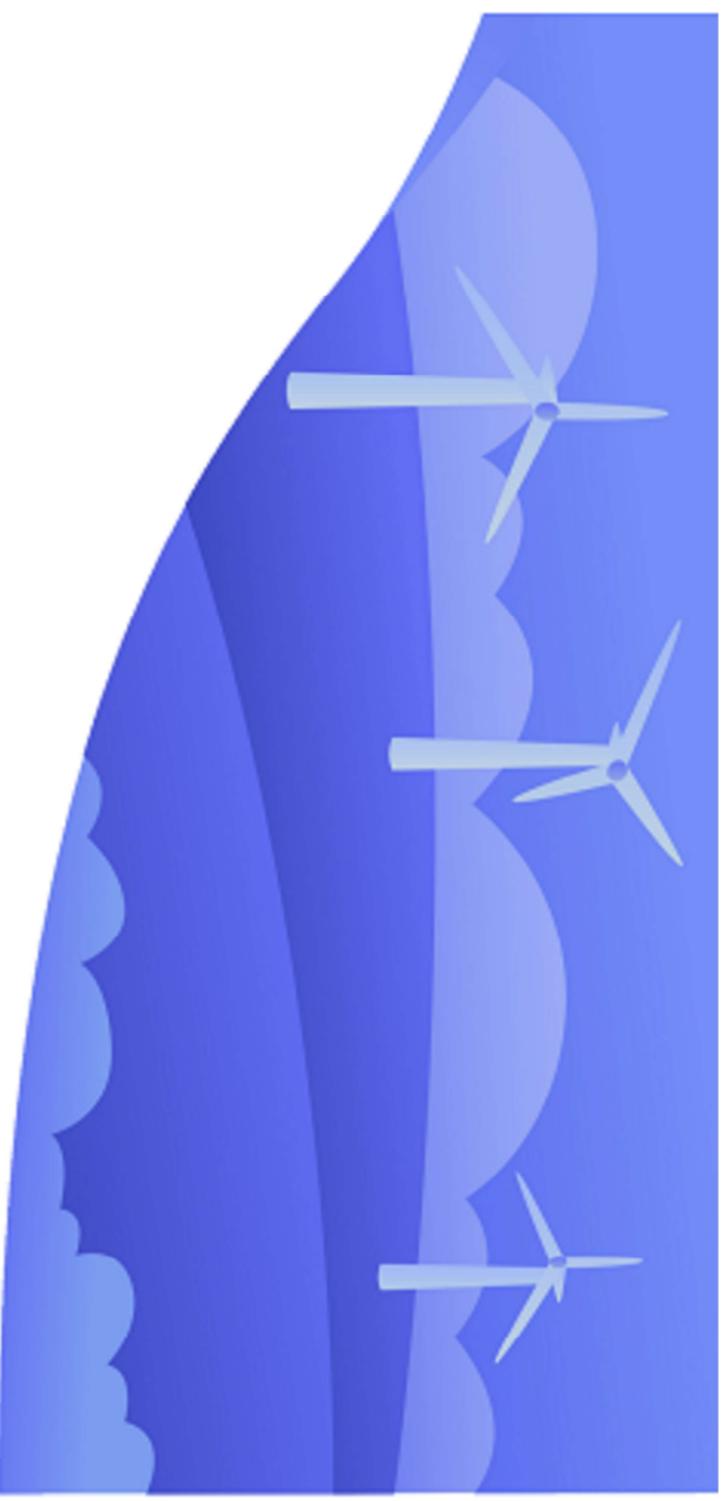
The image shows a user sign-up interface. It consists of five input fields arranged vertically, each with a corresponding icon to its right: a person icon, a telephone icon, an envelope icon, a padlock icon, and a musical note icon. Below these fields is a large blue button with the text "SIGN UP".



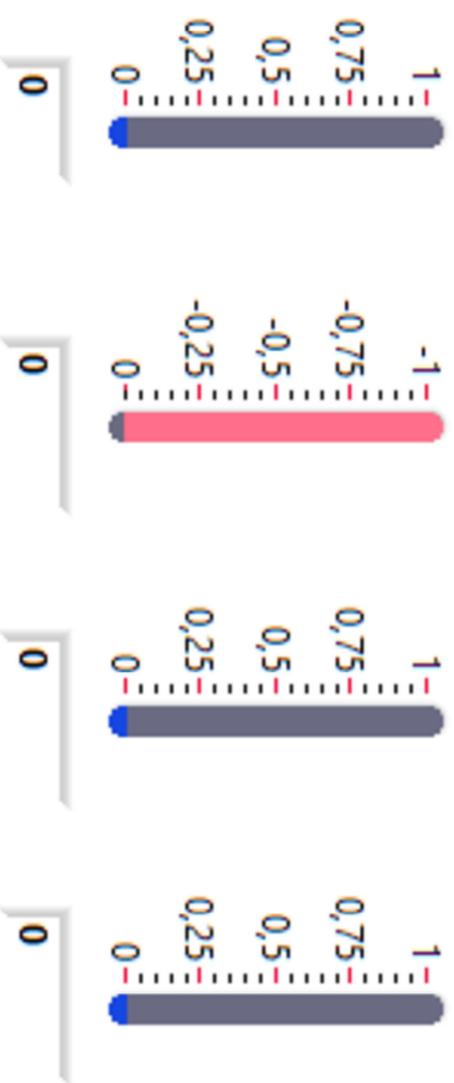
PROMOTION
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS

MAIN MENU

- Data History
- Real Time Data
- Breaker Control
- Error Log
- Simulation
- Help



SIMULATOR CONNECTION



Filename 

BACK



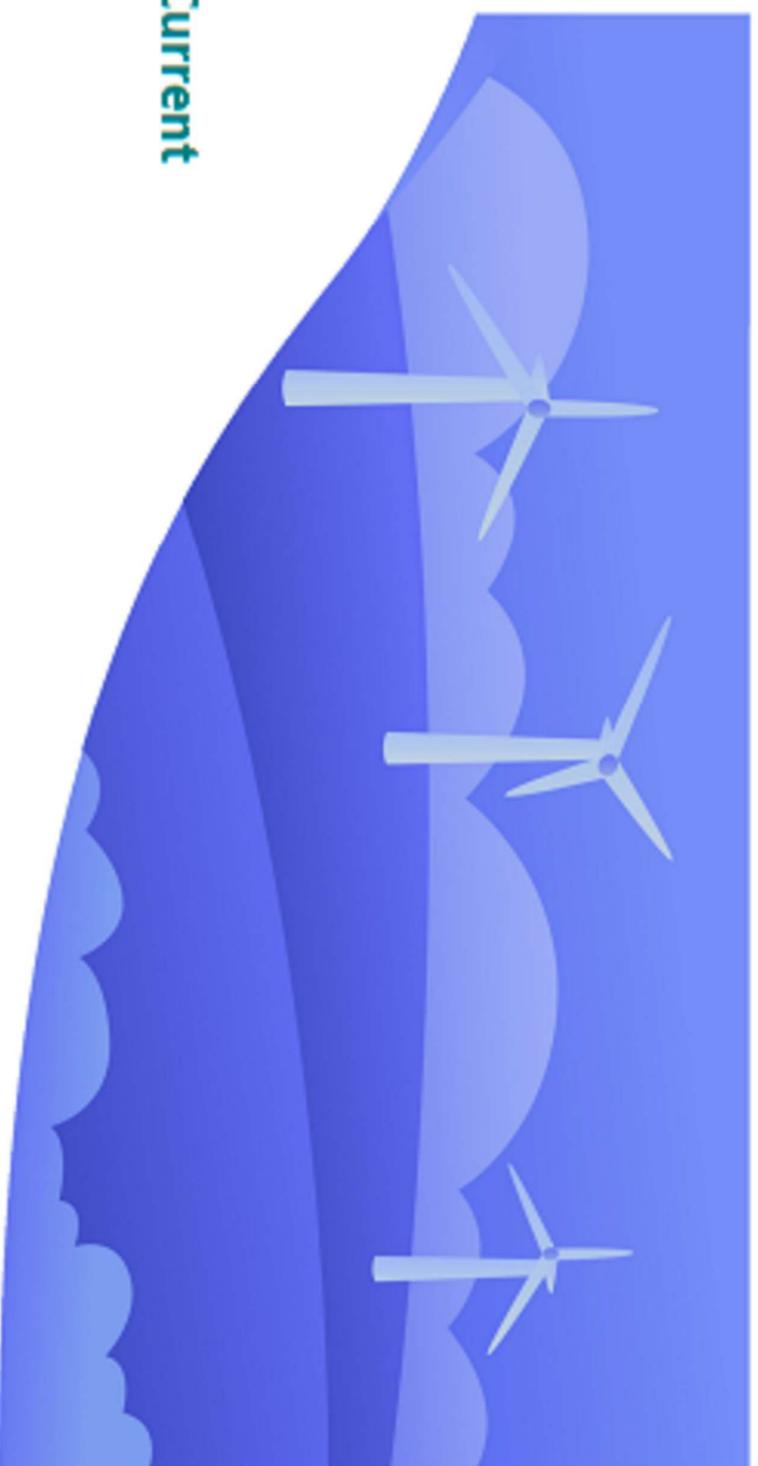
PROMOTION
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS

REAL TIME



Active Power Reactive Power Voltage Current

ACTIVE POWER



SIMULATOR CONNECTION



BACK

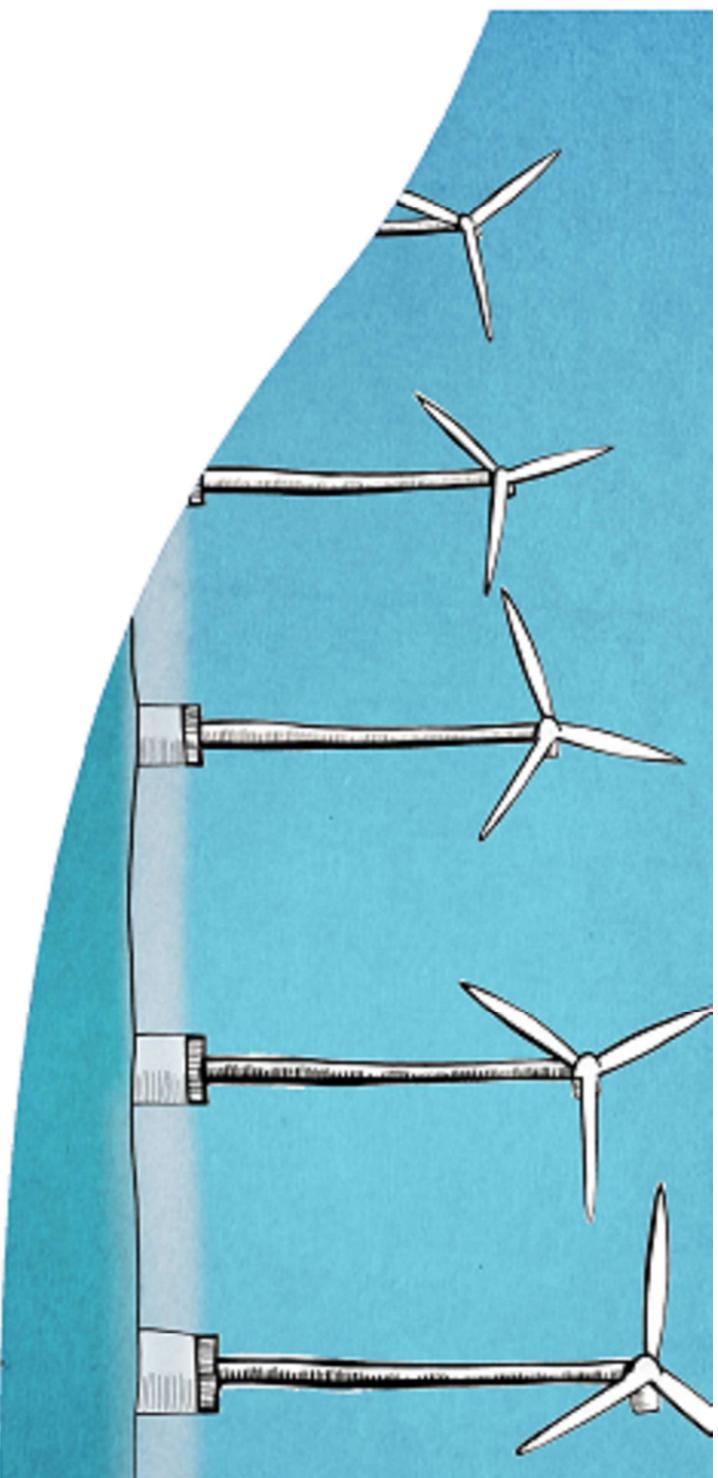


PROMOTION
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS

SIMULADOR

Control Graph

Control



Current

 A

Voltage

 V

Active Power

 MW

Reactive Power

 MVA

STOP



PROMOTION
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS

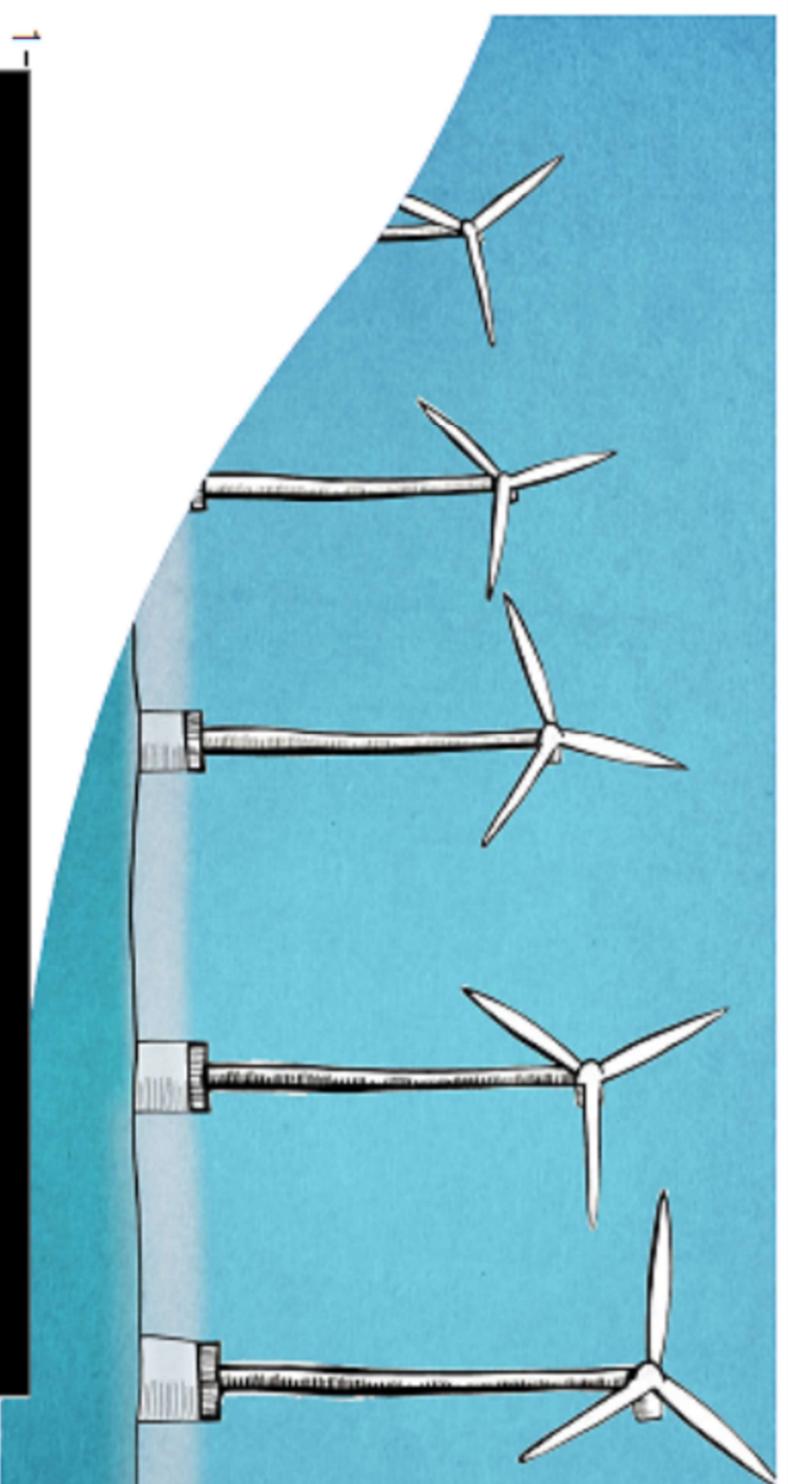
SIMULADOR

Control Graph

Graph

- Voltage
- Off/On
- Active Power
- Off/On
- Current
- Off/On
- Reactive Power
- Off/On

- Voltage
- Current
- Active Power
- Reactive Power



TRABAJO FINAL DE GRADO EN INGENIERIA EN TECNOLOGIAS INDUSTRIALES
UNIVERSITAT POLITÈCNICA DE VALÈNCIA



Proyecto: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SUPERVISIÓN, CONTROL Y ADQUISICIÓN DE DATOS (SCADA) PARA UN PROTOTIPO A ESCALA DE PARQUE EÓLICO.

Plano: "Simulator" Interfaz de usuario 2/2
Autor: Julio Yustas Talamantes

Fecha: Agosto 2020
Escala:

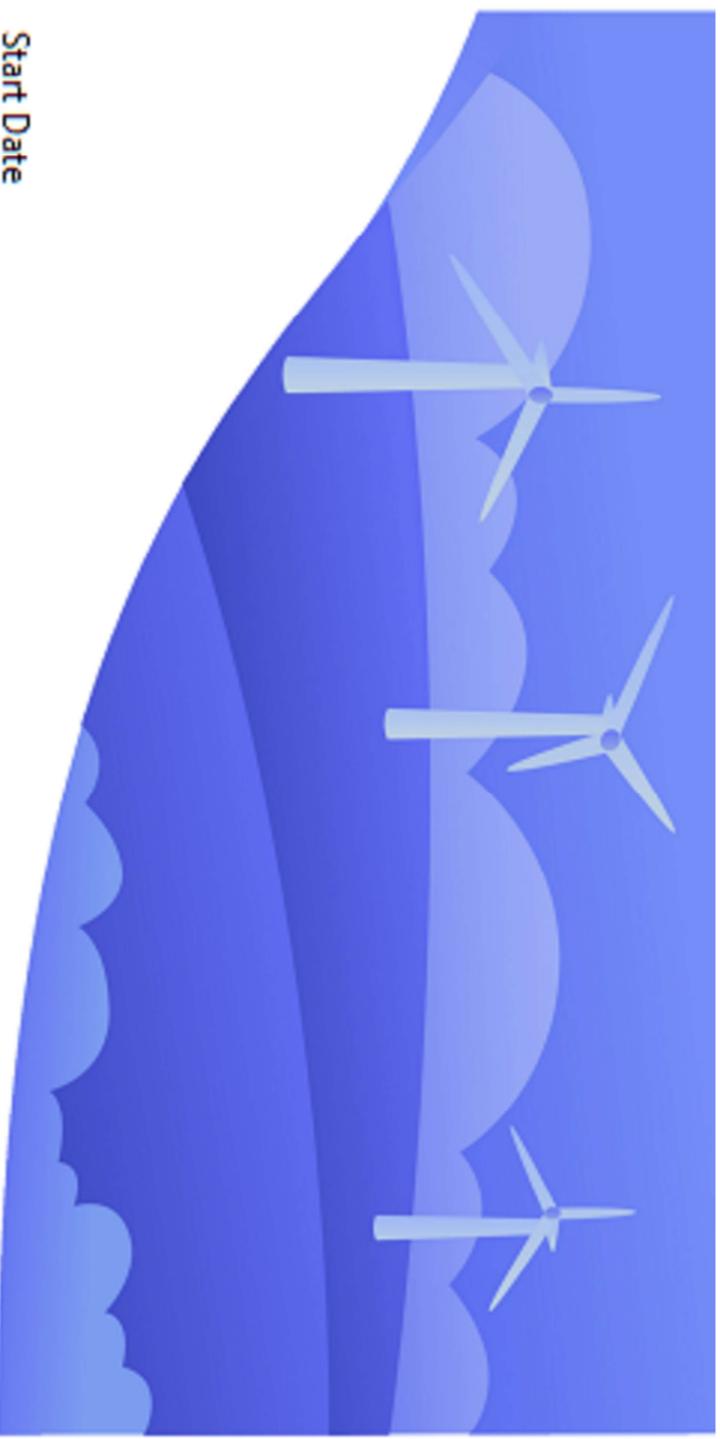
Nº Plano: **22**



PROMOTION
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS

DATA HISTORY

Plot



START DATE

OPEN CALENDAR

Start Date
00:00:00,000
DD/MM/YYYY

CONFIRM DATE



END DATE

OPEN CALENDAR

End Date
00:00:00,000
DD/MM/YYYY

CONFIRM DATE

BACK



TRABAJO FINAL DE GRADO EN INGENIERIA EN TECNOLOGIAS INDUSTRIALES
UNIVERSITAT POLITÈCNICA DE VALÈNCIA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL VALENCIA

Proyecto:

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SUPERVISIÓN, CONTROL Y ADQUISICIÓN DE DATOS (SCADA) PARA UN PROTOTIPO A ESCALA DE PARQUE EÓLICO.

Plano:

"Data History" Interfaz de usuario 1/3

Autor:

Julio Yustas Talamantes

Fecha:

Agosto 2020

Escala:

Nº Plano:

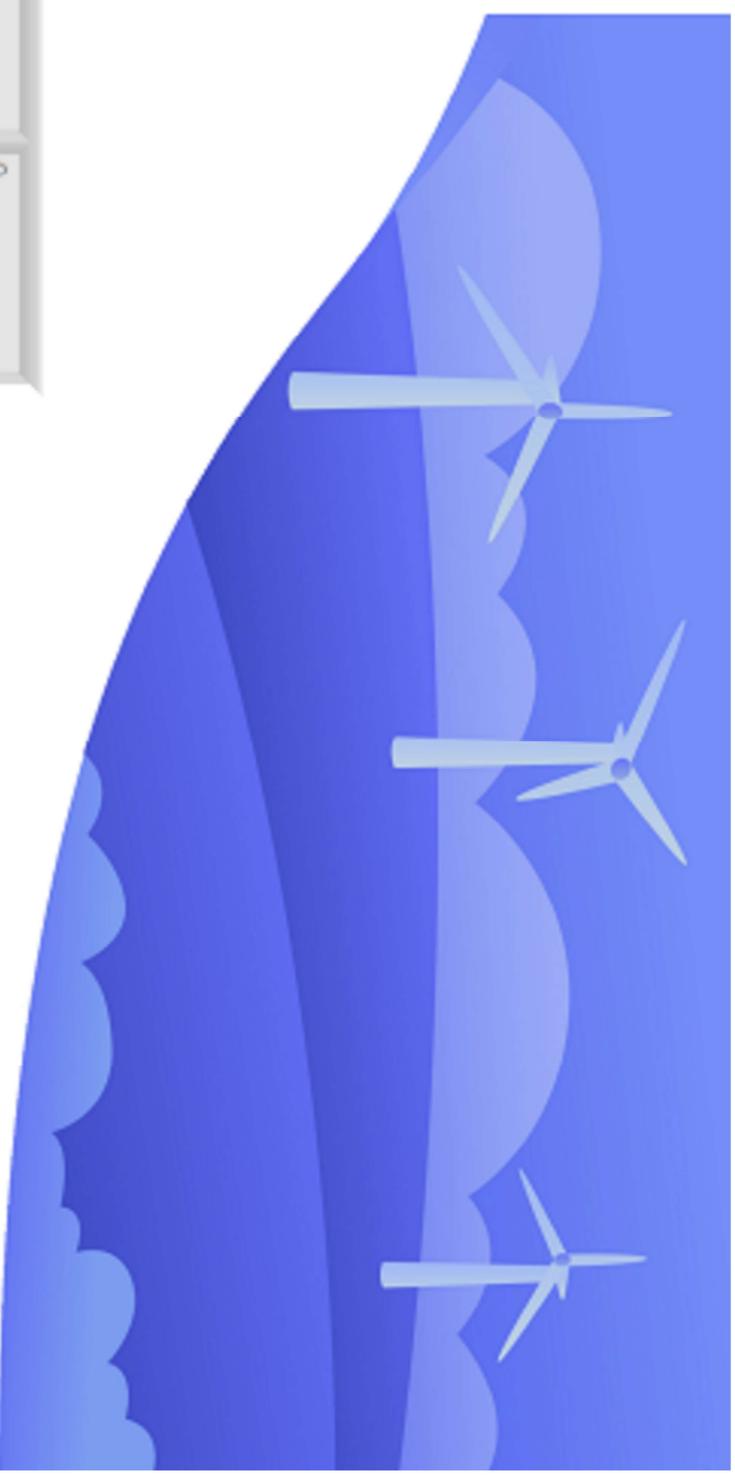
23



PROMOTION
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS

DATA HISTORY

Data Graph



0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Do you want to generate a data file to see all? ●

Generate

BACK

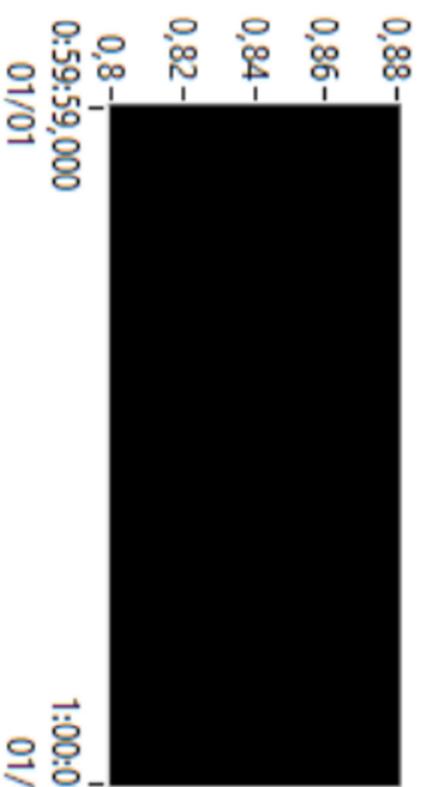


PROMOTION
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS

DATA HISTORY

Data Graph

Active



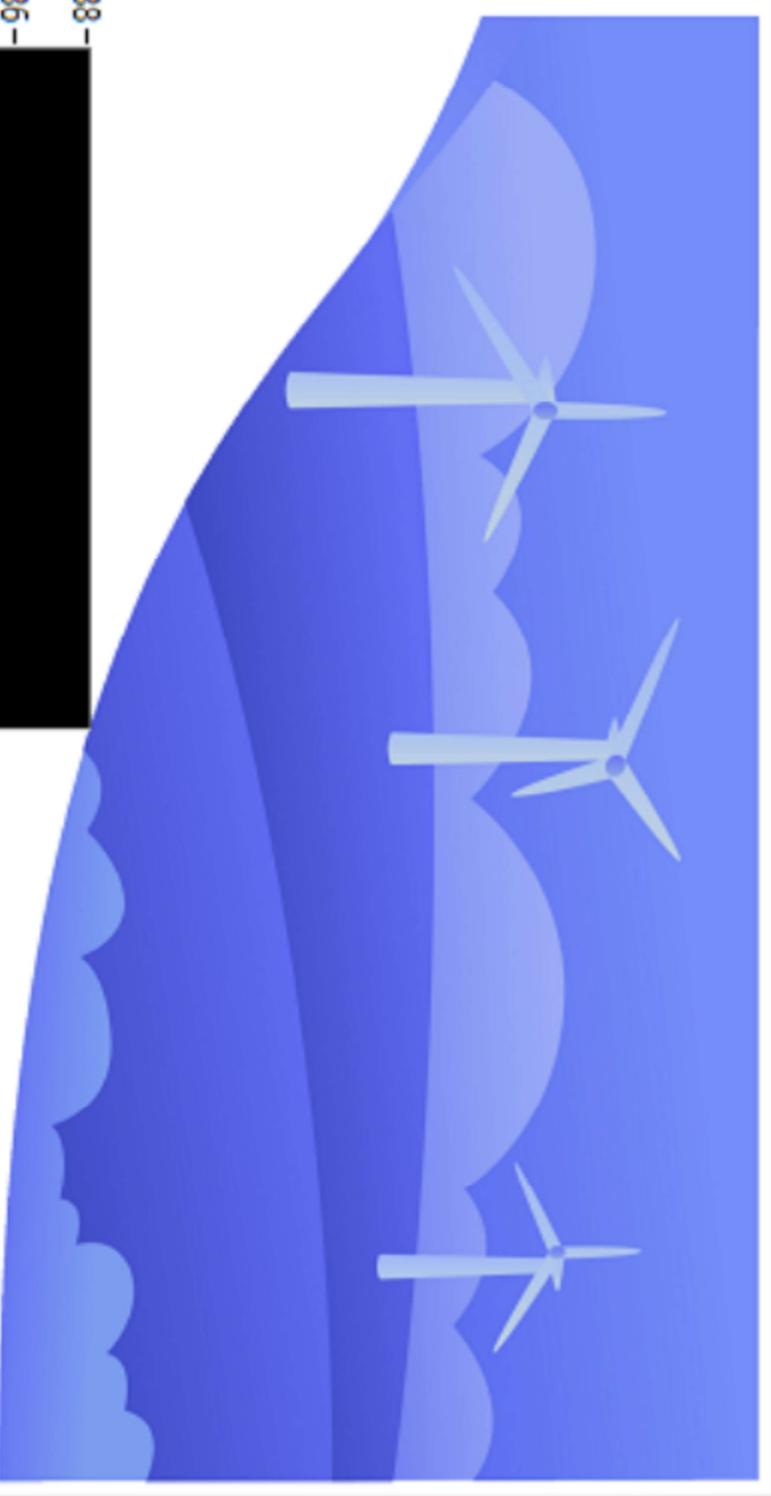
Voltage



Reactive



Current



Begin Try

0

End Try

0

BACK

Enter the desired date:

00:00:00,000

DD/MM/YYYY

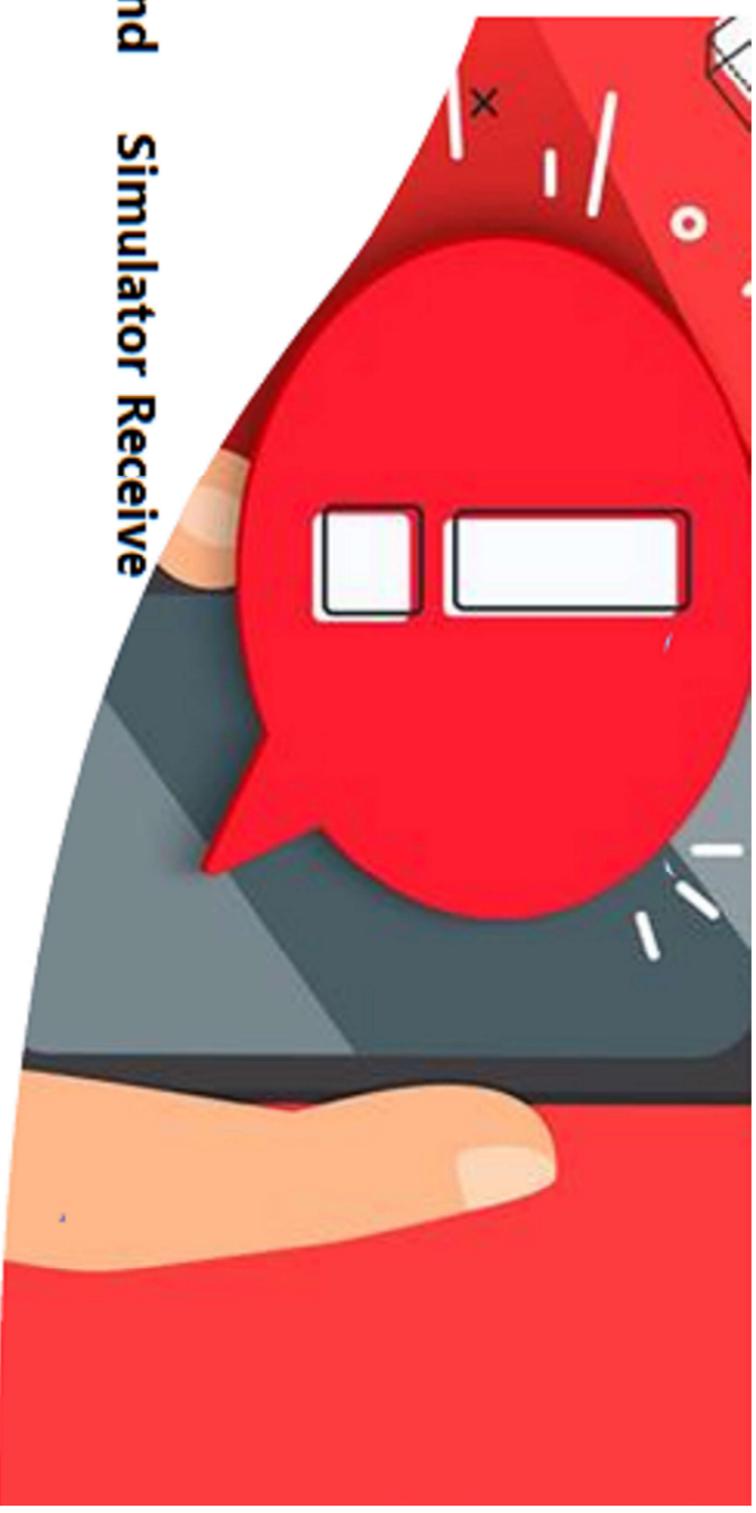
Close window



PROMOTION
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS

ERROR LOG

Real Time Data His. / Breaker Simulator Send Simulator Receive



Real Time Voltage Error

status code d source

Real Time Active Power Error

status code d source

Real Time Reactive Power Error

status code d source

Real Time Current Error

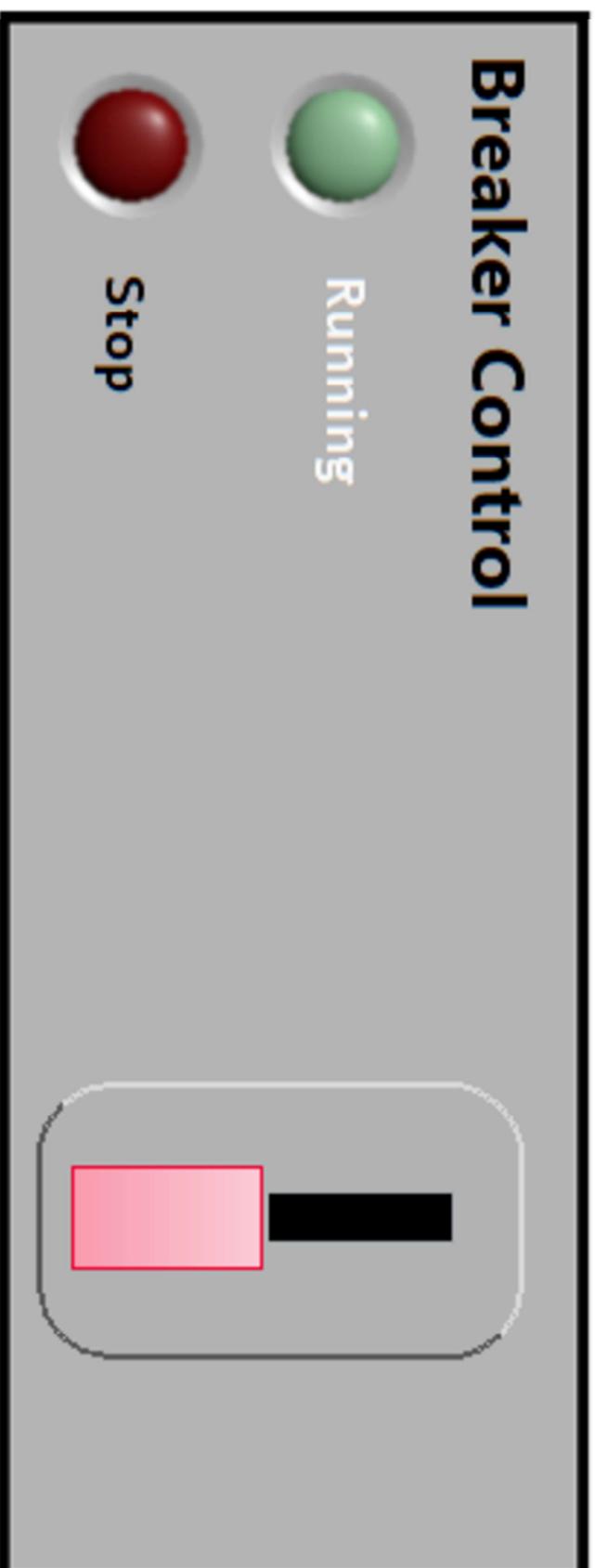
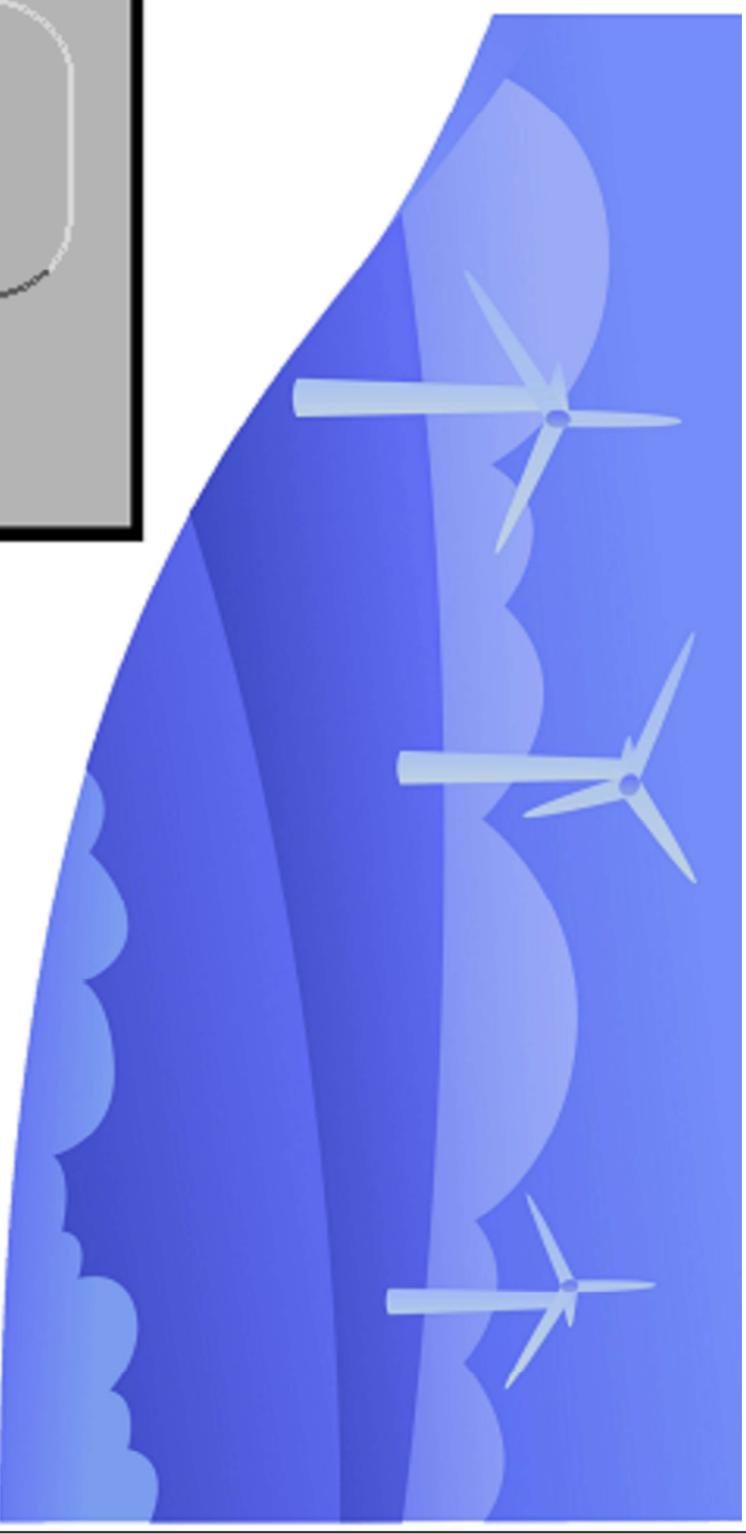
status code d source

BACK



PROMOTION
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS

BREAKER





PROMOTION
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS

HELP

This program is part of the European **PROMOTION** project.
PROMOTION seeks to develop meshed HVDC offshore grids on the basis of cost-effective and reliable technological innovation.

For more information about the project



For more information about the program



Soledad Inmaculada Bernal Pérez Profesora Contratada Doctor PDI



Julio Yustas Talamantes



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA



Parte V

Manuales de usuario

Capítulo V.1

Manual de usuario (Castellano)

JULIO YUSTAS TALAMANTES



MANUAL DE USUARIO

CONTENIDO



1
ACERCA DEL
PROGRAMA

2
ESTRUCTURA DEL
PROGRAMA

3
INTERFAZ DE USUARIO

4
REQUISITOS TÉCNICOS

5
INSTALACIÓN

6
INICIAR APLICACIÓN

7
SIGN UP

CONTENIDO



8
LOGIN

9
MENÚ PRINCIPAL

10
MONITORIZACIÓN

11
SIMULACIÓN DE
SEÑALES

12
CONSULTA EN LA BASE
DE DATOS

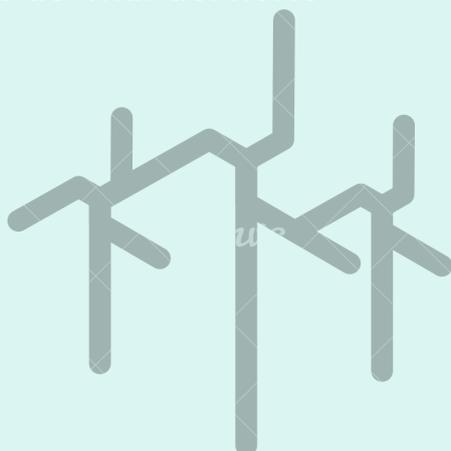
13
CONTROL DEL
"BREAKER"

14
CÓDIGOS DE ERROR Y
FAQS



ACERCA DEL PROGRAMA

El presente programa ha sido desarrollado como parte del Trabajo de Final de Grado de Julio Yustas Talamantes; dicho trabajo se incluye como integrante de la investigación de la Universitat Politècnica de València para el proyecto eólico europeo **"PROMOTioN"** cuyo objetivo principal es explotar todos los recursos eólicos del mar del norte.



Este programa se ha desarrollado para su uso como aplicación de monitorización de parques eólicos marinos.

Su cometido principal es proporcionar al operario de una interfaz clara y sencilla para el correcto desempeño de su trabajo.

A su vez, se postula como los cimientos desde los cuales construir un programa mucho más sofisticado y para el cual se proporciona el "Manual de Programador".



ACERCA DEL PROGRAMA

Los autores de esta aplicación han sido Julio Yustas Talamantes bajo la tutorización de la Doctora Soledad Inmaculada Bernal Pérez (Doctora Ingeniera Industrial).

El presente trabajo se ha desarrollado gracias a la colaboración de la **Universitat Politècnica de València** y el Instituto Universitario de Automática e Informática Industrial (ai2).



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

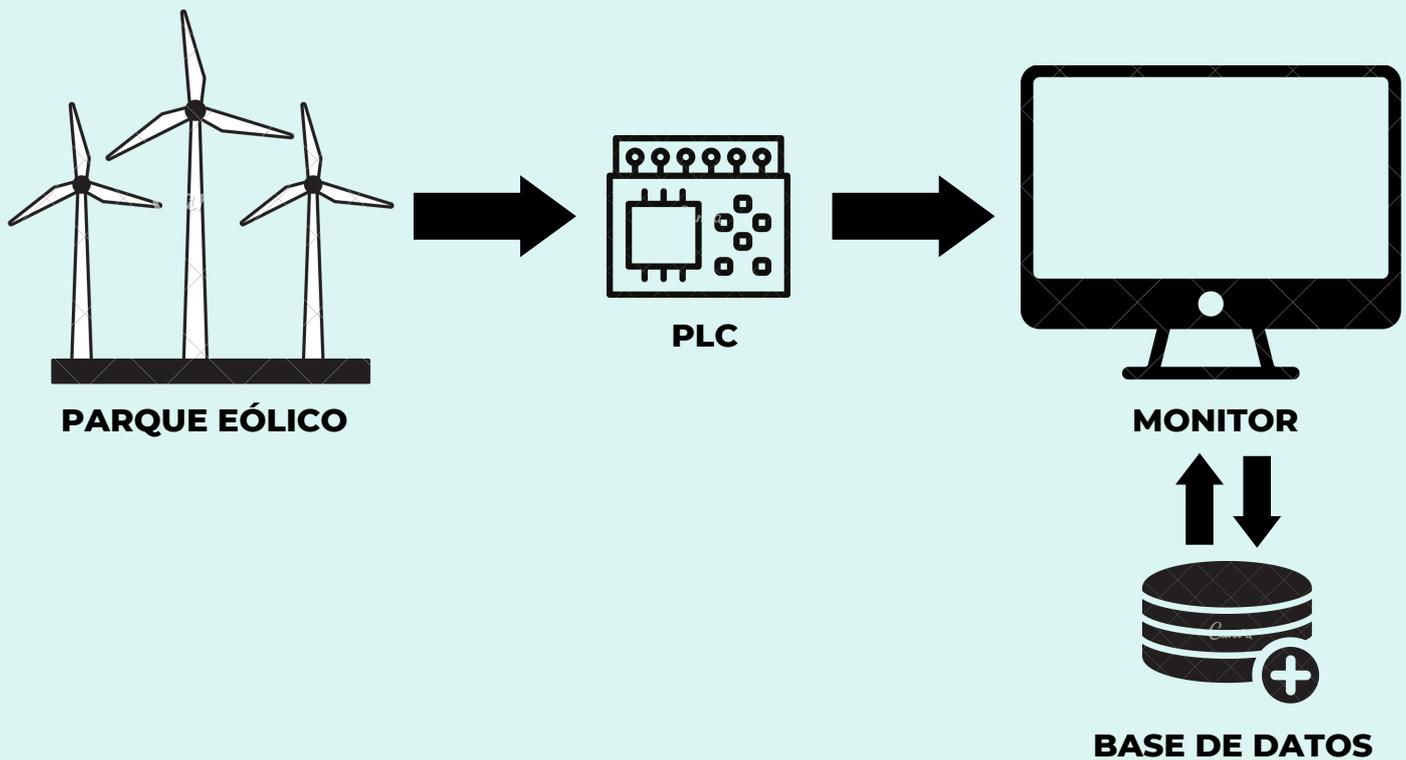


ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA



PROMOTiON
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS





ESTRUCTURA DEL PROGRAMA

Se trata de un sistema SCADA (Supervisión, Control y Adquisición de Datos).

El objetivo principal es controlar un parque eólico desde otro lugar de forma remota, para ello necesitaremos tres elementos clave: el parque eólico, los PLCs y el sistema de monitorización.

Las conexiones entre dichas partes son una parte fundamental para el correcto desempeño del presente programa.

La unión entre el parque eólico y los sensores se realizará mediante un cableado.

Dichos sensores estarán conectados a su vez a la red, con el objetivo de permitir la comunicación mediante protocolo UDP con el simulador o programa SCADA,

Es trabajo del programa SCADA, tratar estos datos y acomodarlos a un entorno atractivo para el trabajador.

A su vez, almacenará dichos datos en una base de datos para permitir su posible consulta posterior.

INTERFAZ DE USUARIO

El programa cuenta con una interfaz de usuario sencilla y atractiva realizada con la ayuda de LabVIEW y Photoshop.

La navegación es sencilla: para avanzar entre las diversas pestañas solo tenemos que elegir la modalidad desde el menú principal o "main menu".

En el caso de querer retroceder, se cuenta en todo momento con un botón de "back" que nos envía a la página anterior.

Para finalizar el programa simplemente tenemos que volver a la pestaña de identificación y darle click al botón de "stop" o en su defecto utilizar el clásico botón de cerrar pestañas de windows.



REQUISITOS TÉCNICOS

DESDE UN SIMULADOR VIRTUAL:



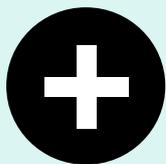
SISTEMA OPERATIVO

Windows 7, Ubuntu 16.04 o macOS Sierra en 64 bits.



NAVEGADOR Y CONEXIÓN RED

Internet Explorer 11, Google Chrome 58 o Firefox 53



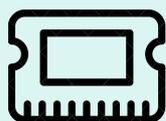
MATLAB

R2017b o Superior



PROCESADOR

Intel Core I5 o superior



MEMORIA RAM

Memoria RAM 8G o superior



TARJETA DE SONIDO Y ALTAVOCES

REQUISITOS TÉCNICOS

DESDE UNA GRANJA EÓLICA REAL:



SISTEMA OPERATIVO

Windows 7, Ubuntu 16.04 o macOS Sierra en 64 bits.

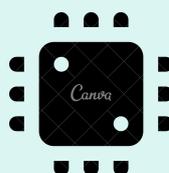


NAVEGADOR

Internet Explorer 11, Google Chrome 58 o Firefox 53

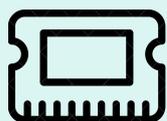


CONEXIÓN RED



PROCESADOR

Intel Core I5 o superior



MEMORIA RAM

Memoria RAM 8G o superior



TARJETA DE SONIDO Y ALTAVOCES

INSTALACIÓN

La instalación del presente programa es bastante sencilla, para ello solo necesitamos seguir una serie de pasos:

La primera etapa consiste en descargar el archivo *promotion.exe* presente en la web de PROMOTioN o en el repositorio adjunto al pie de esta misma página.

Una vez descargado el archivo solo queda darle doble click para ejecutarlo e instalarlo.

Si se desea utilizarlo junto al simulador, deberemos instalar en esa misma computadora, o en otra conectada a la misma red WIFI, el programa MATLAB y ejecutar los simuladores:
Banco_Pruebas_Breaker.mdl ,
Banco_Pruebas_Definitivo.mdl y
simulator.mdl

Si, por el contrario, se desea utilizar un prototipo a escala o una granja eólica real, deberemos de realizar una sincronización de los puertos de envío de los PLCs y los puertos de entrada del programa de acuerdo a la siguiente tabla:

Variable a controlar	Puerto
Tensión	8811
Corriente	8822
Potencia Activa	8833
Potencia Reactiva	8844

Dicha sincronización se realizará en el prototipo a escala o el parque eólico, es decir, siempre sobre el PLC y no sobre el programa.

Control	Envío
Tensión	7700
Corriente	7711
Potencia Activa	7722
Potencia Reactiva	7733

La anterior configuración se usará para la funcionalidad de simulador; dichos puertos son los de envío por parte del simulador y/o PLC.

INICIALIZACIÓN

Para garantizar la correcta inicialización del programa, deberemos comprobar que ambas partes: el ordenador ("host" del sistema SCADA) y los PLCs disponen de conexión a internet.

A su vez, se tendrá que comprobar, de forma análoga a lo expuesto en el apartado anterior, que existe una correcta disposición de los puertos de entrada y salida.

Este último paso es fundamental para garantizar el éxito de la comunicación mediante el protocolo UDP.

Por otro lado, el usuario deberá registrarse con los credenciales proporcionados por la empresa que serán únicos e intranferibles.

En el caso de no disponer de ellos, se podrá dar de alta en la base de datos de operarios autorizados mediante la pestaña de "sign-up".

Esta última pestaña se explicará con más detenimiento en los siguientes puntos de este mismo manual.

A continuación se vuelve a adjuntar la tabla de conexiones UDP:

Variable a controlar	Puerto
Tensión	8811
Corriente	8822
Potencia Activa	8833
Potencia Reactiva	8844

Por último, para empezar a recibir datos del parque eólico, lo único que resta es pulsar sobre el interruptor de inicio de conexión, presente en la pestaña de "main menu".

Para la recepción de datos por parte del software se tendrá que modificar la dirección IP de los protocolos de envío en las pestañas de "Simulator" y "Breaker".

SIGN-UP

La pestaña "sign-up" permite que los usuarios puedan registrarse dentro de la base de datos de clientes.

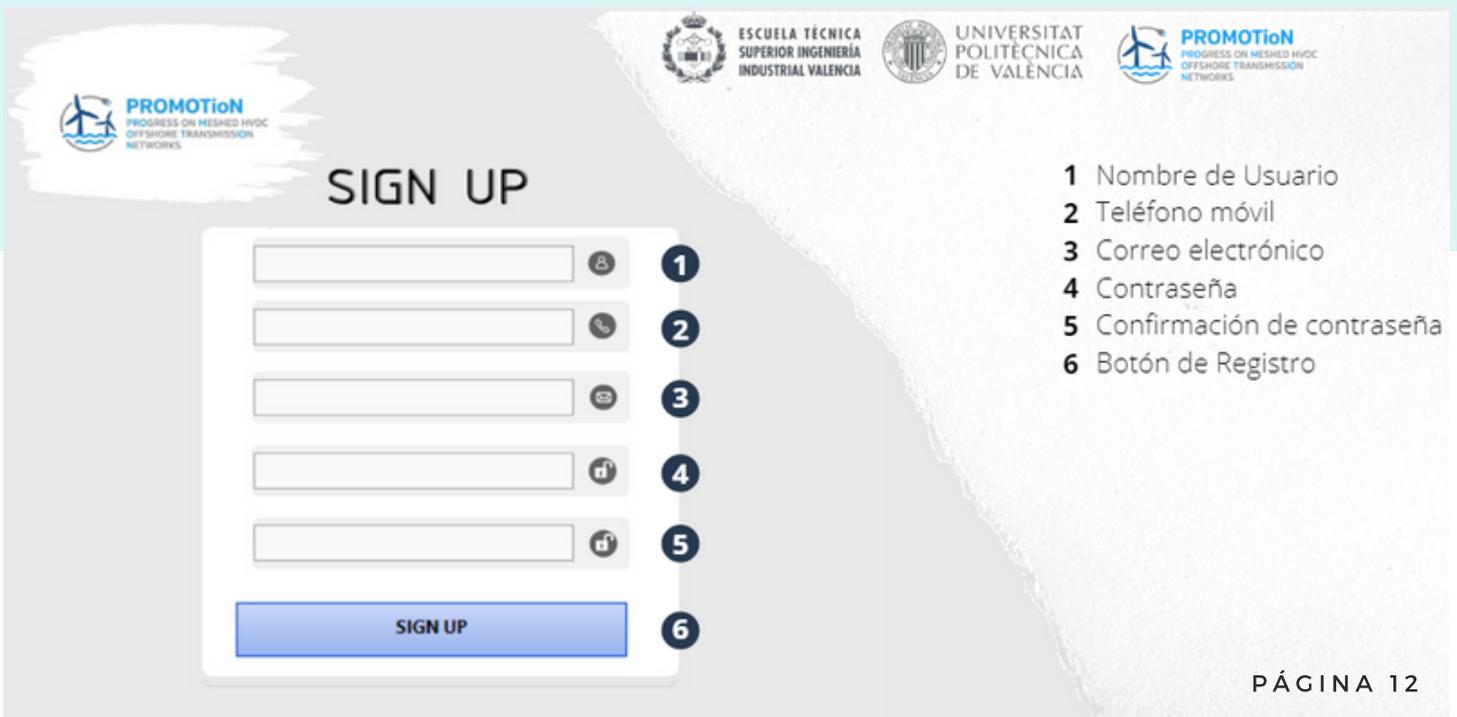
Se puede acceder a esta "front view" exclusivamente desde la pestaña de identificación.

Una vez el usuario se haya dado de alta en la base de datos recibirá un mensaje al correo facilitado confirmando su registro.

En ese momento, el programa devolverá al usuario a la ventana de "login" para que el cliente pueda identificarse y hacer uso todo el potencial del programa

Se notificará error en los siguientes casos: si las contraseñas no coinciden, si el correo ya ha sido previamente registrado y si no se rellenan las casillas imprescindibles.

Se dispone a continuación, de una imagen explicativa de la "front view", identificando cada componente de la misma:



The image shows a 'SIGN UP' form with the following components:

- 1 Nombre de Usuario
- 2 Teléfono móvil
- 3 Correo electrónico
- 4 Contraseña
- 5 Confirmación de contraseña
- 6 Botón de Registro

Logos at the top of the form include: ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA, UNIVERSITAT POLITÈCNICA DE VALÈNCIA, and PROMOTioN PROGRESS ON MESHED HVDC OFFSHORE TRANSMISSION NETWORKS.

LOGIN

Esta ventana permite al usuario identificarse mediante un nombre de usuario y una contraseña. Dichas credenciales pueden haber sido registradas por el mismo operario o venir proporcionadas por el programador mediante una escritura directa sobre la base de datos utilizada por el software.

Como se ha mencionado, se ofrece la posibilidad de que el usuario registre sus datos manualmente. Para ello, debe pulsar sobre el botón "sign-up" localizado en la esquina inferior derecha de la ventana.

En principio, el presente código se encuentra acabado y depurado, por lo que, de no cambiar la base de datos, no sería necesario modificar ningún aspecto.

Se dispone a continuación, de una imagen explicativa de la "front view", identificando cada componente de la misma:

The screenshot shows a web interface for user login. At the top, there are logos for 'PROMOTiON' and 'ESCUOLA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA'. The main content area features a 'User Log in' form with two input fields, a 'LOGIN' button, and a 'Sign Up' button. A red 'STOP' button is located at the bottom center. A list on the right side of the image identifies the numbered callouts: 1 (Username field), 2 (Password field), 3 (Login button), 4 (Sign Up button), and 5 (Stop button).

- 1 Nombre de Usuario
- 2 Contraseña
- 3 Botón de Identificación
- 4 Botón de Registro
- 5 Botón de "Stop"

MENÚ PRINCIPAL

La pestaña "main menu" permite al usuario elegir entre las diferentes funcionalidades del programa: ver datos en tiempo real, consultar el histórico de datos, entrar al simulador, controlar el interruptor automático, consultar los errores y abrir la pestaña de información del programa.

Además, se incluye una breve información sobre los datos en tiempo real, aunque esta funcionalidad se explicará con más detalle en el punto dedicado a la ventana de datos en tiempo real.

Para activar la sincronización, se deberán cumplir todos los parámetros anteriormente citados y accionar la clavija.

Se puede modificar el nombre de la base de datos aunque no se recomienda pues, de cambiarlo, habría que realizar una serie de modificaciones en el código.

1 Botones de menú

2 Clavija de conexión

3 Nombre base de datos

4 Botón de "back"

5 Datos en tiempo real

MONITORIZACIÓN

La ventana "real time" habilita al usuario y/u operario del software obtener el control total sobre el aerogenerador (o conjunto de aerogeneradores).

Se obtienen en todo momento datos sobre la tensión, intensidad, potencia activa y potencia reactiva, representandose en tres formatos: en forma numérica, gráficamente y en forma de barra.

Esta pestaña ofrece un menú interactivo, para que el usuario pueda seleccionar qué variable desea monitorizar.

Además, incorpora un botón de retroceso y una clavija de encendido y apagado de forma análoga a la ventana de "main menu".

1 Clavija
2 Menú
3 Botón de "back"
4 Indicadores

SIMULACIÓN DE SEÑALES

Otra de las funcionalidades del presente software es la simulación de señales; lo que permite generar una especie de simulador o banco de pruebas.

Para esta funcionalidad, igual que para las anteriores, se puede trabajar contra un simulador virtual o un prototipo a escala.

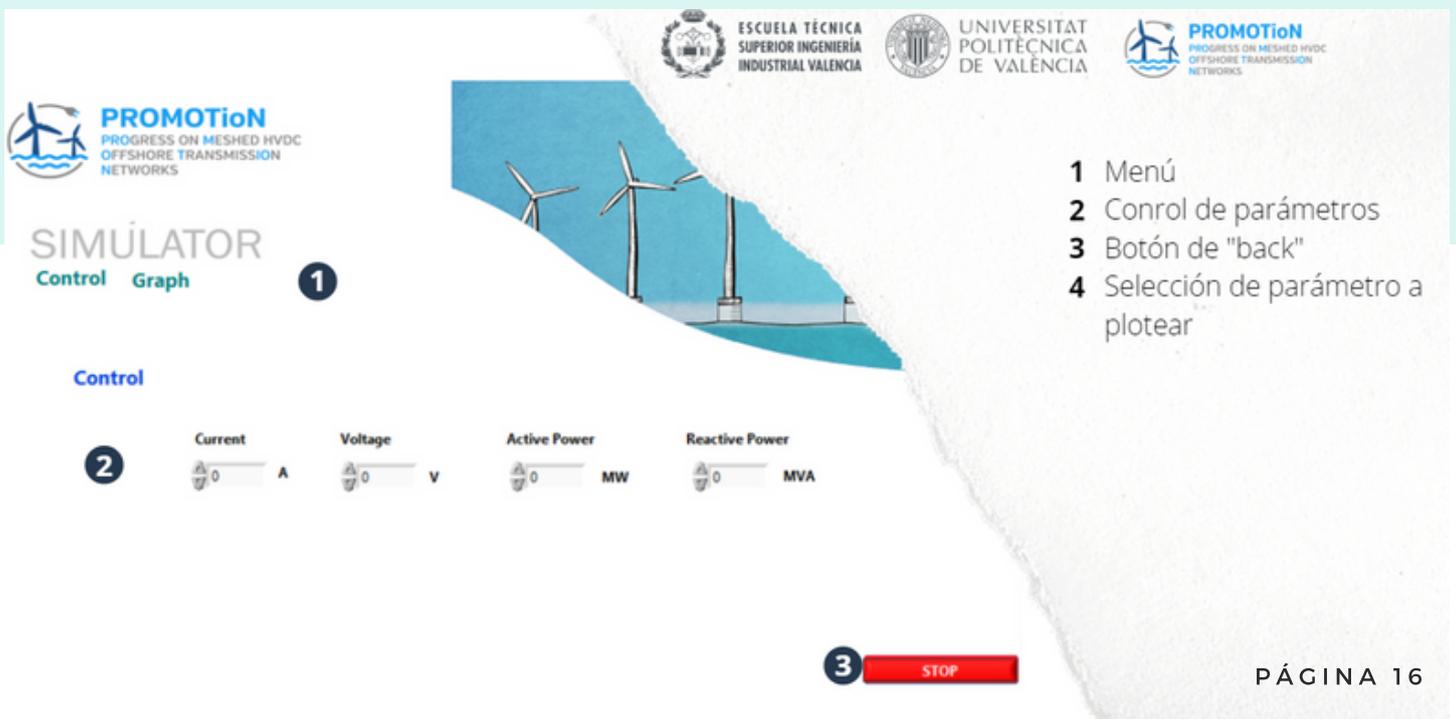
Con el objetivo de poder hacer uso de esta propiedad, se ha generado el código

Banco_Pruebas_Definitivo.mdl en código MATLAB.

Esta pestaña ofrece un menú interactivo, para que el usuario pueda seleccionar si desea enviar datos o ver el comportamiento de éstos en el sistema.

El apartado de gráficos incorpora una nueva funcionalidad, que habilita al usuario a poder elegir que variable quiere plotear en cada momento. Esta funcionalidad viene acompañada de una leyenda para poder identificar el color asociado a cada parámetro.

Además, incorpora un botón de retroceso.



SIMULACIÓN DE SEÑALES

Otra de las funcionalidades del presente software es la simulación de señales; lo que permite generar una especie de simulador o banco de pruebas.

Para esta funcionalidad, igual que para las anteriores, se puede trabajar contra un simulador virtual o un prototipo a escala.

Con el objetivo de poder hacer uso de esta propiedad, se ha generado el código

Banco_Pruebas_Definitivo.mdl en código MATLAB.

Esta pestaña ofrece un menú interactivo, para que el usuario pueda seleccionar si desea enviar datos o ver el comportamiento de éstos en el sistema.

El apartado de gráficos incorpora una nueva funcionalidad, que habilita al usuario a poder elegir que variable quiere plotear en cada momento. Esta funcionalidad viene acompañada de una leyenda para poder identificar el color asociado a cada parámetro.

Además, incorpora un botón de retroceso.

The screenshot displays the software interface for the PROMOTion project. At the top, there are logos for the Escuela Técnica Superior de Ingeniería Industrial de Valencia, the Universitat Politècnica de València, and the PROMOTion project itself. The main interface is divided into several sections:

- Control Panel:** Located on the left, it features a 'SIMULATOR' title and two tabs: 'Control' and 'Graph'. A red 'STOP' button is visible at the bottom right of the control panel area.
- Graph Section:** On the right, there is a graph titled 'Graph' showing a plot of variables over 'Time'. The y-axis ranges from -1 to 1. A legend below the graph identifies the variables: Voltage (black), Current (red), Active Power (green), and Reactive Power (blue).
- Navigation Elements:** A list on the right side of the screenshot provides a numbered key: 1. Menú, 2. Control de parámetros, 3. Botón de "back", 4. Selección de parámetro a plotear.

CONSULTA EN LA BASE DE DATOS

Esta pantalla ofrece al usuario la posibilidad de consultar los datos recibidos por parte del simulador en un intervalo temporal exacto. Para ofrecer estas funcionalidades, la pestaña dispone de dos botones que abren el calendario de inicio y fin del intervalo y dos botones de confirmación con su respectivo LED.

Una vez seleccionado el intervalo, el usuario deberá pulsar el botón de plotear, es ese mismo momento se abrirá una subpestaña cuya función será encontrar los datos dentro de este intervalo consultando la base de datos.

Cuando ya se hayan encontrado, se proyectarán en forma de tabla y gráficamente, ofreciendo al usuario la disponibilidad de descargar estos datos en un fichero.

Además, incorpora un botón de retroceso.

The screenshot shows the 'DATA HISTORY' interface. At the top left is the 'PROMOTion' logo with the tagline 'PROGRESS ON MESHED HVDC OFFSHORE TRANSMISSION NETWORKS'. Below the logo is a blue 'Plot' button (4). To the right is a map of a coastline with a wind turbine icon. Below the map are two date selection sections. Each section has a 'START DATE' or 'END DATE' label, an 'OPEN CALENDAR' button (1), a date input field showing '00:00:00,000 DD/MM/YYYY', and a green 'CONFIRM DATE' button with a small green LED indicator (2). At the bottom center is a red 'BACK' button (3). On the right side, a 'Close window' dialog box is shown, containing the text 'Enter the desired date:', a date input field with '00:00:00,000 DD/MM/YYYY', a calendar icon, and a 'Close window' button (5). At the top right are logos for 'ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA', 'UNIVERSITAT POLITÈCNICA DE VALÈNCIA', and 'PROMOTion'.

- 1 Calendario
- 2 Clavija de confirmación
- 3 Botón de "back"
- 4 Botón de plotear
- 5 Subpestaña de calendario

CONSULTA EN LA BASE DE DATOS

Esta pantalla ofrece al usuario la posibilidad de consultar los datos recibidos por parte del simulador en un intervalo temporal exacto. Para ofrecer estas funcionalidades, la pestaña dispone de dos botones que abren el calendario de inicio y fin del intervalo y dos botones de confirmación con su respectivo LED.

Una vez seleccionado el intervalo, el usuario deberá pulsar el botón de plotear, es ese mismo momento se abrirá una subpestaña cuya función será encontrar los datos dentro de este intervalo consultando la base de datos.

Cuando ya se hayan encontrado, se proyectarán en forma de tabla y gráficamente, ofreciendo al usuario la disponibilidad de descargar estos datos en un fichero.

Además, incorpora un botón de retroceso.

The screenshot shows the 'DATA HISTORY' interface. At the top left, there is a logo for 'PROMOTion' with the text 'PROGRESS ON MESHED HVDC OFFSHORE TRANSMISSION NETWORKS'. Below the logo, the text 'DATA HISTORY' is displayed. To the right of the logo, there are logos for 'ESCUOLA TÉCNICA SUPERIOR INGENIERIA INDUSTRIAL VALENCIA', 'UNIVERSITAT POLITÈCNICA DE VALÈNCIA', and 'PROMOTion'. A blue graphic of a wind turbine is visible in the background. Below the 'DATA HISTORY' text, there are two tabs: 'Data' and 'Graph'. A red circle with the number '6' is next to the 'Data' tab. Below the tabs, there is a table with 10 columns and 10 rows, all containing the number '0'. A red circle with the number '7' is next to the table. To the right of the table, there is a question: 'Do you want to generate a data file to see all?'. Below the question, there is a green button labeled 'Generate' and a red circle with the number '8' next to it. At the bottom center, there is a red button labeled 'BACK' and a red circle with the number '3' next to it. On the right side of the screenshot, there is a legend with three items: '6 Menú', '7 Tabla de datos', and '8 Botón de generación'.

