



Control de inventario de bienes por medio de una aplicación web con integración de una aplicación para dispositivos móviles, utilizando tecnología RFID, para la Universidad de las Fuerzas Armadas – ESPE Sede Sangolquí.

Sampedro Cruz, Wladimir Stephen

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Sistemas e Informática

Trabajo de titulación, previo a la obtención del título de Ingeniero en Sistemas e Informática

Ing. Aguilar Lema, Kléber Augusto MSc.

13 de marzo 2021

1/10/21 08:45

tesis revision para biblioteca

Informe de originalidad

NOMBRE DEL CURSO

Proyecto de Grado

NOMBRE DEL ALUMNO

WLADIMIR STEPHE SAMPEDRO CRUZ

NOMBRE DEL ARCHIVO

WLADIMIR STEPHE SAMPEDRO CRUZ - Documento sin titulo

SE HA CREADO EL INFORME

30 sept 2021

Resumen

Fragmentos marcados	0	0 %
Fragmentos citados o entrecorridos	105	38 %

Kléber
Aguilar L.

Firmado digitalmente por Kléber Aguilar L., Fecha: 2021.10.01 08:53:24 -0500

Coincidencias de la Web

wikipedia.org	29	15 %
universidadviu.com	6	2 %
lebschool.com	7	2 %
coursehero.com	8	2 %
qualitydevs.com	3	2 %
ceupe.com	5	2 %
oracle.com	7	1 %
espe.edu.ec	1	1 %
dipolerfid.es	3	0,9 %
sinnaps.com	2	0,8 %
codmind.com	2	0,7 %
facialix.com	2	0,7 %
acomee.com.mx	1	0,6 %
educalingo.com	1	0,6 %
scrum.mx	2	0,6 %
arquitecturajava.com	1	0,5 %
bisacorporation.com	1	0,5 %
ingenieriadesoftwarei-2020.blogspot.com	1	0,5 %
renatorodgal.blogspot.com	1	0,4 %
docplayer.es	1	0,4 %
como-funciona.co	3	0,4 %
fje.edu	2	0,4 %
emeraldtech.net.ar	2	0,4 %
finanzas.gob.ec	1	0,3 %



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

CERTIFICACIÓN

Certifico que el trabajo de titulación, **“Control de inventario de bienes por medio de una aplicación web con integración de una aplicación para dispositivos móviles, utilizando tecnología RFID, para la Universidad de las Fuerzas Armadas – ESPE Sede Sangolquí”** fue realizado por el señor Sampedro Cruz, Wladimir Stephen el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 30 de marzo 2021



Firmado
digitalmente por
KLEBER
AUGUSTO
AGUILAR LEMA

Aguilar Lema, Kléber Augusto

C. C.: 1710739432



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

RESPONSABILIDAD DE AUTORÍA

Yo, Sampedro Cruz, Wladimir Stephen, con cédula de ciudadanía n°1718163387 declaro que el contenido, ideas y criterios del trabajo de titulación: **Control de inventario de bienes por medio de una aplicación web con integración de una aplicación para dispositivos móviles, utilizando tecnología RFID, para la Universidad de las Fuerzas Armadas – ESPE Sede Sangolquí** de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 30 de marzo 2021

Sampedro Cruz, Wladimir Stephen

C.C.: 1718163387



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

AUTORIZACIÓN DE PUBLICACIÓN

Yo, Sampedro Cruz, Wladimir Stephen, con cédula de ciudadanía n°1718163387, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Control de inventario de bienes por medio de una aplicación web con integración de una aplicación para dispositivos móviles, utilizando tecnología RFID, para la Universidad de las Fuerzas Armadas – ESPE Sede Sangolquí** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 30 de marzo 2021

Sampedro Cruz, Wladimir Stephen

C.C.: 1718163387

DEDICATORIA

Dedico este proyecto de tesis a mis padres y hermanos que han sido quienes me han apoyado incondicionalmente en cada paso para lograr llegar a cumplir este objetivo. Ha sido gracias a sus ánimos, impulsos y ayuda que me han brindado durante este proceso los cuales me han permitido culminar con la carrera.

También quiero dedicar este trabajo a todas las personas que siempre han estado dispuesto a darme una mano y ayudarme dentro de sus posibilidades en el camino para lograr concluir con este proyecto.

Por último, quiero dedicarme este resultado por todo el tiempo, esfuerzo y perseverancia hecho diligentemente para lograr con éxito la finalización de la carrera.

