



**ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ
MANUEL FÉLIX LÓPEZ**

CARRERA DE COMPUTACIÓN

**INFORME DE TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A
LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN CIENCIAS DE LA
COMPUTACIÓN**

**MECANISMO: SISTEMATIZACIÓN DE EXPERIENCIAS PRÁCTICAS
DE INVESTIGACIÓN Y/O INTERVENCIÓN**

TEMA:

**SISTEMA DE INVENTARIO BASADO EN TECNOLOGÍA DE
IDENTIFICACIÓN POR RADIOFRECUENCIA (RFID) EN LA
CARRERA DE COMPUTACIÓN DE LA ESPAM MFL**

AUTORES:

**CHRISTIAN ISRAEL LÓPEZ MORA
JONATHAN GEOVANNY ORMAZA CALDERÓN**

TUTOR:

ING. YIMMY SALVADOR LOOR VERA

CALCETA, JULIO DE 2023

DECLARACIÓN DE AUTORÍA

Nosotros **LÓPEZ MORA CHRISTIAN ISRAEL** y **ORMAZA CALDERÓN JONATHAN GEOVANNY**, con cédulas de ciudadanía 1315223394 y 1315572709 respectivamente, declaramos bajo juramento que el Trabajo de Integración Curricular titulado: “**SISTEMA DE INVENTARIO BASADO EN TECNOLOGÍA DE IDENTIFICACIÓN POR RADIOFRECUENCIA (RFID) EN LA CARRERA DE COMPUTACIÓN DE LA ESPAM MFL**” es de nuestra autoría, que no ha sido previamente presentado para ningún grado o calificación profesional, y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, concedemos a favor de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos, conservando a nuestro favor todos los derechos patrimoniales de autor sobre la obra, en conformidad con el Artículo 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación.

CHRISTIAN I. LÓPEZ MORA

CC: 115223394

JONATHAN G. ORMAZA CALDERÓN

CC: 1315572709

AUTORIZACIÓN DE PUBLICACIÓN

LÓPEZ MORA CHRISTIAN ISRAEL y **ORMAZA CALDERÓN JONATHAN GEOVANNY**, con cédulas de ciudadanía 1315223394 y 1315572709 respectivamente, autorizamos a la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, la publicación en la biblioteca del Trabajo de Integración Curricular titulado: **“SISTEMA DE INVENTARIO BASADO EN TECNOLOGÍA DE IDENTIFICACIÓN POR RADIOFRECUENCIA (RFID) EN LA CARRERA DE COMPUTACIÓN DE LA ESPAM MFL”**, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y total autoría.



CHRISTIAN I. LÓPEZ MORA

CC: 115223394



JONATHAN G. ORMAZA CALDERÓN

CC: 1315572709

CERTIFICACIÓN DEL TUTOR

MGTR. YIMMY SALVADOR LOOR VERA, certifica haber tutelado el Trabajo de Integración Curricular titulado: “**SISTEMA DE INVENTARIO BASADO EN TECNOLOGÍA DE IDENTIFICACIÓN POR RADIOFRECUENCIA (RFID) EN LA CARRERA DE COMPUTACIÓN DE LA ESPAM MFL**”, que ha sido desarrollada por **LÓPEZ MORA CHRISTIAN ISRAEL** y **ORMAZA CALDERÓN JONATHAN GEOVANNY**, previo a la obtención del título de INGENIERO EN CIENCIAS DE LA COMPUTACIÓN de acuerdo al **REGLAMENTO DE LA UNIDAD DE INTEGRACIÓN CURRICULAR DE CARRERAS DE GRADO** de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

ING. YIMMY SALVADOR LOOR VERA

CC: 1311710709

TUTOR

APROBACIÓN DEL TRIBUNAL

Los suscritos integrantes del Tribunal correspondiente, declaramos que hemos **APROBADO** el Trabajo de Integración Curricular titulado: “**SISTEMA DE INVENTARIO BASADO EN TECNOLOGÍA DE IDENTIFICACIÓN POR RADIOFRECUENCIA (RFID) EN LA CARRERA DE COMPUTACIÓN DE LA ESPAM MFL**”, que ha sido desarrollado por **LÓPEZ MORA CHRISTIAN ISRAEL** y **ORMAZA CALDERÓN JONATHAN GEOVANNY**, previo a la obtención del título de INGENIERO EN CIENCIAS DE LA COMPUTACIÓN, de acuerdo al **REGLAMENTO DE LA UNIDAD DE INTEGRACIÓN CURRICULAR DE CARRERAS DE GRADO** de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

MGTR. DANIEL A. MERA MARTINEZ

CC: 1301932156

PRESIDENTE DEL TRIBUNAL

MGTR. FERNANDO R. MOREIRA MOREIRA

CC: 1311726689

MIEMBRO DEL TRIBUNAL

MGTR. RICARDO A. VÉLEZ VALAREZO

CC: 1306391614

MIEMBRO DEL TRIBUNAL

AGRADECIMIENTO

A la prestigiosa institución Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López “ESPAM MFL”, la cual nos brindó la oportunidad de seguir adquiriendo conocimientos en todos estos años.

A cada uno de los docentes de la carrera de Computación que compartieron sus conocimientos en las diferentes aulas de clases y nos ayudaron a formarnos de excelente manera en el área de ciencias de la computación.

LOS AUTORES

DEDICATORIA

Todo lo logrado en estos años se lo dedico a mis padres Hugo Ormaza y Gladys Calderón, por su apoyo incondicional cada día, a mis hermanos Jairo Ormaza, Sandra Ormaza y Erika Ormaza por el apoyo que me brindaron en el transcurso de mi carrera universitaria y a mi esposa Maybe Moreira por todas esas palabras de motivación que me ayudaron a ser la persona que soy hoy en día.

Jonathan Geovanny Ormaza Calderón

DEDICATORIA

Le dedico este trabajo a mi madre pilar fundamental de mi vida a mi tía la cual es mi segunda madre y a mis hermanos los cuales siempre me han apoyado en todo este camino.

Christian Israel López Mora

CONTENIDO GENERAL

DECLARACIÓN DE AUTORÍA	ii
AUTORIZACIÓN DE PUBLICACIÓN	iii
CERTIFICACIÓN DEL TUTOR	iv
APROBACIÓN DEL TRIBUNAL.....	v
AGRADECIMIENTO	vi
DEDICATORIA	vii
DEDICATORIA	viii
CONTENIDO GENERAL.....	ix
CONTENIDO DE TABLAS	x
CONTENIDO DE FIGURAS.....	x
RESUMEN	xi
ABSTRACT.....	xii
1. CAPÍTULO I ANTECEDENTES.....	1
1.1 DESCRIPCIÓN DE LA INSTITUCIÓN	1
1.2 DESCRIPCIÓN DE LA INTERVENCIÓN	2
1.3 OBJETIVOS.....	4
1.3.1 OBJETIVO GENERAL	4
1.3.2 OBJETIVOS ESPECÍFICOS	4
2. CAPÍTULO II. DESARROLLO METODOLÓGICO DE LA INTERVENCIÓN	5
2.1 PLANIFICACIÓN.....	5
2.3 DESARROLLO	6
2.4 PRUEBAS.....	7
3. CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA.....	8
3.1 PLANIFICACIÓN.....	8
3.2 DISEÑO	10
3.3 DESARROLLO	14
3.4 PRUEBAS.....	21
4. CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES.....	24
4.1 CONCLUSIONES	24
4.2 RECOMENDACIONES	25
BIBLIOGRAFÍA	27
ANEXOS	32
ANEXO 1. ELICITACIÓN DE REQUERIMIENTOS	33

ANEXO 2. REQUISITOS DE DISEÑO DEL SISTEMA.....	37
1. INTRODUCCIÓN.....	58
2. OBJETIVO.....	58
3. REQUERIMIENTOS PARA IMPLEMTACIÓN.....	59
4. REQUERIMIENTOS PARA DESARROLLO.....	59
5. INSTALACIÓN Y CONFIGURACIÓN.....	59
6. DESARROLLO DEL MANUAL.....	60
1. OBJETIVO.....	80
2. DEFINICIONES.....	80
3. DESARROLLO DEL MANUAL.....	80

CONTENIDO DE TABLAS

Tabla 1 Lista de requerimientos.....	8
Tabla 2 Resultados de pruebas unitarias.....	27

CONTENIDO DE FIGURAS

Figura 1 Acta de entrevista.....	7
Figura 2 Vista menú principal GitMind.....	11
Figura 3 Diagrama de clases.....	11
Figura 4 Diagrama de arquitectura.....	11
Figura 5 Diagrama Físico de comunicación.....	11
Figura 6 Diagrama Base de datos.....	11
Figura 7 Modelo – Vista - Controlador.....	11
Figura 8 "Paquete Entity Framework.....	15
Figura 9 Mapeo de Base de Datos.....	16
Figura 10 Mapeo de Base de Datos.....	16
Figura 11 Creación de controladores.....	17
Figura 12 Vista Apis.....	19
Figura 13 Prueba Api Postman.....	19
Figura 14 Vistas.....	19
Figura 15 Modelo arduino para lectura RFID.....	20
Figura 16 Api Scann Controller.....	21

RESUMEN

El presente trabajo de integración curricular tuvo como finalidad desarrollar un sistema de inventario basado en la tecnología de identificación por radiofrecuencia para la gestión de equipos tecnológicos en la carrera de Computación de la ESPAM MFL. Para el desarrollo de este trabajo de integración curricular se utilizó la metodología Extreme Programming, basada en cuatro etapas: Planificación, Diseño, Desarrollo, Pruebas. En la etapa de planificación se logró obtener todos los requerimientos necesarios para el perfecto funcionamiento del sistema y se estableció la arquitectura de comunicación entre el hardware y el software, luego, en la etapa del diseño se desarrollaron los diagramas correspondientes de clase y de caso de usos, además se elaboró la arquitectura de comunicación de forma visual, posterior a esto en la etapa de desarrollo se elaboró el backend y frontend en ASP.NET y se estableció la conexión con el lector de identificación por radiofrecuencia, finalmente en la etapa de pruebas se evaluó el sistema desarrollado para validar el funcionamiento de este. La implementación de un sistema de inventario basado en tecnología RFID ha permitido automatizar la gestión de los equipos tecnológicos en la carrera de Computación de la ESPAM MFL. Esta solución ha reemplazado la administración manual, brindando mayor eficiencia y accesibilidad a los datos. Gracias al sistema de inventario web y al lector RFID, se ha logrado agilizar el proceso de registro de datos, reduciendo significativamente el tiempo requerido.

Palabras clave:

Identificación por radiofrecuencia, ASP.NET, sistema de inventario web.

ABSTRACT

The present curricular integration project aimed to develop an inventory system based on radio frequency identification (RFID) technology for managing technological equipment in the Computer Science program at ESPAM MFL. The Extreme Programming methodology was used for the development of this curricular integration project, which is based on four stages: Planning, Design, Development, and Testing. In the planning stage, all the necessary requirements were obtained to ensure the proper functioning of the system, and the communication architecture between the hardware and software was established. In the design stage, the corresponding class and use case diagrams were developed, and the communication architecture was visually designed. Subsequently, in the development stage, the backend and frontend were developed in ASP.NET, and the connection with the RFID reader was established. Finally, in the testing stage, the developed system was evaluated to validate its functionality. The implementation of an RFID-based inventory system has automated the management of technological equipment in the Computer Science program at ESPAM MFL. This solution has replaced manual administration, providing greater efficiency and accessibility to the data. Thanks to the web-based inventory system and the RFID reader, the data registration process has been streamlined, significantly reducing the required time.

Keywords:

Radiofrequency identification, ASP.NET, web inventory system..

CAPÍTULO I ANTECEDENTES

1.1 DESCRIPCIÓN DE LA INSTITUCIÓN

La Escuela Superior Politécnica Agropecuaria de Manabí "Manuel Félix López" (ESPAM MFL), es una universidad pública ecuatoriana ubicada en la ciudad de Calceta perteneciente al cantón Bolívar. La ESPAM MFL es la principal universidad de la zona norte de la provincia de Manabí (ESPAM MFL, 2019) y tiene como finalidad formar profesionales de calidad que sustenten la creación de proyectos en los distintos campos laborales; de la misma manera sean capaces de generar fuentes de trabajo que aporten a los ámbitos económicos, sociales, ambientales, culturales y demás. Su modelo educativo involucra las principales enseñanzas para el desarrollo de las habilidades, destrezas, valores y demás factores que van moldeando la vida profesional de un estudiante, tomando como guías a educadores calificados en el proceso de su formación.

Dentro de la oferta académica de la ESPAM MFL, se encuentran carreras profesionalizantes acorde a las necesidades de la sociedad; como es el caso de la carrera de Computación, la cual tiene como objetivo “formar profesionales para la solución de problemas sociales, regionales y nacionales vinculados al modelo constructivista y desarrollador productivo, dentro de equipos multidisciplinares e interdisciplinares” (ESPAM MFL, 2022). Por tanto, los profesionales en Computación, a lo largo de la carrera universitaria, adquieren conocimientos teóricos que serán utilizados mediante la práctica en proyectos de interés informático; y que serán desarrollados con el fin de suplir algún tipo de necesidad. El edificio de la carrera de computación alberga diversas áreas dedicadas a la gestión, docencia, investigación y vinculación. Con el objetivo de adaptarse a las necesidades actuales de automatización y mantenerse a la vanguardia, se busca mejorar las instalaciones para convertirlo en un edificio inteligente. Una de estas áreas es la UDIV (Unidad de Docencia, Investigación y Vinculación), cuyo propósito principal es fortalecer el aprendizaje de los estudiantes de la carrera de computación de la ESPAM MFL mediante la implementación de una infraestructura automatizada en sus laboratorios. Esta iniciativa busca aprovechar la tecnología más avanzada y satisfacer la creciente demanda de automatización en el proceso de enseñanza.

1.2 DESCRIPCIÓN DE LA INTERVENCIÓN

La necesidad de llevar un registro de información ha llevado a la creación de diversos sistemas de almacenamiento de datos. Estos han supuesto una revolución en la forma en que se puede almacenar y procesar la información, debido a que integran un conjunto de reglas y procedimientos los cuales ayudan a verificar el estado de diversos productos o bienes (Ballestas, 2021) (Peña, 2020) (Merizalde, 2021), y estos han tenido un impacto significativo en la sociedad (Díaz, 2020). Aunque se haya superado la barrera digital, por costumbre, la gestión de estos sigue siendo manual. Por ejemplo, un sistema de inventario manual puede lograr ser muy eficiente, sin embargo, está sujeto a los errores en la introducción de datos erróneos o elaborar algún proceso que esté estructurado de forma incorrecta debido a errores humanos (López, 2019), lo cual puede generar varios problemas al intentar obtener la información.

La evolución tecnológica en los sistemas de inventario ha aumentado significativamente obteniendo varias mejoras en la gestión de información, una de estas es la Tecnología de Identificación por Radiofrecuencia RFID, se basa en la identificación automática mediante el uso de dispositivos que emiten ondas electromagnéticas para la difusión de señales que transfieren información entre objetos distantes (Sánchez, 2021) (Rodríguez, 2021).

La tecnología RFID permite la detección de objetos o productos en grandes cantidades de una forma inalámbrica, haciendo uso de etiquetas o tarjetas donde se puede grabar información de objetos u otros elementos y así obtener una identificación única, esta tarjeta está compuesta por un microchip y una antena, la cual es la encargada de recibir y responder las peticiones de un emisor a un receptor RFID (Niño, 2020) (Plúa & Mejillones, 2022) (Bohórquez & Villegas, 2021). Es decir, presenta una serie de ventajas entre esas se destaca la seguridad de los mismos, verificación de autenticidad, reducción de colisiones de señal y por ende de los costes en su utilización. Rojas (2018), Suarez (2020) y Alencastri (2020) mencionan que la tecnología RFID cuenta con una rapidez y precisión mucho mayor que otras tecnologías de identificación, además su durabilidad es mayor, la cual permitirá ahorrar tiempo y dinero.

El principal inconveniente de esta tecnología es su elevado valor económico lo cual hace difícil su implementación, pero recalcan que esta tecnología puede brindar grandes beneficios a largo plazo (Colcha & Lozano, 2020) (Merizalde, 2022).

Es por ello que los sistemas de inventario con tecnología de identificación por radiofrecuencia tienen mayor eficiencia que los nombrados anteriormente e infalibles (López, 2022), esto debido a que no se requiere la manipulación humana para tareas como la introducción de datos, ya que este automatiza la recopilación de datos en un tiempo muy reducido sin obtener errores en la misma. (Valderrama, 2018) (Parrales, 2022) afirman que, mediante la utilización de esta tecnología se podrá realizar inventarios o consultar sus características asociadas de manera eficiente.

La UDIV de infraestructura maneja un sistema de inventario manual, y como ya se mencionó anteriormente, esto no es lo más óptimo. De esta forma se contribuyó con la Implementación del proyecto que tiene como tema “Elaboración de un sistema de inventario basado en tecnología RFID”, cuya finalidad es agilizar el proceso de inventariado de los equipos tecnológicos de la carrera de Computación de la ESPAM MFL mediante la tecnología de radiofrecuencia para tener una mejor accesibilidad a los datos con mayor eficiencia. Mediante la implementación de esta tecnología, se reemplazó el proceso manual de inventariado que se realizaba dentro de la carrera. Para agilizar este proceso se cuenta con un sistema de inventario web y un lector RFID, el cual recepta todos los datos pertinentes en menos tiempo.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Desarrollar un sistema de inventario basado en tecnología de identificación por radiofrecuencia para la gestión de equipos tecnológicos en la carrera de computación de la ESPAM MFL.

1.3.2 OBJETIVOS ESPECÍFICOS

- Determinar los requerimientos necesarios para implementar el sistema de inventario con tecnología RFID.
- Establecer la arquitectura de comunicación entre el software y hardware.
- Codificar el sistema de inventario.
- Integrar el sistema de inventario con el dispositivo de tecnología RFID.

CAPÍTULO II. DESARROLLO METODOLÓGICO DE LA INTERVENCIÓN

El presente trabajo de titulación fue realizado utilizando la metodología Extreme Programming (XP), la cual se basa en un conjunto de prácticas y técnicas para el desarrollo de software (Abdias, 2022). Para llevar a cabo este proyecto, se aplicaron todas las etapas de la metodología, incluyendo la planificación, el diseño, la codificación y las pruebas. Además, se utilizó la técnica de entrevista para obtener información relevante por parte del cliente y garantizar que los requisitos del sistema fueran comprendidos correctamente. Es importante destacar que este trabajo se realizó con recursos propios, lo que implicó un esfuerzo adicional por parte del equipo de desarrollo.

2.1 PLANIFICACIÓN

En esta etapa se planteó obtener todos los requerimientos necesarios para el desarrollo del sistema de inventario. De acuerdo con (Rodríguez ,2019) una entrevista permite obtener información directa y detallada sobre las necesidades, requisitos y expectativas de los usuarios y partes interesadas del sistema, es por aquello que se eligió realizar una entrevista semiestructurada debido a que esta permite al entrevistador adaptar y ajustar las preguntas según el contexto y las respuestas del entrevistado (Lopezosa ,2020). Para esta entrevista se realizó una guía de preguntas con el propósito de obtener información detallada y precisa que contribuyeron a la elaboración del sistema, durante esta entrevista se escucharon atentamente las respuestas de los entrevistados con el fin de documentar toda la información relevante. Posterior a la entrevista, se estableció el estándar IEEE 830 con el objetivo de documentar los requerimientos establecidos en la entrevista para elaborar el sistema de inventario con tecnología RFID, al seguir este estándar, se utiliza un lenguaje común y una estructura consistente, lo que facilita la comprensión de los requisitos por parte de los desarrolladores, los clientes y los usuarios finales (Rodríguez ,2021).

En esta misma etapa se estableció realizar una revisión bibliográfica sobre los protocolos de comunicación de la tecnología RFID entre el hardware y el

software con el propósito de adquirir conocimiento sobre esta tecnología inicialmente se estableció el nombre del tema a investigar, posterior a esto los criterios de búsqueda como año de publicación y tipos de estudios, luego se realizó la búsqueda exhaustiva en bibliotecas digitales o fuentes relevantes para encontrar estudios relacionados al tema para finalmente procesar la información y obtener los resultados.

2.2 DISEÑO

En esta etapa, se planteó elaborar tanto el diagrama de clases como el diagrama de casos de uso para representar de manera detallada la estructura y la funcionalidad del sistema. El diagrama de clases es una representación visual que muestra las clases del sistema, sus atributos y sus relaciones, permitiendo definir la estructura del sistema y las interacciones entre las diferentes entidades (Vega ,2019). Por otro lado, el diagrama de casos de uso describe las diferentes interacciones entre los actores (Garrido ,2021).

Además, en base a la revisión bibliográfica se estableció implementar la arquitectura de comunicación entre el hardware y el software con la finalidad de representar de manera visual la comunicación entre estos.

2.3 DESARROLLO

De acuerdo con los requerimientos obtenidos, y los diagramas elaborados anteriormente se elaboró la base de datos en el gestor de SQL SERVER.

Para el desarrollo del Backend se utilizó Visual Studio y el marco de desarrollo de aplicaciones web ASP.NET, haciendo el uso respectivo de estos se logró realizar todas las tareas de procesamiento de datos como la lógica de negocios y el consumo de servicios de una aplicación desarrollada anteriormente denominada Asignación de espacios para así proporcionar las funcionalidades solicitadas por el cliente.

Luego se elaboró el Frontend basado en el framework de código abierto de Bootstrap que se basa en estilos de CSS (Cascading Style Sheets) el cual permite dar el formato visual al desarrollo web. Finalmente, en esta etapa, se

integró la arquitectura de comunicación elaborada por un arduino, tarjeta de lectura RFID y una app móvil para visualizar el procesamiento de datos. Además, se elaboraron APIs que permiten comunicar la app móvil con el sistema de inventario.

2.4 PRUEBAS

Una vez obtenida la versión final del sistema de inventario se realizaron las pruebas unitarias necesarias para validar la ejecución de cada una de las funcionalidades y se corrigieron cada uno de los errores presentados en esta etapa, y así se logró obtener un sistema completamente funcional y posteriormente su respectiva aprobación.

CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA

En la siguiente descripción de experiencias, se presenta la relación entre los entregables y fases metodológicas previamente establecidas, aplicando las técnicas y métodos descritos en la sección de metodología de la intervención.

3.1 PLANIFICACIÓN

Durante la entrevista aplicada a las personas interesadas en el desarrollo del proyecto se trataron temas relacionados a la obtención de los requerimientos. Durante dicha entrevista, se discutieron aspectos cruciales para comprender y documentar adecuadamente los requisitos del proyecto, se realizaron preguntas detalladas para identificar y capturar todas las necesidades con el fin de obtener una base sólida para la implementación del sistema.