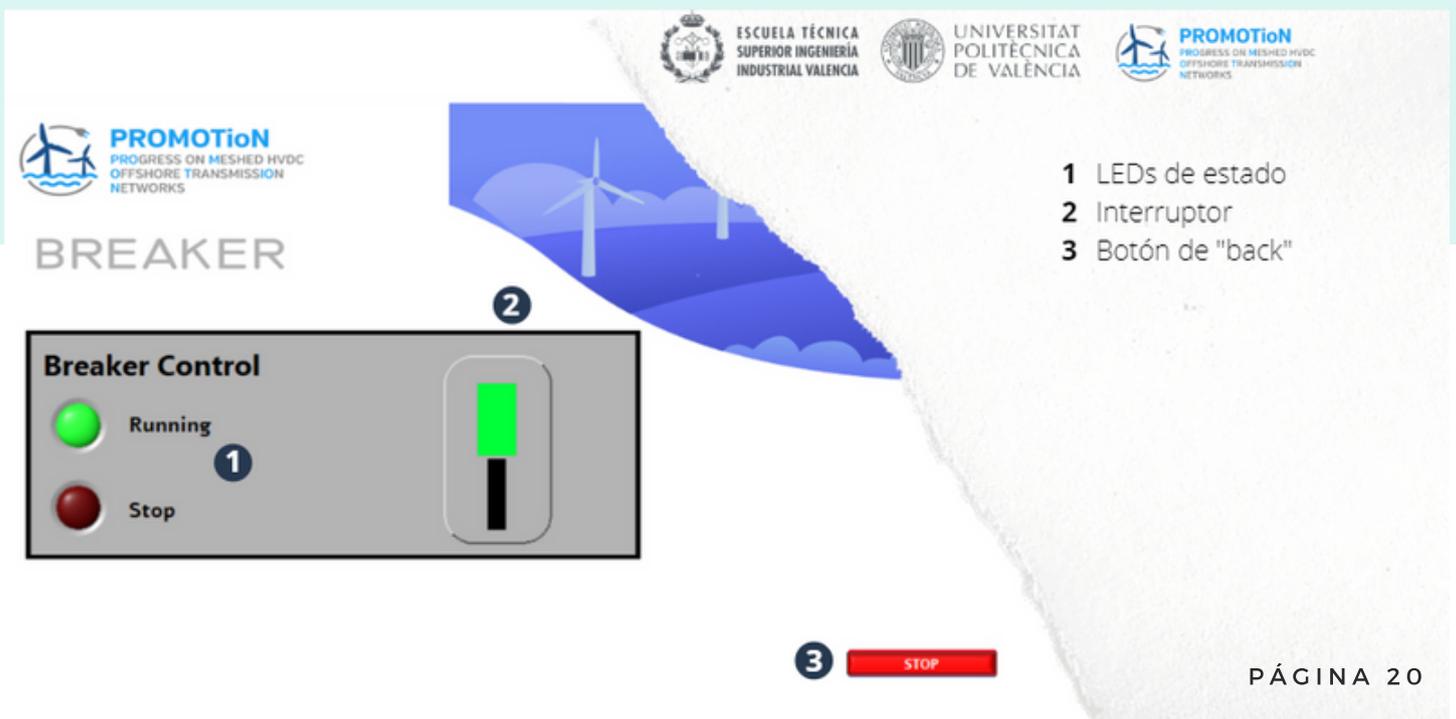
6 Menú
7 Tabla de datos
8 Botón de generación

CONTROL DEL BREAKER

El "breaker" o interruptor automático, es el elemento que permite desconectar el simulador virtual y/o prototipo a escala en cualquier momento de su simulación, evitando posibles incidentes.

La ventana está compuesta de una serie de elementos bastante simples: una gran clavija (que de forma predeterminada se encuentra activa) y dos LED que comunican al usuario de forma más gráfica el estado del "breaker".

Además, de forma análoga al resto de funcionalidades del programa, incorpora un botón de retroceso que devuelve al usuario a la ventana de menú.



CÓDIGOS DE ERROR Y FAQS

La ventana en cuestión permite al usuario del programa consultar los errores que pudieran derivarse del empleo de cualquiera de las otras pestañas del programa: "Real Time", "Data History", "Breaker" y "Simulation".

Se incluye, además, un "tap control" para facilitar la navegación y la visualización de esta "front view".

Cuando exista algún error, este se notificará encendiendo el LED de "status" en rojo e indicando el código y una breve explicación acerca del error en cuestión para que el usuario o, en su defecto, el programador pudiesen solventarlo.

1 Menú
2 Pantallas de error

Real Time Voltage Error

status	code
●	id
source	

Real Time Active Power Error

status	code
●	id
source	

Real Time Reactive Power Error

status	code
●	id
source	

Real Time Current Error

status	code
●	id
source	

BACK

PÁGINA 21

CÓDIGOS DE ERROR Y FAQS

CÓDIGO

DESCRIPCIÓN

#54	<i>La dirección de red está mal formada. Asegúrese de que la dirección tenga un formato válido. Para TCP / IP, la dirección puede ser un nombre de máquina o una dirección IP con el formato xxx.xxx.xxx.xxx. Si este error ocurre al especificar un nombre de máquina, asegúrese de que el nombre de la máquina sea válido. Intente hacer ping al nombre de la máquina. Compruebe que tiene un servidor DNS configurado correctamente. Si está utilizando la función Conexión abierta TCP, asegúrese de que el valor del puerto remoto o el nombre del servicio no sea 0.</i>
#55	<i>La operación de la red está en curso. Si recibe este error mientras usa la función UDP Write.</i>
#56	<i>La operación de la red excedió el límite de tiempo especificado por el usuario o del sistema.</i>
#57	<i>La conexión de red está ocupada.</i>
#58	<i>La función de red no es compatible con el sistema.</i>
#59	<i>Problemas con la escritura UDP en la ventana actual.</i>
#60	<i>El puerto o la dirección de red especificados están actualmente en uso. Seleccione un puerto o una dirección de red disponibles.</i>
#61	<i>El sistema provocó que se cancelara la conexión de red.</i>
#62	<i>El servidor rechazó la conexión de red. Para TCP / IP</i>
#1011	<i>No se encuentran datos para el intervalo temporal seleccionado.</i>
#44	<i>Problema al intentar salvar un fichero</i>

CÓDIGOS DE ERROR Y FAQS

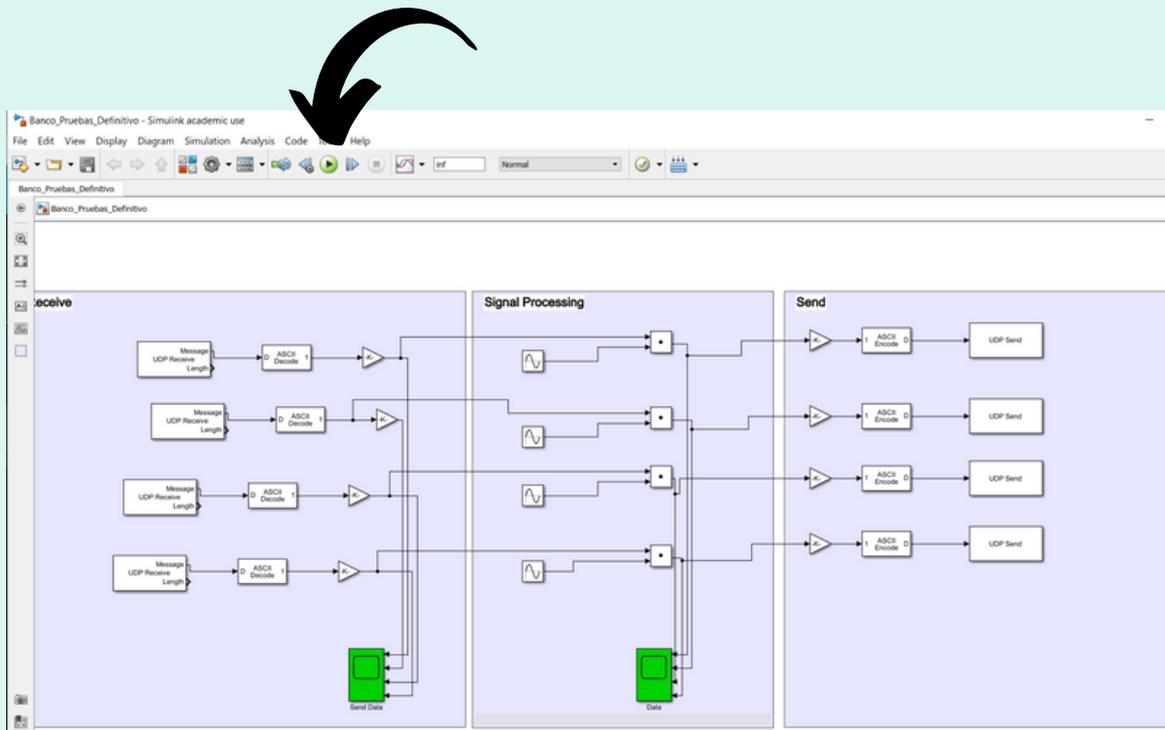
¿Cómo ejecutar el código en LabVIEW?

Para ejecutar un código en LabVIEW el usuario deberá pulsar sobre el botón de "run" dentro de la ventana de "log-in" ya que esta es la ventana principal del programa.

Para saber más sobre LabVIEW consulte: <https://www.ni.com/getting-started/labview-basics/esa/>

¿Cómo ejecutar el código en MATLAB?

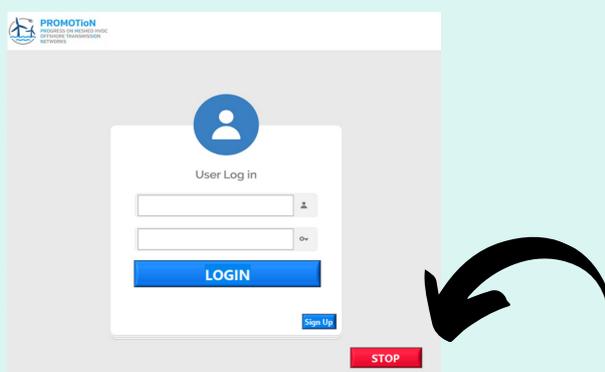
Para ejecutar un código en MATLAB, más concretamente en Simulink. El usuario deberá pulsar sobre el botón de "run" como se puede observar en la siguiente figura.



CÓDIGOS DE ERROR Y FAQS

¿Cómo finalizar el código en LabVIEW?

El programa esta configurado para poder finalizarse sin la necesidad de pulsar el botón de "abort". Para ello, el usuario deberá retroceder hasta la ventana de "log-in" y pulsar sobre el botón de finalización del programa.



¿Cómo pasar el tiempo de UNIX Timestamp a la fecha real?

Para realizar esta conversión se recomienda utilizar convertidores online como: <https://www.unixtimestamp.com/index.php>

¿Qué hacer si tengo problemas durante el registro?

Si el registro genera problemas, lo más probable es que no se disponga de buena conexión red o que el correo electrónico ya haya sido utilizado previamente; para solventarlo se puede utilizar otra dirección de correo electrónico o chequear que la conexión es correcta.

Para más dudas acerca del programa pueden consultar al correo electrónico juyusta@etsii.upv.es

Capítulo V.2

Manual de usuari (Valencià)

JULIO YUSTAS TALAMANTES



MANUAL D'USUARI

CONTINGUT



1
ACOSTA DEL
PROGRAMA

2
ESTRUCTURA DEL
PROGRAMA

3
INTERFÍCIE D'USUARI

4
REQUERIMENTS
TÈCNICS

5
INSTAL·LACIÓ

6
INICIAR APLICACIÓ

7
SIGN UP

CONTENIDO



8
LOGIN

9
MENÚ PRINCIPAL

10
MONITORITZACIÓ

11
SIMULACIÓ DE
SENYALS

12
RECERCA EN LA BASE
DE DADES

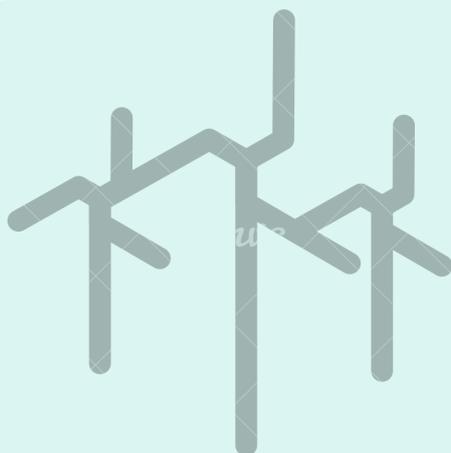
13
CONTROL DEL
"BREAKER"

14
CODIS D'ERROR I
FAQS



ACOSTA DEL PROGRAMA

Aquest programa ha sigut desenvolupat com a part del Treball de Final de Grau de Julio Yustas Talamantes; aquest treball s'inclou dins de l'investigació per part de l'Universitat Politècnica de València per al projecte eòlic europeu "**PROMOTiON**" l'objectiu del qual es explotar tots els recursos de caràcter eòlic presents al mar del nord.



Aquest programa ha sigut desenvolupat per al seu ús com aplicació de monitorització de parcs eòlics a la mar.

La seua comesa principal és proporcionar a l'operari, una interfície clara y senzilla per al correcte acompliment del seu treball.

Al mateix temps, es postula com els fonaments des dels quals construir un programa molt més sofisticat i per al qual es proporciona el "Manual de Programador"



ACOSTA DEL PROGRAMA

Els autors d'aquesta aplicació han sigut Julio Yustas Talamantes baix la tutorització de la Doctora Soledad Inmaculada Bernal Pérez.

Aquest treball s'ha pogut desenvolupar gràcies a la col·laboració de l'Universitat Politècnica de València i de l'Institut Universitari d'Automàtica i Informàtica Industrial



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

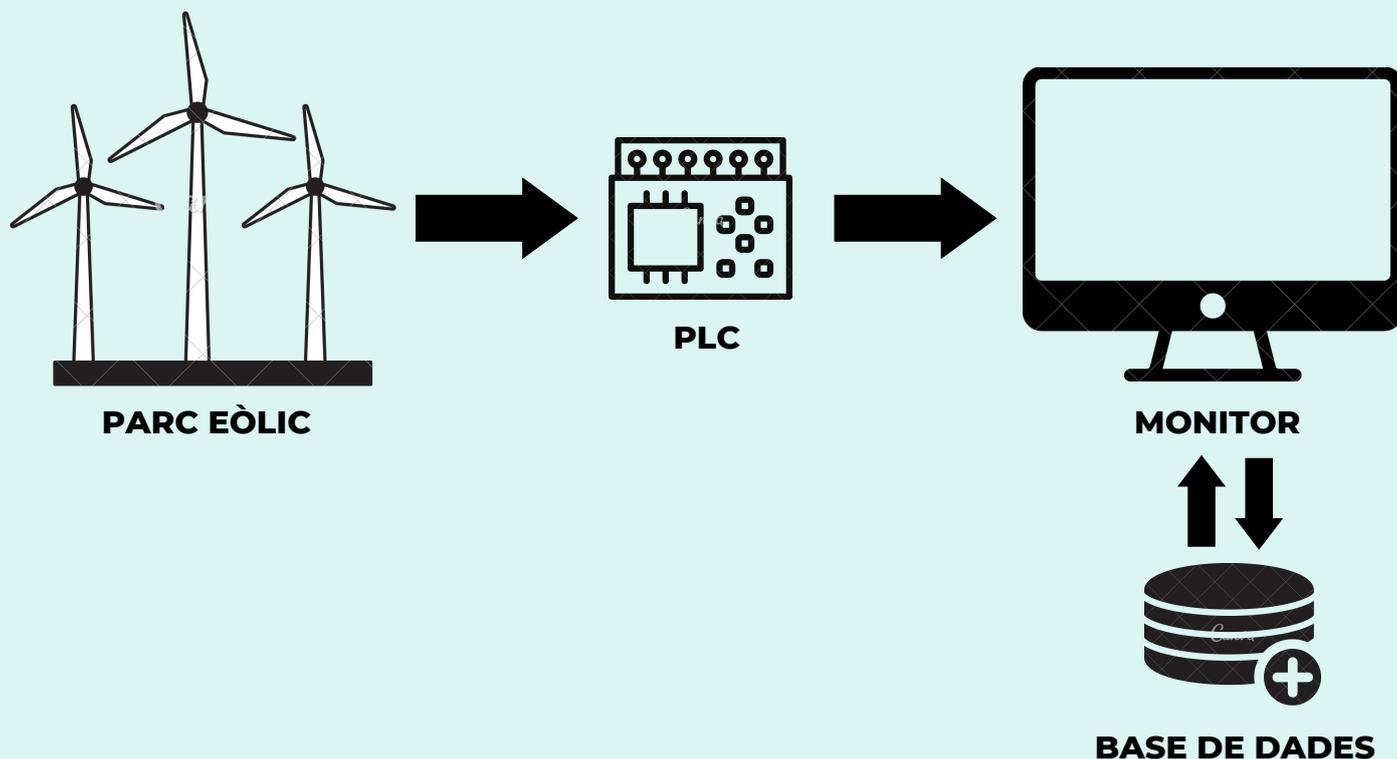


ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA



PROMOTioN
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS





ESTRUCTURA DEL PROGRAMA

Es tracta d'un sistema SCADA (Supervisió, Control i Adquisició de Dades).

L'objectiu principal es controlar un parc eòlic des d'altre lloc de manera remota, per això, necessitarem tres elements clau: el parc eòlic, els PLCs i el sistema de monitorització.

Les connexions entre aquestes parts son una part fonamental per al correcte desenvolupament del present programa.

La unió entre el parc eòlic i els sensors es realitzarà mitjançant un cablejat.

Aquests sensors estaran connectats al seu torn a la xarxa, amb l'objectiu de permetre la comunicació mitjançant protocol UDP amb el simulador o programa SCADA,

És treball del programa SCADA, tractar aquestes dades i acomodarlos a un entorn atractiu per al treballador.

Al seu torn, emmagatzemarà aquestes dades en una base de dades per a permetre la seua possible consulta posterior.

INTERFÍCIE D'USUARI

El programa compta amb una interfície d'usuari senzilla i atractiva realitzada amb l'ajuda de LabVIEW i Photoshop.

La navegació és senzilla: per a avançar entre les diverses pestanyes només hem de triar la modalitat des del menú principal o "main menu".

En el cas de voler retrocedir, es compta en tot moment amb un botó de "back" que ens envia a la pàgina anterior.

Per a finalitzar el programa simplement hem de tornar a la pestanya d'identificació i donar-li clic al botó de "stop" o en el seu defecte, utilitzar el clàssic botó de tancar pestanyes de windows



REQUERIMENTS TÈCNICS

DES D'UN SIMULADOR VIRTUAL:



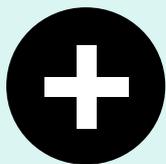
SISTEMA OPERATIU

Windows 7, Ubuntu 16.04 o macOS Sierra en 64 bits.



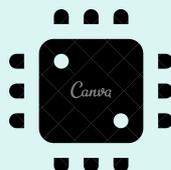
NAVEGADOR Y CONEXIÓ RED

Internet Explorer 11, Google Chrome 58 o Firefox 53



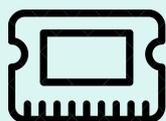
MATLAB

R2017b o superior



PROCESSADOR

Intel Core I5 o superior



MEMÒRIA RAM

Memòria RAM 8G o superior



TARGETA DE SO Y ALTAVEUS

REQUERIMENTS TÈCNICS

DES D' UNA GRANJA EÒLICA REAL:



SISTEMA OPERATIU

Windows 7, Ubuntu 16.04 o macOS Sierra en 64 bits.

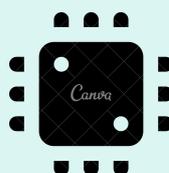


NAVEGADOR

Internet Explorer 11, Google Chrome 58 o Firefox 53

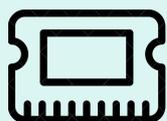


CONEXIÓ RED



PROCESSADOR

Intel Core I5 o superior



MEMÒRIA RAM

Memòria RAM 8G o superior



TARGETA DE SO I ALTAVEUS

INSTAL·LACIÓ

La instal·lació del present programa és prou senzilla, per a això només necessitem seguir una sèrie de passos:

La primera etapa consisteix a descarregar l'arxiu *promotion.exe* present en la web de PROMOTiON o en el repositori adjunt al peu d'aquesta mateixa pàgina.

Una vegada descarregat l'arxiu només queda donar-li doble clic per a executar-lo i instal·lar-ho.

Si es desitja utilitzar-ho al costat del simulador, haurem d'instal·lar en aqueixa mateixa computadora, o en una altra connectada a la mateixa xarxa WIFI, el programa MATLAB i executar els simuladors: *Banco_Pruebas_Breaker.mdl*, *Banco_Pruebas_Definitivo.mdl* i *simulator.mdl*

Si, per contra, es desitja utilitzar un prototip a escala o una granja eòlica real, haurem de realitzar una sincronització dels ports d'enviament dels PLCs i els ports d'entrada del programa d'acord amb la següent taula:

Variable a controlar	Port
Tensió	8811
Corrent	8822
Potència Activa	8833
Potència Reactiva	8844

Aquesta sincronització es realitzarà en el prototip a escala o el parc eòlic, és a dir, sempre sobre el PLC i no sobre el programa.

Control	Enviament
Tensió	7700
Corrent	7711
Potència Activa	7722
Potència Reactiva	7733

La anterior configuració s'utilitzarà per a la funcionalitat de simulador; aquests ports són els d'enviament per part del simulador i/o PLC.

INICIALITZACIÓ

Per a garantir la correcta inicialització del programa, haurem de comprovar que totes dues parts: l'ordinador ("host" del sistema SCADA) i els PLCs disposen de connexió a internet.

Al seu torn, s'haurà de comprovar, de manera anàloga a l'exposat en l'apartat anterior, que existeix una correcta disposició dels ports d'entrada i eixida.

Aquest últim pas és fonamental per a garantir l'èxit de la comunicació mitjançant el protocol UDP.

D'altra banda, l'usuari haurà de registrar-se amb els credencials proporcionats per l'empresa que seran únics i intransferibles.

En el cas de no disposar d'ells, es podrà donar d'alta en la base de dades d'operaris autoritzats mitjançant la pestanya de "sign-up".

Aquesta última pestanya s'explicarà amb més deteniment en els següents punts d'aquest mateix manual.

A continuació es torna a adjuntar la taula de connexions UDP:

Variable a controlar	Port
Tensió	8811
Corrent	8822
Potència Activa	8833
Potència Reactiva	8844

Finalment, per a començar a rebre dades del parc eòlic, l'única cosa que resta és prémer sobre l'interruptor d'inici de connexió, present en la pestanya de "main menu".

Per a la recepció de dades per part del programari s'haurà de modificar l'adreça IP dels protocols d'enviament en les pestanyes de "Simulator" i "Breaker".

SIGN-UP

La pestanya "sign-up" permet que els usuaris puguen registrar-se dins de la base de dades de clients.

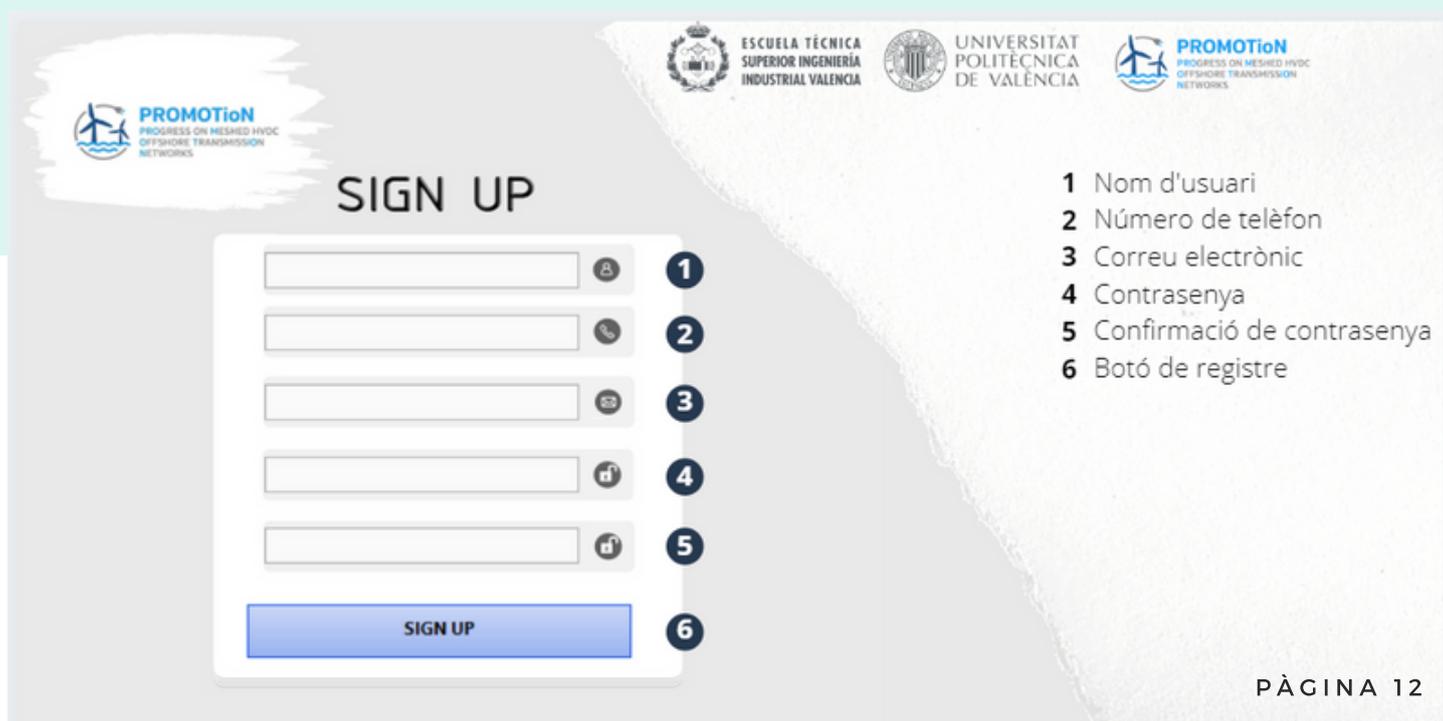
Es pot accedir a aquesta "front view" exclusivament des de la pestanya d'identificació.

Una vegada l'usuari s'haja donat d'alta en la base de dades rebrà un missatge al correu confirmant el seu registre.

En aquest moment, el programa retornarà a l'usuari a la finestra de "login" perquè el client pugui identificar-se i fer ús tot el potencial del programa

Es notificarà error en els següents casos: si les contrasenyes no coincideixen, si el correu ja ha sigut prèviament registrat i si no s'emplenen les caselles imprescindibles.

Es disposa a continuació, d'una imatge explicativa de la "front view", identificant cada component d'aquesta:



The image shows a 'SIGN UP' form with the following components:

- 1 Nom d'usuari
- 2 Número de telèfon
- 3 Correu electrònic
- 4 Contrasenya
- 5 Confirmació de contrasenya
- 6 Botó de registre

The form includes logos for ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA, UNIVERSITAT POLITÈCNICA DE VALÈNCIA, and PROMOTiON (PROGRESS ON MESHED HVDC OFFSHORE TRANSMISSION NETWORKS).

LOGIN

Aquesta finestra permet a l'usuari identificar-se mitjançant un nom d'usuari i una contrasenya. Aquestes credencials poden haver sigut registrades pel mateix operari o vindre proporcionades pel programador mitjançant una escriptura directa sobre la base de dades utilitzada pel programa.

Com s'ha esmentat, s'ofereix la possibilitat que l'usuari registre les seues dades manualment. Per a això, ha de prémer sobre el botó "sign-up" localitzat a la cantonada inferior dreta de la finestra.

En principi, el present codi es troba acabat i depurat, per la qual cosa, de no canviar la base de dades, no seria necessari modificar cap aspecte.

Es disposa a continuació, d'una imatge explicativa de la "front view", identificant cada component d'aquesta:

The screenshot shows a login window titled "User Log in" with a blue user icon. It contains two input fields for username and password, a blue "LOGIN" button, and a smaller blue "Sign Up" button. A red "STOP" button is located at the bottom center. Logos for PROMOTiON, ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA, and UNIVERSITAT POLITÈCNICA DE VALÈNCIA are visible at the top.

- 1 Nom d'usuari
- 2 Contrasenya
- 3 Botó d'identificació
- 4 Botó de registre
- 5 Botó de stop

MENÚ PRINCIPAL

La pestanya "main menu" permet a l'usuari triar entre les diferents funcionalitats del programa: veure dades en temps real, consultar l'històric de dades, entrar al simulador, controlar l'interruptor automàtic, consultar els errors i obrir la pestanya d'informació del programa.

A més, s'inclou una breu informació sobre les dades en temps real, encara que aquesta funcionalitat s'explicarà amb més detall en el punt dedicat a la finestra de dades en temps real.

Per a activar la sincronització, s'hauran de complir tots els paràmetres anteriorment citats i accionar la clavilla.

Es pot modificar el nom de la base de dades encara que no es recomana perquè, de canviar-lo, caldria realitzar una sèrie de modificacions en el codi.

1 Botons de menú
2 Clavilla de connexió
3 Nom en la base de dades
4 Botó de "back"
5 Dades en temps real

MONITORITZACIÓ

La finestra "real time" habilita a l'usuari i/o operari del programa obtenir el control total sobre l'aerogenerador (o conjunt d'aerogeneradors).

S'obtenen en tot moment dades sobre la tensió, intensitat, potència activa i potència reactiva, es representa en tres formats: en forma numèrica, gràficament i en forma de barra.

Aquesta pestanya ofereix un menú interactiu, perquè l'usuari pugui seleccionar quina variable desitja monitorar.

A més, incorpora un botó de reculada i una clavilla d'encesa i apagat de manera anàloga a la finestra de "main menu".

1 Clavilla
2 Menú
3 Botó de "back"
4 Indicadors

SIMULACIÓ DE SENYALS

Una altra de les funcionalitats del present programa és la simulació de senyals; el que permet generar una espècie de simulador o banc de proves.

Per a aquesta funcionalitat, igual que per a les anteriors, es pot treballar contra un simulador virtual o un prototip a escala.

Amb l'objectiu de poder fer ús d'aquesta propietat, s'ha generat el codi *Banco_Pruebas_Definitivo.mdl* en codi de MATLAB.

Aquesta pestanya ofereix un menú interactiu, perquè l'usuari pugui seleccionar si desitja enviar dades o veure el comportament d'aquests en el sistema.

L'apartat de gràfics incorpora una nova funcionalitat, que habilita a l'usuari a poder triar que variable vol plotejar a cada moment. Aquesta funcionalitat ve acompanyada d'una llegenda per a poder identificar el color associat a cada paràmetre.

The screenshot displays the 'SIMULATOR' interface with the 'Control' tab selected. At the top, there are logos for 'PROMOTiON', 'ESCUOLA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA', 'UNIVERSITAT POLITÈCNICA DE VALÈNCIA', and 'PROMOTiON'. The main area shows four sliders for 'Current' (A), 'Voltage' (V), 'Active Power' (MW), and 'Reactive Power' (MVA). A red 'STOP' button is visible at the bottom right. A legend on the right side of the image identifies the numbered elements: 1 Menú, 2 Control de paràmetres, 3 Botó de "back", and 4 Seleccio de paràmetres per representar.

SIMULACIÓ DE SENYALS

Una altra de les funcionalitats del present programa és la simulació de senyals; el que permet generar una espècie de simulador o banc de proves.

Per a aquesta funcionalitat, igual que per a les anteriors, es pot treballar contra un simulador virtual o un prototip a escala.

Amb l'objectiu de poder fer ús d'aquesta propietat, s'ha generat el codi

Banco_Pruebas_Definitivo.mdl en codi de MATLAB.

Aquesta pestanya ofereix un menú interactiu, perquè l'usuari pugui seleccionar si desitja enviar dades o veure el comportament d'aquests en el sistema.

L'apartat de gràfics incorpora una nova funcionalitat, que habilita a l'usuari a poder triar que variable vol plotejar a cada moment. Aquesta funcionalitat ve acompanyada d'una llegenda per a poder identificar el color associat a cada paràmetre.

The screenshot displays the 'SIMULATOR Control Graph' interface. At the top, logos for 'ESCUOLA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA', 'UNIVERSITAT POLITÈCNICA DE VALÈNCIA', and 'PROMOTION' are visible. The interface includes a 'Control' section with checkboxes for 'Voltage', 'Current', 'Active Power', and 'Reactive Power', each with an 'Off/On' option. A 'Graph' section shows a plot area with a y-axis ranging from -1 to 1 and an x-axis labeled 'Time'. A legend below the graph identifies the colors for Voltage (red), Current (green), Active Power (blue), and Reactive Power (orange). A 'STOP' button is located at the bottom right. Numbered callouts (1, 2, 3, 4) point to the menu, control panel, back button, and parameter selection options respectively.

- 1 Menú
- 2 Control de paràmetres
- 3 Botó de "back"
- 4 Selecció de paràmetres per representar

RECERCA EN LA BASE DE DADES

Aquesta pantalla ofereix a l'usuari la possibilitat de consultar les dades rebudes per part del simulador en un interval temporal exacte. Per a oferir aquestes funcionalitats, la pestanya disposa de dos botons que obrin el calendari d'inici i fin de l'interval i dos botons de confirmació amb el seu respectiu LED.

Una vegada seleccionat l'interval, l'usuari haurà de prémer el botó de plotejar, en aquest mateix moment s'obrirà una subpestanya la funció de la qual serà trobar les dades dins d'aquest interval consultant la base de dades.

Quan ja s'hagen trobat, es projectaran en forma de taula i gràficament, oferint a l'usuari la disponibilitat de descarregar aquestes dades en un fi*chero.

A més, incorpora un botó per a retrocedir.

The screenshot shows the 'DATA HISTORY' interface. At the top, there are logos for 'PROMOTioN', 'ESCUOLA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA', 'UNIVERSITAT POLITÈCNICA DE VALÈNCIA', and 'PROMOTioN'. The main area contains a 'Plot' button (4), 'START DATE' and 'END DATE' sections, each with an 'OPEN CALENDAR' button (1) and a 'CONFIRM DATE' button (2). A 'BACK' button (3) is at the bottom. A calendar pop-up window (5) is shown on the right with the text 'Enter the desired date:' and a 'Close window' button.

1 Calendari
 2 Botó de confirmació
 3 Botó de "back"
 4 Botó de representació
 5 Subpestanya de calendari

RECERCA EN LA BASE DE DADES

Aquesta pantalla ofereix a l'usuari la possibilitat de consultar les dades rebudes per part del simulador en un interval temporal exacte. Per a oferir aquestes funcionalitats, la pestanya disposa de dos botons que obrin el calendari d'inici i fin de l'interval i dos botons de confirmació amb el seu respectiu LED.

Una vegada seleccionat l'interval, l'usuari haurà de prémer el botó de plotejar, en aquest mateix moment s'obrirà una subpestanya la funció de la qual serà trobar les dades dins d'aquest interval consultant la base de dades.

Quan ja s'hagen trobat, es projectaran en forma de taula i gràficament, oferint a l'usuari la disponibilitat de descarregar aquestes dades en un fi*chero.

A més, incorpora un botó per a retrocedir.

ESCUOLA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

PROMOTion
PROGRESS ON MESHED HVDC OFFSHORE TRANSMISSION NETWORKS

DATA HISTORY

Data Graph **6**

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Do you want to generate a data file to see all?

Generate **8**

7

3 BACK

6 Menú
7 Taula de dades
8 Botó de generació

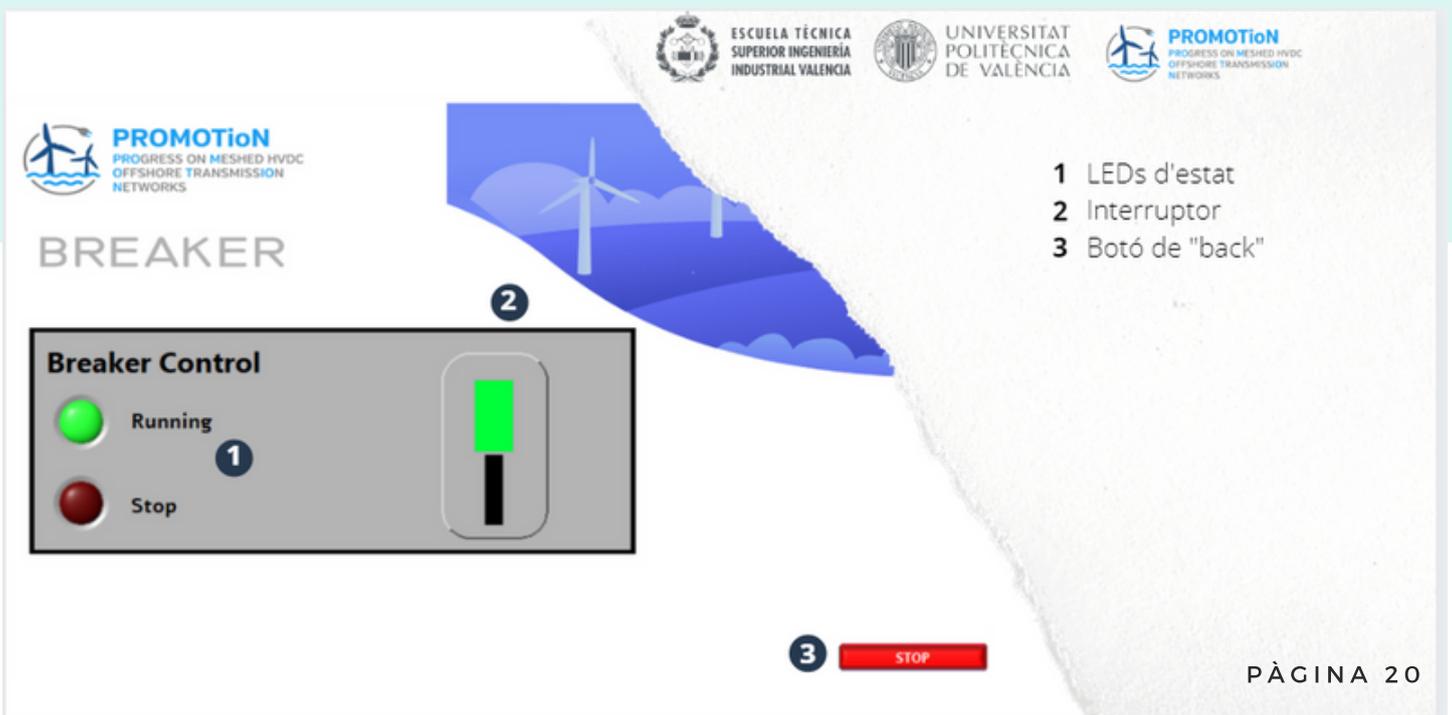
PÀGINA 19

CONTROL DEL BREAKER

El "breaker" o interruptor automàtic, és l'element que permet desconnectar el simulador virtual i/o prototip a escala en qualsevol moment de la seua simulació, evitant possibles incidents.

La finestra està composta d'una sèrie d'elements bastant simples: una gran clavilla (que de manera predeterminada es troba activa) i dos LED que comuniquen a l'usuari de forma més gràfica l'estat del "breaker".

A més, de manera anàloga a la resta de funcionalitats del programa, incorpora un botó de reculada que retorna a l'usuari a la finestra de menú.



CODIS D'ERROR I FAQS

La finestra en qüestió permet a l'usuari del programa consultar els errors que pogueren derivar-se de l'ús de qualsevol de les altres pestanyes del programa: "Real Time", "Data History", "Breaker" i "Simulation".

S'inclou, a més, un "tap control" per a facilitar la navegació i la visualització d'aquesta "front view".

Quan existisca algun error, aquest es notificarà encenent el LED de "status" en roig i indicant el codi i una breu explicació sobre l'error en qüestió perquè l'usuari o, en defecte d'això, el programador pogueren solucionar-ho.

ESCUOLA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

PROMOTion
PROGRESS ON MESHED HVDC OFFSHORE TRANSMISSION NETWORKS

PROMOTion
PROGRESS ON MESHED HVDC OFFSHORE TRANSMISSION NETWORKS

ERROR LOG

1 Menú
2 Finestres d'error

Real Time Data His. / Breaker Simulator Send Simulator Receive

Real Time Voltage Error

status	code
●	
source	

Real Time Active Power Error

status	code
●	
source	

Real Time Reactive Power Error

status	code
●	
source	

Real Time Current Error

status	code
●	
source	

2

BACK

PÀGINA 21

CODIS D'ERROR I FAQS

CÓDIGO

#54

DESCRIPCIÓN

La direcció de xarxa està mal formada. Assegure's que la direcció tinga un format vàlid. Per a TCP / IP, l'adreça pot ser un nom de màquina o una adreça IP amb el format xxx.xxx.xxx.xxx. Si aquest error ocorre en especificar un nom de màquina, assegure's que el nom de la màquina siga vàlid. Intente fer ping al nom de la màquina. Comprove que té un servidor DNS configurat correctament. Si està utilitzant la funció Connexió oberta TCP, assegure's que el valor del port remot o el nom del servei no siga 0.

#55

L'operació de la xarxa està en curs. Si rep aquest error mentre usa la funció UDP Write.

#56

L'operació de la xarxa va excedir el límit de temps especificat per l'usuari o del sistema.

#57

La connexió de xarxa està ocupada.

#58

La funció de xarxa no és compatible amb el sistema.

#59

Problemes amb l'escriptura UDP en la finestra actual.

#60

El port o la direcció de xarxa especificats estan actualment en ús. Seleccione un port o una direcció de xarxa disponibles.

#61

El sistema va provocar que es cancel·lara la connexió de xarxa.

#62

El servidor va rebutjar la connexió de xarxa. Per a TCP / IP

#1011

No es troben dades per a l'interval temporal seleccionat.

#44

Problema al intentar salvar un fitxer.

CODIS D'ERROR I FAQS

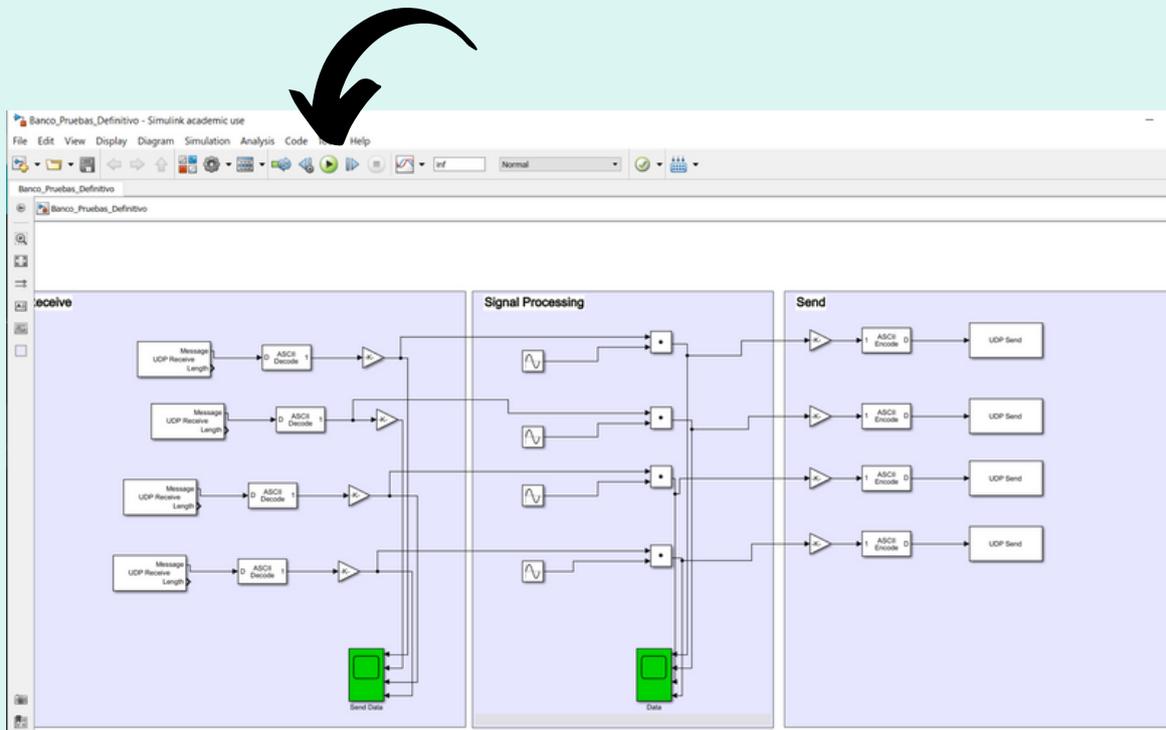
Com executar el codi en LabVIEW?

Per a executar un codi en LabVIEW l'usuari deurà polsar sobre el botó de "run" dins de la finestra de "log-in" ja que és aquesta la finestra principal del programa.

Per saber més sobre LabVIEW pots consultar: <https://www.ni.com/getting-started/labview-basics/esa/>

Com executar el codi en MATLAB?

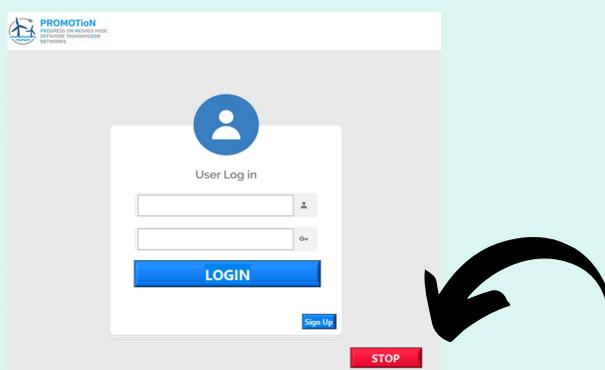
Per a executar un codi en MATLAB, més específicament en Simulink. L'usuari deurà polsar sobre el botó de "run" com es pot comprovar en la següent figura.



CODIS D'ERROR I FAQS

Com finalitzar el codi en LabVIEW?

El programa està configurat per a poder concloure sense la necessitat de pulsar el botó de "abort". Para això, l'usuari deurà retrocedir fins a la finestra de "log-in" i pulsar sobre el botó de finalització del programa.



Com passar el temps de UNIX Timestamp a la data real?

Per a realitzar aquesta conversió es recomana utilitzar el convertidors online, per exemple: <https://www.unixtimestamp.com/index.php>

Què fer si tinc problemes durant el registre?

Si el registre genera problemes, el més probable és que no es disposi de una bona connexió red o que el correu electrònic ja haja sigut utilitzat previament; per a solucionar-ho es pot utilitzar altra direcció de correu electrònic o comprovar que la connexió és correcta.

Para més dubtes acosta del programa poden consultar al correu electrònic juyusta@etsii.upv.es

Capítulo V.3

User Manual (English)

JULIO YUSTAS TALAMANTES



USER MANUAL

CONTENT



1
ABOUT THE SOFTWARE

2
SOFTWARE STRUCTURE

3
USER INTERFACE

4
**TECHNICAL
REQUIREMENTS**

5
INSTALLATION

6
START

7
SIGN UP

CONTENIDO



8
LOGIN

9
MAIN MENU

10
MONITORING

11
SIGNAL SIMULATION

12
DATABASE SEARCH

13
BREAKER CONTROL

14
ERROR CODES & FAQs



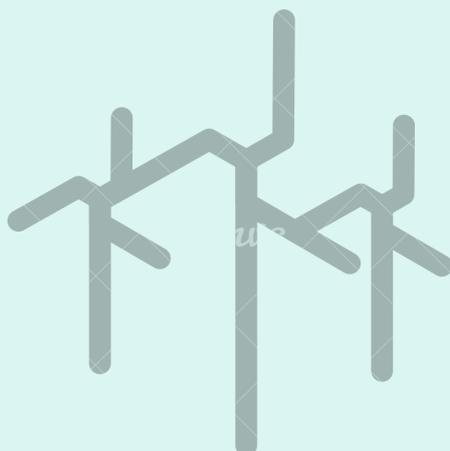
ABOUT THE SOFTWARE

This program has been developed as part of the Final Degree Work of Julio Yustas Talamantes; this work is included as part of the research of the Universitat Politècnica de València for the European wind project "**PROMOTION**" whose main objective is to exploit all the wind resources of the North Sea.

This program has been developed for use as an offshore wind farm monitoring application.

Its main purpose is to provide the operator with a clear and simple interface for the correct performance of his work.

At the same time, it is postulated as the foundation from which to build a much more sophisticated program and for which the "Programmer's Manual" is provided.





ABOUT THE SOFTWARE

The authors of this software have been Julio Yustas Talamantes under the tutorship of Dr. Soledad Inmaculada Bernal Pérez (Industrial Engineering Doctor).

This work has been developed thanks to the collaboration of the Universitat Politècnica de València and the Instituto Universitario de Automática e Informática Industrial (ai2).



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

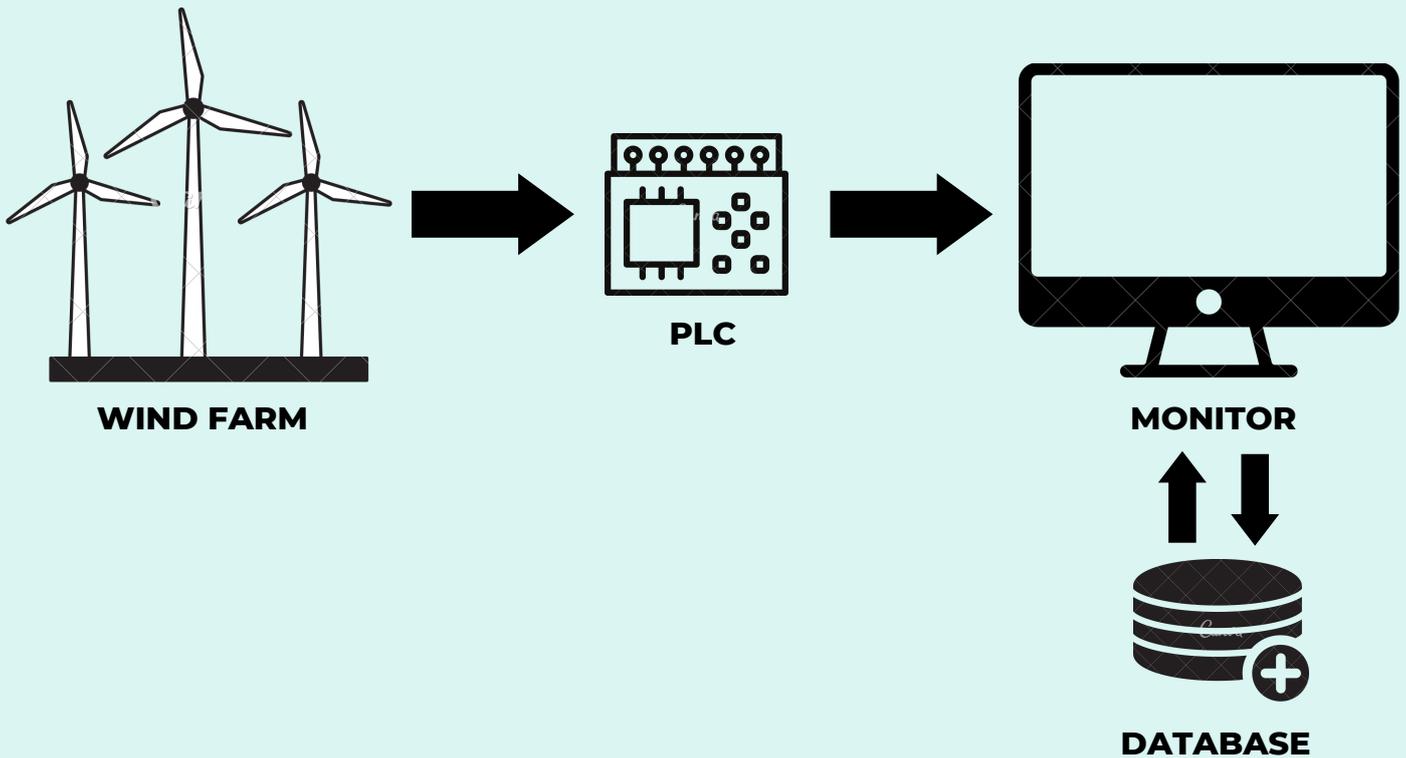


ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA



PROMOTiON
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS





SOFTWARE STRUCTURE

It is a SCADA (Supervisory Control and Data Acquisition) system.

The main objective is to control a wind farm from another location remotely. For this purpose we will need three key elements: the wind farm, the PLCs and the monitoring system.

The connections between these parts are a fundamental for the correct performance of this program.

The connection between the wind farm and the sensors will be made by a cable.

These sensors will, in turn, be connected to the network, in order to allow communication via UDP protocol with the simulator or SCADA program,

Is the job of the SCADA software to process this data and accommodate it in an attractive way to the worker.

In turn, it will store these data in a database to allow for possible later consultation.

USER INTERFACE

The program has a simple and attractive user interface made with the help of LabVIEW and Photoshop.

Navigation is simple: to advance between the various tabs we only have to choose the functionality from the main menu.

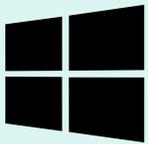
In the case of wanting to go back, there is always a "back" button that sends us to the previous page.

To end the program we simply have to return to the identification tab and click on the "stop" button or use the classic windows button to close tabs.



TECHNICAL REQUIREMENTS

FROM A VIRTUAL SIMULATOR:



OPERATING SYSTEM

Windows 7, Ubuntu 16.04 or macOS Sierra 64 bits.



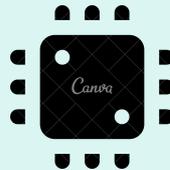
BROWSER

Internet Explorer 11, Google Chrome 58 or Firefox 53



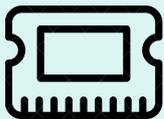
MATLAB

R2017b or superior



PROCESSOR

Intel Core I5 or superior



RAM MEMORY

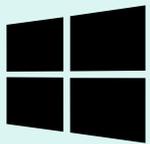
RAM memory 8G or superior



SOUND CARD AND SPEAKERS

TECHNICAL REQUIREMENTS

FROM A REAL WIND FARM:



OPERATING SYSTEM

Windows 7, Ubuntu 16.04 or macOS Sierra 64 bits.



BROWSER

Internet Explorer 11, Google Chrome 58 or Firefox 53

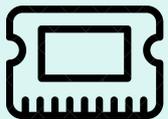


NETWORK CONNECTION



PROCESSOR

Intel Core I5 or superior



RAM MEMORY

RAM memory 8G or superior



SOUND CARD AND SPEAKERS

INSTALLATION

The installation of this program is quite simple, for it, we only need to follow this steps:

The first step is to download the *promotion.exe* file from the PROMOTioN website or from the repository at the bottom of this page.

Once the file has been downloaded, all that remains is to double-click on it to run and install it.

If we want to use it with the simulator, we should install the MATLAB program on the same computer, or on another, connected to the same WIFI network, and run the following simulators:

Banco_Pruebas_Breaker.mdl ,
Banco_Pruebas_Definitivo.mdl
 and *simulator.mdl*

If it is necessary to use a scale prototype or a real wind farm, we must carry out a synchronization of the PLC sending ports and the program entry ports in accordance with the following table:

Control Variable	Port
Voltage	8811
Current	8822
Active Power	8833
Reactive Power	8844

This synchronization will be done on the scale prototype or the wind farm, that is, always on the PLC and not on the program.

Control	Sending
Voltage	7700
Current	7711
Active Power	7722
Reactive Power	7733

The previous configuration will be used for the simulator functionality.

INSTALLATION

To guarantee the correct initialization of the program, we must check that both parts: the computer ("host" of the SCADA system) and the PLCs have correct Internet connection.

At the same time, it will be necessary to check, in a similar way to what has been explained in the previous section, that there is a correct disposition of the input and output ports.

This last step is fundamental to guarantee the communication success of the UDP protocol.

On the other hand, the user will have to register himself with the credentials provided by the company which will be unique and non-transferable.

In the case of not having them, it will be possible to register in the database of authorized operators through the *"sign-up"* tab.

This last page will be explained in more detail in the following points of this manual.

Control Variable	Port
Voltage	8811
Current	8822
Active Power	8833
Reactive Power	8844

Finally, to start receiving data from the wind farm, the only thing left to do is clicking on the connection start switch, in the *"main menu"* tab.

To receive data from the software, the IP address of the sending protocols must be changed in the *"Simulator"* and *"Breaker"* tabs.

SIGN-UP

The "sign-up" tab allows users to register within the customer database.

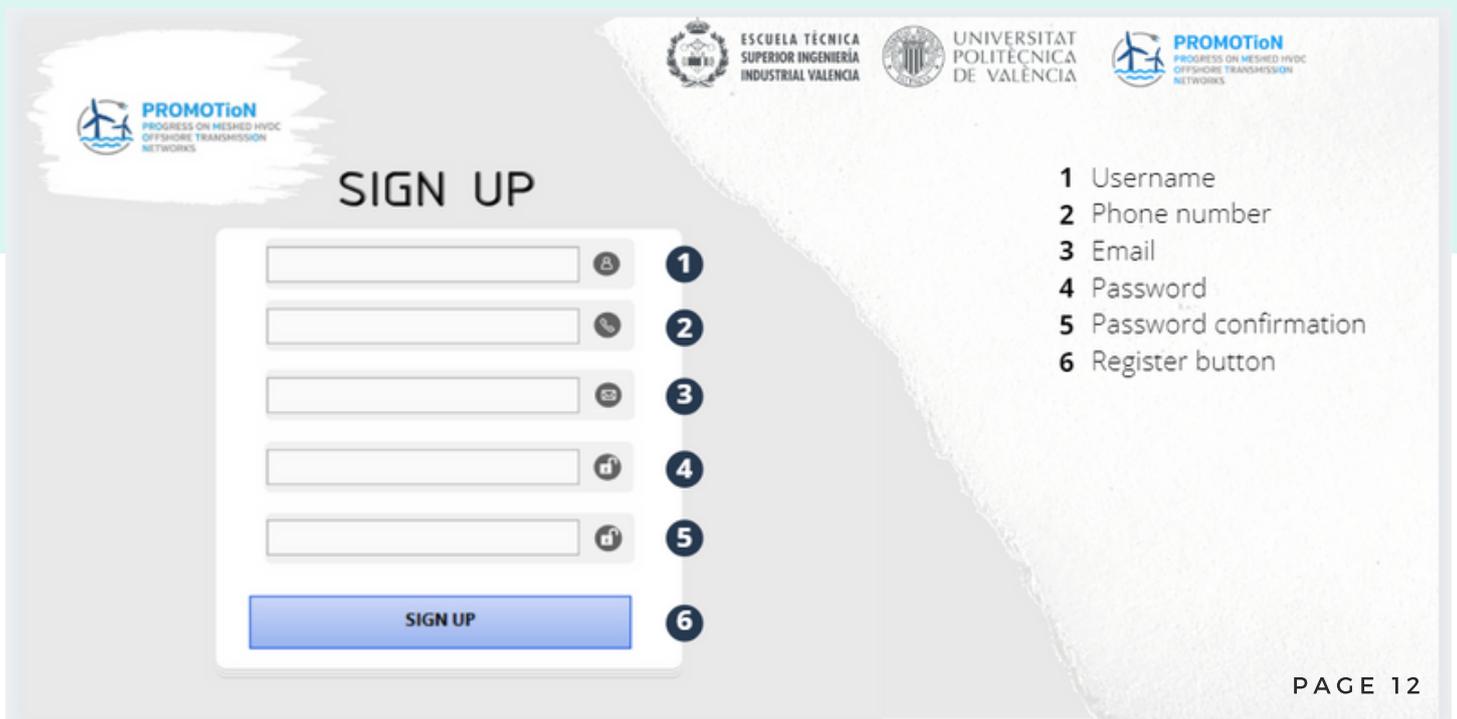
This "front view" can be accessed exclusively from the login tab.

Once the user has been registered in the database, he or she will receive a message to the e-mail provided confirming his or her registration.

At this moment, the program will return the user to the "login" window, there the customer can identify himself and make use of the full potential of the program

An error will be notified in the following cases: if the passwords do not match, if the mail has been previously registered and if the essential boxes are not filled in.

Just below there is an explanatory image of the "front view", identifying each component of it:



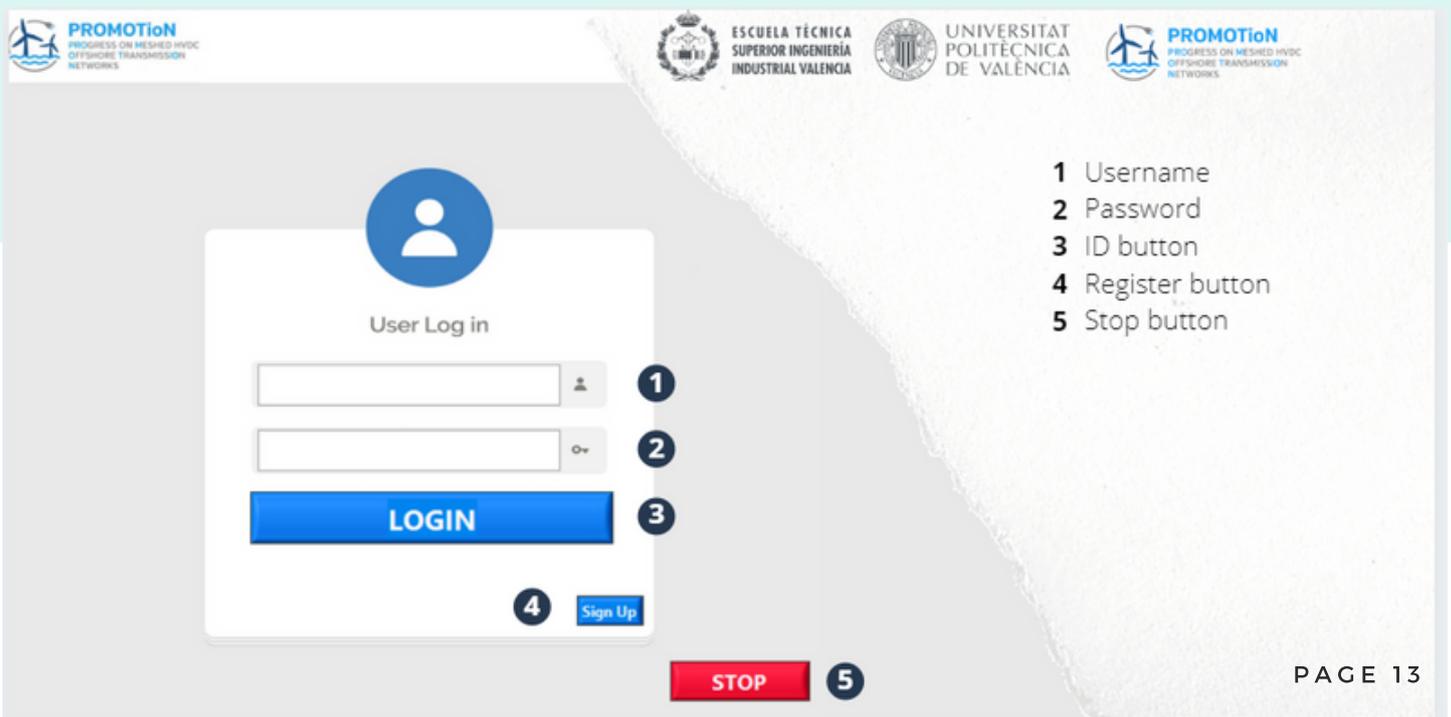
LOGIN

This window allows the user to identify himself by a user name and a password. These credentials can be registered by the operator himself or provided by the programmer by writing directly to the database.

As is mentioned above, the user can register his signup data manually. To do this, click on the "sign-up" button located in the lower right-hand corner of this window.

This code is finished and debugged, so if the database is not changed, it would not be necessary to modify any aspect.

Just below there is an explanatory image of the "front view", identifying each component of it:



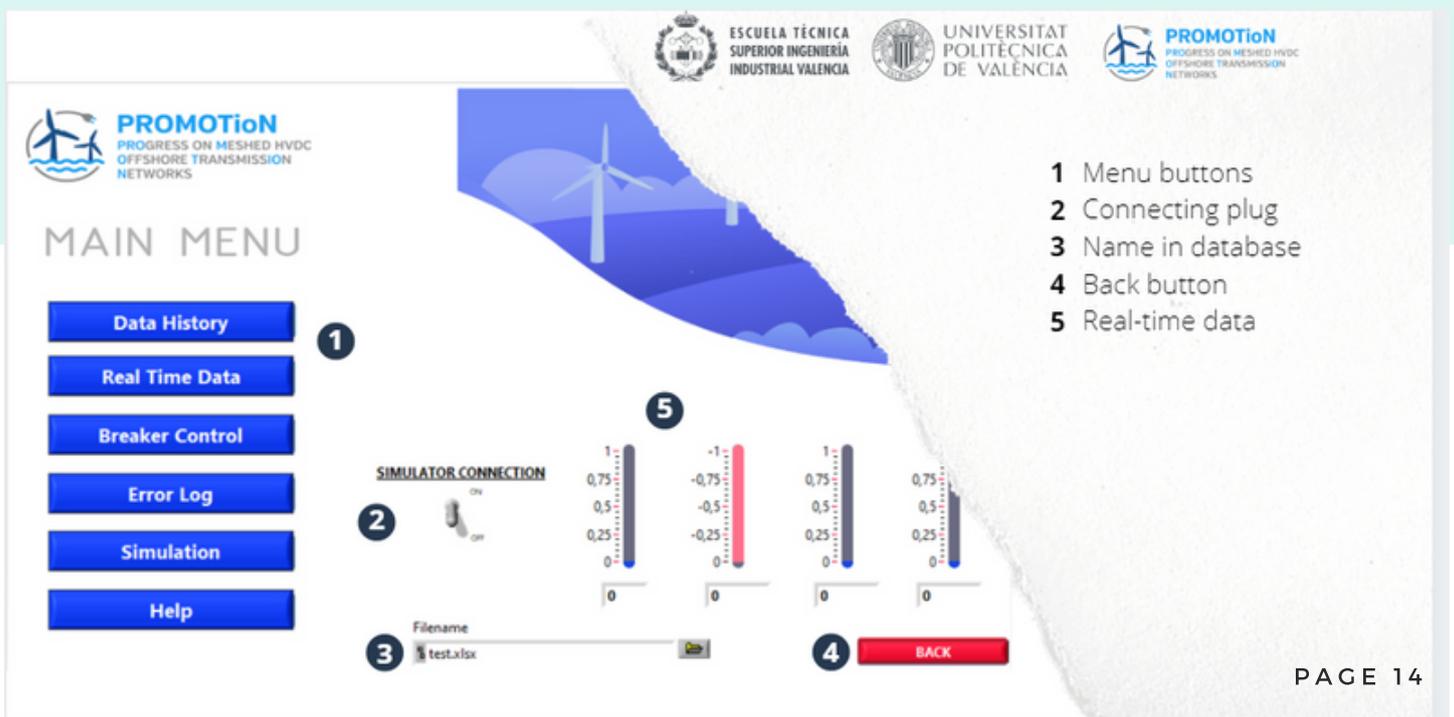
MAIN MENU

The "main menu" tab allows the user to choose between the different program functions: view data in real time, consult the data history, use the simulator, control the circuit breaker, consult errors and open the program information tab.

In addition, a brief information on the real-time data is included, although this functionality will be explained in more detail in the point dedicated to the "real-time" data window.

To activate the synchronization, all the above-mentioned parameters must be met and the plug must be operated.

The name of the database can be modified, although it is not recommended, because, if it is changed, it would be necessary a series of modifications on the code.



- 1 Menu buttons
- 2 Connecting plug
- 3 Name in database
- 4 Back button
- 5 Real-time data

MONITORING

The "real time" window enables the user and/or operator of the software to obtain full control over the wind turbine.

Data on voltage, current, active power and reactive power are obtained at all times, represented in three formats: numerically, graphically and in bar way.

This tab offers an interactive menu, so the user can select which variable he wants to control.

In addition, it incorporates a back button and an on/off pin analogous to the "main menu" window.

ESCUOLA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

PROMOTiON
PROGRESS ON MESHED HVDC OFFSHORE TRANSMISSION NETWORKS

PROMOTiON
PROGRESS ON MESHED HVDC OFFSHORE TRANSMISSION NETWORKS

REAL TIME

Active Power Reactive Power Voltage Current

ACTIVE POWER

SIMULATOR CONNECTION

1

2

3 BACK

4

PAGE 15

SIGNAL SIMULATION

Another feature of this software is the simulation of signals, which allows to generate a kind of simulator or test bench.

