



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Conversión de impresora 3D a sistema semi-automático de
ensayo de muestras biológicas

Trabajo Fin de Grado

Grado en Tecnologías Interactivas

AUTOR/A: Burruchaga Sola, Daniel

Tutor/a: Sogorb Devesa, Tomás Carlos

Cotutor/a: Pelegrí Sebastiá, José

CURSO ACADÉMICO: 2022/2023

Resumen:

El marco del proyecto es un sistema de análisis de muestras biológicas aplicando algoritmos de inteligencia artificial para la detección de determinadas enfermedades. Para evitar los errores y acelerar su análisis de las numerosas muestras. Se pretende automatizar dicho proceso, éste será el desarrollo en que se centrará este proyecto.

Se emplea una **impresora 3D** que recibe instrucciones en código **Gcode**. Estas instrucciones son generadas por un servicio *Python* desde una **Raspberry**. La interfaz gráfica de usuario, desarrollada en *React* y *TypeScript*, juega un papel crucial en la interacción con el sistema y la gestión de las peticiones, enviando estas peticiones a través de *MQTT*. La información se almacena en una base de datos *MongoDB* para su posterior análisis y consulta.

Este enfoque integral combina la **automatización** de procesos con tecnologías avanzadas para mejorar la eficiencia en la obtención de resultados en investigaciones biomédicas.

Automatización, Muestras biológicas, Impresora 3D, Raspberry, GCODE

Abstract:

The project frame is a system for analyzing biological samples using artificial intelligence algorithms to detect specific diseases. To avoid errors and speed up the analysis of numerous samples, the goal is to automate this process, which will be the focus of this project.

A 3D printer is employed to receive instructions in Gcode. These instructions are generated by a Python service from a Raspberry Pi. The user interface, developed in React and TypeScript, plays a crucial role in interacting with the system and managing requests, sending these requests via MQTT. Information is stored in a MongoDB database for subsequent analysis and querying.

This comprehensive approach combines process automation with advanced technologies to enhance efficiency in obtaining results in biomedical research.

Automation, Biological samples, 3D Printer, Raspberry, GCODE.

Prólogo

En la era de la tecnología y la innovación, la ciencia y la medicina han experimentado avances significativos. Sin embargo, aún existen áreas que requieren una mayor eficiencia y precisión, y una de ellas es la recogida de muestras biológicas. Este proyecto, que se presenta a continuación, es un intento de abordar este desafío y revolucionar el proceso de recogida y análisis de las muestras obtenidas.

El valor de este trabajo radica en su enfoque innovador para mecanizar y automatizar un proceso que ha sido tradicionalmente manual y, a veces, propenso a errores. Mediante la integración de tecnologías avanzadas como la impresión 3D y la programación GCODE, este proyecto busca mejorar la precisión y la eficiencia en la recogida de muestras biológicas.

El corazón de este sistema es una impresora 3D controlada por código G CODE a través de una interfaz web, que maneja la parte mecánica del proceso. La ejecución del sistema se realiza a través de una Raspberry Pi, conectada mediante un puerto USB, que ejecuta un servicio Python encargado de la comunicación serie con la impresora.

ÍNDICE

ÍNDICE	2
CAPÍTULO 1:	4
1. Introducción al Problema.....	4
1.1 Problema Propuesto.....	4
1.2 Solución.....	4
1.3 Extras:.....	5
1.4 Motivación del Proyecto.....	5
1.5 Objetivos.....	6
1.6 Impacto Esperado.....	6
1.7 Metodología.....	7
1.8 Estructura.....	10
CAPÍTULO 2:	12
2.1 Estado del arte.....	12
2.2 Crítica al estado del arte.....	12
2.3 Propuesta.....	13
CAPÍTULO 3:	14
3.1 Análisis del problema.....	14
3.1.1 Desafíos de la recogida de muestras biológicas:.....	14
3.1.2 Beneficios de la mecanización y automatización:.....	14
3.1.3 Consideraciones técnicas y tecnológicas:.....	14
3.1.4 Resumen.....	15
3.2 Especificación de requisitos:.....	15
3.2 Análisis de la seguridad.....	16
3.3 Análisis de riesgos.....	17
3.4 Identificación y análisis de soluciones posibles.....	18
3.5 Plan de Trabajo.....	19
3.6 Presupuesto.....	20
CAPÍTULO 4:	22
4.1 Diseño de la solución.....	22
4.2 Arquitectura del Sistema.....	23
4.3 Diseño Detallado.....	24
4.3.1 Arquitectura de Software Módulo Backend.....	24
4.3.1.1 Diseño Detallado del Código.....	24
4.3.1.2 Importación de Módulos.....	24
4.3.1.3 Configuración Inicial.....	24
4.3.1.4 Middleware y Configuración de Express.....	25
4.3.1.5 Configuración del Logger con express-winston.....	25
4.3.1.6 Configuración de Rutas.....	26
4.3.1.7 Ruta de Prueba.....	26
4.3.1.8 Inicio del Servidor.....	26
4.3.2 Módulo FrontEnd React.....	27
4.3.2.1 Arquitectura de Software.....	27

4.3.2.2 Resúmen.....	30
4.3.3 Servicio Python.....	30
4.3.3.1 Importación de módulos.....	30
4.3.3.2 Definición de la clase Service.....	31
4.3.3.3 Función de inicio Impresora 3D.....	32
4.3.3.4 Resúmen.....	32
4.4 Tecnologías Utilizadas.....	33
4.4.1 GCODE.....	33
4.4.2. Impresoras 3D Comunicación Serie.....	33
4.4.3. React.....	33
4.4.4. API REST.....	34
4.4.5. MongoDB.....	34
4.4.6. MQTT.....	35
CAPÍTULO 5.....	35
5.1 Desarrollo de la solución propuesta.....	36
CAPÍTULO 6.....	38
6. Implantación.....	38
6.1 Preparación para la Implantación.....	38
6.2 Puesta en Producción.....	38
6.3 Pruebas y Ajustes.....	39
6.4 Puesta en Marcha y Mantenimiento Continuo.....	42
6.6 Resúmen.....	42
CAPÍTULO 7.....	43
7. Pruebas.....	43
7.1 Pruebas de Verificación.....	43
7.2 Resultados y Conclusión de las Pruebas.....	44
CAPÍTULO 8.....	47
8.1 Conclusiones.....	47
CAPÍTULO 9.....	48
9. Trabajos futuros.....	48
CAPÍTULO 10.....	49
10. Referencias.....	49
ANEXOS.....	50
ANEXO I:.....	53
ANEXO II Objetivos de Desarrollo Sostenible.....	54

CAPÍTULO 1:

1. Introducción al Problema

En el vasto universo de la ciencia y la tecnología, nos encontramos en un punto de inflexión donde la automatización y la precisión se han convertido en imperativos. En este contexto, surge un desafío particular: la recogida y análisis de muestras biológicas. Aunque puede parecer un proceso sencillo, la realidad es que es un procedimiento que requiere una precisión y eficiencia, y aquí es donde nuestro proyecto entra en juego.

Para entender la importancia de este proyecto, es esencial comprender primero el problema que estamos tratando de resolver. La recogida de muestras biológicas es un proceso crucial en muchos campos, desde la medicina hasta la investigación científica. Sin embargo, este proceso ha sido tradicionalmente manual, lo que puede dar lugar a errores y variaciones. Además, la recogida manual de muestras puede ser un proceso lento y laborioso, lo que limita la cantidad de muestras que se pueden recoger y analizar en un período de tiempo determinado.

Nuestro proyecto busca abordar estos desafíos mediante la mecanización y automatización de la recogida de muestras biológicas. Al hacerlo, esperamos mejorar la precisión y eficiencia del proceso, permitiendo la recogida y análisis de un mayor número de muestras en menos tiempo.

Para facilitar la comprensión de este trabajo, hemos incluido un glosario al final del documento que define algunos de los términos técnicos que se utilizan a lo largo de la memoria. Si encuentra algún término o concepto que no comprende, le invitamos a consultar este glosario.

Este proyecto es el resultado de un esfuerzo por integrar diversas tecnologías, incluyendo la estructura y mecanizado usado para la impresión 3D, programación GCODE, Raspberry Pi como cerebro del proyecto, y un prototipo de plataforma web realizada con REACT, Node JS, MongoDB, para crear una solución innovadora a un problema persistente. Aunque este trabajo se centra en un área técnica específica, he hecho todo lo posible para presentarlo de una manera que sea accesible y comprensible para un público amplio.

Espero que este primer tramo introductorio le haya proporcionado una visión general del problema tratado a resolver y de cómo el proyecto se propone hacerlo. Le invitamos a seguir leyendo para obtener una visión más detallada del trabajo y de cómo se espera que contribuya a la mejora de la recogida y análisis de muestras biológicas

1.1 Problema Propuesto.

Para la elección de este proyecto, se plantea el problema de tener que tomar muestras biológicas de varios viales provenientes del Hospital la Fé de Valencia, de una forma manual, para un futuro procesado de estas muestras.

Se plantea un problema de automatizado para la obtención de muestras biológicas, las cuales van a estar a una baja temperatura y su posterior análisis.

1.2 Solución

En un inicio del proyecto, se planteó la idea de cómo obtener una solución mecánica al problema anteriormente descrito. Tras su estudio se planteó el desarrollo de un brazo mecánico colaborativo, pero se desecha la idea debido a sus costes y dificultades técnicas que puede suponer un problema para su desarrollo. También se plantea la idea de un símil de cinta transportadora, la cual fué sustituida por la opción final.

Finalmente la opción elegida fué su mecanización a través del reciclaje de una impresora 3D, ya que he realizado otros proyectos anteriormente con la implementación de código G CODE, por lo que ya conocía de él y podía ser una solución más realista acordes al tiempo y presupuesto.

Para controlar la puesta en marcha de la máquina, se propone una interfaz visual sencilla y accesible inalámbricamente.

1.3 Extras:

API con una base de datos, donde guardar usuarios, medidas y posibilidad futura de variables del algoritmo de IA.

Método de autenticación de usuario mediante JWT.

Interfaz web visual.

Integración de API MQTT para el control, y posibilidad de control externo a través del servidor local MQTT con credenciales.

Acceso a Interfaz mediante Access point wifi LAN.

1.4 Motivación del Proyecto

El Trabajo de Fin de Grado (TFG) que se presenta a continuación nace de una motivación profunda y personal: contribuir a la lucha contra el cáncer. Esta enfermedad, que afecta a millones de personas en todo el mundo, es un desafío médico y científico de gran magnitud. A pesar de los avances significativos en su tratamiento, aún queda mucho por hacer.

La recogida y análisis de muestras biológicas es un componente esencial en la lucha contra el cáncer. Estos procesos permiten a los médicos y científicos entender mejor la enfermedad, desarrollar tratamientos más efectivos y monitorear la respuesta del paciente a estos tratamientos. Sin embargo, la recogida manual de muestras puede ser propensa a errores y variaciones, lo que puede afectar la precisión de los resultados y, en última instancia, la eficacia del tratamiento.

Este proyecto busca abordar este desafío mediante la mecanización y automatización de la recogida de muestras biológicas. Al hacerlo, esperamos mejorar la precisión y eficiencia del proceso, lo que podría tener un impacto significativo en el tratamiento del cáncer.

La motivación para este proyecto no es sólo académica, sino también profundamente humana. Cada mejora en el proceso de recogida y análisis de muestras puede significar un paso más hacia un tratamiento más efectivo y, en última instancia, hacia la cura del cáncer. Es esta posibilidad, la de hacer una contribución, por pequeña que sea, a la lucha contra esta enfermedad, lo que ha impulsado este proyecto desde su concepción.

En última instancia, este TFG es un testimonio de la creencia en el poder de la tecnología y la innovación para mejorar la vida de las personas. A través de este trabajo, se espera no solo cumplir con los requisitos académicos de nuestro grado, sino también hacer una contribución significativa a la lucha contra el cáncer.

1.5 Objetivos

El objetivo principal de este proyecto es desarrollar un sistema automatizado y preciso para la obtención de muestras biológicas mediante el uso de una impresora 3D. Para lograr esto, se plantean los siguientes objetivos:

1. Diseñar un sistema que permita el mecanizado de obtención de muestras biológicas utilizando una impresora 3D. Este sistema estará configurado para controlar la impresora 3D mediante un archivo preconfigurado de posiciones a recorrer, lo que garantizará un proceso automatizado y preciso de obtención de muestras.
2. Configurar las preferencias del sistema para ajustar las variables de mecanizado según el tipo de muestra. Esto asegurará una precisión y calidad óptimas en la obtención de las muestras, teniendo en cuenta las características específicas de cada tipo de muestra biológica o número de viales.
3. Desarrollar una interfaz de usuario intuitiva que permita iniciar y detener la máquina de forma sencilla. Esta interfaz será accesible de forma inalámbrica, lo que brindará comodidad y flexibilidad al operador.
4. Implementar un sistema de almacenamiento y visualización de los datos obtenidos durante el proceso de mecanizado de obtención de muestras biológicas a través de una conexión LAN. Esto permitirá el monitoreo en tiempo real del proceso y el análisis de los datos para detectar posibles problemas y mejorar los resultados. Además, el sistema podrá guardar los parámetros de mecanizado y los resultados obtenidos para futuras referencias y análisis, lo que facilitará la optimización de los procesos y la obtención de muestras biológicas más precisas y confiables.

En resumen, los objetivos de este proyecto son diseñar un sistema automatizado para la obtención de muestras biológicas utilizando una impresora 3D, configurar las preferencias del sistema para garantizar la precisión y calidad óptimas, desarrollar una interfaz de usuario intuitiva y accesible, y establecer un sistema de almacenamiento y visualización de datos para el monitoreo y análisis del proceso. Estos objetivos se plantean con el propósito de mejorar la eficiencia y precisión en la obtención de muestras biológicas, lo que contribuirá al avance de la investigación y el tratamiento en el campo de la biotecnología.

1.6 Impacto Esperado

El objetivo principal de este proyecto es mejorar la eficiencia y precisión en la obtención de muestras biológicas mediante la mecanización y automatización del proceso de recolección. El producto resultante de este trabajo tendrá varias ventajas y mejoras significativas para los diferentes usuarios involucrados. A continuación, se detallan algunos de los impactos esperados:

1. Profesionales de la salud: Los médicos y científicos que trabajan en el campo de la investigación y el tratamiento del cáncer se beneficiarán de este proyecto. La automatización de la obtención de muestras biológicas les permitirá ahorrar tiempo y esfuerzo, ya que no tendrán que realizar este proceso manualmente. Además, la precisión mejorada garantizará resultados más confiables, lo que facilitará la toma de decisiones clínicas y el desarrollo de tratamientos más efectivos.