AGRADECIMIENTO

Agradezco primeramente a Dios por la fortaleza, inteligencia y salud que me permitieron culminar con mi formación profesional y también agradezco a mis padres, hermanos, padrinos, tíos que mostraron siempre su interés en apoyarme en terminar con este proceso.

Segundo, quiero agradecer a mi tutor de tesis el ing. Kléber Aguilar por su ayuda y seguimiento para el correcto desarrollo del proyecto de investigación.

Finalmente quiero agradecer a mi universidad "ESPE", a todos los docentes que forman parte de la carrera de Ingeniería de Sistemas e Informática por compartir parte de sus conocimientos que ayudaron en la formación al impartir cada una de las clases pertenecientes a la malla curricular de la carrera.

TABLA DE CONTENIDOS

CARÁTULA

CERTIFICACIÓN -----	3
RESPONSABILIDAD DE AUTORÍA -----	4
AUTORIZACIÓN DE PUBLICACIÓN -----	5
DEDICATORIA -----	6
AGRADECIMIENTO -----	7
TABLA DE CONTENIDOS -----	8
ÍNDICE DE TABLAS -----	13
ÍNDICE DE FIGURAS -----	14
RESUMEN -----	17
ABSTRACT -----	18
CAPÍTULO I -----	19
INTRODUCCIÓN -----	19
ANTECEDENTES -----	19
PROBLEMÁTICA -----	22
JUSTIFICACIÓN -----	24
OBJETIVOS -----	25
<i>Objetivo general</i> -----	25
<i>Objetivos Específicos</i> -----	25
ALCANCE -----	26
FACTIBILIDAD -----	28
<i>Factibilidad técnica</i> -----	28

<i>Requerimientos de hardware</i> -----	29
<i>Requerimientos de software</i> -----	29
<i>Factibilidad económica</i> -----	29
CAPÍTULO II -----	32
MARCO TEÓRICO -----	32
CONTROL DE INVENTARIOS -----	32
BENEFICIOS DE EJERCER UN BUEN CONTROL DE INVENTARIO-----	34
BASE DE DATOS -----	34
<i>ORACLE</i> -----	34
BACK-END -----	36
<i>Spring boot</i> -----	36
FRONT-END -----	37
<i>Angular</i> -----	37
<i>IONIC Framework</i> -----	38
METODOLOGÍA -----	39
<i>Metodología XP</i> -----	40
FORMAS DE CONTROL FÍSICO DE INVENTARIO-----	42
<i>Código de barras</i> -----	42
<i>Tecnología RFID</i> -----	43
<i>Tipos de dispositivos RFID</i> -----	46
<i>Cómo funciona el sistema RFID</i> -----	47
<i>Aplicaciones presentes y futuras</i> -----	47
<i>Arquitectura</i> -----	48
<i>Tipos de Etiqueta RFID</i> -----	49
<i>Etiquetas Pasivas</i> -----	51
<i>Etiquetas Activas</i> -----	53

<i>Etiquetas semipasivas</i> -----	56
<i>Tipos de antena</i> -----	57
<i>Asociación de etiquetas</i> -----	58
<i>Entornos de etiquetas</i> -----	59
CAPÍTULO III -----	62
ANÁLISIS Y DISEÑO DEL SISTEMA -----	62
ESPECIFICACIONES DE REQUERIMIENTOS DE SOFTWARE-----	62
INTRODUCCIÓN-----	62
PROPÓSITO-----	62
ALCANCE -----	62
HISTORIAS DE USUARIOS-----	63
<i>Concepto</i> -----	63
<i>Historias de usuario</i> -----	63
Historia de usuario 1-----	64
Historia de usuario 2-----	65
Historia de usuario 3-----	66
Historia de usuario 4-----	67
Historia de usuario 5-----	68
Historia de usuario 6-----	69
Historia de usuario 7-----	70
Historia de usuario 8-----	71
DIAGRAMAS ENTIDAD RELACIÓN -----	72
<i>Definición</i> -----	72
<i>Entidades nuevas</i> -----	72
<i>Entidades</i> -----	73
DIAGRAMAS DE CLASES -----	76
<i>Definición</i> -----	76