The image shows two pages of a meeting act. The left page contains the following text:

1. PROGRAMACIÓN
Ubicación: Dirección de cámara de computación – Piso 1
Fecha: 02/11/2022
Hora de inicio: 11:00 Am
Hora de fin: 12:00 Am

2. OBJETIVOS
Los objetivos de la reunión fueron:

- Identificar los requisitos del sistema.
- Establecer el funcionamiento del sistema.
- Establecer el diseño del sistema.

3. AGENDA
Los temas de la agenda que se trataron son:

3.1. REQUERIMIENTOS FUNCIONALES

- El sistema debe contar con un medio de autenticación de usuarios para poder identificar los diferentes roles
- El sistema debe ser capaz de leer los objetos dentro de su rango haciendo uso del lector RFID
- El sistema debe mandar una alerta a el administrador del sistema cuando se identifique que un producto salió de su ubicación
- El sistema debe permitir poder ingresar información de los objetos
- El sistema debe tener claro los diferentes roles de usuarios del sistema
- El sistema debe ser capaz de generar reportes los cuales contaran con información importante y detallada de los objetos dentro del laboratorio
- El sistema debe ser capaz de poder dar de baja a equipos siempre y cuando estos se realicen mediante el usuario administrador
- El sistema no debe eliminar ningún registro de los objetos

4. ASISTENCIA
Los asistentes esta reunión fueron:

The right page contains a table and the following text:

Persona	HORA	
	LLEGADA	SALIDA
LOPEZ MORA CHRISTIAN ISRAEL	11:00 am	12:00 am
ORMAZA CALDERON JONATHAN GEOVANNY	11:00 am	12:00 am
RAMON JOFFRE MOREIRA PICO	11:00 am	12:00 am

5. TEMAS TRATADOS

5.1. Recolección de los requerimientos no funcionales (50 minutos)
Se encuentra documentado en la agenda.

5.2. Adquisición de equipos (10 minutos)
Se estableció una charla sobre los equipos de lectura RFID y sus precios

6. ACUERDOS

6.1. Próxima reunión
Se estableció otra reunión para la fecha 05/01/2023

6.2. Adquisición de equipos
Se acordó en conseguir un producto más económico que el que se tenía planteado originalmente

7. FIRMAS

LOPEZ MORA CHRISTIAN ISRAEL _____
ORMAZA CALDERON JONATHAN GEOVANNY _____
RAMON JOFFRE MOREIRA PICO _____

Figura 1 Acta de entrevista
Fuente: Los Autores

Luego de abordar diferentes temas se logró obtener un panorama totalmente amplio de lo que se pretendía realizar, es así como se procedió a documentar bajo el estándar IEEE 830 cada uno de los requerimientos necesarios y establecidos durante la entrevista, para así obtener la elicitación de requerimientos (**Anexo 1**).

Tabla 1 Lista de requerimientos

N° Requerimiento	Requerimiento	Alta/Eencial
RF001	Asignación de equipos	Alta/Eencial
RF002	Lectura de equipos con el uso del lector RFID	Alta/Eencial
RF003	No eliminar ningún registro	Alta/Eencial
RF004	Reportes	Alta/Eencial
RF005	Interfaz simple y de fácil comprensión	Alta/Eencial
RF006	Verificar autenticación de usuario.	Alta/Eencial
RF007	Ingreso, edición, eliminación y asignación de los laboratorios	Alta/Eencial
RF008	Ingreso, edición, eliminación y asignación de los usuarios	Alta/Eencial
RF009	Ingreso y edición de los equipos	Alta/Eencial
RF010	Validación de tipos de usuarios	Alta/Eencial

Fuente: Los Autores

Para establecer los requisitos del diseño del sistema se tuvo en cuenta aspectos como la facilidad de uso, accesibilidad, estética y funcionalidad (**Anexo 2**).

Además, en esta etapa del proyecto, se llevó a cabo un estudio bibliográfico exhaustivo sobre la comunicación RFID, centrándonos específicamente en los protocolos SPI, USB y HTTP en el desarrollo de software. El objetivo principal de este estudio fue recopilar información relevante y actualizada sobre estos protocolos y su aplicación en el ámbito de la comunicación RFID.

Para garantizar la relevancia de la información recopilada, se establecieron criterios de búsqueda específicos. Se seleccionaron publicaciones con un año de publicación comprendido entre 2019 y 2023 para asegurarnos de incluir la literatura más reciente y actualizada sobre el tema. Además, se limitó la búsqueda a estudios que se encontraran en formato de tesis, artículos o libros, ya que estos suelen ser los tipos de publicaciones más rigurosas y confiables para obtener información académica y técnica.

El proceso de búsqueda bibliográfica se llevó a cabo en diversas fuentes de información, como bases de datos académicas, bibliotecas digitales y repositorios especializados. Se utilizaron palabras clave relacionadas con los protocolos SPI, USB y HTTP, así como con la comunicación RFID, para asegurar la obtención de resultados relevantes.

Una vez recopiladas las fuentes bibliográficas pertinentes, se realizó una revisión detallada de cada publicación seleccionada. Se analizaron los contenidos de los estudios identificados para extraer información relevante sobre los protocolos SPI, USB y HTTP en el contexto de la comunicación RFID (**Anexo 3**). Este exhaustivo estudio bibliográfico nos permitió obtener una sólida base de conocimientos sobre los protocolos SPI, USB y HTTP en relación con la comunicación RFID.

3.2 DISEÑO

Para la elaboración de los diagramas de clase y casos de uso se tuvo en consideración los requisitos documentados en la etapa anterior. Estos requisitos de software establecidos anteriormente son una parte fundamental en la elaboración de diagramas de clase y diagramas de casos de uso, ya que proporcionan la base para comprender las funcionalidades y características que el sistema debe tener.

La elaboración de estos diagramas se llevó a cabo utilizando el software GitMind, el cual ofrece una interfaz intuitiva y de fácil uso para crear diagramas. Una de las ventajas principales de GitMind es la posibilidad de colaborar en tiempo real, lo que facilita el trabajo en equipo al permitir que varias personas trabajen simultáneamente en el mismo diagrama, además GitMind cuenta con una amplia variedad de plantillas predefinidas

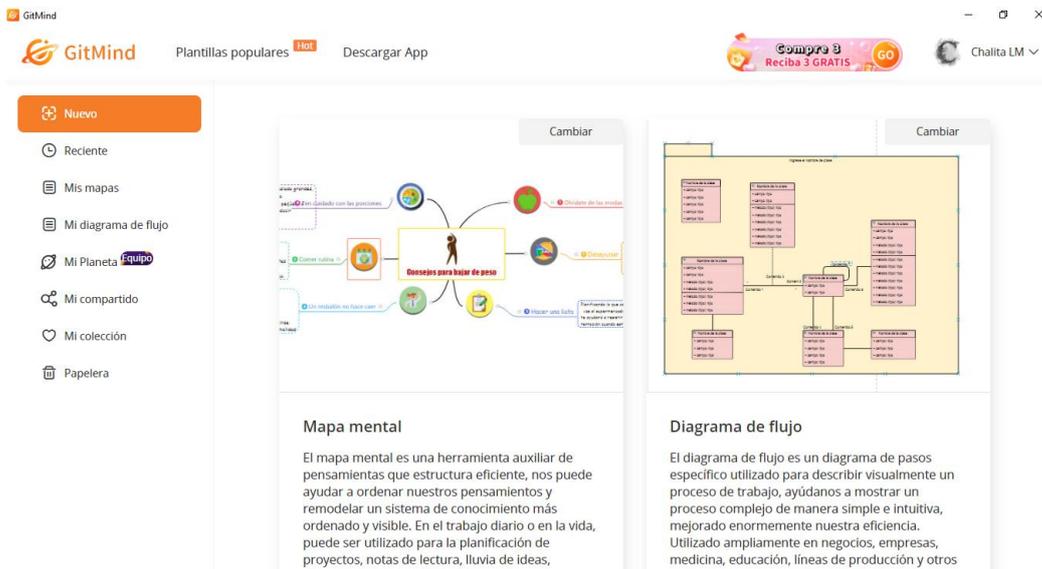


Figura 2 Vista menú principal GitMind
Fuente: Los Autores

Durante la elaboración del diagrama de clases, se estableció la estructura del sistema de manera clara y organizada. Este diagrama proporcionó una representación visual de las clases y sus relaciones, permitiendo visualizar de forma intuitiva cómo interactúan entre sí. Además, se incluyeron los atributos y métodos correspondientes a cada clase, brindando información detallada sobre las características y comportamientos de cada entidad en el sistema.

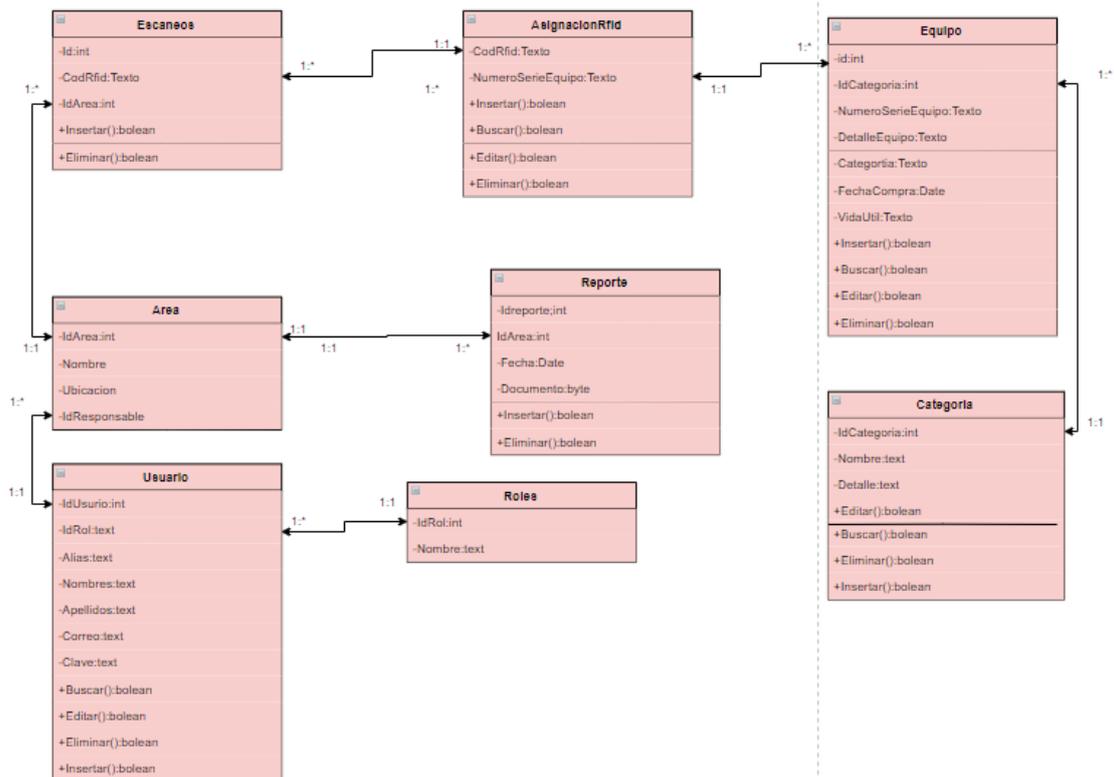


Figura 3 Diagrama de clases
Fuente: Los Autores

Durante la elaboración del diagrama de casos de uso, se establecieron las interacciones entre los actores y el sistema, proporcionando una representación visual de los diferentes casos de uso y sus relaciones. Este diagrama permitió identificar de manera clara y concisa las funcionalidades y acciones que el sistema debe realizar en respuesta a las interacciones de los usuarios. Cada caso de uso se representa de manera individual, describiendo los pasos o etapas necesarios para completar dicha función (**Anexo 4**). Finalmente, en esta etapa de diseño se elaboró la arquitectura de comunicación entre el hardware y el software

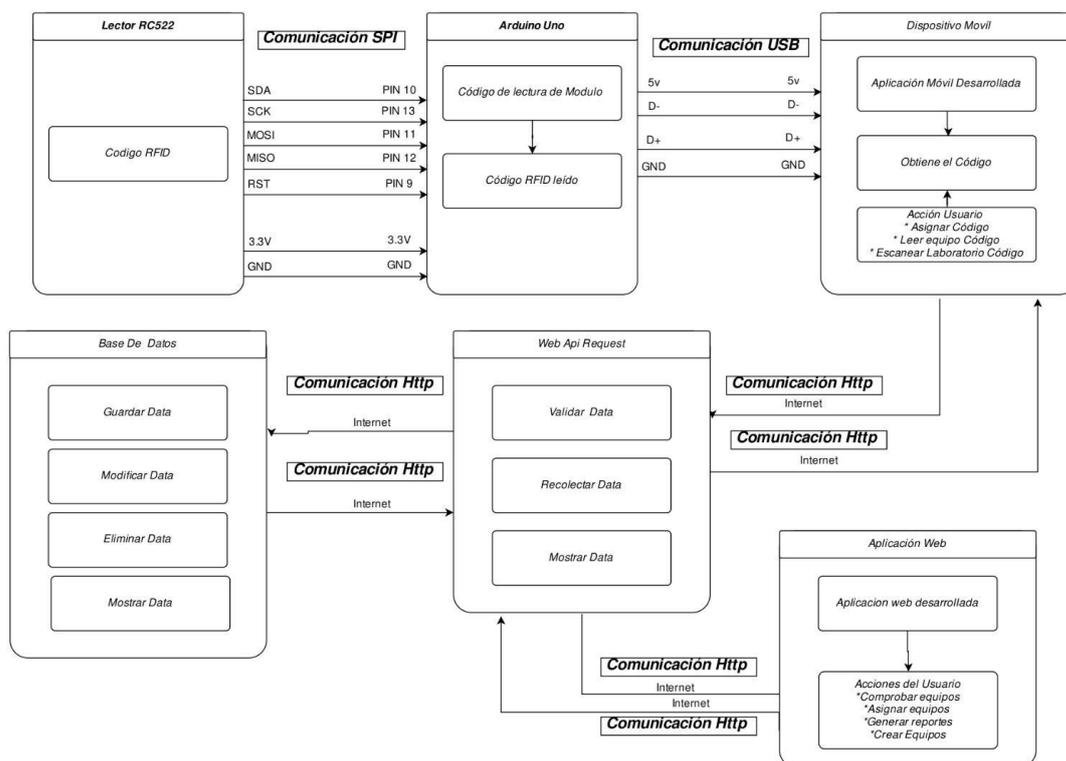


Figura 4 Diagrama de arquitectura
Fuente: Los Autores

Estos diagramas cuentan con el siguiente funcionamiento, la tarjeta de lectura RFID RC522 se comunica con el Arduino Uno mediante el protocolo SPI, esta es una forma de comunicación síncrona que permite la transferencia de datos entre dispositivos electrónicos. En este contexto, el Arduino Uno actúa como el controlador principal y la tarjeta RFID RC522 es el dispositivo periférico. A través del protocolo SPI, el Arduino Uno envía y recibe comandos y datos a la tarjeta RFID RC522 para realizar operaciones como la lectura de etiquetas RFID.

El Arduino Uno, al recibir los datos de la tarjeta RFID RC522, procesa esta información y utiliza el protocolo de comunicación USB para establecer una conexión con el dispositivo móvil.

El dispositivo móvil cuenta con una aplicación instalada que permite recibir datos del arduino e interactuar con los servicios (APIs implementados en el sistema de inventario a través de solicitudes HTTP estas solicitudes pueden ser del tipo GET, POST. Finalmente, las APIs actúan como intermediarios entre el sistema de inventario y la base de datos, facilitando la comunicación y la interacción entre ambos.

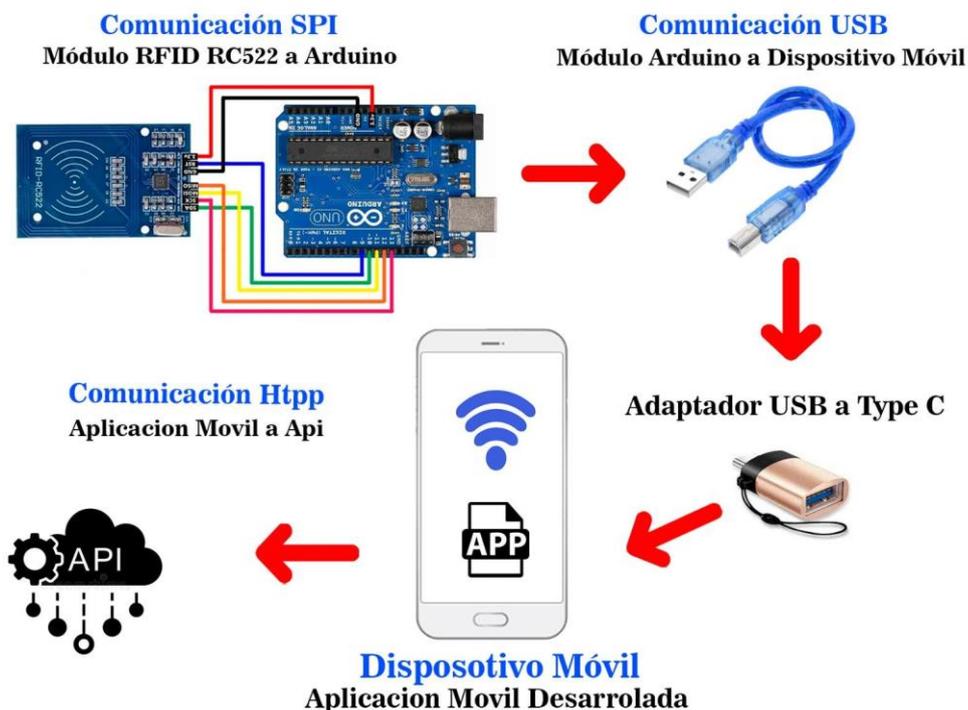


Figura 5 Diagrama Físico de comunicación
Fuente: Los Autores

3.3 DESARROLLO

En base a los requerimientos obtenidos y los diagramas elaborados, se procedió a la creación de la base de datos utilizando el gestor de SQL Server. Durante este proceso, se diseñaron las tablas correspondientes a cada clase, definiendo adecuadamente sus campos y tipos de datos, una vez que las tablas fueron creadas, se procedió a establecer las relaciones entre ellas, asegurando la integridad referencial y garantizando la coherencia de los datos en el sistema. Este enfoque permitió construir una base de datos sólida y eficiente, que cumpliera con los requerimientos solicitados.

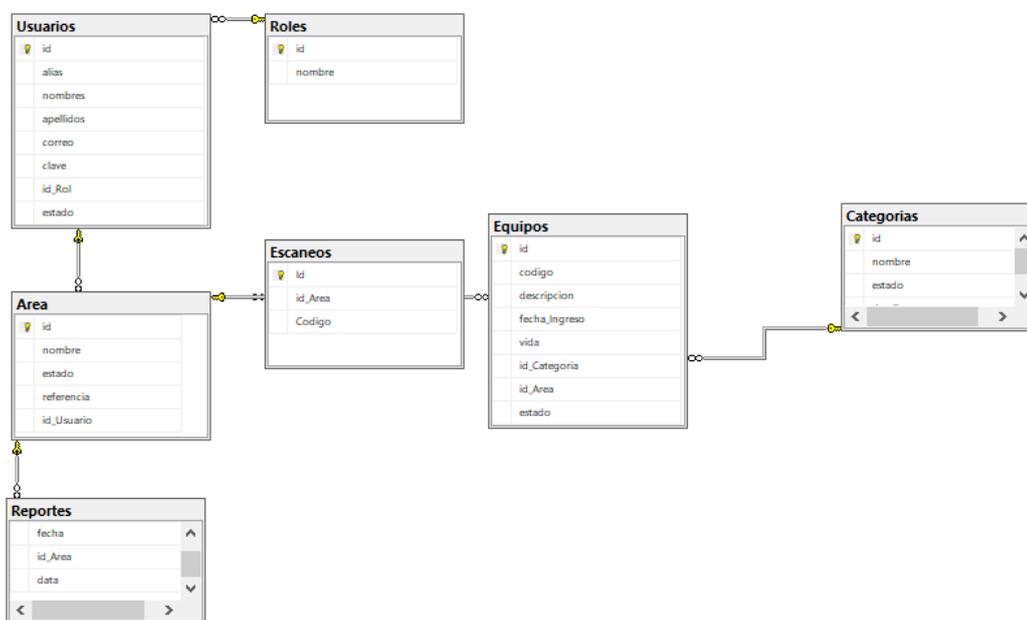


Figura 6 Diagrama Base de datos
Fuente: Los Autores

En el proceso de desarrollo del Backend, se empleó Visual Studio como entorno de desarrollo y el marco de desarrollo de aplicaciones web ASP.NET. Una de las primeras acciones consistió en crear un proyecto con estructuras MVC (Modelo, Vista, Controlador).

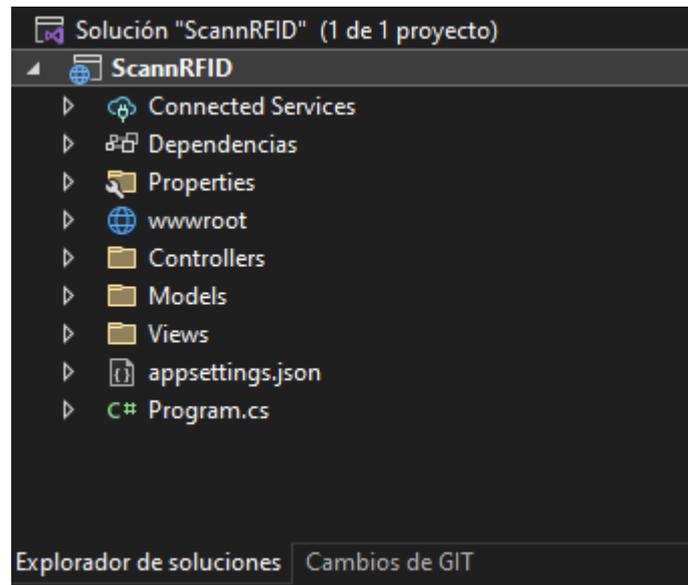


Figura 7 Modelo – Vista - Controlador

Fuente: Los Autores

Luego se instaló el paquete del Entity Framework, un mapeador objeto-relacional (ORM, por sus siglas en inglés) que posibilita el mapeo ágil y sencillo de la base de datos en modelos. Además la incorporación de este framework permitió llevar a cabo consultas directamente desde el Backend sin la necesidad de crear procedimientos almacenados, el uso del Entity Framework contribuyó a mejorar la eficiencia y la flexibilidad en la interacción entre la aplicación y la base de datos.

