



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de un programa de control para una máquina de
ensayos biaxial

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: García de la Vega García, Assier

Tutor/a: Sánchez López, Miguel

CURSO ACADÉMICO: 2022/2023

Resumen

Este documento de final de carrera tratará de explicar los pasos a seguir para desarrollar una aplicación web de página única, que es capaz de utilizarse en diferentes los dispositivos.

Se trata de una interfaz de control y visualización de datos obtenidos a través de un Arduino conectado a una máquina de ensayos biaxial desarrollada para probar la resistencia mecánica de lienzos para el estudio de su conservación.

Palabras clave: Aplicación Web, multiplataforma, JavaScript, Arduino, puerto serie.

Abstract

This undergraduate tesis project document will try to explain the steps to follow in the development of a single page web application, that will work on multiples devices.

This application is an inteface that controls and displays data acquired from an Arduino board connected to a biaxial tensile tester created to study the mechanical aspects of cavas painting conservation.

Keywords: Web application, multi-platform, JavaScript, Arduino, serial port.

Tabla de contenidos

1. Introducción.....	6
1.1. Motivación	7
1.2. Objetivos.....	8
1.3. Estructura del documento.....	8
2. Máquina de ensayos biaxial	10
3. Estado del arte	13
4. Análisis del problema	14
5. Diseño de la solución	16
5.1 Arquitectura	16
5.2 Diagrama de casos de uso	17
5.3 Casos de uso	19
6. Tecnología utilizada.....	25
6.1. Vue.js.....	25
6.2. Github	27
6.3. Visual Studio Code.....	27
6.4. PLESK.....	27
7. Desarrollo de la solución.....	29
8. Implantación.....	40
9. Conclusiones	43
10. Trabajos futuros	44
11. Referencias	45
Anexo I: Objetivos de desarrollo sostenible	48



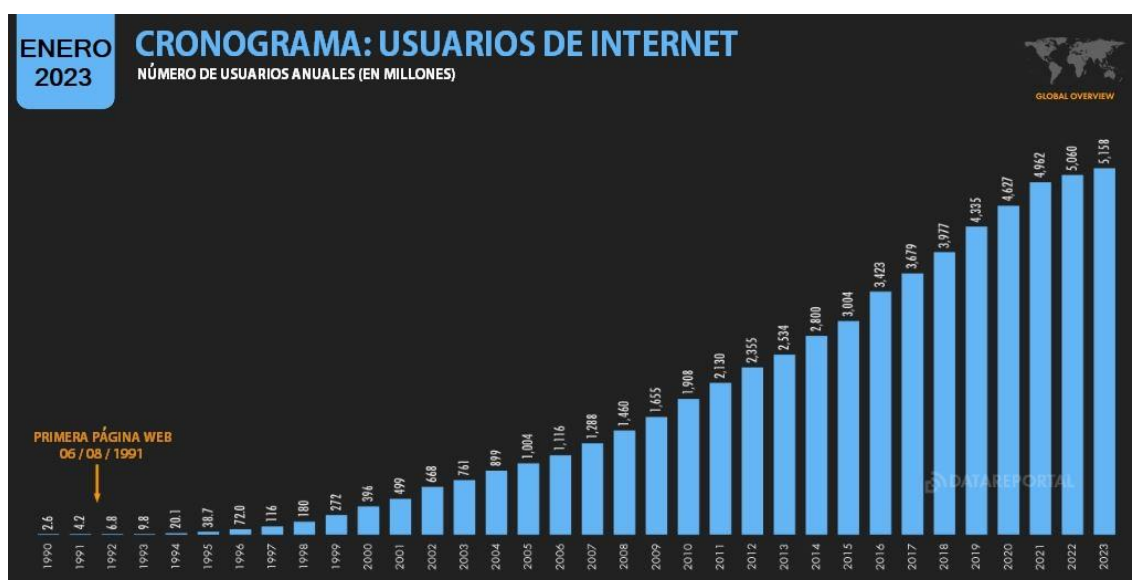
ÍNDICE DE GRÁFICAS, ILUSTRACIONES, TABLAS Y ESQUEMAS

Gráfica 1 : Cronología: Usuarios de Internet [2]	6
Ilustración 1 : Máquina de ensayos biaxial	11
Ilustración 2 : Muestra de lienzo	12
Ilustración 3 : Aplicación de Miguel Sánchez López	14
Ilustración 4 : Mensaje enviado por la aplicación	15
Ilustración 5 : Configuración Vue create	26
Ilustración 6 : Directorios de la aplicación	29
Ilustración 7 : Plantilla de un archivo .vue	30
Ilustración 8 : Barra de navegación	31
Ilustración 9 : Pantalla de bienvenida	31
Ilustración 10 : Pantalla de control y captura	32
Ilustración 11 : Pantalla de estudio de datos fuera de línea	32
Ilustración 12 : Pantalla terminal	33
Ilustración 13 : Dashboard de Plesk	40
Ilustración 14 : Configuración de Github en Plesk	41
Ilustración 15 : Configuración de una aplicación Node.js en Plesk	42
Tabla 1 : Caso de uso documentación	19
Tabla 2 : Caso de uso conexión	20
Tabla 3 : Caso de uso terminal	21
Tabla 4 : Caso de uso configuración	21
Tabla 5 : Caso de uso lanzar test	22
Tabla 6 : Caso de uso mover ejes	23
Tabla 7 : Caso de uso datos offline	24
Tabla 8 : Objetivos de Desarrollo Sostenible	48
Esquema 1 : Electrónica y flujo de datos	11
Esquema 2 : Petición-Respuesta PWA	17
Esquema 3 : Diagrama de casos de uso	18

1. Introducción

La pandemia de la COVID-19 ha dejado una marca indeleble en la vida tecnológica, transformando profundamente la forma en que el mundo interactúa con la tecnología [1]. Esta transformación ha sido especialmente notoria en los países más desarrollados, donde las restricciones de movimiento y el distanciamiento social llevaron a un aumento drástico en la dependencia de Internet para una amplia gama de actividades cotidianas; como trabajar, comunicarse con familiares y amigos, estudiar o incluso explorar nuevas aficiones.

La imagen de una sociedad encerrada en sus hogares, recurriendo a Internet como su principal vía de conexión con el mundo exterior, se ha vuelto una escena común en esta era post-pandémica. Tareas que antes se realizaban principalmente de manera presencial, como el trabajo, la comunicación con familiares y amigos, la educación y la exploración de nuevas aficiones, ahora se han trasladado en gran medida al ámbito digital. Esto ha sido posible gracias al continuo crecimiento del número de usuarios de Internet a lo largo del tiempo.



Gráfica 1 : Cronología: Usuarios de Internet [2]

Desde la creación de la primera página web, el número de usuarios de Internet ha experimentado un crecimiento incesante. En la actualidad, aproximadamente el 64,4% de la población mundial tiene acceso a Internet [3]. Esta expansión no solo se ha traducido en un aumento de la conectividad, sino también en una mayor diversidad de dispositivos disponibles para acceder a la red. Desde teléfonos móviles y tabletas hasta los tradicionales ordenadores de sobremesa y portátiles, la tecnología ha evolucionado para adaptarse a las necesidades de cada usuario.

Internet se ha convertido en un vasto y versátil recurso que atiende a una amplia gama de necesidades. Desde brindar entretenimiento a través de "memes" que alegran el estado de ánimo, hasta proporcionar acceso a información esencial para el estudio y la adquisición de nuevas habilidades que pueden cambiar la vida de una persona. Incluso en campos altamente especializados, como la ingeniería, Internet se ha convertido en una fuente invaluable de recursos, facilitando la obtención de aplicaciones de alta calidad para el manejo de equipos como las máquinas de ensayos biaxiales.

Además, no podemos pasar por alto la importancia de la iniciativa "Open Source" o software de código abierto en este panorama tecnológico en constante evolución. Este enfoque de desarrollo de software se basa en la colaboración abierta [4], lo que no solo significa proporcionar acceso al código fuente, sino también fomentar una comunidad de desarrolladores que trabajan de manera conjunta para mejorar y enriquecer el software. La accesibilidad a Internet ha facilitado la formación de equipos de trabajo altamente creativos y eficientes, lo que ha llevado al desarrollo de soluciones tecnológicas cada vez más avanzadas y accesibles para una amplia audiencia.

Quizás lo más destacable de esta evolución tecnológica es su capacidad para eliminar barreras económicas. Gracias a Internet y al software de código abierto, se están creando herramientas y recursos que pueden ser utilizados por personas de diversos orígenes económicos, permitiendo un acceso equitativo a la tecnología y sus beneficios.

En resumen, la pandemia de la COVID-19 ha acelerado la transformación de la vida tecnológica, impulsando un aumento en la dependencia de Internet y promoviendo la colaboración abierta a través del software de código abierto. Estos cambios están democratizando el acceso a la tecnología y abriendo nuevas oportunidades para personas de todo el mundo, allanando el camino hacia un futuro más conectado e inclusivo.

1.1. Motivación

Desde siempre, me he sentido fascinado por los proyectos de código libre, y me maravilla cómo personas sin una formación académica exhaustiva logran crear soluciones asombrosas al abordar problemas técnicos complejos. Mis intereses y aficiones se alinean con esta mentalidad, y a lo largo de los años he tenido la oportunidad de involucrarme en proyectos que han ampliado mi comprensión de la tecnología y su accesibilidad.

Hace años, ensamblé mi primera impresora 3D, una experiencia que me mostró el poder de la fabricación digital y la importancia de compartir conocimientos en comunidades de código abierto. Esta impresora 3D comparte piezas con la máquina de ensayos que vamos a discutir en este documento, lo que resalta la versatilidad y la interoperabilidad que se pueden lograr a través de la colaboración en proyectos de código abierto.

Además de la impresión 3D, durante algunos años he estado inmerso en el mundo de las aplicaciones web. Una de estas aplicaciones sigue principios y funcionamientos muy similares a la aplicación que estamos a punto de desarrollar, que se centrará en el control y la programación de microcomputadoras para drones de carreras. Mi experiencia en este campo me ha brindado una comprensión profunda de los desafíos y las oportunidades que ofrece el desarrollo de software orientado a dispositivos especializados.

Además, durante este último año, realizando las prácticas de empresa, he trabajado con uno de los Frameworks más novedosos para la creación de aplicaciones web, VueJS.

Por otro lado, me siento profundamente motivado por el propósito subyacente de este proyecto, que es crear un programa que pueda instalarse en diferentes sistemas operativos sin añadir complejidad innecesaria. Creo firmemente que todas las personas deben tener la oportunidad de explorar y experimentar con diversas tecnologías, sin que las limitaciones financieras sean un obstáculo. La accesibilidad y la democratización de la tecnología son principios fundamentales que guían mi enfoque hacia este proyecto, y estoy comprometido en hacer que estas soluciones estén al alcance de un público amplio y diverso.

1.2. Objetivos

El objetivo central de este proyecto es la creación de una aplicación versátil diseñada para el control y la visualización de datos obtenidos de una máquina de ensayos biaxial. Esta aplicación tiene la capacidad de gestionar los cuatro motores de la máquina y recopilar datos de los cuatro sensores, dos en cada eje.

Uno de los principios fundamentales que guían este proyecto es la aspiración de que la aplicación sea altamente compatible con una amplia variedad de dispositivos y sistemas operativos. Esto no solo aumentará su utilidad, sino que también contribuirá a su accesibilidad universal, permitiendo que un público diverso pueda beneficiarse de su uso y contribuir a su desarrollo continuo.

La visión subyacente es transformar esta aplicación en un recurso ampliamente accesible, no solo para su uso actual, sino también para su evolución futura. Esto implica la posibilidad de revivir dispositivos obsoletos que de otra manera no tendrían un propósito en la actualidad y ofrecer la oportunidad de reutilizarlos de manera significativa. Al mismo tiempo, esta iniciativa busca reducir la necesidad de adquirir nuevos dispositivos debido a las demandas de potencia, fomentando así la creación de un programa eficiente en términos de requisitos de hardware.

El proyecto se centra en desarrollar una aplicación flexible y accesible para el control y visualización de datos en máquinas de ensayos biaxiales, con el objetivo de abarcar una amplia gama de dispositivos y sistemas operativos. Esta iniciativa busca no solo mejorar la eficiencia y utilidad de las máquinas existentes, sino también promover la reutilización de equipos obsoletos y reducir la necesidad de inversión en nuevas tecnologías.

1.3. Estructura del documento

Este documento se compone de los siguientes capítulos:

-Introducción: En este capítulo introductorio, se presentan las motivaciones que impulsaron la creación de este trabajo de fin de grado, junto con los objetivos que orientan su desarrollo. Se proporciona una visión general de lo que los lectores pueden esperar encontrar en los siguientes capítulos.

-Máquina de ensayos biaxial: Este capítulo se centra en introducir en detalle la máquina de ensayos biaxial que sirve como contexto principal para el desarrollo del software de control. Se describen sus componentes clave y su función en el proceso de ensayo.