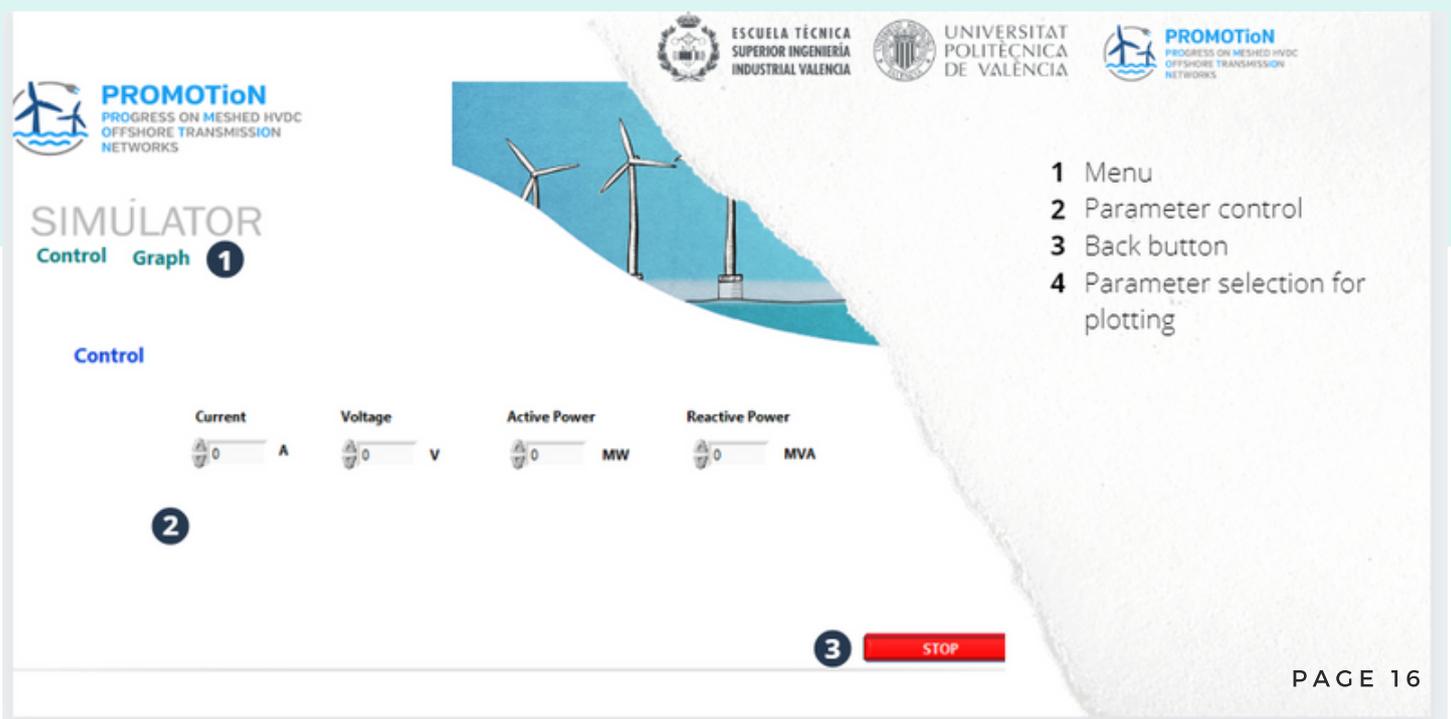
For this functionality, as for the previous ones, it is possible to work against a virtual simulator or a scale prototype.

For using this property, the code *Banco_Pruebas_Definitivo* has been generated on MATLAB

This tab offers an interactive menu, so that the user can choose whether to send data or view the behavior of data in the system.

The graphic section incorporates a new functionality, which enables the user to choose which variable wants to plot at any given time. This functionality is accompanied by a legend to identify the color associated with each parameter.

It also incorporates a back button.



SIGNAL SIMULATION

Another feature of this software is the simulation of signals, which allows to generate a kind of simulator or test bench.

For this functionality, as for the previous ones, it is possible to work against a virtual simulator or a scale prototype.

For using this property, the code *Banco_Pruebas_Definitivo* has been generated on MATLAB

This tab offers an interactive menu, so that the user can choose whether to send data or view the behavior of data in the system.

The graphic section incorporates a new functionality, which enables the user to choose which variable wants to plot at any given time. This functionality is accompanied by a legend to identify the color associated with each parameter.

It also incorporates a back button.

ESCUOLA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

PROMOTioN
PROGRESS ON MESHED HVDC OFFSHORE TRANSMISSION NETWORKS

SIMULATOR
Control Graph

1

Graph

Voltage Off/On

Current Off/On

4 Active Power Off/On

Reactive Power Off/On

Voltage

Current

Active Power

Reactive Power

1
0,75
0,5
0,25
0
-0,25
-0,5
-0,75
-1
-1

Time

3 STOP

1 Menu
2 Parameter control
3 Back button
4 Parameter selection for plotting

PAGE 17

DATABASE SEARCH

This screen offers the user the possibility to consult the data received by the simulator in any exact time interval. To offer these functionalities, the tab has two buttons that open the start and end calendar and two buttons of confirmation with their respective LED.

Once the interval is selected, the user must press the plot button, at that moment a sub-tab will be opened. Its function will be to find the data within this interval by consulting the database.

When they have been found, they will be projected in table form and plotted, offering the user the availability to download these data in a file.

It also incorporates a back button.

The screenshot shows the 'DATA HISTORY' interface. At the top, there are logos for 'PROMOTion', 'ESCUOLA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA', 'UNIVERSITAT POLITÈCNICA DE VALÈNCIA', and 'PROMOTion'. The main area contains a 'Plot' button (4), 'START DATE' and 'END DATE' sections with 'OPEN CALENDAR' buttons (1) and 'CONFIRM DATE' buttons (2). A 'BACK' button (3) is at the bottom. A 'Calendar subtab' (5) is shown as a separate window with a date input field and a 'Close window' button.

1 Calendar
 2 Confirmation button
 3 Back button
 4 Plotting button
 5 Calendar subtab

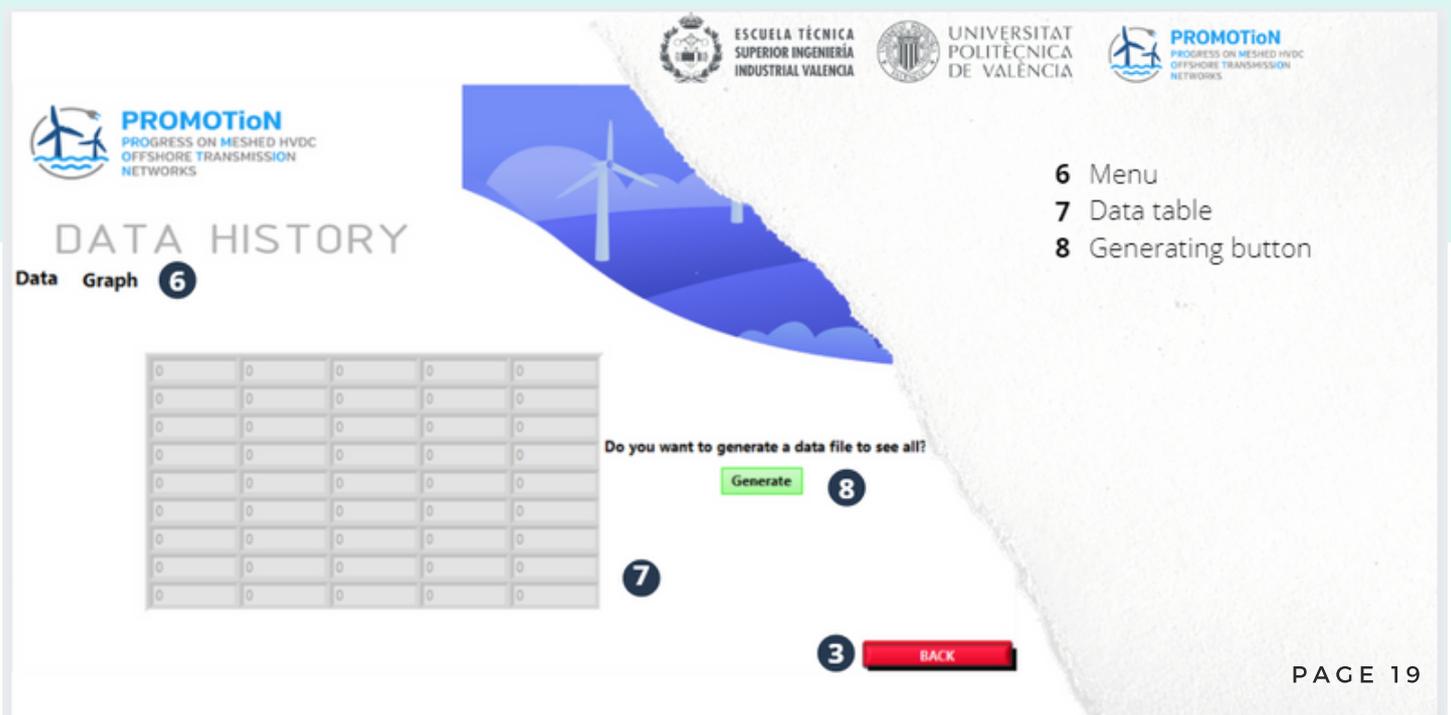
DATABASE SEARCH

This screen offers the user the possibility to consult the data received by the simulator in any exact time interval. To offer these functionalities, the tab has two buttons that open the start and end calendar and two buttons of confirmation with their respective LED.

Once the interval is selected, the user must press the plot button, at that moment a sub-tab will be opened. Its function will be to find the data within this interval by consulting the database.

When they have been found, they will be projected in table form and plotted, offering the user the availability to download these data in a file.

It also incorporates a back button.



BREAKER CONTROL

The "breaker" is the element that allows the virtual simulator and / or scale prototype to be disconnected at any time during the simulation, avoiding potential incidents.

The window is composed of a series of quite simple elements: a large pin (which is active by default) and two LEDs that communicate the status of the "breaker" to the user in a more graphic way.

In addition, similarly to the rest of the program's functionalities, it incorporates a back button that returns the user to the menu window.



ERROR CODES & FAQs

This window enables the user consulting the errors that could arise from the use of any of the other tabs of the program: "Real Time", "Data History", "Breaker" and "Simulation".

A "tap control" is also included to facilitate navigation and display of this "front view".

When there is an error, it will be notified by turning on the "status" LED in red and indicating the code and a brief explanation about the error in question so that the user or the programmer can solve it.

1 Menu
2 Error screens

BACK

PAGE 21

ERROR CODES & FAQs

CODE

DESCRIPTION

#54

The Network address is ill-formed. Make sure the address is in a valid format. For TCP/IP, the address can be either a machine name or an IP address in the form xxx.xxx.xxx.xxx. If this error occurs when specifying a machine name, make sure the machine name is valid. Try to ping the machine name. Check that you have a DNS server properly configured.

#55

The network operation is in progress. If you receive this error while using the UDP Write function, refer to the KnowledgeBase at ni.com for more information.

#56

The network operation exceeded the user-specified or system time limit.

#57

The network connection is busy.

#58

The network function is not supported by the system.

#59

The network is down, unreachable, or has been reset.

#60

The specified port or network address is currently in use. Select an available port or network address.

#61

The system could not allocate the necessary memory.

#62

The system caused the network connection to be aborted.

#1011

No data found for the selected time frame.

#44

Problem trying to save a file.

ERROR CODES & FAQs

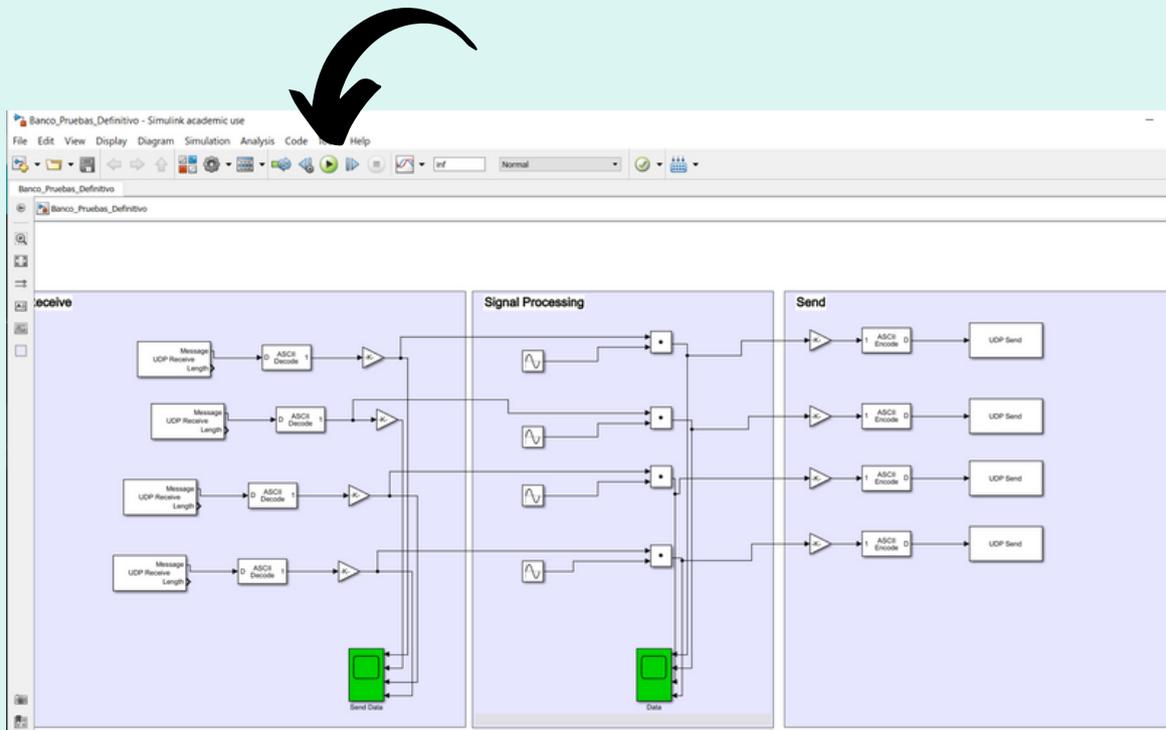
How to run LabVIEW code?

To execute a code in LabVIEW the user must click on the "run" button inside the "log-in" window since this is the main window of the program.

To know more about LabVIEW, please consult: <https://www.ni.com/getting-started/labview-basics/esa/>

How to run MATLAB code?

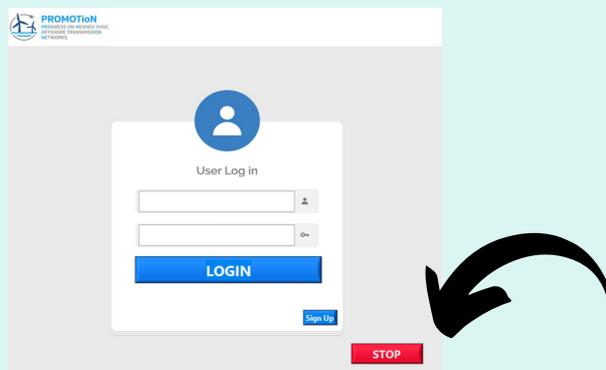
To execute a code in MATLAB, more specifically in Simulink. The user will have to click on the "run" button as shown in the figure below.



ERROR CODES & FAQs

How to finalize LabVIEW code?

The program is configured to be completed without the need to press the "abort" button. To do this, the user must go back to the "log-in" window and click on the program end button.



How to finish the code in LabVIEW?

To make this conversion, is recommended using the online converter, for example: <https://www.unixtimestamp.com/index.php>

What to do if problems occur during the register?

If the registration generates problems, it is most likely that you do not have a good network connection or that the e-mail has been used previously; to solve this you can use another e-mail address or check that the connection is correct.

For more information about the program, please contact us by e-mail: juyusta@etsii.upv.es

Capítulo V.4

Benutzerhanbuch (Deutsch)

JULIO YUSTAS TALAMANTES



BENUTZERHANBUCH

INHALT



1
ÜBER DAS PROGRAMM

2
PROGRAMMSTRUKTUR

3
**BENUTZER-
OBERFLÄCHE**

4
**TECHNISCHE
ANFORDERUNGEN**

5
INSTALLIERUNG

6
**APP
STARTEN/BEGINNEN**

7
SIGN UP

INHALT



8
LOGIN

9
HAUPTMENÜ

10
ÜBERWACHUNG

11
SIGNALSIMULATION

12
DATENBANKABFRAGE

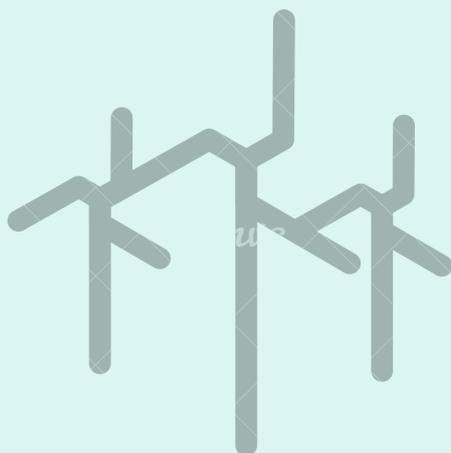
13
"BREAKER"-
KONTROLLE

14
FEHLERCODES UND
HÄUFIG GESTELLTE
FRAGEN



ÜBER DAS PROGRAMM

Dieses Programm wurde als Teil des Abschlussarbeits von Julio Yustas Talamantes entwickelt; die ernannte Arbeit ist an der Investigation der Universitat Politècnica de València für das europäische Windenergieprojekt "**PROMOTioN**" beteiligt, wessen Hauptziel die Nutzung aller Windenergieressourcen in der Nordsee ist.



Dieses Programm wurde für seine Nutzung als Überwachungsapp für Offshore-Windparks entwickelt.

Seine Hauptaufgabe ist, dem Bediener eine klare und einfache Benutzeroberfläche für die korrekte Ausführung seiner Arbeit bieten.

Wiederum, es kandidiert als die Grundlagen, auf denen man ein viel mehr hochentwickeltes Programm erbauen könnte. Dafür bietet man das folgende Programmiererhandbuch.



ÜBER DAS PROGRAMM

Die Autoren dieser App waren Julio Yustas Talamantes, unter die Betreuung der Doktorin Soledad Inmaculada Bernal Pérez (Dr. Wirtschaftsingenieurin).

Die vorliegende Arbeit wurde dank der Zusammenarbeit der **Universitat Politècnica de València** mit dem Instituto Universitario de Automática e Informática Industrial (ai2) entwickelt.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

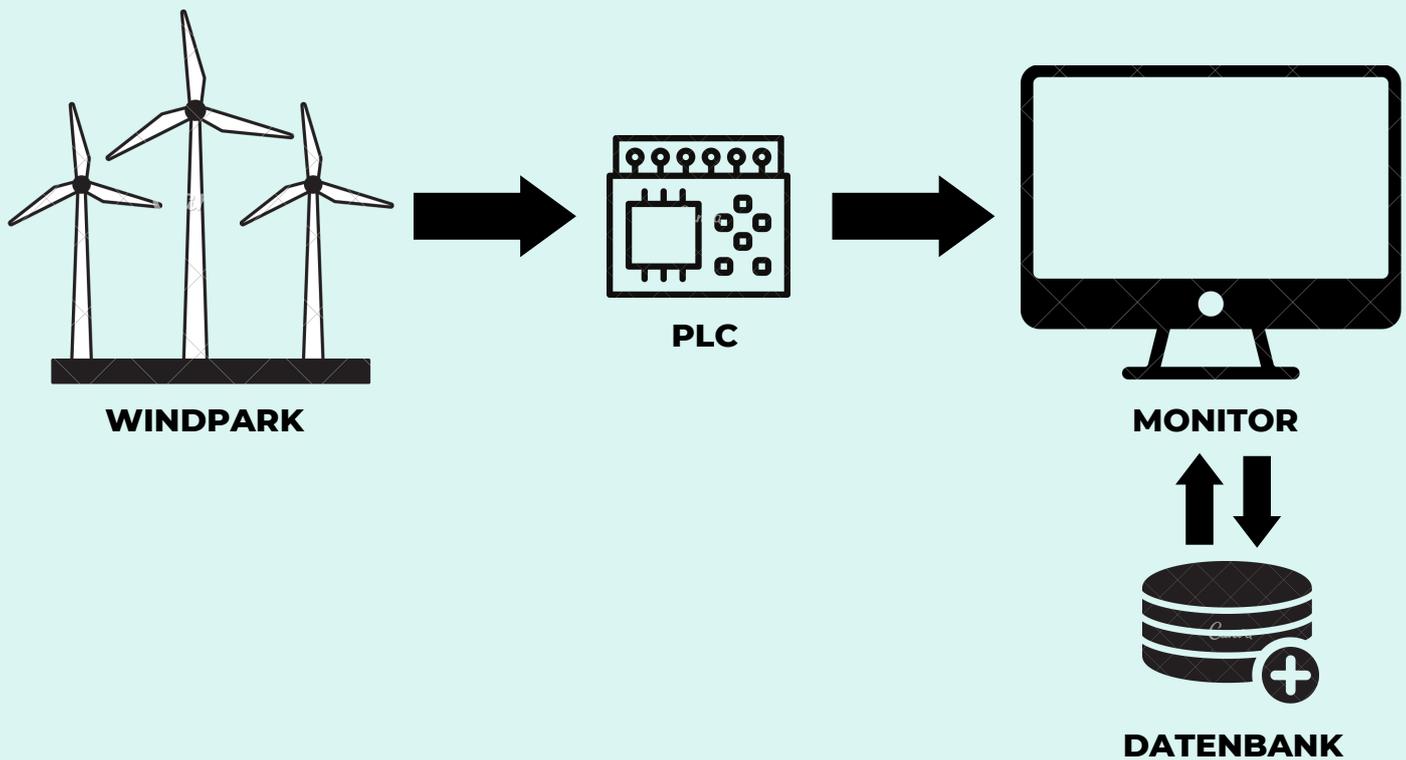


ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA



PROMOTiON
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS





PROGRAMM-STRUKTUR

Es handelt sich um ein SCADA-System (Beaufsichtigung, Kontrolle und Datenerfassung).

Das Hauptziel ist, ein Windpark an einem anderen Standort aus der Ferne kontrollieren. Dafür braucht man 3 Kernelemente: der Windpark, die PLCs und das Monitoringsystem.

Die Verbindungen zwischen den Bestandteilen sind ein wesentlicher Aspekt für die ordnungsgemäße Leistung des vorliegenden Programms.

Die Verbindung zwischen den Windpark und die Sensoren wird durch eine Verkabelung stattfinden.

Diese Sensoren werden wiederum am Netz angeschlossen, mit der Absicht, die Kommunikation durch UDP Protokoll mit dem Simulator oder dem SCADA-Programm zu erlauben.

Es ist Aufgabe des SCADA-Programms, diese Daten zu behandeln und an ein attraktives Umfeld für den Arbeiter anpassen.

Wiederum, wird es diese Daten in einer Datenbank speichern, um eine spätere Abfrage zu erlauben.

BENUTZEROBERFLÄCHE

Das Programm besitzt eine einfache und ansprechende Benutzeroberfläche, mit der Hilfe von LabVIEW und Photoshop realisiert/vollbracht.

Die Navigation ist leicht: um sich zwischen den verschiedenen Tabs zu bewegen, muss man nur den Modus vom Hauptmenü oder "main menu" auswählen.

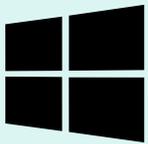
Im Fall, dass man zurückkehren möchte, hat man jederzeit eine "back" Taste zur Verfügung, die zur vorgangenen Seite führt.

Um das Programm zu beenden, muss man einfach zur Identifizierungstab zurückgehen, und auf die "stop" Taste klicken, falls nicht möglich, den klassischen Tabsschliesserbutton von Windows verwenden.



TECHNISCHE ANFORDERUNGEN

VON EINEM VIRTUELLEN SIMULATOR:



BETRIEBSSYSTEM

Windows 7, Ubuntu 16.04 oder macOS Sierra von 64 bits.



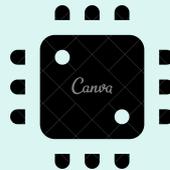
BROWSER UND NETZANSCHLUSS

Internet Explorer 11, Google Chrome 58 oder Firefox 53



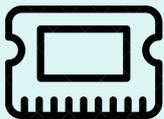
MATLAB

R2017b oder höher



PROZESSOR

Intel Core I5 oder höher



RAM-SPEICHER

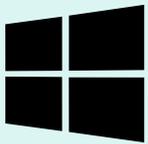
RAM-Speicher 8G oder höher



SOUNDKARTE UND LAUTSPRECHER

TECHNISCHE ANFORDERUNGEN

VON EINER REELLEN WINDFARM:



BETRIEBSSYSTEM

Windows 7, Ubuntu 16.04 oder macOS Sierra von 64 bits.



BROWSER

Internet Explorer 11, Google Chrome 58 oder Firefox 53

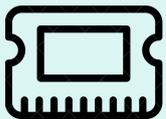


NETZANSCHLUSS



PROZESSOR

Intel Core I5 oder höher



RAM-SPEICHER

RAM-Speicher 8G oder höher



SOUNDKARTE UND LAUTSPRECHER

INSTALLIERUNG

Die Installierung des vorliegenden Programms ist ziemlich einfach, dafür braucht man nur eine Reihe von Schritten:

Die erste Phase besteht darin, die Datei *promotion.exe*, auf die PROMOTioN Webseite oder im beiliegendem Repository in der Fussnote dieser Seite vorhanden, herunterzuladen.

Einmal ist das Archiv heruntergeladen, muss man nur doppelt klicken, um es durchzuführen und installieren.

Wenn man das Programm zusammen mit dem Simulator verwenden möchte, muss man im selben Computer -oder in einem anderen, zum selben WLAN Netz angeschlossen- das MATLAB Programm installieren und die Simulatoren durchführen:

Banco_Pruebas_Breaker.mdl,
Banco_Pruebas_Definitivo.mdl
 und *simulator.mdl*

(*Datenbank_Breaker.mdl*,
Datenbank_Endgültig.mdl)

Wenn man, dagegen, einen Prototyp im Masstab oder eine echte Windfarm verwenden möchte, soll man eine Synchronisierung der Sendeanschlussnummer der PLCs und der Eingänge des Programms durchführen; mit der folgenden Tabelle übereinstimmend

Variable zu kontrollieren	Anschluss
Spannung	8811
Strom	8822
Wirkleistung	8833
Blindleistung	8844

Die ernannte Synchronisierung wird im Prototyp im Masstabsab oder im Windpark durchgeführt, nämlich immer über das PLC und nicht über das Programm.

Kontrolle	Versand
Spannung	7700
Strom	7711
Wirkleistung	7722
Blindleistung	7733

Die vorherige Einstellung wird für die Simulatorfunktionalität verwendet; die Eingänge/Anschlüsse sind denen, die seitens des Simulators und/oder PLC senden.

ERST- AKTIVIERUNG

Um die ordnungsgemäße Initialisierung des Programms zu gewährleisten, muss man überprüfen, dass beide Seiten - der Computer ("host" für das SCADA-System) und die PLCs - über Internetverbindung verfügen.

Wiederum, sollte man verifizieren, in ähnlicher Weise als im vorherigen Absatz dargelegt, dass eine korrekte Anordnung der Eingang- und Ausgangsanschlüsse besteht.

Dieser letzte Schritt ist von grundlegender Bedeutung, um den Erfolg der Kommunikation via UDP Protokoll sicherzustellen.

Andererseits, muss der Benutzer sich mit den einmaligen und nicht übertragbaren Login-Daten, die die Firma zur Verfügung gestellt hat, anmelden.

Im Fall, über diese nicht zu verfügen, kann man sich in der Datenbank autorisierter Bediener registrieren, mittels der "sign-up" Registerkarte.

Dieser letzte Tab wird ausführlicher in den nächsten Punkten dieser Bedienungsanleitung erklärt.

Nun wird wieder die Tabelle der UDP Anschlüsse beigefügt:

Variable zu kontrollieren	Eingang
Spannung	8811
Strom	8822
Wirkleistung	8833
Blindleistung	8844

Schliesslich, um die ersten Daten vom Windpark empfangen zu können, bleibt nur übrig, auf den Schalter zu drücken, der die Verbindung startet -in der "main menu" Registerkarte vorhanden-.

Für den Datempfang seitens des Softwares soll man die IP-Adresse der Versandsprotokolle auf den "Simulator" und "Breaker" Tabs zu verändern.

SIGN-UP

Die "sign-up" Registerkarte ermöglicht, dass die Benutzer sich im Kundendatenbank registrieren können.

Man kann auf diese "front view" ausschliesslich vom Identifizierungstab zugreifen.

Wenn erst einmal der Nutzer sich im Datenbank eingeschrieben hat, wird er/sie eine E-Mail in der gegebenen Adresse erhalten, die die Eintragung bestätigt.

In diesem Moment wird das Programm den Benutzer auf die "login" Registerkarte zurücksenden, damit der Kunde sich anmelden und vom vollen Potenzial des Programms Gebrauch machen kann.

Ein Fehler wird in folgenden Fällen gemeldet: wenn beide Kennworte nicht übereinstimmen, wenn die E-Mail Adresse zuvor registriert wurde und wenn die nötigen Kästchen nicht eingefüllt werden.

Darunter verfügt man über eine erläuternde Abbildung der "front view", auf der man jede ihrer Komponenten identifiziert:

The image shows a registration form titled "SIGN UP" with the following fields and a button:

- 1 Benutzername
- 2 Telefonnummer
- 3 E-Mail Adresse
- 4 Kennwort
- 5 Kennwortbestätigung
- 6 Registrieren (Registertaste)

The form also includes logos for ESCUELA TÉCNICA SUPERIOR INGENIERIA INDUSTRIAL VALENCIA, UNIVERSITAT POLITÈCNICA DE VALÈNCIA, and PROMOTiON.

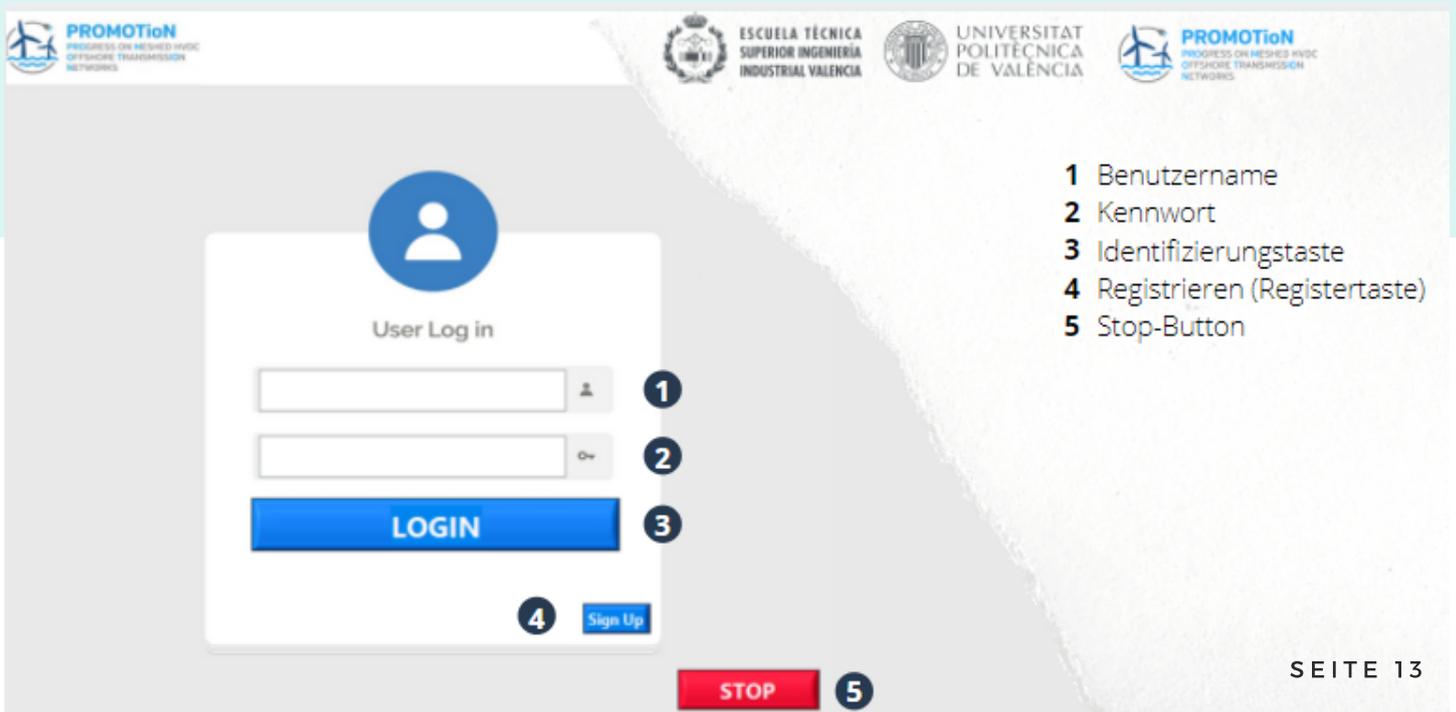
LOGIN

Dieses Fenster erlaubt den Nutzer, sich mit einem Benutzernamen und Kennwort zu identifizieren. Diese Login-Daten können vom selben Bediener registriert, oder vom Programmierer mittels einer direkten Schrift über die Datenbank, die der Software verwendet, zur Verfügung gestellt werden.

Wie vorher erwähnt, bietet man die Möglichkeit, dass der Benutzer die Daten manuell registrieren kann. Dafür muss er auf die "sign-up" Taste drücken, die sich in der unteren rechter Ecke des Fensters befindet.

Grundsätzlich ist der vorhandene Code beendet und gereinigt. Deswegen, wenn die Datenbank nicht geändert wird, ist es nicht nötig, etwas zu modifizieren.

Darunter verfügt man über eine erläuternde Abbildung der "front view", auf der man jede ihrer Komponenten erkennen kann:



- 1 Benutzername
- 2 Kennwort
- 3 Identifizierungstaste
- 4 Registrieren (Registertaste)
- 5 Stop-Button

HAUPTMENÜ

Die Registerkarte "main menu" erlaubt den Nutzer, zwischen den verschiedenen Funktionen des Programms zu wählen: Live-Daten ansehen, die Datenhistorie konsultieren, den Simulator betreten, den automatischen Schalter kontrollieren, Fehler konsultieren und das Informationsfenster über das Programm öffnen.

Ausserdem enthält es eine knappe Information über Live-Daten, obwohl diese Funktionalität weiter in dem spezifischen Punkt erklärt wird.

Um die Synchronisierung zu aktivieren, muss man alle vorher erwähnte Vorgaben erfüllen und den Stecker betätigen.

Der Name in der Datenbank kann modifiziert werden, aber das ist nicht empfohlen, denn, wenn geändert, sollte man eine Reihe Veränderungen im Code durchführen.

1 Menü-Tasten
2 Anschlusstaste
3 Name im Datenbank
4 "Back"-Button
5 Live-Daten

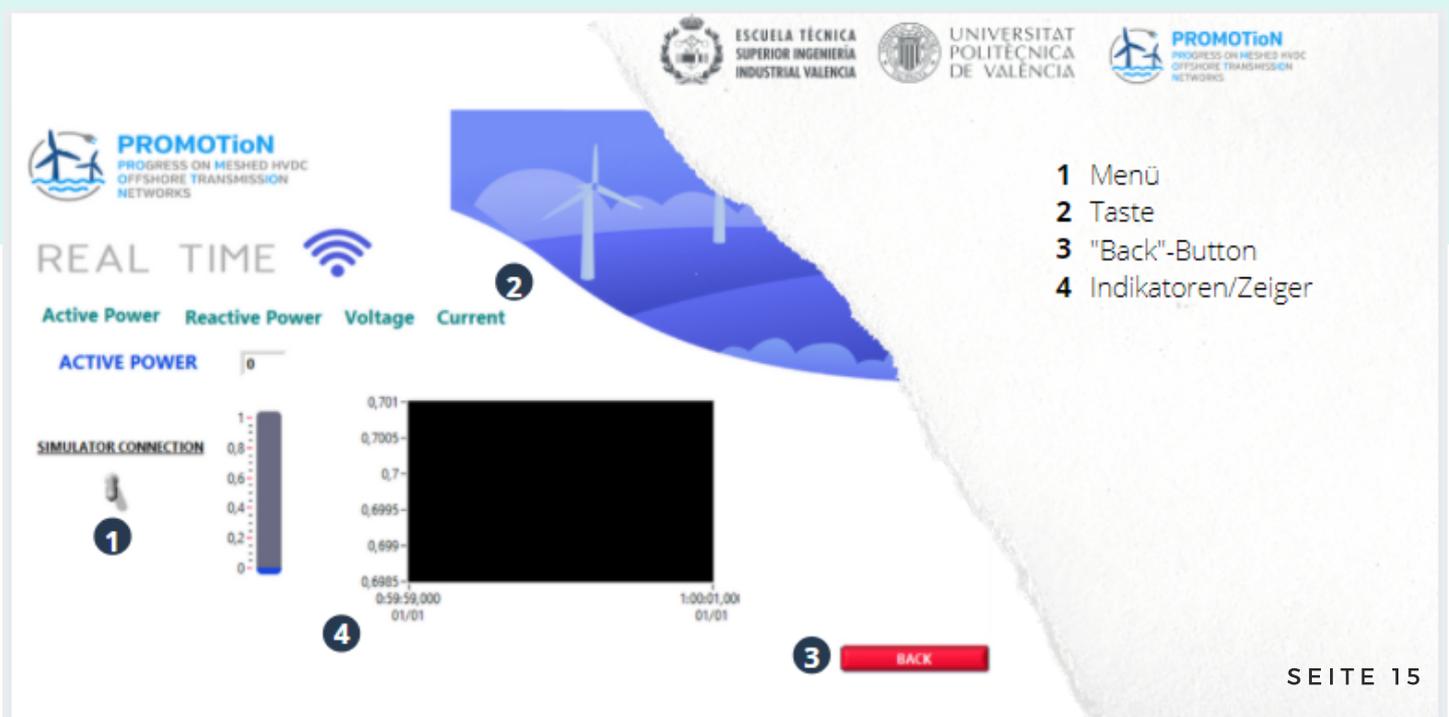
ÜBERWACHUNG

Das Fenster "real time" ermöglicht den Benutzer und/oder Bediener des Softwares, die vollständige Kontrolle über den Windgenerator (oder Gruppe von Windgeneratoren) zu erhalten.

Im jeden Moment erhält man Daten über Spannung, Intensität, Wirkleistung und Blindleistung; in 3 Formaten dargestellt: in numerischer Form, graphisch und als Balkenanzeigen.

Dieser Tab bietet ein interaktives Menü, damit der Nutzer wählen kann, welche Variable er/sie überwachen möchte.

Darüber hinaus integriert es einen Zurück-Button und einen Ein- und Ausschaltungsstecker, in ähnliche Art als das "main menu" Fenster.



- 1 Menü
- 2 Taste
- 3 "Back"-Button
- 4 Indikatoren/Zeiger

SIGNALSIMULATION

Andere der Funktionen des vorhandenen Softwares ist die Signalsimulation, was die Erzeugung einer Art Simulator oder Prüfstand ermöglicht.

Für diese Funktionalität als für die anderen, kann man entweder mit einem virtuellen Simulator oder einem Prototypen im Masstab arbeiten.

Mit dem Ziel, Gebrauch von dieser Eigenschaft zu machen, hat man den Code

Banco_Pruebas_Definitivo.mdl in MATLAB erzeugt.

Diese Registerkarte bietet ein interaktives Menü, damit der Nutzer wählen kann, ob er/sie Daten senden, oder deren Verhalten im System betrachten möchte.

Der Graphikenabsatz fügt eine neue Funktionalität hinzu, was den Nutzer erlaubt, welche Variable er/sie in jedem Moment graphisch darstellen möchte. Diese Funktion wird mit einer Legende begleitet, mit der man die Farbe zugeordnet zu jedem Parameter identifizieren kann

Zusätzlich enthält es einen Zurück-Button.

The screenshot displays the 'SIMULATOR' interface with a 'Control' tab selected. At the top, logos for 'PROMOTion', 'ESCUOLA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA', 'UNIVERSITAT POLITÈCNICA DE VALÈNCIA', and 'PROMOTion' are visible. The main area shows four control sliders: 'Current' (A), 'Voltage' (V), 'Active Power' (MW), and 'Reactive Power' (MVA). A legend on the right lists four items: 1 Menü, 2 Parameterkontrolle, 3 "Back"-Button, and 4 Parameterauswahl für grafische Darstellung. A red 'STOP' button is located at the bottom right of the interface.

SIGNALSIMULATION

Andere der Funktionen des vorhandenen Softwares ist die Signalsimulation, was die Erzeugung einer Art Simulator oder Prüfstand ermöglicht.

Für diese Funktionalität als für die anderen, kann man entweder mit einem virtuellen Simulator oder einem Prototypen im Masstab arbeiten.

Mit dem Ziel, Gebrauch von dieser Eigenschaft zu machen, hat man den Code

Banco_Pruebas_Definitivo.mdl in MATLAB erzeugt.

Diese Registerkarte bietet ein interaktives Menü, damit der Nutzer wählen kann, ob er/sie Daten senden, oder deren Verhalten im System betrachten möchte.

Der Graphikenabsatz fügt eine neue Funktionalität hinzu, was den Nutzer erlaubt, welche Variable er/sie in jedem Moment graphisch darstellen möchte. Diese Funktion wird mit einer Legende begleitet, mit der man die Farbe zugeordnet zu jedem Parameter identifizieren kann

Zusätzlich enthält es einen Zurück-Button.

The screenshot displays the PROMOTion SIMULATOR interface. At the top, logos for ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA, UNIVERSITAT POLITÈCNICA DE VALÈNCIA, and PROMOTion are visible. The main window is titled 'SIMULATOR Control Graph'. On the left, there is a 'Graph' control panel with checkboxes for 'Voltage', 'Current', 'Active Power', and 'Reactive Power', each with an 'Off/On' option. Below these are color-coded waveforms for each parameter. A large graph area on the right shows a plot of these parameters over 'Time'. A red 'STOP' button is located at the bottom right of the graph area. A list of numbered annotations (1-4) points to specific UI elements: 1 points to the 'Control Graph' title, 2 points to the parameter checkboxes, 3 points to the 'STOP' button, and 4 points to the parameter legend.

- 1 Menü
- 2 Parameterkontrolle
- 3 "Back"-Button
- 4 Parameterauswahl für grafische Darstellung

DATENBANKANFRAGE

Diese Registerkarte bietet den Nutzer die Möglichkeit, empfangene Daten vom Simulator in einem präzisen Zeitintervall zu konsultieren. Um diese Funktionen zu leisten, verfügt der Tab über zwei Tasten, die zwei Kalender für die Ermittlung der Start- und Endtermine und zwei Bestätigungsbuttons, mit seinen entsprechenden LED.

Einmal ist das Zeitintervall gewählt, muss den Benutzer auf den Darstellungsbutton drücken, in diesem Moment wird sich eine andere Registerkarte öffnen, deren Funktion ist, die Daten in der Zeitspanne durch eine Datenbankabfrage zu finden,

Wenn sie gefunden werden, werden sie in Tabellenform und grafisch projiziert. Sie bieten den Nutzer die Gelegenheit, diese Daten in einer Datei herunterzuladen.

Zudem gibt es einen Zurück-Button.

The screenshot shows the 'DATA HISTORY' interface. At the top left is the 'PROMOTion' logo with the text 'PROGRESS ON MESHED HVDC OFFSHORE TRANSMISSION NETWORKS'. Below the logo is a 'Plot' button (4). The main area contains two date selection sections. Each section has a 'START DATE' or 'END DATE' label, an 'OPEN CALENDAR' button (1), a date input field (00:00:00,000 DD/MM/YYYY), and a 'CONFIRM DATE' button (2). A 'BACK' button (3) is located at the bottom center. On the right side, there is a 'Close window' dialog box (5) with the text 'Enter the desired date:' and a date input field (00:00:00,000 DD/MM/YYYY). The background features a stylized graphic of wind turbines and a map of the Mediterranean coast.

1 Kalender
 2 Bestätigungsbutton
 3 "Back"-Button
 4 Darstellungsbutton
 5 Zusätzlicher Kalendertab

DATENBANKANFRAGE

Diese Registerkarte bietet den Nutzer die Möglichkeit, empfangene Daten vom Simulator in einem präzisen Zeitintervall zu konsultieren. Um diese Funktionen zu leisten, verfügt der Tab über zwei Tasten, die zwei Kalender für die Ermittlung der Start- und Endtermine und zwei Bestätigungsbuttons, mit seinen entsprechenden LED.

Einmal ist das Zeitintervall gewählt, muss den Benutzer auf den Darstellungsbutton drücken, in diesem Moment wird sich eine andere Registerkarte öffnen, deren Funktion ist, die Daten in der Zeitspanne durch eine Datenbankabfrage zu finden,

Wenn sie gefunden werden, werden sie in Tabellenform und grafisch projiziert. Sie bieten den Nutzer die Gelegenheit, diese Daten in einer Datei herunterzuladen.

Zudem gibt es einen Zurück-Button.

The screenshot shows a software interface titled "DATA HISTORY". At the top left, there is a logo for "PROMOTion" with the tagline "PROGRESS ON MESHED HVDC OFFSHORE TRANSMISSION NETWORKS". To the right are logos for "ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA" and "UNIVERSITAT POLITÈCNICA DE VALÈNCIA". The interface features a "Data" tab and a "Graph" tab, with a circled "6" next to "Data". Below the tabs is a data table with 10 rows and 5 columns, all containing "0". To the right of the table is a question: "Do you want to generate a data file to see all?". Below this question is a green "Generate" button with a circled "8" next to it. At the bottom center is a red "BACK" button with a circled "3" next to it. A circled "7" is placed below the data table. On the right side of the screenshot, there is a legend:

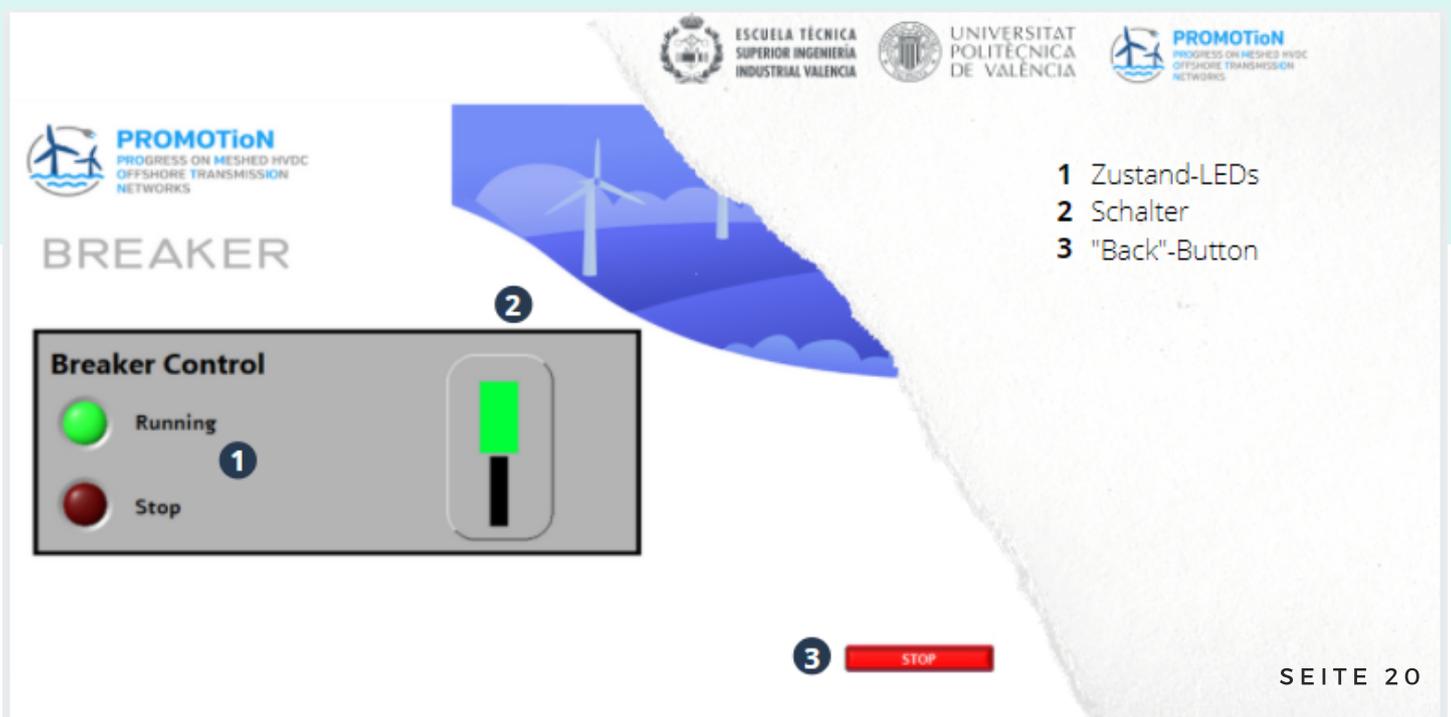
- 6 Menü
- 7 Datentabelle
- 8 Erzeugungstaste

"BREAKER"-KONTROLLE

Der "breaker" oder automatische Schalter ist das Element, das die Abschaltung des virtuellen Simulators und/oder des Prototypen im Massstab in jedem Moment der Simulation erlaubt, sodass mögliche Störfälle vermieden werden.

Dieses Fenster stellt sich aus einer Reihe an ziemlich einfachen Bestandteilen zusammen: einen grossen Stecker (der sich standardmässig aktiviert befindet) und zwei LED, die den Nutzer anschaulicher den Zustand des "Breakers" mitteilen.

Ausserdem, in ähnliche Weise als der Rest der Funktionen des Programms, enthält es einen Zurück-Button, der den Benutzer zum Menüfenster zurücksendet.



FEHLERCODES UND HÄUFIG GESTELLTE FRAGEN

Dieses Fenster ermöglicht den Nutzer des Programms, die Fehler, die sich von der Verwendung der anderen Registerkarten des Programms ableiten können ("Real Time", "Data History", und "Breaker" "Simulation"), zu konsultieren.

Zudem wird ein "tap control" ergänzt, um die Navigation und die Anzeige dieser "front view" erleichtert werden.

Wenn ein Fehler vorliegt, wird dieser benachrichtigt, indem der "status" LED rot eingeschaltet wird und der Code und eine knappe Erklärung des Fehlers angegeben werden, damit der Nutzer oder der Programmierer es lösen können.

ESCUOLA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

PROMOTion
PROGRESS ON MESHED HVDC OFFSHORE TRANSMISSION NETWORKS

ERROR LOG

1 Menü

2 Fehlerbildschirme

Real Time Voltage Error

status	code
●	id
source	

Real Time Active Power Error

status	code
●	id
source	

Real Time Reactive Power Error

status	code
●	id
source	

Real Time Current Error

status	code
●	id
source	

BACK

SEITE 21

FEHLERCODES UND HÄUFIG GESTELLTE FRAGEN

CODE	BESCHREIBUNG
#54	<i>Die Netzwerkadresse ist falsch geformt. Stellen Sie sicher, dass die Adresse ein gültiges Format hat. Für TCP/IP, kann die Adresse einen Maschinennamen oder eine IP-Adresse im Format xxx.xxx.xxx.xxx. Falls der Fehler auftritt, wenn ein Maschinennamen bestimmt wird, vergewissern Sie sich, dass der Maschinennamen gültig ist. Versuchen Sie, einen Ping-Test für den Maschinennamen durchzuführen. Überprüfen Sie, dass Sie einen richtig konfigurierten DNS-Server besitzen. Wenn Sie im Moment die Funktion Öffene Verbindung TCP verwenden, stellen Sie sicher, dass der Wert auf den Remoteport oder Servicename nicht 0 ist.</i>
#55	<i>Der Netzwerkbetrieb ist gerade im Gange. Falls Sie diesen Fehler erhalten, während Sie die UDP Write Funktion benutzen.</i>
#56	<i>Der Netzwerkbetrieb hat die Zeitgrenze überschritten, die der Nutzer oder das System vorgegeben hat.</i>
#57	<i>Die Netzwerkverbindung ist besetzt.</i>
#58	<i>Die Netzwerkfunktion ist mit dem System unvereinbar.</i>
#59	<i>Probleme mit der UDP Schrift im aktuellen Fenster.</i>
#60	<i>Die angegebenen Port oder Netzwerkadresse werden gerade gebraucht. Wählen Sie einen verfügbaren Port oder Netzwerkadresse.</i>
#61	<i>Das System hat verursacht, dass die Netzwerkverbindung abgebrochen wurde.</i>
#62	<i>Der Server hat die Netzwerkverbindung abgelehnt. Für TCP/IP</i>
#1011	<i>Man findet keine Daten für das ausgewählte Zeitintervall.</i>
#44	<i>Problem beim Versuch, eine Datei zu retten</i>

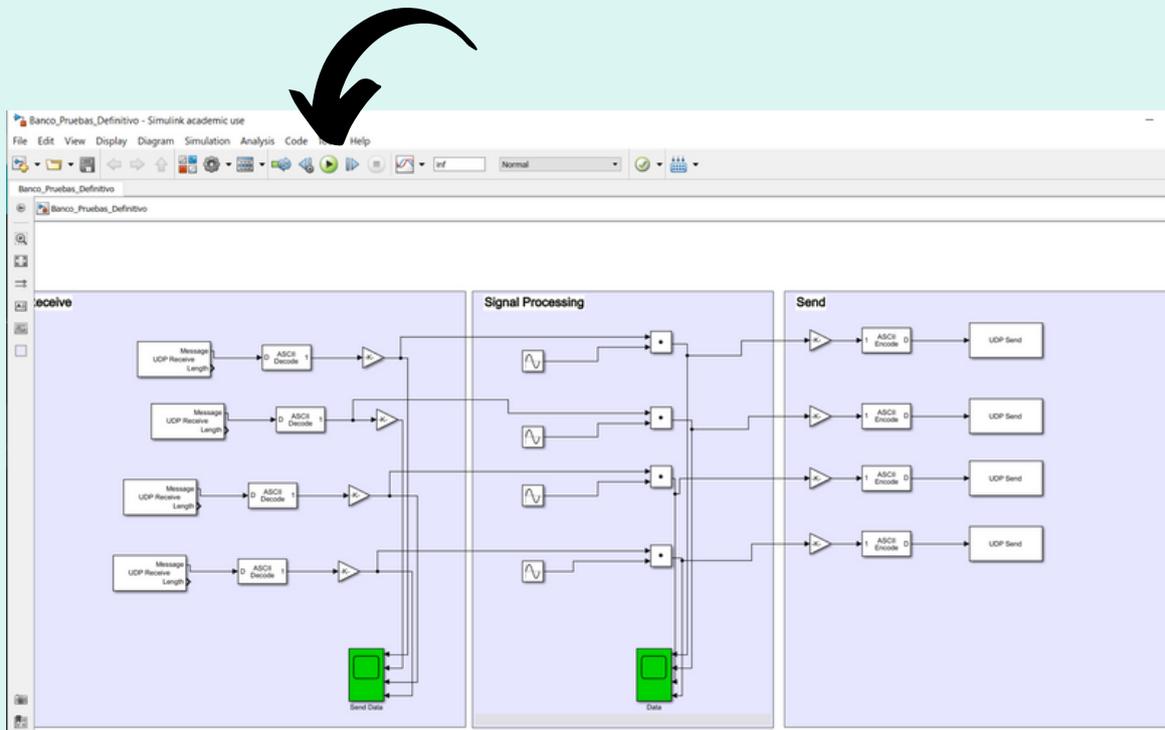
FEHLERCODES UND HÄUFIG GESTELLTE FRAGEN

Wie kann man den Code in LabVIEW ausführen?

Um einen Code in LabVIEW auszuführen, muss der Nutzer auf den "run"-Button im "log-in" Fenster drücken, denn dieses ist das Hauptfenster des Programms. Wenn Sie mehr über LabVIEW erfahren möchten, besuchen Sie folgende Seite: <https://www.ni.com/getting-started/labview-basics/esa/>

Wie kann man den Code in MATLAB ausführen?

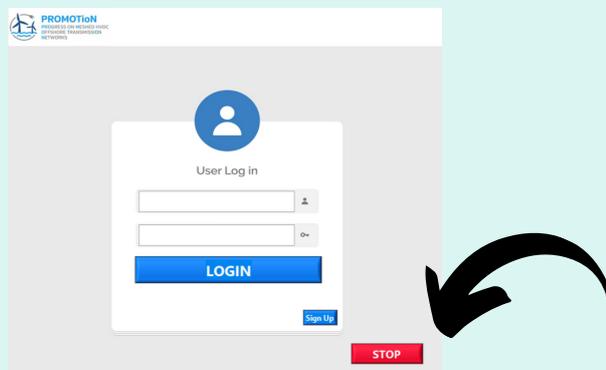
Um den Code in MATLAB auszuführen, genauer gesagt im Simulink, muss der Benutzer auf den "run"-Button drücken, wie man auf der folgenden Abbildung beobachten kann.



FEHLERCODES UND HÄUFIG GESTELLTE FRAGEN

Wie kann man den Code in LabVIEW beenden?

Das Programm ist so eingestellt, dass es beendet werden kann, ohne auf die "abort" Taste zu klicken. Dafür muss der Nutzer zum "log-in" Fenster zurückgehen, und auf Abschluss-Button des Programms drücken.



Wie kann man die UNIX Timestamp-Zeit zum echten Datum machen?

Um diese Umstellung durchzuführen, sind online Konverter wie <https://www.unixtimestamp.com/index.php> empfohlen

Was kann man tun, wenn Probleme während der Anmeldung geschehen?

Falls die Anmeldung Probleme ergibt, verfügt man wahrscheinlich nicht über eine gute Netzwerkverbindung, vermutlich wurde die E-Mail Adresse schon vorher verwendet. Um diese zu lösen, kann man entweder mit einer anderen E-Mail Adresse versuchen, oder die Verbindung checken.

Für weitere Fragen über das Programm können Sie an die E-Mail Adresse juyusta@etsii.upv.es schreiben

Parte VI

Manuales de programador

Capítulo VI.1

Manual de programador (Castellano)

JULIO YUSTAS TALAMANTES



MANUAL DE PROGRAMADOR

CONTENIDO



1
**ACERCA DEL
PROGRAMA**

2
**ESTRUCTURA DEL
PROGRAMA**

3
INTERFAZ DE USUARIO

4
REQUISITOS TÉCNICOS

5
INSTALACIÓN

6
INICIAR APLICACIÓN

7
**REGISTRO
"BLOCK DIAGRAM"**

CONTENIDO



8
IDENTIFICACIÓN
"BLOCK DIAGRAM"

9
MENÚ PRINCIPAL
"BLOCK DIAGRAM"

10
DATOS A TIEMPO REAL
"BLOCK DIAGRAM"

11
SIMULADOR
"BLOCK DIAGRAM"

12
HISTORIAL DE DATOS
"BLOCK DIAGRAM"

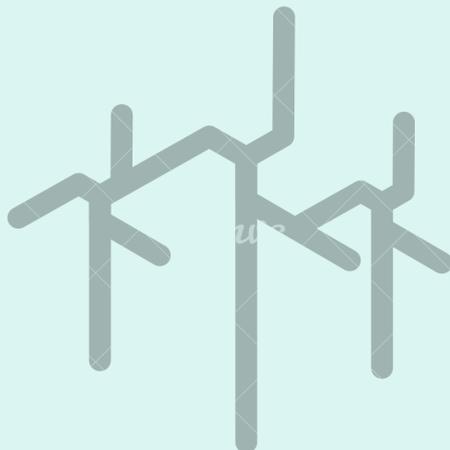
13
INTERRUPTOR
AUTOMÁTICO
"BLOCK DIAGRAM"

14
CÓDIGOS DE ERROR Y
FAQS



ACERCA DEL PROGRAMA

El presente programa ha sido desarrollado como parte del Trabajo de Final de Grado de Julio Yustas Talamantes; dicho trabajo se incluye como integrante de la investigación de la Universitat Politècnica de València para el proyecto eólico europeo "**PROMOTioN**" cuyo objetivo principal es explotar todos los recursos eólicos del mar del norte.



Este programa se ha desarrollado para su uso como aplicación de monitorización de parques eólicos marinos.

Su cometido principal es proporcionar al operario de una interfaz clara y sencilla para el correcto desempeño de su trabajo.

A su vez, se postula como los cimientos desde los cuales construir un programa mucho más sofisticado y para el cual se ha proporcionado el presente manual.



ACERCA DEL PROGRAMA

Los autores de esta aplicación han sido Julio Yustas Talamantes bajo la tutorización de la Doctora Soledad Inmaculada Bernal Pérez (Doctora Ingeniera Industrial).

El presente trabajo se ha desarrollado gracias a la colaboración de la **Universitat Politècnica de València** y el Instituto Universitario de Automática e Informática Industrial (ai2).



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

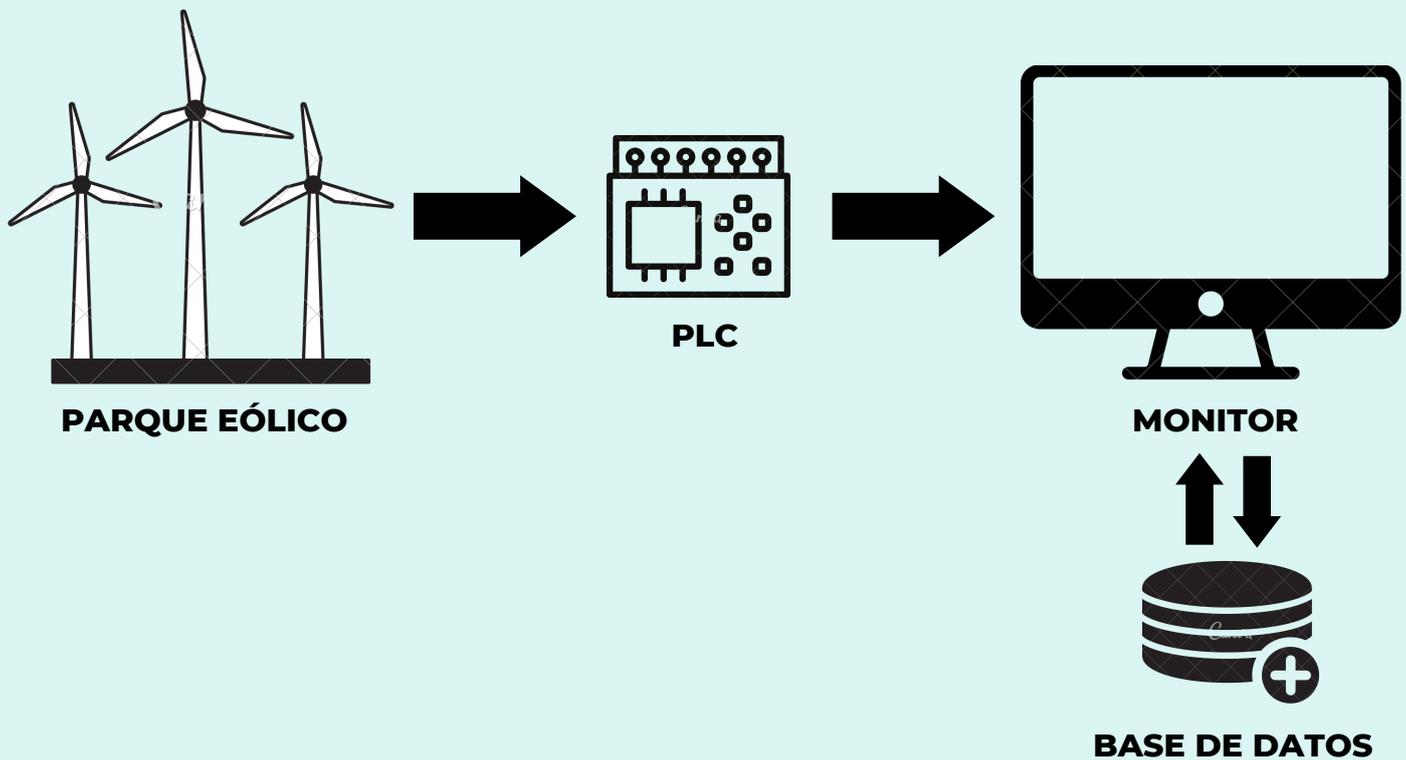


ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA



PROMOTION
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS





ESTRUCTURA DEL PROGRAMA

Se trata de un sistema SCADA (Supervisión, Control y Adquisición de Datos).

El objetivo principal es controlar un parque eólico desde otro lugar de forma remota, para ello necesitaremos tres elementos clave: el parque eólico, los PLCs y el sistema de monitorización.

Las conexiones entre dichas partes son una parte fundamental para el correcto desempeño del presente programa.

La unión entre el parque eólico y los sensores se realizará mediante un cableado.

Dichos sensores estarán conectados a su vez a la red, con el objetivo de permitir la comunicación mediante protocolo UDP con el simulador o programa SCADA,

Es trabajo del programa SCADA, tratar estos datos y acomodarlos a un entorno atractivo para el trabajador.

A su vez, almacenará dichos datos en una base de datos para permitir su posible consulta posterior,

INTERFAZ DE USUARIO

El programa cuenta con una interfaz de usuario sencilla y atractiva realizada con la ayuda de LabVIEW y Photoshop.

La navegación es sencilla: para avanzar entre las diversas pestañas solo tenemos que elegir la modalidad desde el menú principal o "main menu".

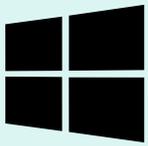
En el caso de querer retroceder, se cuenta en todo momento con un botón de "back" que nos envía a la página anterior.

Para finalizar el programa simplemente tenemos que volver a la pestaña de identificación y darle click al botón de "stop" o en su defecto utilizar el clásico botón de cerrar pestañas de windows.



REQUISITOS TÉCNICOS

DESDE UN SIMULADOR VIRTUAL:



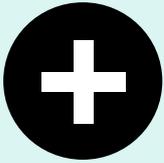
SISTEMA OPERATIVO

Windows 7, Ubuntu 16.04 o macOS Sierra en 64 bits.



NAVEGADOR Y CONEXIÓN RED

Internet Explorer 11, Google Chrome 58 o Firefox 53



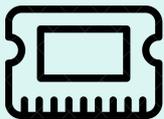
MATLAB

R2017b o Superior



PROCESADOR

Intel Core I5 o superior



MEMORIA RAM

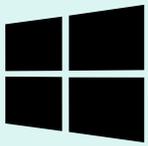
Memoria RAM 8G o superior



TARJETA DE SONIDO Y ALTAVOCES

REQUISITOS TÉCNICOS

DESDE UNA GRANJA EÓLICA REAL:



SISTEMA OPERATIVO

Windows 7, Ubuntu 16.04 o macOS Sierra en 64 bits.

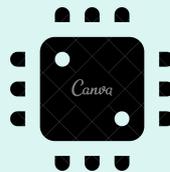


NAVEGADOR

Internet Explorer 11, Google Chrome 58 o Firefox 53

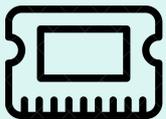


CONEXIÓN RED



PROCESADOR

Intel Core I5 o superior



MEMORIA RAM

Memoria RAM 8G o superior



TARJETA DE SONIDO Y ALTAVOCES

INSTALACIÓN

La instalación del presente programa es bastante sencilla, para ello solo necesitamos seguir una serie de pasos:

La primera etapa consiste en descargar LabVIEW con las extensiones siguientes: *LabVIEW Control Design and Simulation Module (32-bit)*, *LabVIEW Database Connectivity Toolkit (32-bit)*, *LabVIEW Report Generation Toolkit (32-bit)* y *LabVIEW Runtime (32-bit)*.

Una vez descargado el programa se deberá descargar el archivo .zip adjuntado en el repositorio y descomprimirlo. Posteriormente ejecutar el proyecto PROMOTioN.

Si se desea utilizarlo junto al simulador, deberemos instalar en esa misma computadora, o en otra conectada a la misma red WIFI, el programa MATLAB y ejecutar los simuladores: *parque.mdl* y *simulator.mdl*

Si, por el contrario, se desea utilizar un prototipo a escala o una granja eólica real, deberemos de realizar una sincronización de los puertos de envío de los PLCs y los puertos de entrada del programa de acuerdo a la siguiente tabla:

Variable a controlar	Puerto
Tensión	8811
Corriente	8822
Potencia Activa	8833
Potencia Reactiva	8844

Dicha sincronización se realizará en el prototipo a escala o el parque eólico, es decir, siempre sobre el PLC y no sobre el programa.

Control	Envío
Tensión	7700
Corriente	7711
Potencia Activa	7722
Potencia Reactiva	7733

La anterior configuración se usará para la funcionalidad de simulador; dichos puertos son los de envío por parte del simulador y/o PLC.

INICIALIZACIÓN

Para garantizar la correcta inicialización del programa, deberemos comprobar que ambas partes: el ordenador ("host" del sistema SCADA) y los PLCs disponen de conexión a internet.

A su vez, se tendrá que comprobar, de forma análoga a lo expuesto en el apartado anterior, que existe una correcta disposición de los puertos de entrada y salida.

Este último paso es fundamental para garantizar el éxito de la comunicación mediante el protocolo UDP.

Por otro lado, el usuario deberá registrarse con los credenciales proporcionados por la empresa que serán únicos e intranferibles.

En el caso de no disponer de ellos, se podrá dar de alta en la base de datos de operarios autorizados mediante la pestaña de "sign-up".

Esta última pestaña se explicará con más detenimiento en los siguientes puntos de este mismo manual.