<i>Clases</i>	77
CAPÍTULO IV	78
DESARROLLO Y CODIFICACIÓN	78
INTRODUCCIÓN	78
DESARROLLO Y CONFIGURACIÓN DEL SERVIDOR (BACKEND)	79
<i>Creación de proyecto</i>	79
<i>Procesos</i>	79
<i>Anotaciones</i>	83
DESARROLLO DE LA APLICACIÓN WEB	91
<i>Creación de proyecto</i>	91
CAPÍTULO V	98
RESULTADOS OBTENIDOS Y PRUEBAS	98
PRUEBAS DE RENDIMIENTO	98
<i>Definición</i>	98
<i>Uso de batería</i>	98
<i>Uso de red</i>	99
<i>Uso de memoria</i>	100
<i>Uso de CPU</i>	101
PRUEBAS DE ACEPTACIÓN	103
<i>Definición</i>	103
CAPÍTULO VI	115
CONCLUSIONES Y RECOMENDACIONES	115
CONCLUSIONES	115
RECOMENDACIONES	116

GLOSARIOS DE TÉRMINOS	117
REFERENCIAS BIBLIOGRÁFICAS	118
ANEXOS	121

ÍNDICE DE TABLAS

Tabla 1 Proyección de costos	30
Tabla 2 Comparación entre códigos de barras vs RFID	46
Tabla 3 Dispositivos	60
Tabla 4 Historia 1.....	64
Tabla 5 Historia 2.....	65
Tabla 6 Historia 3.....	66
Tabla 7 Historia 4.....	67
Tabla 8 Historia 5.....	68
Tabla 9 Historia 6.....	69
Tabla 10 Historia 7.....	70
Tabla 11 Historia 8.....	71
Tabla 12 Prueba de aceptación 1	103
Tabla 13 Prueba de aceptación 2	104
Tabla 14 Prueba de aceptación 3	104
Tabla 15 Prueba de aceptación 4	105
Tabla 16 Prueba de aceptación 5	106
Tabla 17 Prueba de aceptación 6	107
Tabla 18 Prueba de aceptación 7	108
Tabla 19 Prueba de aceptación 8	109
Tabla 20 Prueba de aceptación 9	109
Tabla 21 Prueba de aceptación 10	110
Tabla 22 Prueba de aceptación 11	111
Tabla 23 Prueba de aceptación 12	112
Tabla 24 Prueba de aceptación 13	112
Tabla 25 Prueba de aceptación 14	113

ÍNDICE DE FIGURAS

Figura 1 Ionic	39
Figura 2 Metodologías	40
Figura 3 Lector de Códigos de barras	42
Figura 4 RFID READER.....	50
Figura 5 Baterías.....	55
Figura 6 Gráficos.....	55
Figura 7 Funcionarios	72
Figura 8 Activos RFID	72
Figura 9 Archivos	73
Figura 10 Provee.....	73
Figura 11 Funcionarios	74
Figura 12 Activos	75
Figura 13 Custodios	76
Figura 14 Funcionarios	77
Figura 15 Funcionarios	77
Figura 16 Crear	79
Figura 17 Crear	80
Figura 18 Crear	80
Figura 19 Crear	81
Figura 20 Fichero	81
Figura 21 Fichero	82
Figura 22 Fichero	82
Figura 23 Fichero	83
Figura 24 Activo.....	84
Figura 25 Activo Rfid.....	84

Figura 26 ArchivoJustif	85
Figura 27 Área	85
Figura 28 Auditoría.....	86
Figura 29 Dirección.....	86
Figura 30 Funcionario.....	87
Figura 31 Sección.....	87
Figura 32 Subsección	88
Figura 33 Fichero	89
Figura 34 Activo Rfid.....	89
Figura 35 Activo Rfid.....	90
Figura 36 Fichero	90
Figura 37 Activo Rfid.....	91
Figura 38 Fichero	92
Figura 39 Fichero	93
Figura 40 Fichero	93
Figura 41 Fichero	94
Figura 42 Fichero	94
Figura 43 Fichero	95
Figura 44 Fichero	95
Figura 45 Modelo	96
Figura 46 Fichero	96
Figura 47 Energía	98
Figura 48 Red.....	99
Figura 49 Red.....	99
Figura 50 Memoria	100
Figura 51 Memoria	101