2. Pacientes: Los pacientes que se someten a pruebas y análisis de muestras biológicas también se verán beneficiados. La automatización del proceso reducirá el tiempo de espera para obtener los resultados, lo que permitirá un diagnóstico más rápido y un inicio temprano del tratamiento. Además, la precisión mejorada ayudará a evitar errores en los resultados, lo que garantizará un tratamiento adecuado y personalizado.
3. Investigadores: Los investigadores en el campo de la biotecnología y la medicina podrán aprovechar los datos recopilados durante el proceso de obtención de muestras. La conexión LAN y el sistema de almacenamiento permitirán el monitoreo en tiempo real y el análisis de los datos, lo que facilitará la detección de patrones y la identificación de posibles problemas. Esto contribuirá al avance de la investigación y al desarrollo de nuevos enfoques y terapias.

1.7 Metodología

Según la metodología SCRUM, “ Scrum se basa en el empirismo y el pensamiento Lean. El empirismo afirma que el conocimiento proviene de la experiencia y de la toma de decisiones con base en lo observado. El pensamiento Lean reduce el desperdicio y se enfoca en lo esencial “ (Ken Schwaber y Jeff Sutherland, 2020)

Acostumbrado a la metodología SCRUM, que tratamos en este Grado, se ha decidido aplicar esta metodología que es un marco de trabajo ágil para la gestión de proyectos de software. En particular, se puede considerar una metodología SCRUM unipersonal para el desarrollo del proyecto, ya que el estudiante de grado podría desempeñar los roles de Scrum Master, Product Owner y Equipo de Desarrollo.

En esta metodología, se pueden utilizar las ceremonias y artefactos de SCRUM adaptados a una situación unipersonal. Por ejemplo, se pueden realizar sesiones de sprint planning y sprint review para definir los objetivos y revisar el progreso del proyecto en ciclos cortos de una o dos semanas. Asimismo, se pueden utilizar herramientas como el backlog del producto para planificar y monitorear el trabajo.

También se ha aplicado la metodología ágil Kanban, mediante la herramienta Trello. Kanban es un método popular de gestión de flujo de trabajo Lean para definir, gestionar y mejorar servicios que realizan trabajos de conocimiento. Ayuda a visualizar el trabajo, maximizar la eficiencia y mejorar de forma continua. El trabajo se representa en tableros Kanban, lo que te permite optimizar la entrega de trabajo.

En resumen, al combinar la metodología SCRUM y Kanban, en un proyecto de grado en Ingeniería, se puede alcanzar una mayor eficiencia y calidad en el desarrollo de resolución de problemas aprovechando las ventajas de ambos enfoques.

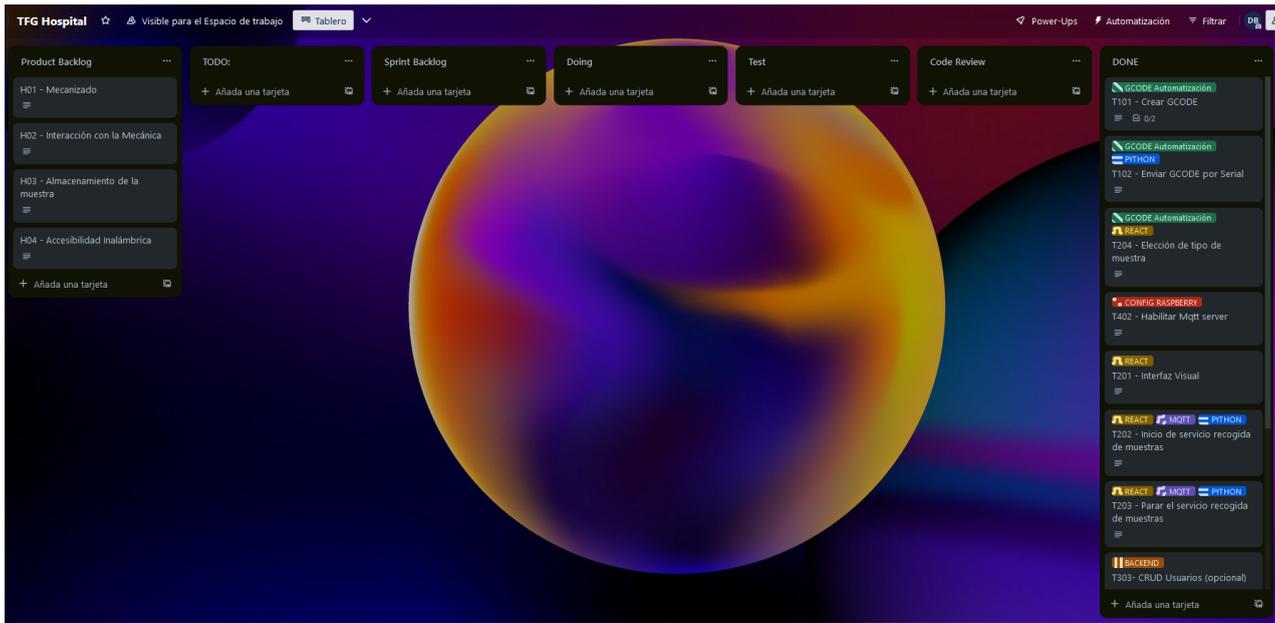


Figura 1: Kanban del proyecto en utilizando la herramienta Trello.

Disponible en [TRELLO](https://trello.com).

Desarrollo y resultados del trabajo

Una vez escogido las diferentes tecnologías seleccionadas para el desarrollo del trabajo se plantean cuatro historias de usuario a desarrollar:

H01 - Mecanizado

Como usuario, quiero poder ingresar a una interfaz gráfica web en React para poder activar el código GCODE y poder iniciar la recogida de medidas mediante un sensor y el mecanizado con una impresora 3D, para poder obtener medidas con mayor precisión y eficiencia.

Se ha subdividido en varias tareas:

- T101- Crear GCODE

Se debe de crear un código capaz de recorrer las muestras en lenguaje GCODE. Como pruebas de aceptación se debe de probar la impresora 3D con este código y realizar el recorrido esperado.

- T102 - Enviar GCODE por comunicación serie (python)

El código anterior debe de ser transmitido a la impresora 3D a través de comunicación serie. Se usará un servicio en Python para realizar este paso.

H02 - Interacción con la Mecánica

Como usuario, quiero tener la opción de poder iniciar el proceso mecánico de obtención de muestras mediante la impresora 3D. Para poder interactuar con ello es necesario una interfaz gráfica accesible para el usuario.

Se ha subdividido en varias tareas:

- T201 - Interfaz Visual

Se debe de crear una interfaz visual desde la cual poder interactuar con la máquina. Debe de constar de un inicio de sesión. Debe de constar de una tabla de recogida de datos. Debe de constar de un apartado de edición de recorrido / inicio / parada.

- T202 - Inicio de servicio recogida de muestras

Debe de constar de un botón en la interfaz visual que inicie este servicio de recogida de muestras. A través de MQTT se realizará el inicio de un servicio de recogida de muestras.

- T203 - Parar el servicio recogida de muestras

Debe de constar de un botón en la interfaz visual que detenga este servicio de recogida de muestras. A través de MQTT se realizará el stop de un servicio de recogida de muestras.

- T204 - Elección de tipo de muestra

Debe de constar de dos configuraciones disponibles para 8,16,32 viales. Por lo cual deben de haber códigos G CODE editables.

H03 - Almacenamiento de la muestra

Como usuario quiero poder obtener las medidas obtenidas mediante el sensor y almacenarlas en nuestra base de datos.

Se ha subdividido en varias tareas:

- T301 - Comunicación Interfaz Visual

Realizar la configuración necesaria para conectar el backend con la interfaz visual. Se deben de realizar varias llamadas como el inicio de sesión, recolección de muestras, etc.

- T302 - CRUD Medidas

Se debe poder crear, editar, eliminar y obtener los datos relacionados con las medidas.

- T303 - CRUD Usuarios (opcional)

Se debe poder crear, editar, eliminar y obtener los datos relacionados con los Usuarios. Debe de contar con un token JWT para inicio de sesión.

H04 - Accesibilidad Inalámbrica

Como usuario, quiero poder comunicarme con esta interfaz sin estar conectado mediante un cable. Si no que desde cualquier dispositivo debe de ser accesible.

Se ha subdividido en varias tareas:

- T401 - Habilitar Access Point en Raspberry

El proyecto debe de constar de un Access point wifi en la raspberry pi para poder acceder a estos servicios.

- T402 - Habilitar Mqtt server

El proyecto debe de constar de un servidor MQTT en la raspberry pi de modo local.

1.8 Estructura

El primer capítulo establece una introducción detallada al problema en cuestión, a la par que introduce la solución propuesta, expone la motivación detrás del proyecto, define los objetivos a cumplir, describe la metodología empleada y expone la estructura planificada para el desarrollo del trabajo.

En el segundo capítulo, se profundiza en el estado actual del campo de estudio, realizando un análisis crítico del panorama tecnológico vigente y proponiendo innovaciones que cubran las deficiencias identificadas. Se explora en detalle el estado del arte, identificando oportunidades para mejorar la tecnología y la metodología existente.

El tercer capítulo aborda con rigurosidad el análisis de la seguridad, identificando y evaluando posibles riesgos inherentes al desarrollo y la implementación de la solución. Se proponen soluciones para mitigar estos riesgos, se presenta un plan de trabajo con pasos concretos para llevar a cabo la implementación y se detalla un presupuesto que cubra los recursos necesarios.

El cuarto capítulo se centra en el diseño de la solución, detallando minuciosamente la arquitectura del sistema, estableciendo los componentes clave y sus interacciones. Asimismo, se describe el diseño detallado de cada componente, proporcionando una visión precisa de cómo cada elemento contribuye a la operación global del sistema.

En el quinto capítulo, se explora en profundidad la etapa de implementación, detallando el proceso de transformar el diseño en una solución funcional. Se destacan los aspectos técnicos y se discuten los desafíos que surgieron durante este proceso.

El sexto capítulo trata de la etapa de implantación, presentando la integración del sistema en un entorno operativo real. Se describen las pruebas realizadas para verificar la funcionalidad del sistema y se discuten los ajustes realizados en respuesta a los resultados de las pruebas.

El séptimo capítulo se dedica a las pruebas, dividiéndose en verificación, validación y pruebas de carga. Se profundiza en la realización de pruebas exhaustivas para confirmar la funcionalidad, la adhesión a los requisitos y la capacidad del sistema para manejar situaciones de alta demanda.

El octavo capítulo concentra las conclusiones obtenidas del proyecto, relacionándolas con los conocimientos adquiridos durante los estudios previos. Se evalúa el logro de los objetivos y se brinda una perspectiva global de los resultados alcanzados.

El noveno capítulo proyecta futuras direcciones de trabajo, identificando áreas de mejora y posibles expansiones de la solución desarrollada.

Por último, el décimo capítulo proporciona una recopilación de las referencias bibliográficas que respaldan las afirmaciones y el análisis presente en el trabajo. Además, se incorpora un glosario o diccionario de términos especializados al final del TFG, permitiendo al lector consultar el significado preciso de terminología específica relacionada con el ámbito de la memoria.

CAPÍTULO 2

2.1 Estado del arte

En el estado del arte de este proyecto, se busca investigar y analizar las soluciones existentes relacionadas con la mecanización y automatización de la recogida de muestras biológicas, así como su posterior análisis y conclusión. A continuación, se describirán algunas tecnologías y enfoques relevantes encontrados en el estado del arte:

Mecanización y automatización de la recogida de muestras biológicas: En la actualidad, existen diferentes enfoques para automatizar la recogida de muestras biológicas. Estos pueden incluir dispositivos robóticos especializados diseñados para manipular y recolectar muestras, así como sistemas de automatización que utilizan brazos mecánicos para realizar tareas específicas. Un estudio por la universidad de Perú, (Gutierrez Zamudio, 2019) ***[Ref. N° 10]** donde trata de recrear una impresora 3D con motores Nema 17, típicos en las impresoras 3D, con fines de crear una automatización de recolecta de muestras biológicas.

Al realizar el estudio del estado del arte, se han examinado otras soluciones existentes que abordan la automatización de la recogida de muestras biológicas. Sin embargo, el enfoque propuesto en este proyecto innovador, resuelve el problema de forma más sencilla y reciclada. La integración de una interfaz web y el uso de tecnologías como React, Node.js y MongoDB también proporciona una plataforma completa y versátil para el control y análisis de los datos recolectados.

2.2 Crítica al estado del arte

Al analizar los trabajos previos presentados en la ETSINF relacionados con la mecanización y automatización de la recogida de muestras biológicas, se observa una falta de enfoque específico en el uso de impresoras 3D y la integración con una Raspberry Pi para el control y supervisión del proceso.

Aunque se han encontrado algunos trabajos que abordan la automatización de la recogida de muestras, estos suelen centrarse en sistemas robóticos especializados o en la implementación de brazos mecánicos. Sin embargo, se ha identificado una laguna en cuanto al uso de impresoras 3D como una herramienta central en el proceso de mecanización.

Además, se ha observado que muchos de los trabajos previos carecen de una interfaz web completa para el control y configuración del sistema, así como una integración adecuada de tecnologías como React, Node.js y MongoDB para el manejo de datos y análisis posteriores.

Por lo tanto, el objetivo de este TFG es cubrir estas lagunas y proporcionar una solución integral que aproveche las ventajas de las impresoras 3D y la Raspberry Pi, permitiendo la mecanización y automatización eficiente de la recogida de muestras biológicas. Además, la implementación de una interfaz web completa y el uso de tecnologías modernas en el desarrollo permitirán una mayor flexibilidad y facilidad de uso en comparación con los enfoques existentes.

En resumen, el análisis crítico del estado del arte ha revelado la necesidad de desarrollar un enfoque más específico que utilice impresoras 3D y una Raspberry Pi para la mecanización y automatización de la recogida de muestras biológicas. Además, la integración de una interfaz web completa y el uso de tecnologías modernas fortalecerán la solución propuesta y la diferenciarán de los enfoques existentes.

**Diseño de robot para toma de sangre y rotulado de muestras en hospitales (Publicación n.º 13763)*

2.3 Propuesta

La propuesta de este proyecto es desarrollar un sistema que mecanice y automatice la recogida de muestras biológicas, su posterior análisis y conclusión. El espacio de conocimiento y tecnología que este trabajo busca llenar es el de la automatización y optimización de procesos en el campo de la biología y la investigación científica.

Lo que diferencia y mejora este trabajo en comparación con las soluciones existentes es la combinación de tecnologías y enfoques de manera original y novedosa. La utilización de una impresora 3D controlada mediante código G CODE, junto con una Raspberry Pi y una interfaz web.

Si bien este proyecto puede considerarse una prueba de concepto, su originalidad radica en la combinación de tecnologías existentes de manera innovadora y en su potencial para mejorar la eficiencia y precisión en la recogida de muestras biológicas. Al automatizar tareas que antes requerían una intervención manual intensiva, este trabajo tiene el potencial de agilizar los procesos de investigación y análisis, permitiendo un avance más rápido y efectivo en el campo de la biología.