-Estado del arte: En esta sección, se ofrece un análisis del estado actual de la tecnología relacionada con la máquina de ensayos biaxial. Se exploran posibles alternativas y se presenta una crítica fundamentada sobre la situación actual en el campo de estudio.

-Análisis del problema: Este capítulo establece el punto de partida para el desarrollo de la aplicación, identificando y analizando las limitaciones y desafíos clave que deben abordarse durante el proceso. Se delimita el alcance y se detalla el contexto del proyecto.

-Diseño de la solución: En esta sección, se presentan las pautas y estrategias de diseño que guiarán el desarrollo del software de control. Se explican las decisiones de diseño y la arquitectura propuesta para abordar los problemas identificados en el capítulo anterior.

-Desarrollo de la solución: Aquí se describe la estructura y los pasos necesarios para replicar el desarrollo de la aplicación. Se incluyen ejemplos prácticos y explicaciones detalladas de cómo se lleva a cabo el proceso de creación del software.

-Implantación: Este capítulo explica el proceso de poner en funcionamiento un servidor web utilizando los recursos proporcionados por la Universidad Politècnica de València. Se detallan los pasos prácticos para la implementación exitosa de la aplicación.

-Conclusiones: En esta sección, se realiza un análisis integral del trabajo realizado y se examina su relación con los estudios cursados. Se resumen los hallazgos clave y se destacan las lecciones aprendidas a lo largo del proceso.

-Trabajos futuros: En este último capítulo, se exponen las posibles mejoras y desarrollos adicionales que podrían llevarse a cabo en la aplicación en el futuro. Se ofrece una visión prospectiva de las áreas de investigación posteriores y las oportunidades para continuar el trabajo.

2. Máquina de ensayos biaxial

Para facilitar la comprensión de los próximos capítulos y para proporcionar un contexto sólido que aborde tanto las oportunidades como las limitaciones, es esencial presentar la máquina de ensayos biaxial que sirve como caso de estudio en este documento.

Comenzaremos esta presentación destacando a los distintos actores involucrados en la creación y desarrollo de esta máquina de ensayos:

-Antonio Iaccarino Idelson: Antonio Iaccarino Idelson es un distinguido profesor de la Universidad de la Tuscia en Viterbo, Italia [5]. Es un investigador y restaurador con experiencia internacional. Antonio es el autor de la tesis titulada "An open-source biaxial tensile tester with automated pre-tensioning for mechanical studies of canvas paintings" [6]. Su valiosa contribución en la creación de esta máquina de ensayos biaxial aporta una perspectiva única a este proyecto.

-Roger Groves (Presentación): Roger Groves es el director de un grupo de investigación especializado en Testing No Destructivo y Estudio de la Salud Estructural [7] en la Universidad Técnica de Delft, Países Bajos. El Profesor Groves supervisó la tesis de Antonio Iaccarino en Delft.

-Miguel Sánchez López: Miguel Sánchez López es un profesor titular de universidad (PDI) en la Universidad Politécnica de València. Pertenece al Departamento de Informática de Sistemas y Computadores y forma parte del grupo de Redes de Computadores [8]. Además de ser el tutor de este trabajo de fin de grado. Miguel desempeñó un papel fundamental en la colaboración y desarrollo de la aplicación de control, basada en Python, que opera la máquina de ensayos. Su experiencia técnica y conocimientos en informática fortalecen la base tecnológica de este proyecto.

Esta presentación más detallada de los implicados proporciona una visión más completa de sus roles y contribuciones en la creación de la máquina de ensayos biaxial y establece un contexto sólido para comprender su importancia en el desarrollo del software de control.

La máquina de ensayos biaxial, tal como se describe en la tesis de Antonio Iaccarino, es un dispositivo diseñado con el objetivo primordial de lograr una máquina de ensayos económica y de fácil ensamblaje, con la intención de compartir su desarrollo con la mayor cantidad posible de investigadores, sin comprometer la calidad. Esto se debe a que las máquinas de ensayos similares suelen estar al alcance de muy pocos laboratorios debido a sus costos prohibitivos.

En capítulos posteriores de este documento, exploraremos las alternativas disponibles en mayor detalle.

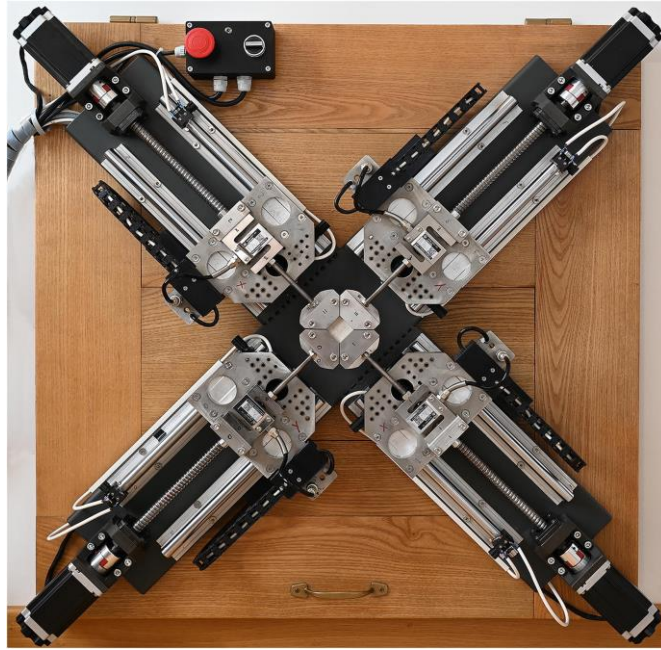
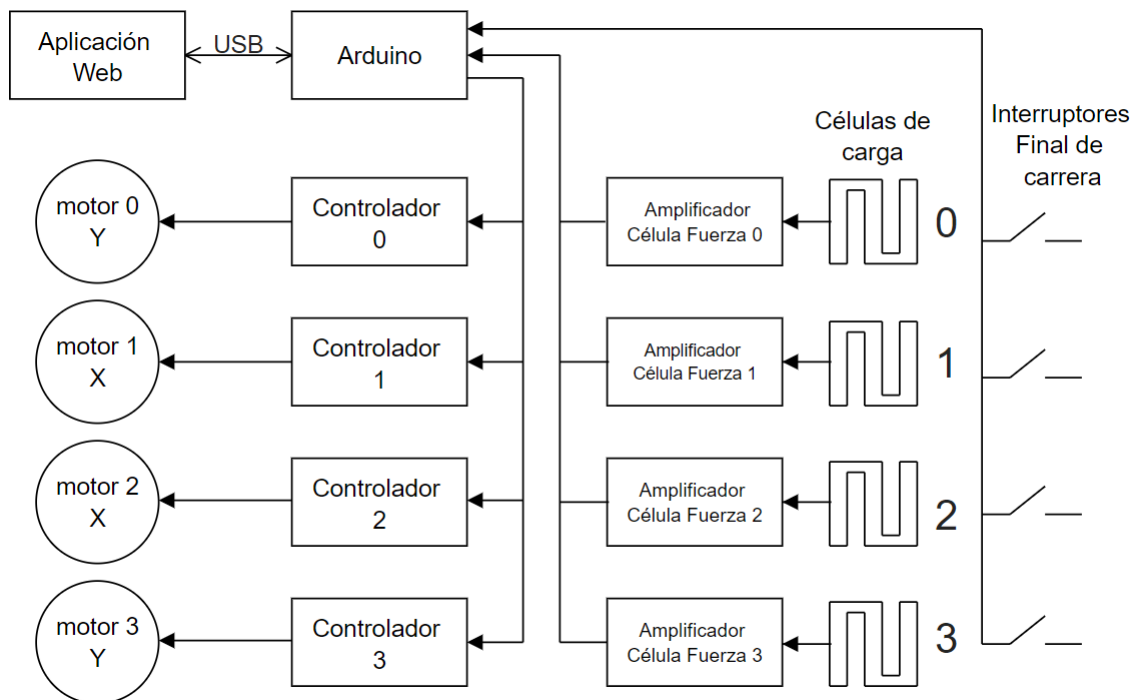


Ilustración 1 : Máquina de ensayos biaxial

La figura anterior representa la máquina, con forma de cruz, que estira una muestra de tela colocada en su centro. Esta máquina está diseñada para aplicar fuerza en los cuatro sentidos de un plano cartesiano formado por los ejes X e Y. Sus componentes están conectados a una placa microcontroladora Arduino, la cual desempeña un papel fundamental en el control preciso del desplazamiento de los ejes y la recopilación de datos de fuerza.



Esquema 1 : Electrónica y flujo de datos

Desarrollo de un programa de control para una máquina de ensayos biaxial

En el esquema anterior, se muestran las partes clave relacionadas con el diseño de la aplicación que estamos desarrollando: los motores y las células de carga que miden la fuerza. En total, hay cuatro de cada tipo, dos en cada eje.

Los motores serán controlados desde la interfaz de manera manual o automáticamente mediante las pruebas definidas. Estos motores girarán con precisión para medir la distancia recorrida durante los ensayos, y los motores enfrentados permitirán la tracción en direcciones opuestas. Las células de carga miden la fuerza de resistencia ejercida por la muestra una vez estirado por los motores, y los datos de estas fuerzas son transmitidos al Arduino y posteriormente a la aplicación de control.

El propósito fundamental de esta máquina, en términos simplificados, es llevar a cabo pruebas de resistencia en lienzos. Estas pruebas se realizan con el fin de estudiar los aspectos mecánicos relacionados con la conservación de pinturas sobre lienzo y, por ende, determinar qué tratamientos pueden ser más beneficiosos para su preservación.

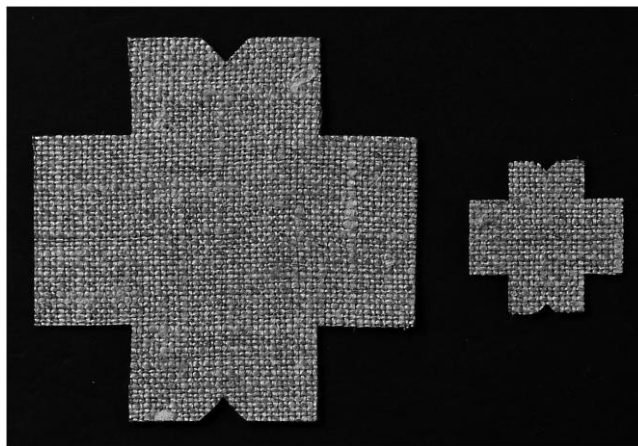


Ilustración 2 : Muestra de lienzo

Las muestras de lienzo tienen una forma similar a la que se muestra en la imagen anterior y son cortadas con precisión mediante láser.

Algunas de las pruebas y controles que pueden llevarse a cabo actualmente incluyen con la aplicación basada en Python son:

- Movimiento de ejes: Esto permite elegir qué ejes se moverán y cuánta distancia se desplazarán.

- Ensayo de tracción a baja tensión: Simula la tensión del lienzo y el desgaste típico al que se somete.

- Ensayo de rendimiento y fallo: Estas pruebas evalúan la resistencia del lienzo en condiciones extremas.

- Ensayo de carga constante: En esta prueba, se mantiene una tensión constante en el lienzo, ajustando la posición del motor según sea necesario. Este tipo de ensayo puede ser de larga duración para evaluar la resistencia a largo plazo del lienzo.

Esta descripción proporciona un contexto más claro sobre la máquina de ensayos biaxial, sus componentes clave y los objetivos de las pruebas que se realizan en ella.

3. Estado del arte

En la era de Internet, es común que los usuarios desarrollen herramientas, tanto de hardware como de software, para abordar problemas o por pura pasión. Sin embargo, este fenómeno puede no ser tan frecuente en todos los campos. En nuestro caso, la creación de un software de control para una máquina de ensayos puede no ser una práctica extendida. Por el contrario, la mayoría de productos similares suelen estar en manos de empresas que desarrollan soluciones para otras empresas, lo que aumenta significativamente el costo del software y lo vuelve menos accesible para un público más amplio.

En favor a estas grandes empresas, es importante reconocer que el desarrollo de software de alta calidad que cumpla con rigurosos requisitos técnicos de precisión y seguridad, y que sea compatible con diversas arquitecturas de hardware, no es una tarea sencilla. A menudo, el software está fuertemente vinculado al hardware en estos casos.

Algunos ejemplos de estas empresas líderes en el mercado incluyen:

-INSTRON [9]: Una empresa estadounidense con más de 75 años de experiencia en la fabricación de máquinas de pruebas de materiales con diversas especificaciones. Ofrecen varias soluciones de software como mejoras para sus máquinas, según su sitio web.

-SHIMADZU [10]: Una empresa japonesa que se dedica a la creación de productos para diversas industrias, como la aeroespacial y la biofarmacéutica. También cuentan con un software específico para máquinas de ensayos.