Figura 52 CPU	101
Figura 53 CPU	102

RESUMEN

El presente proyecto que se ha desarrollado para la Universidad de las Fuerzas Armadas ESPE consta de una aplicación web y una aplicación móvil para ser implementadas en el proceso de toma física de Inventarios para la Unidad de Bienes de esta. Se ha investigado sobre nuevas tecnologías para el control de inventarios entre las que se escogió la tecnología RFID, la misma que tiene como objetivo reemplazar a la tecnología actual de código de barras que se utilizan para etiquetar los bienes de la Universidad. En el desarrollo del presente proyecto se utilizó la metodología XP que me permitió realizar las aplicaciones web y móvil teniendo como expertos a las personas a cargo del proceso de la toma física y el apoyo de la unidad de Tecnologías de Computación de la ESPE, se ha integrado las nuevas aplicaciones con los sistemas actuales de control de inventario de la ESPE: Olympto y eSBye. El sistema web permite realizar la búsqueda de bienes asignados por el número de cédula o nombre del custodio o por la marca, modelo, serie, ubicación o código de los bienes, también permite conocer los bienes faltantes por custodio al momento de realizar la toma física. En el sistema web se podrá realizar la generación de un acta por custodio donde se encuentra los bienes faltantes, sobrantes y constatados. Por otro lado, la aplicación móvil posee la integración con la tecnología RFID, permitiendo realizar la lectura de los tags de RFID colocados en cada bien, también se puede realizar la búsqueda de los bienes por custodio o por características del bien a buscar. La aplicación permite la generación del acta de bienes por custodio, la cual se guarda en una carpeta interna del dispositivo que puede ser exportada a otro dispositivo para su almacenamiento o impresión.

PALABRAS CLAVE:

- **TECNOLOGÍA RFID**
- **CÓDIGO DE BARRAS**
- **TOMA FÍSICA**

ABSTRACT

Current Project has been developed for University of Armed Forces ESPE, it consists of a web application and a mobile application which will be implemented in the process of physical taking of the inventory by the Department of Goods of this University. There are new technologies for inventory control have been investigated, and RFID technology was chosen for this project whose main purpose to is replace to the current barcode technology which is nowadays used to label the University Goods. During the development of the present project the XP methodology was used the same which permitted to carry out web and mobile applications, having as experts people who are in charge of taking a physical inventory and the support of Computer Technologies Unit which belongs to ESPE, the new applications have been integrated with the current inventory control systems to control the inventory of ESPE: Olympo and eSBye. Web system allows to carry out the search of assigned goods with identification number, custodian name, brand, model, series, location or code of goods; it also permits to know the missing goods per custodian at the time of physical taking. In the web system it will be able to generate a report per custodian where missing, surplus and verified goods will be found. On the other hand, mobile application has the integration with RFID technology, allowing the reading of RFID tags stuck on each goods, it is also possible to carry out the search of goods per custodian or for the characteristics of the goods to be searched. Application permits the generations of reports of goods per custodian which is saved in an internal folder of the device which can be exported to another device to its storage or printing.

KEYWORDS:

- **RFID TECHNOLOGY**
- **BARCODE**
- **PHYSICAL INVENTORY TAKING**

CAPÍTULO I

INTRODUCCIÓN

ANTECEDENTES

La Universidad de las Fuerzas Armadas ESPE es considerada una de las más emblemáticas del país por su constante innovación y aporte al desarrollo productivo del Ecuador. Fundada en 1922, la Universidad se distingue por entregar soluciones prácticas a las necesidades y preocupaciones de la sociedad ecuatoriana, contribuyendo a la generación de nuevos conocimientos a través de la docencia, la investigación y la vinculación con la sociedad. En el 2014, fue catalogada por el prestigioso Ranking Mundial de Universidades QS entre las 250 mejores de América Latina y la cuarta mejor del Ecuador. Actualmente, nuestra universidad preside la REDU (Red de Universidades y Escuelas Politécnicas para la Investigación y Posgrados) conformada por más de 20 universidades ecuatorianas. La Universidad es parte del Sistema de Educación Superior del Ecuador, integrada por el campus matriz en Sangolquí, las sedes Latacunga y Santo Domingo de los Tsáchilas, así como las Unidades Académicas Especiales y el Instituto de Idiomas; cuenta con más de 13.000 estudiantes, entre civiles y militares, de ellos 8.309 son hombres y 5.606 son mujeres. Es un centro de educación superior público regulado por la Constitución de la República del Ecuador y la Ley Orgánica de Educación Superior. Luego de la firma del Estatuto de creación, el 26 de junio del 2013, y aprobado por el Consejo de Educación Superior (CES). Está dividida en varios departamentos y unidades (ESPE, 2021).

Una de las unidades que pertenece a la Universidad y que además cuenta con una regulación por parte del Estado es la Unidad de Bienes, entre los procesos que

realiza está la constatación física de bienes existentes y el control y seguimiento de estos junto con los custodios asignados, para el efecto cuenta con dos sistemas de inventario: eSByE y OLYMPO.