Figura 2: Impresora Witbox Utilizada

En resumen, este proyecto propone una solución original y novedosa que combina tecnologías como la impresión 3D, el control mediante código G CODE, la Raspberry Pi y una interfaz web. Al mecanizar y automatizar la recogida de muestras biológicas, este trabajo busca mejorar la eficiencia, precisión y velocidad de los procesos de investigación y análisis, brindando así una contribución significativa al espacio de conocimiento y tecnológico existente.

CAPÍTULO 3:

3.1 Análisis del problema

El análisis en profundidad del problema que resuelve este proyecto de mecanización y automatización de la recogida de muestras biológicas involucra comprender detalladamente los desafíos y oportunidades que surgen al abordar este problema. A continuación, se presenta un análisis exhaustivo del problema:

3.1.1 Desafíos de la recogida de muestras biológicas:

- Proceso manual y propenso a errores: La recogida manual de muestras biológicas es un proceso que requiere precisión y cuidado para evitar contaminaciones y errores en la recolección. La automatización busca minimizar estos errores humanos y mejorar la calidad de las muestras recolectadas.
- Tiempo y recursos necesarios: La recogida manual de muestras puede ser un proceso lento y requerir una cantidad significativa de personal y recursos. La mecanización y automatización buscan agilizar este proceso y reducir la dependencia de recursos humanos.
- Necesidad de precisión y repetibilidad: Para garantizar resultados consistentes y confiables en el análisis de las muestras, es crucial que el proceso de recogida sea preciso y repetible. La automatización puede contribuir a alcanzar estos objetivos al eliminar variaciones inherentes a la intervención humana.

3.1.2 Beneficios de la mecanización y automatización:

- Mejora de la eficiencia y productividad: Al automatizar la recogida de muestras biológicas, se reduce la dependencia de recursos humanos y se agiliza el proceso, lo que resulta en una mayor eficiencia y productividad.
- Reducción de errores: La mecanización y automatización minimizan la posibilidad de errores humanos en la recogida de muestras, lo que mejora la calidad y confiabilidad de los datos obtenidos.
- Mayor control y precisión: Al utilizar una impresora 3D y código G CODE para controlar la parte mecánica del proceso, se puede lograr un mayor control y precisión en la recogida de muestras, lo que resulta en muestras más consistentes y representativas.
- Capacidad de escalabilidad: La automatización permite que el proceso de recogida de muestras se pueda escalar fácilmente para satisfacer necesidades crecientes o cambiar las demandas de la investigación o el análisis biológico.

3.1.3 Consideraciones técnicas y tecnológicas:

- Integración de hardware y software: El proyecto requiere la integración de componentes físicos, como la impresora 3D y la Raspberry Pi, con software de control y comunicación para garantizar un funcionamiento armonioso y coordinado.
- Implementación de una interfaz web: La interfaz web basada en React y el backend en Node.js permiten controlar y configurar el sistema, así como visualizar y analizar los datos obtenidos. Esto requiere habilidades de desarrollo web y conocimientos en tecnologías específicas.
- Comunicación y sincronización: El servicio Python en la Raspberry Pi debe ser capaz de comunicarse con la impresora 3D mediante el lenguaje GCODE y

- mantener una sincronización precisa para garantizar un control efectivo de la parte mecánica del proceso.
- Seguridad y privacidad: Dado que se están recopilando datos biológicos, es fundamental implementar medidas de seguridad y privacidad para proteger la integridad y confidencialidad de los datos.

3.1.4 Resumen

En resumen, este proyecto aborda los desafíos asociados con la recogida manual de muestras biológicas mediante la mecanización y automatización del proceso. Además, se deben tener en cuenta consideraciones técnicas y tecnológicas para lograr una implementación exitosa y garantizar la seguridad y privacidad de los datos recolectados.

3.2 Especificación de requisitos:

Requisitos funcionales:

1.1 Control de la impresora 3D:

Permitir el envío de comandos GCODE a la impresora 3D para controlar sus movimientos y operaciones.

Controlar el inicio y finalización del proceso de impresión.

Posibilidad futura de configurar parámetros de impresión, como velocidad, temperatura y precisión a través de código GCODE.

1.2 Interfaz web:

Proporcionar una interfaz web intuitiva y fácil de usar para controlar la parte mecánica del sistema.

Permitir la configuración de preferencias del sistema, como opciones de impresión y ajustes de calibración.

Posibilidad futura de mostrar información en tiempo real sobre el estado de la impresora, como temperatura, progreso de impresión y posibles errores.

1.3 Comunicación entre la Raspberry Pi y la impresora 3D:

Establecer una comunicación serial entre la Raspberry Pi y la impresora 3D utilizando un puerto USB.

Implementar un servicio Python en la Raspberry Pi para gestionar la comunicación con la impresora 3D y enviar comandos GCODE.

1.4 Almacenamiento de datos:

Utilizar una base de datos local en MongoDB para almacenar los datos de las muestras biológicas recogidas.

Permitir la inserción, consulta y modificación de datos de muestras en la base de datos.

Garantizar la integridad y confidencialidad de los datos almacenados.

2. Requisitos no funcionales:

2.1 Rendimiento:

El sistema debe ser capaz de procesar los comandos GCODE y controlar la impresora 3D de manera eficiente y en tiempo real. 2.1.2 La interfaz web debe responder de forma rápida y fluida, sin retrasos significativos en la actualización de la información.

2.2 Usabilidad:

La interfaz web debe ser intuitiva y fácil de navegar, con una estructura clara y un diseño atractivo. 2.2.2 Proporcionar retroalimentación visual y mensajes de error claros en caso de operaciones incorrectas o fallidas.

2.3 Seguridad:

Implementar medidas de seguridad para proteger el acceso no autorizado a la interfaz web y la base de datos.

Garantizar la privacidad y confidencialidad de los datos de las muestras biológicas almacenadas.

2.4 Escalabilidad:

Diseñar el sistema de manera que pueda adaptarse y expandirse en el futuro, permitiendo la integración de nuevas funcionalidades y componentes si es necesario.

3.2 Análisis de la seguridad

1. Acceso controlado y autenticación:

Implementar un sistema de autenticación seguro para controlar el acceso a la interfaz web y a las funcionalidades del sistema. Esto puede incluir autenticación basada en contraseñas, autenticación de dos factores o incluso autenticación biométrica, según el nivel de seguridad requerido.

Asignar roles y privilegios de usuario para restringir el acceso a funciones específicas dentro del sistema. Por ejemplo, solo permitir a usuarios autorizados la configuración del sistema o la visualización de datos sensibles.

2. Protección de datos personales y privacidad:

Cumplir con las regulaciones de protección de datos aplicables, como el Reglamento General de Protección de Datos (RGPD) en Europa u otras leyes de privacidad de datos en tu jurisdicción. Asegurarse de que los datos personales de los usuarios se recolectan, almacenen y utilicen de acuerdo con las regulaciones pertinentes.

Utilizar técnicas de encriptación para proteger los datos personales y sensibles almacenados en la base de datos, asegurando que solo los usuarios autorizados puedan acceder a ellos.

3. Protección contra accesos no autorizados:

Implementar medidas de seguridad para proteger el sistema contra accesos no autorizados, como ataques de fuerza bruta o intentos de manipulación de datos. Esto puede incluir la implementación de firewalls, sistemas de detección de intrusiones y auditorías de seguridad regulares.

Mantener actualizados los sistemas operativos, software y librerías utilizados en el proyecto, aplicando parches de seguridad y actualizaciones recomendadas por los proveedores.

4. Seguridad física y ambiente controlado:

Garantizar la seguridad física de los dispositivos utilizados en el proyecto, como la Raspberry Pi y la impresora 3D. Esto implica protegerlos contra robos, daños o accesos no autorizados.

Mantener un ambiente controlado para la recogida de muestras biológicas, garantizando que se cumplan los estándares de seguridad y manejo adecuado de sustancias biológicas.

5. Capacitación y concienciación:

Proporcionar capacitación a los usuarios y el personal involucrado en el proyecto sobre las prácticas de seguridad y la importancia de proteger la privacidad de los datos personales y el manejo adecuado de las muestras biológicas.

Fomentar una cultura de seguridad dentro del equipo y promover buenas prácticas de seguridad, como el uso de contraseñas seguras, el reporte de incidentes de seguridad y la actualización regular de los sistemas.

Es importante destacar que la seguridad es un aspecto crítico y debe considerarse en todas las etapas del proyecto, desde el diseño hasta la implementación y el mantenimiento continuo..

3.3 Análisis de riesgos

Riesgo de aceptación:

Resistencia al cambio: Existe el riesgo de que los usuarios y las partes interesadas muestren resistencia al cambio al adoptar el nuevo sistema de mecanización y automatización de la recogida de muestras biológicas. Pueden surgir preocupaciones sobre la confiabilidad, la eficacia o la familiaridad con el nuevo enfoque tecnológico. Es importante abordar estas preocupaciones mediante una comunicación efectiva, demostrando los beneficios y capacitando adecuadamente a los usuarios en la utilización del nuevo sistema.

Falta de comprensión o conciencia: Si los usuarios y las partes interesadas no comprenden completamente los beneficios o el propósito del proyecto, pueden mostrar una menor disposición a aceptarlo. Es fundamental comunicar de manera clara y efectiva los objetivos, las ventajas y el impacto positivo del sistema automatizado en la recogida de muestras biológicas.

Limitaciones técnicas o de recursos: Si el sistema implementado presenta limitaciones técnicas o requiere recursos adicionales significativos, como actualizaciones de hardware o entrenamiento especializado, esto puede afectar la aceptación. Es importante garantizar que el sistema sea accesible, compatible con los recursos existentes y que los usuarios tengan el apoyo necesario para su uso efectivo.

Indicadores de satisfacción:

Retroalimentación de los usuarios: Medir la satisfacción a través de encuestas, entrevistas o cuestionarios dirigidos a los usuarios y las partes interesadas involucradas en el proceso de recogida de muestras. Las preguntas pueden estar relacionadas con la facilidad de uso, la eficiencia, la calidad de las muestras recolectadas y la mejora general en comparación con el método manual anterior.

Eficiencia y productividad mejoradas: Comparar métricas de rendimiento antes y después de la implementación, como el tiempo necesario para recolectar muestras, la reducción de errores o la cantidad de muestras procesadas en un período determinado. Estos indicadores pueden demostrar la eficiencia y productividad mejoradas logradas con el nuevo sistema automatizado.

Cumplimiento de los objetivos del proyecto: Evaluar si el sistema automatizado cumple con los objetivos y requisitos establecidos en la etapa de planificación del proyecto. Esto puede incluir criterios específicos, como la precisión de la recogida de muestras, la capacidad de personalización o la interoperabilidad con otros sistemas o herramientas utilizadas en el análisis y la conclusión de las muestras biológicas.

3.4 Identificación y análisis de soluciones posibles

La identificación y análisis de las soluciones posibles para el proyecto de mecanización y automatización de la recogida de muestras biológicas es un paso crucial para determinar la opción más adecuada. Aunque se mencionó que se consideraron propuestas de un brazo robótico y una cinta transportadora, finalmente se decidió reciclar una impresora 3D para lograr los objetivos del proyecto. A continuación, se detallan las posibles soluciones y el análisis realizado:

1. Brazo robótico:

Propuesta: Utilizar un brazo robótico para realizar la recogida automatizada de muestras biológicas.

Análisis: Si bien un brazo robótico podría proporcionar una solución precisa y controlada, su implementación puede requerir un presupuesto significativo debido a los costos asociados con el hardware y la programación especializada. Además, el brazo robótico puede requerir una integración compleja con el sistema existente y la interfaz web. Dado el presupuesto limitado, esta opción fue denegada.



Figura 3: Brazo robótico para manipulación de muestras biológicas

2. Cinta transportadora:

Propuesta: Implementar una cinta transportadora para el transporte automático de las muestras biológicas.

Análisis: Una cinta transportadora puede ofrecer una solución eficiente y continua para la recogida de muestras. Sin embargo, su implementación puede requerir un espacio físico considerable y un diseño específico para acomodar las necesidades del proyecto. Además, la integración con el sistema existente y la interfaz web puede ser un desafío. Considerando los recursos disponibles y las restricciones de espacio, se determinó que esta opción no era la más viable.

3. Reciclaje de impresora 3D:

Propuesta: Reutilizar una impresora 3D existente para mecanizar y automatizar la recogida de muestras biológicas.

Análisis: Al reciclar una impresora 3D, se aprovechan los componentes y la funcionalidad existentes, lo que puede reducir los costos y simplificar la implementación. La impresora 3D ofrece una plataforma mecánica adecuada para controlar el proceso de recogida de muestras, y su control mediante código G CODE permite una programación precisa y personalizable. Además, la integración con una Raspberry Pi, una interfaz web y una base de datos local proporciona una solución completa y flexible. Dadas las consideraciones de presupuesto y recursos, esta opción se seleccionó como la más adecuada.

El análisis consideró los aspectos económicos, técnicos y de integración para evaluar las diferentes propuestas. Teniendo en cuenta el presupuesto limitado y la necesidad de una solución funcional y eficiente, la opción de reciclar una impresora 3D se consideró la mejor elección. Esto permitió aprovechar los componentes existentes, utilizar el control mediante código G CODE y la integración con una Raspberry Pi y una interfaz web para lograr los objetivos del proyecto.

3.5 Plan de Trabajo

Este plan se divide en varias fases y estima el esfuerzo requerido en horas para cada una de ellas.

1. Investigación y planificación (30 horas):

Revisión y estudio del estado del arte sobre la mecanización de la recogida de muestras biológicas (10 horas).

Definición de requisitos y especificación del proyecto (10 horas).

Planificación del cronograma y estimación de recursos (10 horas).

2. Desarrollo del controlador de impresora 3D (40 horas):

Investigación y familiarización con el hardware de la impresora 3D, Raspberry Pi (10 horas).

Programación del controlador de la impresora 3D utilizando Python y la comunicación serial con GCODE (20 horas).

Pruebas y depuración del controlador (10 horas).

3. Desarrollo de la interfaz de usuario (60 horas):

Diseño de la interfaz de usuario (10 horas).

Desarrollo del front-end utilizando React (30 horas).

Desarrollo del back-end utilizando Node.js para la conexión con la Raspberry Pi (20 horas).

4. Implementación de la API y base de datos (30 horas):

Diseño y configuración de la API para la comunicación entre la interfaz de usuario y la base de datos (10 horas).

Implementación de la base de datos local en MongoDB (10 horas).

Pruebas de la API y la base de datos (10 horas).

5. Pruebas, integración y ajustes (40 horas):

Realización de pruebas unitarias y de integración para garantizar el correcto funcionamiento de todas las partes del sistema (20 horas).