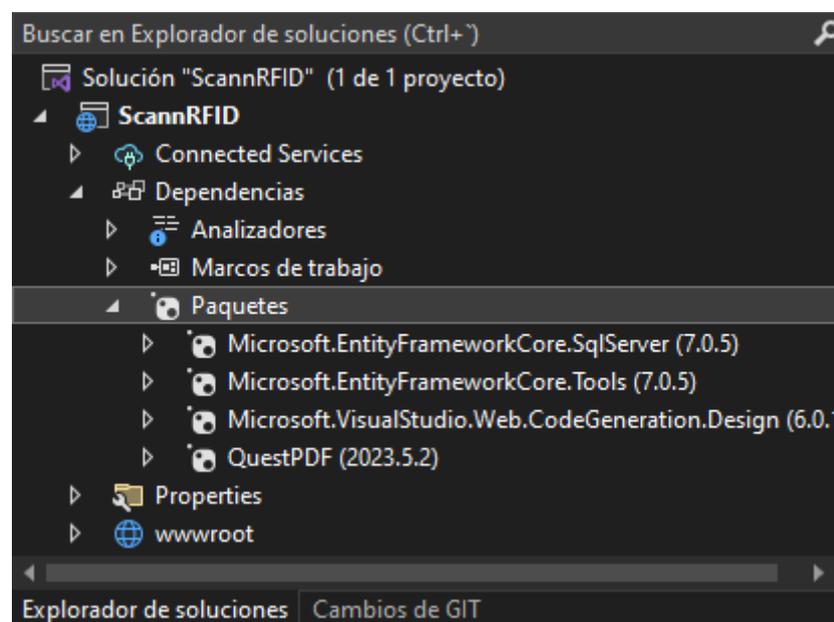
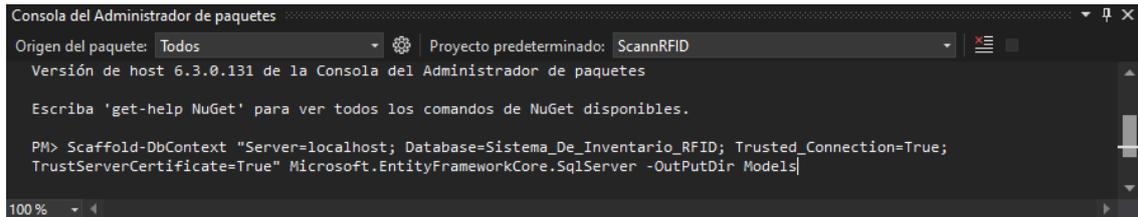


Figura 8 Paquete Entity Framework

Fuente: Los Autores

Por medio del comando presentado en la siguiente figura se logró mapear la base de datos en la carpeta Models



```

Consola del Administrador de paquetes
Origen del paquete: Todos Proyecto predeterminado: ScannRFID
Versión de host 6.3.0.131 de la Consola del Administrador de paquetes

Escriba 'get-help NuGet' para ver todos los comandos de NuGet disponibles.

PM> Scaffold-DbContext "Server=localhost; Database=Sistema_De_Inventario_RFID; Trusted_Connection=True; TrustServerCertificate=True" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models
  
```

Figura 9 Mapeo de Base de Datos
Fuente: Los Autores

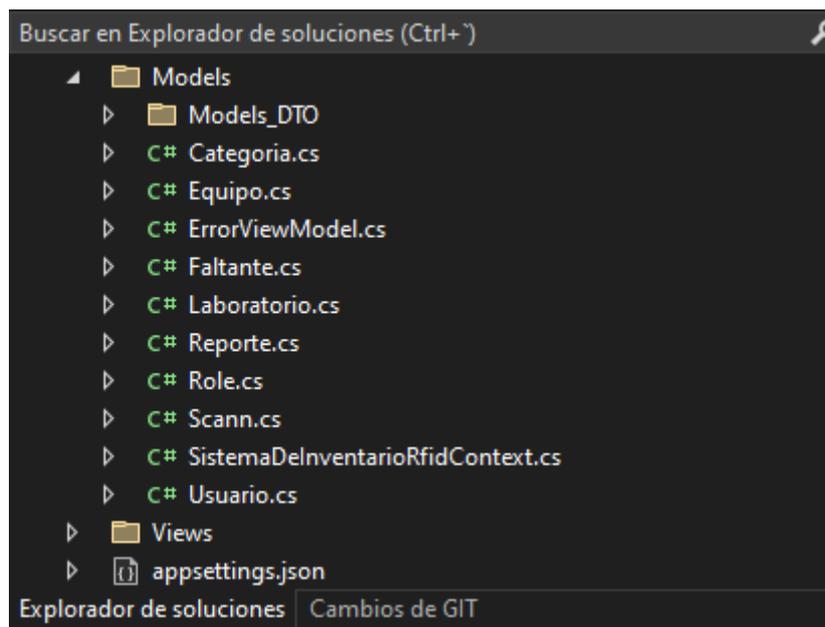


Figura 10 Mapeo de Base de Datos
Fuente: Los Autores

Para completar la estructura del Backend, se crearon los controladores y las vistas correspondientes.

Estos controladores son módulos que se encargan de recibir las solicitudes enviadas por el usuario desde el Frontend o cualquier otra interfaz. Estos controladores actúan como intermediarios entre el cliente y el servidor, y se encargan de manejar las diferentes rutas y acciones que el usuario desea realizar.

Los controladores también se encargan de gestionar la lógica de negocio. Esto implica aplicar reglas específicas de la aplicación, validar los datos

proporcionados por el usuario, generar respuestas adecuadas y manejar posibles errores o excepciones.

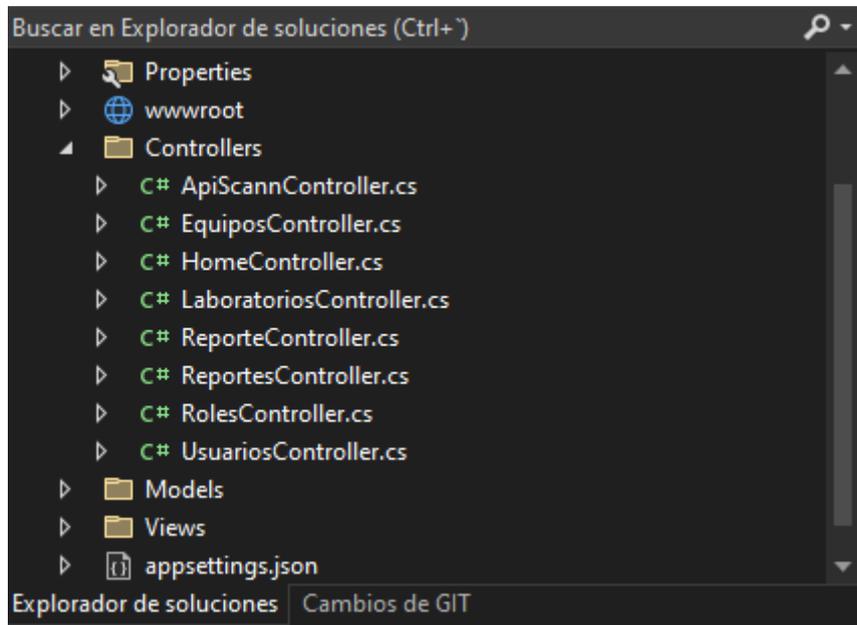


Figura 11 Creación de controladores
Fuente: Los Autores

Además se consumieron APIs desarrolladas en el proyecto “ASIGNACIÓN DE ESPACIOS” con el fin de obtener información almacenada.

Nombre de aplicación	Inicio	API
Aula		
API		Description
GET api/Aula		No documentation available.
GET api/Aula/{Codigo}		No documentation available.
POST api/Aula		No documentation available.
PUT api/Aula/{id}		No documentation available.
DELETE api/Aula/{id}		No documentation available.

Figura 12 Vista Apis
Fuente: Los Autores

Inicialmente se obtuvo el acceso a las APIs, luego se verificaron que estas funcionaran de forma correcta a través de postman para su posterior integración dentro del sistema de inventario.

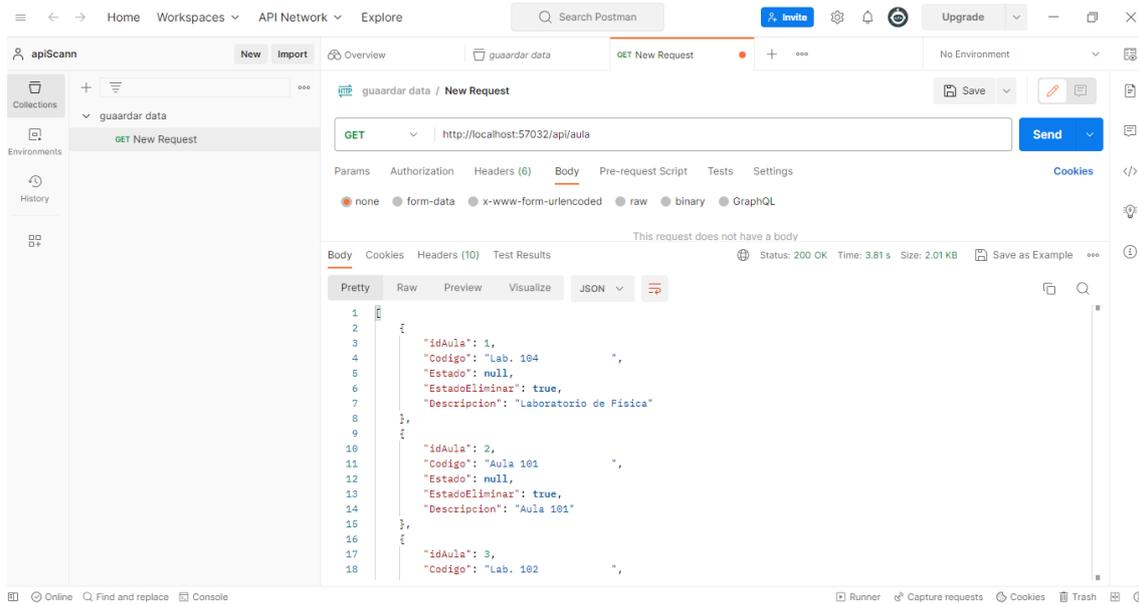


Figura 13 Prueba Api Postman
Fuente: Los Autores

Las vistas, por su parte, se encargaron de la presentación de los datos al usuario, y se utilizó el framework de código abierto de Bootstrap que se basa en estilos de CSS (Cascading Style Sheets) el cual permite dar el formato visual al desarrollo web(**Anexo 5**)

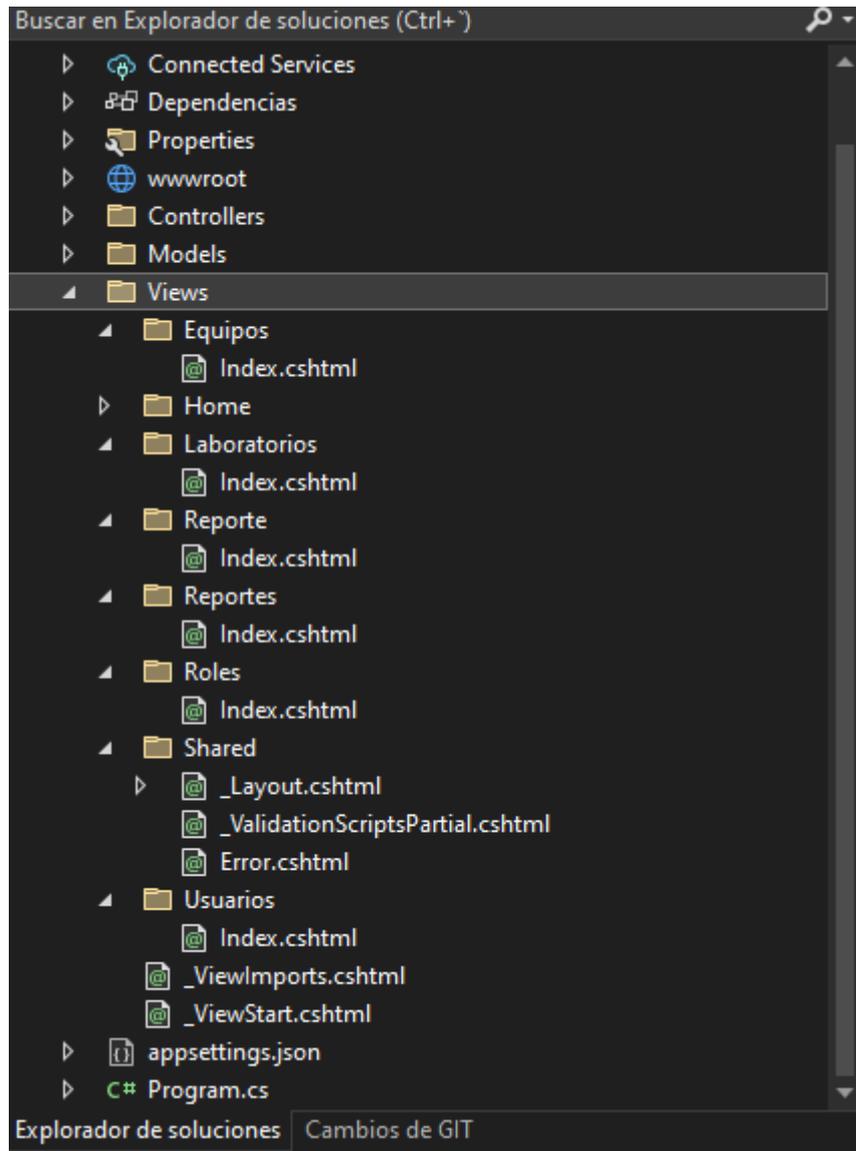


Figura 14 Vistas
Fuente: Los Autores

La elaboración del lector RFID se realizó a través de un arduino modelo Uno, una tarjeta de lectura RFID módulo RC522 y una aplicación móvil.

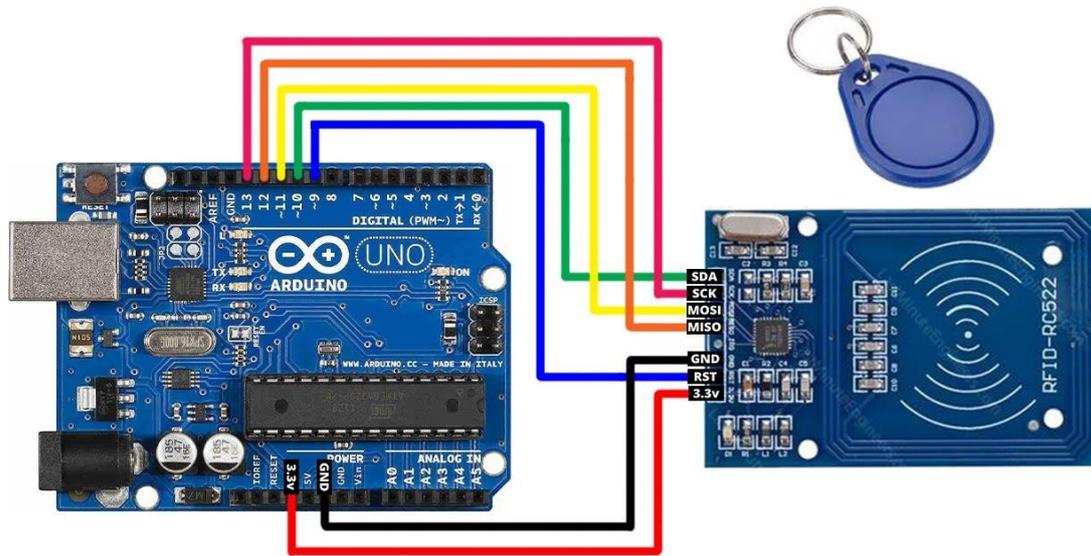


Figura 15 Modelo arduino para lectura RFID
Fuente: Los Autores

Basado en este esquema se procedió a conectar cada uno de los pines para el respectivo funcionamiento.

Se procedió a programar en el arduino mediante el software Arduino el protocolo de comunicación SPI (**Anexo 6**), luego se tuvo la necesidad de implementar una aplicación móvil en AdroidStudio que diera una vista gráfica del procesamiento de datos del arduino con el tarjeta RFID, esta aplicación era capaz de procesar los datos mediante el protocolo USB (**Anexo 7**).

Finalmente se implementaron APIs (Interfaces de Programación de Aplicaciones) que permiten la comunicación entre la aplicación móvil con el sistema de inventario. Estas APIs son puntos de entrada que facilitan la interacción entre el lector RFID(app móvil, arduino, tarjeta RFID) y el sistema, permitiendo el intercambio de información de manera eficiente y segura. Mediante estas APIs, se estableció un protocolo de comunicación que define cómo se envían y reciben los datos entre el lector RFID y el sistema de inventario. Esto incluye la definición de las rutas de acceso, los métodos de solicitud (como GET, POST) y los formatos de intercambio de datos como JSON. Al recibir una solicitud desde el lector RFID, las APIs procesan la información y la envían al sistema de inventario correspondiente. Esto puede implicar la actualización de la base de datos con la lectura de nuevos productos,

la verificación de existencias o cualquier otra operación requerida en el contexto del inventario.

```

3 referencias
public class ApiScannController : Controller
{
    private readonly ILogger<ApiScannController> _logger;
    private Models.SistemaDeInventarioRfidContext _db;

    0 referencias
    public ApiScannController(ILogger<ApiScannController> logger, SistemaDeInventarioRfidContext
    {
        _logger = logger;
        _db = db;
    }

    [HttpPost]
    0 referencias
    public async Task<IActionResult> GuardarDatos(int lab, string cod)...

    [HttpGet]
    0 referencias
    public async Task<IActionResult> ObtenerLaboratorios()...

    [HttpGet]
    0 referencias
    public async Task<IActionResult> ObtenerCategorias()...

    [HttpGet]
    0 referencias
    public async Task<IActionResult> ObtenerEquipo(string codigo)...

    [HttpPost]
    0 referencias
    public IActionResult GuardarEquipoNuevo(Equipo_Save_DTO equipo)...
}

```

Figura 16 Api Scann Controller
Fuente: Los Autores

3.4 PRUEBAS

Durante la etapa de pruebas de la aplicación, se realizó una serie de actividades para asegurar de que todo funcionara correctamente antes de su lanzamiento oficial. Fue un proceso muy importante para detectar y corregir cualquier error o problema antes de que los usuarios pudieran utilizarla.

En primer lugar, se planificó cuidadosamente todas las pruebas que se pretendían realizar. Además, se establecieron los criterios de aceptación que debían cumplirse para considerar que las pruebas eran exitosas.

Se procedió a ejecutar las pruebas paso a paso (**ANEXO 7**). Se probó la funcionalidad de la aplicación, asegurándose de que todas las características operaran correctamente. También se realizaron las pruebas de rendimiento para observar el comportamiento de la aplicación bajo diferentes cargas de trabajo y pruebas de seguridad para identificar posibles vulnerabilidades. Durante las pruebas, se registraron cuidadosamente todos los resultados

obtenidos. Se documentó cada uno de los errores y problemas encontrados de manera precisa, incluyendo la descripción del error, la forma de reproducirlo y cualquier información relevante para su posterior corrección. Se asignó una prioridad a cada problema identificado.

Una vez que finalizaron todas las pruebas y se corrigieron los errores encontrados, se volvió a realizar una revisión exhaustiva de la aplicación para asegurar de que todo estuviera funcionando correctamente.

Tabla 2 Resultados de pruebas unitarias

ID	DESCRIPCIÓN TEST	OBJETIVO	CONDICIÓN	RESULTADO OBTENIDO
1	Verificación de usuario (Login)	Comprobar la identidad del usuario	Usuario Existente ,Contraseña correcta	Éxito
2	Creación de usuario	Ingresar usuario	Cuidar datos sensibles	Éxito
3	Edición de usuario	Modificar datos usuario	Existencia de usuario, Contraseña encriptada, Modelo sea válido	Éxito
4	Eliminación de usuario	Eliminar usuario	Existencia de usuario	Éxito
5	Listar usuario	Obtener listado usuarios	Usuario válido ,Cuidar datos sensibles	Éxito
6	Buscar usuario	Obtener dato usuario	Existencia de usuario, Cuidar datos sensibles	Éxito
7	Encriptación de contraseña	Encriptar contraseña	Longitud mayor a 5 caracteres	Éxito
8	Creación de área	Ingresar área	Que el área no exista, Modelo es válido	Éxito
9	Listar área	Obtener listado de área	Área válida	Éxito
10	Buscar área	Obtener dato de área	Existencia de área , Área válida	Éxito
11	Creación de equipo	Ingresar equipo	No Existencia de equipo ,Modelo sea válido	Éxito
12	Edición de equipo	Modificar datos equipo	Existencia de equipo ,Modelo sea válido	Éxito
13	Eliminación de equipo	Eliminar equipo	Existencia de equipo	Éxito
14	Listar equipo	Obtener listado de equipo	Equipo válido	Éxito

15	Buscar equipo	Obtener dato de equipo	Existencia de equipo	Éxito
16	Ingresar Escaneos	Agregar equipos escaneados	Equipo asignado al área, el equipo no esté escaneado	Éxito
17	Verificación de áreas escaneadas	Verificar áreas escaneadas	Seleccionar área a verificar	Éxito
18	Resultados parciales del escaneo	Obtener resultados de los equipo escaneados	Mostrar resultados con su estado	Éxito
19	Lista de escaneos	Listar registros de escaneos	Escaneos Válidos	Éxito
20	Generación de reportes	Generar Reporte	Registro de escaneo completo	Éxito
21	Lista reportes	Ver registro de reportes	Fecha ingresada valida, Reporte exista	Éxito

Fuente: Los Autores

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- En conclusión, el proceso de recopilación de requerimientos para el desarrollo del sistema de inventario fue efectivo gracias al uso de entrevistas semi estructuradas, permitiendo obtener un panorama completo del proyecto. Los requerimientos identificados en las entrevistas fueron documentados y analizados detalladamente para cumplir con las necesidades y expectativas de los usuarios, y se identificaron las funcionalidades necesarias para el sistema. Además se tuvo en cuenta aspectos como la facilidad de uso, accesibilidad, estética y funcionalidad en el diseño del sistema. Con esto se garantiza un sistema eficiente y satisfactorio para los usuarios.
- La elaboración de los diagramas de clases y casos de uso, fueron elementos esenciales en el proceso de desarrollo del sistema. Estos componentes permiten una comprensión clara de los requisitos. Al analizar y representar visualmente la estructura y las interacciones del sistema a través de los diagramas de clases y casos de uso, se logró una mejor organización y comunicación entre el equipo de desarrollo.
- Desarrollar una base de datos es un proceso importante y esencial en cualquier sistema o proyecto que requiera almacenar y gestionar información de manera organizada y estructurada. En este proceso, se debe analizar cuidadosamente los requerimientos del sistema y definir las tablas y campos necesarios para satisfacer dichos requerimientos. Además, es importante establecer las relaciones y las claves primarias y foráneas para garantizar la integridad de los datos. El uso de la arquitectura Modelo-Vista-Controlador (MVC) en ASP.NET permite la mantenibilidad y escalabilidad del proyecto de forma sencilla. Es importante considerar la accesibilidad, la compatibilidad de dispositivos y la usabilidad durante el proceso de desarrollo.
- Finalmente, la etapa de pruebas de la aplicación fue un proceso fundamental y minucioso para garantizar su correcto funcionamiento

antes de su lanzamiento oficial. A través de una planificación cuidadosa, la ejecución de pruebas exhaustivas y la documentación precisa de los resultados, se logró identificar y corregir errores, vulnerabilidades y problemas de usabilidad. La documentación detallada de los errores y la asignación de prioridades facilitaron la posterior corrección de los problemas identificados, optimizando el tiempo y los recursos del equipo de desarrollo.