El eSByE, el cual tiene como finalidad mejorar el control financiero y administrativo de los bienes y existencias, que son propiedad de los organismos y entidades que conforman el presupuesto general del estado. El mismo permite la ejecución de los procesos relacionados al manejo, administración, registros contables y obtención de reportes de los bienes y existencias, para el análisis de la información y la toma de decisiones. La interoperabilidad de este aplicativo con el Sistema Integrado de Gestión Financiera (eSIGEF), permite al usuario obtener información conciliada, respecto a cada proceso que se esté ejecutando (Finanzas, 2009).

El otro sistema de inventario que actualmente sigue usando la Universidad es el OLYMPO, este permite llevar un registro histórico de todos los bienes de la ESPE, mientras que eSByE se utiliza para llevar el registro de los bienes a partir del año 2017 por lo que no tiene la información anterior a ese año.

La forma en la que se lleva el control de los bienes de la Universidad es por medio de etiquetas de códigos de barras, los mismos que son impresos por una persona encargada de la Unidad a cargo y luego es colocado en cada bien. Actualmente se pretende utilizar una tecnología que reemplace al código de barras para lograr disminuir el tiempo al momento de realizar la toma física.

Con el avance de la tecnología se han definido nuevas formas para identificar un bien y controlar de mejor manera un inventario, uno de estos sistemas se conoce como RFID que significa identificación por radiofrecuencia.

Este es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio. Las tecnologías RFID se agrupan dentro de las denominadas auto ID (identificación automática) (Lázaro Guillén Lázaro, 2011).

Aunque en la actualidad la tecnología más extendida para la identificación de objetos es la de los códigos de barras, éstos presentan algunas desventajas, como la imposibilidad de ser reprogramados. El origen de la tecnología RFID consistió en usar chips de silicio que pudieran transferir los datos que almacenaban al lector sin contacto físico, de forma equivalente a los lectores de infrarrojos utilizados para leer los códigos de barras (Peris, 2008).

Las etiquetas RFID son unos dispositivos pequeños, similares a una pegatina, que pueden ser adheridas o incorporadas a un bien del inventario. Contienen antenas para permitirles recibir y responder a peticiones por radiofrecuencia desde un emisor-receptor RFID. Las etiquetas pasivas no necesitan alimentación eléctrica interna, mientras que las activas sí lo requieren. Una de las ventajas del uso de radiofrecuencia (en lugar, por ejemplo, de infrarrojos) es que no se requiere visión directa entre el emisor y receptor (Peris, 2008).

Este sistema se pretende implementar en la Universidad de las Fuerzas Armadas – ESPE sede Sangolquí para la mejora de la toma física de los bienes pertenecientes a la misma.

PROBLEMÁTICA

Actualmente la Universidad de la Fuerzas Armadas - ESPE utiliza 2 sistemas para realizar la toma física y para el control de los bienes: el primero es el sistema eSByE que opera desde el año 2017 a la actualidad. El segundo sistema es el OLYMPO el cual posee la información histórica de los bienes, permiten tener el inventario de bienes de manera digital.

Por otro lado, de acuerdo con el reglamento general sustitutivo para la administración, utilización, manejo y control de los bienes e inventarios del sector público que cita el artículo 54 de la sección III Constatación Física, el cual menciona que “se debe realizar al menos una vez al año una constatación física de los bienes e inventario de la universidad, con el objetivo de confirmar la ubicación, existencia real y la nómina de los responsables de su tenencia y conservación de los bienes”.

Para poder cumplir con lo dispuesto en el artículo mencionado anteriormente la Unidad de Bienes de la ESPE, ha tomado la decisión de realizar la toma física por departamentos y unidades, y así lograr realizar el control de bienes de toda la universidad.

El proceso para la toma física de inventarios de la Universidad se realiza de la siguiente manera:

- Una persona asignada de la Unidad de Bienes imprime la lista de bienes con sus respectivos custodios.
- Dicha persona se debe acercar a los diferentes puestos de trabajo.
- Se realiza la lectura de los bienes por medio del código de barras.
- Se compara los bienes de la lista impresa con los bienes que se hizo la lectura.

- Genera un acta donde constan los bienes faltantes, sobrantes o alguna novedad acerca de los mismos.
- En caso de bienes faltantes el custodio debe justificar la causa presentando documentación de descargo.
- Firman el acta tanto el custodio y la persona asignada de la Unidad de Bienes para realizar la toma física.