Identificación y corrección de posibles errores o problemas (10 horas).

Ajustes y mejoras adicionales según la retroalimentación y las necesidades del proyecto (10 horas).

6. Documentación y cierre (20 horas):

Elaboración de la documentación final del proyecto, incluyendo la memoria del TFG y manuales de uso (15 horas).

Revisión final y preparación para la presentación (5 horas).

Total estimado: 220 horas.

3.6 Presupuesto

1. Investigación de tecnologías:

Costos de acceso a revistas científicas, publicaciones, libros o recursos en línea para investigar las tecnologías y conceptos relevantes.

2. Desarrollo del controlador de impresora 3D:

Costos de hardware, como una Raspberry Pi y un conversor FTDI para la comunicación con la impresora 3D.

Costos de desarrollo de software para el controlador de la impresora 3D, incluyendo la programación en Python y el desarrollo de la comunicación serial con GCODE.

3. Desarrollo de la interfaz de usuario:

Costos de desarrollo de software para el front-end de la interfaz de usuario basado en React y el back-end en Node.js para la conexión con la Raspberry Pi.

Costos de diseño de interfaz y experiencia de usuario (UX/UI) para crear una interfaz atractiva y fácil de usar.

4. API y base de datos:

Costos de desarrollo de software para la implementación de una API que permita la comunicación entre la interfaz de usuario y la base de datos.

Costos de configuración y mantenimiento de la base de datos local en MongoDB .

5. Infraestructura y hosting:

Costos asociados con el alojamiento de la interfaz de usuario y la API, que pueden incluir servicios de alojamiento web, servidores dedicados o servicios en la nube.

YSOLVE S.L

Avenida Hermanos Machado 13, 204 46019 Valencia
danielburruchaga@ysolve.es
(+34) 627669080



Presupuesto

Fecha: 1/06/2023

Presupuesto para:

DNI:

CIF:

Realizado por:

Daniel Burruchaga Sola

Invoice

100030

Proyecto:

Hospital la Fé de Valencia

Fecha de vencimiento

15/08/2023

Description

Price

Inicio de proyecto

2400 €

Desarrollo del controlador de impresora 3D

3200 €

Diseño y desarrollo de la interfaz de usuario

4750 €

Backend Microservicios

6700 €

Infraestructura y hosting (OPCIONAL)

599.75 €

TOTAL

€17,649.75

Pago 60% al inicio del proyecto, 40% a la finalización

SIN IVA

"Al firmar el presente acuerdo consta en acta que ambas partes estan de acuerdo en el presupuesto a falta de formalizar contrato"

Tabla 1: Presupuesto del proyecto.

CAPÍTULO 4

4.1 Diseño de la solución

La solución planteada se basa en una comunicación cliente-servidor, donde el cliente se conecta a través del punto de acceso WiFi de la Raspberry Pi y accede a una interfaz web para controlar y configurar la impresora 3D. Además, se utiliza un servicio MQTT para la comunicación con dispositivos hardware externos, y se implementa un servicio de autenticación API para garantizar la seguridad y el acceso adecuado. Los datos recolectados se almacenan en una base de datos local.

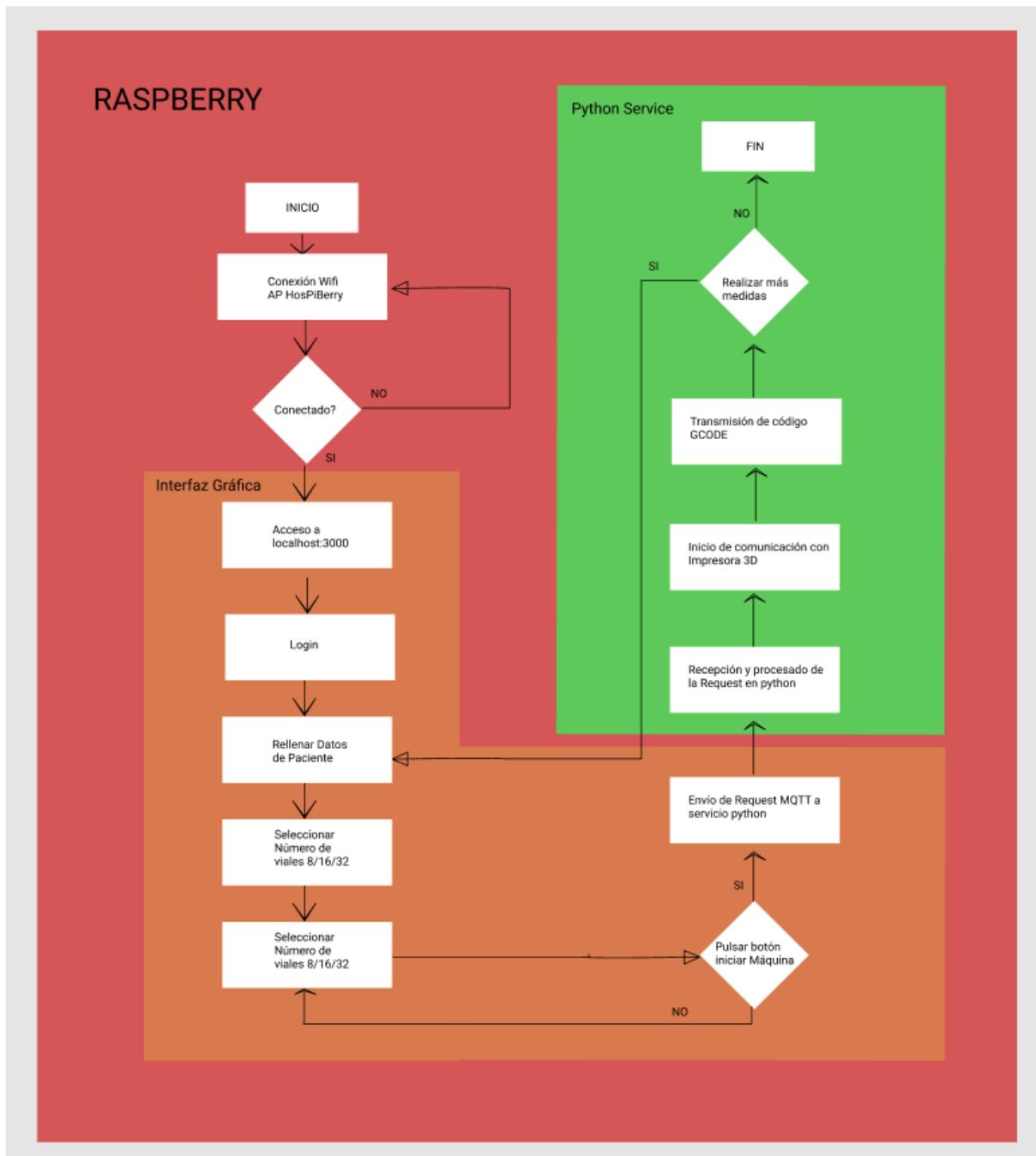


Tabla 2: Flujo del software desarrollado.

4.2 Arquitectura del Sistema

La arquitectura del proyecto seleccionado se basa en una comunicación cliente-servidor, donde el cliente se conecta a través del punto de acceso WiFi proporcionado por la Raspberry Pi. A continuación, se explica la arquitectura en detalle:

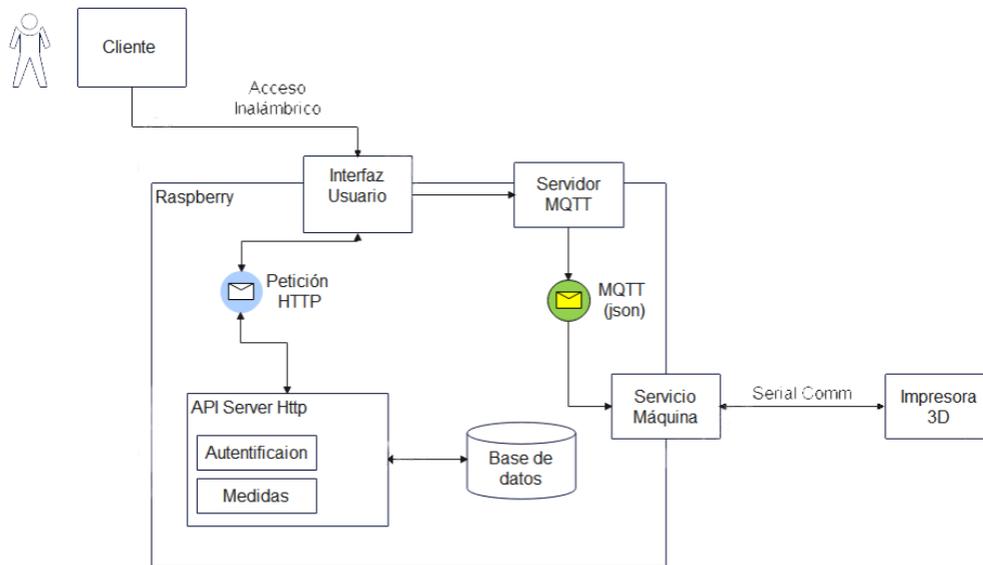


Tabla 3: Arquitectura del proyecto.

1. Punto de acceso WiFi:
 - La Raspberry Pi actúa como un punto de acceso WiFi, permitiendo que los clientes se conecten a ella a través de una red inalámbrica.
2. Interfaz web:
 - El cliente accede a una interfaz web a través de un puerto específico en la Raspberry Pi. Esta interfaz proporciona las funcionalidades necesarias para controlar y configurar la impresora 3D.
3. Control de la impresora 3D:
 - La interfaz web permite al cliente enviar comandos de control a la impresora 3D. Estos comandos, en forma de código G CODE, se envían a través de la comunicación serial para controlar los movimientos y la operación de la impresora.
4. Servicio MQTT:
 - Se desarrolla un servicio MQTT en Python para la comunicación entre el cliente y otros dispositivos hardware, como sensores externos. MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería ligero y eficiente para la comunicación entre dispositivos IoT. Se elige MQTT debido a su escalabilidad y flexibilidad para la implementación de sensores externos y la obtención de datos adicionales.
5. Servicio de autenticación API:
 - Se implementa un servicio de autenticación API para gestionar la seguridad y el acceso a la interfaz web y los datos almacenados. Este servicio permite autenticar a los usuarios y garantizar que solo los usuarios autorizados puedan acceder y controlar la impresora 3D y visualizar los datos almacenados.
6. Almacenamiento de datos:
 - Se utiliza una base de datos local para almacenar los datos obtenidos durante el proceso de mecanización de las muestras biológicas. Se menciona anteriormente que se utiliza MongoDB como base de datos para almacenar los datos recolectados.

4.3 Diseño Detallado

4.3.1 Arquitectura de Software Módulo Backend

4.3.1.1 Diseño Detallado del Código

El código proporcionado es una implementación de un servidor Express.js que expone varias rutas y configuraciones para manejar solicitudes HTTP. A continuación, se presenta un desglose detallado del diseño y funcionamiento de este código:

4.3.1.2 Importación de Módulos

```
import express, { Application } from 'express';
import * as http from 'http';
import * as winston from 'winston';
import * as expressWinston from 'express-winston';
import cors from 'cors';
import {CommonRoutesConfig} from './router/common.routes.config';
import {AuthRoutes} from './router/auth/auth.route.config';
import {UsersRoutes} from './router/users/users.route.config';
import {MeasuresRoutes} from './router/measures/measures.route.config';
import {ControlRoutes} from './router/controlMachine/controlMachine.route.config';
import debug from 'debug';
import './config/connection'
```

Figura 4: Código importación de módulos.

En esta sección, se importan los módulos necesarios para construir el servidor. Esto incluye los módulos de Express, HTTP, logging (winston y express-winston), manejo de CORS, rutas de la aplicación y módulos personalizados. Además, se importa el módulo 'debug' para habilitar mensajes de depuración y se establece la conexión a una base de datos utilizando el archivo 'connection.js'.

4.3.1.3 Configuración Inicial

```
const app: Application = express();
const server: http.Server = http.createServer(app);
const PORT = 7878;
const routes: Array<CommonRoutesConfig> = [];
const debugLog: debug.IDebugger = debug('app')
```

Figura 5: Código configuración inicial backend

En esta sección, se crea una instancia de la aplicación Express y un servidor HTTP utilizando esa aplicación. Se define el puerto en el que el servidor escuchará las solicitudes entrantes. Se inicializa un arreglo para almacenar las configuraciones de rutas (objeto 'routes') y se crea una instancia de 'debug' para mensajes de depuración específicos de la aplicación.

4.3.1.4 Middleware y Configuración de Express

```
app.use(express.json());

// here we are adding middleware to allow cross-origin requests
app.use(cors());
app.use(function(req, res, next) {
  res.header('Access-Control-Allow-Origin', '*');
  res.header('Access-Control-Allow-Methods', 'GET,PUT,POST,DELETE');
  res.header('Access-Control-Allow-Headers', 'Content-Type, Authorization');
  next();
});
```

Figura 6: Código Middleware configuración express

Aquí, se configuran varios middlewares de Express. 'express.json()' permite que las solicitudes entrantes se analicen como JSON. 'cors()' habilita el manejo de solicitudes de origen cruzado. El middleware 'function(req, res, next)' agrega encabezados CORS adicionales para permitir diferentes métodos y tipos de encabezados.

4.3.1.5 Configuración del Logger con express-winston

```
// here we are preparing the expressWinston logging middleware configuration,
// which will automatically log all HTTP requests handled by Express.js
const loggerOptions: expressWinston.LoggerOptions = {
  transports: [new winston.transports.Console()],
  format: winston.format.combine(
    winston.format.json(),
    winston.format.prettyPrint(),
    winston.format.colorize({ all: true })
  ),
};

if (!process.env.DEBUG) {
  loggerOptions.meta = false; // when not debugging, log requests as one-liners
}

// initialize the logger with the above configuration
app.use(expressWinston.logger(loggerOptions));
```

Figura 7: Código configuración logger

Aquí, se configura el middleware *expressWinston.logger* para que registre automáticamente todas las solicitudes HTTP manejadas por Express. Se utiliza la librería *winston* para gestionar los registros y se definen las opciones de formato y transporte.

4.3.1.6 Configuración de Rutas

```
// here we are adding the UserRoutes to our array,
// after sending the Express.js application object to have the routes added to our app!
routes.push(new UsersRoutes(app));
routes.push(new ControlRoutes(app));
routes.push(new AuthRoutes(app));
routes.push(new MeasuresRoutes(app));

// this is a simple route to make sure everything is working properly
const runningMessage = `Server running at http://localhost:${PORT}`;

app.get('/', (req: express.Request, res: express.Response) => {
  res.status(200).send(runningMessage)
});
```

Figura 8: Código configuración de rutas

Se crean instancias de las clases que definen las rutas para diferentes recursos de la aplicación (usuarios, control de máquina, autenticación y medidas) y se agregan al arreglo 'routes'. Cada instancia de ruta recibe la aplicación Express como parámetro para agregar sus rutas y funcionalidades correspondientes.