A continuación se vuelve a adjuntar la tabla de conexiones UDP:

Variable a controlar	Puerto
Tensión	8811
Corriente	8822
Potencia Activa	8833
Potencia Reactiva	8844

Por último, para empezar a recibir datos del parque eólico, lo único que resta es pulsar sobre el interruptor de inicio de conexión, presente en la pestaña de "main menu".

REGISTRO "BLOCK DIAGRAM"

La pestaña "sign-up" permite que los usuarios puedan registrarse dentro de la base de datos de clientes.

En los recortes adjuntados se aprecia el código de programación para la ventana de registro de usuario.

El primer componente a destacar es un gran bucle "while" dentro del cual se engloban varios elementos:

- Un bucle de la misma tipología cuyo objetivo es repetir el programa tantas veces como sea necesario hasta que se rellenen los campos de registro.
- Un reinicio de las variables a completar, esto se realizará al inicio de la ventana y en el caso de que se tenga que repetir el registro por ser incorrecto.
- La comprobación de que las diferentes condiciones para que el registro se haya realizado de forma exitosa se cumplen: todos los campos completados y el usuario y correo electrónico no utilizado previamente.
- Además, podemos encontrar un timing y sendas condiciones para la finalización de ambos bucles.

- 1** Bucle "while".
- 2** Reinicio de variables
- 3** Temporizador
- 4** Clavija registro
- 5** Nombre de usuario
- 6** Correo electrónico
- 7** Escritura en b. de datos
- 8** Condicionales
- 9** Mensaje
- 10** Estructura de envío
- 11** Mensajes de error

REGISTRO "BLOCK DIAGRAM"

Continuando en el exterior de la mencionada estructura, encontramos una estructura condicional que tiene la función de almacenar los parámetros en la base de datos "login.lvm", en el caso de que se hayan cumplido todas las condiciones anteriores.

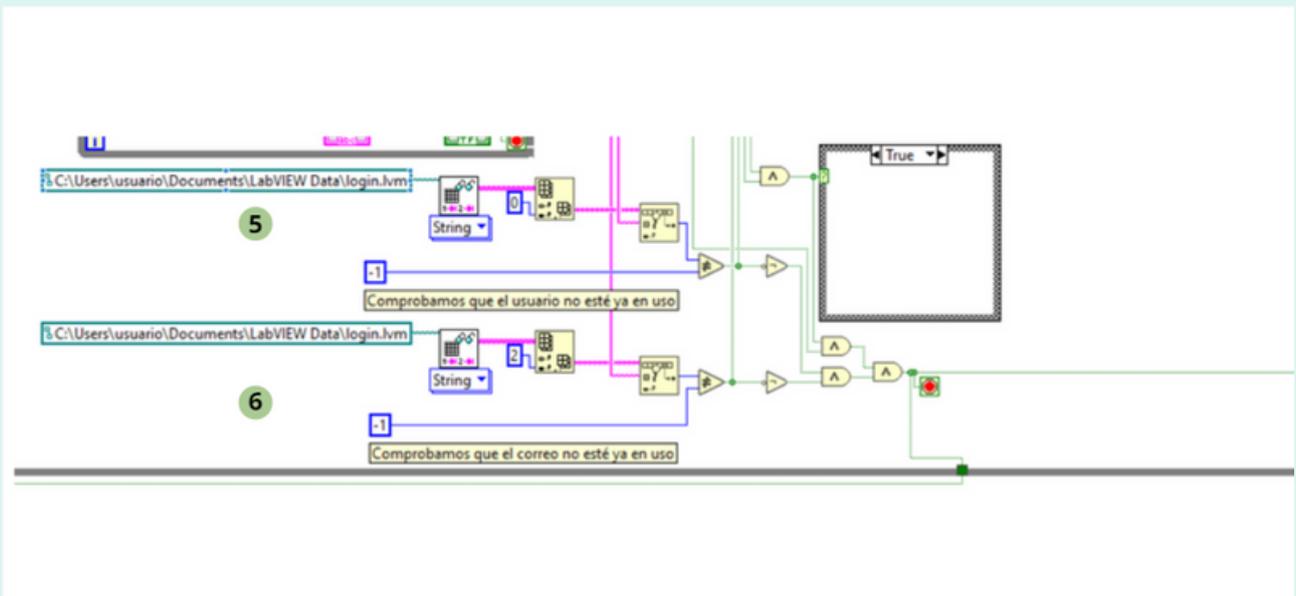
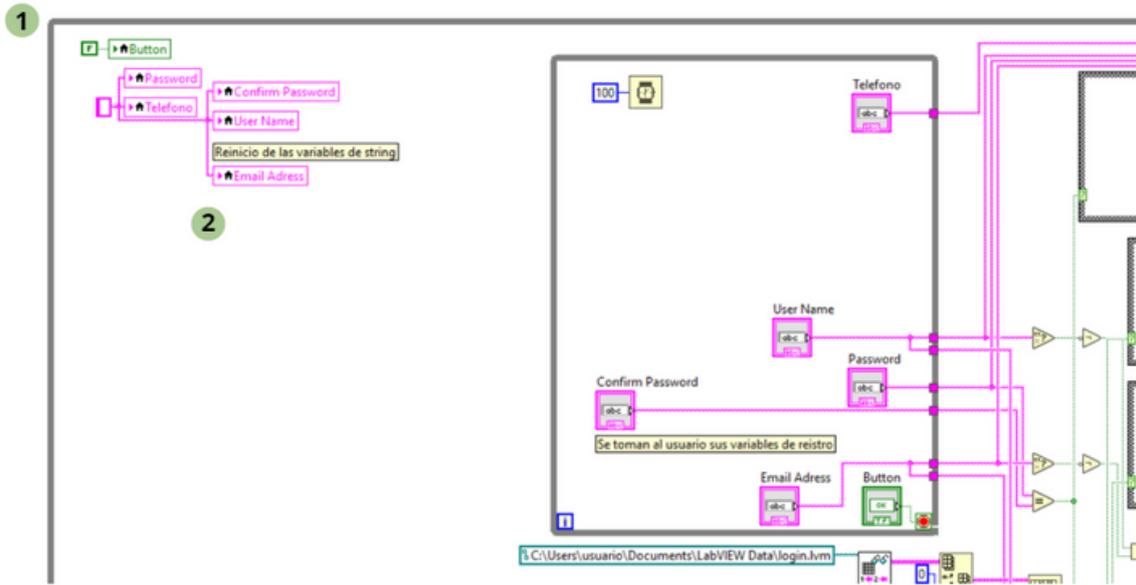
Por último, podemos observar un inicio de una serie de variables de tipo "string", texto, utilizadas posteriormente en la estructura que encontramos justo debajo y cuya función es enviar un correo de confirmación al usuario que se acaba de registrar.

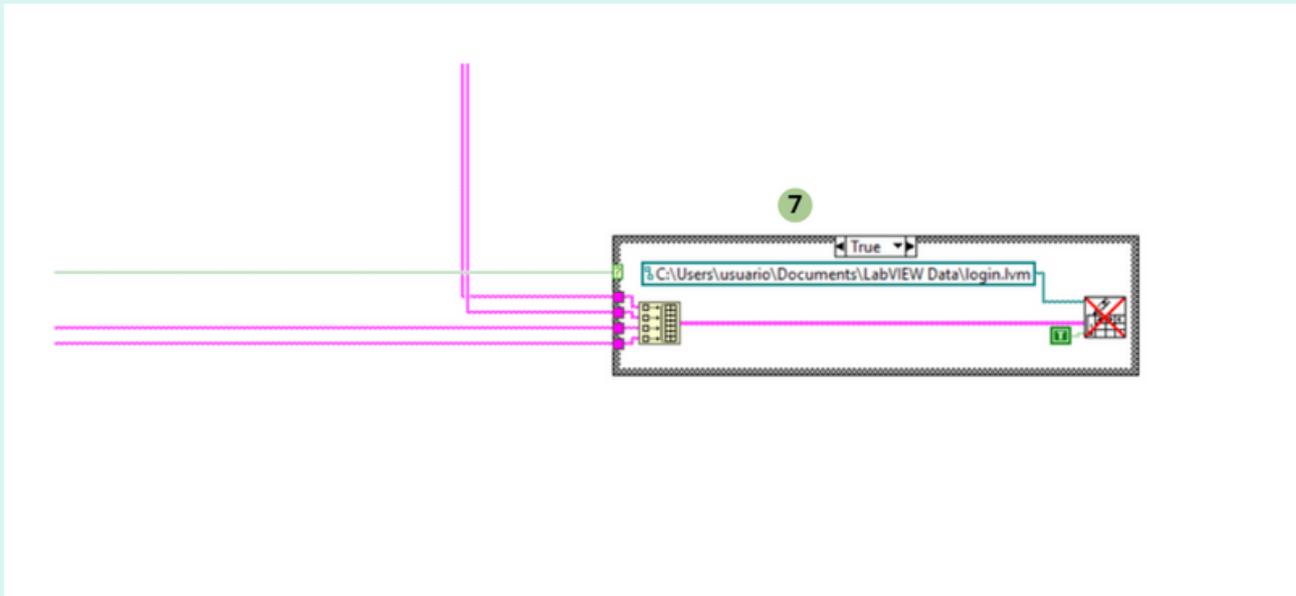
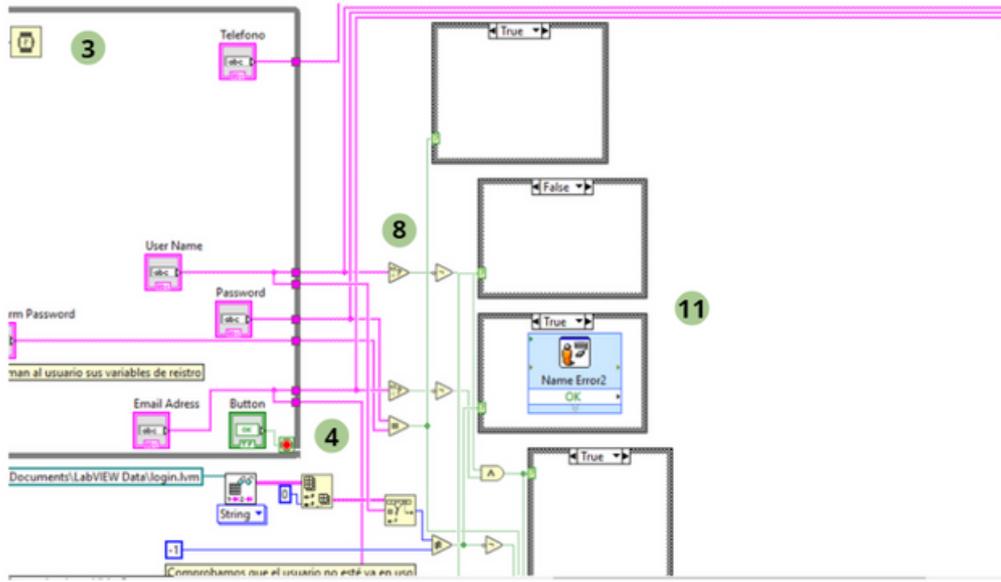
En caso de que todo lo mencionado anteriormente se haya llevado a cabo de forma exitosa, el programa generará un mensaje de confirmación al usuario.

Los principales elementos a modificar por parte de un programador son: la sustitución de la base de datos tipo .lvm por otro tipo de base de datos más sofisticada; la introducción de un elemento de confirmación mediante una combinación aleatoria de números enviada en el correo de confirmación, etc.

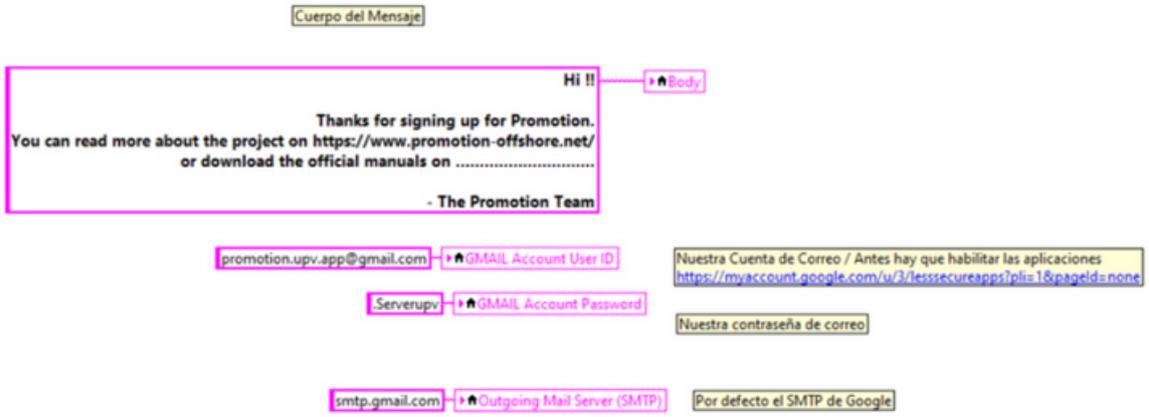
A su vez, se podrá modificar el contenido del mensaje y el diseño del mismo.

- 1** Bucle "while".
- 2** Reinicio de variables
- 3** Temporizador
- 4** Clavija registro
- 5** Nombre de usuario
- 6** Correo electrónico
- 7** Escritura en b. de datos
- 8** Condicionales
- 9** Mensaje
- 10** Estructura de envío
- 11** Mensajes de error

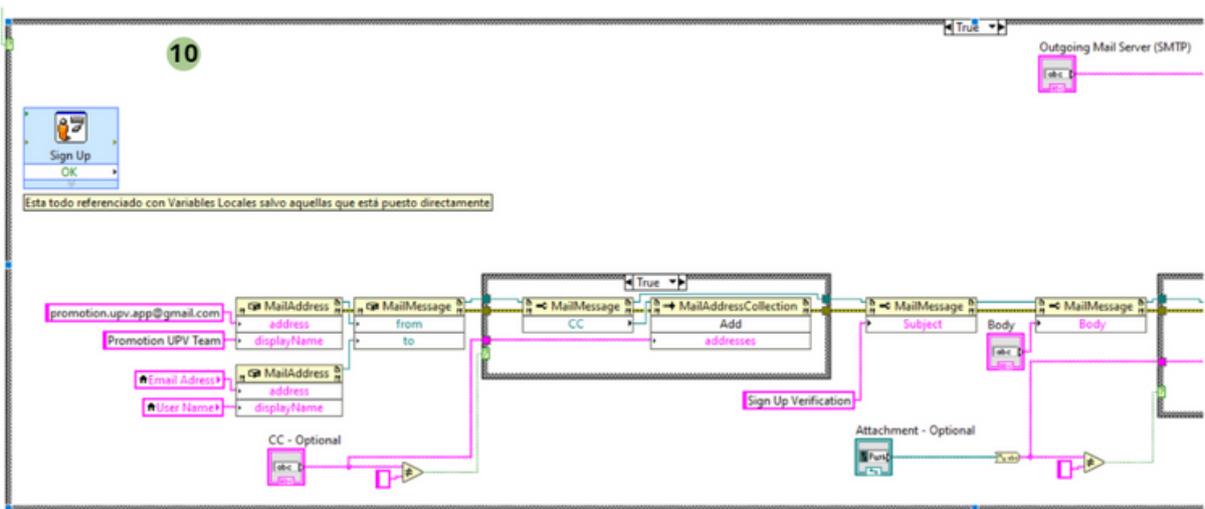


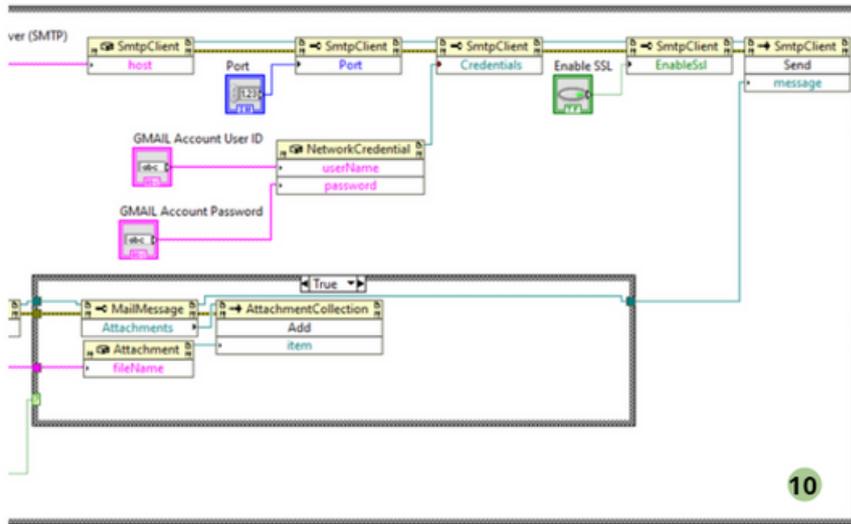


9



10





IDENTIFICACIÓN "BLOCK DIAGRAM"

Se presenta ahora el diagrama de bloques o programación del entorno gráfico para la identificación de usuario.

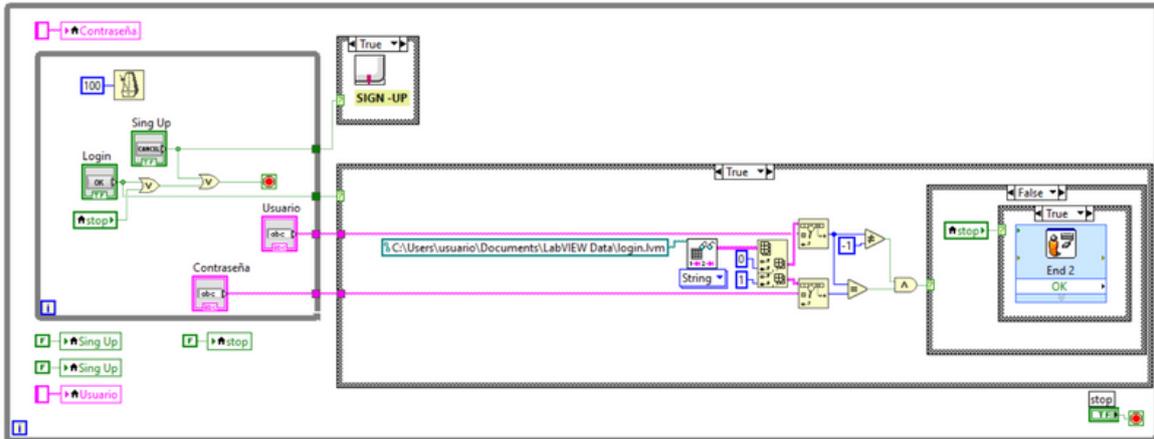
El presente código se sitúa en el interior de un bucle "while" cuya función es repetir el programa, siempre y cuando no se valide la opción que finaliza el bucle de repetición, es decir, botón de "stop" pulsado, lo cual generaría un mensaje de cierre y finalizaría el programa; a la vez que reinicia las variables y los campos de relleno.

Dentro de este bucle se encuentra otro bucle "while" con las casillas para introducir las credenciales: nombre de usuario y contraseña, y las clavijas ("login" y "sign-up"). Serán estas dos últimas las encargadas de permitir el avance hasta los siguientes bloques.

Sendos bloques son condicionales: el primero de ellos ejecutará la pestaña de registro en el caso de que la validación del bucle "while" anterior se haya hecho mediante el pulsado del botón "sign-up".

El segundo condicional se ejecuta cuando se pulsa el botón de "login"; en este momento el programa abre la base de datos "login.lvm" y se produce una consulta línea a línea del nombre de usuario y contraseña. Dependiendo de si los datos introducidos se han encontrado o no, el programa autorizará al usuario pasar a la pestaña de "main menu" o le emitirá un mensaje de error y le instará a continuar probando, lo que devolvería al cliente al estado inicial.

- | | |
|--------------------------------|-------------------|
| 1 Bucle "while". | 7 Búsqueda |
| 2 Reinicio de variables | 8 Acciones |
| 3 Temporizador | |
| 4 Clavijas | |
| 5 Campos de relleno | |
| 6 Estructura "sign-up" | |



MENÚ PRINCIPAL "BLOCK DIAGRAM"

En este "block diagram" encontramos dos elementos clave que actúan de forma simultánea y totalmente independiente:

Un primer bucle "while" que repite el código desde el inicio del programa a la vez que se reinician las variables utilizadas en el mismo; este bucle se repite constantemente incluso si se valida alguna de las estructuras condicionales que se localizan dentro del mismo. La única forma de finalizarlo es tecleando sobre el botón de "back".

Otro bucle "while" dentro de una estructura condicional que se habilita una vez se conecta la clavija de recepción de datos en tiempo real; dentro de este bucle tenemos, repetido tantas veces como datos tenemos a recibir, el proceso de apertura, lectura

y cierre del protocolo UDP, así como el tratado de la información para poder representarla de forma correcta en la "front view".

En este diagrama de bloques no hay nada a modificar a no ser que el programador desee incorporar una nueva funcionalidad al programa o modificar el recibo de datos en tiempo real.

- 1** Reinicio de variables
- 2** Bucle "while".
- 3** Temporizador
- 4** Clavijas
- 5** Inicio recepción datos
- 6** Estructura recepción

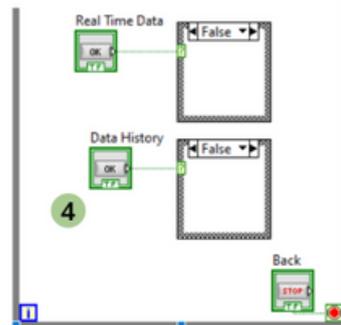
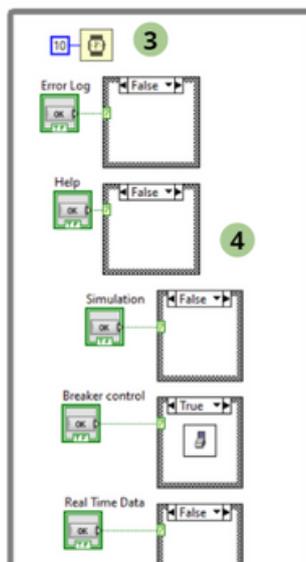
- 7** Reinicio variables protocolo UDP

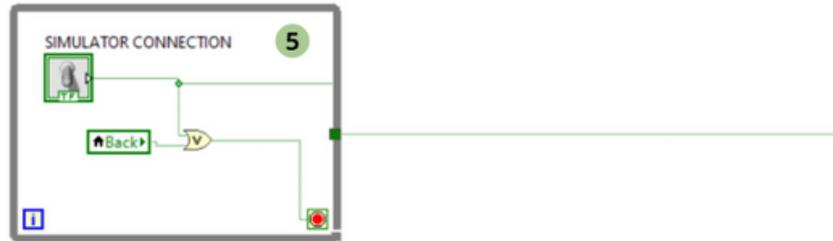
1

▶▶SIMULATOR CONNECTION

- ▶▶Help
- ▶▶Back
- ▶▶Simulation
- ▶▶Data History
- ▶▶Real Time Data
- ▶▶Error Log
- ▶▶Breaker control

2

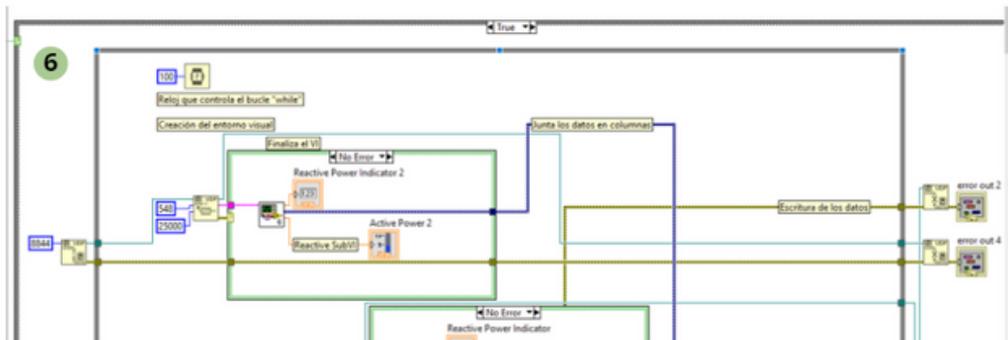




7

- Voltage Indicator
- Active Power
- Reactive Power
- Active Power
- Active Power Indicator
- Reactive Power Indicator

6



DATOS EN TIEMPO REAL

El entorno de programación de esta ventana cuenta con un bucle "while" dentro del cual se halla la clavija de conexión con el simulador.

Una vez conectada, se validará la estructura de casos. Dentro de esta tenemos otro bucle "while" que ejecuta el conjunto de bloques de recepción de datos de forma continuada hasta que se pulse el botón de retroceso.

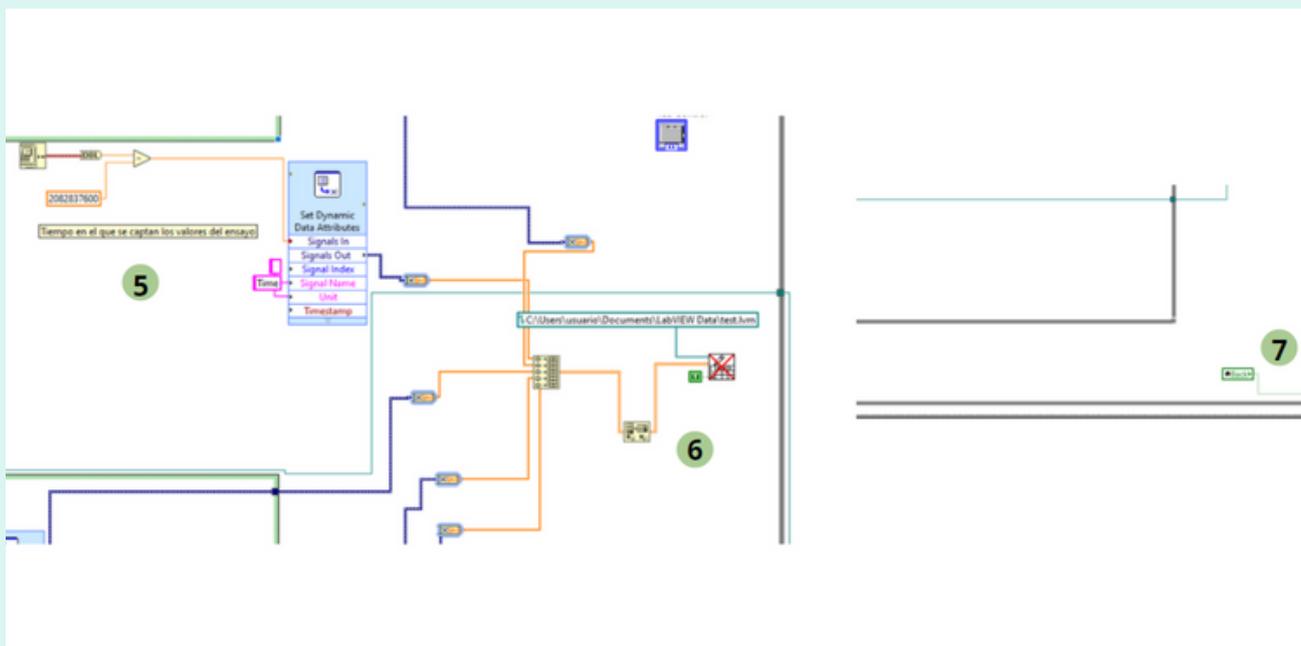
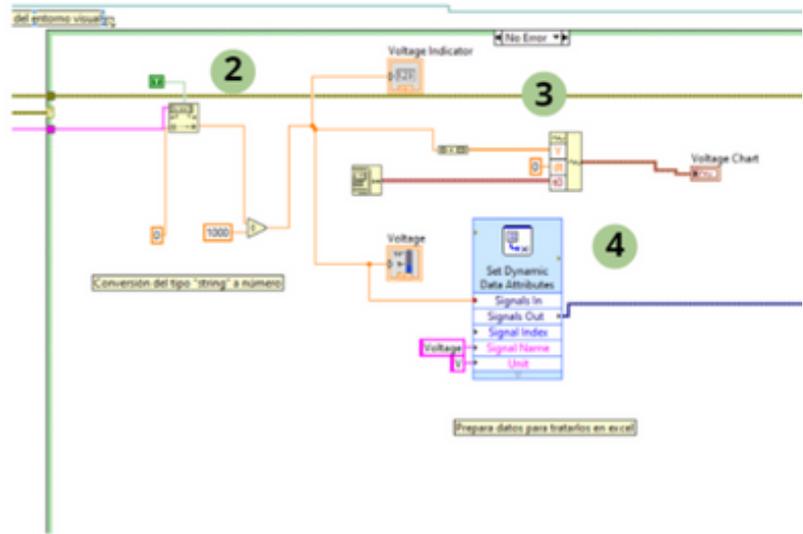
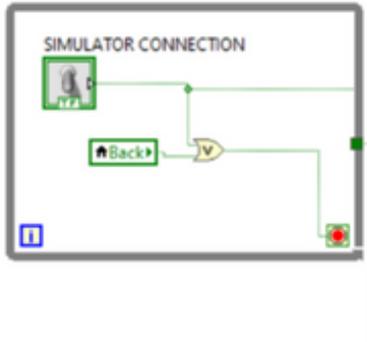
La recepción de datos consta de un bloque de apertura del protocolo de envío UDP dónde se asigna el puerto local a utilizar,

un bloque de lectura que recibe la señal, una serie de bloques de tratamiento de señal y un bloque de cierre del protocolo. Como en el resto de las ventanas tenemos también unos indicadores de errores que serán utilizados por la pestaña de errores.

Las modificaciones podrían ser cambiar el tipo de base de datos, cambiar los puertos utilizados o el tipo de variables recibidas; así como la naturaleza de los indicadores utilizados y la frecuencia de recogida de datos.

- 1** Bucle inicio
- 2** Recepción de datos
- 3** Indicadores
- 4** Preparación de datos
- 5** Tiempo
- 6** Escritura de datos
- 7** Botón de retroceso

1 SIMULATOR CONNECTION



SIMULADOR "BLOCK DIAGRAM"

Este "block diagram" reúne aspectos ya utilizados en las pestañas de "breaker" y de "real time".

Se trata de un gran bucle "while", dentro del cual se realizan dos tareas simultáneas: el envío de datos y la recepción de datos vía UDP. Para ello, se seguirá el esquema típico de "apertura, envío y cierre" o "apertura, recepción y cierre", según el caso.

Tanto en el envío como en la recepción se dispone de un indicador de error, utilizado por la página de "Error Log"; por su parte, la recepción continúa con un sistema de ploteo.

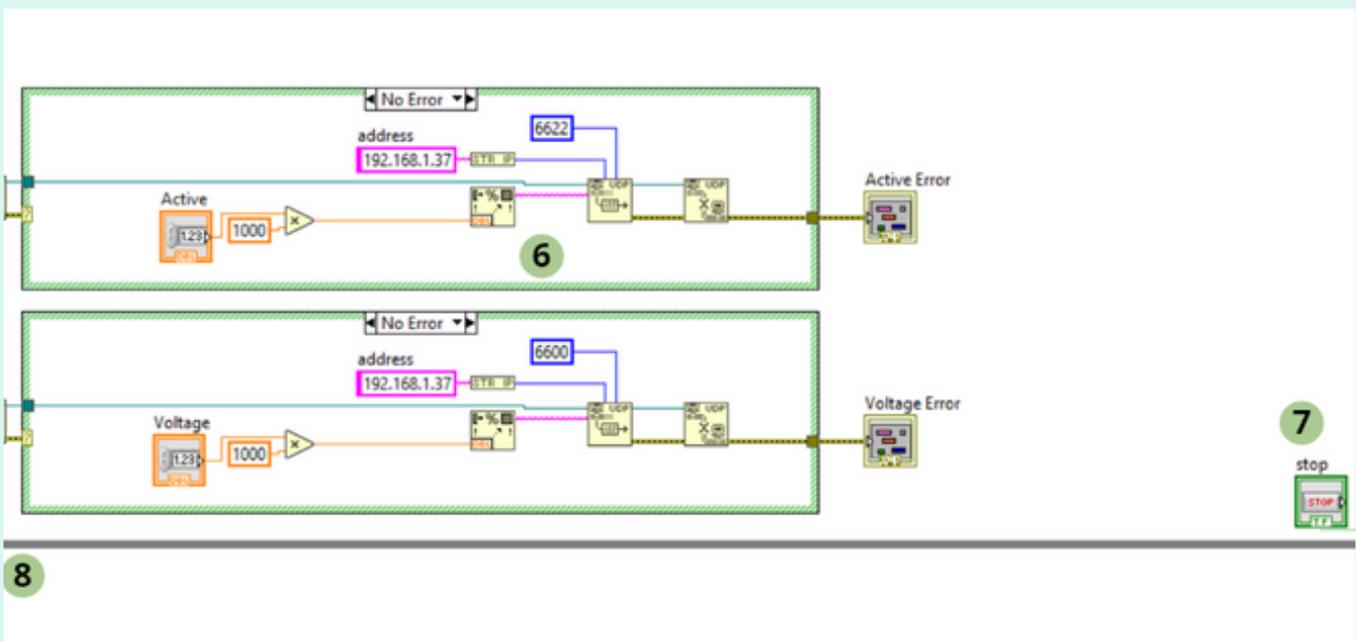
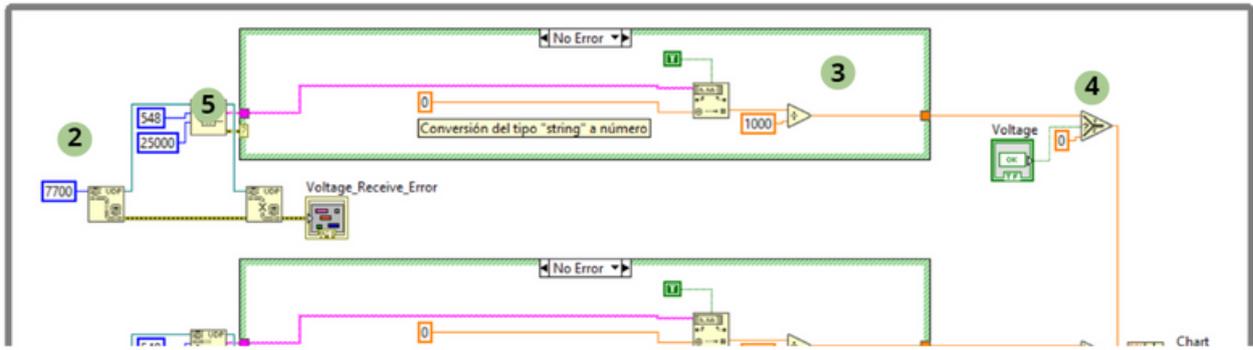
Se ha empleado un operador booleano para ofrecer al usuario la posibilidad de elegir si se plotea o no la variable en cuestión, acompañado de una gráfica para visualizar los mismos.

Se dispone, al igual que en el resto de las pestañas, de una clavija de retroceso que envía al usuario al menú principal.

Las variables de recepción y envío, así como los indicadores son totalmente modificables a gusto del programador.

- 1** Reinicio de variables
- 2** Apertura UDP
- 3** Recepción de datos
- 4** Clavijas
- 5** Gráfica
- 6** Estructura de envío
- 7** Botón de "back"
- 8** Bucle "while"

1



HISTORIAL DE DATOS

Esta funcionalidad del programa esta compuesta por dos subpestañas: la primera de ellas permite seleccionar la fecha y la segunda representa los datos.

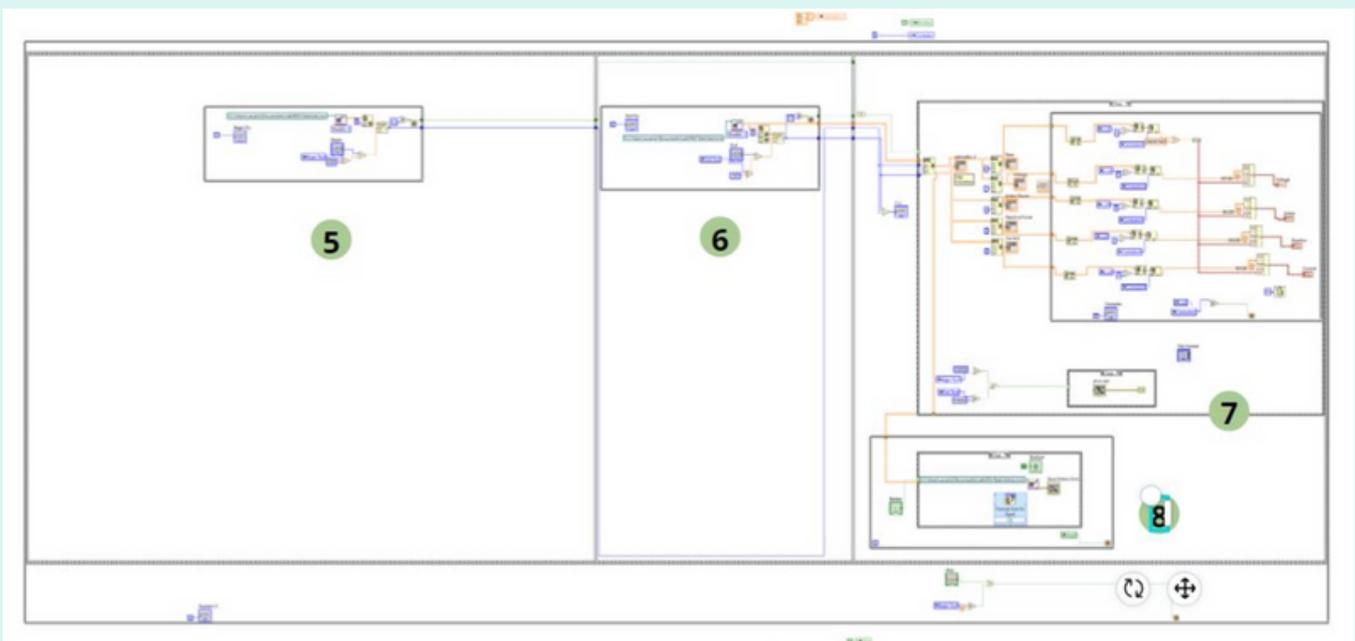
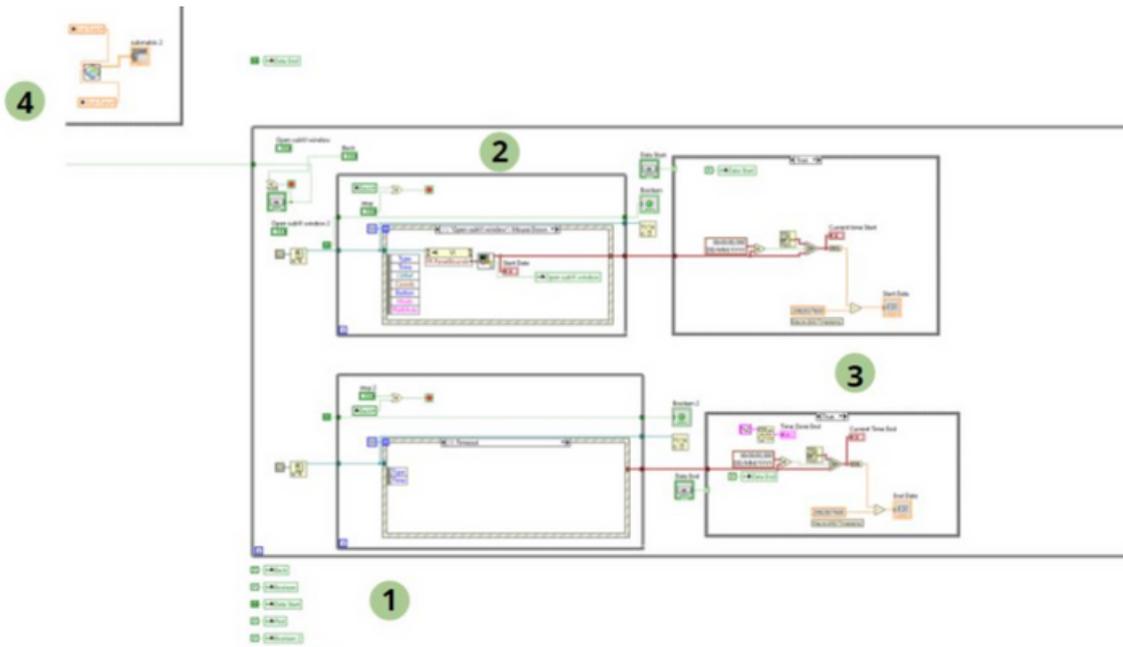
Para la selección de fecha se dispone de un gran bucle "while" dentro del cual encontramos sendos bloques condicionales que se ejecutarán cuando se seleccione la fecha y se confirme. El gran bucle "while" finalizará cuando se cumplan estos dos parámetros y el usuario pulse sobre el botón de plotear.

En ese momento el programa abrirá la otra pestaña, esta segunda ventana está constituida por una secuencia: dos bucles "while",

dónde el software busca los datos más cercanos y dos últimos bucles "while", el primero formado por bloques de tratamiento de matrices para el ploteo y el segundo con un condicional para el almacenaje de los datos ploteados. Todo esto dentro de un gran bucle "while" que se ejecutará hasta que el usuario indique lo contrario.

Este es el código más desarrollado del software y cualquier modificación requiere de mucho cuidado para no desajustar el conjunto; se recomienda no realizar ningún cambio fuera de la posible modificación de la base de datos.

- 1** Reinicio de variables
- 2** Bucle "while"
- 3** Estructuras condicionales
- 4** Apertura segunda pestaña
- 5** Búsqueda de inicio
- 6** Búsqueda de fin
- 7** Ploteo
- 8** Salvado de datos



INTERRUPTOR AUTOMÁTICO

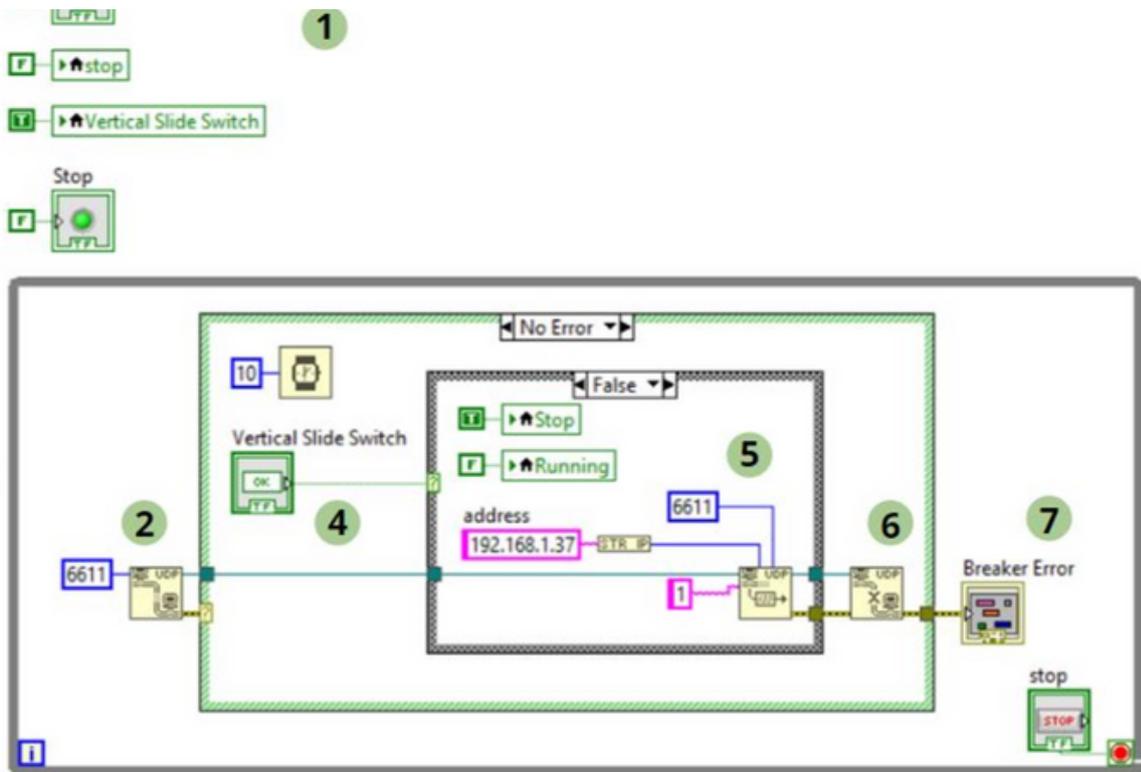
El código de programación de esta ventana cuenta con un reinicio de las variables (incluidos los dos LED) y un bucle "while" que, de forma análoga a los anteriores, se reproduce hasta que se valida la condición de salida.

Dentro de este bucle encontramos un envío vía protocolo UDP; primero abrimos el protocolo y asignamos un puerto (en este caso el 6611).

Si este proceso se realiza sin problemas, se validará la estructura condicional. Dentro de ella tenemos el interruptor, y según su posición, se abrirá una u otra parte de la estructura condicional, enviando un 1 si se ha abierto el circuito o un 0 si, por el contrario, está cerrado.

Dentro de esta última estructura tenemos el envío que se realizará después de introducir el puerto de destino y la dirección (la cual habrá que modificar de dato numérico a "string").

- | | |
|---------------------------------|-----------------------------|
| 1 Reinicio de variables | 7 Error |
| 2 Apertura UDP | 8 Botón de retroceso |
| 3 Bucle "while" | |
| 4 Interruptor | |
| 5 Estructura condicional | |
| 6 Cierre protocolo UDP | |



CÓDIGOS DE ERROR Y FAQs

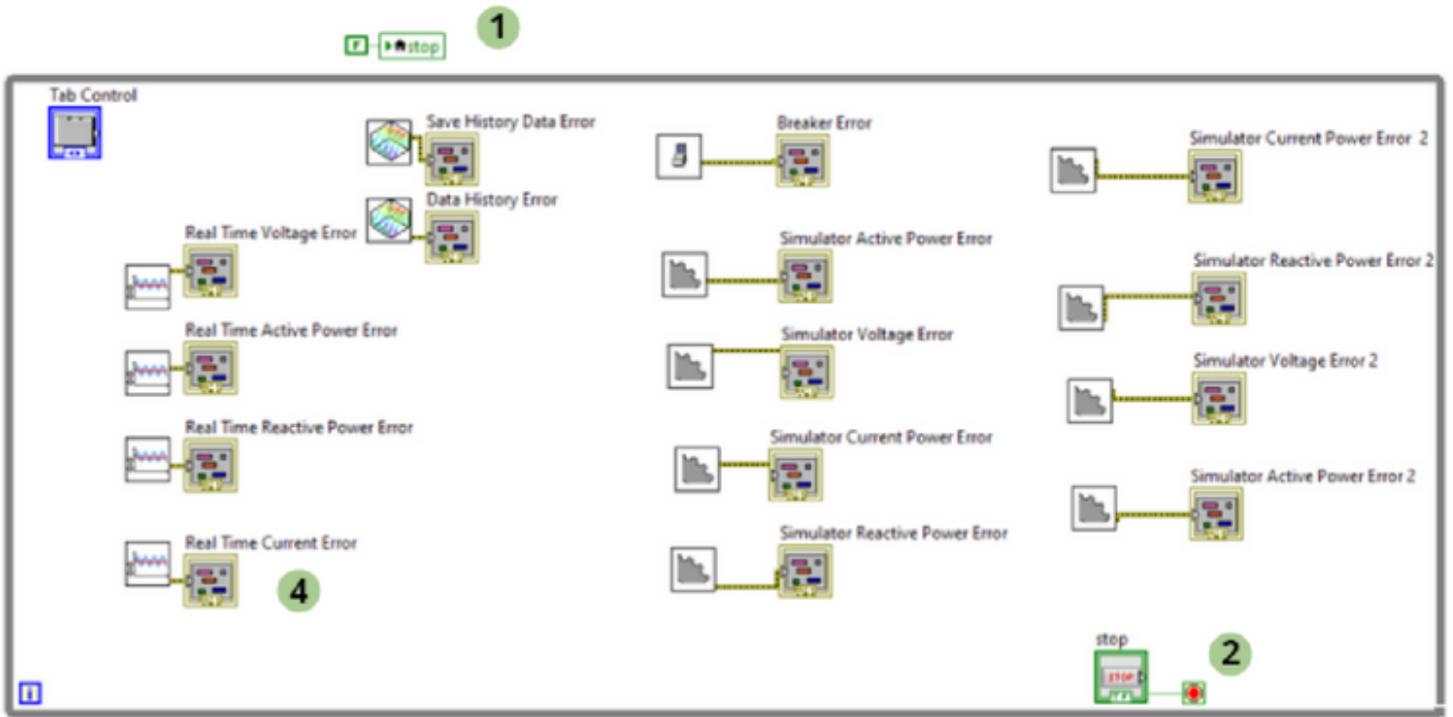
El entorno de programación de la pestaña de errores no es más que una especie de "repositorio" donde se agrupa toda la información sobre errores presente en el resto de las pestañas, y que se ha hecho invisible por motivos de comodidad.

Por lo tanto, se dispondrá de un bucle "while" que se repetirá mientras no se cumpla la condición de salida y, dentro de él, se encontrarán las llamadas a los errores de las diferentes ventanas junto a un indicador gráfico para cada una de ellas.

La única modificación posible es añadir más indicadores de error, para ello, habrá que seguir con la estructura expuesta: llamada a la pestaña + indicador.

- 1** Reinicio de variables
- 2** Botón de retroceso
- 3** Bucle "while"
- 4** Llamadas a los vi

3



CÓDIGOS DE ERROR Y FAQS

CÓDIGO

DESCRIPCIÓN

#54	<i>La dirección de red está mal formada. Asegúrese de que la dirección tenga un formato válido. Para TCP / IP, la dirección puede ser un nombre de máquina o una dirección IP con el formato xxx.xxx.xxx.xxx. Si este error ocurre al especificar un nombre de máquina, asegúrese de que el nombre de la máquina sea válido. Intente hacer ping al nombre de la máquina. Compruebe que tiene un servidor DNS configurado correctamente. Si está utilizando la función Conexión abierta TCP, asegúrese de que el valor del puerto remoto o el nombre del servicio no sea 0.</i>
#55	<i>La operación de la red está en curso. Si recibe este error mientras usa la función UDP Write.</i>
#56	<i>La operación de la red excedió el límite de tiempo especificado por el usuario o del sistema.</i>
#57	<i>La conexión de red está ocupada.</i>
#58	<i>La función de red no es compatible con el sistema.</i>
#59	<i>Problemas con la escritura UDP en la ventana actual.</i>
#60	<i>El puerto o la dirección de red especificados están actualmente en uso. Seleccione un puerto o una dirección de red disponibles.</i>
#61	<i>El sistema provocó que se cancelara la conexión de red.</i>
#62	<i>El servidor rechazó la conexión de red. Para TCP / IP</i>
#1011	<i>No se encuentran datos para el intervalo temporal seleccionado.</i>
#44	<i>Problema al intentar salvar un fichero</i>

CÓDIGOS DE ERROR Y FAQS

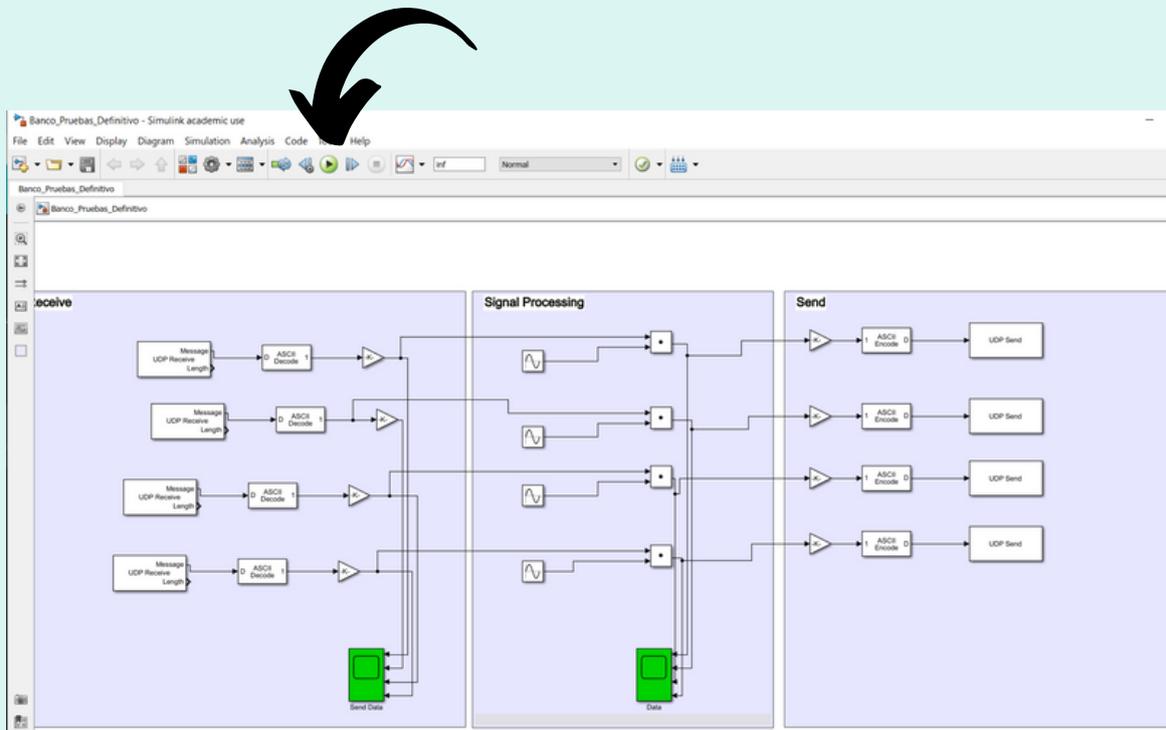
¿Cómo ejecutar el código en LabVIEW?

Para ejecutar un código en LabVIEW el usuario deberá pulsar sobre el botón de "run" dentro de la ventana de "log-in" ya que esta es la ventana principal del programa.

Para saber más sobre LabVIEW consulte: <https://www.ni.com/getting-started/labview-basics/esa/>

¿Cómo ejecutar el código en MATLAB?

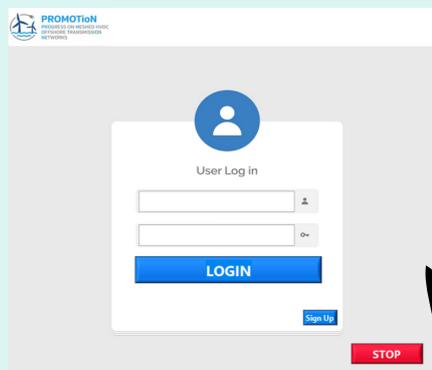
Para ejecutar un código en MATLAB, más concretamente en Simulink. El usuario deberá pulsar sobre el botón de "run" como se puede observar en la siguiente figura.



CÓDIGOS DE ERROR Y FAQS

¿Cómo finalizar el código en LabVIEW?

El programa esta configurado para poder finalizarse sin la necesidad de pulsar el botón de "abort". Para ello, el usuario deberá retroceder hasta la ventana de "log-in" y pulsar sobre el botón de finalización del programa.



¿Cómo pasar el tiempo de UNIX Timestamp a la fecha real?

Para realizar esta conversión se recomienda utilizar convertidores online como: <https://www.unixtimestamp.com/index.php>

¿Qué hacer si tengo problemas durante el registro?

Si el registro genera problemas, lo más probable es que no se disponga de buena conexión red o que el correo electrónico ya haya sido utilizado previamente; para solventarlo se puede utilizar otra dirección de correo electrónico o chequear que la conexión es correcta.

Para más dudas acerca del programa pueden consultar al correo electrónico jyusta@etsii.upv.es

Capítulo VI.2

Manual de programador (Valencià)

JULIO YUSTAS TALAMANTES



MANUAL DE PROGRAMADOR

CONTINGUT



1
ACOSTA DEL
PROGRAMA

2
ESTRUCTURA DEL
PROGRAMA

3
INTERFÍCIE D'USUARI

4
REQUERIMENTS
TÈCNICS

5
INSTALACIÓ

6
INICIAR APLICACIÓ

7
REGISTRE
"BLOCK DIAGRAM"

CONTENIDO



8

**IDENTIFICACIÓ
"BLOCK DIAGRAM"**

9

**MENÚ PRINCIPAL
"BLOCK DIAGRAM"**

10

**DADES EN TEMPS
REAL "BLOCK
DIAGRAM"**

11

**SIMULADOR
"BLOCK DIAGRAM"**

12

**HISTÒRIC DE DADES
"BLOCK DIAGRAM"**

13

**INTERRUPTOR
AUTOMÀTIC
"BLOCK DIAGRAM"**

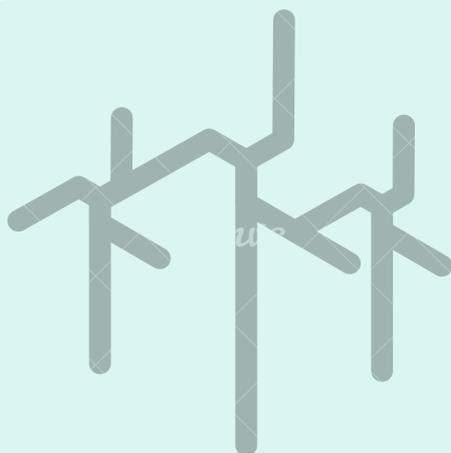
14

**CÓDIS D'ERROR I
FAQS**



ACOSTA DEL PROGRAMA

Aquest programa ha sigut desenvolupat com a part del Treball de Final de Grau de Julio Yustas Talamantes; aquest treball s'inclou dins de l'investigació per part de l'Universitat Politècnica de València per al projecte eòlic europeu "**PROMOTiON**" l'objectiu del qual es explotar tots els recursos de caràcter eòlic presents al mar del nord.



Aquest programa ha sigut desenvolupat per al seu ús com aplicació de monitorització de parcs eòlics a la mar.

La seua comesa principal és proporcionar a l'operari, una interfície clara y senzilla per al correcte acompliment del seu treball.

Al mateix temps, es postula com els fonaments des dels quals construir un programa molt més sofisticat i per al qual es proporciona aquest manual.



ACOSTA DEL PROGRAMA

Els autors d'aquesta aplicació han sigut Julio Yustas Talamantes baix la tutorització de la Doctora Soledad Inmaculada Bernal Pérez.

Aquest treball s'ha pogut desenvolupar gràcies a la col·laboració de l'Universitat Politècnica de València i de l'Institut Universitari d'Automàtica i Informàtica Industrial



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

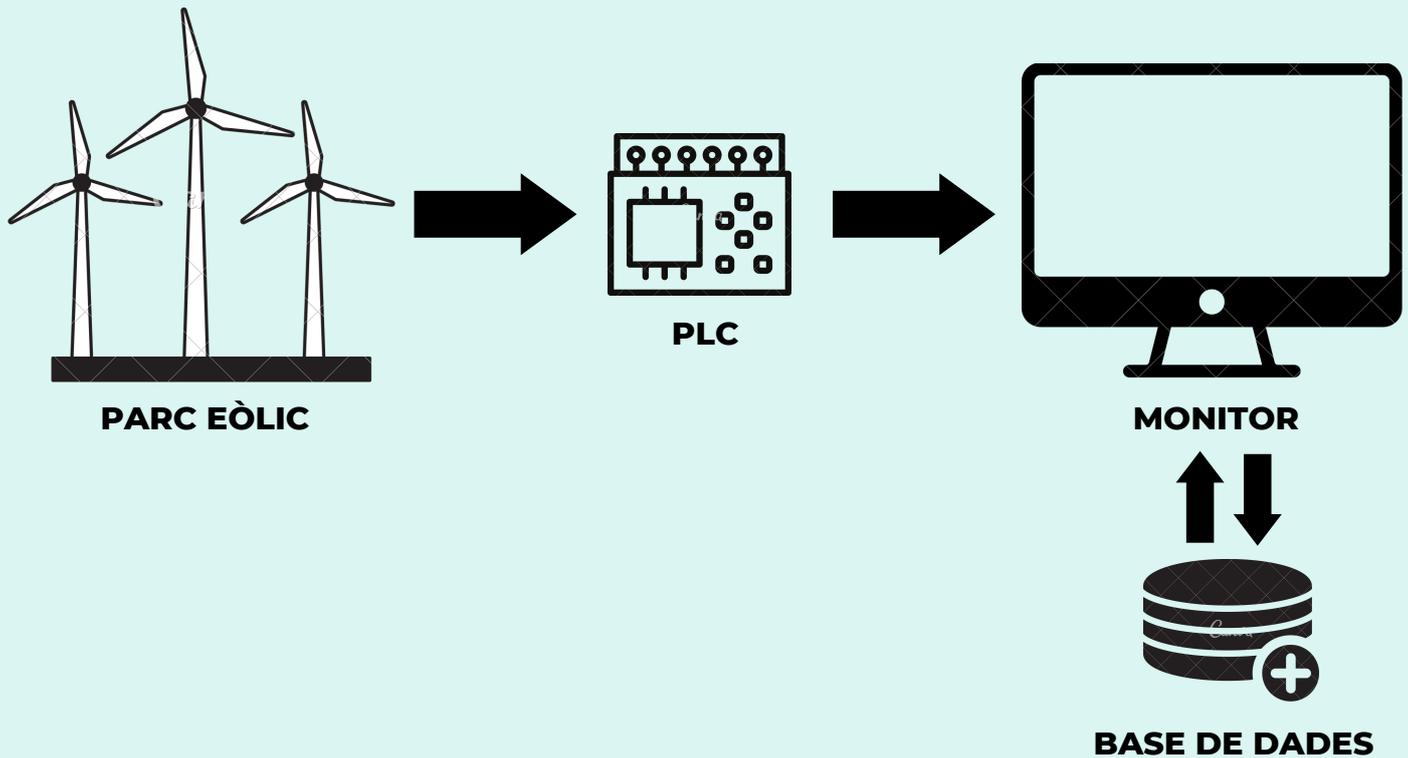


ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA



PROMOTioN
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS





ESTRUCTURA DEL PROGRAMA

Es tracta d'un sistema SCADA (Supervisió, Control i Adquisició de Dades).

L'objectiu principal es controlar un parc eòlic des d'altre lloc de manera remota, per això, necessitem tres elements clau: el parc eòlic, els PLCs i el sistema de monitorització.

Les connexions entre aquestes parts són una part fonamental per al correcte desenvolupament del present programa.

La unió entre el parc eòlic i els sensors es realitzarà mitjançant un cablejat.

Aquests sensors estaran connectats al seu torn a la xarxa, amb l'objectiu de permetre la comunicació mitjançant protocol UDP amb el simulador o programa SCADA,

És treball del programa SCADA, tractar aquestes dades i acomodar-los a un entorn atractiu per al treballador.

Al seu torn, emmagatzemarà aquestes dades en una base de dades per a permetre la seua possible consulta posterior.

INTERFÍCIE D'USUARI

El programa compta amb una interfície d'usuari senzilla i atractiva realitzada amb l'ajuda de LabVIEW i Photoshop.

La navegació és senzilla: per a avançar entre les diverses pestanyes només hem de triar la modalitat des del menú principal o "main menu".

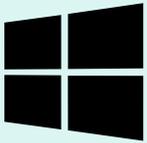
En el cas de voler retrocedir, es compta en tot moment amb un botó de "back" que ens envia a la pàgina anterior.

Per a finalitzar el programa simplement hem de tornar a la pestanya d'identificació i donar-li clic al botó de "stop" o en el seu defecte, utilitzar el clàssic botó de tancar pestanyes de windows



REQUERIMENTS TÈCNICS

DES D'UN SIMULADOR VIRTUAL:



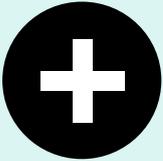
SISTEMA OPERATIU

Windows 7, Ubuntu 16.04 o macOS Sierra en 64 bits.



NAVEGADOR Y CONEXIÓ RED

Internet Explorer 11, Google Chrome 58 o Firefox 53



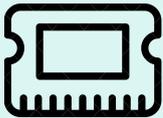
MATLAB

R2017b o superior



PROCESSADOR

Intel Core I5 o superior



MEMÒRIA RAM

Memòria RAM 8G o superior



TARGETA DE SO Y ALTAVEUS

REQUISITOS TÉCNICOS

DESDE UNA GRANJA EÓLICA REAL:



SISTEMA OPERATIU

Windows 7, Ubuntu 16.04 o macOS Sierra en 64 bits.

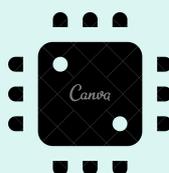


NAVEGADOR

Internet Explorer 11, Google Chrome 58 o Firefox 53

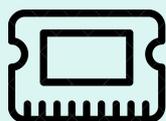


CONEXIÓ RED



PROCESSADOR

Intel Core I5 o superior



MEMÒRIA RAM

Memòria RAM 8G o superior



TARGETA DE SO I ALTAVEUS

INSTAL·LACIÓ

La instal·lació del present programa és prou senzilla, per a això només necessitem seguir una sèrie de passos:

La primera etapa consisteix a descarregar l'arxiu *promotion.exe* present en la web de PROMOTiON o en el repositori adjunt al peu d'aquesta mateixa pàgina.

Una vegada descarregat l'arxiu només queda donar-li doble clic per a executar-lo i instal·lar-ho.

Si es desitja utilitzar-ho al costat del simulador, haurem d'instal·lar en aqueixa mateixa computadora, o en una altra connectada a la mateixa xarxa WIFI, el programa MATLAB i executar els simuladors: *Banco_Pruebas_Breaker.mdl*, *Banco_Pruebas_Definitivo.mdl* i *simulator.mdl*

Si, per contra, es desitja utilitzar un prototip a escala o una granja eòlica real, haurem de realitzar una sincronització dels ports d'enviament dels PLCs i els ports d'entrada del programa d'acord amb la següent taula:

Variable a controlar	Port
Tensió	8811
Corrent	8822
Potència Activa	8833
Potència Reactiva	8844

Aquesta sincronització es realitzarà en el prototip a escala o el parc eòlic, és a dir, sempre sobre el PLC i no sobre el programa.

Control	Envío
Tensió	7700
Corrent	7711
Potència Activa	7722
Potència Reactiva	7733

La anterior configuració s'utilitzarà per a la funcionalitat de simulador; aquests ports són els d'enviament per part del simulador i/o PLC.

INICIALITZACIÓ

Per a garantir la correcta inicialització del programa, haurem de comprovar que totes dues parts: l'ordinador ("*host" del sistema SCADA) i els PLCs disposen de connexió a internet.

Al seu torn, s'haurà de comprovar, de manera anàloga a l'exposat en l'apartat anterior, que existeix una correcta disposició dels ports d'entrada i eixida.

Aquest últim pas és fonamental per a garantir l'èxit de la comunicació mitjançant el protocol UDP.

D'altra banda, l'usuari haurà de registrar-se amb els credencials proporcionats per l'empresa que seran únics i intransferibles.

En el cas de no disposar d'ells, es podrà donar d'alta en la base de dades d'operaris autoritzats mitjançant la pestanya de "sign-up".

Aquesta última pestanya s'explicarà amb més deteniment en els següents punts d'aquest mateix manual.

A continuació es torna a adjuntar la taula de connexions UDP:

Variable a controlar	Port
Tensió	8811
Corrent	8822
Potència Activa	8833
Potència Reactiva	8844

Finalment, per a començar a rebre dades del parc eòlic, l'única cosa que resta és prémer sobre l'interruptor d'inici de connexió, present en la pestanya de "*main menu*".

Per a la recepció de dades per part del programari s'haurà de modificar l'adreça IP dels protocols d'enviament en les pestanyes de "*Simulator*" i "*Breaker*".

REGISTRE "BLOCK DIAGRAM"

La pestanya "sign-up" permet que els usuaris puguin registrar-se dins de la base de dades de clients.

En les retallades adjuntades s'aprecia el codi de programació per a la finestra de registre d'usuari.

El primer component a destacar és un gran bucle "while" dins del qual s'engloben diversos elements:

- Un bucle de la mateixa tipologia l'objectiu de la qual és repetir el programa tantes vegades com siga necessari fins que s'emplenen els camps de registre.
- Un reinici de les variables a completar, això es realitzarà a l'inici de la finestra i en el cas que s'haja de repetir el registre per ser incorrecte.
- La comprovació que les diferents condicions perquè el registre s'haja realitzat de manera reeixida es compleixen: tots els camps completats i l'usuari i correu electrònic no utilitzat prèviament.
- A més, podem trobar un timing i sengles condicions per a la finalització de tots dos bucles.

- 1** Bucle "while".
- 2** Reinici de variables
- 3** Temporitzador
- 4** Clavilla registre
- 5** Nom d'usuari
- 6** Correu electrònic.
- 7** Escriptura en b. de dades
- 8** Condicionals
- 9** Missatge
- 10** Estructura d'enviament
- 11** Missatges d'error

REGISTRE "BLOCK DIAGRAM"

Continuant en l'exterior de l'esmentada estructura, trobem una estructura condicional que té la funció d'emmagatzemar els paràmetres en la base de dades "login.lvm", en el cas que s'hagen complit totes les condicions anteriors.

