



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

DIGITALIZACIÓN DE UN BANCO DE ENSAYOS Y COMPROBACIÓN EXPERIMENTAL

Trabajo Fin de Grado

Grado en Ingeniería Eléctrica

Autor: Leordean, Cosmin Adrian

Tutor: Puche Panadero, Rubén

CURSO ACADÉMICO 2020/2021



RESUMEN

El presente Trabajo Fin de Grado consiste en la digitalización de un puesto de prácticas de accionamientos eléctricos. Para ello, se ha modelizado el puesto de prácticas, motores, convertidores de potencia, la red y los procesos. Esta modelización ha sido implementada mediante la herramienta informática Matlab-Simulink, la cual permite modelizar sistemas multifísicos. Así, para la parte de electrónica de potencia y electromagnetismo se ha hecho uso de la librería *SimPowerSystem*.

Para que la digitalización sea operativa se ha implementado una interfaz gráfica de usuario desde donde se puedan realizar los diferentes ensayos como si se estuviese en el puesto de prácticas. Se destaca así que la digitalización del puesto de prácticas permitirá al usuario final, en este caso, el alumno, poder practicar o ensayar desde casa. La implementación de esta interfaz intuitiva y configurable se ha realizado con AppDesigner.

Todo ello se debe validar experimentalmente, y para ello se ha tenido que desarrollar un sistema de adquisición de magnitudes eléctrico-mecánicas como son la velocidad de giro del motor y el par motor ejercido del puesto de prácticas. Este sistema de adquisición está basado en un microcontrolador de altas prestaciones y bajo coste como es el ESP32, además del desarrollo de una placa de prototipado para la adecuación a la adquisición de las magnitudes. Así, se permite finalmente mostrar de forma local mediante una pantalla LCD los valores o de forma remota mediante una conexión *Bluetooth* a una aplicación basada en Android para no solo ver los valores actuales sino también para ver históricos. Este sistema se ha realizado de una forma versátil permitiendo adquirir y medir ambas magnitudes, par y velocidad, de dos formas distintas. Por otro lado, la aplicación Android ha sido diseñada mediante la herramienta AppInventor, perteneciente al MIT.

Finalmente, la comprobación experimental ha sido efectuada, validando el prototipo y el modelo digital del banco de ensayos, el cual posee un error inferior al 3%.



ABSTRACT

The following Bachelor's Final Thesis consist on the digitalization of a real electric drives test bench. To do so, the test bench has been modelled, including power converters, drives and processes. This model has been implemented using the informatic tool *Matlab-Simulink* because it allowed us to design multi-physic systems. In the same way, to implement the power electronics and electromagnetism components, the *SimPowerSystem* library has been used.

To guarantee the operability of the digitalization a graphical user interface has been implemented, so different tests could be done as if we were on the real test bench. This way, the digitalization allows the final user, in this case, the student, to practice from home. In addition, the implementation of this intuitive and configurable graphical interface has been done using *AppDesigner*.

Furthermore, the obtained results must be verified experimentally, so a measuring system which returns two important electromechanical values, such as drive's speed and torque, has been designed. This measuring system is based on a low cost and high performance microcontroller, such as the ESP32, besides the development of a prototype board for the adequacy of the results. Moreover, the obtained results could be shown on a local way using a LCD screen or on a remote way on an Android application, which also allows the user to check past results. This system has been made on a versatile way, so the two variables, torque and speed, could be obtained at the same time and from different sensors. Regarding the Android application, it has been made using the AppInventor tool, from the MIT.

At last, the experimental verification has been made, validating the prototype and the digital test bench model, resulting on an error lower than a 3%.



RESUM

El present Treball Fi de Grau consisteix en la digitalització d'un lloc de pràctiques d'accionaments elèctrics. Per això, s'ha modelitzat el lloc de pràctiques, motors, convertidors de potència, la xarxa i els processos. Esta modelització ha sigut implementada mitjançant la ferramenta informàtica Matlab-Simulink, la qual permet modelitzar sistemes multifísics. Així, per a la part d'electrònica de potència i electromagnetisme s'ha fet ús de la llibreria SimPowerSystem. Per a què la digitalització siga operativa s'ha implementat una interfície gràfica d'usuari, des d'on es puguen realitzar els diferents assajos com si s'estiguera en el lloc de pràctiques, destacant que la digitalització del lloc de pràctiques permetrà a l'usuari final, en este cas, l'alumne, poder practicar o assajar des de casa. La implementació d'esta interfície intuïtiva i configurable s'ha realitzat amb AppDesigner.

Tot això s'ha de validar experimentalment, i per tal s'ha hagut de desenvolupar un sistema d'adquisició de magnituds electromecàniques com són la velocitat de gir del motor i el parell motor exercit del lloc de pràctiques. Este sistema d'adquisició està basat en un microcontrolador d'altas prestacions i baix cost com és l'ESP32, a més del desenvolupament d'una placa de prototipat per a l'adequació a l'adquisició de les magnituds. Així, es permet finalment mostrar de forma local mitjançant una pantalla LCD els valors o de forma remota per mitjà d'una connexió Bluetooth a una aplicació basada en Android, per a no sols veure els valors actuals, sinó també per a veure històrics. Este sistema s'ha realitzat d'una forma versàtil, permetent adquirir i mesurar ambdues magnituds, parell i velocitat, de dos formes diferents. D'altra banda, l'aplicació Android ha sigut dissenyada mitjançant la ferramenta AppInventor, la qual pertany al MIT.

Finalment, la comprovació experimental ha sigut realitzada, validant el prototip i el model digital del banc d'assajos, el qual posseïx un error inferior al 3%.



ÍNDICE

MEMORIA	9
1. OBJETO DEL PROYECTO	10
2. ANTECEDENTES	12
3. FACTORES A CONSIDERAR: ESTUDIO DE NECESIDADES, LIMITACIONES Y CONDICIONANTES	13
3.1. ESPECIFICACIONES DEL ENCARGO	13
3.2. ESTUDIO DE NECESIDADES PROPIAS	13
3.3. FACTORES A CONSIDERAR: LIMITACIONES Y CONDICIONANTES	14
4. ESTUDIO DE ALTERNATIVAS	15
4.1. ALTERNATIVAS DE SOFTWARE	15
4.2. ALTERNATIVAS DE HARDWARE	16
5. DESCRIPCIÓN DETALLADA DE LA SOLUCIÓN ADOPTADA	17
5.1. MODELO EN SIMULINK DE LOS SISTEMAS	17
5.1.1. RED	18
5.1.1.1. ALIMENTACIÓN AC	18
5.1.1.2. ALIMENTACIÓN DC	18
5.1.2. CONVERTIDOR	19
5.1.2.1. CONTROL AC	20
5.1.2.2. CONTROL DC	28
5.1.3. MOTOR	29
5.1.3.1. MOTOR IM	30
5.1.3.2. MOTOR DC	32
5.1.3.3. SELECCIÓN DE MOTOR	32
5.1.4. PROCESO	33
5.1.4.1. PAR CONSTANTE: CINTA TRANSPORTADORA	33
5.1.4.2. PAR CUADRÁTICO: BOMBA CENTRIFUGADORA	35
5.1.4.3. POTENCIA CONSTANTE: BOBINADORA	36
5.1.4.4. PAR CONSTANTE CON CAMBIO DE CARGA: CINTA TRANSPORTADORA DE VARIAS CARGAS	37
5.1.5. VISUALIZACIÓN	39
5.2. DISEÑO DE LA INTERFAZ GRÁFICA	40
5.2.1. ELEMENTOS UTILIZADOS	40
5.2.2. COMUNICACIÓN CON SIMULINK	41
5.2.3. PROGRAMACIÓN DE LA INTERFAZ – FUNCIONES CALLBACK	42
5.2.4. DISEÑO DE LA INTERFAZ GRÁFICA	43
5.2.4.1. SELECCIÓN DE COMPONENTES	44
5.2.4.2. VISUALIZACIÓN DE RESULTADOS	49
5.2.4.3. CONSIGNAS	51
5.3. DESARROLLO DEL PROTOTIPO DE MEDICIÓN	51
5.3.1. ESP32 WROVER – B	53



Digitalización de un banco de ensayos y comprobación experimental

5.3.2. MEDICIÓN DE VELOCIDAD	54
5.3.2.1. ENCODER	54
5.3.2.2. ÓPTICO	56
5.3.2.3. SELECTORES	57
5.3.3. MEDICIÓN DE PAR	58
5.3.3.1. SENSOR DE EFECTO HALL	59
5.3.3.2. PINZA AMPERIMÉTRICA	60
5.3.3.3. SELECTORES Y RELÉS	61
5.3.4. SALIDA 0 – 10 V	62
5.3.5. RESET	62
5.3.6. PANTALLA LCD	63
5.3.7. ALIMENTACIÓN DEL SISTEMA	64
5.3.8. APLICACIÓN ANDROID PARA VISUALIZACIÓN Y MODIFICACIÓN DE PARÁMETROS	66
5.3.9. SISTEMA FÍSICO FINAL	68
5.4. COMPARATIVA ENTRE SIMULACIÓN Y SISTEMA REAL	70
5.4.1. RESULTADOS DE SIMULACIÓN DEL MODELO	72
5.4.1.1. CONTROL ESCALAR	72
5.4.1.2. CONTROL DIRECTO DE PAR	76
5.4.2. RESULTADOS DE MEDICIÓN	82
5.4.2.1. CONTROL ESCALAR	82
5.4.2.2. CONTROL DIRECTO DE PAR	84
5.4.3. COMPARATIVA ENTRE RESULTADOS DE SIMULACIÓN Y MEDICIÓN	87
5.4.3.1. CINTA TRANSPORTADORA	87
5.4.3.2. BOMBA CENTRIFUGADORA	88
5.4.3.3. BOBINADORA	88
5.4.3.4. CINTA TRANSPORTADORA DE VARIAS CARGAS	89
6. CONCLUSIONES	90
PLANOS	91
PLANO 1 – ESQUEMA DE CONEXIONADO	92
PLANO 2 – CIRCUITO DE SISTEMA DE MEDICIÓN	93
PLIEGO DE CONDICIONES	94
1. INTRODUCCIÓN	95
2. CONDICIONES GENERALES	95
2.1. CONDICIONES TÉCNICAS	95
2.2. CONDICIONES LEGALES	98
2.3. CONDICIONES ECONÓMICAS	98



Digitalización de un banco de ensayos y comprobación experimental

PRESUPUESTO	100
<hr/>	
1. DESGLOSE DE PARTIDAS	101
2. RESUMEN DE PRESUPUESTO	106
3. JUSTIFICACIÓN DE PRESUPUESTO	107
ANEXOS	108
<hr/>	
ANEXO 1 – MODELADO DE BANCO DE ENSAYOS	109
ANEXO 2 – CÓDIGO INTERFAZ USUARIO	159
ANEXO 3 – CÓDIGO DE INTERFAZ GRÁFICA DE MOTOR PERSONALIZADO	184
ANEXO 4 – CÓDIGO DE ESP32 WROVER – B	195
ANEXO 5 – CÓDIGO DE APP ANDROID	206



MEMORIA



1. OBJETO DEL PROYECTO

Los objetivos principales del presente Trabajo de Fin de Grado se pueden dividir en cuatro partes diferenciadas de la siguiente forma:

• **Modelado de banco de ensayos**

En primer lugar, se pretende comprender el funcionamiento en profundidad de los diferentes componentes del banco de ensayos modelándolos en *Simulink* de la manera más precisa y parecida posible respecto a su análogo real. Seguidamente, se aspira al aprendizaje del desarrollo de la interfaz gráfica – en adelante, aplicación – que permita acceder a dichos modelos intuitivamente a un usuario, con conocimientos básicos de simulación, y, además, obtener la información de una forma sencilla y rápida.

• **Diseño de interfaz gráfica**

En segundo lugar, se trata el diseño de una interfaz gráfica la cual permita que el usuario pueda observar el comportamiento de un accionamiento eléctrico en un determinado proceso. Para ello, esta interfaz gráfica ejecutará una simulación del modelo implementado a partir de los parámetros asignados por el usuario y, finalmente, mostrará de forma gráfica los resultados obtenidos.

Dicha interfaz gráfica se desarrolla utilizando la herramienta *AppDesigner* del software *MATLAB*, por lo que la simulación se realiza mediante el entorno de programación de alto nivel *Simulink*, perteneciente al mismo software.

• **Comprobación experimental**

Más adelante, para comprobar la precisión de los modelos y la veracidad de los datos obtenidos, se realiza una validación experimental, proceso básico en cualquier campo de la ciencia para corroborar valores teóricos o, en este caso, simulados. Para ello, será necesario el desarrollo de un sistema de medición.

• **Desarrollo de sistema de medición**

Finalmente, se pretende diseñar sistema de medición que permita obtener en tiempo real los parámetros más importantes del accionamiento, como son el par y la velocidad. Este sistema de medición estará basado en un microcontrolador de altas prestaciones, permitiendo la lectura de diferentes señales de sensores. Posteriormente, mediante la adecuación de dichas señales, se obtiene el valor en tiempo real de cada parámetro, el cual se muestra, por un lado, en una pantalla LCD en el propio banco de ensayos y, por otro lado, en una aplicación Android diseñada mediante la herramienta *AppInventor*.

Digitalización de un banco de ensayos y comprobación experimental

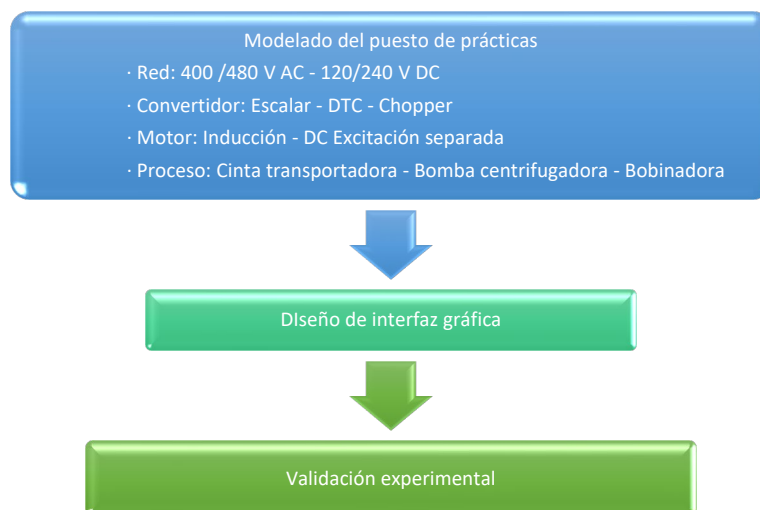


Diagrama 1. Evolución de los objetivos durante la realización del proyecto

Ante la necesidad de implementación o actualización de un proceso para una pequeña, mediana o incluso gran empresa, esta aplicación permite asistir en la toma de decisiones acerca del nuevo equipamiento que se ha de adquirir. Así, se podrá variar cada componente hasta encontrar la combinación más adecuada y suficiente para satisfacer las necesidades del proceso estudiado. De esta forma, obtenemos otro de los objetivos importantes del presente proyecto, que se trata de generar un ahorro económico por parte de la empresa, evitando la adquisición de productos no adecuados o de características muy superiores a las realmente necesitadas, incrementado así el coste de la inversión.

Por último y de la misma manera, se pretende incrementar la eficiencia de los procesos industriales mediante el uso de la aplicación diseñada en este proyecto, reduciendo así las emisiones indirectas de gases de efecto invernadero y contribuyendo a la mejora de la sostenibilidad de empresas industriales, responsables de una considerable parte de las emisiones globales.



2. ANTECEDENTES

Para el desarrollo de los antecedentes, podemos distinguir entre dos ámbitos, el académico y el industrial.

En cuanto al ámbito académico, podemos destacar la necesidad de la existencia de una aplicación que permita mostrar de manera didáctica, intuitiva y directa el comportamiento de diferentes accionamientos eléctricos. De esta forma, el alumnado podría obtener una visión menos abstracta acerca de cada materia tratada. Por otro lado, existe la necesidad de asistencia voluntaria al laboratorio por parte del alumnado para complementar y corroborar el estudio realizado de una materia en concreto. Sin embargo, éste no se encuentra siempre disponible o accesible y, además, no se permite el uso de los componentes en puntos de trabajo críticos, que puedan dañarlos. Asimismo, se puede destacar la imposibilidad de simular procesos concretos.

En cuanto al ámbito industrial, podemos destacar la tendencia de las empresas a reducir al máximo el valor las inversiones en nuevo equipamiento, ajustando al máximo las necesidades a cubrir. Así, los tiempos de retorno de dichas inversiones quedarían reducidos, de la misma forma, permitiendo también la financiación a otras secciones de la misma empresa. De esta manera, se trata de elegir siempre el equipo adecuado a los requisitos. Sin embargo, a lo largo de la historia, la elección de estos componentes ha resultado en una compleja decisión, debido a la dificultad de la estimación del comportamiento de los equipos. Esto ha derivado en la errónea selección del equipamiento, en algunos casos por no alcanzar las demandas de los procesos a controlar y, en otros casos, por sobredimensionar demasiado la instalación, dando lugar a una pérdida de eficiencia y, como se ha comentado anteriormente, un incremento de la inversión a realizar. Además, y de la misma manera, existe también la tendencia industrial de realizar un *Digital Twin*, el cual consiste en un modelo digital del sistema que se desea implementar, o que ya se ha implementado, para comprobar su funcionamiento en situaciones límite, o situaciones poco comunes o imprevisibles.

Por tanto, la digitalización del banco de ensayos de accionamientos eléctricos pretende satisfacer estas necesidades, de cara a, por un lado, mejorar y facilitar el aprendizaje y la comprensión del funcionamiento de accionamientos eléctricos concretos y, por otro lado, reducir costes e incrementar la eficiencia y durabilidad de los componentes eléctricos.



3. FACTORES A CONSIDERAR: ESTUDIO DE NECESIDADES, LIMITACIONES Y CONDICIONANTES

3.1. ESPECIFICACIONES DEL ENCARGO

Para un correcto desarrollo del proyecto, se debe garantizar un cumplimiento de una serie de condicionantes impuestos en orden cronológico de ejecución, los cuales se detallan a continuación:

- Digitalización de un banco de ensayos, desde la alimentación hasta la carga, pasando por el convertidor y el motor.
- Correcto funcionamiento de los modelos ejecutados
- Diseño de una interfaz gráfica para el acceso sencillo a los resultados de la digitalización.
- Diseño de un sistema de medición para la comprobación de los resultados obtenidos de los modelos
- Validación de los resultados obtenidos mediante la comparación entre los resultados de simulación y los resultados recogidos por el sistema de medición implementado.

3.2. ESTUDIO DE NECESIDADES PROPIAS

Se detallan a continuación las necesidades principales que surgen para, por un lado, la parte de *Software*, donde se trata de digitalizar un banco de ensayos e implementar la interfaz gráfica, y por otro lado la parte de *Hardware*, donde se implementa el sistema de medición.

Necesidades de *Software*

- Uso de un software ampliamente extendido y que permita modelar en diferentes ámbitos.
- Que permita implementar y simular los sistemas tratados en el presente proyecto.
- Que permita visualizar resultados de la simulación.
- Que permita almacenar los mismos resultados.



Digitalización de un banco de ensayos y comprobación experimental

- Que permita la introducción de librerías de máquinas eléctricas.
- Que permita implementar cálculos matemáticos.
- Que permita el diseño de una interfaz gráfica que pueda acceder a la simulación.
- Que el diseño de la interfaz gráfica sea intuitivo y sencillo.

Necesidades de Hardware

- Un microcontrolador económico para la lectura de diversas señales.
- Que la programación del microcontrolador sea sencilla e intuitiva.
- Que posea suficientes entradas y salidas.
- Una alta velocidad de procesamiento.

3.3. FACTORES A CONSIDERAR: LIMITACIONES Y CONDICIONANTES

En un principio, la idea para la elaboración de este proyecto se trataba de realizar la digitalización de un banco de ensayos y poder observar la evolución de las variables en tiempo real. Sin embargo, tras diversos intentos y búsquedas, se llegó a la conclusión de que no era posible realizar una visualización en tiempo real de los resultados del sistema debido a la limitación impuesta por la librería utilizada donde se encuentran la mayoría de los componentes eléctricos y electrónicos utilizados: *Spezialiced Power Systems*, de *Simscape*.

Seguidamente, tras decidir únicamente ver el resultado de cada variable una vez finalizada la simulación en vez de en tiempo real, la validación experimental se pensaba realizar utilizando sistemas de medición comerciales o ya implementados anteriormente. Sin embargo, tras la oportunidad de realizar prácticas en el Laboratorio de Máquinas Eléctricas y Accionamientos Eléctricos se pudo acordar la realización de un sistema de medición para la validación de datos del presente proyecto final de grado.

De esta forma, se ha conseguido aplicar los conocimientos de diferentes ámbitos del grado, como pueden ser de *Control de Máquinas y Accionamientos Eléctricos*, *Electrónica*, *Programación Informática en Arduino*, *Circuito eléctricos* o incluso *Electrónica de potencia*.

Finalmente, en cuanto a las limitaciones y condicionantes surgidos para la realización del presente Trabajo Fin de Grado, se destaca la necesidad de tener acceso a un banco de ensayos real, con el fin de poder realizar pruebas con el sistema de medición en primera instancia y posteriormente validar los resultados. Además, se requiere un



ordenador personal para el diseño del modelo de simulación, la interfaz gráfica e incluso para la programación del microcontrolador del sistema de medición.

4. ESTUDIO DE ALTERNATIVAS

Se tratan en este punto las diferentes alternativas disponibles para, por un lado, la realización de las simulaciones mencionadas y, por otro lado, para la implementación del sistema de medición.

4.1. ALTERNATIVAS DE SOFTWARE

Para la modelación de las simulaciones existen diferentes programas que se pueden agrupar en dos grandes grupos en función del método de resolución que utiliza: métodos analíticos y métodos de elementos finitos.

a. Métodos analíticos

En primer lugar, los métodos analíticos son aquellos que se basan en fórmulas matemáticas para la resolución de un determinado problema. Por lo general, este tipo de métodos, en sistemas complejos adquieren una menor precisión en comparación con los métodos de elementos finitos, aunque la resolución es considerablemente más rápida.

Así, existen diferentes programas que utilizan este tipo de métodos de resolución, como pueden ser PSCAD o PSIM.

b. Métodos de elementos finitos

En segundo lugar, los métodos de elementos finitos son aquellos basados en la aproximación de soluciones de ecuaciones diferenciales parciales, las cuales en diversos sistemas son complejas de resolver. En general, el tiempo requerido para hallar una solución utilizando este método es elevado, aunque depende plenamente de la capacidad de computación del ordenador utilizado. En cambio, la precisión conseguida mediante este método es muy elevada.

Así, existen también diversos programas que utilizan estos métodos, como pueden ser JMAG o MATLAB.

De esta forma, se ha optado por un software para realizar la implementación de los modelos de simulación que utilice métodos de elementos finitos ya que la precisión en este tipo de sistemas es uno de los factores clave. Seguidamente, se ha tratado de que



Digitalización de un banco de ensayos y comprobación experimental

el software utilizado permita fácil conexión con una interfaz gráfica externa, o incluso permita la creación de una. Así, además de poseer licencia educativa, MATLAB reúne los requisitos básicos para la realización del proyecto, ya que cuenta también con la herramienta AppDesigner.

4.2. ALTERNATIVAS DE HARDWARE

En cuanto a las alternativas de hardware, para la implementación del sistema de medición, se han barajado varias opciones. Estas opciones son, por un lado, el microcontrolador ESP32 y, por otro, la placa electrónica Arduino.

De esta forma, se ha escogido utilizar el ESP32, concretamente la versión WROVER – B, debido a la disponibilidad de este en el laboratorio y a su bajo precio. Además, posee mejores prestaciones y rendimiento que la placa Arduino, lo cual nos permite realizar mediciones con más precisión.

5. DESCRIPCIÓN DETALLADA DE LA SOLUCIÓN ADOPTADA

La descripción detallada de la solución adaptada será fragmentada de forma general en las diferentes partes que componen el Diagrama 1. Así, la explicación será desarrollada en el mismo orden cronológico al de la implementación de la solución.

5.1. MODELO EN SIMULINK DE LOS SISTEMAS

El banco de ensayos propuesto a digitalizar se compone de los elementos descritos a continuación, de forma que coincida lo máximo posible con el material disponible en el laboratorio y, por tanto, con lo existente en una industria real.

De esta forma, se diferencian los principales componentes a desarrollar detalladamente en este punto, los cuales forman parte del modelo realizado en Simulink:

- Red
- Convertidor
- Motor
- Proceso
- Medición

Se muestra en la Figura 1 el esquema general del modelo, de manera que se puede observar el conexionado básico entre cada uno de los componentes.

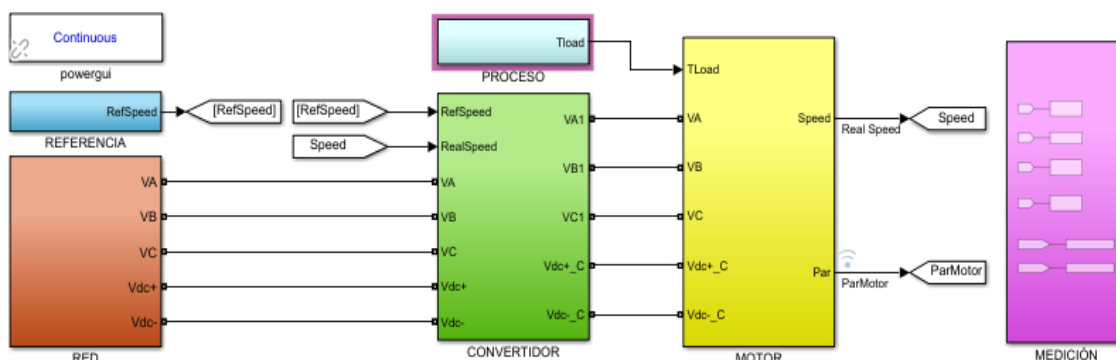


Figura 1. Esquema general de conexionado en Simulink

Digitalización de un banco de ensayos y comprobación experimental

Seguidamente, podemos detallar el funcionamiento de cada uno de los componentes.

5.1.1. RED

Este subsistema, denominado *Red*, se trata del primer componente del modelo, el cual es responsable de generar una tensión de alimentación para el resto de las componentes.

Así, observamos en la Figura 2 que posee 5 salidas, de las cuales tres de ellas corresponden a la alimentación en corriente alterna trifásica, mientras que las dos restantes corresponden a la alimentación en corriente continua.

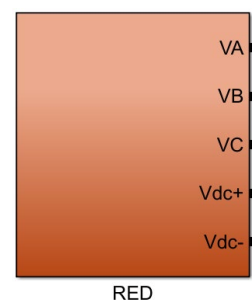


Figura 2. Subsistema 'Red'

Si observamos la composición detallada de este subsistema, tendremos:

5.1.1.1. ALIMENTACIÓN AC

Por un lado, la alimentación en corriente alterna se realiza mediante una *Fuente de tensión trifásica programable*, como se muestra en la figura siguiente.

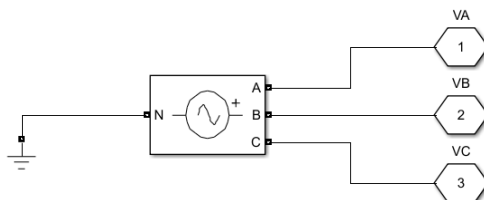


Figura 3. Modelado de fuentes de alimentación - AC

5.1.1.2. ALIMENTACIÓN DC

Por otro lado, la alimentación en corriente continua se realiza mediante una *Fuente de tensión en corriente continua*, como se muestra a continuación.

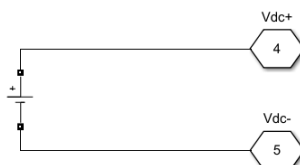


Figura 4. Modelado de fuentes de alimentación - DC

Digitalización de un banco de ensayos y comprobación experimental

Así, los valores de tensión aplicados a cada una de las fuentes serán seleccionados por el usuario desde la interfaz gráfica, descrita en el punto 7.2, en función del convertidor y motor elegido.

5.1.2. CONVERTIDOR

En segundo lugar, podemos observar el convertidor, responsable principalmente de generar, a partir de la tensión y frecuencia proporcionada por la red de alimentación, una tensión y frecuencia adecuada a la velocidad de consigna o al par dependiente del proceso a ejecutar.

De esta forma, se puede ver que este bloque posee 7 entradas, de las cuales tres de ellas (VA, VB y VC) pertenecen a la alimentación trifásica, dos (Vdc+ y Vdc-) pertenecen a la alimentación en continua, provenientes del subsistema explicado anteriormente. Por otro lado, una de las entradas corresponde a la velocidad de consigna (RefSpeed), la cual se obtiene a partir de un valor introducido por el usuario en la interfaz gráfica, y la restante a la velocidad real de giro del motor en cada momento (RealSpeed), obtenida a partir de un bloque de medición, desarrollado más adelante.

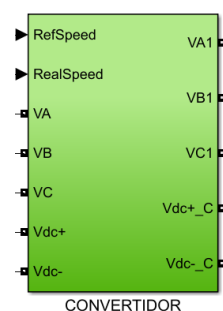


Figura 5. Subsistema "Convertidor"

Por otro lado, observamos que posee 5 salidas, de las cuales tres (VA1, VB1 y VC1) se corresponden a la tensión de alimentación trifásica del subsistema *Motor*, a una frecuencia determinada, y las dos restantes (Vdc+_C y Vdc-_C) a la tensión de alimentación continua del subsistema *Motor*.

Más detalladamente, distinguimos dos tipos de convertidores, como se puede observar en la Figura 5, uno para el caso de controlar la corriente alterna y otro para el caso de corriente continua.

Digitalización de un banco de ensayos y comprobación experimental

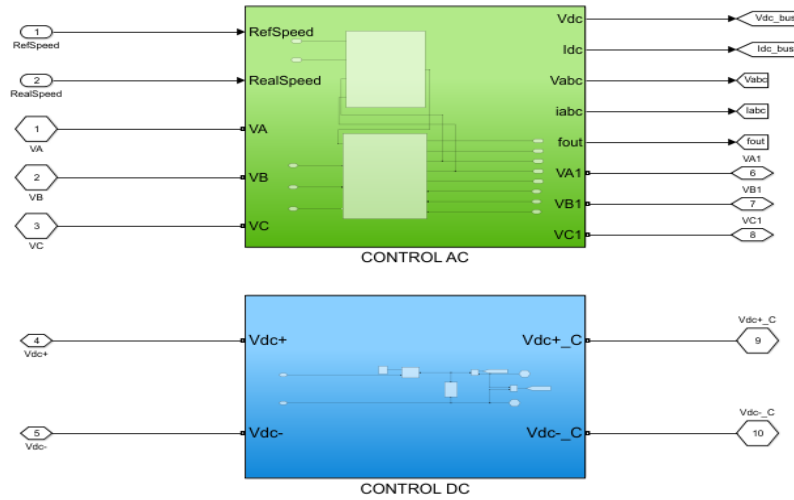


Figura 6. Control AC y Control DC

5.1.2.1. CONTROL AC

Para este primer caso tendremos dos principales bloques, uno que representa el *Hardware* del convertidor, el cual es común a los diferentes tipos de control de corriente alterna, y otro que representa el *Software*, el cual variará en función del tipo de control escogido.

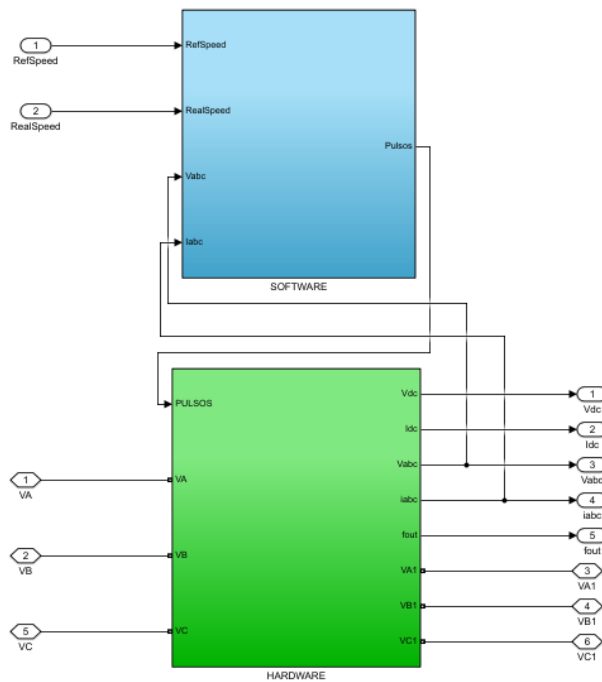


Figura 7. Composición del Control AC

Digitalización de un banco de ensayos y comprobación experimental

5.1.2.1.1 HARDWARE

En primer lugar, detallaremos la estructura física del hardware del convertidor AC. Este hardware no es más que convertidor AC/AC controlado, cuyo circuito principal se puede ver detallado en la figura siguiente.

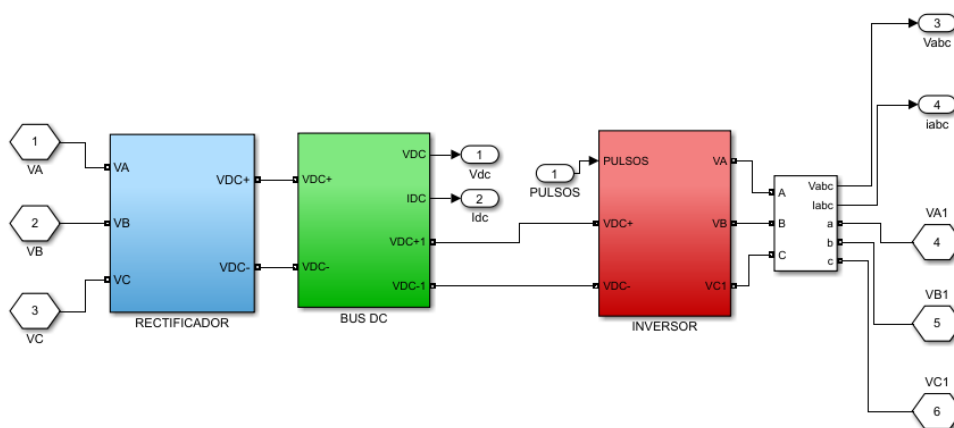


Figura 8. Esquema de Hardware

Se puede observar, de izquierda a derecha, que en primer lugar tenemos un *Rectificador*, el cual está compuesto de un puente de diodos, cuya función es convertir la corriente alterna en continua.

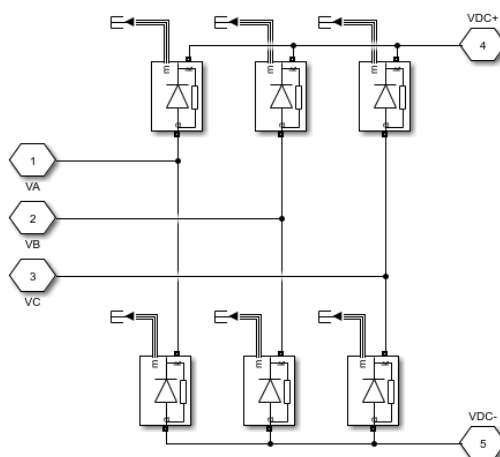


Figura 9. Puente rectificador

Más adelante, se puede ver un *Bus DC*, en serie con la salida en corriente continua del rectificador, compuesto únicamente por un condensador en paralelo, cuya función principal es reducir el rizado de tensión derivado del rectificado de la corriente.

Digitalización de un banco de ensayos y comprobación experimental

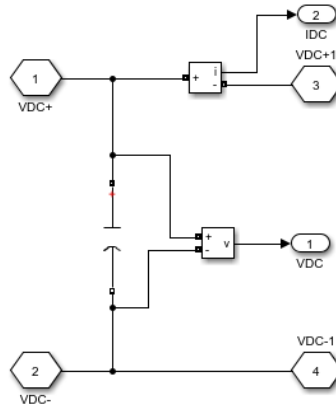


Figura 10. Bus DC

Por último, tenemos el bloque *Inversor*, en serie con el bloque del bus DC. Este bloque inversor está compuesto por un puente de transistores, cuya función es convertir la corriente continua en corriente alterna, a una tensión y frecuencia determinada. Se puede observar que dependen de una de las entradas denominada *Pulsos*, la cual es proveniente del software del *Control AC*. Así, los transistores estarán controlados por estos pulsos y se conseguirá dicha conversión de corriente continua a alterna a tensión y frecuencia determinada.

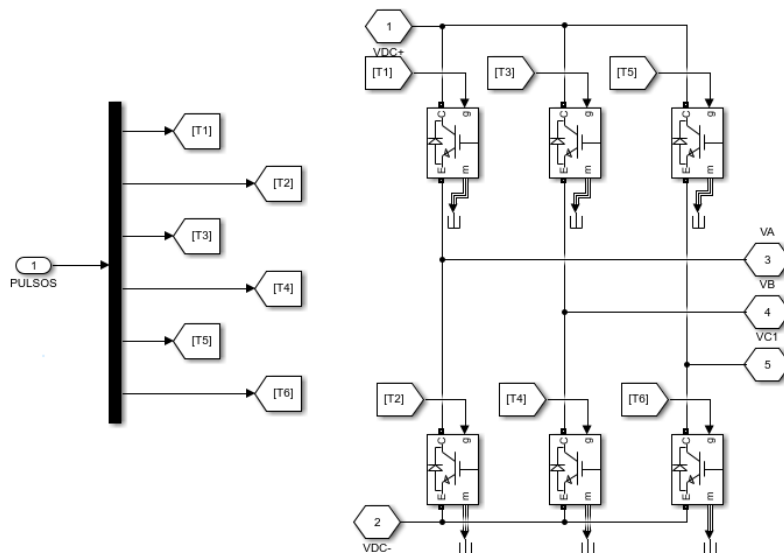


Figura 11. Puente inversor

Consecuentemente, se puede detallar el origen de dichos pulsos.

5.1.2.1.2. SOFTWARE

En cuanto al componente software del convertidor de corriente alterna, se han implementado dos tipos de control diferentes:

- Control escalar (V/F)
- Control directo de par (DTC)

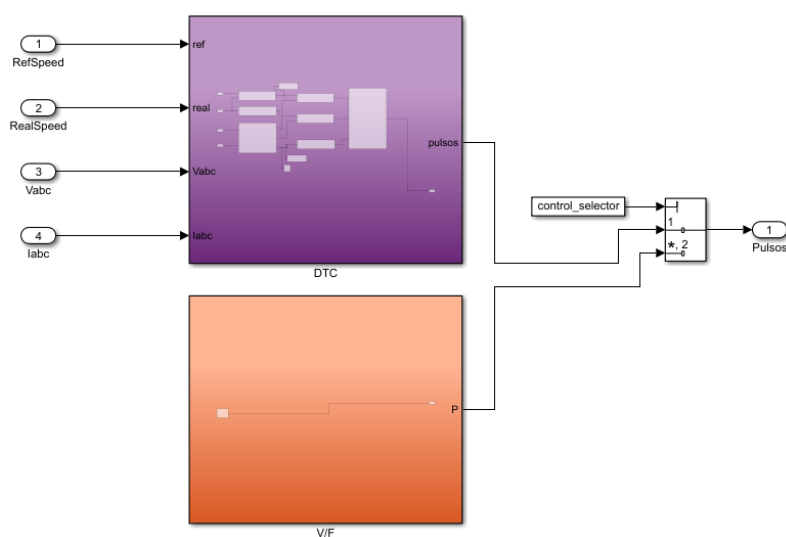


Figura 12. Tipos de control implementados

De esta forma, posteriormente se podrá comparar la respuesta de cada uno de los sistemas de control.

5.1.2.1.2.1. CONTROL ESCALAR (V/F)

El control escalar es uno de los tipos de control de corriente alterna más sencillos existentes hasta la fecha. Su propósito fundamental es mantener el flujo magnético constante, manteniendo la relación tensión – frecuencia constante.

Así, está principalmente diseñado para procesos en los que no exista un gran cambio de carga o para procesos en los que no se requiere cambios bruscos de velocidad.

Por otro lado, sabemos que la velocidad de un motor de corriente alterna viene determinada por

$$n = \frac{60 \cdot f}{p}$$

Digitalización de un banco de ensayos y comprobación experimental

Donde

n : velocidad [rpm]

f : frecuencia de alimentación [Hz]

p : pares de polos

Por tanto, vemos que para conseguir que el motor de corriente alterna gire a una determinada velocidad, podemos modificar su frecuencia de alimentación o su número de pares de polos. Siendo la segunda opción inviable debido a su elevado coste y nula posibilidad de variación de velocidad de forma rápida y eficiente, la primera resulta la mejor opción ya que, utilizando el control escalar podemos fácilmente regular este parámetro.

En cuanto a la implementación del control escalar, se ha utilizado un generador de onda PWM a la cual se le puede asignar una frecuencia determinada, a partir de la velocidad de consigna y utilizando la expresión matemática anteriormente descrita. A nivel de implementación real, este generador de PWM se realiza a partir de la comparación entre una onda senoidal y una onda triangular. Sin embargo, para este caso se ha utilizado directamente el generador de onda PWM, de forma que simplifique el sistema y mejore el rendimiento de la simulación.

De esta forma, el circuito resultante es el siguiente:

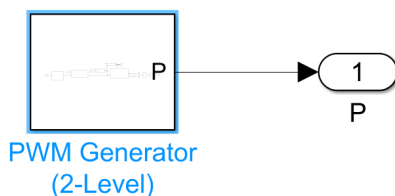


Figura 13. Esquema de control escalar

Se puede observar que, en este caso, a diferencia del que se procederá a detallar a continuación, no existe realimentación alguna acerca de la velocidad del motor, o corriente demandada, es decir, independientemente de la situación del motor, la tensión y frecuencia de alimentación será siempre la misma.

Existen, sin embargo, técnicas que permiten un arranque más gradual mediante una rampa de aceleración, o un arranque con un par resistente elevado, mediante un *Voltage Boost*, entre otros, aunque para esta simulación no ha sido necesario implementarlos.

5.1.2.1.2.2. CONTROL DIRECTO DE PAR (DTC)

El segundo tipo de control implementado es el *Control directo de par*. Este tipo de control se caracteriza, principalmente, por tener una buena respuesta de par, incluso a velocidad cero. Además, la respuesta de velocidad es muy precisa, a diferencia del anterior, y no necesita realimentación del estado del motor, únicamente de la tensión y corriente suministrada a este.

Sin embargo, es considerablemente más costoso que el control escalar y su implementación es más compleja. De esta forma, podremos realizar posteriormente una comparación entre estos tipos de control donde las diferencias sean claramente más perceptibles.

El esquema general del control directo de par se detalla en la Figura 13.

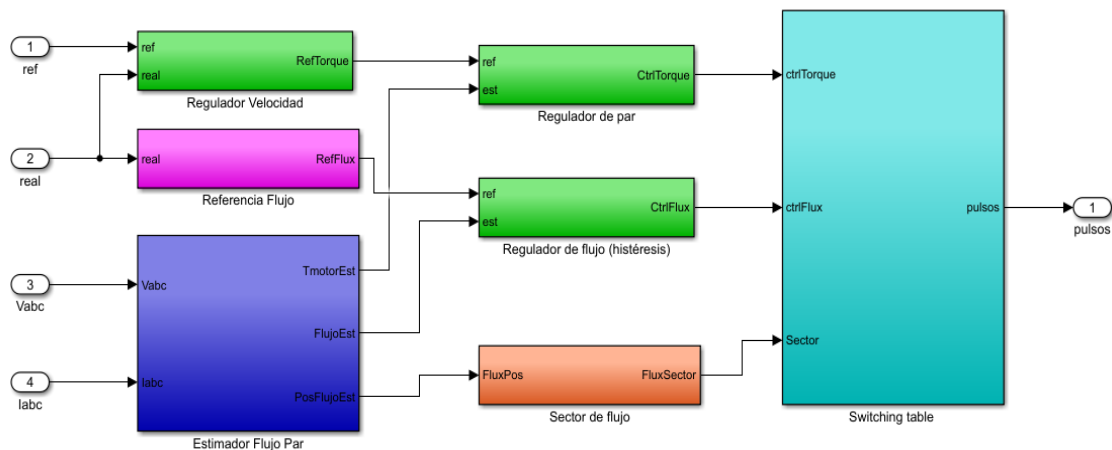


Figura 14. Esquema de control directo de par

Se explica a continuación, de forma breve, el funcionamiento básico de este tipo de control.

En un primer lugar, se obtienen las medidas de tensión y corriente suministradas al motor, como se ha comentado anteriormente, a partir de la salida del propio convertidor, las cuales se convierten al plano α, β . A partir de dichas mediciones, se puede estimar por un lado la velocidad de giro del rotor, aunque para este modelo directamente se ha tomado la velocidad real, y, por otro lado, el vector flujo magnético del estator y el par mecánico. Además, como se puede observar en las ecuaciones descritas a continuación, que estiman las variables mencionadas se deben de conocer con anterioridad las características eléctricas de la máquina eléctrica, ya sea mediante un ensayo manual o uno autónomo realizado por los convertidores modernos.

Digitalización de un banco de ensayos y comprobación experimental

$$\psi_s = \int (\bar{V}_s - \bar{I}_s \cdot R_s) dt$$

$$T_e = K_L \cdot |\psi_s| \cdot |\psi_r| \cdot \sin \gamma$$

Donde:

$$K_L = \frac{3 \cdot P \cdot L_M}{2 \cdot \sigma \cdot L_S \cdot L_r} \cdot L_S$$

$$\sigma = 1 - \frac{L_M^2}{L_S \cdot L_r}$$

Que para este caso se ha simplificado como

$$K_L = 0.75$$

Y:

$\psi_s =$ Flujo del estator

$\bar{V}_s =$ Vector tensión del estator

$\bar{I}_s =$ Vector corriente del estator

$T_e =$ Par ejercido

$\psi_r =$ Flujo del rotor

$\gamma =$ Ángulo entre el vector flujo del estator y el vector flujo del rotor

$L_S =$ Inductancia del estator

$L_r =$ Inductancia del rotor

$L_M =$ Inductancia mutua

$P =$ Número de pares de polos

De esta forma, este estimador de flujo y par quedaría modelado de la forma en la que se indica en la Figura 14.

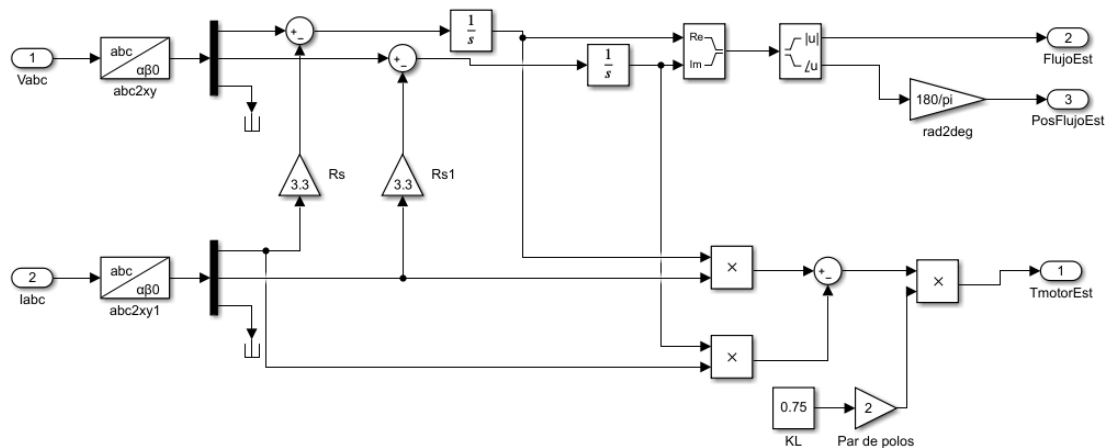


Figura 15. Modelado del estimador de flujo y par

Digitalización de un banco de ensayos y comprobación experimental

En segundo lugar, una vez conocidos dichos parámetros, se procede a calcular el error entre estos y los valores de flujo y par de referencia. Estos valores de referencia son calculados a partir de la velocidad estimada del punto anterior y un control PID únicamente para la obtención del par de referencia.

Así, a estos errores de par y flujo obtenidos se les aplica una banda de histéresis, de forma que se determine, mediante una tabla de interrupciones predefinida y, dependiendo de la posición, o sector en que se encuentre, el flujo, el vector de tensión necesario (secuencia de pulsos) de aplicar para obtener el par adecuado. Más concreta e intuitivamente, como se muestra en la figura siguiente, observamos por un lado los seis vectores de tensión que se pueden aplicar a la máquina eléctrica. Por otro lado, se puede ver el vector flujo del estator y el vector flujo del rotor, los cuales rotan respecto al origen de coordenadas de acuerdo con el vector tensión aplicado y la velocidad real del motor, respectivamente, y donde el segundo tiende a seguir al primero.

Así, una vez aumentada la carga en el rotor de la máquina eléctrica, se generará una ralentización del vector flujo del rotor, generando así una mayor diferencia entre el vector flujo del estator y el vector flujo del rotor, es decir, aumentando γ y, consecuentemente, aumentando el par necesario a aplicar, lo cual se estima con lo explicado anteriormente.

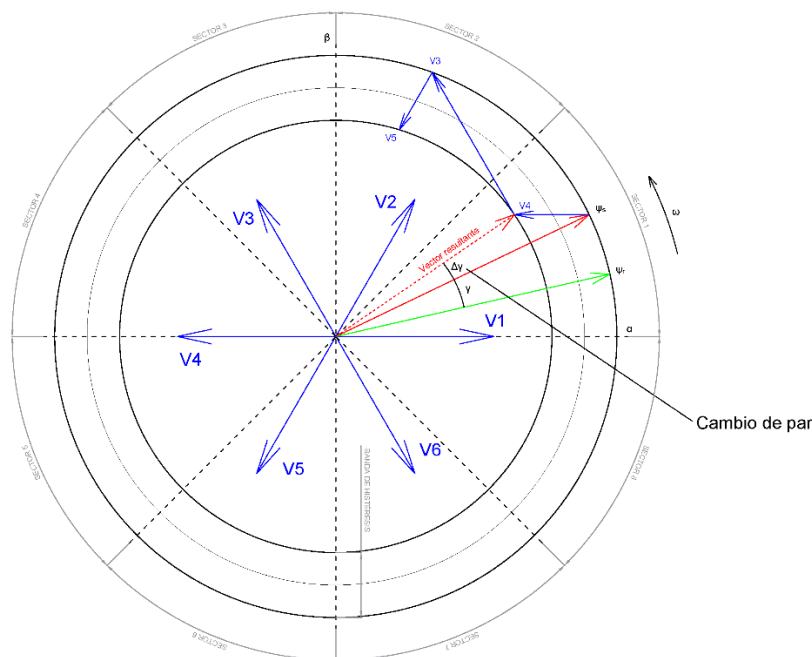


Figura 16. Funcionamiento gráfico del control directo de par – Elaboración propia

5.1.2.2. CONTROL DC

En cuanto al control realizado en corriente continua, se ha implementado únicamente un tipo, el *Chopper DC*, el cual, a nivel de funcionamiento, como veremos a continuación, es muy simple. Además, ofrece un buen control de la señal, aunque no de manera muy eficiente.

5.1.2.2.1. CHOPPER DC

El *Chopper DC* es un convertidor DC/DC el cual permite generar una tensión media de salida ajustable a partir de una tensión de entrada fija.

Para ello, cuenta con un semiconductor, en este caso un MOSFET, el cual conmuta a una alta frecuencia, permitiendo obtener una tensión media de salida proporcional al ciclo de trabajo programado. Además, posee un diodo de retorno, simulando también la eliminación del retorno, ya que en este caso las cargas son inductivas.

De esta forma, obtenemos el siguiente circuito modelado.

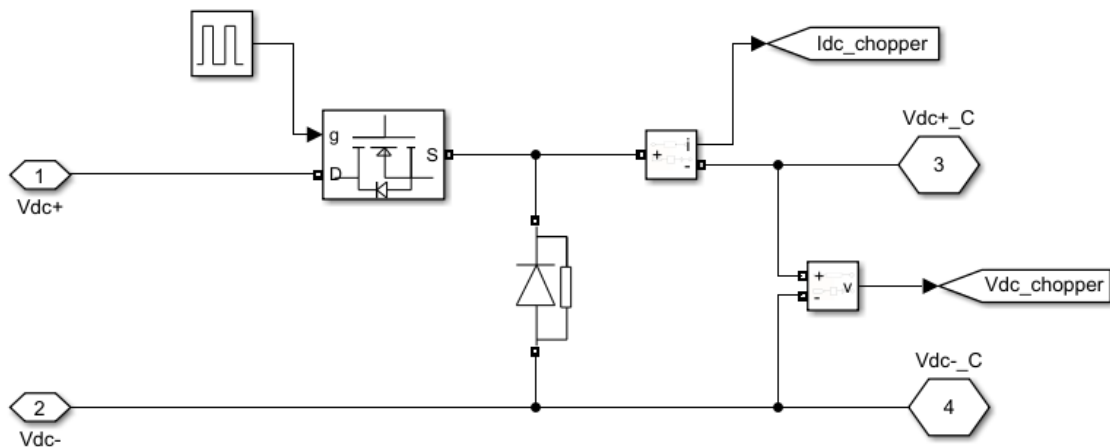


Figura 17. Modelado del Chopper DC

Finalmente, podemos comentar que, como hemos visto, de esta forma, el valor de tensión suministrado al motor puede ser modificado, lo cual conduce a que la velocidad del motor de DC también puede ser modificada.

5.1.3. MOTOR

El siguiente subsistema, denominado *Motor*, contiene los diferentes tipos de máquinas eléctricas implementadas.

Por un lado, como se puede ver en la figura adjunta, posee seis entradas, de las cuales tres de ellas corresponden a la alimentación en corriente alterna proveniente del convertidor, las dos siguientes también corresponden a la alimentación proveniente del convertidor, pero en corriente continua, y la última pertenece al par de carga, el cual proviene del subsistema *Proceso*, que, como describiremos más adelante, puede ser fijo o variable.

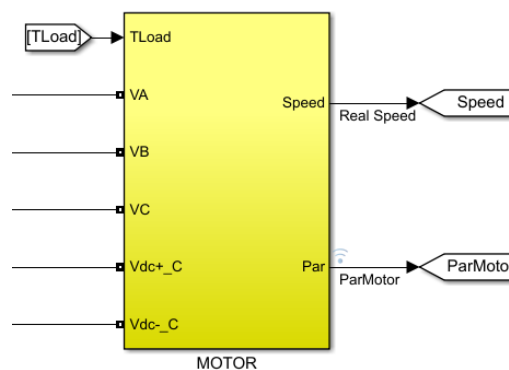


Figura 18. Subsistema Motor

Por otro lado, se puede observar que este subsistema *Motor* posee dos salidas, que concretamente son dos de las principales: la velocidad de giro del motor y el par motor que ejerce.

Dentro de dicho subsistema, distinguimos dos tipos de motores que se han implementado: el motor de inducción y el motor de corriente continua. Estos motores se podrían haber modelado a partir de sus ecuaciones características, sin embargo, a nivel de simulación, no es eficiente. Por tanto, se han utilizado los bloques pertenecientes a la librería *Specialized Power Systems* de *Simscape*, tal y como se muestra a continuación.

Digitalización de un banco de ensayos y comprobación experimental

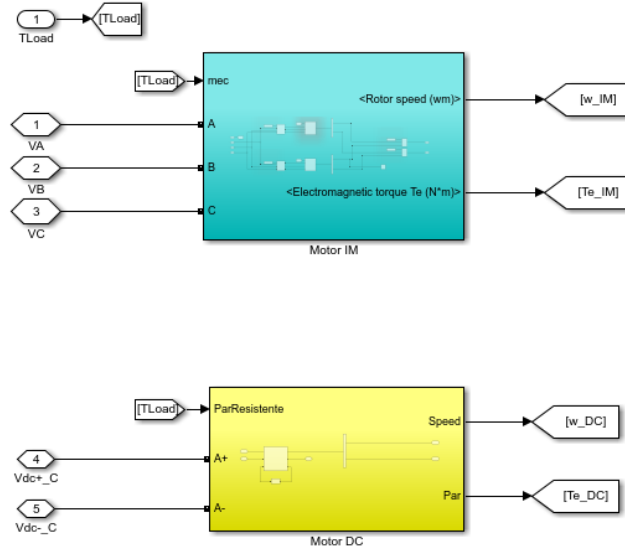


Figura 19. Tipos de motores implementados

Así, podemos observar que la alimentación a los motores de inducción proviene de la salida alterna trifásica del convertidor y la alimentación del motor de corriente continua proviene de la salida continua del convertidor.

De esta forma podemos estudiar más en detalle cada tipo de máquina eléctrica.

5.1.3.1. MOTOR IM

En este primer caso, distinguiremos dos tipos de motores de inducción: uno predefinido y otro personalizable.

El primer caso, se trata de un motor de jaula de ardilla, predefinido en la librería de *Specialized Power Systems*, con las siguientes características principales:

$$\begin{aligned}
 P_n &= 4 \text{ kW} \\
 U_n &= 400 \text{ V} \\
 f_n &= 50 \text{ Hz} \\
 n_s &= 1430 \text{ rpm} \\
 P_{polos} &= 2
 \end{aligned}$$

El segundo caso se compone por un motor de jaula de ardilla también, aunque sus valores característicos son personalizables desde la interfaz gráfica, como se detalla en el punto 7.2. De esta forma, la configuración de dicho bloque de motor asíncrono de

Digitalización de un banco de ensayos y comprobación experimental

inducción quedaría en función de una serie de variables, como se detalla a continuación en la Figura 19.

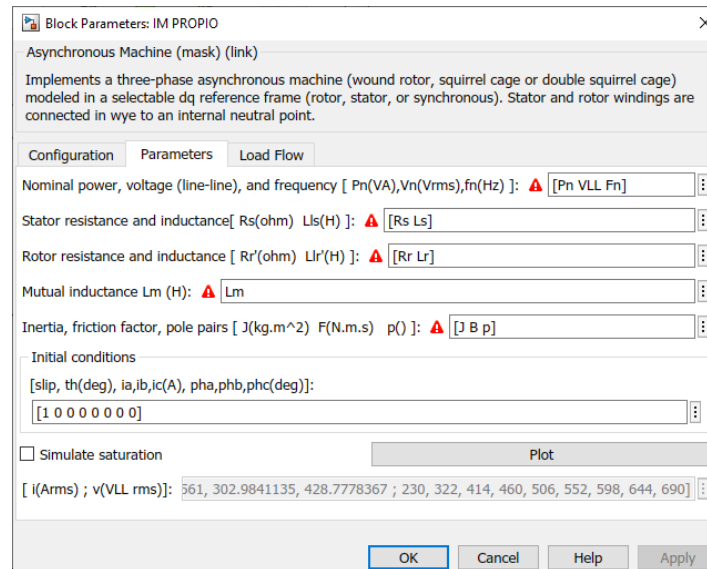


Figura 20. Configuración de motor personalizable

Por otro lado, al igual que en casos anteriores, existen diferentes interruptores que son controlados desde la propia interfaz gráfica, permitiendo elegir un motor, un convertidor, o el tipo de alimentación de red necesario. En este caso, vendrán controlados por las variables *IM_PROPIO*, *IM_SYM* y *IM_selector*, como se puede ver en la Figura 20.

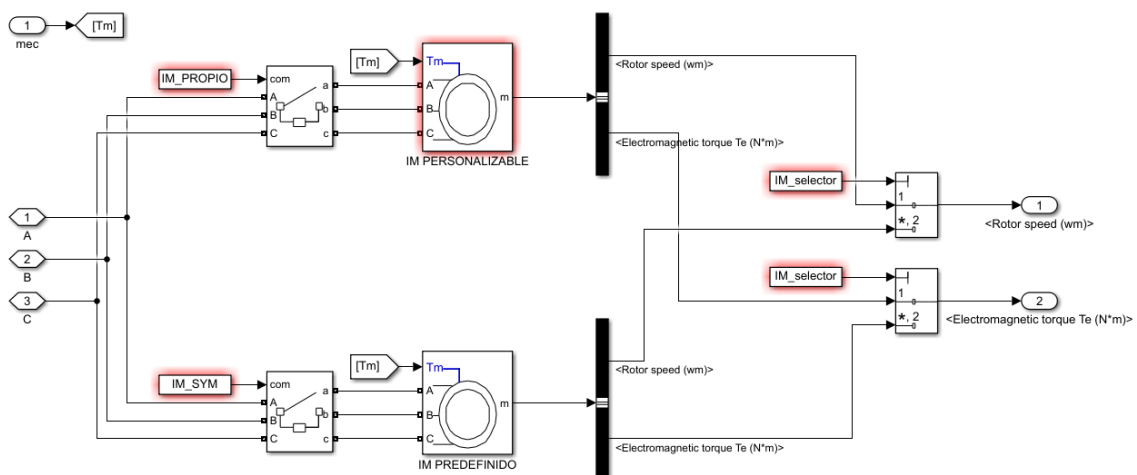


Figura 21. Modelo de motores de inducción

Finalmente, observamos a la derecha de la Figura 20 la selección realizada por la variable *IM_selector*, que permite que la salida del subsistema de *Motor IM* sea la velocidad y el par del motor personalizable o del motor predefinido.

5.1.3.2. MOTOR DC

En este segundo caso, tenemos un motor de corriente continua de excitación separada. Concretamente, las características principales de este motor son:

$$P_n = 5 \text{ HP}$$

$$U_n = 240 \text{ V}$$

$$v_n = 1750 \text{ rpm}$$

$$U_{field} = 150 \text{ V}$$

Donde la tensión de campo U_{field} en este caso se obtiene a partir de una fuente de tensión continua.

Así, tal y como se muestra en la figura siguiente, conseguimos obtener el valor de velocidad (en rad/s, posteriormente convertido a rpm) y de par ejercido por la máquina de corriente continua.

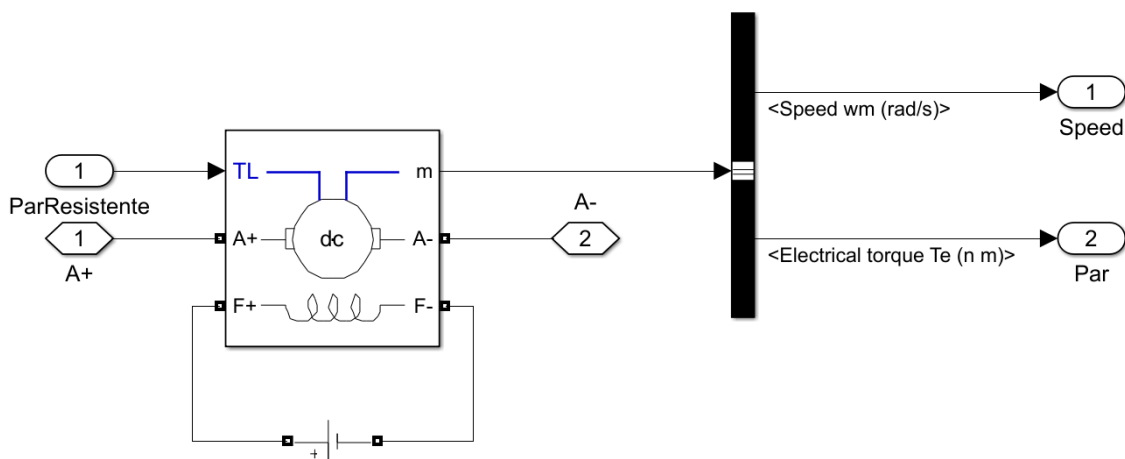


Figura 22. Modelo de motor de corriente continua

5.1.3.3. SELECCIÓN DE MOTOR

Finalmente, como se ha comentado anteriormente, debemos de seleccionar la señal del motor correspondiente, lo cual lo haremos a partir de la variable M_sel_out , como se muestra a continuación. Esta variable, también será controlada desde la interfaz gráfica posteriormente detallada.

Digitalización de un banco de ensayos y comprobación experimental

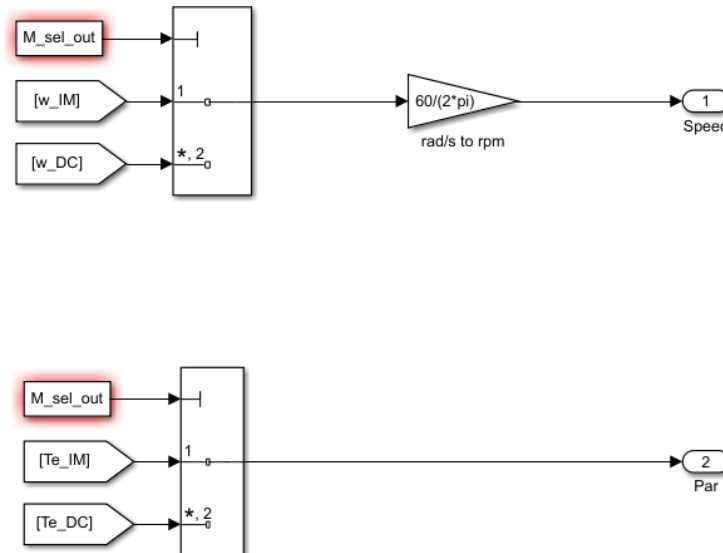


Figura 23. Modelado de selección de velocidad y par de motores

5.1.4. PROCESO

El siguiente subsistema a tratar será el *Proceso*. Este subsistema simula los diferentes tipos de procesos más comunes y presentes en una industria real, a partir de una serie de expresiones matemáticas básicas.

De esta forma, se han implementado cuatro procesos diferentes:

- Par constante: cinta transportadora
- Par cuadrático: bomba centrifugadora
- Potencia constante: bobinadora
- Par constante con cambio de carga: cinta transportadora de varias cargas

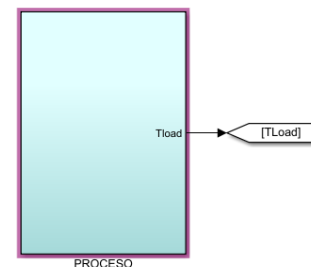


Figura 24. Subsistema Proceso

Los cuales se describen detalladamente a continuación.

5.1.4.1. PAR CONSTANTE: CINTA TRANSPORTADORA

En este primer tipo de proceso, la cinta transportadora se caracteriza por tener siempre el mismo nivel de par resistente. Así, como se muestra en el gráfico adjunto, el par permanece constante independientemente de la velocidad, mientras que la potencia crece proporcionalmente con la velocidad.

Digitalización de un banco de ensayos y comprobación experimental

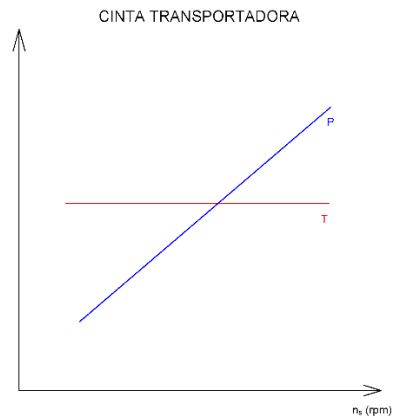


Figura 25. Evolución de par y potencia en función de velocidad en cinta transportadora



Figura 26. Cinta transportadora en transporte de paquetería. Fuente: Forbo Movement

Por tanto, en este caso, el par resistente siempre será el indicado en la interfaz gráfica por el usuario, T_{load_ref} .

Cinta transportadora

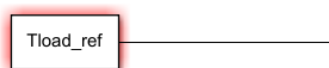


Figura 27. Modelo de par constante

5.1.4.2. PAR CUADRÁTICO: BOMBA CENTRIFUGADORA

En este caso, el par resistente es proporcional al cuadrado de la velocidad y, de igual forma, la potencia consumida, como se muestra en el gráfico siguiente. Este fenómeno ocurre principalmente en procesos que involucren el movimiento de fluidos, como puede ser la bomba centrífuga simulada.

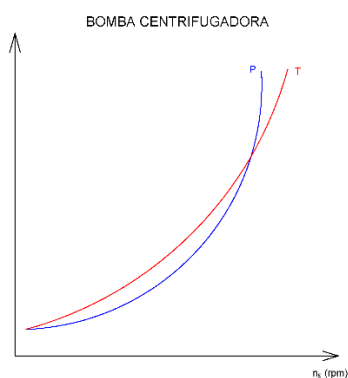


Figura 28. Evolución de par y potencia en función de velocidad en bomba centrífuga



Figura 29. Bomba centrífuga en industria química. Fuente: Cindex

Así, la implementación de este proceso en el modelo quedaría de la siguiente forma:

Bomba centrífuga

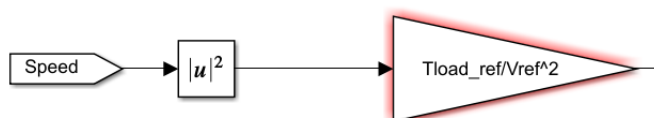


Figura 30. Modelo de bomba centrífuga

Digitalización de un banco de ensayos y comprobación experimental

Donde:

Speed: velocidad instantánea

$T_{load_{ref}}$: par resistente introducido

V_{ref} : velocidad de referencia

Dando lugar al comportamiento descrito en la Figura 25.

5.1.4.3. POTENCIA CONSTANTE: BOBINADORA

En este caso, la potencia ejercida por el motor es siempre idéntica, mientras que el par disminuye de forma inversamente proporcional con la velocidad. Este fenómeno ocurre mayoritariamente en bobinadoras, aunque también en algunas máquinas herramienta de corte o incluso en máquinas laminadoras. Su comportamiento, a nivel gráfico, se describe en la figura siguiente.

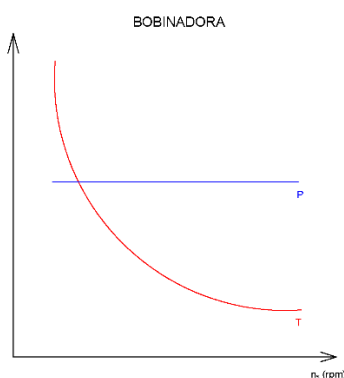


Figura 31. Evolución de par y potencia en función de velocidad en bobinadora



Figura 32. Bobinadora de cable. Fuente: Loimex

De esta forma, el modelo quedaría implementado de la siguiente forma

Bobinadora

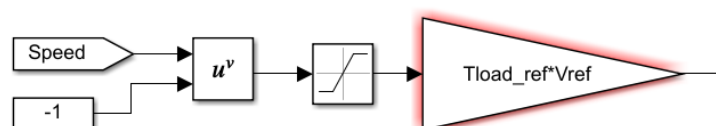


Figura 33. Modelo de bobinadora

donde las variables utilizadas son

Speed: velocidad instantánea
Tload_{ref}: par resistente introducido
Vref: velocidad de referencia

El bloque de saturación se ha introducido con el fin de evitar que la inversa de la velocidad, a velocidades muy bajas, diese valores matemáticamente posibles muy elevados, que a nivel físico no son reales. Así, se puede evitar que durante los primeros instantes de simulación el sistema no se desestabilice.

5.1.4.4. PAR CONSTANTE CON CAMBIO DE CARGA: CINTA TRANSPORTADORA DE VARIAS CARGAS

En este caso, el par también es constante. Sin embargo, se ha programado para que en el instante inicial el par resistente fuese el mismo que para el caso de par constante, *Tload_{ref}*, mientras que en el instante $t = 0.2 s$, el par resistente se reduzca a la mitad, *Tload_{ref}*, permitiendo así observar la evolución del sistema cuando existe una diferencia sustancial de carga.