4.3.1.7 Ruta de Prueba

```
// this is a simple route to make sure everything is working properly
const runningMessage = `Server running at http://localhost:${PORT}`;

app.get('/', (req: express.Request, res: express.Response) => {
  res.status(200).send(runningMessage)
});
```

Figura 9: Código configuración de rutas de test

Esta ruta se utiliza para verificar que el servidor está funcionando correctamente. Cuando se accede a la raíz ('/'), el servidor responde con un mensaje indicando que está en funcionamiento.

4.3.1.8 Inicio del Servidor

```
server.listen(PORT, () => {
  routes.forEach((route: CommonRoutesConfig) => {
    debugLog(`Routes configured for ${route.getName()}`);
  });
  // our only exception to avoiding console.log(), because we
  // always want to know when the server is done starting up
  console.log(runningMessage);
});
```

Figura 10: Código inicio de servidor express

Se inicia el servidor HTTP para escuchar en el puerto definido (7878). Antes de esto, se recorre el arreglo de rutas para imprimir en la consola qué rutas están configuradas. Finalmente, se muestra un mensaje en la consola indicando que el servidor está en funcionamiento.

Este diseño detallado abarca la estructura y el flujo de trabajo del código proporcionado, mostrando cómo se configuran las rutas, los middlewares y el servidor en general para crear una aplicación Express.js funcional.

4.3.2 Módulo FrontEnd React

4.3.2.1 Arquitectura de Software

El código comienza importando las bibliotecas y componentes necesarios:

```
import React from 'react'
import '../node_modules/bootstrap/dist/css/bootstrap.min.css'
import './styles/App.css'

import { AuthProvider } from "../context/index";
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom'
import history from "./history";
import { LoginPage } from './components/login.component'
import SignUp from './components/signup.component'
import Table from './components/table.component'
import EditAlgorithm from './components/editAlgorithm.component'
import AdminPanel from './Pages/adminPanel.component'
import NavBar from './components/NavBar.component'
import { useAuth } from "./Hooks";
import { PrivateRoute } from './Router/PrivateRoute'
import { NextPageFromLogin } from './Router/NextPageFromLogin'
import { MQTTProvider } from './context/MqttContext';
```

Figura 11: Código importación de módulos frontend

```
function App() {
  const { isAuthenticated } = useAuth(); // 'isAuthenticated' is assigned a value bu

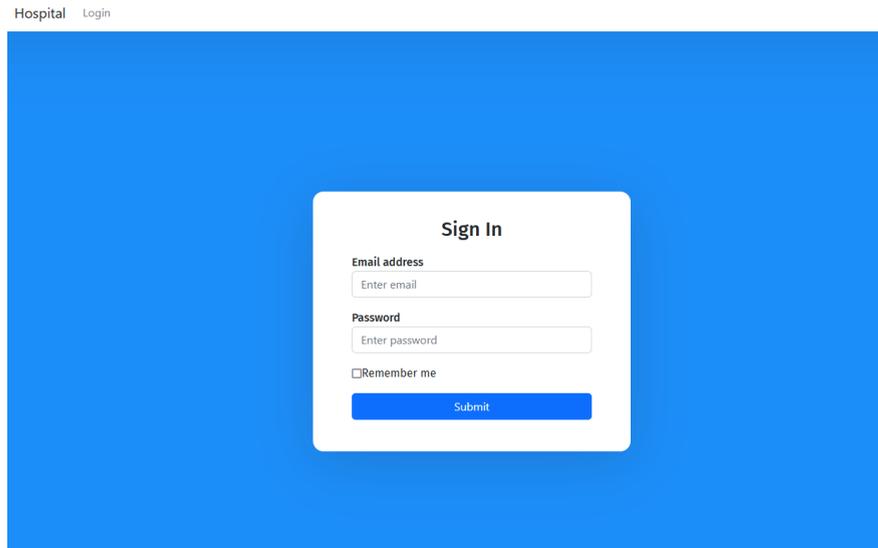
  return (
    <AuthProvider>
      <div className="App">
        <Router history={history}>
          <NavBar />
          <MQTTProvider>
            <Routes>
              <Route path="/sign-in" element={
                <NextPageFromLogin ><LoginPage /></NextPageFromLogin> /> />
              <Route path="/sign-up" element={
                <PrivateRoute ><SignUp /></PrivateRoute> /> />
              <Route path="/" element={
                <PrivateRoute >
                  <Table />
                </PrivateRoute> /> />
              <Route path="/algorithm" element={
                <PrivateRoute ><EditAlgorithm /></PrivateRoute> /> />
              <Route path="/admin" element={
                <PrivateRoute ><AdminPanel value={false} /></PrivateRoute> /> />
            </Routes>
          </MQTTProvider>
        </Router>
      </div>
    </AuthProvider>
  )
}

export default App
```

Figura 12: Código función principal App

La función **App** es el componente principal que renderiza la aplicación:

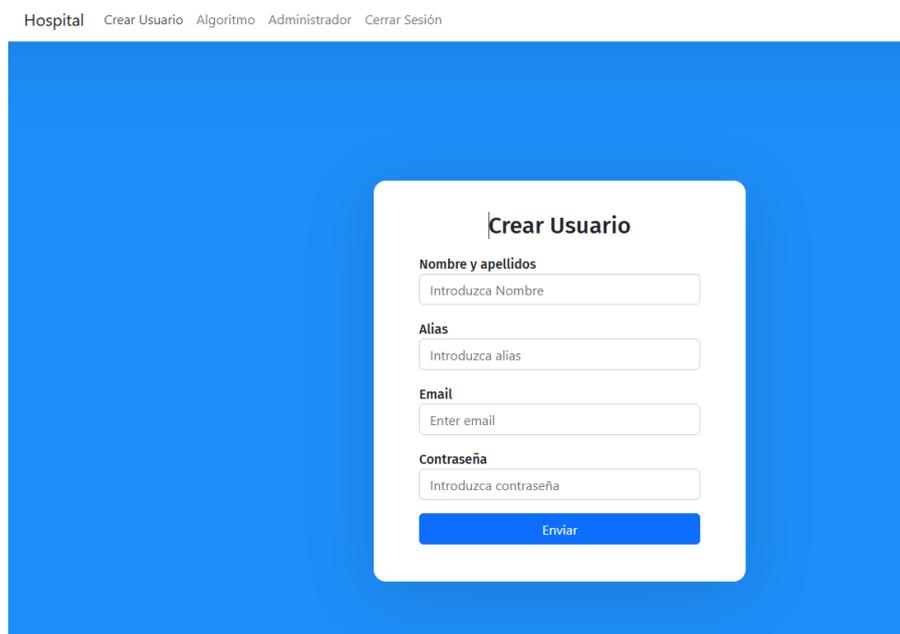
1. **AuthProvider**: Este componente provee el contexto de autenticación para la aplicación.
2. **BrowserRouter**: Se utiliza para gestionar el enrutamiento de la aplicación.
3. **NavBar**: Representa la barra de navegación superior de la aplicación.
4. **MQTTProvider**: Provee el contexto para la comunicación MQTT.
5. **Routes**: Define y engloba las rutas y los componentes correspondientes a renderizar para cada ruta.
6. **Rutas** (`/sign-in`, `/sign-up`, `/`, `/algorithm`, `/admin`): Cada una de estas rutas define qué componente se renderiza cuando el usuario accede a ellas. Se utilizan los componentes:
 - a. **LoginPage**: Página que renderiza un formulario de inicio de sesión.



The screenshot shows a web application interface with a blue background. At the top left, there are navigation links: "Hospital" and "Login". In the center, there is a white rounded rectangle titled "Sign In". Inside this rectangle, there are four input fields: "Email address" with the placeholder "Enter email", "Password" with the placeholder "Enter password", and a checkbox labeled "Remember me". Below these fields is a blue button labeled "Submit".

Figura 12: Interfaz de usuario inicio de sesión

- b. **SignUp**: Página que renderiza un formulario que permite crear nuevos usuarios.



The screenshot shows a web application interface with a blue background. At the top, there are navigation links: "Hospital", "Crear Usuario", "Algoritmo", "Administrador", and "Cerrar Sesión". In the center, there is a white rounded rectangle titled "Crear Usuario". Inside this rectangle, there are four input fields: "Nombre y apellidos" with the placeholder "Introduzca Nombre", "Alias" with the placeholder "Introduzca alias", "Email" with the placeholder "Enter email", and "Contraseña" with the placeholder "Introduzca contraseña". Below these fields is a blue button labeled "Enviar".

Figura 13: Interfaz de usuario creación de usuario

- c. Home (“/”) Página inicial de la interfaz, dedicada al control de la máquina e introducción de datos del paciente de las futuras muestras a obtener.

The screenshot shows the 'Control de Máquina' section with a form titled 'Introduzca datos de muestras'. It includes input fields for 'Nombre', 'Estado', 'Apellido', and 'DNI', and a dropdown menu for 'Número de medidas a analizar' set to '8'. A blue button labeled 'Iniciar Máquina' is positioned below the form. Below this is the 'Muestras' section, which features a search bar with the placeholder 'input search text' and a 'Search' button. Underneath is a table with columns for 'Name', 'Surname', 'State', 'DNI', and 'Data'. The table is currently empty, and a pagination control shows '10 / page'.

Figura 14: Interfaz de usuario página inicial

- d. EditAlgorithm: Página preparada para el modo de testeo del algoritmo que clasifica externamente las medidas, para su facilitar su calibración.

Hospital | Crear Usuario | Algoritmo | Administrador | Cerrar Sesión |

The screenshot displays the 'Editar Parámetros Algoritmo' section. It contains three input fields labeled 'VarA', 'VarB', and 'VarC', each with a placeholder value matching its label. A blue button labeled 'Cambio de variables' is located at the bottom of the form.

Figura 15: Interfaz de usuario editar algoritmo

e. AdminPanel: Panel de administración de usuarios.

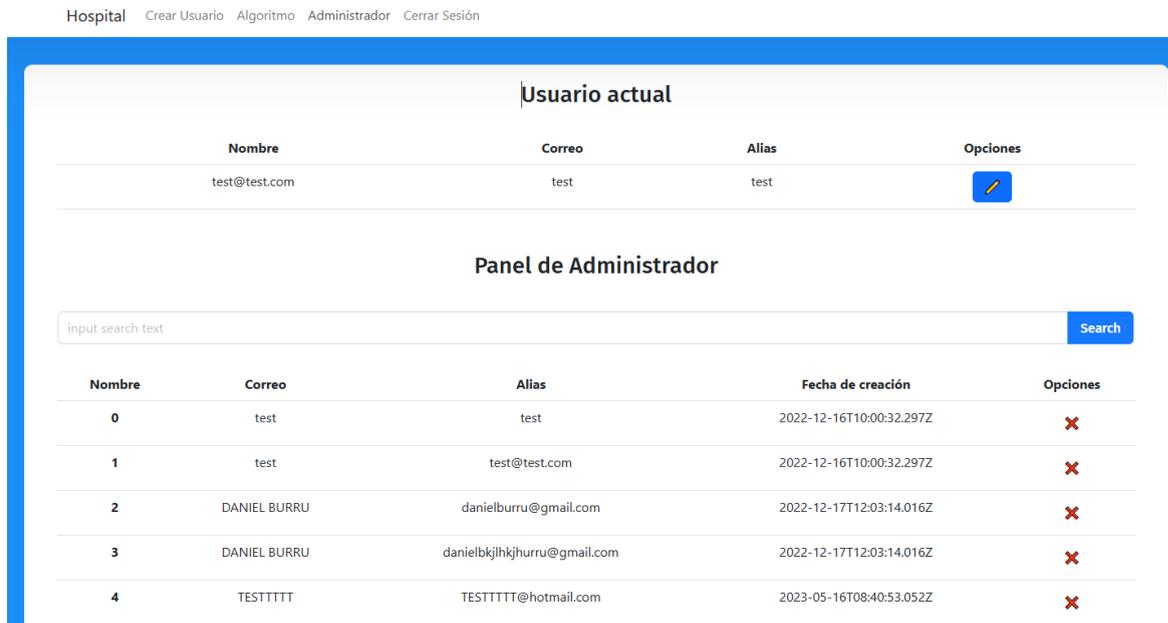


Figura 16: Interfaz de usuario panel administrador

7. PrivateRoute: Es un componente que controla el acceso a las rutas protegidas, asegurándose de que solo los usuarios autenticados puedan acceder a ellas.
8. NextPageFromLogin: Es un componente que redirige a la página siguiente después de que el usuario inicie sesión.

4.3.2.2 Resumen

La aplicación está estructurada para gestionar la autenticación de usuarios, el enrutamiento de diferentes páginas y la presentación de contenido dependiendo de la autenticación del usuario. Se utilizan contextos, rutas privadas y componentes de interfaz de usuario para lograr esto. Cada componente y biblioteca tiene un papel específico en la creación de una experiencia de usuario completa y segura.

4.3.3 Servicio Python

4.3.3.1 Importación de módulos

```
from MqttController import *
from mqtt_credentials import *
from hospiPython import *
```

Figura 17: Importación de módulos del servicio python

Se han importando varios módulos necesarios para el funcionamiento del programa. Los módulos importados parecen estar relacionados con la comunicación MQTT (MqttController), las

credenciales para la conexión MQTT (`mqtt_credentials`), y otro módulo llamado "hospiPython" para la comunicación e inicio de la impresora 3D.