-GIBITRE [11]: Otra empresa, de origen italiano, que se especializa en máquinas de pruebas de tracción, incluyendo una de un solo eje. Han desarrollado su propio software para estas máquinas.

Todas estas empresas comparten la característica de que, para obtener un software de alta calidad, es necesario adquirir una máquina de la misma empresa. Además, por lo general, solo proporcionan cotizaciones después de llevar a cabo un estudio detallado de las necesidades del usuario, a menudo a través de un formulario.

Este enfoque restringido ha llevado a que solo unos pocos laboratorios tengan acceso a maquinaria y software de este tipo. Sin embargo, proyectos como este tienen el potencial de eliminar la barrera financiera, lo que permitiría que más personas puedan aprovechar estas tecnologías de vanguardia.



4. Análisis del problema

Como punto de partida, consideramos la solución propuesta por el tutor de este trabajo, Miguel Sánchez López.

La solución consiste en una interfaz de control y visualización de datos que gestiona una máquina de 2 ejes utilizada para el ensayo de lienzos y telas. La base de esta solución es una placa de microcontrolador de código abierto, Arduino, que recopila datos y comandos para el control de la máquina.

Esta interfaz la desarrolló en Python con la premisa de crear un programa de control que fuese de fácil instalación y que se pudiese instalar en diferentes entornos o sistemas operativos.

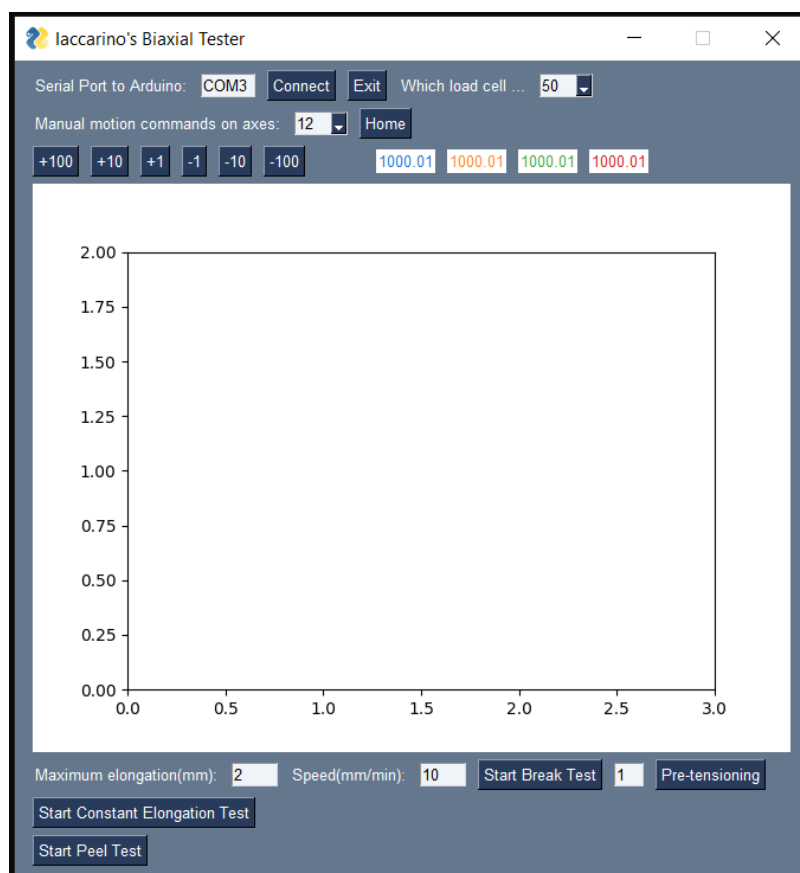


Ilustración 3 : Aplicación de Miguel Sánchez López

Como se puede apreciar en la anterior captura, tareas tan simples como seleccionar qué ejes van a moverse de forma controlada por el usuario (“manual motion commands on axes”), no son bastante intuitivas, ya que requieren saber del 0 al 3 que eje es que motores. Por lo tanto, una mejora importante sería implementar una interfaz más amigable y comprensible para los usuarios, especialmente aquellos que no están familiarizados con los términos técnicos específicos de la industria.

Además, se debe considerar la adaptación de la interfaz al idioma del usuario, dado que se manejan términos altamente especializados en el sector, lo que podría dificultar la comprensión en otros idiomas.

Es importante destacar que el desarrollo de la aplicación se encuentra limitado por la funcionalidad del programa cargado en la placa Arduino. Este programa recibe comandos predefinidos para llevar a cabo diversas acciones y se comunica mediante líneas de texto finalizadas con un retorno de carro.

A continuación, se muestra una captura con los comandos disponibles:

```
Open port COM3
Iaccarino's biaxial tester
by misan
Available serial commands: (lines end with CRLF or LF)
H for help menu
T for sensors tare
S stop motion
D toggle display
P peel test
V123 change speed to 123 mm/min
R resets the load cell (current force=0 Newtons)
F123.34 sets the pulling force to 123.34 Newtons. It will move to keep that force
X123 moves 123 steps (it can be negative too)
L123 sets the max elongation value to 123 mm for any operation
C123 sets the maximum force value to 123 Newtons
K123 sets the pre-tensioning force X-axis value to 123 Newtons
J123 sets the pre-tensioning force Y-axis value to 123 Newtons
W123 pre-tensions the active axes to the set values for 123 seconds
M123 selects the motors to move, -1=all, 0..3:invidual, 12,30, 255=none
@1.1 2.2 3.3 4.4 changes the multipliers of all load cells to these new values
FIN:? prints out current number of steps and the current force on the load cell

SEND: @8816.89,8701.5,8859.1,8965.37,
TIMEOUT SERIAL
```

Ilustración 4 : Mensaje enviado por la aplicación

La captura de pantalla muestra un mensaje enviado por la aplicación a través de un terminal, que es la forma en que Arduino recibe y envía datos a los motores y células de carga. Por lo tanto, será necesario incorporar en la aplicación una forma sencilla de acceder y utilizar un terminal para facilitar la comunicación con la placa Arduino.

Este análisis del problema sienta las bases para el desarrollo de una interfaz de control más intuitiva y versátil que aborde las limitaciones actuales y mejore la experiencia del usuario.

5. Diseño de la solución

La característica principal de esta aplicación de página única radica en su capacidad para ejecutar todo su contenido en una sola página. Este enfoque tiene ventajas significativas al reducir la cantidad de peticiones al servidor para cargar distintos elementos. Desde el momento en que se carga la página web, toda la información necesaria para ejecutar la aplicación está disponible, lo que significa que la aplicación se envía por completo al navegador y no se recarga durante su uso.

Este enfoque tiene un impacto positivo en la experiencia del usuario, ya que mejora los tiempos de respuesta y agiliza la navegación. Además, transforma eficazmente una aplicación que, en su forma tradicional, sería de escritorio, en una aplicación web moderna y eficiente.

Para lograr esta eficiencia y reactividad, hemos aprovechado el framework Vue.js para JavaScript [12]. Vue.js simplifica la creación, diseño y actualización de aplicaciones web, permitiendo un desarrollo más rápido y una funcionalidad más rica.

5.1 Arquitectura

Durante nuestros estudios de grado, hemos aprendido que las aplicaciones cliente-servidor generalmente siguen una arquitectura de software de tres capas o niveles [13]:

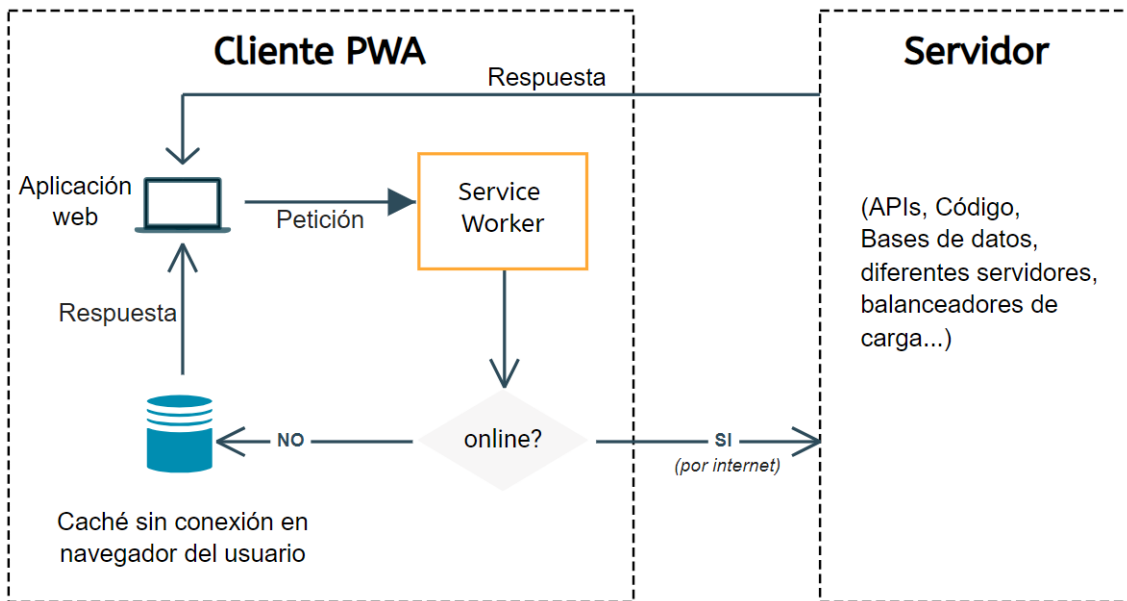
-Capa de persistencia: En esta capa se almacenan los datos necesarios para el funcionamiento de la aplicación y se ejecuta en el servidor.

-Capa de lógica de negocio: Es el núcleo de la aplicación donde se procesan los datos de la capa de persistencia y se suministra la información para la capa de presentación. También se ejecuta en el servidor.

-Capa de presentación: Esta capa comprende la interfaz de usuario y la comunicación de la aplicación para mostrar los datos. Se ejecuta en el navegador web.

Hoy en día, los navegadores son excepcionalmente potentes y cuentan con tecnologías avanzadas, como cachés de datos y procesos en segundo plano, que nos permiten crear aplicaciones que no dependen constantemente de hacer peticiones al servidor.

Como se ilustra en el esquema siguiente, nuestra aplicación no va a funcionar como acabamos de explicar. Todo lo necesario para el funcionamiento de nuestra aplicación se envía desde el servidor al navegador, y este último se encarga del resto.



Esquema 2 : Petición-Respuesta PWA

Esto es posible gracias al concepto de Renderizado en el Lado del Cliente (Client-Side Rendering) [14], que permite que las páginas se construyan en el navegador utilizando JavaScript. Toda la lógica, la obtención de datos y la navegación se gestionan en el cliente y no en el servidor.

En el contexto de nuestra aplicación, todas las tres capas se ejecutan en el navegador:

-Capa de persistencia: Solo necesitamos almacenar datos relacionados con las traducciones. Los demás datos son obtenidos por el Arduino y se almacenan en la caché del navegador.

-Capa de lógica de negocio: La versatilidad de JavaScript nos permite crear métodos y funciones complejas para el procesamiento de los datos recibidos del Arduino.

-Capa de presentación: La utilización de Vue.js potencia la reactividad de la interfaz de usuario, mejorando la percepción y la interacción de los datos por parte del usuario.

Este diseño de tres capas ejecutado en el navegador aprovecha al máximo las capacidades de los navegadores modernos, permitiendo una experiencia fluida y eficiente para el usuario final.

5.2 Diagrama de casos de uso

Para comprender cómo interactúan los diversos actores con nuestra aplicación, es esencial definir los actores involucrados y sus respectivos roles:

-Usuario Nuevo: Se trata de una persona ajena al equipo de desarrollo de la máquina de ensayos que tiene la intención de comenzar a utilizar el sistema completo. Este usuario representa a aquellos que se están familiarizando por primera vez con la aplicación y el hardware.