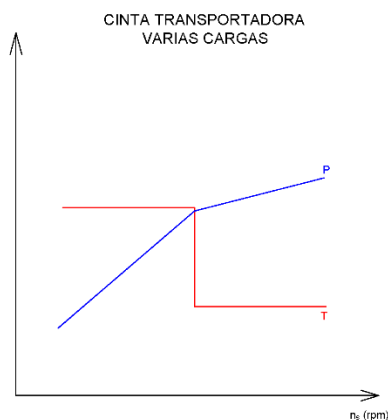


Figura 34. Evolución de par y potencia en función de velocidad en cinta transportadora con varias cargas

Digitalización de un banco de ensayos y comprobación experimental

De esta forma, el modelo quedaría implementado de la siguiente manera

Cinta transportadora varias cargas

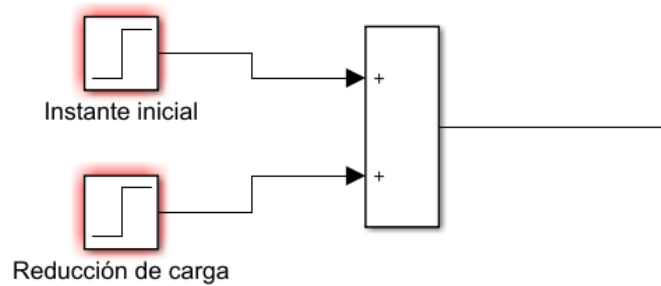


Figura 35. Modelo de cinta transportadora de varias cargas

Donde se puede observar un primer bloque de escalón que se inicializa en el instante inicial, $t = 0 s$, con un par igual a $Tload_ref$, y un segundo bloque que reduce a la mitad la carga, el cual actúa en el tiempo $t = 0.2 s$ y con una carga de $-Tload_ref/2$, como se ha comentado anteriormente.

SUBSISTEMA COMPLETO

Una vez estudiados los procesos del subsistema por separado, obtendríamos de forma conjunta el siguiente modelo:

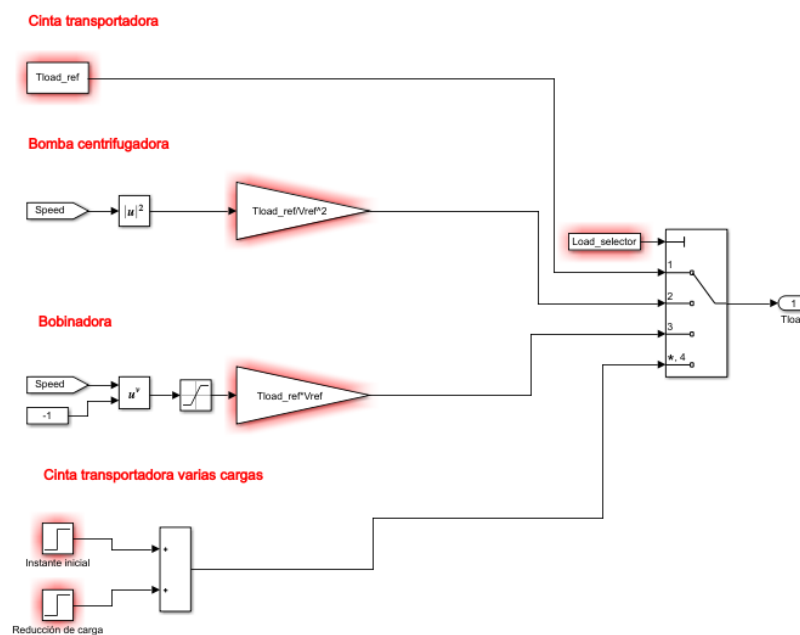


Figura 36. Modelo de procesos completo

Digitalización de un banco de ensayos y comprobación experimental

Se puede destacar, finalmente, el selector a la derecha del modelo, que permite asignar una carga u otra a partir de la variable *Load_selector*, la cual se modifica, de nuevo, a partir de lo seleccionado por el usuario en la interfaz gráfica.

5.1.5. VISUALIZACIÓN

El último subsistema del modelo implementado se denomina *Mediciones*. Este subsistema consta de recapitular las variables del resto del modelo que se desea obtener para enviar a la interfaz gráfica, las cuales son:

- Tensión trifásica de salida del convertidor - ***Vabc***
- Corriente trifásica de salida del convertidor - ***Iabc***
- Tensión del bus de continua del convertidor / Tensión de salida del convertidor de corriente continua - ***Vdc***
- Corriente del bus de continua del convertidor / Corriente de salida del convertidor de corriente continua - ***Idc***
- Velocidad del motor – ***Speed_f***
- Par del motor – ***Tload_f***

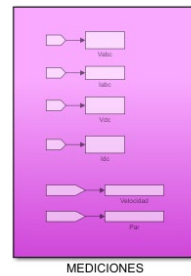


Figura 37. Subsistema Mediciones

Para el envío de dicha información a la interfaz gráfica, se transmite directamente al *Workspace* de *Matlab*, mediante un bloque *To Workspace*, de donde la interfaz diseñada en *AppDesigner*, lee los valores instantáneos, permitiendo realizar una representación gráfica de la evolución de cada variable respecto al tiempo.

Así, el modelo dentro de dicho subsistema es de la siguiente forma:

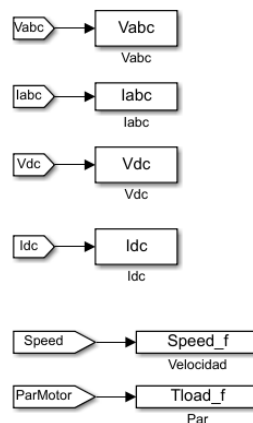


Figura 38. Envío de información a la interfaz gráfica

5.2. DISEÑO DE LA INTERFAZ GRÁFICA

La interfaz gráfica se trata del vínculo principal entre el usuario y la herramienta de simulación. Así, se permite que el usuario no requiera un alto conocimiento de programación, ni de bajo ni de alto nivel, únicamente es indispensable poseer capacidad de interpretar los resultados obtenidos y conocer los diferentes componentes eléctricos de un proceso industrial.

Para el diseño y desarrollo de la interfaz gráfica se ha hecho uso de la herramienta *App Designer* del software Matlab, ya que permite una conexión directa con *Simulink*, donde se encuentra el modelo de simulación. Además, como se describirá a continuación, la herramienta genera una base de proyecto en la cual se permite comenzar a añadir componentes para su programación, sin necesidad de configurar la creación de una interfaz gráfica. Asimismo, se permite modificar parámetros básicos, como pudiera ser icono de la aplicación, versión o título, entre otros, de la propia interfaz gráfica de una forma rápida e intuitiva.

5.2.1. ELEMENTOS UTILIZADOS

Por otro lado, la herramienta *App Designer* posee una Librería de Componentes la cual contiene diferentes elementos que se pueden añadir a la aplicación, como pueden ser botones, gráficas, desplegables, campos de texto o numéricos, etc. Los elementos principales de esta librería son:

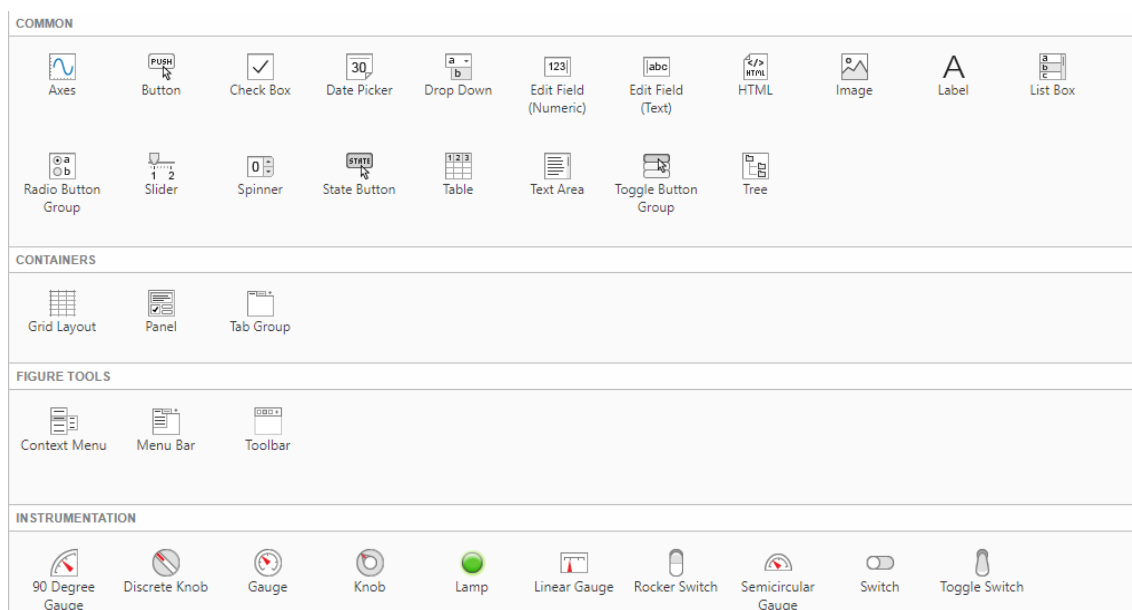


Figura 39. Elementos principales de la librería de componentes

Digitalización de un banco de ensayos y comprobación experimental

Más concretamente, los elementos utilizados para el diseño son los siguientes:

- Título
- Imagen
- Desplegable
- Botón
- Campo numérico editable
- Gráfica
- Botón de selección

Cuya simbología se representa en la Tabla 1.








						
Título	Imagen	Desplegable	Botón	Campo numérico editable	Gráfica	Botón de selección

Tabla 1. Elementos utilizados

5.2.2. COMUNICACIÓN CON SIMULINK

En cuanto a la comunicación entre la interfaz gráfica y el modelo de simulación diseñado en *Simulink*, como se ha comentado en puntos anteriores, se ha realizado a través del espacio de trabajo (*Workspace*) de *Matlab*.

Al iniciar la aplicación, se generan todas las variables posibles y se inicializan a 0. Estas variables comprenden, tanto la selección de convertidor, motor, proceso, etc., como la velocidad de consigna o parámetros que se deseen visualizar, introducido todo por el usuario.

Seguidamente, al realizar una selección o introducción de un valor, la variable correspondiente cambia de valor en el espacio de trabajo de *Matlab*. Más adelante, el modelo de simulación en *Simulink* es capaz de leer las variables correspondientes y aplicarlas en el bloque correcto, de forma que, por ejemplo, un bloque *Constante* que controle que la selección de la señal de salida – velocidad y par – que se envía a la interfaz gráfica puede obtener el valor 0 (ningún valor seleccionado), el valor 1 (enviar valor de motor de corriente alterna) o valor 2 (enviar valor de motor de corriente continua).

Digitalización de un banco de ensayos y comprobación experimental

Hemos observado, por tanto, que la interfaz gráfica puede escribir valores en el *Workspace* y que Simulink puede leer valores del mismo lugar. Además, cada uno puede realizar su operación inversa, es decir, la interfaz puede leer y el modelo de simulación puede escribir. De hecho, la forma en la que se envía la información gráfica que nos proporciona la información de los motores desde el modelo a la interfaz es exactamente la misma.



Diagrama 2. Intercambio de información entre modelo e interfaz

5.2.3. PROGRAMACIÓN DE LA INTERFAZ – FUNCIONES CALLBACK

Para realizar la programación de la interfaz gráfica se han utilizado funciones *Callback*, ya implementadas en el propio *App Designer*. Estas funciones, técnicamente, son funciones que se pasan como parámetros a otras funciones.

Para este caso concreto, la definición es la misma, aunque para una mejor comprensión, las funciones *Callback* son las funciones que se ejecutan cuando se interacciona con alguno de los elementos descritos en la Tabla 1.

De esta forma, cada vez que se escoge alguna opción de uno de los desplegables existentes, se ejecuta una función que lee qué opción del desplegable se ha escogido y en función de dicha elección, se guarda un valor concreto en una variable existente en el *Workspace*. Además, también se permite que, en función de lo escogido en dicho desplegable, otros elementos de la aplicación de inhabiliten o incluso aparezcan o desaparezcan, para evitar confusiones y obtener una interfaz gráfica más intuitiva. Este mismo proceso ocurre en cualquier otro elemento de la aplicación, de forma que obtengamos pleno control sobre las elecciones del usuario.

5.2.4. DISEÑO DE LA INTERFAZ GRÁFICA

A rasgos generales y al iniciar la aplicación, la interfaz gráfica de esta posee el aspecto descrito en la figura siguiente.



Figura 40. Interfaz gráfica

De esta forma, se puede observar en la parte superior de la aplicación diferentes desplegables que permiten la selección de componentes. En la parte central observamos un apartado de visualización de resultados y, finalmente, en la parte inferior de la aplicación podemos ver las consignas de comenzar la simulación y velocidad deseada, como se muestra en la Figura 37.

Así, podemos describir cada uno de los apartados, sus posibilidades y su correspondiente funcionamiento

Digitalización de un banco de ensayos y comprobación experimental



Figura 41. Apartados principales de la aplicación

5.2.4.1. SELECCIÓN DE COMPONENTES

En este primer apartado es posible seleccionar por el usuario cada uno de los componentes de la simulación descrita en el punto 7.1.

Así, de forma esquematizada, la estructura de selección de dichos componentes es la siguiente.

Digitalización de un banco de ensayos y comprobación experimental

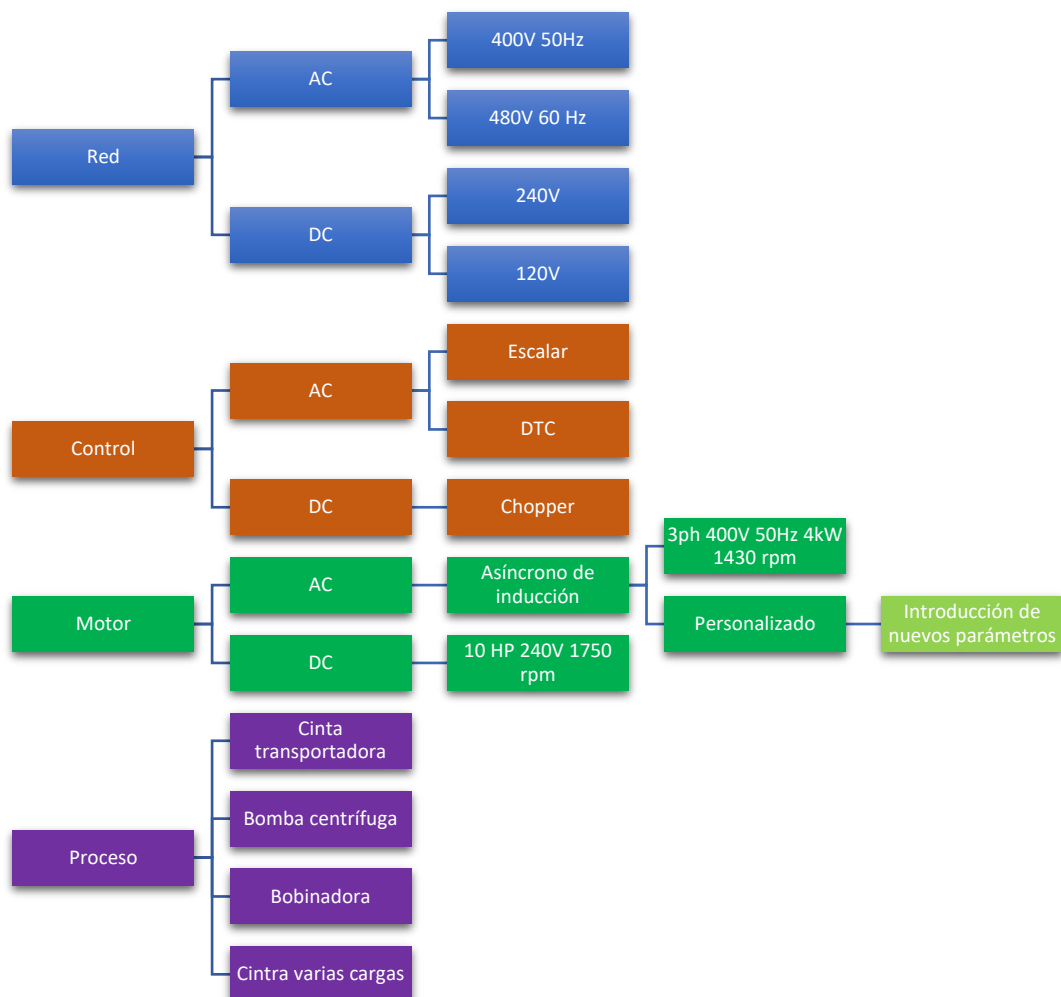


Diagrama 3. Esquema de posibilidades de selección

Seguidamente, para seleccionar en primer lugar el tipo de red a utilizar, se hace click que el desplegable correspondiente y nos ofrece las posibilidades mostradas en el Diagrama 3.

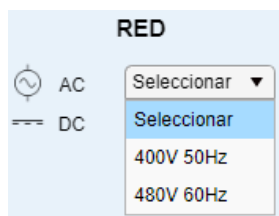


Figura 42. Desplegable para la selección del tipo de red alterna

Así, una vez seleccionada, por ejemplo, cualquier red de corriente alterna, se inhabilitan todas las demás posibilidades de seleccionar motores o controles de corriente continua, evitando así que el usuario seleccione de manera inintencionada

Digitalización de un banco de ensayos y comprobación experimental

componentes incompatibles, aunque a nivel de simulación no genere ninguna pérdida económica ni material. Ocurre la misma dinámica al

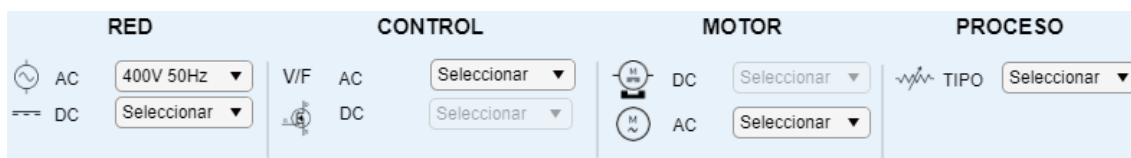


Figura 43. Inhabilitación de componentes DC al seleccionar red DC.

Por otro lado, en cuanto a la selección del tipo de control, se ha realizado de forma simplificada, ya que únicamente se nos permite seleccionar el tipo y no modificar otros parámetros.

Más adelante, en cuanto a la selección del motor, el de corriente continua concretamente no es modificable, mientras que el de corriente alterna nos permite seleccionar entre uno asíncrono de inducción predefinido y otro personalizado, como se muestra en la figura siguiente.

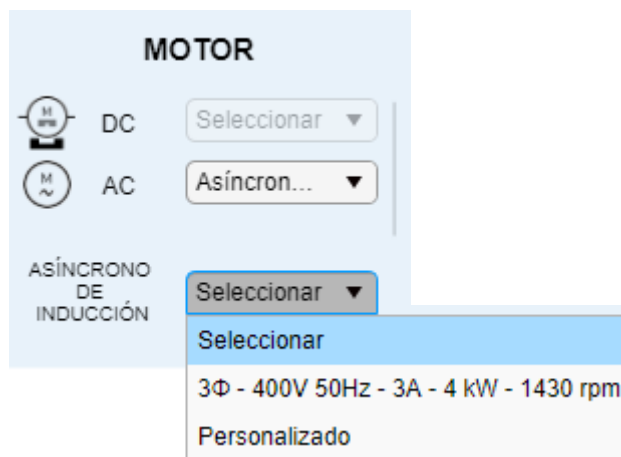


Figura 44. Selección de motor asíncrono

Así, una vez seleccionado el motor personalizado se habilita un botón en pantalla de la forma siguiente

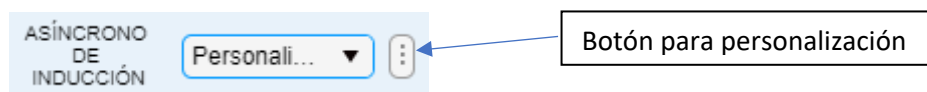


Figura 45. Botón de personalización

el cual ejecuta otra interfaz gráfica la cual nos permite introducir los parámetros característicos de otro motor asíncrono de inducción, tal y como se muestra a continuación.

Digitalización de un banco de ensayos y comprobación experimental

Parámetros motor personalizado

MOTOR DE INDUCCIÓN PERSONALIZADO

Pares de polos Pn (W) ULL (V) J (kg·m²)

B (N·m·s) fn (Hz)

Datos de ensayo

Resistencia estática	Ensayo de vacío	Ensayo de cortocircuito
Ruv (Ω) <input type="text" value="0"/>	U10 (V) <input type="text" value="0"/>	Ucc (V) <input type="text" value="0"/>
Ruw (Ω) <input type="text" value="0"/>	I10 (A) <input type="text" value="0"/>	Icc (A) <input type="text" value="0"/>
Rww (Ω) <input type="text" value="0"/>	P10 (W) <input type="text" value="0"/>	Pcc (W) <input type="text" value="0"/>
	Q10 (VA) <input type="text" value="0"/>	Qcc (VA) <input type="text" value="0"/>

Parámetros del circuito equivalente

Rs (Ω) <input type="text" value="0"/>	Rr (Ω) <input type="text" value="0"/>	
Lm (H) <input type="text" value="0"/>	Lr (H) <input type="text" value="0"/>	Ls (H) <input type="text" value="0"/>

Aceptar Cancelar

Figura 46. Interfaz gráfica de selección de motor personalizado

En este caso se pueden observar, por un lado, en la parte superior de la imagen, los parámetros generales del motor que se desea utilizar. Más concretamente, se detallan las características mecánicas y nominales del motor que, en cualquier caso, se deben introducir.

Por otro lado, debajo de dicho apartado, encontramos el resto de los parámetros eléctricos, los cuales, como se puede ver en la figura, se pueden introducir a partir de la medición de resistencia estática, ensayo de vacío y ensayo de cortocircuito, o directamente a partir del circuito equivalente de la máquina. Esto, permite que el usuario no tenga que realizar las transformaciones correspondientes ahorrando tiempo y evitando posibles errores de cálculo.

Así, los parámetros de la parte superior de la aplicación se envían al modelo de *Simulink* de la misma forma que el resto de parámetros de la aplicación completa. Además, los parámetros del circuito equivalente también se envían o, en caso de ser introducidos los datos de los ensayos, como se ha comentado anteriormente, se calculan los parámetros del circuito equivalente a partir de dichos datos de los ensayos, y se envían al modelo de igual forma.



Digitalización de un banco de ensayos y comprobación experimental

Para el cálculo de dichos parámetros de circuito equivalente se ejecuta una función denominada

function datos_motor_propio_IM(app)

la cual lee las variables globales que almacenan los datos de los ensayos y realiza los siguientes cálculos.

Las resistencias de estator y rotor se rigen por las siguientes ecuaciones

$$R_s = \frac{R_{uv} + R_{uw} + R_{vw}}{3/2}$$

$$R_r = R_{cc} - R_s$$

donde

$$R_{cc} = \frac{P_{cc}}{3 \cdot I_{cc}^2}$$

Por otro lado, la inductancia mutua sigue las siguientes ecuaciones

$$L_m = \frac{X_m}{2 \cdot \pi \cdot 50}$$

donde

$$X_m = \frac{U_{10}^2}{Q_{10}}$$

Y las inductancias de rotor y estator las siguientes

$$L_s = L_r$$

$$L_s = \frac{X_{ds}}{2 \cdot \pi \cdot 50}$$

donde

$$X_{ds} = \frac{X_{cc}}{2}$$

Finalmente, el resto de parámetros nominales y características mecánicas del motor ensayado se envían de la misma forma.

En cuanto al proceso, tendremos las posibilidades de selección descritas en el Diagrama 3.

Digitalización de un banco de ensayos y comprobación experimental

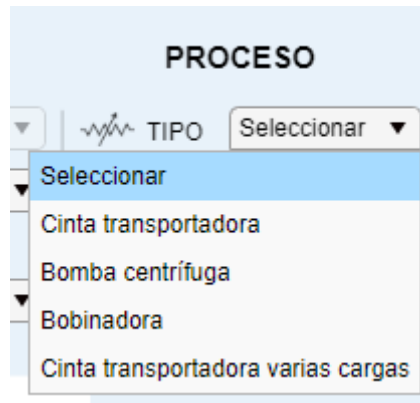
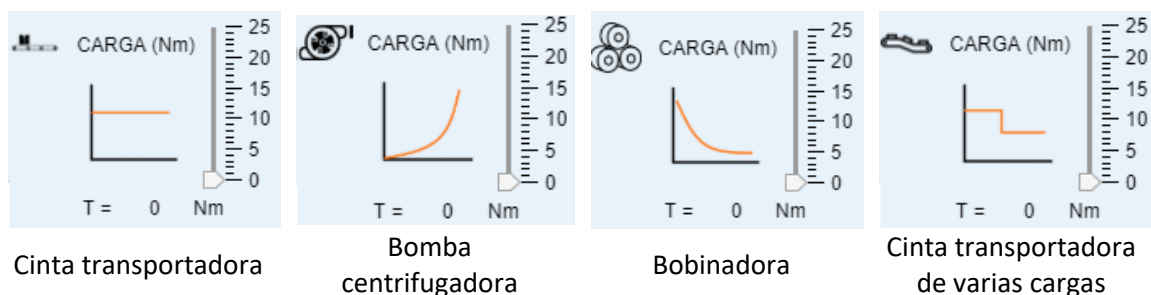


Figura 47. Desplegable de proceso

Así, una vez seleccionado el tipo, se habilita un *Slider* el cual posibilita la selección del nivel de carga que se desea utilizar. Además, se muestra, para cada tipo, un icono con la evolución de la carga similar a lo indicado en la Figura 24, Figura 26, Figura 28 y Figura 30. Además, se muestra el valor exacto seleccionado en el *Slider* debajo de dicho icono para obtener una visión más precisa de la carga suministrada.



De esta forma, el valor obtenido en dicho *Slider* se envía al modelo de *Simulink* de la misma forma, a través del espacio de trabajo de *Matlab*.

5.2.4.2. VISUALIZACIÓN DE RESULTADOS

Por otro lado, tenemos el apartado de visualización de resultados, donde podemos observar que existe un componente de la librería denominado *Gráfica* a la izquierda de la pantalla y otro componente denominado *Botón de selección* a la derecha.

Digitalización de un banco de ensayos y comprobación experimental

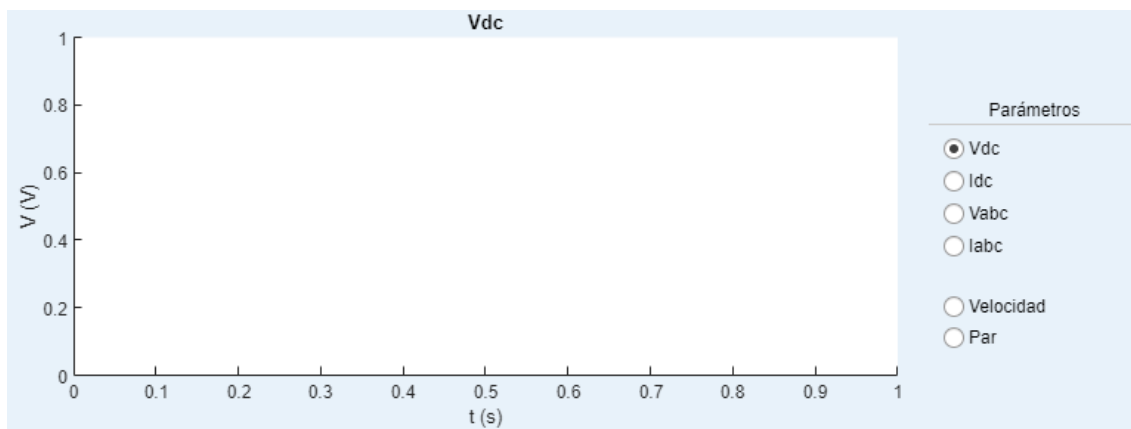


Figura 48. Apartado de visualización

En primer lugar, en cuanto a la gráfica, se muestra por defecto el valor de V_{dc} . Además, este tipo de gráficas posee la opción de exportar como imagen el resultado, ampliar o reducir, e incluso poder ver el valor del eje Y en cada instante de tiempo. Todo ello, se realiza mediante los botones que aparecen en la parte superior derecha de la gráfica al situar el cursor encima, excepto el último, que se obtiene interactuando directamente con la gráfica y haciendo click en el propio resultado.

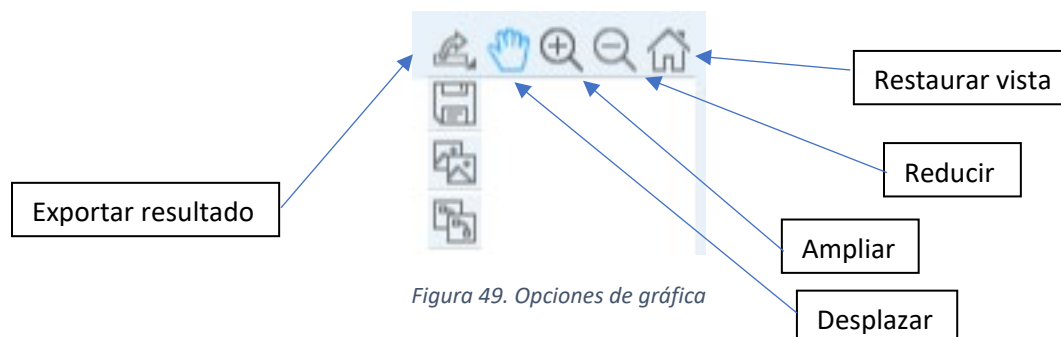


Figura 49. Opciones de gráfica

En segundo lugar, en cuanto al botón de selección, tenemos los diferentes parámetros que se pueden visualizar. Por defecto siempre que se ejecuta el sistema se selecciona automáticamente V_{dc} , al igual que la gráfica.

Así, podemos observar que los parámetros que se nos permite visualizar son los mismos que se enviaban desde el bloque de *Medición*:

- V_{dc} : tensión del bus de continua del inversor (en caso de AC) ó tensión de alimentación continua (en caso de DC).
- I_{dc} : corriente del bus de continua del inversor (en caso de AC) ó corriente de alimentación continua (en caso de DC).
- V_{abc} : tensión de alimentación al motor trifásica

Digitalización de un banco de ensayos y comprobación experimental

- Iabc: corriente de alimentación al motor trifásica
- Velocidad
- Par ejercido

De esta forma, en función de lo seleccionado en este *Botón de selección*, se cambia lo mostrado en la gráfica del mismo apartado.

5.2.4.3. CONSIGNAS

En último lugar, tenemos el apartado de consignas, el cual, por un lado, posee un campo numérico editable que permite ajustar la velocidad de referencia del modelo modificando a su vez la variable V_{ref} del propio espacio de trabajo de *Matlab* y, consecuentemente, la misma en el modelo de *Simulink*. Por otro lado, posee el botón de *Start* el cual inicia la simulación. Este botón únicamente se encuentra activo cuando se ha seleccionado tanto el tipo de red, control, motor y proceso, como la velocidad de referencia descrita en este mismo punto, con el fin de evitar incongruencias en el resultado final mostrado en la gráfica central de la aplicación.

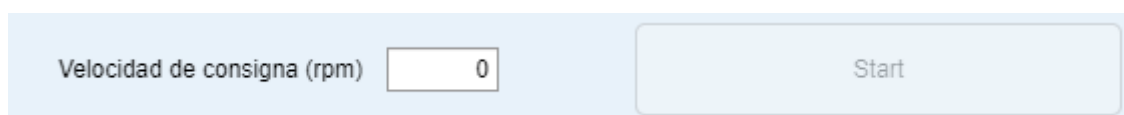


Figura 50. Apartado de consignas

5.3. DESARROLLO DEL PROTOTIPO DE MEDICIÓN

Para garantizar una correcta interpretación de los resultados mostrados por la simulación se ha desarrollado un sistema de medición que permita conocer dos de las variables más importantes: velocidad y par.

Así, se ha tratado de que sirva además como sistema de medición no solo para este caso en concreto, sino también para otros usos que puedan aparecer en laboratorios de la entidad. Por ello, se han implementado varios tipos de medición tanto de par como de velocidad para así poder comparar la precisión en la medición con cada sensor.

De esta forma, se han implementado una serie de selectores que permiten seleccionar el sensor de par o el sensor de velocidad más adecuado.

Digitalización de un banco de ensayos y comprobación experimental

Por otro lado, se ha instalado, de la misma forma, una pantalla LCD que permite la visualización de dichos parámetros en tiempo real.

Seguidamente, se ha implementado además, una salida de 0 a 10 V, la cual es proporcional a la velocidad y que servirá como realimentación de velocidad a los diferentes aparatos de control de máquinas eléctricas.

Así, se ha introducido también un pulsador cuya finalidad es resetear el microcontrolador que recibe y procesa las señales de los sensores y envía la información relevante a la pantalla LCD, además de generar la realimentación de velocidad. En este caso, el microcontrolador utilizado es el ESP32 WROVER-B.

Se indica en el apartado *PLANOS* del presente documento el circuito completo del sistema de medición.

De esta forma, el sistema implementado se refleja en la figura siguiente.



Figura 51. Sistema de medición implementado

Digitalización de un banco de ensayos y comprobación experimental

Se procede a continuación a la explicación detallada de cada uno de los sistemas descritos previamente.

5.3.1. ESP32 WROVER – B

Para esta aplicación concreta se ha decidido utilizar el ESP32 WROVER – B, debido a varias razones:

- Tamaño reducido
- 4 MB de memoria flash
- 8 MB de PSRAM
- Frecuencia de procesador desde 80 MHz hasta 240 MHz
- Bluetooth
- Diferentes protocolos de comunicación: SPI, I2C.
- Posee también WiFi, pero en este caso no se utiliza.
- Programación sencilla mediante el IDE de Arduino
- Amplio número de PINs
- Dos núcleos

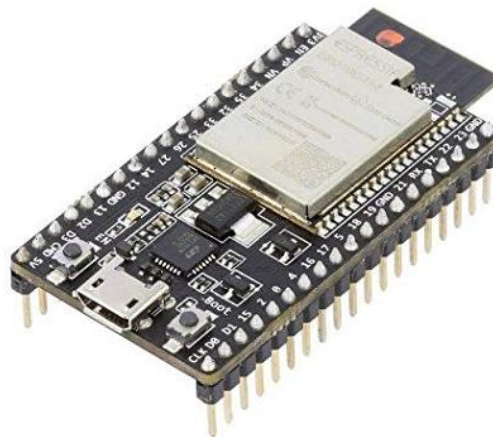


Figura 52. ESP32 WROVER-B

5.3.2. MEDICIÓN DE VELOCIDAD

En cuanto a la medición de velocidad, se han implementado dos sensores diferentes:

- Encoder
- Óptico

Para obtener la velocidad a partir de estos sensores, se debe conocer la onda que genera cada uno.

En este caso, las ondas de ambos sensores son idénticas, cuadradas y de frecuencia proporcional a la velocidad. Sin embargo, la resolución del Encoder es considerablemente superior a la del Óptico, por lo que ambas señales no se pueden tratar de la misma forma. Así, se debe de distinguir un sensor u otro, utilizando un selector posteriormente definido, para que, mediante software se pueda aplicar la resolución adecuada a la conversión entre frecuencia y velocidad.

La conexión para ambos sensores se realiza tal y como se detalla en la Figura 49.

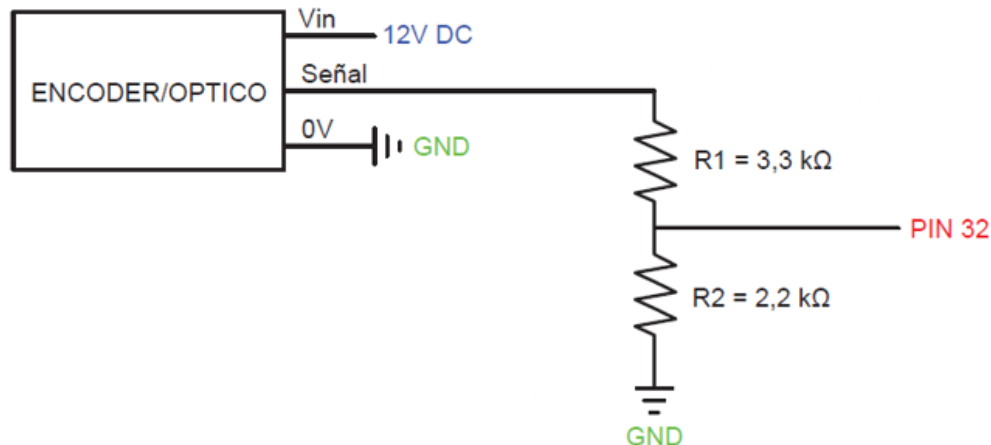


Figura 53. Conexión de Encoder y Óptico

5.3.2.1. ENCODER

El Encoder es uno de los sensores de velocidad más utilizados en la industria principalmente debido a su precisión, precio y fiabilidad.

Su funcionamiento básico es sencillo ya que se basa principalmente en la emisión de luz a través de un disco, con ranuras como se muestra en la figura siguiente, el cual se

Digitalización de un banco de ensayos y comprobación experimental

encuentra anclado al eje del motor. Así, el disco gira a la misma velocidad que el motor.

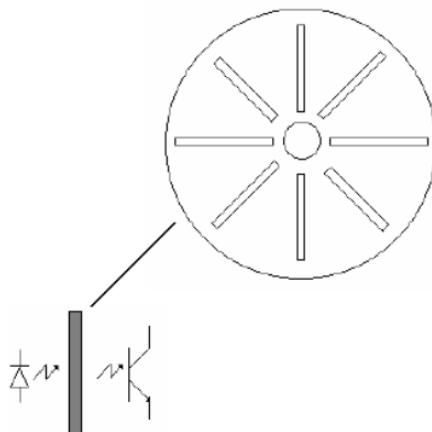


Figura 54. Encoder - Recursos UPV

Se emite, por tanto, la luz a través de dicho disco y se recibe al otro lado del disco mediante un diodo fotoemisor y un transistor fotoreceptor. De esta forma, a medida que aumenta la velocidad del motor, las veces que se activa por segundo dicho transistor fotoreceptor es mayor.

Así, en este caso, el disco cuenta con 1024 ranuras lo que significa que la resolución del encoder será

$$\text{Resolución} = 1024 \frac{\text{pulsos}}{\text{vuelta}}$$

De esta forma, obtenemos una onda cuadrada cuya frecuencia máxima vendrá determinada por la ecuación siguiente, donde la velocidad máxima n_{max} será igual a 1500 rpm, ya que es la velocidad nominal del motor presente en el laboratorio.

$$f_{max} = \frac{\text{Resolución} \cdot n_{max}}{60} \rightarrow f_{max} = \frac{1024 \cdot 1500}{60} = 25600 \text{ Hz}$$

Por otro lado, el encoder utilizado requiere de una tensión de alimentación de 12 V en corriente continua para poder funcionar. Así, obtenemos una señal de onda cuadrada cuya tensión máxima es de 7 V. Sin embargo, el microcontrolador no admite más de 3.3 V a la entrada, por lo que no sería posible leer la señal, lo que ocurriría sería que acabaría dañando el microcontrolador irreversiblemente.

Por tanto, se ha implementado un divisor de tensión como se muestra en la Figura 49, cuya finalidad es reducir dicha tensión a un valor máximo de:

$$V_{encPIN} = 2.33 \text{ V}$$



Digitalización de un banco de ensayos y comprobación experimental

Así, la señal de entrada en el PIN 32 del microcontrolador ESP32 WROVER – B cumple con las especificaciones máximas. Lo siguiente a implementar es la transformación de dicha señal en valores de frecuencia manejables en el software.

Para ello, se utilizan interrupciones. Las interrupciones son detecciones de algún suceso en algún momento del programa. En este caso, lo que se busca es detectar cuándo, en el PIN 32, existe un flanco de subida, que representaría el comienzo positivo de la señal cuadrada de entrada. Una vez detectado dicho flanco de subida, se anotará en una variable denominada *flancos*, para que, al cabo de 1 segundo, se pueda determinar el número de flancos de subida, o, en otras palabras, la frecuencia de la onda cuadrada. Para ello, se ha hecho uso de la función

attachInterrupt(digitalPinToInterrupt(pinFreq), mideFrecuencia, RISING);

donde el primer parámetro de la función es el PIN 32 en este caso, donde tenemos la señal cuadrada de entrada, el tercer parámetro es la condición que se debe cumplir para ejecutar la función del segundo parámetro, que en este caso debe detectar un flanco de subida (RISING) para ejecutar la función *mideFrecuencia*, la cual lo único que realiza es aumentar la variable *flancos* una unidad cada vez que se ejecuta, como se ha comentado anteriormente.

Así, a partir de la frecuencia medida, al cabo de cada segundo, se puede calcular la velocidad de giro de la misma forma que calculábamos la frecuencia máxima:

$$n = \frac{60 \cdot f}{\text{Resolución}}$$

De esta forma, el valor obtenido se representa en pantalla cada segundo, exactamente después de ser calculado.

5.3.2.2. ÓPTICO

En cuanto al sensor óptico, se trata de un tipo de sensor cuyo funcionamiento se basa en la emisión y recepción de una señal la cual rebota en una serie de cuatro adhesivos situados en el rotor del motor. De esta forma la señal recibida también es cuadrada. Sin embargo, es de mucha menor resolución, ya que únicamente puede recibir 4 pulsos por vuelta, a diferencia de los 1024 que permitía el encoder.

$$\text{Resolución} = 4 \frac{\text{pulsos}}{\text{vuelta}}$$



Figura 55. Sensor óptico

Se puede observar, por tanto, que la precisión de la medida de este sensor es mínima, por lo que solo se utilizará con fines didácticos. En cualquier otro caso, el sensor más conveniente de utilizar es el encoder.

Por todo lo demás, el funcionamiento es el mismo que en el caso del encoder. Se alimenta a 12 V y genera una señal cuadrada de valor máximo igual a 7V la cual se reduce a 2.33 V gracias a un divisor de tensión. Así, a través de las interrupciones realizadas por software, se miden los pulsos obtenidos por segundo y se calcula la velocidad del motor a partir de la frecuencia de dichos pulsos, teniendo en cuenta que la resolución en este caso es diferente, cuyo valor se modifica a partir de los selectores descritos a continuación.

5.3.2.3. SELECTORES

En cuanto a los selectores, tendremos uno de dos posiciones, el cual contiene una entrada de alimentación a 3.3 V y dos salidas, una dirigida al PIN 22 y otra dirigida al PIN 23. Ambos pines contienen una resistencia de 10 k Ω unida con la tierra, de forma que la lectura de la posición seleccionada pueda ser más precisa.

De esta forma, en función de lo recibido en los pines 22 y 23, se tomará una decisión correspondiente en cuanto a lo que mostrar por pantalla y a la resolución utilizada para el cálculo de la velocidad

Así, el circuito correspondiente es el siguiente

Digitalización de un banco de ensayos y comprobación experimental

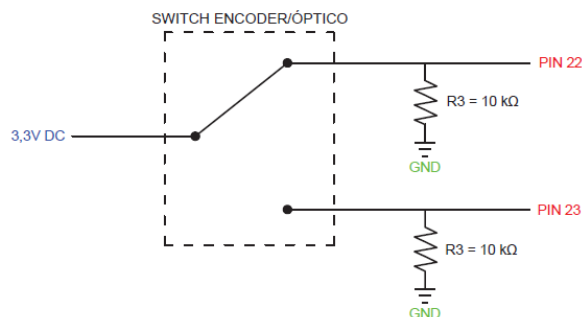


Figura 56. Selector Encoder - Óptico

5.3.3. MEDICIÓN DE PAR

En cuanto a la medición de par, se han implementado los dos sensores siguientes:

- Sensor de efecto Hall
- Pinza amperimétrica

En este caso, cada señal tiene una forma particular, aunque ambas senoidales, una de ellas está desplazada verticalmente. Por tanto, cada una deberá de ser tratada de forma particular, por lo que se hará uso de unos relés para que, puesto que la entrada para ambas señales es la misma, se pueda asignar un tratamiento de señal u otro, en función de lo seleccionado. Así, se muestra a continuación el circuito correspondiente:

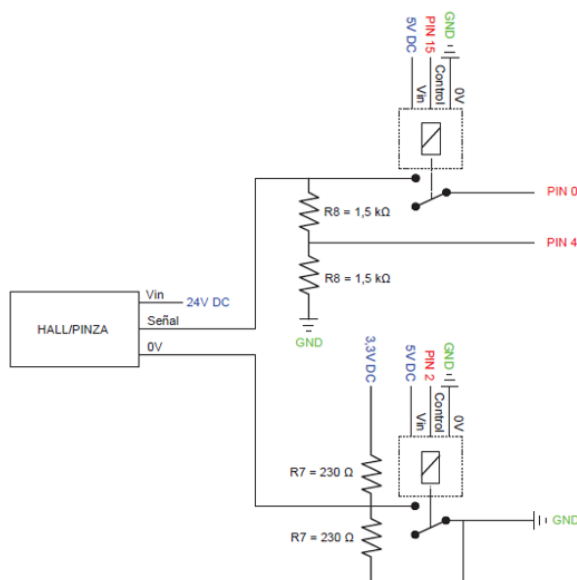


Figura 57. Circuito de medición de par

Digitalización de un banco de ensayos y comprobación experimental

De esta forma, lo medido en cada caso corresponderá con la tensión eficaz (RMS) de la onda, desplazado cierto valor positiva o negativamente, dependiendo del caso. Así, para corregir dicho desplazamiento, se asignan en cada caso unas constantes, las cuales permiten ajustar dicha medición de tensión RMS al valor real.

Además, la tensión es medida directamente en la carga, la cual en este caso es un servomotor controlado con otro variador en modo control de par.

Finalmente, podemos comentar que, en ambos casos, la medida del valor de tensión se ha realizado haciendo uso de un temporizador, por software. Así, se han tomado medidas de la tensión cada milisegundo y se ha almacenado cada valor en un vector, para que, al obtener mil muestras, se pueda realizar la media cuadrática con un valor suficiente de muestras, que nos puedan brindar un resultado lo más preciso posible. Para ello, se ha realizado el siguiente cálculo con todas las muestras.

$$V_{rms} = \sqrt{\sum_{i=1}^{N=1000} \frac{v_i^2}{N}}$$

5.3.3.1. SENSOR DE EFECTO HALL

En cuanto al sensor de efecto hall, se ha utilizado un modelo similar al HTSR25, el cual, como se ve en la imagen siguiente, posee un orificio donde es posible situar el cable de la carga, el cual genera una tensión inducida en el sensor y éste la detecta y la envía como salida. De esta forma, se consigue un aislamiento entre los circuitos de igual manera, evitando posibles accidentes que puedan ocurrir.



Figura 58. Sensor de efecto hall

La señal generada por este sensor se trata de una señal, como se ha comentado anteriormente, alterna, pero desplazada verticalmente hasta el punto de únicamente contener parte continua. Esta señal, por defecto está centrada en 2.5 V y con carga máxima, el valor máximo de la onda se sitúa cercano a los 0 – 5 V.

Digitalización de un banco de ensayos y comprobación experimental

De esta forma, se ha decidido implementar un divisor de tensión, el cual rebaje a la mitad dicha amplitud de onda, utilizando dos resistencias del mismo valor. Así, se consigue que la onda se centre en 1,25 V y como máximo llegue a 0 – 2,5 V, valores que son admisibles para introducir al microcontrolador a través del PIN 4.

Por otro lado, como se puede observar en la Figura 53, la alimentación se realiza de 24V. La alimentación de este sensor debe de ser de 5V, sin embargo, al encontrarse ya instalado en el laboratorio, se ha de realizar a 24V, ya que posee un divisor de tensión instalado para evitar introducir otros niveles de tensión en proyectos previos.

Finalmente, una vez obtenida la tensión eficaz correspondiente, se procede a calcular, cada segundo, el par ejercido por la máquina, el cual se rige por la siguiente ecuación:

$$T = V_{rms} \cdot K_{res} \cdot K_{volt2curr} \cdot K_{curr2torque} \cdot K_{ajuste1}$$

Donde K_{res} es la constante que relaciona la tensión real y la resolución de la entrada del microcontrolador

$$K_{res} = \frac{3,3}{4095}$$

$K_{volt2curr}$ es la constante que convierte la tensión en corriente

$$K_{volt2curr} = 10$$

$K_{curr2torque}$ es la constante que convierte la corriente en par, a partir de los parámetros nominales del motor

$$K_{curr2torque} = \frac{T_n}{I_n} \rightarrow K_{curr2torque} = \frac{15.5 \text{ N}}{14.4 \text{ A}} = 1.076$$

Y $K_{ajuste1}$ es la constante que ajusta el valor obtenido al real, lo cual se realiza cambiando dicho valor y obteniendo a carga máxima un valor de par igual a 15.5 N.

5.3.3.2. PINZA AMPERIMÉTRICA

En cuanto a la pinza amperimétrica, el funcionamiento es el mismo. La principal diferencia es que esta pinza es manejable desde fuera, es decir, se puede poner o quitar de manera sencilla sin necesidad ni de desconectar el circuito ni de desarmar parte de él.

Digitalización de un banco de ensayos y comprobación experimental

Además, a diferencia del caso anterior, este sensor no requiere de alimentación, pues únicamente con la tensión inducida se puede generar una salida, la cual es una señal alterna senoidal pura.

Puesto que el microcontrolador ESP32 WROVER – B no es capaz de leer de manera correcta las señales de alterna, se ha introducido otro divisor de tensión, como se puede observar en la Figura 53, en la tierra de la señal. De esta forma, la señal queda similar a la del sensor de efecto hall, sin componente alterna, pero desplazada de forma diferente.

Así, la onda resultante queda centrada en 1.5 V, teniendo su máximo y su mínimo, respectivamente en 3V y en 0V, siendo admisible para su medición con el microcontrolador escogido.

El cálculo del par se realiza de la misma forma que en el caso anterior, tomando un número suficiente de muestras durante un segundo, calculando la media cuadrática y realizando la siguiente ecuación, donde podemos observar que $K_{ajuste1}$ en este caso ya no existe y se ha sustituido por $K_{ajuste2}$, ya que este valor requiere un ajuste ligeramente diferente.

$$T = V_{rms} \cdot K_{res} \cdot K_{volt2curr} \cdot K_{curr2torque} \cdot K_{ajuste2}$$

Una vez calculado, se envía a la pantalla LCD, actualizando así el valor mostrado cada segundo.

5.3.3.3. SELECTORES Y RELÉS

En cuanto al selector, su funcionamiento es prácticamente similar al utilizado en el caso del sensor de velocidad. La única diferencia es que, por un lado, se encuentra conectado a otros pines, tal y como se muestra en la figura siguiente, y, por otro lado, controlan, a través del software, los dos relés mostrados en la Figura 53, de forma que se utilice un divisor de tensión u otro.

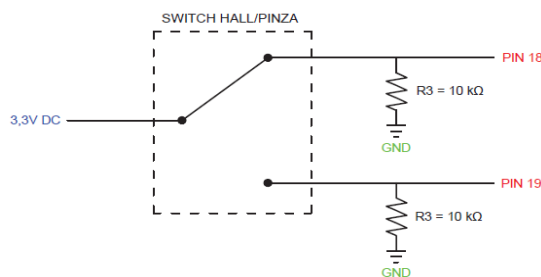


Figura 59. Circuito de selectores de par

5.3.4. SALIDA 0 – 10 V

Seguidamente, tenemos una salida de tensión del microcontrolador ESP32 WROVER – B, la cual, como se ha comentado anteriormente, es proporcional a la velocidad de giro del motor y sirve como realimentación para los controles de velocidad.

El microcontrolador utilizado permite que las salidas analógicas tengan una tensión comprendida entre 0V y 3.3V, por lo que para conseguir dichos 10V, deberemos utilizar un amplificador operacional, concretamente el OPA705. El circuito resultante se detalla en la Figura 56.

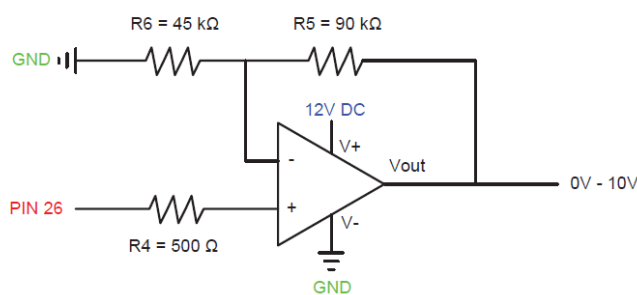


Figura 60. Circuito amplificador de tensión

De esta forma, de acuerdo con su equivalente matemático, la ganancia del sistema viene determinada por:

$$A_v = 1 + \frac{R_5}{R_6}$$

Por lo que nuestra salida por el PIN 26 nos proporcionaría, de acuerdo con los valores escogidos en el circuito, una salida de valor

$$V_{out} = V_{PIN26} \cdot 3$$

Así, conseguimos una tensión de salida máxima muy próxima a 10 V, suficiente para este caso, y proporcional a la velocidad.

5.3.5. RESET

Por otro lado, tenemos el botón de Reset. Este botón es simplemente un pulsador que permite, o no, el paso de la corriente, entre la salida de 3,3 V del microcontrolador y una de las entradas analógicas del mismo. De esta forma, detectaremos que está

Digitalización de un banco de ensayos y comprobación experimental

pulsado cuando exista una tensión positiva en dicho PIN, concretamente el 5, y posteriormente se ejecutará la función que resetea el dispositivo:

ESPrestart();

De esta forma conseguimos que, si por algún motivo el programa se encuentra parado o funcionando correctamente, podamos reiniciar el microcontrolador.

El circuito es muy simple también, se detalla a continuación:

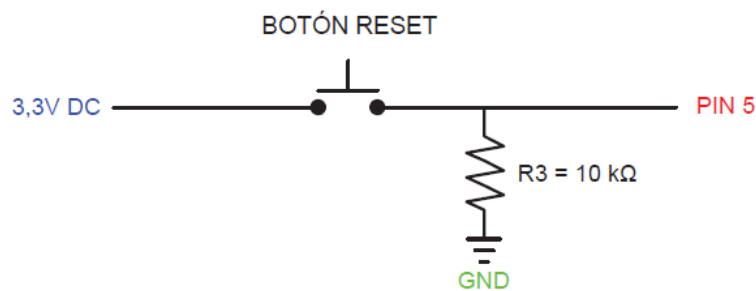


Figura 61. Circuito de Reset

5.3.6. PANTALLA LCD

Más adelante, se ha implementado una pantalla LCD de tamaño 20x4 casillas, la cual permite mostrarnos los valores en tiempo real del sistema. Concretamente, se muestran los valores de velocidad y par, como se ha mencionado en otros apartados, cada segundo.

La pantalla utilizada se trata de una NHD-0420D3Z-NSW-BBW-V3_revB, la cual se ha programado utilizando una librería particular denominada

nhdlcd.h

Además, el protocolo de comunicación utilizado para transferir información a la pantalla LCD es SPI, aunque también podría haberse utilizado I2C. Así, al usar SPI, se ha tenido que cortocircuitar la resistencia R2 presente en la parte posterior de la pantalla.

Por otro lado, los pines utilizados son los siguientes:

- 1 – *Slave: PIN 25*
- 2 – *Sin conectar*
- 3 – *Clock: PIN 14*
- 4 – *Data: PIN 13*
- 5 – *GND*
- 6 – *5V DC*

Digitalización de un banco de ensayos y comprobación experimental

De esta forma y mediante las funciones predefinidas en la librería mencionada, se ha conseguido transmitir la información a la pantalla, la cual ha resultado con un diseño similar al mostrado a continuación.

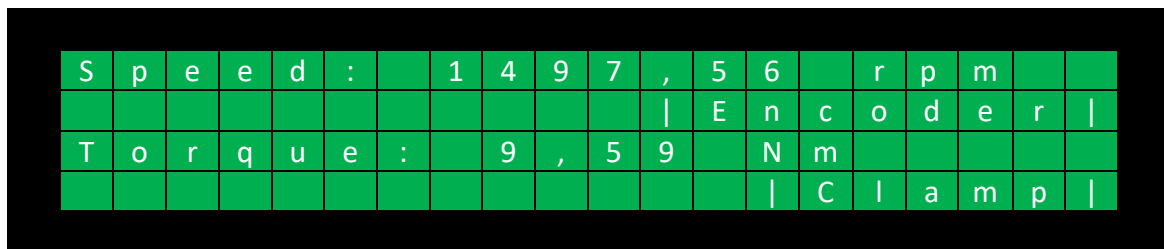


Figura 62. Ejemplo de resultado de LCD

Se puede observar, además, que a la izquierda de la pantalla se muestra el sensor utilizado en cada caso, cuyos valores pueden ser los mostrados en pantalla, u *Óptico* y *Hall*, respectivamente.

5.3.7. ALIMENTACIÓN DEL SISTEMA

En cuanto a la alimentación utilizada en el sistema, podemos observar que existen diferentes componentes, cada uno con una tensión de alimentación requerida diferente. Concretamente, existen tres niveles de tensión solicitados:

$$V_1 = 24 V$$

$$V_2 = 12 V$$

$$V_3 = 5 V$$

$$V_4 = 3.3 V$$

$$V_{red} = 230 V$$

Más detalladamente, los componentes que requieren cada nivel de tensión se han indicado en los circuitos respectivos, aunque se detallan a continuación como resumen y justificación del circuito de alimentación implementado.

- Tensión de alimentación de 24 V
 - o Sensor de efecto Hall

- Tensión de alimentación de 12 V
 - o Encoder
 - o Sensor óptico

Digitalización de un banco de ensayos y comprobación experimental

- Amplificador Operacional OPA705
- Tensión de alimentación de 5 V
 - ESP32 WROVER – B
 - Relés
 - Pantalla LCD

- Tensión de alimentación de 3.3 V
 - Selectores
 - Botón de reset

Así, la disposición de la alimentación es la siguiente:

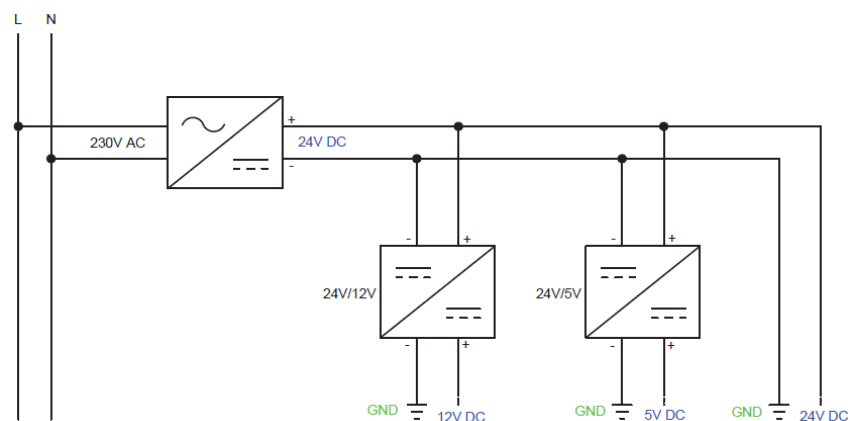


Figura 63. Circuito de alimentación

donde se puede observar que el sistema se encuentra conectado a la red mediante una serie de convertidores. Detalladamente, existe un convertidor AC/DC de 230V a 24V, de donde podemos obtener el primer nivel de tensión requerido para el sistema. Seguidamente, se puede observar la existencia de dos convertidores DC/DC, los cuales son de potencia máxima igual a 1 W, suficiente para esta aplicación, y uno convierte los 24V DC en 12V DC, mientras que el otro convierte los 24V DC en 5V DC, obteniendo los dos siguientes niveles de tensión requeridos. Por último, necesitamos obtener el último nivel de tensión, los 3,3V DC, los cuales se adquieren a partir de una de las patillas del microcontrolador ESP32 WROVER – B, sin necesidad de realizar ninguna conversión adicional.

5.3.8. APLICACIÓN ANDROID PARA VISUALIZACIÓN Y MODIFICACIÓN DE PARÁMETROS

En última instancia, se va a tratar de explicar el desarrollo de una aplicación Android cuya finalidad principal es tener la capacidad de mostrar en el dispositivo móvil los mismos valores de velocidad y par que los mostrados en pantalla y poder modificar ciertos parámetros anteriormente descritos. Así, no se requerirá ningún conocimiento acerca del lenguaje de programación utilizado para modificar valores.

De esta forma, los parámetros que son posibles de modificar se guardan en la memoria EEPROM que posee el propio microcontrolador ESP32 WROVER – B, de forma que queden permanentemente guardados incluso si se apaga o resetea, hasta que se deseen modificar de nuevo. Estos parámetros, concretamente, son:

$$\begin{aligned} & \text{Resolución}_{Encoder} \\ & \text{Resolución}_{Pinza} \\ & K_{ajuste1} = K_{Hall} \\ & K_{ajuste1} = K_{Pinza} \end{aligned}$$

Cuyos valores predeterminados son:

$$\begin{aligned} \text{Resolución}_{Encoder} &= 1024 \\ \text{Resolución}_{Óptico} &= 4 \\ K_{Hall} &= 1.0 \\ K_{Pinza} &= 1.0 \end{aligned}$$

De esta forma, en caso de cambiar de sensor de velocidad a uno de mayor o menor resolución, se podrá modificar directamente desde la aplicación. Además, en caso de cambiar de sensor de par y este proporcione un nivel de tensión centrado en otro punto, o con variación diferente, de podrá ajustar mediante las constantes de la misma forma que la resolución de la aplicación.

Además, es posible conocer desde la aplicación dichos valores que están guardados en la memoria EEPROM mediante un botón denominado *Obtener Parámetros*, como se mostrará más adelante.

Por otro lado, para realizar la conexión entre el microcontrolador ESP32 WROVER – B y la aplicación Android, se ha utilizado el BluetoothSerial, que envía y recibe información directamente a través del puerto Serie. Además, es mucho más simple que otros métodos de transmisión de información a través de Bluetooth, como pudiera ser el Bluetooth Low Energy.

De esta forma, Los intercambios de información realizados entre el microcontrolador y la aplicación Android quedan resumidos en el siguiente diagrama.

Digitalización de un banco de ensayos y comprobación experimental



Diagrama 4. Transmisión de información entre aplicación Android e información en ESP32 WROVER - B

De esta forma, la aplicación diseñada tiene el siguiente aspecto.



Figura 64. Interfaz de la aplicación Android

Digitalización de un banco de ensayos y comprobación experimental

Donde, si se toca donde se encuentra el logo Bluetooth siguiente:



Figura 65. Logo Bluetooth

se accede a otra pantalla donde se permite seleccionar el dispositivo Bluetooth con el que se desea conectar. Se muestra a continuación el diseño de dicha pantalla, donde se puede observar que el dispositivo denominado *PUESTO 6* se encuentra ya vinculado con el teléfono y disponible para conectar.

Más adelante, se puede observar en la Figura 60 los diferentes parámetros descritos y su correspondiente botón para obtener los que se encuentran guardados. Además, en la parte inferior se observa el valor real de velocidad y par, en tiempo real. Por otro lado, justo debajo del logo de Bluetooth se puede ver un *Botón de Slide* con el cual podemos entrar en modo edición y, así, editar los parámetros. Dicho botón se ha implementado debido a diferentes incompatibilidades existentes entre la transmisión de información por BluetoothSerial y las interrupciones y temporizadores, siendo estos dos últimos desactivados una vez activado el *Botón de Slide* y posteriormente activados al terminar de editar los parámetros.

Finalmente, podemos comentar que la aplicación se ha desarrollado haciendo uso de la aplicación AppInventor2, del Massachusetts Institute of Technology, puesto que se caracteriza por un ahorro de tiempo considerable, tanto en la implementación como en el aprendizaje del desarrollo de aplicaciones Android. Además, se ha tratado desde un inicio desarrollar dicha aplicación mediante el software Android Studio, aunque debido a los dos últimos motivos comentados, se ha decidido trasladarlo directamente a AppInventor2.

5.3.9. SISTEMA FÍSICO FINAL

A continuación, se muestra el sistema de medición finalmente implementado en una placa perforada de prototipado, con sus componentes y pistas soldadas.

Digitalización de un banco de ensayos y comprobación experimental

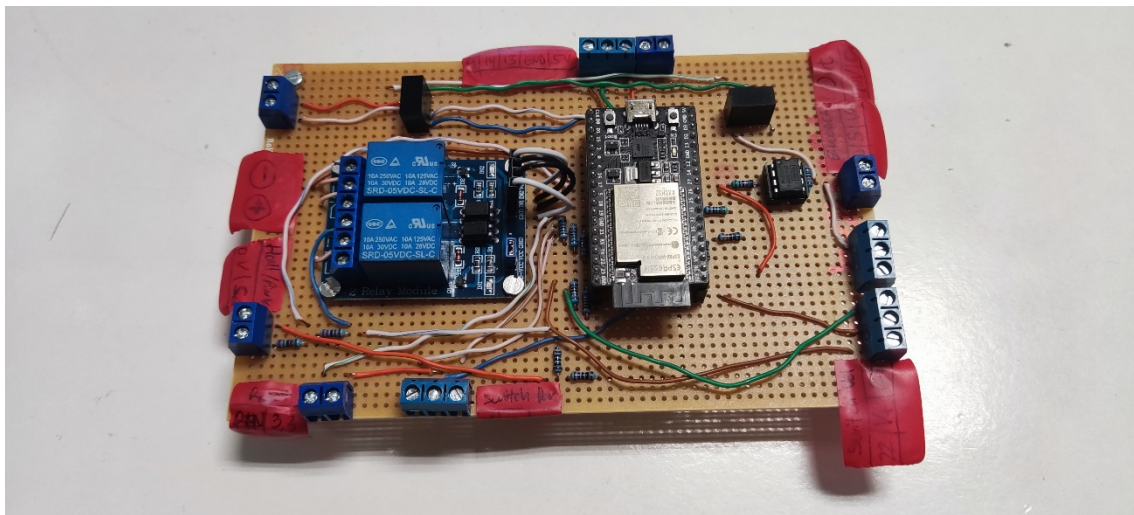


Figura 66. Sistema de medición soldado en placa perforada

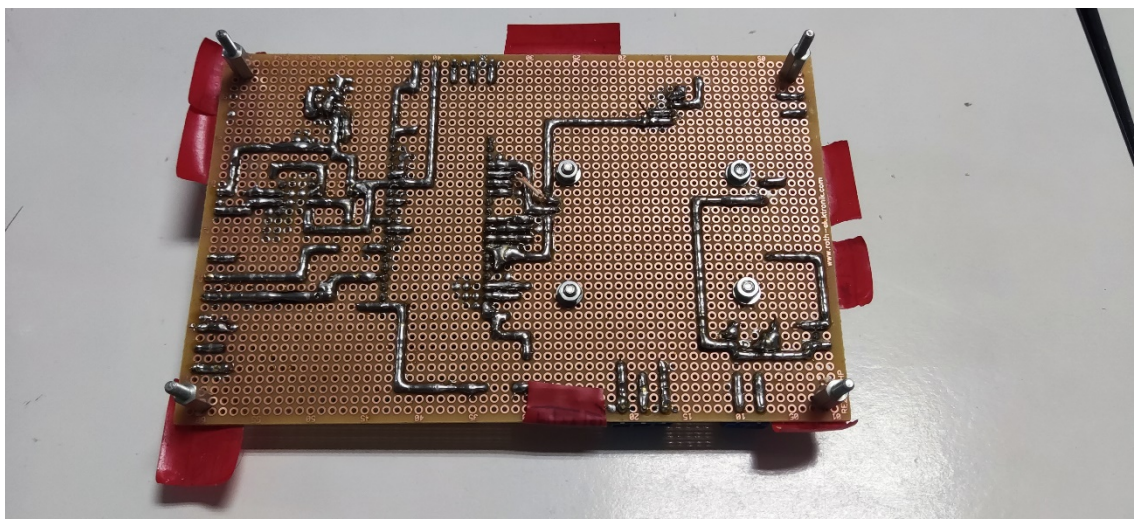


Figura 67. Reverso de sistema de medición soldado en placa perforada

5.4. COMPARATIVA ENTRE SIMULACIÓN Y SISTEMA REAL

En el presente punto se desarrollará una comparativa de la respuesta de la velocidad y de par de un sistema concreto entre la simulación realizada y el sistema de medición de parámetros en tiempo real implementado. Así, se asumirá que los resultados obtenidos mediante el sistema de medición diseñado son los correctos ya que, tras diversos ajustes y comparaciones con otros sistemas de medición, consigue un resultado con suficiente precisión para el requisito actual.

Por otro lado, puesto que de forma física únicamente se ha dispuesto de un motor asíncrono de inducción (Modelo: 1LE10011AB422AA4Z), se ha comprobado experimentalmente la respuesta de dicho motor, cuyas características principales mostradas en su placa de características son las siguientes.

$$\begin{aligned}P_n &= 2,20 \text{ kW} \\V_{LL} &= 400 \text{ V} \\f_n &= 50 \text{ Hz} \\P_{\text{polos}} &= 2 \text{ pares de polos}\end{aligned}$$

Para la implementación de este motor en el modelo de simulación, se ha partido de los resultados de los ensayos de vacío y cortocircuito y la medición de la resistencia del estator, además del coeficiente de fricción, B , y del momento de inercia, J , cuyos valores en este caso son:

- Resistencia del estator

$$R_{uv} = 6.6 \Omega$$

$$R_{uw} = 6.5 \Omega$$

$$R_{vw} = 6.7 \Omega$$

- Ensayo de vacío – conexión en estrella a tensión nominal

$$U_{10} = 400.33 \text{ V}$$

$$I_{10} = 3.43 \text{ A}$$

$$P_{10} = 312 \text{ W}$$

$$Q_{10} = 2354 \text{ VAr}$$



Digitalización de un banco de ensayos y comprobación experimental

- Ensayo de cortocircuito con conexión en estrella

$$U_{cc} = 88.03 V$$

$$I_{cc} = 4.75 A$$

$$P_{cc} = 420 W$$

$$Q_{cc} = 586 VAr$$

- Coeficientes

$$B = 0.002985 N \cdot m \cdot s$$

$$J = 0.0131 kg \cdot m^2$$

Estos valores se han introducido en la segunda interfaz gráfica desarrollada en el punto 6.2.4.1, donde se personaliza el motor de inducción a partir de los resultados de los ensayos o a partir del circuito equivalente.

Por la otra parte, no se ha comprobado experimentalmente el comportamiento del motor de corriente continua y, consecuentemente, tampoco el control de corriente continua implementado en la simulación (chopper DC) ni la alimentación correspondiente (120 V DC ó 240 V DC). Sin embargo, se mostrarán de igual forma en este punto algunos de los resultados obtenidos en la simulación de estos sistemas de corriente continua.

Así, sí se ha comprobado el funcionamiento de ambos controles de corriente alterna (Escalar y DTC), mientras que únicamente se ha utilizado la alimentación de 400 V AC de tensión y 50 Hz de frecuencia, que es la disponible en nuestra ubicación. Además, se ha comprobado el comportamiento para los cuatro tipos de carga implementados (cinta transportadora, bomba centrifugadora, bobinadora y cinta transportadora de varias cargas).

Por tanto, la dinámica para la comparación de los comportamientos de cada modelo será la siguiente:

Por un lado, se van a mostrar los diferentes comportamientos de par y velocidad, así como de algunos parámetros eléctricos, primeramente utilizando el control escalar y seguidamente utilizando el control directo de par, todo ello para diferentes niveles de carga y diferentes tipos de carga.

Por otro lado, se realizarán las mismas versiones que en el caso anterior, pero anotando el valor del sistema en estado estacionario. No se realiza, por tanto, una evolución en el tiempo similar a la obtenida en la simulación debido a diferentes limitaciones del sistema de medición.

Digitalización de un banco de ensayos y comprobación experimental

Finalmente, se compararán los valores obtenidos en la simulación en estado estacionario con los valores obtenidos en el sistema real, de forma exhaustiva y realizando el cálculo de la variación obtenida correspondiente.

5.4.1. RESULTADOS DE SIMULACIÓN DEL MODELO

Como se ha comentado anteriormente, los primeros resultados mostrados son los correspondientes al motor de inducción utilizando el control escalar.

Para todos los casos estudiados a continuación, la consigna de velocidad será fija y de valor igual a:

$$V_{ref} = 1200 \text{ rpm}$$

5.4.1.1. CONTROL ESCALAR

5.4.1.1.1. CINTA TRANSPORTADORA

Para el caso de la cinta transportadora, situaremos una carga, la cual es constante, igual a:

$$T_1 = 5.05 \text{ Nm}$$

De esta forma, se muestra a continuación su evolución de velocidad y par en el tiempo de simulación.