4.3.3.2 Definición de la clase Service

En este bloque se están inicializando varias variables y objetos de inicio en el constructor de la clase Service:

```
class Service():
    def __init__(self):
        self.status = 0
        # Public Mqtt Server to test "Mqtt": Unknown word.
        self.mqttCredentials = MqttCredentials("broker.hivemq.com","", "",1883) "mqtt": Unknown word.
        # Local Mqtt Server "Mqtt": Unknown word.
        self.mqttCredentials = MqttCredentials("192.168.4.1","ysolve","ysolve2022",1883) "mqtt": Unknown word.
        self.mqttController = MqttConnection(self.mqttCredentials) "mqtt": Unknown word.
        self.mqttController.create_loop_mqtt_receive(self.processRequest)
        self.actualData = {
            "name": "not-provided",
            "surname": "not-provided",
            "state": "not-provided",
            "dni": "not-provided",
            "selectedOption": "not-provided"
        }
```

Figura 18: Clase principal del servicio python

1. `self.status`: Inicializado con el valor 0. Marca el valor inicial del estado del servicio.
2. `self.mqttCredentials`: Se crea una instancia de la clase *MqttCredentials* con diferentes parámetros. Se crea una instancia usando credenciales para un servidor público MQTT (`broker.hivemq.com`) y luego se sobrescribe con credenciales para un servidor MQTT local (`192.168.4.1`) ya que el broker público *hivemq* se utilizó anteriormente para pruebas.
3. `self.mqttController`: Se crea una instancia de la clase *MqttConnection* utilizando las credenciales MQTT definidas anteriormente. Luego se llama al método `create_loop_mqtt_receive` en `self.mqttController` pasando `self.processRequest` como argumento. Esta función llamada `processRequest` será la que ejecutará cada función dependiendo de qué petición sea recibida por el servicio.
4. `self.actualData`: Un diccionario que almacena información predeterminada sobre datos de la petición actual.

```
def processRequest(self,request):
    if(request["typeRequest"] == 0):
        ## Iniciar Machine "Iniciar": Unknown word.
        self.actualData = request["data"]
        startMachine(request["data"]["quantity"])
        print("startMachine")
```

Figura 19: Función de procesamiento de las peticiones MQTT del servicio python

Este bloque define el método `processRequest`, procesa las solicitudes recibidas a través de MQTT. Si el tipo de solicitud `typeRequest` es 0, actualiza los datos actuales con los datos proporcionados en la solicitud y llama a una función llamada `startMachine` con una cantidad específica de viales.

```

service = Service()

while (service.status == 0):
    pass

```

Figura 20: Bucle principal del servicio python

En este bloque se crea una instancia de la clase Service y se almacena en la variable service. Luego, hay un bucle que se ejecuta mientras el valor de service.status sea 0.

4.3.3.3 Función de inicio Impresora 3D

Esta función se encarga de abrir un archivo G-code, enviar sus comandos línea por línea a la impresora 3D a través de un puerto serie. Más tarde, cierra los recursos una vez que se hayan enviado todos los comandos.

```

def startMachine(type):
    port= "/dev/ttyUSB0"
    file = "/home/pi/Desktop/HospitalLaFe_FrontEnd/gcode_explain.txt"    "gcode": Unknown word.
    ## show values ##
    print( "USB Port: %s" % port )
    print( "Gcode file: %s" % file )    "Gcode": Unknown word.

    # Open serial port
    s = serial.Serial(port,115200)
    print( 'Opening Serial Port')

    # Open g-code file
    #f = open('/media/UNTITLED/shoulder.g','r');
    f = open(file,'r')
    print( ('Opening gcode file'))    "gcode": Unknown word.

    # Wake up
    s.write(str("\r\n\r\n").encode()) # Hit enter a few times to wake the print(robot    "rbot": Unknown word.
    time.sleep(2) # Wait for print(robot to initialize    "rbot": Unknown word.
    s.flushInput() # Flush startup text in serial input
    print( 'Sending gcode')    "gcode": Unknown word.

    # Stream g-code
    for line in f:
        l = removeComment(line)
        l = l.strip() # Strip all EOL characters for streaming
        if (l.isspace()==False and len(l)>0) :
            print( 'Sending: ' + l)
            s.write((str(l) + '\n').encode()) # Hit enter a few times to wake the print(robot    "rbot": Unknown word.
            grbl_out = s.readline() # Wait for response with carriage return    "grbl": Unknown word.
            print( ' : ' + str(grbl_out.strip()))    "grbl": Unknown word.

    # Wait here until printing is finished to close serial port and file.
    #raw_input(" Press <Enter> to exit.")

    # Close file and serial port
    f.close()
    s.close()

```

Figura 21: Función de inicio de comunicación con impresora 3D

4.3.3.4 Resumen

Este código está relacionado con la gestión de comunicación MQTT y procesamiento de solicitudes en nuestro servicio. Trata de gestionar las comunicaciones entre la interfaz web y la impresora 3D a través de un servicio y un gestor de peticiones.

Una vez recibida la petición necesaria para el inicio, crea una conexión serie con la impresora 3D y envía un documento con código G CODE predeterminado.

4.4 Tecnologías Utilizadas

4.4.1 GCODE

GCODE es un lenguaje de programación utilizado en la fabricación asistida por computadora (CAM) para controlar máquinas de herramientas CNC (Control Numérico Computarizado) como fresadoras, tornos y cortadoras láser.

El GCODE contiene una serie de instrucciones sincronas que indican a la máquina cómo mover sus ejes para realizar una tarea específica, como cortar o perforar materiales. Estas instrucciones incluyen movimientos lineales y circulares, cambios de herramienta, velocidades de movimiento, retrocesos y saltos mediante secuencias y coordenadas.

El GCODE se escribe en forma de texto, con cada línea que contiene una sola instrucción y un número de secuencia opcional. Las líneas se envían a la máquina herramienta a través de un puerto de comunicaciones o una unidad de almacenamiento de datos.

Personalmente descubrí este lenguaje, realizando automatizaciones de corte con cnc en la empresa Tecnosel en Xirivella, Valencia. Gracias a esto pude relacionar fácilmente esta tecnología con la implementación de nuestro problema.

4.4.2. Impresoras 3D Comunicación Serie

Las impresoras 3D son herramientas fundamentales en la actualidad para la fabricación de prototipos y piezas personalizadas. Una de las formas más comunes de comunicación entre una impresora 3D y un ordenador es a través de una conexión serie.

Esta comunicación serie se establece mediante un cable USB y utiliza el protocolo de comunicación RS-232 para enviar y recibir datos entre la impresora y el ordenador. Los datos se transmiten en formato binario y se organizan en paquetes de información que contienen las instrucciones de impresión para la máquina.

Es importante tener en cuenta que la velocidad de transmisión de datos en la comunicación serie puede afectar la calidad de impresión y la precisión de los detalles en la pieza final. Por esta razón, es esencial elegir una velocidad de transmisión adecuada para cada aplicación y asegurarse de que la conexión entre la impresora y el ordenador sea estable y confiable.

Personalmente monté una impresora 3D en el transcurso del 1º año de la Universidad, por lo que también estaba acostumbrado a esta tecnología.

4.4.3. React

La tecnología React se ha convertido en una herramienta esencial para el desarrollo de aplicaciones web modernas y de alta calidad en el ámbito del Frontend. Esta tecnología, creada por Facebook, se basa en la construcción de componentes reutilizables que se combinan para formar interfaces de usuario dinámicas y altamente interactivas.

Además, React utiliza una arquitectura de programación declarativa, lo que significa que los desarrolladores pueden centrarse en describir el comportamiento de la interfaz de usuario, en lugar de preocuparse por los detalles de cómo se implementa en el navegador.

Gracias a su amplia comunidad y ecosistema de bibliotecas y herramientas, React se ha convertido en una de las tecnologías más populares para el desarrollo Frontend, siendo ampliamente utilizada por empresas líderes en tecnología en todo el mundo.

4.4.4. API REST

REST (Representational State Transfer) es un estilo arquitectónico muy utilizado para desarrollar servicios web y API.

Las API que siguen este estilo se diseñan con el objetivo de ser simples, escalables y flexibles, y se utilizan comúnmente en aplicaciones web y móviles, así como en arquitecturas de microservicios y IoT. Por otro lado, Node.js es un entorno de tiempo de ejecución JavaScript back-end de código abierto y multiplataforma que permite a los desarrolladores ejecutar código JavaScript fuera de un navegador web.

Este entorno proporciona un modelo de E/S sin bloqueo y basado en eventos que lo hace eficiente y capaz de manejar grandes cantidades de datos. Además, cuenta con una comunidad activa y numerosas bibliotecas y módulos para ayudar a los desarrolladores a crear aplicaciones de manera más rápida y sencilla.

Express.js es un marco de aplicación web popular para Node.js que proporciona características y herramientas para construir servidores web, manejar solicitudes y respuestas HTTP, enrutar solicitudes a controladores específicos, manejar middleware y mucho más. Este marco se destaca por su simplicidad, flexibilidad y escalabilidad, lo que lo convierte en una opción preferida para los desarrolladores que crean aplicaciones web con Node.js.

Algunas de las características y beneficios clave de Express.js son:

- **Minimalista y flexible:** Express.js proporciona una estructura minimalista y flexible que permite a los desarrolladores crear aplicaciones de la forma que deseen.
- **Enrutamiento:** Express.js facilita la definición de rutas para manejar solicitudes HTTP y asignarlas a funciones o controladores específicos.
- **Middleware:** Express.js permite a los desarrolladores definir funciones de middleware que se pueden usar para manejar tareas comunes como autenticación, registro, manejo de errores y más.

API robusta: Express.js proporciona una API robusta para manejar solicitudes y respuestas HTTP, lo que permite a los desarrolladores crear aplicaciones web de alto rendimiento.

4.4.5. MongoDB

MongoDB es una base de datos NoSQL ampliamente utilizada que ofrece un enfoque flexible y escalable para el almacenamiento y recuperación de datos. A diferencia de las bases de datos relacionales, MongoDB no utiliza tablas y filas, sino que organiza los datos en documentos JSON BSON (formato binario JSON). Esto permite una estructura de datos más dinámica y adaptable a medida que los requisitos de la aplicación evolucionan.

En lugar de SQL, MongoDB utiliza un modelo de consulta basado en documentos. Los desarrolladores pueden interactuar con MongoDB a través de su lenguaje de consulta, que permite realizar consultas y actualizaciones utilizando estructuras de datos similares a los objetos que se utilizan en el código de la aplicación.

Algunas características clave de MongoDB incluyen la capacidad de manejar grandes volúmenes de datos, la capacidad de escalar horizontalmente añadiendo nodos adicionales, y una fuerte flexibilidad en cuanto a la estructura de los datos. MongoDB es especialmente adecuado para aplicaciones que manejan datos semi-estructurados o no estructurados, como datos en constante evolución o datos provenientes de sensores.

4.4.6. MQTT

MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería ligero y eficiente diseñado para la comunicación entre dispositivos en redes de baja capacidad y alta latencia, como Internet de las cosas (IoT). Fue desarrollado con el objetivo de permitir que los dispositivos se comuniquen de manera confiable y eficiente en entornos con recursos limitados.

En MQTT, los dispositivos se dividen en dos roles principales: el "publicador" (publisher), la plataforma web, y el "suscriptor" (subscriber), el servicio de python en la raspberry pi. Los publicadores envían mensajes a canales llamados "temas" (topics), y los suscriptores pueden suscribirse a estos temas para recibir los mensajes. Este modelo de publicación/suscripción permite una comunicación asíncrona y de bajo consumo de ancho de banda, ya que los dispositivos solo intercambian datos cuando es necesario.

CAPÍTULO 5

5.1 Desarrollo de la solución propuesta

El desarrollo de la solución propuesta en el proyecto TFG ha experimentado cambios significativos desde su primera versión hasta la solución final. Inicialmente, hace aproximadamente dos años la primera versión se desarrolló utilizando HTML y PHP para ejecutar un código Python encargado de ejecutar el GCODE. Sin embargo, esta implementación presentó problemas y dificultades técnicas, lo que llevó a buscar una nueva solución.



Figura 22: Primera versión del proyecto realizada con html y php

Se decidió desarrollar una nueva versión utilizando React como tecnología principal para el frontend. Esta elección permitió crear una interfaz web más dinámica y moderna, mejorando la experiencia del usuario. Además, se optó por utilizar MQTT como protocolo de comunicación local, lo cual facilitó la interacción con la parte de mecanizado.



Figura 23: Última versión del proyecto realizada con react, typescript y python.

En cuanto al backend, se decidió implementar una arquitectura basada en microservicios. Esto permitió dividir la lógica del sistema en componentes independientes y escalables, lo que facilitó su mantenimiento y futuras expansiones. Cada microservicio se encargó de gestionar una

funcionalidad específica, como la comunicación con la impresora 3D, la gestión de la base de datos, entre otros. Se adjunta la documentación del backend creada con [Postman](#).

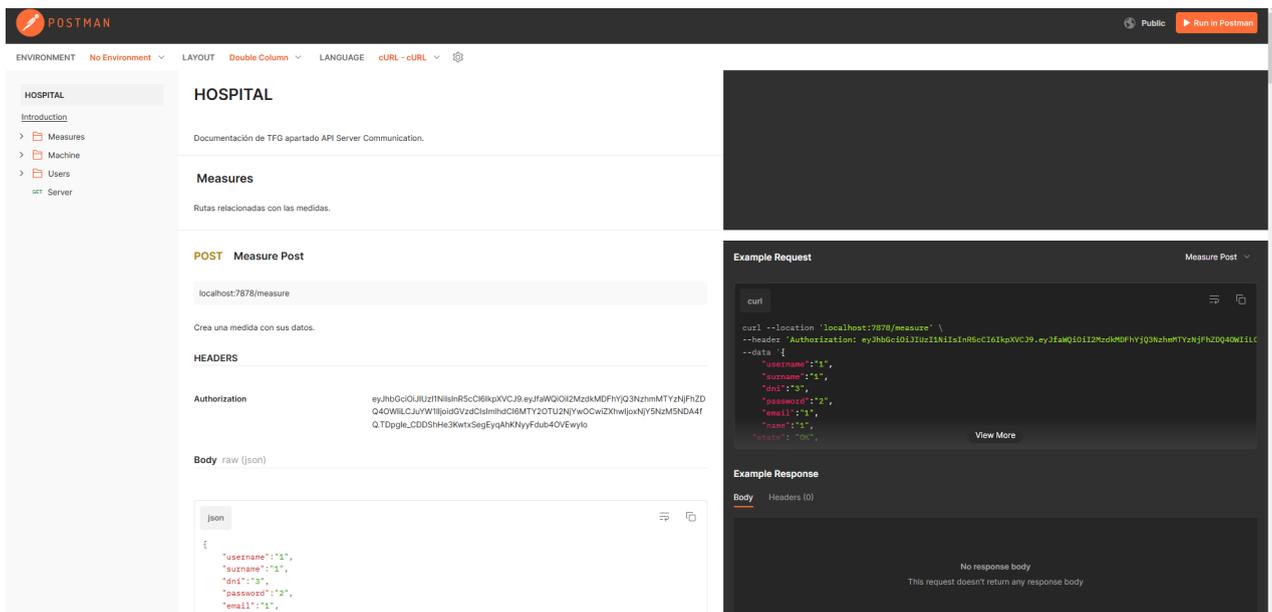


Figura 24: Documentación y pruebas realizadas con Postman

Durante el desarrollo de la solución final, se encontraron diversas dificultades y se tomaron decisiones importantes. Algunos de los problemas encontrados en la implementación inicial podrían haber estado relacionados con la comunicación entre los diferentes componentes, la gestión de la ejecución del código Python, o la escalabilidad y flexibilidad de la solución.

Con la nueva arquitectura basada en React y microservicios, se logró una solución más robusta y flexible. Se mejoró la comunicación local utilizando MQTT, lo que permitió una interacción más eficiente entre los diferentes elementos del sistema. Además, la implementación de microservicios facilitó la gestión y mantenimiento de cada componente de forma independiente.