Durante este proceso las personas de la Unidad de Bienes han identificado varios inconvenientes a la hora de realizar la toma física, los cuales son:

- Que las etiquetas de códigos de barras son a veces ilegibles
- La falta de etiquetas de códigos de barras en algunos bienes
- El tiempo en encontrar la etiqueta de código de barras en cada bien
- La gran cantidad de bienes que forman parte de la toma física

El número aproximado actual de bienes que la Universidad de las Fuerzas Armadas – ESPE Matriz posee es de 93298, y la unidad de bienes cuenta con 6 administrativos para realizar todos los procesos además de la toma física.

JUSTIFICACIÓN

De acuerdo con el reglamento general sustitutivo para la administración, utilización, manejo y control de los bienes e inventarios del sector público que cita el artículo 54 de la sección III Constatación Física, el cual menciona que “se debe realizar al menos una vez al año una constatación física de los bienes e inventario de la universidad, con el objetivo de confirmar la ubicación, existencia real y la nómina de los responsables de su tenencia y conservación de los bienes”.

Considerando la cantidad de bienes: 93298 aproximadamente, de acuerdo con datos estadísticos de la Unidad de Bienes es necesario un tiempo de 19 meses para realizar la toma física completa de la Universidad de las Fuerzas Armadas ESPE – Matriz.

En este contexto el sistema va a hacer desarrollado usando la tecnología RFID como reemplazo de los códigos de barras, integrado a la aplicación móvil lo que ayudara en el proceso de la toma física, ya que no depende de una computadora para realizar la consulta de los bienes que posee cada custodio y además se puede realizar la lectura varios bienes de un área específica simultáneamente, dependiendo del equipo usado para la lectura de tags.

La aplicación emitirá el acta de la constancia de bienes faltantes y/o sobrantes de acuerdo con el proceso de toma física, permitiendo al personal de bienes enviar dicha acta luego de la constatación física, con lo cual agiliza el proceso de manera positiva, para su posterior firma.

OBJETIVOS

Objetivo general

- Desarrollar e implementar una aplicación web con integración de una aplicación para dispositivos móviles que permita el control de inventario de bienes, utilizando tecnología RFID, para la Universidad de la Fuerzas Armadas – ESPE sede Sangolquí.

Objetivos Específicos

- Revisar y analizar el sistema eSByE y el sistema OLYMPO de inventarios para el control de bienes que actualmente utiliza la universidad.
- Investigar sobre el funcionamiento de la tecnología RFID
- Cumplir con las etapas del proceso de desarrollo de software para generar una aplicación web y móvil.
- Realizar pruebas de integración entre la tecnología RFID y el sistema web y móvil.
- Utilizar metodologías que permitan una adecuada implementación de la aplicación móvil y web en la Unidad de Bienes de la ESPE – Matriz.
- Integración con los sistemas que actualmente interviene en la toma física.
- Realizar una capacitación del funcionamiento de la aplicación WEB y móvil.

ALCANCE

El proyecto contará con: un sistema web y una aplicación móvil que utilizará la tecnología RFID para la lectura de las etiquetas que utilicen radio frecuencia. El desarrollo de estas aplicaciones se basa en los sistemas eSByE y OLYMPO que actualmente utiliza la universidad para el control y la constatación física de los bienes.

El sistema web permitirá a las personas que realizan la toma física consultar los bienes por los siguientes criterios de búsqueda:

- Custodio, por su número de identificación o sus nombres.
- Marca
- Modelo
- Serie
- Ubicación, por departamento, sección, subsección o unidad.

Desde el sistema web la persona de bienes que realice la constatación física puede generar el acta por custodio.

También desde el sistema web se puede visualizar los procesos pendientes en la constatación física. El proceso finalizará una vez se haya generado el acta que corresponda a los bienes constatados por custodio y sea firmado por las partes que intervienen.

La aplicación móvil al igual que el sistema web permitirá realizar la búsqueda de los bienes por las siguientes opciones de búsqueda:

- Custodio, por su número de identificación o sus nombres.
- Marca
- Modelo
- Serie

- Ubicación, por departamento, sección, subsección o unidad.

Por otro lado, la aplicación contará con la integración con la tecnología RFID, lo que permitirá realizar la lectura de las etiquetas que utilicen la tecnología de radio frecuencia ubicados en los bienes.

También desde la aplicación móvil se puede generar el acta en base a la lectura de los bienes realizados y en comparación con los bienes que tenga asignado el custodio al que se le esté realizando la constatación física.