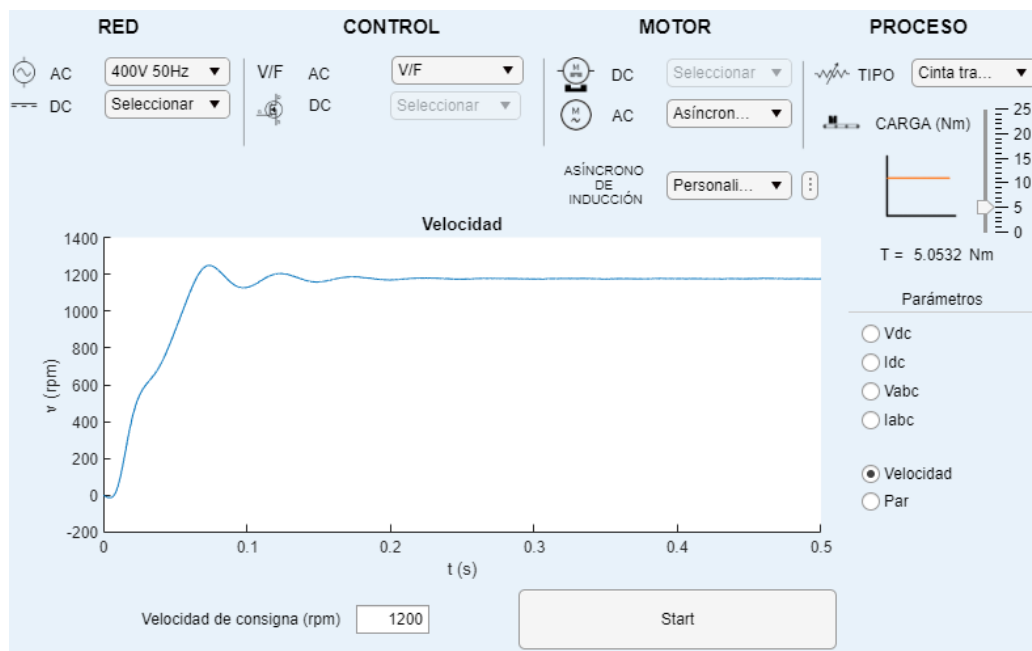


Figura 68. Respuesta de velocidad. Cinta transportadora – Simulación control escalar

Digitalización de un banco de ensayos y comprobación experimental

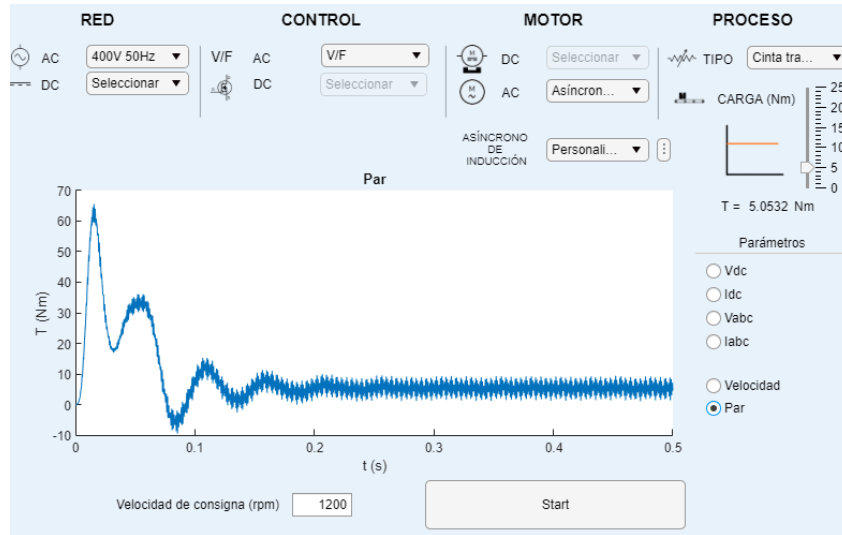


Figura 69. Respuesta de par. Cinta transportadora – Simulación control escalar

Así, se resumen en la tabla siguiente los valores de velocidad y par obtenidos en estado estacionario:

Resultado	Par resistente = 5,05 Nm
Velocidad (rpm)	1177,1
Par (Nm)	5,24

Tabla 2. Resumen de resultados. Cinta transportadora - Simulación control escalar

Por otro lado, podemos mostrar la evolución de la corriente suministrada al motor, la cual se consigue en este caso que sea senoidal gracias a la conmutación PWM en los transistores del inversor.

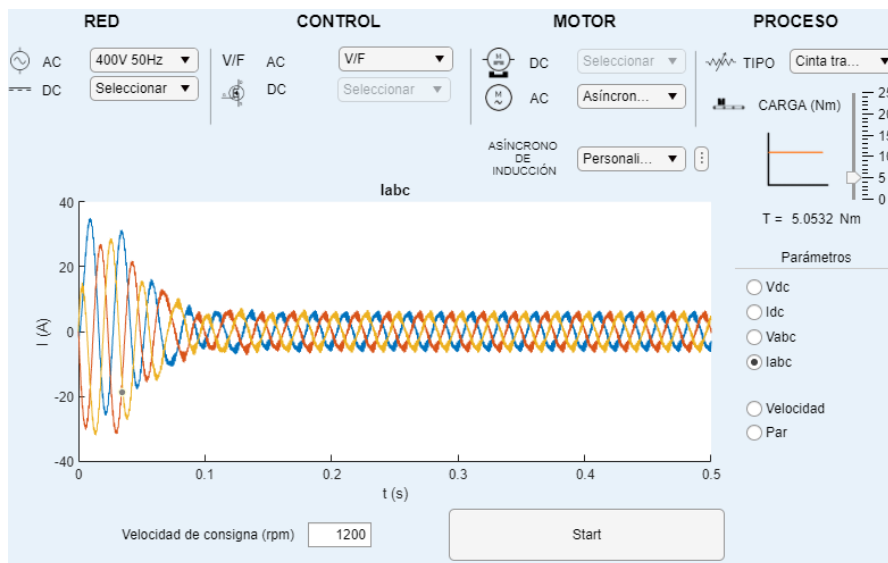


Figura 70. Respuesta de corriente alterna – Simulación control escalar

Digitalización de un banco de ensayos y comprobación experimental

Por otro lado, podemos observar el comportamiento de la tensión del bus de continua, para poder corroborar que la conversión de 400V AC a continua se ha realizado correctamente. Se obtiene así, la siguiente evolución:

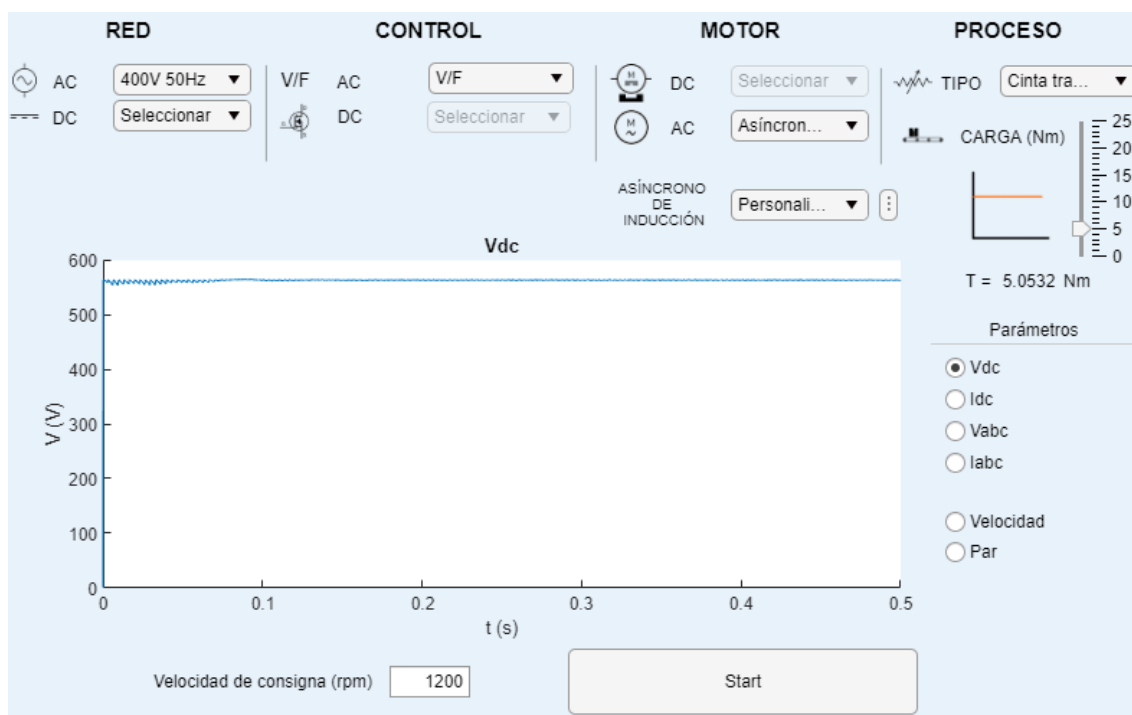


Figura 71. Respuesta de tensión continua. Cinta transportadora – Simulación control escalar

Se puede observar, por tanto, que el valor de tensión del bus de continua en estado estacionario es:

$$V_{dc} = 563.05 \text{ V}$$

Este valor es correcto, debido a que es considerablemente próximo al valor teórico que se debería obtener, que es:

$$V_{dc_{teo}} = 400V \cdot \sqrt{2} \rightarrow V_{dc_{teo}} = 565.68 \text{ V}$$

Finalmente, podemos comentar la existencia de diversas oscilaciones durante el transitorio de cada uno de los parámetros, debido principalmente al tipo de control utilizado, aunque alcanza el estado estacionario en un tiempo aceptablemente reducido.

Digitalización de un banco de ensayos y comprobación experimental

5.4.1.1.2. BOMBA CENTRIFUGADORA

En este segundo caso, el par de carga utilizado ha sido de valor igual a:

$$T_2 = 6.91 \text{ Nm}$$

Así, las respuestas de par y velocidad han sido:

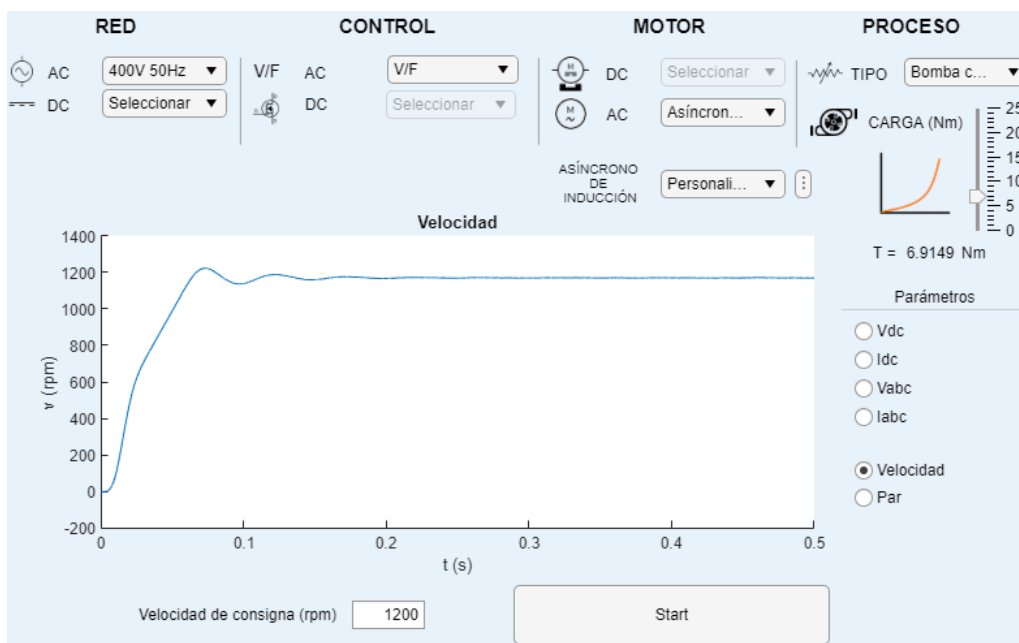


Figura 72. Respuesta de velocidad. Bomba centrífuga – Simulación

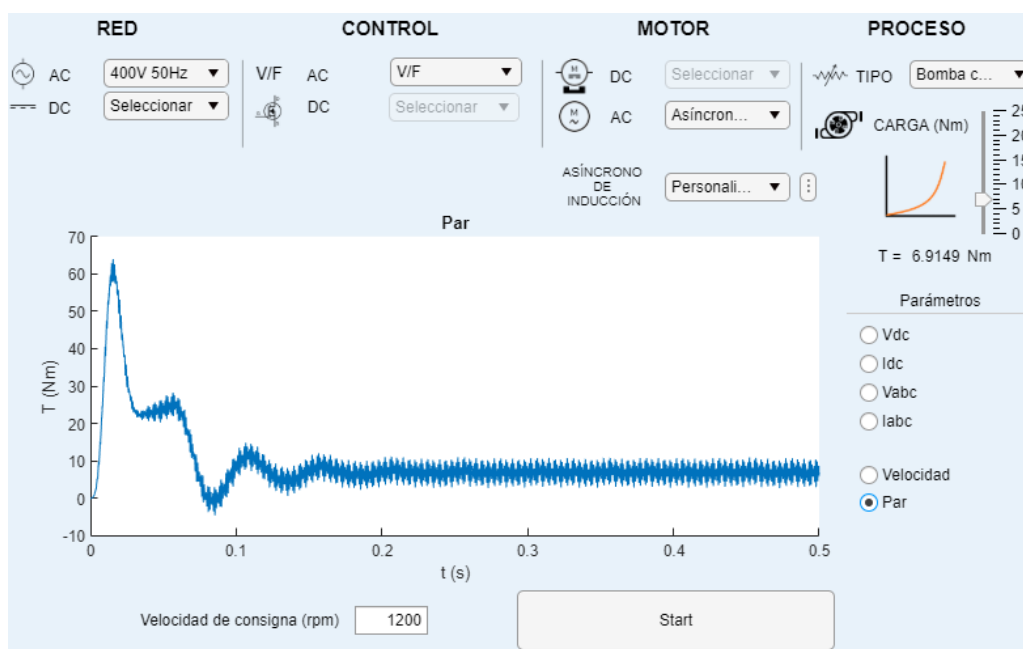


Figura 73. Respuesta de par. Bomba centrífuga – Simulación

Digitalización de un banco de ensayos y comprobación experimental

Cuyos valores concretos en estado estacionario se muestran en la tabla siguiente.

Resultado	Par resistente = 6,91 Nm
Velocidad (rpm)	1169,14
Par (Nm)	7,25

Tabla 3. Resumen de resultados. Bomba centrífuga – Simulación control escalar

De esta forma, observamos una menor oscilación en el transitorio de la velocidad, e incluso en el par, debido a que la carga incrementa cuadráticamente y a que la carga utilizada en este caso es mayor, impidiendo así que los picos de velocidad alcancen valores elevados como en el caso anterior.

5.4.1.2. CONTROL DIRECTO DE PAR

5.4.1.2.1. BOBINADORA

En el caso de que se implemente una carga como pudiera ser una bobinadora, la respuesta de velocidad se muestra a continuación, utilizando el siguiente nivel de carga:

$$T_3 = 2.9255 \text{ Nm}$$

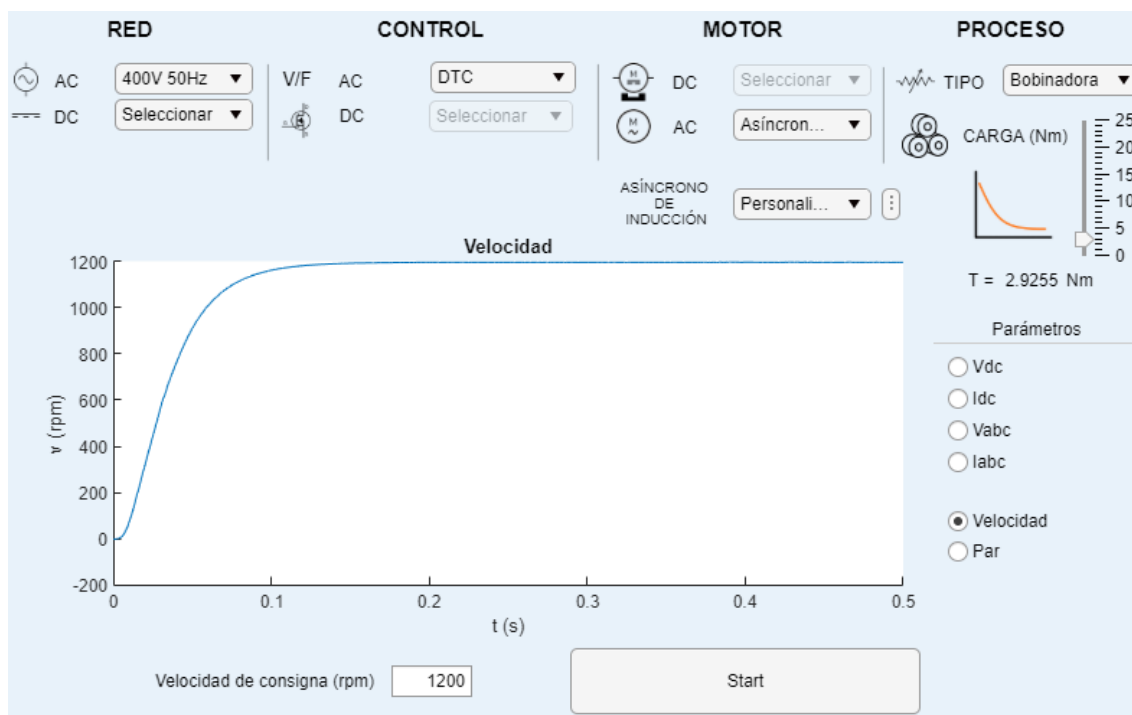


Figura 74. Respuesta de velocidad. Bobinadora - Simulación control directo de par

Digitalización de un banco de ensayos y comprobación experimental

En el caso del par, observamos el siguiente comportamiento:

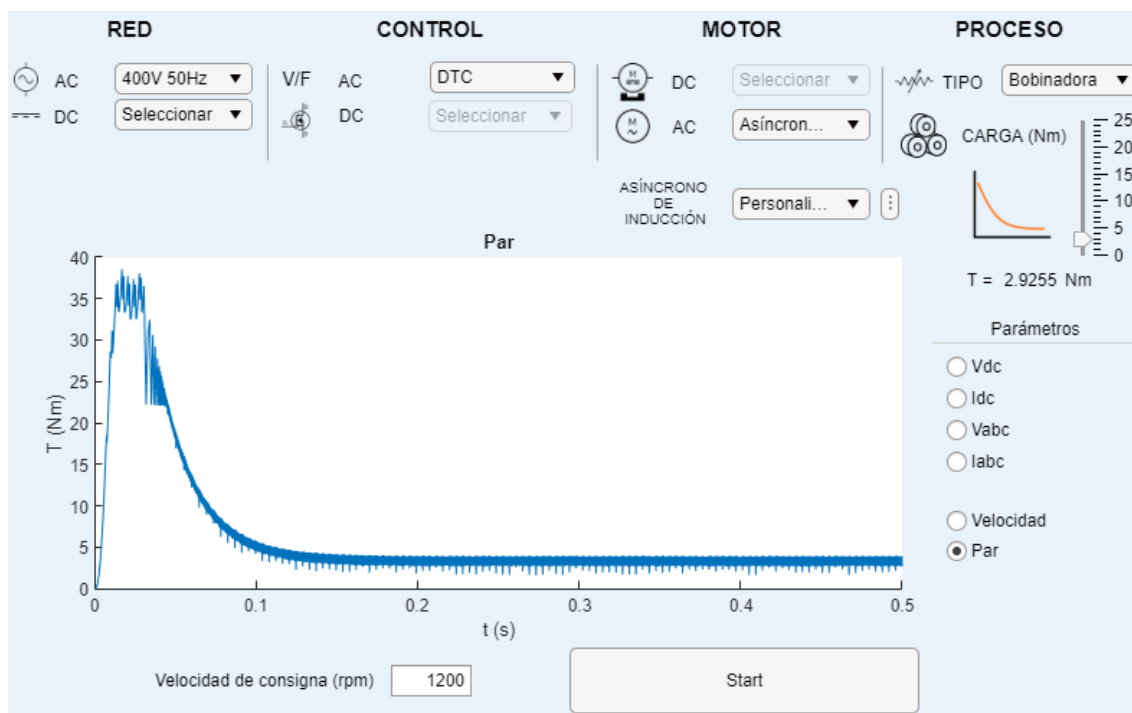


Figura 75. Respuesta de par. Bobinadora - Simulación control directo de par

Se puede observar a simple vista cómo la evolución de par y velocidad es mucho más fluida que en el caso de haber utilizado el control escalar e incluso a pesar de resultar una carga más compleja que las dos anteriores estudiadas. Así, el valor de velocidad en estado estacionario será mucho más próximo al de consigna, mientras que el de par será más próximo al de la carga, tal y como se muestra en la tabla siguiente.

Resultado	Par resistente = 2,92 Nm
Velocidad (rpm)	1195,17
Par (Nm)	2,93

Tabla 4. Resumen de resultados. Bobinadora – Simulación control directo de par

Además, se muestran a continuación los resultados de tensión y corriente del bus de continua y de tensión y corriente de salida del convertidor.

Digitalización de un banco de ensayos y comprobación experimental

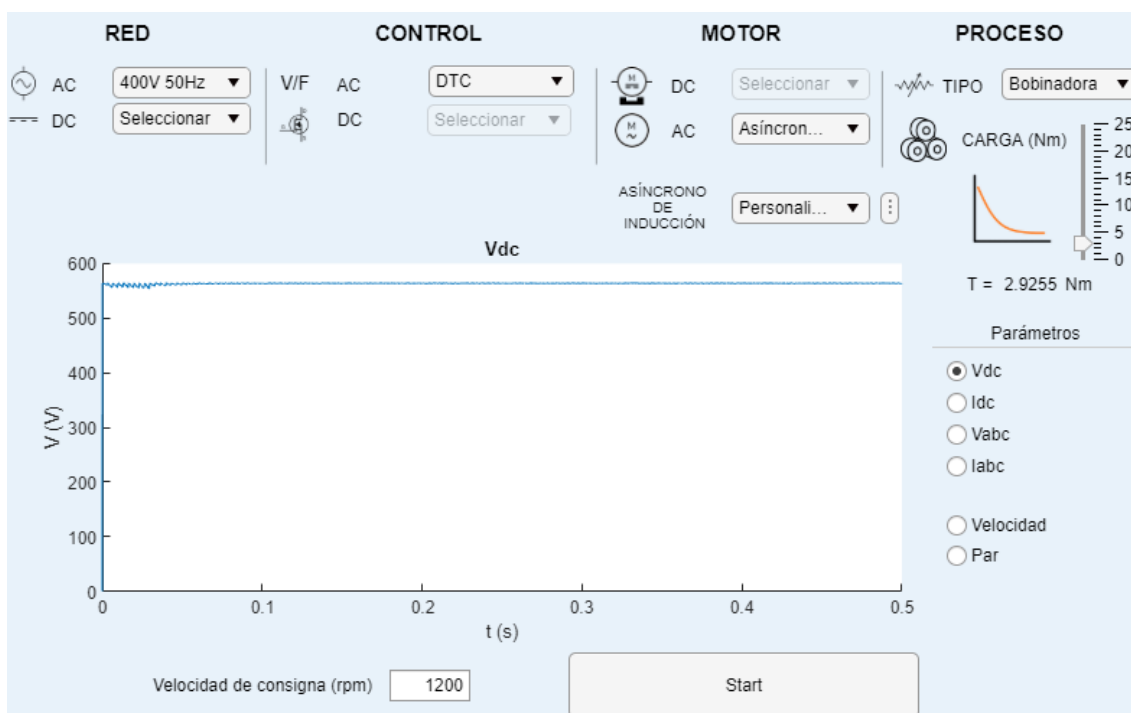


Figura 76. Respuesta de tensión continua. Bobinadora - Simulación control directo de par

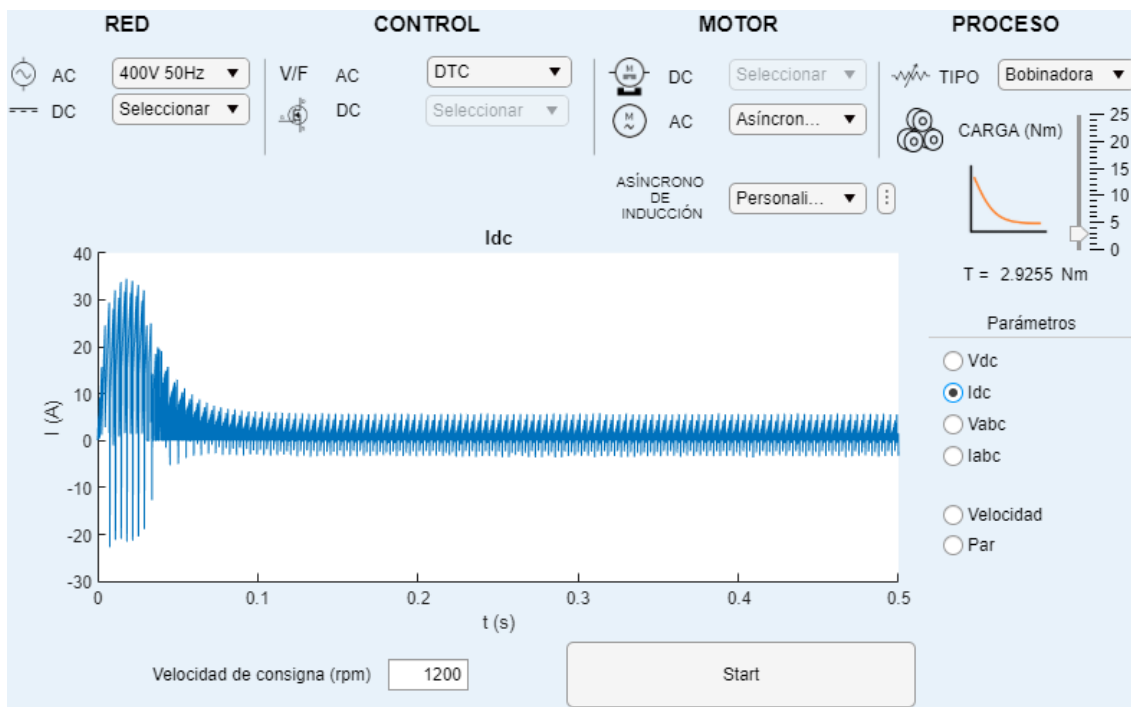


Figura 77. Respuesta de corriente continua. Bobinadora - Simulación control directo de par

Digitalización de un banco de ensayos y comprobación experimental

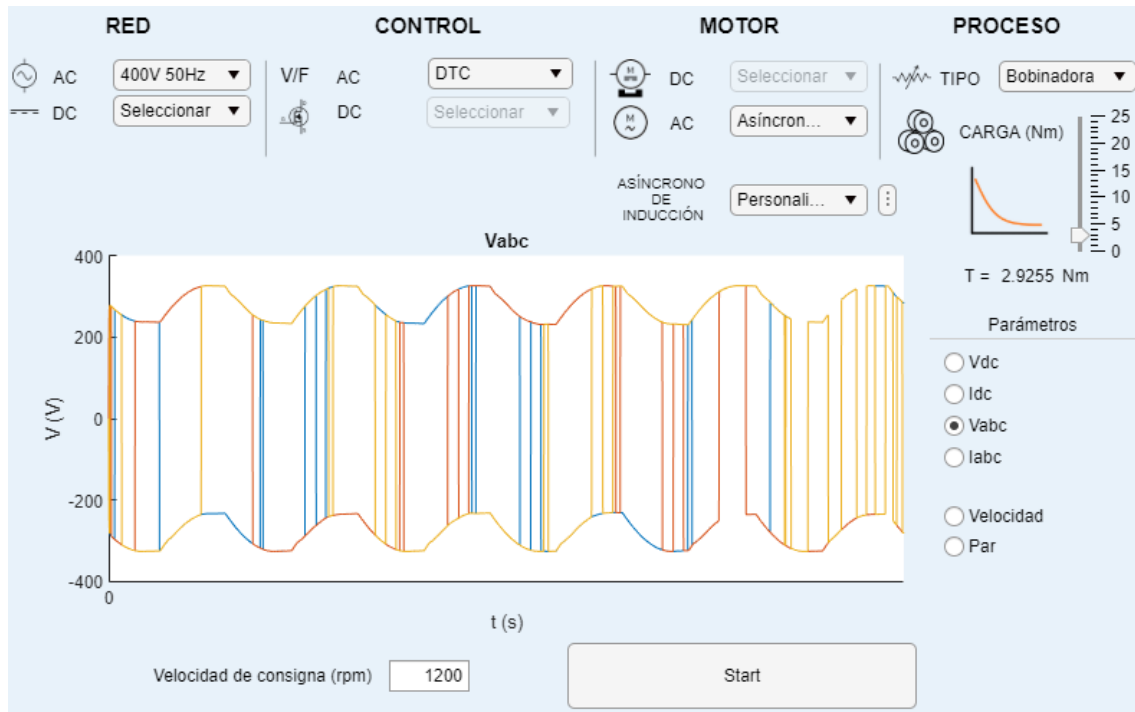


Figura 78. Respuesta de tensión alterna. Bobinadora - Simulación control directo de par

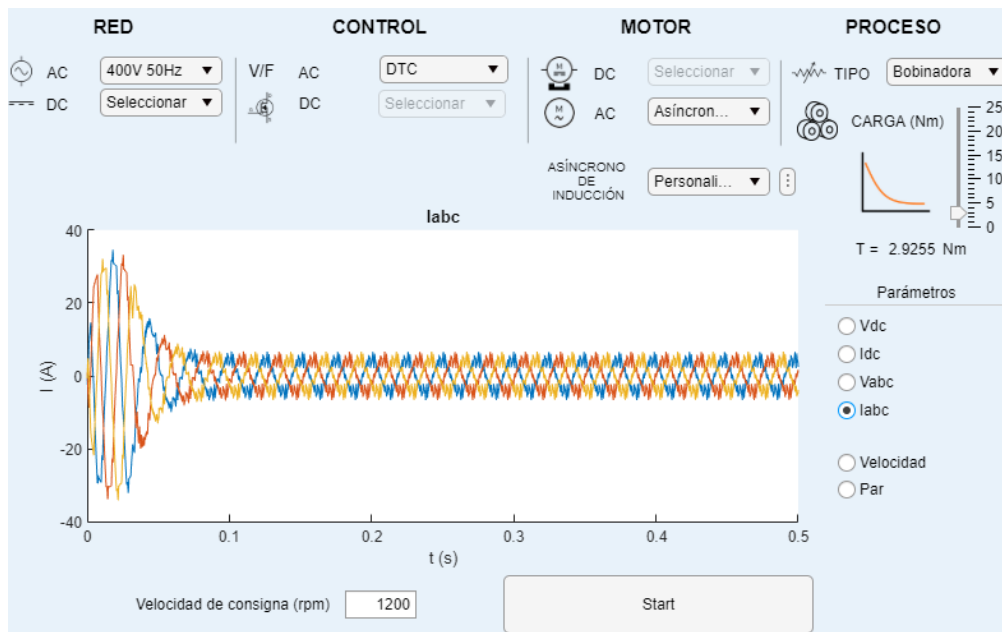


Figura 79. Respuesta de corriente alterna. Bobinadora - Simulación control directo de par

Digitalización de un banco de ensayos y comprobación experimental

5.4.1.2.2. CINTA TRANSPORTADORA DE VARIAS CARGAS

En último lugar, se muestran las respuestas de velocidad y par para el caso de la cinta transportadora de varias cargas. En este caso se ha utilizado un par de carga superior al de los anteriores para poder observar con claridad el cambio de éste y su repercusión en el sistema cuando se utiliza el control directo de par.

$$T_4 = 11.17 \text{ Nm}$$

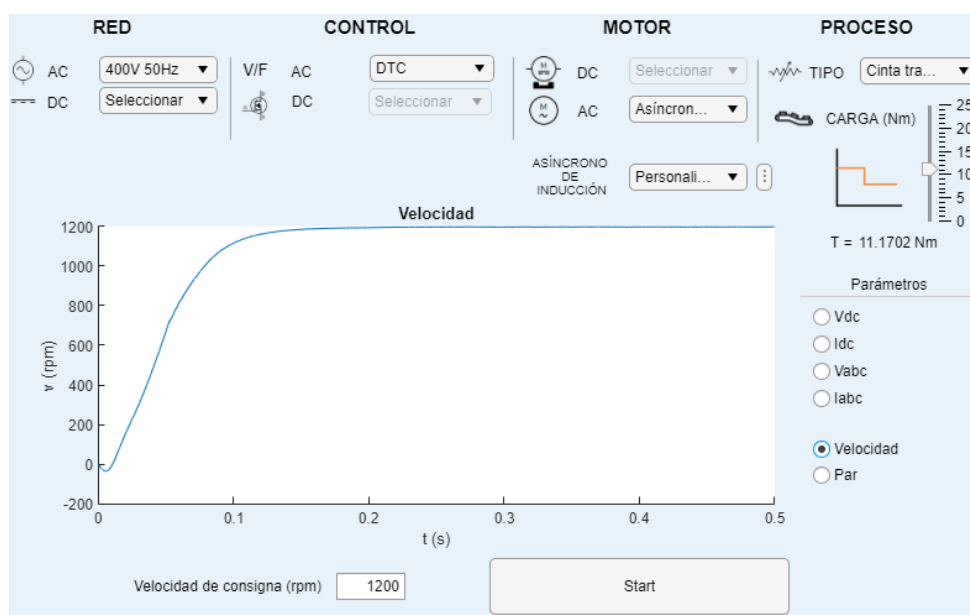


Figura 80. Respuesta de velocidad. Cinta transportadora de varias cargas - Simulación control directo de par

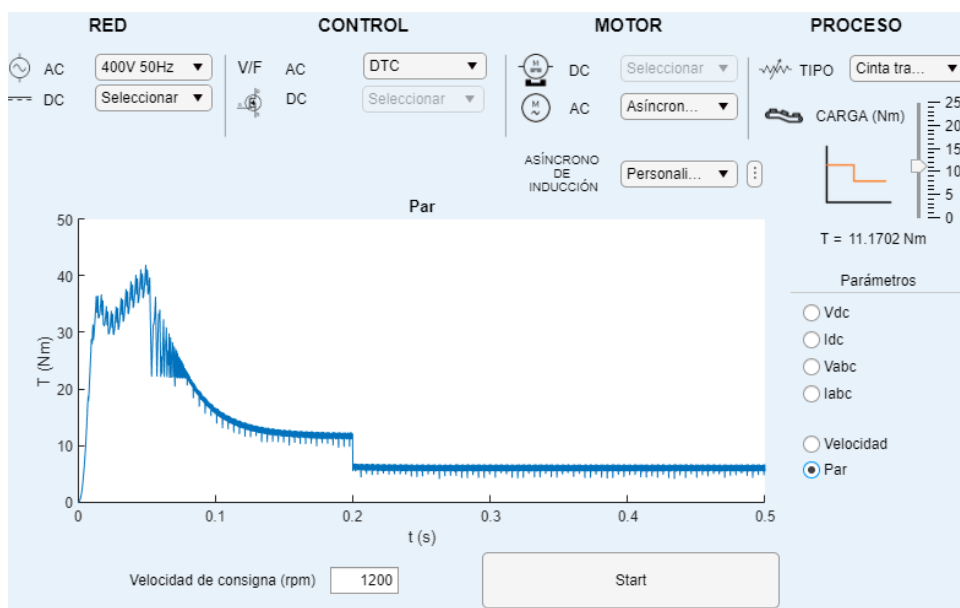


Figura 81. Respuesta de par. Cinta transportadora de varias cargas - Simulación control directo de par

Digitalización de un banco de ensayos y comprobación experimental

En este caso se puede observar, por un lado, que en el momento que cambia la carga la variación de velocidad es prácticamente despreciable, se mantiene en el mismo valor estacionario. Por otro lado, vemos que el par durante el estado estacionario realiza el mismo comportamiento que en el caso anterior de la bobinadora: se mantiene prácticamente al unísono con el par de carga, lo cual es característica principal de este tipo de control.

De esta forma, se resume en la siguiente tabla el valor de cada parámetro en estado estacionario, para cada carga.

Resultado	Par resistente = 11,17 Nm - t = 0 s	Par resistente = 5,59 Nm - t = 0,2 s
Velocidad (rpm)	1195,40	1195,40
Par (Nm)	11,23	5,62

Tabla 5. Resumen de resultados. Cinta transportadora de varias cargas - Simulación control directo de par

Análogamente se puede observar en la gráfica de la tensión suministrada por el convertidor el cambio de carga en el instante $t = 0.2$ s. Por tanto, como complemento, se incluye a continuación dicha representación gráfica con el fin de corroborar el comportamiento del par recién obtenido.

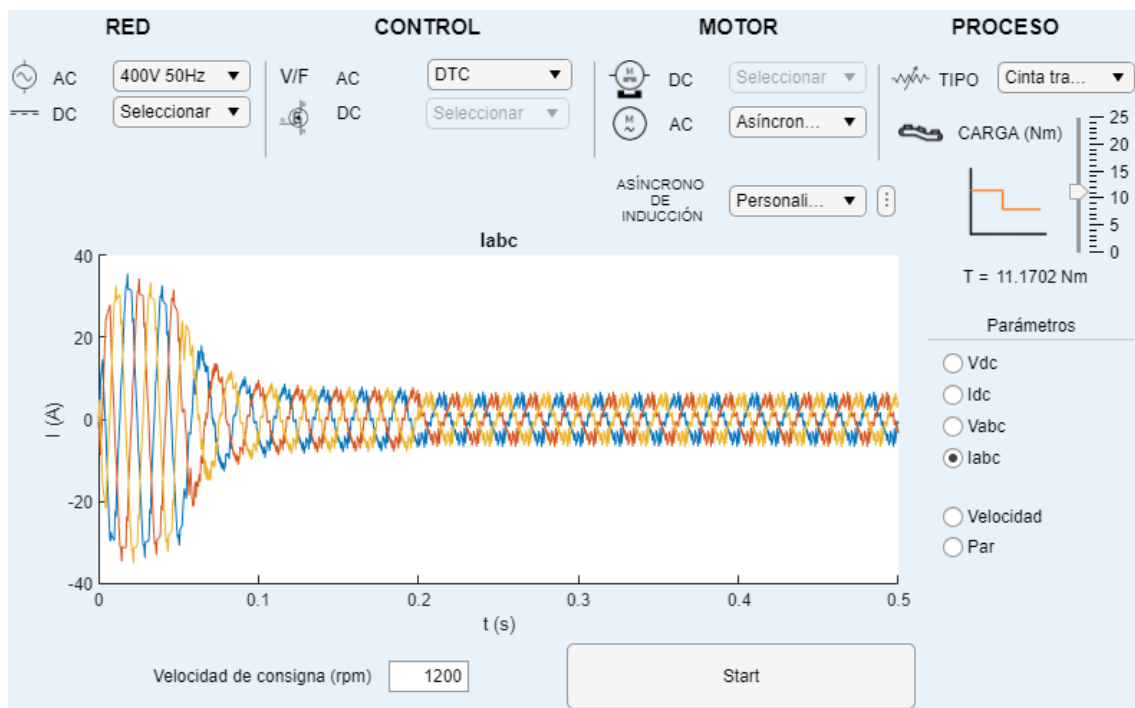


Figura 82. Respuesta de corriente alterna. Cinta transportadora de varias cargas - Simulación control directo de par

5.4.2. RESULTADOS DE MEDICIÓN

Se detallan a continuación los resultados obtenidos vía Bluetooth en la app diseñada. Se mostrará, por un lado, una captura de pantalla de la aplicación Android y, por otro lado, una tabla resumen como en el caso de los resultados del modelo de simulación.

5.4.2.1. CONTROL ESCALAR

5.4.2.1.1. CINTA TRANSPORTADORA

En este primer caso, se ha tratado de que el par de carga fuese, aproximadamente, del mismo valor que el anterior:

$$T_1 = 5,05 Nm$$

Así, se obtienen los siguientes valores de velocidad y par en estado estacionario, para el caso de la cinta transportadora.

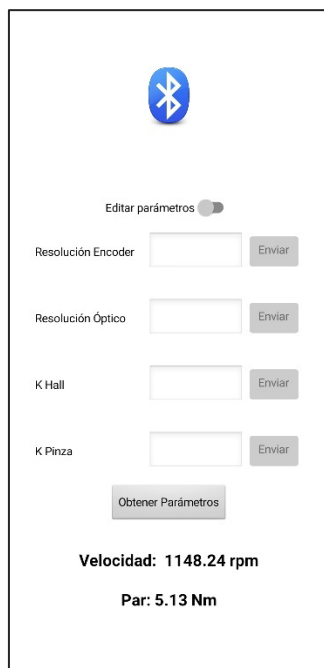


Figura 83. Respuesta de velocidad y par. Cinta transportadora - Medición escalar

Digitalización de un banco de ensayos y comprobación experimental

Resultado	Par resistente = 5,05 Nm
Velocidad (rpm)	1148,24
Par (Nm)	5,13

Tabla 6. Resumen de resultados. Cinta transportadora - Medición escalar

5.4.2.1.2. BOMBA CENTRIFUGADORA

Para este caso, el par máximo se ha tratado de que tuviese un valor igual a:

$$T_2 = 6,91 \text{ Nm}$$

De esta forma, las respuestas de velocidad y par en estado estacionario han resultado con los siguientes valores.

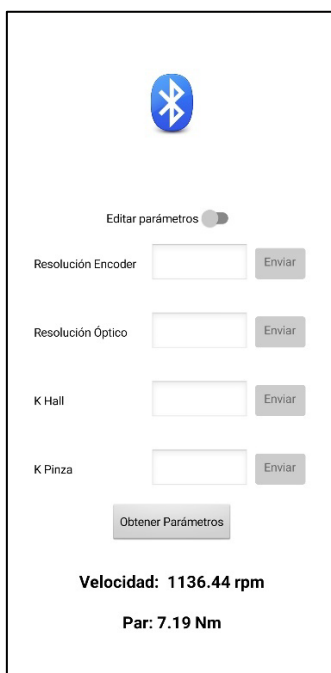


Figura 84. Respuesta de velocidad y par. Bomba centrifugadora - Medición escalar

Resultado	Par resistente = 6,91 Nm
Velocidad (rpm)	1136,44
Par (Nm)	7,19

Tabla 7. Resumen de resultados. Bomba centrifugadora - Medición escalar

5.4.2.2. CONTROL DIRECTO DE PAR

5.4.2.2.1. BOBINADORA

En este tercer caso, el par de carga máximo se ha tratado de que tuviese un valor igual a:

$$T_3 = 2,92 \text{ Nm}$$

Así, el valor de la velocidad y el par en estado estacionario resulta de la forma siguiente.

Bluetooth icon

Editar parámetros

Resolución Encoder Enviar

Resolución Óptico Enviar

K Hall Enviar

K Pinza Enviar

Obtener Parámetros

Velocidad: 1200.16 rpm

Par: 2.93 Nm

Figura 85. Respuesta de velocidad y par. Bobinadora - Medición control directo de par

Resultado	Par resistente = 2,92 Nm
Velocidad (rpm)	1200,16
Par (Nm)	2,93

Tabla 8. Resumen de resultados. Bobinadora - Medición control directo de par

Digitalización de un banco de ensayos y comprobación experimental

5.4.2.2.2. CINTA TRANSPORTADORA DE VARIAS CARGAS

En cuarto y último lugar, se ha dividido la medición en dos partes, una con el par inicial determinado y la otra con la mitad de dicho par, resultando en un comportamiento igual al de la cinta transformadora (primer caso) pero con control directo de par.

Así, el par de carga introducido se ha tratado de que sea de valor igual a:

$$T_4 = 11,17 Nm$$

Mientras que para la otra parte se ha tratado de que su valor sea igual a:

$$T_{4_2} = 5,59 Nm$$

Así, el resultado de par y velocidad, en cada caso ha sido el siguiente.

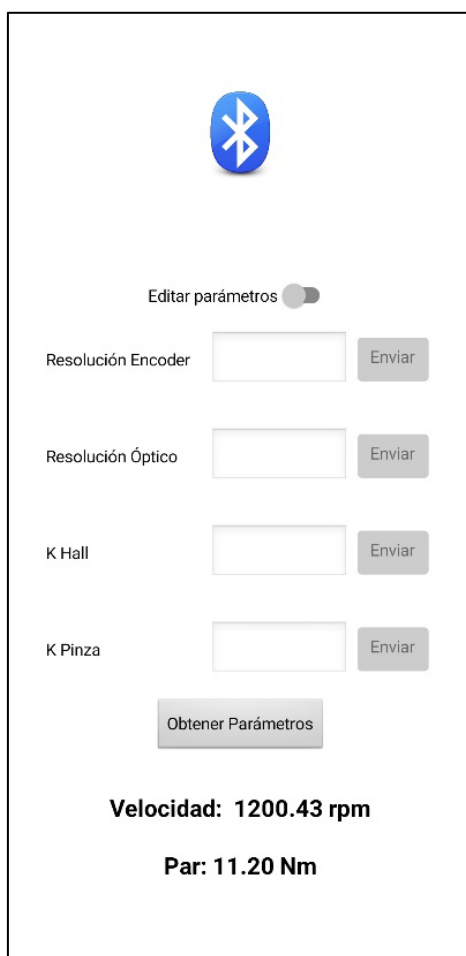


Figura 86. Respuesta de velocidad y par. Cinta transportadora de varias cargas T4 - Medición control directo de par

Digitalización de un banco de ensayos y comprobación experimental

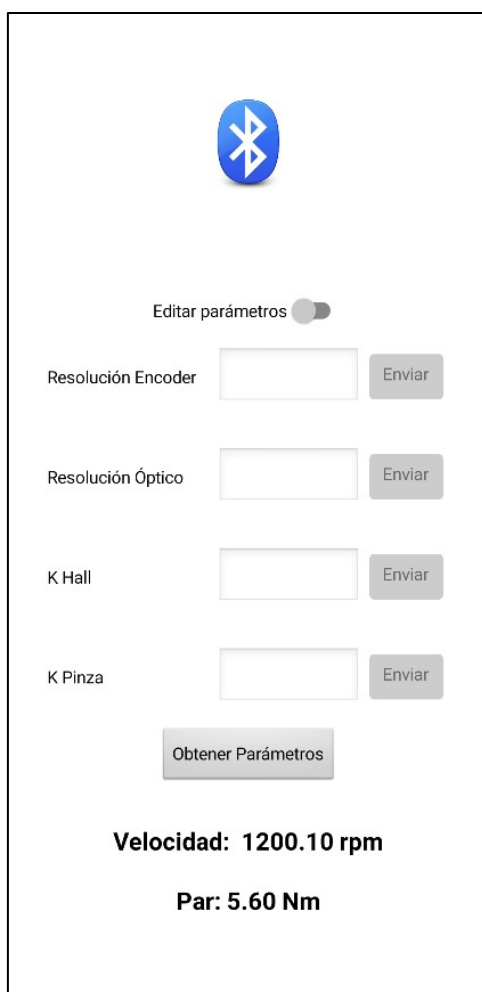


Figura 87. Respuesta de velocidad y par. Cinta transportadora de varias cargas T4 2 - Medición control directo de par

Resultado	Par resistente = 11,17 Nm - t = 0 s	Par resistente = 5,59 Nm - t = 0,2 s
Velocidad (rpm)	1200,43	1200,10
Par (Nm)	11,20	5,60

Tabla 9. Resumen de resultados. Cinta transportadora de varias cargas - Medición control directo de par

5.4.3. COMPARATIVA ENTRE RESULTADOS DE SIMULACIÓN Y MEDICIÓN

En este último punto donde se detalla la solución adoptada se tratará de comprobar la variación entre las mediciones realizadas y los resultados del modelo de simulación implementado. Además, se tratará de argumentar las variaciones existentes entre los resultados, haciendo uso, o de la base teórica anteriormente comentada, o de aspectos característicos de la aparamenta comercial utilizada para simular cada uno de los casos.

Así, para el cálculo de dicha variación, se ha utilizado, en todos los casos, la siguiente expresión:

$$\Delta (\%) = \frac{p_1 - p_2}{p_1} \cdot 100$$

Donde:

$\Delta (\%) =$ *variación en forma porcentual*

$p_1 =$ *valor del primer resultado*

$p_2 =$ *valor del segundo resultado*

De esta forma, se consiguen representar las variaciones de los resultados superiores a los de la medición con signo positivo, mientras que las variaciones de los resultados inferiores a los de la medición con signo negativo.

Así, se analiza a continuación, caso por caso, para cada carga, la desviación de los resultados.

5.4.3.1. CINTA TRANSPORTADORA

En este primer caso, teniendo como carga la cinta transportadora y de control el escaler, los resultados obtenidos han sido los siguientes.

Par resistente = 5,05 Nm			
Resultado	Simulación	Real	Variación (%)
Velocidad (rpm)	1177,1	1148,24	2,51%
Par (Nm)	5,24	5,13	2,14%

Tabla 10. Resumen de variación. Cinta transportadora – Escalar

Se puede observar, a partir de los resultados obtenidos en la Tabla 10, que existe una variación considerable de entorno al 2% entre lo simulado y lo medido, tanto para la velocidad como para el par. Esto es muy probablemente debido a que al realizar la simulación no se hayan tenido en cuenta o se hayan considerado nulos algunos factores que puedan generar resistencia mecánica al sistema. También, puede ser debido a que los parámetros obtenidos de los ensayos con el cual se ha implementado

Digitalización de un banco de ensayos y comprobación experimental

el motor en el modelo de Simulink no hayan sido del todo exactos comparados con los valores reales de la máquina.

Sin embargo, a pesar de todo ello, se ha conseguido una respuesta muy buena tanto en la simulación como en el sistema real, donde además podemos considerar muy reducida la variación existente del sistema.

Por otro lado, se puede ver que el valor obtenido de velocidad en ambos casos es muy diferente al valor de consigna, es inferior. Al tratarse de una máquina de inducción asíncrona, existe un deslizamiento, es decir, una pérdida de revoluciones, originado por la carga impuesta, no siendo corregido por el control escalar.

5.4.3.2. BOMBA CENTRIFUGADORA

En este segundo caso, utilizando como carga una bomba centrifugadora y un control escalar, los resultados son los siguientes.

Par resistente = 6,91 Nm			
Resultado	Simulación	Real	Variación (%)
Velocidad (rpm)	1169,14	1136,44	2,88%
Par (Nm)	7,25	7,19	0,83%

Tabla 11. Resumen de variación. Bomba centrifugadora – Escalar

Se puede observar, por tanto, la existencia de un deslizamiento en ambos casos y una variación entre el sistema real y la simulación de aproximadamente un 2 %, muy similar al del caso anterior. Por otro lado, se puede ver una variación inferior al 1 % en el par, aunque en ambos casos se aleja considerablemente de la consigna de carga, lo cual se debe muy probablemente a la ligera inestabilidad de ambos sistemas.

5.4.3.3. BOBINADORA

En el tercer caso, utilizando como carga la bobinadora y un control directo de par, los resultados son los siguientes.

Par resistente = 2,92 Nm			
Resultado	Simulación	Real	Variación (%)
Velocidad (rpm)	1195,17	1200,16	-0,42%
Par (Nm)	2,93	2,93	0,00%

Tabla 12. Resumen de variación. Bobinadora – Control directo de par

Se puede observar una menor variación entre los resultados de velocidad de la simulación y los de la medición, inferior al 0.50 %. Se debe principalmente a una mejor implementación del control directo de par en el modelo de *Simulink* que en el caso anterior donde se utilizaba el control escalar. Por otro lado, se puede observar que el resultado de velocidad de la simulación difiere ligeramente de la velocidad de

Digitalización de un banco de ensayos y comprobación experimental

consigna, de nuevo probablemente debido a diferentes consideraciones ligeramente erróneas implementadas en el modelo. En el caso del sistema real, la velocidad se ajusta prácticamente exacta a la velocidad de consigna, ya que se cuenta con controles comerciales de un alto coste, con la precisión que ello conlleva.

En cuanto al par, se puede observar que la variación es nula, aunque probablemente debido a la casualidad de los parámetros elegidos. Aun así, ambos valores se asemejan de forma suficiente a los valores de par de carga introducidos.

5.4.3.4. CINTA TRANSPORTADORA DE VARIAS CARGAS

En este último caso, utilizando la cinta transportadora de varias cargas y el control directo de par, se obtienen los siguientes resultados, en primer lugar para la primera carga y en segundo lugar para la carga reducida cuando $t = 0,20$ s.

Par resistente = 11,17 Nm - t = 0 s			
Resultado	Simulación	Real	Variación (%)
Velocidad (rpm)	1195,4	1200,43	-0,42%
Par (Nm)	11,23	11,2	0,27%

Tabla 13. Resultados de variación. Cinta transportadora de varias cargas T4 - Control directo de par

Par resistente = 5,59 Nm - t = 0,2 s			
Resultado	Simulación	Real	Variación (%)
Velocidad (rpm)	1195,40	1200,10	-0,39%
Par (Nm)	5,62	5,60	0,36%

Tabla 14. Resultados de variación. Cinta transportadora de varias cargas T4 2 - Control directo de par

Se puede ver, para ambas cargas, una variación mínima, menor al 0,50% en todos los casos. Además, las velocidades y pares, tanto reales como simulados, se acercan de forma muy próxima a los valores de consigna, de la misma forma que hemos observado en el apartado anterior. Además, se puede comentar que las variaciones son menores que las del apartado anterior, en el caso de la velocidad, debido a una más sencilla implementación de la carga en el modelo de simulación.



6. CONCLUSIONES

El presente Trabajo de Fin de Grado ha tratado diversos campos que abarca la ingeniería, desde el punto de vista laboral y de investigación.

Así, se ha introducido el ámbito de la simulación que, como se comentaba al comienzo de esta memoria, se trata de uno de los procesos primordiales al comienzo de la realización de una actividad industrial.

Más adelante, se han tratado las interfaces gráficas, las cuales se encuentran presentes en prácticamente cualquier ámbito, no sólo de la ingeniería, sino también en otras especializaciones o incluso en la vida cotidiana.

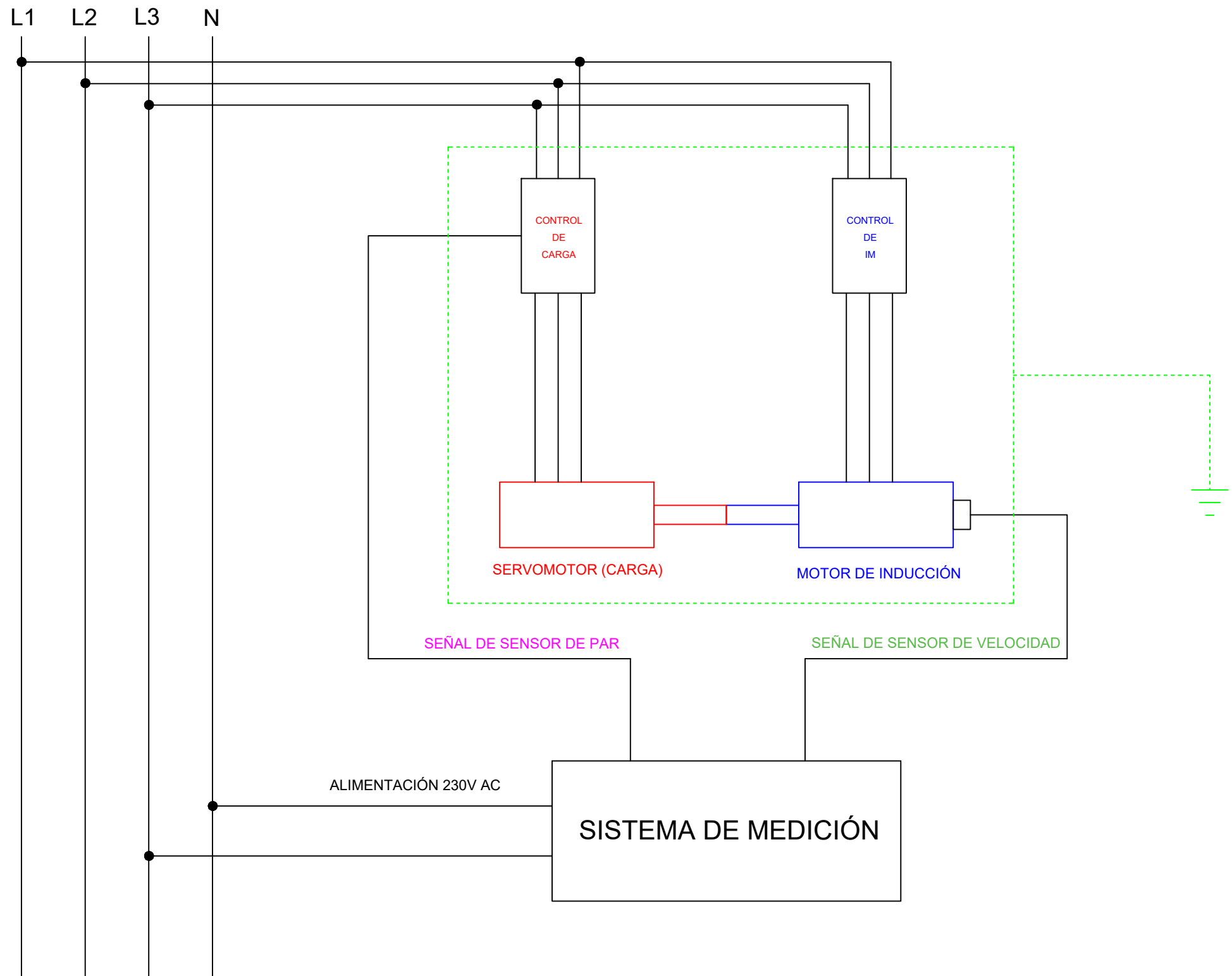
Seguidamente, mediante el diseño del sistema de medición y su respectiva aplicación Android se han comprendido diferentes conceptos que se encuentran presentes en todo el proceso de diseño y desarrollo de nuevos componentes o aparatos electrónicos.

Finalmente, se ha realizado una valoración de los resultados obtenidos, tanto en la simulación a partir de las gráficas, conectadas al modelo de *Simulink* y visualizadas en la interfaz gráfica, como en el sistema real a partir de la información mostrada en la pantalla LCD o en la misma aplicación Android. De esta forma, se ha observado que la variación entre los resultados obtenidos es de entorno al 2% en el peor de los casos, mientras que en el mejor de los casos se encuentra en menos del 0.5%, eso sí, haciendo uso de un control directo de par, el cual es considerablemente más costoso que el control escalar también implementado.

En conclusión, el desarrollo del conjunto del proyecto ha facilitado la comprensión del funcionamiento de la metodología utilizada en la industria para el diseño de nuevos componentes. Esto, ha conducido a un exitoso resultado en cuanto a precisión de los sistemas, aunque queda abierto a la implementación de nuevos tipos de máquinas eléctricas, convertidores e incluso procesos, además de a su transformación para la visualización de los resultados de la simulación en tiempo real, idea principal del presente proyecto. Además, en cuanto a la parte del sistema de medición implementado, queda de igual manera abierto a su transformación con el fin de obtener los resultados en el mismo formato que la simulación, aunque tanto esta modificación como la comentada anteriormente acerca de la visualización de los resultados de simulación en tiempo real requiere una mayor inversión de tiempo y muy probablemente de recursos.



PLANOS



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



LEORDEAN, COSMIN ADRIAN
GRADO EN INGENIERÍA ELÉCTRICA

TRABAJO DE FIN DE GRADO - MEDICIÓN DE PAR Y VELOCIDAD

SITUACIÓN: VALENCIA

PETICIONARIO: UNIVERSIDAD POLITÉCNICA DE VALENCIA

PLANO
ESQUEMA DE CONEXIONADO

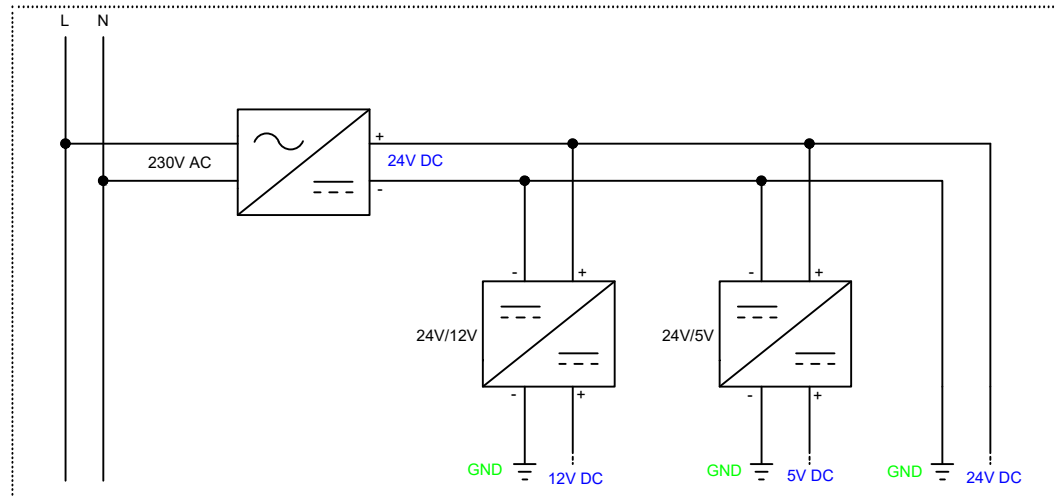
FECHA
06/2021

Nº
01

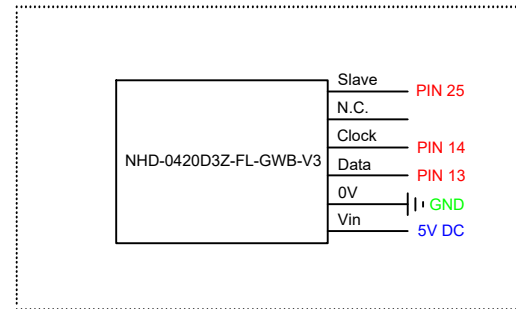
ESCALA
S/E

REFERENCIA
M-01

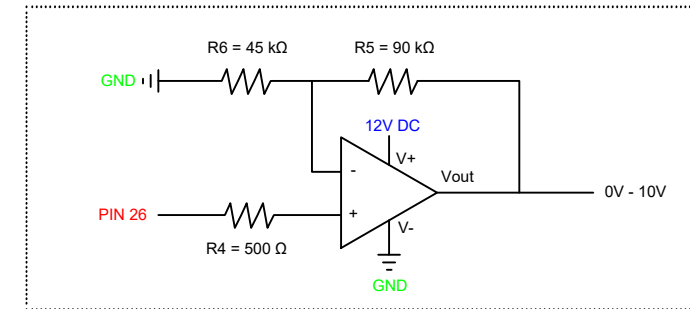
ALIMENTACIÓN



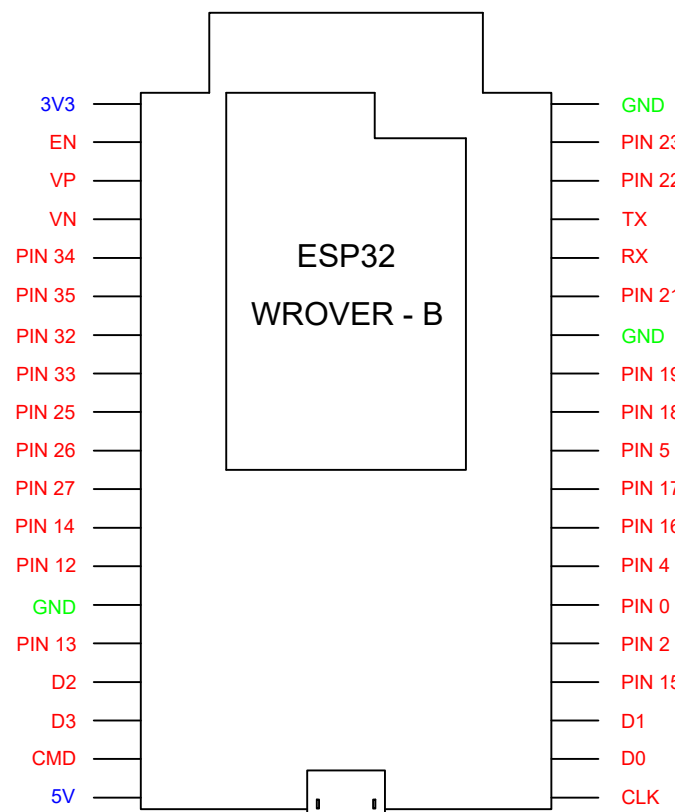
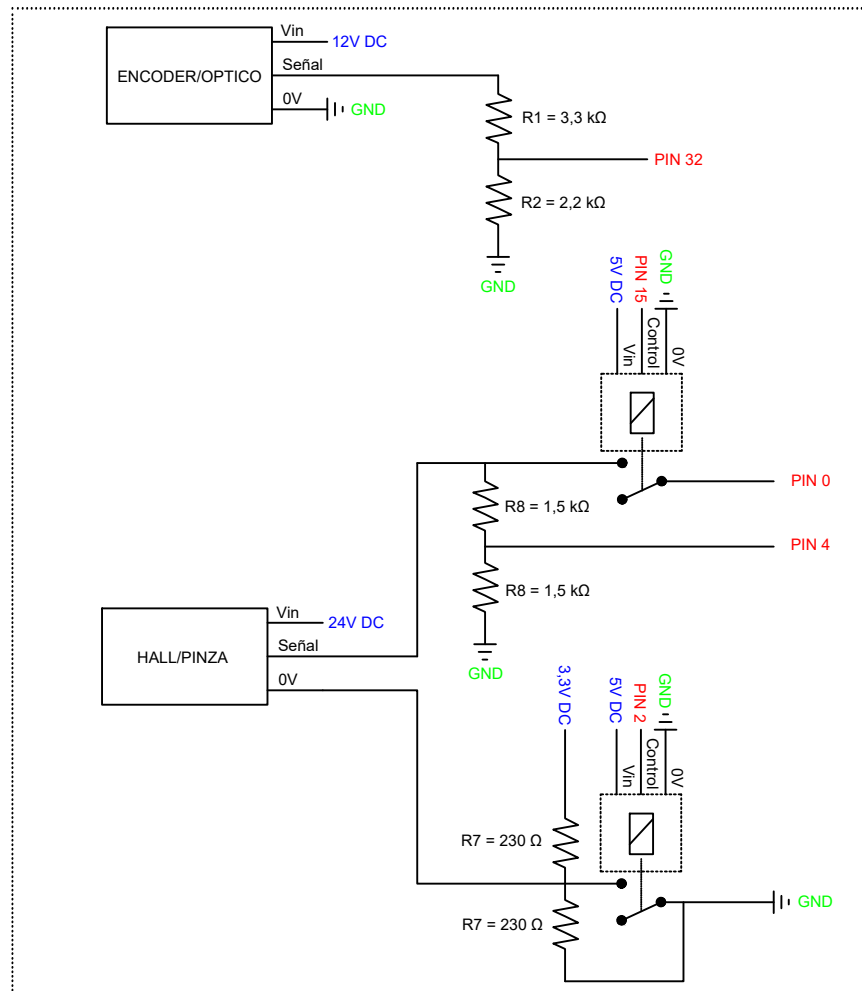
PANTALLA LCD



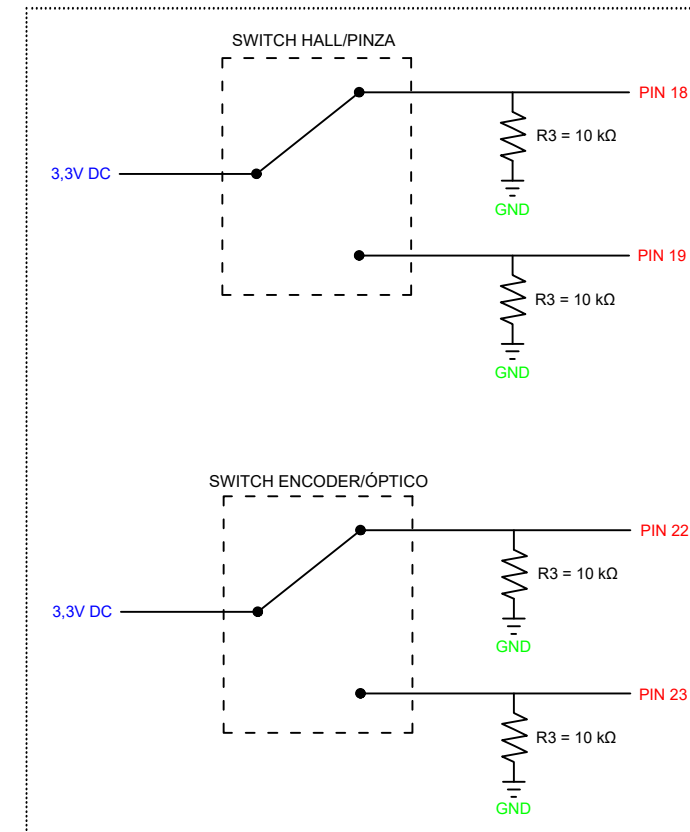
SALIDA 0V - 10V



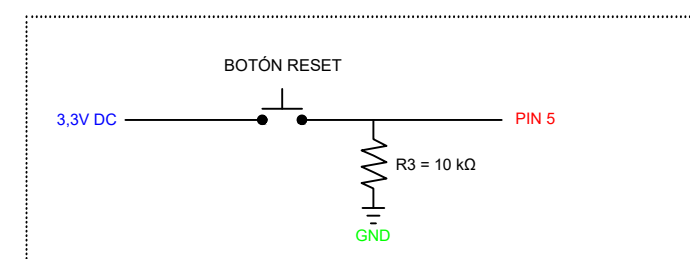
SEÑALES DE ENTRADA



SELECTORES



RESET



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



LEORDEAN, COSMIN ADRIAN
GRADO EN INGENIERÍA ELÉCTRICA

TRABAJO DE FIN DE GRADO - MEDICIÓN DE PAR Y VELOCIDAD

PLANO

CIRCUITO

SITUACIÓN: VALENCIA

FECHA

06/2021

Nº

02

PETICIONARIO: UNIVERSIDAD POLITÉCNICA DE VALENCIA

ESCALA

S/E

REFERENCIA

M-01



PLIEGO DE CONDICIONES



1. INTRODUCCIÓN

Se detalla en el presente pliego de condiciones el conjunto de las especificaciones, tanto técnicas y legales, como económicas para el correcto desarrollo del proyecto. Así, se muestran, en cada uno de los conjuntos, los requisitos contractuales de cada uno de los factores que forman el contrato.

De esta forma, en cuanto a lo técnico, específicamente se desarrollarán las condiciones y requisitos del desarrollo *hardware* y *software* del proyecto, con las características descritas en la memoria de este mismo documento. Más adelante, en cuanto a lo legal, se indican las condiciones de obtención y uso del *software*, así como del *hardware*. Finalmente, en cuanto a lo económico, se detallan las condiciones generales de pago, garantías y adquisición del producto.

2. CONDICIONES GENERALES

2.1. CONDICIONES TÉCNICAS

Para poder garantizar el correcto funcionamiento del conjunto del sistema, en un primer lugar se deberán cumplir las condiciones técnicas descritas en este punto.

Así, en cuanto al software implementado se exigirán las siguientes condiciones mínimas, con el fin de garantizar un resultado claro y eficiente.

- Condiciones de software
- Se deberá poseer un equipo informático, preferentemente, un ordenador personal.
- Dicho equipo informático deberá cumplir con las especificaciones mínimas para la instalación de un software de gran requisito de procesamiento de datos, como es MATLAB junto con la librería Simulink y la herramienta AppDesigner. Se detallan a continuación los requisitos mínimos establecidos por la empresa desarrolladora, MathWorks, de dicho software para garantizar el correcto funcionamiento del modelo de simulación.