CAPÍTULO 6

6. Implantación

En esta sección, se abordará la fase de implantación del proyecto, donde el desarrollo realizado se lleva a explotación para poner el sistema en funcionamiento. En esta etapa, se instala y pone en producción el sistema desarrollado, permitiendo la prueba y obtención de resultados de la solución final.

6.1 Preparación para la Implantación

Antes de la implantación, es fundamental preparar el entorno para asegurar una transición sin problemas desde el desarrollo hasta la explotación. Se deben tener en cuenta los siguientes pasos:

1. **Configuración de la Raspberry Pi:** Asegurarse de que la Raspberry Pi esté configurada correctamente como un punto de acceso WiFi y que tenga acceso a la red inalámbrica. Por seguridad se añade una contraseña WPA2 al punto de acceso.
2. **Servidor Web:** Configurar el servidor web en la Raspberry Pi para alojar la interfaz web y garantizar que el puerto específico para el acceso esté disponible y configurado correctamente.
3. **Conexión Serial:** Configurar la comunicación serial entre la Raspberry Pi y la impresora 3D para permitir el envío de comandos de control en forma de código GCODE.
4. **Instalación de MQTT:** Desplegar el servicio MQTT desarrollado en Python en la Raspberry Pi para habilitar la comunicación eficiente entre el cliente y otros dispositivos hardware, como sensores externos.
5. **Despliegue del Servicio de Autenticación API:** Instalar y configurar el servicio de autenticación API en el servidor web para garantizar la seguridad y el acceso controlado a la interfaz web y los datos almacenados.
6. **Base de Datos:** Configurar y asegurarse de que la base de datos local esté operativa en la Raspberry Pi, utilizando MongoDB para almacenar los datos recopilados durante el proceso de mecanización.
7. **Cron Jobs:** Configurar el inicio de los servicios necesarios, para su inicio conforme se encienda el proyecto.

Estos procesos detallados, solo serán necesarios realizarlos para este proyecto, posteriormente si se desea crear otro proyecto bastará con el archivo .ISO creado de la imagen preconfigurada.

6.2 Puesta en Producción

Una vez que el entorno esté preparado, se puede proceder con la puesta en producción del sistema desarrollado:

1. **Instalación del Sistema:** Cargar todos los archivos y componentes necesarios en la Raspberry Pi, asegurando que la estructura de directorios y las dependencias estén configuradas adecuadamente.
2. **Pruebas Iniciales:** Se han llevado a cabo pruebas exhaustivas para verificar que todos los componentes del sistema estén funcionando correctamente. Esto incluye probar la

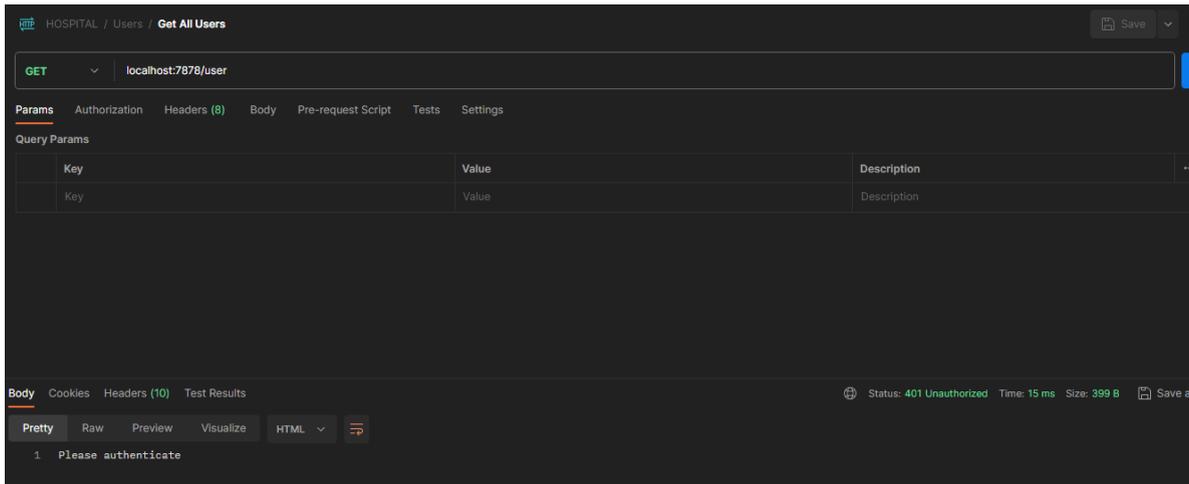


Figura 25: Pruebas realizadas de getUsers a través de postman en Backend

Una vez introducido el header: Authorization, con un token válido podemos obtener un código 200 con la respuesta correcta de la API.

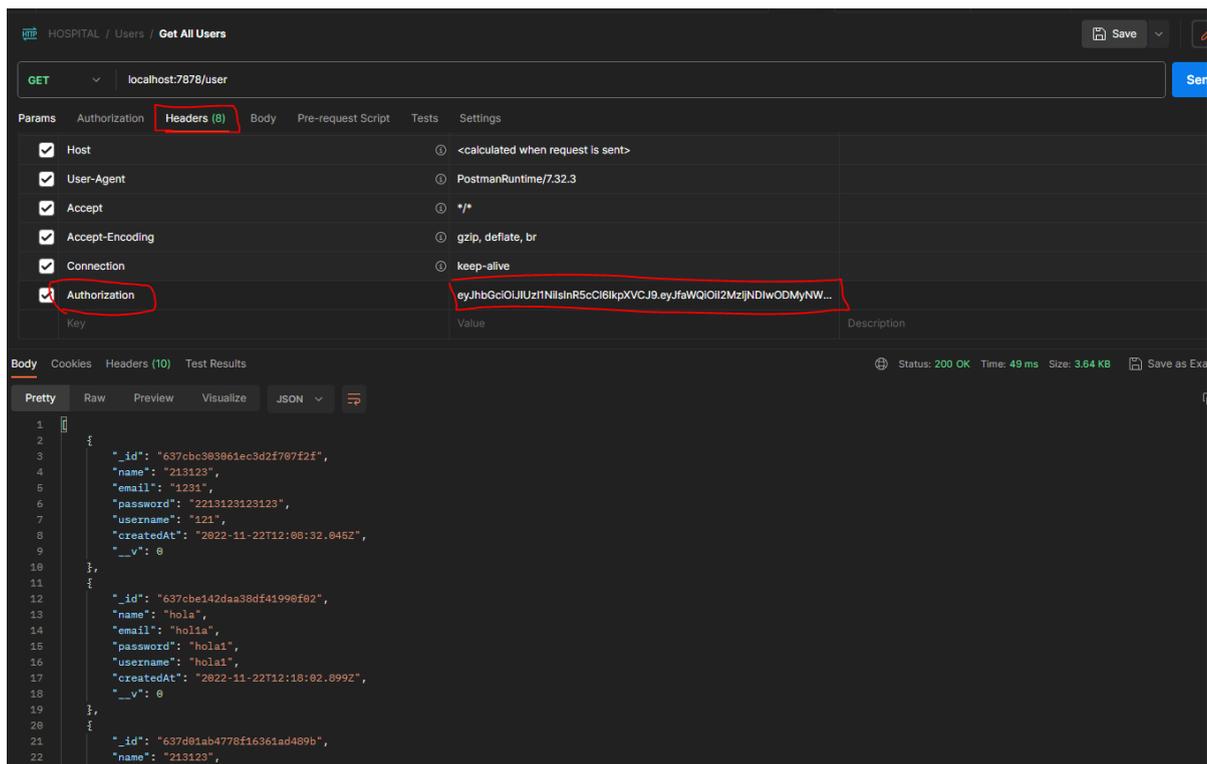


Figura 26: Pruebas realizadas de getAllUsers a través de postman en Backend

- Pruebas de almacenamiento:** Comprobar que los datos recopilados se almacenan correctamente en la base de datos local y pueden recuperarse de manera precisa.

Una de las pruebas de almacenamiento se han realizado manualmente creando nuevos usuarios, desde la ruta de creación de nuevos usuarios.

Figura 27: Pruebas realizadas de almacenamiento a través de la interfaz

Y desde la ruta de administración de usuarios se puede comprobar que se han creado satisfactoriamente y podemos recuperar los datos.

Hospital [Crear Usuario](#) [Algoritmo](#) [Administrador](#) [Cerrar Sesión](#)

Usuario actual

Nombre	Correo	Alias	Opciones
test@test.com	test	test	

Panel de Administrador

input search text Search

Nombre	Correo	Alias	Fecha de creación	Opciones
0	test	test	2022-12-16T10:00:32.297Z	
1	test	test@test.com	2022-12-16T10:00:32.297Z	
2	DANIEL BURRU	danielburru@gmail.com	2022-12-17T12:03:14.016Z	
3	DANIEL BURRU	danielbkjlkjhurr@gmail.com	2022-12-17T12:03:14.016Z	
4	TESTTTTT	TESTTTTT@hotmail.com	2023-05-16T08:40:53.052Z	
5	TEST_TFG	TEST_TFG@hotmail.com	2023-08-21T16:10:12.130Z	

Figura 28: Comprobando el registro en la base de datos de la aplicación

6.4 Puesta en Marcha y Mantenimiento Continuo

Una vez que el sistema ha pasado las pruebas y se ha validado, se considera listo para su puesta en marcha completa. Durante esta fase, es importante llevar a cabo un seguimiento constante y brindar mantenimiento para asegurarse de que el sistema siga funcionando sin problemas y se adapte a las necesidades cambiantes del usuario, para ello se ha realizado una copia de la tarjeta SD card preconfigurada, obteniendo así un archivo .ISO replicable para cualquier puesta en marcha.

La puesta en marcha bastará con tener una impresora 3D genérica, de las dimensiones necesarias dependiendo de la cantidad de viales a analizar, y una raspberry pi conectada con el archivo .ISO nombrado anteriormente.

Como mantenimiento continuo, será necesario mantener el hardware y realizar limpiezas continuas de la impresora y engrasar las barras deslizantes de la impresora 3D.

6.6 Resumen

La fase de implantación lleva el desarrollo del proyecto a la explotación, donde se instala y pone en producción el sistema desarrollado. A través de una cuidadosa preparación, pruebas exhaustivas y validación de resultados, se garantiza que el sistema funcione correctamente y cumpla con los objetivos establecidos. El mantenimiento continuo asegura que el sistema siga siendo eficiente y confiable a lo largo del tiempo.

CAPÍTULO 7

7. Pruebas

En esta sección, se detallarán las pruebas realizadas para verificar el correcto funcionamiento de la solución implementada. Se abordarán las pruebas de verificación, validación y, en su caso, pruebas de carga. Estas pruebas son esenciales para garantizar que el sistema cumple con los requisitos y expectativas establecidos, así como para evaluar su eficiencia y rendimiento.

7.1 Pruebas de Verificación

Las pruebas de verificación se centran en confirmar que cada componente individual del sistema funcione correctamente según lo diseñado. Esto incluye validar el funcionamiento de las características y la comunicación entre los diferentes elementos del sistema.

1. **Prueba del Punto de Acceso WiFi:** Verificación de que la Raspberry Pi actúa como un punto de acceso WiFi y permite que los clientes se conecten de manera exitosa. Para ello se ha iniciado la raspberry pi, y se ha comprobado que inicia un punto de acceso wifi llamado HosPiBerry.
2. **Prueba de la Interfaz Web:** Confirmación de que la interfaz web es accesible a través del puerto específico en la Raspberry Pi y que proporciona las funcionalidades necesarias para controlar y configurar la impresora 3D. Para ello desde un navegador se ha accedido al localhost y el puerto estipulado, accediendo así a la propia interfaz.
3. **Prueba de Comunicación Serial:** Verificación de que la interfaz web puede enviar comandos de control en forma de código G CODE a través de la comunicación serial para dirigir la impresora 3D. Para ello se ha utilizado una conexión a través de putty, un gestor de conexiones, y se ha enviado varios comandos de GCODE para verificar el movimiento de la impresora 3D.
4. **Prueba del Servicio MQTT:** Validación de que el servicio MQTT permite una comunicación efectiva entre el cliente y otros dispositivos hardware, como sensores externos. Para ello se ha utilizado MQTT Fx un gestor MQTT, desde el cual se puede suscribir o publicar en un tópico específico.
5. **Prueba del Servicio de Autenticación API:** Confirmación de que el servicio de autenticación API gestiona de manera adecuada la seguridad y el acceso controlado a la interfaz web y los datos almacenados. Para ello se ha utilizado Postman un gestor de peticiones HTTP, realizando peticiones y verificando los resultados devueltos. Un dato relevante es la información del usuario entregada en el JWT, el cual se ha confirmado a través de la aplicación jwt.io.
6. **Prueba de Almacenamiento de Datos:** Verificación de que los datos recolectados durante el proceso se almacenan correctamente en la base de datos local. Para ello se ha procedido a crear usuarios a través de la propia plataforma y verificar su almacenamiento y visualización en ella misma.

7.2 Resultados y Conclusión de las Pruebas

Las pruebas se han centrado en varios apartados separados, una vez concluidas estas pruebas individuales, se ha procedido a realizar pruebas conjuntas.

Las pruebas realizadas en este proceso de validación han arrojado resultados satisfactorios en general.

En primer lugar, la prueba del Punto de Acceso WiFi confirmó que la Raspberry Pi actúa como un punto de acceso WiFi exitosamente, permitiendo que los clientes se conecten sin problemas al punto de acceso denominado "HosPiBerry" con contraseña WPA2 HosPiBerry2023. Es accesible

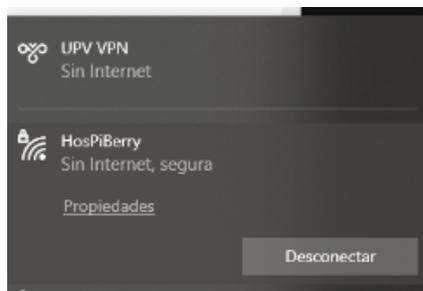


Figura 29: Comprobando la conexión wifi con la raspberry

En cuanto a la Prueba de la Interfaz Web, se ha verificado que la interfaz web es accesible a través del puerto especificado en la Raspberry Pi, ip 192.168.4.1:3000, la interfaz web., proporcionando todas las funcionalidades necesarias para controlar y configurar la impresora 3D, y manejo de usuarios, autenticación, etc.



Figura 30: Comprobando la accesibilidad de la interfaz

La Prueba de Comunicación Serial demostró que el servicio python es capaz de enviar comandos de control en forma de código G CODE a través de la comunicación serial para dirigir la impresora 3D. Esta capacidad es esencial para la operación precisa de la impresora.

En relación con la Prueba del Servicio MQTT, se ha validado que el servicio permite una comunicación efectiva entre el cliente y los dispositivos hardware externos como la impresora 3d a través del servicio python.