Para tener una mayor seguridad tanto el sistema web como la aplicación móvil integrados a los sistemas eSByE y Olympo, solo podrán ser usadas por las personas que pertenezcan a la Unidad de Bienes de la ESPE – Sede Sangolquí, permitiendo tener un mayor control de la información sensible y confidencial.

FACTIBILIDAD

Factibilidad se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señaladas. Generalmente la factibilidad se determina sobre un proyecto. Estos resultados se entregan a la gerencia, quienes son los que aprueban la realización del sistema informático. El estudio de factibilidad es una tarea que suele estar organizada y realizada por los analistas de sistemas. El estudio consume aproximadamente entre un 5% y un 10% del costo estimado total del proyecto, y el período de elaboración de este varía dependiendo del tamaño y tipo de sistema a desarrollar (Edu, 2021).

En el presente proyecto se han considerado:

Factibilidad técnica

Después de haber cursado la carrera de ingeniería en sistemas e informática, varios cursos tomados, la experiencia laboral en desarrollo de software. Los conocimientos adquiridos son:

- Telecomunicaciones
- Aplicaciones web
- Web Services
- Aplicaciones móviles
- Aplicaciones distribuidas
- Base de datos
- Lógica matemática
- Lógica de programación

Requerimientos de hardware

Los requerimientos mínimos para el desarrollo de este sistema son los siguientes:

- Portátil marca Dell core I7, con una RAM de 16 GB, una tarjeta de video radeon.
- Impresora Epson multifunción
- Flash Memory 8GB
- Tags RFID UHF Gen2
- Lector RFID 1128 Bluetooth UHF

Requerimientos de software

Los requerimientos mínimos para el desarrollo de este sistema son los siguientes:

- Android studio 4.1
- Angular CLI 7.6
- S.O. Windows 10
- Spring Boot 2.3.0
- Visual Studio Code 1.53

Factibilidad económica

Para la ejecución del proyecto se necesita un presupuesto aproximado de \$5.998 dólares. Al ser un proyecto de titulación, dicho valor será asumido en su totalidad por el estudiante.

Tabla 1*Proyección de costos*

	Descripción	Costo unitario	Cantidad	Total
COSTOS FIJOS				
Hardware				
	Portátil Dell i7	\$800	1	\$800
	Impresora Epson	\$200	1	\$200
	Flash 8GB	\$20	1	\$20
Software				
	SO Windows	\$0	1	\$0
	Android Studio	\$0	1	\$0
	Visual Studio Code	\$0	1	\$0
Talento humano				
	Horas de investigación	\$8	16	\$128
	Horas de desarrollo	\$8	60	\$480
	Horas de pruebas	\$8	20	\$160
Otros				
	Internet	\$25	5	\$125
COSTOS VARIABLES				
Hardware				
	Lector RFID	\$1000	1	\$1000
	Antena RFID	\$1500	1	\$1500
	Tags RFID	\$1	100	\$100
Talento humano				
	Horas extras	\$5	20	\$100
Transporte				
	Gasolina	\$40	5	\$200
Otros				
	Hojas de impresión	\$3,50	10	\$35
	Tintas	\$10	8	\$80

Descripción	Costo unitario	Cantidad	Total
CD	\$2	10	\$20
Imprevistos	\$1000	1	\$1000
		TOTAL:	\$5.948

Nota. Se detalla los diferentes rubros que intervienen en el desarrollo del proyecto.

CAPÍTULO II

MARCO TEÓRICO

Control de inventarios

El control de inventario se refiere a todos los procesos que ayudan al suministro, accesibilidad y almacenamiento de productos para minimizar los tiempos y costos relacionados con el manejo de una empresa, por ende, es un mecanismo que contribuye a que la organización administre de manera eficiente el movimiento y almacenamiento de mercancía, así como el flujo de información y recursos que resultan de ello. Involucra distintos aspectos, pero en términos generales se subdivide en lo correspondiente a gestión y optimización (Salas, 2009).

La gestión se encarga de mantener la productividad en las operaciones relacionadas con la administración del inventario, mientras que la optimización se ocupa de incrementar las ganancias de la empresa provenientes del uso y manejo de este (Salas, 2009).

Para realizar el control de inventario se debe tomar las siguientes consideraciones:

- Mantener un catálogo con los productos que se manejan: Es importante organizar la información que se posea sobre las existencias y complementarla con detalles, además de depurarla de manera constante, esto facilita la visualización de necesidades y oportunidades del inventario en tiempo real (Salas, 2009).