Digitalización de un banco de ensayos y comprobación experimental

Sistema Operativo	Requisito de versión
Windows	Windows 7 SP1 en adelante, Windows Server 2008 SP2 en Adelante.
Mac	macOS 10.10 -10.11
Linux	Kernel 2.6 o superior, glibc 2.11 o superior

Tabla 15. Requisitos de sistema operativo

Componente	Requisito mínimo
Procesador	Intel o AMD x86-64 con soporte de instrucciones AVX2
Disco	2 GB para la instalación de MATLAB, 4-6 GB para una instalación típica
RAM	1 GB mínimo, 4 GB recomendado
Tarjeta gráfica	Soporte para OpenGL 3.3 recomendado con 1 GB en GPU

Tabla 16. Requisitos mínimos de componentes del equipo informático

- La conexión a internet será muy recomendada, pero no imprescindible, con el fin de poder trasladar sugerencias, dudas o informes de errores al servicio técnico.
- El desarrollador del software descrito en la presente memoria no tendrá acceso, ni de manera directa, ni indirecta, al equipo informático del promotor.
- Se garantiza el funcionamiento del modelo e interfaz gráfica descritos en la memoria. Cualquier otra función implementada o cambiada hasta el momento de la adquisición del producto no será responsabilidad del desarrollador, salvo anexo o modificación del documento, en cuyo caso esta condición quedará parcialmente inválida.
- El software se ha desarrollado con fin educativo, por lo que cualquier uso fuera de este ámbito, ya sea laboral o económico, no será responsabilidad del desarrollador.
- El uso de Matlab, Simulink y sus correspondientes herramientas se realizará contando estrictamente con una licencia, ya sea adquirida, de prueba o dedicada a estudiantes. Cualquier error de funcionamiento en el software desarrollado sin hacer uso de alguna de estas licencias no será responsabilidad del autor.
- En caso de desear visualizar parámetros y/o resultados en tiempo real, además de modificar variables del sistema real, se deberá poseer un dispositivo móvil con sistema operativo Android. Se detallan a continuación las características mínimas para el uso de la aplicación móvil.



Digitalización de un banco de ensayos y comprobación experimental

Componente	Requisito mínimo
Sistema operativo	Android 4.4 (KitKat)
Procesador	DualCore
RAM	1 GB
Bluetooth	Sí
Wi-Fi	Prescindible. Recomendable para la facilidad de instalación

Tabla 17. Requisitos mínimos de dispositivo Android

· Cualquier uso que no cumpla con estos requisitos mínimos para el dispositivo Android implicarán la ausencia de responsabilidad del desarrollador.

· La adquisición y uso del software desarrollado indicará la aceptación de las condiciones indicadas en este punto.

- Condiciones de hardware

· Utilización estricta del mismo modelo de microcontrolador descrito en esta memoria, impidiendo su reemplazo.

· La alimentación de tensión al sistema deberá realizarse a 230 V en corriente alterna, con una frecuencia de 50 Hz.

· La onda de alimentación deberá cumplir con los estándares de calidad usualmente requeridos, sobre todo en cuanto a contenido de armónicos, variación de frecuencia y huecos de tensión.

· Cualquier manipulación del componente hardware anulará cualquier garantía que pudiera tener el producto.

· Las conexiones entre el sistema de medición y los sensores, convertidores o cualquier otra aparamenta apta, se deberá realizar conforme a los planos adjuntos. De no ser así, el desarrollador carecerá de responsabilidad ante cualquier fallo eléctrico o mecánico que pudiera derivar de la incorrecta conexión.

· En cuanto al sistema de medición, este deberá implementarse en una placa agujereada o PCB, con el fin de garantizar la compactibilidad del sistema.

· La implementación se deberá realizar mediante soldadura con estaño, asegurando la correcta conexión entre componentes y sujeción a la placa agujereada.

· Cualquier sobrante de pieza metálica, o punzante, posterior a la ejecución de la soldadura deberá ser retirada previo a la entrega del producto, con el fin de prevenir posibles riesgos a la manipulación superficial del sistema.



Digitalización de un banco de ensayos y comprobación experimental

- Las conexiones con los bornes de potencia donde se encuentran tanto las entradas de alimentación y sensores, como las demás salidas, deberán efectuarse de forma que no exista parte conductora accesible del cable empleado. Además, el cableado deberá realizarse de forma que se evite el contacto directo con otro cableado.
- Los zócalos en los que se encuentra el microcontrolador y el amplificador operacional deberán ser robustos, estables y que permitan una sencilla extracción o integración del componente correspondiente, en caso de ser necesario, por parte del servicio técnico cualificado.
- El sistema de medición hardware no podrá ser manipulado superficialmente mientras se encuentre alimentado a partir de la red, ni aún utilizando guantes aislantes reglamentarios. De esta forma se evita cualquier posible daño humano e incluso material.

2.2. CONDICIONES LEGALES

Para la adquisición del sistema al completo, es decir, incluyendo tanto los componentes hardware como los componentes software, el promotor se deberá poner en contacto con el autor del presente proyecto o, en su defecto, con la Universidad Politécnica de Valencia.

Asimismo, se deberán cumplir y abonar las correspondientes partidas presentadas en el presupuesto por ambas partes. El promotor, por tanto, deberá recibir un albarán de compra, o cualquier otro documento que certifique la adquisición, en el mismo momento de la compra.

Finalmente, se deberá respetar la licencia de todo software secundario utilizado, ya sea mediante compra o adquisición de versión de prueba, o similar, de forma que se garantice el uso legal y responsable del producto desarrollado.

2.3. CONDICIONES ECONÓMICAS

- El precio de venta del producto deberá ser adecuado respecto al trabajo realizado y rango próximo al de productos similares.
- Se prohíbe la competencia desleal frente a otras instituciones



Digitalización de un banco de ensayos y comprobación experimental

- Una vez abonado el importe correspondiente por parte del promotor, el autor deberá entregar el producto con anterioridad a las 48 horas próximas a la compra.
- El método de pago podrá ser, tanto en efectivo, como con tarjeta.
- El precio indicado por el autor comprende la totalidad del producto, sin gastos secundarios ocultos.
- Se dará una garantía al promotor de funcionamiento del sistema de, al menos, 2 años. En caso de mal funcionamiento, el autor o servicio técnico correspondiente deberá cubrir las reparaciones necesarias para que el sistema funcione sin problema alguno.
- El autor o servicio técnico correspondiente podrá solicitar al promotor que abone una cantidad determinada, de acuerdo con las horas trabajadas y componentes reemplazados, en caso de mal funcionamiento del sistema y no cumplir las especificaciones descritas en las Condiciones Técnicas de este mismo documento.
- El promotor tendrá derecho a reclamar al autor la corrección del precio de oferta del producto, en caso de que este último publicitase el producto con un precio muy superior al mostrado en el presupuesto del presente documento.



PRESUPUESTO



Digitalización de un banco de ensayos y comprobación experimental

1. DESGLOSE DE PARTIDAS

CÓDIGO	RESUMEN	UDS	LONGITUD	ANCHURA	ALTURA	PARCIALES	CANTIDAD	PRECIO	IMPORTE
CAPÍTULO C01 Personal									
1.1	h Alumno								
	Horas invertidas por el alumno para el diseño, implementación y redacción del proyecto completo.								
	Modelo Simulink	1	50,00			50,00			
	Interfaz Gráfica	1	80,00			80,00			
	Sistema de medición	1	150,00			150,00			
	Comprobación experimental	1	20,00			20,00			
	Redacción de proyecto	1	60,00			60,00			
							360,00	10,00	3.600,00
	TOTAL CAPÍTULO C01 Personal								3.600,00
CAPÍTULO C02 Informática									
2.1	h Equipo informático								
	Lenovo ThinkPad E14. Procesador: Intel Core i5-10210U (4C / 8T, 1.6/4.2GHz, 6MB) RAM: 1 x 8 GB SO-DIMM DDR4-2666 Almacenamiento: 256 GB SSD M.2 2242 PCIe NVMe 3.0x2 Pantalla: 14" FHZ (1920x1080) WVA 250nits Anti-glare Gráfica: Integrada Intel UHD Graphics Sistema operativo: Windows 10 Pro 64								
	Precio de equipo (sin IVA): 660,33 €								
	Se contabilizan horas de uso en el proyecto suponiendo un periodo de amortización de 3 años. Teniendo 2000 horas anuales trabajadas, resultan 6000 horas hasta la amortización del equipo. Por tanto, el precio por hora de uso del equipo será de 0.11 €/h.								
	Modelo Simulink	1	50,00			50,00			
	Interfaz Gráfica	1	80,00			80,00			
	Sistema de medición	1	150,00			150,00			
	Comprobación experimental	1	20,00			20,00			
	Redacción de proyecto	1	60,00			60,00			
							360,00	0,11	39,60
2.2	h Licencia MATLAB								
	Licencia perpetua de Matlab, incluyendo Simulink y Simscape Electrical. Permite la ejecución del software diseñado.								
	Precio MATLAB (sin IVA): 1.657,89 € Precio Simulink (sin IVA): 2.479,34 € Precio Simscape Electrical (sin IVA): 2.479,34 € Precio total del paquete (sin IVA): 6.616,57 €								
	Se contabilizan horas de uso en el proyecto suponiendo un periodo de amortización de 3 años. Teniendo 2000 horas anuales trabajadas, resultan 6000 horas hasta la amortización del software. Por tanto, el precio por hora de uso del software completo será de 3,31 €/h.								
	Modelo Simulink	1	50,00			50,00			
	Interfaz Gráfica	1	80,00			80,00			
	Comprobación experimental	1	20,00			20,00			
							150,00	3,31	496,50



Digitalización de un banco de ensayos y comprobación experimental

2.3.	<p>h Licencia Presto Licencia anual de Presto, software utilizado para la realización del presupuesto.</p> <p>Precio Presto (sin IVA): 420,00 €</p> <p>Se contabilizan horas de uso en el proyecto suponiendo un periodo de amortización de 1 año. Teniendo 2000 horas anuales trabajadas, resultan 2000 horas hasta la amortización del software. Por tanto, el precio por hora de uso del software completo será de 0,21 €/h.</p> <table border="0"> <tr> <td>Realización de presupuesto</td> <td style="text-align: right;">1</td> <td style="text-align: right;">2,00</td> <td style="text-align: right;">2,00</td> <td></td> <td></td> </tr> </table>	Realización de presupuesto	1	2,00	2,00								
Realización de presupuesto	1	2,00	2,00										
			2,00	0,21	0,42								
2.4.	<p>h Licencia Microsoft Office Licencia anual de paquete Office, software utilizado para la realización de la memoria y presentación.</p> <p>Precio Office (sin IVA): 69 €</p> <p>Se contabilizan horas de uso en el proyecto suponiendo un periodo de amortización de 1 año. Teniendo 2000 horas anuales trabajadas, resultan 2000 horas hasta la amortización del software. Por tanto, el precio por hora de uso del software completo será de 0,03 €/h.</p> <table border="0"> <tr> <td>Redacción de proyecto</td> <td style="text-align: right;">1</td> <td style="text-align: right;">60,00</td> <td style="text-align: right;">60,00</td> <td></td> <td></td> </tr> </table>	Redacción de proyecto	1	60,00	60,00								
Redacción de proyecto	1	60,00	60,00										
			60,00	0,03	1,80								
2.5	<p>h Dispositivo Android Dispositivo Xiaomi Poco F3 Pantalla: 6,67" AMOLED 120 Hz Procesador: Qualcomm Snapdragon 870 Memoria RAM: 6 Gb Almacenamiento: 128 Gb Software: Android 11 MIUI 12 Conectividad: 5G, LTE, Bluetooth 5.1, WiFi 6</p> <p>Precio de equipo (sin IVA): 247,93 €</p> <p>Se contabilizan horas de uso en el proyecto suponiendo un periodo de amortización de 2 años. Teniendo 2000 horas anuales trabajadas, resultan 4000 horas hasta la amortización del equipo. Por tanto, el precio por hora de uso del equipo será de 0.06 €/h.</p> <table border="0"> <tr> <td>Sistema de medición</td> <td style="text-align: right;">1</td> <td style="text-align: right;">150,00</td> <td style="text-align: right;">150,00</td> <td></td> <td></td> </tr> <tr> <td>Comprobación experimental</td> <td style="text-align: right;">1</td> <td style="text-align: right;">20,00</td> <td style="text-align: right;">20,00</td> <td></td> <td></td> </tr> </table>	Sistema de medición	1	150,00	150,00			Comprobación experimental	1	20,00	20,00		
Sistema de medición	1	150,00	150,00										
Comprobación experimental	1	20,00	20,00										
			170,00	0,06	10,20								
	TOTAL CAPÍTULO C02 Informática.....				548,52								
CAPÍTULO C03 Material													
3.1	<p>ud ESP32 WROVER - B ESP32 WROVER - B con 4MB de memoria Flash, 8MB de PSRAM, Bluetooth, 2 núcleos, WiFi y protocolo SPI de comunicación</p>		1,00	10,25	10,25								
3.2	<p>ud NHD-0420D3Z-NSW-BBW-V3 Módulos de pantalla - LCD, OLED de caracteres y numérico Transmisivo 5 x 8 puntos STN - Neumático súper trenzado LED - Blanco I°C, RS-232, SPI 98.00mm x 60.00mm x 13.50mm</p>		1,00	24,58	24,58								
3.3	<p>ud Módulo Relés SRD-05VDC-SL-C Módulo de 2 relés aislado por optoacoplador y con LED de encendido y apagado para cada relé</p>		1,00	2,50	2,50								



Digitalización de un banco de ensayos y comprobación experimental

3.4	<p>ud Conversor AC/DC de 230V a 24V</p> <ul style="list-style-type: none"> • Carcasa de plástico de forma escalonada • Entrada de gama completa • Tamaño compacto • Clase de aislamiento II • Se puede instalar en carril DIN industrial TS-35/7,5 o 15 • Tensión de salida dc ajustable • Consumo de energía sin carga <0,3 W 	<hr/> <p>1,00 16,46 16,46</p>
3.5	<p>ud B2405S-1W</p> <p>Convertidor DC/DC aislado de 1W de potencia (B2405S-1W). Tensión de entrada: 24V (21,6 – 26,4V) Tensión de salida: 5V Corriente de salida: 200mA Eficiencia: 80% Encapsulado: SIP</p>	<hr/> <p>1,00 2,95 2,95</p>
3.6	<p>ud B2412S-1W</p> <p>Convertidor DC/DC aislado de 1W de potencia (B2412S-1W). Tensión de entrada: 24V (21,6 – 26,4V) Tensión de salida: 12V Corriente de salida: 200mA Eficiencia: 80% Encapsulado: SIP</p>	<hr/> <p>1,00 2,95 2,95</p>
3.7	<p>ud Resistencia 3.3 kOhm</p> <p>Resistencia 3.3kO Potencia Nominal 0.6W Tolerancia ±0.5% Axial/Radial Axial Tecnología Película de Metal Paquete/Carcasa 0623 Mínima Temperatura de Funcionamiento -55°C Máxima Temperatura de Funcionamiento +155°C</p>	<hr/> <p>1,00 2,95 2,95</p>
3.8	<p>ud Resistencia 2.2 kOhm</p> <ul style="list-style-type: none"> • Alta potencia en formato compacto • excelente estabilidad a largo plazo • disipación nominal 2W a 70 °C. • tolerancia De Resistencia de ±5 % • ±300ppm/°C Coeficiente de temperatura • tensión del elemento Limitador 350V • tensión de sobrecarga 600V (máx.) • 1000MO de resistencia De Aislamiento • recubrimiento resistente a disolventes • Alta capacidad de sobrecarga/sobretensión • Construcción completamente ignífuga 	<hr/> <p>1,00 0,05 0,05</p> <hr/> <p>1,00 0,09 0,09</p>



Digitalización de un banco de ensayos y comprobación experimental

3.9	<p>ud Resistencia 10 kOhm Resistencias de película metálica fina profesionales Disipación de potencia nominal de hasta 1 W Excelente estabilidad general: clase 0,25 Amplia gama profesional: 0,22 O a 22 MO Terminación de cable sin plomo (Pb) Estañado puro que ofrece compatibilidad con procesos de soldadura con y sin plomo Aplicaciones industriales, telecomunicaciones, equipos médicos y automoción</p>	5,00	0,07	0,35
3.10	<p>ud Resistencia 500 Ohm Resistencia: 500 Ohms Frecuencia de energía: 600 mW Tolerancia: 1 % Coeficiente de temperatura: 50 PPM / C Temperatura operativa mínima: - 55 C Temperatura operativa máxima: + 155 C Voltaje máximo: 350 V</p>	1,00	0,09	0,09
3.11	<p>ud Resistencia 90 kOhm Resistencia 90 kOhms Tolerancia ±5% Potencia (Vatios) 0.25W, 1/4W</p>	1,00	0,09	0,09
3.12	<p>ud Resistencia 45 kOhm Resistencia: 45 kOhms Frecuencia de energía: 166 mW (1/6 W) Tolerancia: 5 % Coeficiente de temperatura: - 350 PPM / C, + 500 PPM / C Temperatura operativa mínima: - 55 C Temperatura operativa máxima: + 155 C Voltaje máximo: 150 V</p>	1,00	0,09	0,09
3.13	<p>ud Resistencia 230 Ohm Resistencia: 230 Ohms Tolerancia: 5 % Coeficiente de temperatura: - 350 PPM / C, + 500 PPM / C Temperatura operativa mínima: - 55 C Temperatura operativa máxima: + 155 C Voltaje máximo: 150 V</p>	2,00	0,08	0,16
3.14	<p>ud Resistencia 1,5 kOhm Resistencia: 1,5 kOhms Tolerancia: 5 % Coeficiente de temperatura: - 350 PPM / C, + 500 PPM / C Temperatura operativa mínima: - 55 C Temperatura operativa máxima: + 155 C Voltaje máximo: 150 V</p>	2,00	0,08	0,16



Digitalización de un banco de ensayos y comprobación experimental

3.15	<p>ud Placa perforada Papel SRBP FR2 1,5 mm, soldado lacado Fibra de vidrio EPOXI FR4 1,5 mm; nivelada por aire caliente (HAL, sin plomo) Tamaño de placa 100 x 160 x 1,6 mm Diseño de orificio 37 x 57 Diám. del orificio de 1,0 mm Paso de 2,54 mm Espesor del cobre 35 µm</p>	1,00	5,55	5,55								
3.16	<p>ud Amplificador operacional OPA705PA Empaquetado / Estuche: PDIP-8 Corriente de salida por canal: 10 mA Vos - Voltaje de compensación de la entrada: 5 mV Temperatura operativa mínima: - 40 C Temperatura operativa máxima: + 85 C Ib - Corriente de polarización de entrada: 10 pA CMRR - Ratio de rechazo en modo común : 77 dB Serie: OPA705 Empaquetado: Tube Tipo de amplificador: Low Input Bias Current Amplifier Características: High Clload Drive Altura: 4.57 mm Tipo de entrada: Rail-to-Rail</p>	1,00	1,49	1,49								
3.17	<p>ud Conmutador de palanca Tipo de interruptor de palanca Cantidad de todas las posiciones 3 Número de posiciones estables 2 posiciones Configuración de contactos SP3T Modo de actuación (La posición entre paréntesis es una posición temporal) ON-OFF-(ON) Capacidad de carga de contactos CA @R (a carga resistiva) 2A / 250V CA</p>	2,00	2,93	5,86								
3.18	<p>ud Pulsador empotrable Pulsador empotrable OFF-ON abierto en reposo Electro DH 11.542 Características: - Pulsador empotrable OFF-ON. - Fijación a rosca - Circuito abierto. - Actúa al pulsar. - Ref. 11.542 - Funcionamiento OFF-ON (abierto en reposo) - Potencia 0'5A/ 250V</p>	1,00	0,95	0,95								
3.19	<p>m Cableado rígido Cable de conexión rígido monofilar de cobre estañado, perfecto para tareas de prototipado y diseño de circuitos electrónicos.</p> <p>- Conductores: 0.5mm - Diámetro Exterior Total: 1mm</p> <table border="0" style="width: 100%;"> <tr> <td>Cableado interno</td> <td style="text-align: right;">1</td> <td style="text-align: right;">8,00</td> <td style="text-align: right;">8,00</td> </tr> <tr> <td>Cableado externo</td> <td style="text-align: right;">1</td> <td style="text-align: right;">10,00</td> <td style="text-align: right;">10,00</td> </tr> </table>	Cableado interno	1	8,00	8,00	Cableado externo	1	10,00	10,00	18,00	0,19	3,42
Cableado interno	1	8,00	8,00									
Cableado externo	1	10,00	10,00									



Digitalización de un banco de ensayos y comprobación experimental

3.20	<p>g Estaño trimetal MBO 0.5mm Hilo de estaño, aleación de estaño, plomo y cobre (60/38/2) con flux CA2 de diámetro 0.5mm en carrete de 100gr.</p> <p>Este estaño de alta calidad de MBO sustituye un parte de plomo por cobre lo que junto al núcleo de flux que incorpora garantiza una fácil aplicación y una soldadura perfecta.</p> <p>Composición:</p> <ul style="list-style-type: none"> - Estaño: 60% - Plomo: 38% - Cobre: 2% <p>Medición en gramos. Estaño utilizado en placa</p>										
	<table border="0" style="width: 100%;"> <tr> <td style="width: 10%; text-align: center;">1</td> <td style="width: 30%; text-align: right;">80,00</td> <td style="width: 30%; text-align: right;">80,00</td> <td style="width: 10%;"></td> <td style="width: 10%;"></td> </tr> <tr> <td></td> <td></td> <td style="border-top: 1px solid black; text-align: right;">80,00</td> <td style="border-top: 1px solid black; text-align: right;">0,07</td> <td style="border-top: 1px solid black; text-align: right;">5,60</td> </tr> </table>	1	80,00	80,00					80,00	0,07	5,60
1	80,00	80,00									
		80,00	0,07	5,60							
	<p>TOTAL CAPÍTULO C03 Material..... 83,64</p> <hr/> <p>TOTAL 4.232,16</p>										

2. RESUMEN DE PRESUPUESTO

CAPITULO	RESUMEN	EUROS	%
C01	Personal.....	3.600,00	85,06
C02	Informática.....	548,52	12,96
C03	Material.....	83,64	1,98
	TOTAL EJECUCIÓN MATERIAL	4.232,16	
	13,00 % Gastos generales.....	550,18	
	6,00 % Beneficio industrial.....	253,93	
		SUMA DE G.G. y B.I.	804,11
	21,00 % I.V.A.....	1.057,62	
	TOTAL PRESUPUESTO CONTRATA	6.093,89	
	TOTAL PRESUPUESTO GENERAL	6.093,89	

Asciende el presupuesto general a la expresada cantidad de SEIS MIL NOVENTA Y TRES EUROS con OCHENTA Y NUEVE CÉNTIMOS

Valencia, a 13 de julio de 2021.



3. JUSTIFICACIÓN DE PRESUPUESTO

Al tratarse de un presupuesto realizado teniendo en cuenta una serie de gastos iniciales básicos, como son las horas invertidas y las licencias y equipos informáticos necesarios para el diseño y desarrollo del sistema, es necesario justificar ciertos aspectos relativos a los precios estipulados en el presupuesto mostrado en los puntos anteriores.

En primer lugar, en cuanto al capítulo C01, de personal, sólo existe una partida referente a las horas invertidas por el alumno para el desarrollo del proyecto. Se asigna así un precio de 10 €/h, el cual es próximo al de cargos similares a la fecha de redacción del proyecto.

En segundo lugar, en cuanto al capítulo C02, de informática, existen diversas partidas relativas al equipo informático utilizado o licencias de software necesarias. Puesto que ninguno de ambos se ha utilizado en su totalidad y durante todo su periodo útil para el desarrollo de este proyecto, se ha estimado, como se puede ver en el punto 1 de este apartado, un precio por hora. De esta forma, asumiendo un tiempo de amortización t_{am} , el cálculo del precio por hora quedaría de la siguiente forma:

$$P_{hora}[\text{€/h}] = \frac{P_{inform} [\text{€}]}{t_{am} [\text{h}]}$$

Así, en cada caso, a partir de las horas de uso de cada equipo o software y este precio por hora, se obtiene el importe final de cada partida.

En tercer lugar, se comprenden únicamente los materiales utilizados para el diseño del prototipo de medición, por lo que no se ha realizado ningún cálculo adicional.

Finalmente, puesto a que cada vez que se realiza una venta del sistema en conjunto no se ha de diseñar, desarrollar ni comprar licencias u equipo nuevo, el precio obtenido del presupuesto se desvía de un precio de venta real que se podría ofertar. Así, se estima un número medio de ventas, el cual se sitúa en 1000 unidades del sistema completo. En cada una de estas mil unidades se encontrará la partida de materiales completa, pues esta siempre se tendrá que incorporar de nuevo al prototipo. Por otro lado, en cada una de ellas se añadirá la parte proporcional del coste del diseño y desarrollo del sistema, además del equipo informático y licencias de software necesarias. De esta forma, el precio final vendrá determinado por:

$$PEM = P_{materiales} + \frac{P_{h-lic-eq}}{1000}$$

Así, el presupuesto de ejecución material [PEM] final y el precio final (P_{final}) considerando IVA, Gastos Generales y Beneficio Industrial, del sistema completo será:

$$PEM = 87,79 \text{ €} \rightarrow P_{final} = 126,41 \text{ €}$$

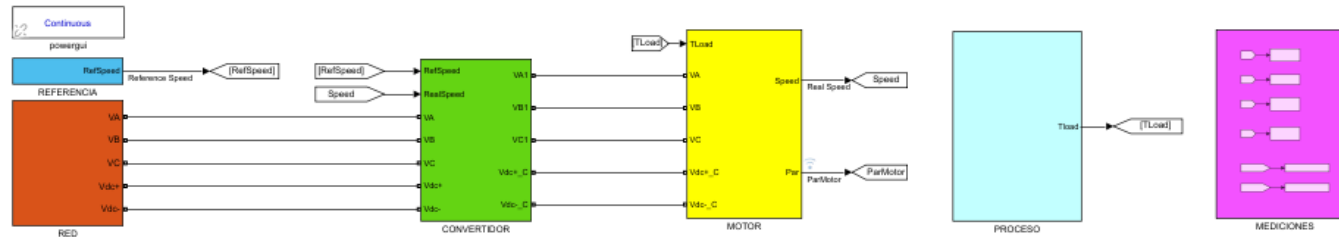


ANEXOS



ANEXO 1 – MODELADO DE BANCO DE ENSAYOS

MODELO



adria

13-Jul-2021 21:57:55

Tabla de contenidos

[Model - MODELO](#)
[System - MODELO](#)
[System - MODELO/CONVERTIDOR](#)
[System - MODELO/CONVERTIDOR/CONTROL AC](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/HARDWARE](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/BUS DC](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/INVERSOR](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/RECTIFICADOR](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC/Estimador Flujo Par](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC/Flux Sector](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC/Referencia Flujo](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC/Regulador Flux \(hysteresis\)](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC/Regulador Torque \(hysteresis\)](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC/Regulador Velocidad](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC/Switching table](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/V/F](#)
[System - MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/myPWM](#)
[System - MODELO/CONVERTIDOR/CONTROL DC](#)
[System - MODELO/MEDICIONES](#)
[System - MODELO/MOTOR](#)
[System - MODELO/MOTOR/Motor DC](#)
[System - MODELO/MOTOR/Motor IM](#)
[System - MODELO/PROCESO](#)
[System - MODELO/RED](#)
[System - MODELO/REFERENCIA](#)
[Appendix](#)

Lista de tablas

1. [From Block Properties](#)
2. [Goto Block Properties](#)
3. [PSB option menu block Block Properties](#)
4. [Constant Block Properties](#)

5. [From Block Properties](#)
6. [Goto Block Properties](#)
7. [Inport Block Properties](#)
8. [MultiPortSwitch Block Properties](#)
9. [PMIOPort Block Properties](#)
10. [Inport Block Properties](#)
11. [Outport Block Properties](#)
12. [PMIOPort Block Properties](#)
13. [Inport Block Properties](#)
14. [Outport Block Properties](#)
15. [PMIOPort Block Properties](#)
16. [Three-Phase VI Measurement Block Properties](#)
17. [Current Measurement Block Properties](#)
18. [Outport Block Properties](#)
19. [PMIOPort Block Properties](#)
20. [Series RLC Branch Block Properties](#)
21. [Voltage Measurement Block Properties](#)
22. [Demux Block Properties](#)
23. [From Block Properties](#)
24. [Goto Block Properties](#)
25. [IGBT/Diode Block Properties](#)
26. [Inport Block Properties](#)
27. [PMIOPort Block Properties](#)
28. [Terminator Block Properties](#)
29. [Diode Block Properties](#)
30. [PMIOPort Block Properties](#)
31. [Terminator Block Properties](#)
32. [Constant Block Properties](#)
33. [Inport Block Properties](#)
34. [MultiPortSwitch Block Properties](#)
35. [Outport Block Properties](#)
36. [Display Block Properties](#)
37. [Inport Block Properties](#)
38. [Outport Block Properties](#)
39. [ComplexToMagnitudeAngle Block Properties](#)
40. [Constant Block Properties](#)
41. [Demux Block Properties](#)
42. [Gain Block Properties](#)
43. [Inport Block Properties](#)
44. [Integrator Block Properties](#)
45. [Outport Block Properties](#)
46. [Product Block Properties](#)
47. [RealImagToComplex Block Properties](#)
48. [Sum Block Properties](#)
49. [Terminator Block Properties](#)
50. [abc to Alpha-Beta-Zero Transformation Block Properties](#)
51. [Constant Block Properties](#)
52. [DataTypeConversion Block Properties](#)
53. [Gain Block Properties](#)
54. [Inport Block Properties](#)
55. [Logic Block Properties](#)
56. [Mux Block Properties](#)
57. [Outport Block Properties](#)
58. [RelationalOperator Block Properties](#)
59. [Sum Block Properties](#)
60. [Abs Block Properties](#)

61. [Fcn Block Properties](#)
62. [Inport Block Properties](#)
63. [Outport Block Properties](#)
64. [Saturate Block Properties](#)
65. [Display Block Properties](#)
66. [Inport Block Properties](#)
67. [Outport Block Properties](#)
68. [Relay Block Properties](#)
69. [Sum Block Properties](#)
70. [DataTypeConversion Block Properties](#)
71. [Display Block Properties](#)
72. [Gain Block Properties](#)
73. [Inport Block Properties](#)
74. [Logic Block Properties](#)
75. [Outport Block Properties](#)
76. [Relay Block Properties](#)
77. [Sum Block Properties](#)
78. [Constant Block Properties](#)
79. [Derivative Block Properties](#)
80. [Display Block Properties](#)
81. [Gain Block Properties](#)
82. [Inport Block Properties](#)
83. [Integrator Block Properties](#)
84. [Outport Block Properties](#)
85. [Product Block Properties](#)
86. [Saturate Block Properties](#)
87. [Sum Block Properties](#)
88. [Constant Block Properties](#)
89. [Display Block Properties](#)
90. [Inport Block Properties](#)
91. [Lookup2D Block Properties](#)
92. [MultiPortSwitch Block Properties](#)
93. [Outport Block Properties](#)
94. [Sum Block Properties](#)
95. [Outport Block Properties](#)
96. [PWM Generator \(2-Level\) Block Properties](#)
97. [Constant Block Properties](#)
98. [Demux Block Properties](#)
99. [Gain Block Properties](#)
100. [Inport Block Properties](#)
101. [Integrator Block Properties](#)
102. [Logic Block Properties](#)
103. [Mux Block Properties](#)
104. [Outport Block Properties](#)
105. [Product Block Properties](#)
106. [RelationalOperator Block Properties](#)
107. [Repeating table Block Properties](#)
108. [Sum Block Properties](#)
109. [Trigonometry Block Properties](#)
110. [Current Measurement Block Properties](#)
111. [Diode Block Properties](#)
112. [DiscretePulseGenerator Block Properties](#)
113. [Goto Block Properties](#)
114. [Mosfet Block Properties](#)
115. [PMIOPort Block Properties](#)
116. [Voltage Measurement Block Properties](#)

117. [From Block Properties](#)
118. [ToWorkspace Block Properties](#)
119. [Constant Block Properties](#)
120. [From Block Properties](#)
121. [Gain Block Properties](#)
122. [Goto Block Properties](#)
123. [Inport Block Properties](#)
124. [MultiPortSwitch Block Properties](#)
125. [Outport Block Properties](#)
126. [PMIOPort Block Properties](#)
127. [BusSelector Block Properties](#)
128. [DC Voltage Source Block Properties](#)
129. [DC machine Block Properties](#)
130. [Inport Block Properties](#)
131. [Outport Block Properties](#)
132. [PMIOPort Block Properties](#)
133. [Asynchronous Machine Block Properties](#)
134. [BusSelector Block Properties](#)
135. [Constant Block Properties](#)
136. [From Block Properties](#)
137. [Goto Block Properties](#)
138. [Inport Block Properties](#)
139. [MultiPortSwitch Block Properties](#)
140. [Outport Block Properties](#)
141. [PMIOPort Block Properties](#)
142. [Three-Phase Breaker Block Properties](#)
143. [Constant Block Properties](#)
144. [From Block Properties](#)
145. [Gain Block Properties](#)
146. [Math Block Properties](#)
147. [MultiPortSwitch Block Properties](#)
148. [Outport Block Properties](#)
149. [Saturate Block Properties](#)
150. [Step Block Properties](#)
151. [Sum Block Properties](#)
152. [DC Voltage Source Block Properties](#)
153. [Ground Block Properties](#)
154. [PMIOPort Block Properties](#)
155. [Three-Phase Programmable Voltage Source Block Properties](#)
156. [Constant Block Properties](#)
157. [Outport Block Properties](#)
158. [Block Type Count](#)
159. [Model Variables](#)
160. [Model Functions](#)

Model - MODELO

Full Model Hierarchy

1. [MODELO](#)
 1. [CONVERTIDOR](#)
 1. [CONTROL AC](#)
 1. [HARDWARE](#)

1. [BUS DC](#)
2. [INVERSOR](#)
3. [RECTIFICADOR](#)

2. [SOFTWARE](#)

1. [DTC](#)

1. [Estimador Flujo Par](#)
2. [Flux Sector](#)
3. [Referencia Flujo](#)
4. [Regulador Flux \(hysteresis\)](#)
5. [Regulador Torque \(hysteresis\)](#)
6. [Regulador Velocidad](#)
7. [Switching table](#)

2. [V/F](#)
3. [myPWM](#)

2. [CONTROL DC](#)

2. [MEDICIONES](#)

3. [MOTOR](#)

1. [Motor DC](#)
2. [Motor IM](#)

4. [PROCESO](#)

5. [RED](#)

6. [REFERENCIA](#)

Simulation Parameter	Value
Solver	ode4
RelTol	1e-3
Refine	1
MaxOrder	5
FixedStep	0.00001
ZeroCross	on

[\[more info\]](#)

System - MODELO

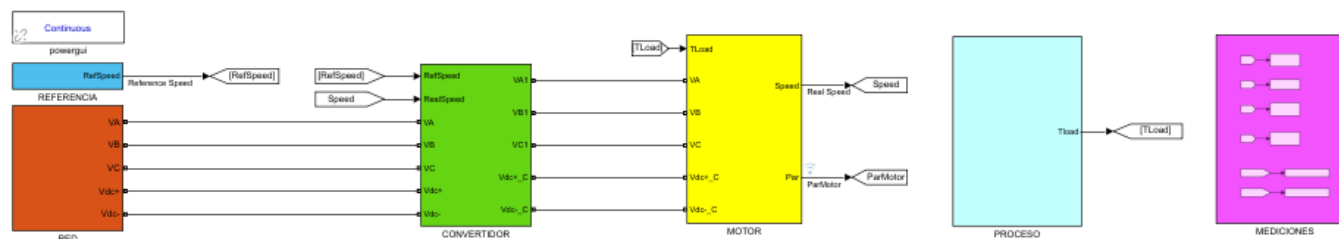


Tabla 1. From Block Properties

Name	Goto Tag	Icon Display	Goto Blk Name	Goto Blk Location	Defined In Blk
------	----------	--------------	---------------	-------------------	----------------

From	Speed	Tag	Goto2	MODELO	rad/s to rpm
From2	TLoad	Tag	Goto4	MODELO	Multiport Switch
From4	RefSpeed	Tag	Goto1	MODELO	Constant

Tabla 2. Goto Block Properties

Name	Goto Tag	Icon Display	Tag Visibility	From Blk	From Blk Location	Used By Blk
Goto1	RefSpeed	Tag	local	From4	MODELO	Sum1
Goto2	Speed	Tag	global	From2 , From1 , From1 , From	MODELO/PROCESO , MODELO/PROCESO , MODELO/MEDICIONES , MODELO	Math Function1 , Math Function , Velocidad , Sum1 , Abs
Goto3	ParMotor	Tag	global	From3	MODELO/MEDICIONES	Par
Goto4	TLoad	Tag	local	From2	MODELO	Sum , 1_Tb2 , 1_Tb2

Tabla 3. PSB option menu block Block Properties

Name	Simulation Mode	Iterations	Frequencyindice	Pbase	Err Max	Units V	Units W	Function Messages	Echomessages	Current Source Switches	Disable Snubber Devices	Disable Ron Switches	Disable Vf Switches	Display Equations	Method	X0status	Nonlinear Tolerance	NMax Iteration	Continue On Max Iteration
powergui	Continuous	50	0	100e6	1e-4	kV	MW	off	off	off	off	off	off	off	off	blocks	1e-4	100	off

System - [MODELO/CONVERTIDOR](#)

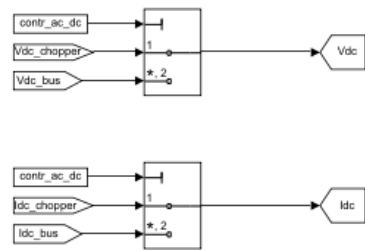
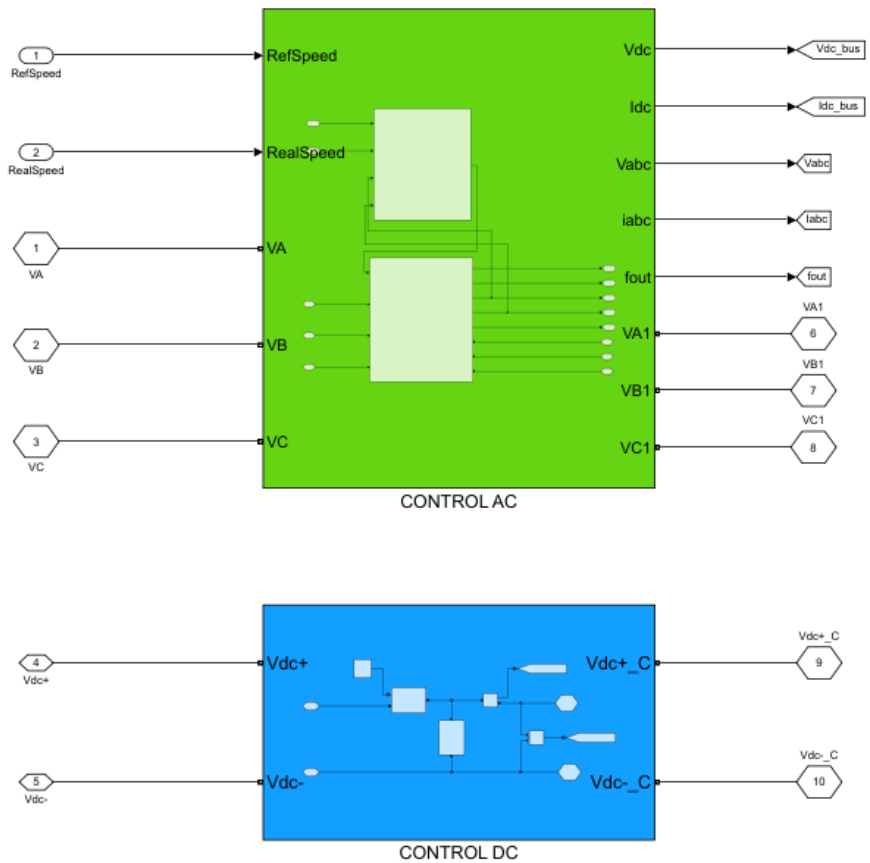


Tabla 4. Constant Block Properties

Name	Value	Out Data Type Str	Lock Scale	Sample Time	Frame Period
Constant	contr_ac_dc	Inherit: Inherit from 'Constant value'	off	inf	inf
Constant1	contr_ac_dc	Inherit: Inherit from 'Constant value'	off	inf	inf

Tabla 5. From Block Properties

Name	Goto Tag	Icon Display	Goto Blk Name	Goto Blk Location	Defined In Blk
From	Vdc_chopper	Tag	Goto1	MODELO/CONVERTIDOR/CONTROL DC	do not delete this gain
From1	Vdc_bus	Tag	Goto4	MODELO/CONVERTIDOR	do not delete this gain
From2	Idc_bus	Tag	Goto5	MODELO/CONVERTIDOR	do not delete this gain
From3	Idc_chopper	Tag	Goto	MODELO/CONVERTIDOR/CONTROL DC	do not delete this gain

Tabla 6. Goto Block Properties

Name	Goto Tag	Icon Display	Tag Visibility	From Blk	From Blk Location	Used By Blk
Goto	Vdc	Tag	global	From7	MODELO/MEDICIONES	Vdc
Goto1	fout	Tag	global	Unconnected	Unconnected	MODELO (model)
Goto2	Iabc	Tag	global	From8	MODELO/MEDICIONES	Iabc
Goto3	Vabc	Tag	global	From5	MODELO/MEDICIONES	Vabc
Goto4	Vdc_bus	Tag	global	From1	MODELO/CONVERTIDOR	Multiport Switch
Goto5	Idc_bus	Tag	global	From2	MODELO/CONVERTIDOR	Multiport Switch1
Goto6	Idc	Tag	global	From6	MODELO/MEDICIONES	Idc

Tabla 7. Inport Block Properties

Name	Port	Defined In Blk
RealSpeed	2	rad/s to rpm
RefSpeed	1	Constant

Tabla 8. MultiPortSwitch Block Properties

Name	Data Port Order	Inputs	Data Port Indices	Data Port For Default	Diagnostic For Default	Input Same DT	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow	Allow Diff Input Sizes
Multiport Switch	One-based contiguous	2	{1,2,3}	Last data port	Error	off	Inherit: Inherit via internal rule	off	Floor	off	off
Multiport Switch1	One-based contiguous	2	{1,2,3}	Last data port	Error	off	Inherit: Inherit via internal rule	off	Floor	off	off

Tabla 9. PMIOPort Block Properties

Name	Port	Side
VA	1	Left
VA1	6	Right
VB	2	Left
VB1	7	Right
VC	3	Left
VC1	8	Right
Vdc+	4	Left
Vdc+_C	9	Right
Vdc-	5	Left
Vdc-_C	10	Right

System - [MODELO/CONVERTIDOR/CONTROL AC](#)

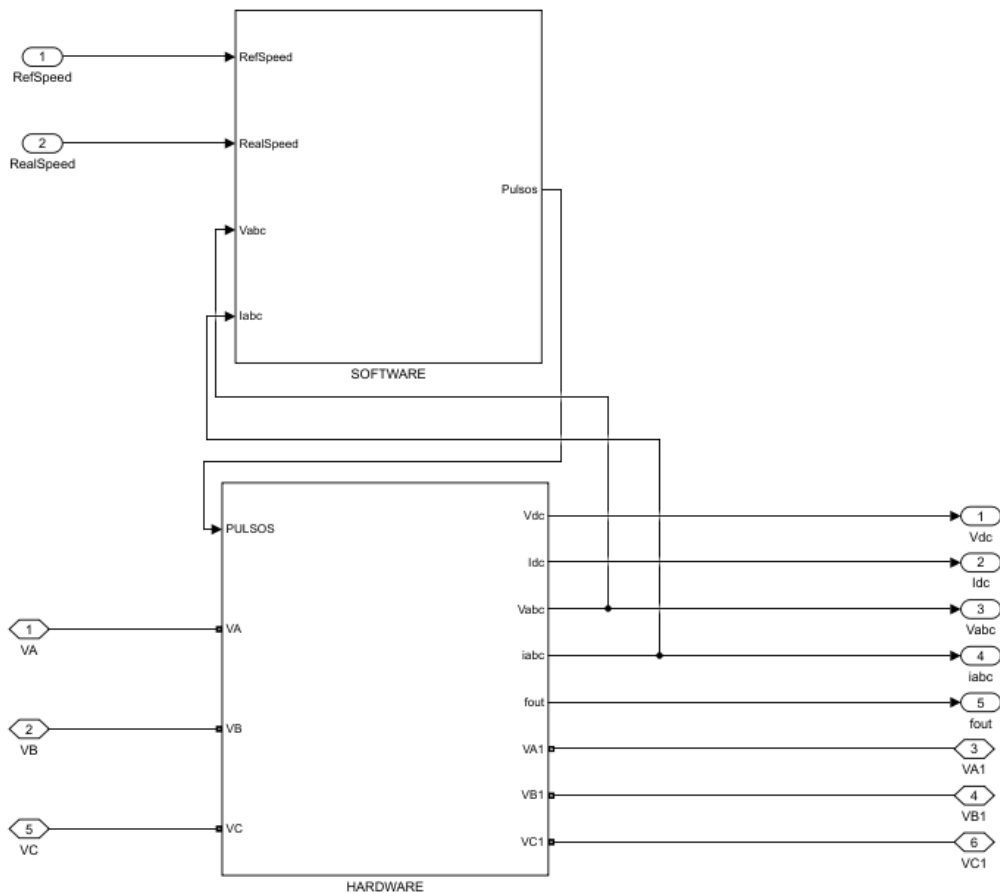


Tabla 10. Inport Block Properties

Name	Port	Defined In Blk
RealSpeed	2	rad/s to rpm
RefSpeed	1	Constant

Tabla 11. Outport Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Outport Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Un Connected	Output When Unconnected Value	Vector Params As 1DFor Out When Unconnected	Used By Blk
fout	5	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	MODELO (model)
iabc	4	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Gain3, iabc
Idc	2	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Multiport Switch1
Vabc	3	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Gain3, Vabc

Vdc	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Multiport Switch
-----	---	------	-------------	-----	---------	---------	------	-----	--------	------	-----	-----	---	----	----------------------------------

Tabla 12. PMIOPort Block Properties

Name	Port	Side
VA	1	Left
VA1	3	Right
VB	2	Left
VB1	4	Right
VC	5	Left
VC1	6	Right

System - [MODELO/CONVERTIDOR/CONTROL AC/HARDWARE](#)

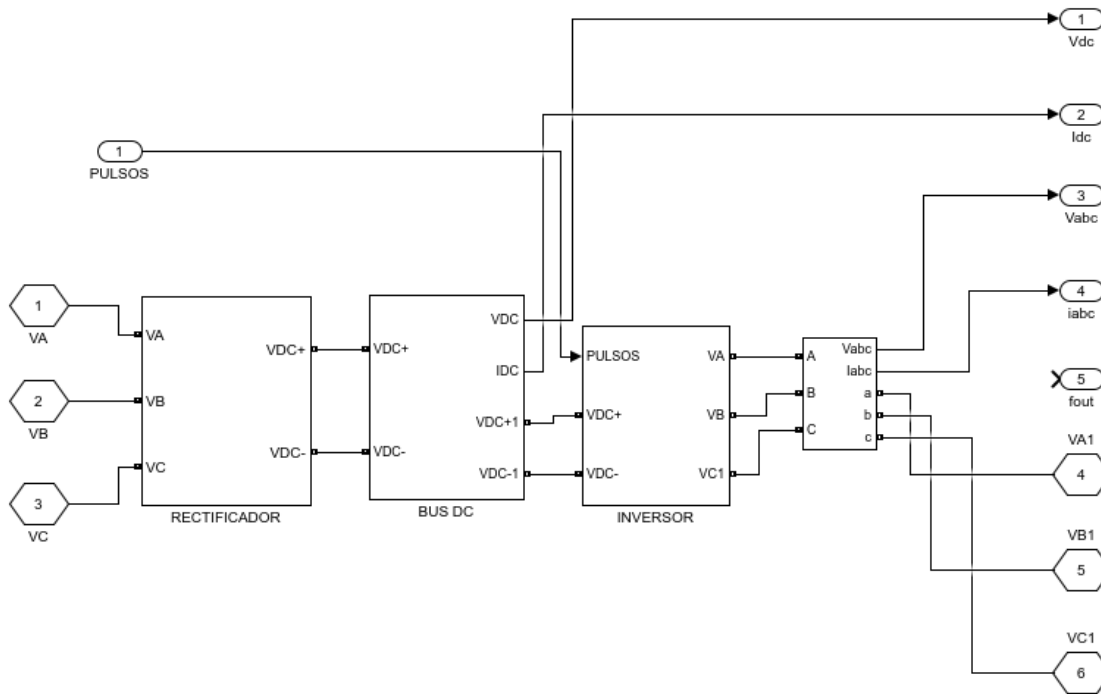


Tabla 13. Inport Block Properties

Name	Port	Defined In Blk
PULSOS	1	Multiport Switch

Tabla 14. Outport Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Output Is	Source Of Initial Output	Output When	Must Resolve To Signal	Output When Un	Output When Unconnected	Vector Params As 1DFor Out	Used By Blk
------	------	---------------	--------------	------------	------	--------------	-------------	------------------	--------------------------	-------------	------------------------	----------------	-------------------------	----------------------------	-------------

								Virtual	Value	Disabled	Object	Connected	Value	When Unconnected	
fout	5	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Unconnected
iabc	4	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Gain3, iabc
Idc	2	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Multiport Switch1
Vabc	3	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Gain3, Vabc
Vdc	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Multiport Switch

Tabla 15. PMIOPort Block Properties

Name	Port	Side
VA	1	Left
VA1	4	Right
VB	2	Left
VB1	5	Right
VC	3	Left
VC1	6	Right

Tabla 16. Three-Phase VI Measurement Block Properties

Name	Voltage Measurement	Set Label V	Vpu	Current Measurement	Set Label I	Ipu
Three-Phase V-I Measurement	phase-to-ground	off	off	yes	off	off

System - [MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/BUS DC](#)

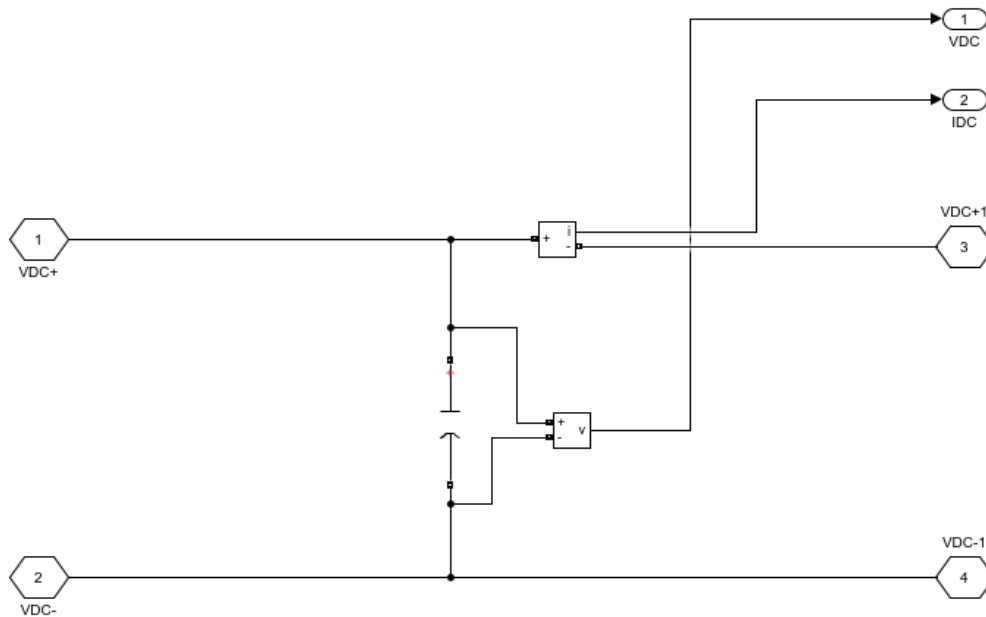


Tabla 17. Current Measurement Block Properties

Name
Current Measurement

Tabla 18. Output Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Output Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Unconnected	Output When Unconnected Value	Vector Params As 1D For Out When Unconnected	Used By Blk
IDC	2	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Multiport Switch1
VDC	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Multiport Switch

Tabla 19. PMIOPort Block Properties

Name	Port	Side
VDC+	1	Left
VDC+1	3	Right
VDC-	2	Left
VDC-1	4	Right

Tabla 20. Series RLC Branch Block Properties

Name	Branch Type	Capacitance	Setx 0	Measurements
Series RLC Branch	C	0.005	off	None

Tabla 21. Voltage Measurement Block Properties

Name
Voltage Measurement

System - [MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/INVERSOR](#)

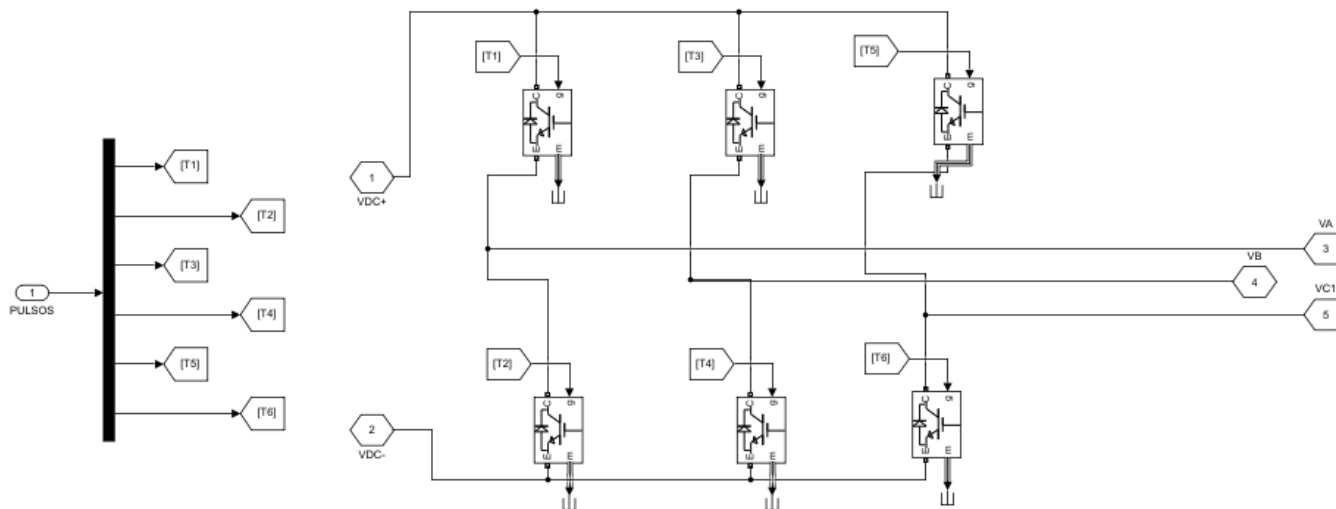


Tabla 22. Demux Block Properties

Name	Outputs	Display Option	Bus Selection Mode
Demux	6	bar	off

Tabla 23. From Block Properties

Name	Goto Tag	Icon Display	Goto Blk Name	Goto Blk Location	Defined In Blk
From	T1	Tag	Goto	MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/INVERSOR	Demux
From1	T2	Tag	Goto1	MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/INVERSOR	Demux
From2	T3	Tag	Goto2	MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/INVERSOR	Demux
From3	T4	Tag	Goto3	MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/INVERSOR	Demux
From4	T5	Tag	Goto4	MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/INVERSOR	Demux
From5	T6	Tag	Goto5	MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/INVERSOR	Demux

Tabla 24. Goto Block Properties

Name	Goto Tag	Icon Display	Tag Visibility	From Blk	From Blk Location	Used By Blk
Goto	T1	Tag	local	From	MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/INVERSOR	Data Type Conversion
Goto1	T2	Tag	local	From1	MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/INVERSOR	Data Type Conversion
Goto2	T3	Tag	local	From2	MODELO/CONVERTIDOR/CONTROL	Data Type Conversion

					AC/HARDWARE/INVERSOR	
Goto3	T4	Tag	local	From3	MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/INVERSOR	Data Type Conversion
Goto4	T5	Tag	local	From4	MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/INVERSOR	Data Type Conversion
Goto5	T6	Tag	local	From5	MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/INVERSOR	Data Type Conversion

Tabla 25. IGBT/Diode Block Properties

Name	Ron	Rs	Cs	Measurements
IGBT/Diode	1e-3	1e5	inf	on
IGBT/Diode1	1e-3	1e5	inf	on
IGBT/Diode2	1e-3	1e5	inf	on
IGBT/Diode3	1e-3	1e5	inf	on
IGBT/Diode4	1e-3	1e5	inf	on
IGBT/Diode5	1e-3	1e5	inf	on

Tabla 26. Inport Block Properties

Name	Port	Defined In Blk
PULSOS	1	Multiport Switch

Tabla 27. PMIOPort Block Properties

Name	Port	Side
VA	3	Right
VB	4	Right
VC1	5	Right
VDC+	1	Left
VDC-	2	Left

Tabla 28. Terminator Block Properties

Name
Terminator
Terminator1
Terminator2
Terminator3
Terminator4
Terminator5

System - [MODELO/CONVERTIDOR/CONTROL AC/HARDWARE/RECTIFICADOR](#)

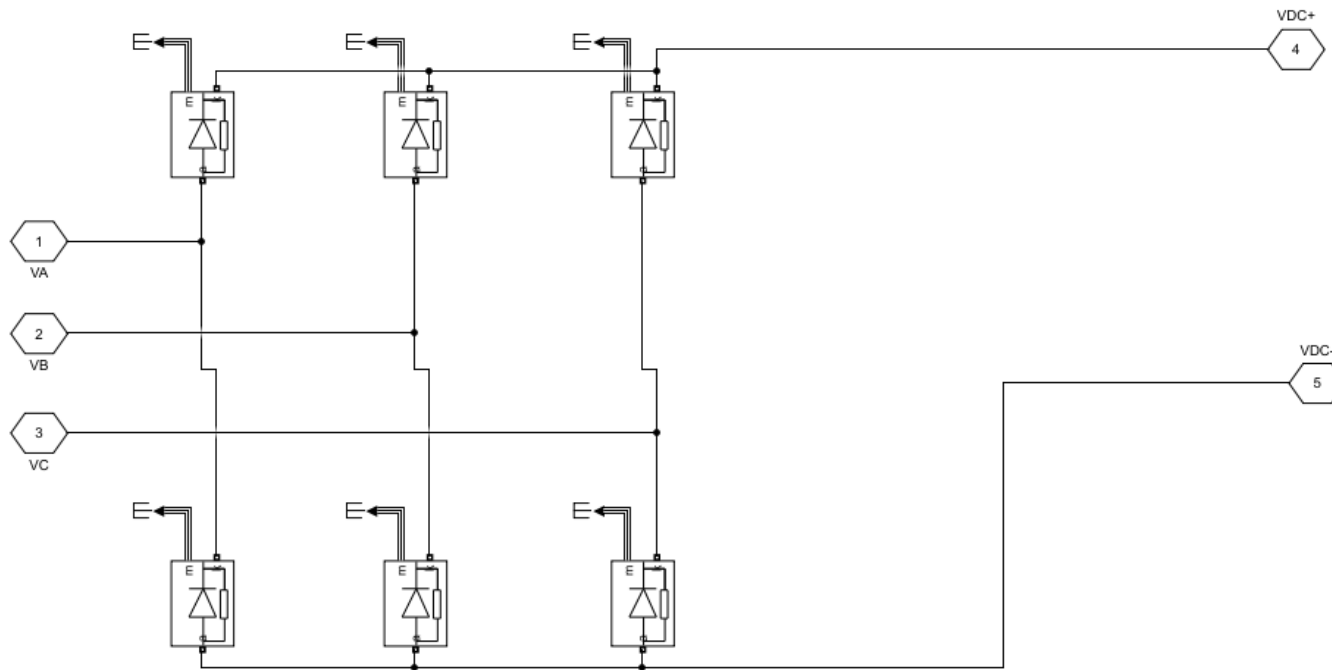


Tabla 29. Diode Block Properties

Name	Ron	Lon	Vf	IC	Rs	Cs	Measurements
Diode	0.001	0	0.8	0	500	250e-9	on
Diode1	0.001	0	0.8	0	500	250e-9	on
Diode2	0.001	0	0.8	0	500	250e-9	on
Diode3	0.001	0	0.8	0	500	250e-9	on
Diode4	0.001	0	0.8	0	500	250e-9	on
Diode5	0.001	0	0.8	0	500	250e-9	on

Tabla 30. PMIOPort Block Properties

Name	Port	Side
VA	1	Left
VB	2	Left
VC	3	Left
VDC+	4	Right
VDC-	5	Right

Tabla 31. Terminator Block Properties

Name
Terminator
Terminator1
Terminator2

Terminator3
Terminator4
Terminator5

System - [MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE](#)

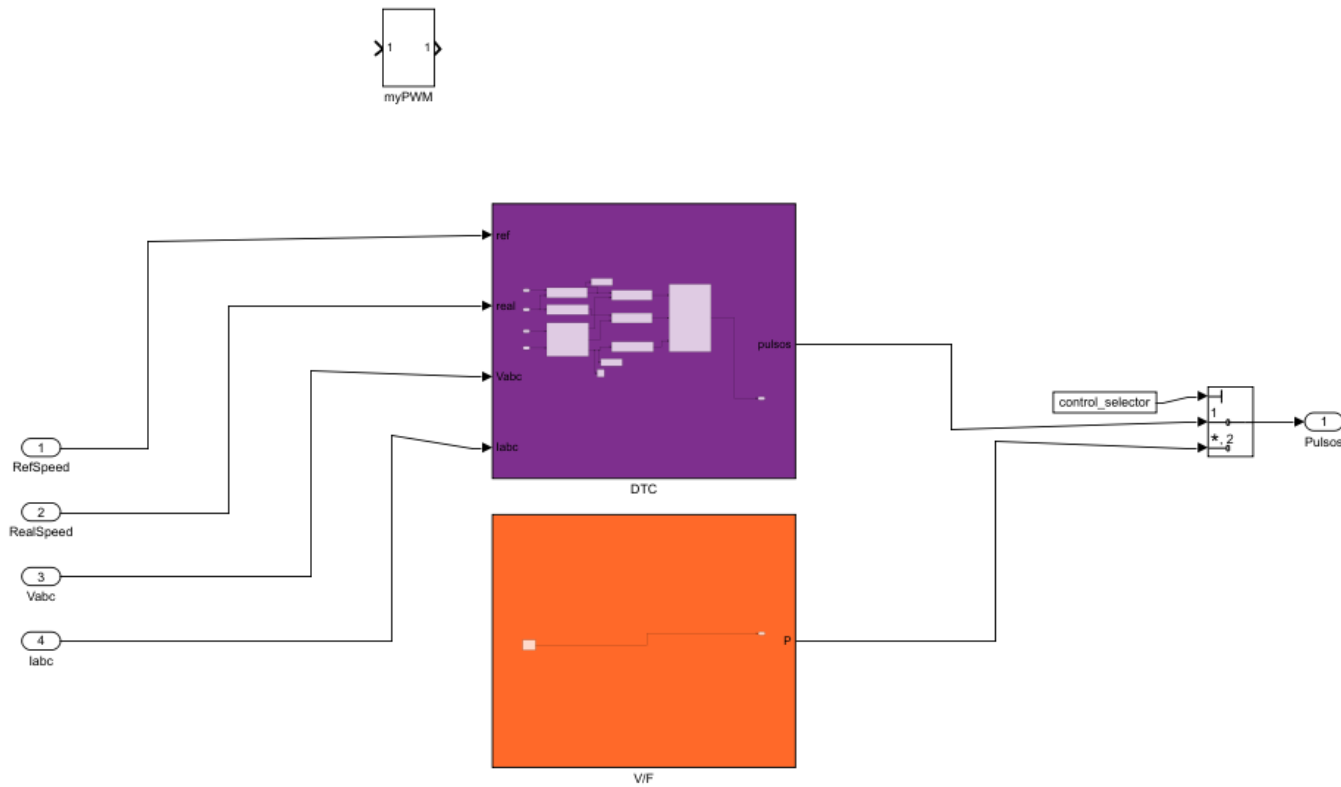


Tabla 32. Constant Block Properties

Name	Value	Out Data Type Str	Lock Scale	Sample Time	Frame Period
Constant	control_selector	Inherit: Inherit from 'Constant value'	off	inf	inf

Tabla 33. Inport Block Properties

Name	Port	Defined In Blk
Iabc	4	Kv
RealSpeed	2	rad/s to rpm
RefSpeed	1	Constant
Vabc	3	Kv1

Tabla 34. MultiPortSwitch Block Properties

Name	Data Port Order	Inputs	Data Port Indices	Data Port For Default	Diagnostic For Default	Input Same DT	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow	Allow Diff Input Sizes
Multiport Switch	One-based contiguous	2	{1,2,3}	Last data port	None	off	Inherit: Inherit via internal rule	off	Floor	off	off

Tabla 35. Outport Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Outport Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Unconnected	Output When Unconnected Value	Vector Params As 1D For Out When Unconnected	Used By Blk
Pulsos	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Demux

System - [MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC](#)

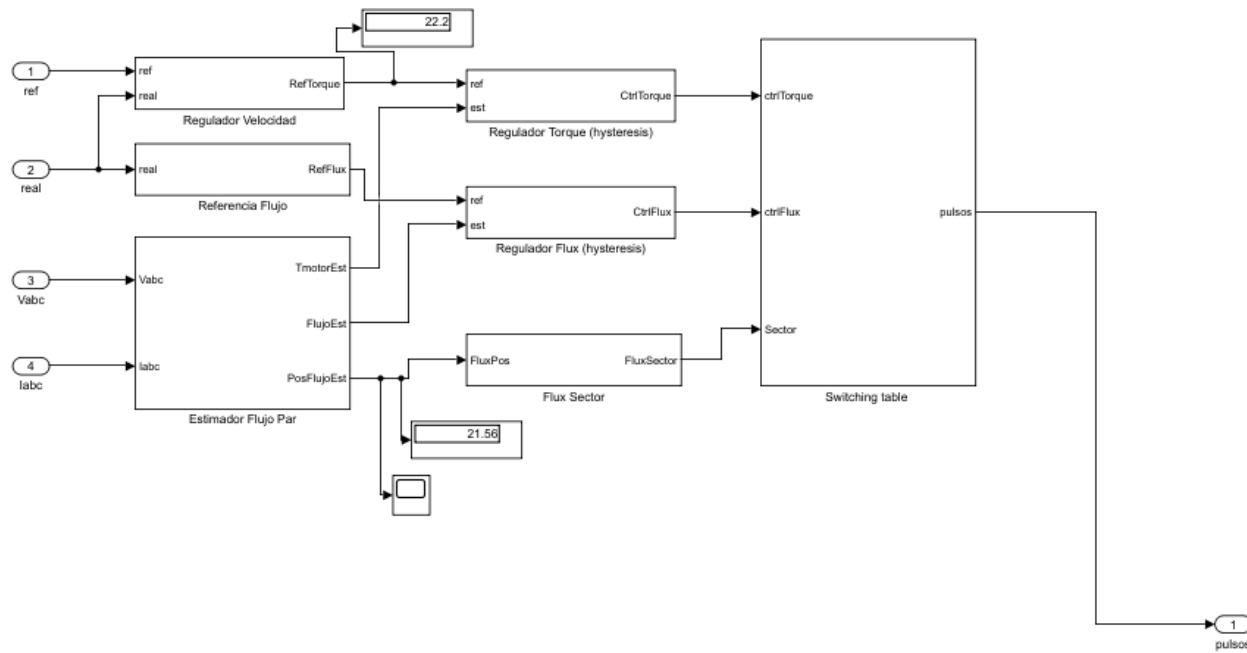


Tabla 36. Display Block Properties

Name	Format	Decimation	Floating
Display	short	1	off
Display1	short	1	off

Tabla 37. Inport Block Properties

Name	Port	Defined In Blk
Iabc	4	Kv

real	2	rad/s to rpm
ref	1	Constant
Vabc	3	Kv1

Tabla 38. Outport Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Outport Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Unconnected	Output When Unconnected Value	Vector Params As 1D For Out When Unconnected	Used By Blk
pulsos	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Multiport Switch

System - [MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC/Estimador Flujo Par](#)

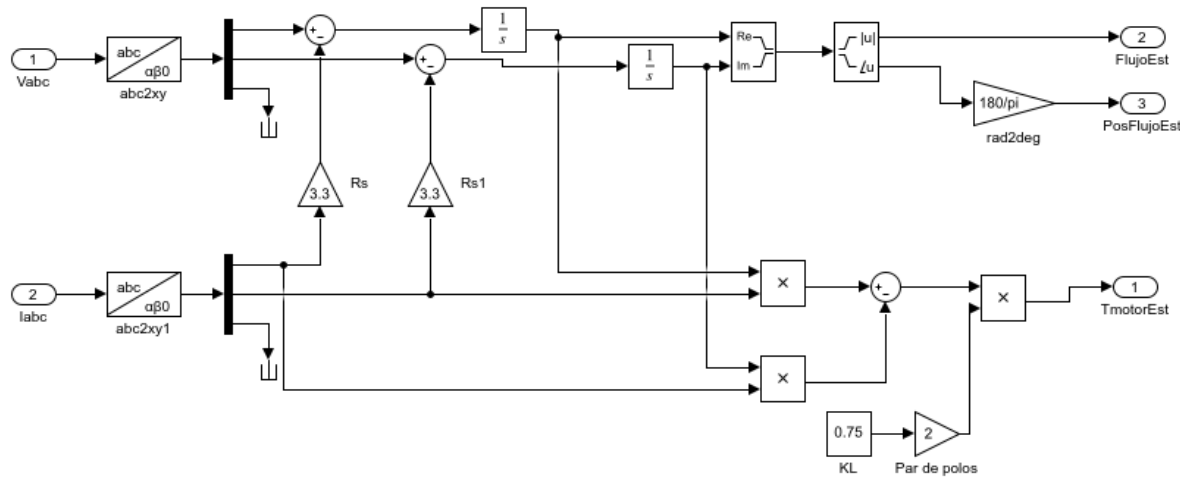


Tabla 39. ComplexToMagnitudeAngle Block Properties

Name	Output
Complex to Magnitude-Angle	Magnitude and angle

Tabla 40. Constant Block Properties

Name	Value	Out Data Type Str	Lock Scale	Sample Time	Frame Period
KL	0.75	Inherit: Inherit from 'Constant value'	off	inf	inf

Tabla 41. Demux Block Properties

Name	Outputs	Display Option	Bus Selection Mode
Demux	3	bar	off
Demux1	3	bar	off

Tabla 42. Gain Block Properties

Name	Gain	Multiplication	Param Data Type Str	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
Par de polos	2	Element-wise(K.*u)	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off
rad2deg	180/pi	Element-wise(K.*u)	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off
Rs	3.3	Element-wise(K.*u)	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off
Rs1	3.3	Element-wise(K.*u)	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off

Tabla 43. Inport Block Properties

Name	Port	Defined In Blk
labc	2	Kv
Vabc	1	Kv1

Tabla 44. Integrator Block Properties

Name	External Reset	Initial Condition Source	Wrap State	Wrapped State Upper Value	Wrapped State Lower Value	Absolute Tolerance	Zero Cross	Continuous State Attributes
Integrator	none	internal	off	pi	-pi	auto	on	"
Integrator1	none	internal	off	pi	-pi	auto	on	"

Tabla 45. Outport Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Outport Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Un Connected	Output When Unconnected Value	Vector Params As 1DFor Out When Unconnected	Used By Blk
FlujoEst	2	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Display, Sum
PosFlujoEst	3	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Relational Operator12, Relational Operator11, Relational Operator17, Relational Operator10, Relational Operator15, Relational Operator16, Relational Operator13, Relational Operator14, Relational Operator8,