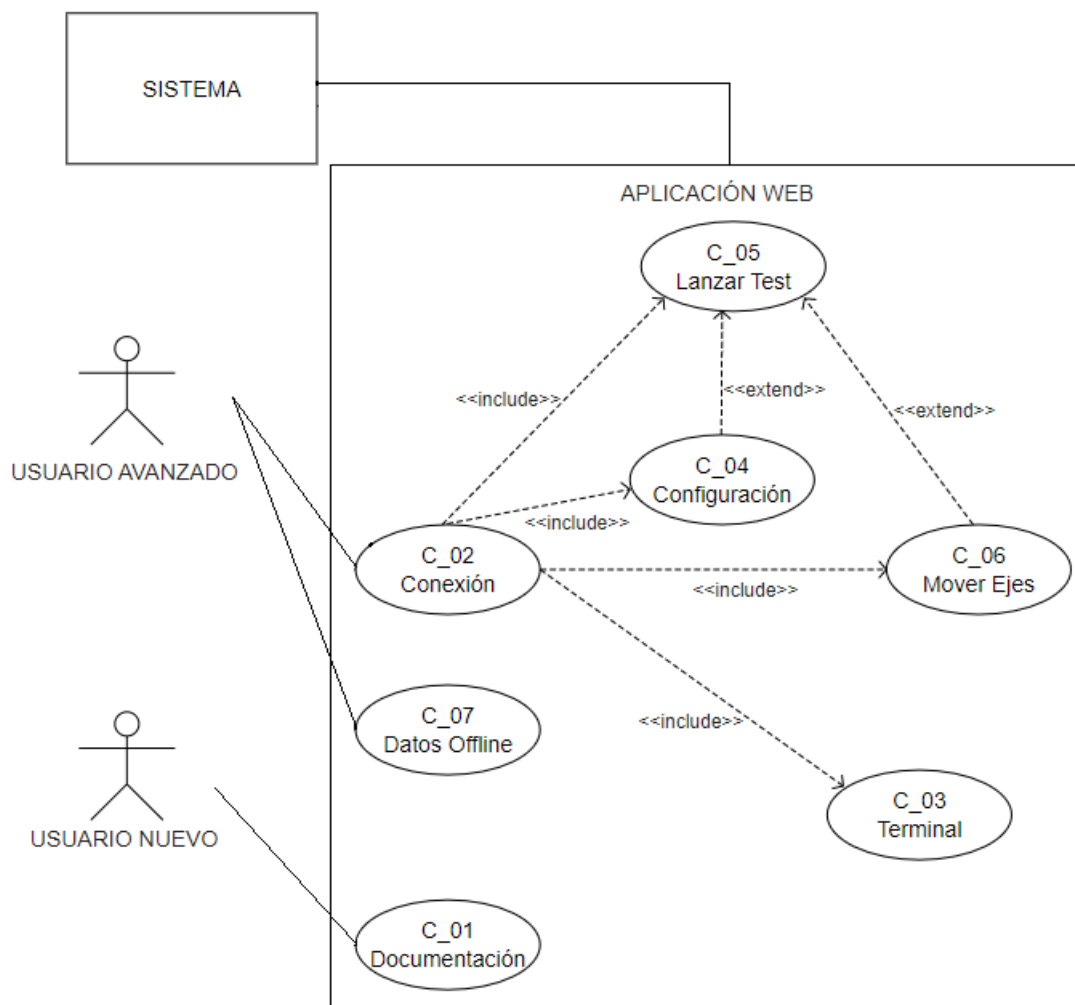
Desarrollo de un programa de control para una máquina de ensayos biaxial

-Usuario Experto: Este actor es una persona que ya ha instalado la máquina de ensayos, ha cargado el firmware en la placa Arduino y ha utilizado el software del tutor en al menos una ocasión. Los usuarios expertos tienen un conocimiento previo y experiencia con la aplicación y la maquinaria.

-Aplicación Web: Hace referencia a la aplicación descrita en este trabajo, que se trata de la interfaz de control y visualización de datos que los usuarios interactúan para gestionar la máquina de ensayos biaxial.

-Sistema: En este contexto, el sistema se refiere a la combinación de la placa de prototipado Arduino y la máquina de ensayos biaxial, que trabajan en conjunto para realizar pruebas y recopilar datos.

A continuación, se presenta un diagrama de casos de uso [15] que representa cómo estos actores interactúan entre sí y con la aplicación:



Esquema 3 : Diagrama de casos de uso

A continuación, definiremos las relaciones entre los casos de uso del gráfico anterior:

-Asociación <<include>>: Esta relación indica que un caso de uso incluye otro. En otras palabras, el caso de uso que incluye a otro requiere que ambos casos de uso se realicen para completar la funcionalidad deseada. Esta relación se utiliza cuando un caso de uso más grande involucra la ejecución de uno o más casos de uso más pequeños. Por ejemplo, en nuestro contexto, podríamos tener un caso de uso "Realizar prueba de tracción" que incluye casos de uso más pequeños como "Configurar parámetros de prueba" y "Registrar resultados de prueba". En este caso, para llevar a cabo una prueba de tracción, se deben realizar ambos subcasos de uso.

-Asociación <<extend>>: Esta relación indica que un caso de uso añade funcionalidad a otro. En otras palabras, el caso de uso agrega otro agrega funcionalidad que es independiente del caso de uso extendido. Esta relación se utiliza cuando hay escenarios en los que ciertas funcionalidades son compatibles entre sí. Por ejemplo, en el contexto de nuestra aplicación, podríamos tener un caso de uso "Mover Ejes" que extiende el caso de uso "Lanzar Test". Esto significa que el usuario puede dejar los ejes en una posición predeterminada antes de lanzar la prueba que moverá estos ejes.

Estas relaciones ayudan a aclarar cómo se interconectan los casos de uso y cómo se comportan en diferentes situaciones. La asociación <<include>> representa dependencias funcionales, mientras que la asociación <<extend>> representea un añadido de funcionalidad entre casos de uso dentro de la aplicación.

5.3 Casos de uso

En esta sección, procederemos a desarrollar en detalle cada uno de los casos de uso representados en el diagrama anterior. Esto permitirá una comprensión más precisa de las funcionalidades generales de la aplicación web, así como de las acciones que pueden llevar a cabo los actores involucrados.

Nombre del caso de uso	C_01 Documentación.
Actores	Usuario Nuevo.
Relación con otros casos de uso	
Explicación	Un usuario que desconozca el sistema puede encontrar toda la información necesaria para replicar una máquina de ensayos como la del estudio y recursos software para poder utilizar la aplicación web.
Flujo básico	1) El usuario se dirige a la página principal de la web. 2) El usuario hace clic en el botón de "más información" que le dirige a una vista de la aplicación que tiene toda la información.
Flujo alternativo	
Precondiciones	Debe tener acceso a internet y conocer la dirección de la página web de la aplicación.
Postcondiciones	

Tabla 1 : Caso de uso documentación

Desarrollo de un programa de control para una máquina de ensayos biaxial

Nombre del caso de uso	C_02 Conexión.
Actores	Usuario Avanzado.
Relación con otros casos de uso	C_03, C_04, C_05 y C_06.
Explicación	Para conectar la máquina de ensayos con la aplicación, se tiene que navegar por la aplicación hasta encontrar el botón que gestiona la conexión por el puerto serie en la parte superior. Existe una etiqueta de estado de conexión.
Flujo básico	<ol style="list-style-type: none"> 1) El usuario se dirige a la página principal de la web. 2) El usuario hace clic en el botón de “comencemos” que le dirige a una vista principal de la aplicación web, la pantalla de ensayos en tiempo real. 3) En la barra superior pulsa el botón “seleccionar puerto”. 4) Selecciona el puerto mediante el dialogo del navegador. 5) Pulsa el botón Conectar.
Flujo alternativo	5) Si no ha resuelto satisfactoriamente la conexión, no cambia el estado de la conexión.
Precondiciones	Debe tener acceso a internet, conocer la dirección de la página web de la aplicación y poseer una máquina de ensayos biaxial conectada a un Arduino con el firmware cargado.
Postcondiciones	

Tabla 2 : Caso de uso conexión

Nombre del caso de uso	C_03 Terminal.
Actores	Usuario Avanzado.
Relación con otros casos de uso	C_02.
Explicación	El Arduino envía a la aplicación web mensajes con los que podemos conocer el estado del funcionamiento de la máquina de ensayos. También se pueden enviar comandos preestablecidos para comprobar funcionalidades.
Flujo básico	1) El usuario se dirige al menú y selecciona el apartado “Terminal”. 2) El usuario tiene establecida la conexión con el Arduino. 3) Se pueden leer los mensajes emitidos por el Arduino y enviar sentencias.
Flujo alternativo	2) No está establecida la conexión con el Arduino. 3) No se muestra ningún mensaje ni se permite introducir ningún comando.
Precondiciones	Debe tener una conexión establecida con el Arduino.
Postcondiciones	Si se ha establecido la conexión correctamente se imprimirá por la pantalla del terminal un mensaje de bienvenida seguido de otro de ayuda describiendo los comandos a utilizar.

Tabla 3 : Caso de uso terminal

Nombre del caso de uso	C_04 Configuración.
Actores	Usuario Avanzado.
Relación con otros casos de uso	C_02 y C_05.
Explicación	Modificación de los parámetros predefinidos antes de lanzar una prueba.
Flujo básico	1) El usuario se encuentra en la vista “Real Time”. 2) El usuario tiene establecida la conexión con el Arduino. 3) Se pueden modificar los parámetros de “velocidad”, “multiplicador de células de carga” y “elongación máxima”.
Flujo alternativo	2) No está establecida la conexión con el Arduino.
Precondiciones	Debe de conocer los parámetros que desea modificar ya que podría poner en riesgo las pruebas.
Postcondiciones	Si se ha establecido la conexión correctamente, la configuración se aplicará cuando se lance una prueba.

Tabla 4 : Caso de uso configuración

Nombre del caso de uso	C_05 Lanzar Test.
Actores	Usuario Avanzado.
Relación con otros casos de uso	C_02, C_04 y C_06.
Explicación	Se puede ordenar al Arduino que lance unas pruebas definidas.
Flujo básico	<ol style="list-style-type: none"> 1) El usuario se encuentra en la vista “Real Time”. 2) El usuario tiene establecida la conexión con el Arduino. 3) En el apartado “Lanzar Pruebas” seleccionar la prueba deseada. 4) Esperar a que termine la prueba y consultar los datos en tiempo real de las fuerzas leídas por las células de carga en la gráfica.
Flujo alternativo	<ol style="list-style-type: none"> 2) No está establecida la conexión con el Arduino.
Precondiciones	Debe de tener configurados los parámetros deseados antes de iniciar una prueba.
Postcondiciones	Si se ha establecido la conexión correctamente, y la prueba ha comenzado de forma exitosa, se empieza a imprimir en el gráfico los datos recibidos por las células de carga.

Tabla 5 : Caso de uso lanzar test

Nombre del caso de uso	C_06 Mover Ejes.
Actores	Usuario Avanzado.
Relación con otros casos de uso	C_02 y C_05.
Explicación	Se puede ordenar al Arduino que mueva los motores en diferentes sentidos. Ya sea de 1 en 1, por parejas en el mismo eje o todos a la vez.
Flujo básico	<ol style="list-style-type: none"> 1) El usuario se encuentra en la vista “Real Time”. 2) El usuario tiene establecida la conexión con el Arduino. 3) El usuario debe marcar que motores desea mover en esquema de la máquina biaxial. 4) El usuario debe utilizar los botones bajo el apartado “Mover motores” para seleccionar la distancia deseada.
Flujo alternativo	<ol style="list-style-type: none"> 2) No está establecida la conexión con el Arduino. 3) Selecciona un patrón de movimiento erróneo y recibe una alerta.
Precondiciones	Debe de tener iniciada una conexión y elegir un patrón de movimiento de motores permitido.
Postcondiciones	Se muestra en cada eje la distancia actual de cada célula de carga.

Tabla 6 : Caso de uso mover ejes

Nombre del caso de uso	C_07 Datos Offline.
Actores	Usuario Avanzado.
Relación con otros casos de uso	
Explicación	Si se poseen archivos que contengan datos recogidos por pruebas y estructurados de forma correcta, se permite la visualización de esos datos sin conexión con el Arduino.
Flujo básico	<ol style="list-style-type: none"> 1) El usuario se encuentra en la vista "Offline". 2) El usuario posee un archivo con datos estructurados de una prueba realizada. 3) El usuario selecciona el archivo con ayuda del asistente. 4) El usuario puede consultar los datos de la prueba en el gráfico.
Flujo alternativo	<ol style="list-style-type: none"> 2) El usuario no dispone de un archivo con los datos bien estructurados. 3) El usuario se dirige al apartado de "información".
Precondiciones	Se precisa seguir un orden en el archivo .csv para poder mostrar los datos correctamente en el gráfico de fuerzas.
Postcondiciones	Se muestra en el gráfico de fuerzas la prueba de forma detallada.

Tabla 7 : Caso de uso datos offline

6. Tecnología utilizada

En esta sección, se explorará en detalle la tecnología utilizada en el desarrollo de la solución para el problema planteado. Esta tecnología es fundamental para comprender cómo se logra la funcionalidad de la aplicación web y su capacidad de instalación en una variedad de dispositivos y sistemas operativos:

-SPA (Single Page Application o Aplicación de Página Única): Una de las tecnologías centrales empleadas en este proyecto es SPA, que significa "Single Page Application" o "Aplicación de Página Única". Esta tecnología se basa en la creación de una página web que contiene todo su contenido en un solo archivo HTML. En este archivo, se incorpora tanto el código HTML, como el JavaScript y el CSS. La finalidad de esta aproximación es proporcionar una experiencia de usuario más fluida y rápida al eliminar la necesidad de realizar constantes peticiones al servidor web.