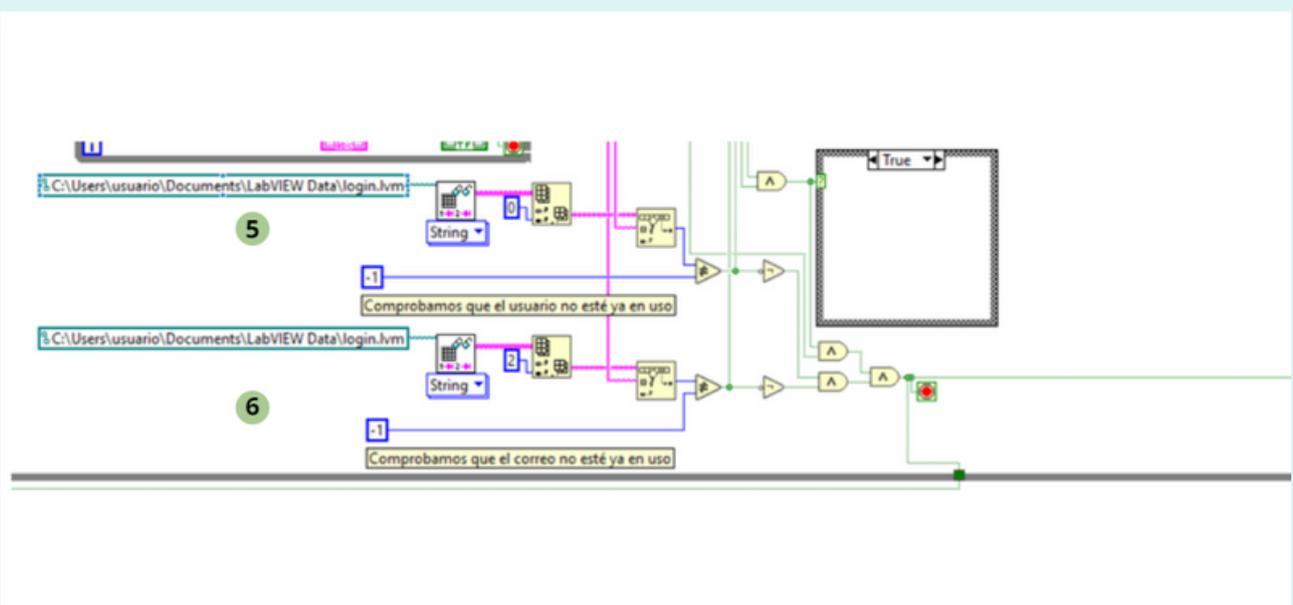
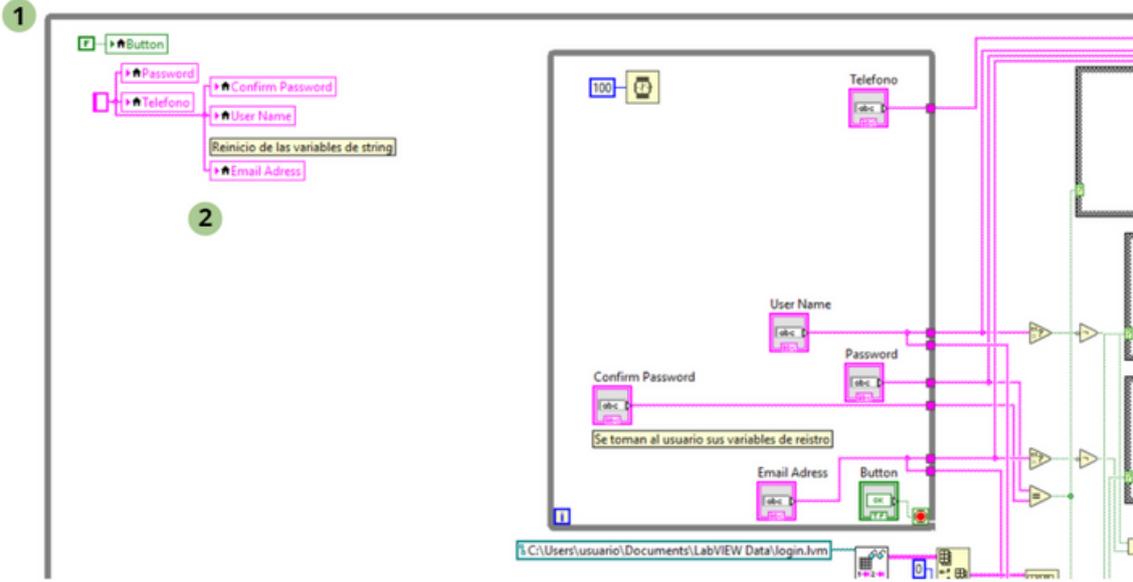
Finalment, podem observar un inici d'una sèrie de variables de tipus "string", text, utilitzades posteriorment en l'estructura que trobem just davall i la funció de la qual és enviar un correu de confirmació a l'usuari que s'acaba de registrar.

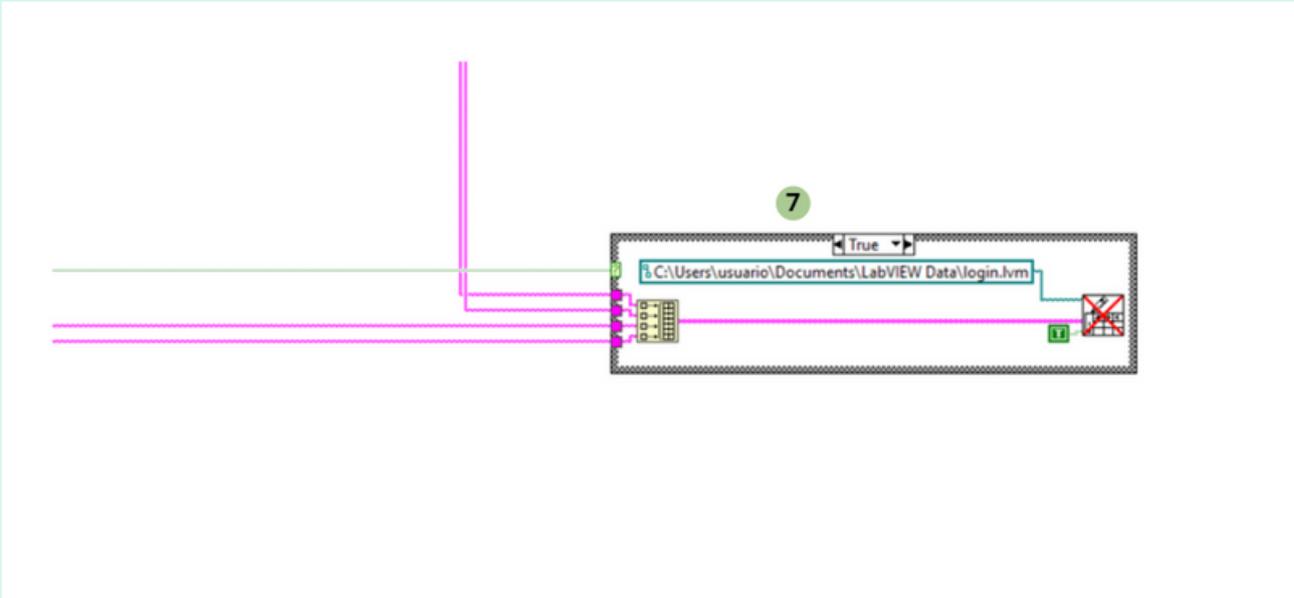
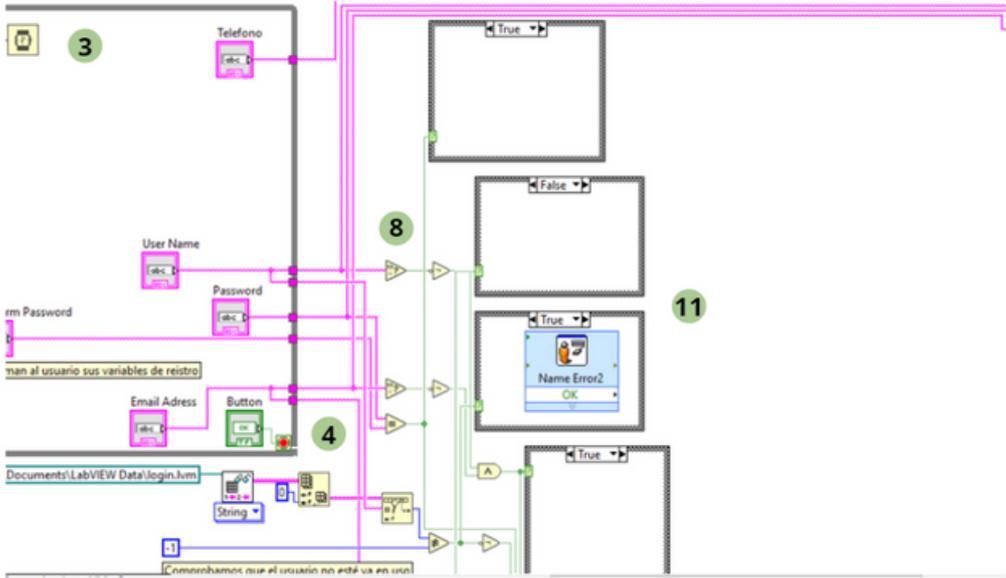
En cas que tot l'esmentat anteriorment s'haja dut a terme de manera reeixida, el programa generarà un missatge de confirmació a l'usuari.

Els principals elements a modificar per part d'un programador són: la substitució de la base de dades tipus .lvm per una altra mena de base de dades més sofisticada; la introducció d'un element de confirmació mitjançant una combinació aleatòria de números enviada en el correu de confirmació, etc.

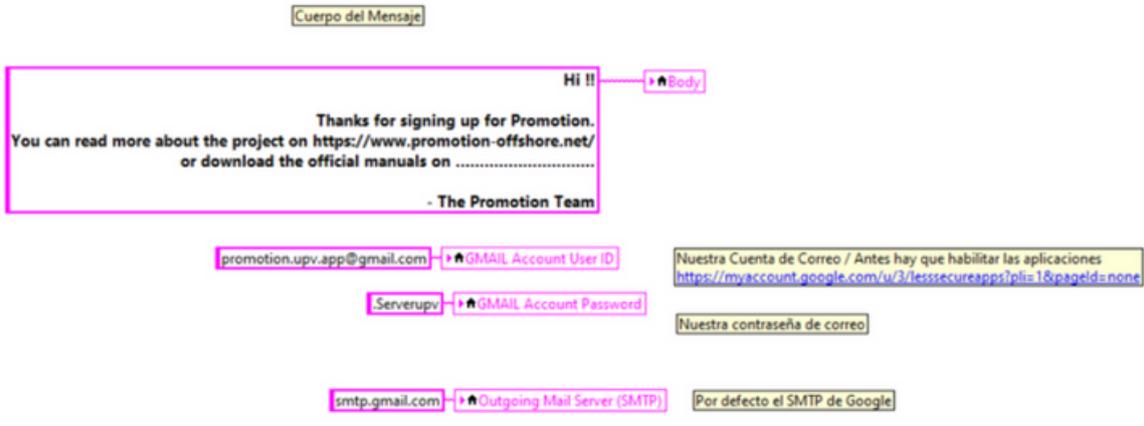
Al seu torn, es podrà modificar el contingut del missatge i el disseny d'aquest.

- 1** Bucle "while".
- 2** Reinici de variables
- 3** Temporitzador
- 4** Clavilla registre
- 5** Nom d'usuari
- 6** Correu electrònic.
- 7** Escriptura en b. de dades
- 8** Condicionals
- 9** Missatge
- 10** Estructura d'enviament
- 11** Missatges d'error

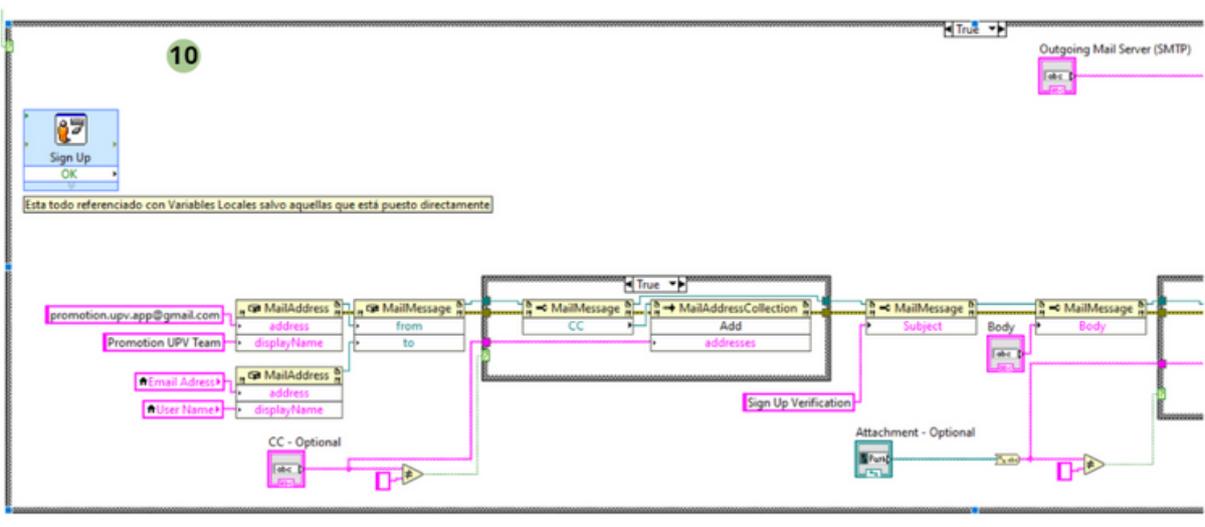


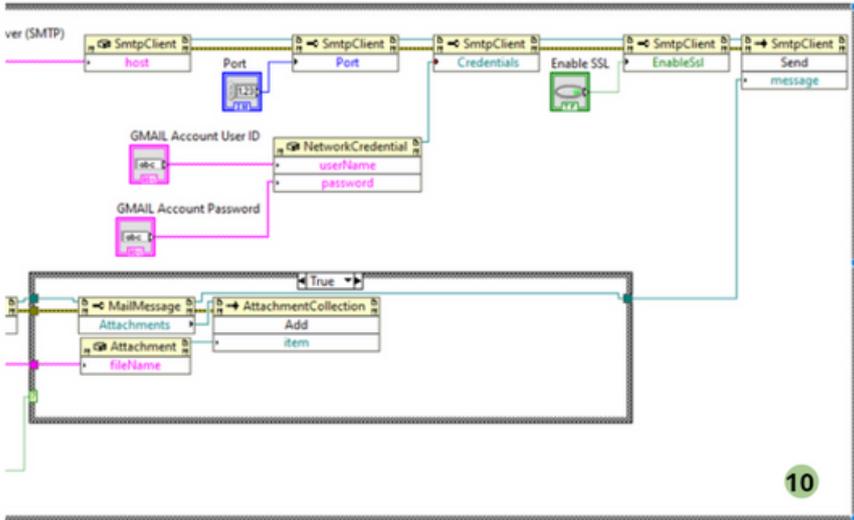


9



10





IDENTIFICACIÓ "BLOCK DIAGRAM"

Es presenta ara el diagrama de blocs o programació de l'entorn gràfic per a la identificació d'usuari.

El present codi se situa a l'interior d'un bucle "while" la funció del qual és repetir el programa, sempre que no es valide l'opció que finalitza el bucle de repetició, és a dir, botó de "stop" premut, la qual cosa generaria un missatge de tancament i finalitzaria el programa; alhora que reinicia les variables i els camps de farciment.

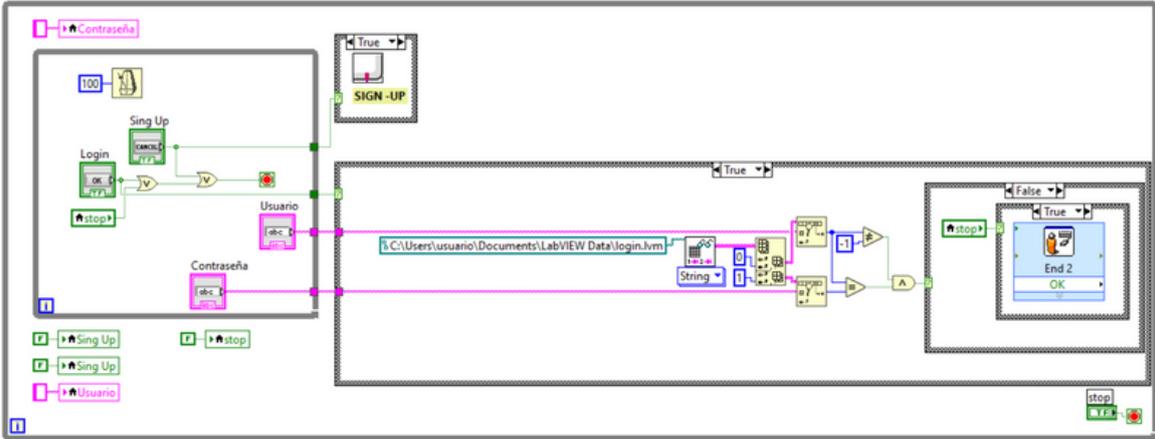
Dins d'aquest bucle es troba un altre bucle "while" amb les caselles per a introduir les credencials: nom d'usuari i contrasenya, i les clavilles ("login" i "sign-up"). Seran aquestes dues últimes les encarregades de permetre l'avanç fins als següents blocs.

Sengles blocs són condicionals: el primer d'ells executarà la pestanya de registre en el cas que la validació del bucle "while" anterior s'haja fet mitjançant el premut del botó "sign-up".

El segon condicional s'executa quan es prem el botó de "login"; en aquest moment el programa obri la base de dades "login.lvm" i es produeix una consulta línia a línia del nom d'usuari i contrasenya.

Depenent de si les dades introduïdes s'han trobat o no, el programa autoritzarà l'usuari passar a la pestanya de "main menu" o li emetrà un missatge d'error i li instarà a continuar provant, la qual cosa retornaria al client a l'estat inicial.

- 1** Bucle "*while".
- 2** Reinici de variables
- 3** Temporitzador
- 4** Clavilles
- 5** Campos de farcit
- 6** Estructura "*sign-*up"
- 7** Cerca
- 8** Accions



MENÚ PRINCIPAL "BLOCK DIAGRAM"

En aquest "block diagram" trobem dos elements clau que actuen de manera simultània i totalment independent:

Un primer bucle "while" que repeteix el codi des de l'inici del programa alhora que es reinicien les variables utilitzades en aquest; aquest bucle es repeteix constantment fins i tot si es valida alguna de les estructures condicionals que es localitzen dins d'aquest. L'única manera de finalitzar-ho és teclejant sobre el botó de "back".

Un altre bucle "while" dins d'una estructura condicional que s'habilita una vegada es connecta la clavilla de recepció de dades en temps real; dins d'aquest bucle tenim, repetit tantes vegades com dades tenim a rebre, el procés d'obertura, lectura

i tancament del protocol UDP, així com el tractat de la informació per a poder representar-la de manera correcta en la "front view".

En aquest diagrama de blocs no hi ha res a modificar llevat que el programador desitge incorporar una nova funcionalitat al programa o modificar el rebut de dades en temps real.

- 1** Reinici de variables
- 2** Bucle "while".
- 3** Temporitzador
- 4** Clavilles
- 5** Inici recepció dades
- 6** Estructura recepció

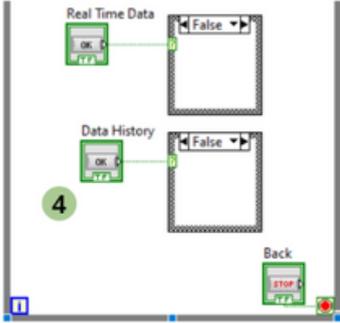
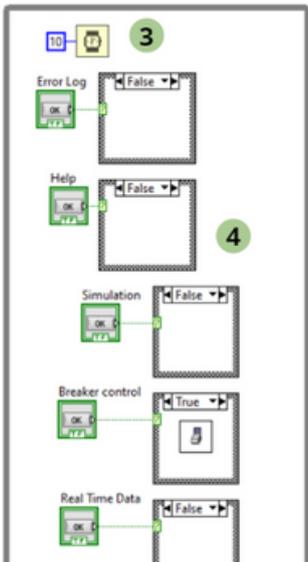
- 7** Reinici variables protocol UDP

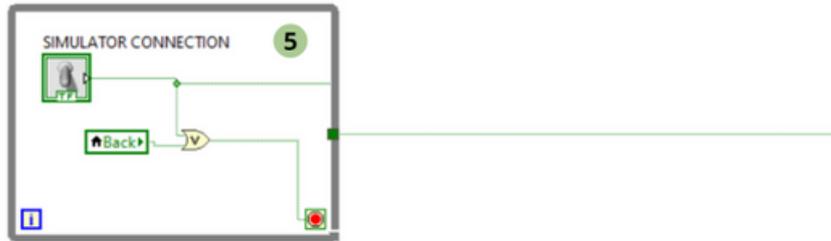
1

▶▶SIMULATOR CONNECTION

- ▶▶Help
- ▶▶Back
- ▶▶Simulation
- ▶▶Data History
- ▶▶Real Time Data
- ▶▶Error Log
- ▶▶Breaker control

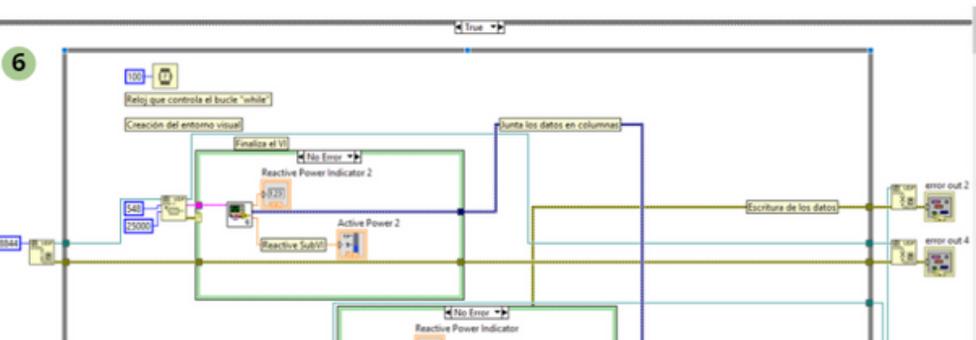
2





7

- Voltage Indicator
- Active Power
- Reactive Power
- Active Power
- Active Power Indicator
- Reactive Power Indicator



DADES EN TEMPS REAL

L'entorn de programació d'aquesta finestra compta amb un bucle "while" dins del qual està la clavilla de connexió amb el simulador.

Una vegada connectada, es validarà l'estructura de casos. Dins d'aquesta tenim un altre bucle "while" que executa el conjunt de blocs de recepció de dades de forma continuada fins que es prema el botó de reculada.

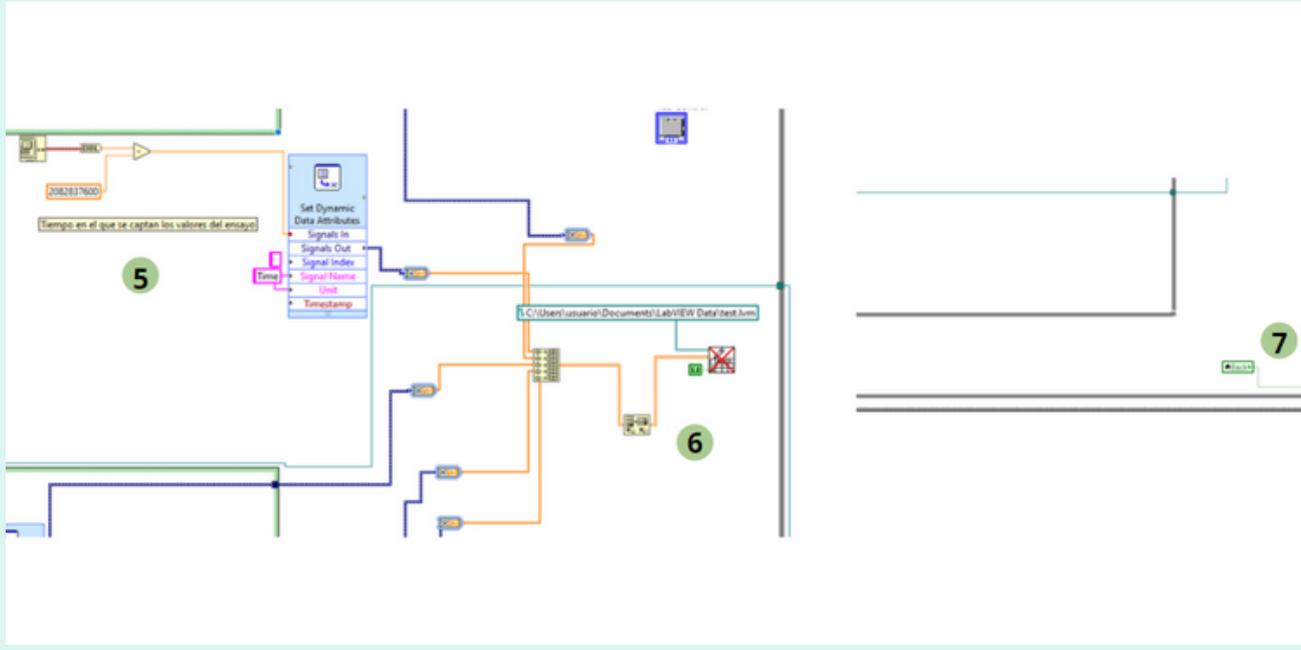
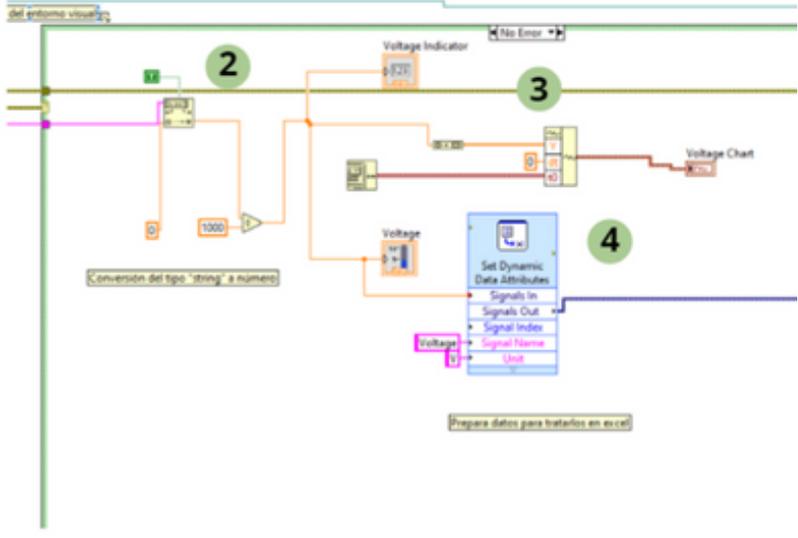
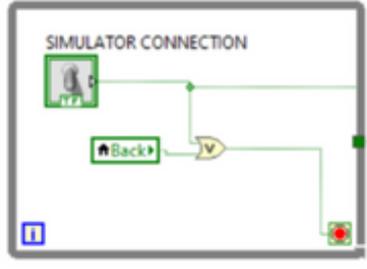
La recepció de dades consta d'un bloc d'obertura del protocol d'enviament UDP on s'assigna el port local a utilitzar,

un bloc de lectura que rep el senyal, una sèrie de blocs de tractament de senyal i un bloc de tancament del protocol. Com en la resta de les finestres tenim també uns indicadors d'errors que seran utilitzats per la pestanya d'errors.

Les modificacions podrien ser canviar el tipus de base de dades, canviar els ports utilitzats o el tipus de variables rebudes; així com la naturalesa dels indicadors utilitzats i la freqüència de recollida de dades.

- 1** Bucle inici
- 2** Recepció de dades
- 3** Indicadores
- 4** Preparació de dades
- 5** Temps
- 6** Escripura de dades
- 7** Botó de reculada

1 SIMULATOR CONNECTION



SIMULADOR "BLOCK DIAGRAM"

Aquest "block diagram" reuneix aspectes ja utilitzats en les pestanyes de "breaker" i de "real time".

Es tracta d'un gran bucle "while", dins del qual es realitzen dues tasques simultànies: l'enviament de dades i la recepció de dades via UDP. Per a això, se seguirà l'esquema típic de "obertura, enviament i tancament" o "obertura, recepció i tancament", segons el cas.

Tant en l'enviament com en la recepció es disposa d'un indicador d'error, utilitzat per la pàgina de "Error Log"; per part seua, la recepció continua amb un sistema de representació.

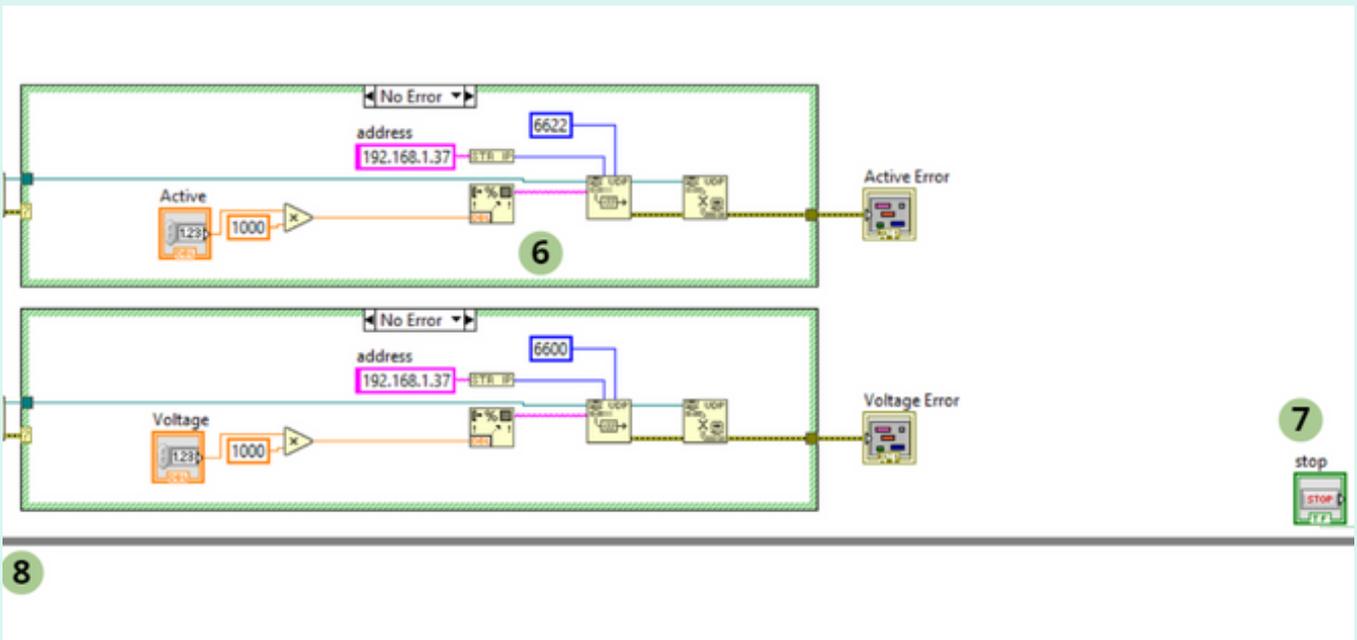
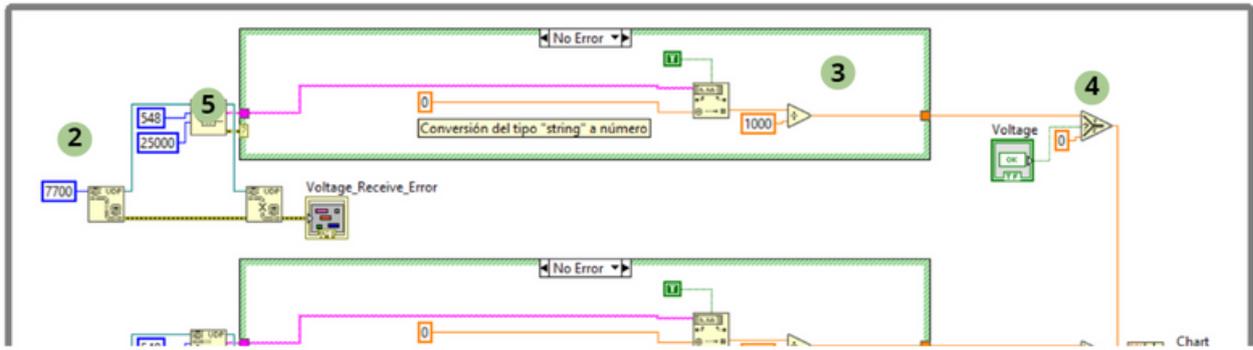
S'ha emprat un operador booleà per a oferir a l'usuari la possibilitat de triar si es ploteja o no la variable en qüestió, acompanyat d'una gràfica per a visualitzar els mateixos.

Es disposa, igual que en la resta de les pestanyes, d'una clavilla de reculada que envia a l'usuari al menú principal.

Les variables de recepció i enviament, així com els indicadors són totalment modificables a gust del programador.

- 1** Reinici de variables
- 2** Obertura UDP
- 3** Recepció de dades
- 4** Clavilles
- 5** Gràfica
- 6** Estructura d'enviament
- 7** Botó de "back"
- 8** Bucle "while"

1



HISTÒRIC DE DADES

Aquesta funcionalitat del programa aquesta composta per dues subfinestres: la primera d'elles permet seleccionar la data i la segona representa les dades.

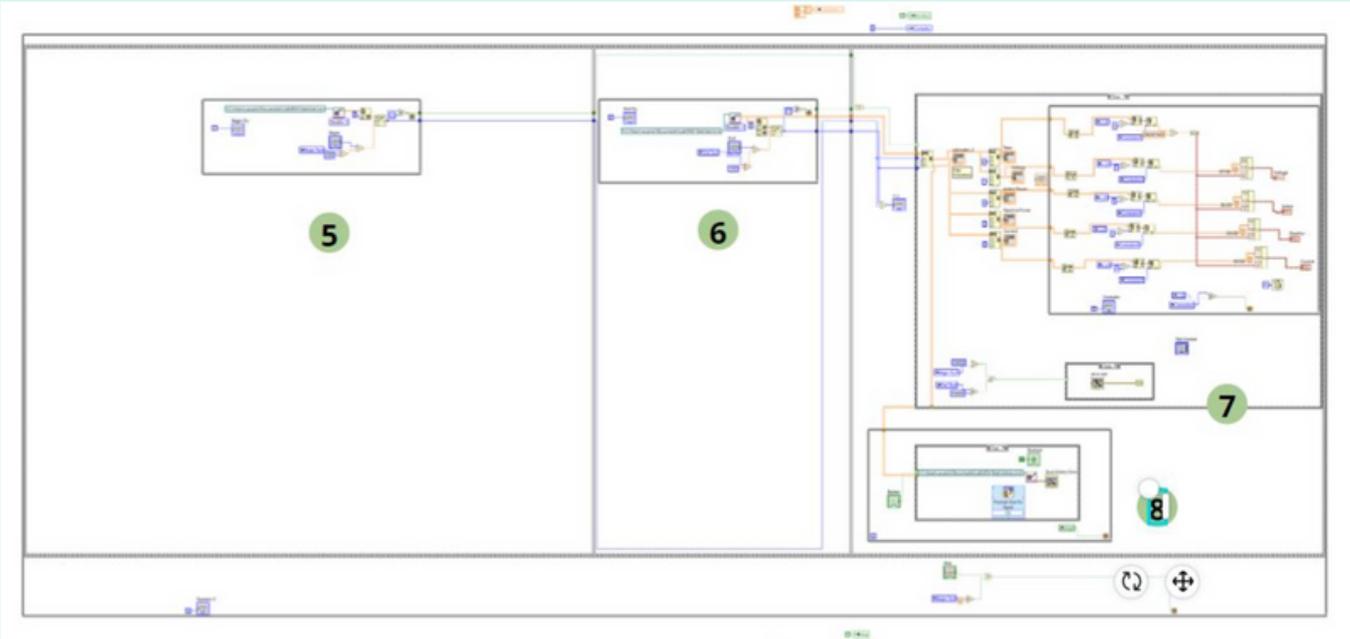
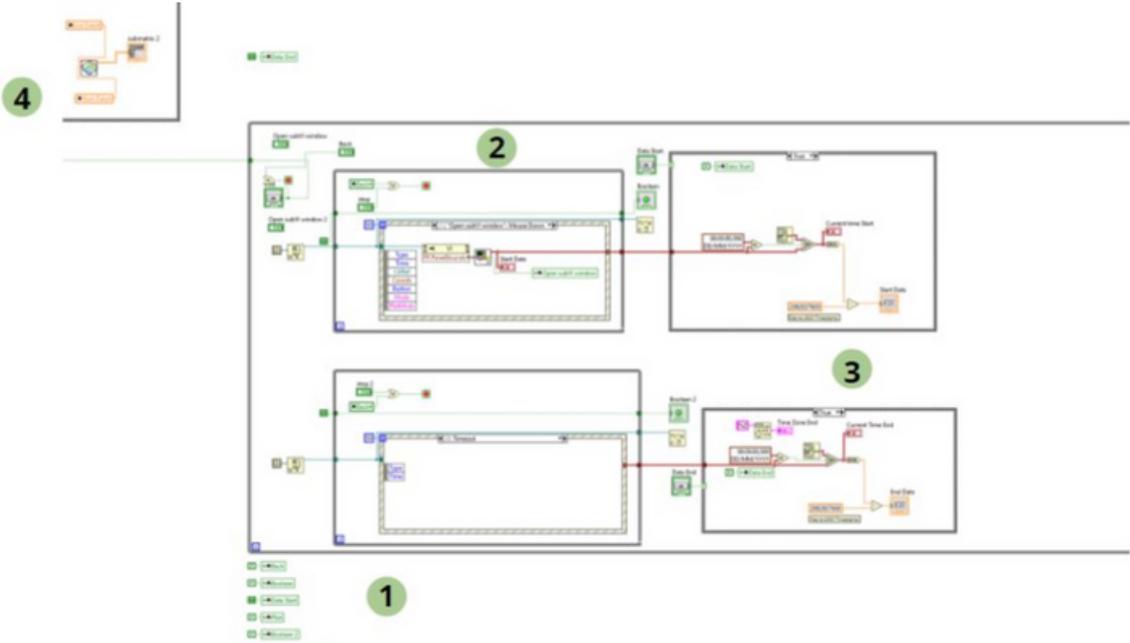
Per a la selecció de data es disposa d'un gran bucle "while" dins del qual trobem sengles blocs condicionals que s'executaran quan se seleccione la data i es confirme. El gran bucle "while" finalitzarà quan es complisquen aquests dos paràmetres i l'usuari preme sobre el botó de plotejar.

En aqueix moment el programa obrirà l'altra pestanya, aquesta segona finestra està constituïda per la següent seqüència: dos bucles "while",

on el programa busca les dades més pròximes i dos últims bucles "while", el primer format per blocs de tractament de matrius per a plotejar i el segon amb un condicional per al magatzematge de les dades plotejades. Tot això dins d'un gran bucle "while" que s'executarà fins que l'usuari indique el contrari.

Aquest és el codi més desenvolupat del programari i qualsevol modificació requereix de molta cura per a no desajustar el conjunt; es recomana no realitzar cap canvi fora de la possible modificació de la base de dades.

- 1** Reinicio de variables
- 2** Bucle "while"
- 3** Estructuras condicionales
- 4** Apertura segunda pestaña
- 5** Búsqueda de inicio
- 6** Búsqueda de fin
- 7** Ploteo
- 8** Salvado de datos



INTERRUPTOR AUTOMÀTIC

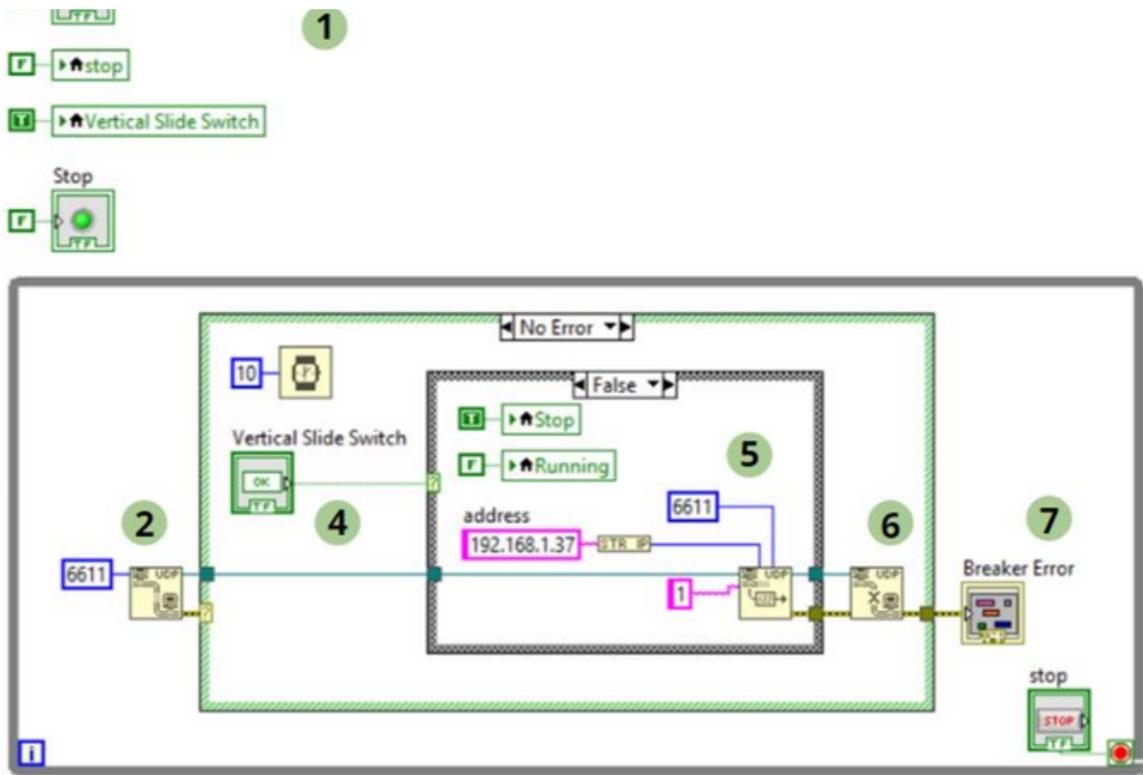
El codi de programació d'aquesta finestra compta amb un reinici de les variables (inclosos els dos LED) i un bucle "while" que, de manera anàloga als anteriors, es reproduïx fins que es valida la condició d'eixida.

Dins d'aquest bucle trobem un enviament via protocol UDP; primer obrim el protocol i assignem un port (en aquest cas el 6611).

Si aquest procés es realitza sense problemes, es validarà l'estructura condicional. Dins d'ella tenim l'interruptor, i segons la seua posició, s'obrirà l'una o l'altra part de l'estructura condicional, enviant un 1 si s'ha obert el circuit o un 0 si, per contra, està tancat.

Dins d'aquesta última estructura tenim l'enviament que es realitzarà després d'introduir el port de destí i l'adreça (la qual caldrà modificar de dada numèrica a "string").

- | | |
|---------------------------------|------------------------------|
| 1 Reinici de variables | 7 Error |
| 2 Obertura UDP | 8 Botó per retrocedir |
| 3 Bucle "while" | |
| 4 Interruptor | |
| 5 Estructura condicional | |
| 6 Tancament protocol UDP | |



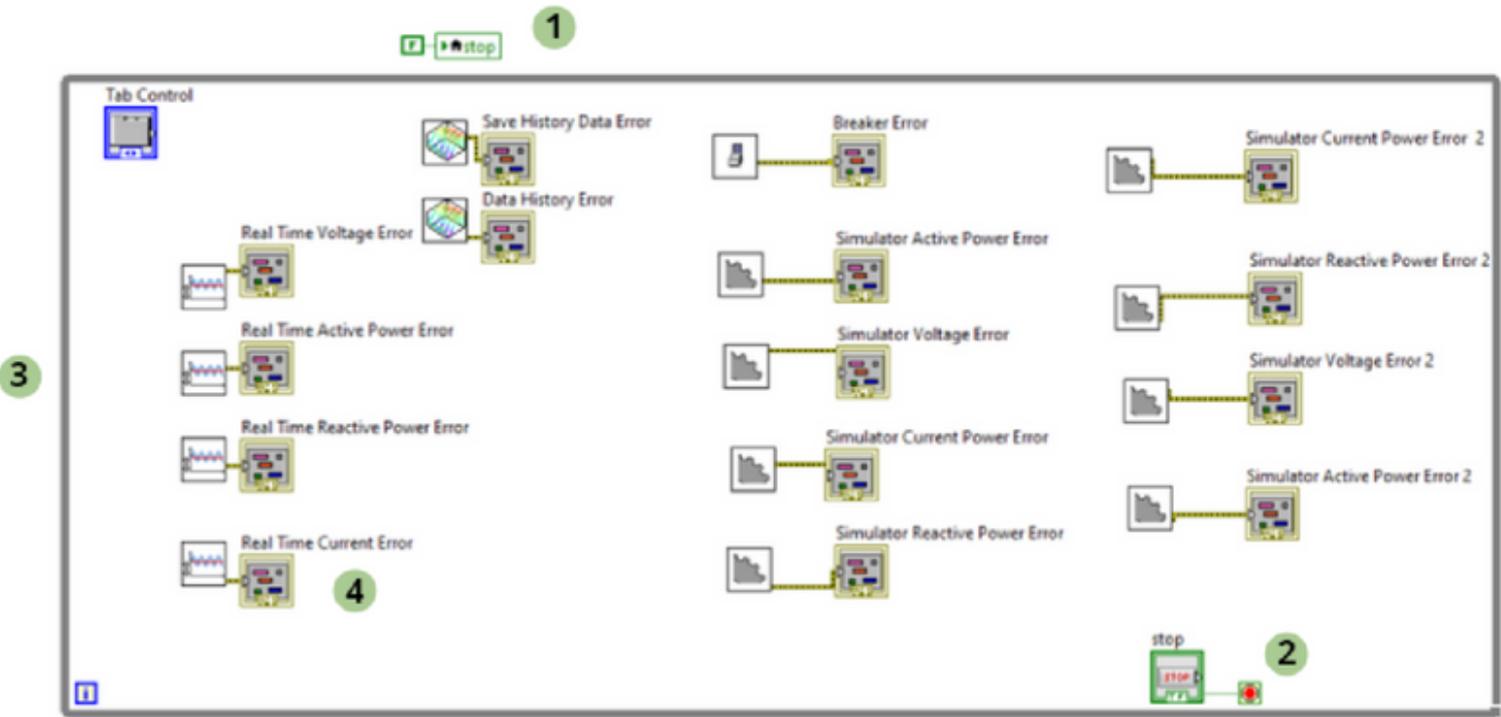
CODIS D'ERROR I FAQS

L'entorn de programació de la pestanya d'errors no és més que una espècie de "repositori" on s'agrupa tota la informació sobre errors present en la resta de les pestanyes, i que s'ha fet invisible per motius de comoditat.

Per tant, es disposarà d'un bucle "while" que es repetirà mentre no es complisca la condició d'eixida i, dins d'ell, es trobaran les anomenades als errors de les diferents finestres al costat d'un indicador gràfic per a cadascuna d'elles.

La única modificació possible és afegir més indicadors d'error, per a això, caldrà seguir amb l'estructura exposada: anomenada a la pestanya + indicador.

- 1** Reinici de variables
- 2** Botó de reculada
- 3** Bucle "while"
- 4** Anomenades als vaig veure



CODIS D'ERROR I FAQS

CÓDIGO

#54

DESCRIPCIÓN

La direcció de xarxa està mal formada. Assegure's que la direcció tinga un format vàlid. Per a TCP / IP, l'adreça pot ser un nom de màquina o una adreça IP amb el format xxx.xxx.xxx.xxx. Si aquest error ocorre en especificar un nom de màquina, assegure's que el nom de la màquina siga vàlid. Intente fer ping al nom de la màquina. Comprove que té un servidor DNS configurat correctament. Si està utilitzant la funció Connexió oberta TCP, assegure's que el valor del port remot o el nom del servei no siga 0.

#55

L'operació de la xarxa està en curs. Si rep aquest error mentre usa la funció UDP Write.

#56

L'operació de la xarxa va excedir el límit de temps especificat per l'usuari o del sistema.

#57

La connexió de xarxa està ocupada.

#58

La funció de xarxa no és compatible amb el sistema.

#59

Problemes amb l'escriptura UDP en la finestra actual.

#60

El port o la direcció de xarxa especificats estan actualment en ús. Seleccione un port o una direcció de xarxa disponibles.

#61

El sistema va provocar que es cancel·lara la connexió de xarxa.

#62

El servidor va rebutjar la connexió de xarxa. Per a TCP / IP

#1011

No es troben dades per a l'interval temporal seleccionat.

#44

Problema al intentar salvar un fitxer.

CODIS D'ERROR I FAQS

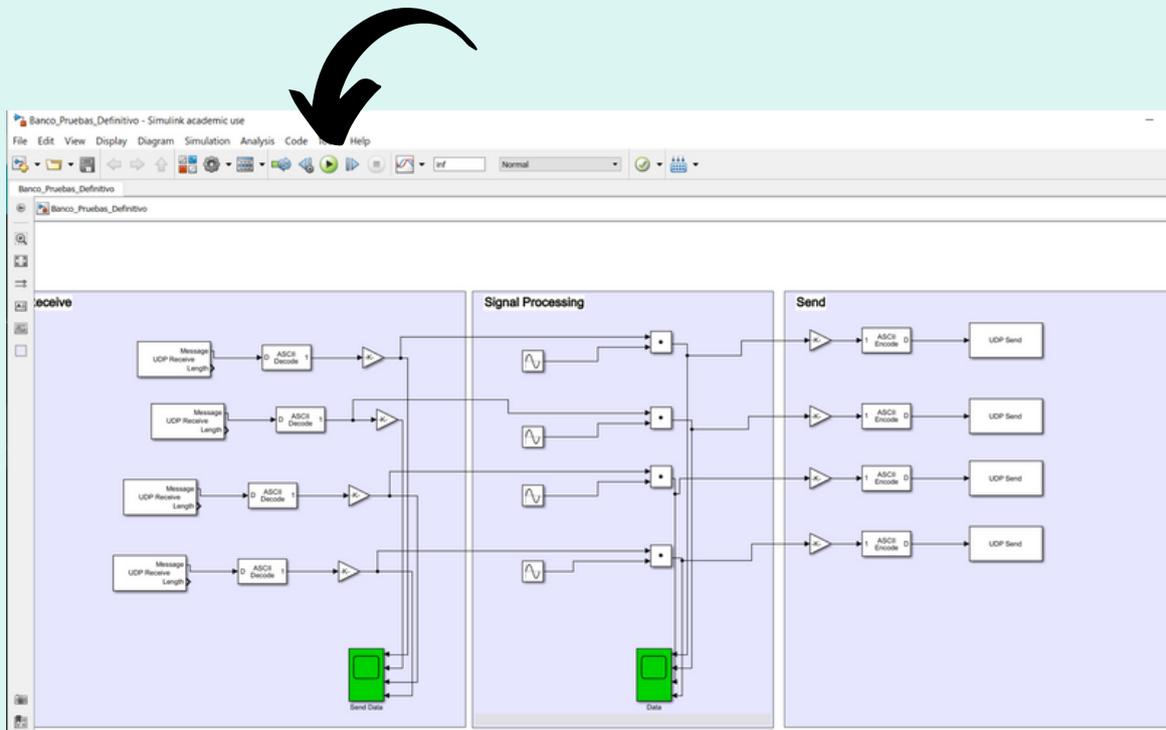
Com executar el codi en LabVIEW?

Per a executar un codi en LabVIEW l'usuari deurà polsar sobre el botó de "run" dins de la finestra de "log-in" ja que és aquesta la finestra principal del programa.

Per saber més sobre LabVIEW pots consultar: <https://www.ni.com/getting-started/labview-basics/esa/>

Com executar el codi en MATLAB?

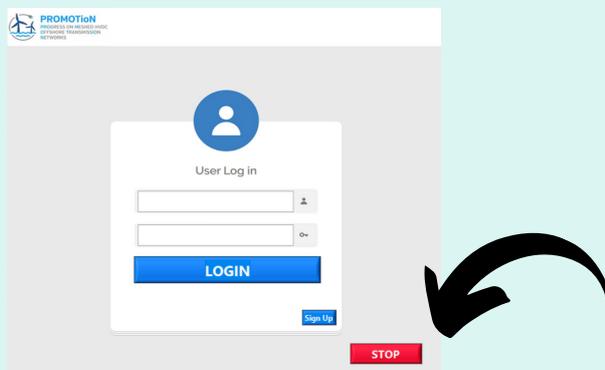
Per a executar un codi en MATLAB, més específicament en Simulink. L'usuari deurà polsar sobre el botó de "run" com es pot comprovar en la següent figura.



CODIS D'ERROR I FAQS

Com finalitzar el codi en LabVIEW?

El programa està configurat per a poder concloure sense la necessitat de pulsar el botó de "abort". Para això, l'usuari deurà retrocedir fins a la finestra de "log-in" i pulsar sobre el botó de finalització del programa.



Com passar el temps de UNIX Timestamp a la data real?

Per a realitzar aquesta conversió es recomana utilitzar el convertidors online, per exemple: <https://www.unixtimestamp.com/index.php>

Què fer si tinc problemes durant el registre?

Si el registre genera problemes, el més probable és que no es disposi de una bona connexió red o que el correu electrònic ja haja sigut utilitzat previament; per a solucionar-ho es pot utilitzar altra direcció de correu electrònic o comprovar que la connexió és correcta.

Para més dubtes acosta del programa poden consultar al correu electrònic juyusta@etsii.upv.es

Capítulo VI.3

Programmer's Manual (English)

JULIO YUSTAS TALAMANTES



PROGRAMMER'S MANUAL

CONTENT



1
ABOUT THE PROGRAM

2
SOFTWARE STRUCTURE

3
USER INTERFACE

4
**TECHNICAL
REQUIREMENTS**

5
INSTALLATION

6
START

7
**REGISTER
"BLOCK DIAGRAM"**

CONTENT



8
IDENTIFICATION
"BLOCK DIAGRAM"

9
MAIN MENU
"BLOCK DIAGRAM"

10
REAL-TIME DATA
"BLOCK DIAGRAM"

11
SIMULATOR
"BLOCK DIAGRAM"

12
DATA HISTORY
"BLOCK DIAGRAM"

13
AUTOMATIC SWITCH
"BLOCK DIAGRAM"

14
ERROR CODES & FAQs



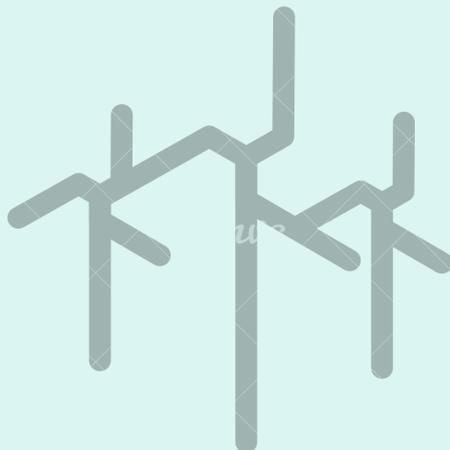
ABOUT THE SOFTWARE

This program has been developed as part of the Final Degree Work of Julio Yustas Talamantes; this work is included as part of the research of the Universitat Politècnica de València for the European wind project "PROMOTioN" whose main objective is to exploit all the wind resources of the North Sea.

This program has been developed for use as an offshore wind farm monitoring application.

Its main purpose is to provide the operator with a clear and simple interface for the correct performance of his work.

At the same time, it is postulated as the foundation from which to build a much more sophisticated program and for which the "Programmer's Manual" is provided





ABOUT THE SOFTWARE

The authors of this software have been Julio Yustas Talamantes under the tutorship of Dr. Soledad Inmaculada Bernal Pérez (Industrial Engineering Doctor).

This work has been developed thanks to the collaboration of the Universitat Politècnica de València and the Instituto Universitario de Automática e Informática Industrial (ai2).



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

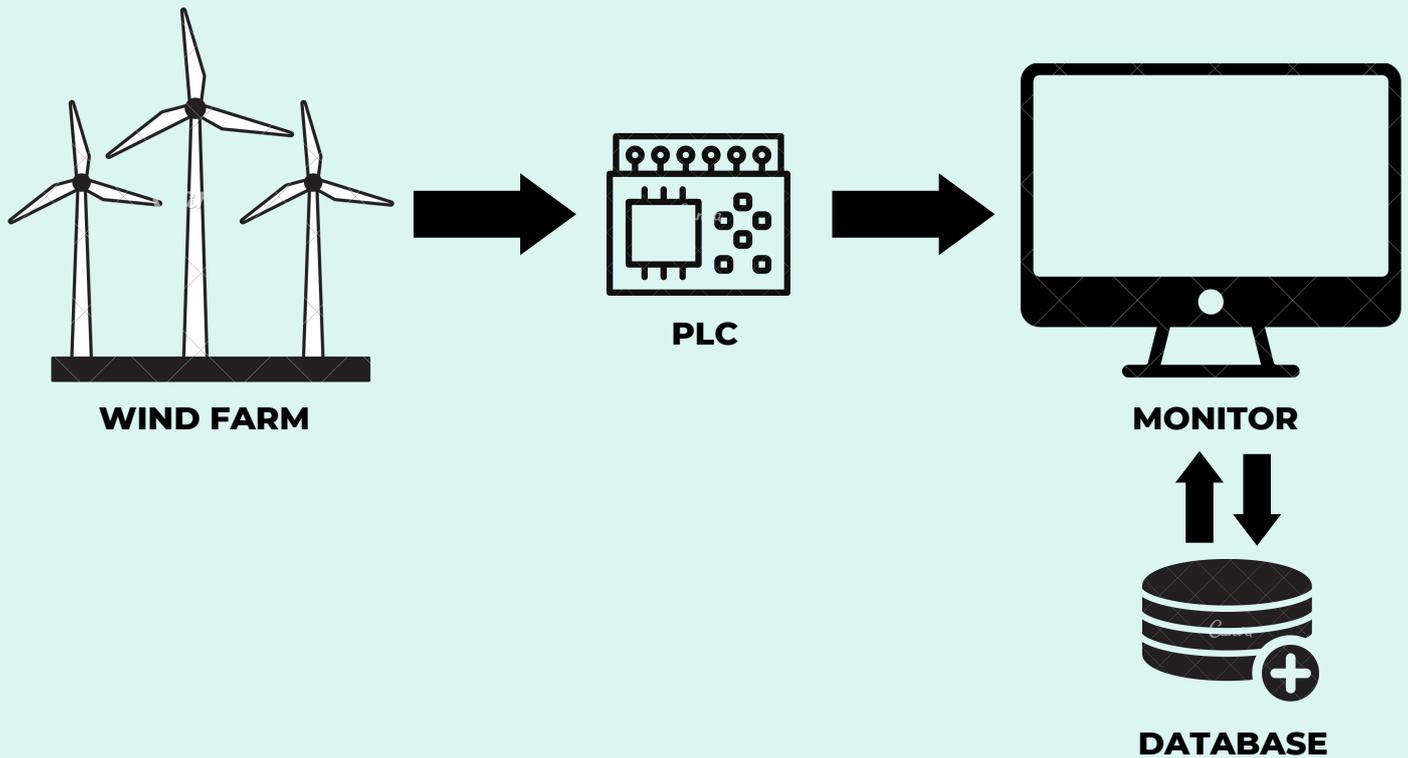


ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA



PROMOTiON
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS





SOFTWARE STRUCTURE

It is a SCADA (Supervisory Control and Data Acquisition) system.

The main objective is to control a wind farm from another location remotely. For this purpose we will need three key elements: the wind farm, the PLCs and the monitoring system.

The connections between these parts are a fundamental for the correct performance of this program.

The connection between the wind farm and the sensors will be made by a cable.

These sensors will, in turn, be connected to the network, in order to allow communication via UDP protocol with the simulator or SCADA program,

Is the job of the SCADA software to process this data and accommodate it in an attractive way to the worker.

In turn, it will store these data in a database to allow for possible later consultation.

USER INTERFACE

The program has a simple and attractive user interface made with the help of LabVIEW and Photoshop.

Navigation is simple: to advance between the various tabs we only have to choose the functionality from the main menu.

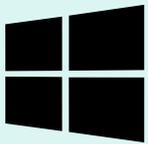
In the case of wanting to go back, there is always a "back" button that sends us to the previous page.

To end the program we simply have to return to the identification tab and click on the "stop" button or use the classic windows button to close tabs.



TECHNICAL REQUIREMENTS

FROM A VIRTUAL SIMULATOR:



OPERATING SYSTEM

Windows 7, Ubuntu 16.04 or macOS Sierra 64 bits.



BROWSER AND NETWORK CONNECTION

Internet Explorer 11, Google Chrome 58 or Firefox 53



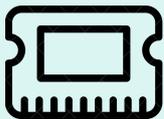
MATLAB

R2017b or superior



PROCESSOR

Intel Core I5 or superior



RAM MEMORY

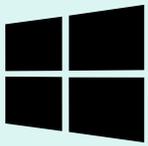
RAM memory 8G or superior



SOUND CARD AND SPEAKERS

TECHNICAL REQUIREMENTS

FROM A REAL WIND FARM:



OPERATING SYSTEM

Windows 7, Ubuntu 16.04 or macOS Sierra 64 bits.



BROWSER

Internet Explorer 11, Google Chrome 58 or Firefox 53

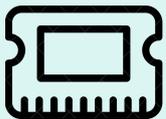


NETWORK CONNECTION



PROCESSOR

Intel Core I5 or superior



RAM MEMORY

RAM memory 8G or superior



SOUND CARD AND SPEAKERS

INSTALLATION

The installation of this program is quite simple, for it, we only need to follow this steps:

The first step is to download the *promotion.exe* file from the PROMOTioN website or from the repository at the bottom of this page.

Once the file has been downloaded, all that remains is to double-click on it to run and install it.

If we want to use it with the simulator, we should install the MATLAB program on the same computer, or on another, connected to the same WIFI network, and run the following simulators:

Banco_Pruebas_Breaker.mdl ,
Banco_Pruebas_Definitivo.mdl
 and *simulator.mdl*

If it is necessary to use a scale prototype or a real wind farm, we must carry out a synchronization of the PLC sending ports and the program entry ports in accordance with the following table:

Control Variable	Port
Voltage	8811
Current	8822
Active Power	8833
Reactive Power	8844

This synchronization will be done on the scale prototype or the wind farm, that is, always on the PLC and not on the program.

Control	Sending
Voltage	7700
Current	7711
Active Power	7722
Reactive Power	7733

The previous configuration will be used for the simulator functionality.

INITIALIZATION

To guarantee the correct initialization of the program, we must check that both parts: the computer ("host" of the SCADA system) and the PLCs have correct Internet connection.

At the same time, it will be necessary to check, in a similar way to what has been explained in the previous section, that there is a correct disposition of the input and output ports.

This last step is fundamental to guarantee the communication success of the UDP protocol.

On the other hand, the user will have to register himself with the credentials provided by the company which will be unique and non-transferable.

In the case of not having them, it will be possible to register in the database of authorized operators through the "sign-up" tab.

This last page will be explained in more detail in the following points of this manual.

Control Variable	Port
Voltage	8811
Current	8822
Active Power	8833
Reactive Power	8844

Finally, to start receiving data from the wind farm, the only thing left to do is clicking on the connection start switch, in the "main menu" tab.

To receive data from the software, the IP address of the sending protocols must be changed in the "Simulator" and "Breaker" tabs.

REGISTER "BLOCK DIAGRAM"

The "sign-up" tab enables user to register in the client database.

On the attached pictures, it is possible to notice the programming code for the user sign-up window.

The first component to highlight is a big while loop, which encapsulates various elements within:

- A loop of the same typology, whose objective is repeating the program so many times as necessary, until the registration boxes are filled in.
- A resetting of the variables to complete, this will take place at the beginning of the window and in the case of having to repeat the registration for being incorrect.
- The verification that the different conditions for the register to be made successfully are fulfilled: every box has been filled in and the user and e-mail have not been used previously.
- In addition, a timer and two conditions for the end of both loops can be found.

- 1** While loop
- 2** Variables' restart
- 3** Timer
- 4** Registration key
- 5** Username
- 6** E-mail
- 7** Writing in database
- 8** Conditionals
- 9** Message
- 10** Sending structure
- 11** Error messages

REGISTER "BLOCK DIAGRAM"

Continuing on the exterior of the mentioned structure, there is a conditional structure which has the function to store the parameters in the database *"login.lvm"*, in the case that all previous conditions have been met.

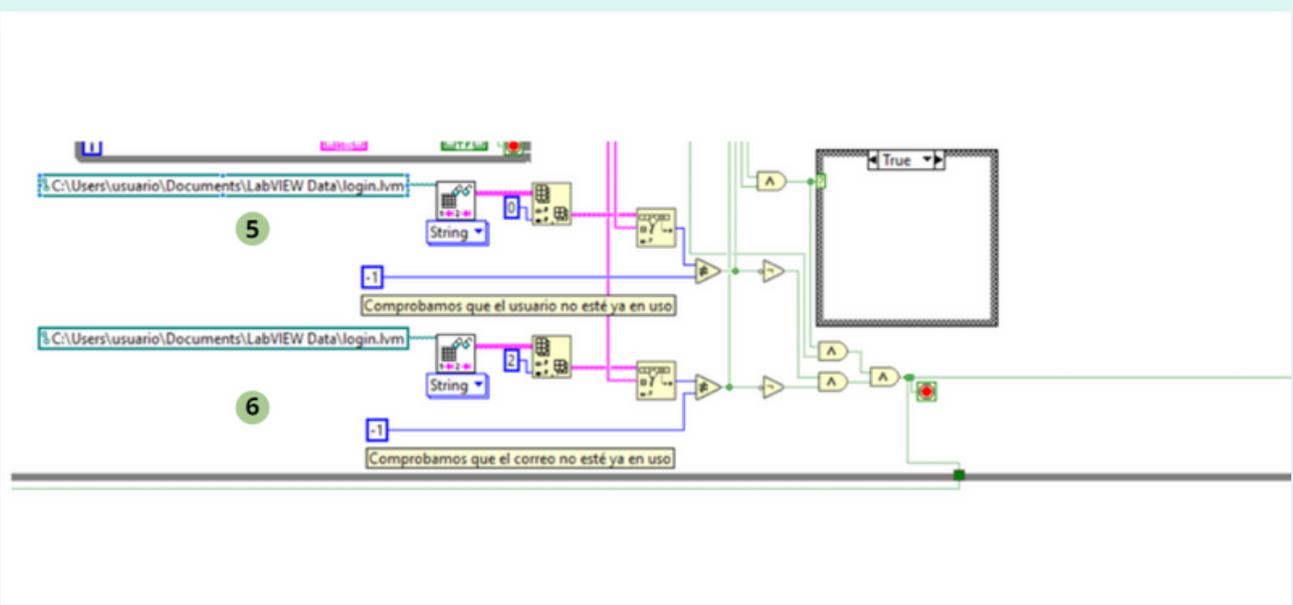
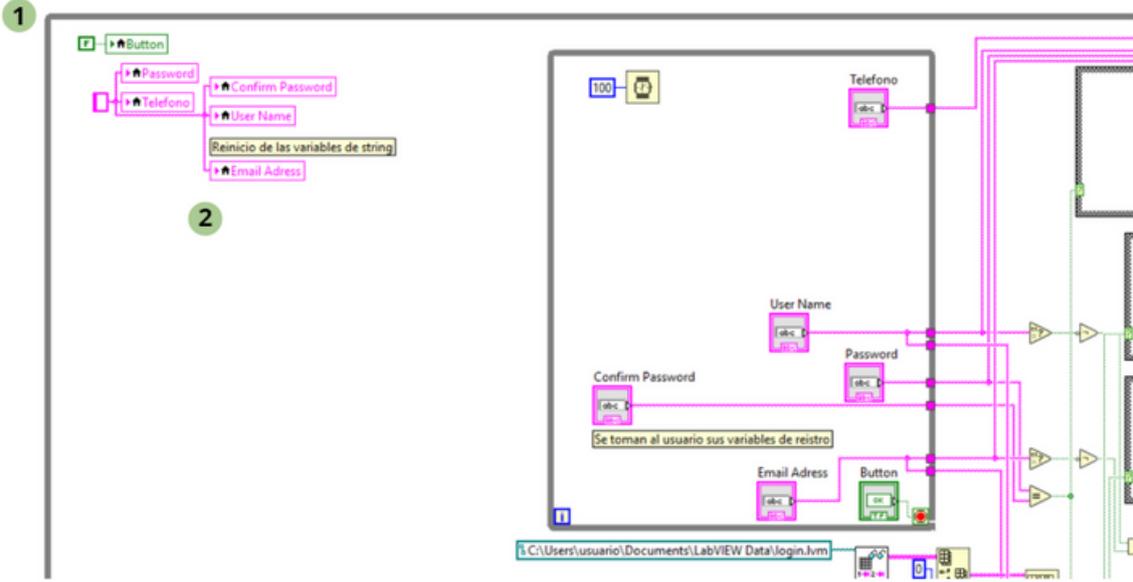
Last of all, the beginning of a series of variables type "string", text, can be observed. They are used later in the structure found just below, and their task is sending a confirmation e-mail to the user that has recently signed up.