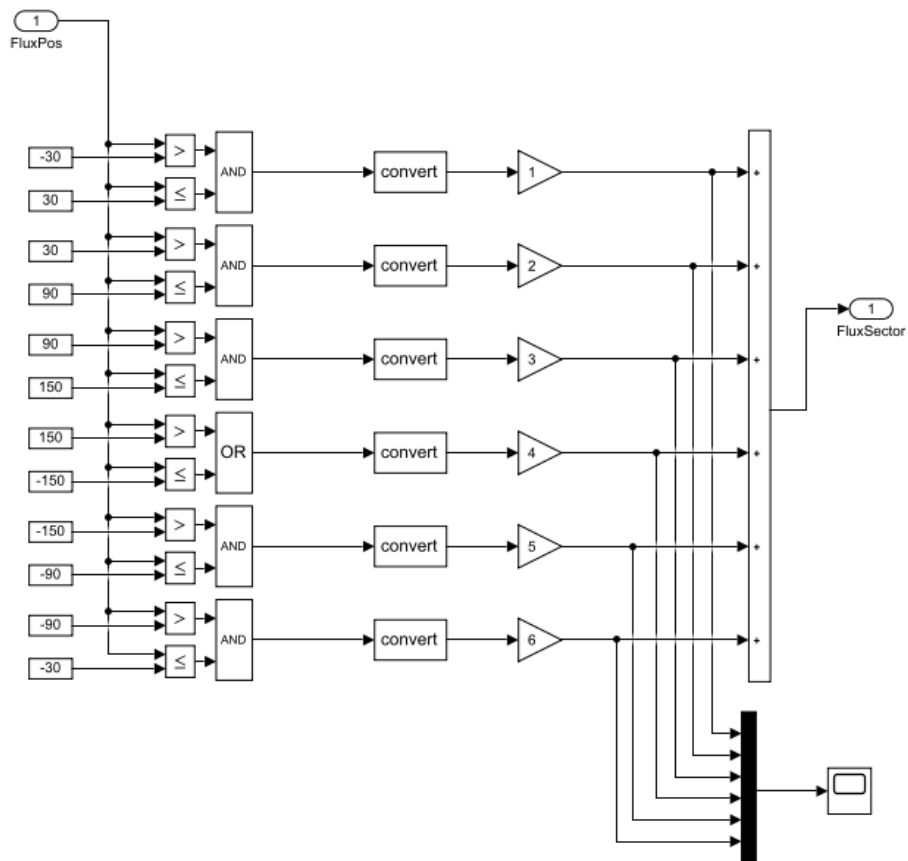


Tabla 51. Constant Block Properties

Name	Value	Out Data Type Str	Lock Scale	Sample Time	Frame Period
Constant	-30	Inherit: Inherit from 'Constant value'	off	inf	inf
Constant10	-90	Inherit: Inherit from 'Constant value'	off	inf	inf
Constant11	-30	Inherit: Inherit from 'Constant value'	off	inf	inf
Constant12	150	Inherit: Inherit from 'Constant value'	off	inf	inf
Constant13	150	Inherit: Inherit from 'Constant value'	off	inf	inf
Constant14	-150	Inherit: Inherit from 'Constant value'	off	inf	inf
Constant15	-150	Inherit: Inherit from 'Constant value'	off	inf	inf
Constant16	-90	Inherit: Inherit from 'Constant value'	off	inf	inf
		Inherit: Inherit from 'Constant			

Constant6	30	value'	off	inf	inf
Constant7	30	Inherit: Inherit from 'Constant value'	off	inf	inf
Constant8	90	Inherit: Inherit from 'Constant value'	off	inf	inf
Constant9	90	Inherit: Inherit from 'Constant value'	off	inf	inf

Tabla 52. DataTypeConversion Block Properties

Name	Out Data Type Str	Lock Scale	Convert Real World	Rnd Meth	Saturate On Integer Overflow
Data Type Conversion	Inherit: Inherit via back propagation	off	Real World Value (RWV)	Zero	on
Data Type Conversion1	Inherit: Inherit via back propagation	off	Real World Value (RWV)	Zero	on
Data Type Conversion2	Inherit: Inherit via back propagation	off	Real World Value (RWV)	Zero	on
Data Type Conversion3	Inherit: Inherit via back propagation	off	Real World Value (RWV)	Zero	on
Data Type Conversion4	Inherit: Inherit via back propagation	off	Real World Value (RWV)	Zero	on
Data Type Conversion5	Inherit: Inherit via back propagation	off	Real World Value (RWV)	Zero	on

Tabla 53. Gain Block Properties

Name	Gain	Multiplication	Param Data Type Str	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
Gain	1	Element-wise(K.*u)	Inherit: Same as input	Inherit: Same as input	off	Floor	on
Gain1	2	Element-wise(K.*u)	Inherit: Same as input	Inherit: Same as input	off	Floor	on
Gain2	3	Element-wise(K.*u)	Inherit: Same as input	Inherit: Same as input	off	Floor	on
Gain3	4	Element-wise(K.*u)	Inherit: Same as input	Inherit: Same as input	off	Floor	on
Gain4	5	Element-wise(K.*u)	Inherit: Same as input	Inherit: Same as input	off	Floor	on
Gain5	6	Element-wise(K.*u)	Inherit: Same as input	Inherit: Same as input	off	Floor	on

Tabla 54. Inport Block Properties

Name	Port	Defined In Blk
FluxPos	1	rad2deg

Tabla 55. Logic Block Properties

Name	Operator	Inputs	Icon Shape	All Ports Same DT	Out Data Type Str
Logical Operator	AND	2	rectangular	on	Inherit: Logical (see Configuration Parameters: Optimization)
Logical Operator1	AND	2	rectangular	on	Inherit: Logical (see Configuration Parameters: Optimization)
Logical Operator2	AND	2	rectangular	on	Inherit: Logical (see Configuration Parameters: Optimization)
Logical Operator3	OR	2	rectangular	on	Inherit: Logical (see Configuration Parameters: Optimization)

Logical Operator4	AND	2	rectangular	on	Inherit: Logical (see Configuration Parameters: Optimization)
Logical Operator5	AND	2	rectangular	on	Inherit: Logical (see Configuration Parameters: Optimization)

Tabla 56. Mux Block Properties

Name	Inputs	Display Option
Mux	6	bar

Tabla 57. Outport Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Outport Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Un Connected	Output When Unconnected Value	Vector Params As 1DFor Out When Unconnected	Used By Blk
FluxSector	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Display1 , Flux = -1 , Flux = 1

Tabla 58. RelationalOperator Block Properties

Name	Operator	Input Same DT	Out Data Type Str	Zero Cross	Rnd Meth
Relational Operator10	>	on	Inherit: Logical (see Configuration Parameters: Optimization)	off	Nearest
Relational Operator11	>	on	Inherit: Logical (see Configuration Parameters: Optimization)	off	Nearest
Relational Operator12	<=	on	Inherit: Logical (see Configuration Parameters: Optimization)	off	Nearest
Relational Operator13	<=	on	Inherit: Logical (see Configuration Parameters: Optimization)	off	Nearest
Relational Operator14	>	on	Inherit: Logical (see Configuration Parameters: Optimization)	off	Nearest
Relational Operator15	<=	on	Inherit: Logical (see Configuration Parameters: Optimization)	off	Nearest
Relational Operator16	>	on	Inherit: Logical (see Configuration Parameters: Optimization)	off	Nearest
Relational Operator17	<=	on	Inherit: Logical (see Configuration Parameters: Optimization)	off	Nearest
Relational Operator6	<=	on	Inherit: Logical (see Configuration Parameters: Optimization)	off	Nearest
Relational Operator7	>	on	Inherit: Logical (see Configuration Parameters: Optimization)	off	Nearest
Relational Operator8	<=	on	Inherit: Logical (see Configuration Parameters: Optimization)	off	Nearest
Relational Operator9	>	on	Inherit: Logical (see Configuration Parameters: Optimization)	off	Nearest

Tabla 59. Sum Block Properties

Name	Icon Shape	Inputs	Collapse Mode	Collapse Dim	Input Same DT	Accum Data	Out Data Type	Lock Scale	Rnd Meth	Saturate On
------	------------	--------	---------------	--------------	---------------	------------	---------------	------------	----------	-------------

						Type Str	Str			Integer Overflow
Add	rectangular	+++++	All dimensions	1	off	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off

System - [MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC/Referencia Flujo](#)

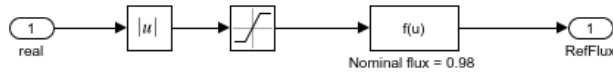


Tabla 60. Abs Block Properties

Name	Zero Cross	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
Abs	on	Inherit: Same as input	off	Floor	off

Tabla 61. Fcn Block Properties

Name	Expr
Nominal flux = 0.98	1500*0.98/u(1)

Tabla 62. Inport Block Properties

Name	Port	Defined In Blk
real	1	rad/s to rpm

Tabla 63. Outport Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Outport Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Un Connected	Output When Unconnected Value	Vector Params As 1DFor Out When Unconnected	Used By Blk
RefFlux	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Display1, Sum

Tabla 64. Saturate Block Properties

Name	Upper Limit	Lower Limit	Linearize As Gain	Zero Cross	Out Data Type Str	Lock Scale	Rnd Meth
Saturation1	inf	1500	on	off	Inherit: Same as input	off	Floor

System - [MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC/Regulador Flux \(hysteresis\)](#)

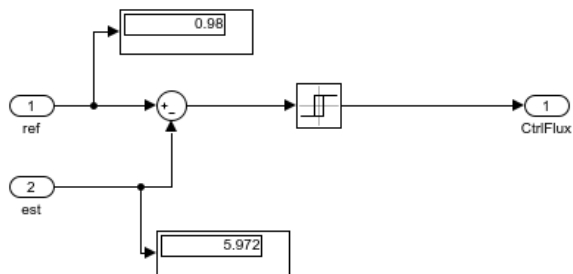


Tabla 65. Display Block Properties

Name	Format	Decimation	Floating
Display	short	1	off
Display1	short	1	off

Tabla 66. Inport Block Properties

Name	Port	Defined In Blk
est	2	Complex to Magnitude-Angle
ref	1	Nominal flux = 0.98

Tabla 67. Outport Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Outport Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Unconnected	Output When Unconnected Value	Vector Params As 1D For Out When Unconnected	Used By Blk
CtrlFlux	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Display, Multiport Switch1

Tabla 68. Relay Block Properties

Name	On Switch Value	Off Switch Value	On Output Value	Off Output Value	Out Data Type Str	Lock Scale	Input Processing	Zero Cross
Relay	0.98/20	-0.98/20	1	2	Inherit: All ports same datatype	off	Elements as channels (sample based)	on

Tabla 69. Sum Block Properties

Name	Icon Shape	Inputs	Collapse Mode	Collapse Dim	Input Same DT	Accum Data Type Str	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
Sum	round	+-	All dimensions	1	off	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off

System - [MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC/Regulador Torque \(hysteresis\)](#)

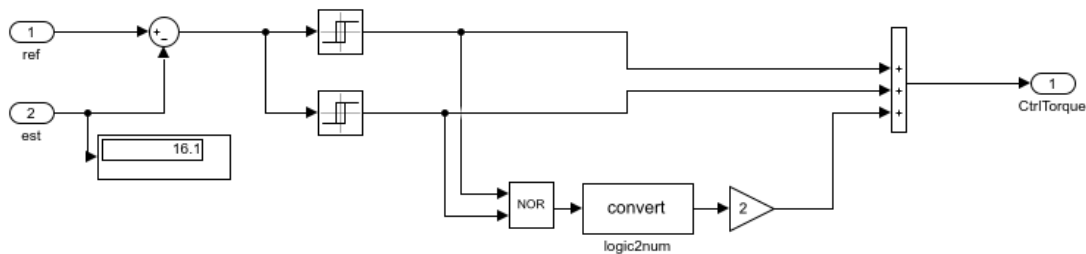


Tabla 70. DataTypeConversion Block Properties

Name	Out Data Type Str	Lock Scale	Convert Real World	Rnd Meth	Saturate On Integer Overflow
logic2num	Inherit: Inherit via back propagation	off	Real World Value (RWV)	Floor	off

Tabla 71. Display Block Properties

Name	Format	Decimation	Floating
Display	short	1	off

Tabla 72. Gain Block Properties

Name	Gain	Multiplication	Param Data Type Str	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
Gain	2	Element-wise(K.*u)	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off

Tabla 73. Inport Block Properties

Name	Port	Defined In Blk
est	2	Product2
ref	1	Saturation

Tabla 74. Logic Block Properties

Name	Operator	Inputs	Icon Shape	All Ports Same DT	Out Data Type Str
Logical Operator	NOR	2	rectangular	off	boolean

Tabla 75. Outport Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Outport Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Un Connected	Output When Unconnected Value	Vector Params As 1D For Out When Unconnected	Used By Blk
CtrlTorque	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Display2 , Flux = -1 , Flux = 1

Tabla 76. Relay Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Outport Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Un Connected	Output When Unconnected Value	Vector Params As 1D For Out When Unconnected	Used By Blk

Name	On Switch Value	Off Switch Value	On Output Value	Off Output Value	Out Data Type Str	Lock Scale	Input Processing	Zero Cross
Relay	22.2/2	0	1	0	Inherit: All ports same datatype	off	Elements as channels (sample based)	on
Relay1	0	-22.2/2	0	3	Inherit: All ports same datatype	off	Elements as channels (sample based)	on

Tabla 77. Sum Block Properties

Name	Icon Shape	Inputs	Collapse Mode	Collapse Dim	Input Same DT	Accum Data Type Str	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
Sum	round	+ -	All dimensions	1	off	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off
Sum1	rectangular	++++	All dimensions	1	off	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off

System - [MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC/Regulador Velocidad](#)

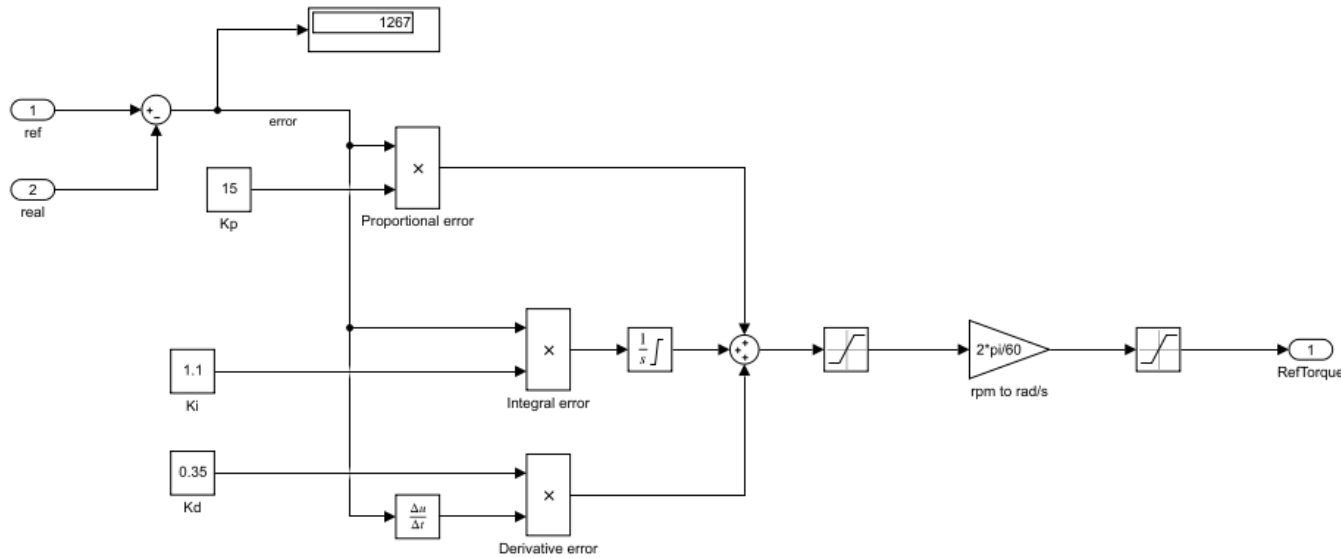


Tabla 78. Constant Block Properties

Name	Value	Out Data Type Str	Lock Scale	Sample Time	Frame Period
Kd	0.35	Inherit: Inherit from 'Constant value'	off	inf	inf
Ki	1.1	Inherit: Inherit from 'Constant value'	off	inf	inf
Kp	15	Inherit: Inherit from 'Constant value'	off	inf	inf

Tabla 79. Derivative Block Properties

Name	Coefficient In TFapproximation

Derivative	inf
------------	-----

Tabla 80. Display Block Properties

Name	Format	Decimation	Floating
Display	short	1	off

Tabla 81. Gain Block Properties

Name	Gain	Multiplication	Param Data Type Str	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
rpm to rad/s	2*pi/60	Element-wise(K.*u)	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off

Tabla 82. Inport Block Properties

Name	Port	Defined In Blk
real	2	rad/s to rpm
ref	1	Constant

Tabla 83. Integrator Block Properties

Name	External Reset	Initial Condition Source	Limit Output	Upper Saturation Limit	Lower Saturation Limit	Wrap State	Wrapped State Upper Value	Wrapped State Lower Value	Absolute Tolerance	Zero Cross	Continuous State Attributes
Integrator	none	internal	on	1500	-1500	off	pi	-pi	auto	on	"

Tabla 84. Outport Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Outport Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Unconnected	Output When Unconnected Value	Vector Params As IDFor Out When Unconnected	Used By Blk
RefTorque	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Sum, Display

Tabla 85. Product Block Properties

Name	Inputs	Multiplication	Collapse Mode	Collapse Dim	Input Same DT	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
Derivative error	2	Element-wise(.*)	All dimensions	1	off	Inherit: Inherit via internal rule	off	Floor	off
Integral error	2	Element-wise(.*)	All dimensions	1	off	Inherit: Inherit via internal rule	off	Floor	off
Proportional error	2	Element-wise(.*)	All dimensions	1	off	Inherit: Inherit via internal rule	off	Floor	off

Tabla 86. Saturate Block Properties

Name	Upper Limit	Lower Limit	Linearize As Gain	Zero Cross	Out Data Type Str	Lock Scale	Rnd Meth
Saturation	22.2	-22.2	on	off	Inherit: Same as input	off	Floor
Saturation1	1500	-1500	on	off	Inherit: Same as input	off	Floor

Tabla 87. Sum Block Properties

Name	Icon Shape	Inputs	Collapse Mode	Collapse Dim	Input Same DT	Accum Data Type Str	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
Sum1	round	+-	All dimensions	1	off	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off
Sum3	round	+++	All dimensions	1	off	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off

System - [MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/DTC/](#)Switching table

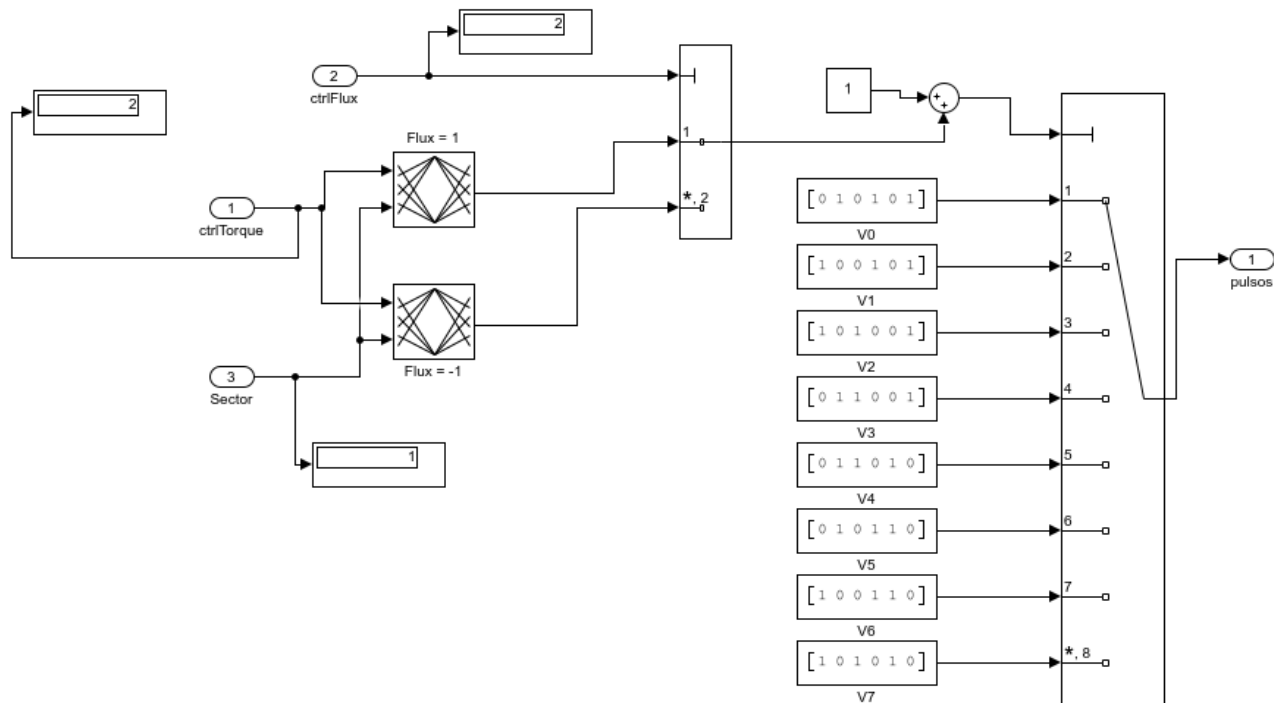


Tabla 88. Constant Block Properties

Name	Value	Out Data Type Str	Lock Scale	Sample Time	Frame Period
Constant	1	Inherit: Inherit from 'Constant value'	off	inf	inf
V0	[0 1 0 1 0 1]	Inherit: Inherit from 'Constant value'	off	inf	inf
V1	[1 0 0 1 0 1]	Inherit: Inherit from 'Constant value'	off	inf	inf
V2	[1 0 1 0 0 1]	Inherit: Inherit from 'Constant value'	off	inf	inf
V3	[0 1 1 0 0 1]	Inherit: Inherit from 'Constant value'	off	inf	inf

V4	[0 1 1 0 1 0]	Inherit: Inherit from 'Constant value'	off	inf	inf
V5	[0 1 0 1 1 0]	Inherit: Inherit from 'Constant value'	off	inf	inf
V6	[1 0 0 1 1 0]	Inherit: Inherit from 'Constant value'	off	inf	inf
V7	[1 0 1 0 1 0]	Inherit: Inherit from 'Constant value'	off	inf	inf

Tabla 89. Display Block Properties

Name	Format	Decimation	Floating
Display	short	1	off
Display1	short	1	off
Display2	short	1	off

Tabla 90. Inport Block Properties

Name	Port	Defined In Blk
ctrlFlux	2	Relay
ctrlTorque	1	Sum1
Sector	3	Add

Tabla 91. Lookup2D Block Properties

Name	Row Index	Column Index	Table	Look Up Meth	Input Same DT	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
Flux = -1	[1;2;3]	[1;2;3;4;5;6]	[3 4 5 6 1 2;7 0 7 0 7 0;5 6 1 2 3 4]	Interpolation-Extrapolation	off	Inherit: Same as first input	off	Floor	on
Flux = 1	[1;2;3]	[1;2;3;4;5;6]	[2 3 4 5 6 1;0 7 0 7 0 7;6 1 2 3 4 5]	Interpolation-Extrapolation	off	Inherit: Same as first input	off	Floor	on

Tabla 92. MultiPortSwitch Block Properties

Name	Data Port Order	Inputs	Data Port Indices	Data Port For Default	Diagnostic For Default	Input Same DT	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow	Allow Diff Input Sizes
Multiport Switch	One-based contiguous	8	{1,2,3}	Last data port	Error	off	Inherit: Inherit via internal rule	off	Floor	off	off
Multiport Switch1	One-based contiguous	2	{1,2,3}	Last data port	Error	off	Inherit: Inherit via internal rule	off	Floor	off	off

Tabla 93. Output Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Output Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Unconnected	Output When Unconnected Value	Vector Params As 1D For Out When Unconnected	Used By Blk
pulsos	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Multiport Switch

Tabla 94. Sum Block Properties

Name	Icon Shape	Inputs	Collapse Mode	Collapse Dim	Input Same DT	Accum Data Type Str	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
Sum	round	++	All dimensions	1	off	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off

System - [MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/V/F](#)

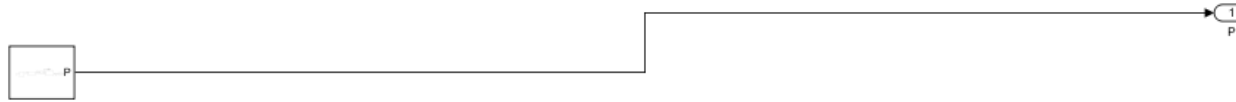


Tabla 95. Outport Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Output Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Unconnected	Output When Unconnected Value	Vector Params As 1D For Out When Unconnected	Used By Blk
P	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Multiport Switch

Tabla 96. PWM Generator (2-Level) Block Properties

Name	Modulator Type	Modulator Mode	Fc	Pc	Min Max	Sampling Technique	Modulating Signals	M	Freq	Phase	Ts	Show Carrier Output
PWM Generator (2-Level)	Three-phase bridge (6 pulses)	Unsynchronized	1000	90	[-1 1]	Natural	on	1	Vref*p/60	0	0	off

System - [MODELO/CONVERTIDOR/CONTROL AC/SOFTWARE/myPWM](#)

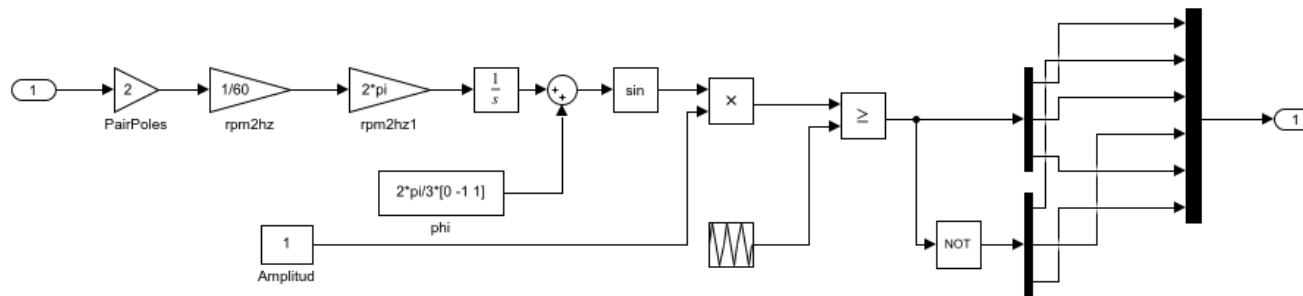


Tabla 97. Constant Block Properties

Name	Value	Out Data Type Str	Lock Scale	Sample Time	Frame Period
Amplitud	1	Inherit: Inherit from 'Constant value'	off	inf	inf
		Inherit: Inherit from 'Constant			

phi	$2\pi/3*[0 -1 1]$	value'	off	inf	inf
-----	-------------------	--------	-----	-----	-----

Tabla 98. Demux Block Properties

Name	Outputs	Display Option	Bus Selection Mode
Demux	3	bar	off
Demux1	3	bar	off

Tabla 99. Gain Block Properties

Name	Gain	Multiplication	Param Data Type Str	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
PairPoles	2	Element-wise(K.*u)	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off
rpm2hz	1/60	Element-wise(K.*u)	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off
rpm2hz1	2π	Element-wise(K.*u)	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off

Tabla 100. Inport Block Properties

Name	Port	Defined In Blk
In1	1	Unconnected

Tabla 101. Integrator Block Properties

Name	External Reset	Initial Condition Source	Wrap State	Wrapped State Upper Value	Wrapped State Lower Value	Absolute Tolerance	Zero Cross	Continuous State Attributes
Integrator	none	internal	off	pi	-pi	auto	on	"

Tabla 102. Logic Block Properties

Name	Operator	Inputs	Icon Shape	All Ports Same DT	Out Data Type Str
Logical Operator	NOT	2	rectangular	off	boolean

Tabla 103. Mux Block Properties

Name	Inputs	Display Option
Mux	6	bar

Tabla 104. Outport Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Outport Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Un Connected	Output When Unconnected Value	Vector Params As 1DFor Out When Unconnected	Used By Blk
Out1	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Unconnected

Tabla 105. Product Block Properties

Name	Inputs	Multiplication	Collapse Mode	Collapse Dim	Input Same DT	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On
------	--------	----------------	---------------	--------------	---------------	-------------------	------------	----------	-------------

									Integer Overflow
Product	2	Element-wise(.*)	All dimensions	1	off	Inherit: Inherit via internal rule	off	Floor	off

Tabla 106. RelationalOperator Block Properties

Name	Operator	Input Same DT	Out Data Type Str	Zero Cross	Rnd Meth
Relational Operator	>=	off	boolean	on	Nearest

Tabla 107. Repeating table Block Properties

Name	Rep seq t	Rep seq y
Repeating Sequence	[0 0.25 0.75 1]/2000	[0 1 -1 0]

Tabla 108. Sum Block Properties

Name	Icon Shape	Inputs	Collapse Mode	Collapse Dim	Input Same DT	Accum Data Type Str	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
Sum	round	++	All dimensions	1	off	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off

Tabla 109. Trigonometry Block Properties

Name	Operator	Approximation Method	Number Of Iterations	Output Signal Type
Trigonometric Function	sin	None	11	auto

System - [MODELO/CONVERTIDOR/CONTROL DC](#)

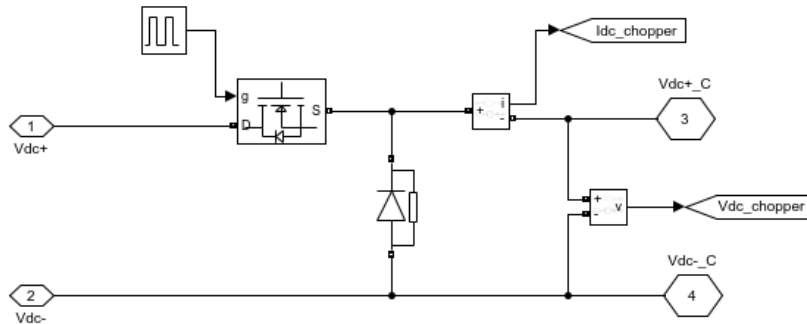


Tabla 110. Current Measurement Block Properties

Name
Current Measurement

Tabla 111. Diode Block Properties

Name	Ron	Lon	Vf	IC	Rs	Cs	Measurements
Diode	0.001	0	0.8	0	500	250e-9	off

Tabla 112. DiscretePulseGenerator Block Properties

Name	Pulse Type	Time Source	Amplitude	Period	Pulse Width	Phase Delay	Sample Time
Pulse Generator	Time based	Use simulation time	5	0.004	fix((Vref)/1750*100)	0	1

Tabla 113. Goto Block Properties

Name	Goto Tag	Icon Display	Tag Visibility	From Blk	From Blk Location	Used By Blk
Goto	Idc_chopper	Tag	global	From3	MODELO/CONVERTIDOR	Multiport Switch1
Goto1	Vdc_chopper	Tag	global	From	MODELO/CONVERTIDOR	Multiport Switch

Tabla 114. Mosfet Block Properties

Name	Ron	Lon	Rd	Vfd	IC	Rs	Cs	Measurements
Mosfet	0.1	0	0.01	0	0	1e5	inf	off

Tabla 115. PMIOPort Block Properties

Name	Port	Side
Vdc+	1	Left
Vdc+_C	3	Right
Vdc-	2	Left
Vdc-_C	4	Right

Tabla 116. Voltage Measurement Block Properties

Name
Voltage Measurement

System - [MODELO/MEDICIONES](#)

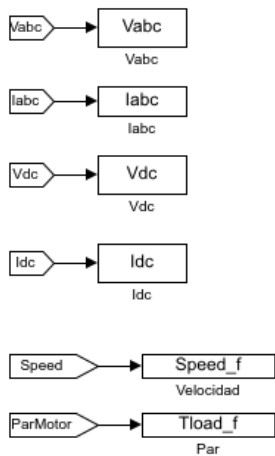


Tabla 117. From Block Properties

Name	Goto Tag	Icon Display	Goto Blk Name	Goto Blk Location	Defined In Blk
------	----------	--------------	---------------	-------------------	----------------

From1	Speed	Tag	Goto2	MODELO	rad/s to rpm
From3	ParMotor	Tag	Goto3	MODELO	Multiport Switch1
From5	Vabc	Tag	Goto3	MODELO/CONVERTIDOR	Kv1
From6	Idc	Tag	Goto6	MODELO/CONVERTIDOR	Multiport Switch1
From7	Vdc	Tag	Goto	MODELO/CONVERTIDOR	Multiport Switch
From8	Iabc	Tag	Goto2	MODELO/CONVERTIDOR	Kv

Tabla 118. ToWorkspace Block Properties

Name	Variable Name	Max Data Points	Decimation	Save Format	Save 2DSignal	Fixpt As Fi
Iabc	Iabc	inf	1	Timeseries	3-D array (concatenate along third dimension)	on
Idc	Idc	inf	1	Timeseries	3-D array (concatenate along third dimension)	on
Par	Tload_f	inf	1	Timeseries	3-D array (concatenate along third dimension)	on
Vabc	Vabc	inf	1	Timeseries	3-D array (concatenate along third dimension)	on
Vdc	Vdc	inf	1	Timeseries	3-D array (concatenate along third dimension)	on
Velocidad	Speed_f	inf	1	Timeseries	3-D array (concatenate along third dimension)	on

System - [MODELO/MOTOR](#)

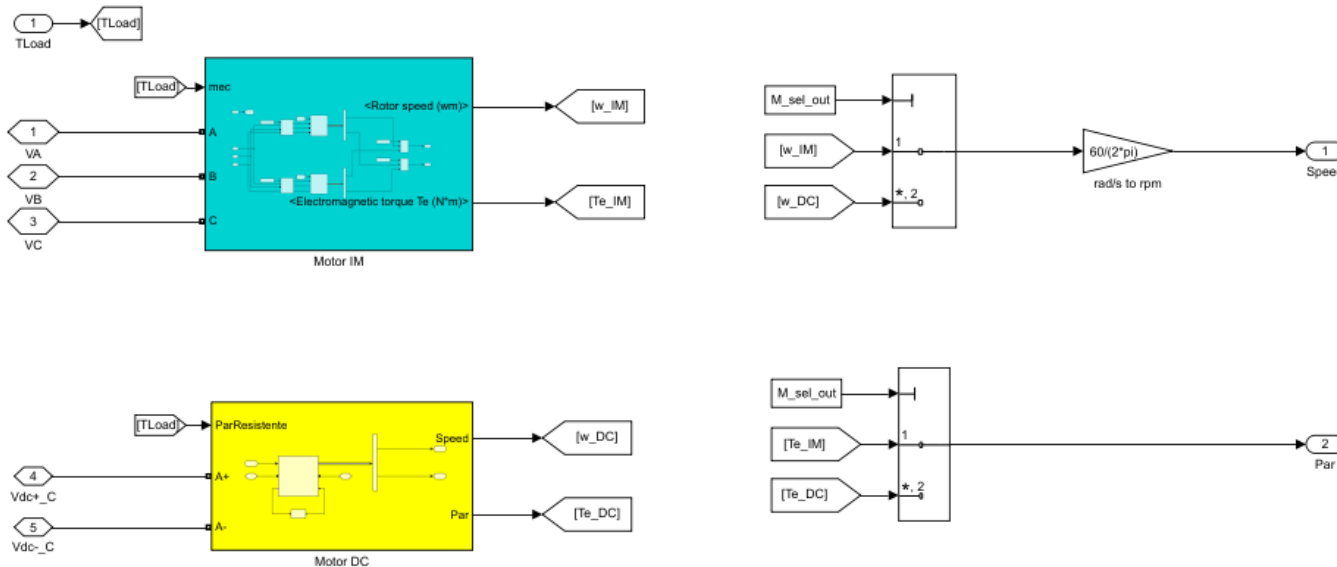


Tabla 119. Constant Block Properties

--	--	--	--	--	--

Name	Value	Out Data Type Str	Lock Scale	Sample Time	Frame Period
Constant2	M_sel_out	Inherit: Inherit from 'Constant value'	off	inf	inf
Constant3	M_sel_out	Inherit: Inherit from 'Constant value'	off	inf	inf

Tabla 120. From Block Properties

Name	Goto Tag	Icon Display	Goto Blk Name	Goto Blk Location	Defined In Blk
From1	TLoad	Tag	Goto	MODELO/MOTOR	Multiport Switch
From3	w_IM	Tag	Goto2	MODELO/MOTOR	Multiport Switch
From5	Te_IM	Tag	Goto4	MODELO/MOTOR	Multiport Switch1
From6	TLoad	Tag	Goto	MODELO/MOTOR	Multiport Switch
From7	w_DC	Tag	Goto5	MODELO/MOTOR	Demux
From8	Te_DC	Tag	Goto6	MODELO/MOTOR	Demux

Tabla 121. Gain Block Properties

Name	Gain	Multiplication	Param Data Type Str	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
rad/s to rpm	60/(2*pi)	Element-wise(K.*u)	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off

Tabla 122. Goto Block Properties

Name	Goto Tag	Icon Display	Tag Visibility	From Blk	From Blk Location	Used By Blk
Goto	TLoad	Tag	local	From6 , From1	MODELO/MOTOR , MODELO/MOTOR	Sum, 1_Tb2, 1_Tb2
Goto2	w_IM	Tag	local	From3	MODELO/MOTOR	Multiport Switch
Goto4	Te_IM	Tag	local	From5	MODELO/MOTOR	Multiport Switch1
Goto5	w_DC	Tag	local	From7	MODELO/MOTOR	Multiport Switch
Goto6	Te_DC	Tag	local	From8	MODELO/MOTOR	Multiport Switch1

Tabla 123. Inport Block Properties

Name	Port	Defined In Blk
TLoad	1	Multiport Switch

Tabla 124. MultiPortSwitch Block Properties

Name	Data Port Order	Inputs	Data Port Indices	Data Port For Default	Diagnostic For Default	Input Same DT	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow	Allow Diff Input Sizes
Multiport Switch	One-based contiguous	2	{1,2,3}	Last data port	Error	off	Inherit: Inherit via internal rule	off	Floor	off	off
Multiport Switch1	One-based contiguous	2	{1,2,3}	Last data port	Error	off	Inherit: Inherit via internal rule	off	Floor	off	off

Tabla 125. Output Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Output Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Unconnected	Output When Unconnected Value	Vector Params As 1D For Out When Unconnected	Used By Blk
Par	2	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Par
Speed	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Math Function1 , Math Function , Velocidad , Sum1 , Abs

Tabla 126. PMIOPort Block Properties

Name	Port	Side
VA	1	Left
VB	2	Left
VC	3	Left
Vdc+_C	4	Left
Vdc-_C	5	Left

System - [MODELO/MOTOR/Motor DC](#)

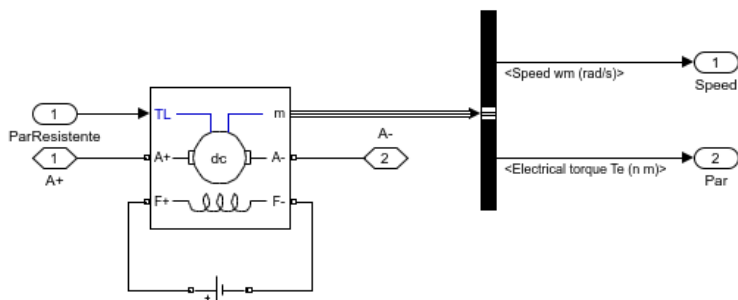


Tabla 127. BusSelector Block Properties

Name	Output Signals	Output As Bus	Input Signals
Bus Selector	Speed wm (rad/s), Electrical torque Te (n m)	off	Speed wm (rad/s) Armature current ia (A) Field current if (A) Electrical torque Te (n m)

Tabla 128. DC Voltage Source Block Properties

Name	Amplitude	Measurements
DC Voltage Source	300	None

Tabla 129. DC machine Block Properties

Name	Preset Model	Mechanical Load	Field Type	Measurement Bus	RLa	RLf	Laf	J	Bm	Tf	W0	Ts Block
DC Machine	04: 10HP 240V 1750RPM Field:300V	Torque TL	Wound	off	[1.086 0.01216]	[180 71.47]	0.6458	0.04251	0.003406	1.046	1	-1

Tabla 130. Inport Block Properties

Name	Port	Defined In Blk
ParResistente	1	Multiport Switch

Tabla 131. Output Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Output Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Unconnected	Output When Unconnected Value	Vector Params As 1DFor Out When Unconnected	Used By Blk
Par	2	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Multiport Switch1
Speed	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Multiport Switch

Tabla 132. PMIOPort Block Properties

Name	Port	Side
A+	1	Left
A-	2	Left

System - [MODELO/MOTOR/Motor IM](#)

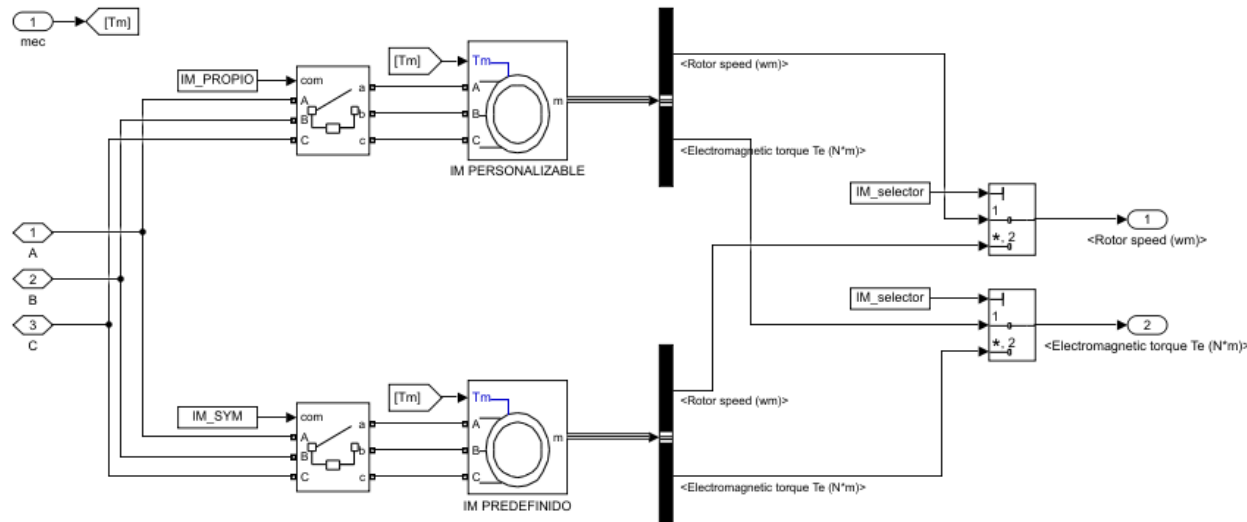


Tabla 133. Asynchronous Machine Block Properties

Name	Rotor Type	Preset Model	Mechanical Load	Reference Frame	Measurement Bus	Nominal Parameters	Stator	Rotor	Lm	Mechanical	Initial Conditions	Simulate Saturation	Saturation	Ts Block	Iterative Model	Iterative Discrete Model	Pmec
IM PERSONALIZABLE	Squirrel-cage	No	Torque Tm	Rotor	off	[Pn VLL Fn]	[Rs Ls]	[Rr Lr]	Lm	[J B p]	[1 0 0 0 0 0 0]	off	[14.03593122, 27.81365428, 53.79336849, 72.68890987, 97.98006896, 148.6815601, 215.7428561, 302.9841135, 428.7778367 ; 230, 322, 414, 460, 506, 552, 598, 644, 690]	-1	Trapezoidal non iterative	Trapezoidal non iterative	1.492e+006
IM PREDEFINIDO	Squirrel-cage	15: 5.4 HP (4KW) 400 V 50Hz 1430 RPM	Torque Tm	Rotor	off	[4000 400 50]	[1.405 0.005839]	[1.395 0.005839]	0.1722	[0.0131 0.002985 2]	[1 0 0 0 0 0 0]	off	[14.03593122, 27.81365428, 53.79336849, 72.68890987, 97.98006896, 148.6815601, 215.7428561, 302.9841135, 428.7778367 ; 230, 322, 414, 460, 506, 552, 598, 644, 690]	-1	Trapezoidal non iterative	Trapezoidal non iterative	1.492e+006

Tabla 134. BusSelector Block Properties

Name	Output Signals	Output As Bus	Input Signals
Bus Selector	Mechanical.Rotor speed (wm),Mechanical.Electromagnetic torque Te (N*m)	off	{ Rotor measurements , Rotor current ir_a (A) Rotor current ir_b (A) Rotor current ir_c (A) Rotor current iq (A) Rotor current id (A) Rotor flux phir_q (V s) Rotor flux phir_d (V s) Rotor voltage Vr_q (V) Rotor voltage Vr_d (V) } { Stator measurements , Stator current is_a (A) Stator current is_b (A) Stator current is_c (A) Stator current is_q (A) Stator current is_d (A) Stator flux phis_q (V s) Stator flux phis_d (V s) Stator voltage vs_q (V) Stator voltage vs_d (V) } Lm (H) { Mechanical , Rotor speed (wm) Electromagnetic torque Te (N*m) Rotor angle thetam (rad) }
Bus Selector1	Mechanical.Rotor speed	off	{ Rotor measurements , Rotor current ir_a (A) Rotor current ir_b (A) Rotor current ir_c (A) Rotor current iq (A) Rotor current id (A) Rotor flux phir_q (V s) Rotor flux phir_d (V s) Rotor voltage Vr_q (V) Rotor voltage Vr_d (V) } { Stator measurements , Stator current is_a (A) Stator current is_b (A) Stator current is_c (A) Stator

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Output Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Unconnected	Output When Unconnected Value	Params As ID For Out When Unconnected	Used By Blk
<Electromagnetic torque Te (N*m)>	2	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Multiport Switch1
<Rotor speed (wm)>	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Multiport Switch

Tabla 141. PMIOPort Block Properties

Name	Port	Side
A	1	Left
B	2	Left
C	3	Left

Tabla 142. Three-Phase Breaker Block Properties

Name	Initial State	Switch A	Switch B	Switch C	Switch Times	External	Breaker Resistance	Snubber Resistance	Snubber Capacitance	Measurements
Three-Phase Breaker	open	on	on	on	[1/60 5/60]	on	0	1e3	inf	None
Three-Phase Breaker1	open	on	on	on	[1/60 5/60]	on	0.01	1e3	inf	None

System - [MODELO](#)/PROCESO

Cinta transportadora

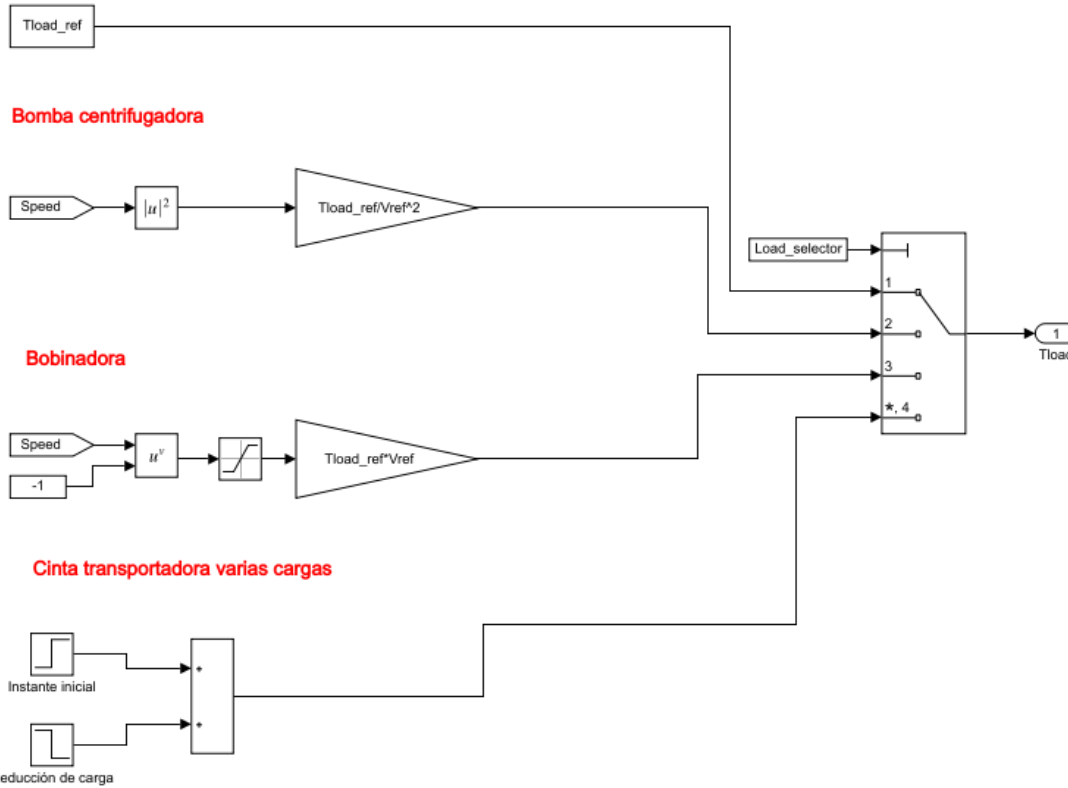


Tabla 143. Constant Block Properties

Name	Value	Out Data Type Str	Lock Scale	Sample Time	Frame Period
Constant	Tload_ref	Inherit: Inherit from 'Constant value'	off	inf	inf
Constant1	-1	Inherit: Inherit from 'Constant value'	off	inf	inf
Constant2	Load_selector	Inherit: Inherit from 'Constant value'	off	inf	inf

Tabla 144. From Block Properties

Name	Goto Tag	Icon Display	Goto Blk Name	Goto Blk Location	Defined In Blk
From1	Speed	Tag	Goto2	MODELO	rad/s to rpm
From2	Speed	Tag	Goto2	MODELO	rad/s to rpm

Tabla 145. Gain Block Properties

Name	Gain	Multiplication	Param Data Type Str	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow

Gain	Tload_ref/Vref^2	Element-wise(K.*u)	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off
Gain1	Tload_ref*Vref	Element-wise(K.*u)	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off

Tabla 146. Math Block Properties

Name	Operator	Signed Power	Output Signal Type	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow	Intermediate Results Data Type Str	Algorithm Type	Iterations
Math Function	magnitude^2	off	auto	Inherit: Same as first input	off	Floor	on	Inherit: Inherit via internal rule	Newton-Raphson	3
Math Function1	pow	off	auto	Inherit: Same as first input	off	Floor	on	Inherit: Inherit via internal rule	Newton-Raphson	3

Tabla 147. MultiPortSwitch Block Properties

Name	Data Port Order	Inputs	Data Port Indices	Data Port For Default	Diagnostic For Default	Input Same DT	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow	Allow Diff Input Sizes
Multiport Switch	One-based contiguous	4	{1,2,3}	Last data port	Error	off	Inherit: Inherit via internal rule	off	Floor	off	off

Tabla 148. Output Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Outputport Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Unconnected	Output When Unconnected Value	Vector Params As 1DFor Out When Unconnected	Used By Blk
Tload	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Sum, 1_Tb2, 1_Tb2

Tabla 149. Saturate Block Properties

Name	Upper Limit	Lower Limit	Linearize As Gain	Zero Cross	Out Data Type Str	Lock Scale	Rnd Meth
Saturation	0.01	0	on	on	Inherit: Same as input	off	Floor

Tabla 150. Step Block Properties

Name	Time	Before	After	Out Data Type Str	Sample Time	Zero Cross
Instante inicial	0	0	Tload_ref	double	0	on
Reducción de carga	0.2	0	-Tload_ref/2	double	0	on

Tabla 151. Sum Block Properties

Name	Icon Shape	Inputs	Collapse Mode	Collapse Dim	Input Same DT	Accum Data Type Str	Out Data Type Str	Lock Scale	Rnd Meth	Saturate On Integer Overflow
Add	rectangular	++	All dimensions	1	off	Inherit: Inherit via internal rule	Inherit: Inherit via internal rule	off	Floor	off

System - [MODELO/RED](#)

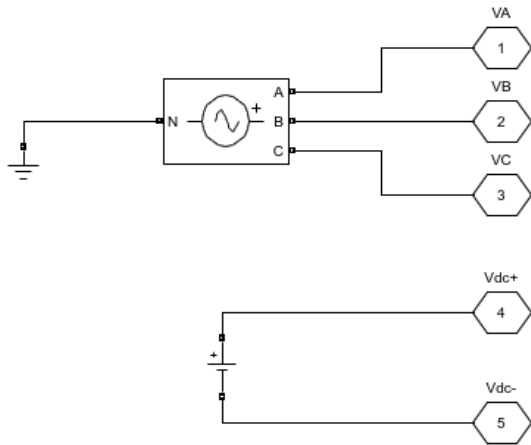


Tabla 152. DC Voltage Source Block Properties

Name	Amplitude	Measurements
DC Voltage Source	Vdc	None

Tabla 153. Ground Block Properties

Name	Physical Domain	Sub Class Name	Left Port Type	Right Port Type
Ground	powersysdomain	unknown	p1	p1

Tabla 154. PMIOPort Block Properties

Name	Port	Side
VA	1	Right
VB	2	Right
VC	3	Right
Vdc+	4	Right
Vdc-	5	Right

Tabla 155. Three-Phase Programmable Voltage Source Block Properties

Name	Positive Sequence	Variation Entity	Harmonic Generation	Bus Type
Three-Phase Programmable Voltage Source	[Vac 0 fac]	None	off	swing

System - [MODELO/REFERENCIA](#)

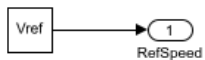


Tabla 156. Constant Block Properties

Name	Value	Out Data Type Str	Lock Scale	Sample Time	Frame Period
Constant	Vref	Inherit: Inherit from 'Constant value'	off	inf	inf

Tabla 157. Outport Block Properties

Name	Port	Storage Class	Icon Display	Lock Scale	Unit	Var Size Sig	Signal Type	Ensure Outport Is Virtual	Source Of Initial Output Value	Output When Disabled	Must Resolve To Signal Object	Output When Unconnected	Output When Unconnected Value	Vector Params As 1DFor Out When Unconnected	Used By Blk
RefSpeed	1	Auto	Port number	off	inherit	Inherit	auto	off	Dialog	held	off	off	0	on	Sum1

Appendix

Tabla 158. Block Type Count

BlockType	Count	Block Names
PMIOPort	55	VDC+ , VDC+1 , VDC- , VDC-1 , VA , VB , VC1 , VDC+ , VDC- , VA , VB , VC , VDC+ , VDC- , VA , VA1 , VB , VB1 , VC , VC1 , VA , VA1 , VB , VB1 , VC , VC1 , Vdc+ , Vdc+_C , Vdc- , Vdc-_C , VA , VA1 , VB , VB1 , VC , VC1 , Vdc+ , Vdc+_C , Vdc- , Vdc-_C , A+ , A- , A , B , C , VA , VB , VC , Vdc+_C , Vdc-_C , VA , VB , VC , Vdc+ , Vdc-
Constant	40	Constant , KL , Constant , Constant10 , Constant11 , Constant12 , Constant13 , Constant14 , Constant15 , Constant16 , Constant6 , Constant7 , Constant8 , Constant9 , Kd , Ki , Kp , Constant , V0 , V1 , V2 , V3 , V4 , V5 , V6 , V7 , Amplitud , phi , Constant , Constant1 , Constant2 , Constant3 , Constant , Constant1 , Constant2 , Constant3 , Constant , Constant1 , Constant2 , Constant
Outport	33	IDC , VDC , Idc , Vabc , Vdc , fout , iabc , Idc , FlujoEst , PosFlujoEst , TmotorEst , FluxSector , RefFlux , CtrlFlux , CtrlTorque , RefTorque , pulsos , pulsos , Pulsos , P , Out1 , Vabc , Vdc , fout , iabc , Par , Speed , <Electromagnetic torque Te (N*m)> , <Rotor speed (wm)> , Par , Speed , Tload , RefSpeed
Inport	31	PULSOS , PULSOS , RealSpeed , RefSpeed , iabc , Vabc , FluxPos , iabc , real , est , ref , est , ref , real , ref , Sector , ctrlFlux , ctrlTorque , Vabc , real , ref , iabc , RealSpeed , RefSpeed , Vabc , In1 , RealSpeed , RefSpeed , ParResistente , mec , TLoad
From	29	From , From1 , From2 , From3 , From4 , From5 , From , From1 , From2 , From3 , From , From2 , From4 , From1 , From3 , From5 , From6 , From7 , From8 , From1 , From3 , From5 , From6 , From7 , From8 , From , From1 , From1 , From2
SubSystem	25	CONVERTIDOR , CONTROL AC , HARDWARE , BUS DC , INVERSOR , RECTIFICADOR , SOFTWARE , DTC , Estimador Flujo Par , Flux Sector , Referencia Flujo , Regulador Flux (hysteresis) , Regulador Torque (hysteresis) , Regulador Velocidad , Switching table , V/E , myPWM , CONTROL DC , MEDICIONES , MOTOR , Motor DC , Motor IM , PROCESO , RED , REFERENCIA
Goto	25	Goto , Goto1 , Goto2 , Goto3 , Goto4 , Goto5 , Goto , Goto1 , Goto , Goto1 , Goto2 , Goto3 , Goto4 , Goto5 , Goto6 , Goto1 , Goto2 , Goto3 , Goto4 , Goto , Goto2 , Goto4 , Goto5 , Goto6 , Goto
Gain	18	Par de polos , Rs , Rs1 , rad2deg , Gain , Gain1 , Gain2 , Gain3 , Gain4 , Gain5 , Gain , rpm to rad/s , PairPoles , rpm2hz , rpm2hz1 , rad/s to rpm , Gain , Gain1
Terminator	14	Terminator , Terminator1 , Terminator2 , Terminator3 , Terminator4 , Terminator5 , Terminator , Terminator1 , Terminator2 , Terminator3 , Terminator4 , Terminator5 , Terminator , Terminator1
RelationalOperator	13	Relational Operator10 , Relational Operator11 , Relational Operator12 , Relational Operator13 , Relational Operator14 , Relational Operator15 , Relational Operator16 , Relational Operator17 , Relational Operator6 , Relational Operator7 , Relational Operator8 , Relational Operator9 , Relational Operator
Sum	12	Sum , Sum1 , Sum2 , Add , Sum , Sum , Sum1 , Sum1 , Sum3 , Sum , Sum , Add
MultiPortSwitch	10	Multiport Switch , Multiport Switch1 , Multiport Switch , Multiport Switch , Multiport Switch1 , Multiport Switch , Multiport Switch1 , Multiport Switch , Multiport Switch1 , Multiport Switch
Display	9	Display , Display1 , Display , Display1 , Display , Display , Display , Display1 , Display2
Logic	8	Logical Operator , Logical Operator1 , Logical Operator2 , Logical Operator3 , Logical Operator4 , Logical Operator5 , Logical Operator , Logical Operator

Product	7	Product , Product1 , Product2 , Derivative error , Integral error , Proportional error , Product
Diode (m)	7	Diode , Diode1 , Diode2 , Diode3 , Diode4 , Diode5 , Diode
DataTypeConversion	7	Data Type Conversion , Data Type Conversion1 , Data Type Conversion2 , Data Type Conversion3 , Data Type Conversion4 , Data Type Conversion5 , logic2num
ToWorkspace	6	Iabc , Idc , Par , Yabc , Ydc , Velocidad
IGBT/Diode (m)	6	IGBT/Diode , IGBT/Diode1 , IGBT/Diode2 , IGBT/Diode3 , IGBT/Diode4 , IGBT/Diode5
Demux	5	Demux , Demux , Demux1 , Demux , Demux1
Saturate	4	Saturation1 , Saturation , Saturation1 , Saturation
Integrator	4	Integrator , Integrator1 , Integrator , Integrator
Relay	3	Relay , Relay , Relay1
BusSelector	3	Bus Selector , Bus Selector , Bus Selector1
abc to Alpha-Beta-Zero Transformation (m)	2	abc2xy , abc2xy1
Voltage Measurement (m)	2	Voltage Measurement , Voltage Measurement
Three-Phase Breaker (m)	2	Three-Phase Breaker , Three-Phase Breaker1
Step	2	Instante inicial , Reducción de carga
Scope	2	Scope , Scope
Mux	2	Mux , Mux
Math	2	Math Function , Math Function1
Lookup2D	2	Flux = -1 , Flux = 1
DC Voltage Source (m)	2	DC Voltage Source , DC Voltage Source
Current Measurement (m)	2	Current Measurement , Current Measurement
Asynchronous Machine (m)	2	IM PERSONALIZABLE , IM PREDEFINIDO
Trigonometry	1	Trigonometric Function
Three-Phase VI Measurement (m)	1	Three-Phase V-I Measurement
Three-Phase Programmable Voltage Source (m)	1	Three-Phase Programmable Voltage Source
Series RLC Branch (m)	1	Series RLC Branch
Repeating table (m)	1	Repeating Sequence
RealImagToComplex	1	Real-Imag to Complex
PWM Generator (2-Level) (m)	1	PWM Generator (2-Level)
PSB option menu block (m)	1	powergui
Mosfet (m)	1	Mosfet
Ground (m)	1	Ground
Fcn	1	Nominal flux = 0.98
DiscretePulseGenerator	1	Pulse Generator
Derivative	1	Derivative
DC machine (m)	1	DC Machine
ComplexToMagnitudeAngle	1	Complex to Magnitude-Angle
Abs	1	Abs

Tabla 159. Model Variables

Variable Name	Parent Blocks	Calling character vector	Value
B	IM PERSONALIZABLE	B	0.0030
Fn	IM PERSONALIZABLE IM PERSONALIZABLE	Fn [Pn VLL Fn]	50
IM_PROPIO	Constant	IM_PROPIO	0
		IM_SYM	

IM_SYM	Constant1		1
IM_selector	Constant2 Constant3	IM_selector IM_selector	2
J	IM PERSONALIZABLE	J	0.0131
Lm	IM PERSONALIZABLE	Lm	0.2167
Load_selector	Constant2	Load_selector	1
Lr	IM PERSONALIZABLE	Lr	0.0138
Ls	IM PERSONALIZABLE IM PERSONALIZABLE	Ls [Rs Ls]	0.0138
M_sel_out	Constant2 Constant3	M_sel_out M_sel_out	1
Pn	IM PERSONALIZABLE IM PERSONALIZABLE	Pn [Pn VLL Fn]	2200
Rr	IM PERSONALIZABLE	Rr	2.9050
Rs	IM PERSONALIZABLE IM PERSONALIZABLE	Rs [Rs Ls]	3.3000
Tload_ref	Reducción de carga Gain1 Gain Constant Gain Gain1 Instante inicial Reducción de carga	-Tload_ref/2 Tload_ref*Vref Tload_ref/Vref^2 Tload_ref Tload_ref Tload_ref Tload_ref Tload_ref	7.4468
VLL	IM PERSONALIZABLE IM PERSONALIZABLE	VLL [Pn VLL Fn]	400
Vac	Three-Phase Programmable Voltage Source Three-Phase Programmable Voltage Source	Vac [Vac 0 fac]	400
Vdc	DC Voltage Source	Vdc	0
Vref	Gain1 Gain PWM Generator (2-Level) Pulse Generator Gain Gain1 Constant Pulse Generator	Tload_ref*Vref Tload_ref/Vref^2 Vref Vref Vref Vref Vref fix((Vref)/1750*100)	1200
contr_ac_dc	Constant Constant1	contr_ac_dc contr_ac_dc	2
control_selector	Constant	control_selector	1
fac	Three-Phase Programmable Voltage Source Three-Phase Programmable Voltage Source	[Vac 0 fac] fac	50
	PWM Generator (2-Level)	p	

p	IM PERSONALIZABLE	p	2
---	-----------------------------------	---	---

Tabla 160. Model Functions

Function Name	Parent Blocks	Calling character vector
fix	Pulse Generator	fix((Vref)/1750*100)
pi	Integrator Integrator1 Integrator Integrator rad2deg phi rpm to rad/s rpm2hz1 rad/s to rpm Integrator Integrator1 Integrator Integrator	- π - π - π - π $180/\pi$ $2*\pi/3*[0 \ -1 \ 1]$ $2*\pi/60$ $2*\pi$ $60/(2*\pi)$ π π π π



ANEXO 2 – CÓDIGO INTERFAZ USUARIO

Digitalización de un banco de ensayos y comprobación experimental

```
classdef INTERF_GENERAL < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        BancodeensayosUIFigure          matlab.ui.Figure
        REDLabel                          matlab.ui.control.Label
        ICOac                              matlab.ui.control.Image
        ICODc                              matlab.ui.control.Image
        ICovf                              matlab.ui.control.Label
        Separador1                        matlab.ui.control.Image
        CARGANmSliderLabel                matlab.ui.control.Label
        CARGANmSlider                      matlab.ui.control.Slider
        VelocidaddeconsignarpmEditFieldLabel matlab.ui.control.Label
        VelocidaddeconsignarpmEditField  matlab.ui.control.NumericEditField
        ParametrosButtonGroup            matlab.ui.container.ButtonGroup
        VdcButton                          matlab.ui.control.RadioButton
        IdcButton                          matlab.ui.control.RadioButton
        VabcButton                         matlab.ui.control.RadioButton
        IabcButton                         matlab.ui.control.RadioButton
        VelocidadButton                   matlab.ui.control.RadioButton
        ParButton                          matlab.ui.control.RadioButton
        ACDropDown                         matlab.ui.control.DropDown
        ACDropDownLabel                   matlab.ui.control.Label
        DCDropDown                         matlab.ui.control.DropDown
        DCDropDownLabel                   matlab.ui.control.Label
        EERRDropDown                       matlab.ui.control.DropDown
        EERRDropDownLabel                 matlab.ui.control.Label
        StartButton                       matlab.ui.control.Button
        ICOeerr                            matlab.ui.control.Image
        ACDropDown_2Label                  matlab.ui.control.Label
        ACDropDown_2                       matlab.ui.control.DropDown
        CONTROLLabel                       matlab.ui.control.Label
        Separador2                          matlab.ui.control.Image
        MOTORLabel                         matlab.ui.control.Label
        ICODcmotor                         matlab.ui.control.Image
        ICOacmotor                         matlab.ui.control.Image
        DCMDropDownLabel                   matlab.ui.control.Label
        DCMDropDown                        matlab.ui.control.DropDown
        ACMDropDownLabel                   matlab.ui.control.Label
        ACMDropDown                        matlab.ui.control.DropDown
        PROCESOLabel                       matlab.ui.control.Label
        ICOCarga                            matlab.ui.control.Image
        Separador3                          matlab.ui.control.Image
        TIPODropDownLabel                  matlab.ui.control.Label
        TIPODropDown                       matlab.ui.control.DropDown
        ASINCRONODEINDUCCIONLabel          matlab.ui.control.Label
        ASINCRONOIDropDown                  matlab.ui.control.DropDown
        ASINCRONODEROTORBOBINADOLabel     matlab.ui.control.Label
        ASINCRONORBDropDown                matlab.ui.control.DropDown
        SINCRONOLabel                      matlab.ui.control.Label
    end
end
```

Digitalización de un banco de ensayos y comprobación experimental

```
SINCRONODropDown      matlab.ui.control.DropDown
PMSMLabel              matlab.ui.control.Label
PMSMDropDown          matlab.ui.control.DropDown
RELUCTANCIALabel      matlab.ui.control.Label
RELUCTANCIADropDown  matlab.ui.control.DropDown
ParametrosIM          matlab.ui.control.Button
TconstICO              matlab.ui.control.Image
Tconst2pICO           matlab.ui.control.Image
TcuadICO               matlab.ui.control.Image
PconstICO              matlab.ui.control.Image
bobinadoraICO         matlab.ui.control.Image
bombaICO               matlab.ui.control.Image
cintaICO               matlab.ui.control.Image
cinta2pICO             matlab.ui.control.Image
ICovectorial_2        matlab.ui.control.Image
DCDropDown_2Label     matlab.ui.control.Label
DCDropDown_2          matlab.ui.control.DropDown
TLabel                 matlab.ui.control.Label
NmLabel                matlab.ui.control.Label
Label                  matlab.ui.control.Label
UIAxes                 matlab.ui.control.UIAxes

end

properties (Access = private)
    redselected = 0;
    controlselected = 0;
    motorselected = 0;
    motortypeselected = 0;
    loadselected = 0;
end

methods (Access = private)

function visibleOffMotorType(app)
    app.ASINCRONOIDropDown.Visible = 'off';
    app.ASINCRONODEINDUCCIONLabel.Visible = 'off';
    app.ASINCRONORBDropDown.Visible = 'off';
    app.ASINCRONODEROTORBOBINADOLabel.Visible = 'off';
    app.SINCRONODropDown.Visible = 'off';
    app.SINCRONOLabel.Visible = 'off';
    app.PMSMDropDown.Visible = 'off';
    app.PMSMLabel.Visible = 'off';
    app.RELUCTANCIADropDown.Visible = 'off';
    app.RELUCTANCIALabel.Visible = 'off';
end

function checkSelected(app)
    S = app.redselected;
    C = app.controlselected;
    M = app.motorselected;
```


Digitalización de un banco de ensayos y comprobación experimental

```
TM = app.motortypeselected;
L = app.loadselected;

SUMA = S+C+M+TM+L;

if(SUMA > 4)
    app.StartButton.Enable = 'on';
else
    app.StartButton.Enable = 'off';
end

end

function datos_motor_propio_IM(app)
if(evalin('base','ensayo_par')== 0 && evalin('base','CirEq_par')== 0)
    % Resistencia del estator
    % -----
    RuV = 6.6;      % Ohm
    RuW = 6.5;      % Ohm
    RvW = 6.7;      % Ohm

    % Ensayo de vacío - conexión en estrella a tensión nominal
    % -----
    U10 = 400.33;   % V
    I10 = 3.43;     % A
    P10 = 312;      % W
    Q10 = 2354;     % VAR

    % Ensayo de cortocircuito con conexión en estrella
    % -----
    Ucc = 88.03;    % V
    Icc = 4.75;     % A
    Pcc = 420;      % W
    Qcc = 586;      % VAR

    % Parámetros nominales
    % -----
    p = 2;          % Pares de polos
    Pn = 2200;      % W
    VLL = 400;      % V
    J = 0.0131;     % kg·m^2
    B = 0.002985;   % N·m·s
    Fn = 50;        % Hz

    % Resistencia estator
    % -----
    Rs = (RuV + RuW + RvW)/3/2;      % Parámetro circuito equivalente

    % Inductancia mutua
    % -----
    xm = U10^2/Q10;
```

Digitalización de un banco de ensayos y comprobación experimental

```
Lm = Xm/(2*pi*50); % Parámetro circuito equivalente

% Resistencia de cortocircuito
% -----
Rcc = Pcc/(3*Icc^2);

% Resistencia del rotor
% -----
Rr = Rcc - Rs; % Parámetro circuito equivalente

% Reactancia de cortocircuito
% -----
Xcc = Qcc/(3*Icc^2);

% Obtención de Lds, Ldr
% -----
Xds = Xcc/2;
Ls = Xds/(2*pi*50);
Lr = Ls; % Parámetros circuito equivalente
assignin('base','Rs',Rs);
assignin('base','Lm',Lm);
assignin('base','Rr',Rr);
assignin('base','Lr',Lr);
assignin('base','Ls',Ls);
assignin('base','Pn',Pn);
assignin('base','p',p);
assignin('base','VLL',VLL);
assignin('base','J',J);
assignin('base','B',B);
assignin('base','Fn',Fn);
end

if(evalin('base','ensayo_par') == 1)
% Resistencia estátor
% -----
Rs = (Ruv + Ruw + Rvw)/3/2; % Parámetro circuito equivalente

% Inductancia mutua
% -----
Xm = U10^2/Q10;
Lm = Xm/(2*pi*50); % Parámetro circuito equivalente

% Resistencia de cortocircuito
% -----
Rcc = Pcc/(3*Icc^2);

% Resistencia del rotor
% -----
Rr = Rcc - Rs; % Parámetro circuito equivalente

% Reactancia de cortocircuito
```