4.2 RECOMENDACIONES

- Se recomienda seguir un proceso metódico para la obtención de requerimientos del proyecto, además se aconseja documentar cada uno de los requerimientos establecidos durante la entrevista y realizar un análisis detallado para asegurar que se cumplen todas las necesidades y expectativas de los usuarios. Es importante que se tenga una comprensión clara de lo que se quiere lograr con el sistema.
- Realizar una revisión exhaustiva de los diagramas de clases y casos de uso antes de avanzar en la implementación del sistema. Esto ayudará a identificar posibles inconsistencias, omisiones o errores en la comprensión de los requisitos. Es fundamental que todos los miembros del equipo de desarrollo tengan una visión clara y compartida de la estructura y funcionalidades del sistema. Además los diagramas de clases y casos de uso deben ser discutidos y validados con todas las partes involucradas, de modo que se pueda obtener retroalimentación temprana y se puedan realizar ajustes si es necesario. Esto asegurará que el sistema se alinee con las expectativas y necesidades de los usuarios finales.
- Seguir buenas prácticas y estándares en el desarrollo de la base de datos para garantizar su eficiencia, seguridad y escalabilidad. Esto incluye la normalización de las tablas, la implementación de reglas de integridad de datos y la implementación de medidas de seguridad adecuadas para proteger los datos sensibles. Además, es importante probar y monitorear la base de datos de manera regular para detectar y corregir cualquier problema o error que pueda surgir con el tiempo. Utilizar la arquitectura Modelo-Vista-Controlador (MVC) en ASP.NET

para proyectos de backend que requieran una organización clara y separación de responsabilidades en el desarrollo. Además, esta arquitectura facilita la mantenibilidad y escalabilidad del proyecto, así como una mejor prueba y desarrollo, lo que a su vez conduce a un resultado final muy eficiente. Probar exhaustivamente el diseño en diferentes dispositivos y navegadores para asegurarse de que la experiencia del usuario sea consistente y sin problemas.

- Registrar todos los resultados de las pruebas de forma detallada. Documentar los errores y problemas encontrados de manera precisa, incluyendo descripciones claras y pasos para reproducirlos. Esto facilitará su corrección y seguimiento posterior. Asignar prioridades a los errores y problemas identificados durante las pruebas. Esto ayudará al equipo de desarrollo a enfocarse en los problemas más críticos y corregirlos de manera oportuna.

BIBLIOGRAFÍA

- Cedeño, A. (2020). Análisis y propuesta de un sistema de control de inventarios (rfid) para el manejo eficiente de las bodegas de la empresa mamut andino c.a. Repositorio general de la Universidad de Guayaquil.<http://repositorio.ug.edu.ec/handle/redug/53945>
- Mayorga, A. (2022). Análisis y diseño de un sistema que permita el control de acceso al personal administrativo ubicado en la matriz de la universidad técnica de babahoyo, utilizando tecnología RFID. Repositorio general de la Universidad Técnica de Babahoyo. <http://dspace.utb.edu.ec/handle/49000/11695>
- Quitian, B. (2021). Sistema de control de inventarios para el laboratorio corporativo (sede Medellín) de la empresa Postobón S.A. (Vol. 3, Issue 2). Repositorio general de la Universidad de Antioquia. https://bibliotecadigital.udea.edu.co/bitstream/10495/24220/5/QuitianSamuel_2021_SistemaInventariosLaboratorio.pdf
- Sánchez, B , y Villegas, R. (2021). Prototipo aplicativo para administración, nutrición y control de clientes mediante tecnología rfid en el gimnasio belgica. Repositorio general de la Universidad de Guayaquil. <http://repositorio.ug.edu.ec/handle/redug/52253>
- Colcha, C., Vera, L. (2020). Análisis técnico y económico de la implementación de un sistema de control de inventarios, mediante un prototipo escalable, usando tecnología rfid y software libre, para la empresa Hidroelec. Repositorio general de la Universidad de Guayaquil.<http://repositorio.ug.edu.ec/handle/redug/48822>
- Cueva, R. (2020). Diferencias entre gestión de inventarios y gestión de almacén. in applied microbiology and biotechnology (vol. 2507, edición 1). <https://doi.org/10.1016/j.solener.2019.02.027%0ahttps://www.golder.com/insights/block-caving-a-viable-alternative/%0a???>
- ESPAM MFL. (2022). Oferta de grados. <http://www.espam.edu.ec/web/oferta/grado/computacion.aspx>

- López, K. (2019). Propuesta de sistema de control de inventarios en la empresa laboratorios qfa Ltda . Repositorio General de la Unicatolica Lumen Gentium. <https://repository.unicatolica.edu.co/handle/20.500.12237/2239>
- Lopez, D. (2022). Desarrollo de prototipo de sistema de control y monitoreo de usuarios del comedor universitario, basado en tecnología rfid, en la universidad nacional amazónica de madre de dios. Repositorio de la universidad nacional amazónica de madre de Dios. <https://repositorio.unamad.edu.pe/handle/20.500.14070/867>
- Intriago, A. (2021). Diseño de un sistema de control y gestión de inventarios en una bodega de producto terminado. Repositorio de la universidad politécnica salesiana. <https://dspace.ups.edu.ec/handle/123456789/15717>
- Solís, K. (2022). Diseño de un Sistema de Inventario con Tecnología RFID para Optimizar Procesos de Almacenamiento en Comisión de Tránsito del Ecuador (Issue No 1, 2022). <https://doi.org/10.47711/0868-6351-190>
- Rondón, C, Bayona, J, Meneses, L, Parada, D, Castro, S, Martínez, E. (2020). Análisis de herramientas para gestión bibliográfica y control de acceso utilizando tecnología RFID. Eco Matemático, 11(2), 39–49. <https://doi.org/10.22463/17948231.3018>
- Baqué, J. (2022). Análisis y diseño de un sistema video portero ip con control de acceso rfid para mejorar la seguridad en la sala de docentes dos de la carrera de tecnologías de la información. Repositorio general de la universidad estatal del sur de Manabí. <http://repositorio.unesum.edu.ec/handle/53000/3065>
- Peña, E. (2020). Mejora del control de inventarios de boletería en el Fondo de Empleados Nacional de la Organización Terpel. S.A. <https://repository.uniminuto.edu/handle/10656/12378>
- Plúa, K., & Rodríguez, S. (2022). Prototipo de seguridad y control utilizando tarjetas rfid para los estudiantes de la escuela de educación básica particular colinas de la florida en la ciudad de guayaquil. Repositorio

general de la Universidad de Guayaquil.<http://repositorio.ug.edu.ec/handle/redug/64335>

Rodríguez, J. (2021). Análisis y eficiencia de comunicaciones y recursos en sistemas de identificación por radiofrecuencia. <https://ruidera.uclm.es/xmlui/handle/10578/28611>

Rojas, E. (2018). Obtenido de Aplicación de RFID para la optimización en el registro de ingresos de lotes de celulares a un centro de distribución: https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/9459/Rojas_je.pdf?sequence=3

Sánchez, J. (2021). Implementación de un sistema de identificación por radiofrecuencia para el control de inventario en la Universidad Politécnica de Sinaloa (Vol. 91). Repositorio general de la Universidad Politécnica de Sinaloa.<http://repositorio.upsin.edu.mx/formatos/A047SANCHEZLIZARRAGAJOSELUIS.10923.pdf>

Soto, G. (2021). Desarrollo de un sistema para la gestión de dispositivos ciberfísicos. caso: monitoreo de una edificación inteligente. Repositorio general de la Universidad de Sonora. <http://www.repositorioinstitucional.uson.mx/handle/20.500.12984/6531>

Suarez, M. (2020). Sistema Electrónico Basado En Tecnologías Wifi E Identificación Por Radiofrecuencia (Rfid) Para El Registro Automático De Personas En Lugares De Gran Afluencia. In Paper Knowledge . Toward a Media History of Documents.Repositorio general de la Escuela Superior Politécnica del Chimborazo. <http://dspace.espace.edu.ec/handle/123456789/621>

ESPAM MFL. (2019) Estatuto de la ESPAM MFL. Repositorio general de la ESPAM MFL. <http://www.espam.edu.ec/recursos/sitio/espam/ESTATUTOESPAMMFL20190910-CES.pdf>

Valderrama, J. (2018). Obtenido de Propuesta para la implementación de la tecnología RFID en el control de inventarios en una empresa facilitadora

- de pagos electrónicos. Repositorio general .
<https://repository.unimilitar.edu.co/handle/10654/20721?show=full>
- Department of Electrical Engineering, B. (2019). Ultralightweight RFID Authentication Protocols for. Hindawi.<https://www.hindawi.com/journals/scn/2019/3295616/>.
- Souza, W. (2022). UTILIZAÇÃO DA ETIQUETA RFID NA GESTÃO DO PATRIMÔNIO PÚBLICO: O. Araguaina.Repositorio general de la universidad Federal do Tocantins .<https://repositorio.uft.edu.br/bitstream/11612/4569/1/ISAAC%20ALVES%20DIAS%20-%20TCC%20-%20LOG%c3%8dSTICA.pdf>
- Abdias, A. (2022). Tecnología arduino con Extreme Programing para la incubación artificial de huevos de pollos criollos con monitoreo web socket en la Cooperativa de Ahorro y Crédito Neuroventas, 2019. Repositorio general de la Universidad Puclla Perú .http://repositorio.unu.edu.pe/bitstream/handle/UNU/5587/B8_2022_UNU_SISTEMAS_2022_T_ABDIAS_ESPIRITU_CESAR_PASTOR_V1.pdf?sequence=1&isAlloweaaad=y
- Francisco, P.(2022). Implementación de funcionalidad de dispositivo de almacenamiento masivo USB en MicroPython como solución para almacenar en memoria flash el bring-up software en ECU's.I Repositorio Institucional de CIATEQ Digital.
<https://ciateq.repositorioinstitucional.mx/jspui/bitstream/1020/473/1/QuirartePelayoFranciscoJ%20MSIM%202020.pdf>.
- Intriago, P. (2018). Analizador de protocolo SPI .Repositorio general de la Espol.
<https://www.dspace.espol.edu.ec/handle/123456789/42249>.
- Oviedo, B. (2019). Visualizador de tráfico de red de comunicación basado en la Arquitectura TCP/IP. Universidad y Sociedad vol.11no.2.http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202019000200193
- Rodríguez, S. (2019). Un recorrido por la técnica de la entrevista en la recolección de datos cuantitativos. diagramación y compilación, 117.

<http://uba.edu.ve/wp-content/uploads/2021/03/6.JORNADAS-2019.pdf#page=117>

Lopezosa, C.(2020) . Entrevistas semiestructuradas con NVivo: pasos para un análisis cualitativo eficaz.<https://repositori.upf.edu/handle/10230/44605>

Rodríguez, A .(2021). Estándares IEEE sobre el análisis y diseño de software en proyectos de control de información. <https://revistas.uvp.mx/index.php/nextia/article/view/89>

Vega, A .(2019). Método basado en la programación por capas para generar código automático desde el diagrama de clases. Revista Peruana de Computación y Sistemas 2019 2(2):25-42 .<https://revistasinvestigacion.unmsm.edu.pe/index.php/rpcsis/article/download/17015/14753/61315>

Garrido, A .(2021). Casos de uso y diagramas de casos de uso. Repositorio general Universidad Politécnica de Valencia.<https://riunet.upv.es/handle/10251/167637>

ANEXOS

ANEXO 1. ELICITACIÓN DE REQUERIMIENTOS

Código de requisito	<u>RF001</u>
Nombre de requisito	Asignación de equipos
Tipo	<u>Requisitos de producto</u> Requisitos de proyecto
Prioridad del requisito	<u>Alta/Esencial</u> Media/Deseado Baja/Opcional
Descripción	El sistema debe permitir asignar a los equipos un código RFID
Proceso	Cuando se establezca una etiqueta para el equipo el sistema debe pedir el serial de dicha pc para hacer la asignación
Entradas	Serial del equipo
Salidas	*Mensaje de error en el caso de problemas al almacenar información *Guardado exitoso, *Edición exitosa
Restricciones	Solo podrá acceder el usuario administrador

Anexo 1: Asignación de equipos.

Código de requisito	<u>RF002</u>
Nombre de requisito	Lectura de equipos con el uso del lector RFID
Tipo	<u>Requisitos de producto</u> Requisitos de proyecto
Prioridad del requisito	<u>Alta/Esencial</u> Media/Deseado Baja/Opcional
Descripción	El sistema debe ser capaz de leer los objetos dentro de su rango haciendo uso del lector RFID
Proceso	Cuando se requiera mostrar los equipos de un laboratorio se deberá hacer uso del lector RFID una vez hecha la búsqueda se deberá mostrar que el registro se tomó con éxito
Entradas	Equipos identificados por el lector RFID
Salidas	*Mensaje de error en el caso de no recibir información del lector *Lectura exitosa
Restricciones	no aplica.

Anexo 2: Lectura de equipos con el uso del lector RFID

Código de requisito	<u>RF003</u>
Nombre de requisito	No eliminar ningún registro
Tipo	<u>Requisitos de producto</u> Requisitos de proyecto
Prioridad del requisito	<u>Alta/Esencial</u> Media/Deseado Baja/Opcional

Descripción	El sistema no debe eliminar ningún registro de los objetos
Proceso	El usuario no debe tener habilitada la opción de eliminar cualquier registro del sistema
Entradas	Acción de usuario
Salidas	*Mensaje de advertencia notificando que no se puede eliminar el registro
Restricciones	No aplica.

Anexo 3: No eliminar ningún registro

Código de requisito	<u>RF004</u>
Nombre de requisito	Reportes
Tipo	<u>Requisitos de producto</u> Requisitos de proyecto
Prioridad del requisito	<u>Alta/Eencial</u> Media/Deseado Baja/Opcional
Descripción	El sistema debe ser capaz de generar reportes, los cuales contarán con información importante y detallada de los objetos dentro del laboratorio
Proceso	Cuando se cree un reporte el mismo deberá contar con información importante de los equipos como lo son el tiempo de vida útil de los objetos, si se encuentra en el lugar correspondiente, además mostrar el custodio de los objetos
Entradas	Equipos identificados en el sistema
Salidas	*Mensaje de error en el caso de no poder generar el reporte *Generación de reporte exitoso
Restricciones	no aplica.

Anexo 4: Reportes.

Código de requisito	<u>RF005</u>
Nombre de requisito	Interfaz simple y de fácil comprensión
Tipo	<u>Requisitos de producto</u> Requisitos de proyecto
Prioridad del requisito	<u>Alta/Eencial</u> Media/Deseado Baja/Opcional
Descripción	El diseño del sistema debe ser intuitivo de fácil comprensión no debe incluir opciones no deseadas
Proceso	Se deben colocar solo las opciones tratadas
Entradas	Ninguna
Salidas	Diseño sistema

Tipo	Requisitos de producto Requisitos de proyecto
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe permitir ingresar, editar y eliminar la información de los usuarios, además de permitir que se le puedan asignar laboratorios
Proceso	Debe permitir guardar el usuario sin la necesidad de asignar un laboratorio y al momento de eliminar debe colocar el laboratorio asignado a otro docente o en estado en espera de asignación
Entradas	Información de los laboratorios, información de docente encargado
Salidas	*Mensaje de error en el caso problemas al almacenar información *Guardado exitoso, *Edición exitosa, *Eliminación exitosa
Restricciones	Solo podrá acceder el usuario administrador

Anexo 8: Ingreso, edición, eliminación y asignación de los usuarios. Fuente: Los Autores

Código de requisito	RF009
Nombre de requisito	Ingreso y edición de los equipos
Tipo	Requisitos de producto Requisitos de proyecto
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe permitir ingresar y editar la información de los equipos
Proceso	Cuando se ingresen los datos correspondientes para las etiquetas de los objetos, el sistema debe permitir asignar dichos datos
Entradas	Información de los equipos
Salidas	*Mensaje de error en el caso problemas al almacenar información *Guardado exitoso, *Edición exitosa
Restricciones	Solo podrá acceder el usuario administrador

Anexo 9: Ingreso y edición de los equipos. Fuente: Los Autores

Código de requisito	RF010
Nombre de requisito	Validación de tipos de usuarios
Tipo	Requisitos de producto Requisitos de proyecto
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe tener claro los diferentes roles de usuarios del sistema

Proceso	Cuando acceda a la aplicación debe identificar el tipo de usuario para poder restringir y permitir accesos a las diferentes opciones del sistema
Entradas	Credenciales
Salidas	*Mensaje de advertencia diciendo que no cuenta con el acceso necesario
Restricciones	no aplica.

Anexo 9: Validación tipo usuarios. Fuente: Los Autores

ANEXO 2. REQUISITOS DE DISEÑO DEL SISTEMA

Código de requisito	<u>RD001</u>
Nombre de requisito	Responsivo
Prioridad del requisito	<u>Alta/Esencial</u> Media/Deseado Baja/Opcional
Descripción	La interfaz debe ser capaz de poder verse bien tanto en dispositivos móviles como en ordenadores

Anexo 10: Responsivo

Código de requisito	<u>RD002</u>
Nombre de requisito	Intuitivo
Prioridad del requisito	<u>Alta/Esencial</u> Media/Deseado Baja/Opcional
Descripción	El sistema debe ser capaz de manejarse sin mucha complejidad

Anexo 11: Intuitivo.

Código de requisito	<u>RD003</u>
Nombre de requisito	Minimalista
Prioridad del requisito	<u>Alta/Esencial</u> Media/Deseado Baja/Opcional
Descripción	Las opciones del sistema no deben ser demasiadas solo las planteadas para mantener un estilo minimalista y de fácil comprensión

Anexo 12: Minimalista.

Código de requisito	<u>RD004</u>
---------------------	---------------------

Nombre de requisito	Paleta de color del sistema
Prioridad del requisito	Alta/Esencial <u>Media/Deseado</u> Baja/Opcional
Descripción	La paleta de colores del sistema se debe decantar por colores pasteles los cuales no sean fuerte a la vista

Anexo 13: Paleta de color del sistema.

Código de requisito	<u>RD005</u>
Nombre de requisito	Fuente de texto
Prioridad del requisito	Alta/Esencial <u>Media/Deseado</u> Baja/Opcional
Descripción	Las fuentes del texto deben ser comprensibles

Anexo 14: Fuente de texto.

ANEXO 3. REVISIÓN BIBLIOGRÁFICA PROTOCOLOS DE COMUNICACIÓN.

Estudio de los protocolos SPI, USB Y HTTP para su implementación en un sistema de inventario RFID: Revisión bibliográfica

Resumen

La siguiente revisión bibliográfica se centra en analizar los protocolos de comunicación SPI, USB y HTTP para su aplicación en un sistema de inventario RFID. Se realizó una investigación exhaustiva que abarcó un total de 20 artículos, con el objetivo de identificar las ventajas y desventajas de cada protocolo. Los hallazgos obtenidos permiten una selección informada del protocolo de comunicación más adecuado, teniendo en cuenta las necesidades específicas del sistema.

Palabras claves: Protocolo de comunicación, SPI, USB, HTTP, Sistema Inventario

Abstract

The following bibliographical review focuses on analyzing the SPI, USB and HTTP communication protocols for their application in an RFID inventory system. An exhaustive investigation was carried out that included a total of 20 articles, with the objective of identifying the advantages and disadvantages of each protocol. The findings obtained allow an informed selection of the most appropriate communication protocol, taking into account the specific needs of the system.

Keywords: Communication protocol, SPI, USB, HTTP, Inventory System

Introducción

En el área correspondiente al desarrollo de software, la elección de protocolos de comunicación adecuados desempeña un papel fundamental para garantizar un buen funcionamiento de las aplicaciones (David 2019). Según (Quiñones 2020) las necesidades y requisitos de cada sistema, definen que protocolos pueden ser utilizados para permitir la transferencia de datos de manera eficiente y confiable. En el ámbito actual se vuelve necesario realizar un análisis bibliográfico de los protocolos SPI, USB Y HTTP el cual nos ayude a identificar correctamente sus ventajas en la aplicación de nuestro sistema.

Esta revisión bibliográfica tiene como objetivo proporcionar una base sólida de conocimiento sobre los protocolos SPI, USB y HTTP, y cómo se aplican en el desarrollo de sistemas de inventario RFID. Al entender las particularidades de cada protocolo, podremos aprovechar al máximo sus beneficios y asegurar un funcionamiento eficiente y confiable de nuestro sistema.

Materiales y métodos

Para el desarrollo de este proyecto, se llevó a cabo una investigación exhaustiva para comprender las utilidades de cada protocolo, tanto por separado como en conjunto. Con el objetivo de obtener una visión completa y actualizada, se revisaron un total de 20 artículos.

Estos artículos seleccionados han proporcionado información detallada y precisa sobre los protocolos en cuestión, abordando aspectos clave como su funcionamiento, características distintivas, ventajas y limitaciones. La inclusión de estos artículos ha permitido fundamentar de manera sólida y respaldar nuestras conclusiones y recomendaciones en este proyecto de desarrollo.

Resultados

Los resultados obtenidos revelaron una serie de ventajas y desventajas asociadas a cada uno de los protocolos. Para una mayor claridad y comprensión, se presentarán las tablas correspondientes que resumen dichos resultados,

detallando las ventajas y desventajas identificadas para el protocolo SPI, USB y HTTP, respectivamente.

Protocolo SPI:

Ventajas	Desventajas
Alta velocidad de transferencia de datos	Limitaciones en la distancia de transmisión
Simplicidad en su implementación	Requiere cableado específico entre dispositivos
Comunicación directa y rápida	No es adecuado para aplicaciones bidireccionales
Bajo consumo de energía	
Ampliamente utilizado en aplicaciones de bajo nivel	

Protocolo USB:

Ventajas	Desventajas
Alta versatilidad y compatibilidad	Mayor complejidad en la implementación
Conexión en caliente y detección automática de dispositivos	Limitaciones en la distancia de transmisión
Comunicación directa y rápida	
Transferencia de datos y suministro de energía simultáneamente	
Interfaz estándar y fácil de usar	

Protocolo HTTP:

Ventajas	Desventajas
-----------------	--------------------

Ampliamente utilizado en aplicaciones	No es adecuado para aplicaciones de baja latencia o tiempo real
Compatibilidad con diferentes dispositivos y sistemas operativos	Limitaciones en la transferencia de grandes volúmenes de datos
Interfaz intuitiva y bien documentada	Posibles problemas de seguridad y privacidad en entornos no cifrados

Conclusión

Tras realizar una exhaustiva revisión bibliográfica sobre los protocolos de comunicación SPI, USB y HTTP, se pueden extraer conclusiones significativas sobre las ventajas y desventajas de cada uno de ellos para su implementación.