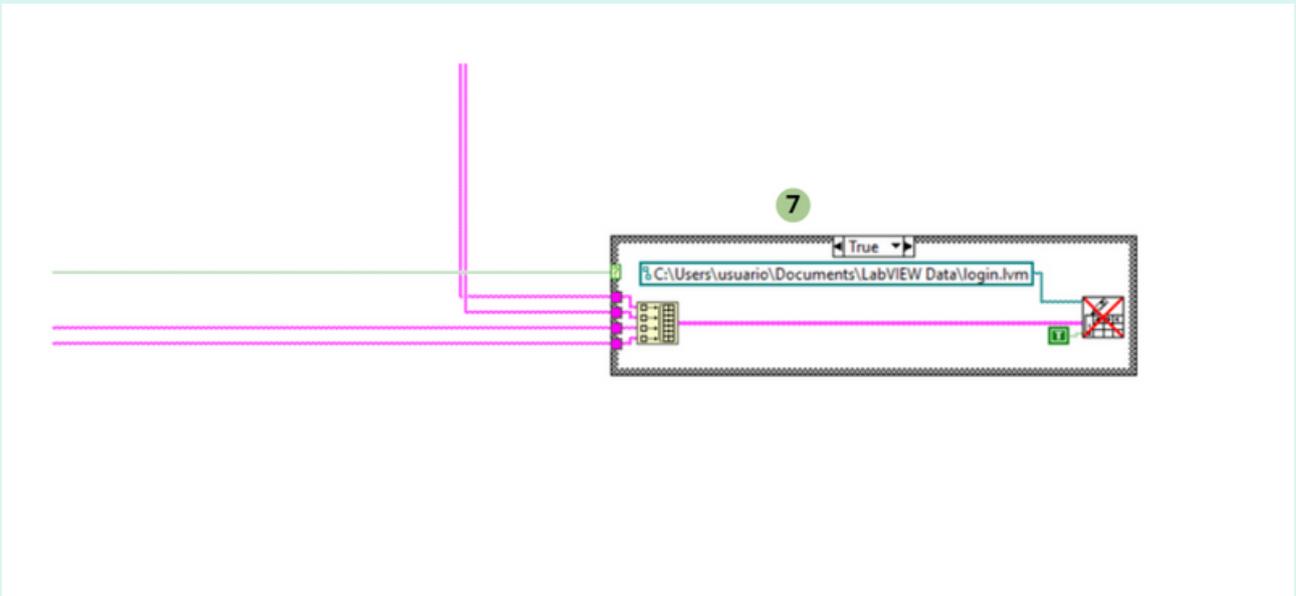
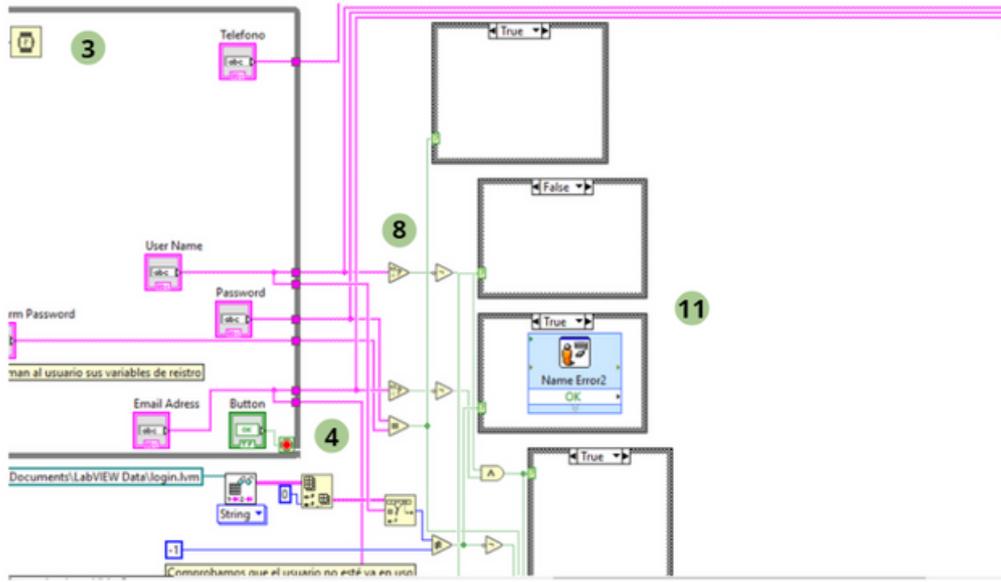
In the case that the abovementioned has been conducted successfully, the program will generate a confirmation message for the user.

The main items to modify by the programmer are: the replacement of the database type .lvm by a more sophisticated type of database; the introduction of a confirmation element through a random combination of numbers sent in the confirmation e-mail, etc.

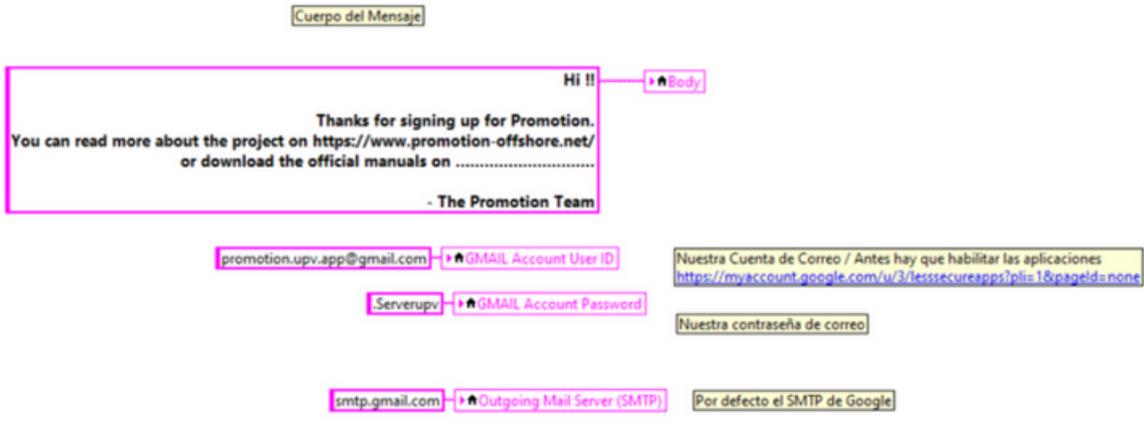
Likewise, the content of the message and its design can be edited.

- 1** While loop
- 2** Variable restart
- 3** Timer
- 4** Registration key
- 5** Username
- 6** E-mail
- 7** Writing in database
- 8** Conditionals
- 9** Message
- 10** Sending structure
- 11** Error messages

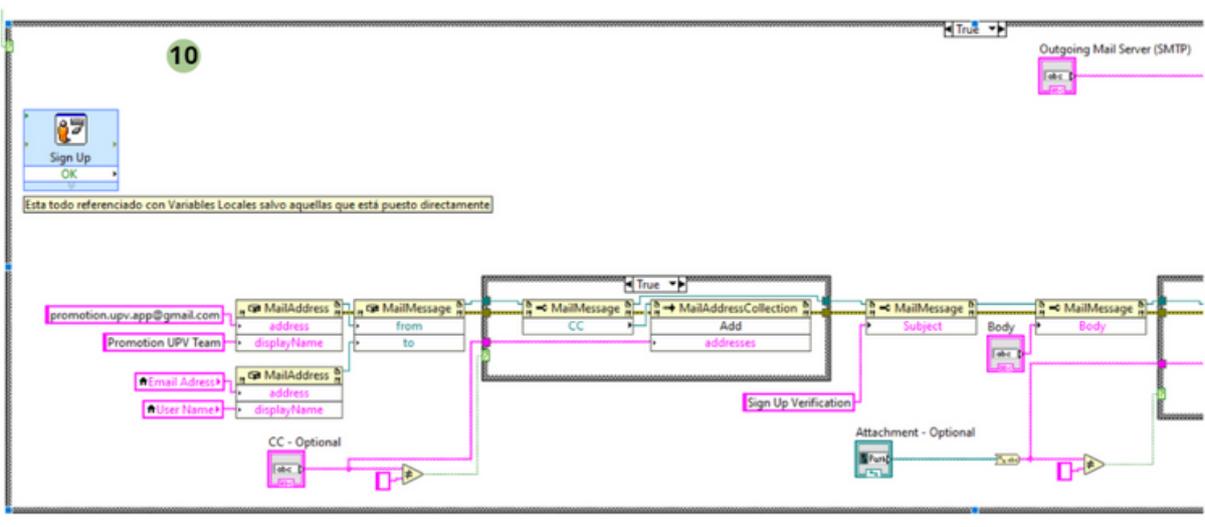


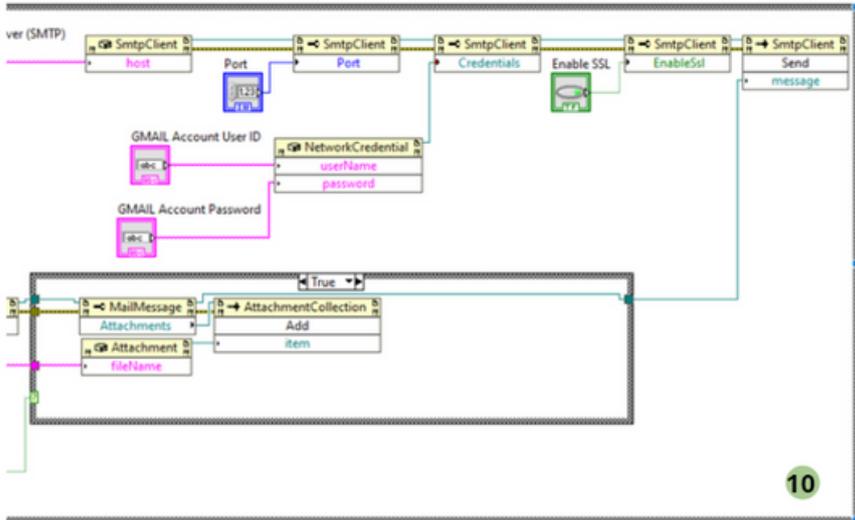


9



10





IDENTIFICATION "BLOCK DIAGRAM"

Now the block diagram or programming of the graphic environment for user identification is presented.

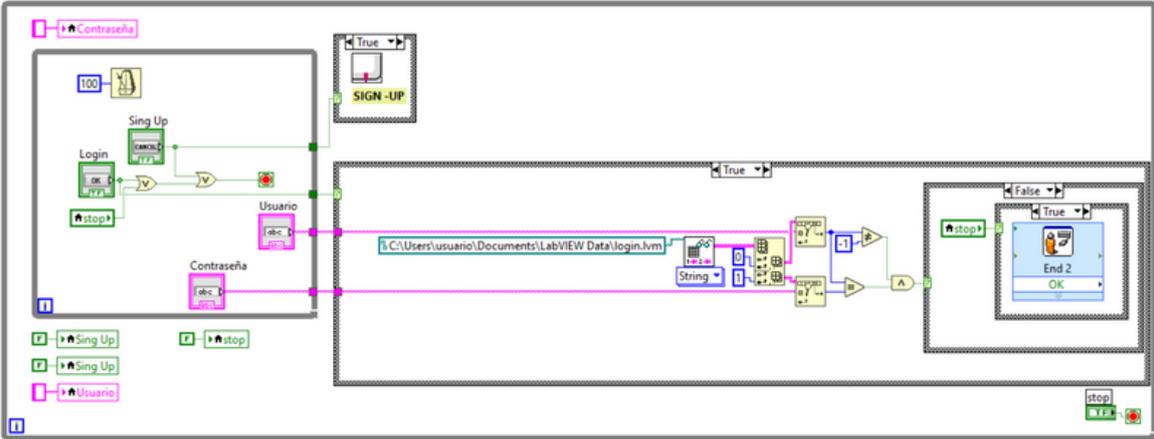
The current code is located within a while loop whose function is repeating the program, provided that the option ending the repetition loop is not validated. That is, *stop* button pressed, which will produce a closed message and end the program, as well as restarting the variables and filling out fields.

Inside this loop, there is another while loop to find, with the boxes to enter the credentials: username and password and the keys (*login* and *sign-up*). These two last buttons are in charge to allow the advance to the next blocks.

Both blocks are conditional: the first of them will execute the register tab in the case that the validation of the while loop has been carried out pressing the "*sign-up*" button.

The second conditional is executed when the "*login*" button is pressed; in this moment the program opens the database "*login.lvm*" and a line-to-line consultation about the username and password takes place. Depending on whether the entered data have been found or not, the program will authorize the user to go to the "*main menu*" page or will deliver an error message and urge him/her to keep trying. This will return the user to the initial state.

- | | |
|------------------------------|------------------|
| 1 While loop | 7 Search |
| 2 Variable restart | 8 Actions |
| 3 Timer | |
| 4 Keys | |
| 5 Filling out fields | |
| 6 "Sign-up" structure | |



MAIN MENU "BLOCK DIAGRAM"

In this "block diagram" there are two key elements acting simultaneously and independently:

A first while loop which repeats the code since the beginning of the program, as well as the employed variables are reseted. This loop repeats itself constantly, even if one of the conditional structures located inside is validated. The only way to finish it is clicking on the "back" button.

Another while loop inside a conditional structure that is enabled once the real-time data reception plug is connected. Inside this loop there is also, repeated as many times as data there are to receive, the opening, reading and closing

procedures of the UDP protocol, as well as the handling of information to be able to represent it correctly on the "front view".

In this block diagram there is not anything to modify, unless the programmer wishes to bring in a new functionality to the program or altering the real-time data reception.

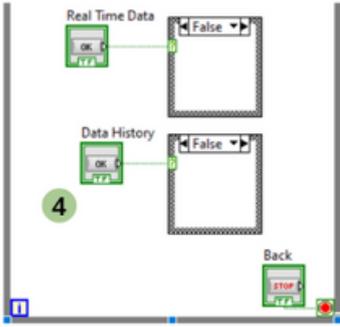
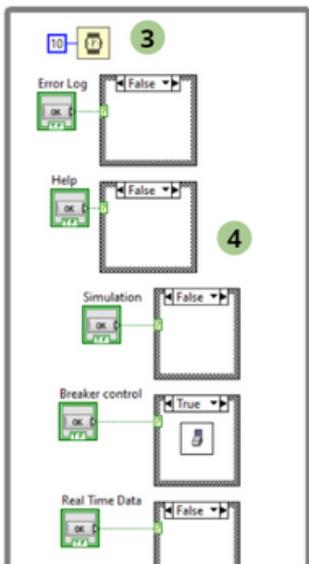
- 1** Variable restart
- 2** While loop
- 3** Timer
- 4** Keys
- 5** Data reception beginning
- 6** Reception structure
- 7** UDP protocol variables restart

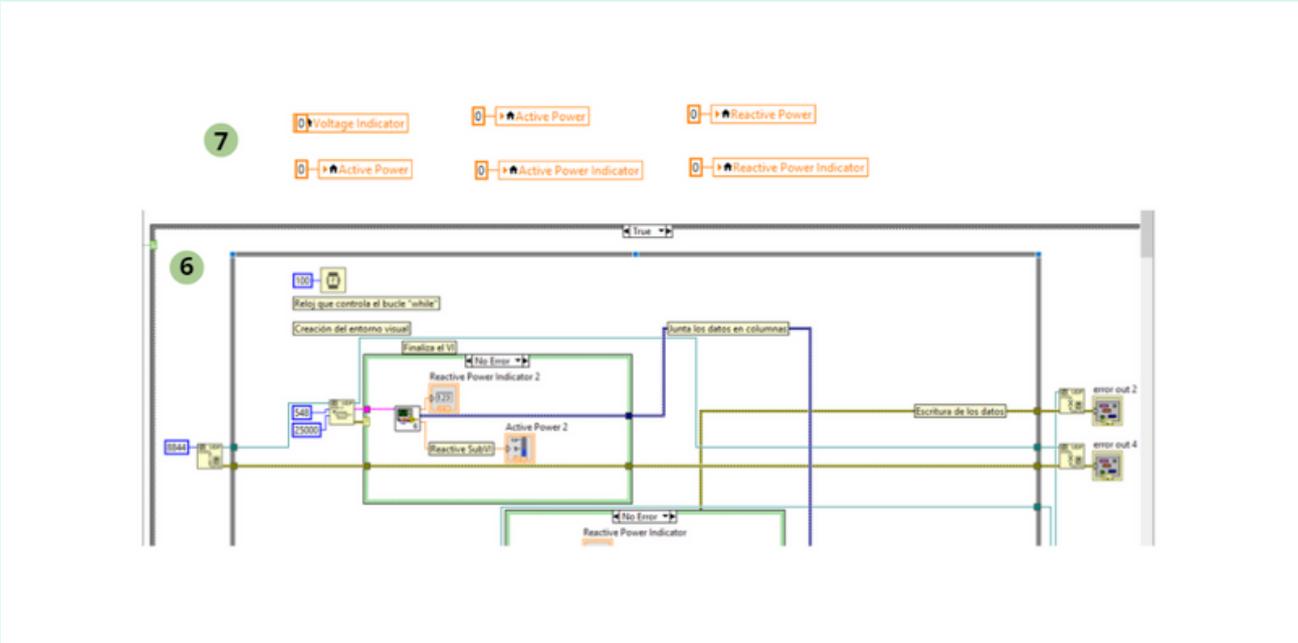
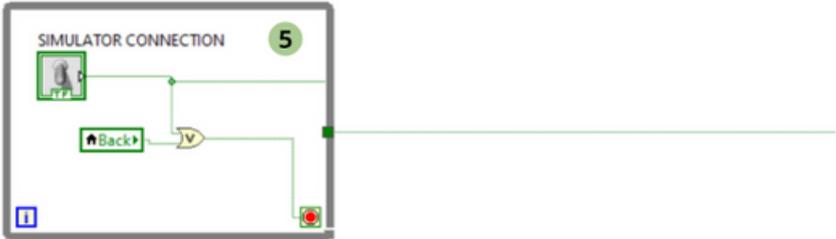
1

▶▶SIMULATOR CONNECTION

- ▶▶Help
- ▶▶Back
- ▶▶Simulation
- ▶▶Data History
- ▶▶Real Time Data
- ▶▶Error Log
- ▶▶Breaker control

2





REAL-TIME DATA

The programming environment in this window counts on a while loop, in which there is the connection plug with the simulator.

Once connected, the cases structure will be validated. Within, there is another while loop which executes the data reception block sets continuously, until the return button is pressed.

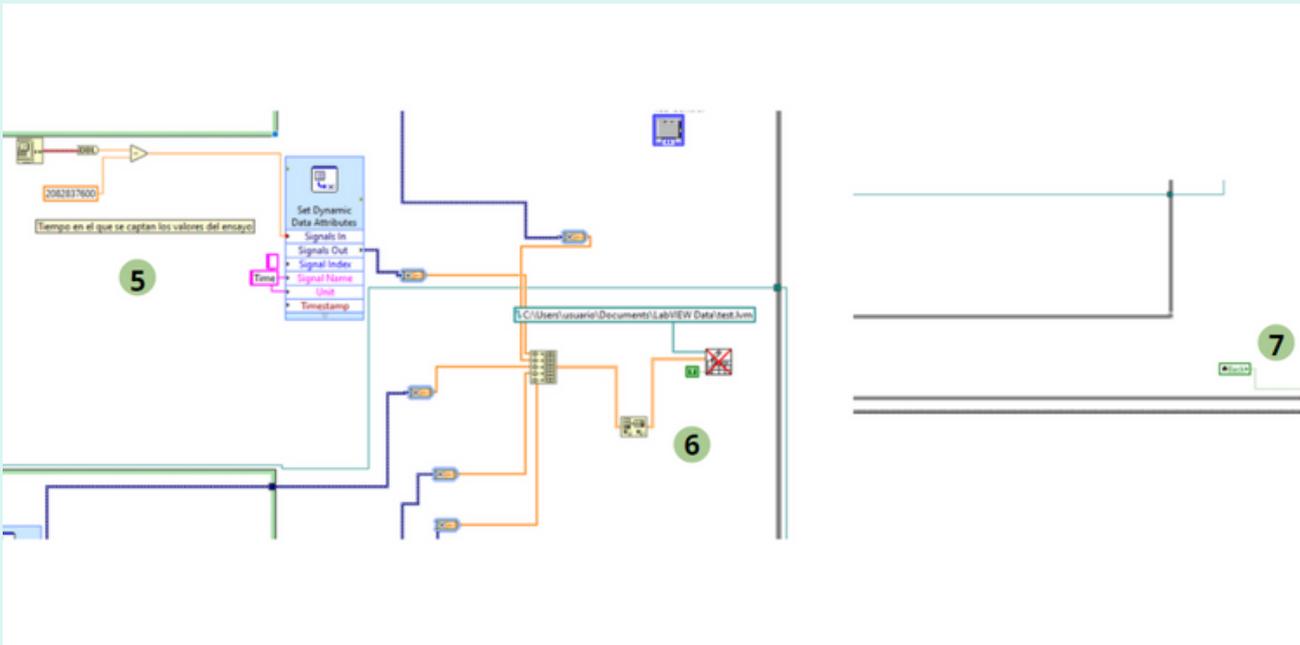
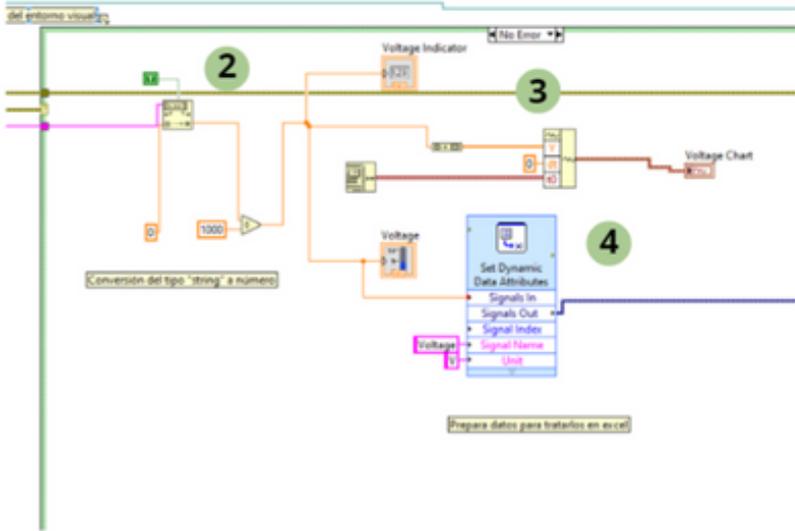
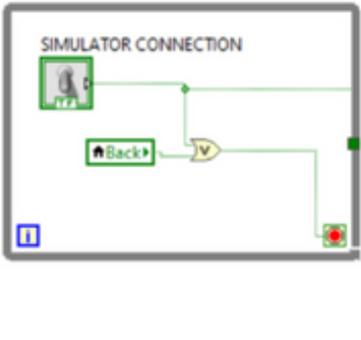
The data reception consists of an UDP sending protocol opening block, where the applying local port is assigned; a reading block which receives the signal; a series of signal-handling blocks and a closing protocol block.

Like in the rest of the windows, some error indicators appear, which will be used by the error tab.

Modifications could be changing the database type, the employed ports and the type of received variables; just as the character of the employed indicators and the data reception frequency.

- 1** While loop
- 2** Data reception
- 3** Indicators
- 4** Data preparation
- 5** Time
- 6** Data writing
- 7** Return button

1 SIMULATOR CONNECTION



SIMULATOR "BLOCK DIAGRAM"

This "block diagram" gathers aspects that have already been used in the *"breaker"* and *"real time"* tabs.

It is a big while loop, where two simultaneous tasks are carried out: the data sending and data reception via UDP. To this end, the typical "opening, sending and closure" or "opening, reception and closure" scheme will be followed, according to the case.

In both the sending and reception there is an error indicator available, employed by the *"error log"* page: for its part, the reception goes on with a plotting system.

A Boolean operator has been used to offer the user the possibility to choose plotting the variable at issue or not, matched with a graph to visualize them.

Just like in the rest of the tabs, this one has a return button that sends the user back to the main menu.

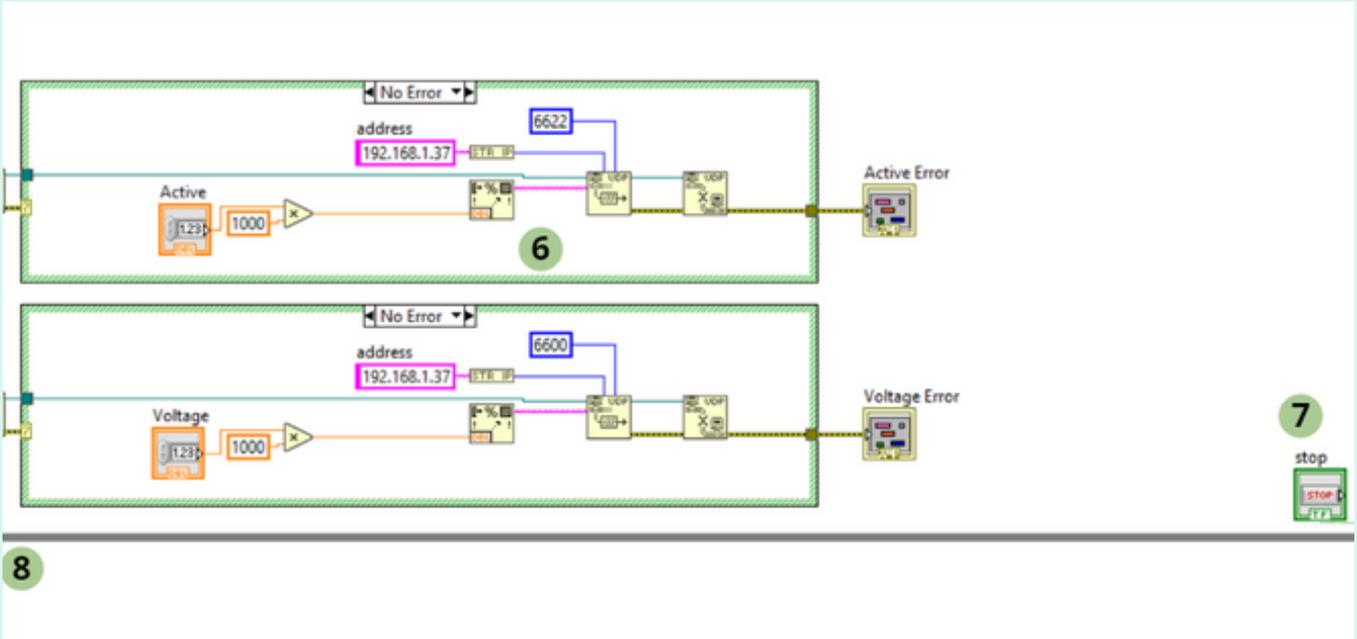
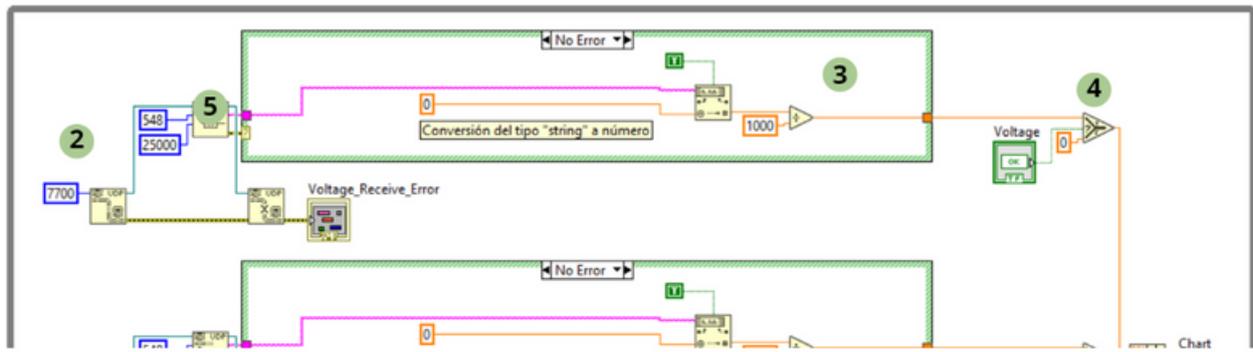
The reception and sending variables, as well as the indicators are completely modifiable according to the programmer's preferences.

- 1** Variable restart
- 2** UDP opening
- 3** Data reception
- 4** Keys
- 5** Graph
- 6** Sending structure

- 7** "Back" button
- 8** While loop

1

- Current
- Voltage
- stop
- Reactive Power
- Active Power
- Chart



8

DATA HISTORY

This program's feature is composed by two subtabs: the first of them allows to select the date, and the second represents the data.

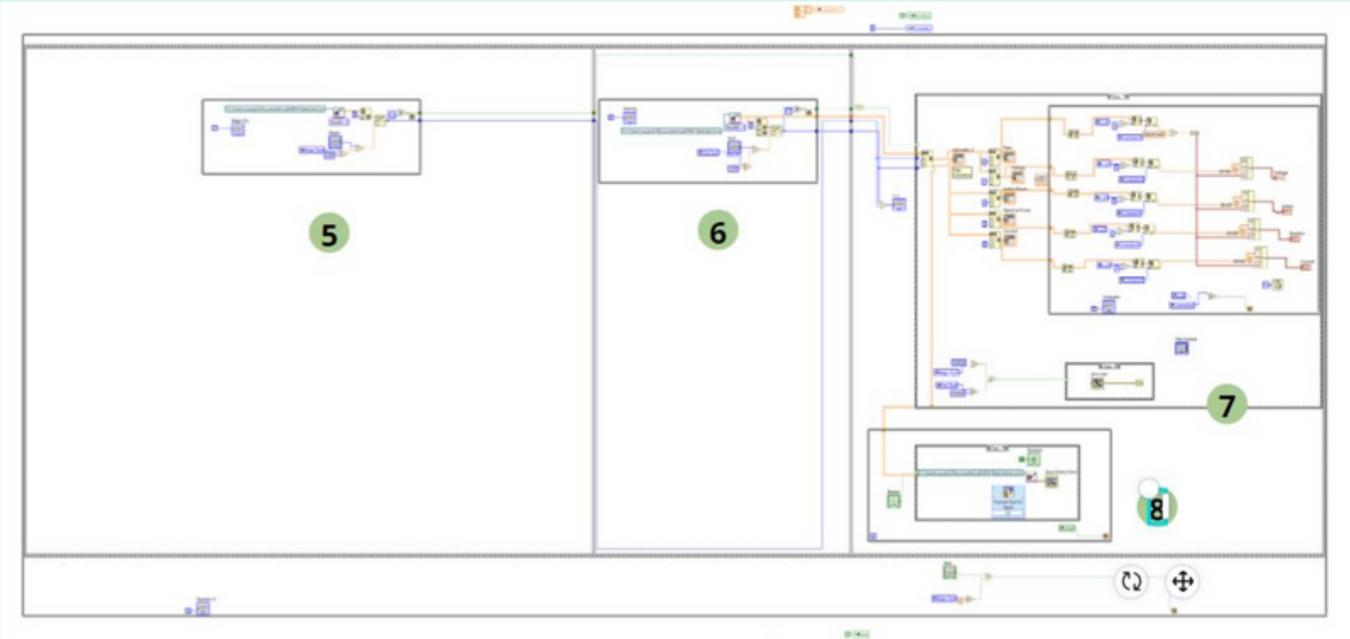
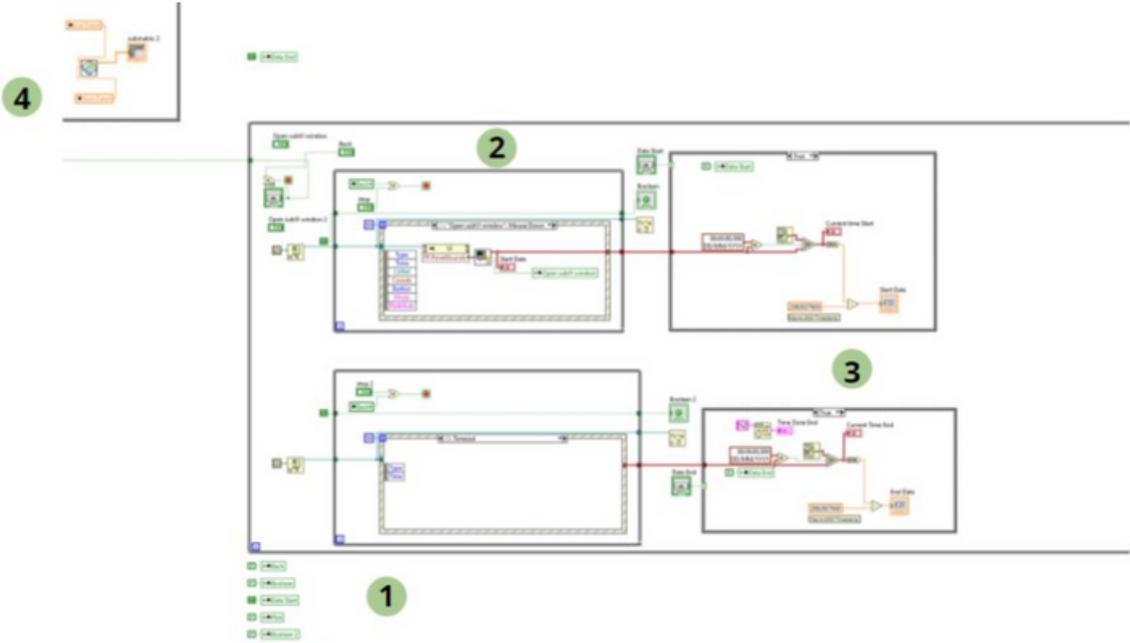
For the date selection there is a big while loop. Within, there are two conditional blocks to be found, which will be executed when the date is chosen and confirmed. The big while loop will end when these two parameters are fulfilled and the user clicks the plotting button.

In this moment the program will open another window, these second window is composed by a sequence: two while loops

,where the software looks for the next data and two last while loops. The first consists of matrix-handling blocks for plotting and the second comprises a conditional for the storage of plotted data. All this inside a big while loop, which will be run until the user suggests otherwise.

This is the most developed software code, and every modification requires a lot of caution to not mismatch the whole. It is advised not to conduct any changes, besides the possible modification of the database.

- 1** Variable restart
- 2** While loop
- 3** Conditional structures
- 4** Second tab opening
- 5** Beginning search
- 6** Ending search
- 7** Plotting
- 8** Data saving



AUTOMATIC SWITCH

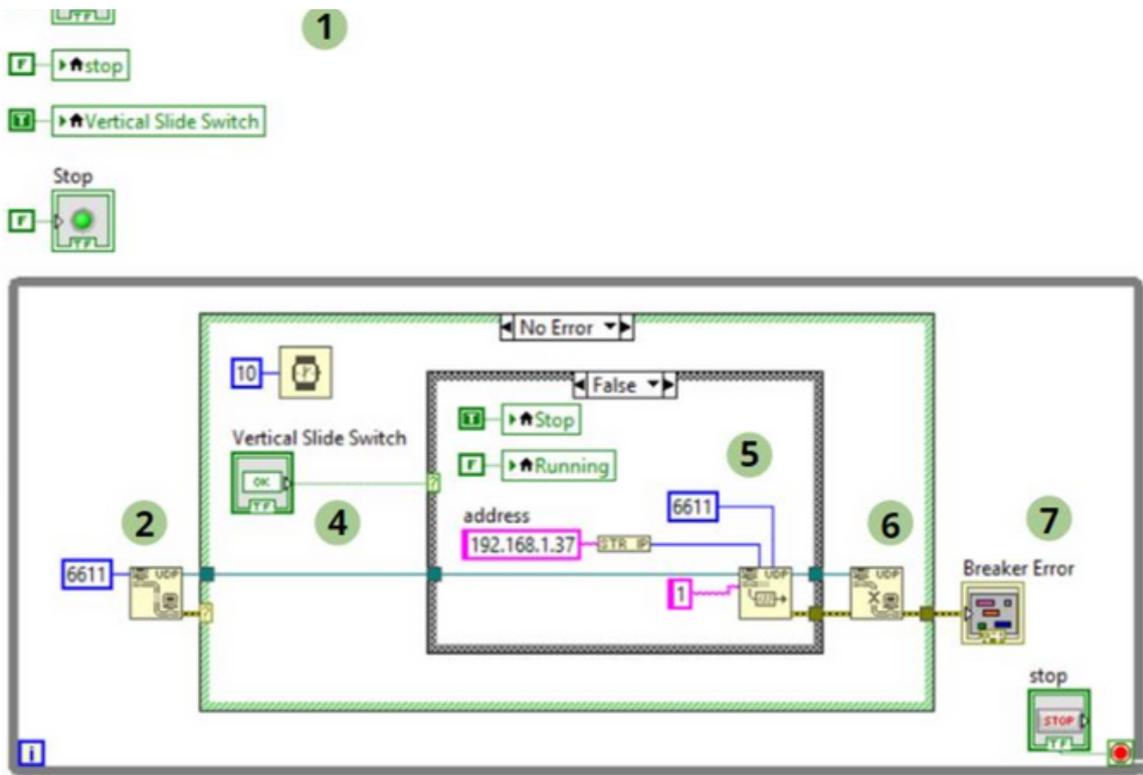
This window's programation code is composed by a variable resetting (including both LEDs) and a while loop that, similarly to the preceding, it is reproduced until until de exit condition is validated.

In this loop there is a sending via UDP protocol to find; first of all, the protocol is opened, then a port is assigned (in this case the 6611).

If this procedure happens without problems, the conditional structure is validated. Within it, there is the switcher and according to its position, one or another part of the conditional structure is opened or a 0 if, on the contrary, it is closed.

Inside this last structure there is the delivery that will take place after introducing the destination port and the address (which should be modified from numerical fact to *"string"*).

- 1** Variable restart
- 2** UDP opening
- 3** While loop
- 4** Switch
- 5** Conditional structure
- 6** UDP protocol closing
- 7** Error
- 8** Return button



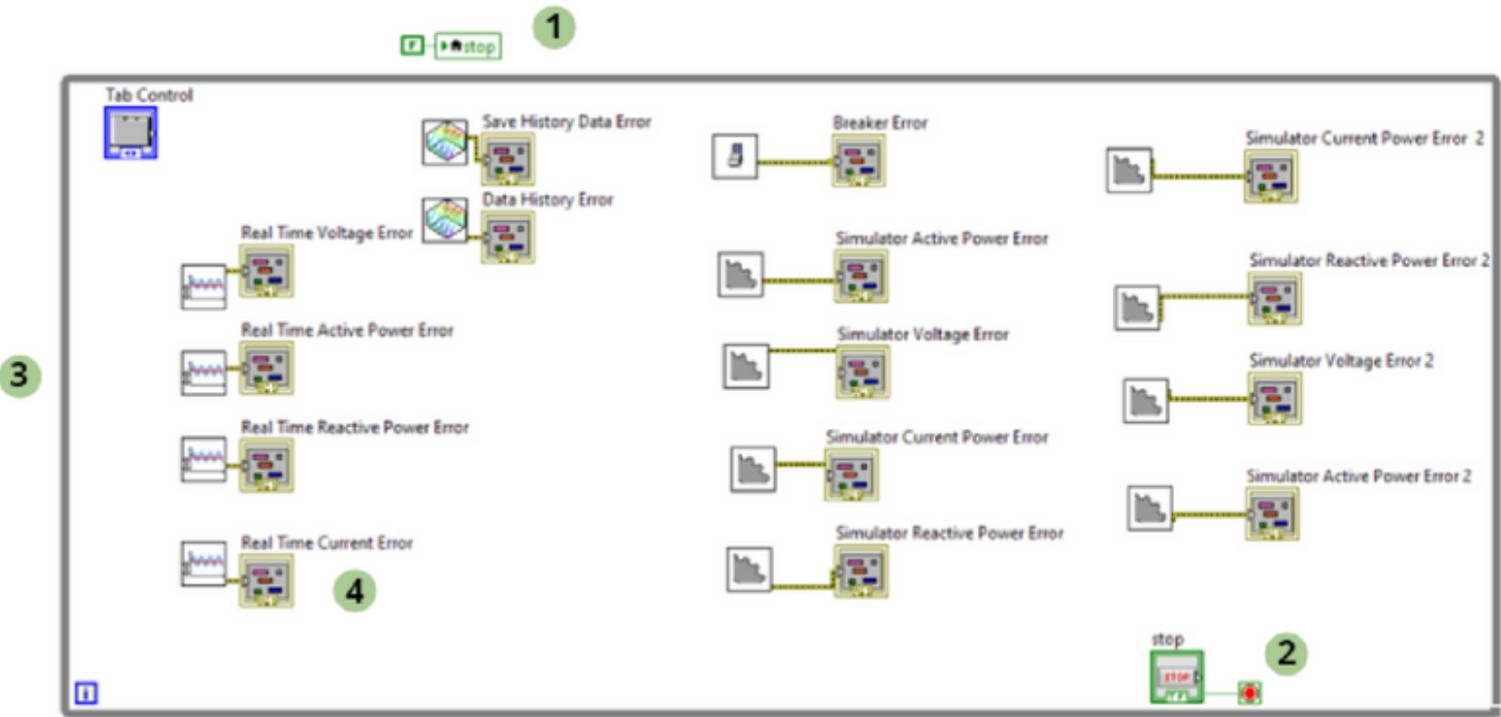
ERROR CODES & FAQs

The programming environment of the error tab is no more than a sort of repository where all information about mistakes in the rest of tabs is grouped, and has been made invisible for convenience purposes.

That's why there will be a while loop available, that will be repeating itself until the output condition is fulfilled. Within, there are appeals to the errors occurring in the different windows, together with a graphic indicator for each of them.

The only possible change is adding more error indicators. To this effect, the exposed structure must be kept: tab call + indicator

- 1** Variable restart
- 2** Return button
- 3** While loop
- 4** Appeals to other tabs



ERROR CODES & FAQs

CODE

DESCRIPTION

#54	<i>The Network address is ill-formed. Make sure the address is in a valid format. For TCP/IP, the address can be either a machine name or an IP address in the form xxx.xxx.xxx.xxx. If this error occurs when specifying a machine name, make sure the machine name is valid. Try to ping the machine name. Check that you have a DNS server properly configured.</i>
#55	<i>The network operation is in progress. If you receive this error while using the UDP Write function, refer to the KnowledgeBase at ni.com for more information.</i>
#56	<i>The network operation exceeded the user-specified or system time limit.</i>
#57	<i>The network connection is busy.</i>
#58	<i>The network function is not supported by the system.</i>
#59	<i>The network is down, unreachable, or has been reset.</i>
#60	<i>The specified port or network address is currently in use. Select an available port or network address.</i>
#61	<i>The system could not allocate the necessary memory.</i>
#62	<i>The system caused the network connection to be aborted.</i>
#1011	<i>No data found for the selected time frame.</i>
#44	<i>Problem while trying to save a file.</i>

ERROR CODES & FAQs

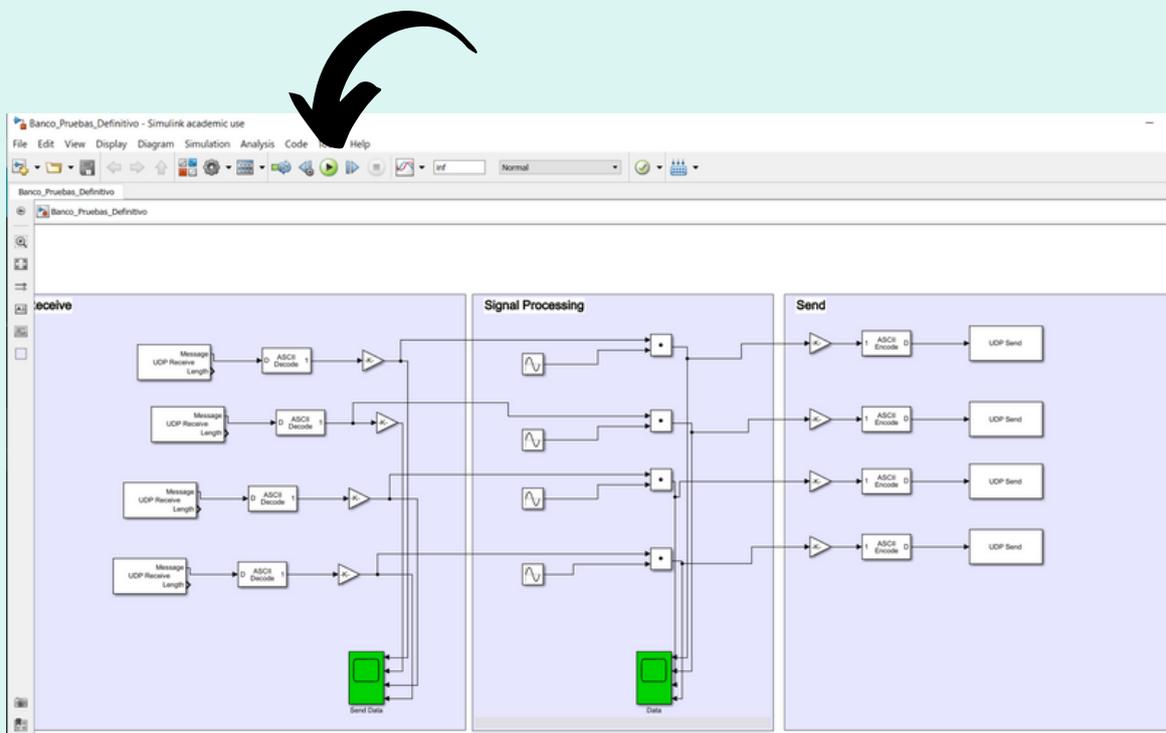
How to run LabVIEW code?

To execute a code in LabVIEW the user must click on the "run" button inside the "log-in" window since this is the main window of the program.

To know more about LabVIEW, please consult: <https://www.ni.com/getting-started/labview-basics/esa/>

How to run MATLAB code?

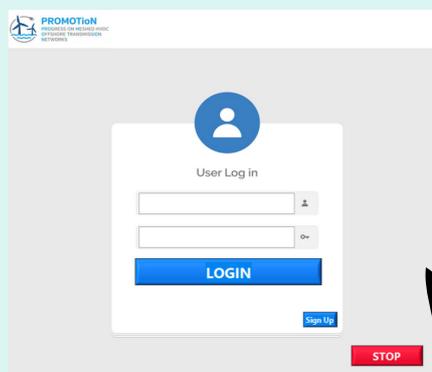
To execute a code in MATLAB, more specifically in Simulink. The user will have to click on the "run" button as shown in the figure below.



ERROR CODES & FAQs

How to finalize LabVIEW code?

The program is configured to be completed without the need to press the "abort" button. To do this, the user must go back to the "log-in" window and click on the program end button.



How to finish the code in LabVIEW?

To make this conversion, is recommended using the online converter, for example: <https://www.unixtimestamp.com/index.php>

What to do if problems occur during the register?

If the registration generates problems, it is most likely that you do not have a good network connection or that the e-mail has been used previously; to solve this you can use another e-mail address or check that the connection is correct.

*For more information about the program, please contact us by e-mail:
juyusta@etsii.upv.es*

Capítulo VI.4

Programmiererhanbuch (Deutsch)

JULIO YUSTAS TALAMANTES



PROGRAMMIERER- HANBUCH

INHALT



1
ÜBER DAS PROGRAMM

2
PROGRAMMSTRUKTUR

3
**BENUTZEROBERFLÄCH
E**

4
**TECHNISCHE
ANFORDERUNGEN**

5
INSTALLIERUNG

6
**APP
STARTEN/BEGINNEN**

7
**REGISTER "BLOCK
DIAGRAM"**

INHALT



8

**IDENTIFIZIERUNG
"BLOCK DIAGRAM"**

9

**HAUPTMENÜ
"BLOCK DIAGRAM"**

10

**LIVE-DATEN "BLOCK
DIAGRAM"**

11

**SIMULATOR
"BLOCK DIAGRAM"**

12

**DATENHISTORIE
"BLOCK DIAGRAM"**

13

**AUTOMATISCHER
SCHALTER
"BLOCK DIAGRAM"**

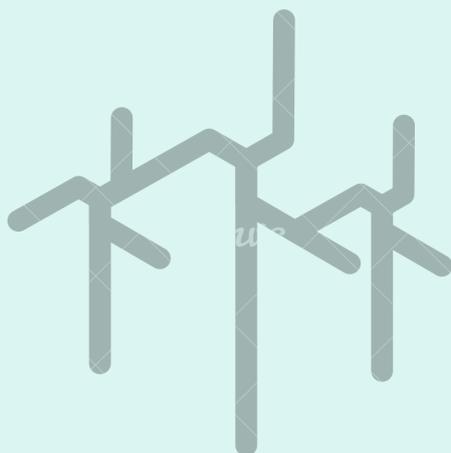
14

**FEHLERCODES UND
HÄUFIG GESTELLTE
FRAGEN**



ÜBER DAS PROGRAMM

Dieses Programm wurde als Teil des Abschlussarbeits von Julio Yustas Talamantes entwickelt; die ernannte Arbeit ist an der Investigation der Universitat Politècnica de València für das europäische Windenergieprojekt "PROMOTioN" beteiligt, wessen Hauptziel die Nutzung aller Windenergieressourcen in der Nordsee ist.



Dieses Programm wurde für seine Nutzung als Überwachungsapp für Offshore-Windparks entwickelt.

Seine Hauptaufgabe ist, dem Bediener eine klare und einfache Benutzeroberfläche für die korrekte Ausführung seiner Arbeit bieten.

Wiederum, es kandidiert als die Grundlagen, auf denen man ein viel mehr hochentwickeltes Programm erbauen könnte. Dafür bietet man das vorhandene Programmiererhandbuch.



ÜBER DAS PROGRAMM

Die Autoren dieser App waren Julio Yustas Talamantes, unter die Betreuung der Doktorin Soledad Inmaculada Bernal Pérez (Dr. Wirtschaftsingenieurin).

Die vorliegende Arbeit wurde dank der Zusammenarbeit der Universitat Politècnica de València mit dem Instituto Universitario de Automática e Informática Industrial (ai2) entwickelt.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

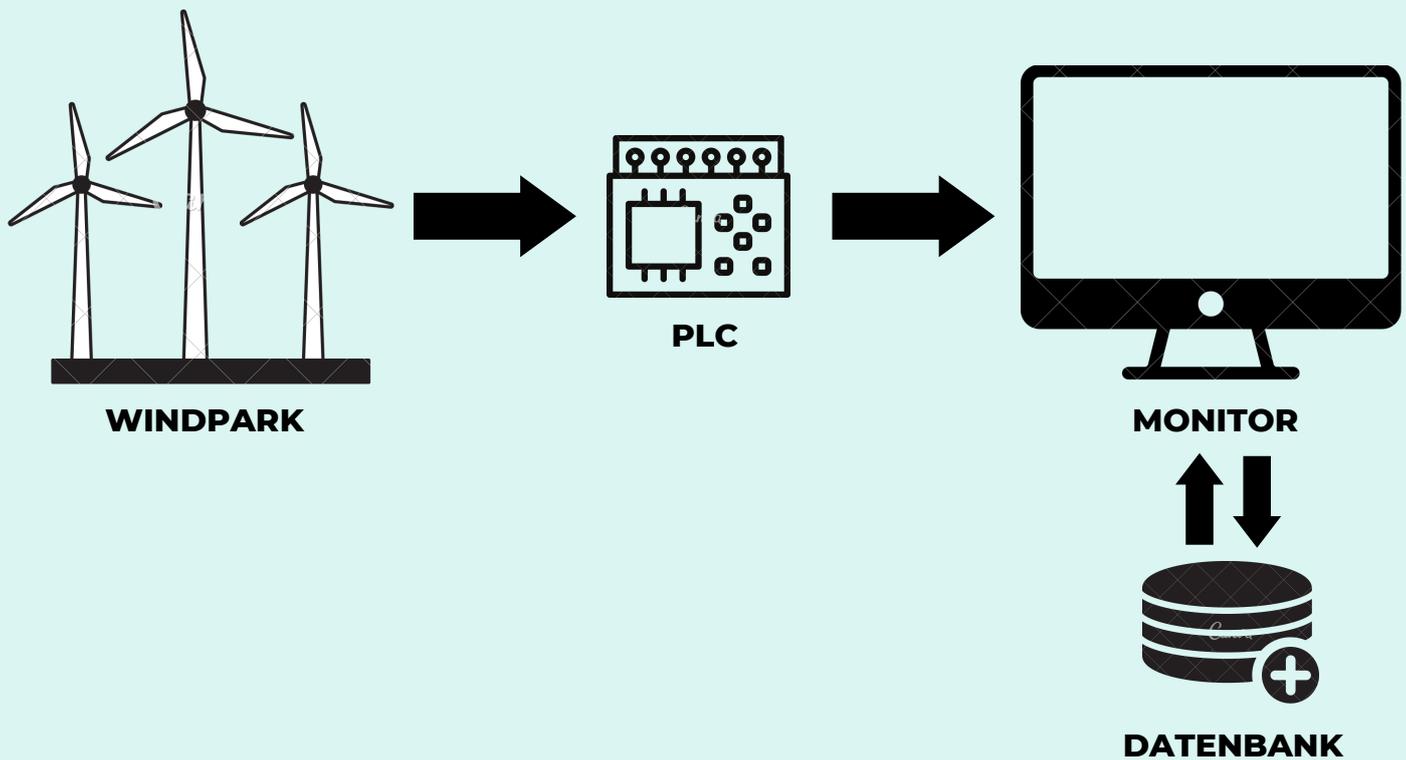


ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA



PROMOTiON
PROGRESS ON MESHED HVDC
OFFSHORE TRANSMISSION
NETWORKS





PROGRAMM-STRUKTUR

Es handelt sich um ein SCADA-System (Beaufsichtigung, Kontrolle und Datenerfassung).

Das Hauptziel ist, ein Windpark an einem anderen Standort aus der Ferne kontrollieren. Dafür braucht man 3 Kernelemente: der Windpark, die PLCs und das Monitoringsystem.

Die Verbindungen zwischen den Bestandteilen sind ein wesentlicher Aspekt für die ordnungsgemäße Leistung des vorliegenden Programms.

Die Verbindung zwischen den Windpark und die Sensoren wird durch eine Verkabelung stattfinden.

Diese Sensoren werden wiederum am Netz angeschlossen, mit der Absicht, die Kommunikation durch UDP Protokoll mit dem Simulator oder dem SCADA-Programm zu erlauben.

Es ist Aufgabe des SCADA-Programms, diese Daten zu behandeln und an ein attraktives Umfeld für den Arbeiter anpassen.

Wiederum, wird es diese Daten in einer Datenbank speichern, um eine spätere Abfrage zu erlauben.

BENUTZEROBERFLÄCHE

Das Programm besitzt eine einfache und ansprechende Benutzeroberfläche, mit der Hilfe von LabVIEW und Photoshop realisiert/vollbracht.

Die Navigation ist leicht: um sich zwischen den verschiedenen Tabs zu bewegen, muss man nur den Modus vom Hauptmenü oder "main menu" auswählen.

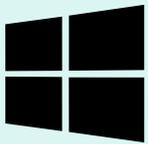
Im Fall, dass man zurückkehren möchte, hat man jederzeit eine "back" Taste zur Verfügung, die zur vorgangenen Seite führt.

Um das Programm zu beenden, muss man einfach zur Identifizierungstab zurückgehen, und auf die "stop" Taste klicken, falls nicht möglich, den klassischen Tabsschliesserbutton von Windows verwenden.



TECHNISCHE ANFORDERUNGEN

VON EINEM VIRTUELLEN SIMULATOR:



BETRIEBSSYSTEM

Windows 7, Ubuntu 16.04 oder macOS Sierra von 64 bits.



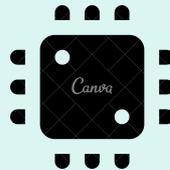
BROWSER UND NETZANSCHLUSS

Internet Explorer 11, Google Chrome 58 oder Firefox 53



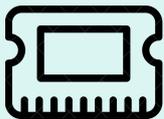
MATLAB

R2017b oder höher



PROZESSOR

Intel Core I5 oder höher



RAM-SPEICHER

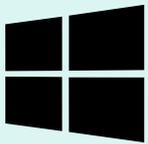
RAM-Speicher 8G oder höher



SOUNDKARTE UN LAUTSPRECHER

TECHNISCHE ANFORDERUNGEN

VON EINER REELLEN WINDFARM:



BETRIEBSSYSTEM

Windows 7, Ubuntu 16.04 oder macOS Sierra von 64 bits.

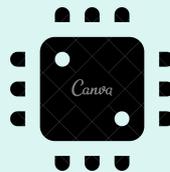


BROWSER

Internet Explorer 11, Google Chrome 58 oder Firefox 53

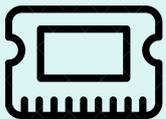


NETZANSCHLUSS



PROZESSOR

Intel Core I5 oder höher



RAM-SPEICHER

RAM-Speicher 8G oder höher



SOUNDKARTE UND LAUTSPRECHER

INSTALLIERUNG

Die Installierung des vorliegenden Programms ist ziemlich einfach, dafür braucht man nur eine Reihe von Schritten:

Die erste Phase besteht darin, LabVIEW mit den folgenden Erweiterungen herunterzuladen: *LabVIEW Control Design and Simulation Module (32-bit)*, *LabVIEW Database Connectivity Toolkit (32-bit)*, *LabVIEW Report Generation Toolkit (32-bit)* und *LabVIEW Runtime (32-bit)*.

Einmal hat man das Programm heruntergeladen, muss man die im Repository beigefügte .zip Datei auch herunterladen und dekomprimieren. Später soll man das Projekt PROMOTioN ausführen.

Wenn man das Programm zusammen mit dem Simulator verwenden möchte, muss man im selben Computer -oder in einem anderen, zum selben WLAN Netz angeschlossen- das MATLAB Programm installieren und die Simulatoren durchführen: *parque.mdl* oder *simulator.mdl*

Wenn man, dagegen, einen Prototyp im Masstab oder eine echte Windfarm verwenden möchte, soll man eine Synchronisierung der Sendeanschlussnummer der PLCs und der Eingänge des Programms durchführen; mit der folgenden Tabelle übereinstimmend:

Variable zu kontrollieren	Anschluss
Spannung	8811
Strom	8822
Wirkleistung	8833
Blindleistung	8844

Die ernannte Synchronisierung wird im Prototyp im Masstabsab oder im Windpark durchgeführt, nämlich immer über das PLC und nicht über das Programm.

Kontrolle	Versand
Spannung	7700
Strom	7711
Wirkleistung	7722
Blindleistung	7733

Die vorherige Einstellung wird für die Simulatorfunktionalität verwendet; die Eingänge/Anschlüsse sind denen, die seitens des Simulators und/oder PLC senden.

ERST- AKTIVIERUNG

Um die ordnungsgemässe Initialisierung des Programms zu gewährleisten, muss man überprüfen, dass beide Seiten - der Computer ("host" für das SCADA-System) und die PLCs - über Internetverbindung verfügen.

Wiederum, sollte man verifizieren, in ähnlicher Weise als im vorherigen Absatz dargelegt, dass eine korrekte Anordnung der Eingang- und Ausgangsanschlüsse besteht.

Dieser letzte Schritt ist von grundlegender Bedeutung, um den Erfolg der Kommunikation via UDP Protokoll sicherzustellen.

Andererseits, muss der Benutzer sich mit den einmaligen und nicht übertragbaren Login-Daten, die die Firma zur Verfügung gestellt hat, anmelden.

Im Fall, über diese nicht zu verfügen, kann man sich im Datenbank autorisierter Bediener registrieren, mittels die "sign-up" Registerkarte.

Dieser letzte Tab wird ausführlicher in den nächsten Punkten dieser Bedienungsanleitung erklärt.

Nun wird wieder die Tabelle der UDP Anschlüsse beigefügt:

Variable zu kontrollieren	Anschluss
Spannung	8811
Strom	8822
Wirkleistung	8833
Blindleistung	8844

Schliesslich, um die ersten Daten vom Windpark empfangen zu können, bleibt nur übrig, auf den Schalter zu drücken, der die Verbindung startet -in der "main menu" Registerkarte vorhanden-.

REGISTER "BLOCK DIAGRAM"

Die "sign-up" Registerkarte ermöglicht, dass die Benutzer sich im Kundendatenbank registrieren können.

In den beigelegten Abbildungen kann man den Programmcode für das Benutzerregistrierungsfenster erkennen.

Die erste Komponente zu hervorheben ist eine grosse Schleife "while", dass einige Bestandteile umfasst:

- Ein Loop der gleichen Typologie, dessen Ziel ist, das Programm beliebig oft zu wiederholen, bis alle Anmeldefelder ausgefüllt werden.

- Ein Neustart der Variablen zu ergänzen, dies findet beim Start des Fensters statt und wenn die Anmeldung wiederholt werden muss, weil sie fehlerhaft erfolgt.
- Die Feststellung, dass die verschiedenen Bedingungen für das Register erfolgreich erfüllt sind: alle Felder ausgefüllt, und das User und die E-Mail Adresse nicht vorher benutzt.
- Ausserdem kann man ein Timer und jeweilige Bedingungen für den Abschluss beider Schleifen finden.

- | | |
|---------------------------------------|---------------------------------|
| 1 Schleife while | 7 Register in Datenbank |
| 2 Variablenneustart | 8 Bedingungen |
| 3 Zeitschaltuhr | 9 Nachricht |
| 4 Registrieren (Registertaste) | 10 Struktur des Versands |
| 5 Benutzername | 11 Fehlermeldungen |
| 6 E-Mail Adresse | |

REGISTER "BLOCK DIAGRAM"

Im Aussenbereich der genannten Struktur findet man eine Konditionalstruktur, deren Funktion die Aufbewahrung der Parameter im Datenbank "*login.lvm*" ist, für den Fall, dass alle vordere Bedingungen erfüllt werden.

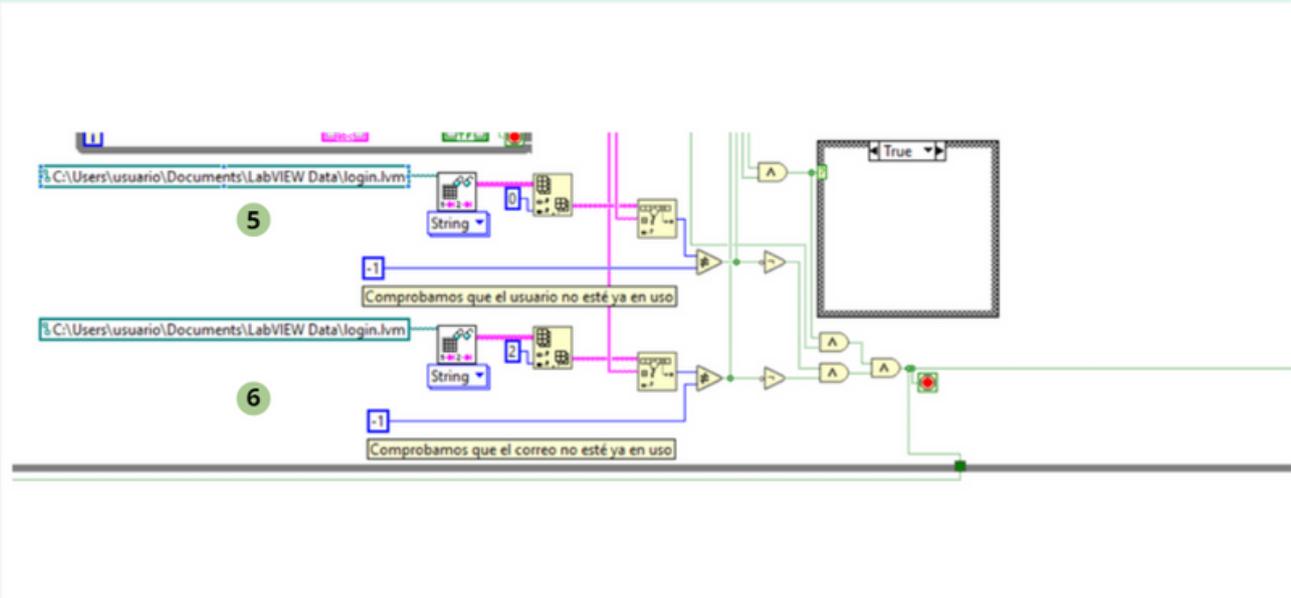
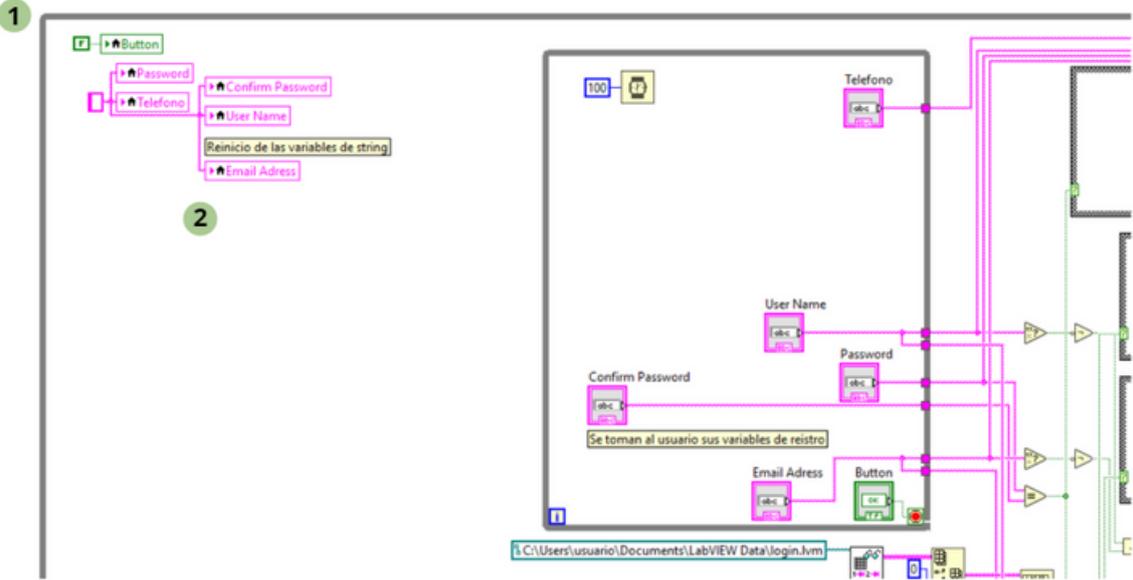
Zuletzt kann man den Start einer Reihe an Variablen Art "string", Text, beobachten, die später in der direkt unteren Struktur benutzt werden. Deren Funktion ist, eine Bestätigungsmail an dem neu registrierten User zu senden.