Digitalización de un banco de ensayos y comprobación experimental

```
% -----  
Xcc = Qcc/(3*Icc^2);  
  
% Obtención de Lds, Ldr  
% -----  
Xds = Xcc/2;  
Ls = Xds/(2*pi*50);  
Lr = Ls; % Parámetros circuito equivalente  
assignin('base','Rs', Rs);  
assignin('base','Lm', Lm);  
assignin('base','Rr', Rr);  
assignin('base','Lr', Lr);  
assignin('base','Ls', Ls);  
end  
end  
end  
  
% Callbacks that handle component events  
methods (Access = private)  
  
% Button pushed function: StartButton  
function StartButtonPushed(app, event)  
    global simout;  
    datos_motor_propio_IM(app)  
    simout = sim('MODELO.slx', 'StopTime', '0.5');  
    app.VdcButton.Value = 1;  
    plot(app.UIAxes, simout.Vdc.Time, simout.Vdc.Data);  
end  
  
% Callback function  
function IabcCheckBoxValueChanged(app, event)  
  
end  
  
% Selection changed function: ParametrosButtonGroup  
function ParametrosButtonGroupSelectionChanged(app, event)  
    global simout;  
    selectedButton = app.ParametrosButtonGroup.SelectedObject;  
    cla(app.UIAxes);  
    switch(selectedButton)  
        case app.IabcButton  
            plot(app.UIAxes, simout.Iabc.Time, simout.Iabc.Data);  
            app.UIAxes.Title.String = 'Iabc';  
            app.UIAxes.XLabel.String = 't (s)';  
            app.UIAxes.YLabel.String = 'I (A)';  
        case app.VabcButton  
            plot(app.UIAxes, simout.Vabc.Time, simout.Vabc.Data);  
            app.UIAxes.Title.String = 'Vabc';  
            app.UIAxes.XLabel.String = 't (s)';  
            app.UIAxes.YLabel.String = 'V (V)';
```

Digitalización de un banco de ensayos y comprobación experimental

```
case app.IdcButton
    plot(app.UIAxes, simout.Idc.Time, simout.Idc.Data);
    app.UIAxes.Title.String = 'Idc';
    app.UIAxes.XLabel.String = 't (s)';
    app.UIAxes.YLabel.String = 'I (A)';
case app.VdcButton
    plot(app.UIAxes, simout.Vdc.Time, simout.Vdc.Data);
    app.UIAxes.Title.String = 'Vdc';
    app.UIAxes.XLabel.String = 't (s)';
    app.UIAxes.YLabel.String = 'V (V)';
case app.ParButton
    plot(app.UIAxes, simout.Tload_f.Time, simout.Tload_f.Data);
    app.UIAxes.Title.String = 'Par';
    app.UIAxes.XLabel.String = 't (s)';
    app.UIAxes.YLabel.String = 'T (Nm)';
case app.VelocidadButton
    plot(app.UIAxes, simout.Speed_f.Time, simout.Speed_f.Data);
    app.UIAxes.Title.String = 'Velocidad';
    app.UIAxes.XLabel.String = 't (s)';
    app.UIAxes.YLabel.String = 'v (rpm)';
end
end

% Callback function
function REDButtonGroupSelectionChanged(app, event)

end

% Value changed function: ACDropDown
function ACDropDownValueChanged(app, event)
    value = app.ACDropDown.Value;
    assignin("base", 'Vdc', 0);
    assignin("base", "ensayo_par", 0);
    assignin("base", "CirEq_par", 0);
    switch value
        case 1 % Seleccionar
            app.ACMDropDown.Enable = "on";
            app.DCMDropDown.Enable = "on";
            app.ACDropDown_2.Enable = "on";
            app.DCDropDown_2.Enable = "on";
            assignin('base', 'vac', 0);
            assignin('base', 'fac', 0);
            app.redselected = 0;
        case 2 % 400V 50Hz
            app.ACMDropDown.Enable = "on";
            app.DCMDropDown.Enable = "off";
            app.ACDropDown_2.Enable = "on";
            app.DCDropDown_2.Enable = "off";
            assignin('base', 'vac', 400);
            assignin('base', 'fac', 50);
            app.redselected = 1;
    end
end
```

Digitalización de un banco de ensayos y comprobación experimental

```
case 3 % 480V 60Hz
    app.ACMDropDown.Enable = "on";
    app.DCMDropDown.Enable = "off";
    app.ACDropDown_2.Enable = "on";
    app.DCDropDown_2.Enable = "off";
    assignin('base', 'fac', 60);
    assignin('base', 'vac', 480);
    app.redselected = 1;
end

if(app.DCDropDown.Value ~= 1)
    app.DCDropDown.Value = 1;
end
if(app.EERRDropDown.Value ~= 1)
    app.EERRDropDown.Value = 1;
end
checkSelected(app);
end

% Value changed function: DCDropDown
function DCDropDownValueChanged(app, event)
    value = app.DCDropDown.Value;
    assignin("base", "ensayo_par", 0);
    assignin("base", "CirEq_par", 0);
    assignin('base', 'vac', 0);
    assignin("base", 'fac', 0);
    switch value
        case 1 % Seleccionar
            app.ACMDropDown.Enable = "on";
            app.DCMDropDown.Enable = "on";
            app.ACDropDown_2.Enable = "on";
            app.DCDropDown_2.Enable = "on";

            app.redselected = 0;
            assignin("base", 'vdc', 0);
        case 2 % 240V
            app.ACMDropDown.Enable = "off";
            app.DCMDropDown.Enable = "on";
            app.ACDropDown_2.Enable = "off";
            app.DCDropDown_2.Enable = "on";
            app.ASINCRONOIDropDown.Visible = "off";
            app.ASINCRONODEINDUCCIONLabel.Visible = 'off';
            app.ParametrosIM.Visible = 'off';
            app.redselected = 1;
            assignin("base", 'vdc', 240);
        case 3 % 120V
            app.ACMDropDown.Enable = "off";
            app.DCMDropDown.Enable = "on";
            app.ACDropDown_2.Enable = "off";
            app.DCDropDown_2.Enable = "on";
            app.ASINCRONOIDropDown.Visible = "off";
```

Digitalización de un banco de ensayos y comprobación experimental

```
        app.ASINCRONODEINDUCCIONLabel.Visible = 'off';
        app.ParametrosIM.Visible = 'off';
        app.redselected = 1;
        assignin("base", 'vdc', 120);
    end
    if(app.ACDropDown.Value ~= 1)
        app.ACDropDown.Value = 1;
    end
    if(app.EERRDropDown.Value ~= 1)
        app.EERRDropDown.Value = 1;
    end
    checkSelected(app);
end

% Value changed function: EERRDropDown
function EERRDropDownValueChanged(app, event)
    value = app.EERRDropDown.Value;
    switch value
        case 1 % Seleccionar
            app.redselected = 0;
        case 2 % Solar
            app.redselected = 1;
        case 3 % Eolica
            app.redselected = 1;
    end
    if(app.ACDropDown.Value ~= 1)
        app.ACDropDown.Value = 1;
    end
    if(app.DCDropDown.Value ~= 1)
        app.DCDropDown.Value = 1;
    end
    checkSelected(app);
end

% Value changed function: TIPODropDown
function TIPODropDownValueChanged(app, event)
    value = app.TIPODropDown.Value;
    global tipocarga;
    tipocarga = value;
    app.TconstICO.Visible = 'off';
    app.Tconst2pICO.Visible = 'off';
    app.TcuadICO.Visible = 'off';
    app.PconstICO.Visible = 'off';

    app.cinta2pICO.Visible = 'off';
    app.cintaICO.Visible = 'off';
    app.bobinadoraICO.Visible = 'off';
    app.bombaICO.Visible = 'off';

    app.TLabel.Visible = "off";
    app.Label.Visible = "off";
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.NmLabel.Visible = "off";
switch value
case '1' % Seleccionar
    app.CARGANmSlider.Visible = 'off';
    app.CARGANmSliderLabel.Visible = 'off';
    app.loadselected = 0;
case '2' % Cinta transportadora
    app.CARGANmSlider.Visible = 'on';
    app.CARGANmSliderLabel.Visible = 'on';
    app.TconstICO.Visible = 'on';
    app.cintaICO.Visible = 'on';
    app.TLabel.Visible = "on";
    app.Label.Visible = "on";
    app.NmLabel.Visible = "on";
    app.loadselected = 1;
case '3' % Bomba centrífuga
    app.CARGANmSlider.Visible = 'on';
    app.CARGANmSliderLabel.Visible = 'on';
    app.TcuadICO.Visible = 'on';
    app.bombaICO.Visible = 'on';
    app.TLabel.Visible = "on";
    app.Label.Visible = "on";
    app.NmLabel.Visible = "on";
    app.loadselected = 1;
case '4' % Bobinadora
    app.CARGANmSlider.Visible = 'on';
    app.CARGANmSliderLabel.Visible = 'on';
    app.PconstICO.Visible = 'on';
    app.bobinadoraICO.Visible = 'on';
    app.TLabel.Visible = "on";
    app.Label.Visible = "on";
    app.NmLabel.Visible = "on";
    app.loadselected = 1;
case '5' % Cinta transp. varias cargas
    app.CARGANmSlider.Visible = 'on';
    app.CARGANmSliderLabel.Visible = 'on';
    app.Tconst2pICO.Visible = 'on';
    app.cinta2pICO.Visible = 'on';
    app.TLabel.Visible = "on";
    app.Label.Visible = "on";
    app.NmLabel.Visible = "on";
    app.loadselected = 1;
end
checkSelected(app);
end

% Value changed function: ACMDropDown
function ACMDropDownValueChanged(app, event)
    value = app.ACMDropDown.Value;
    % Ocultar todos los tipos de motor
    visibleOffMotorType(app);
```

Digitalización de un banco de ensayos y comprobación experimental

```
switch value
    case '1' % Seleccionar
        app.motorselected = 0;
        assignin("base", 'M_sel_out', 0);
    case '2' % Asíncrono de inducción
        app.ASINCRONOIDropDown.Visible = 'on';
        app.ASINCRONODEINDUCCIONLabel.Visible = 'on';
        assignin("base", 'M_sel_out', 1);
        app.motorselected = 1;
    case '3' % Asíncrono de rotor bobinado
        app.ASINCRONORBDropDown.Visible = 'on';
        app.ASINCRONODEROTORBOBINADOLabel.Visible = 'on';
        app.motorselected = 1;
    case '4' % Síncrono
        app.SINCRONODropDown.Visible = 'on';
        app.SINCRONOLabel.Visible = 'on';
        app.motorselected = 1;
    case '5' % Síncrono de imanes permanentes
        app.PMSMDropDown.Visible = 'on';
        app.PMSMLabel.Visible = 'on';
        app.motorselected = 1;
    case '6' % Reluctancia variable
        app.RELUCTANCIADropDown.Visible = 'on';
        app.RELUCTANCIALabel.Visible = 'on';
        app.motorselected = 1;
end
if(app.DCMDropDown.Value ~= '1')
    app.DCMDropDown.Value = '1';
end
checkSelected(app);
end

% Value changed function: ASINCRONORBDropDown
function ASINCRONORBDropDownValueChanged(app, event)
    value = app.ASINCRONORBDropDown.Value;
    switch value
        case '1'
            app.motortypeselectd = 0;
        case '2'
            app.motortypeselectd = 1;
    end
    checkSelected(app);
end

% Value changed function: ACDropDown_2
function ACDropDown_2ValueChanged(app, event)
    value = app.ACDropDown_2.Value;
    switch value
        case '1' % Seleccionar
            app.controlselected = 0;
            assignin("base", "control_selector", 0);
```


Digitalización de un banco de ensayos y comprobación experimental

```
        assignin("base", "contr_ac_dc", 0);
    case '2' % V/F
        app.controlselected = 1;
        assignin("base", "control_selector", 2);
        assignin("base", "contr_ac_dc", 2);
    case '3' % DTC
        app.controlselected = 1;
        assignin("base", "control_selector", 1);
        assignin("base", "contr_ac_dc", 2);
    end
    checkSelected(app);
end

% Value changed function: RELUCTANCIADropDown
function RELUCTANCIADropDownValueChanged(app, event)
    value = app.RELUCTANCIADropDown.Value;
    switch value
        case '1' % Seleccionar
            app.motortypeselected = 0;
        case '2' % 400V 50Hz
            app.motortypeselected = 1;
    end
    checkSelected(app);
end

% Value changed function: PMSMDropDown
function PMSMDropDownValueChanged(app, event)
    value = app.PMSMDropDown.Value;
    switch value
        case '1' % Seleccionar
            app.motortypeselected = 0;
        case '2' % 400V 50Hz
            app.motortypeselected = 1;
    end
    checkSelected(app);
end

% Value changed function: SINCRONODropDown
function SINCRONODropDownValueChanged(app, event)
    value = app.SINCRONODropDown.Value;
    switch value
        case '1' % Seleccionar
            app.motortypeselected = 0;
        case '2' % 400V 50Hz
            app.motortypeselected = 1;
    end
    checkSelected(app);
end

% Value changed function: ASINCRONOIDropDown
function ASINCRONOIDropDownValueChanged(app, event)
```

Digitalización de un banco de ensayos y comprobación experimental

```
value = app.ASINCRONOIDropDown.Value;
switch value
    case '1' % Seleccionar
        app.motortypeselected = 0;
        assignin("base",'IM_PROPIO', 0);
        assignin("base",'IM_SYM', 0);
        assignin("base",'M_sel_out', 0);
        assignin("base",'IM_selector', 0);
    case '2' % 400v 50Hz
        app.motortypeselected = 1;
        assignin("base",'IM_PROPIO', 0);
        assignin("base",'IM_SYM', 1);
        assignin("base",'M_sel_out', 1);
        assignin("base",'IM_selector', 2);
    case '3' % Personalizado
        app.motortypeselected = 1;
        assignin("base",'IM_PROPIO', 1);
        assignin("base",'IM_SYM', 0);
        assignin("base",'M_sel_out', 1);
        assignin("base",'IM_selector', 1);
        app.ParametrosIM.Visible = 'on';
end
checkSelected(app);
end

% Value changed function: DCMDropDown
function DCMDropDownValueChanged(app, event)
    value = app.DCMDropDown.Value;
    assignin("base",'IM_PROPIO', 0);
    assignin("base",'IM_SYM', 0);
    assignin("base",'IM_selector', 0);
    switch value
        case '1' % Seleccionar
            app.motorselected = 0;
            assignin("base", "contr_ac_dc", 0);
        case '2' % 10HP 1750rpm 240V
            app.motorselected = 2;
            assignin("base", "M_sel_out", 2);
            assignin("base", "contr_ac_dc", 1);
            app.ASINCRONOIDropDown.Visible = 'off';
            app.ASINCRONODEINDUCCIONLabel.Visible = 'off';
        case '3' % Shunt
            app.motorselected = 2;
            assignin("base", "M_sel_out", 2);
            assignin("base", "contr_ac_dc", 1);
            app.ASINCRONOIDropDown.Visible = 'off';
            app.ASINCRONODEINDUCCIONLabel.Visible = 'off';
    end
    if(app.ACMDropDown.Value ~= '1')
        app.ACMDropDown.value = '1';
    end
end
```

Digitalización de un banco de ensayos y comprobación experimental

```
        checkSelected(app);
    end

    % Callback function
    function VECTORIALDropDownValueChanged(app, event)
        value = app.VECTORIALDropDown.Value;
        switch value
            case 1 % Seleccionar
                app.controlselected = 0;
                assignin("base", "control_selector", 0);
                assignin("base", "contr_ac_dc", 0);
            case 2 % FOC
                app.controlselected = 1;
                assignin("base", "control_selector", 1);
                assignin("base", "contr_ac_dc", 2);
            case 3 % DTC
                app.controlselected = 1;
                assignin("base", "control_selector", 1);
                assignin("base", "contr_ac_dc", 2);
        end
        if(app.ACDropDown_2.Value ~= '1')
            app.ACDropDown_2.Value = '1';
        end
        checkSelected(app);
    end

    % Value changed function: VelocidaddeconsignarpmEditField
    function velocidaddeconsignarpmEditFieldValueChanged(app, event)
        value = app.VelocidaddeconsignarpmEditField.Value;
        assignin('base', 'vref', value);
    end

    % Value changed function: CARGANmSlider
    function CARGANmSliderValueChanged(app, event)
        value = app.CARGANmSlider.Value;
        global tipocarga;
        app.Label.Text = num2str(value);
        switch tipocarga
            case '1'
                assignin('base', 'Tload_ref', 0);

                assignin('base', 'Load_selector', 0);
            case '2'
                assignin('base', 'Tload_ref', value);
                assignin('base', 'Load_selector', 1);
            case '3'
                assignin('base', 'Tload_ref', value);
                assignin('base', 'Load_selector', 2);
            case '4'
                assignin('base', 'Tload_ref', value);
                app.Label.Text = "Tload_ref";
        end
    end
end
```

Digitalización de un banco de ensayos y comprobación experimental

```
        assignin('base','Load_selector',3);
    case '5'
        assignin('base','Tload_ref', value);
        assignin('base','Tload_ref42', value/2);
        assignin('base','Tload_ref1',0);
        assignin('base','Tload_ref2',0);
        assignin('base','Tload_ref3',0);
        assignin('base','Load_selector',4);
    end
end

% Button pushed function: ParametrosIM
function ParametrosIMButtonPushed(app, event)
    param_IM;
end

% Value changed function: DCDropDown_2
function DCDropDown_2ValueChanged(app, event)
    value = app.DCDropDown_2.Value;
    assignin("base", "control_selector", 0);
    switch value
        case '1' % Seleccionar
            app.controlselected = 0;
            assignin("base", "control_selector", 0);
            assignin("base", "contr_ac_dc", 0);
        case '2' % Chopper
            app.controlselected = 1;
    end
    if(app.ACDropDown_2.value ~= '1')
        app.ACDropDown_2.Value = '1';
    end
    checkSelected(app);
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

    % Create BancodeensayosUIFigure and hide until all components are created
    app.BancodeensayosUIFigure = uifigure('Visible', 'off');
    app.BancodeensayosUIFigure.Color = [0.9098 0.949 0.9804];
    app.BancodeensayosUIFigure.Position = [100 100 794 528];
    app.BancodeensayosUIFigure.Name = 'Banco de ensayos';
    app.BancodeensayosUIFigure.Icon = 'AC.png';
    app.BancodeensayosUIFigure.Resize = 'off';

    % Create REDLabel
    app.REDLabel = uilabel(app.BancodeensayosUIFigure);
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.REDLabel.HorizontalAlignment = 'center';
app.REDLabel.FontSize = 15;
app.REDLabel.FontWeight = 'bold';
app.REDLabel.Position = [1 478 171 32];
app.REDLabel.Text = 'RED';

% Create ICOac
app.ICOac = uimage(app.BancodeensayosUIFigure);
app.ICOac.Position = [1 445 27 27];
app.ICOac.ImageSource = 'AC.png';

% Create ICODc
app.ICODc = uimage(app.BancodeensayosUIFigure);
app.ICODc.Position = [1 420 27 27];
app.ICODc.ImageSource = 'DC.png';

% Create ICovf
app.ICovf = uilabel(app.BancodeensayosUIFigure);
app.ICovf.HorizontalAlignment = 'center';
app.ICovf.FontSize = 13;
app.ICovf.Position = [189 445 27 27];
app.ICovf.Text = 'V/F';

% Create Separador1
app.Separador1 = uimage(app.BancodeensayosUIFigure);
app.Separador1.Position = [171 369 23 133];
app.Separador1.ImageSource = 'a.png';

% Create CARGANmsliderLabel
app.CARGANmsliderLabel = uilabel(app.BancodeensayosUIFigure);
app.CARGANmsliderLabel.HorizontalAlignment = 'right';
app.CARGANmsliderLabel.Visible = 'off';
app.CARGANmsliderLabel.Position = [659 408 78 22];
app.CARGANmsliderLabel.Text = 'CARGA (Nm)';

% Create CARGANmslider
app.CARGANmslider = uislider(app.BancodeensayosUIFigure);
app.CARGANmslider.Limits = [0 25];
app.CARGANmslider.Orientation = 'vertical';
app.CARGANmslider.ValueChangedFcn = createCallbackFcn(app,
@CARGANmsliderValueChanged, true);
app.CARGANmslider.Visible = 'off';
app.CARGANmslider.Position = [747 336 3 94];

% Create velocidaddeconsignarpmEditFieldLabel
app.velocidaddeconsignarpmEditFieldLabel =
uilabel(app.BancodeensayosUIFigure);
app.velocidaddeconsignarpmEditFieldLabel.HorizontalAlignment = 'right';
app.velocidaddeconsignarpmEditFieldLabel.Position = [96 30 160 22];
app.velocidaddeconsignarpmEditFieldLabel.Text = 'velocidad de consigna
(rpm)';
```

Digitalización de un banco de ensayos y comprobación experimental

```
% Create velocidaddeconsignarpmEditField
app.velocidaddeconsignarpmEditField =
uieditfield(app.BancodeensayosUIFigure, 'numeric');
app.velocidaddeconsignarpmEditField.ValueChangedFcn = createCallbackFcn(app,
@velocidaddeconsignarpmEditFieldValueChanged, true);
app.velocidaddeconsignarpmEditField.Position = [268 30 56 22];

% Create ParametrosButtonGroup
app.ParametrosButtonGroup = uibuttongroup(app.BancodeensayosUIFigure);
app.ParametrosButtonGroup.SelectionChangedFcn = createCallbackFcn(app,
@ParametrosButtonGroupSelectionChanged, true);
app.ParametrosButtonGroup.BorderType = 'none';
app.ParametrosButtonGroup.TitlePosition = 'centertop';
app.ParametrosButtonGroup.Title = 'Parámetros';
app.ParametrosButtonGroup.BackgroundColor = [0.9098 0.949 0.9804];
app.ParametrosButtonGroup.Position = [644 109 143 185];

% Create VdcButton
app.VdcButton = uiradiobutton(app.ParametrosButtonGroup);
app.VdcButton.Text = 'Vdc';
app.VdcButton.Position = [11 140 58 22];
app.VdcButton.Value = true;

% Create IdcButton
app.IdcButton = uiradiobutton(app.ParametrosButtonGroup);
app.IdcButton.Text = 'Idc';
app.IdcButton.Position = [11 118 65 22];

% Create VabcButton
app.VabcButton = uiradiobutton(app.ParametrosButtonGroup);
app.VabcButton.Text = 'Vabc';
app.VabcButton.Position = [11 96 65 22];

% Create IabcButton
app.IabcButton = uiradiobutton(app.ParametrosButtonGroup);
app.IabcButton.Text = 'Iabc';
app.IabcButton.Position = [11 74 45 22];

% Create velocidadButton
app.velocidadButton = uiradiobutton(app.ParametrosButtonGroup);
app.velocidadButton.Text = 'velocidad';
app.velocidadButton.Position = [11 33 74 22];

% Create ParButton
app.ParButton = uiradiobutton(app.ParametrosButtonGroup);
app.ParButton.Text = 'Par';
app.ParButton.Position = [11 12 41 22];

% Create ACDropDown
app.ACDropDown = uidropdown(app.BancodeensayosUIFigure);
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.ACDropDown.Items = {'Seleccionar', '400V 50Hz', '480V 60Hz'};
app.ACDropDown.ItemsData = [1 2 3];
app.ACDropDown.ValueChangedFcn = createCallbackFcn(app,
@ACDropDownValueChanged, true);
app.ACDropDown.Position = [76 448 96 22];
app.ACDropDown.Value = 1;

% Create ACDropDownLabel
app.ACDropDownLabel = uilabel(app.BancodeensayosUIFigure);
app.ACDropDownLabel.HorizontalAlignment = 'right';
app.ACDropDownLabel.Position = [26 446 25 22];
app.ACDropDownLabel.Text = 'AC';

% Create DCDropDown
app.DCDropDown = uidropdown(app.BancodeensayosUIFigure);
app.DCDropDown.Items = {'Seleccionar', '240V', '120V'};
app.DCDropDown.ItemsData = [1 2 3];
app.DCDropDown.ValueChangedFcn = createCallbackFcn(app,
@DCDropDownValueChanged, true);
app.DCDropDown.Position = [76 423 96 22];
app.DCDropDown.Value = 1;

% Create DCDropDownLabel
app.DCDropDownLabel = uilabel(app.BancodeensayosUIFigure);
app.DCDropDownLabel.HorizontalAlignment = 'right';
app.DCDropDownLabel.Position = [26 421 25 22];
app.DCDropDownLabel.Text = 'DC';

% Create EERRDropDown
app.EERRDropDown = uidropdown(app.BancodeensayosUIFigure);
app.EERRDropDown.Items = {'Seleccionar', 'solar', 'Eólica'};
app.EERRDropDown.ItemsData = [1 2 3];
app.EERRDropDown.ValueChangedFcn = createCallbackFcn(app,
@EERRDropDownValueChanged, true);
app.EERRDropDown.Visible = 'off';
app.EERRDropDown.Position = [76 398 96 22];
app.EERRDropDown.Value = 1;

% Create EERRDropDownLabel
app.EERRDropDownLabel = uilabel(app.BancodeensayosUIFigure);
app.EERRDropDownLabel.HorizontalAlignment = 'right';
app.EERRDropDownLabel.Visible = 'off';
app.EERRDropDownLabel.Position = [26 396 39 22];
app.EERRDropDownLabel.Text = 'EERR';

% Create StartButton
app.StartButton = uibutton(app.BancodeensayosUIFigure, 'push');
app.StartButton.ButtonPushedFcn = createCallbackFcn(app, @StartButtonPushed,
true);

app.StartButton.Enable = 'off';
app.StartButton.Position = [392 18 243 46];
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.StartButton.Text = 'Start';

% Create ICOeerr
app.ICOeerr = uiimage(app.BancodeensayosUIFigure);
app.ICOeerr.Visible = 'off';
app.ICOeerr.Position = [1 395 27 27];
app.ICOeerr.ImageSource = 'EERR.png';

% Create ACDropDown_2Label
app.ACDropDown_2Label = uilabel(app.BancodeensayosUIFigure);
app.ACDropDown_2Label.HorizontalAlignment = 'right';
app.ACDropDown_2Label.Position = [223 446 25 22];
app.ACDropDown_2Label.Text = 'AC';

% Create ACDropDown_2
app.ACDropDown_2 = uidropdown(app.BancodeensayosUIFigure);
app.ACDropDown_2.Items = {'seleccionar', 'V/F', 'DTC'};
app.ACDropDown_2.ItemsData = {'1', '2', '3'};
app.ACDropDown_2.ValueChangedFcn = createCallbackFcn(app,
@ACDropDown_2ValueChanged, true);
app.ACDropDown_2.Position = [295 449 101 22];
app.ACDropDown_2.Value = '1';

% Create CONTROLLabel
app.CONTROLLabel = uilabel(app.BancodeensayosUIFigure);
app.CONTROLLabel.HorizontalAlignment = 'center';
app.CONTROLLabel.FontSize = 15;
app.CONTROLLabel.FontWeight = 'bold';
app.CONTROLLabel.Position = [189 478 212 32];
app.CONTROLLabel.Text = 'CONTROL';

% Create Separador2
app.Separador2 = uiimage(app.BancodeensayosUIFigure);
app.Separador2.Position = [400 369 23 133];
app.Separador2.ImageSource = 'a.png';

% Create MOTORLabel
app.MOTORLabel = uilabel(app.BancodeensayosUIFigure);
app.MOTORLabel.HorizontalAlignment = 'center';
app.MOTORLabel.FontSize = 15;
app.MOTORLabel.FontWeight = 'bold';
app.MOTORLabel.Position = [422 478 179 32];
app.MOTORLabel.Text = 'MOTOR';

% Create ICODcmotor
app.ICODcmotor = uiimage(app.BancodeensayosUIFigure);
app.ICODcmotor.ScaleMethod = 'fill';
app.ICODcmotor.Position = [407 436 58 43];
app.ICODcmotor.ImageSource = 'DCM.png';

% Create ICOacmotor
```


Digitalización de un banco de ensayos y comprobación experimental

```
app.ICOacmotor = uimage(app.BancodeensayosUIFigure);
app.ICOacmotor.ScaleMethod = 'fill';
app.ICOacmotor.Position = [403 398 66 56];
app.ICOacmotor.ImageSource = 'acm.png';

% Create DCMDropDownLabel
app.DCMDropDownLabel = uilabel(app.BancodeensayosUIFigure);
app.DCMDropDownLabel.HorizontalAlignment = 'right';
app.DCMDropDownLabel.Position = [455 445 25 22];
app.DCMDropDownLabel.Text = 'DC';

% Create DCMDropDown
app.DCMDropDown = uidropdown(app.BancodeensayosUIFigure);
app.DCMDropDown.Items = {'Seleccionar', '10 HP 240 v 1750 rpm'};
app.DCMDropDown.ItemsData = {'1', '2'};
app.DCMDropDown.ValueChangedFcn = createCallbackFcn(app,
@DCMDropDownValueChanged, true);
app.DCMDropDown.Position = [505 447 96 22];
app.DCMDropDown.Value = '1';

% Create ACMDropDownLabel
app.ACMDropDownLabel = uilabel(app.BancodeensayosUIFigure);
app.ACMDropDownLabel.HorizontalAlignment = 'right';
app.ACMDropDownLabel.Position = [455 414 25 22];
app.ACMDropDownLabel.Text = 'AC';

% Create ACMDropDown
app.ACMDropDown = uidropdown(app.BancodeensayosUIFigure);
app.ACMDropDown.Items = {'Seleccionar', 'Asíncrono de Inducción'};
app.ACMDropDown.ItemsData = {'1', '2', ''};
app.ACMDropDown.ValueChangedFcn = createCallbackFcn(app,
@ACMDropDownValueChanged, true);
app.ACMDropDown.Position = [505 416 96 22];
app.ACMDropDown.Value = '1';

% Create PROCESOLabel
app.PROCESOLabel = uilabel(app.BancodeensayosUIFigure);
app.PROCESOLabel.HorizontalAlignment = 'center';
app.PROCESOLabel.FontSize = 15;
app.PROCESOLabel.Fontweight = 'bold';
app.PROCESOLabel.Position = [612 478 171 32];
app.PROCESOLabel.Text = 'PROCESO';

% Create ICOCarga
app.ICOCarga = uimage(app.BancodeensayosUIFigure);
app.ICOCarga.Position = [618 444 27 27];
app.ICOCarga.ImageSource = 'Res.png';

% Create Separador3
app.Separador3 = uimage(app.BancodeensayosUIFigure);
app.Separador3.Position = [598 367 23 133];
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.Separador3.ImageSource = 'a.png';

% Create TIPODropDownLabel
app.TIPODropDownLabel = uilabel(app.BancodeensayosUIFigure);
app.TIPODropDownLabel.HorizontalAlignment = 'right';
app.TIPODropDownLabel.Position = [645 445 34 22];
app.TIPODropDownLabel.Text = 'TIPO';

% Create TIPODropDown
app.TIPODropDown = uidropdown(app.BancodeensayosUIFigure);
app.TIPODropDown.Items = {'seleccionar', 'Cinta transportadora', 'Bomba
centrífuga', 'Bobinadora', 'Cinta transportadora varias cargas'};
app.TIPODropDown.ItemsData = {'1', '2', '3', '4', '5'};
app.TIPODropDown.ValueChangedFcn = createCallbackFcn(app,
@TIPODropDownValueChanged, true);
app.TIPODropDown.Position = [692 447 96 22];
app.TIPODropDown.Value = '1';

% Create ASINCRONODEINDUCCIONLabel
app.ASINCRONODEINDUCCIONLabel = uilabel(app.BancodeensayosUIFigure);
app.ASINCRONODEINDUCCIONLabel.HorizontalAlignment = 'center';
app.ASINCRONODEINDUCCIONLabel.WordWrap = 'on';
app.ASINCRONODEINDUCCIONLabel.FontSize = 10;
app.ASINCRONODEINDUCCIONLabel.Visible = 'off';
app.ASINCRONODEINDUCCIONLabel.Position = [422 345 73 54];
app.ASINCRONODEINDUCCIONLabel.Text = 'ASÍNCRONO DE INDUCCIÓN';

% Create ASINCRONOIDropDown
app.ASINCRONOIDropDown = uidropdown(app.BancodeensayosUIFigure);
app.ASINCRONOIDropDown.Items = {'seleccionar', '3φ - 400V 50Hz - 3A - 4 kw -
1430 rpm', 'Personalizado'};
app.ASINCRONOIDropDown.ItemsData = {'1', '2', '3'};
app.ASINCRONOIDropDown.ValueChangedFcn = createCallbackFcn(app,
@ASINCRONOIDropDownValueChanged, true);
app.ASINCRONOIDropDown.Visible = 'off';
app.ASINCRONOIDropDown.Position = [505 361 96 22];
app.ASINCRONOIDropDown.Value = '1';

% Create ASINCRONODEROTORBOBINADOLabel
app.ASINCRONODEROTORBOBINADOLabel = uilabel(app.BancodeensayosUIFigure);
app.ASINCRONODEROTORBOBINADOLabel.HorizontalAlignment = 'center';
app.ASINCRONODEROTORBOBINADOLabel.WordWrap = 'on';
app.ASINCRONODEROTORBOBINADOLabel.FontSize = 10;
app.ASINCRONODEROTORBOBINADOLabel.Visible = 'off';
app.ASINCRONODEROTORBOBINADOLabel.Position = [423 345 73 54];
app.ASINCRONODEROTORBOBINADOLabel.Text = 'ASÍNCRONO DE ROTOR BOBINADO';

% Create ASINCRONORBDropDown
app.ASINCRONORBDropDown = uidropdown(app.BancodeensayosUIFigure);
app.ASINCRONORBDropDown.Items = {'seleccionar', '3φ - 400V 50Hz - 3A - 1.5
kw - 2900 rpm'};
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.ASINCRONORBDropDown.ItemsData = {'1', '2'};
app.ASINCRONORBDropDown.ValueChangedFcn = createCallbackFcn(app,
@ASINCRONORBDropDownValueChanged, true);
app.ASINCRONORBDropDown.Visible = 'off';
app.ASINCRONORBDropDown.Position = [506 361 96 22];
app.ASINCRONORBDropDown.Value = '1';

% Create SINCRONOLabel
app.SINCRONOLabel = uilabel(app.BancodeensayosUIFigure);
app.SINCRONOLabel.HorizontalAlignment = 'center';
app.SINCRONOLabel.WordWrap = 'on';
app.SINCRONOLabel.FontSize = 10;
app.SINCRONOLabel.Visible = 'off';
app.SINCRONOLabel.Position = [423 345 73 54];
app.SINCRONOLabel.Text = 'SÍNCRONO';

% Create SINCRONODropDown
app.SINCRONODropDown = uidropdown(app.BancodeensayosUIFigure);
app.SINCRONODropDown.Items = {'Seleccionar', '3 $\phi$  - 400V 50Hz - 3A - 1.5 kw -
2900 rpm'};
app.SINCRONODropDown.ItemsData = {'1', '2'};
app.SINCRONODropDown.ValueChangedFcn = createCallbackFcn(app,
@SINCRONODropDownValueChanged, true);
app.SINCRONODropDown.Visible = 'off';
app.SINCRONODropDown.Position = [506 361 96 22];
app.SINCRONODropDown.Value = '1';

% Create PMSMLabel
app.PMSMLabel = uilabel(app.BancodeensayosUIFigure);
app.PMSMLabel.HorizontalAlignment = 'center';
app.PMSMLabel.WordWrap = 'on';
app.PMSMLabel.FontSize = 10;
app.PMSMLabel.Visible = 'off';
app.PMSMLabel.Position = [407 345 89 54];
app.PMSMLabel.Text = 'SÍNCRONO DE IMANES PERMANENTES';

% Create PMSMDropDown
app.PMSMDropDown = uidropdown(app.BancodeensayosUIFigure);
app.PMSMDropDown.Items = {'Seleccionar', '3 $\phi$  - 400V 50Hz - 3A - 1.5 kw -
2900 rpm'};
app.PMSMDropDown.ItemsData = {'1', '2'};
app.PMSMDropDown.ValueChangedFcn = createCallbackFcn(app,
@PMSMDropDownValueChanged, true);
app.PMSMDropDown.Visible = 'off';
app.PMSMDropDown.Position = [506 361 96 22];
app.PMSMDropDown.Value = '1';

% Create RELUCTANCIALabel
app.RELUCTANCIALabel = uilabel(app.BancodeensayosUIFigure);
app.RELUCTANCIALabel.HorizontalAlignment = 'center';
app.RELUCTANCIALabel.WordWrap = 'on';
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.RELUCTANCIALabel.FontSize = 10;
app.RELUCTANCIALabel.Visible = 'off';
app.RELUCTANCIALabel.Position = [408 345 89 54];
app.RELUCTANCIALabel.Text = 'RELUCTANCIA VARIABLE';

% Create RELUCTANCIADropDown
app.RELUCTANCIADropDown = uidropdown(app.BancodeensayosUIFigure);
app.RELUCTANCIADropDown.Items = {'seleccionar', '3 $\Phi$  - 400V 50Hz - 3A - 1.5
kw - 2900 rpm'};
app.RELUCTANCIADropDown.ItemsData = {'1', '2'};
app.RELUCTANCIADropDown.ValueChangedFcn = createCallbackFcn(app,
@RELUCTANCIADropDownValueChanged, true);
app.RELUCTANCIADropDown.Visible = 'off';
app.RELUCTANCIADropDown.Position = [507 361 96 22];
app.RELUCTANCIADropDown.Value = '1';

% Create ParametrosIM
app.ParametrosIM = uibutton(app.BancodeensayosUIFigure, 'push');
app.ParametrosIM.ButtonPushedFcn = createCallbackFcn(app,
@ParametrosIMButtonPushed, true);
app.ParametrosIM.Visible = 'off';
app.ParametrosIM.Position = [608 361 13 22];
app.ParametrosIM.Text = ':';

% Create TconstICO
app.TconstICO = uiimage(app.BancodeensayosUIFigure);
app.TconstICO.Visible = 'off';
app.TconstICO.Position = [668 336 61 70];
app.TconstICO.ImageSource = 'Parconstante.png';

% Create Tconst2pICO
app.Tconst2pICO = uiimage(app.BancodeensayosUIFigure);
app.Tconst2pICO.Visible = 'off';
app.Tconst2pICO.Position = [668 336 61 70];
app.Tconst2pICO.ImageSource = 'Parcuadr.png';

% Create TcuadICO
app.TcuadICO = uiimage(app.BancodeensayosUIFigure);
app.TcuadICO.Visible = 'off';
app.TcuadICO.Position = [669 337 61 70];
app.TcuadICO.ImageSource = 'Parconstantecambio.png';

% Create PconstICO
app.PconstICO = uiimage(app.BancodeensayosUIFigure);
app.PconstICO.Visible = 'off';
app.PconstICO.Position = [668 336 61 70];
app.PconstICO.ImageSource = 'potconst.png';

% Create bobinadoraICO
app.bobinadoraICO = uiimage(app.BancodeensayosUIFigure);
app.bobinadoraICO.Visible = 'off';
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.bobinadoraICO.Position = [620 404 36 31];
app.bobinadoraICO.ImageSource = 'bobina.png';

% Create bombaICO
app.bombaICO = uiimage(app.BancodeensayosUIFigure);
app.bombaICO.Visible = 'off';
app.bombaICO.Position = [620 404 36 31];
app.bombaICO.ImageSource = 'bomba.png';

% Create cintaICO
app.cintaICO = uiimage(app.BancodeensayosUIFigure);
app.cintaICO.Visible = 'off';
app.cintaICO.Position = [620 405 36 31];
app.cintaICO.ImageSource = 'cinta.png';

% Create cinta2pICO
app.cinta2pICO = uiimage(app.BancodeensayosUIFigure);
app.cinta2pICO.Visible = 'off';
app.cinta2pICO.Position = [620 404 36 31];
app.cinta2pICO.ImageSource = 'cinta2p.png';

% Create ICovectorial_2
app.ICovectorial_2 = uiimage(app.BancodeensayosUIFigure);
app.ICovectorial_2.Position = [181 415 45 30];
app.ICovectorial_2.ImageSource = 'mosfet.png';

% Create DCDropDown_2Label
app.DCDropDown_2Label = uilabel(app.BancodeensayosUIFigure);
app.DCDropDown_2Label.HorizontalAlignment = 'right';
app.DCDropDown_2Label.Position = [226 422 23 22];
app.DCDropDown_2Label.Text = 'DC';

% Create DCDropDown_2
app.DCDropDown_2 = uidropdown(app.BancodeensayosUIFigure);
app.DCDropDown_2.Items = {'seleccionar', 'Chopper'};
app.DCDropDown_2.ItemsData = {'1', '2'};
app.DCDropDown_2.ValueChangedFcn = createCallbackFcn(app,
@DCDropDown_2ValueChanged, true);
app.DCDropDown_2.Position = [294 422 100 22];
app.DCDropDown_2.Value = '1';

% Create TLabel
app.TLabel = uilabel(app.BancodeensayosUIFigure);
app.TLabel.Visible = 'off';
app.TLabel.Position = [668 308 25 22];
app.TLabel.Text = 'T =';

% Create NmLabel
app.NmLabel = uilabel(app.BancodeensayosUIFigure);
app.NmLabel.Visible = 'off';
app.NmLabel.Position = [736 308 25 22];
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.NmLabel.Text = 'Nm';

% Create Label
app.Label = uilabel(app.BancodeensayosUIFigure);
app.Label.HorizontalAlignment = 'center';
app.Label.Visible = 'off';
app.Label.Position = [687 308 50 22];
app.Label.Text = '0';

% Create UIAxes
app.UIAxes = uiaxes(app.BancodeensayosUIFigure);
title(app.UIAxes, 'Vdc')
xlabel(app.UIAxes, 't (s)')
ylabel(app.UIAxes, 'v (V)')
zlabel(app.UIAxes, 'z')
app.UIAxes.PlotBoxAspectRatio = [2.43668122270742 1 1];
app.UIAxes.XTick = [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
app.UIAxes.Position = [26 65 605 305];

% Show the figure after all components are created
app.BancodeensayosUIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = INTERF_GENERAL

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.BancodeensayosUIFigure)

if nargin == 0
clear app
end
end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app.BancodeensayosUIFigure)
end
end
end
```



ANEXO 3 – CÓDIGO DE INTERFAZ GRÁFICA DE MOTOR PERSONALIZADO



Digitalización de un banco de ensayos y comprobación experimental

```
classdef INTERF_PERSONALIZ < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        ParmetrosmotorpersonalizadoUIFigure matlab.ui.Figure
        ButtonGroup matlab.ui.container.ButtonGroup
        ensayo matlab.ui.control.RadioButton
        Circ_Eq matlab.ui.control.RadioButton
        RuvLabel matlab.ui.control.Label
        RuvEditField matlab.ui.control.NumericEditField
        RuwLabel matlab.ui.control.Label
        RuwEditField matlab.ui.control.NumericEditField
        RvwLabel matlab.ui.control.Label
        RvwEditField matlab.ui.control.NumericEditField
        U10VEditFieldLabel matlab.ui.control.Label
        U10VEditField matlab.ui.control.NumericEditField
        I10AEditFieldLabel matlab.ui.control.Label
        I10AEditField matlab.ui.control.NumericEditField
        P10WEditFieldLabel matlab.ui.control.Label
        P10WEditField matlab.ui.control.NumericEditField
        Q10VAEditFieldLabel matlab.ui.control.Label
        Q10VAEditField matlab.ui.control.NumericEditField
        UccVEditFieldLabel matlab.ui.control.Label
        UccVEditField matlab.ui.control.NumericEditField
        IccAEditFieldLabel matlab.ui.control.Label
        IccAEditField matlab.ui.control.NumericEditField
        PccWEditFieldLabel matlab.ui.control.Label
        PccWEditField matlab.ui.control.NumericEditField
        QccVAEditFieldLabel matlab.ui.control.Label
        QccVAEditField matlab.ui.control.NumericEditField
        ResistenciaestaticaLabel matlab.ui.control.Label
        EnsayodevacoLabel matlab.ui.control.Label
        EnsayodecortocircuitoLabel matlab.ui.control.Label
        RsEditFieldLabel matlab.ui.control.Label
        RsEditField matlab.ui.control.NumericEditField
        LmHEditFieldLabel matlab.ui.control.Label
        LmHEditField matlab.ui.control.NumericEditField
        RrEditFieldLabel matlab.ui.control.Label
        RrEditField matlab.ui.control.NumericEditField
        LrHEditFieldLabel matlab.ui.control.Label
        LrHEditField matlab.ui.control.NumericEditField
        LSHEditFieldLabel matlab.ui.control.Label
        LSHEditField matlab.ui.control.NumericEditField
        AceptarButton matlab.ui.control.Button
        CancelarButton matlab.ui.control.Button
        MOTORDEINDUCCINPERSONALIZADOLabel matlab.ui.control.Label
        ParesdepolosEditFieldLabel matlab.ui.control.Label
        ParesdepolosEditField matlab.ui.control.NumericEditField
        PnwEditFieldLabel matlab.ui.control.Label
        PnwEditField matlab.ui.control.NumericEditField
    end
end
```


Digitalización de un banco de ensayos y comprobación experimental

```
ULLVEditFieldLabel    matlab.ui.control.Label
ULLVEditField         matlab.ui.control.NumericEditField
Jkgm2EditFieldLabel  matlab.ui.control.Label
Jkgm2EditField       matlab.ui.control.NumericEditField
BNmsEditFieldLabel   matlab.ui.control.Label
BNmsEditField        matlab.ui.control.NumericEditField
fnHzEditFieldLabel   matlab.ui.control.Label
fnHzEditField        matlab.ui.control.NumericEditField
end
```

```
methods (Access = private)
```

```
function enableEnsayoEditField(app)
    % Enable
    app.RuVEditField.Enable = 'on';
    app.RuWEditField.Enable = 'on';
    app.RvWEditField.Enable = 'on';
    app.U10VEditField.Enable = 'on';
    app.I10AEditField.Enable = 'on';
    app.P10WEditField.Enable = 'on';
    app.Q10VAEditField.Enable = 'on';
    app.UccVEditField.Enable = 'on';
    app.IccaEditField.Enable = 'on';
    app.PccWEditField.Enable = 'on';
    app.QccVAEditField.Enable = 'on';
```

```
    % Disable
    app.RsEditField.Enable = 'off';
    app.RrEditField.Enable = 'off';
    app.LmHEditField.Enable = 'off';
    app.LrHEditField.Enable = 'off';
    app.LsHEditField.Enable = 'off';
```

```
end
function enableCirc_EqEditField(app)
    % Disable
    app.RuVEditField.Enable = 'off';
    app.RuWEditField.Enable = 'off';
    app.RvWEditField.Enable = 'off';
    app.U10VEditField.Enable = 'off';
    app.I10AEditField.Enable = 'off';
    app.P10WEditField.Enable = 'off';
    app.Q10VAEditField.Enable = 'off';
    app.UccVEditField.Enable = 'off';
    app.IccaEditField.Enable = 'off';
    app.PccWEditField.Enable = 'off';
    app.QccVAEditField.Enable = 'off';

    % Enable
    app.RsEditField.Enable = 'on';
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.RrEditField.Enable = 'on';
app.LmHEditField.Enable = 'on';
app.LrHEditField.Enable = 'on';
app.LsHEditField.Enable = 'on';
end
end

% Callbacks that handle component events
methods (Access = private)

% Code that executes after component creation
function startupFcn(app)

end

% Selection changed function: ButtonGroup
function ButtonGroupSelectionChanged(app, event)
    selectedButton = app.ButtonGroup.SelectedObject;
    switch(selectedButton)
        case app.ensayo
            enableEnsayoEditField(app);
            assignin("base", "ensayo_par", 1);
            assignin("base", "CirEq_par", 0);
        case app.Circ_Eq
            enableCirc_EqEditField(app);
            assignin("base", "ensayo_par", 0);
            assignin("base", "CirEq_par", 1);
    end
end

% Button pushed function: AceptarButton
function AceptarButtonPushed(app, event)
    assignin("base", "parameters_changed", 1);

    assignin('base', 'p', app.ParesdepolosEditField.value);
    assignin('base', 'Pn', app.PnwEditField.value);
    assignin('base', 'VLL', app.ULLVEditField.value);
    assignin('base', 'J', app.Jkgm2EditField.value);
    assignin('base', 'B', app.BNmsEditField.value);
    assignin('base', 'Fn', app.fnHzEditField.value);

    assignin('base', 'Ruv', app.RuvEditField.value);
    assignin('base', 'Ruw', app.RuwEditField.value);
    assignin('base', 'Rvw', app.RvwEditField.value);
    assignin('base', 'U10', app.U10VEditField.value);
    assignin('base', 'I10', app.I10AEditField.value);
    assignin('base', 'P10', app.P10WEditField.value);
    assignin('base', 'Q10', app.Q10VAEditField.value);
    assignin('base', 'Ucc', app.UccVEditField.value);
    assignin('base', 'Icc', app.IccAEditField.value);
```

Digitalización de un banco de ensayos y comprobación experimental

```
assignin('base','Pcc', app.PccWEditField.Value);
assignin('base','Qcc', app.QccVAEditField.Value);

assignin('base','Rs', app.RsEditField.Value);
assignin('base','Rr', app.RrEditField.Value);
assignin('base','Lm', app.LmHEditField.Value);
assignin('base','Lr', app.LrHEditField.Value);
assignin('base','Ls', app.LsHEditField.Value);
app.ParmetrosmotorpersonalizadoUIFigure.Visible = 'off';
end

% Button pushed function: CancelarButton
function CancelarButtonPushed(app, event)
    assignin("base", "parameters_changed", 0);
    delete(app.ParmetrosmotorpersonalizadoUIFigure);
end
end

% Component initialization
methods (Access = private)

    % Create UIFigure and components
    function createComponents(app)

        % Create ParmetrosmotorpersonalizadoUIFigure and hide until all components
are created
        app.ParmetrosmotorpersonalizadoUIFigure = uifigure('visible', 'off');
        app.ParmetrosmotorpersonalizadoUIFigure.Position = [100 100 640 480];
        app.ParmetrosmotorpersonalizadoUIFigure.Name = 'Parámetros motor
personalizado';
        app.ParmetrosmotorpersonalizadoUIFigure.Icon = 'acm.png';

        % Create ButtonGroup
        app.ButtonGroup = uibuttongroup(app.ParmetrosmotorpersonalizadoUIFigure);
        app.ButtonGroup.SelectionChangedFcn = createCallbackFcn(app,
@ButtonGroupSelectionChanged, true);
        app.ButtonGroup.Position = [23 74 594 289];

        % Create ensayo
        app.ensayo = uiradiobutton(app.ButtonGroup);
        app.ensayo.Text = 'Datos de ensayo';
        app.ensayo.Position = [11 262 112 22];
        app.ensayo.Value = true;

        % Create Circ_Eq
        app.Circ_Eq = uiradiobutton(app.ButtonGroup);
        app.Circ_Eq.Text = 'Parámetros del circuito equivalente';
        app.Circ_Eq.Position = [11 75 210 22];

        % Create RuvLabel
        app.RuvLabel = uilabel(app.ButtonGroup);
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.RuvLabel.HorizontalAlignment = 'right';
app.RuvLabel.Position = [74 208 47 22];
app.RuvLabel.Text = 'Ruv ( $\Omega$ )';

% Create RuvEditField
app.RuvEditField = uieditfield(app.ButtonGroup, 'numeric');
app.RuvEditField.Position = [136 208 42 22];

% Create RuwLabel
app.RuwLabel = uilabel(app.ButtonGroup);
app.RuwLabel.HorizontalAlignment = 'right';
app.RuwLabel.Position = [71 176 50 22];
app.RuwLabel.Text = 'Ruw ( $\Omega$ )';

% Create RuwEditField
app.RuwEditField = uieditfield(app.ButtonGroup, 'numeric');
app.RuwEditField.Position = [136 176 42 22];

% Create RvwLabel
app.RvwLabel = uilabel(app.ButtonGroup);
app.RvwLabel.HorizontalAlignment = 'right';
app.RvwLabel.Position = [72 144 49 22];
app.RvwLabel.Text = 'Rvw ( $\Omega$ )';

% Create RvwEditField
app.RvwEditField = uieditfield(app.ButtonGroup, 'numeric');
app.RvwEditField.Position = [136 144 42 22];

% Create U10VEditFieldLabel
app.U10VEditFieldLabel = uilabel(app.ButtonGroup);
app.U10VEditFieldLabel.HorizontalAlignment = 'right';
app.U10VEditFieldLabel.Position = [222 208 47 22];
app.U10VEditFieldLabel.Text = 'U10 (V)';

% Create U10VEditField
app.U10VEditField = uieditfield(app.ButtonGroup, 'numeric');
app.U10VEditField.Position = [284 208 42 22];

% Create I10AEditFieldLabel
app.I10AEditFieldLabel = uilabel(app.ButtonGroup);
app.I10AEditFieldLabel.HorizontalAlignment = 'right';
app.I10AEditFieldLabel.Position = [227 176 42 22];
app.I10AEditFieldLabel.Text = 'I10 (A)';

% Create I10AEditField
app.I10AEditField = uieditfield(app.ButtonGroup, 'numeric');
app.I10AEditField.Position = [284 176 42 22];

% Create P10WEditFieldLabel
app.P10WEditFieldLabel = uilabel(app.ButtonGroup);
app.P10WEditFieldLabel.HorizontalAlignment = 'right';
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.P10WEditFieldLabel.Position = [219 144 50 22];
app.P10WEditFieldLabel.Text = 'P10 (W)';

% Create P10WEditField
app.P10WEditField = uieditfield(app.ButtonGroup, 'numeric');
app.P10WEditField.Position = [284 144 42 22];

% Create Q10VAEditFieldLabel
app.Q10VAEditFieldLabel = uilabel(app.ButtonGroup);
app.Q10VAEditFieldLabel.HorizontalAlignment = 'right';
app.Q10VAEditFieldLabel.Position = [214 112 55 22];
app.Q10VAEditFieldLabel.Text = 'Q10 (VA)';

% Create Q10VAEditField
app.Q10VAEditField = uieditfield(app.ButtonGroup, 'numeric');
app.Q10VAEditField.Position = [284 112 42 22];

% Create UccVEditFieldLabel
app.UccVEditFieldLabel = uilabel(app.ButtonGroup);
app.UccVEditFieldLabel.HorizontalAlignment = 'right';
app.UccVEditFieldLabel.Position = [378 208 45 22];
app.UccVEditFieldLabel.Text = 'Ucc (V)';

% Create UccVEditField
app.UccVEditField = uieditfield(app.ButtonGroup, 'numeric');
app.UccVEditField.Position = [438 208 42 22];

% Create ICCAEditFieldLabel
app.ICCAEditFieldLabel = uilabel(app.ButtonGroup);
app.ICCAEditFieldLabel.HorizontalAlignment = 'right';
app.ICCAEditFieldLabel.Position = [383 176 40 22];
app.ICCAEditFieldLabel.Text = 'Icc (A)';

% Create ICCAEditField
app.ICCAEditField = uieditfield(app.ButtonGroup, 'numeric');
app.ICCAEditField.Position = [438 176 42 22];

% Create PccWEditFieldLabel
app.PccWEditFieldLabel = uilabel(app.ButtonGroup);
app.PccWEditFieldLabel.HorizontalAlignment = 'right';
app.PccWEditFieldLabel.Position = [375 144 48 22];
app.PccWEditFieldLabel.Text = 'Pcc (W)';

% Create PccWEditField
app.PccWEditField = uieditfield(app.ButtonGroup, 'numeric');
app.PccWEditField.Position = [438 144 42 22];

% Create QccVAEditFieldLabel
app.QccVAEditFieldLabel = uilabel(app.ButtonGroup);
app.QccVAEditFieldLabel.HorizontalAlignment = 'right';
app.QccVAEditFieldLabel.Position = [370 112 53 22];
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.QccVAEditFieldLabel.Text = 'Qcc (VA)';

% Create QccVAEditField
app.QccVAEditField = uieditfield(app.ButtonGroup, 'numeric');
app.QccVAEditField.Position = [438 112 42 22];

% Create ResistenciaestaticaLabel
app.ResistenciaestaticaLabel = uilabel(app.ButtonGroup);
app.ResistenciaestaticaLabel.HorizontalAlignment = 'center';
app.ResistenciaestaticaLabel.FontWeight = 'bold';
app.ResistenciaestaticaLabel.Position = [62 234 127 22];
app.ResistenciaestaticaLabel.Text = 'Resistencia estática';

% Create EnsayodevacoLabel
app.EnsayodevacoLabel = uilabel(app.ButtonGroup);
app.EnsayodevacoLabel.HorizontalAlignment = 'center';
app.EnsayodevacoLabel.FontWeight = 'bold';
app.EnsayodevacoLabel.Position = [210 234 127 22];
app.EnsayodevacoLabel.Text = 'Ensayo de vacío';

% Create EnsayodecortocircuitoLabel
app.EnsayodecortocircuitoLabel = uilabel(app.ButtonGroup);
app.EnsayodecortocircuitoLabel.HorizontalAlignment = 'center';
app.EnsayodecortocircuitoLabel.FontWeight = 'bold';
app.EnsayodecortocircuitoLabel.Position = [355 234 142 22];
app.EnsayodecortocircuitoLabel.Text = 'Ensayo de cortocircuito';

% Create RSEditFieldLabel
app.RSEditFieldLabel = uilabel(app.ButtonGroup);
app.RSEditFieldLabel.HorizontalAlignment = 'right';
app.RSEditFieldLabel.Position = [81 46 40 22];
app.RSEditFieldLabel.Text = 'Rs ( $\Omega$ )';

% Create RSEditField
app.RSEditField = uieditfield(app.ButtonGroup, 'numeric');
app.RSEditField.Enable = 'off';
app.RSEditField.Position = [136 46 42 22];

% Create LmHEditFieldLabel
app.LmHEditFieldLabel = uilabel(app.ButtonGroup);
app.LmHEditFieldLabel.HorizontalAlignment = 'right';
app.LmHEditFieldLabel.Position = [79 14 42 22];
app.LmHEditFieldLabel.Text = 'Lm (H)';

% Create LmHEditField
app.LmHEditField = uieditfield(app.ButtonGroup, 'numeric');
app.LmHEditField.Enable = 'off';
app.LmHEditField.Position = [136 14 42 22];

% Create RREditFieldLabel
app.RREditFieldLabel = uilabel(app.ButtonGroup);
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.RrEditFieldLabel.HorizontalAlignment = 'right';
app.RrEditFieldLabel.Position = [231 46 38 22];
app.RrEditFieldLabel.Text = 'Rr ( $\Omega$ )';

% Create RrEditField
app.RrEditField = uieditfield(app.ButtonGroup, 'numeric');
app.RrEditField.Enable = 'off';
app.RrEditField.Position = [284 46 42 22];

% Create LrEditFieldLabel
app.LrEditFieldLabel = uilabel(app.ButtonGroup);
app.LrEditFieldLabel.HorizontalAlignment = 'right';
app.LrEditFieldLabel.Position = [233 14 36 22];
app.LrEditFieldLabel.Text = 'Lr (H)';

% Create LrEditField
app.LrEditField = uieditfield(app.ButtonGroup, 'numeric');
app.LrEditField.Enable = 'off';
app.LrEditField.Position = [284 14 42 22];

% Create LsEditFieldLabel
app.LsEditFieldLabel = uilabel(app.ButtonGroup);
app.LsEditFieldLabel.HorizontalAlignment = 'right';
app.LsEditFieldLabel.Position = [385 14 38 22];
app.LsEditFieldLabel.Text = 'Ls (H)';

% Create LsEditField
app.LsEditField = uieditfield(app.ButtonGroup, 'numeric');
app.LsEditField.Enable = 'off';
app.LsEditField.Position = [438 14 42 22];

% Create AceptarButton
app.AceptarButton = uibutton(app.ParmetrosmotorpersonalizadoUIFigure,
'push');
app.AceptarButton.ButtonPushedFcn = createCallbackFcn(app,
@AceptarButtonPushed, true);
app.AceptarButton.Position = [208 28 100 22];
app.AceptarButton.Text = 'Aceptar';

% Create CancelarButton
app.CancelarButton = uibutton(app.ParmetrosmotorpersonalizadoUIFigure,
'push');
app.CancelarButton.ButtonPushedFcn = createCallbackFcn(app,
@CancelarButtonPushed, true);
app.CancelarButton.Position = [331 28 100 22];
app.CancelarButton.Text = 'Cancelar';

% Create MOTORDEINDUCCINPERSONALIZADOLabel
app.MOTORDEINDUCCINPERSONALIZADOLabel =
uilabel(app.ParmetrosmotorpersonalizadoUIFigure);
app.MOTORDEINDUCCINPERSONALIZADOLabel.FontSize = 15;
```

Digitalización de un banco de ensayos y comprobación experimental

```
app.MOTORDEINDUCCINPERSONALIZADOLabel.FontWeight = 'bold';
app.MOTORDEINDUCCINPERSONALIZADOLabel.Position = [163 443 313 30];
app.MOTORDEINDUCCINPERSONALIZADOLabel.Text = 'MOTOR DE INDUCCIÓN
PERSONALIZADO';

% Create ParesdepolosEditFieldLabel
app.ParesdepolosEditFieldLabel =
uilabel(app.ParmetrosmotorpersonalizadoUIFigure);
app.ParesdepolosEditFieldLabel.HorizontalAlignment = 'right';
app.ParesdepolosEditFieldLabel.Position = [48 404 86 22];
app.ParesdepolosEditFieldLabel.Text = 'Pares de polos';

% Create ParesdepolosEditField
app.ParesdepolosEditField =
uieditfield(app.ParmetrosmotorpersonalizadoUIFigure, 'numeric');
app.ParesdepolosEditField.Position = [149 404 42 22];

% Create PnWEditFieldLabel
app.PnWEditFieldLabel = uilabel(app.ParmetrosmotorpersonalizadoUIFigure);
app.PnWEditFieldLabel.HorizontalAlignment = 'right';
app.PnWEditFieldLabel.Position = [208 404 43 22];
app.PnWEditFieldLabel.Text = 'Pn (w)';

% Create PnWEditField
app.PnWEditField = uieditfield(app.ParmetrosmotorpersonalizadoUIFigure,
'numeric');
app.PnWEditField.Position = [266 404 42 22];

% Create ULLVEditFieldLabel
app.ULLVEditFieldLabel = uilabel(app.ParmetrosmotorpersonalizadoUIFigure);
app.ULLVEditFieldLabel.HorizontalAlignment = 'right';
app.ULLVEditFieldLabel.Position = [326 404 47 22];
app.ULLVEditFieldLabel.Text = 'ULL (v)';

% Create ULLVEditField
app.ULLVEditField = uieditfield(app.ParmetrosmotorpersonalizadoUIFigure,
'numeric');
app.ULLVEditField.Position = [388 404 42 22];

% Create Jkgm2EditFieldLabel
app.Jkgm2EditFieldLabel = uilabel(app.ParmetrosmotorpersonalizadoUIFigure);
app.Jkgm2EditFieldLabel.HorizontalAlignment = 'right';
app.Jkgm2EditFieldLabel.Position = [452 404 62 22];
app.Jkgm2EditFieldLabel.Text = 'J (kg·m^2)';

% Create Jkgm2EditField
app.Jkgm2EditField = uieditfield(app.ParmetrosmotorpersonalizadoUIFigure,
'numeric');
app.Jkgm2EditField.Position = [529 404 42 22];

% Create BNmsEditFieldLabel
```


Digitalización de un banco de ensayos y comprobación experimental

```
app.BNmsEditFieldLabel = uilabel(app.ParmetrosmotorpersonalizadoUIFigure);
app.BNmsEditFieldLabel.HorizontalAlignment = 'right';
app.BNmsEditFieldLabel.Position = [77 373 57 22];
app.BNmsEditFieldLabel.Text = 'B (N·m·s)';

% Create BNmsEditField
app.BNmsEditField = uicontrol(app.ParmetrosmotorpersonalizadoUIFigure,
'numeric');
app.BNmsEditField.Position = [149 373 42 22];

% Create fnHzEditFieldLabel
app.fnHzEditFieldLabel = uilabel(app.ParmetrosmotorpersonalizadoUIFigure);
app.fnHzEditFieldLabel.HorizontalAlignment = 'right';
app.fnHzEditFieldLabel.Position = [210 373 41 22];
app.fnHzEditFieldLabel.Text = 'fn (Hz)';

% Create fnHzEditField
app.fnHzEditField = uicontrol(app.ParmetrosmotorpersonalizadoUIFigure,
'numeric');
app.fnHzEditField.Position = [266 373 42 22];

% Show the figure after all components are created
app.ParmetrosmotorpersonalizadoUIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = INTERF_PERSONALIZ

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.ParmetrosmotorpersonalizadoUIFigure)

% Execute the startup function
runStartupFcn(app, @startupFcn)
if nargin == 0
    clear app
end
end

% Code that executes before app deletion
function delete(app)
% Delete UIFigure when app is deleted
delete(app.ParmetrosmotorpersonalizadoUIFigure)
end
end
end
```



ANEXO 4 – CÓDIGO DE ESP32 WROVER – B



Digitalización de un banco de ensayos y comprobación experimental

```
#include <nhdlcd.hpp> // Librería LCD
#include <EEPROM.h>
#include "BluetoothSerial.h"

#define SIZE_ARRAY 1000

const byte dataPin = 13; // Pin data LCD
const byte clockPin = 14; // Pin clock LCD
const byte slaveSelectPin = 25; // Pin slave LCD
nhdlcd lcd(dataPin, clockPin, slaveSelectPin); // Inicializa LCD
int pinFreq = 32; // Pin de entrada de
señal a pulsos
int pinSelEncoder = 22;
int pinSelOptico = 23;
int pinSelClamp = 18;
int pinSelHall = 19;

float frecuencia = 0, frec_it = 0;
int flancos = 0; // Frecuencia
inicial
float Tmedicion = 1000; // Tiempo de
medición de frecuencia [s]
int num_mediciones = 2;
float resolucion = 0; // Resolución
del encoder [bits]
String f1; // Declarado global
sólo para ajustar texto en LCD
float ajusteVelocidad; // Factor a
multiplicar a la velocidad
float aux, aux1 = 0;
float vel;

// PARÁMETROS MOTOR
float vel_nominal = 1500;

// PARÁMETROS PARA LECTURA DE PAR
int pinParClamp = 0, n_lecturas = 1000, pinParHall = 4;
float suma_tensionClamp = 0, par = 0;
double tension_rms = 0, suma_tensionHall = 0, tensionHall[SIZE_ARRAY],
tensionClamp[SIZE_ARRAY];
int tension_array[1000];
float conversion_corriente_par = 1.076, volt2curr = 10;

// PARÁMETROS PARA AJUSTE DE PINZA
float pc1 = 5, oc1 = 14.3, pc2 = 1.439, oc2 = 3.85, pc3 = 1.439, oc3 =
4;

// PARÁMETROS PARA AJUSTE DE HALL
float ph1 = 6.175, oh1 = 7.1, ph2 = 6.175, oh2 = 7, ph3 = 6.175, oh3 =
7, ph4 = 6.175, oh4 = 7.15;

// PARÁMETROS SALIDA 0 - 10 V
int pinOutVel = 26;

// RESET
```



Digitalización de un banco de ensayos y comprobación experimental

```
int pinReset = 5;
float try_reset = 0;

// CONTROL HALL - PINZA
int pinOutClamp = 15;
int pinOutHall = 2;

// TIMER
hw_timer_t * timer = NULL;
volatile SemaphoreHandle_t timerSemaphore;
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;
int buff = 0, buffer_full = 0;

// DOBLE NUCLEO
TaskHandle_t Task1;

// PARAMETROS DE AJUSTE
//int resEnc = 1024, resOpt = 4;
//float kHall = 1, kPinza = 1;
int resEnc, resOpt;
float kHall, kPinza;

// BLUETOOTH

#ifdef CONFIG_BT_ENABLED || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and
enable it
#endif

BluetoothSerial SerialBT;

String receivedData;
int availableToChangeData = 0;
String dataSent;

void IRAM_ATTR mideTension() {
    //String taskMessage = "Task running on core ";
    portENTER_CRITICAL_ISR(&timerMux);
    //taskMessage = taskMessage + xPortGetCoreID();
    tensionHall[buff] = analogRead(pinParHall) * 3.3 / 4095;
    tensionClamp[buff] = analogRead(pinParClamp) * 3.3 / 4095;
    buff++;
    if (buff >= SIZE_ARRAY) {
        buff = 0;
        buffer_full = 1;
    }
    portEXIT_CRITICAL_ISR(&timerMux);
    xSemaphoreGiveFromISR(timerSemaphore, NULL);
}

void sendData(void * pvParameters) {
    for (;;) {
        dataSent = "";
        dataSent.concat(String(vel));
    }
}
```

Digitalización de un banco de ensayos y comprobación experimental

```
dataSent.concat("|");
dataSent.concat(String(par));
dataSent.concat("|");

if (receivedData == "getParam") {
    sendParam();
    receivedData = "";
} else {
    SerialBT.print(dataSent);
}
if (receivedData == "runTimer") {
    timerAlarmEnable(timer);
    attachInterrupt(digitalPinToInterrupt(pinFreq), mideFrecuencia,
RISING);
    availableToChangeData = 0;
    receivedData = "";
}

if (SerialBT.available()) {
    receivedData = SerialBT.readString();
    if (receivedData == "modifyParameters") {
        timerAlarmDisable(timer);
        detachInterrupt(digitalPinToInterrupt(pinFreq));
        availableToChangeData = 1;
    }
    if (availableToChangeData == 1) {
        if (receivedData.charAt(0) == 'a') {
            receivedData.remove(0, 1);
            resEnc = receivedData.toInt();
            writeEPROM(resEnc, resOpt, kHall, kPinza);
            receivedData = "";
        }
        if (receivedData.charAt(0) == 'b') {
            receivedData.remove(0, 1);
            resOpt = receivedData.toInt();
            writeEPROM(resEnc, resOpt, kHall, kPinza);
            receivedData = "";
        }
        if (receivedData.charAt(0) == 'c') {
            receivedData.remove(0, 1);
            kHall = receivedData.toFloat();
            writeEPROM(resEnc, resOpt, kHall, kPinza);
            receivedData = "";
        }
        if (receivedData.charAt(0) == 'd') {
            receivedData.remove(0, 1);
            kPinza = receivedData.toFloat();
            writeEPROM(resEnc, resOpt, kHall, kPinza);
            receivedData = "";
        }
    }
}

if (buffer_full == 1) {
    printOnLCD();
}
```



Digitalización de un banco de ensayos y comprobación experimental

```
    }
    if (try_reset == HIGH) {
        tryReset();
    }
    readEPROM(&resEnc, &resOpt, &kHall, &kPinza);
    delay(Tmediccion);
}

vTaskDelete(NULL);
}

void setup() {
    Serial.begin(115200);

    if (!EEPROM.begin(128)) /*Tamaño es 4 de 32 bits*/
    {
        Serial.println("Failed to initialise EEPROM");
        Serial.println("Restarting...");
        delay(1000);
    }

    SerialBT.begin("PUESTO 6"); //Bluetooth device name

    pinMode(pinFreq, INPUT);
    pinMode(pinSelEncoder, INPUT);
    pinMode(pinSelEncoder, INPUT);
    pinMode(pinParClamp, INPUT);
    pinMode(pinOutVel, OUTPUT);
    pinMode(pinReset, INPUT);
    pinMode(pinOutClamp, OUTPUT);
    pinMode(pinOutHall, OUTPUT);

    xTaskCreatePinnedToCore(sendData, "Task1", 10000, NULL, 0, &Task1,
0);

    timerSemaphore = xSemaphoreCreateBinary();

    // Crear timer
    timer = timerBegin(2, 3200, true);

    // Interrupción del timer
    timerAttachInterrupt(timer, &mideTension, true);

    // Tiempo de alarma de llamada a la funcion (en microsegundos)
    timerAlarmWrite(timer, 1000, true);

    // Habilitar alarmas
    timerAlarmEnable(timer);

    attachInterrupt(digitalPinToInterrupt(pinFreq), mideFrecuencia,
RISING);

    //writeEPROM(resEnc, resOpt, kHall, kPinza);
```