En primer lugar, el protocolo SPI se destaca por su alta velocidad de transferencia de datos, lo que lo hace especialmente adecuado para aplicaciones que requieren una rápida transmisión de información. Su implementación es relativamente sencilla y su bajo consumo de energía lo convierte en una opción eficiente en términos de consumo. Sin embargo, se deben tener en cuenta las limitaciones en la distancia de transmisión y su idoneidad para aplicaciones más complejas que requieran comunicación bidireccional.

Por otro lado, el protocolo USB ofrece una gran versatilidad y compatibilidad con una amplia variedad de dispositivos y sistemas operativos. Su capacidad de conexión en caliente y detección automática de dispositivos lo convierten en una opción conveniente en entornos donde se requiere una fácil conectividad. No obstante, se debe considerar un mayor consumo de energía.

En cuanto al protocolo HTTP, es ampliamente utilizado en aplicaciones web y ofrece compatibilidad con diversos dispositivos y sistemas operativos. Su interfaz intuitiva y bien documentada facilita la comunicación a través de solicitudes y respuestas. Sin embargo, es importante tener en cuenta que puede presentar limitaciones en la transferencia de grandes volúmenes de datos, y pueden surgir preocupaciones de seguridad y privacidad en entornos no cifrados.

Bibliografía

- Auccasi, M (2021). Diseño de una red inalámbrica para el servicio de pesaje en el sector agroindustrial utilizando Balanzas Super SS mediante protocolo SPI basado en microcontrolador ESP32. Repositorio institucional Universidad Ricardo Palma. <https://repositorio.urp.edu.pe/handle/20.500.14138/4851>
- González, J (2022). Diseño de un IP CORE para una interfaz SPI. Pistas Educativas, No. 137, n. <https://pistaseducativas.celaya.tecnm.mx/index.php/pistas/article/view/2276>
- Garrido, F (2022). Expansor de puertos con protocolos de comunicación SPI en FPGA. Repositorio General Universidad Politécnica de Madrid. <https://oa.upm.es/72677/>
- Luise, R (2019). Sistema Protocolo Integrado (SPI) nas universidades federais Brasileiras. Memorias del xiii congreso de archivología del MERCOSUR 21 - 25 de octubre de 2019. Montevideo – uruguay (Pag 294). https://d1wqtxts1xzle7.cloudfront.net/70437895/Memorias_XIII_CAM-libre.pdf?1632885432=&response-content-disposition=inline%3B+filename%3DMemorias_XIII_Congreso_de_Archivologia_d.pdf&Expires=1686521336&Signature=W~sytFwnzHRUxtR5Ac iR05GCrTmOk-GK5fZkqY32dmwP2s6-fJK80JeZHSTnrYARlwjV1elcCSFvxKpcgtZsyqWC1TL2gGMJcbnMLbyai PPorGPfE4m5KIXgcEa84tziUPsztuK8iqCfctryzd4PbzXZlINV76m~ePI4W bUPHtWtb1~5gla-aL8LznlZnrjuPlx1ebLTGGF-2NRLs40N1DdKXizwsy5QIKR8BoUZzZw-0ZBuuDtJx6l2DR937KJYdKMZcln9ZirAN-tlwAv2~4GWaRE-10yhayCMpFKuH8Jy8CBT1rds~cKbGp~Xzvqu0xallcxcBslzergwhiG5-Q__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA#page=936
- Tamayo, J (2022). Desarrollo e instalación del sistema "SADM-1.2" para su validación a bordo de la misión suborbital CSM-2018-A. Repositorio

General de la Escuela Superior de Mecánica y Eléctrica unidad Culhuacan. <https://tesis.ipn.mx/handle/123456789/31133>

Zambrano, A (2021). Sistema ambulatorio de monitorización de crisis epilépticas. Repositorio General Universidad Corl. <https://ruc.udc.es/dspace/handle/2183/28293>.

Ortega, P (2022). Desarrollo de un sistema USB para el aprendizaje y la implementación en dispositivos programables PIC18. Repositorio General Universidad UCCI. <https://repositorio.ecci.edu.co/handle/001/2921>.

Moreno, P (2022). Hackeando computadores por el puerto USB Hacking computers by USB ports. Repositorio General de la universidad Complutense de Madrid. <https://eprints.ucm.es/id/eprint/75027/>.

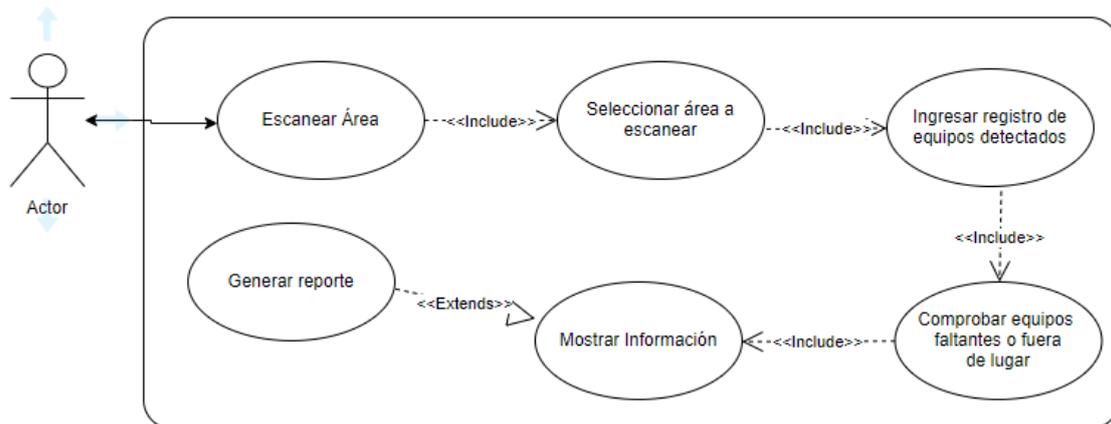
Cruz, J (2019). Evaluación técnica de implementación de tarjeta Arduino para interconexión con termopares y manómetros. https://d1wqtxts1xzle7.cloudfront.net/59035078/NolascoCruzTA22120190425-19586-1wc8z49-libre.pdf?1556256961=&response-content-disposition=inline%3B+filename%3DEVALUACION_TECNICA_DE_IMPLEMENTACION_DE.pdf&Expires=1686522278&Signature=arwEcWMwjLD5WvinXs5h9C8pG5ypk7sEfMe7PF4RE39TgePBMXsNycQ0CqqjfJWNL7IHKMPVpyumXM6MBe3DdoXXteuDAF~OyKTXtrvt4W6UR2x6eHKZt0v4K9a61VkB40wv0M26OisnxxaCY4uM9u4jMaX8d1nSn4JliKnTjoCz4eTxQvBIZ1M-CaGLWuj6MtYOkxYCp6KNINZbNG-r70tvU0FrE80HbM1azB7hHgryzjSy2JEjMOy2gM31riFK51erZVY70c~SK1Z1o9q0JRO1Q1BK4HJWKGb~rcNdFpzSmEczsN3ZoEin5LMCc1vfZC9SmVj5Mu4PAIfAggw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

Doallo, M (2021). Modulador por amplitud de pulso implementado sobre FPGA con software libre. <https://ieeexplore.ieee.org/abstract/document/9505335>.

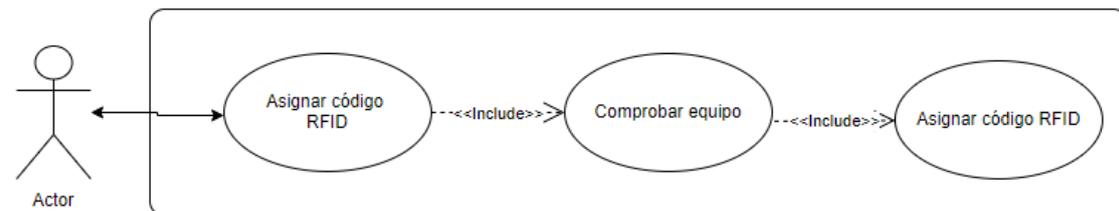
Nivio, Guillan (2022). Desarrollo de un receptor de telemetría aeronáutica basado en Radio Definida por Software. Repositorio Genral Itema. <http://rinfi.fi.mdp.edu.ar/handle/123456789/700>

- Quispe, A (2022). Sistema de adquisición de datos para la determinación de la resistencia al esfuerzo cortante del suelo – caso de estudio: implementación de un dispositivo USB-HID para un equipo de corte de la empresa Labomat. Repositorio General de la universidad mayor de San Andrés. <https://repositorio.umsa.bo/handle/123456789/30936>
- Ana, C (2022). Las mejores prácticas para la implementación de un API REST. Repositorio General Universidad Mayor de San Simón. <http://ddigital.umss.edu.bo:8080/jspui/handle/123456789/33404>
- Pires, F (2021). Api (APPLICATION PROGRAMMING INTERFACE) de sistemas de avaliação de atendimento. <http://www.jornacitec.fatecbt.edu.br/index.php/XJTC/XJTC/paper/view/2454>
- Berves, D (2022). API para el desarrollo de aplicaciones IoT personalizadas usando FIWARE. <https://rctd.uic.cu/rctd/article/view/132>
- Millan, C (2019). Security in API and API managers. Repositorio Feneral de la Universitat Oberta de Catalunya. <https://openaccess.uoc.edu/handle/10609/95668>
- Batista, D (2022). Evaluación de la Integración del Protocolo MQTT en una Plataforma de Ciudades Inteligentes. <https://sol.sbc.org.br/journals/index.php/reic/article/view/2221>
- Ruiz, M (2019). Servicio Web API REST sobre el Framework Spring, Hibernate, JSON Web Token y BBDD Oracle. Repositorio de la Universidad de Sevilla. <https://idus.us.es/handle/11441/89487>
- David, J (2019). Desarrollo de una herramienta de software para la gestión del mantenimiento de infraestructura en el SENA regional Antioquia. Repositorio general Cintex. <https://revistas.pascualbravo.edu.co/index.php/cintex/article/view/331>.
- Quiñones, M (2020) Desarrollo y evaluación de un gateway móvil IoT para redes 4G LTE. Enfoque UTE vol.11 no.4 Quito oct./dic. 2020. http://scielo.senescyt.gob.ec/scielo.php?pid=S1390-65422020000400016&script=sci_arttext

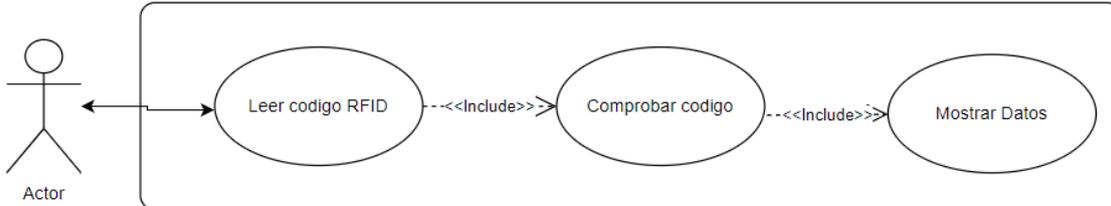
ANEXO 4. DIAGRAMAS DE CASO DE USO



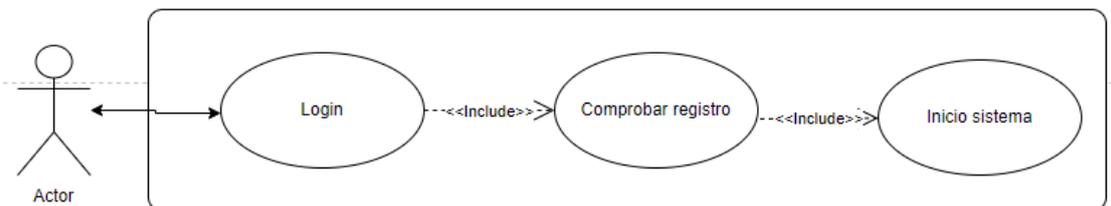
Anexo 15: Diagrama de flujo escaneo de área



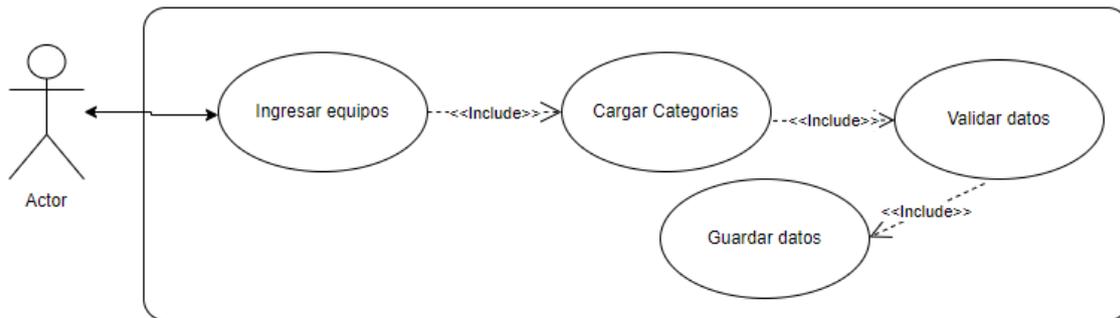
Anexo 16: Diagrama de flujo asignación Rfid



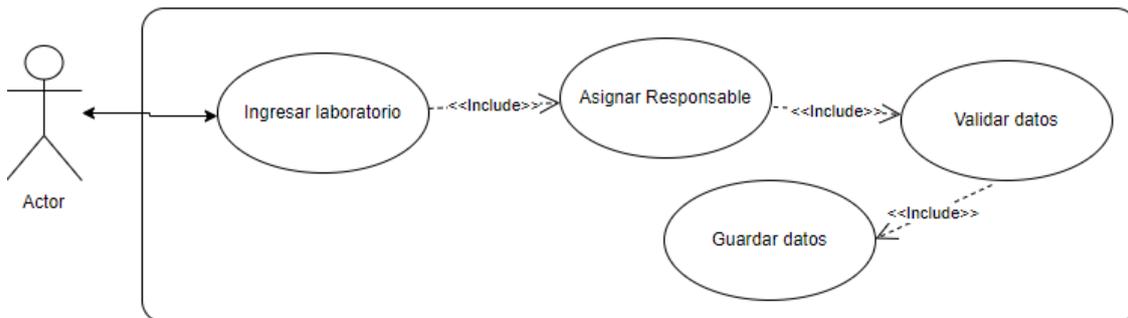
Anexo 17: Diagrama de flujo Lectura de códigos



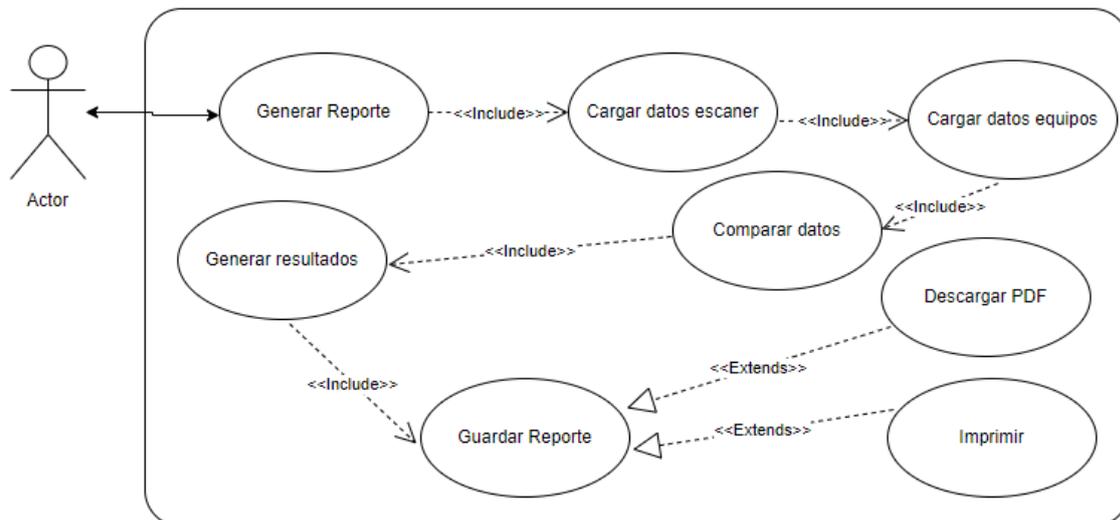
Anexo 18: Diagrama de flujo Lectura de códigos



Anexo 19:Diagrama de flujo Lectura de códigos



Anexo 20:Diagrama de flujo Lectura de códigos



Anexo 21:Diagrama de flujo Lectura de códigos

ANEXO 5. DESARROLLO DE LAS VISTAS DEL APLICATIVO WEB



Jonathan Ormaza

Usuarios Agregar

Mostrando registros del 1 al 1 de un total de 1 registros

Alias	Nombres	Apellidos	Correo	Rol	Estado	Acciones
Chema	Jonathan Geovanny	Ormaza Calderon	jonathan.ormaza@espam.edu.ec	Admin	Activo	 

Mostrar registros Anterior Siguiente Buscar:

2022 © ESPAM MFL

Anexo 22: Menu usuario



Jonathan Ormaza

Usuarios Agregar

Mostrando registros del 1 al 1 de un total de 1 registros

Alias	Nombres	Apellidos	Correo	Rol	Estado	Acciones
Chema	Jonathan Geovanny	Ormaza Calderon	jonathan.ormaza@espam.edu.ec	Admin	Activo	 

Mostrar registros Anterior Siguiente Buscar:

2022 © ESPAM MFL

Agregar un nuevo usuario

NOMBRES

APELLIDOS

ALIAS

CORREO

CLAVE

ROL

Anexo 23: Agregar usuario



Jonathan Ormaza

Usuarios Agregar

Mostrando registros del 1 al 1 de un total de 1 registros

Alias	Nombres	Apellidos	Correo	Rol	Estado	Acciones
Chema	Jonathan Geovanny	Ormaza Calderon	jonathan.ormaza@espam.edu.ec	Admin	Activo	 

Mostrar registros Anterior Siguiente Buscar:

2022 © ESPAM MFL

Editar usuario

NOMBRES

APELLIDOS

ALIAS

ROL

Anexo 24: Editar usuario



Jonathan Ormaza

Laboratorios Agregar

Mostrando registros del 1 al 2 de un total de 2 registros

Nombre	Referencia	Responsable	Estado	Acciones
201	Planta Baja	Jonathan Geovanny Ormaza Calderon	Activo	+ -
202	Planta Baja	Sandro Antonio Palau Delgado	Activo	+ -

Mostrar 5 registros Anterior 1 Siguiente Buscar:

2022 © ESPAM MFL

- 🏠 Inicio
- 👤 Usuarios
- 🏢 Laboratorios
- 🖨️ Equipos
- 🔑 Roles
- 🔍 Verificar
- 📄 Reporte

Anexo 25: Editar usuario



Jonathan Ormaza

Equipos Agregar

Mostrando registros del 1 al 5 de un total de 6 registros (filtrado de un total de 11 registros)

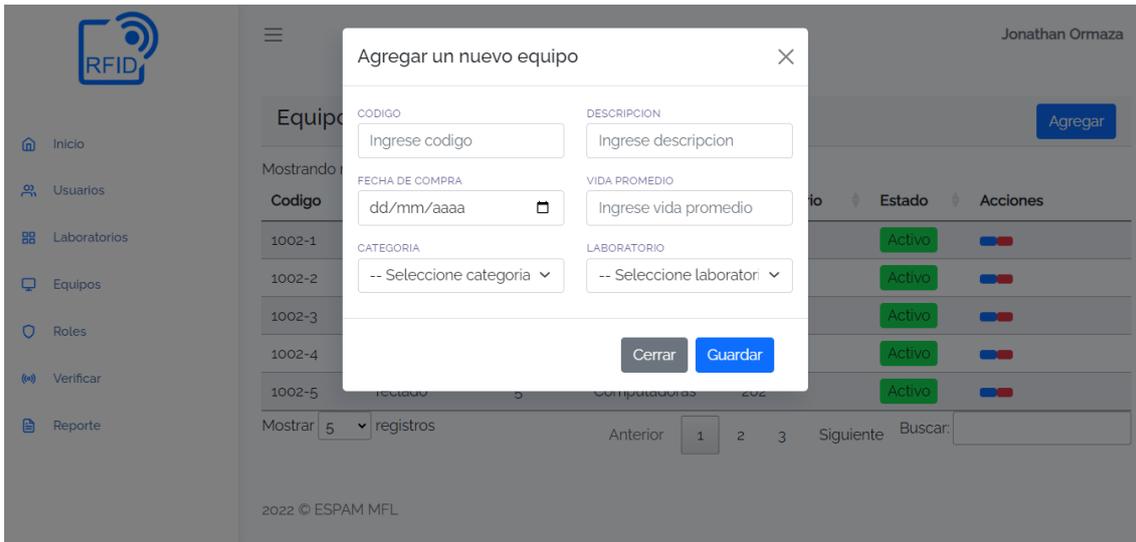
Codigo	Descripcion	Vida	Categoria	Laboratorio	Estado	Acciones
1002-1	Mause	5	Computadoras	202	Activo	+ -
1002-2	Teclado	5	Computadoras	202	Activo	+ -
1002-3	Teclado	5	Computadoras	202	Activo	+ -
1002-4	Teclado	5	Computadoras	202	Activo	+ -
1002-5	Teclado	5	Computadoras	202	Activo	+ -

Mostrar 5 registros Anterior 1 2 Siguiente Buscar:

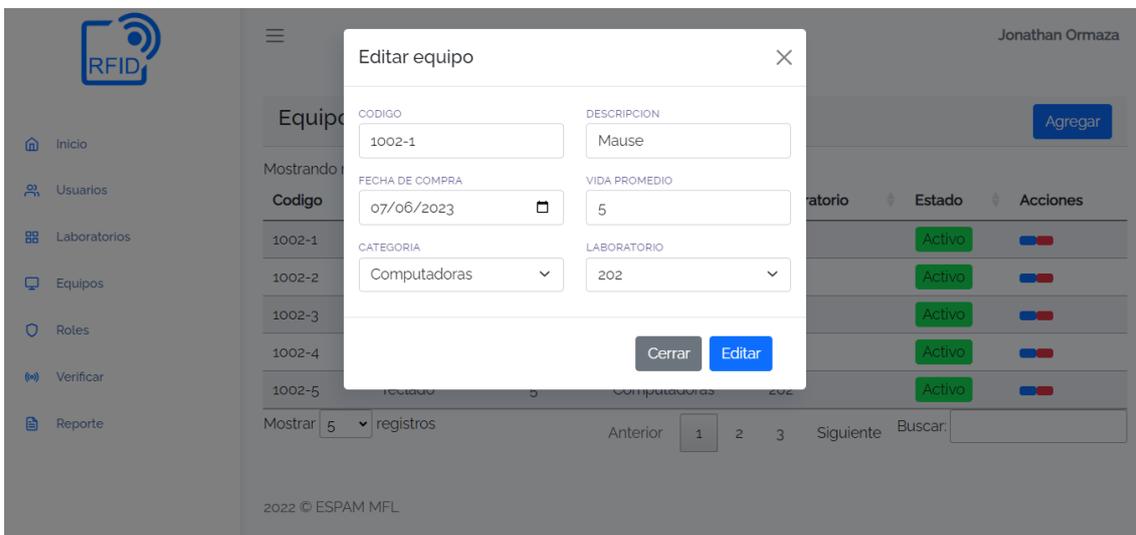
2022 © ESPAM MFL

- 🏠 Inicio
- 👤 Usuarios
- 🏢 Laboratorios
- 🖨️ Equipos
- 🔑 Roles
- 🔍 Verificar
- 📄 Reporte

Anexo 26: Menú equipos



Anexo 27: Agregar equipo



Anexo 28: Editar Equipo



- [Inicio](#)
- [Usuarios](#)
- [Laboratorios](#)
- [Equipos](#)
- [Roles](#)
- [Verificar](#)
- [Reporte](#)

☰
Jonathan Ormaza

Verificar

LABORATORIOS

▼
Generar

Codigo	Descripcion	Laboratorio	Estado
Ningún dato disponible en esta tabla			

2022 © ESPAM MFL

Anexo 29: Verificar Laboratorio



- [Inicio](#)
- [Usuarios](#)
- [Laboratorios](#)
- [Equipos](#)
- [Roles](#)
- [Verificar](#)
- [Reporte](#)

☰
Jonathan Ormaza

Verificar

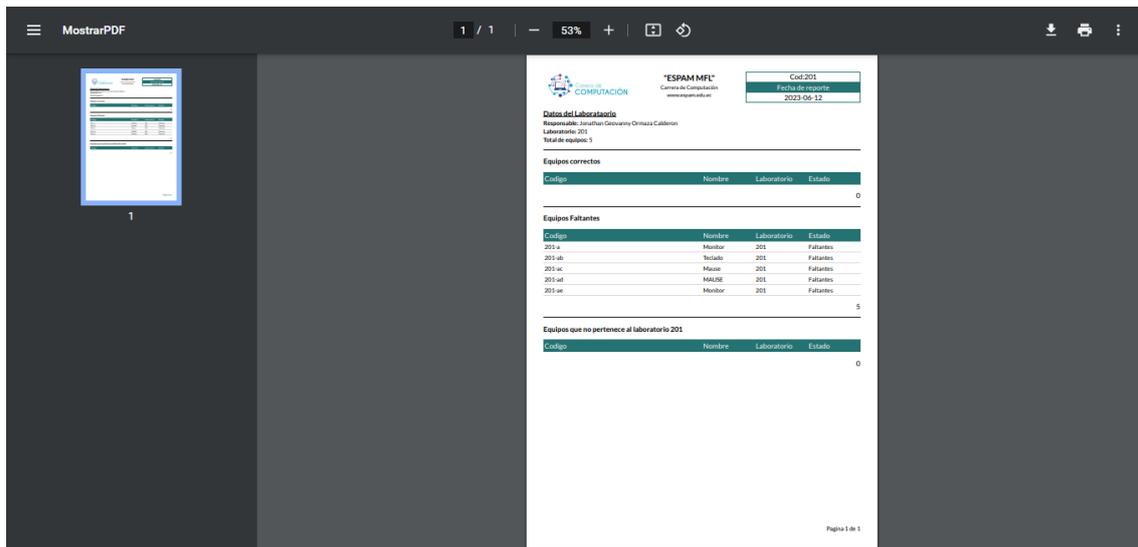
LABORATORIOS

▼
Generar

Codigo	Descripcion	Laboratorio	Estado
1002-1	Mause	202	Completo
1002-2	Teclado	202	Faltantes
1002-3	Teclado	202	Faltantes
1002-4	Teclado	202	Faltantes
1002-5	Teclado	202	Faltantes
1002-6	Mause	202	Faltantes

2022 © ESPAM MFL

Anexo 30: Registro verificación



Anexo 31 :Reporte generado

ANEXO 6. CÓDIGO ARDUINO UNO

```
//Se importan las librerías
#include <SPI.h>
#include <MFRC522.h>

// Pin 9 para el reset del RC522
#define RST_PIN 9

// Pin 10 para el SS (SDA) del RC522
#define SS_PIN_ENB 10

// Creamos el objeto para el RC522
MFRC522 MyLectorRF(SS_PIN_ENB, RST_PIN);

// String para almacenar el código RFID
String rfidCode = "";

// Iniciamos la comunicación serial
void setup() {
  Serial.begin(9600);

// Iniciamos el Bus SPI
  SPI.begin();

// Iniciamos el MyLectorRF
  MyLectorRF.PCD_Init();
  Serial.println("Control Inicializado ...");
```

```

}

/// Verificamos si se ha detectado alguna tarjeta
void loop() {
  if (MyLectorRF.PICC_IsNewCardPresent()) {

    /// Determinamos el código de la tarjeta
    if (MyLectorRF.PICC_ReadCardSerial()) {

      /// Recuperamos el ID de la tarjeta
      rfidCode = "";
      for (byte i = 0; i < MyLectorRF.uid.size; i++) {
        if (MyLectorRF.uid.uidByte[i] < 0x10) {
          rfidCode += "0";
        }
        rfidCode += String(MyLectorRF.uid.uidByte[i], HEX);
      }
      /// Imprimimos el código RFID sin espacios
      rfidCode.replace(" ", "");
      Serial.println(rfidCode);
      delay(100);

      /// Terminamos la lectura de la tarjeta actual
      MyLectorRF.PICC_HaltA();
    }
  }
}

```

ANEXO 6. DESARROLLO DE LAS VISTAS DEL APLICATIVO MÓVIL

9:27   

CÓDIGO RFID

05ae5183

Información adicional

Número Serie: 6024569

Detalle equipo:
Computadora i7 4gb ram 500 sdd

Área asignada:
Laboratorio 101 informatica

Responsable:
Ing. Yimmy Loor Vera

Fecha Ingreso:
17/04/2022

Vida útil programada :
48 MESES

 **Escanear área**

Asignar código **Leer Código**

Anexo 32:Diagrama de flujo Lectura de códigos

CÓDIGO RFID	
05ae5183	
Información adicional	
Número Serie:	<u>6024569</u>
Detalle equipo:	Computadora i7 4gb ram 500 sdd
Área asignada:	Laboratorio 101 informatica ▼
Responsable:	Ing. Yimmy Loor Vera
Fecha Ingreso:	17/04/2022
Vida útil programada :	48 MESES
Asignar N° Serie	
	Escanear área
Asignar código	Leer Código

Anexo 33:Diagrama de flujo Lectura de códigos

CÓDIGO RFID

05ae5183

Información adicional

Área a escanear:

Laboratorio 101 informática ▼

Numeros de equipos Identificados

5



Escanear área

Asignar código

Leer Código

Anexo 34: Diagrama de flujo Lectura de códigos

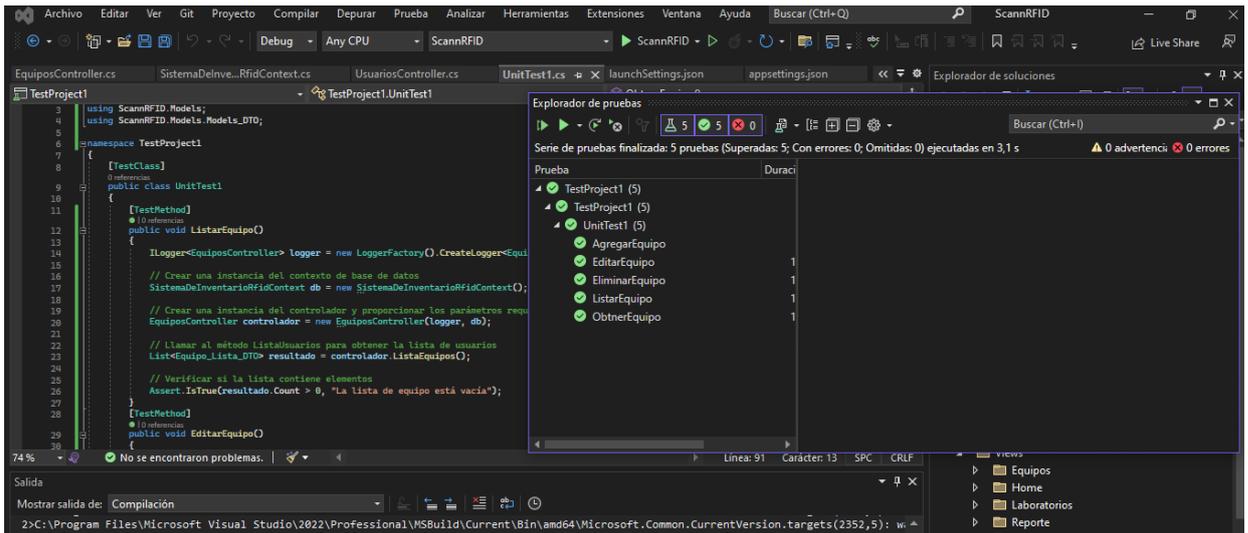
ANEXO 7. PRUEBAS UNITARIAS

Visual Studio interface showing unit tests for a user management system. The Explorer de pruebas window displays the following test results:

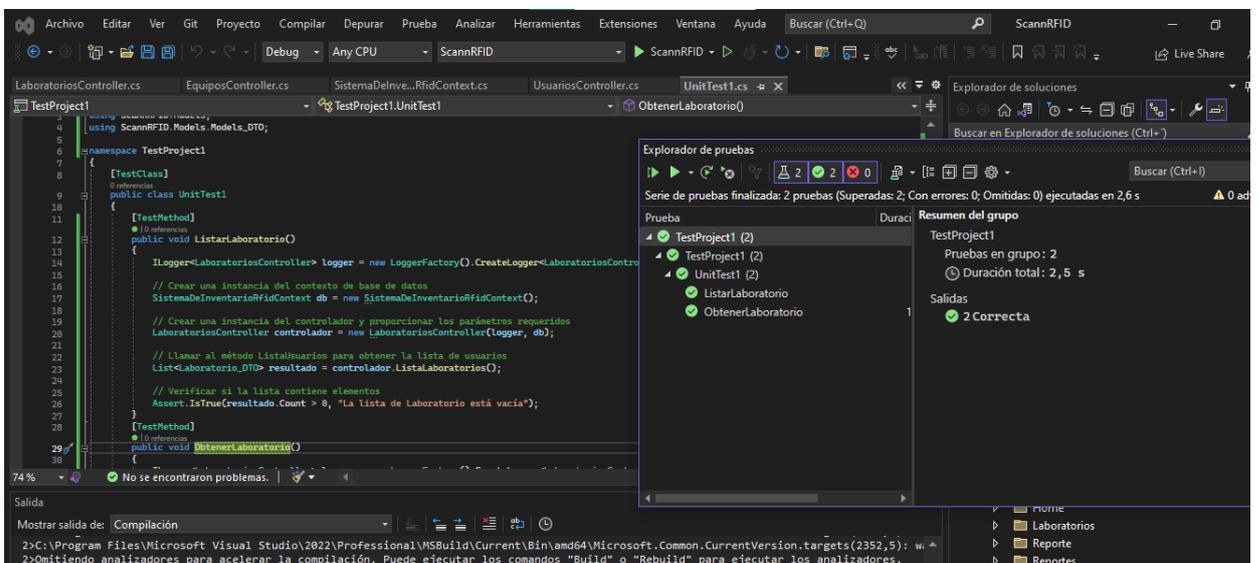
Prueba	Duración
TestProject1 (6)	
TestProject1 (6)	
UnitTest1 (6)	
AgregarUsuario	1
EditarUsuario	1
EliminarUsuario	1
ListarUsuario	1
ObtenerUsuario	1
VerificarLogin	1

Resumen de los detalles de la prueba:
 ListarUsuario
 Origen: UnitTest1.cs línea 12
 Duración: 116 ms

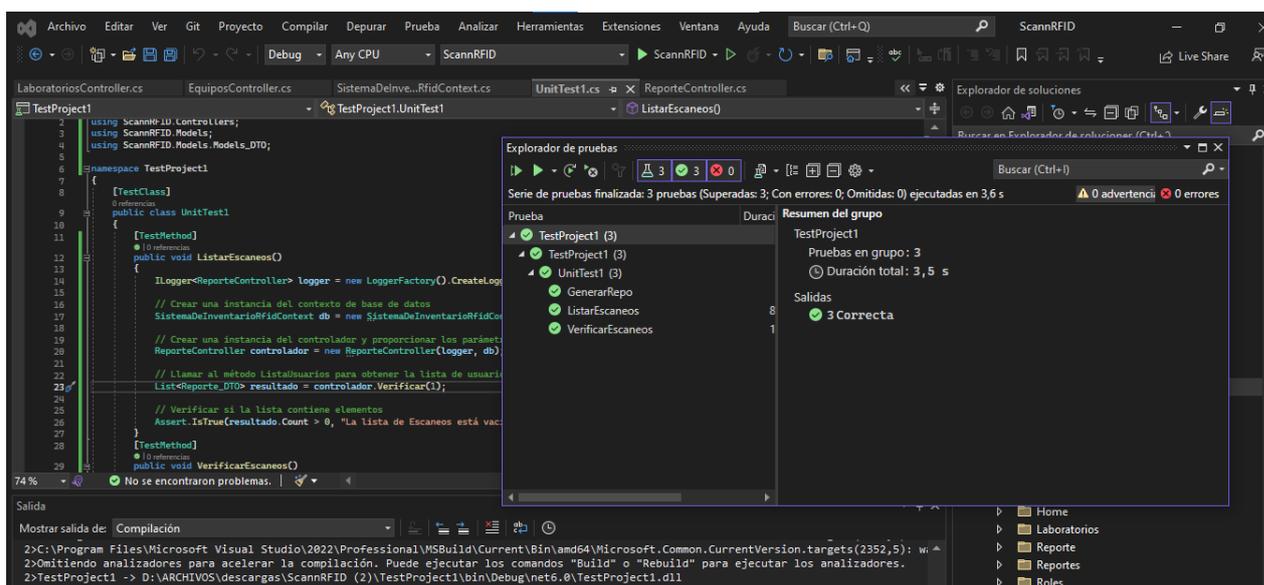
Anexo 35: Prueba Unitaria Usuarios



Anexo 36: Prueba Unitaria Equipos



Anexo 37: Prueba Unitaria Laboratorio



Anexo 38: Prueba Unitaria Escaneos y Reportesda

ANEXO 8. MANUAL DE PROGRAMADOR

Sistema inventario RFID

Manual de programador

Versión: 1.0

Fecha: JUNIO/2023

Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso

Contenido

1. INTRODUCCIÓN	58
2. OBJETIVO	58
3. REQUERIMIENTOS PARA IMPLEMTACIÓN	59
4. REQUERIMIENTOS PARA DESARROLLO	59
5. DESARROLLO DEL MANUAL	60

1. INTRODUCCIÓN

Este manual tiene como objetivo proporcionarte una guía completa y detallada para desarrollar y personalizar la aplicación de acuerdo con tus necesidades.

En este manual, encontrarás una amplia gama de información técnica y recursos que te ayudarán a comprender la arquitectura de la aplicación, las herramientas utilizadas, los flujos de trabajo recomendados y las mejores prácticas de desarrollo. Ya seas un desarrollador experimentado o estés dando tus primeros pasos en el desarrollo de aplicaciones, este manual está diseñado para brindarte el conocimiento necesario para maximizar el potencial de la aplicación de sistema de inventario RFID.

2. OBJETIVO

El objetivo principal de este Manual de Desarrollador es proporcionar a los programadores y desarrolladores una guía completa y detallada para

comprender y utilizar eficazmente las herramientas, técnicas y recursos necesarios en el desarrollo de la aplicación sistema de inventario RFID.

3. REQUERIMIENTOS PARA IMPLEMENTACIÓN

El sistema puede ser instalado en cualquier sistema operativo que cumpla con los

siguientes requerimientos:

- Gestor de base de datos Microsoft SQL Server
- Navegador web
- Servidor Web
- Android 8.0 o superior

4. REQUERIMIENTOS PARA DESARROLLO

Para la edición del sistema se necesita:

- Gestor de base de datos Microsoft SQL Server
- Visual studio
- Android studio
- Arduino ide

5. INSTALACIÓN Y CONFIGURACIÓN

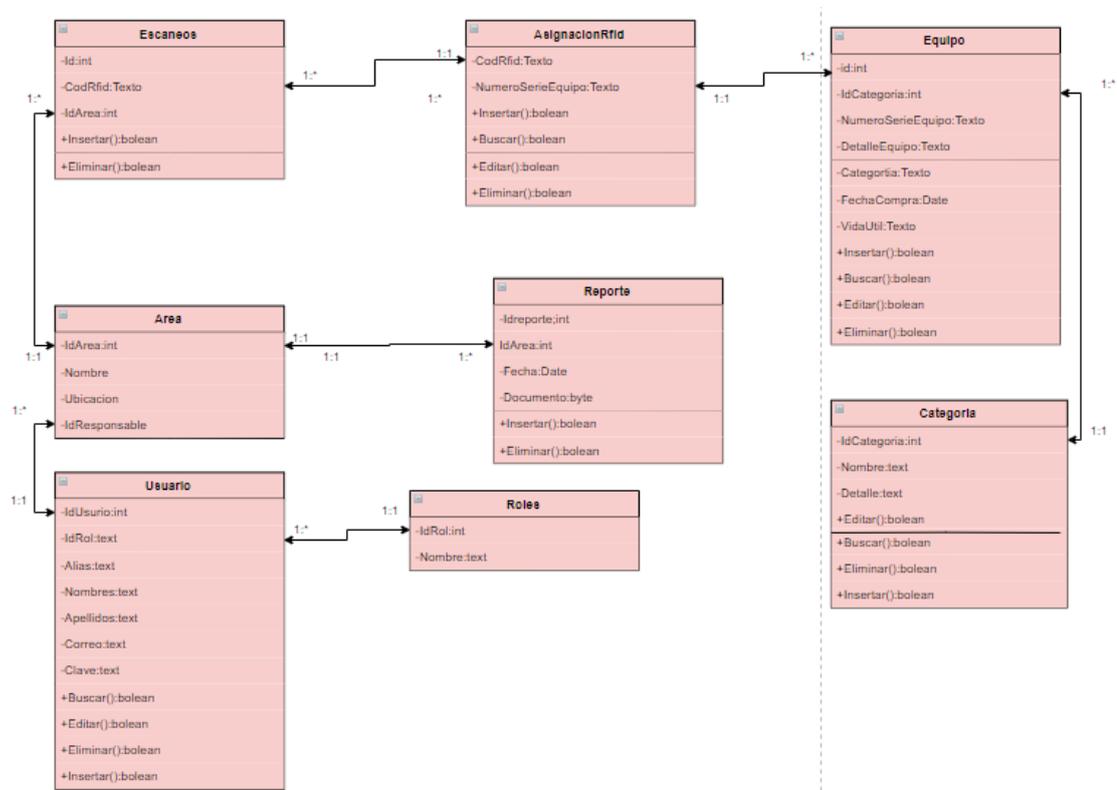
Para la instalación del sistema web, se requiere publicar la solución y subirla al servidor de destino, de manera similar a como se hace con cualquier otra aplicación web. Esto implica asegurarse de tener acceso al servidor y los privilegios necesarios para subir y configurar los archivos de la solución web. Una vez que los archivos están en el servidor, se pueden seguir los pasos habituales para configurar la base de datos, ajustar la configuración y garantizar que todos los componentes estén correctamente instalados y funcionando.

En cuanto a la aplicación Android, simplemente se necesita el archivo APK de instalación. Este archivo se puede obtener a través de diferentes métodos, como descargarlo desde una tienda de aplicaciones, recibirlo por correo electrónico o transferirlo directamente al dispositivo móvil. Una vez que se tiene el archivo APK, se puede ejecutar en el dispositivo Android y seguir las instrucciones de instalación para tener la aplicación funcionando en el dispositivo.

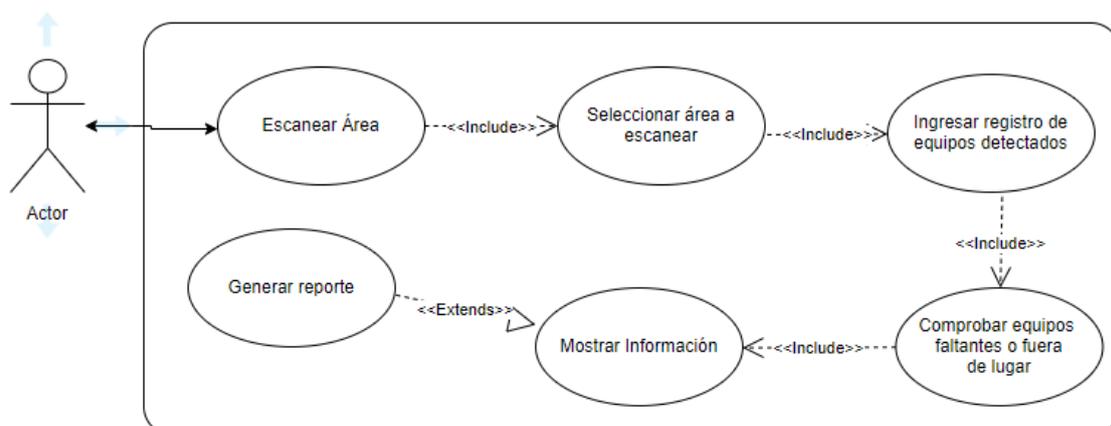
Recuerda que, tanto para la instalación del sistema web como para la aplicación Android, es importante tener en cuenta los requisitos del sistema y asegurarse de que se cumplan para garantizar un funcionamiento adecuado.