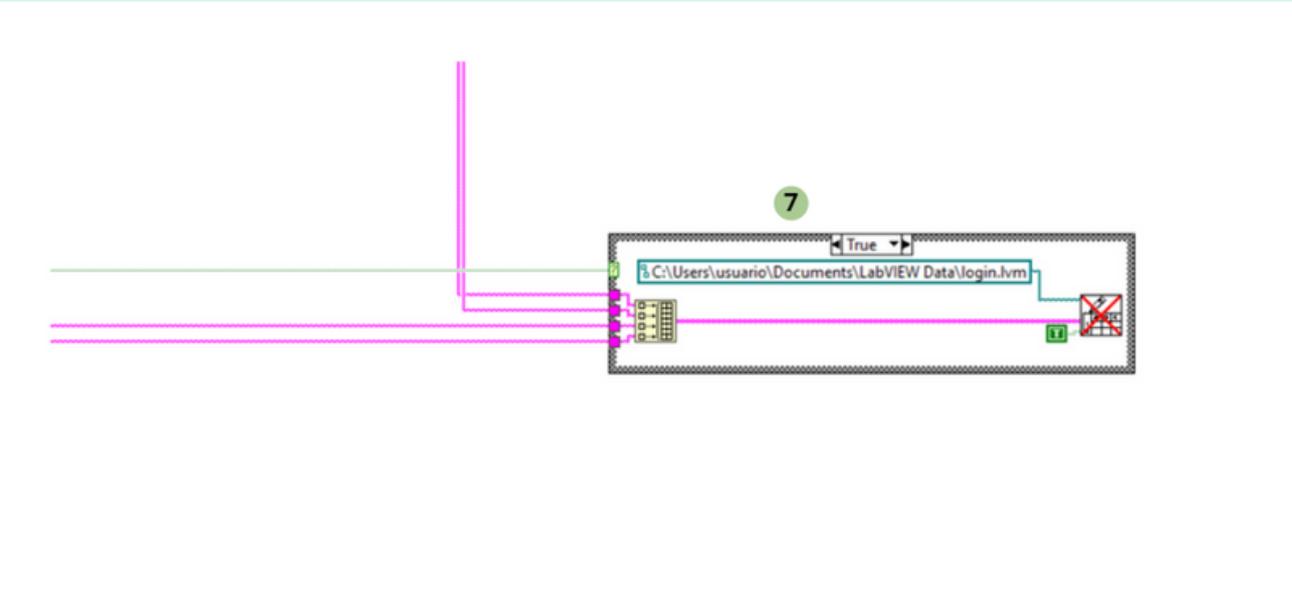
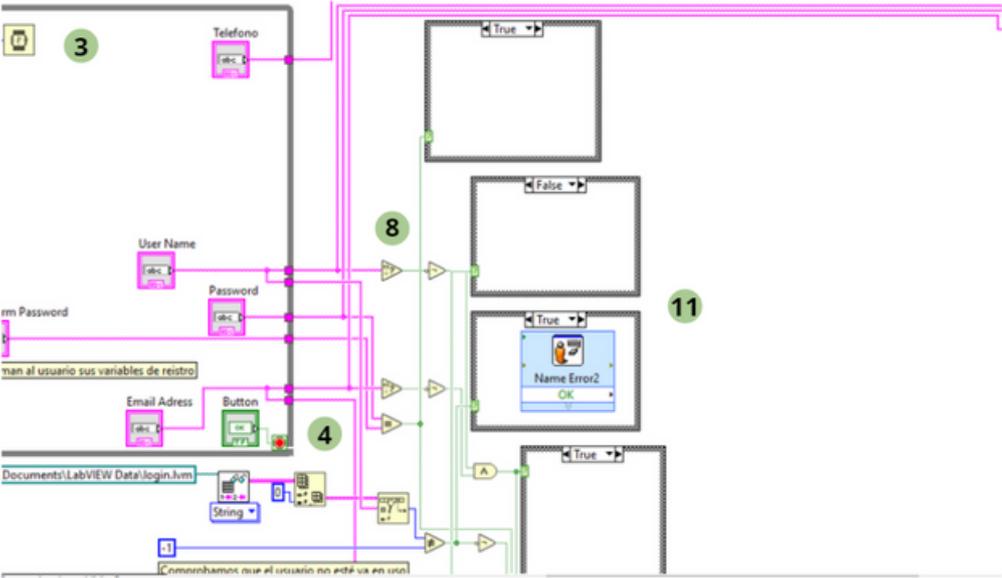
Im Fall, dass alles oben genannte erfolgreich stattgefunden hat, wird das Programm eine Bestätigungsnachricht für den Nutzer erzeugen.

Die Hauptelemente, die von einem Programmierer geändert werden könnten, sind: die Ersetzung einer Datenbank Typ *.lvm* durch einen höheren entwickelten Datenbanktyp; die Einführung eines Bestätigungselementes durch eine zufällige Zahlenkombination, die in der Bestätigungsmail gesendet wird, etc

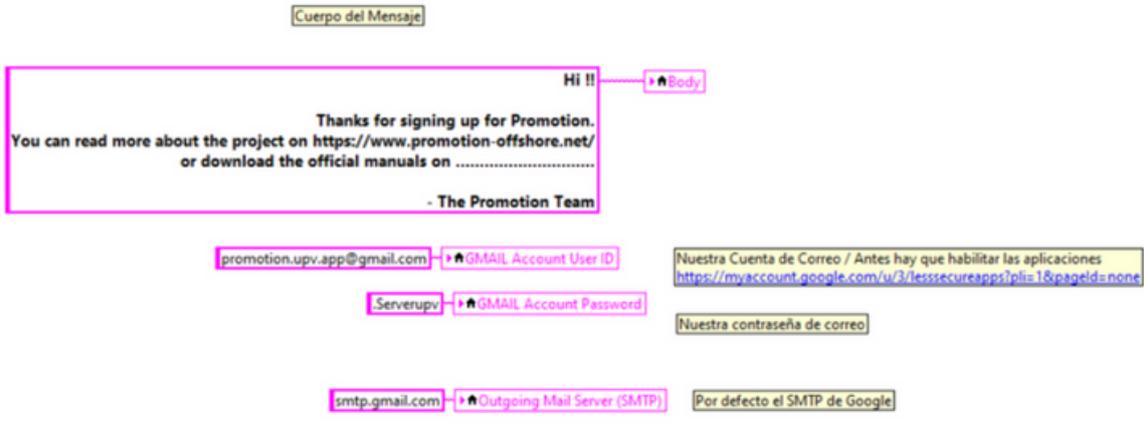
Wiederum kann man den Inhalt der Mitteilung und ihr Design modifizieren.

- | | |
|---------------------------------------|---------------------------------|
| 1 Schleife while | 7 Register in Datenbank |
| 2 Variablenneustart | 8 Bedingungen |
| 3 Zeitschaltuhr | 9 Nachricht |
| 4 Registrieren (Registertaste) | 10 Struktur des Versands |
| 5 Benutzername | 11 Fehlermeldungen |
| 6 E-Mail Adresse | |

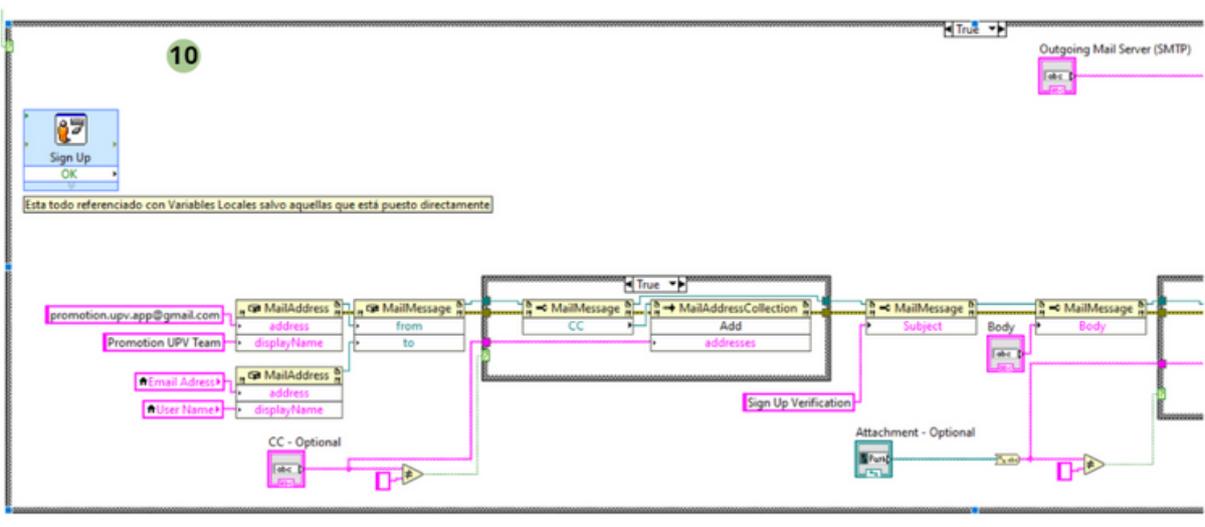


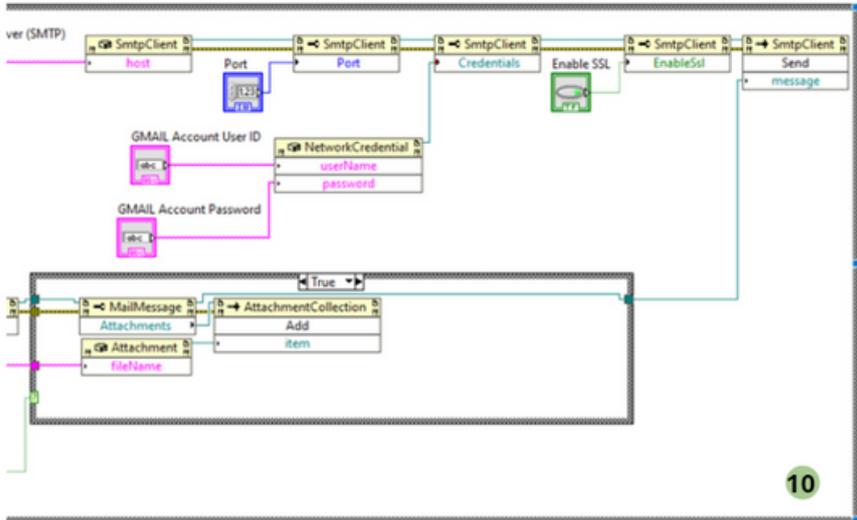


9



10





IDENTIFIZIERUNG "BLOCK DIAGRAM"

Mittlerweile wird das Blockschaltbild oder Programmierung des grafischen Umfelds für die Useridentifizierung dargelegt,

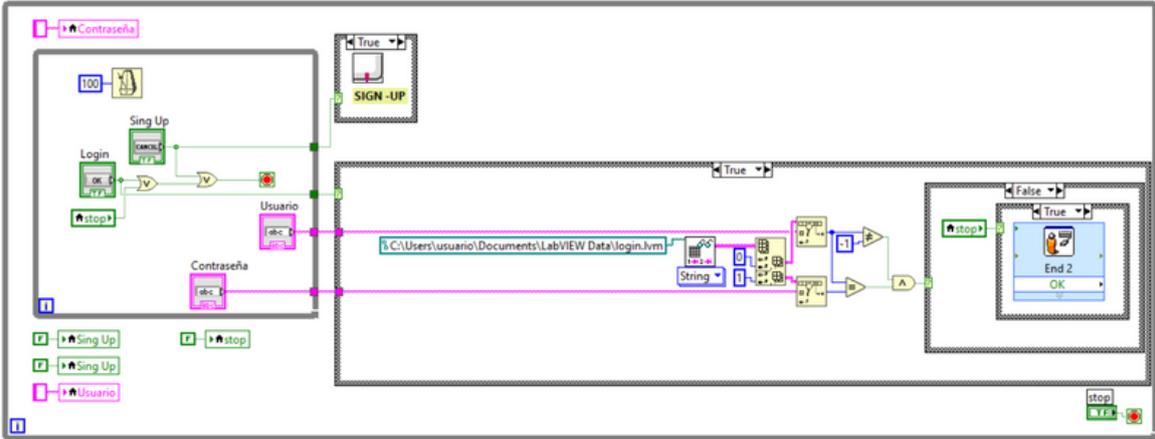
Der vorhandene Code befindet sich in einer Schleife while. Deren Funktion ist, das Programm zu wiederholen, sofern die Option, die den Wiederholungsloop validiert, nicht validiert wird -das heisst, "stop"-Button gedrückt. Das würde eine Schliessmeldung erzeugen und würde das Programm beenden; auf einmal würden auch die Variablen und die Registerfelder zurückgesetzt.

In dieser Schleife while befindet sich eine andere Schleife while, mit den Kästchen, wo die Login-Daten eingegeben werden: Benutzername und Kennwort, und die Tasten ("login" und "sign-up"). Die zwei letzten sind dafür zuständig, den Durchgang zu den nächsten Blöcken zu erlauben.

Beide Blöcke sind bedingt: der erste wird die Registrierungregisterkarte ausführen, wenn die Bestätigung der früheren Schleife while mittels den "sign-up"-Button erfolgt ist.

Der zweite Konditionalblock wird ausgeführt, wenn den "login"-Button gedrückt wird; in diesem Moment öffnet das Programm die Datenbank "login.lvm" und findet eine direkte Anfrage von Benutzername und Kennwort statt. Je nachdem, ob die eingesetzten Daten gefunden werden oder nicht, wird das Programm dem User den Durchgang zum "main menu"-Tab erlauben, oder wird es eine Fehlermeldung abgeben und den Nutzer auffordern, es wieder einmal zu versuchen, was den Kunden zum Ausgangszustand bringen würde.

- 1** Schleife while
- 2** Variablenneustart
- 3** Zeitschaltuhr
- 4** Tasten
- 5** Registerfelder
- 6** "Sign-up" Struktur
- 7** Suche
- 8** Aktionen



HAUPTMENÜ "BLOCK DIAGRAM"

in diesem "block diagram" findet man zwei Kernelemente, die gleichzeitig und unabhängig agieren:

Eine erste Schleife while, die den Code vom Beginn des Programms an wiederholt, als zugleich die verwendeten Variablen zurückgesetzt werden. Dieser Loop wiederholt sich ständig, auch wenn eine der Konditionalstrukturen, die sich darin befinden, validiert wird. Der einzige Weg, es zu beenden, ist, auf den "back"-Button zu klicken.

Eine andere Schleife while in einer Konditionalstruktur, die aktiviert wird, wenn den Live-Datenempfangstecker angeschlossen wird. Dieser Loop enthält, so oft wiederholt als Daten zu empfangen sind; die Öffnungs-, Lese- und Schlussverfahren

des UDP Protokolls, als auch die Informationbehandlung, um sie richtig auf die "front view" darstellen zu können.

En este diagrama de bloques no hay nada a modificar a no ser que el programador desee incorporar una nueva funcionalidad al programa o modificar el recibo de datos en tiempo real.

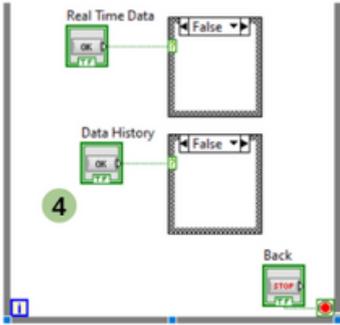
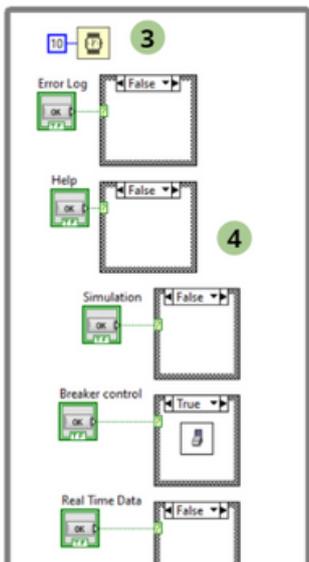
- 1** Variablenneustart
- 2** Schleife while
- 3** Zeitschaltuhr
- 4** Tasten
- 5** Beginn des Datenempfangs
- 6** Empfangstruktur
- 7** Zurücksetzen der Variablen vom UDP-Protokoll

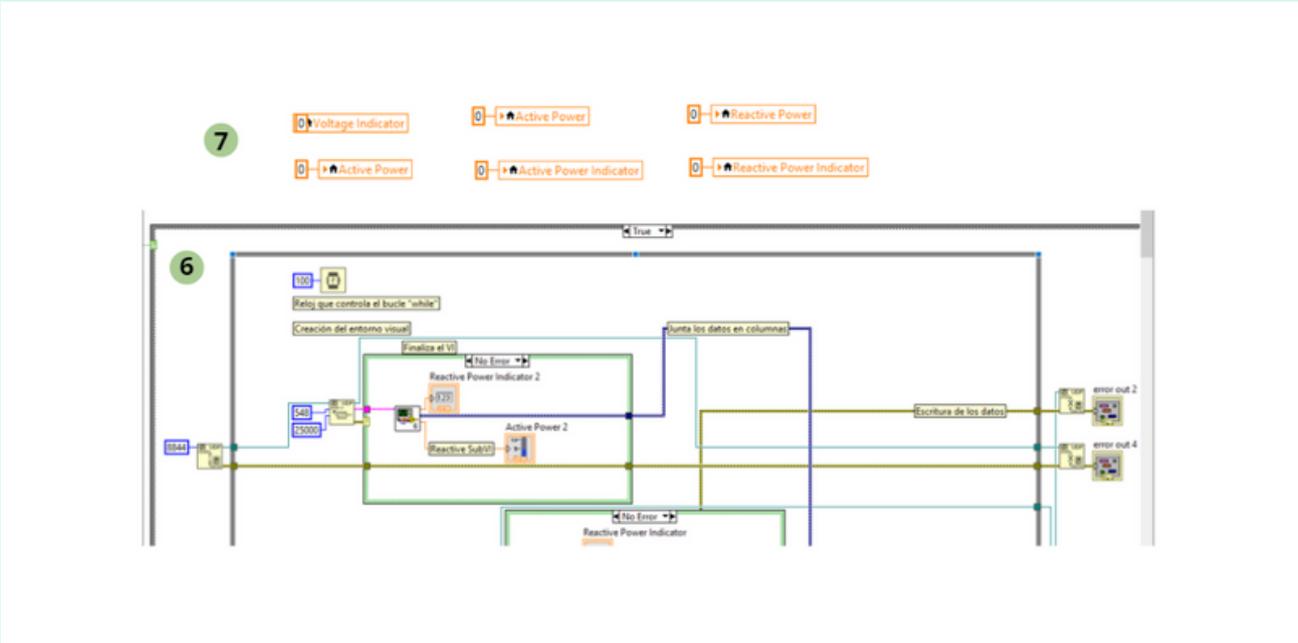
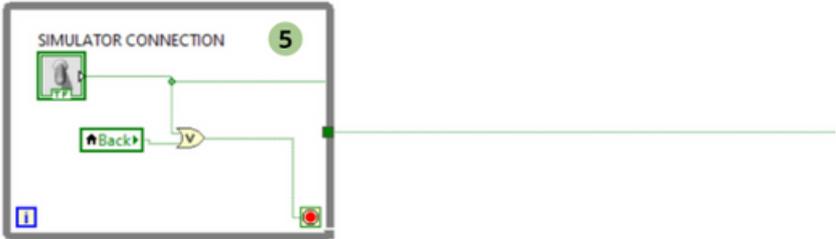
1

▶▶SIMULATOR CONNECTION

- ▶▶Help
- ▶▶Back
- ▶▶Simulation
- ▶▶Data History
- ▶▶Real Time Data
- ▶▶Error Log
- ▶▶Breaker control

2





LIVE-DATEN

Die Programmierumgebung dieses Fensters besitzt eine Schleife while, in der sich den Anschlussstecker mit dem Simulator befindet.

Einmal verbunden, wird die Fällestruktur validiert. Hier gibt es eine zusätzliche Schleife while, die das Ensemble der Datenempfangblöcke kontinuierlich ausführt, bis den Zurück-Button gedrückt wird.

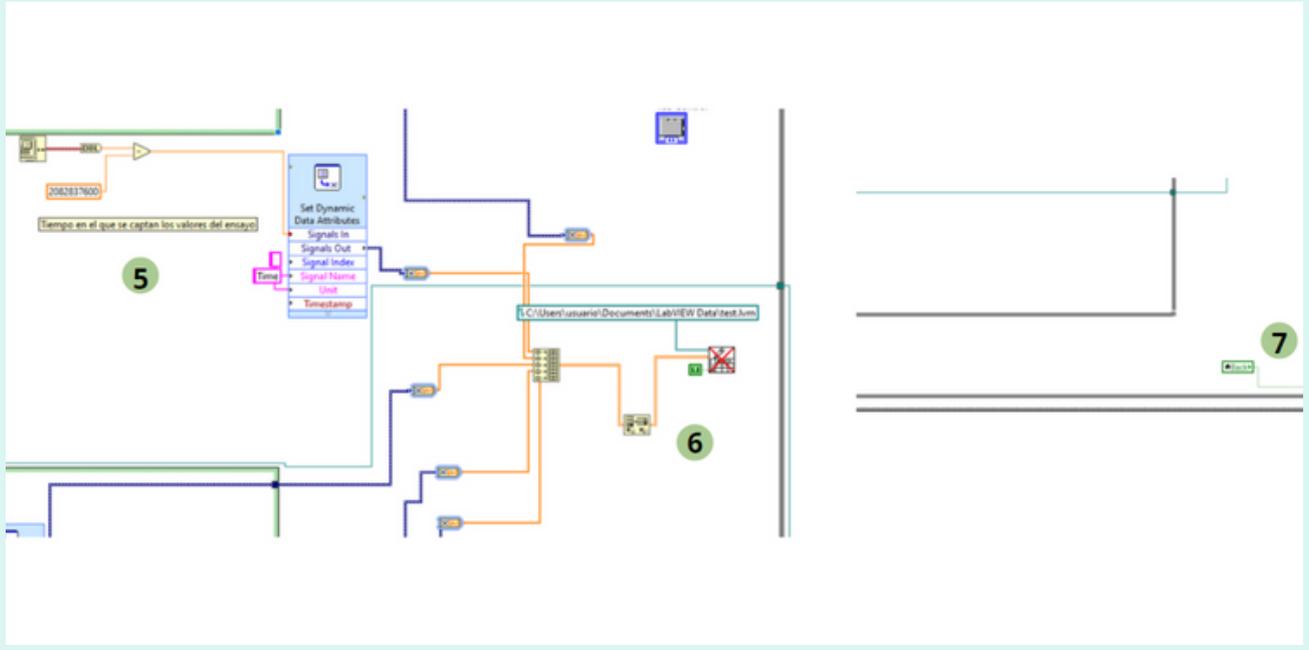
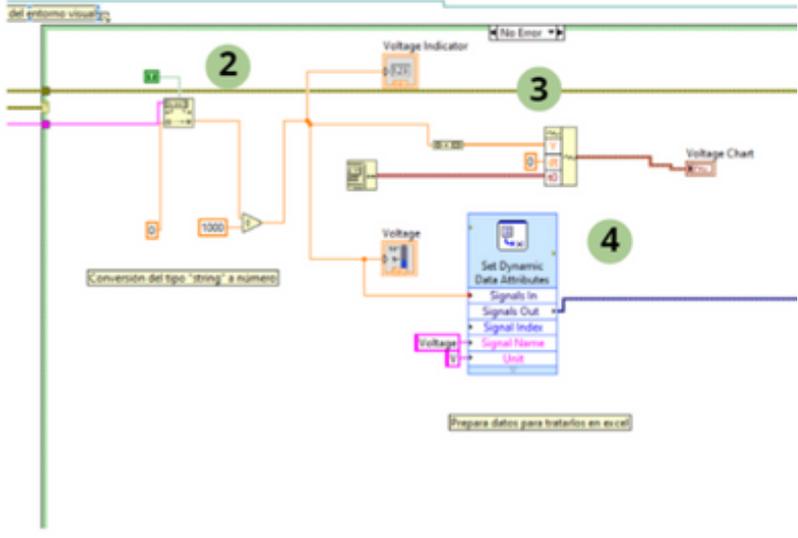
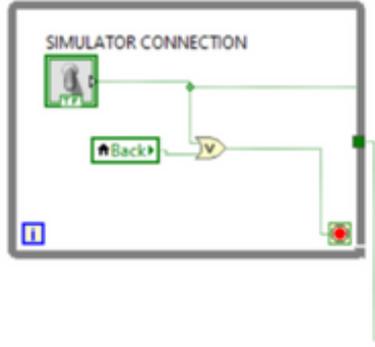
Der Datenempfang besteht aus dem Öffnungsblock der UDP Versandsprotokoll, wo der lokale Port zu verwenden zugeordnet wird; ein Leseblock, der das Signal empfängt, eine Reihe von Signalbehandlungblöcken und ein Block für den Schliessprotokoll.

Wie in den restlichen Fenster enthält dieses auch Fehleranzeigen, die vom Fehlertab benutzt werden.

Einige Modifizierungen könnten sein: den Datenbanktyp, die verwendeten Ports oder den Typ der empfangenen Variablen zu verändern; sowie die Art der angewendeten Indikatoren und die Frequenz der Datensammlung.

- 1** Start-Loop
- 2** Datenempfang
- 3** Indikatoren
- 4** Datenvorbereitung
- 5** Zeit
- 6** Schreiben von Daten
- 7** Zurück-Button

1 SIMULATOR CONNECTION



SIMULATOR "BLOCK DIAGRAM"

Dieser "block diagram" vereint Aspekte, die schon in den "breaker" und "real time" Registerkarten verwendet werden.

Es handelt sich um eine grosse Schleife while, in der sich zwei simultane Aufgaben erfolgen: die Datenübermittlung und der Datemempfang via UDP. Dafür wird das typische Schema "Öffnung, Versand und Schliessung" oder "Öffnung, Empfang und Schliessung" verfolgt wird.

Sowohl beim Versand als auch beim Empfang verfügt man über eine Fehleranzeige, die vom "error log" Seite benutzt wird. Ihrerseits, setzt der Empfang mit einem Grafikdarstellungssystem fort.

Es wurde einen Booleschen Operator angewendet, um den Nutzer die Möglichkeit zu geben, die Variable grafisch darzustellen oder nicht. Dieser wird von einer Tabelle begleitet, wo diese abgebildet sind.

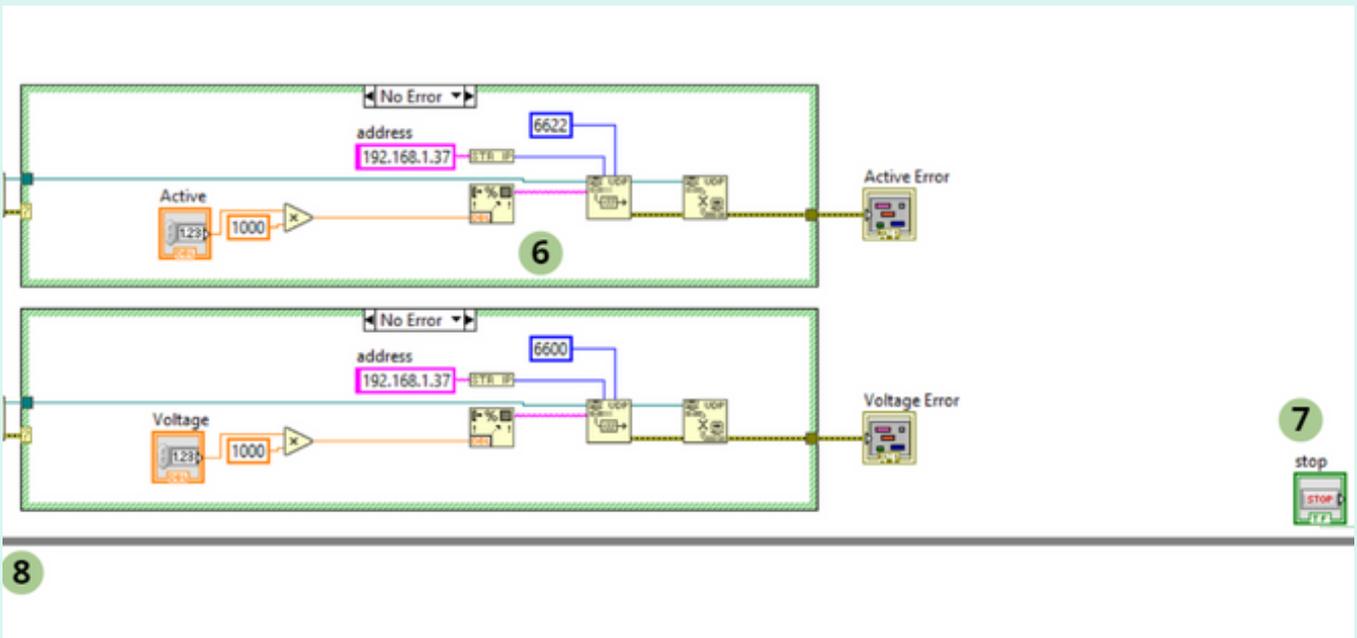
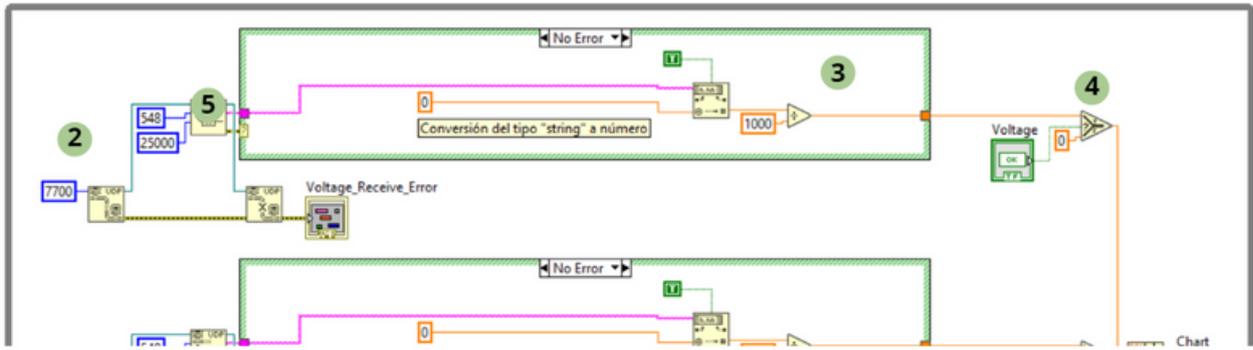
Wie auch in den restlichen Registerkarten verfügt man über eine Zurück-Taste, die den User zum Hauptmenü sendet.

Die Versand- und Empfangvariablen, als auch die Indikatoren, sind je nach Programmiererwünsche vollständig veränderbar.

- 1** variablenneustart
- 2** UDP Öffnung
- 3** Datenempfang
- 4** Tasten
- 5** Grafik
- 6** Versandstruktur

- 7** "Back"-Button
- 8** Schleife while

1



DATENHISTORIE

Diese Programmfunktionalität besteht aus zwei zusätzlichen Registerkarten: die erste ermöglicht, das Datum zu wählen, und die zweite stellt die Daten dar.

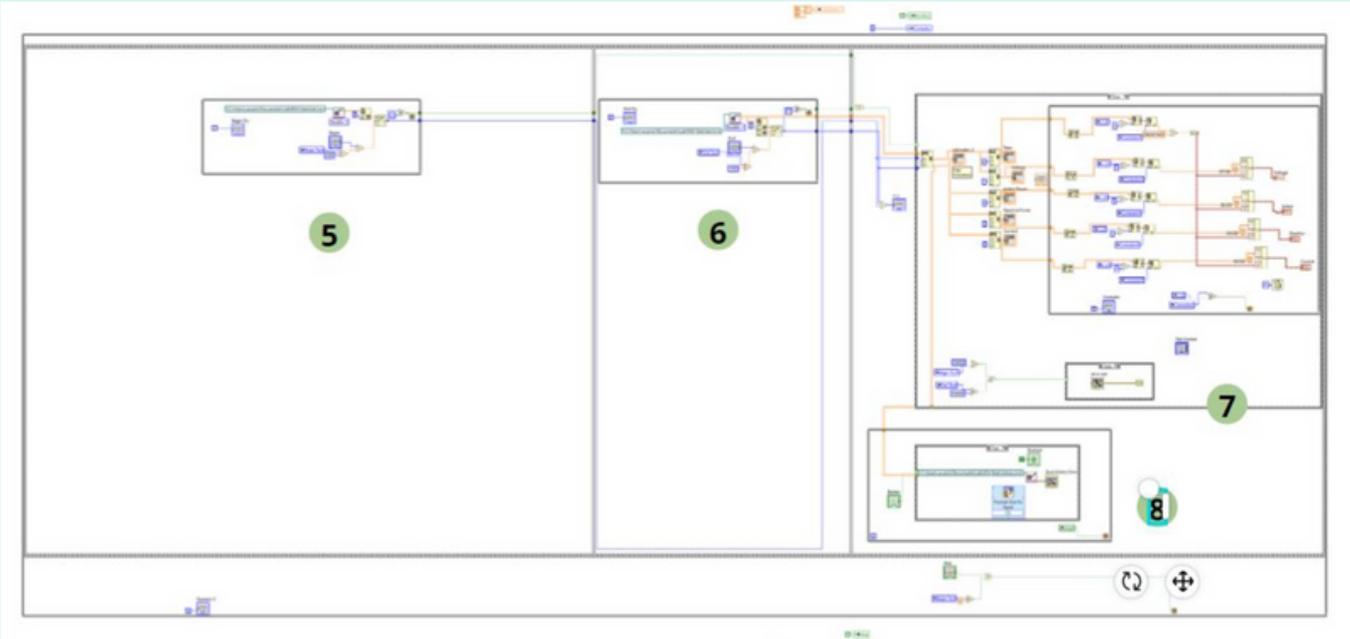
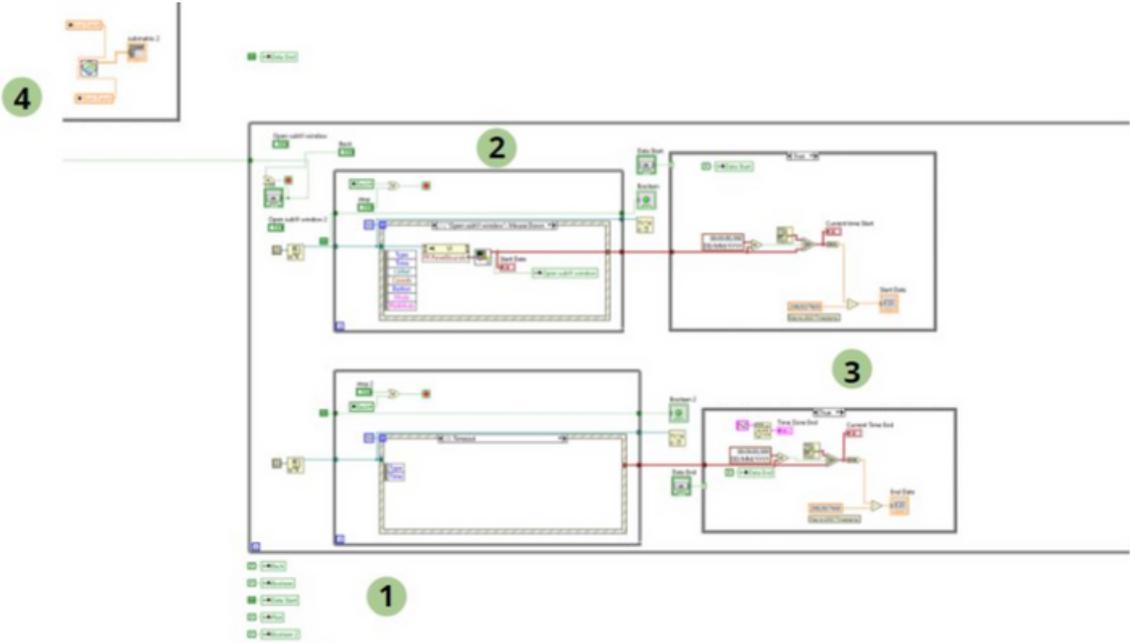
Für die Datenauswahl verfügt man über eine grosse Schleife `while`, in der zwei Konditionalblöcke zu finden sind, die sich ausführen, wenn das Datum ausgewählt und bestätigt wird. Die grosse Schleife `while` endet, wenn diese zwei Parameter erfüllt sind, und der Nutzer auf den Darstellungstaste drückt.

In diesem Moment öffnet das Programm den anderen Tab. Dieser zweite Fenster besteht aus einer Sequenz: zwei Schleifen `while`,

wo die Software die näheren Daten sucht und zwei letzten Schleifen `while`. `dónde el software busca los datos más cercanos y dos últimos bucles "while"`. Die erste ist aus Matrixbehandlungsblöcken für die grafische Darstellung gebildet, und die zweite aus einer Bedingung für die Aufbewahrung der dargestellten Daten; alles in einer grossen Schleife `while`, die sich ausführen wird, bis den User anders hinweist.

Dieser ist der entwickelste Code der Software und jede Veränderung verlangt viel Vorsicht, damit das Ensemble nicht verstellt wird. Man empfiehlt, keine Änderung ausser der möglichen Modifizierung der Datenbank durchzuführen.

- | | |
|--------------------------------------|---------------------------|
| 1 Variablenneustart | 7 Darstellung |
| 2 Schleife <code>while</code> | 8 Datenspeicherung |
| 3 Konditionalstrukturen | |
| 4 Öffnung des zweiten Tabs | |
| 5 Startsuche | |
| 6 Endsuche | |



AUTOMATISCHER SCHALTER

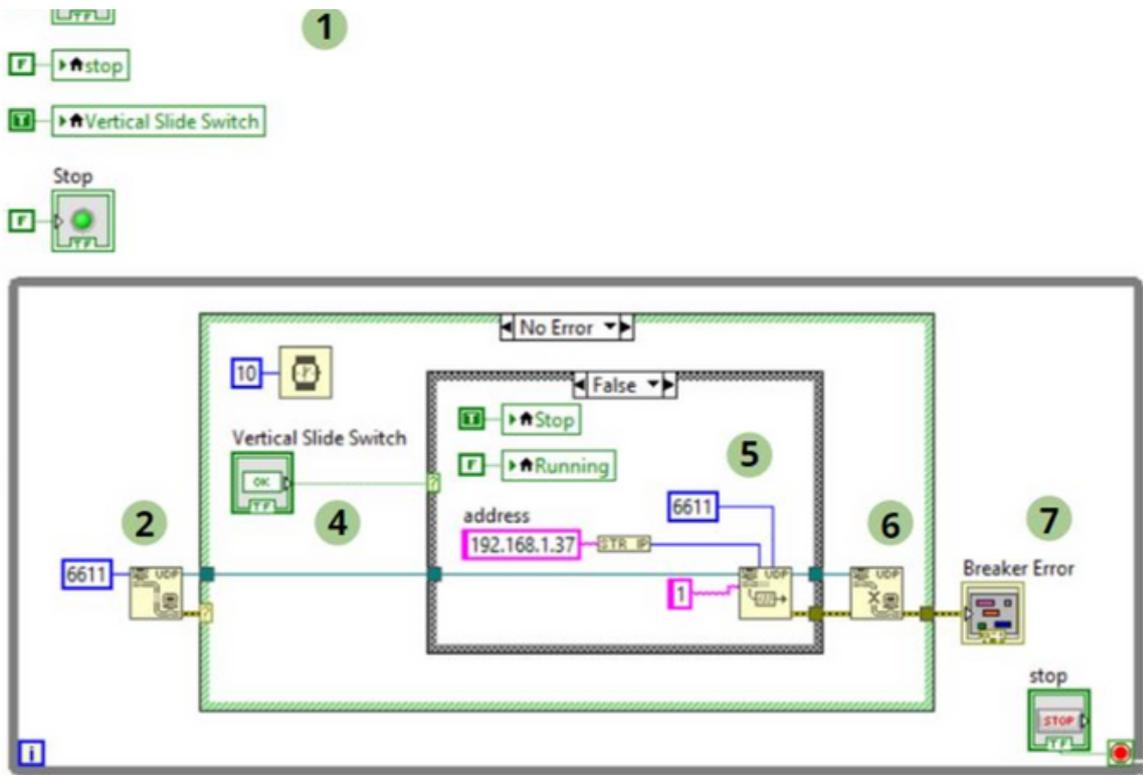
Der Programmiercode dieses Fensters besitzt einen Variablenneustart (die zwei LEDs enthalten) und eine Schleife while, die auf gleiche Weise wie die vorigen, sich abspielt, bis die Ausgangsbedingung eintritt.

In diesem Loop findet man einen Versand via UDP Protokoll: zuerst öffnet man das Protokoll, dann teilt man einen Port zu (in diesem Fall 6611).

Wenn der Prozess ohne Probleme erfolgt, wird die Konditionalstruktur validiert. Dort gibt es den Schalter und, nach seiner Position, wird einen oder anderen Teil der Konditionalstruktur geöffnet. Eine 1 wird gesendet, wenn der Kreislauf geöffnet ist oder eine 0, wenn im Gegenteil er geschlossen ist.

In dieser letzten Struktur hat man den Versand, der stattfinden wird, nachdem der Eingangport und die Adresse (die von numerischer Information zum "string" umwandeln muss) eingegeben sind.

- 1** Variablenneustart
- 2** UDP-Öffnung
- 3** Schleife while
- 4** Schalter
- 5** Konditionalstruktur
- 6** Schliessung des UDP-Protokolls
- 7** Fehler
- 8** Zurück-Button



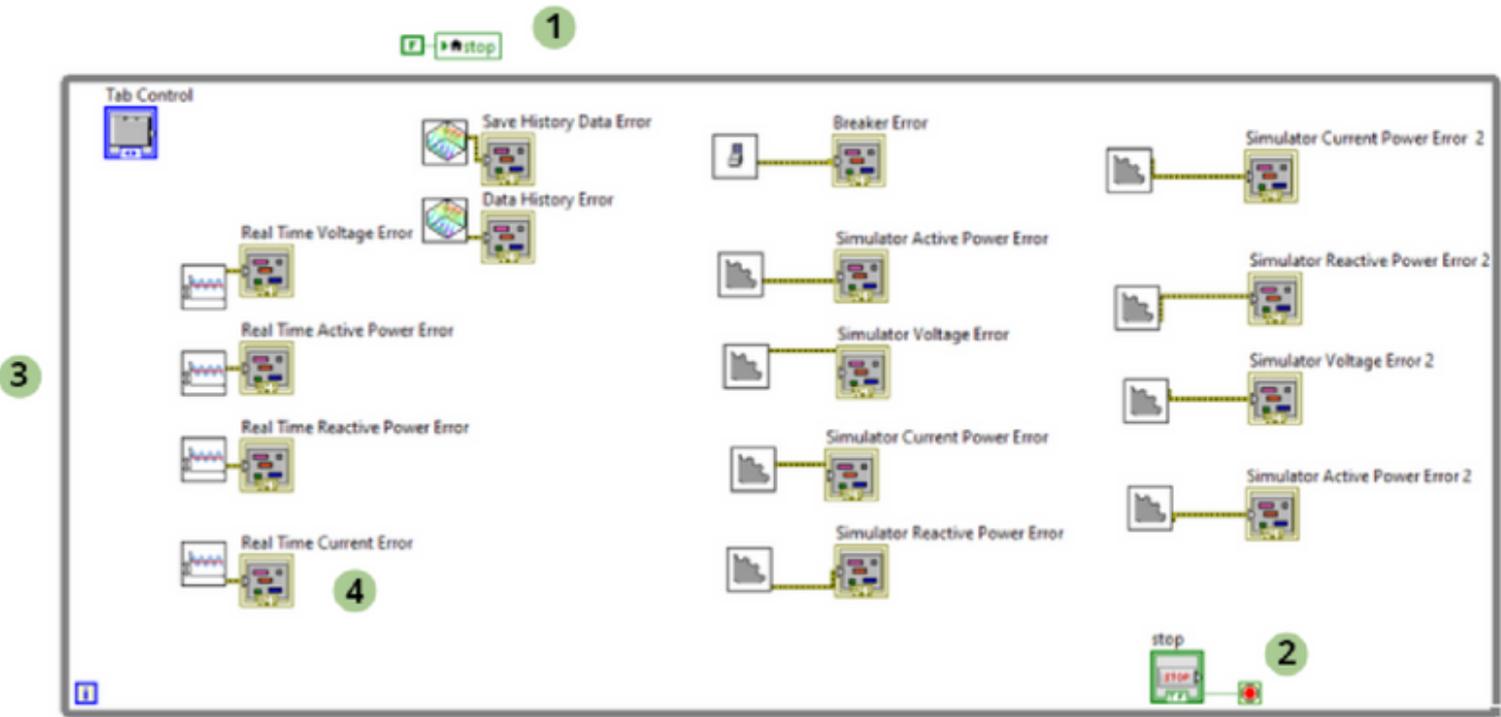
FEHLERCODES UND HÄUFIG GESTELLTE FRAGEN

Die Programmierumgebung der Fehlerregisterkarte ist nur eine Art "Repository", wo alle Informationen über Fehler in den restlichen Tabs gruppiert werden, und aus Bequemheitsgründen unsichtbar gestaltet ist.

Daher verfügt man über eine Schleife `while`, die sich wiederholt, solange die Ausgangsbedingung nicht erfüllt wird. Darin befinden sich die Fehler der verschiedenen Fenster, zusammen mit einer grafischen Anzeige für jede von ihnen.

Die einzige mögliche Modifizierung ist, mehr Fehleranzeigen hinzuzufügen. Dafür soll man mit der dargelegten Struktur fortfahren: Ruf an die Registerkarte + Indikator

- 1** Variablenneustart
- 2** Zurück-Button
- 3** Schleife `while`
- 4** Appell an die Registarkarten



FEHLERCODES UND HÄUFIG GESTELLTE FRAGEN

CODE

BESCHREIBUNG

#54

Die Netzwerkadresse ist falsch geformt. Stellen Sie sicher, dass die Adresse ein gültiges Format hat. Für TCP/IP, kann die Adresse einen Maschinennamen oder eine IP-Adresse im Format xxx.xxx.xxx.xxx. Falls der Fehler auftritt, wenn ein Maschinennamen bestimmt wird, vergewissern Sie sich, dass der Maschinennamen gültig ist. Versuchen Sie, einen Ping-Test für den Maschinennamen durchzuführen. Überprüfen Sie, dass Sie einen richtig konfigurierten DNS-Server besitzen. Wenn Sie im Moment die Funktion Öffene Verbindung TCP verwenden, stellen Sie sicher, dass der Wert auf den Remoteport oder Servicenamen nicht 0 ist.

#55

Der Netzwerkbetrieb ist gerade im Gange. Falls Sie diesen Fehler erhalten, während Sie die UDP Write Funktion benutzen.

#56

Der Netzwerkbetrieb hat die Zeitgrenze überschritten, die der Nutzer oder das System vorgegeben hat.

#57

Die Netzwerkverbindung ist besetzt.

#58

Die Netzwerkfunktion ist mit dem System unvereinbar.

#59

Probleme mit der UDP Schrift im aktuellen Fenster.

#60

Die angegebenen Port oder Netzwerkadresse werden gerade gebraucht. Wählen Sie einen verfügbaren Port oder Netzwerkadresse.

#61

Das System hat verursacht, dass die Netzwerkverbindung abgebrochen wurde.

#62

Der Server hat die Netzwerkverbindung abgelehnt. Für TCP/IP

#1011

Man findet keine Daten für das ausgewählte Zeitintervall.

#44

Problem beim Versuch, eine Datei zu retten

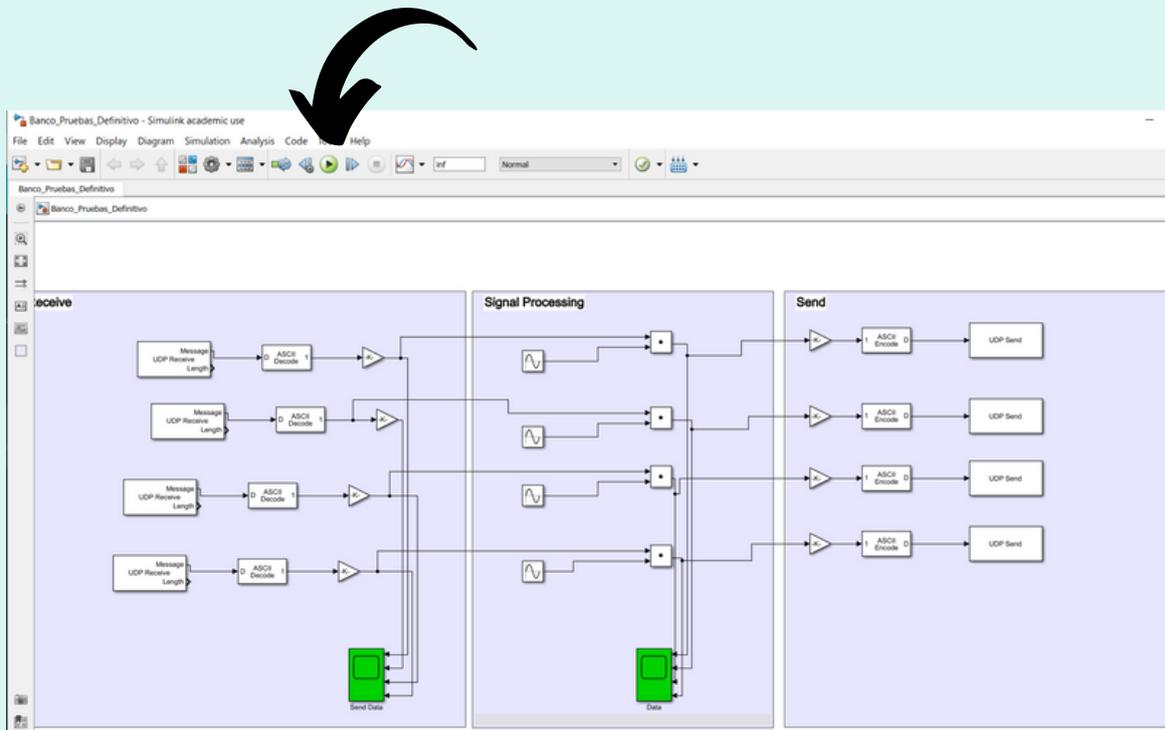
FEHLERCODES UND HÄUFIG GESTELLTE FRAGEN

Wie kann man den Code in LabVIEW ausführen?

Um einen Code in LabVIEW auszuführen, muss der Nutzer auf den "run"-Button im "log-in" Fenster drücken, denn dieses ist das Hauptfenster des Programms. Wenn Sie mehr über LabVIEW erfahren möchten, besuchen Sie folgende Seite: <https://www.ni.com/getting-started/labview-basics/esa/>

Wie kann man den Code in MATLAB ausführen?

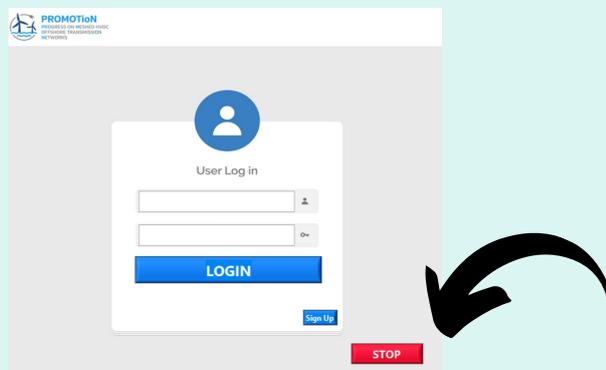
Um den Code in MATLAB auszuführen, genauer gesagt im Simulink, muss der Benutzer auf den "run"-Button drücken, wie man auf der folgenden Abbildung beobachten kann.



FEHLERCODES UND HÄUFIG GESTELLTE FRAGEN

Wie kann man den Code in LabVIEW beenden?

Das Programm ist so eingestellt, dass es beendet werden kann, ohne auf die "abort" Taste zu klicken. Dafür muss der Nutzer zum "log-in" Fenster zurückgehen, und auf Abschluss-Button des Programms drücken.



Wie kann man die UNIX Timestamp-Zeit zum echten Datum machen?

Um diese Umstellung durchzuführen, sind online Konverter wie <https://www.unixtimestamp.com/index.php> empfohlen

Was kann man tun, wenn Probleme während der Anmeldung geschehen?

Falls die Anmeldung Probleme ergibt, verfügt man wahrscheinlich nicht über eine gute Netzwerkverbindung, vermutlich wurde die E-Mail Adresse schon vorher verwendet. Um diese zu lösen, kann man entweder mit einer anderen E-Mail Adresse versuchen, oder die Verbindung checken.

Für weitere Fragen über das Programm können Sie an die E-Mail Adresse juyusta@etsii.upv.es schreiben

Parte VII

Anexos

Capítulo VII.1

Breve historia de la energía eólica

En este capítulo de ampliación se dispone de un breve resumen de la historia de la energía eólica desde sus inicios hasta su situación hoy en día, destacando sus avances más importantes y sus situación en España.

Índice

VII.1.1 Breve historia de la energía eólica.	419
--	-----

VII.1.1 Breve historia de la energía eólica

El inicio de este sistema de generación de energía, tal y como lo se conoce hoy en día, data del 1887, cuando, James Blyth -un profesor del Anderson's College en Glasgow- decidió combinar el tradicional molino de eje vertical con un generador eléctrico de la época.

Este sería el comienzo para un viaje de alrededor de 150 años que acabaría con el desarrollo de lo que hoy en día se entiende como aerogenerador. Este centenar de años ha convertido la energía eólica en una pieza fundamental del panorama energético internacional; proporcionando electricidad en más de 90 países en todo el mundo.

Después de poner en marcha su turbina en julio del 1887, Blyth emitió un escrito al *Philosophical Society of Glasgow*; en el mismo, Blyth describía su prototipo como “un diseño en forma de trípode con 33 pies de altura, cuatro palas de 13 pies cada una fabricadas de lona y una dinamo”.

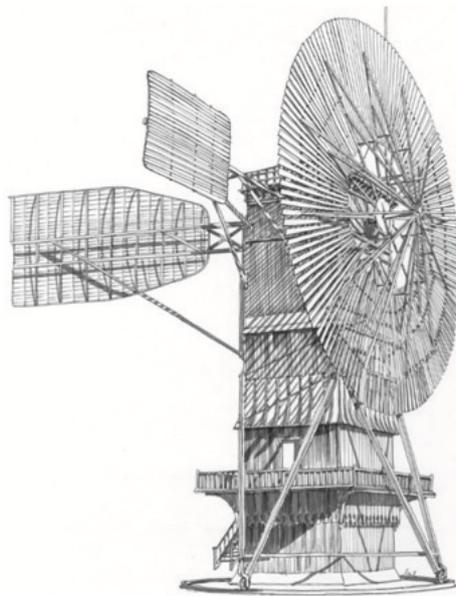


Figura VII.1.1: Boceto de turbina realizado por Brush(windstory)

Al mismo tiempo, al otro lado del océano, otro ingeniero estaba investigando también en este sistema de generación de energía. Para producir energía para su casa en Cleveland, Ohio; Mr. Charles F. Brush, diseñó un pequeño aerogenerador en el patio trasero de su vivienda.

Unos años más tarde, el meteorólogo danés Poul la Cour estaba a punto de lograr un gran avance en el mundo de la energía eólica. En 1891, una vez conocidos los estudios de Brush y Blyth; fabricó su propio aerogenerador.

El aerogenerador danés estaba conectado a dos generadores de 9 kW de corriente continua. En este prototipo, la Cour introdujo la primera solución conocida hasta entonces para tratar las fluctuaciones naturales de viento. Estas fluctuaciones se habían convertido en uno de los principales baches para el desarrollo de este sistema en la época y en el futuro, ya que modificaban notablemente la frecuencia del voltaje de salida.

Para paliar dicho efecto adverso, la Cour fabricó un segundo eje para el molino y lo conectó al eje principal mediante un sistema de poleas. Dicho invento fue bautizado por el mismo la Cour como “Kratostate”.



Figura VII.1.2: Diseño de aerogenerador ideado por la Cour (**windstory**)

Durante sus estudios, la Cour diseñó un nuevo y revolucionario método para almacenar la energía producida por su invento. Inspirado por los estudios de eruditos italianos de la época, la Cour fabricó un sistema que almacenaba la energía directamente en tubo de agua, donde el oxígeno se separaría del hidrógeno mediante un proceso conocido como hidrólisis. Este segundo invento sentaría las bases para lo que hoy se conoce como pila de hidrógeno y para el desarrollo del vector hidrógeno en términos generales.

Esta elegante solución se torció rápido cuando el propio la Cour se dio cuenta de que se originaban problemas de explosión, debidos a que las partículas de oxígeno no conseguían separarse completamente de las de hidrógeno.

De aquellos primeros prototipos y conceptos de aerogenerador del S.XIX, se inició una época donde primaron aquellas turbinas pequeñas, que abastecían granjas y casas individuales; generalmente por ser emplazamientos donde no llegaba la red eléctrica general.

Para contextualizar las condiciones históricas donde se pudieron desarrollar, debe quedar constancia de que la configuración de la industria eléctrica en los inicios del S.XX desterró a la energía eólica casi por completo del panorama al adoptar un modelo central de producción y distribución a partir de turbinas de vapor que favorecía a grandes compañías, quienes contaban con el favor de las autoridades. Por tanto, todas aquellas innovaciones o alternativas que no encajaran en este marco único no tenían cabida.

Así se desarrolló la electrificación en estas décadas en Estados Unidos y Europa, quien imitó el mismo modelo, mientras que los entusiastas en la materia trataban de completar la transición de los molinos de viento, que por su funcionalidad giraban lentamente, hacia la turbina eólica moderna, que debía alcanzar velocidades muy superiores para ser capaz de generar energía.

La principal innovación corrió a cargo de Stuart y Fales en 1917, quienes implantaron aspas con un perfil más aerodinámico, cuya importancia reside en su misma forma. Junto a un ángulo de ataque óptimo entre 10 y 15° y aspas trenzadas desde su base hasta la punta, este diseño logra maximizar el output eléctrico porque incrementa su velocidad nominal. Mediante esta implementación patentada (casi 10 años después) y su perfeccionamiento, se obtienen los primeros aerogeneradores modernos.

Gracias a este progreso y a la iniciativa empresarial, en los años 20 compañías como HEBCO y más adelante Jacobs Wind Electric Company con su modelo de 3 aspas se introdujeron en la producción y comercialización de turbinas eólicas de pequeño tamaño para abastecer a aquellos granjeros cuyas propiedades no carecían de acceso a la red eléctrica general. La geografía estadounidense, junto a la disposición del servicio eléctrico focalizado en áreas urbanas, dejaron ese nicho abierto. El nuevo invento encontró además un uso popular para recargar las baterías de las radios. Aunque la empresa de los hermanos Jacobs se posicionó a la vanguardia absoluta de la tecnología por su capacidad de inventiva, el coste de la obtención de electricidad a partir de esta fuente no era competitivo. Esto se debía a que se había seguido trabajando en ampliar, mejorar y optimizar las turbinas a vapor, de las que se alimentaban los holdings eléctricos; a mayor velocidad. La opción imperante llevaba la delantera, y los progresos en la energía eólica no serían suficientes, por lo que efectivamente quedaba relegada a cubrir los vacíos del sistema eléctrico general, y no por mucho tiempo más.

Mientras tanto, en el continente europeo, Kurt Bilau proseguía con los avances al aplicar sus conocimientos en la reparación de molinos de viento tradicionales después de la primera Guerra Mundial a las nuevas turbinas eólicas: construyó su prototipo en 1921, aunque presentaba ciertas complicaciones. Para controlar el giro con fuertes vientos, y facilitar el inicio del movimiento desde la posición de reposo; inventó las "Ventikanten", aspas abatibles adaptables a las condiciones meteorológicas. Pese a su efectividad, no encontró mercado en una Alemania sumida en una fuerte crisis económica. Desde la vertiente teórica, Albert Betz definió el "límite Betz" en varias publicaciones: se trata del límite superior de la eficiencia de una turbina, la máxima energía proveniente del viento que es capaz de extraer y convertir. Ascende a 59.3%. También asienta otras bases teóricas y emplea por primera vez vocabulario común para hablar de energía eólica.

También la URSS, en la búsqueda de soluciones para satisfacer sus necesidades energéticas, se embarcó en esta nueva vía. Sus esfuerzos de investigación se materializaron en 1931 en la turbina Balaclava emplazada en Crimea, de 100kW: el primer aerogenerador conectado a la red, en todo el mundo.

Esta primera fase de desarrollo de las turbinas eólicas después de su despunte finalizó cuando el presidente Roosevelt se comprometió, por un lado, a acabar con los abusos de las compañías eléctricas, y por otro a erradicar el desempleo fruto de la Gran Depresión. Combinó estos objetivos en el "Public Utility Holding Company Act" (1935) y la Ley de Electrificación Rural de 1936, por las que primero las eléctricas quedaron bajo la autoridad regulatoria de la Comisión Federal de Energía, manteniendo su trato previo, operar en un entorno protegido donde obtener rentas fijas a cambio de dar servicio a los clientes; y después se animó el surgimiento de cooperativas eléctricas rurales que vendieran la electricidad producida en plantas públicas a clientes en ubicaciones remotas a precio de coste.

Varios préstamos e incentivos para expandir la red de transmisión hasta las residencias de los nuevos usuarios concluyeron la electrificación rural estadounidense en torno a los años 50, a cambio de que esas granjas desmantelaran sus turbinas eólicas.

Con ello, desaparecen las opciones para esta fuente de energía a nivel comercial para ser instalada en fábricas individuales y hogares, y solo les quedaba una posible transformación como equipamientos de gran escala, compatibles con el modelo de estación central y el sistema de transmisión AC.

Tomó el relevo del desarrollo Alemania, donde se planteaba diversificar sus fuentes de obtención de energía más allá del carbón, para poder exportarlo. El ingeniero Hermann Honnef tenía grandes visiones para el futuro, como instalaciones de enormes parques eólicos marítimos, pero dadas las posibilidades del momento resultaban inviables. Ante la controversia que despertaban sus ideas y sin el apoyo de otros estudiosos, se limitó a aportar innovaciones más realistas.

La historia de la energía eólica sufrió un nuevo bache tras el breve periodo de entreguerras, pues la segunda Guerra Mundial paralizó de nuevo todo el desarrollo, e incluso se retrocedió, ya que el conflicto bélico exigía una reorientación de la industria. Ni siquiera en Estados Unidos la posguerra supone un ambiente favorable, hasta que llega la prosperidad en la década de los 50.

Ante un increíble despegue en la demanda de electricidad en los domicilios, se originó la necesidad de una oferta a la altura. En la Comisión Federal de Energía, Percy Holbrook Thomas y otros investigadores valoraban la posibilidad de relanzar un plan de investigación en energía eólica que sirviera a los 2 objetivos energéticos nacionales prioritarios: mantener los precios de la electricidad bajos, y hacer llegar el suministro a todas las áreas rurales. Así, la expansión que acabaría con las pequeñas turbinas eólicas de las granjas y su comercialización abría a su vez una oportunidad para la energía eólica a gran escala. La inversión total del proyecto de aerogeneradores de 7,5MW se estimaba en 5 millones de dólares, muy por debajo de turbinas antecesoras y por tanto una cifra que sí rivalizaba con las propulsadas a vapor, pero aun así no se le concedió financiación pública al proyecto.

En Reino Unido en cambio, se avivó un interés que se canalizó en un comité específico dentro de la Electrical Research Association. E.W Golding lideraba un programa en el mismo sentido, para autoabastecer energéticamente el país sin depender de las importaciones de combustibles. Se erigieron 2 turbinas experimentales que acabaron fracasando pese al esfuerzo que se puso en arreglar inconvenientes y el gobierno se retiró de la experimentación con aerogeneradores a gran escala, porque el progreso se desenvolvía lentamente y no reportaba frutos económicos suficientes para seguir invirtiendo.

Tras un parón, revivió la idea de la autosuficiencia energética gracias al informe de Dixy Lee Ray, y con ella la investigación en energía eólica, unas décadas después al otro lado del océano con la crisis mundial del petróleo de 1973. Los avances de la NASA en el proyecto Culebra se complementaron con la construcción de una turbina de 100kW para retomar la actividad, y finalmente sus impulsores lograron convencer al Congreso para que les otorgara la financiación necesaria: en 1974, se destinaron 1,8 millones de dólares al proyecto. Con este flujo monetario entrante, por fin existía una base económica para organizar seminarios y reuniones de grupos de expertos donde compartir conocimientos. En ellos, Ronald L. Thomas y Joseph M. Savino, analizaron varios diseños anteriores de turbinas eólicas y dedujeron 2 vertientes efectivas:

- La propuesta de Ulrich Hütter (100kW): un aerogenerador de giro rápido, situado a sotavento, con 2 aspas y control de inclinación; construido con materiales ligeros. Su ejecución era notablemente mejor que la de su competidor, aunque presentaba un problema fundamental, las cargas cíclicas/alternantes.
- Por el contrario, el modelo de Johannes Juul (200kW) giraba más lentamente, estaba emplazado a barlovento y se controlaba mediante detención pasiva. Las 3 aspas estaban específicamente diseñadas para paralizar el funcionamiento cuando soplaran vientos fuertes. Su fuerte residía en la fiabilidad, más que en la aerodinámica o la estética.

En la actualidad, el aerogenerador tipo parte de la base de Juul, pero incorpora elementos ligeros como fibra de vidrio o compuestos de carbón para potenciar el desempeño. Aun así, en esta época no se buscaba aunar lo mejor de ambas alternativas, sino escoger la mejor y trabajar en ella. Finalmente, triunfó en la disputa la turbina de Hütter, que se prefería por su baja velocidad de arranque y su rendimiento global.

Por tanto, se partía de estos prototipos pequeños, empezando desde los 100kW, para poder aprender de ellos y sus deficiencias, mejorarlos; y paulatinamente ir construyendo aerogeneradores más potentes conforme se fueran adquiriendo conocimientos y experiencia. Se buscaban las turbinas gigantes como objetivo final, por su competitividad en costes: las inversiones en estos equipamientos solo salen rentables si se reparten entre una producción de millones de kWh.

Dinamarca reaccionó de una forma similar a la coyuntura de 1973, dada la necesidad imperiosa de la calefacción en los meses más fríos. Los primeros planes gubernamentales fruto de esta nueva mentalidad de convertirse en una nación energéticamente independiente apostaban por la energía nuclear, pero se cancelaron en 1985 por la fuerte oposición popular a esta fuente energética. Paralelamente, entre 1974 y 1979 se forjó un fuerte compromiso desde todos los sectores de la sociedad danesa hacia la reinención que posibilitó el posicionamiento del país como potencia mundial en energía eólica. Fueron por tanto los mismos ciudadanos quienes alcanzaron el propósito gubernamental, quien trató de proveer bases académicas y también técnicas (Risø) donde apoyar el desarrollo. Aquí, se tomó el modelo de Johannes Juul como punto de partida.

1. Por ejemplo, Cristian Riisager y Erik Nielsen, tomando la hoja de ruta establecida en la conferencia de las Naciones Unidas, construyeron pequeños aerogeneradores con los materiales que hallaban a su disposición. Por suerte, se toparon pronto con Torgny Moller, un periodista entusiasta de la energía eólica quien se convirtió en el mayor prescriptor de su trabajo. Gracias a su excelente publicidad, recibieron un aluvión de pedidos hasta la entrada en el mercado de los grandes fabricantes.
2. Otro caso muy ilustrativo de este movimiento social es la historia de Tvind, el proyecto de Mogens Amdi Petersen de erigir un modelo alternativo de escuela, que también pretendía ser autosuficiente en el sentido energético: la comunidad de profesores y estudiantes fabricó conjuntamente el aerogenerador más grande operativo hasta la fecha (2MW) en 1978, sin apenas conocimientos previos en la materia y partiendo del diseño de Hütter. La proeza se cosechó un gran éxito por suponer una fuente de inspiración e innovación, además de publicar informaciones muy valiosas para el posterior desarrollo. A pesar de ello, no se pudo emplear para la comercialización ni el aprovechamiento de energía eléctrica a gran escala.

3. Svend Adolfsen, partiendo de los moldes de Erik Grove-Neilsen -componente estándar de todas las turbinas danesas-, consiguió que su diseño sea aprobado oficialmente al probar su buen funcionamiento y fiabilidad. La producción en masa no tardó en llegar.

La administración danesa también tomó partido, firmando acuerdos internacionales y con el proyecto práctico que erigiría las turbinas Nibe-A y Nibe-B. Sin embargo, las acciones de los fabricantes daneses individuales surtieron más efecto que estos grandes programas de investigación científica gubernamental, que seguían un enfoque demasiado reglamentario y formal que no les permitía progresar con la misma agilidad que quienes actuaban por su cuenta y en pequeños grupos.

Adicionalmente, en 1977 las circunstancias políticas proporcionaron indirectamente otro empuje adicional a la industria mediante subsidios para inversiones en energía renovable y préstamos para los exportadores. Gracias a los incentivos, fabricar y exportar aerogeneradores se había tornado en un negocio muy lucrativo.

De este capítulo de reinención de energía eólica, surgieron 2 de los 5 mayores fabricantes mundiales de aerogeneradores: Vestas y Bonus Energy, que tras ser adquirida por Siemens y fusionada con Gamesa, se posiciona a la misma altura que el gigante Vestas. De hecho, se considera que la industria global nace con el lanzamiento del modelo V-10 de Vestas en 1979. Aun así, tomó tiempo que el resto de países siguieran la estela de Dinamarca, y más en que las empresas pudieran sobrevivir en base a una demanda mucho mayor de turbinas. En los años 80, las políticas estadounidenses asentaron la base perfecta para la prosperidad de las compañías danesas.

Esta época dorada terminó abruptamente con el colapso del mercado de California en 1986, y las pocas empresas en Dinamarca y Alemania que siguieron tratando de comercializar aerogeneradores tuvieron que luchar por su supervivencia; al tiempo que los múltiples intentos de establecer programas nacionales de investigación y desarrollo de energía eólica fracasaban reiteradamente en todos los países. El panorama estaba plagado de incertidumbre cuando la Comunidad Económica Europea se reunió en Madrid en los años 90 para poner en marcha los WEGA, instalaciones de testeo de turbinas de capacidad superior al megavatio. De nuevo, la utilidad de construir grandes aerogeneradores se ponía en duda, al no satisfacer nunca las expectativas de los inversores. Un nuevo estudio que confirmó la viabilidad de la energía eólica a gran escala mientras se emplearan materiales ligeros, fuertes y flexibles puso fin a la controversia, y los principales productores europeos (Bonus Energy, Enercon, Nordic, Vestas y WEG) colaboraron para erigir 5 aerogeneradores, agregándoles configuraciones de velocidad variable y convertidores de potencia que los hicieron más controlables y desarrollando herramientas digitales y estrategias para compatibilizar las turbinas con los requerimientos de la red. Así pues, el éxito real de este programa radicó en que potenció la experiencia y el desarrollo y consolidación de recursos en Europa.