En lugar de páginas tradicionales, SPA utiliza el concepto de "vistas", que son los diferentes apartados de la página web. Aunque mejora la experiencia de usuario, puede dificultar el posicionamiento en motores de búsqueda (SEO) debido a la falta de múltiples páginas con contenido único. Sin embargo, este enfoque se compensa con otras ventajas en términos de rendimiento y fluidez.

En este proyecto, la SPA está construida con JavaScript, y para lograrlo, se ha utilizado el framework Vue.js, que se presentará en detalle a continuación.

-PWA (Progressive Web App o Aplicación Web Progresiva) [16]: Otra tecnología esencial utilizada es PWA, que significa "Progressive Web App" o "Aplicación Web Progresiva". Una PWA dota a una página web de la capacidad de comportarse casi como una aplicación nativa, permitiendo incluso su instalación en dispositivos. Esto es posible gracias a un componente clave llamado "service worker" que se ejecuta en segundo plano para gestionar solicitudes y almacenamiento de archivos. Además, las PWA ofrecen funcionalidades offline, lo que significa que la aplicación puede seguir siendo útil incluso cuando el usuario no está conectado a Internet.

La tecnología PWA es fundamental en este proyecto, ya que permite que la aplicación esté disponible en una amplia gama de dispositivos sin verse limitada por un sistema operativo específico. Esto no solo mejora la accesibilidad, sino que también puede ahorrar recursos a empresas y usuarios que de otro modo tendrían que desarrollar aplicaciones móviles específicas.

6.1. Vue.js

Para la elaboración de la solución he decidido utilizar el entorno de trabajo de JavaScript: Vue.js. Esta herramienta nos permite crear interfaces web de manera sencilla, tras la compilación genera documentos HTML, CSS y JavaScript para darle a la web toda la funcionalidad necesaria.

Vue trabaja con plantillas, cada una es un archivo independiente que incluye su código HTML, CSS y JavaScript. Estas plantillas permiten vincular de forma declarativa el DOM. Nos permite la creación de páginas reactivas, evitando tener que recargar la página para que cambien los datos que contiene.

Desarrollo de un programa de control para una máquina de ensayos biaxial

Estas plantillas permiten vincular de forma declarativa el DOM, lo que facilita la creación de páginas reactivas sin necesidad de recargar la página para actualizar datos.

Las plantillas pueden separarse en componentes y vistas:

-Los componentes son fragmentos de páginas que se pueden reutilizar, en nuestro caso la barra de navegación o el pie de página.

-Las vistas son componentes y agrupaciones, que se utilizan para presentar datos y pueden ser accesibles por URL gracias a Vue Router.

Para crear la base de la aplicación, se ha utilizado Vue-CLI, que es un sistema completo que entrega las herramientas necesarias para la creación, desarrollo y despliegue de aplicaciones web. Como se puede ver en la siguiente captura, a través del terminal seguimos un breve proceso en el cual elegimos diferentes funcionalidades añadidas según nos interese.

```
D:\MSI\Documents\UNIVERSIDAD\CODEcli>vue create app-biaxial

Vue CLI v5.0.8
? Please pick a preset: Manually select features
? Check the features needed for your project: PWA, Router, Vuex
? Choose a version of Vue.js that you want to start the project with 3.x
? Use history mode for router? (Requires proper server setup for index fallback in production) No
? Where do you prefer placing config for Babel, ESLint, etc.? In dedicated config files
? Save this as a preset for future projects? No

Vue CLI v5.0.8
✔ Creating project in D:\MSI\Documents\UNIVERSIDAD\CODEcli\app-biaxial.
✔ Initializing git repository...
✔ Installing CLI plugins. This might take a while...
```

Ilustración 5 : Configuración Vue create

Entre las configuraciones seleccionadas para este proyecto se encuentran:

-Soporte PWA: Gracias a este plugin, la aplicación web se puede instalar en varios navegadores, como Chrome, Edge, Firefox o Safari.

-Vue Router: Este plugin facilita la gestión de enlaces y redirecciones, simplificando el comportamiento de la aplicación web de página única.

-Vuex (actualmente Pinia): Otra herramienta de Vue que permite compartir datos entre diferentes componentes y vistas de la aplicación.

Además de los plugins mencionados, se han incorporado las siguientes librerías:

-Bootstrap 5: Esta librería proporciona una colección de herramientas que simplifican la aplicación de estilos CSS a elementos HTML, facilitando la decoración del contenido web.

-Chart.js: Esta potente herramienta permite la generación de tablas y gráficos estadísticos, especialmente útiles para representar datos de manera reactiva, es decir, que cambian cuando los datos cambian.

-Node SerialPort: Esta librería de JavaScript es esencial para la comunicación con la placa de Arduino a través del puerto serie.

-Vue-i18n: Esta librería permite la implementación de un sistema de traducciones basado en etiquetas, lo que facilita la internacionalización de la aplicación.

Vue.js ofrece un entorno de desarrollo con un servidor local, que permite modificar el código mientras se tiene una vista previa en el navegador. Cada vez que se guarda un archivo de código, la página se recarga automáticamente, lo que agiliza el proceso de creación de páginas web dinámicas. Esta funcionalidad se puede ejecutar desde una terminal en el directorio principal de la aplicación mediante el comando "npm run serve".

6.2. Github

Para cumplir con uno de los objetivos fundamentales de este proyecto, que es hacer que la aplicación web sea accesible para todo el mundo y fomentar su desarrollo, se utiliza GitHub.

GitHub es una plataforma que permite compartir y gestionar proyectos de software de forma colaborativa. Además, es esencial para recuperar el repositorio completo durante el despliegue de la aplicación, un proceso que se detallará en un capítulo posterior.

6.3. Visual Studio Code

Visual Studio Code de Microsoft es la herramienta principal utilizada para la creación y edición de los diferentes directorios y archivos que contienen el código del proyecto. Este editor de código fuente es muy poderoso y cuenta con un repositorio extenso de extensiones que facilitan la implementación de código de manera ágil y limpia. Algunas de las extensiones utilizadas en este proyecto incluyen:

-Git: Utilizado para subir los cambios a GitHub de manera sencilla y mantener un historial de versiones.

-Volar: Una suite de herramientas para el desarrollo de plantillas de Vue.

-ESLint: Esta extensión analiza el código para evitar errores y sugiere mejoras para lograr un rendimiento óptimo.

Visual Studio Code es elegido por su gran comunidad de usuarios y su gratuidad, lo que lo convierte en una herramienta atractiva para el desarrollo de proyectos web.

Este editor puede que no sea tan potente como por ejemplo PHPStorm, pero su gran atractivo principal es que tiene una gran comunidad de usuario y que es gratuito.

6.4. PLESK

Para probar y compartir la aplicación web descrita en este documento, se ha utilizado Plesk, un software de alojamiento de páginas y aplicaciones web con un panel de control sencillo.

La Universidad Politécnica de Valencia proporciona acceso gratuito a un servidor web a sus estudiantes y docentes.

Desarrollo de un programa de control para una máquina de ensayos biaxial

Ha sido una tecnología muy necesaria, para poder probar y compartir la aplicación web detallada en este documento. En un capítulo posterior se detallará su uso y como realizar el despliegue de una aplicación Node.Js.

7. Desarrollo de la solución

A continuación, comenzaremos describiendo la creación de nuestra aplicación Vue base para el proyecto siguiendo una serie de pasos fundamentales. Seguidamente definiremos los otros desarrollos importantes para que nuestra aplicación adquiera el comportamiento y funcionalidad deseada.

APLICACIÓN WEB VUE:

Dado que se trata de una aplicación Node.js, es esencial establecer un entorno adecuado antes de comenzar el desarrollo. A continuación, se detallan los pasos clave para configurar el entorno de desarrollo y crear la estructura inicial de la aplicación Vue 3:

-Paso 1: Instalación de NVM y Node.js: Primero, es recomendable instalar NVM (Node Version Manager) para gestionar diferentes versiones de Node.js en su sistema. Esto proporciona flexibilidad al trabajar en proyectos con diferentes requisitos de versión de Node.js. Luego, instale Node.js y NPM utilizando NVM para asegurarse de tener las últimas versiones disponibles.

-Paso 2: Creación de la Aplicación Vue: Utilice Vue-CLI, una herramienta oficial de Vue.js, para crear la estructura inicial de la aplicación. A través del asistente de Vue-CLI, puede seleccionar las opciones y configuraciones deseadas para su proyecto. Estas opciones incluyen configuraciones para Vue 3 y otras dependencias.

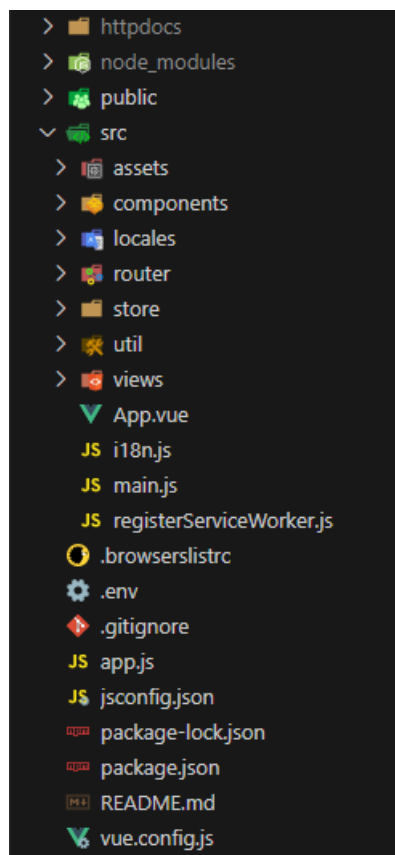


Ilustración 6 : Directorios de la aplicación

Desarrollo de un programa de control para una máquina de ensayos biaxial

-Paso 3: Estructura del Proyecto: La estructura del proyecto final debe ser organizada y comprensible. El siguiente esquema muestra los principales directorios y archivos que encontrará en su proyecto:

-/assets: Este directorio es utilizado para almacenar recursos estáticos como imágenes, fuentes, y otros archivos que se utilizan en su aplicación.

-/components: Aquí se encuentran los componentes reutilizables de la aplicación, como la barra de navegación (NavBar.vue) y el pie de página (Footer.vue).

-/views: Este directorio contiene las diferentes vistas únicas de la aplicación. Cada vista representa una página de la aplicación y es única en su diseño y contenido.

-App.vue: Este archivo es el componente principal de la aplicación y actúa como la raíz definida por Vue.js. En él, se configuran y muestran las diferentes vistas de la aplicación.

-main.js: Este archivo es responsable de inicializar la aplicación Vue y configurar los complementos principales. Aquí se define el punto de entrada de la aplicación.

Los archivos de Vue tienen la siguiente estructura:

```
<template>
</template>
<script>
</script>
<style>
</style>
```

Ilustración 7 : Plantilla de un archivo .vue

-Existe un apartado “template” donde escribiremos con código HTML, la estructura del documento Web.

- “Scripts”, escrito en JavaScript, encontramos un área para nuestros métodos y propiedades que necesitamos para modificar el modelo de objeto de documento (DOM).

-Y finalmente “style”, que es una zona de código donde escribiremos las sentencias CSS para la modificación del estilo.

El componente principal es “App.vue”, que es la raíz de la aplicación definida por Vue.js, donde mostraremos las diferentes vistas, que definiré a continuación.

Componentes Reutilizables:

Dentro del directorio /components, encontramos dos componentes reutilizables clave:

NavBar.vue: Este componente representa la barra superior de navegación de la aplicación, que incluye elementos como el logo de la aplicación, un indicador de estado de conexión,

botones para modificar la conexión, traducción de la página y un menú desplegable para navegar por la aplicación.



Ilustración 8 : Barra de navegación

Footer.vue: Este componente simple muestra información sobre el creador de la aplicación y proporciona detalles adicionales sobre su desarrollo.

Vistas Únicas:

Las vistas en el directorio /views representan las diferentes páginas de la aplicación web. Aquí se describen algunas de las vistas clave:

- HomeView.vue: Esta vista de bienvenida orienta a los usuarios, tanto nuevos como habituales, sobre el funcionamiento básico de la aplicación y cómo comenzar a utilizarla.