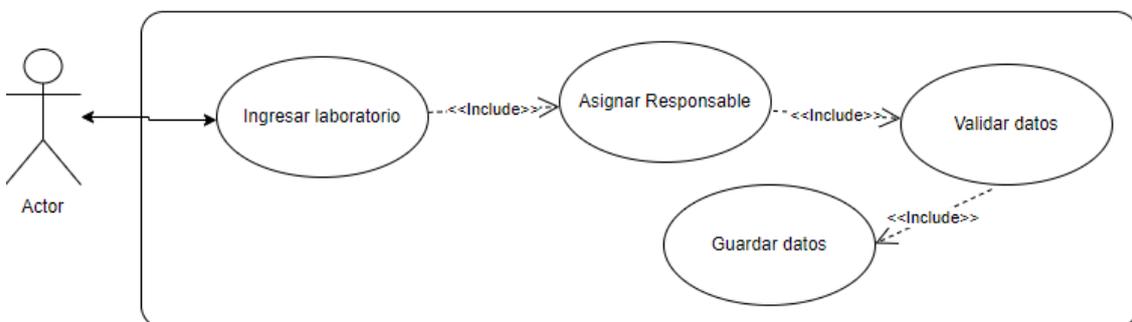
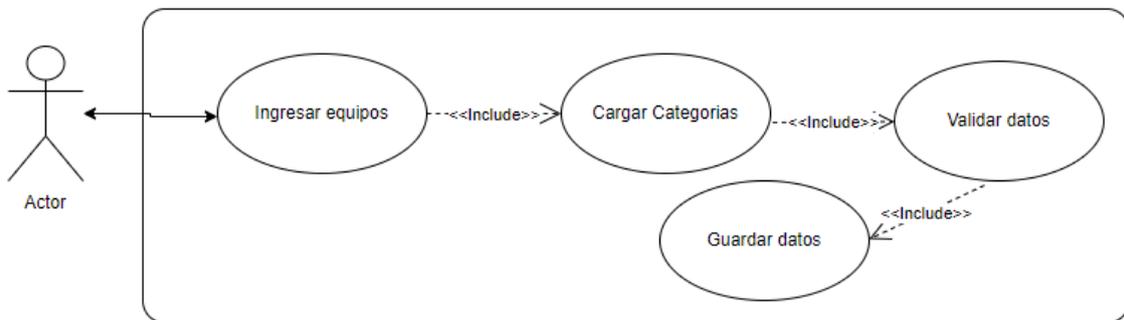
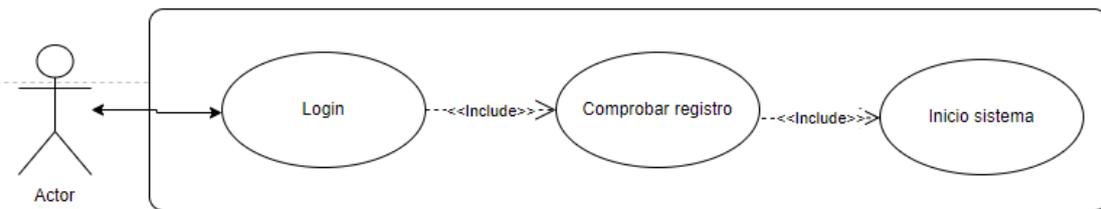
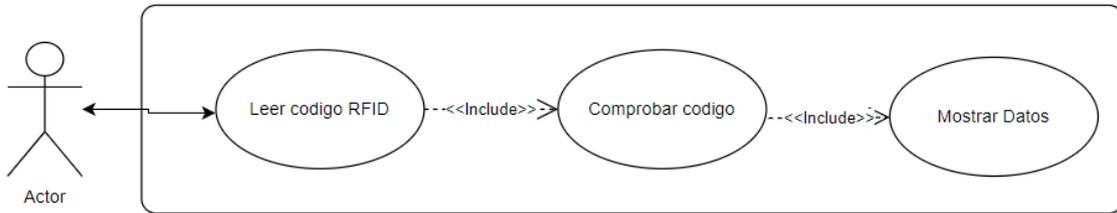
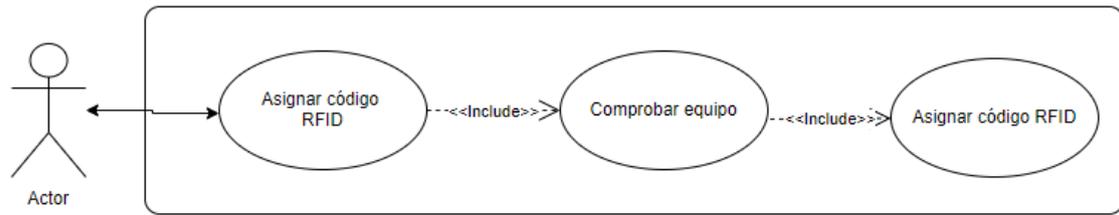
6. DESARROLLO DEL MANUAL

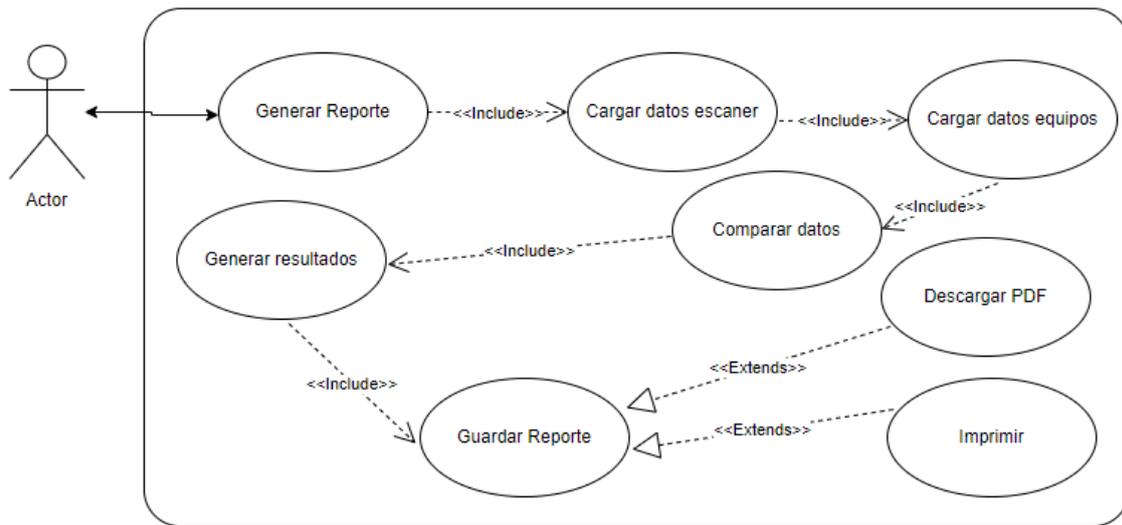
6.1. DIAGRAMA DE CLASES



6.2. DIAGRAMA CASO DE USO







6.3. APLICACIÓN MÓVIL

Librerías aplicadas en Android Studio.

```

dependencies {
    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.5.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    //Libreria para la comunicacion con el arduino
    implementation 'com.github.mik3y:usb-serial-for-android:3.5.1'
    //Libreria para la comunicacion con la apis del sistema
    implementation 'com.squareup.retrofit2:retrofit:2.9.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
    //Libreria para habilitar las notificaciones
    implementation 'com.google.android.material:material:1.4.0'
}
  
```

Manifiesto de la aplicación con los permisos USB y de internet.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.USB_PERMISSION" />
    <uses-sdk tools:overrideLibrary="com.hoho.android.usbserial" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:usesCleartextTraffic="true"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application1"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.MyApplication1"
        android:requestLegacyExternalStorage="true">
    </application>
    <activity
  
```

ApiInterface es donde se realiza la configuración con la Api.

```

13 public interface ApiInterface {
14
15     1 usage
16     @FormUrlEncoded
17     @POST("/ApiScann/GuardarDatos")
18     Call<ResponseBody> getUserInformation(@Field("lab") int lab, @Field("cod") String cod) ;
19
20     1 usage
21     @FormUrlEncoded
22     @POST("/ApiScann/GuardarEquipoNuevo")
23     Call<ResponseBody> setNewEquipoCode(
24         @Field("codigo") String codigo,
25         @Field("descripcion") String descripcion,
26         @Field("idCategoria") int idCategoria,
27         @Field("fechaIngreso") String fechaIngreso,
28         @Field("idLaboratorio") int idLaboratorio,
29         @Field("vida") double vida
30     );
31
32     1 usage
33     @GET("/ApiScann/ObtenerEquipo")
34     Call<Equipo> getEquipoInformation(@Query("codigo") String codigo);

```

En la clase RetrofitClinet se guarda información básica de las Api.

```

1 package com.example.myapplication1;
2
3 import ...
4
5 5 usages
6 public class RetrofitClinet {
7     3 usages
8     public static Retrofit retrofit;
9     5 usages
10    public static Retrofit getInstance(){
11        if(retrofit==null){
12            retrofit=new Retrofit.Builder()
13                .baseUrl("http://scannrfid-001-site1.etempurl.com/")
14                .addConverterFactory(GsonConverterFactory.create())
15                .build();
16        }
17        return retrofit;
18    }
19 }

```

Clases de los diferentes objetos utilizados con sus constructores además de sus métodos Get y Set.

```

1 package com.example.myapplication1;
2
3 public class Categoria {
4     3 usages
5     private int id;
6     3 usages
7     private String nombre;
8     3 usages
9     private String detalle;
10    3 usages
11    private int estado;
12
13    public Categoria(int id, String nombre, String detalle, int estado) {
14        this.id = id;
15        this.nombre = nombre;
16        this.detalle = detalle;
17        this.estado = estado;
18    }
19
20    2 usages
21    public int getId() { return id; }
22
23    public void setId(int id) { this.id = id; }

```

The image displays three sequential screenshots of an IDE, likely Android Studio, showing the development of a Java class hierarchy for an Android application. The left sidebar shows the project structure, and the main editor displays the code for different classes.

Equipo.java

```

4
5 public class Equipo {
6     private int id;
7     private String codigo;
8     private String descripcion;
9     private String fechaIngreso;
10    private double vida;
11    private int estado;
12    private Categoria categoria;
13    private Laboratorio laboratorio;
14    private String responsable;
15
16    public Equipo(int id, String codigo, String descripcion, String fechaIngreso, double vida, int estado, Categoria cat
  
```

Laboratorio.java

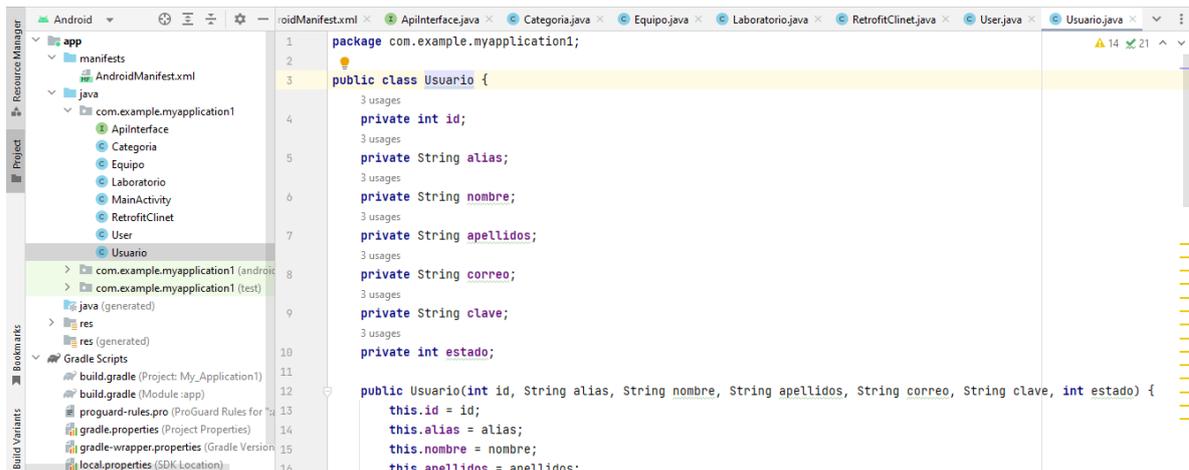
```

1 package com.example.myapplication1;
2
3 import java.util.Calendar;
4
5 public class Laboratorio {
6     private int id;
7     private String nombre;
8     private String referencia;
9     private int estado;
10    private Usuario usuario;
11
12    public Laboratorio(int id, String nombre, String referencia, int estado, Usuario usuario) {
13        this.id = id;
14        this.nombre = nombre;
15        this.referencia = referencia;
16        this.estado = estado;
17        this.usuario = usuario;
  
```

User.java

```

1 package com.example.myapplication1;
2
3 public class User {
4     private int lab;
5     private String cod;
6
7     public User(int lab, String cod) {
8         this.lab = lab;
9         this.cod = cod;
10    }
11
12    public int getLab() {
13        return lab;
14    }
15
16    public void setLab(int lab) { this.lab = lab; }
17
18    public String getCod() { return cod; }
19
20    public void setCod(String cod) { this.cod = cod; }
  
```



Librerías importadas en el MainActivity:

```

import android.Manifest;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.pm.PackageManager;
import android.hardware.usb.UsbDevice;
import android.hardware.usb.UsbManager;
import android.media.MediaPlayer;
import android.os.Build;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;
import com.google.android.material.snackbar.Snackbar;
import com.hoho.android.usbserial.driver.UsbSerialDriver;
import com.hoho.android.usbserial.driver.UsbSerialPort;
import com.hoho.android.usbserial.driver.UsbSerialProber;
import com.hoho.android.usbserial.util.SerialInputOutputManager;
import java.io.IOException;

```

```

import java.text.DateFormat;
import java.text.DecimalFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.concurrent.Executors;
import okhttp3.*;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

```

Declaración de variables

```

private TextView
titulodescripcion,tituloarea,titulocategoria,tituloresponsable,titulof
echa,titulovidapromedio;
private TextView resultTextView,resultTextView2,resultTextView3;
private TextView
txtdescripcion_Get,txtarea_Get,txtcategoria_Get,txtresponsable_Get,txt
fechacompra_Get,txtvidapromedio_Get;
private TextView
txtdescripcion_Set,txtresponsable_Set,txtfechacompra_Set,txtvidapromed
io_Set;
private Spinner txtarea_Set,txtcategoria_Set;
private Button Btingresar;

private SerialInputOutputManager usbIoManager;
private UsbSerialPort usbSerialPort;

```

Método onCreate

//En este método se carga la información del Sistema.

@Override

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

```

//Cargarlayout

```

    setContentView(R.layout.activity_main);
    mp=MediaPlayer.create(this,R.raw.sound) ;

```

//Establece la comunicacion entre los elementos y las variables

```

    asignarCodigos();

```

//Ocultar opciones del get

```

    MostrarGet();

```

```

//Cargar spit categoria
    ObtenerCategoria();
//Cargar spit laboratorio
    ObtenerLaboratorio();
//Se cargan lo permisos necesarios
    String[] permissions = {Manifest.permission.ACCESS_FINE_LOCATION};
    int requestCode = 1;
//Se valida los permisos
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        requestPermissions(permissions, requestCode);
    }
}

```

Método onResume

```

//Cuando la aplicación regresa del segundo plano
@Override
protected void onResume() {
    super.onResume();
    registerReceiver(usbReceiver, new
IntentFilter(UsbManager.ACTION_USB_DEVICE_ATTACHED));
    registerReceiver(usbReceiver, new
IntentFilter(UsbManager.ACTION_USB_DEVICE_DETACHED));
    // Obtener el primer controlador de serie USB de la lista
    UsbManager usbManager = (UsbManager)
getSystemService(Context.USB_SERVICE);
    List<UsbSerialDriver> availableDrivers =
UsbSerialProber.getDefaultProber().findAllDrivers(usbManager);
    // Si la aplicación tiene permisos para acceder al dispositivo USB
    if (!availableDrivers.isEmpty()) {
        UsbSerialDriver driver = availableDrivers.get(0);
        UsbDevice device = driver.getDevice();
        if (usbManager.hasPermission(device)) {
            usbSerialPort = driver.getPorts().get(0);
            try {
                // Abrir el puerto serie USB y configurar los parámetros de
comunicación
                usbSerialPort.open(usbManager.openDevice(device));
                usbSerialPort.setParameters(9600, 8,
UsbSerialPort.STOPBITS_1, UsbSerialPort.PARITY_NONE);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

// Crear un SerialInputOutputManager para manejar las entradas y
// salidas del puerto
        usbIoManager = new SerialInputOutputManager(usbSerialPort,
this);
        Executors.newSingleThreadExecutor().submit(usbIoManager);
    } else {
// Solicitar permiso al usuario
        PendingIntent permissionIntent =
PendingIntent.getBroadcast(this, 0, new Intent(ACTION_USB_PERMISSION),
0);
        usbManager.requestPermission(device, permissionIntent);
    }
}
}
}

```

Método onPause

//Se carga cuando el estado se pone en pausa

```

@Override
protected void onPause() {
    super.onPause();
    unregisterReceiver(usbReceiver);
    if (usbSerialPort != null) {
        try {
            usbSerialPort.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        usbSerialPort = null;
    }
    if (usbIoManager != null) {
        usbIoManager.stop();
        usbIoManager = null;
    }
}
}

```

Método onNewData

//En este metodo se carga solo cuando se detecte un Nuevo código RFID

```

public void onNewData(byte[] data) {
    String receivedData = new String(data);
    runOnUiThread(new Runnable() {
        @Override
        public void run() {

```

```

// Eliminar espacios en blanco alrededor de la cadena
String function = receivedData.trim();
resultTextView.append(function);
resultTextView3.setText("." +
resultTextView.getText().toString().length());
if (resultTextView.getText().toString().length() == 8) {
    String sortedData =
sortData(resultTextView.getText().toString());

resultTextView2.setText(resultTextView.getText().toString());

    if (txtdescripcion_Get.getVisibility()==View.VISIBLE) {
        ObtenerEquipo(resultTextView.getText().toString());
    } else if
(titulodescripcion.getVisibility()==View.GONE) {
        InsertarData ( resultTextView.getText().toString());
    }
    mp.start();
}
//Limpiar data
resultTextView.setText("");
// Establecer la longitud a 0
resultTextView3.setText(".0");
}
}
});
}

```

Método BroadcastReceiver

```

//En este método se detecta la conexión USB
private final BroadcastReceiver usbReceiver = new BroadcastReceiver()
{
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if (action != null) {
            switch (action) {
                case UsbManager.ACTION_USB_DEVICE_ATTACHED:
                    // Dispositivo USB conectado, manejarlo si es
                    necesario
                    break;
                case UsbManager.ACTION_USB_DEVICE_DETACHED:
                    // Dispositivo USB desconectado, manejarlo si es

```

necesario

```

        break;
    }
}
};

```

Método onRequestPermissionsResult

En este método se piden los permisos de la aplicación

```

@Override
public void onRequestPermissionsResult(int requestCode, String[]
permissions, int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == 1) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            // Permiso concedido, realizar las acciones necesarias
            UsbManager usbManager = (UsbManager)
getSystemService(Context.USB_SERVICE);
            List<UsbSerialDriver> availableDrivers =
UsbSerialProber.getDefaultProber().findAllDrivers(usbManager);
            if (!availableDrivers.isEmpty()) {
                UsbSerialDriver driver = availableDrivers.get(0);
                UsbDevice device = driver.getDevice();
                if (usbManager.hasPermission(device)) {
                    usbSerialPort = driver.getPorts().get(0);
                    try {
                        usbSerialPort.open(usbManager.openDevice(device));
                        usbSerialPort.setParameters(9600, 8,
UsbSerialPort.STOPBITS_1, UsbSerialPort.PARITY_NONE);
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                    usbIoManager = new
SerialInputOutputManager(usbSerialPort, this);
                    Executors.newSingleThreadExecutor().submit(usbIoManager);
                }
            }
}
}

```

```

    } else {
        // Permiso denegado, manejar el escenario sin permisos
    }
}
}

```

Método para cargar spit

```

    private void ObtenerCategoria() {
//Conexion con api
        ApiInterface apiInterface =
        RetrofitClinet.getRetrofitInstance().create(ApiInterface.class);
        Call<List<Categoria>> call =
        apiInterface.getCategoriaInformation();
        call.enqueue(new Callback<List<Categoria>>() {
            @Override
//Respuesta de la api
            public void onResponse(Call<List<Categoria>> call,
            Response<List<Categoria>> response) {
                ArrayList<Categoria> data = (ArrayList<Categoria>)
                response.body();
                ArrayAdapter<Categoria> adapter = new
                ArrayAdapter<Categoria>(MainActivity.this,
                android.R.layout.simple_spinner_item, data) {
                    @NonNull
                    @Override
//asignacion de spit
                    public View getView(int position, @Nullable View
                    convertView, @NonNull ViewGroup parent) {
                        TextView textView = (TextView)
                        super.getView(position, convertView, parent);

                        textView.setText(data.get(position).getNombre());
                        return textView;
                    }

                    @Override
                    public View getDropDownView(int position,
                    @Nullable View convertView, @NonNull ViewGroup parent) {
                        TextView textView = (TextView)
                        super.getDropDownView(position, convertView, parent);

```

```

textView.setText(data.get(position).getNombre());
        return textView;
    }
};

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdo
wn_item);

txtcategoria_Set.setAdapter(adapter);
txtcategoria_Set.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent,
View view, int position, long id) {
        Categoria selectedLaboratorio=
data.get(position);
        String selectedId =
String.valueOf(selectedLaboratorio.getId());

        Log.e(TAG, "Selected ID: " + selectedId);
    }

    @Override
    public void onNothingSelected(AdapterView<?>
parent) {
        // No se seleccionó ningún elemento
    }
});

}

@Override
public void onFailure(Call<List<Categoria>> call,
Throwable t) {
    }
} );
}
}

```

Método obtener equipo

```

private void ObtenerEquipo(String codigo) {

//Limpiar contenedores get
    limpiarget();
    ApiInterface apiInterface =
RetrofitClinet.getRetrofitInstance().create(ApiInterface.class);
    Call<Equipo> call = apiInterface.getEquipoInformation(codigo);
    call.enqueue(new Callback<Equipo>() {
        @Override
        public void onResponse(Call<Equipo> call, Response<Equipo>
response) {
            if (response.isSuccessful()) {
                Equipo equipo = response.body();
                // Accede a los atributos del equipo y realiza las
operaciones necesarias

                txtdescripcion_Get.setText(equipo.getDescripcion());

txtarea_Get.setText(equipo.getLaboratorio().getNombre());

txtcategoria_Get.setText(equipo.getCategoria().getDetalle());
                txtresponsable_Get.setText(equipo.getResponsable());
                txtfechacompra_Get.setText(equipo.getFechaIngreso());
                String equipoStr = equipo.toString();
                Log.d(TAG, "Contenido del objeto equipo: " +
equipoStr);

txtvidapromedio_Get.setText(convertToFormat(equipo.getVida()));

            } else {
                // La respuesta no fue exitosa, maneja el error
            }
        }
    });

    @Override
    public void onFailure(Call<Equipo> call, Throwable t) {
        // Maneja la falla en la solicitud
    }
}

```

```

        Log.e(TAG, "Error en la solicitud: " + t.getMessage());
    }
});
}