Se eligió España como sede por su reciente incorporación al ámbito de la energía eólica también desde la primera crisis del petróleo, donde debió actuar rápidamente por su fuerte dependencia de estas importaciones. Desde 1975 se iniciaron las acciones para su sustitución: la transición energética. El primer parque eólico español albergó los primeros 5 prototipos en la Empordà en 1984, el mismo año en que Ecotènia puso en marcha su primer diseño, robusto y específico para los vientos de las montañas del país. Tras su expansión nacional y su incursión en el mercado internacional, Alstom adquirió la compañía como parte de sus acciones de revitalización, y su división Alstom Wind acabó integrándose en GE Energía Renovable. Made y la Corporación de

Energía Hidroeléctrica de Navarra también participaron en el despegue de la energía eólica en España, junto a Acciona Windpower.

Todas estas iniciativas se apoyaron en numerosas políticas energéticas favorables, tanto desde el gobierno central como los autonómicos. Las CCAA trataban de atraer productores de turbinas eólicas hacia sus regiones para el desarrollo industrial y crecimiento económico, aunque algunas apostaron fuertemente por la fabricación por parte de compañías locales, antes de conceder licencias externas. Así se explica la oportunidad de mercado que vio Gamesa para acudir a Vestas, que ya estaba recuperada del batacazo del mercado californiano y en 1994 se hallaba a la vanguardia, y proponerle una joint venture. La compañía danesa aspiraba a ser la mayor empresa energética moderna y ya exportaba turbinas a países por todo el mundo. Sin embargo, el contexto español presentaba fuertes restricciones, y en eso consistía la oferta de Gamesa: una vía de entrada en el país, tomándole como socio estratégico. Surge así Gamesa Eólica, gracias al acuerdo de derechos exclusivos para emplear la tecnología de Vestas en España, a cambio de acotar su expansión a los territorios de influencia tradicional española. Con los años, a Gamesa Eólica se le quedó pequeño el territorio acordado, su tecnología propia superaba a la heredada de Vestas y rompió su alianza con ellos al continuar creciendo por encima de las prohibiciones. Realmente Gamesa salió ganando en esta joint venture, pues a cambio de la participación en el regulado mercado español, acabó adquiriendo todo el know how tecnológico para despegar por su cuenta, mientras que a Vestas le surgió un nuevo y muy potente competidor global. Gamesa encontró un nuevo socio en Repower, juntándose con Siemens Wind Power en 2016 y conformando Siemens Gamesa Energía Renovable para competir contra Vestas como mayor fabricante mundial de aerogeneradores.

Actualmente, Iberdrola, quien permaneció pasivamente durante los 90 cosecha un éxito rotunda y consistente a partir de 2011 gracias a su subsidiaria Iberdrola Renovables, al convertirse en la principal operadora y propietaria de parques eólicos. Acciona Windpower se fusionó con Nordex, Acciona Energy permanece involucrada por su cuenta. Todos estos participantes que sitúan a España en el quinto lugar del mercado mundial se gestaron durante "el milagro eólico español", y así, mientras Dinamarca y Alemania se siguieron moviendo en su línea habitual, Europa se consolidó en el año 2000 a la cabeza de la energía eólica, con el 75 % de la capacidad global.

Tras las idas y venidas del S.XX, la energía eólica llegó por fin para quedarse como vía muy viable de energía eólica gracias a la maduración de la tecnología, sólidas fundaciones legales en Europa, reformulaciones en las políticas estadounidenses y la aparición de los mercados chino e hindú.

En Europa, la UE impulsa acciones gracias a sus directivas en claro favor de las fuentes de energía renovables, que determinan objetivos para los Estados miembros y a pesar de no ser vinculantes, proporcionan una hoja de ruta para que los países puedan contribuir a lograr el propósito comunitario, sin especificar los medios por los que podrían alcanzarse esas cifras. Ya fuera mediante incentivos fiscales o sistemas de cuotas, incluso la emisión de "certificados verdes" comercializables, del 2000 al 2005 la capacidad europea total se duplicó. A su vez, se incorporan los mercados en crecimiento de Estados Unidos, China y la India. El ritmo de esta evolución fue tan rápido que dio lugar a escasez de oferta, porque los fabricantes de aerogeneradores y proveedores de componentes no pudieron anticiparse pese a las previsiones del boom. Para adaptarse a este nuevo panorama, Vestas y NEG Micon se fusionaron y se vieron atraídas empresas tradicionalmente enfocadas hacia otros sectores como el conglomerado GE con la adquisición de Enron Wind o Siemens comprando Bonus Energy y conformando Siemens Wind Power. El progreso se frenó por

la crisis económica, pero permanece vigente la directiva de 2009 que marca un nuevo objetivo: 20% de energía obtenida de fuentes renovables para 2020. A partir de esa fecha, reina cierta incertidumbre, aunque se baraja la propuesta del Paquete de Energías Limpias de la Comisión Europea de 2016: 27% para 2030.

Paralelamente, se desarrollan y consolidan instalaciones que hasta el momento resultaban utópicas, siendo el mayor exponente los parques eólicos offshore. El primero de ellos, Vindeby en 1991, sienta precedente, aunque su auge deberá esperar hasta el S.XXI cuando el nivel tecnológico es capaz de exprimir su emplazamiento favorable, donde soplan vientos más fuertes y se deriva por tanto mayor capacidad de producción. Su posición en el mar también abre una senda para la continuación del crecimiento eólico, pues bastantes de las mejores ubicaciones terrestres están ya copadas.

Plantea asimismo varios desafíos, como construir enormes aerogeneradores resistentes al viento, las olas y la corrosión que deben persistir en buenas condiciones y ser sencillos de mantener; además de instalar los cimientos en aguas turbulentas y el cableado en el fondo marino para vincularlos a la red terrestre. Requieren mucho tiempo y elevadas inversiones iniciales por su colosal tamaño, pero una vez se salvan los desafíos comentados las plantas eólicas offshore son muy provechosas.

Los líderes en este ámbito son Reino Unido, con 5.1GW instalados y Siemens Gamesa, proveedor de los aerogeneradores offshore del mundo. De hecho, el 90% los 12.6GW de capacidad global se concentran en las costas inglesas, danesas y alemanas.

Sin embargo, ya no se trata solamente de un proyecto europeo y estadounidense: se ha exportado a China e India, y más tarde a Latinoamérica, África y Oriente Medio; 90 países con presencia eólica. Casos destacables son los de Marruecos, México y Brasil; con motivaciones de transición energética.

Gracias a las innovaciones más recientes, la energía eólica resulta la más barata en muchos casos. Aun así, crecientes cantidades incorporándose a las redes eléctricas presentan un desafío importante al que responder mediante nuevas soluciones tecnológicas.

Realmente, las ideas que se están poniendo en práctica, al igual que el concepto de los parques eólicos offshore que nace con Honnef, se contemplaron en el S.XIX por los visionarios e impulsores que comenzaron con la vía de obtener energía del viento.

- La primera, que tiene su origen con Brush: baterías integradas en las turbinas o situadas estratégicamente en la red de distribución para suavizar las fluctuaciones y garantizar un flujo de corriente más estable. Permiten incorporar la electricidad cuando hace falta en la red, y si la línea está saturada, la almacenan hasta que pueda ser aceptada. Aquí sobresalen los esfuerzos de Vestas y GE.
- La segunda, que ya llevaba a cabo Poul la Cour: el hidrógeno que se obtiene de un proceso de electrólisis, a partir del excedente de energía eólica. Su utilidad y aprovechamiento radican en que es otra fuente de energía en desarrollo, que puede transportarse fácilmente en gasoductos, o comprimirse y almacenarse. De hecho, a pesar de hallarse todavía en fase de testeo y estudio, ciertos expertos confían en las alianzas entre energía eólica e hidrógeno como el futuro del panorama energético.

Con todo esto, el camino recorrido por la energía eólica desde el S.XIX ha sido largo y ha debido superar altibajos y retrocesos, abriéndose paso en un modelo eléctrico donde se excluían las al-

ternativas y contra el que tampoco podía competir en precio. Se llegó a comercializar en ciertos momentos del tiempo para propietarios individuales, pero pronto se perfiló su destino final como objeto de investigación y de grandes proyectos públicos nacionales que pretendían bien posicionarse a la vanguardia de la técnica, o diversificar en fuentes energéticas para reaccionar a las crisis y no sobreexplotar sus recursos, o no depender tanto de las importaciones de combustibles fósiles.

Sin embargo, no fue hasta el cambio al S.XXI cuando despegó definitivamente, llevando la capacidad de producción, la tecnología, el panorama empresarial, las políticas nacionales y supranacionales al siguiente nivel y haciendo realidad las visiones de quienes dedicaron sus vidas a optimizar esta fuente de energía, en un contexto donde no existían recursos ni voluntad suficiente. Ahora, por fin el viento sopla a su favor.

Capítulo VII.2

Contexto energético

En este capítulo de ampliación se dispone de un análisis acerca de la situación energética a diferentes niveles y la posición de la energía eólica en cada uno de estos contextos.

Índice

VII.2.1 Situación energética	431
VII.2.1.1 Situación energética: Nivel mundial.	431
VII.2.1.2 Situación energética: Nivel europeo	433
VII.2.1.3 Situación energética: Nivel nacional	435
VII.2.2 Problema energético	436
VII.2.2.1 La sostenibilidad energética	436
VII.2.2.2 El Cambio Climático: Un problema Real	437
VII.2.3 Energías renovables	438
VII.2.3.1 Definición	438
VII.2.3.2 Clasificación	438
VII.2.3.3 Papel en el problema energético	438
VII.2.4 Energías Eólica	439
VII.2.4.1 ¿Qué es la energía eólica y como se produce?	439
VII.2.4.2 Determinación del emplazamiento de un aerogenerador	440
VII.2.4.3 Usos de la energía eólica	442
VII.2.4.4 Clasificación	442
VII.2.4.5 Aerogenerador.	444
VII.2.4.6 Ventajas e inconvenientes de la energía eólica.	446

VII.2.1 Situación energética

A lo largo de este punto se analizará, de forma objetiva, la situación energética a diferentes niveles, prestando especial atención al lugar que ocupan las energías renovables en dicha "fotografía energética".

VII.2.1.1 Situación energética: Nivel mundial

La población mundial continua con la tendencia al alza de los últimos ochenta años y se prevé que se mantenga, al menos hasta 2050. La mayor parte de este crecimiento se producirá en países subdesarrollados lo que generará un desequilibrio energético importante que contribuirá al que se conoce como "problema energético".

En la siguiente figura VII.2.1 se puede observar una previsión sobre la evolución de la demografía según el nivel de desarrollo de los países cada dos décadas⁷.

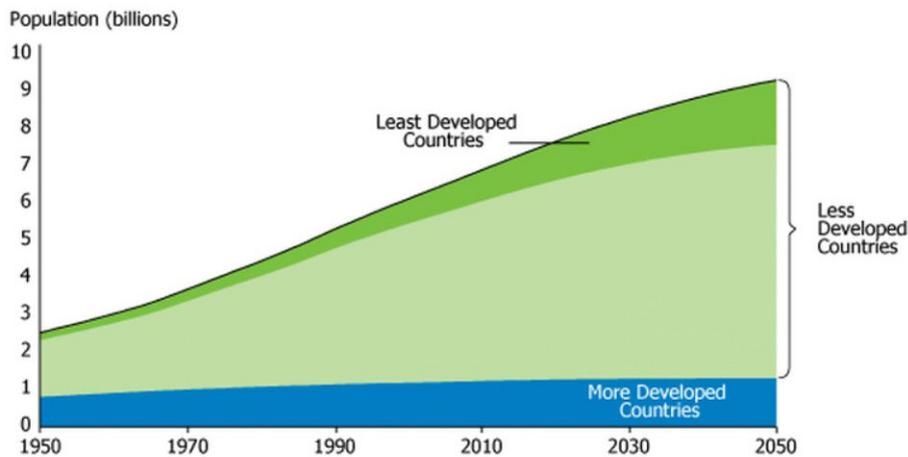


Figura VII.2.1: Evolución poblacional

El aumento poblacional irá acompañado de un incremento en la demanda energética; el objetivo de estos países menos desarrollados pasa ahora por convertir esta expansión en una oportunidad para reducir su pobreza y urbanizar zonas rurales.

Las figuras VII.2.2a⁸ y VII.2.2b⁹ muestran la diferencia de energía requerida por región mundial y de donde proviene. Se evidencia así el peso tan pequeño de lo que comúnmente se denomina como "energías limpias".

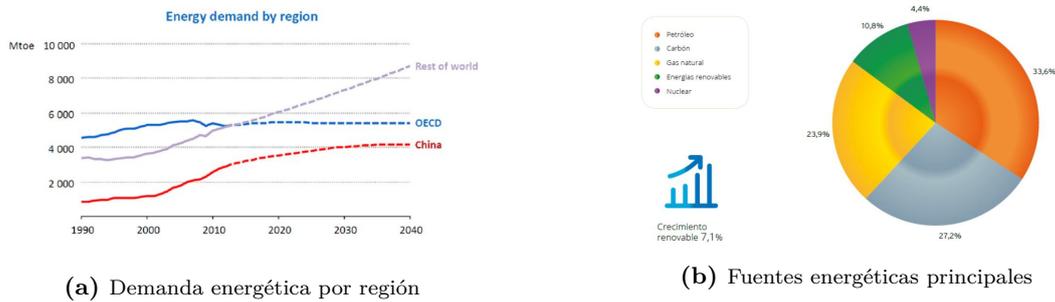


Figura VII.2.2: Infografía sobre el consumo y la demanda global

No existiría ningún problema con este panorama energético de no ser por el origen de toda la energía producida hoy en día. Más del 80 % de la demanda actual se cubre con energía generada a partir de combustibles fósiles, lo que supone un importante agravio a la hora de cumplir con los acuerdos energéticos de París y Kyoto.

Se prevé, por tanto, una gran expansión del modelo energético europeo-estadounidense, que irá sustituyendo en los próximos años a los obsoletos modelos atomizados de algunas naciones concentradas, sobre todo, en Asia y África.

Para finalizar se dispone de la infografía de Statistical Review of World Energy de BP, publicada anualmente. En ella se pueden comprobar los diferentes sistemas energéticos presentes en la actualidad VII.2.3. Por un lado, se observa el modelo de Oriente Medio, basado principalmente en los combustibles fósiles dada la abundancia de recursos petrolíferos en la zona; por otro lado, el modelo europeo que premia el consumo de energía por partes de fuentes renovables.

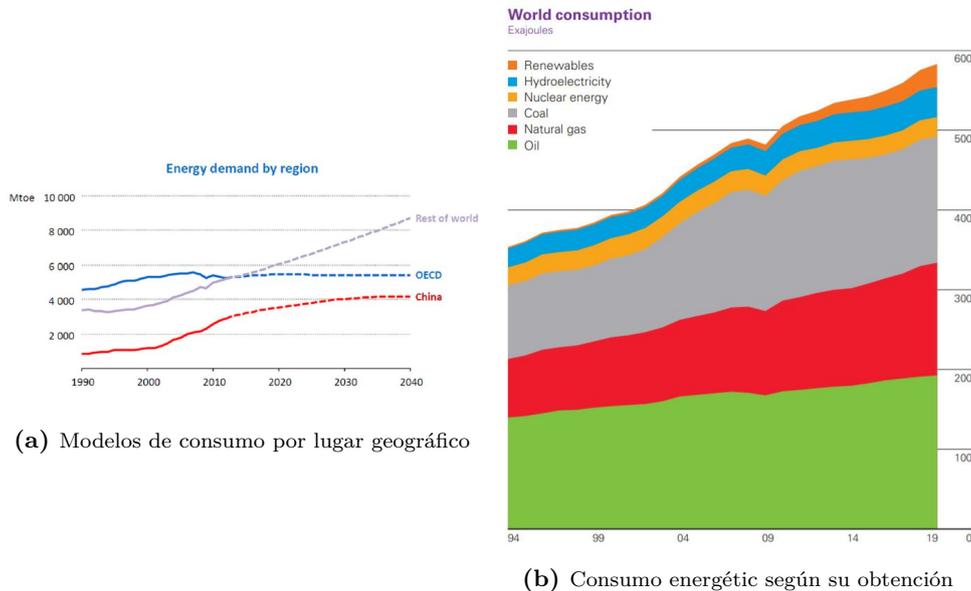


Figura VII.2.3: Recortes de la infografía energética publicada por BP

VII.2.1.2 Situación energética: Nivel europeo

El temprano desarrollo tecnológico de la Unión Europea en comparación con otros continentes ha desencadenado en un consumo energético estable en las últimas décadas, muy diferente, por ejemplo, a la evolución del continente asiático donde la demanda crece exponencialmente año tras año.

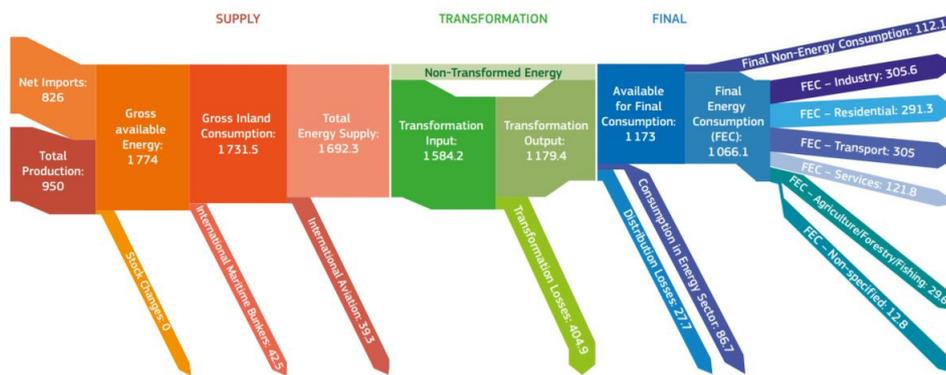


Figura VII.2.4: EU-28 Energy Flow 2000. Fuente: Eurostat, May 2019

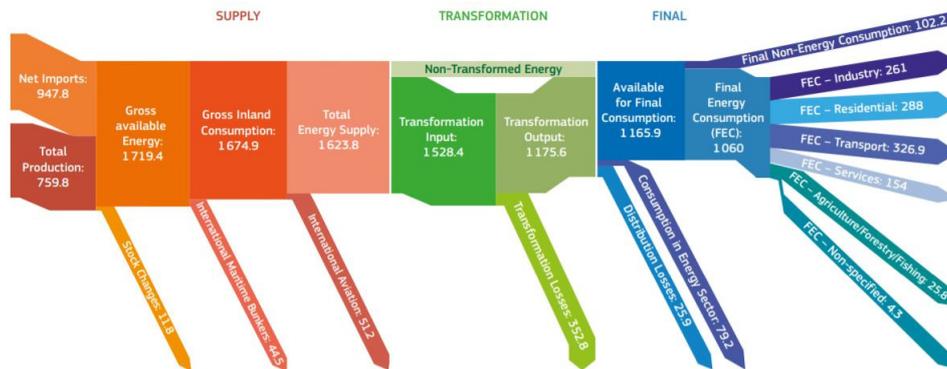


Figura VII.2.5: EU-28 Energy Flow 2017. Fuente: Eurostat, May 2019

Esta vinculación a los combustibles fósiles tiene su inicio en la primera y segunda revolución industrial cuando, con el objetivo de desarrollarse lo antes posible, los países europeos quedaron fuertemente ligados a este tipo de combustible.

Todo esto provoca que, si se continua con un escenario continuista, manteniendo la proporción actual de uso de las fuentes de energía para cubrir las necesidades de la UE hasta 2030, el continente europeo no cumplirá con el protocolo de Kyoto. A pesar de ello, los países que conforman la unión europea son los principales valedores de este protocolo iniciado en el año 2005.

Lo anterior se puede comprobar en las figuras VII.2.7a y VII.2.7b; en la primera de ella se pueden ver cuáles son las principales fuentes energéticas de la unión europea en el año 2000 y en la segunda en el año 2017.

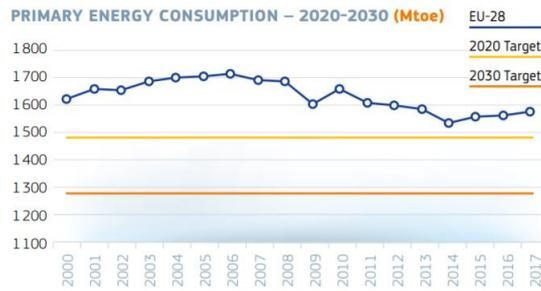
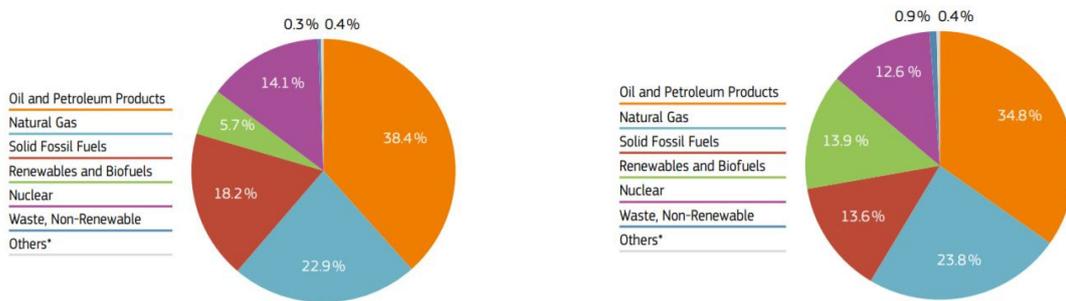


Figura VII.2.6: Consumo energético primario. Fuente: EU energy in figures. Statistical pocketbook 2019



(a) Total Primary Products 2000: 1729.6 Mtoe. Fuente: Eurostat, May 2019 (b) Total Primary Products 2015: 1729.6 Mtoe. Fuente: Eurostat, May 2019

Figura VII.2.7: Total Primary Products

Analizando con más profundidad el consumo energético europeo, se pueden observar que la mayoría de energía se destina a industria y transporte VII.2.7 y VII.2.6 y el origen de la energía consumida fluctúa, en gran medida, dependiendo del país del que se esté hablando.

En el terreno de la energía eólica, como se puede observar en la siguiente figura¹⁰ VII.2.8, se avanza a ritmo cuasi exponencial, lo que augura un gran futuro para este recurso energético. (Union 2019)



Figura VII.2.8: Energía eólica en Europa

Por último, se destaca que en el postrero informe de BP para la unión europea se informaba que, según sus cálculos, las energías renovables se convertirán en la mayor fuente de energía entre el 2035-2040 en la UE, superando al petróleo. Entre las fuentes de energía renovable, se enfatiza que la eólica, tema principal de este proyecto, representará aproximadamente dos tercios de la energía total renovable en el año 2040.

VII.2.1.3 Situación energética: Nivel nacional

A la hora de analizar la situación energética de un país, resulta indispensable retroceder años atrás para comprender de donde se viene y leer los objetivos y planes propuestos para ver a donde se quiere llegar.

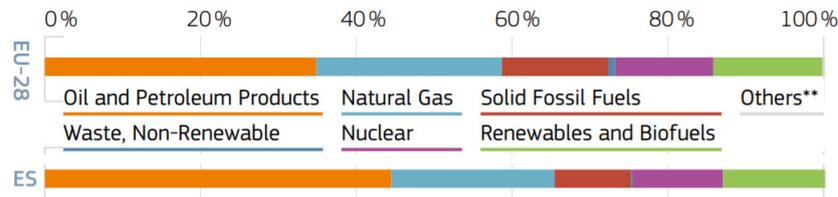


Figura VII.2.9: Procedencia de la energía en España. Fuente: Eurostat, May 2019

A su vez, se requiere de un análisis intenso de las principales variables energéticas de todo país desarrollado: el consumo energético primario y final, la intensidad energética y el grado de autoabastecimiento, como se puede observar en los recursos gráficos VII.2.9 y VII.2.10.

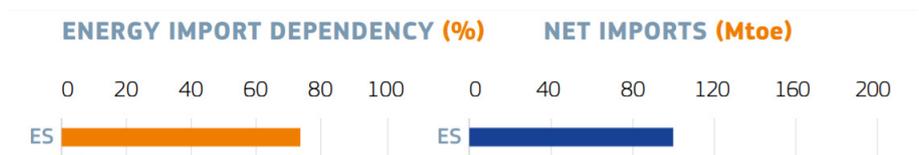


Figura VII.2.10: Dependencia exterior de España. Fuente: Eurostat, May 2019

España se ha caracterizado a lo largo de su historia por tener un grado de autoabastecimiento exiguo debido, entre otros aspectos, a su escasa producción energética primaria y su baja apuesta por las energías renovables.

La incorporación, hace décadas, de España en la unión europea, provocó un giro radical en el consumo energético del país. Se incrementó en gran medida el consumo de energía primaria lo que hizo indispensable invertir en nuevas plantas energéticas, (principalmente no renovables como las plantas de ciclo combinado). La principal misión de este tipo de plantas era adquirir un rol de "interruptor", activándose en los momentos en el que se incrementara la demanda y deteniéndose cuando su rendimiento ya no resultara necesario.

La producción de energías renovables aumentó gratamente entre los años 2005 y 2009. A pesar de esto, la dependencia energética de España se cifra en torno a un 80%, lo que supone un verdadero lastre para el desarrollo tecnológico español.

Actualmente se vive en una etapa de "transición energética", iniciada después de la firma del acuerdo de París en 2015 contra el calentamiento global, en la que España se comprometía a descarbonizarse antes del año 2050.

Dicho acuerdo contaba con la potenciación de las energías renovables, aboliendo el famoso "interruptor al sol" y apoyando a la ya extendida energía eólica en España; contribuyendo, entre otros, a proyectos como el que se tiene entre manos, proyectos de energía eólica offshore o energía eólica marina

En la figura VII.2.10, se puede analizar la evolución de la dependencia energética española frente a la europea en los últimos años. (Union 2019)

%	2017 Renewable Energy Shares				Indicative 2017/2018	2020 RES Target	2030 RES Target
	RES Transport	RES Electricity	RES Heating and Cooling	Overall RES Share			
EU-28	7.4	30.7	19.5	17.5	n.a.	20.0	32.0
ES	5.9	36.3	17.5	17.5	16.0	20.0	

Figura VII.2.11: Objetivos de energías renovables en España. Fuente: Eurostat-RES SHARES, March 2019

VII.2.2 Problema energético

Como se ha descrito en los anteriores puntos de este mismo capítulo; el mundo se encuentra en una compleja encrucijada energética: por una parte, cada vez existe mayor demanda de recursos energéticos y por otra, las principales fuentes de energía actuales se basan en recursos limitados y su operación conlleva riesgos negativos sobre el problema.

Se plantean, por tanto, dos grandes desafíos para el futuro energético mundial: satisfacer la creciente demanda energética a un precio razonable y hacerlo de una forma medioambientalmente sostenible.

Estos factores determinantes del problema energético se representan gráficamente en la figura VII.2.12

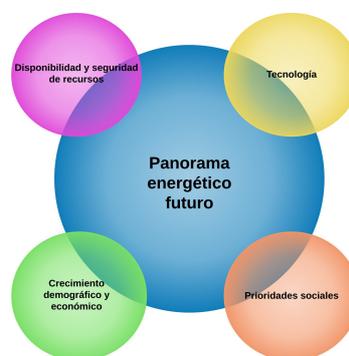


Figura VII.2.12: Principales factores del denominado como problema energético

VII.2.2.1 La sostenibilidad energética

El "World Energy Council" define la sostenibilidad energética como el equilibrio entre tres dimensiones principales: la seguridad energética, la sostenibilidad ambiental y la equidad social.

En otras palabras, se trata intentar conseguir un equilibrio real entre las diferentes variables: económica, social y ambiental. Económicamente los proyectos de generación de energía tienen que ser realizables y el precio de venta del kwh ha de ser accesible para el usuario; esto lleva a la segunda dimensión, la social, donde se debe respetar no solo su economía sino también las tradiciones y rasgos propios de los habitantes de cada zona. Por último, para alcanzar dicha sostenibilidad se tiene que disponer de un panorama ambiental favorable con bajas emisiones de

dióxido de carbono y demás gases de efecto invernadero y con un recurso energético renovable. (Wang 2015)

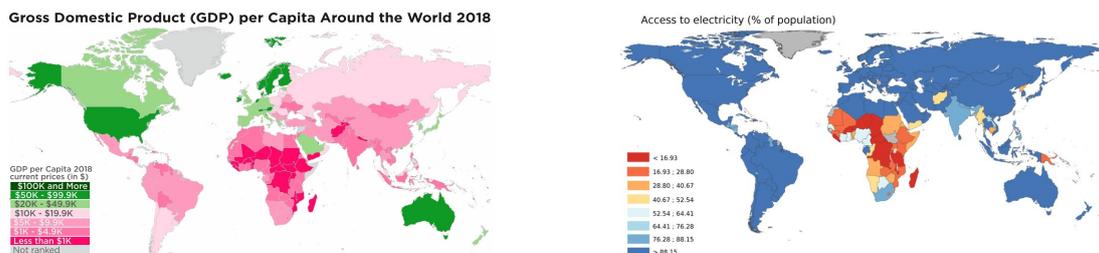


Figura VII.2.13: Pilares para alcanzar la sostenibilidad energética

VII.2.2.2 El Cambio Climático: Un problema Real

En los últimos 80 años, las concentraciones atmosféricas de dióxido de carbono, metano y óxido nitroso han aumentado a niveles sin precedente; en esta tesitura, energías renovables se presentan como un vector clave para encontrar una solución sostenible para este problema ambiental y económico, proporcionando energía a nuevas zonas electrificadas de forma limpia.

El problema ambiental se convierte ahora en un problema económico debido a la alta necesidad de inversión que requieren parques fotovoltaicos y eólicos; dicho factor se intensifica cuando se tiene en cuenta la debilidad económica de los países más necesitados de esta electrificación. Esto último se puede comprobar gráficamente en las figuras VII.2.14a y VII.2.14b ; donde se observa el PIB per cápita de los países y su porcentaje de electrificación respectivamente.



(a) Producto interior bruto según el país. Fuente: Irena 10/06/2020 (b) Porcentaje de electrificación según el país. Fuente: Actua- litix 2012

Figura VII.2.14: PIB vs. % de electrificación

VII.2.3 Energías renovables

VII.2.3.1 Definición

La Real Academia Española de la lengua define energías renovables como: "energía cuyas fuentes se presentan en la naturaleza de modo continuo y prácticamente inagotable"(Española 2014).

Se basan en el aprovechamiento de recursos energéticos naturales, generados prácticamente en su totalidad, bien de forma directa o indirecta, por el Sol. (Peñalvo 2020a)

VII.2.3.2 Clasificación

Como se ha expuesto anteriormente, la mayor parte de los métodos de generación de energía renovable dependen, directa o indirectamente del sol.

Siguiendo dicho criterio se pueden dividir estas fuentes en: fuentes de origen solar (solar directa y solar indirecta), fuentes de origen humano, fuentes de origen gravitacional y fuentes de origen terrestre. Se ha realizado un esquema VII.2.15 en el que se especifica su categoría y el recurso utilizado para la producción de energía.

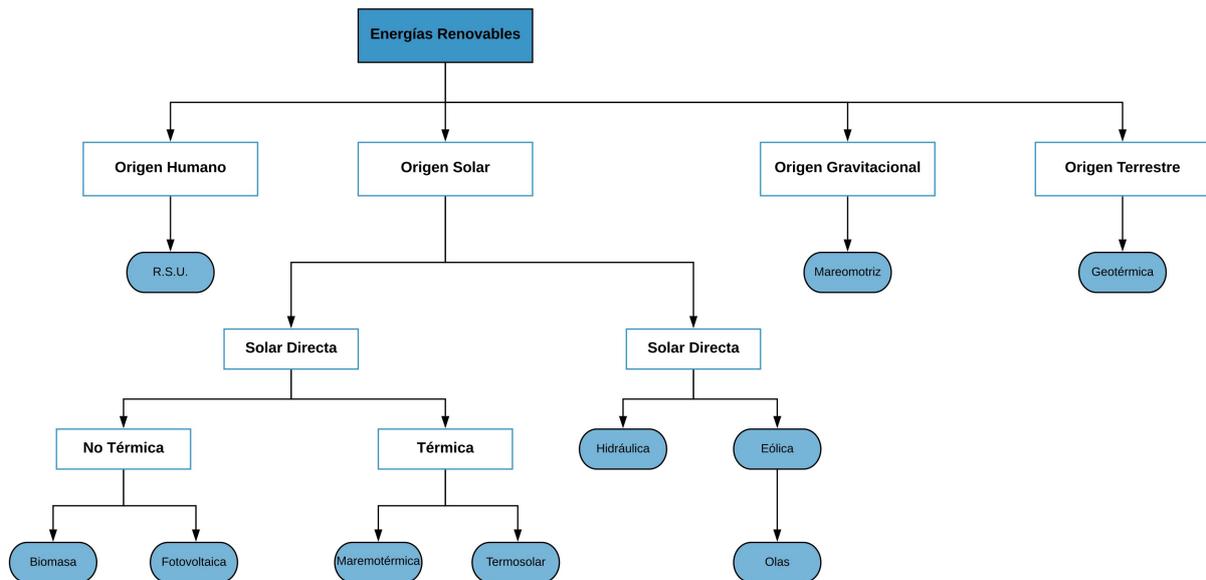


Figura VII.2.15: Esquema de las energías renovables y su procedencia

VII.2.3.3 Papel en el problema energético

Como se ha observado con anterioridad, las fuentes de energía renovable se erigen como la principal alternativa a los anticuados combustibles fósiles para abordar la denominada como "transición energética".

Después de muchos años de críticas por su bajo desarrollo tecnológico y su elevado coste energético, se puede afirmar que ahora sí, las energías renovables han alcanzado su madurez tecnológica ofreciendo un precio de venta bastante competitivo.

Por poner un ejemplo práctico, las instalaciones de autoconsumo fotovoltaico para una vivienda estándar han disminuido en apenas diez años de unos 25000€, a 5000-6000€, con un periodo de retorno de la inversión inferior a los 5 años, sin tener en cuenta la compensación energética aprobada recientemente en países como España. A continuación, se expone un ejemplo de cálculo ambiental para una instalación de autoconsumo fotovoltaico de 2 kW, realizada durante mi estancia en prácticas en la consultoría energética "Solats SL" VII.2.16.

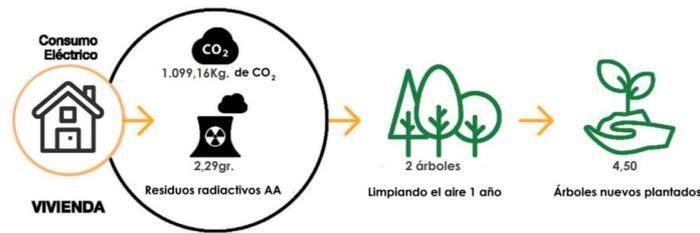


Figura VII.2.16: Ejemplo de impacto ambiental para autoconsumo fotovoltaico de 2 kW. Fuente: Prácticas personales en Solats SL

Un ejemplo práctico como el anteriormente expuesto permite comprobar el gran potencial de estas fuentes de energía a la hora de buscar una solución al problema energético, no solo por su capacidad para generar energía limpia sino por su aptitud para hacerlo de forma atomizada, facilitando el acceso a la electricidad para muchos usuarios.

VII.2.4 Energías Eólica

A lo largo de este punto se definirá el concepto de "energía eólica", se clasificarán los distintos tipos de aerogeneradores y se realizará una breve explicación técnica a la vez que se analizará el procedimiento para determinar la viabilidad de un proyecto eólico y sus respectivos cálculos energéticos.

VII.2.4.1 ¿Qué es la energía eólica y como se produce?

La energía eólica es una fuente de energía renovable cuyo recurso energético es la radiación solar y su diferencia de absorción por parte de la corteza terrestre que genera corrientes de aire denominadas "viento".

El viento se produce debido a la variación de presión atmosférica que de forma continua se produce en el planeta y se define en función de su velocidad y dirección. En cuanto al sentido siempre es el mismo, de las zonas de alta a las zonas de baja presión.

La Tierra devuelve constantemente a la atmósfera el calor que recibe del Sol; en las zonas donde menos calor se libera, la presión de los gases atmosféricos aumenta, mientras que allí donde se libera más calor, el aire se calienta y la presión disminuye. Todo esto genera las zonas de altas y bajas presiones VII.2.17.

La potencia eólica, como se puede comprobar en la siguiente fórmula, depende de diferentes variables: la densidad del aire ρ , la superficie de captación A y la velocidad del viento v_0 .

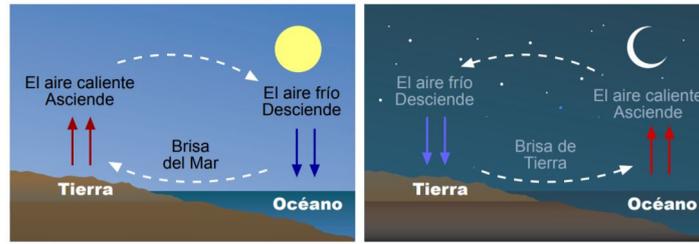


Figura VII.2.17: Ciclo eólico. Fuente: *Energía Eólica: Guía Técnica*

$$P_T \approx C_p \left(A \rho \frac{v_0^3}{2} \right)$$

Figura VII.2.18: Potencia eólica

C_p es un coeficiente de rendimiento de potencia que hace referencia a la fracción mecánica obtenible respecto a la energía eólica presente, su máximo teórico es de alrededor del 59%.

VII.2.4.2 Determinación del emplazamiento de un aerogenerador

VII.2.4.2.1 Rosa de los vientos

El primer elemento a utilizar para este estudio es la denominada como rosa de los vientos:

La rosa de los vientos¹¹ es el recurso más ampliamente utilizado para analizar el viento; se trata de un diagrama polar que representa el tanto por ciento con la dirección y el módulo de una zona terrestre en concreto. Existen dos variantes de la rosa de los vientos:

Rosa de los vientos de procedencia: este diagrama proporciona la dirección o direcciones principales con su frecuencia en una infografía de estilo circular permitiendo así ubicar el aerogenerador en el sitio idóneo.

Rosa de los vientos de potencia: se empleará en el caso de que no se distinga claramente el viento dominante, sirviendo como pieza para inclinar la balanza a un lado u otro.

Se puede observar un ejemplo en la figura VII.2.19, en este caso existen dos direcciones de viento dominantes: NNE y SE. Analizando el segundo diagrama se puede determinar que es la segunda dirección la que proporciona mayor potencia y será, por lo tanto, el emplazamiento idóneo para la turbina.

VII.2.4.2.2 Orografía y obstáculos

Se entiende como orografía la disposición del terreno en un lugar determinado.

En este sentido, se puede comprobar que los accidentes del terreno suaves como colinas o vaguadas influyen positivamente en la aceleración local del viento, por el contrario, los accidentes abruptos disipan la energía del viento debido las turbulencias generadas. Esto se puede observar en la figura VII.2.20. (*Energía Eólica: Guía Técnica*)

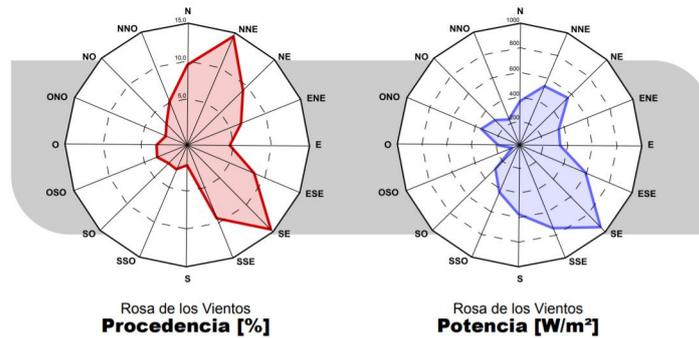


Figura VII.2.19: Ejemplo de rosa de los vientos. Fuente: *Energía Eólica: Guía Técnica*

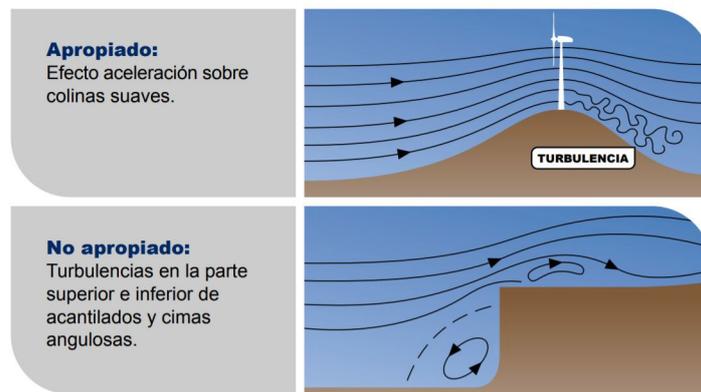


Figura VII.2.20: Ejemplo de la influencia de la orografía. Fuente: *Energía Eólica: Guía Técnica*

Como se ha visto anteriormente, los obstáculos generan turbulencias, que afectan negativamente sobre el rendimiento del aerogenerador. Para evitar este agravante se dispone de dos opciones: colocar la turbina eólica a barlovento¹² o instalar el aerogenerador a entre 7 y 10 veces el diámetro del obstáculo; todo dependerá de si el obstáculo es poroso o no poroso¹³ VII.2.21.

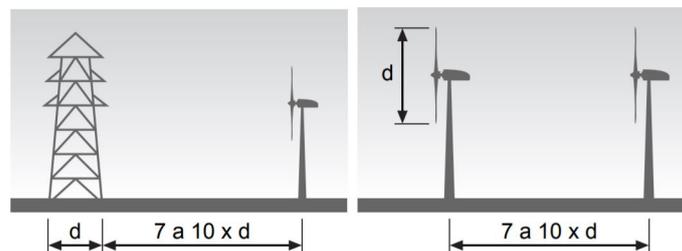


Figura VII.2.21: Influencia de los obstáculos y separación necesaria. Fuente: *Energía Eólica: Guía Técnica*

Otro factor es la rugosidad, comúnmente utilizada en el cálculo de naves industriales para comprobar su resistencia a viento, también tiene uso en el ámbito de la generación de energía. La rugosidad se clasifica numéricamente; mientras que la rugosidad mínima, o de clase 0 se da en medios acuáticos como el mar, la de clase 3 se da en zonas urbanas o de gran masificación de edificios.

Como resumen de todos los pasos anteriores se tiene la web <https://globalwindatlas.info/> que recoge todos los datos y proporciona un entorno atractivo en el que se puede identificar los parámetros y cálculos más relevantes de nuestro estudio, introduciendo los parámetros de diseño.

En la figura VII.2.22 se puede apreciar un recorte de esta interfaz web.

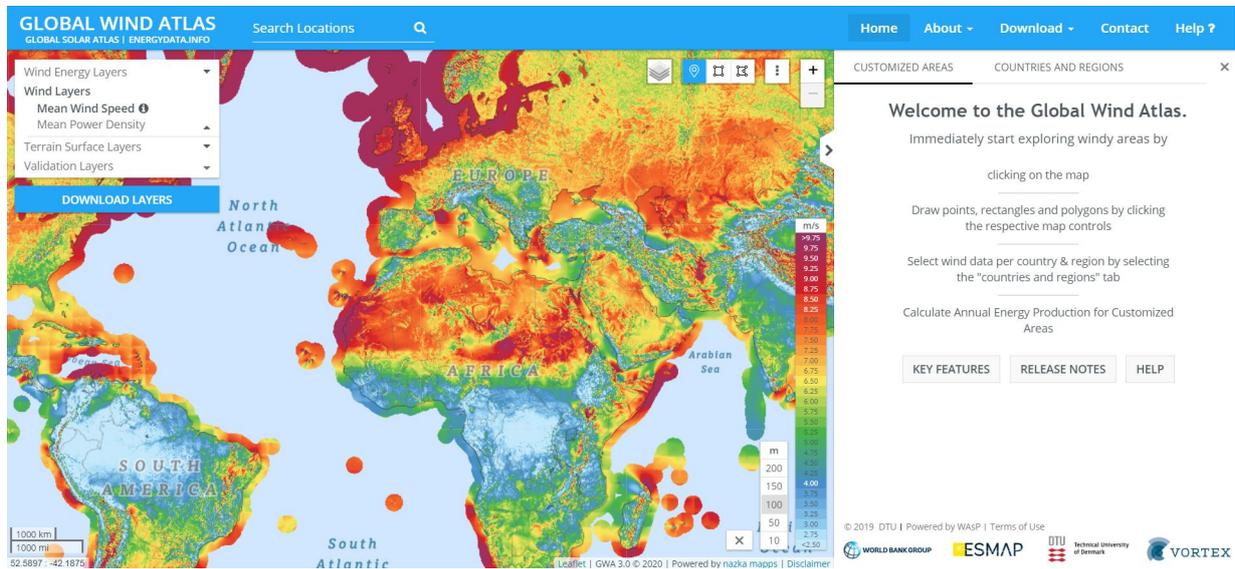


Figura VII.2.22: Recorte del "Global Wind Atlas". Atlas 21/06/2020

VII.2.4.3 Usos de la energía eólica

La energía eólica es una de las fuentes de energía renovable más antigua; utilizada a lo largo de la historia para el bombeo de agua o la molienda de grano.

No fue hasta los años 20 cuando empezó a utilizarse como generador de energía eléctrica gracias a la creación del "WECS" (wind electricity conversion systems). Abandonado durante la electrificación norteamericana y europea, se volvería a utilizar como fuente de energía eléctrica a partir del 1970.

VII.2.4.4 Clasificación

A grandes rasgos los generadores eólicos se pueden dividir en dos grandes grupos:

- Generadores de eje horizontal "HAWT" (Horizontal Axis Wind Turbine)
- Generadores de eje vertical "VAW" (por su nombre en inglés: "Vertical Axis Wind Turbine").

Aerogeneradores de eje vertical: Se dividen a su vez en aerogeneradores de tipo Savonius, Darrieus y Darrieus- Savonius; observables todos ellos en la figura VII.2.23.

El aerogenerador Savonius es el modelo más sencillo que hay; tiene algunos aspectos positivos como que es poco ruidoso o que es útil para aplicaciones de baja potencia, pero también muchas desventajas al necesitar de un control de velocidad y un dispositivo de paro independientes entre otros aspectos.

Por otro lado, el tipo Darrieus corrige la falta de eficiencia del anterior modelo y eleva su campo de trabajo, pudiéndose utilizar para aplicaciones de más alta potencia.

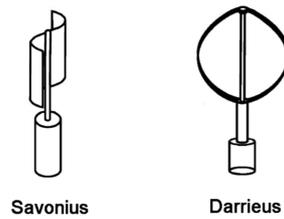


Figura VII.2.23: Boceto de ambos tipos de aerogeneradores Gate 14/06/2020

Aerogeneradores de eje horizontal: Son los aerogeneradores de uso más extendido y los más identificables por la mayor parte de la población. Se dividen en aerogeneradores a barlovento, llamados así porque el viento encuentra antes al rotor que a la torre y los de sotavento VII.2.25.

Los primeros de ellos tienen mayor eficiencia que los generadores a sotavento al no presentar interferencias fluidicas, por el contrario, tiene como inconveniente que necesitan una aleta direccional al no poder alinearse con el viento de forma autónoma.

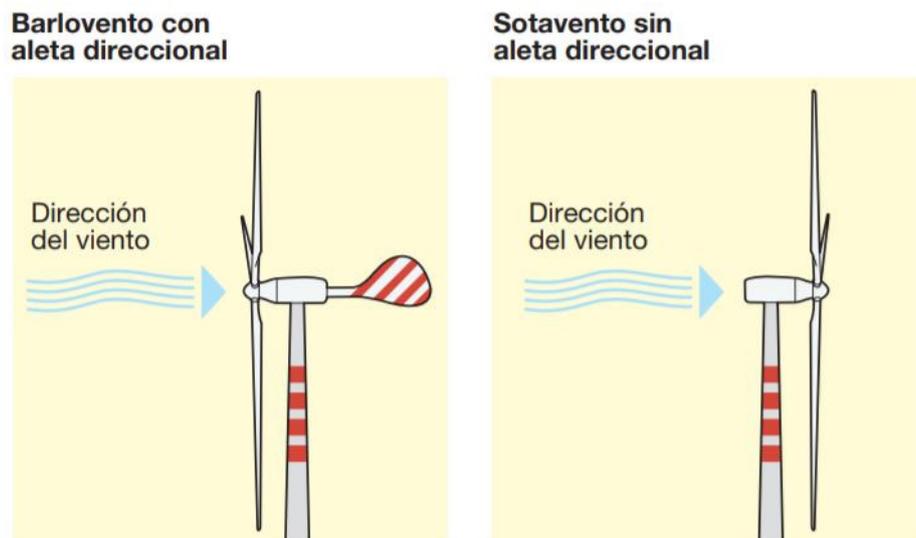


Figura VII.2.24: Aerogeneradores a barlovento y a sotavento. Fuente: Cuaderno de aplicaciones técnicas n.12: Plantas eólicas

Se pueden clasificar también según el número de palas que presenten. En la siguiente tabla se confrontan las principales características de ambos grupos.

2 PALAS	3 PALAS
Menor peso y coste del rotor	Mayor compensación de las fuerzas aerodinámicas
Más ruidoso	Mayor estabilidad mecánica.
Instalación más sencilla	Par motor más uniforme
Diseño más complejo	Menor impacto visual.

Figura VII.2.25: Diferencias entre los aerogeneradores de 2 y 3 palas. Fuente: Cuaderno de aplicaciones técnicas n.12: Plantas eólicas

Para finalizar, se adjunta una infografía VII.2.27 donde se representan los tipos de aerogenerador en función del coeficiente de potencia C_p y velocidad específica: ¹⁴

$$\lambda = \frac{v_t}{v_1} = \frac{\Omega R_{radio}}{v_1}$$

Figura VII.2.26: Velocidad específica

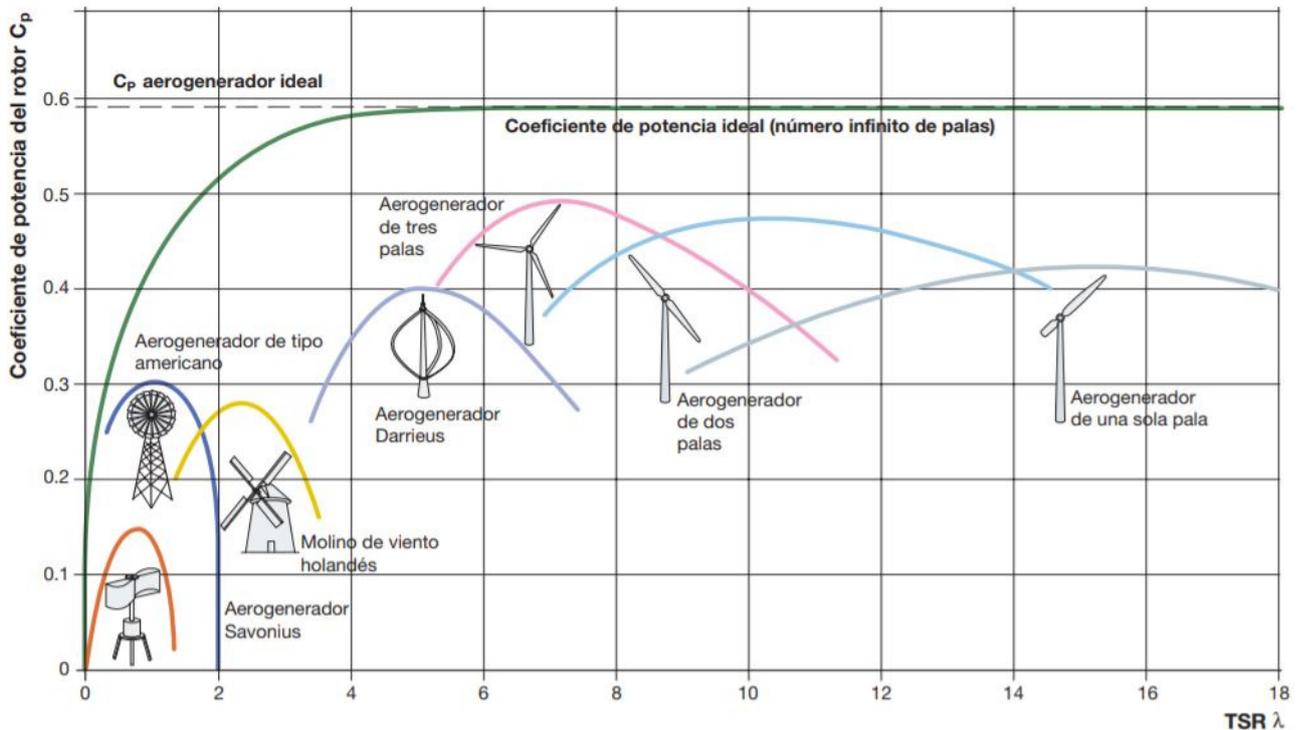


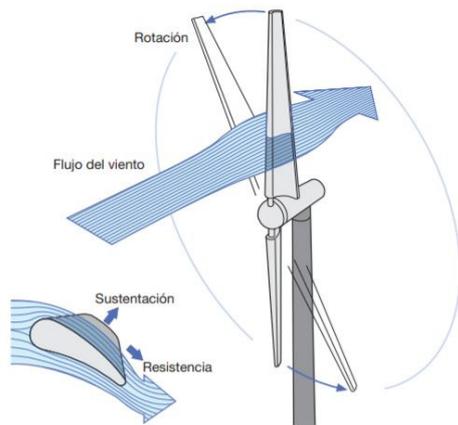
Figura VII.2.27: Infografía en función de la velocidad específica y el coeficiente de potencia. Fuente: Cuaderno de aplicaciones técnicas n.12: Plantas eólicas

VII.2.4.5 Aerogenerador

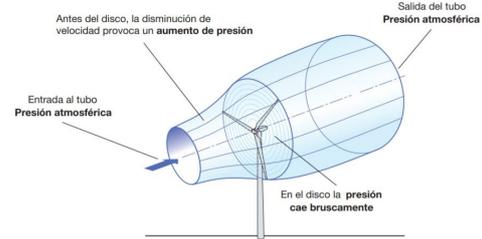
VII.2.4.5.1 Principio de funcionamiento

Un aerogenerador transforma la energía cinética del viento en energía eléctrica pasando por el estado intermedio en forma de energía mecánica de rotación a través de las palas VII.2.28b.

Dichos aerogeneradores pueden ser de sustentación o resistencia. Un aerogenerador necesita una velocidad de viento comprendida entre la mínima (velocidad de acoplamiento o "cut-in") entre 3-5 m/s y una máxima (velocidad de desconexión o "cut-of") de 25 m/s; entregando la potencia nominal a una velocidad entre 12-14 m/s.



(a) Fuente: *Cuaderno de aplicaciones técnicas n.12: Plantas eólicas*



(b) Fuente: *Cuaderno de aplicaciones técnicas n.12: Plantas eólicas*

Figura VII.2.28: Principio de funcionamiento de una turbina eólicas

VII.2.4.5.2 Tipología

Como se ha podido vislumbrar a lo largo de este capítulo, existen dos grandes grupos de parques eólicos: "onshore" o terrestres y "offshore" o marinos.

Los parques eólicos en tierra son el tipo de parque eólico más popular en el mundo hoy en día, siendo la principal fuente de energía renovable en países como España; sin embargo, existe un creciente interés en los países desarrollados por construir parques eólicos en alta mar. Esta inclinación se agiganta sobre todo en los países colindantes con el mar del norte, debido a sus ideales características para la instalación de este tipo de granjas energéticas.

La primera gran diferencia entre ambos bloques se obtiene al observar la eficiencia y capacidad de las granjas; estas dos variables suelen ser mas favorables en instalaciones marinas debido a que las velocidades del viento son más elevadas y consistentes por lo tanto tienen la posibilidad de generar más electricidad a un ritmo más estable.

Mejora también el impacto paisajístico (si se aleja lo suficiente de la costa) y la contaminación acústica (en la figura VII.2.29 se puede observar una comparativa de la generación de decibelios de algunas operaciones cotidianas).

Por su parte los parques marinos conllevan un costo económico de ejecución y mantenimiento muy superior, además de ser mucho menos accesibles y necesitar, por lo tanto, un sistema de monitorización y control más robusto.

En la mejora de la comunicación y las líneas de transmisión de los parques eólicos marinos es donde PROMOTiON ve la oportunidad de mejora que permita hacer, de las granjas offshore una posibilidad real para el futuro energético mundial.

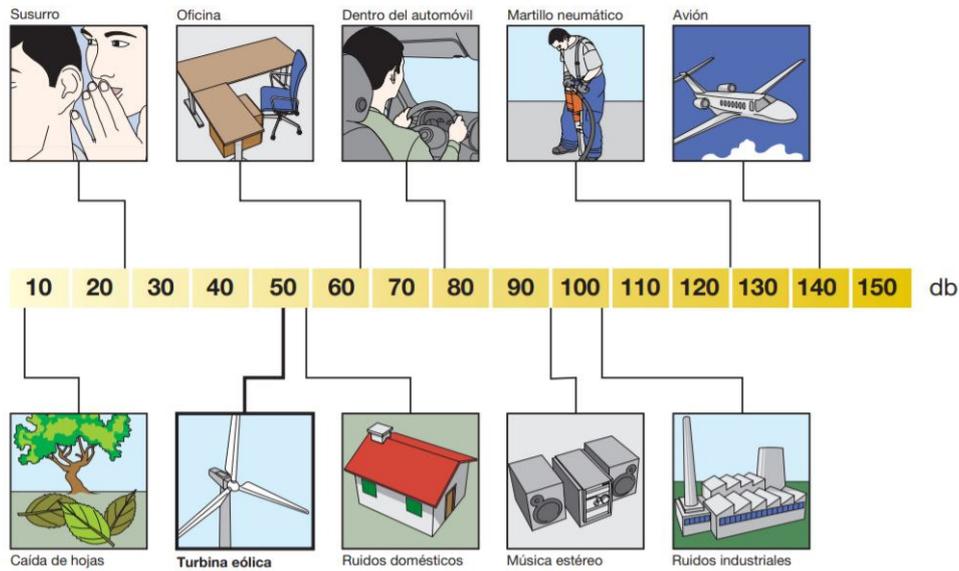


Figura VII.2.29: Comparativa generación de ruido de un generador frente a otras acciones. Fuente: *Cuaderno de aplicaciones técnicas n.12: Plantas eólicas*

<i>Ventajas</i>	<i>Inconvenientes</i>
<i>Fuente de energía limpia y renovable</i>	<i>Intermitencia</i>
<i>Bajos costos de operación</i>	<i>Contaminación acústica</i>
<i>Elevada generación de potencia eléctrica</i>	<i>Impacto paisajístico</i>
<i>Posibilidad de emplearse para el autoconsumo</i>	<i>Impacto ambiental</i>
	<i>Elevado costo de la instalación</i>
	<i>Gran uso de terreno con baja densidad</i>

Tabla VII.2.1: Ventajas e inconvenientes de la energía eólica

VII.2.4.6 Ventajas e inconvenientes de la energía eólica

Su principal ventaja es que trabaja con un recurso natural inagotable como es el viento; su desventaja capital radica en su intermitencia que, además de no asegurar un suministro constante puede crear desajustes en la red eléctrica nacional asimismo puede generar descuadres respecto a la curva de consumo características de la población, generando energía sin posibilidad de ser consumida con la necesidad de almacenarla.

Bibliografía

- ABB. *Cuaderno de aplicaciones técnicas n.12: Plantas eólicas*. Ed. por ABB. ABB (vid. págs. 443-446, 450).
- Actualitix (2012). “Access to electricity”. En: *actualitix.com* (vid. pág. 437).
- Adalid, Rafael Ors Carot; Antonio Martí Campoy; Alberto J. Pérez Jiménez; Joaquín Gracia Morán; Luis J. Saiz. *APUNTES DE INFORMÁTICA INDUSTRIAL* (vid. pág. 450).
- Agarwal, Tarun (01/07/2020). “How SCADA Systems Work?” En: <https://www.elprocus.com/>.
- Arts, American Academy of y Sciences (2007). *The World’s Energy Problemand what we can do about it*. Ed. por Berkele. Berkele.
- Atlas, Global Wind (21/06/2020). “Global Wind Atlas”. En: <https://globalwindatlas.info/> (vid. pág. 442).
- Blanco, M. I. (2009). “The economics of wind energy. Renewable and Sustainable Energy Reviews”. En:
- Enerdata (10/06/2020). “Estadísticas energéticas mundiales | Enerdata”. En: <https://datos.enerdata.net/> (vid. pág. 450).
- Energía, Agencia Andaluza de la. *Energía Eólica: Guía Técnica*. Agencia Andaluza de la EnergÃa (vid. págs. 440, 441).
- España, Gobierno de. *Ley 31/1995, de 8 de noviembre, BOE núm. 269, de 10 de noviembre de 1995*.
- Española, Real Academia (2014). *Diccionario de la Lengua Espanola RAE*. Diccionario de la lengua española. Planeta Publishing Corporation. ISBN: 9786070723513. URL: <https://books.google.es/books?id=naqwoQEACAAJ> (vid. pág. 438).
- Estado, Jefatura del. *Ley 3/2018, BOE, núm. 294, de 5 de diciembre de 2018, Jefatura del Estado*.
- *Ley 37/2003, BOE, núm. 276, de 18 de noviembre de 2003, páginas 40494 a 40505 (12 págs.), Jefatura del Estado*.
- *Ley 4/1989, BOE, núm. 74, de 28 de marzo de 1989, páginas 8262 a 8269 (8 págs.), Jefatura del Estado*.
- Gate, Research (14/06/2020). “Research Gate”. En: <https://www.researchgate.net/> (vid. pág. 443).
- Guisado, David (02/08/2020). “¿Qué son los Pliegos de Condiciones en el Proyecto?” En: <https://www.aparejadorsevilla.es/los-pliegos-de-condiciones-proyecto/> (vid. pág. 121).
- Güncelleme, Son (03/07/2020). “SpaceX”. En: <https://www.cnntrk.com/bilim-teknoloji/spacex-hakkinda-bilmeniz-gerekenler> (vid. pág. 20).

- Hemami, A. (2012). *Wind Turbine Technology*. Delmar, Cengage Learning Series in Renewable Energies. Cengage Learning. ISBN: 9781435486461. URL: <https://books.google.es/books?id=urNSjGoyn8QC>.
- Industria, Energía y Turismo Ministerio de. *Real Decreto 947/2015,BOE, núm. 249, de 17 de octubre de 2015, páginas 97340 a 97342 (3 págs.),Ministerio de Industria, Energía y Turismo*.
- Industria, Turismo y comercio Ministerio de. *Real Decreto 1614/2010,BOE, núm. 298, de 8 de diciembre de 2010, páginas 101853 a 101859 (7 págs.),Ministerio de Industria, Turismo y Comercio*.
- Irena (10/06/2020). “Visualizing GDP per Capita by Country”. En: <https://howmuch.net/> (vid. pág. 437).
- ISO. *ISO/IEC 25000, System and Software Quality Requirements and Evaluation*. 22 páginas.
- Kostal (10/08/2020). En: <https://www.kostal-solar-electric.com/> (vid. pág. 45).
- Patricia López, Jose M. Drake y. *Verificación y Validación*.
- Peñalvo, Elisa (2020a). *Introducción a las energías renovables*. Ed. por Universidad Politécnica de Valencia (vid. pág. 438).
- (2020b). *Introducción a las energías renovables. Energía eólica*. Ed. por Universidad Politécnica de Valencia. Universidad Politécnica de Valencia.
- Penin, A.R. (2007). *Sistemas SCADA 2a*. Marcombo. ISBN: 9788426714503.
- Pérez, Soledad Inmaculada Bernal (2020a). “WP16 MS79 Real time CHIL Set Up”. En: *PROMOTioN*.
- (2020b). “WP3 Self start and black start operation”. En: *PROMOTioN*.
- Pérez-López, Esteban (2015). “Los sistemas SCADA en la automatización industrial”. En: *Tecnología en Marcha*.
- Piasecka I., Tomporowski y co. (2019). “Life Cycle Analysis of Ecological Impacts of an Offshore and a Land-Based Wind Power Plant.” En: *Faculty of Mechanical Engineering, University of Science and Technology in Bydgoszcz*.
- Presidencia, Ministerio de la. *Real Decreto 2177/2004,BOE, núm. 274, de 13 de noviembre de 2004, páginas 37486 a 37489 (4 págs.),Ministerio de la Presidencia*.
- PROMOTioN (27/07/2020). “PROMOTioN”. En: <https://www.promotion-offshore.net/>.
- Rojas, Pablo (01/06/2020). “La Sostenibilidad Energética”. En: <https://www.elfinancierocr.com/opinion/la-sostenibilidad-energetica/3NF5VSKKSRDBVATA3CQTIVV754/story/>.
- Romero, Eduardo (2009). *El problema energético mundial*. Ed. por Repsol. Repsol.
- Salah, Dr. Mohammad. “SCADA Systems”. En: *Mechatronics Engineering Department Hashemite University* ().
- Team, Logitek (10/07/2020). “Qué es OPC y qué es un OPC Server.” En: <https://www.kepserverexopc.com/>.
- Tomporowski A., Piasecka y co. (2018). “Comparison Analysis of Blade Life Cycles of Land-Based and Offshore Wind Power Plants. Polish Maritime Research”. En: *TP University of Science and Technology*.
- Tracy Adams, P.E. “SCADA System Fundamentals”. En: *Continuing Education and Development, Inc.* ().
- Union, European (2019). *EU energy in figures. Statistical pocketbook 2019*. Ed. por Drukarnia INTERAK SP ZOO. Luxembourg: Publications Office of the European Union, 2019 (vid. págs. 434, 435).
- Unión Europea., Parlamento Europeo y el Consejo de la. *Directiva 2009/147/CE,Diario Oficial de la Unión Europea,Parlamento Europeo y el Consejo de la Unión Europea*.

- *Directiva 2009/28/CE, Diario Oficial de la Unión Europea, Parlamento Europeo y el Consejo de la Unión Europea.*
- Unknown (04/06/2020). “3 Architectures of SCADA systems”. En: <https://automationforum.co/3-architectures-of-scada-systems>.
- Wang, José Pablo Rojas (2015). “La sostenibilidad energética”. En: *CEGESTI* (vid. pág. 437).
- “What is LabVIEW?” (03/07/2020). En: <https://www.electronics-notes.com>.
- “Why do I need to be concerned about global energy problems?” En: *geni.org* ().
- Wikipedia (23/07/2020[a]). “ACK”. En: <https://es.wikipedia.org/wiki/ACK> (vid. pág. 450).
- (23/07/2020[b]). “SYN”. En: <https://es.wikipedia.org/wiki/SYN> (vid. pág. 450).

Notas

¹**Offshore:** La energía eólica offshore es aquella fuente de energía que se obtiene al aprovechar la fuerza del viento que se produce en alta mar.

²**TCP:** El protocolo TCP (Transmission Control Protocol) se encarga de trocear los mensajes en paquetes antes de enviarlos a la red y recomponerlos en el destino. Este protocolo proporciona al nivel de aplicación un servicio orientado a conexión, que incluye la entrega garantizada de los datos (TCP asegura que los datos llegan completos, sin errores y ordenados), y control de flujo. (*APUNTES DE INFORMÁTICA INDUSTRIAL*)

³**UDP:** El protocolo UDP (User Datagram Protocol) ofrece al nivel de aplicación un servicio no orientado a conexión, en el que no se garantiza la entrega de datos, aunque es más rápido que TCP. (*APUNTES DE INFORMÁTICA INDUSTRIAL*)

⁴**Front View:** se denomina así a la pantalla visible por el usuario en un programa desarrollado mediante lenguaje de LabVIEW.

⁵**SYN:** es un bit de control dentro del segmento TCP, que se utiliza para sincronizar los números de secuencia iniciales ISN de una conexión en el procedimiento de establecimiento de tres fases (3 way handshake)(Wikipedia 23/07/2020[b])

⁶**ACK:** en comunicaciones entre computadores, es un mensaje que el destino de la comunicación envía al origen de esta para confirmar la recepción de un mensaje.(Wikipedia 23/07/2020[a])

⁷Fuente: United Nations Population Division. World Population Prospects: The 2010 Revision

⁸Fuente: International Energy Agency, World Energy Outlook, 2014

⁹ Fuente: (Enerdata 10/06/2020)

¹⁰EU energy in figures. Statistical pocketbook 2019

¹¹**Rosa de los vientos:** recurso ingenieril esencial para determinar el emplazamiento y/o la viabilidad de una instalación eólica.

¹²**Barlovento:** Se dice que un aerogenerador está a barlovento cuando se sitúa por delante del obstáculo.

¹³**Obstáculo poroso:** dicese de un obstáculo que deja pasar el viento de forma parcial.

¹⁴**Velocidad específica (TSR):** La velocidad específica (TSR, Tip Speed Ratio), identificada por λ , se define como la relación entre la velocidad tangencial en el extremo de la pala y la velocidad del viento a la entrada del tubo de flujo. (*Cuaderno de aplicaciones técnicas n.12: Plantas eólicas*)