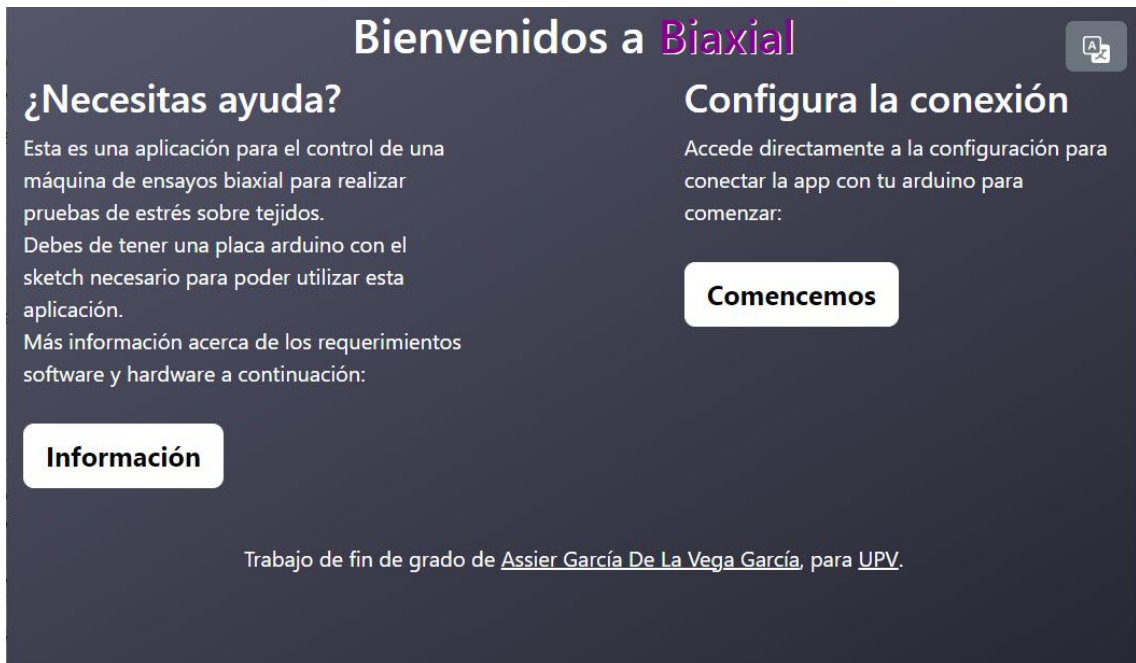


Ilustración 9 : Pantalla de bienvenida

- RealTimeView.vue: En esta vista, los usuarios pueden interactuar con un gráfico de fuerzas en tiempo real. Ofrece controles para el movimiento de los ejes, configuración de parámetros y la posibilidad de ejecutar pruebas en tiempo real.

Desarrollo de un programa de control para una máquina de ensayos biaxial



Ilustración 10 : Pantalla de control y captura

- OfflineView.vue: Esta vista permite a los usuarios revisar datos de pruebas guardados localmente en forma de gráficos y estadísticas.

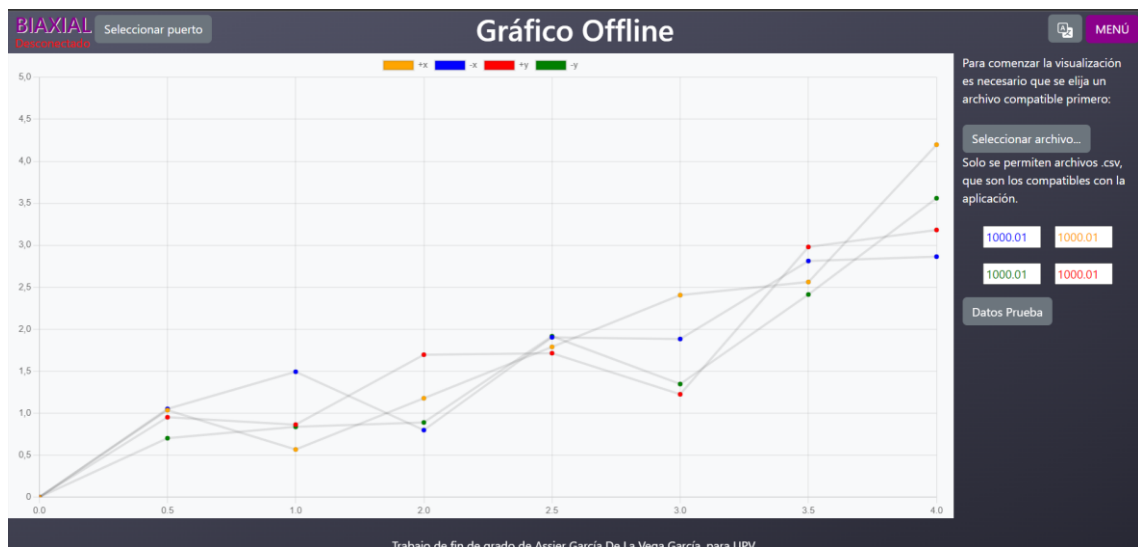


Ilustración 11 : Pantalla de estudio de datos fuera de línea

- ConsoleView.vue: Una vista de consola que muestra los mensajes enviados por la placa de prototipado Arduino. Aquí, los usuarios pueden leer los mensajes, mientras que los comandos se envían desde otras páginas de la aplicación.

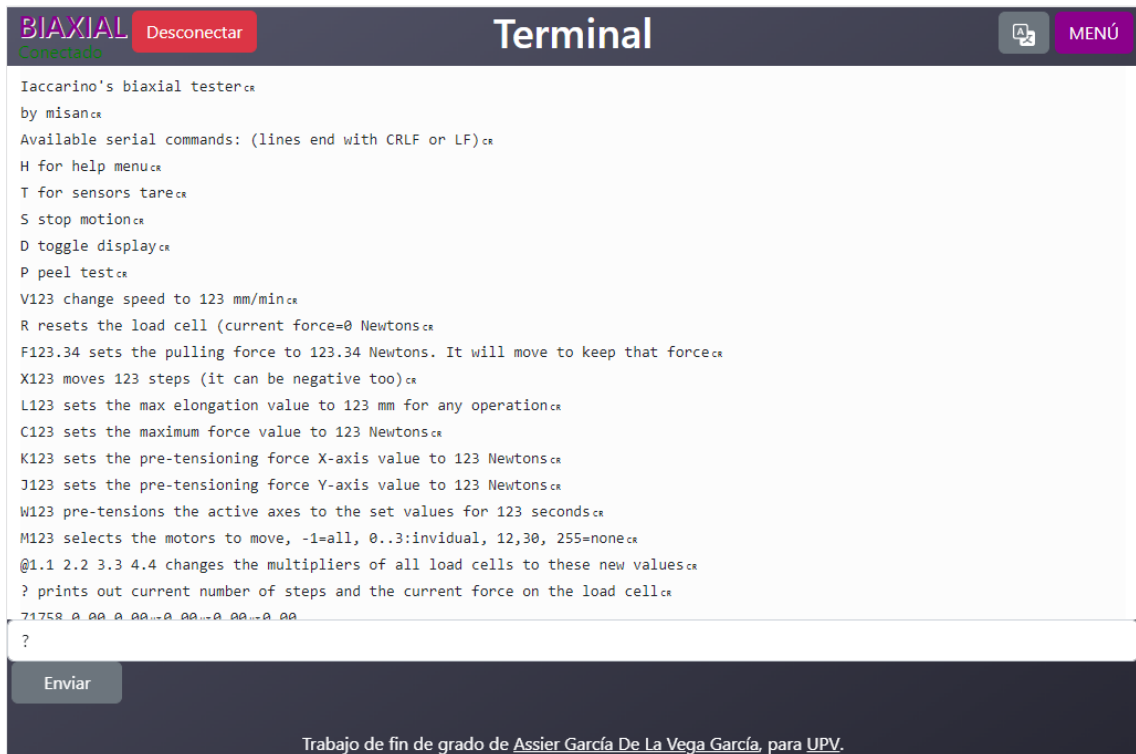


Ilustración 12 : Pantalla terminal

- AboutView.vue: Una página informativa que proporciona una descripción detallada del funcionamiento de la aplicación. También ofrece enlaces para descargar el firmware de Arduino y un documento científico relacionado con el desarrollo de la máquina de ensayos biaxial.

La estructura de la aplicación Vue se ha diseñado de manera organizada y modular para facilitar el desarrollo y la comprensión. Los componentes reutilizables permiten un enfoque eficiente en la creación de elementos comunes, mientras que las vistas únicas ofrecen experiencias específicas para cada página de la aplicación. Este enfoque garantiza que la aplicación sea mantenible y escalable a medida que evoluciona con nuevas características y mejoras.

ROUTER:

El enrutador (Router), es una pieza de código que permite que podamos navegar por las diferentes páginas de la web y gestionar los historiales. Todo esto sin complicar las direcciones url.

Tenemos un directorio “router” en el que encontramos un archivo JavaScript “index” donde se define cada la ruta de cada vista o página que se va a querer acceder.

```
const routes = [
{
  path: "/console",
  name: "Console",
  component: ConsoleView,
},
],
```



La constante `routes` es un arreglo que define las rutas de la aplicación web. Cada ruta está representada como un objeto con tres propiedades principales:

-`path`: La URL que activará esta ruta. Por ejemplo, `/` será la página del terminal, `/console` será la página del terminal, etc.

-`name`: Un nombre único para identificar esta ruta. Esto es útil para navegar a estas rutas programáticamente.

-`component`: El componente de vista que se asociará con esta ruta. Cada componente de vista representa una página o una sección de la aplicación.

Luego cuando queremos desarrollar un hipervínculo u otra manera de navegar entre las páginas, solo tenemos que utilizar la etiqueta de html "Routerlink" de la siguiente forma:

```
<button class="btn btn-light">  
  <RouterLink class="link" to="/Console"> Terminal </RouterLink>  
</button>
```

En este caso, utilizado en el menú desplegable de la barra de navegación, se le asigna a un botón un hipervínculo con la funcionalidad de redirigir a la vista Terminal con la ruta `"/Console"`.

Esta clase de enrutador Vue.js se encarga de definir y gestionar las rutas de la aplicación web. Cada ruta está asociada a una vista específica, lo que permite que la aplicación muestre las páginas correctas cuando el usuario navega a través de ellas. Esto es fundamental para la navegación y la organización de una aplicación web Vue.js.

STORE:

La parte de código que puede ser la más importante del trabajo, se trata del "store". Este código se integra en un "store" definido con la biblioteca Pinia, que permite que este código se ejecute constantemente en la aplicación web y gestione la comunicación serie. Vamos a desglosar las partes clave del código y su funcionalidad.

Para la conexión de nuestra aplicación web con la placa Arduino, vamos a utilizar la interfaz de comunicación bidireccional puerto serie (también conocido como puerto serie RS-232 o UART). Nos vamos a ayudar de la biblioteca WEB SERIAL API [17].

Vamos a desglosar las partes clave del código y su funcionalidad.

Definición del "store" y Variables Iniciales:

El código comienza importando las bibliotecas y utilidades necesarias, como `defineStore`, `ref`, `hex`, `getUsbInfo`, `encodeWithHtml`, y `decode`. Estas utilidades ayudan en diversas tareas, como el manejo de la información USB, codificación y decodificación de datos, y otras operaciones esenciales.

Luego, se definen constantes como `vid_pid`, `encoder`, y `decoder` para realizar operaciones iniciales con los datos del puerto serie y codificación/decodificación de datos.

A continuación, se define un "store" llamado useConnectionStore, que contendrá todos los datos y funciones relacionados con la comunicación serie.

Se inicializan variables de estado, como id, vendor, product, port, y otros que se utilizan para rastrear la información de la conexión y el estado de la comunicación.

Acciones Asíncronas:

El "store" define varias acciones asíncronas que permiten interactuar con la comunicación serie y realizar tareas específicas, como seleccionar un puerto, inicializar la conexión, conectarse al dispositivo, monitorizar la comunicación, escribir datos, cerrar la conexión, obtener datos, detener la recepción de datos, enviar comandos y más.

-La acción selectPort permite seleccionar un puerto de comunicación USB. Esto es especialmente útil si se tienen múltiples dispositivos conectados.

-La acción init inicializa la conexión con un dispositivo específico identificado por su ID de vendedor (VID) y su ID de producto (PID).

-La acción connect abre la conexión con el dispositivo, configurando los parámetros de la comunicación, como velocidad de baudios y otros.

```
async connect() {
  if (!this.port) return
  try {
    await this.port.open(this.options)
    this.open = !!this.port?.readable
    this.monitor()
  }
  catch (e) {
    window.alert(e.message)
  }
},
```

-La acción monitor se ejecuta continuamente para leer los datos del puerto serie y almacenarlos en el "store". Esta función está diseñada para ejecutarse constantemente mientras la conexión esté abierta.

```
async monitor() {
  while (this.open && this.port?.readable) {
    this.open = true
    const reader = this.port.readable.getReader()
    this._reader = reader
    try {
      let aux = '';
      while (this.open) {
        const { value, done } = await reader.read()
        if (done) {
          this.open = false
          break;
        }
      }
    }
  }
}
```



```
    }
    const decoded = decoder.decode(value)
    this.messages.push(decoded)
    aux += decoded;
    if(aux.includes(this.separator)){
        this.messagesV2.push(aux);
        aux = '';
    }
}
} catch (error) {
    console.error('reading error', error)
} finally {
    reader.releaseLock()
}
},
```

-Las acciones write y close permiten enviar datos al dispositivo y cerrar la conexión, respectivamente.

```
async write(data) {
    if (this.port?.writable) {
        const writer = this.port.writable.getWriter()
        await writer.write(encoder.encode(data))
        writer.releaseLock()
    }
},
```

-Las acciones getData y stopData se utilizan para iniciar y detener la recepción continua de datos desde el dispositivo.