Digitalización de un banco de ensayos y comprobación experimental

```
    readEPROM(&resEnc, &resOpt, &kHall, &kPinza);
}

void loop() {

    if (digitalRead(pinSelEncoder) == HIGH) {

        resolucion = resEnc;
        ajusteVelocidad = 0.9853;
        if (digitalRead(pinSelClamp) == HIGH) {
            digitalWrite(pinOutHall, LOW);
            digitalWrite(pinOutClamp, HIGH);
            readTorqueClamp();
        } else if (digitalRead(pinSelHall) == HIGH) {
            digitalWrite(pinOutClamp, LOW);
            digitalWrite(pinOutHall, HIGH);
            readTorqueHall();
        }
        frecuencia = float(flancos) * 1000 / float(Tmedicion);
        flancos = 0;
        //Serial.println("ENCODER");
    } else if (digitalRead(pinSelOptico) == HIGH) {
        resolucion = resOpt;
        ajusteVelocidad = 1.0;
        if (digitalRead(pinSelClamp) == HIGH) {
            readTorqueClamp();
        } else if (digitalRead(pinSelHall) == HIGH) {
            readTorqueHall();
        }
        for (int i = 0; i < num_mediciones; i++) {
            delay(Tmedicion / num_mediciones);
            frec_it = float(flancos) * 1000.0 / (Tmedicion /
float(num_mediciones));
            aux1 = aux1 + frec_it;
            flancos = 0;
        }
        frecuencia = aux1 / float(num_mediciones);
        Serial.println("OPTICO");
    }
    try_reset = digitalRead(pinReset);

    delay(Tmedicion);
    //Serial.println(frecuencia);
    aux1 = 0;
    frecuencia = 0;
}

void mideFrecuencia() {
    flancos++;
}

void printTorque() { // Función para imprimir
    la frecuencia en el LCD
    ajustarFila2();
```



Digitalización de un banco de ensayos y comprobación experimental

```
// Freq -> int to char
aux = frecuencia;
String freq = String(par);
char freq1[80];
String f2 = ajustarSeleccion();
f1.concat(freq);
f1.concat(f2);
f1.toCharArray(freq1, 80);
// Print frequency ("Frecuencia: XXXXXHz")
lcd.write(freq1);

// Print on Serial
//Serial.println(frecuencia);
}

void printVel() { // Función para imprimir la
velocidad en LCD
    vel = (60.0 * aux) / resolucion; // Calcula velocidad a partir de
frecuencia
    // Vel -> int to char
    String velchar = String(vel);
    char vell[40];
    String v1 = "Speed: ";
    String v2 = " rpm";
    v1.concat(velchar);
    v1.concat(v2);
    v1.toCharArray(vell, 40);
    // Print speed ("Velocidad: XXXXrpm")
    lcd.write(vell);
    if ( vel <= 1500) {
        digitalWrite(pinOutVel, vel * 255 / vel_nominal);
    } else {
        digitalWrite(pinOutVel, 255);
    }
    // Print on Serial
    //Serial.println(vel);
}

void printOnLCD() {
    lcd.write(Command::ClearScreen); // Borrar datos de pantalla
    printVel();
    printTorque();
}

void ajustarFila2() { // Función para ajustar la
segunda fila de la pantalla
    if (vel < 0) {
        f1 = " Torque: ";
    } else if (vel < 10) {
        f1 = " Torque: ";
    } else if (vel < 100) {
        f1 = " Torque: ";
    } else if (vel < 1000) {
        f1 = " Torque: ";
    } else if (vel < 10000) {
```


Digitalización de un banco de ensayos y comprobación experimental

```
    fl = " Torque: ";
} else if (vel < 100000) {
    fl = " Torque: ";
} else if (vel < 1000000) {
    fl = "Torque: ";
}
}

String ajustarSeleccion() {
    String ajuste;
    if (par < 10) {
        if (digitalRead(pinSelClamp) == HIGH) { // Clamp
            if (digitalRead(pinSelEncoder) == HIGH) {
                // Encoder + Clamp
                ajuste = " Nm          |Encoder|          |Clamp|";
            } else if (digitalRead(pinSelOptico) == HIGH) {
                // Optico + Clamp
                ajuste = " Nm          |Optico|          |Clamp|";
            }
        } else if (digitalRead(pinSelHall) == HIGH) { // Hall
            if (digitalRead(pinSelEncoder) == HIGH) {
                // Encoder + Hall
                ajuste = " Nm          |Encoder|          |Hall|";
            } else if (digitalRead(pinSelOptico) == HIGH) {
                // Optico + Hall
                ajuste = " Nm          |Optico|          |Hall|";
            }
        }
    } else {
        if (digitalRead(pinSelClamp) == HIGH) { // Clamp
            if (digitalRead(pinSelEncoder) == HIGH) {
                // Encoder + Clamp
                ajuste = " Nm          |Encoder|          |Clamp|";
            } else if (digitalRead(pinSelOptico) == HIGH) {
                // Optico + Clamp
                ajuste = " Nm          |Optico|          |Clamp|";
            }
        } else if (digitalRead(pinSelHall) == HIGH) { // Hall
            if (digitalRead(pinSelEncoder) == HIGH) {
                // Encoder + Hall
                ajuste = " Nm          |Encoder|          |Hall|";
            } else if (digitalRead(pinSelOptico) == HIGH) {
                // Optico + Hall
                ajuste = " Nm          |Optico|          |Hall|";
            }
        }
    }
}

return ajuste;
}

void readTorqueClamp() {

    ////////////////////////////////////LECTURA DE PAR////////////////////////////////////
}
```

Digitalización de un banco de ensayos y comprobación experimental

```
if (buffer_full) {
    for (int h = 0; h < n_lecturas; h++) {
        suma_tensionClamp = suma_tensionClamp + tensionClamp[h] *
tensionClamp[h];
    }
    tension_rms = sqrt(suma_tensionClamp / float(n_lecturas));
    //Serial.println("RMS CLAMP: ");
    //Serial.println(tension_rms);
    suma_tensionClamp = 0;

    par = tension_rms * volt2curr * conversion_corriente_par * kPinza;
}

void readTorqueHall() {
    ////////////////////////////////////LECTURA DE PAR RMS////////////////////////////////////
    if (buffer_full) {
        for (int k = 0; k < n_lecturas; k++) {
            suma_tensionHall = suma_tensionHall + tensionHall[k] *
tensionHall[k];
        }
        tension_rms = sqrt(suma_tensionHall / double(n_lecturas));
        suma_tensionHall = 0;
    }

    par = tension_rms * volt2curr * conversion_corriente_par * kHall;
}

void tryReset() {
    lcd.write(Command::ClearScreen);           // Borrar datos de pantalla
    int tdelay = 200;
    // Animacion 2
    lcd.write("    Resetting.    ");
    delay(tdelay);
    lcd.write("        ..        ");
    delay(tdelay);
    lcd.write(Command::ClearScreen);           // Borrar datos de pantalla
    lcd.write("    Resetting.    ");
    delay(tdelay);
    lcd.write("        ....        ");
    delay(tdelay);
    lcd.write(Command::ClearScreen);           // Borrar datos de pantalla
    lcd.write("    Resetting.    ");
    delay(tdelay);
    lcd.write("        .....        ");
    delay(tdelay);
    lcd.write(Command::ClearScreen);           // Borrar datos de pantalla
    lcd.write("    Resetting.    ");
    delay(tdelay);
    lcd.write("        .....        ");
    delay(tdelay);
    lcd.write(Command::ClearScreen);           // Borrar datos de pantalla
    lcd.write("    Resetting.    ");
    delay(tdelay);
}
```

Digitalización de un banco de ensayos y comprobación experimental

```
lcd.write(" ..... ");
delay(tdelay);
lcd.write(Command::ClearScreen); // Borrar datos de pantalla
lcd.write(" Resetting. ");
delay(tdelay);
lcd.write(" ..... ");
delay(tdelay);
lcd.write(Command::ClearScreen); // Borrar datos de pantalla
lcd.write(" Resetting. ");
delay(tdelay);
lcd.write(" ..... ");
delay(tdelay);
lcd.write(Command::ClearScreen); // Borrar datos de pantalla
lcd.write(" Resetting. ");
delay(tdelay);
lcd.write(" ..... ");
delay(tdelay);
lcd.write(Command::ClearScreen); // Borrar datos de pantalla
lcd.write(" Resetting. ");
delay(tdelay);
lcd.write(" ..... ");
delay(tdelay);
lcd.write(Command::ClearScreen); // Borrar datos de pantalla
lcd.write(" Resetting. ");
delay(tdelay);
lcd.write(" .. ");
delay(tdelay);
lcd.write(Command::ClearScreen); // Borrar datos de pantalla
lcd.write(" Resetting. ");

ESP.restart();
}

void writeEPROM(int resEnc, int resOpt, float kHall, float kPinza) {
int address = 0;
Serial.println("1");
EEPROM.writeInt(address, resEnc);
address += sizeof(int);
Serial.println("2");
EEPROM.writeInt(address, resOpt);
address += sizeof(int);
Serial.println("3");
EEPROM.writeFloat(address, kHall);
address += sizeof(float);
Serial.println("4");
EEPROM.writeFloat(address, kPinza);
address += sizeof(float);
Serial.println("55");
EEPROM.commit();
Serial.println("6");
}
```



Digitalización de un banco de ensayos y comprobación experimental

```
}  
  
void readEPROM(int* resEnc, int* resOpt, float* kHall, float* kPinza)  
{  
    int address = 0;  
    Serial.println("1");  
    resEnc[0] = EEPROM.readInt(address);  
    address += sizeof(int);  
    Serial.println("2");  
    resOpt[0] = EEPROM.readInt(address);  
    address += sizeof(int);  
    Serial.println("3");  
    kHall[0] = EEPROM.readFloat(address);  
    address += sizeof(float);  
    Serial.println("4");  
    kPinza[0] = EEPROM.readFloat(address);  
    address += sizeof(float);  
    Serial.println("5");  
}  
  
void sendParam() {  
    readEPROM(&resEnc, &resOpt, &kHall, &kPinza);  
    dataSent.concat(String(resEnc));  
    dataSent.concat("|");  
    dataSent.concat(String(resOpt));  
    dataSent.concat("|");  
    dataSent.concat(String(kHall));  
    dataSent.concat("|");  
    dataSent.concat(String(kPinza));  
    dataSent.concat("|");  
    SerialBT.print(dataSent);  
}
```



ANEXO 5 – CÓDIGO DE APP ANDROID



Digitalización de un banco de ensayos y comprobación experimental

```
package appinventor.ai_adrian_leordean.Parametros_v1_copy;

import android.os.Bundle;
import androidx.fragment.app.FragmentTransaction;
import com.google.appinventor.components.common.PropertyTypeConstants;
import
com.google.appinventor.components.runtime.AppInventorCompatActivity;
import com.google.appinventor.components.runtime.BluetoothClient;
import com.google.appinventor.components.runtime.Button;
import com.google.appinventor.components.runtime.Clock;
import com.google.appinventor.components.runtime.Component;
import com.google.appinventor.components.runtime.EventDispatcher;
import com.google.appinventor.components.runtime.Form;
import
com.google.appinventor.components.runtime.HandlesEventDispatching;
import
com.google.appinventor.components.runtime.HorizontalArrangement;
import com.google.appinventor.components.runtime.Label;
import com.google.appinventor.components.runtime.ListPicker;
import com.google.appinventor.components.runtime.Switch;
import com.google.appinventor.components.runtime.TextBox;
import com.google.appinventor.components.runtime.VerticalArrangement;
import
com.google.appinventor.components.runtime.errors.PermissionException;
import
com.google.appinventor.components.runtime.errors.YailRuntimeError;
import com.google.appinventor.components.runtime.util.RetValManager;
import
com.google.appinventor.components.runtime.util.RuntimeErrorAlert;
import com.google.youngandroid.runtime;
import gnu.expr.Language;
import gnu.expr.ModuleBody;
import gnu.expr.ModuleInfo;
import gnu.expr.ModuleMethod;
import gnu.kawa.functions.Apply;
import gnu.kawa.functions.Format;
import gnu.kawa.functions.GetNamedPart;
import gnu.kawa.functions.IsEqual;
import gnu.kawa.reflect.Invoke;
import gnu.kawa.reflect.SlotGet;
import gnu.kawa.reflect.SlotSet;
import gnu.lists.Consumer;
import gnu.lists.FString;
import gnu.lists.LList;
import gnu.lists.Pair;
import gnu.lists.PairWithPosition;
import gnu.lists.VoidConsumer;
import gnu.mapping.CallContext;
import gnu.mapping.Environment;
import gnu.mapping.SimpleSymbol;
import gnu.mapping.Symbol;
import gnu.mapping.Values;
import gnu.mapping.WrongType;
import gnu.math.IntNum;
import kawa.lang.Promise;
```

Digitalización de un banco de ensayos y comprobación experimental

```
import kawa.lib.lists;
import kawa.lib.misc;
import kawa.lib.strings;
import kawa.standard.Scheme;
import kawa.standard.require;

/* compiled from: Screen1.yail */
public class Screen1 extends Form implements Runnable {
    static final SimpleSymbol Lit0 = ((SimpleSymbol) new
SimpleSymbol("Screen1").readResolve());
    static final SimpleSymbol Lit1 = ((SimpleSymbol) new
SimpleSymbol("getMessage").readResolve());
    static final SimpleSymbol Lit10 = ((SimpleSymbol) new
SimpleSymbol("ScreenOrientation").readResolve());
    static final PairWithPosition Lit100 = PairWithPosition.make(Lit8,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 585896);
    static final SimpleSymbol Lit101 = ((SimpleSymbol) new
SimpleSymbol("Button7$Click").readResolve());
    static final FString Lit102 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
    static final SimpleSymbol Lit103 = ((SimpleSymbol) new
SimpleSymbol("HorizontalArrangement2").readResolve());
    static final IntNum Lit104 = IntNum.make(-1010);
    static final FString Lit105 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
    static final FString Lit106 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit107 = ((SimpleSymbol) new
SimpleSymbol("Label2").readResolve());
    static final IntNum Lit108 = IntNum.make(-1035);
    static final FString Lit109 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit11 = ((SimpleSymbol) new
SimpleSymbol("ShowListsAsJson").readResolve());
    static final FString Lit110 = new
FString("com.google.appinventor.components.runtime.TextBox");
    static final SimpleSymbol Lit111 = ((SimpleSymbol) new
SimpleSymbol("TextBox2").readResolve());
    static final IntNum Lit112 = IntNum.make(-1030);
    static final FString Lit113 = new
FString("com.google.appinventor.components.runtime.TextBox");
    static final FString Lit114 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit115 = ((SimpleSymbol) new
SimpleSymbol("Label12").readResolve());
    static final FString Lit116 = new
FString("com.google.appinventor.components.runtime.Label");
    static final FString Lit117 = new
FString("com.google.appinventor.components.runtime.Button");
    static final FString Lit118 = new
FString("com.google.appinventor.components.runtime.Button");
```



Digitalización de un banco de ensayos y comprobación experimental

```
static final PairWithPosition Lit119 = PairWithPosition.make(Lit8,
PairWithPosition.make(Lit8, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 794799),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 794793);
static final SimpleSymbol Lit12 = ((SimpleSymbol) new
SimpleSymbol(PropertyTypeConstants.PROPERTY_TYPE_BOOLEAN).readResolve(
));
static final PairWithPosition Lit120 = PairWithPosition.make(Lit8,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 794816);
static final SimpleSymbol Lit121 = ((SimpleSymbol) new
SimpleSymbol("Button2$Click").readResolve());
static final FString Lit122 = new
FString("com.google.appinventor.components.runtime.Label");
static final SimpleSymbol Lit123 = ((SimpleSymbol) new
SimpleSymbol("Label17").readResolve());
static final FString Lit124 = new
FString("com.google.appinventor.components.runtime.Label");
static final FString Lit125 = new
FString("com.google.appinventor.components.runtime.Button");
static final IntNum Lit126 = IntNum.make(-1004);
static final IntNum Lit127 = IntNum.make(-1005);
static final FString Lit128 = new
FString("com.google.appinventor.components.runtime.Button");
static final PairWithPosition Lit129 = PairWithPosition.make(Lit8,
PairWithPosition.make(Lit8, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 884884),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 884878);
static final SimpleSymbol Lit13 = ((SimpleSymbol) new
SimpleSymbol("Sizing").readResolve());
static final PairWithPosition Lit130 = PairWithPosition.make(Lit8,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 884901);
static final SimpleSymbol Lit131 = ((SimpleSymbol) new
SimpleSymbol("Button8$Click").readResolve());
static final FString Lit132 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
static final SimpleSymbol Lit133 = ((SimpleSymbol) new
SimpleSymbol("HorizontalArrangement1").readResolve());
static final IntNum Lit134 = IntNum.make(-1010);
static final FString Lit135 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
static final FString Lit136 = new
```




Digitalización de un banco de ensayos y comprobación experimental

```
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit137 = ((SimpleSymbol) new
SimpleSymbol("Label13").readResolve());
    static final IntNum Lit138 = IntNum.make(-1035);
    static final FString Lit139 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit14 = ((SimpleSymbol) new
SimpleSymbol("Title").readResolve());
    static final FString Lit140 = new
FString("com.google.appinventor.components.runtime.TextBox");
    static final SimpleSymbol Lit141 = ((SimpleSymbol) new
SimpleSymbol("TextBox3").readResolve());
    static final IntNum Lit142 = IntNum.make(-1030);
    static final FString Lit143 = new
FString("com.google.appinventor.components.runtime.TextBox");
    static final FString Lit144 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit145 = ((SimpleSymbol) new
SimpleSymbol("Label13").readResolve());
    static final FString Lit146 = new
FString("com.google.appinventor.components.runtime.Label");
    static final FString Lit147 = new
FString("com.google.appinventor.components.runtime.Button");
    static final FString Lit148 = new
FString("com.google.appinventor.components.runtime.Button");
    static final PairWithPosition Lit149 = PairWithPosition.make(Lit8,
PairWithPosition.make(Lit8, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 1093807),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 1093801);
    static final SimpleSymbol Lit15 = ((SimpleSymbol) new
SimpleSymbol("TitleVisible").readResolve());
    static final PairWithPosition Lit150 = PairWithPosition.make(Lit8,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 1093824);
    static final SimpleSymbol Lit151 = ((SimpleSymbol) new
SimpleSymbol("Button1$Click").readResolve());
    static final FString Lit152 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit153 = ((SimpleSymbol) new
SimpleSymbol("Label18").readResolve());
    static final FString Lit154 = new
FString("com.google.appinventor.components.runtime.Label");
    static final FString Lit155 = new
FString("com.google.appinventor.components.runtime.Button");
    static final IntNum Lit156 = IntNum.make(-1004);
    static final IntNum Lit157 = IntNum.make(-1005);
    static final FString Lit158 = new
FString("com.google.appinventor.components.runtime.Button");
    static final PairWithPosition Lit159 = PairWithPosition.make(Lit8,
PairWithPosition.make(Lit8, LList.Empty,
```



Digitalización de un banco de ensayos y comprobación experimental

```
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 1183892),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 1183886);
    static final SimpleSymbol Lit16 = ((SimpleSymbol) new
SimpleSymbol("Button3").readResolve());
    static final PairWithPosition Lit160 = PairWithPosition.make(Lit8,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 1183909);
    static final SimpleSymbol Lit161 = ((SimpleSymbol) new
SimpleSymbol("Button9$Click").readResolve());
    static final FString Lit162 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
    static final SimpleSymbol Lit163 = ((SimpleSymbol) new
SimpleSymbol("HorizontalArrangement4").readResolve());
    static final IntNum Lit164 = IntNum.make(-1010);
    static final FString Lit165 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
    static final FString Lit166 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit167 = ((SimpleSymbol) new
SimpleSymbol("Label4").readResolve());
    static final IntNum Lit168 = IntNum.make(-1035);
    static final FString Lit169 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit17 = ((SimpleSymbol) new
SimpleSymbol("Enabled").readResolve());
    static final FString Lit170 = new
FString("com.google.appinventor.components.runtime.TextBox");
    static final SimpleSymbol Lit171 = ((SimpleSymbol) new
SimpleSymbol("TextBox4").readResolve());
    static final IntNum Lit172 = IntNum.make(-1030);
    static final FString Lit173 = new
FString("com.google.appinventor.components.runtime.TextBox");
    static final FString Lit174 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit175 = ((SimpleSymbol) new
SimpleSymbol("Label14").readResolve());
    static final FString Lit176 = new
FString("com.google.appinventor.components.runtime.Label");
    static final FString Lit177 = new
FString("com.google.appinventor.components.runtime.Button");
    static final FString Lit178 = new
FString("com.google.appinventor.components.runtime.Button");
    static final PairWithPosition Lit179 = PairWithPosition.make(Lit8,
PairWithPosition.make(Lit8, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 1392815),
"/tmp/1626207119500_0.7964086826725233-
```



Digitalización de un banco de ensayos y comprobación experimental

```
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 1392809);
    static final SimpleSymbol Lit18 = ((SimpleSymbol) new
SimpleSymbol("Button2").readResolve());
    static final PairWithPosition Lit180 = PairWithPosition.make(Lit8,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 1392832);
    static final SimpleSymbol Lit181 = ((SimpleSymbol) new
SimpleSymbol("Button4$Click").readResolve());
    static final FString Lit182 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit183 = ((SimpleSymbol) new
SimpleSymbol("Label19").readResolve());
    static final FString Lit184 = new
FString("com.google.appinventor.components.runtime.Label");
    static final FString Lit185 = new
FString("com.google.appinventor.components.runtime.Button");
    static final IntNum Lit186 = IntNum.make(-1004);
    static final IntNum Lit187 = IntNum.make(-1005);
    static final FString Lit188 = new
FString("com.google.appinventor.components.runtime.Button");
    static final PairWithPosition Lit189 = PairWithPosition.make(Lit8,
PairWithPosition.make(Lit8, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 1482900),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 1482894);
    static final SimpleSymbol Lit19 = ((SimpleSymbol) new
SimpleSymbol("Button1").readResolve());
    static final PairWithPosition Lit190 = PairWithPosition.make(Lit8,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 1482917);
    static final SimpleSymbol Lit191 = ((SimpleSymbol) new
SimpleSymbol("Button10$Click").readResolve());
    static final FString Lit192 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
    static final SimpleSymbol Lit193 = ((SimpleSymbol) new
SimpleSymbol("HorizontalArrangement8").readResolve());
    static final FString Lit194 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
    static final FString Lit195 = new
FString("com.google.appinventor.components.runtime.Button");
    static final SimpleSymbol Lit196 = ((SimpleSymbol) new
SimpleSymbol("Button5").readResolve());
    static final FString Lit197 = new
FString("com.google.appinventor.components.runtime.Button");
    static final PairWithPosition Lit198 = PairWithPosition.make(Lit8,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
```



Digitalización de un banco de ensayos y comprobación experimental

```
_v1_copy/Screen1.yail", 1540185);
    static final SimpleSymbol Lit199 = ((SimpleSymbol) new
SimpleSymbol("Button5$Click").readResolve());
    static final SimpleSymbol Lit2 = ((SimpleSymbol) new
SimpleSymbol("*the-null-value*").readResolve());
    static final SimpleSymbol Lit20 = ((SimpleSymbol) new
SimpleSymbol("Button4").readResolve());
    static final FString Lit200 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit201 = ((SimpleSymbol) new
SimpleSymbol("Label15").readResolve());
    static final FString Lit202 = new
FString("com.google.appinventor.components.runtime.Label");
    static final FString Lit203 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
    static final SimpleSymbol Lit204 = ((SimpleSymbol) new
SimpleSymbol("HorizontalArrangement6").readResolve());
    static final IntNum Lit205 = IntNum.make(-1006);
    static final FString Lit206 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
    static final FString Lit207 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit208 = ((SimpleSymbol) new
SimpleSymbol("Label5").readResolve());
    static final SimpleSymbol Lit209 = ((SimpleSymbol) new
SimpleSymbol("FontBold").readResolve());
    static final SimpleSymbol Lit21 = ((SimpleSymbol) new
SimpleSymbol("Screen1$Initialize").readResolve());
    static final SimpleSymbol Lit210 = ((SimpleSymbol) new
SimpleSymbol("FontSize").readResolve());
    static final IntNum Lit211 = IntNum.make(20);
    static final IntNum Lit212 = IntNum.make(-1004);
    static final FString Lit213 = new
FString("com.google.appinventor.components.runtime.Label");
    static final FString Lit214 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit215 = ((SimpleSymbol) new
SimpleSymbol("Label6").readResolve());
    static final IntNum Lit216 = IntNum.make(-1004);
    static final FString Lit217 = new
FString("com.google.appinventor.components.runtime.Label");
    static final FString Lit218 = new
FString("com.google.appinventor.components.runtime.Label");
    static final SimpleSymbol Lit219 = ((SimpleSymbol) new
SimpleSymbol("Label7").readResolve());
    static final SimpleSymbol Lit22 = ((SimpleSymbol) new
SimpleSymbol("Initialize").readResolve());
    static final IntNum Lit220 = IntNum.make(-1004);
    static final FString Lit221 = new
FString("com.google.appinventor.components.runtime.Label");
    static final FString Lit222 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
```



Digitalización de un banco de ensayos y comprobación experimental

```
static final SimpleSymbol Lit223 = ((SimpleSymbol) new
SimpleSymbol("HorizontalArrangement7").readResolve());
static final IntNum Lit224 = IntNum.make(-1006);
static final FString Lit225 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
static final FString Lit226 = new
FString("com.google.appinventor.components.runtime.Label");
static final SimpleSymbol Lit227 = ((SimpleSymbol) new
SimpleSymbol("Label8").readResolve());
static final IntNum Lit228 = IntNum.make(-1004);
static final FString Lit229 = new
FString("com.google.appinventor.components.runtime.Label");
static final FString Lit23 = new
FString("com.google.appinventor.components.runtime.VerticalArrangement
");
static final FString Lit230 = new
FString("com.google.appinventor.components.runtime.Label");
static final SimpleSymbol Lit231 = ((SimpleSymbol) new
SimpleSymbol("Label9").readResolve());
static final IntNum Lit232 = IntNum.make(-1004);
static final FString Lit233 = new
FString("com.google.appinventor.components.runtime.Label");
static final FString Lit234 = new
FString("com.google.appinventor.components.runtime.Label");
static final SimpleSymbol Lit235 = ((SimpleSymbol) new
SimpleSymbol("Label10").readResolve());
static final IntNum Lit236 = IntNum.make(-1004);
static final FString Lit237 = new
FString("com.google.appinventor.components.runtime.Label");
static final FString Lit238 = new
FString("com.google.appinventor.components.runtime.BluetoothClient");
static final FString Lit239 = new
FString("com.google.appinventor.components.runtime.BluetoothClient");
static final SimpleSymbol Lit24 = ((SimpleSymbol) new
SimpleSymbol("VerticalArrangement1").readResolve());
static final FString Lit240 = new
FString("com.google.appinventor.components.runtime.Clock");
static final SimpleSymbol Lit241 = ((SimpleSymbol) new
SimpleSymbol("Clock1").readResolve());
static final FString Lit242 = new
FString("com.google.appinventor.components.runtime.Clock");
static final SimpleSymbol Lit243 = ((SimpleSymbol) new
SimpleSymbol("IsConnected").readResolve());
static final SimpleSymbol Lit244 = ((SimpleSymbol) new
SimpleSymbol("BytesAvailableToReceive").readResolve());
static final IntNum Lit245 = IntNum.make(0);
static final PairWithPosition Lit246 = PairWithPosition.make(Lit6,
PairWithPosition.make(Lit6, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2044119),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2044111);
```



Digitalización de un banco de ensayos y comprobación experimental

```
static final SimpleSymbol Lit247 = ((SimpleSymbol) new
SimpleSymbol("ReceiveText").readResolve());
static final PairWithPosition Lit248 = PairWithPosition.make(Lit6,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2044389);
static final PairWithPosition Lit249 =
static final SimpleSymbol Lit25 = ((SimpleSymbol) new
SimpleSymbol("AlignVertical").readResolve());
static final PairWithPosition Lit250 =
PairWithPosition.make(Lit43, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2044561);
static final PairWithPosition Lit251 = PairWithPosition.make(Lit6,
PairWithPosition.make(Lit6, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2044598),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2044590);
static final PairWithPosition Lit252 =
PairWithPosition.make(Lit43, PairWithPosition.make(Lit6, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2044757),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2044751);
static final PairWithPosition Lit253 =
PairWithPosition.make(Lit43, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2044928);
static final PairWithPosition Lit254 = PairWithPosition.make(Lit6,
PairWithPosition.make(Lit6, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2044965),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2044957);
static final PairWithPosition Lit255 =
PairWithPosition.make(Lit43, PairWithPosition.make(Lit6, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2045124),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2045118);
static final PairWithPosition Lit256 =
PairWithPosition.make(Lit43, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
```



Digitalización de un banco de ensayos y comprobación experimental

```
_v1_copy/Screen1.yail", 2045295);  
    static final PairWithPosition Lit257 = PairWithPosition.make(Lit6,  
PairWithPosition.make(Lit6, LList.Empty,  
"/tmp/1626207119500_0.7964086826725233-  
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros  
_v1_copy/Screen1.yail", 2045332),  
"/tmp/1626207119500_0.7964086826725233-  
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros  
_v1_copy/Screen1.yail", 2045324);  
    static final PairWithPosition Lit258 =  
PairWithPosition.make(Lit43, PairWithPosition.make(Lit6, LList.Empty,  
"/tmp/1626207119500_0.7964086826725233-  
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros  
_v1_copy/Screen1.yail", 2045493),  
"/tmp/1626207119500_0.7964086826725233-  
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros  
_v1_copy/Screen1.yail", 2045487);  
    static final PairWithPosition Lit259 =  
PairWithPosition.make(Lit43, LList.Empty,  
"/tmp/1626207119500_0.7964086826725233-  
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros  
_v1_copy/Screen1.yail", 2045664);  
    static final IntNum Lit26 = IntNum.make(2);  
    static final IntNum Lit260 = IntNum.make(4);  
    static final PairWithPosition Lit261 = PairWithPosition.make(Lit6,  
PairWithPosition.make(Lit6, LList.Empty,  
"/tmp/1626207119500_0.7964086826725233-  
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros  
_v1_copy/Screen1.yail", 2045701),  
"/tmp/1626207119500_0.7964086826725233-  
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros  
_v1_copy/Screen1.yail", 2045693);  
    static final PairWithPosition Lit262 =  
PairWithPosition.make(Lit43, PairWithPosition.make(Lit6, LList.Empty,  
"/tmp/1626207119500_0.7964086826725233-  
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros  
_v1_copy/Screen1.yail", 2045862),  
"/tmp/1626207119500_0.7964086826725233-  
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros  
_v1_copy/Screen1.yail", 2045856);  
    static final PairWithPosition Lit263 =  
PairWithPosition.make(Lit43, LList.Empty,  
"/tmp/1626207119500_0.7964086826725233-  
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros  
_v1_copy/Screen1.yail", 2046033);  
    static final IntNum Lit264 = IntNum.make(5);  
    static final PairWithPosition Lit265 = PairWithPosition.make(Lit6,  
PairWithPosition.make(Lit6, LList.Empty,  
"/tmp/1626207119500_0.7964086826725233-  
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros  
_v1_copy/Screen1.yail", 2046070),  
"/tmp/1626207119500_0.7964086826725233-  
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros  
_v1_copy/Screen1.yail", 2046062);  
    static final PairWithPosition Lit266 =
```



Digitalización de un banco de ensayos y comprobación experimental

```
PairWithPosition.make(Lit43, PairWithPosition.make(Lit6, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2046231),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2046225);
    static final PairWithPosition Lit267 =
PairWithPosition.make(Lit43, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2046402);
    static final IntNum Lit268 = IntNum.make(6);
    static final PairWithPosition Lit269 = PairWithPosition.make(Lit6,
PairWithPosition.make(Lit6, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2046439),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2046431);
    static final SimpleSymbol Lit27 = ((SimpleSymbol) new
SimpleSymbol("Height").readResolve());
    static final PairWithPosition Lit270;
    static final SimpleSymbol Lit271 = ((SimpleSymbol) new
SimpleSymbol("Clock1$Timer").readResolve());
    static final SimpleSymbol Lit272 = ((SimpleSymbol) new
SimpleSymbol("Timer").readResolve());
    static final SimpleSymbol Lit273 = ((SimpleSymbol) new
SimpleSymbol("get-simple-name").readResolve());
    static final SimpleSymbol Lit274 = ((SimpleSymbol) new
SimpleSymbol("android-log-form").readResolve());
    static final SimpleSymbol Lit275 = ((SimpleSymbol) new
SimpleSymbol("add-to-form-environment").readResolve());
    static final SimpleSymbol Lit276 = ((SimpleSymbol) new
SimpleSymbol("lookup-in-form-environment").readResolve());
    static final SimpleSymbol Lit277 = ((SimpleSymbol) new
SimpleSymbol("is-bound-in-form-environment").readResolve());
    static final SimpleSymbol Lit278 = ((SimpleSymbol) new
SimpleSymbol("add-to-global-var-environment").readResolve());
    static final SimpleSymbol Lit279 = ((SimpleSymbol) new
SimpleSymbol("add-to-events").readResolve());
    static final IntNum Lit28 = IntNum.make(-2);
    static final SimpleSymbol Lit280 = ((SimpleSymbol) new
SimpleSymbol("add-to-components").readResolve());
    static final SimpleSymbol Lit281 = ((SimpleSymbol) new
SimpleSymbol("add-to-global-vars").readResolve());
    static final SimpleSymbol Lit282 = ((SimpleSymbol) new
SimpleSymbol("add-to-form-do-after-creation").readResolve());
    static final SimpleSymbol Lit283 = ((SimpleSymbol) new
SimpleSymbol("send-error").readResolve());
    static final SimpleSymbol Lit284 = ((SimpleSymbol) new
SimpleSymbol("dispatchEvent").readResolve());
    static final SimpleSymbol Lit285 = ((SimpleSymbol) new
SimpleSymbol("dispatchGenericEvent").readResolve());
```


Digitalización de un banco de ensayos y comprobación experimental

```
static final SimpleSymbol Lit286 = ((SimpleSymbol) new
SimpleSymbol("lookup-handler").readResolve());
static final SimpleSymbol Lit29 = ((SimpleSymbol) new
SimpleSymbol("Width").readResolve());
static final SimpleSymbol Lit3 = ((SimpleSymbol) new
SimpleSymbol("g$Received_data").readResolve());
static final FString Lit30 = new
FString("com.google.appinventor.components.runtime.VerticalArrangement
");
static final FString Lit31 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
static final SimpleSymbol Lit32 = ((SimpleSymbol) new
SimpleSymbol("HorizontalArrangement5").readResolve());
static final IntNum Lit33 = IntNum.make(-1020);
static final FString Lit34 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
static final FString Lit35 = new
FString("com.google.appinventor.components.runtime.ListPicker");
static final SimpleSymbol Lit36 = ((SimpleSymbol) new
SimpleSymbol("ListPicker1").readResolve());
static final IntNum Lit37 = IntNum.make(80);
static final SimpleSymbol Lit38 = ((SimpleSymbol) new
SimpleSymbol("Image").readResolve());
static final FString Lit39 = new
FString("com.google.appinventor.components.runtime.ListPicker");
static final SimpleSymbol Lit4 = ((SimpleSymbol) new
SimpleSymbol("AlignHorizontal").readResolve());
static final SimpleSymbol Lit40 = ((SimpleSymbol) new
SimpleSymbol("Elements").readResolve());
static final SimpleSymbol Lit41 = ((SimpleSymbol) new
SimpleSymbol("BluetoothClient1").readResolve());
static final SimpleSymbol Lit42 = ((SimpleSymbol) new
SimpleSymbol("AddressesAndNames").readResolve());
static final SimpleSymbol Lit43;
static final SimpleSymbol Lit44 = ((SimpleSymbol) new
SimpleSymbol("ListPicker1$BeforePicking").readResolve());
static final SimpleSymbol Lit45 = ((SimpleSymbol) new
SimpleSymbol("BeforePicking").readResolve());
static final SimpleSymbol Lit46 = ((SimpleSymbol) new
SimpleSymbol("Selection").readResolve());
static final SimpleSymbol Lit47 = ((SimpleSymbol) new
SimpleSymbol("Connect").readResolve());
static final PairWithPosition Lit48 = PairWithPosition.make(Lit8,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 250022);
static final SimpleSymbol Lit49 = ((SimpleSymbol) new
SimpleSymbol("Switch1").readResolve());
static final IntNum Lit5 = IntNum.make(3);
static final SimpleSymbol Lit50 = ((SimpleSymbol) new
SimpleSymbol("On").readResolve());
static final SimpleSymbol Lit51 = ((SimpleSymbol) new
SimpleSymbol("SendText").readResolve());
```



Digitalización de un banco de ensayos y comprobación experimental

```
static final PairWithPosition Lit52 = PairWithPosition.make(Lit8,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 250171);
static final SimpleSymbol Lit53 = ((SimpleSymbol) new
SimpleSymbol("ListPicker1$AfterPicking").readResolve());
static final SimpleSymbol Lit54 = ((SimpleSymbol) new
SimpleSymbol("AfterPicking").readResolve());
static final FString Lit55 = new
FString("com.google.appinventor.components.runtime.Switch");
static final SimpleSymbol Lit56 = ((SimpleSymbol) new
SimpleSymbol("Text").readResolve());
static final FString Lit57 = new
FString("com.google.appinventor.components.runtime.Switch");
static final PairWithPosition Lit58 = PairWithPosition.make(Lit8,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 286858);
static final SimpleSymbol Lit59 = ((SimpleSymbol) new
SimpleSymbol("Button7").readResolve());
static final SimpleSymbol Lit6;
static final SimpleSymbol Lit60 = ((SimpleSymbol) new
SimpleSymbol("Visible").readResolve());
static final SimpleSymbol Lit61 = ((SimpleSymbol) new
SimpleSymbol("Button8").readResolve());
static final SimpleSymbol Lit62 = ((SimpleSymbol) new
SimpleSymbol("Button9").readResolve());
static final SimpleSymbol Lit63 = ((SimpleSymbol) new
SimpleSymbol("Button10").readResolve());
static final PairWithPosition Lit64 = PairWithPosition.make(Lit8,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 287858);
static final SimpleSymbol Lit65 = ((SimpleSymbol) new
SimpleSymbol("Switch1$Changed").readResolve());
static final SimpleSymbol Lit66 = ((SimpleSymbol) new
SimpleSymbol("Changed").readResolve());
static final FString Lit67 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
static final SimpleSymbol Lit68 = ((SimpleSymbol) new
SimpleSymbol("HorizontalArrangement3").readResolve());
static final IntNum Lit69 = IntNum.make(-1010);
static final SimpleSymbol Lit7 = ((SimpleSymbol) new
SimpleSymbol("AppName").readResolve());
static final FString Lit70 = new
FString("com.google.appinventor.components.runtime.HorizontalArrangeme
nt");
static final FString Lit71 = new
FString("com.google.appinventor.components.runtime.Label");
static final SimpleSymbol Lit72 = ((SimpleSymbol) new
SimpleSymbol("Label1").readResolve());
static final IntNum Lit73 = IntNum.make(-1035);
static final FString Lit74 = new
FString("com.google.appinventor.components.runtime.Label");
```



Digitalización de un banco de ensayos y comprobación experimental

```
static final FString Lit75 = new
FString("com.google.appinventor.components.runtime.TextBox");
static final SimpleSymbol Lit76 = ((SimpleSymbol) new
SimpleSymbol("TextBox1").readResolve());
static final SimpleSymbol Lit77 = ((SimpleSymbol) new
SimpleSymbol("NumbersOnly").readResolve());
static final SimpleSymbol Lit78 = ((SimpleSymbol) new
SimpleSymbol("TextAlignment").readResolve());
static final IntNum Lit79 = IntNum.make(1);
static final SimpleSymbol Lit8;
static final IntNum Lit80 = IntNum.make(-1030);
static final FString Lit81 = new
FString("com.google.appinventor.components.runtime.TextBox");
static final FString Lit82 = new
FString("com.google.appinventor.components.runtime.Label");
static final SimpleSymbol Lit83 = ((SimpleSymbol) new
SimpleSymbol("Label11").readResolve());
static final FString Lit84 = new
FString("com.google.appinventor.components.runtime.Label");
static final FString Lit85 = new
FString("com.google.appinventor.components.runtime.Button");
static final SimpleSymbol Lit86 = ((SimpleSymbol) new
SimpleSymbol("Shape").readResolve());
static final FString Lit87 = new
FString("com.google.appinventor.components.runtime.Button");
static final PairWithPosition Lit88 = PairWithPosition.make(Lit8,
PairWithPosition.make(Lit8, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 495791),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 495785);
static final PairWithPosition Lit89 = PairWithPosition.make(Lit8,
LList.Empty, "/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 495808);
static final SimpleSymbol Lit9 = ((SimpleSymbol) new
SimpleSymbol("Icon").readResolve());
static final SimpleSymbol Lit90 = ((SimpleSymbol) new
SimpleSymbol("Button3$Click").readResolve());
static final SimpleSymbol Lit91 = ((SimpleSymbol) new
SimpleSymbol("Click").readResolve());
static final FString Lit92 = new
FString("com.google.appinventor.components.runtime.Label");
static final SimpleSymbol Lit93 = ((SimpleSymbol) new
SimpleSymbol("Label16").readResolve());
static final FString Lit94 = new
FString("com.google.appinventor.components.runtime.Label");
static final FString Lit95 = new
FString("com.google.appinventor.components.runtime.Button");
static final IntNum Lit96 = IntNum.make(-1004);
static final IntNum Lit97 = IntNum.make(-1005);
static final FString Lit98 = new
FString("com.google.appinventor.components.runtime.Button");
```

Digitalización de un banco de ensayos y comprobación experimental

```
static final PairWithPosition Lit99 = PairWithPosition.make(Lit8,
PairWithPosition.make(Lit8, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 585879),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 585873);
public static Screen1 Screen1;
static final ModuleMethod lambda$Fn1 = null;
static final ModuleMethod lambda$Fn10 = null;
static final ModuleMethod lambda$Fn11 = null;
static final ModuleMethod lambda$Fn12 = null;
static final ModuleMethod lambda$Fn13 = null;
static final ModuleMethod lambda$Fn14 = null;
static final ModuleMethod lambda$Fn15 = null;
static final ModuleMethod lambda$Fn16 = null;
static final ModuleMethod lambda$Fn17 = null;
static final ModuleMethod lambda$Fn18 = null;
static final ModuleMethod lambda$Fn19 = null;
static final ModuleMethod lambda$Fn2 = null;
static final ModuleMethod lambda$Fn20 = null;
static final ModuleMethod lambda$Fn21 = null;
static final ModuleMethod lambda$Fn22 = null;
static final ModuleMethod lambda$Fn23 = null;
static final ModuleMethod lambda$Fn24 = null;
static final ModuleMethod lambda$Fn25 = null;
static final ModuleMethod lambda$Fn26 = null;
static final ModuleMethod lambda$Fn27 = null;
static final ModuleMethod lambda$Fn28 = null;
static final ModuleMethod lambda$Fn29 = null;
static final ModuleMethod lambda$Fn3 = null;
static final ModuleMethod lambda$Fn30 = null;
static final ModuleMethod lambda$Fn31 = null;
static final ModuleMethod lambda$Fn32 = null;
static final ModuleMethod lambda$Fn33 = null;
static final ModuleMethod lambda$Fn34 = null;
static final ModuleMethod lambda$Fn35 = null;
static final ModuleMethod lambda$Fn36 = null;
static final ModuleMethod lambda$Fn37 = null;
static final ModuleMethod lambda$Fn38 = null;
static final ModuleMethod lambda$Fn39 = null;
static final ModuleMethod lambda$Fn4 = null;
static final ModuleMethod lambda$Fn40 = null;
static final ModuleMethod lambda$Fn41 = null;
static final ModuleMethod lambda$Fn42 = null;
static final ModuleMethod lambda$Fn43 = null;
static final ModuleMethod lambda$Fn44 = null;
static final ModuleMethod lambda$Fn45 = null;
static final ModuleMethod lambda$Fn46 = null;
static final ModuleMethod lambda$Fn47 = null;
static final ModuleMethod lambda$Fn48 = null;
static final ModuleMethod lambda$Fn49 = null;
static final ModuleMethod lambda$Fn5 = null;
static final ModuleMethod lambda$Fn50 = null;
```



Digitalización de un banco de ensayos y comprobación experimental

```
static final ModuleMethod lambda$Fn51 = null;
static final ModuleMethod lambda$Fn52 = null;
static final ModuleMethod lambda$Fn53 = null;
static final ModuleMethod lambda$Fn54 = null;
static final ModuleMethod lambda$Fn55 = null;
static final ModuleMethod lambda$Fn56 = null;
static final ModuleMethod lambda$Fn57 = null;
static final ModuleMethod lambda$Fn58 = null;
static final ModuleMethod lambda$Fn59 = null;
static final ModuleMethod lambda$Fn6 = null;
static final ModuleMethod lambda$Fn60 = null;
static final ModuleMethod lambda$Fn61 = null;
static final ModuleMethod lambda$Fn62 = null;
static final ModuleMethod lambda$Fn63 = null;
static final ModuleMethod lambda$Fn64 = null;
static final ModuleMethod lambda$Fn65 = null;
static final ModuleMethod lambda$Fn66 = null;
static final ModuleMethod lambda$Fn67 = null;
static final ModuleMethod lambda$Fn68 = null;
static final ModuleMethod lambda$Fn69 = null;
static final ModuleMethod lambda$Fn7 = null;
static final ModuleMethod lambda$Fn70 = null;
static final ModuleMethod lambda$Fn71 = null;
static final ModuleMethod lambda$Fn72 = null;
static final ModuleMethod lambda$Fn73 = null;
static final ModuleMethod lambda$Fn74 = null;
static final ModuleMethod lambda$Fn75 = null;
static final ModuleMethod lambda$Fn76 = null;
static final ModuleMethod lambda$Fn77 = null;
static final ModuleMethod lambda$Fn78 = null;
static final ModuleMethod lambda$Fn79 = null;
static final ModuleMethod lambda$Fn8 = null;
static final ModuleMethod lambda$Fn80 = null;
static final ModuleMethod lambda$Fn81 = null;
static final ModuleMethod lambda$Fn82 = null;
static final ModuleMethod lambda$Fn83 = null;
static final ModuleMethod lambda$Fn84 = null;
static final ModuleMethod lambda$Fn85 = null;
static final ModuleMethod lambda$Fn86 = null;
static final ModuleMethod lambda$Fn87 = null;
static final ModuleMethod lambda$Fn9 = null;
public Boolean $Stdebug$Mnform$St;
public final ModuleMethod $define;
public BluetoothClient BluetoothClient1;
public Button Button1;
public final ModuleMethod Button1$Click;
public Button Button10;
public final ModuleMethod Button10$Click;
public Button Button2;
public final ModuleMethod Button2$Click;
public Button Button3;
public final ModuleMethod Button3$Click;
public Button Button4;
public final ModuleMethod Button4$Click;
public Button Button5;
```



Digitalización de un banco de ensayos y comprobación experimental

```
public final ModuleMethod Button5$Click;
public Button Button7;
public final ModuleMethod Button7$Click;
public Button Button8;
public final ModuleMethod Button8$Click;
public Button Button9;
public final ModuleMethod Button9$Click;
public Clock Clock1;
public final ModuleMethod Clock1$Timer;
public HorizontalArrangement HorizontalArrangement1;
public HorizontalArrangement HorizontalArrangement2;
public HorizontalArrangement HorizontalArrangement3;
public HorizontalArrangement HorizontalArrangement4;
public HorizontalArrangement HorizontalArrangement5;
public HorizontalArrangement HorizontalArrangement6;
public HorizontalArrangement HorizontalArrangement7;
public HorizontalArrangement HorizontalArrangement8;
public Label Label1;
public Label Label10;
public Label Label11;
public Label Label12;
public Label Label13;
public Label Label14;
public Label Label15;
public Label Label16;
public Label Label17;
public Label Label18;
public Label Label19;
public Label Label2;
public Label Label3;
public Label Label4;
public Label Label5;
public Label Label6;
public Label Label7;
public Label Label8;
public Label Label9;
public ListPicker ListPicker1;
public final ModuleMethod ListPicker1$AfterPicking;
public final ModuleMethod ListPicker1$BeforePicking;
public final ModuleMethod Screen1$Initialize;
public Switch Switch1;
public final ModuleMethod Switch1$Changed;
public TextBox TextBox1;
public TextBox TextBox2;
public TextBox TextBox3;
public TextBox TextBox4;
public VerticalArrangement VerticalArrangement1;
public final ModuleMethod add$Mnto$Mncomponents;
public final ModuleMethod add$Mnto$Mnevents;
public final ModuleMethod add$Mnto$Mnform$Mndo$Mnafter$Mncreation;
public final ModuleMethod add$Mnto$Mnform$Mnenvironment;
public final ModuleMethod add$Mnto$Mnglobal$Mnvar$Mnenvironment;
public final ModuleMethod add$Mnto$Mnglobal$Mnvars;
public final ModuleMethod android$Mnlog$Mnform;
public LList components$Mnto$Mncreate;
```

Digitalización de un banco de ensayos y comprobación experimental

```
public final ModuleMethod dispatchEvent;
public final ModuleMethod dispatchGenericEvent;
public LList events$Mnto$Mnregister;
public LList form$Mndo$Mnafter$Mncreation;
public Environment form$Mnenvironment;
public Symbol form$Mnname$Mnsymbol;
public final ModuleMethod get$Mnsimple$Mnname;
public Environment global$Mnvar$Mnenvironment;
public LList global$Mnvars$Mnto$Mncreate;
public final ModuleMethod is$Mnbound$Mnin$Mnform$Mnenvironment;
public final ModuleMethod lookup$Mnhandler;
public final ModuleMethod lookup$Mnin$Mnform$Mnenvironment;
public final ModuleMethod onCreate;
public final ModuleMethod process$Mnexception;
public final ModuleMethod send$Mnerror;

static {
    SimpleSymbol simpleSymbol = (SimpleSymbol) new
SimpleSymbol("list").readResolve();
    Lit43 = simpleSymbol;
    SimpleSymbol simpleSymbol2 = (SimpleSymbol) new
SimpleSymbol("number").readResolve();
    Lit6 = simpleSymbol2;
    Lit270 = PairWithPosition.make(simpleSymbol,
PairWithPosition.make(simpleSymbol2, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2046600),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2046594);
    SimpleSymbol simpleSymbol3 = (SimpleSymbol) new
SimpleSymbol(PropertyTypeConstants.PROPERTY_TYPE_TEXT).readResolve();
    Lit8 = simpleSymbol3;
    Lit249 = PairWithPosition.make(simpleSymbol3,
PairWithPosition.make(Lit8, LList.Empty,
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2044411),
"/tmp/1626207119500_0.7964086826725233-
0/youngandroidproject/./src/appinventor/ai_adrian_leordean/Parametros
_v1_copy/Screen1.yail", 2044405);
}

public Screen1() {
    ModuleInfo.register(this);
    frame frame2 = new frame();
    frame2.$main = this;
    this.get$Mnsimple$Mnname = new ModuleMethod(frame2, 1, Lit273,
FragmentTransaction.TRANSIT_FRAGMENT_OPEN);
    this.onCreate = new ModuleMethod(frame2, 2, "onCreate",
FragmentTransaction.TRANSIT_FRAGMENT_OPEN);
    this.android$Mnlog$Mnform = new ModuleMethod(frame2, 3,
Lit274, FragmentTransaction.TRANSIT_FRAGMENT_OPEN);
    this.add$Mnto$Mnform$Mnenvironment = new ModuleMethod(frame2,
```

Digitalización de un banco de ensayos y comprobación experimental

```
4, Lit275, 8194);
    this.lookup$Mnin$Mnform$Mnenvironment = new
ModuleMethod(frame2, 5, Lit276, 8193);
    this.is$Mnbound$Mnin$Mnform$Mnenvironment = new
ModuleMethod(frame2, 7, Lit277,
FragmentTransaction.TRANSIT_FRAGMENT_OPEN);
    this.add$Mnto$Mnglobal$Mnvar$Mnenvironment = new
ModuleMethod(frame2, 8, Lit278, 8194);
    this.add$Mnto$Mnevents = new ModuleMethod(frame2, 9, Lit279,
8194);
    this.add$Mnto$Mncomponents = new ModuleMethod(frame2, 10,
Lit280, 16388);
    this.add$Mnto$Mnglobal$Mnvars = new ModuleMethod(frame2, 11,
Lit281, 8194);
    this.add$Mnto$Mnform$Mndo$Mnafter$Mncreation = new
ModuleMethod(frame2, 12, Lit282,
FragmentTransaction.TRANSIT_FRAGMENT_OPEN);
    this.send$Mnerror = new ModuleMethod(frame2, 13, Lit283,
FragmentTransaction.TRANSIT_FRAGMENT_OPEN);
    this.process$Mnexception = new ModuleMethod(frame2, 14,
"process-exception", FragmentTransaction.TRANSIT_FRAGMENT_OPEN);
    this.dispatchEvent = new ModuleMethod(frame2, 15, Lit284,
16388);
    this.dispatchGenericEvent = new ModuleMethod(frame2, 16,
Lit285, 16388);
    this.lookup$Mnhandler = new ModuleMethod(frame2, 17, Lit286,
8194);
    ModuleMethod moduleMethod = new ModuleMethod(frame2, 18,
(Object) null, 0);
    moduleMethod.setProperty("source-location",
"/tmp/runtime7688346696957088787.scm:622");
    lambda$Fn1 = moduleMethod;
    this.$define = new ModuleMethod(frame2, 19, "$define", 0);
    lambda$Fn2 = new ModuleMethod(frame2, 20, (Object) null, 0);
    lambda$Fn3 = new ModuleMethod(frame2, 21, (Object) null, 0);
    this.Screen1$Initialize = new ModuleMethod(frame2, 22, Lit21,
0);
    lambda$Fn4 = new ModuleMethod(frame2, 23, (Object) null, 0);
    lambda$Fn5 = new ModuleMethod(frame2, 24, (Object) null, 0);
    lambda$Fn6 = new ModuleMethod(frame2, 25, (Object) null, 0);
    lambda$Fn7 = new ModuleMethod(frame2, 26, (Object) null, 0);
    lambda$Fn8 = new ModuleMethod(frame2, 27, (Object) null, 0);
    lambda$Fn9 = new ModuleMethod(frame2, 28, (Object) null, 0);
    this.ListPicker1$BeforePicking = new ModuleMethod(frame2, 29,
Lit44, 0);
    this.ListPicker1$AfterPicking = new ModuleMethod(frame2, 30,
Lit53, 0);
    lambda$Fn10 = new ModuleMethod(frame2, 31, (Object) null, 0);
    lambda$Fn11 = new ModuleMethod(frame2, 32, (Object) null, 0);
    this.Switch1$Changed = new ModuleMethod(frame2, 33, Lit65, 0);
    lambda$Fn12 = new ModuleMethod(frame2, 34, (Object) null, 0);
    lambda$Fn13 = new ModuleMethod(frame2, 35, (Object) null, 0);
    lambda$Fn14 = new ModuleMethod(frame2, 36, (Object) null, 0);
    lambda$Fn15 = new ModuleMethod(frame2, 37, (Object) null, 0);
    lambda$Fn16 = new ModuleMethod(frame2, 38, (Object) null, 0);
```


Digitalización de un banco de ensayos y comprobación experimental

```
lambda$Fn17 = new ModuleMethod(frame2, 39, (Object) null, 0);  
lambda$Fn18 = new ModuleMethod(frame2, 40, (Object) null, 0);  
lambda$Fn19 = new ModuleMethod(frame2, 41, (Object) null, 0);  
lambda$Fn20 = new ModuleMethod(frame2, 42, (Object) null, 0);  
lambda$Fn21 = new ModuleMethod(frame2, 43, (Object) null, 0);  
this.Button3$Click = new ModuleMethod(frame2, 44, Lit90, 0);  
lambda$Fn22 = new ModuleMethod(frame2, 45, (Object) null, 0);  
lambda$Fn23 = new ModuleMethod(frame2, 46, (Object) null, 0);  
lambda$Fn24 = new ModuleMethod(frame2, 47, (Object) null, 0);  
lambda$Fn25 = new ModuleMethod(frame2, 48, (Object) null, 0);  
this.Button7$Click = new ModuleMethod(frame2, 49, Lit101, 0);  
lambda$Fn26 = new ModuleMethod(frame2, 50, (Object) null, 0);  
lambda$Fn27 = new ModuleMethod(frame2, 51, (Object) null, 0);  
lambda$Fn28 = new ModuleMethod(frame2, 52, (Object) null, 0);  
lambda$Fn29 = new ModuleMethod(frame2, 53, (Object) null, 0);  
lambda$Fn30 = new ModuleMethod(frame2, 54, (Object) null, 0);  
lambda$Fn31 = new ModuleMethod(frame2, 55, (Object) null, 0);  
lambda$Fn32 = new ModuleMethod(frame2, 56, (Object) null, 0);  
lambda$Fn33 = new ModuleMethod(frame2, 57, (Object) null, 0);  
lambda$Fn34 = new ModuleMethod(frame2, 58, (Object) null, 0);  
lambda$Fn35 = new ModuleMethod(frame2, 59, (Object) null, 0);  
this.Button2$Click = new ModuleMethod(frame2, 60, Lit121, 0);  
lambda$Fn36 = new ModuleMethod(frame2, 61, (Object) null, 0);  
lambda$Fn37 = new ModuleMethod(frame2, 62, (Object) null, 0);  
lambda$Fn38 = new ModuleMethod(frame2, 63, (Object) null, 0);  
lambda$Fn39 = new ModuleMethod(frame2, 64, (Object) null, 0);  
this.Button8$Click = new ModuleMethod(frame2, 65, Lit131, 0);  
lambda$Fn40 = new ModuleMethod(frame2, 66, (Object) null, 0);  
lambda$Fn41 = new ModuleMethod(frame2, 67, (Object) null, 0);  
lambda$Fn42 = new ModuleMethod(frame2, 68, (Object) null, 0);  
lambda$Fn43 = new ModuleMethod(frame2, 69, (Object) null, 0);  
lambda$Fn44 = new ModuleMethod(frame2, 70, (Object) null, 0);  
lambda$Fn45 = new ModuleMethod(frame2, 71, (Object) null, 0);  
lambda$Fn46 = new ModuleMethod(frame2, 72, (Object) null, 0);  
lambda$Fn47 = new ModuleMethod(frame2, 73, (Object) null, 0);  
lambda$Fn48 = new ModuleMethod(frame2, 74, (Object) null, 0);  
lambda$Fn49 = new ModuleMethod(frame2, 75, (Object) null, 0);  
this.Button1$Click = new ModuleMethod(frame2, 76, Lit151, 0);  
lambda$Fn50 = new ModuleMethod(frame2, 77, (Object) null, 0);  
lambda$Fn51 = new ModuleMethod(frame2, 78, (Object) null, 0);  
lambda$Fn52 = new ModuleMethod(frame2, 79, (Object) null, 0);  
lambda$Fn53 = new ModuleMethod(frame2, 80, (Object) null, 0);  
this.Button9$Click = new ModuleMethod(frame2, 81, Lit161, 0);  
lambda$Fn54 = new ModuleMethod(frame2, 82, (Object) null, 0);  
lambda$Fn55 = new ModuleMethod(frame2, 83, (Object) null, 0);  
lambda$Fn56 = new ModuleMethod(frame2, 84, (Object) null, 0);  
lambda$Fn57 = new ModuleMethod(frame2, 85, (Object) null, 0);  
lambda$Fn58 = new ModuleMethod(frame2, 86, (Object) null, 0);  
lambda$Fn59 = new ModuleMethod(frame2, 87, (Object) null, 0);  
lambda$Fn60 = new ModuleMethod(frame2, 88, (Object) null, 0);  
lambda$Fn61 = new ModuleMethod(frame2, 89, (Object) null, 0);  
lambda$Fn62 = new ModuleMethod(frame2, 90, (Object) null, 0);  
lambda$Fn63 = new ModuleMethod(frame2, 91, (Object) null, 0);  
this.Button4$Click = new ModuleMethod(frame2, 92, Lit181, 0);  
lambda$Fn64 = new ModuleMethod(frame2, 93, (Object) null, 0);
```

Digitalización de un banco de ensayos y comprobación experimental

```
lambda$Fn65 = new ModuleMethod(frame2, 94, (Object) null, 0);
lambda$Fn66 = new ModuleMethod(frame2, 95, (Object) null, 0);
lambda$Fn67 = new ModuleMethod(frame2, 96, (Object) null, 0);
this.Button10$Click = new ModuleMethod(frame2, 97, Lit191, 0);
lambda$Fn68 = new ModuleMethod(frame2, 98, (Object) null, 0);
lambda$Fn69 = new ModuleMethod(frame2, 99, (Object) null, 0);
this.Button5$Click = new ModuleMethod(frame2, 100, Lit199, 0);
lambda$Fn70 = new ModuleMethod(frame2, 101, (Object) null, 0);
lambda$Fn71 = new ModuleMethod(frame2, 102, (Object) null, 0);
lambda$Fn72 = new ModuleMethod(frame2, 103, (Object) null, 0);
lambda$Fn73 = new ModuleMethod(frame2, 104, (Object) null, 0);
lambda$Fn74 = new ModuleMethod(frame2, 105, (Object) null, 0);
lambda$Fn75 = new ModuleMethod(frame2, 106, (Object) null, 0);
lambda$Fn76 = new ModuleMethod(frame2, 107, (Object) null, 0);
lambda$Fn77 = new ModuleMethod(frame2, 108, (Object) null, 0);
lambda$Fn78 = new ModuleMethod(frame2, 109, (Object) null, 0);
lambda$Fn79 = new ModuleMethod(frame2, 110, (Object) null, 0);
lambda$Fn80 = new ModuleMethod(frame2, 111, (Object) null, 0);
lambda$Fn81 = new ModuleMethod(frame2, 112, (Object) null, 0);
lambda$Fn82 = new ModuleMethod(frame2, 113, (Object) null, 0);
lambda$Fn83 = new ModuleMethod(frame2, 114, (Object) null, 0);
lambda$Fn84 = new ModuleMethod(frame2, 115, (Object) null, 0);
lambda$Fn85 = new ModuleMethod(frame2, 116, (Object) null, 0);
lambda$Fn86 = new ModuleMethod(frame2, 117, (Object) null, 0);
lambda$Fn87 = new ModuleMethod(frame2, 118, (Object) null, 0);
this.Clock1$Timer = new ModuleMethod(frame2, 119, Lit271, 0);
}

public Object lookupInFormEnvironment(Symbol symbol) {
    return lookupInFormEnvironment(symbol, Boolean.FALSE);
}

public void run() {
    CallContext instance = CallContext.getInstance();
    Consumer consumer = instance.consumer;
    instance.consumer = VoidConsumer.instance;
    try {
        run(instance);
        th = null;
    } catch (Throwable th) {
        th = th;
    }
    ModuleBody.runCleanup(instance, th, consumer);
}

public final void run(CallContext $ctx) {
    String obj;
    Consumer $result = $ctx.consumer;
    Object find = require.find("com.google.youngandroid.runtime");
    try {
        ((Runnable) find).run();
        this.$Stdebug$Mnform$St = Boolean.FALSE;
        this.form$Mnenvironment =
Environment.make(misc.symbol$To$String(Lit0));
FString stringAppend =
```

Digitalización de un banco de ensayos y comprobación experimental

```
strings.stringAppend(misc.symbol$To$string(Lit0), "-global-vars");
    if (stringAppend == null) {
        obj = null;
    } else {
        obj = stringAppend.toString();
    }
    this.global$Mnvar$Mnenvironment = Environment.make(obj);
    Screen1 = null;
    this.form$Mnname$Mnsymbol = Lit0;
    this.events$Mnto$Mnregister = LList.Empty;
    this.components$Mnto$Mncreate = LList.Empty;
    this.global$Mnvars$Mnto$Mncreate = LList.Empty;
    this.form$Mndo$Mnafter$Mncreation = LList.Empty;
    Object find2 =
require.find("com.google.youngandroid.runtime");
    try {
        ((Runnable) find2).run();
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addGlobalVarToCurrentFormEnvironment(Lit3,
""), $result);
        } else {
            addToGlobalVars(Lit3, lambda$Fn2);
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
            runtime.setAndCoerceProperty$Ex(Lit0, Lit4, Lit5,
Lit6);
            runtime.setAndCoerceProperty$Ex(Lit0, Lit7,
"Parametros_v1", Lit8);
            runtime.setAndCoerceProperty$Ex(Lit0, Lit9,
"settings-icon.png", Lit8);
            runtime.setAndCoerceProperty$Ex(Lit0, Lit10,
"portrait", Lit8);
            runtime.setAndCoerceProperty$Ex(Lit0, Lit11,
Boolean.TRUE, Lit12);
            runtime.setAndCoerceProperty$Ex(Lit0, Lit13,
"Responsive", Lit8);
            runtime.setAndCoerceProperty$Ex(Lit0, Lit14,
"Screen1", Lit8);
Values.writeValues(runtime.setAndCoerceProperty$Ex(Lit0, Lit15,
Boolean.FALSE, Lit12), $result);
        } else {
            addToFormDoAfterCreation(new Promise(lambda$Fn3));
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
            runtime.addToCurrentFormEnvironment(Lit21,
this.Screen1$Initialize);
        } else {
            addToFormEnvironment(Lit21,
this.Screen1$Initialize);
        }
    }
}
```



Digitalización de un banco de ensayos y comprobación experimental

```
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "Screen1", "Initialize");
        } else {
            addToEvents(Lit0, Lit22);
        }
        this.VerticalArrangement1 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit0, Lit23, Lit24,
lambda$Fn4), $result);
        } else {
            addToComponents(Lit0, Lit30, Lit24, lambda$Fn5);
        }
        this.HorizontalArrangement5 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit24, Lit31, Lit32,
lambda$Fn6), $result);
        } else {
            addToComponents(Lit24, Lit34, Lit32, lambda$Fn7);
        }
        this.ListPicker1 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit32, Lit35, Lit36,
lambda$Fn8), $result);
        } else {
            addToComponents(Lit32, Lit39, Lit36, lambda$Fn9);
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
            runtime.addToCurrentFormEnvironment(Lit44,
this.ListPicker1$BeforePicking);
        } else {
            addToFormEnvironment(Lit44,
this.ListPicker1$BeforePicking);
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "ListPicker1", "BeforePicking");
        } else {
            addToEvents(Lit36, Lit45);
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
            runtime.addToCurrentFormEnvironment(Lit53,
this.ListPicker1$AfterPicking);

```

Digitalización de un banco de ensayos y comprobación experimental

```
    } else {
        addToFormEnvironment(Lit53,
this.ListPicker1$AfterPicking);
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "ListPicker1", "AfterPicking");
    } else {
        addToEvents(Lit36, Lit54);
    }
    this.Switch1 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit24, Lit55, Lit49,
lambda$Fn10), $result);
    } else {
        addToComponents(Lit24, Lit57, Lit49, lambda$Fn11);
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
        runtime.addToCurrentFormEnvironment(Lit65,
this.Switch1$Changed);
    } else {
        addToFormEnvironment(Lit65, this.Switch1$Changed);
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "Switch1", "Changed");
    } else {
        addToEvents(Lit49, Lit66);
    }
    this.HorizontalArrangement3 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit24, Lit67, Lit68,
lambda$Fn12), $result);
    } else {
        addToComponents(Lit24, Lit70, Lit68, lambda$Fn13);
    }
    this.Label1 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit68, Lit71, Lit72,
lambda$Fn14), $result);
    } else {
        addToComponents(Lit68, Lit74, Lit72, lambda$Fn15);
    }
    this.TextBox1 = null;
```

Digitalización de un banco de ensayos y comprobación experimental

```
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit68, Lit75, Lit76,
lambda$Fn16), $result);
    } else {
        addToComponents(Lit68, Lit81, Lit76, lambda$Fn17);
    }
    this.Label11 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit68, Lit82, Lit83,
lambda$Fn18), $result);
    } else {
        addToComponents(Lit68, Lit84, Lit83, lambda$Fn19);
    }
    this.Button3 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit68, Lit85, Lit16,
lambda$Fn20), $result);
    } else {
        addToComponents(Lit68, Lit87, Lit16, lambda$Fn21);
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
        runtime.addToCurrentFormEnvironment(Lit90,
this.Button3$Click);
    } else {
        addToFormEnvironment(Lit90, this.Button3$Click);
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "Button3", "Click");
    } else {
        addToEvents(Lit16, Lit91);
    }
    this.Label16 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit68, Lit92, Lit93,
lambda$Fn22), $result);
    } else {
        addToComponents(Lit68, Lit94, Lit93, lambda$Fn23);
    }
    this.Button7 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit68, Lit95, Lit59,
```



Digitalización de un banco de ensayos y comprobación experimental

```
lambda$Fn24), $result);
    } else {
        addToComponents(Lit68, Lit98, Lit59, lambda$Fn25);
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
        runtime.addToCurrentFormEnvironment(Lit101,
this.Button7$Click);
    } else {
        addToFormEnvironment(Lit101, this.Button7$Click);
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "Button7", "Click");
    } else {
        addToEvents(Lit59, Lit91);
    }
    this.HorizontalArrangement2 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit24, Lit102,
Lit103, lambda$Fn26), $result);
    } else {
        addToComponents(Lit24, Lit105, Lit103,
lambda$Fn27);
    }
    this.Label2 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit103, Lit106,
Lit107, lambda$Fn28), $result);
    } else {
        addToComponents(Lit103, Lit109, Lit107,
lambda$Fn29);
    }
    this.TextBox2 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit103, Lit110,
Lit111, lambda$Fn30), $result);
    } else {
        addToComponents(Lit103, Lit113, Lit111,
lambda$Fn31);
    }
    this.Label12 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit103, Lit114,
Lit115, lambda$Fn32), $result);
```



Digitalización de un banco de ensayos y comprobación experimental

```
    } else {
        addToComponents(Lit103, Lit116, Lit115,
lambda$Fn33);
    }
    this.Button2 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit103, Lit117,
Lit18, lambda$Fn34), $result);
    } else {
        addToComponents(Lit103, Lit118, Lit18,
lambda$Fn35);
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
        runtime.addToCurrentFormEnvironment(Lit121,
this.Button2$Click);
    } else {
        addToFormEnvironment(Lit121, this.Button2$Click);
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "Button2", "Click");
    } else {
        addToEvents(Lit18, Lit91);
    }
    this.Label17 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit103, Lit122,
Lit123, lambda$Fn36), $result);
    } else {
        addToComponents(Lit103, Lit124, Lit123,
lambda$Fn37);
    }
    this.Button8 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit103, Lit125,
Lit61, lambda$Fn38), $result);
    } else {
        addToComponents(Lit103, Lit128, Lit61,
lambda$Fn39);
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
        runtime.addToCurrentFormEnvironment(Lit131,
this.Button8$Click);
    } else {
        addToFormEnvironment(Lit131, this.Button8$Click);
    }
}
```


Digitalización de un banco de ensayos y comprobación experimental

```
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "Button8", "Click");
    } else {
        addToEvents(Lit61, Lit91);
    }
    this.HorizontalArrangement1 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit24, Lit132,
Lit133, lambda$Fn40), $result);
    } else {
        addToComponents(Lit24, Lit135, Lit133,
lambda$Fn41);
    }
    this.Label3 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit133, Lit136,
Lit137, lambda$Fn42), $result);
    } else {
        addToComponents(Lit133, Lit139, Lit137,
lambda$Fn43);
    }
    this.TextBox3 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit133, Lit140,
Lit141, lambda$Fn44), $result);
    } else {
        addToComponents(Lit133, Lit143, Lit141,
lambda$Fn45);
    }
    this.Label13 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit133, Lit144,
Lit145, lambda$Fn46), $result);
    } else {
        addToComponents(Lit133, Lit146, Lit145,
lambda$Fn47);
    }
    this.Button1 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit133, Lit147,
Lit19, lambda$Fn48), $result);
    }
}
```



Digitalización de un banco de ensayos y comprobación experimental

```
        } else {
            addToComponents(Lit133, Lit148, Lit19,
lambda$Fn49);
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
            runtime.addToCurrentFormEnvironment(Lit151,
this.Button1$Click);
        } else {
            addToFormEnvironment(Lit151, this.Button1$Click);
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "Button1", "Click");
        } else {
            addToEvents(Lit19, Lit91);
        }
        this.Label18 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit133, Lit152,
Lit153, lambda$Fn50), $result);
        } else {
            addToComponents(Lit133, Lit154, Lit153,
lambda$Fn51);
        }
        this.Button9 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit133, Lit155,
Lit62, lambda$Fn52), $result);
        } else {
            addToComponents(Lit133, Lit158, Lit62,
lambda$Fn53);
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
            runtime.addToCurrentFormEnvironment(Lit161,
this.Button9$Click);
        } else {
            addToFormEnvironment(Lit161, this.Button9$Click);
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "Button9", "Click");
        } else {
            addToEvents(Lit62, Lit91);
        }
        this.HorizontalArrangement4 = null;
    }
}
```