CAPÍTULO 8

8.1 Conclusiones

El proyecto de mecanizado a través de la impresora 3D ha sido satisfactoriamente desarrollado, alcanzando los objetivos planteados inicialmente. A lo largo de este Trabajo de Fin de Grado, se ha demostrado el dominio de diferentes tecnologías y su integración para resolver el problema planteado en la introducción de la memoria.

Durante el desarrollo del proyecto, se han enfrentado diversos desafíos y se han encontrado problemas técnicos, pero se han logrado superar mediante soluciones adecuadas. La integración de la Raspberry Pi, el control mediante código G CODE y la comunicación con el hardware de la impresora 3D han sido aspectos clave que se han abordado de manera efectiva.

Se ha aprendido a utilizar tecnologías de vanguardia como el Internet de las Cosas (IoT), la comunicación MQTT y el desarrollo de aplicaciones web utilizando React y Node.js. Estas tecnologías no fueron ampliamente abordadas durante la carrera, por lo que se ha requerido un esfuerzo adicional para adquirir los conocimientos necesarios.

Durante el desarrollo del proyecto, se han identificado errores y dificultades, pero se han utilizado estrategias de resolución de problemas para superarlos. En retrospectiva, algunos errores podrían haberse evitado con una planificación más detallada y una mayor atención a los detalles. Estos errores han sido aprendizajes valiosos que contribuirán al crecimiento profesional.

A nivel personal, este proyecto ha supuesto un importante crecimiento en el manejo de nuevas tecnologías, la capacidad de integrar conocimientos de diferentes disciplinas. Se ha fortalecido la capacidad de investigar, planificar y desarrollar soluciones tecnológicas de manera autónoma.

En cuanto a nuevos conocimientos y tecnologías adquiridos, ha sido necesario aprender sobre el control de impresoras 3D, la comunicación serial mediante código GCODE, el desarrollo de servicios MQTT, la implementación de una interfaz de usuario con React y Node.js, y la gestión de una base de datos local con MongoDB . Estos conocimientos han sido fundamentales para el éxito del proyecto y han permitido alcanzar un nivel de dominio adecuado en estas tecnologías.

Este proyecto combina diversas tecnologías aprendidas durante el grado de Tecnologías Interactivas, como IoT, programación web, electrónica y microcontroladores, protocolos de comunicación aplicados al proyecto.

El desarrollo del proyecto involucra actividades como investigación, diseño, programación, pruebas y documentación.

Como estudiante de Tecnologías Interactivas, he aplicado conocimientos y habilidades adquiridas en diversas áreas tecnológicas para enfrentar este desafío.

En resumen, el proyecto de mecanizado a través de la impresora 3D ha sido una experiencia enriquecedora que ha permitido aplicar los conocimientos adquiridos durante la carrera y explorar nuevas tecnologías. Se han logrado los objetivos planteados inicialmente y se ha demostrado un dominio adecuado en la resolución del problema planteado.

CAPÍTULO 9

9. Trabajos futuros

El proyecto de mecanizado y automatización de la recogida de muestras biológicas ofrece diversas oportunidades para futuras mejoras y ampliaciones. Algunas de las posibles áreas de trabajo futuro incluyen:

1. Envío de muestras a través de la API:

Actualmente, el proyecto permite el control y configuración de la impresora 3D para realizar el mecanizado de las muestras. Un trabajo futuro podría enfocarse en ampliar la funcionalidad del sistema para permitir el envío de muestras biológicas a través de la API. Esto podría implicar la integración de dispositivos de transporte o sistemas de manipulación adicionales que permitan la carga y descarga automatizada de las muestras en la impresora 3D.

2. Comunicación mediante MQTT o API HTTP:

En el proyecto actual, se ha utilizado el protocolo MQTT para la comunicación con dispositivos externos, como sensores, en un futuro. Esto brindaría flexibilidad para interactuar con dispositivos y sistemas externos sin depender exclusivamente de la API HTTP.

3. Sensorización de temperatura u otras medidas:

Una posible mejora sería la incorporación de sensores adicionales, como sensores de temperatura, para monitorear las condiciones durante el proceso de recogida de muestras biológicas. Esto permitiría obtener datos más completos y precisos sobre las condiciones ambientales y realizar un seguimiento más detallado de los parámetros relevantes para el análisis de las muestras.

4. Optimización del rendimiento y la eficiencia:

Se podrían realizar trabajos futuros para optimizar el rendimiento y la eficiencia del sistema. Esto podría involucrar mejoras en el tiempo de respuesta de la interfaz web, la reducción del consumo de energía de los dispositivos o la optimización de los algoritmos utilizados para el control de la impresora 3D. Estas mejoras podrían aumentar la productividad y la eficiencia del proceso de mecanizado de las muestras biológicas.

5. Integración con sistemas de análisis de datos:

Otra posibilidad sería la integración del sistema con plataformas o sistemas de análisis de datos, o edición de algoritmos de análisis. Esto permitiría una mayor automatización y análisis de los resultados obtenidos durante el proceso de mecanizado. Los datos recolectados podrían ser procesados automáticamente y utilizados para generar informes, realizar análisis estadísticos o incluso para entrenar modelos de aprendizaje automático.

CAPÍTULO 10

10. Referencias

1. GitHub - mqttjs/MQTT.js: The MQTT client for Node.js and the browser. (s.f.). GitHub. <https://github.com/mqttjs/MQTT.js#readme>
2. Short introduction — pySerial 3.4 documentation. (s.f.). Welcome to pySerial's documentation — pySerial 3.4 documentation. <https://pyserial.readthedocs.io/en/latest/shortintro.html>
3. Using TypeScript with MongoDB (s.f.) Medium <https://medium.com/swlh/using-typescript-with-mongodb-393caf7adfef>
4. Tu propio punto de acceso WiFi con Raspberry Pi. (s.f.). Hipertextual. <https://hipertextual.com/2020/07/punto-acceso-wifi-raspberry-pi>
5. Install Mosquitto Broker Raspberry Pi | Random Nerd Tutorials. (s.f.). Random Nerd Tutorials. <https://randomnerdtutorials.com/how-to-install-mosquitto-broker-on-raspberry-pi/>
6. ¿Qué es el GCODE? Transformando tus modelos 3D al lenguaje de las impresoras 3D - 3Dfils. (s.f.). 3Dfils |Filamentos para impresora 3D. <https://www.3dfils.com/es/blog/entrada/que-es-el-gcode.html>
7. Qué es la impresión 3D, importancia y cómo funciona la fabricación aditiva. (s.f.). aula21 | Formación para la Industria. <https://www.cursosaula21.com/que-es-la-impresion-3d/>
8. Magaña, L. M. L. (2020, 17 de enero). Qué es Json Web Token y cómo funciona. OpenWebinars.net. <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>
9. *Diseño de robot para toma de sangre y rotulado de muestras en hospitales* (Publicación n.º 13763) [Tesis Ingeniero Mecatrónico, Pontificia Universidad Católica del Perú]. (2019). tesis.pucp.edu.pe. <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/13763> **Cita en el texto:**(*Diseño de robot para toma de sangre y rotulado de muestras en hospitales*, 2019)
10. Ken Schwaber. (2020). La guía scrum: La guía definitiva de scrum: Las reglas del juego Noviembre 2020) [\[Pdf\]](#). **Cita en el texto.**

ANEXOS

DICCIONARIO:

- Muestras biológicas: Son materiales biológicos, como sangre, orina, saliva, tejidos, etc., que se utilizan para realizar análisis y diagnósticos médicos o científicos.
- Impresión 3D: Es un proceso de fabricación aditiva que crea objetos tridimensionales a partir de un modelo digital.
- Interfaz web: Es una interfaz de usuario que se accede a través de un navegador web y que permite interactuar con una aplicación o sistema.
- ODS: Objetivos de Desarrollo Sostenible, son un conjunto de objetivos establecidos por las Naciones Unidas para lograr un futuro sostenible para todos.
- MQTT: Protocolo de mensajería ligero y de publicación/suscripción utilizado para la comunicación entre dispositivos en redes de sensores y sistemas embebidos
- Mecanización: Proceso de automatizar una tarea o proceso mediante el uso de máquinas o dispositivos mecánicos.
- Automatización: Proceso de hacer que un sistema o proceso funcione automáticamente, sin la necesidad de intervención humana.
- Código G CODE: Lenguaje de programación utilizado para controlar máquinas CNC, como impresoras 3D, mediante la definición de movimientos y acciones específicas.
- Raspberry Pi: Ordenador de placa reducida (SBC) utilizado para diversos proyectos de electrónica y programación.
- Interfaz visual: Interfaz de usuario que permite interactuar con una aplicación o sistema de forma visual.
- Backend: Parte de un sistema informático que se encarga del procesamiento y almacenamiento de datos.
- Frontend: Parte de un sistema informático que se encarga de la presentación y la interacción con el usuario.
- Node.js: Entorno de tiempo de ejecución de JavaScript utilizado para construir aplicaciones de red escalables.
- MongoDB : Sistema de gestión de bases de datos no relacional de código abierto.
- Autenticación de usuario JWT: Método de autenticación de usuario que utiliza tokens JSON web (JWT) para verificar la identidad del usuario.
- API: Interfaz de programación de aplicaciones que permite la comunicación entre diferentes aplicaciones o sistemas.
- LAN: Red de área local, que conecta dispositivos en un área geográfica limitada, como una casa o una oficina.
- FTDI: Conversor USB a serie utilizado para la comunicación entre la Raspberry Pi y la impresora 3D.
- React: Biblioteca de JavaScript utilizada para construir interfaces de usuario interactivas.
- Python: Lenguaje de programación de alto nivel utilizado para el desarrollo de aplicaciones web, científicas y de automatización.
- Base de datos: Sistema de almacenamiento de datos estructurados que permite la gestión y manipulación eficiente de la información.
- Servidor: Computadora o sistema informático que proporciona servicios o recursos a otros dispositivos o usuarios en una red.
- Control de acceso: Proceso de autenticación y autorización que permite o deniega el acceso a recursos o funcionalidades específicas en un sistema.
- Seguridad de datos: Conjunto de medidas y prácticas utilizadas para proteger la confidencialidad, integridad y disponibilidad de los datos almacenados y transmitidos en un sistema.
- Desarrollo web: Proceso de creación y mantenimiento de sitios web y aplicaciones web utilizando tecnologías y lenguajes de programación específicos.
- API REST: Arquitectura de diseño de software que permite la comunicación entre sistemas a través de solicitudes HTTP utilizando los métodos GET, POST, PUT y DELETE.

- Git: Sistema de control de versiones distribuido utilizado para el seguimiento de cambios en el código fuente de un proyecto.
- Testing: Proceso de verificación y validación de un software para garantizar su correcto funcionamiento y detectar posibles errores o fallos.
- Documentación: Conjunto de documentos o recursos que describen el funcionamiento, uso y mantenimiento de un software o sistema.
- Framework: Conjunto de herramientas, bibliotecas y estándares que proporcionan una estructura y funcionalidades comunes para el desarrollo de aplicaciones..
- Scrum: Marco de trabajo ágil utilizado en la gestión de proyectos de desarrollo de software, que se basa en iteraciones cortas y en la colaboración entre el equipo de desarrollo y el cliente.
- Metodología ágil: Enfoque de desarrollo de software que se basa en la adaptabilidad, la colaboración y la entrega incremental, en contraposición a los enfoques tradicionales y más rígidos.
- Debugging: Proceso de identificación y corrección de errores o fallos en un programa o sistema informático.
- Repositorio: Almacenamiento centralizado de archivos y versiones de código fuente, utilizado en sistemas de control de versiones como Git.
- Control de versiones: Sistema que permite el seguimiento de cambios en el código fuente de un proyecto, facilitando la colaboración y la gestión de versiones.
- Despliegue: Proceso de implementación y puesta en funcionamiento de una aplicación o sistema en un entorno de producción.
- Depuración: Proceso de identificación y corrección de errores o fallos en un programa o sistema informático.
- Lenguaje de programación: Lenguaje formal utilizado para escribir instrucciones que se pueden ejecutar por un ordenador o sistema informático.
- Sintaxis: Conjunto de reglas y estructuras gramaticales que definen la forma correcta de escribir instrucciones en un lenguaje de programación.
- Semántica: Significado y comportamiento de las instrucciones escritas en un lenguaje de programación.
- Variable: Espacio de memoria reservado para almacenar un valor o dato específico durante la ejecución de un programa.
- Función: Bloque de código que realiza una tarea específica y puede ser llamado y reutilizado en diferentes partes de un programa.
- Objeto: Instancia de una clase que contiene datos y métodos específicos.
- Testing: Proceso de verificación y validación de un software para garantizar su correcto funcionamiento y detectar posibles errores o fallos.
- Despliegue: Proceso de implementación y puesta en funcionamiento de una aplicación o sistema en un entorno de producción.
- Mantenimiento: Actividad de corrección, mejora y actualización de un software o sistema para garantizar su correcto funcionamiento y adaptación a los cambios y requerimientos.
- Control de versiones: Sistema que permite el seguimiento de cambios en el código fuente de un proyecto, facilitando la colaboración y la gestión de versiones.
- Repositorio: Almacenamiento centralizado de archivos y versiones de código fuente, utilizado en sistemas de control de versiones como Git.
- Integración continua: Práctica de desarrollo de software que consiste en realizar integraciones y pruebas automáticas de forma regular y frecuente para detectar problemas de manera temprana.
- Scrum: Marco de trabajo ágil utilizado en la gestión de proyectos de desarrollo de software, que se basa en iteraciones cortas y en la colaboración.

ANEXO I:

Se adjunta el código en su totalidad de todo el proyecto.

ANEXO II Objetivos de Desarrollo Sostenible

Relación con los Objetivos de Desarrollo Sostenible (ODS):

Este proyecto también se alinea con los Objetivos de Desarrollo Sostenible (ODS) establecidos por las Naciones Unidas. En particular, se relaciona con los siguientes ODS:

1. Salud y bienestar (ODS 3): Al mejorar la eficiencia y precisión en la obtención de muestras biológicas, este proyecto contribuye a mejorar la salud y el bienestar de las personas al facilitar un diagnóstico y tratamiento más precisos y oportunos.

2. Innovación e infraestructuras (ODS 9): La mecanización y automatización de la obtención de muestras biológicas representan una innovación tecnológica que mejora las infraestructuras y los procesos existentes en el campo de la biotecnología.

3. Alianzas para lograr los objetivos (ODS 17): Este proyecto fomenta la colaboración entre diferentes actores, como profesionales de la salud, investigadores y desarrolladores de tecnología, para lograr un objetivo común: mejorar la obtención y análisis de muestras biológicas.

En resumen, este proyecto se alinea con los Objetivos de Desarrollo Sostenible de la ONU. En particular: Salud y bienestar, Innovación e infraestructuras y alianzas para lograr los objetivos.