-Otras acciones, como sendCommand, getLastMessage, y changeCanReadData, se utilizan para enviar comandos específicos al dispositivo, obtener el último mensaje recibido y controlar la recepción de datos.

```
async sendCommand(command, params){
    let cmd = '';
    if(typeof params !== 'undefined'){
        cmd = this.prepend + command + params + this.append;
    }
    else{
        cmd = this.prepend + command + this.append;
    }
    this.cargandoDatos = true;
    await this.write(encode(cmd));
    this.cargandoDatos = false;
},
```

Uso de Event Listeners:

El código también utiliza Event Listeners para detectar cuándo un dispositivo USB se conecta o desconecta físicamente. Estos eventos se manejan en las funciones onconnect y ondisconnect.

Exportación del Store:

Finalmente, el "store" se exporta para que pueda ser utilizado en otros componentes de la aplicación web.

En resumen, este código crea un "store" que administra la comunicación serie con un dispositivo externo, como una placa Arduino. Proporciona funciones para seleccionar el puerto, iniciar la conexión, enviar y recibir datos, y monitorear continuamente la comunicación. Esto permite que la aplicación web interactúe con el dispositivo conectado y realice operaciones específicas según los comandos enviados y los datos recibidos.

TRADUCCIONES:

Para poder llevar a cabo la funcionalidad de traducción de la aplicación sin necesitar una conexión a internet. El siguiente código se encarga de configurar y cargar las traducciones (locales) para una aplicación Vue.js utilizando la biblioteca vue-i18n. Veamos una descripción detallada de este código:

```
import { createI18n } from 'vue-i18n'

function loadLocaleMessages() {
  const locales = require.context('./locales', true, /[A-Za-z0-9-
_,\s]+\\.json$/i)
  const messages = {}
  locales.keys().forEach(key => {
    const matched = key.match(/([A-Za-z0-9-_]+)\./i)
    if (matched && matched.length > 1) {
      const locale = matched[1]
      messages[locale] = locales(key).default
    }
  })
  return messages
}

export default createI18n({
  legacy: false,
  globalInjection: true,
  locale: process.env.VUE_APP_I18N_LOCALE || 'en',
  fallbackLocale: process.env.VUE_APP_I18N_FALLBACK_LOCALE || 'en',
  messages: loadLocaleMessages()
})
```



1. Importación de createI18n:

El código comienza importando la función `createI18n` de la biblioteca `vue-i18n`. Esta función se utiliza para crear una instancia de `Vue I18n`, que se utilizará para gestionar las traducciones en la aplicación.

2. Función `loadLocaleMessages`:

La función `loadLocaleMessages` se utiliza para cargar las traducciones desde archivos JSON ubicados en la carpeta `locales`. Estos archivos deben seguir una convención de nomenclatura específica, como `[idioma].json`, donde `[idioma]` es el código de idioma, por ejemplo, `en.json` para inglés.

`Require.context` se utiliza para cargar dinámicamente todos los archivos JSON de traducción en la carpeta `locales`.

Luego, se itera sobre cada archivo JSON cargado y se extraen las traducciones correspondientes. Estas traducciones se almacenan en un objeto `messages`, donde cada clave es el código de idioma y el valor es un objeto que contiene las traducciones para ese idioma.

El objeto `messages` se devuelve al final de la función, y se utilizará como fuente de traducciones para la aplicación.

3. Creación de la instancia de `Vue I18n`:

Se exporta una instancia de `Vue I18n` utilizando `createI18n`. Aquí se configura `Vue I18n` con las siguientes opciones:

-`legacy`: Se establece en `false`. Esto significa que se utiliza la nueva API de `Vue I18n` en lugar de la API heredada. La nueva API proporciona una sintaxis más moderna y mejorada para gestionar las traducciones.

-`globalInjection`: Se establece en `true`. Esto permite que la instancia de `Vue I18n` se inyecte globalmente en la aplicación, lo que facilita el acceso a las traducciones en cualquier componente sin necesidad de importaciones adicionales.

-`locale`: Se establece en `process.env.VUE_APP_I18N_LOCALE || 'en'`. Esta opción define el idioma predeterminado de la aplicación. Si no se encuentra ninguna configuración en el entorno (variable `VUE_APP_I18N_LOCALE`), se utiliza el idioma inglés ('en') como valor predeterminado.

-`fallbackLocale`: Se establece en `process.env.VUE_APP_I18N_FALLBACK_LOCALE || 'en'`. Esta opción define el idioma de respaldo que se utilizará si no se encuentra una traducción para el idioma actual. Al igual que con `locale`, el valor predeterminado es inglés ('en') si no se especifica en el entorno.

-`messages`: Se establece en el resultado de la función `loadLocaleMessages()`. Esto carga las traducciones previamente recopiladas utilizando la función `loadLocaleMessages`.

Después, utilizando la función `$t(etiqueta)` con la etiqueta correspondiente al texto que se quiere mostrar en los diferentes idiomas. El renderizado de los textos no necesita que se recarguen las páginas.

```
<h4>{{ $t(36) }}</h4>
```

En resumen, este código configura y carga las traducciones para una aplicación Vue.js utilizando la biblioteca `vue-i18n`. Permite que la aplicación cambie dinámicamente entre diferentes idiomas según las traducciones proporcionadas en archivos JSON en la carpeta locales. La configuración también se encarga de manejar idiomas predeterminados y de respaldo.

8. Implantación

Para lograr el despliegue exitoso de nuestra aplicación web en un servidor de la Universidad Politécnica de València (UPV), seguimos una serie de pasos cruciales. Aquí se detallan en profundidad:

1. Acceso al Servicio de Publicación Web de la UPV:

Primero, solicitamos el alta en el Servicio de Publicación Web de la UPV, lo que nos otorga acceso a la infraestructura necesaria para desplegar nuestra aplicación web.

2. Acceso al Panel de Control de Plesk:

En aproximadamente una hora después de solicitar el alta, se nos proporciona acceso al panel de control de Plesk. Este panel permite la gestión y configuración del servidor web.

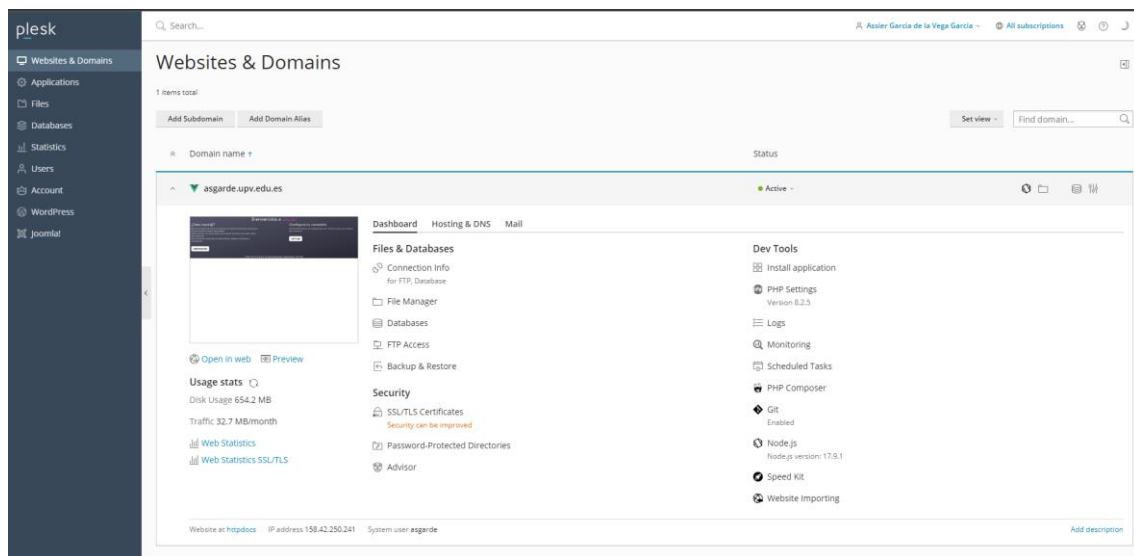


Ilustración 13 : Dashboard de Plesk

3. Preparación Inicial:

Dado que nuestra aplicación no se basa en WordPress, es esencial desinstalar el plugin de WordPress que viene preinstalado en los servidores Plesk. Plesk no recomienda tener múltiples aplicaciones de diferentes tecnologías en un mismo servidor para garantizar un funcionamiento sin problemas.

Para mantener una instalación limpia, eliminamos el contenido del directorio "httpdocs". Este directorio será donde alojaremos nuestra aplicación web una vez compilada.

4. Instalación de Herramientas Requeridas:

Dado que hemos optado por Vue.js, un framework que requiere Node.js para funcionar, instalamos la herramienta Devtool Node.js desde el panel principal de control de Plesk.

Además, para compartir nuestros archivos de desarrollo con otros usuarios de Internet, instalamos el plugin Git, lo que nos permite recuperar una copia de todo el repositorio del proyecto desde GitHub.

5. Configuración de GitHub en Plesk:

Configuramos nuestro repositorio de GitHub en Plesk, proporcionando los datos necesarios de autenticación y las rutas de acceso. Una vez configurado, realizamos una operación "pull" para obtener los archivos de la aplicación y, a continuación, un "deploy". Es importante agregar estos archivos en la raíz de nuestro servidor Plesk.

app-biaxial settings

Remote repository

Properties

Repository name *

Specify a name that is unique within a domain.

Repository URL

To change the URL, create a new repository

If access to the repository is secured by HTTP Basic credentials, specify them below

Username

Password

Webhook URL

Specify this URL in the repository settings of the remote service.

Repository branch *

Deployment settings

Deployment mode *

 Automatic Manual Disabled

Files will be deployed to the production site as soon as they are available in the Plesk repository.

Server path *

Directory on the server where files will be deployed.

Ilustración 14 : Configuración de Github en Plesk

6. Configuración de la Aplicación Node.js:

A pesar de que no podemos modificar el "Document Root" debido a la falta de permisos, durante el desarrollo de la aplicación preparamos la ruta de despliegue de los archivos compilados teniendo en cuenta este inconveniente.

También preparamos un archivo llamado "app.js" para configurar el comportamiento de la aplicación una vez desplegada.

7. Instalación de Dependencias:

Dado que la aplicación web requiere componentes externos a Node.js, como Vue y sus plugins, nos dirigimos a nuestra aplicación Node.js y ejecutamos "NPM install" para instalar las dependencias necesarias. La primera vez que se ejecuta, esto puede llevar algo de tiempo, pero no tardará tanto en ejecuciones posteriores, a menos que se agreguen nuevas bibliotecas.

8. Compilación de la Aplicación:

Para poblar la carpeta "httpdocs" y desplegar nuestra aplicación, ejecutamos el script "build". Vue necesita compilar los archivos de desarrollo para crear la página web final. Es importante recordar que cualquier modificación en la página web requerirá una nueva compilación.

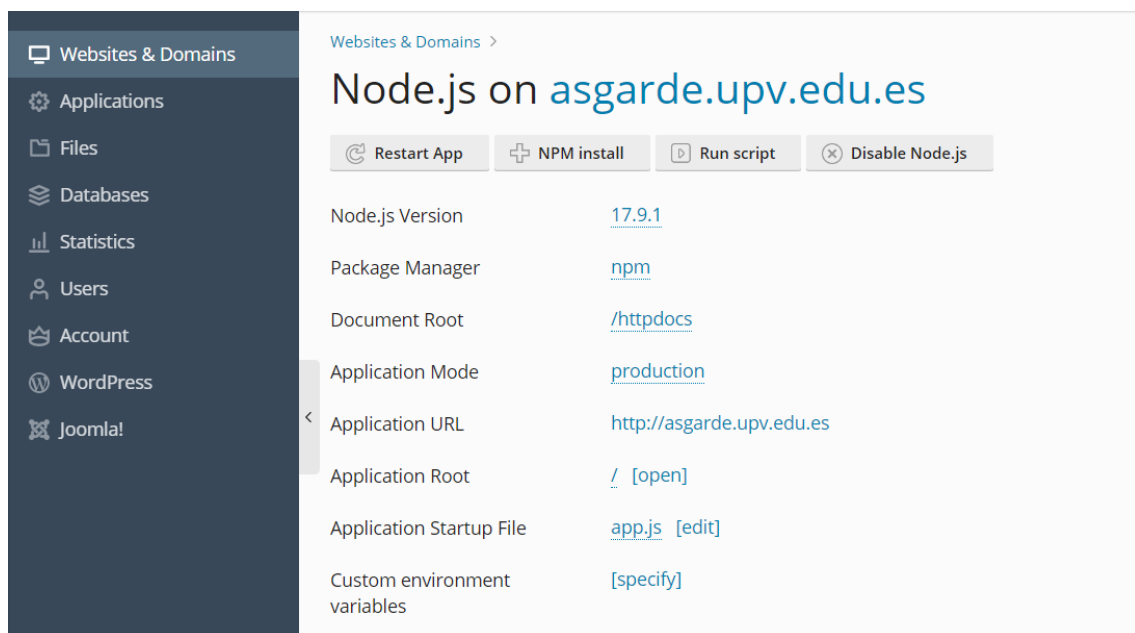


Ilustración 15 : Configuración de una aplicación Node.js en Plesk

De esta manera, hemos logrado desplegar con éxito nuestra aplicación web en un servidor proporcionado por la UPV, y ahora se puede acceder a ella desde <https://asgarde.upv.edu.es/>. Este proceso asegura que nuestra aplicación esté disponible para su uso en una amplia variedad de dispositivos y sistemas operativos, lo que es esencial para su accesibilidad y utilidad.