Digitalización de un banco de ensayos y comprobación experimental

```
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit24, Lit162,
Lit163, lambda$Fn54), $result);
        } else {
            addToComponents(Lit24, Lit165, Lit163,
lambda$Fn55);
        }
        this.Label4 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit163, Lit166,
Lit167, lambda$Fn56), $result);
        } else {
            addToComponents(Lit163, Lit169, Lit167,
lambda$Fn57);
        }
        this.TextBox4 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit163, Lit170,
Lit171, lambda$Fn58), $result);
        } else {
            addToComponents(Lit163, Lit173, Lit171,
lambda$Fn59);
        }
        this.Label14 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit163, Lit174,
Lit175, lambda$Fn60), $result);
        } else {
            addToComponents(Lit163, Lit176, Lit175,
lambda$Fn61);
        }
        this.Button4 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit163, Lit177,
Lit20, lambda$Fn62), $result);
        } else {
            addToComponents(Lit163, Lit178, Lit20,
lambda$Fn63);
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
            runtime.addToCurrentFormEnvironment(Lit181,
this.Button4$Click);
        } else {
            addToFormEnvironment(Lit181, this.Button4$Click);
        }
    }
}
```



Digitalización de un banco de ensayos y comprobación experimental

```
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "Button4", "Click");
    } else {
        addToEvents(Lit20, Lit91);
    }
    this.Label19 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit163, Lit182,
Lit183, lambda$Fn64), $result);
    } else {
        addToComponents(Lit163, Lit184, Lit183,
lambda$Fn65);
    }
    this.Button10 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit163, Lit185,
Lit63, lambda$Fn66), $result);
    } else {
        addToComponents(Lit163, Lit188, Lit63,
lambda$Fn67);
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
        runtime.addToCurrentFormEnvironment(Lit191,
this.Button10$Click);
    } else {
        addToFormEnvironment(Lit191, this.Button10$Click);
    }
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "Button10", "Click");
    } else {
        addToEvents(Lit63, Lit91);
    }
    this.HorizontalArrangement8 = null;
    if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit24, Lit192,
Lit193, Boolean.FALSE), $result);
    } else {
        addToComponents(Lit24, Lit194, Lit193,
Boolean.FALSE);
    }
    this.Button5 = null;
```



Digitalización de un banco de ensayos y comprobación experimental

```
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit193, Lit195,
Lit196, lambda$Fn68), $result);
        } else {
            addToComponents(Lit193, Lit197, Lit196,
lambda$Fn69);
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
            runtime.addToCurrentFormEnvironment(Lit199,
this.Button5$Click);
        } else {
            addToFormEnvironment(Lit199, this.Button5$Click);
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "Button5", "Click");
        } else {
            addToEvents(Lit196, Lit91);
        }
        this.Label15 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit24, Lit200,
Lit201, lambda$Fn70), $result);
        } else {
            addToComponents(Lit24, Lit202, Lit201,
lambda$Fn71);
        }
        this.HorizontalArrangement6 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit24, Lit203,
Lit204, lambda$Fn72), $result);
        } else {
            addToComponents(Lit24, Lit206, Lit204,
lambda$Fn73);
        }
        this.Label5 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit204, Lit207,
Lit208, lambda$Fn74), $result);
        } else {
            addToComponents(Lit204, Lit213, Lit208,
lambda$Fn75);
        }
        this.Label6 = null;

```



Digitalización de un banco de ensayos y comprobación experimental

```
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit204, Lit214,
Lit215, lambda$Fn76), $result);
        } else {
            addToComponents(Lit204, Lit217, Lit215,
lambda$Fn77);
        }
        this.Label7 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit204, Lit218,
Lit219, lambda$Fn78), $result);
        } else {
            addToComponents(Lit204, Lit221, Lit219,
lambda$Fn79);
        }
        this.HorizontalArrangement7 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit24, Lit222,
Lit223, lambda$Fn80), $result);
        } else {
            addToComponents(Lit24, Lit225, Lit223,
lambda$Fn81);
        }
        this.Label8 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit223, Lit226,
Lit227, lambda$Fn82), $result);
        } else {
            addToComponents(Lit223, Lit229, Lit227,
lambda$Fn83);
        }
        this.Label9 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit223, Lit230,
Lit231, lambda$Fn84), $result);
        } else {
            addToComponents(Lit223, Lit233, Lit231,
lambda$Fn85);
        }
        this.Label10 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValue(runtime.addComponentWithinRepl(Lit223, Lit234,
Lit235, lambda$Fn86), $result);
```



Digitalización de un banco de ensayos y comprobación experimental

```
        } else {
            addToComponents(Lit223, Lit237, Lit235,
lambda$Fn87);
        }
        this.BluetoothClient1 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit0, Lit238, Lit41,
Boolean.FALSE), $result);
        } else {
            addToComponents(Lit0, Lit239, Lit41,
Boolean.FALSE);
        }
        this.Clock1 = null;
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
Values.writeValues(runtime.addComponentWithinRepl(Lit0, Lit240,
Lit241, Boolean.FALSE), $result);
        } else {
            addToComponents(Lit0, Lit242, Lit241,
Boolean.FALSE);
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
            runtime.addToCurrentFormEnvironment(Lit271,
this.Clock1$Timer);
        } else {
            addToFormEnvironment(Lit271, this.Clock1$Timer);
        }
        if (runtime.$Stthis$Mnis$Mnthe$Mnrepl$St !=
Boolean.FALSE) {
EventDispatcher.registerEventForDelegation((HandlesEventDispatching)
runtime.$Stthis$Mnform$St, "Clock1", "Timer");
        } else {
            addToEvents(Lit241, Lit272);
        }
        runtime.initRuntime();
    } catch (ClassCastException e) {
        throw new WrongType(e, "java.lang.Runnable.run()", 1,
find2);
    }
    } catch (ClassCastException e2) {
        throw new WrongType(e2, "java.lang.Runnable.run()", 1,
find);
    }
}

static String lambda3() {
    return "";
}

static Object lambda4() {
```



Digitalización de un banco de ensayos y comprobación experimental

```
runtime.setAndCoerceProperty$Ex(Lit0, Lit4, Lit5, Lit6);
runtime.setAndCoerceProperty$Ex(Lit0, Lit7, "Parametros_v1",
Lit8);
runtime.setAndCoerceProperty$Ex(Lit0, Lit9, "settings-
icon.png", Lit8);
runtime.setAndCoerceProperty$Ex(Lit0, Lit10, "portrait",
Lit8);
runtime.setAndCoerceProperty$Ex(Lit0, Lit11, Boolean.TRUE,
Lit12);
runtime.setAndCoerceProperty$Ex(Lit0, Lit13, "Responsive",
Lit8);
runtime.setAndCoerceProperty$Ex(Lit0, Lit14, "Screen1", Lit8);
return runtime.setAndCoerceProperty$Ex(Lit0, Lit15,
Boolean.FALSE, Lit12);
}

public Object Screen1$Initialize() {
runtime.setThisForm();
runtime.setAndCoerceProperty$Ex(Lit16, Lit17, Boolean.FALSE,
Lit12);
runtime.setAndCoerceProperty$Ex(Lit18, Lit17, Boolean.FALSE,
Lit12);
runtime.setAndCoerceProperty$Ex(Lit19, Lit17, Boolean.FALSE,
Lit12);
return runtime.setAndCoerceProperty$Ex(Lit20, Lit17,
Boolean.FALSE, Lit12);
}

static Object lambda5() {
runtime.setAndCoerceProperty$Ex(Lit24, Lit4, Lit5, Lit6);
runtime.setAndCoerceProperty$Ex(Lit24, Lit25, Lit26, Lit6);
runtime.setAndCoerceProperty$Ex(Lit24, Lit27, Lit28, Lit6);
return runtime.setAndCoerceProperty$Ex(Lit24, Lit29, Lit28,
Lit6);
}

static Object lambda6() {
runtime.setAndCoerceProperty$Ex(Lit24, Lit4, Lit5, Lit6);
runtime.setAndCoerceProperty$Ex(Lit24, Lit25, Lit26, Lit6);
runtime.setAndCoerceProperty$Ex(Lit24, Lit27, Lit28, Lit6);
return runtime.setAndCoerceProperty$Ex(Lit24, Lit29, Lit28,
Lit6);
}

static Object lambda7() {
runtime.setAndCoerceProperty$Ex(Lit32, Lit4, Lit5, Lit6);
runtime.setAndCoerceProperty$Ex(Lit32, Lit27, Lit33, Lit6);
return runtime.setAndCoerceProperty$Ex(Lit32, Lit29, Lit28,
Lit6);
}

static Object lambda8() {
runtime.setAndCoerceProperty$Ex(Lit32, Lit4, Lit5, Lit6);
runtime.setAndCoerceProperty$Ex(Lit32, Lit27, Lit33, Lit6);
return runtime.setAndCoerceProperty$Ex(Lit32, Lit29, Lit28,
```


Digitalización de un banco de ensayos y comprobación experimental

```
Lit6);
}

static Object lambda10() {
    runtime.setAndCoerceProperty$Ex(Lit36, Lit27, Lit37, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit36, Lit38, "bluetooth-icon-
31998.png", Lit8);
    return runtime.setAndCoerceProperty$Ex(Lit36, Lit29, Lit37,
Lit6);
}

static Object lambda9() {
    runtime.setAndCoerceProperty$Ex(Lit36, Lit27, Lit37, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit36, Lit38, "bluetooth-icon-
31998.png", Lit8);
    return runtime.setAndCoerceProperty$Ex(Lit36, Lit29, Lit37,
Lit6);
}

public Object ListPicker1$BeforePicking() {
    runtime.setThisForm();
    return runtime.setAndCoerceProperty$Ex(Lit36, Lit40,
runtime.getProperty$1(Lit41, Lit42), Lit43);
}

public Object ListPicker1$AfterPicking() {
    runtime.setThisForm();
    runtime.setAndCoerceProperty$Ex(Lit36, Lit46,
runtime.callComponentMethod(Lit41, Lit47,
LList.list1(runtime.getProperty$1(Lit36, Lit46)), Lit48), Lit8);
    runtime.setAndCoerceProperty$Ex(Lit49, Lit50, Boolean.FALSE,
Lit12);
    return runtime.callComponentMethod(Lit41, Lit51,
LList.list1("runTimer"), Lit52);
}

static Object lambda11() {
    return runtime.setAndCoerceProperty$Ex(Lit49, Lit56, "Editar
parámetros", Lit8);
}

static Object lambda12() {
    return runtime.setAndCoerceProperty$Ex(Lit49, Lit56, "Editar
parámetros", Lit8);
}

public Object Switch1$Changed() {
    runtime.setThisForm();
    if (runtime.getProperty$1(Lit49, Lit50) != Boolean.FALSE) {
        runtime.callComponentMethod(Lit41, Lit51,
LList.list1("modifyParameters"), Lit58);
        runtime.setAndCoerceProperty$Ex(Lit19, Lit17,
Boolean.TRUE, Lit12);
        runtime.setAndCoerceProperty$Ex(Lit18, Lit17,
Boolean.TRUE, Lit12);
    }
}
```



Digitalización de un banco de ensayos y comprobación experimental

```
        runtime.setAndCoerceProperty$Ex(Lit16, Lit17,  
Boolean.TRUE, Lit12);  
        runtime.setAndCoerceProperty$Ex(Lit20, Lit17,  
Boolean.TRUE, Lit12);  
        runtime.setAndCoerceProperty$Ex(Lit59, Lit60,  
Boolean.TRUE, Lit12);  
        runtime.setAndCoerceProperty$Ex(Lit61, Lit60,  
Boolean.TRUE, Lit12);  
        runtime.setAndCoerceProperty$Ex(Lit62, Lit60,  
Boolean.TRUE, Lit12);  
        return runtime.setAndCoerceProperty$Ex(Lit63, Lit60,  
Boolean.TRUE, Lit12);  
    }  
    runtime.setAndCoerceProperty$Ex(Lit19, Lit17, Boolean.FALSE,  
Lit12);  
    runtime.setAndCoerceProperty$Ex(Lit18, Lit17, Boolean.FALSE,  
Lit12);  
    runtime.setAndCoerceProperty$Ex(Lit16, Lit17, Boolean.FALSE,  
Lit12);  
    runtime.setAndCoerceProperty$Ex(Lit20, Lit17, Boolean.FALSE,  
Lit12);  
    runtime.setAndCoerceProperty$Ex(Lit59, Lit60, Boolean.FALSE,  
Lit12);  
    runtime.setAndCoerceProperty$Ex(Lit61, Lit60, Boolean.FALSE,  
Lit12);  
    runtime.setAndCoerceProperty$Ex(Lit62, Lit60, Boolean.FALSE,  
Lit12);  
    runtime.setAndCoerceProperty$Ex(Lit63, Lit60, Boolean.FALSE,  
Lit12);  
    return runtime.callComponentMethod(Lit41, Lit51,  
LList.list1("runTimer"), Lit64);  
    }  
  
    static Object lambda13() {  
        runtime.setAndCoerceProperty$Ex(Lit68, Lit4, Lit5, Lit6);  
        runtime.setAndCoerceProperty$Ex(Lit68, Lit25, Lit26, Lit6);  
        runtime.setAndCoerceProperty$Ex(Lit68, Lit27, Lit69, Lit6);  
        return runtime.setAndCoerceProperty$Ex(Lit68, Lit29, Lit28,  
Lit6);  
    }  
  
    static Object lambda14() {  
        runtime.setAndCoerceProperty$Ex(Lit68, Lit4, Lit5, Lit6);  
        runtime.setAndCoerceProperty$Ex(Lit68, Lit25, Lit26, Lit6);  
        runtime.setAndCoerceProperty$Ex(Lit68, Lit27, Lit69, Lit6);  
        return runtime.setAndCoerceProperty$Ex(Lit68, Lit29, Lit28,  
Lit6);  
    }  
  
    static Object lambda15() {  
        runtime.setAndCoerceProperty$Ex(Lit72, Lit56, "Resolución  
Encoder", Lit8);  
        return runtime.setAndCoerceProperty$Ex(Lit72, Lit29, Lit73,  
Lit6);  
    }  
}
```

Digitalización de un banco de ensayos y comprobación experimental

```
static Object lambda16() {
    runtime.setAndCoerceProperty$Ex(Lit72, Lit56, "Resolución
Encoder    ", Lit8);
    return runtime.setAndCoerceProperty$Ex(Lit72, Lit29, Lit73,
Lit6);
}

static Object lambda17() {
    runtime.setAndCoerceProperty$Ex(Lit76, Lit77, Boolean.TRUE,
Lit12);
    runtime.setAndCoerceProperty$Ex(Lit76, Lit78, Lit79, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit76, Lit29, Lit80,
Lit6);
}

static Object lambda18() {
    runtime.setAndCoerceProperty$Ex(Lit76, Lit77, Boolean.TRUE,
Lit12);
    runtime.setAndCoerceProperty$Ex(Lit76, Lit78, Lit79, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit76, Lit29, Lit80,
Lit6);
}

static Object lambda19() {
    return runtime.setAndCoerceProperty$Ex(Lit83, Lit56, " ",
Lit8);
}

static Object lambda20() {
    return runtime.setAndCoerceProperty$Ex(Lit83, Lit56, " ",
Lit8);
}

static Object lambda21() {
    runtime.setAndCoerceProperty$Ex(Lit16, Lit86, Lit79, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit16, Lit56, "Enviar",
Lit8);
}

static Object lambda22() {
    runtime.setAndCoerceProperty$Ex(Lit16, Lit86, Lit79, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit16, Lit56, "Enviar",
Lit8);
}

public Object Button3$Click() {
    runtime.setThisForm();
    return runtime.callComponentMethod(Lit41, Lit51,
LList.list1(runtime.callYailPrimitive(strings.string$Mnappend,
LList.list2("a", runtime.getProperty$1(Lit76, Lit56)), Lit88,
"join")), Lit89);
}

static Object lambda23() {
```

Digitalización de un banco de ensayos y comprobación experimental

```
        return runtime.setAndCoerceProperty$Ex(Lit93, Lit56, " ",
Lit8);
    }

    static Object lambda24() {
        return runtime.setAndCoerceProperty$Ex(Lit93, Lit56, " ",
Lit8);
    }

    static Object lambda25() {
        runtime.setAndCoerceProperty$Ex(Lit59, Lit27, Lit96, Lit6);
        runtime.setAndCoerceProperty$Ex(Lit59, Lit38, "a.png", Lit8);
        runtime.setAndCoerceProperty$Ex(Lit59, Lit60, Boolean.FALSE,
Lit12);
        return runtime.setAndCoerceProperty$Ex(Lit59, Lit29, Lit97,
Lit6);
    }

    static Object lambda26() {
        runtime.setAndCoerceProperty$Ex(Lit59, Lit27, Lit96, Lit6);
        runtime.setAndCoerceProperty$Ex(Lit59, Lit38, "a.png", Lit8);
        runtime.setAndCoerceProperty$Ex(Lit59, Lit60, Boolean.FALSE,
Lit12);
        return runtime.setAndCoerceProperty$Ex(Lit59, Lit29, Lit97,
Lit6);
    }

    public Object Button7$Click() {
        runtime.setThisForm();
        runtime.callComponentMethod(Lit41, Lit51,
LList.list1(runtime.callYailPrimitive(strings.string$Mnappend,
LList.list2("a", "1024"), Lit99, "join")), Lit100);
        return runtime.setAndCoerceProperty$Ex(Lit76, Lit56, "1024",
Lit8);
    }

    static Object lambda27() {
        runtime.setAndCoerceProperty$Ex(Lit103, Lit4, Lit5, Lit6);
        runtime.setAndCoerceProperty$Ex(Lit103, Lit25, Lit26, Lit6);
        runtime.setAndCoerceProperty$Ex(Lit103, Lit27, Lit104, Lit6);
        return runtime.setAndCoerceProperty$Ex(Lit103, Lit29, Lit28,
Lit6);
    }

    static Object lambda28() {
        runtime.setAndCoerceProperty$Ex(Lit103, Lit4, Lit5, Lit6);
        runtime.setAndCoerceProperty$Ex(Lit103, Lit25, Lit26, Lit6);
        runtime.setAndCoerceProperty$Ex(Lit103, Lit27, Lit104, Lit6);
        return runtime.setAndCoerceProperty$Ex(Lit103, Lit29, Lit28,
Lit6);
    }

    static Object lambda29() {
        runtime.setAndCoerceProperty$Ex(Lit107, Lit56, "Resolución
Óptico ", Lit8);
```

Digitalización de un banco de ensayos y comprobación experimental

```
        return runtime.setAndCoerceProperty$Ex(Lit107, Lit29, Lit108,
Lit6);
    }

    static Object lambda30() {
        runtime.setAndCoerceProperty$Ex(Lit107, Lit56, "Resolución
Óptico", Lit8);
        return runtime.setAndCoerceProperty$Ex(Lit107, Lit29, Lit108,
Lit6);
    }

    static Object lambda31() {
        runtime.setAndCoerceProperty$Ex(Lit111, Lit77, Boolean.TRUE,
Lit12);
        runtime.setAndCoerceProperty$Ex(Lit111, Lit78, Lit79, Lit6);
        return runtime.setAndCoerceProperty$Ex(Lit111, Lit29, Lit112,
Lit6);
    }

    static Object lambda32() {
        runtime.setAndCoerceProperty$Ex(Lit111, Lit77, Boolean.TRUE,
Lit12);
        runtime.setAndCoerceProperty$Ex(Lit111, Lit78, Lit79, Lit6);
        return runtime.setAndCoerceProperty$Ex(Lit111, Lit29, Lit112,
Lit6);
    }

    static Object lambda33() {
        return runtime.setAndCoerceProperty$Ex(Lit115, Lit56, " ",
Lit8);
    }

    static Object lambda34() {
        return runtime.setAndCoerceProperty$Ex(Lit115, Lit56, " ",
Lit8);
    }

    static Object lambda35() {
        runtime.setAndCoerceProperty$Ex(Lit18, Lit86, Lit79, Lit6);
        return runtime.setAndCoerceProperty$Ex(Lit18, Lit56, "Enviar",
Lit8);
    }

    static Object lambda36() {
        runtime.setAndCoerceProperty$Ex(Lit18, Lit86, Lit79, Lit6);
        return runtime.setAndCoerceProperty$Ex(Lit18, Lit56, "Enviar",
Lit8);
    }

    public Object Button2$Click() {
        runtime.setThisForm();
        return runtime.callComponentMethod(Lit41, Lit51,
LList.list1(runtime.callYailPrimitive(strings.string$Mnappend,
LList.list2("b", runtime.getProperty$1(Lit111, Lit56)), Lit119,
"join")), Lit120);
    }
}
```

Digitalización de un banco de ensayos y comprobación experimental

```
}

static Object lambda37() {
    return runtime.setAndCoerceProperty$Ex(Lit123, Lit56, " ",
Lit8);
}

static Object lambda38() {
    return runtime.setAndCoerceProperty$Ex(Lit123, Lit56, " ",
Lit8);
}

static Object lambda39() {
    runtime.setAndCoerceProperty$Ex(Lit61, Lit27, Lit126, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit61, Lit38, "a.png", Lit8);
    runtime.setAndCoerceProperty$Ex(Lit61, Lit60, Boolean.FALSE,
Lit12);
    return runtime.setAndCoerceProperty$Ex(Lit61, Lit29, Lit127,
Lit6);
}

static Object lambda40() {
    runtime.setAndCoerceProperty$Ex(Lit61, Lit27, Lit126, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit61, Lit38, "a.png", Lit8);
    runtime.setAndCoerceProperty$Ex(Lit61, Lit60, Boolean.FALSE,
Lit12);
    return runtime.setAndCoerceProperty$Ex(Lit61, Lit29, Lit127,
Lit6);
}

public Object Button8$Click() {
    runtime.setThisForm();
    runtime.callComponentMethod(Lit41, Lit51,
LList.list1(runtime.callYailPrimitive(strings.string$Mnappend,
LList.list2("b", "4"), Lit129, "join")), Lit130);
    return runtime.setAndCoerceProperty$Ex(Lit111, Lit56, "4",
Lit8);
}

static Object lambda41() {
    runtime.setAndCoerceProperty$Ex(Lit133, Lit4, Lit5, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit133, Lit25, Lit26, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit133, Lit27, Lit134, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit133, Lit29, Lit28,
Lit6);
}

static Object lambda42() {
    runtime.setAndCoerceProperty$Ex(Lit133, Lit4, Lit5, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit133, Lit25, Lit26, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit133, Lit27, Lit134, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit133, Lit29, Lit28,
Lit6);
}
```



Digitalización de un banco de ensayos y comprobación experimental

```
static Object lambda43() {
    runtime.setAndCoerceProperty$Ex(Lit137, Lit56, "K Hall",
Lit8);
    return runtime.setAndCoerceProperty$Ex(Lit137, Lit29, Lit138,
Lit6);
}

static Object lambda44() {
    runtime.setAndCoerceProperty$Ex(Lit137, Lit56, "K Hall",
Lit8);
    return runtime.setAndCoerceProperty$Ex(Lit137, Lit29, Lit138,
Lit6);
}

static Object lambda45() {
    runtime.setAndCoerceProperty$Ex(Lit141, Lit77, Boolean.TRUE,
Lit12);
    runtime.setAndCoerceProperty$Ex(Lit141, Lit78, Lit79, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit141, Lit29, Lit142,
Lit6);
}

static Object lambda46() {
    runtime.setAndCoerceProperty$Ex(Lit141, Lit77, Boolean.TRUE,
Lit12);
    runtime.setAndCoerceProperty$Ex(Lit141, Lit78, Lit79, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit141, Lit29, Lit142,
Lit6);
}

static Object lambda47() {
    return runtime.setAndCoerceProperty$Ex(Lit145, Lit56, " ",
Lit8);
}

static Object lambda48() {
    return runtime.setAndCoerceProperty$Ex(Lit145, Lit56, " ",
Lit8);
}

static Object lambda49() {
    runtime.setAndCoerceProperty$Ex(Lit19, Lit86, Lit79, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit19, Lit56, "Enviar",
Lit8);
}

static Object lambda50() {
    runtime.setAndCoerceProperty$Ex(Lit19, Lit86, Lit79, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit19, Lit56, "Enviar",
Lit8);
}

public Object Button1$Click() {
    runtime.setThisForm();
    return runtime.callComponentMethod(Lit41, Lit51,
```

Digitalización de un banco de ensayos y comprobación experimental

```
LList.list1(runtime.callYailPrimitive(strings.string$Mnappend,  
LList.list2("c", runtime.getProperty$1(Lit141, Lit56)), Lit149,  
"join")), Lit150);  
}  
  
static Object lambda51() {  
    return runtime.setAndCoerceProperty$Ex(Lit153, Lit56, " ",  
Lit8);  
}  
  
static Object lambda52() {  
    return runtime.setAndCoerceProperty$Ex(Lit153, Lit56, " ",  
Lit8);  
}  
  
static Object lambda53() {  
    runtime.setAndCoerceProperty$Ex(Lit62, Lit27, Lit156, Lit6);  
    runtime.setAndCoerceProperty$Ex(Lit62, Lit38, "a.png", Lit8);  
    runtime.setAndCoerceProperty$Ex(Lit62, Lit60, Boolean.FALSE,  
Lit12);  
    return runtime.setAndCoerceProperty$Ex(Lit62, Lit29, Lit157,  
Lit6);  
}  
  
static Object lambda54() {  
    runtime.setAndCoerceProperty$Ex(Lit62, Lit27, Lit156, Lit6);  
    runtime.setAndCoerceProperty$Ex(Lit62, Lit38, "a.png", Lit8);  
    runtime.setAndCoerceProperty$Ex(Lit62, Lit60, Boolean.FALSE,  
Lit12);  
    return runtime.setAndCoerceProperty$Ex(Lit62, Lit29, Lit157,  
Lit6);  
}  
  
public Object Button9$Click() {  
    runtime.setThisForm();  
    runtime.callComponentMethod(Lit41, Lit51,  
LList.list1(runtime.callYailPrimitive(strings.string$Mnappend,  
LList.list2("c", "1"), Lit159, "join")), Lit160);  
    return runtime.setAndCoerceProperty$Ex(Lit141, Lit56, "1",  
Lit8);  
}  
  
static Object lambda55() {  
    runtime.setAndCoerceProperty$Ex(Lit163, Lit4, Lit5, Lit6);  
    runtime.setAndCoerceProperty$Ex(Lit163, Lit25, Lit26, Lit6);  
    runtime.setAndCoerceProperty$Ex(Lit163, Lit27, Lit164, Lit6);  
    return runtime.setAndCoerceProperty$Ex(Lit163, Lit29, Lit28,  
Lit6);  
}  
  
static Object lambda56() {  
    runtime.setAndCoerceProperty$Ex(Lit163, Lit4, Lit5, Lit6);  
    runtime.setAndCoerceProperty$Ex(Lit163, Lit25, Lit26, Lit6);  
    runtime.setAndCoerceProperty$Ex(Lit163, Lit27, Lit164, Lit6);  
    return runtime.setAndCoerceProperty$Ex(Lit163, Lit29, Lit28,
```




Digitalización de un banco de ensayos y comprobación experimental

```
Lit6);
}

static Object lambda57() {
runtime.setAndCoerceProperty$Ex(Lit167, Lit56, "K Pinza",
Lit8);
return runtime.setAndCoerceProperty$Ex(Lit167, Lit29, Lit168,
Lit6);
}

static Object lambda58() {
runtime.setAndCoerceProperty$Ex(Lit167, Lit56, "K Pinza",
Lit8);
return runtime.setAndCoerceProperty$Ex(Lit167, Lit29, Lit168,
Lit6);
}

/* compiled from: Screen1.yail */
public class frame extends ModuleBody {
Screen1 $main;

public int match1(ModuleMethod moduleMethod, Object obj,
CallContext callContext) {
switch (moduleMethod.selector) {
case 1:
callContext.value1 = obj;
callContext.proc = moduleMethod;
callContext.pc = 1;
return 0;
case 2:
if (!(obj instanceof Screen1)) {
return -786431;
}
callContext.value1 = obj;
callContext.proc = moduleMethod;
callContext.pc = 1;
return 0;
case 3:
callContext.value1 = obj;
callContext.proc = moduleMethod;
callContext.pc = 1;
return 0;
case 5:
if (!(obj instanceof Symbol)) {
return -786431;
}
callContext.value1 = obj;
callContext.proc = moduleMethod;
callContext.pc = 1;
return 0;
case 7:
if (!(obj instanceof Symbol)) {
return -786431;
}
callContext.value1 = obj;
```

Digitalización de un banco de ensayos y comprobación experimental

```
        callContext.proc = moduleMethod;
        callContext.pc = 1;
        return 0;
    case 12:
        callContext.value1 = obj;
        callContext.proc = moduleMethod;
        callContext.pc = 1;
        return 0;
    case 13:
        callContext.value1 = obj;
        callContext.proc = moduleMethod;
        callContext.pc = 1;
        return 0;
    case 14:
        if (!(obj instanceof Screen1)) {
            return -786431;
        }
        callContext.value1 = obj;
        callContext.proc = moduleMethod;
        callContext.pc = 1;
        return 0;
    default:
        return super.match1(moduleMethod, obj,
callContext);
    }
}

    public int match2(ModuleMethod moduleMethod, Object obj,
Object obj2, CallContext callContext) {
        switch (moduleMethod.selector) {
            case 4:
                if (!(obj instanceof Symbol)) {
                    return -786431;
                }
                callContext.value1 = obj;
                callContext.value2 = obj2;
                callContext.proc = moduleMethod;
                callContext.pc = 2;
                return 0;
            case 5:
                if (!(obj instanceof Symbol)) {
                    return -786431;
                }
                callContext.value1 = obj;
                callContext.value2 = obj2;
                callContext.proc = moduleMethod;
                callContext.pc = 2;
                return 0;
            case 8:
                if (!(obj instanceof Symbol)) {
                    return -786431;
                }
                callContext.value1 = obj;
                callContext.value2 = obj2;
                callContext.proc = moduleMethod;
```

Digitalización de un banco de ensayos y comprobación experimental

```
        callContext.pc = 2;
        return 0;
    case 9:
        callContext.value1 = obj;
        callContext.value2 = obj2;
        callContext.proc = moduleMethod;
        callContext.pc = 2;
        return 0;
    case 11:
        callContext.value1 = obj;
        callContext.value2 = obj2;
        callContext.proc = moduleMethod;
        callContext.pc = 2;
        return 0;
    case 17:
        callContext.value1 = obj;
        callContext.value2 = obj2;
        callContext.proc = moduleMethod;
        callContext.pc = 2;
        return 0;
    default:
        return super.match2(moduleMethod, obj, obj2,
callContext);
    }
}

    public int match4(ModuleMethod moduleMethod, Object obj,
Object obj2, Object obj3, Object obj4, CallContext callContext) {
        switch (moduleMethod.selector) {
            case 10:
                callContext.value1 = obj;
                callContext.value2 = obj2;
                callContext.value3 = obj3;
                callContext.value4 = obj4;
                callContext.proc = moduleMethod;
                callContext.pc = 4;
                return 0;
            case 15:
                if (!(obj instanceof Screen1)) {
                    return -786431;
                }
                callContext.value1 = obj;
                if (!(obj2 instanceof Component)) {
                    return -786430;
                }
                callContext.value2 = obj2;
                if (!(obj3 instanceof String)) {
                    return -786429;
                }
                callContext.value3 = obj3;
                if (!(obj4 instanceof String)) {
                    return -786428;
                }
                callContext.value4 = obj4;
                callContext.proc = moduleMethod;
```

Digitalización de un banco de ensayos y comprobación experimental

```
        callContext.pc = 4;
        return 0;
    case 16:
        if (!(obj instanceof Screen1)) {
            return -786431;
        }
        callContext.value1 = obj;
        if (!(obj2 instanceof Component)) {
            return -786430;
        }
        callContext.value2 = obj2;
        if (!(obj3 instanceof String)) {
            return -786429;
        }
        callContext.value3 = obj3;
        callContext.value4 = obj4;
        callContext.proc = moduleMethod;
        callContext.pc = 4;
        return 0;
    default:
        return super.match4(moduleMethod, obj, obj2, obj3,
obj4, callContext);
    }
}

public Object apply1(ModuleMethod moduleMethod, Object obj) {
    switch (moduleMethod.selector) {
        case 1:
            return this.$main.getSimpleName(obj);
        case 2:
            try {
                this.$main.onCreate((Bundle) obj);
                return Values.empty;
            } catch (ClassCastException e) {
                throw new WrongType(e, "onCreate", 1, obj);
            }
        case 3:
            this.$main.androidLogForm(obj);
            return Values.empty;
        case 5:
            try {
                return
this.$main.lookupInFormEnvironment((Symbol) obj);
            } catch (ClassCastException e2) {
                throw new WrongType(e2, "lookup-in-form-
environment", 1, obj);
            }
        case 7:
            try {
                return
this.$main.isBoundInFormEnvironment((Symbol) obj) ? Boolean.TRUE :
Boolean.FALSE;
            } catch (ClassCastException e3) {
                throw new WrongType(e3, "is-bound-in-form-
environment", 1, obj);
            }
    }
}
```

Digitalización de un banco de ensayos y comprobación experimental

```
    }  
    case 12:  
        this.$main.addToFormDoAfterCreation(obj);  
        return Values.empty;  
    case 13:  
        this.$main.sendError(obj);  
        return Values.empty;  
    case 14:  
        this.$main.processException(obj);  
        return Values.empty;  
    default:  
        return super.apply1(moduleMethod, obj);  
    }  
}  
  
    public Object apply4(ModuleMethod moduleMethod, Object obj,  
Object obj2, Object obj3, Object obj4) {  
    boolean z = true;  
    switch (moduleMethod.selector) {  
        case 10:  
            this.$main.addToComponents(obj, obj2, obj3, obj4);  
            return Values.empty;  
        case 15:  
            try {  
                try {  
                    try {  
                        return  
this.$main.dispatchEvent((Component) obj, (String) obj2, (String)  
obj3, (Object[]) obj4) ? Boolean.TRUE : Boolean.FALSE;  
                    } catch (ClassCastException e) {  
                        throw new WrongType(e,  
"dispatchEvent", 4, obj4);  
                    }  
                } catch (ClassCastException e2) {  
                    throw new WrongType(e2,  
"dispatchEvent", 3, obj3);  
                }  
            } catch (ClassCastException e3) {  
                throw new WrongType(e3, "dispatchEvent",  
2, obj2);  
            }  
        } catch (ClassCastException e4) {  
            throw new WrongType(e4, "dispatchEvent", 1,  
obj);  
        }  
    case 16:  
        Screen1 screen1 = this.$main;  
        try {  
            Component component = (Component) obj;  
            try {  
                String str = (String) obj2;  
                try {  
                    if (obj3 == Boolean.FALSE) {  
                        z = false;  
                    }  
                }  
            }  
        }  
    }  
}
```

Digitalización de un banco de ensayos y comprobación experimental

```
    }
    try {
screen1.dispatchGenericEvent(component, str, z, (Object[]) obj4);
        return Values.empty;
    } catch (ClassCastException e5) {
        throw new WrongType(e5,
"dispatchGenericEvent", 4, obj4);
    }
    } catch (ClassCastException e6) {
        throw new WrongType(e6,
"dispatchGenericEvent", 3, obj3);
    }
    } catch (ClassCastException e7) {
        throw new WrongType(e7,
"dispatchGenericEvent", 2, obj2);
    }
    } catch (ClassCastException e8) {
        throw new WrongType(e8,
"dispatchGenericEvent", 1, obj);
    }
    default:
        return super.apply4(moduleMethod, obj, obj2, obj3,
obj4);
    }
}

    public Object apply2(ModuleMethod moduleMethod, Object obj,
Object obj2) {
        switch (moduleMethod.selector) {
            case 4:
                try {
                    this.$main.addToFormEnvironment((Symbol) obj,
obj2);
                    return Values.empty;
                } catch (ClassCastException e) {
                    throw new WrongType(e, "add-to-form-
environment", 1, obj);
                }
            case 5:
                try {
                    return
this.$main.lookupInFormEnvironment((Symbol) obj, obj2);
                } catch (ClassCastException e2) {
                    throw new WrongType(e2, "lookup-in-form-
environment", 1, obj);
                }
            case 8:
                try {
                    this.$main.addToGlobalVarEnvironment((Symbol)
obj, obj2);
                    return Values.empty;
                } catch (ClassCastException e3) {
                    throw new WrongType(e3, "add-to-global-var-
environment", 1, obj);
                }
            default:
                return super.apply2(moduleMethod, obj, obj2);
        }
    }
}
```

Digitalización de un banco de ensayos y comprobación experimental

```
    }  
    case 9:  
        this.$main.addToEvents(obj, obj2);  
        return Values.empty;  
    case 11:  
        this.$main.addToGlobalVars(obj, obj2);  
        return Values.empty;  
    case 17:  
        return this.$main.lookupHandler(obj, obj2);  
    default:  
        return super.apply2(moduleMethod, obj, obj2);  
    }  
}  
  
public Object apply0(ModuleMethod moduleMethod) {  
    switch (moduleMethod.selector) {  
        case 18:  
            return Screen1.lambda2();  
        case 19:  
            this.$main.$define();  
            return Values.empty;  
        case 20:  
            return Screen1.lambda3();  
        case 21:  
            return Screen1.lambda4();  
        case 22:  
            return this.$main.Screen1$Initialize();  
        case 23:  
            return Screen1.lambda5();  
        case 24:  
            return Screen1.lambda6();  
        case 25:  
            return Screen1.lambda7();  
        case 26:  
            return Screen1.lambda8();  
        case 27:  
            return Screen1.lambda9();  
        case 28:  
            return Screen1.lambda10();  
        case 29:  
            return this.$main.ListPicker1$BeforePicking();  
        case 30:  
            return this.$main.ListPicker1$AfterPicking();  
        case 31:  
            return Screen1.lambda11();  
        case 32:  
            return Screen1.lambda12();  
        case 33:  
            return this.$main.Switch1$Changed();  
        case 34:  
            return Screen1.lambda13();  
        case 35:  
            return Screen1.lambda14();  
        case 36:  
            return Screen1.lambda15();  
    }  
}
```



Digitalización de un banco de ensayos y comprobación experimental

```
case 37:
    return Screen1.lambda16 ();
case 38:
    return Screen1.lambda17 ();
case 39:
    return Screen1.lambda18 ();
case 40:
    return Screen1.lambda19 ();
case 41:
    return Screen1.lambda20 ();
case 42:
    return Screen1.lambda21 ();
case 43:
    return Screen1.lambda22 ();
case 44:
    return this.$main.Button3$Click ();
case 45:
    return Screen1.lambda23 ();
case 46:
    return Screen1.lambda24 ();
case 47:
    return Screen1.lambda25 ();
case 48:
    return Screen1.lambda26 ();
case 49:
    return this.$main.Button7$Click ();
case 50:
    return Screen1.lambda27 ();
case 51:
    return Screen1.lambda28 ();
case 52:
    return Screen1.lambda29 ();
case 53:
    return Screen1.lambda30 ();
case 54:
    return Screen1.lambda31 ();
case 55:
    return Screen1.lambda32 ();
case 56:
    return Screen1.lambda33 ();
case 57:
    return Screen1.lambda34 ();
case 58:
    return Screen1.lambda35 ();
case 59:
    return Screen1.lambda36 ();
case 60:
    return this.$main.Button2$Click ();
case 61:
    return Screen1.lambda37 ();
case 62:
    return Screen1.lambda38 ();
case 63:
    return Screen1.lambda39 ();
case 64:
```




Digitalización de un banco de ensayos y comprobación experimental

```
        return Screen1.lambda40 ();
    case 65:
        return this.$main.Button8$Click ();
    case 66:
        return Screen1.lambda41 ();
    case 67:
        return Screen1.lambda42 ();
    case 68:
        return Screen1.lambda43 ();
    case 69:
        return Screen1.lambda44 ();
    case 70:
        return Screen1.lambda45 ();
    case 71:
        return Screen1.lambda46 ();
    case 72:
        return Screen1.lambda47 ();
    case 73:
        return Screen1.lambda48 ();
    case 74:
        return Screen1.lambda49 ();
    case 75:
        return Screen1.lambda50 ();
    case 76:
        return this.$main.Button1$Click ();
    case 77:
        return Screen1.lambda51 ();
    case 78:
        return Screen1.lambda52 ();
    case 79:
        return Screen1.lambda53 ();
    case 80:
        return Screen1.lambda54 ();
    case 81:
        return this.$main.Button9$Click ();
    case 82:
        return Screen1.lambda55 ();
    case 83:
        return Screen1.lambda56 ();
    case 84:
        return Screen1.lambda57 ();
    case 85:
        return Screen1.lambda58 ();
    case 86:
        return Screen1.lambda59 ();
    case 87:
        return Screen1.lambda60 ();
    case 88:
        return Screen1.lambda61 ();
    case 89:
        return Screen1.lambda62 ();
    case 90:
        return Screen1.lambda63 ();
    case 91:
        return Screen1.lambda64 ();
```



Digitalización de un banco de ensayos y comprobación experimental

```
case 92:
    return this.$main.Button4$Click();
case 93:
    return Screen1.lambda65();
case 94:
    return Screen1.lambda66();
case 95:
    return Screen1.lambda67();
case 96:
    return Screen1.lambda68();
case 97:
    return this.$main.Button10$Click();
case 98:
    return Screen1.lambda69();
case 99:
    return Screen1.lambda70();
case 100:
    return this.$main.Button5$Click();
case 101:
    return Screen1.lambda71();
case 102:
    return Screen1.lambda72();
case 103:
    return Screen1.lambda73();
case 104:
    return Screen1.lambda74();
case 105:
    return Screen1.lambda75();
case 106:
    return Screen1.lambda76();
case 107:
    return Screen1.lambda77();
case 108:
    return Screen1.lambda78();
case 109:
    return Screen1.lambda79();
case 110:
    return Screen1.lambda80();
case 111:
    return Screen1.lambda81();
case 112:
    return Screen1.lambda82();
case 113:
    return Screen1.lambda83();
case 114:
    return Screen1.lambda84();
case 115:
    return Screen1.lambda85();
case 116:
    return Screen1.lambda86();
case 117:
    return Screen1.lambda87();
case 118:
    return Screen1.lambda88();
case 119:
```

Digitalización de un banco de ensayos y comprobación experimental

```
        return this.$main.Clock1$Timer();
    default:
        return super.apply0(moduleMethod);
    }
}

public int match0(ModuleMethod moduleMethod, CallContext
callContext) {
    switch (moduleMethod.selector) {
        case 18:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 19:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 20:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 21:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 22:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 23:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 24:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 25:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 26:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 27:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 28:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 29:
            callContext.proc = moduleMethod;
```

Digitalización de un banco de ensayos y comprobación experimental

```
callContext.pc = 0;
return 0;
case 30:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 31:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 32:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 33:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 34:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 35:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 36:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 37:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 38:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 39:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 40:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 41:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 42:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 43:
```



Digitalización de un banco de ensayos y comprobación experimental

```
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 44:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 45:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 46:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 47:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 48:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 49:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 50:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 51:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 52:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 53:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 54:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 55:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 56:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
```

Digitalización de un banco de ensayos y comprobación experimental

```
case 57:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
    return 0;
case 58:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
    return 0;
case 59:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
    return 0;
case 60:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
    return 0;
case 61:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
    return 0;
case 62:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
    return 0;
case 63:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
    return 0;
case 64:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
    return 0;
case 65:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
    return 0;
case 66:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
    return 0;
case 67:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
    return 0;
case 68:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
    return 0;
case 69:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
    return 0;
case 70:
    callContext.proc = moduleMethod;
    callContext.pc = 0;
```

Digitalización de un banco de ensayos y comprobación experimental

```
        return 0;
    case 71:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 72:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 73:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 74:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 75:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 76:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 77:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 78:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 79:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 80:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 81:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 82:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 83:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 84:
        callContext.proc = moduleMethod;
```

Digitalización de un banco de ensayos y comprobación experimental

```
        callContext.pc = 0;
        return 0;
    case 85:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 86:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 87:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 88:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 89:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 90:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 91:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 92:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 93:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 94:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 95:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 96:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 97:
        callContext.proc = moduleMethod;
        callContext.pc = 0;
        return 0;
    case 98:
```


Digitalización de un banco de ensayos y comprobación experimental

```
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 99:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 100:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 101:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 102:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 103:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 104:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 105:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 106:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 107:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 108:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 109:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 110:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
case 111:
callContext.proc = moduleMethod;
callContext.pc = 0;
return 0;
```

Digitalización de un banco de ensayos y comprobación experimental

```
        case 112:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 113:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 114:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 115:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 116:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 117:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 118:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        case 119:
            callContext.proc = moduleMethod;
            callContext.pc = 0;
            return 0;
        default:
            return super.match0(moduleMethod, callContext);
    }
}

static Object lambda59() {
    runtime.setAndCoerceProperty$Ex(Lit171, Lit77, Boolean.TRUE,
Lit12);
    runtime.setAndCoerceProperty$Ex(Lit171, Lit78, Lit79, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit171, Lit29, Lit172,
Lit6);
}

static Object lambda60() {
    runtime.setAndCoerceProperty$Ex(Lit171, Lit77, Boolean.TRUE,
Lit12);
    runtime.setAndCoerceProperty$Ex(Lit171, Lit78, Lit79, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit171, Lit29, Lit172,
Lit6);
}

static Object lambda61() {
```

Digitalización de un banco de ensayos y comprobación experimental

```
        return runtime.setAndCoerceProperty$Ex(Lit175, Lit56, " ",
Lit8);
    }

    static Object lambda62() {
        return runtime.setAndCoerceProperty$Ex(Lit175, Lit56, " ",
Lit8);
    }

    static Object lambda63() {
        runtime.setAndCoerceProperty$Ex(Lit20, Lit86, Lit79, Lit6);
        return runtime.setAndCoerceProperty$Ex(Lit20, Lit56, "Enviar",
Lit8);
    }

    static Object lambda64() {
        runtime.setAndCoerceProperty$Ex(Lit20, Lit86, Lit79, Lit6);
        return runtime.setAndCoerceProperty$Ex(Lit20, Lit56, "Enviar",
Lit8);
    }

    public Object Button4$Click() {
        runtime.setThisForm();
        return runtime.callComponentMethod(Lit41, Lit51,
LList.list1(runtime.callYailPrimitive(strings.string$Mnappend,
LList.list2("d", runtime.getProperty$1(Lit171, Lit56)), Lit179,
"join")), Lit180);
    }

    static Object lambda65() {
        return runtime.setAndCoerceProperty$Ex(Lit183, Lit56, " ",
Lit8);
    }

    static Object lambda66() {
        return runtime.setAndCoerceProperty$Ex(Lit183, Lit56, " ",
Lit8);
    }

    static Object lambda67() {
        runtime.setAndCoerceProperty$Ex(Lit63, Lit27, Lit186, Lit6);
        runtime.setAndCoerceProperty$Ex(Lit63, Lit38, "a.png", Lit8);
        runtime.setAndCoerceProperty$Ex(Lit63, Lit60, Boolean.FALSE,
Lit12);
        return runtime.setAndCoerceProperty$Ex(Lit63, Lit29, Lit187,
Lit6);
    }

    static Object lambda68() {
        runtime.setAndCoerceProperty$Ex(Lit63, Lit27, Lit186, Lit6);
        runtime.setAndCoerceProperty$Ex(Lit63, Lit38, "a.png", Lit8);
        runtime.setAndCoerceProperty$Ex(Lit63, Lit60, Boolean.FALSE,
Lit12);
        return runtime.setAndCoerceProperty$Ex(Lit63, Lit29, Lit187,
Lit6);
    }
}
```

Digitalización de un banco de ensayos y comprobación experimental

```
}

public Object Button10$Click() {
    runtime.setThisForm();
    runtime.callComponentMethod(Lit41, Lit51,
LList.list1(runtime.callYailPrimitive(strings.string$Mnappend,
LList.list2("d", "1"), Lit189, "join")), Lit190);
    return runtime.setAndCoerceProperty$Ex(Lit171, Lit56, "1",
Lit8);
}

static Object lambda69() {
    return runtime.setAndCoerceProperty$Ex(Lit196, Lit56, "Obtener
Parámetros", Lit8);
}

static Object lambda70() {
    return runtime.setAndCoerceProperty$Ex(Lit196, Lit56, "Obtener
Parámetros", Lit8);
}

public Object Button5$Click() {
    runtime.setThisForm();
    return runtime.callComponentMethod(Lit41, Lit51,
LList.list1("getParam"), Lit198);
}

static Object lambda71() {
    return runtime.setAndCoerceProperty$Ex(Lit201, Lit56, " ",
Lit8);
}

static Object lambda72() {
    return runtime.setAndCoerceProperty$Ex(Lit201, Lit56, " ",
Lit8);
}

static Object lambda73() {
    runtime.setAndCoerceProperty$Ex(Lit204, Lit4, Lit5, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit204, Lit25, Lit26, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit204, Lit27, Lit205, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit204, Lit29, Lit28,
Lit6);
}

static Object lambda74() {
    runtime.setAndCoerceProperty$Ex(Lit204, Lit4, Lit5, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit204, Lit25, Lit26, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit204, Lit27, Lit205, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit204, Lit29, Lit28,
Lit6);
}

static Object lambda75() {
    runtime.setAndCoerceProperty$Ex(Lit208, Lit209, Boolean.TRUE,
```



Digitalización de un banco de ensayos y comprobación experimental

```
Lit12);
    runtime.setAndCoerceProperty$Ex(Lit208, Lit210, Lit211, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit208, Lit27, Lit212, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit208, Lit56,
"Velocidad: ", Lit8);
}

    static Object lambda76() {
    runtime.setAndCoerceProperty$Ex(Lit208, Lit209, Boolean.TRUE,
Lit12);
    runtime.setAndCoerceProperty$Ex(Lit208, Lit210, Lit211, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit208, Lit27, Lit212, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit208, Lit56,
"Velocidad: ", Lit8);
}

    static Object lambda77() {
    runtime.setAndCoerceProperty$Ex(Lit215, Lit209, Boolean.TRUE,
Lit12);
    runtime.setAndCoerceProperty$Ex(Lit215, Lit210, Lit211, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit215, Lit27, Lit216, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit215, Lit56, "0",
Lit8);
}

    static Object lambda78() {
    runtime.setAndCoerceProperty$Ex(Lit215, Lit209, Boolean.TRUE,
Lit12);
    runtime.setAndCoerceProperty$Ex(Lit215, Lit210, Lit211, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit215, Lit27, Lit216, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit215, Lit56, "0",
Lit8);
}

    static Object lambda79() {
    runtime.setAndCoerceProperty$Ex(Lit219, Lit209, Boolean.TRUE,
Lit12);
    runtime.setAndCoerceProperty$Ex(Lit219, Lit210, Lit211, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit219, Lit27, Lit220, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit219, Lit56, "rpm",
Lit8);
}

    static Object lambda80() {
    runtime.setAndCoerceProperty$Ex(Lit219, Lit209, Boolean.TRUE,
Lit12);
    runtime.setAndCoerceProperty$Ex(Lit219, Lit210, Lit211, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit219, Lit27, Lit220, Lit6);
    return runtime.setAndCoerceProperty$Ex(Lit219, Lit56, "rpm",
Lit8);
}

    static Object lambda81() {
    runtime.setAndCoerceProperty$Ex(Lit223, Lit4, Lit5, Lit6);
    runtime.setAndCoerceProperty$Ex(Lit223, Lit25, Lit26, Lit6);
```



Digitalización de un banco de ensayos y comprobación experimental

```
runtime.setAndCoerceProperty$Ex(Lit223, Lit27, Lit224, Lit6);  
return runtime.setAndCoerceProperty$Ex(Lit223, Lit29, Lit28,  
Lit6);  
}  
  
static Object lambda82() {  
runtime.setAndCoerceProperty$Ex(Lit223, Lit4, Lit5, Lit6);  
runtime.setAndCoerceProperty$Ex(Lit223, Lit25, Lit26, Lit6);  
runtime.setAndCoerceProperty$Ex(Lit223, Lit27, Lit224, Lit6);  
return runtime.setAndCoerceProperty$Ex(Lit223, Lit29, Lit28,  
Lit6);  
}  
  
static Object lambda83() {  
runtime.setAndCoerceProperty$Ex(Lit227, Lit209, Boolean.TRUE,  
Lit12);  
runtime.setAndCoerceProperty$Ex(Lit227, Lit210, Lit211, Lit6);  
runtime.setAndCoerceProperty$Ex(Lit227, Lit27, Lit228, Lit6);  
return runtime.setAndCoerceProperty$Ex(Lit227, Lit56, "Par:",  
Lit8);  
}  
  
static Object lambda84() {  
runtime.setAndCoerceProperty$Ex(Lit227, Lit209, Boolean.TRUE,  
Lit12);  
runtime.setAndCoerceProperty$Ex(Lit227, Lit210, Lit211, Lit6);  
runtime.setAndCoerceProperty$Ex(Lit227, Lit27, Lit228, Lit6);  
return runtime.setAndCoerceProperty$Ex(Lit227, Lit56, "Par:",  
Lit8);  
}  
  
static Object lambda85() {  
runtime.setAndCoerceProperty$Ex(Lit231, Lit209, Boolean.TRUE,  
Lit12);  
runtime.setAndCoerceProperty$Ex(Lit231, Lit210, Lit211, Lit6);  
runtime.setAndCoerceProperty$Ex(Lit231, Lit27, Lit232, Lit6);  
return runtime.setAndCoerceProperty$Ex(Lit231, Lit56, "0",  
Lit8);  
}  
  
static Object lambda86() {  
runtime.setAndCoerceProperty$Ex(Lit231, Lit209, Boolean.TRUE,  
Lit12);  
runtime.setAndCoerceProperty$Ex(Lit231, Lit210, Lit211, Lit6);  
runtime.setAndCoerceProperty$Ex(Lit231, Lit27, Lit232, Lit6);  
return runtime.setAndCoerceProperty$Ex(Lit231, Lit56, "0",  
Lit8);  
}  
  
static Object lambda87() {  
runtime.setAndCoerceProperty$Ex(Lit235, Lit209, Boolean.TRUE,  
Lit12);  
runtime.setAndCoerceProperty$Ex(Lit235, Lit210, Lit211, Lit6);  
runtime.setAndCoerceProperty$Ex(Lit235, Lit27, Lit236, Lit6);  
return runtime.setAndCoerceProperty$Ex(Lit235, Lit56, "Nm",
```

Digitalización de un banco de ensayos y comprobación experimental

```
Lit8);
}

static Object lambda88() {
runtime.setAndCoerceProperty$Ex(Lit235, Lit209, Boolean.TRUE,
Lit12);
runtime.setAndCoerceProperty$Ex(Lit235, Lit210, Lit211, Lit6);
runtime.setAndCoerceProperty$Ex(Lit235, Lit27, Lit236, Lit6);
return runtime.setAndCoerceProperty$Ex(Lit235, Lit56, "Nm",
Lit8);
}

public Object Clock1$Timer() {
runtime.setThisForm();
if (runtime.getProperty$1(Lit41, Lit243) == Boolean.FALSE) {
return Values.empty;
}
if (runtime.callYailPrimitive(Scheme.numGrt,
LList.list2(runtime.callComponentMethod(Lit41, Lit244, LList.Empty,
LList.Empty), Lit245), Lit246, ">") == Boolean.FALSE) {
return Values.empty;
}
runtime.addGlobalVarToCurrentFormEnvironment(Lit3,
runtime.callYailPrimitive(runtime.string$Mnsplit,
LList.list2(runtime.callComponentMethod(Lit41, Lit247,
LList.list1(runtime.callComponentMethod(Lit41, Lit244, LList.Empty,
LList.Empty), Lit248), "|"), Lit249, "split"));
if (runtime.callYailPrimitive(Scheme.numGEq,
LList.list2(runtime.callYailPrimitive(runtime.yail$Mnlist$Mnlength,
LList.list1(runtime.lookupGlobalVarInCurrentFormEnvironment(Lit3,
runtime.$Stthe$Mnnull$Mnvalue$St)), Lit250, "length of list"), Lit79),
Lit251, ">=") != Boolean.FALSE) {
runtime.setAndCoerceProperty$Ex(Lit215, Lit56,
runtime.callYailPrimitive(runtime.yail$Mnlist$Mnget$Mnitem,
LList.list2(runtime.lookupGlobalVarInCurrentFormEnvironment(Lit3,
runtime.$Stthe$Mnnull$Mnvalue$St), Lit79), Lit252, "select list
item"), Lit8);
}
if (runtime.callYailPrimitive(Scheme.numGEq,
LList.list2(runtime.callYailPrimitive(runtime.yail$Mnlist$Mnlength,
LList.list1(runtime.lookupGlobalVarInCurrentFormEnvironment(Lit3,
runtime.$Stthe$Mnnull$Mnvalue$St)), Lit253, "length of list"), Lit26),
Lit254, ">=") != Boolean.FALSE) {
runtime.setAndCoerceProperty$Ex(Lit231, Lit56,
runtime.callYailPrimitive(runtime.yail$Mnlist$Mnget$Mnitem,
LList.list2(runtime.lookupGlobalVarInCurrentFormEnvironment(Lit3,
runtime.$Stthe$Mnnull$Mnvalue$St), Lit26), Lit255, "select list
item"), Lit8);
}
if (runtime.callYailPrimitive(Scheme.numGEq,
LList.list2(runtime.callYailPrimitive(runtime.yail$Mnlist$Mnlength,
LList.list1(runtime.lookupGlobalVarInCurrentFormEnvironment(Lit3,
runtime.$Stthe$Mnnull$Mnvalue$St)), Lit256, "length of list"), Lit5),
Lit257, ">=") != Boolean.FALSE) {
runtime.setAndCoerceProperty$Ex(Lit76, Lit56,
```

Digitalización de un banco de ensayos y comprobación experimental

```
runtime.callYailPrimitive(runtime.yail$Mnlist$Mnget$Mnitem,  
LList.list2(runtime.lookupGlobalVarInCurrentFormEnvironment(Lit3,  
runtime.$Stthe$Mnnull$Mnvalue$St), Lit5), Lit258, "select list item"),  
Lit8);  
}  
if (runtime.callYailPrimitive(Scheme.numGEq,  
LList.list2(runtime.callYailPrimitive(runtime.yail$Mnlist$Mnlength,  
LList.list1(runtime.lookupGlobalVarInCurrentFormEnvironment(Lit3,  
runtime.$Stthe$Mnnull$Mnvalue$St)), Lit259, "length of list"),  
Lit260), Lit261, ">=") != Boolean.FALSE) {  
    runtime.setAndCoerceProperty$Ex(Lit111, Lit56,  
runtime.callYailPrimitive(runtime.yail$Mnlist$Mnget$Mnitem,  
LList.list2(runtime.lookupGlobalVarInCurrentFormEnvironment(Lit3,  
runtime.$Stthe$Mnnull$Mnvalue$St), Lit260), Lit262, "select list  
item"), Lit8);  
}  
if (runtime.callYailPrimitive(Scheme.numGEq,  
LList.list2(runtime.callYailPrimitive(runtime.yail$Mnlist$Mnlength,  
LList.list1(runtime.lookupGlobalVarInCurrentFormEnvironment(Lit3,  
runtime.$Stthe$Mnnull$Mnvalue$St)), Lit263, "length of list"),  
Lit264), Lit265, ">=") != Boolean.FALSE) {  
    runtime.setAndCoerceProperty$Ex(Lit141, Lit56,  
runtime.callYailPrimitive(runtime.yail$Mnlist$Mnget$Mnitem,  
LList.list2(runtime.lookupGlobalVarInCurrentFormEnvironment(Lit3,  
runtime.$Stthe$Mnnull$Mnvalue$St), Lit264), Lit266, "select list  
item"), Lit8);  
}  
if (runtime.callYailPrimitive(Scheme.numGEq,  
LList.list2(runtime.callYailPrimitive(runtime.yail$Mnlist$Mnlength,  
LList.list1(runtime.lookupGlobalVarInCurrentFormEnvironment(Lit3,  
runtime.$Stthe$Mnnull$Mnvalue$St)), Lit267, "length of list"),  
Lit268), Lit269, ">=") != Boolean.FALSE) {  
    runtime.setAndCoerceProperty$Ex(Lit171, Lit56,  
runtime.callYailPrimitive(runtime.yail$Mnlist$Mnget$Mnitem,  
LList.list2(runtime.lookupGlobalVarInCurrentFormEnvironment(Lit3,  
runtime.$Stthe$Mnnull$Mnvalue$St), Lit268), Lit270, "select list  
item"), Lit8);  
}  
return runtime.addGlobalVarToCurrentFormEnvironment(Lit3, "");  
}  
  
public String getSimpleName(Object object) {  
    return object.getClass().getSimpleName();  
}  
  
public void onCreate(Bundle icicle) {  
    AppInventorCompatActivity.setClassicModeFromYail(true);  
    super.onCreate(icicle);  
}  
  
public void androidLogForm(Object message) {  
}  
  
public void addToFormEnvironment(Symbol name, Object object) {  
    androidLogForm(Format.formatToString(0, "Adding ~A to env ~A
```


Digitalización de un banco de ensayos y comprobación experimental

```
with value ~A", name, this.form$Mnenvironment, object));
    this.form$Mnenvironment.put(name, object);
}

    public Object lookupInFormEnvironment(Symbol name, Object
default$Mnvalue) {
    boolean x = ((this.form$Mnenvironment == null ? 1 : 0) + 1) &
true;
    if (x) {
        if (!this.form$Mnenvironment.isBound(name)) {
            return default$Mnvalue;
        }
    } else if (!x) {
        return default$Mnvalue;
    }
    return this.form$Mnenvironment.get(name);
}

    public boolean isBoundInFormEnvironment(Symbol name) {
    return this.form$Mnenvironment.isBound(name);
}

    public void addToGlobalVarEnvironment(Symbol name, Object object)
{
    androidLogForm(Format.formatToString(0, "Adding ~A to env ~A
with value ~A", name, this.global$Mnvar$Mnenvironment, object));
    this.global$Mnvar$Mnenvironment.put(name, object);
}

    public void addToEvents(Object component$Mnname, Object
event$Mnname) {
    this.events$Mnto$Mnregister =
lists.cons(lists.cons(component$Mnname, event$Mnname),
this.events$Mnto$Mnregister);
}

    public void addToComponents(Object container$Mnname, Object
component$Mntype, Object component$Mnname, Object init$Mnthunk) {
    this.components$Mnto$Mncreate =
lists.cons(LList.list4(container$Mnname, component$Mntype,
component$Mnname, init$Mnthunk), this.components$Mnto$Mncreate);
}

    public void addToGlobalVars(Object var, Object val$Mnthunk) {
    this.global$Mnvars$Mnto$Mncreate = lists.cons(LList.list2(var,
val$Mnthunk), this.global$Mnvars$Mnto$Mncreate);
}

    public void addToFormDoAfterCreation(Object thunk) {
    this.form$Mndo$Mnafter$Mncreation = lists.cons(thunk,
this.form$Mndo$Mnafter$Mncreation);
}

    public void sendError(Object error) {
    RetValManager.sendError(error == null ? null :
```

Digitalización de un banco de ensayos y comprobación experimental

```
error.toString());
    }

    public void processException(Object ex) {
        Object apply1 =
Scheme.applyToArgs.apply1(GetNamedPart.getNamedPart.apply2(ex, Lit1));
        RuntimeErrorAlert.alert(this, apply1 == null ? null :
apply1.toString(), ex instanceof YailRuntimeError ?
((YailRuntimeError) ex).getErrorType() : "Runtime Error", "End
Application");
    }

    public boolean dispatchEvent(Component componentObject, String
registeredComponentName, String eventName, Object[] args) {
        boolean x;
        SimpleSymbol registeredObject =
misc.stringTo$Symbol(registeredComponentName);
        if (!isBoundInFormEnvironment(registeredObject)) {
            EventDispatcher.unregisterEventForDelegation(this,
registeredComponentName, eventName);
            return false;
        } else if (lookupInFormEnvironment(registeredObject) !=
componentObject) {
            return false;
        } else {
            try {
Scheme.apply.apply2(lookupHandler(registeredComponentName, eventName),
LList.makeList(args, 0));
                return true;
            } catch (PermissionException exception) {
                exception.printStackTrace();
                if (this == componentObject) {
                    x = true;
                } else {
                    x = false;
                }
                if (!x ? x : IsEqual.apply(eventName,
"PermissionNeeded")) {
                    processException(exception);
                } else {
                    PermissionDenied(componentObject, eventName,
exception.getPermissionNeeded());
                }
                return false;
            } catch (Throwable exception2) {
                androidLogForm(exception2.getMessage());
                exception2.printStackTrace();
                processException(exception2);
                return false;
            }
        }
    }

    public void dispatchGenericEvent(Component componentObject, String
```



Digitalización de un banco de ensayos y comprobación experimental

```
eventName, boolean notAlreadyHandled, Object[] args) {
    Boolean bool;
    boolean x = true;
    Object handler =
lookupInFormEnvironment(misc.string$To$Symbol(strings.stringAppend("any$",
getSimpleName(componentObject), "$", eventName)));
    if (handler != Boolean.FALSE) {
        try {
            Apply apply = Scheme.apply;
            if (notAlreadyHandled) {
                bool = Boolean.TRUE;
            } else {
                bool = Boolean.FALSE;
            }
            apply.apply2(handler, lists.cons(componentObject,
lists.cons(bool, LList.makeList(args, 0)));
        } catch (PermissionException exception) {
            exception.printStackTrace();
            if (this != componentObject) {
                x = false;
            }
            if (!x ? x : IsEqual.apply(eventName,
"PermissionNeeded")) {
                processException(exception);
            } else {
                PermissionDenied(componentObject, eventName,
exception.getPermissionNeeded());
            }
        } catch (Throwable exception2) {
            androidLogForm(exception2.getMessage());
            exception2.printStackTrace();
            processException(exception2);
        }
    }
}

public Object lookupHandler(Object componentName, Object
eventName) {
    String str = null;
    String obj = componentName == null ? null :
componentName.toString();
    if (eventName != null) {
        str = eventName.toString();
    }
    return
lookupInFormEnvironment(misc.string$To$Symbol(EventDispatcher.makeFull
EventName(obj, str)));
}

public void $define() {
    Object reverse;
    Object obj;
    Object reverse2;
    Object obj2;
    Object obj3;
```

Digitalización de un banco de ensayos y comprobación experimental

```
Object var;  
Object component$Mnname;  
Object obj4;  
Language.setDefaults(Scheme.getInstance());  
try {  
    run();  
} catch (Exception exception) {  
    androidLogForm(exception.getMessage());  
    processException(exception);  
}  
Screen1 = this;  
addToFormEnvironment(Lit0, this);  
Object obj5 = this.events$Mnto$Mnregister;  
while (obj5 != LList.Empty) {  
    try {  
        Pair arg0 = (Pair) obj5;  
        Object event$Mninfo = arg0.getCar();  
        Object apply1 = lists.car.apply1(event$Mninfo);  
        String obj6 = apply1 == null ? null :  
apply1.toString();  
        Object apply12 = lists.cdr.apply1(event$Mninfo);  
        EventDispatcher.registerEventForDelegation(this, obj6,  
apply12 == null ? null : apply12.toString());  
        obj5 = arg0.getCdr();  
    } catch (ClassCastException e) {  
        throw new WrongType(e, "arg0", -2, obj5);  
    }  
}  
try {  
    LList components =  
lists.reverse(this.components$Mnto$Mncreate);  
    addToGlobalVars(Lit2, lambda$Fn1);  
    reverse =  
lists.reverse(this.form$Mndo$Mnafter$Mncreation);  
    while (reverse != LList.Empty) {  
        Pair arg02 = (Pair) reverse;  
        misc.force(arg02.getCar());  
        reverse = arg02.getCdr();  
    }  
    obj = components;  
    while (obj != LList.Empty) {  
        Pair arg03 = (Pair) obj;  
        Object component$Mninfo = arg03.getCar();  
        component$Mnname =  
lists.caddr.apply1(component$Mninfo);  
        lists.caddr.apply1(component$Mninfo);  
        Object component$Mnobject =  
Invoke.make.apply2(lists.cadr.apply1(component$Mninfo),  
lookupInFormEnvironment((Symbol) lists.car.apply1(component$Mninfo)));  
        SlotSet.set$Mnfield$Ex.apply3(this, component$Mnname,  
component$Mnobject);  
        addToFormEnvironment((Symbol) component$Mnname,  
component$Mnobject);  
        obj = arg03.getCdr();  
    }  
}
```

Digitalización de un banco de ensayos y comprobación experimental

```
reverse2 =
lists.reverse(this.global$Mnvars$Mnto$Mncreate);
while (reverse2 != LList.Empty) {
  Pair arg04 = (Pair) reverse2;
  Object var$Mnval = arg04.getCar();
  var = lists.car.apply1(var$Mnval);
  addToGlobalVarEnvironment((Symbol) var,
Scheme.applyToArgs.apply1(lists.cadr.apply1(var$Mnval)));
  reverse2 = arg04.getCdr();
}
LList component$Mndescriptors = components;
obj2 = component$Mndescriptors;
while (obj2 != LList.Empty) {
  Pair arg05 = (Pair) obj2;
  Object component$Mninfo2 = arg05.getCar();
  lists.caddr.apply1(component$Mninfo2);
  Object init$Mnthunk =
lists.caddr.apply1(component$Mninfo2);
  if (init$Mnthunk != Boolean.FALSE) {
    Scheme.applyToArgs.apply1(init$Mnthunk);
  }
  obj2 = arg05.getCdr();
}
obj3 = component$Mndescriptors;
while (obj3 != LList.Empty) {
  Pair arg06 = (Pair) obj3;
  Object component$Mninfo3 = arg06.getCar();
  Object component$Mnname2 =
lists.caddr.apply1(component$Mninfo3);
  lists.caddr.apply1(component$Mninfo3);
  callInitialize(SlotGet.field.apply2(this,
component$Mnname2));
  obj3 = arg06.getCdr();
}
} catch (ClassCastException e2) {
  throw new WrongType(e2, "arg0", -2, obj3);
} catch (ClassCastException e3) {
  throw new WrongType(e3, "arg0", -2, obj2);
} catch (ClassCastException e4) {
  throw new WrongType(e4, "add-to-global-var-environment",
0, var);
} catch (ClassCastException e5) {
  throw new WrongType(e5, "arg0", -2, reverse2);
} catch (ClassCastException e6) {
  throw new WrongType(e6, "add-to-form-environment", 0,
component$Mnname);
} catch (ClassCastException e7) {
  throw new WrongType(e7, "lookup-in-form-environment", 0,
obj4);
} catch (ClassCastException e8) {
  throw new WrongType(e8, "arg0", -2, obj);
} catch (ClassCastException e9) {
  throw new WrongType(e9, "arg0", -2, reverse);
} catch (YailRuntimeError exception2) {
  processException(exception2);
}
```



Digitalización de un banco de ensayos y comprobación experimental

```
    }  
  }  
  
  public static SimpleSymbol lambdaSymbolAppend$V(Object[]  
argsArray) {  
    LList symbols = LList.makeList(argsArray, 0);  
    Apply apply = Scheme.apply;  
    ModuleMethod moduleMethod = strings.string$Mnappend;  
    Object obj = LList.Empty;  
    LList lList = symbols;  
    while (lList != LList.Empty) {  
      try {  
        Pair arg0 = (Pair) lList;  
        Object arg02 = arg0.getCdr();  
        Object car = arg0.getCar();  
        try {  
          obj = Pair.make(misc.symbol$To$string((Symbol)  
car), obj);  
          lList = arg02;  
        } catch (ClassCastException e) {  
          throw new WrongType(e, "symbol->string", 1, car);  
        }  
      } catch (ClassCastException e2) {  
        throw new WrongType(e2, "arg0", -2, lList);  
      }  
    }  
    Object apply2 = apply.apply2(moduleMethod,  
LList.reverseInPlace(obj));  
    try {  
      return misc.string$To$Symbol((CharSequence) apply2);  
    } catch (ClassCastException e3) {  
      throw new WrongType(e3, "string->symbol", 1, apply2);  
    }  
  }  
  
  static Object lambda2() {  
    return null;  
  }  
}
```