```

Método insertar data

//Este metodo se aplica para guardar los datos del escáneo

```

private void InsertarData(String cod) {
    ApiInterface apiInterface =
RetrofitClinet.getRetrofitInstance().create(ApiInterface.class);
    Laboratorio selectedCategoria = (Laboratorio)
txtarea_Set.getSelectedItemAt();
    int selectedId = selectedCategoria.getId();

    Call<ResponseBody> call =
apiInterface.getUserInformation(selectedId, cod);
    call.enqueue(new Callback<ResponseBody>() {
        @Override
        public void onResponse(Call<ResponseBody> call,
Response<ResponseBody> response) {
            if (response.isSuccessful()) {
                try {
                    String jsonResponse = response.body().string();
                    if (jsonResponse.equals("Ya existe")) {
                        mostrarMensaje("El código ya existe");
                    } else if (jsonResponse.equals("Registrado")) {
                        mostrarMensaje("Registro exitoso");
                        limpiarset();
                    } else {
                        mostrarMensaje("Respuesta desconocida del
servidor");
                    }
                } catch (IOException e) {
                    e.printStackTrace();
                }
            } else {
                mostrarMensaje("Error en la solicitud, revise el
código");
            }
        }
    });
}

```

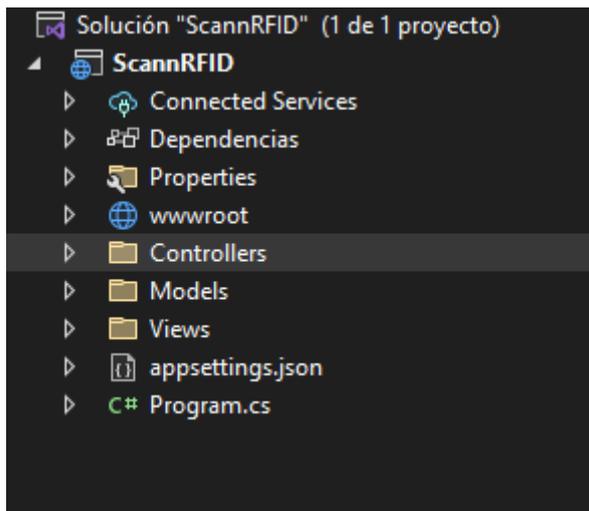
```

    @Override
    public void onFailure(Call<ResponseBody> call, Throwable t) {
        mostrarMensaje("Error en la llamada: " + t.getMessage());
    }
});
}

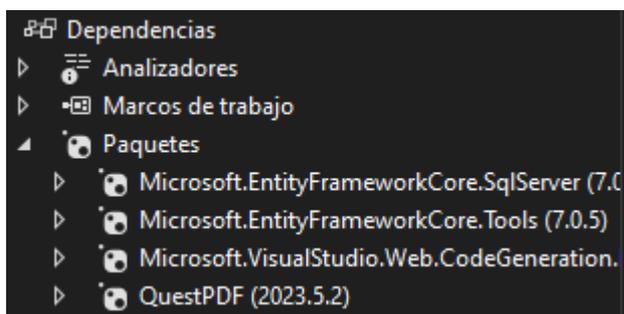
```

6.4. Aplicación web sistema de inventario RFID

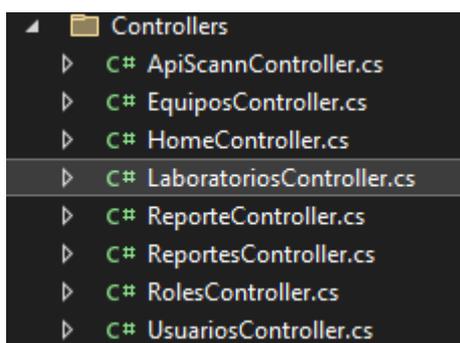
Directorio app



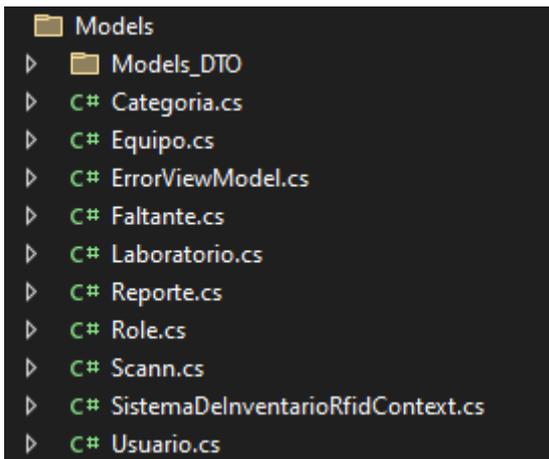
Paquetes necesarios



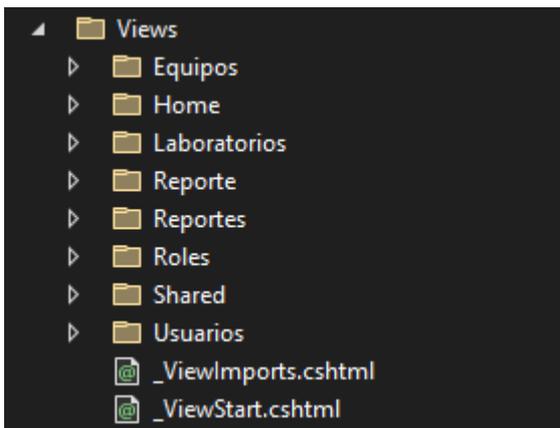
Controllers



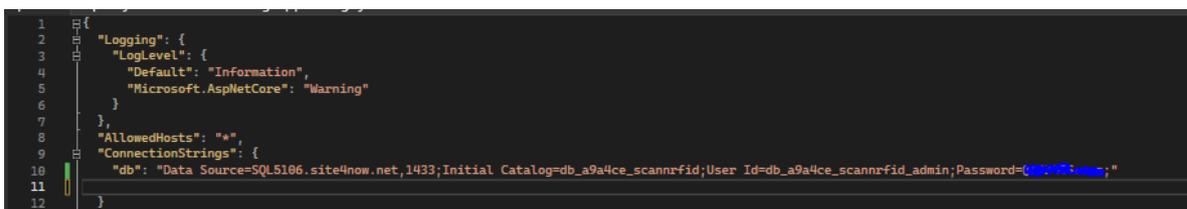
Modelos y Models_DTO



Vista de la aplicación



Cadena de conexión



Sistema listo para trabajar

```

D:\ARCHIVOS\descargas\ScannRFID (2)\ScannRFID\bin\Debug\net6.0\ScannRFID.exe
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7248
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5248
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: D:\ARCHIVOS\descargas\ScannRFID (2)\ScannRFID\

```

6.5. CODIGO CONEXIÓN ARDUINO Y MODULO RFID

//Se importan las librerias

```
#include <SPI.h>
```

```
#include <MFRC522.h>
```

// Pin 9 para el reset del RC522

```
#define RST_PIN 9
```

// Pin 10 para el SS (SDA) del RC522

```
#define SS_PIN_ENB 10
```

// Creamos el objeto para el RC522

```
MFRC522 MyLectorRF(SS_PIN_ENB, RST_PIN);
```

// String para almacenar el código RFID

```
String rfidCode = "";
```

```
void setup() {
```

// Iniciamos la comunicación serial

```
  Serial.begin(9600);
```

// Iniciamos el Bus SPI

```
  SPI.begin();
```

// Iniciamos el MyLectorRF

```
  MyLectorRF.PCD_Init();
```

```
  Serial.println("Control Inicializado ...");
```

```
}
```

```
void loop() {
```

```
/// Verificamos si se ha detectado alguna tarjeta
if (MyLectorRF.PICC_IsNewCardPresent()) {
    /// Determinamos el código de la tarjeta
    if (MyLectorRF.PICC_ReadCardSerial()) {
        /// Recuperamos el ID de la tarjeta
        rfidCode = "";
        for (byte i = 0; i < MyLectorRF.uid.size; i++) {
            if (MyLectorRF.uid.uidByte[i] < 0x10) {
                rfidCode += "0";
            }
            rfidCode += String(MyLectorRF.uid.uidByte[i], HEX);
        }
        /// Imprimimos el código RFID sin espacios
        rfidCode.replace(" ", "");
        Serial.println(rfidCode);
        delay(100);
        /// Terminamos la lectura de la tarjeta actual
        MyLectorRF.PICC_HaltA();
    }
}
}
```

ANEXO 9. MANUAL DE USUARIO

Sistema inventario RFID

Manual de usuario

Versión: 1.0

Fecha: jun/2023

Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso

Contenido

1. OBJETIVO.....	80
2. DEFINICIONES.....	80
3. DESARROLLO DEL MANUAL	80

1. OBJETIVO

El objetivo de este manual de usuario es proporcionar a los usuarios una guía completa y clara sobre el funcionamiento del sistema de inventario RFID. El manual tiene como propósito permitir que los usuarios comprendan y utilicen eficientemente todas las funciones y características del sistema, garantizando así una gestión eficaz y precisa del inventario.

2. DEFINICIONES

SISTEMA DE INVENTARIO RFID: Es una aplicación web que ofrece una solución avanzada para simplificar la gestión de inventarios, aprovechando al máximo las tecnologías RFID. Con la aplicación, los usuarios pueden realizar informes detallados y acceder a una amplia gama de funcionalidades que optimizan el proceso de inventario.

MÓDULO LECTOR: Este módulo se refiere a la unidad de comunicación provista, que consta de un lector, una placa Arduino y una carcasa. El lector es el dispositivo encargado de capturar la información de las etiquetas RFID, mientras que la placa Arduino se encarga de procesar y transmitir los datos. La carcasa brinda protección y soporte físico al conjunto. Juntos, estos componentes forman el módulo lector, esencial para el funcionamiento del sistema.

APLICACIÓN MÓVIL: Nuestra solución incluye una aplicación móvil específicamente diseñada para dispositivos Android. Esta aplicación juega un papel fundamental al facilitar la comunicación entre el módulo lector y el sistema de inventario RFID. A través de la aplicación móvil, los usuarios pueden visualizar y gestionar los datos capturados por el lector RFID, así como acceder a diversas funcionalidades del sistema de inventario. La aplicación móvil proporciona una interfaz intuitiva y fácil de usar, permitiendo una interacción fluida y eficiente con el sistema de inventario RFID.

3. DESARROLLO DEL MANUAL

3.1. *APLICACIÓN MÓVIL*

La aplicación móvil proporciona con las siguientes opciones:

3.1.1. *MENÚ INGRESO EQUIPOS*

En el inicio de esta opción se deberá leer mediante el módulo de lector el código de nuestra tarjeta RFID.

En este menú se deberá llenar los siguientes campos:

Número de serie: Últimos 6 números de serie del equipo.

Detalle: Información breve del equipo.

Área asignada: Seleccionar los laboratorios disponibles.

Fecha de ingreso: Fecha de inicio de uso del equipo.

Vida útil programada: Tiempo en meses de uso recomendado.

Una vez ingresados estos datos el usuario deberá guardar los cambios

The screenshot shows a mobile application interface with the following elements:

- Time: 9:27
- Signal and battery icons.
- Section header: **CÓDIGO RFID**
- RFID Code: **05ae5183**
- Section header: **Información adicional**
- Fields and values:
 - Número Serie:** 6024569
 - Detalle equipo:** Computadora i7 4gb ram 500 sdd
 - Área asignada:** Laboratorio 101 informatica (with a dropdown arrow)
 - Responsable:** Ing. Yimmy Loor Vera
 - Fecha ingreso:** 17/04/2022
 - Vida útil programada :** 48 MESES
- Buttons:
 - Asignar N° Serie (grey button)
 - Escanear área (blue button with a QR icon)
 - Asignar código (white button)
 - Leer Código (blue button)

3.1.2. MENÚ ESCANEADO EQUIPOS

En esta opción, simplemente al leer las etiquetas con el módulo lector, se obtendrá la información del equipo si está ingresado en el inventario, mientras que, si el equipo no está registrado, se mostrará en blanco.

9:27

CÓDIGO RFID

05ae5183

Información adicional

Número Serie: 6024569

Detalle equipo:
Computadora i7 4gb ram 500 sdd

Área asignada:
Laboratorio 101 informatica

Responsable:
Ing. Yimmy Loor Vera

Fecha ingreso:
17/04/2022

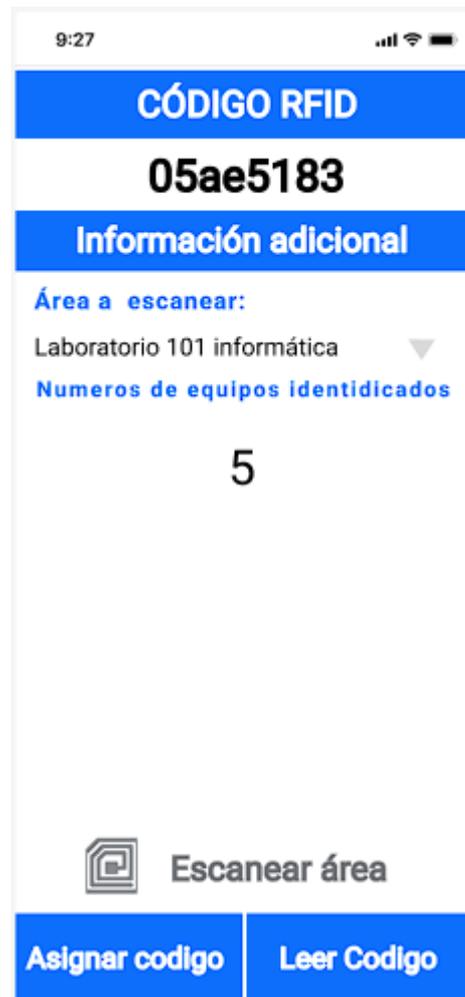
Vida útil programada :
48 MESES

 **Escanear área**

Asignar código **Leer Código**

3.1.3.MENÚ ESCANEAO LABORATORIOS

En este menú, simplemente tienes que seleccionar el laboratorio que deseas escanear. Al leer las etiquetas con el módulo lector, estas se ingresarán automáticamente al sistema de inventario RFID mostrando un contador de los equipos correctamente leídos.



3.2. APLICACIÓN WEB SISTEMA DE INVENTARIO RFID

3.2.1. MENÚ USUARIOS

Icono

Inicio

Usuarios

Laboratorios

Equipos

Roles

Verificar

Reporte

Hola,

Usuarios Agregar

Mostrando registros del 1 al 2 de un total de 2 registros

Alias	Nombres	Apellidos	Correo	Rol	Estado	Acciones
Chema	Jonathan Geovanny	Ormaza Calderon	jonathan.ormaza@espam.edu.ec	Admin	Activo	+ -
Murphy	Carlos Pierre	Quijje Vera	pierre@gmail.com	UDIVS	Activo	+ -

Mostrar 5 registros

Anterior 1 Siguiete

Buscar:

2022 © ESPAM MFL

3.2.1.1. AGREGAR

En este menú se necesitan ingresar las siguientes opciones:

Nombre: Nombres del usuario

Apellidos: Apellidos del usuario

Alias: Nickname del usuario

Correo: Correo electrónico del usuario

Clave: clave de acceso al sistema

Rol: Tipo de acceso del sistema

Agregar un nuevo usuario
✕

NOMBRES

APELLIDOS

ALIAS

CORREO

CLAVE

ROL

Cerrar
Guardar

3.2.1.2. EDITAR

Para acceder al menú de editar se necesita presionar el botón azul de la tabla.

Mostrando registros del 1 al 2 de un total de 2 registros

Alias	Nombres	Apellidos	Correo	Rol	Estado	Acciones
Chema	Jonathan Geovanny	Ormaza Calderon	jonathan.ormaza@espamedu.ec	Admin	Activo	
Murphy	Carlos Pierre	Quijije Vera	pierre@gmail.com	UDIVS	Activo	

Cuando se ingrese a la opción solo se podrán cambiar los siguientes datos.

Editar usuario ✕

NOMBRES

APELLIDOS

ALIAS

ROL

Admin
▼

Cerrar

Editar

3.2.1.3. ELIMINAR

Para eliminar se necesita presionar el botón azul de la tabla.

Mostrando registros del 1 al 2 de un total de 2 registros

Alias	Nombres	Apellidos	Correo	Rol	Estado	Acciones
Chema	Jonathan Geovanny	Ormaza Calderon	jonathan.ormaza@espam.edu.ec	Admin	Activo	- +
Murphy	Carlos Pierre	Quijije Vera	pierre@gmail.com	UDIVS	Activo	- +

3.2.2. LABORATORIOS

Icono

- [Inicio](#)
- [Usuarios](#)
- [Laboratorios](#)
- [Equipos](#)
- [Roles](#)
- [Verificar](#)
- [Reporte](#)

☰
Hola,

Laboratorios Agregar

Mostrando registros del 1 al 2 de un total de 2 registros

Nombre	Referencia	Responsable	Estado	Acciones
201	Planta Baja	Carlos Pierre Quijije Vera	Activo	- +
202	Planta Baja	Sandro Antonio Palau Delgado	Activo	- +

Mostrar 5 registros

Anterior
1
Siguiente
Buscar:

2022 © ESPAM MFL

3.2.2.1. AGREGAR

En este menú se necesitan ingresar las siguientes opciones:

Nombre: Nombres del usuario

Referencia: Ubicación e laboratorio

Responsable: Usuario responsable del área

Agregar un nuevo laboratorio
✕

NOMBRE

REFERENCIA

RESPONSABLE

Cerrar
Guardar

3.2.2.2. EDITAR

Para eliminar se necesita presionar el botón azul de la tabla.

Mostrando registros del 1 al 2 de un total de 2 registros

Nombre	Referencia	Responsable	Estado	Acciones
201	Planta Baja	Carlos Pierre Quijije Vera	Activo	
202	Planta Baja	Sandro Antonio Palau Delgado	Activo	

Mostrar registros

Anterior Siguiente

Buscar:

2022 © ESPAM MFL

Cuando se ingrese a la opción solo se podrán cambiar los siguientes datos.

Editar laboratorio ✕

NOMBRE

REFERENCIA

RESPONSABLE

3.2.2.3. ELIMINAR

Mostrando registros del 1 al 2 de un total de 2 registros

Nombre	Referencia	Responsable	Estado	Acciones
201	Planta Baja	Carlos Pierre Quijije Vera	Activo	
202	Planta Baja	Sandro Antonio Palau Delgado	Activo	

Mostrar registros

Anterior Siguiente

Buscar:

2022 © ESPAM MFL

3.2.3. MENÚ EQUIPOS

Icono

- Inicio
- Usuarios
- Laboratorios
- Equipos
- Roles
- Verificar
- Reporte

Hola,

Equipos Agregar

Mostrando registros del 1 al 4 de un total de 4 registros

Codigo	Descripción	Vida	Categoría	Laboratorio	Estado	Acciones
05ae5183	teclado	0	Computadoras	201	Activo	
05ae51833	Monitor	0	Computadoras	202	Activo	
d257a624	impresora	0	Computadoras	201	Activo	
d257a6245	Mouse	0	Computadoras	201	Activo	

Mostrar registros

Anterior Siguiente

Buscar:

2022 © ESPAM MFL

3.2.3.1. AGREGAR

En este menú se necesitan ingresar las siguientes opciones:

Número de serie: Últimos 6 números de serie del equipo.

Código RFID: Numero de etiqueta RFID.

Detalle: Información breve del equipo.

Categoría: Selección de tipo de equipo.

Área asignada: Seleccionar los laboratorios disponibles.

Fecha de ingreso: Fecha de inicio de uso del equipo.

Vida útil programada: Tiempo en meses de uso recomendado.

Agregar un nuevo equipo
✕

CODIGO

DESCRIPCION

FECHA DE COMPRA

VIDA PROMEDIO

CATEGORIA

LABORATORIO

Cerrar
Guardar

3.2.3.2. *EDITAR*

Para eliminar se necesita presionar el botón azul de la tabla.

Equipos							Agregar
Mostrando registros del 1 al 4 de un total de 4 registros							
Codigo	Descripcion	Vida	Categoria	Laboratorio	Estado	Acciones	
05ae5183	teclado	0	Computadoras	201	Activo	■ ■	
05ae51833	Monitor	0	Computadoras	202	Activo	■ ■	
d257a624	impresora	0	Computadoras	201	Activo	■ ■	
d257a6245	Mouse	0	Computadoras	201	Activo	■ ■	

Mostrar registros
Anterior Siguiente
Buscar:

Cuando se ingrese a la opción solo se podrán cambiar los siguientes datos.

Agregar un nuevo equipo

✕

CODIGO

DESCRIPCION

FECHA DE COMPRA

VIDA PROMEDIO

CATEGORIA

LABORATORIO

Cerrar
Editar

3.2.3.3. ELIMINAR

Equipos

Agregar

Mostrando registros del 1 al 4 de un total de 4 registros

Codigo	Descripcion	Vida	Categoria	Laboratorio	Estado	Acciones
05ae5183	teclado	0	Computadoras	201	Activo	- - +
05ae51833	Monitor	0	Computadoras	202	Activo	- -
d257a624	impresora	0	Computadoras	201	Activo	- -
d257a6245	Mouse	0	Computadoras	201	Activo	- -

Mostrar registros
Anterior Siguiente
Buscar:

3.2.4. MENÚ ROLES

Este menú solo se podrá observar los distintos roles permitidos del usuario

Roles

Mostrando registros del 1 al 4 de un total de 4 registros

ID	Nombre
1	Admin
2	Docente
3	Director de Carrera
4	UDIVS

Mostrar 5 registros Anterior 1 Siguiente

2022 © ESPAM MFL

3.2.5. MENÚ VERIFICAR

Para verificar se tendría que seleccionar el laboratorio en cual se realizó el escaneo donde se mostraran los equipos escaneados con los diferentes estados:

Completo: Equipo perteneciente en el laboratorio.

Faltante: Equipos los cuales no se identificaron.

No pertenece en el laboratorio: Equipo escaneado, pero no perteneciente al laboratorio.

LABORATORIOS

201 Generar

Codigo	Descripcion	Laboratorio	Estado
05ae5183	teclado	201	Completo
d257a624	impresora	201	Completo
d257a6245	Mouse	201	Faltantes
05ae51833	Monitor	202	No pertenecen a este Laboratorio

2022 © ESPAM MFL

3.2.6. MENÚ REPORTE

W

3.2.7. HOJA DE REPORTE

MostrarPDF 1 / 1 53% +

1

COMITÉ DE COMPUTACIÓN **ESPAAM MFL** **Comité de Computación** **www.espaam.fli.edu.ec**

Cod. 201
Fecha de reporte
2023-06-12

Datos del Laboratorio
Responsable: Jonathan Córdova Ormaza Calderón
Laboratorio: 201
Total de equipos: 7

Equipos correctos

Código	Nombre	Laboratorio	Estado
0			

Equipos Faltantes

Código	Nombre	Laboratorio	Estado
201-a	Monitor	201	Faltante
201-ab	Teclado	201	Faltante
201-ac	Mouse	201	Faltante
201-ad	Mouse	201	Faltante
201-ae	Monitor	201	Faltante
5			

Equipos que no pertenecen al laboratorio 201

Código	Nombre	Laboratorio	Estado
0			

Página 1 de 1