9. Conclusiones

La construcción de una aplicación web, como esta, puede no parecer tan atractiva a primera vista, ya que no se suelen ver muchos ejemplos profesionales de este tipo de tecnologías en el contexto de máquinas de ensayo biaxiales. Normalmente, el software diseñado para controlar estas máquinas suele ser costoso, orientado a sistemas cerrados y altamente especializado.

Sin embargo, esta aproximación busca cambiar ese paradigma al crear una pieza de software que sea fácil de desarrollar y accesible para una audiencia más amplia. Es cierto que JavaScript y Vue pueden parecer intimidantes al principio, pero con un poco de conocimiento y práctica, es sorprendente lo que se puede lograr. Además, hoy en día, la mayoría de las personas que necesitan utilizar una máquina de ensayos similar tienen acceso a Internet y pueden obtener una versión instalable del software con solo unos pocos pasos.

Creo firmemente que, si tuviera la oportunidad de conocer mejor el contexto de uso de la aplicación y trabajar de cerca con usuarios reales para comprender cómo utilizan las diferentes pantallas y características, podríamos crear una versión aún mejor y más adaptada a sus necesidades específicas.

Este proyecto ha sido una experiencia de aprendizaje invaluable. No solo me ha proporcionado conocimientos profundos sobre las tecnologías JavaScript y HTML, sino que también me ha enseñado la importancia de la perseverancia en proyectos de esta envergadura. El desarrollo de software es un proceso que requiere tiempo, dedicación constante y un aprendizaje continuo, tanto en términos de conocimientos previos como futuros.

Ahora, considero que he mejorado en la organización de proyectos. He aprendido a establecer metas a corto plazo que me permiten obtener resultados constantes y evitar que los problemas se acumulen con el tiempo. Esta lección sobre gestión del tiempo y organización es invaluable en cualquier campo.

Por último, quiero destacar la importancia de la cultura del código abierto. Creo que esta filosofía debe perdurar y crecer, ya que puede contribuir significativamente a resolver problemas, abordar vulnerabilidades y superar las dificultades asociadas con el aprendizaje y el uso de software. El código abierto fomenta la colaboración, la transparencia y la innovación, y es un recurso invaluable para la comunidad global de desarrolladores y usuarios.



10. Trabajos futuros

El hecho de que la aplicación funcione no significa necesariamente que esté completa. A medida que avanzaba el desarrollo y realizábamos pruebas más exhaustivas, se hicieron evidentes oportunidades para mejorar tanto el funcionamiento como la usabilidad de la aplicación. Aquí destacamos algunas de las posibles mejoras que podrían considerarse para futuras versiones:

-Soporte para Diferentes Interfaces de Comunicación: En la actualidad, la aplicación se conecta a la placa Arduino a través del puerto serie. Sin embargo, considerando que existen otras placas de prototipado que implementan diferentes interfaces de comunicación, una mejora potencial sería la creación de un nuevo "store" que contenga métodos y propiedades necesarios para implementar una conexión utilizando tecnología Wifi u otras interfaces. Esto permitiría una mayor flexibilidad y adaptabilidad a diferentes dispositivos de hardware.

-Guardar Configuraciones Personalizadas: Sería beneficioso incorporar la capacidad de guardar configuraciones personalizadas para la ejecución repetida de pruebas. Esto ahorraría tiempo a los usuarios al eliminar la necesidad de configurar manualmente los parámetros cada vez que realicen una prueba similar. Estas configuraciones podrían incluir velocidades, límites de carga, configuraciones de hardware y más.

-Mejoras en Accesibilidad y Diseño: La colaboración con profesionales de la accesibilidad podría resultar en mejoras significativas en la usabilidad y el diseño de la aplicación. Esto podría incluir la optimización de la interfaz para dispositivos táctiles, mejoras en la legibilidad de las fuentes, opciones de alto contraste y compatibilidad con lectores de pantalla. Además, la integración con dispositivos de control externos, como controladores de accesibilidad, podría hacer que la aplicación sea más accesible para un público diverso.

-Captura de Video y Análisis Visual: Una característica interesante podría ser la incorporación del control de una cámara para capturar video durante las pruebas. Esto permitiría obtener imágenes de alta resolución a cámara lenta que documenten cada etapa de la prueba. Estos videos podrían asociarse con los datos numéricos obtenidos, lo que proporcionaría una visión más completa y clara de lo que sucede en cada momento de la prueba. Esto sería especialmente útil para el análisis posterior y la solución de problemas.

En resumen, la mejora continua es esencial para cualquier proyecto de desarrollo de software. La retroalimentación de los usuarios y la evolución de la tecnología siempre presentarán oportunidades para perfeccionar y expandir una aplicación. Estas mejoras potenciales no solo mejorarían la funcionalidad de la aplicación, sino que también la harían más versátil y accesible para un público más amplio.

11. Referencias

[1] Covid-19: uso y abuso de internet y teléfono móvil en estudiantes universitarios. Consultado en: <https://revista.infad.eu/index.php/IJODAEPA/article/view/2123>

[2] El número de usuarios de internet en el mundo crece un 1,9% y alcanza los 5.160 millones (2023).

Consultado en: <https://marketing4ecommerce.net/usuarios-de-internet-mundo/>

[3] Número de usuarios de Internet en el mundo entre 2005 hasta 2022.

Consultado en: <https://es.statista.com/estadisticas/541434/numero-mundial-de-usuarios-de-internet/>

[4] Open Source Initiative.

Disponible en: <https://opensource.org/>

[5] Ficha personal Antonio Iaccarino Idelson, en LinkedIn.

Consultado en: <https://www.linkedin.com/in/antonio-iaccarino-idelson-32460921/>

[6] An open-source biaxial tensile tester with automated pre-tensioning for mechanical studies of canvas paintings.

Disponible en: <https://www.sciencedirect.com/science/article/pii/S2468067223000196>

[7] Ficha personal Roger M. Groves, en Researchgate.

Consultado en: <https://www.researchgate.net/profile/Roger-Groves>

[8] Ficha personal Miguel Sánchez López, Universidad Politècnica de València.

Consultado en: <https://www.upv.es/ficha-personal/misan>

[9] INSTRON.

Consultado en: <https://www.instron.com/en/products/materials-testing-software>

[10] SHIMADZU.

Consultado en: <https://www.shimadzu.com/an/products/materials-testing/uni-ttm-software/index.html>

[11] GIBITRE.

Consultado en: https://www.gibitre.it/page_sin.php?ProdottoN=SwDinamo#

[12] Kyle Simpson, You Don't Know JS Yet: Get Started. 2020.

Disponible en: <https://github.com/getify/You-Dont-Know-JS>

Carlos Azaustre, Aprendiendo JavaScript: Desde cero hasta ECMAScript 6. 2016.

[13] ¿Qué es la arquitectura de tres niveles?

Consultado en: <https://www.ibm.com/es-es/topics/three-tier-architecture#:~:text=La%20arquitectura%20de%20tres%20niveles%20es%20una%20arquitectura%20de%20aplicaci%C3%B3n,datos%2C%20donde%20se%20almacenan%20y>

[14] Rendering on the Web (Renderizado en el lado del cliente).

Consultado en: [https://web.dev/rendering-on-the-web/#:~:text=Client%20side%20rendering%20\(CSR\)%20means%20rendering%20pages%20directly%20in,and%20keep%20fast%20for%20mobile.](https://web.dev/rendering-on-the-web/#:~:text=Client%20side%20rendering%20(CSR)%20means%20rendering%20pages%20directly%20in,and%20keep%20fast%20for%20mobile.)

[15] Diagrama de casos de uso.

Consultado en: <https://support.microsoft.com/es-es/office/crear-un-diagrama-de-casos-de-uso-uml-92cc948d-fc74-466c-9457-e82d62ee1298>

[16] How to Create a Progressive Web App (PWA Guide For Customers)

Consultado en: <https://www.scorchsoft.com/blog/how-to-create-progressive-web-app/>

[17] Read from and write to a serial port. (Chrome Developers).

Consultado en: <https://developer.chrome.com/articles/serial/>

Servicio de Publicación Web (Universidad Politècnica de València).

Consultado en: <https://www.upv.es/entidades/ASIC/catalogo/527544normalc.html>

[18] Servicio de webs avanzadas (Universidad Politècnica de València).

Consultado en: <https://wiki.upv.es/confluence/display/MANUALES/Servicio+de+webs+avanzadas>

[19] How to Host Node.js Applications on Plesk Obsidian.

Consultado en: <https://www.plesk.com/blog/product-technology/node-js-plesk-onyx/>

Hosting Node.js Applications.

Consultado en: <https://docs.plesk.com/en-US/obsidian/administrator-guide/website-management/hosting-nodejs-applications.76652/>

[20] Deploy Vue Options para Plesk.

Consultado en: <https://cli.vuejs.org/config/#outputdir>

Anexo I: Objetivos de desarrollo sostenible

Grado de relación del trabajo realizado con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	Alto	Medio	Bajo	No procede
ODS 1: Fin de la pobreza				x
ODS 2: Hambre cero				x
ODS 3: Salud y bienestar				x
ODS 4: Educación de calidad		x		
ODS 5: Igualdad de género				x
ODS 6: Agua limpia y saneamiento				x
ODS 7: Energía asequible y no contaminante				x
ODS 8: Trabajo decente y crecimiento económico			x	
ODS 9: Industria, innovación e infraestructuras		x		x
ODS 10: Reducción de las desigualdades				x
ODS 11: Ciudades y comunidades sostenibles				x
ODS 12: Producción y consumo responsables	x			
ODS 13: Acción por el clima			x	
ODS 14: Vida submarina				x
ODS 15: Vida de ecosistemas terrestres				x
ODS 16: Paz, justicia e instituciones sólidas				x
ODS 17: Alianzas para lograr los objetivos		x		

Tabla 8 : Objetivos de Desarrollo Sostenible

El proyecto de desarrollar una aplicación web para controlar una máquina de ensayos biaxial puede estar relacionado con varios Objetivos de Desarrollo Sostenible (ODS) establecidos por las Naciones Unidas. A continuación, se mencionan algunos de los ODS que podrían estar relacionados con este proyecto:

ODS 4 - Educación de Calidad: La aplicación web puede tener un componente educativo al proporcionar información y documentación sobre el funcionamiento de la máquina de ensayos biaxial. Además, facilita el acceso a herramientas de aprendizaje en línea, lo que contribuye a la educación y capacitación en el uso de esta tecnología.

ODS 8 - Trabajo Decente y Crecimiento Económico: La creación de una aplicación web y la mejora de la accesibilidad a esta tecnología pueden generar oportunidades de empleo y contribuir al crecimiento económico, especialmente en sectores relacionados con la ingeniería y la investigación.

ODS 9 - Industria, Innovación e Infraestructura: El desarrollo de una aplicación web que permite controlar y monitorear una máquina de ensayos biaxial representa un avance en la infraestructura tecnológica y la innovación en la industria. Esto puede contribuir a mejorar la eficiencia en la realización de pruebas y ensayos, así como a impulsar el desarrollo de tecnologías de control.

ODS 12 - Producción y Consumo Responsables: La optimización y automatización de los procesos de ensayo y monitoreo a través de la aplicación web pueden reducir el consumo de recursos y minimizar el desperdicio en el proceso de pruebas.

ODS 13 - Acción por el Clima: Si la aplicación se utiliza para el monitoreo y análisis de datos relacionados con pruebas de materiales y estructuras, podría contribuir a una mejor comprensión de cómo los materiales se comportan bajo diferentes condiciones, lo que podría llevar a un diseño más sostenible de estructuras y materiales.

ODS 17 - Alianzas para lograr los Objetivos: La colaboración con instituciones académicas, empresas y organizaciones de investigación en el desarrollo y la implementación de esta aplicación puede promover alianzas que contribuyan a alcanzar los ODS relacionados con la tecnología y la innovación.

Es importante destacar que la relación entre este proyecto y los ODS puede variar según su enfoque y aplicación específica. Sin embargo, al centrarse en la tecnología, la educación y la eficiencia, este proyecto tiene el potencial de contribuir positivamente a varios de los objetivos de desarrollo sostenible establecidos por las Naciones Unidas.