- Clasificar los productos: Separar por grupos semánticos: ya sea por proveedor, éxito de venta o rezago, hará más accesible la información del inventario, así

como acelerará la toma de medidas necesarias (reabastecimiento, re-ofertas, entre otras) (Salas, 2009).

- Para lograr una mejor gestión y actualización del inventario se debe realizar lo siguiente:
 - Inventario perpetuo: Se hace un registro continuo (día a día) de la producción y venta de artículos, por lo que se puede conocer el costo del inventario y las existencias en el mismo sin tener que determinar una fecha de inventariado (Salas, 2009).
 - Inventario periódico: Se eligen fechas específicas para contabilizar la mercancía según las necesidades de la empresa (Salas, 2009).
- Se necesita entender los siguientes conceptos:
 - Stock máximo. Límite de unidades, por artículo, que se desea mantener en almacén según las ganancias y costos que representen (Salas, 2009).
 - Stock mínimo (de seguridad). Existencias mínimas, por artículo, que se requieren en almacén considerando labores de reabastecimiento y las posibles pérdidas que su carencia signifique (Salas, 2009).
 - Punto re-orden. Momento (medido por la cantidad de existencias) en el que se deben realizar órdenes de reabastecimiento tomando en cuenta tiempos y costos de proveedores (Salas, 2009).
- Monitorear y actualizar de manera constante la información recopilada, y el sistema utilizado. Así se podrá solicitar la compra de unidades antes de que se agoten, rotar mercancía generando campañas atractivas, reconocer la utilidad de los métodos implementados, identificar áreas de oportunidad e incorporar mejoras (Salas, 2009).

- Integrar herramientas especializadas acelera el cumplimiento de las actividades relacionadas con el inventario al permitir el acceso a interfaces de gestión automatizadas (Salas, 2009).

Beneficios de ejercer un buen control de inventario

- Información importante y vigente sobre las existencias, mejorando la toma de decisiones
- Resalta la efectividad de la empresa y la eficiencia de sus procedimientos
- Aumenta la calidad de servicio al cliente
- Ayuda a la identificación pertinente de estacionalidad o flujo de los productos
- Optimiza la inversión de recursos (económicos, humanos y temporales)
- Permite tener un mejor conocimiento y control de las entradas, salidas y localización de mercancía, de esta manera se reducen pérdidas, se optimiza el espacio

BASE DE DATOS

ORACLE

Oracle es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hace que sólo se vea en empresas muy grandes y multinacionales, por norma general. En el desarrollo de páginas web pasa lo mismo: como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, MySQL, SQL Server, etc (Heurtel, 2015).

Las principales características de una base de datos orientada a objetos Oracle son:

- Control de acceso, porque utiliza tecnología avanzada que vigila quién accede a los datos.
- Una herramienta cómoda de utilizar, ya que es muy intuitiva.
- Gestión segura de la protección de los datos, así como de las copias de seguridad.
- Gracias a la vista materializada de Oracle se pueden visualizar los datos de consulta en tablas con el formato de filas y columnas.

Existen muchos tipos diferentes de bases de datos. La mejor base de datos para una organización específica depende de cómo pretenda la organización utilizar los datos (Oracle, 2021)

- Bases de datos relacionales. Las bases de datos se hicieron predominantes en la década de 1980. Los elementos de una base de datos relacional se organizan como un conjunto de tablas con columnas y filas. La tecnología de bases de datos relacionales proporciona la forma más eficiente y flexible de acceder a información estructurada (Oracle, 2021).
- Bases de datos orientadas a objetos. La información de una base de datos orientada a objetos se representa en forma de objetos, como en la programación orientada a objetos (Oracle, 2021).
- Bases de datos distribuidas. Una base de datos distribuida consta de dos o más archivos que se encuentran en sitios diferentes. La base de datos puede almacenarse en varios ordenadores, ubicarse en la misma ubicación física o repartirse en diferentes redes (Oracle, 2021).

- Almacenes de datos. Un repositorio central de datos, un almacén de datos es un tipo de base de datos diseñado específicamente para consultas y análisis rápidos (Oracle, 2021).
- Bases de datos NoSQL. Una base de datos NoSQL, o base de datos no relacional, permite almacenar y manipular datos no estructurados y semiestructurados (a diferencia de una base de datos relacional, que define cómo se deben componer todos los datos insertados en la base de datos). Las bases de datos NoSQL se hicieron populares a medida que las aplicaciones web se volvían más comunes y complejas (Oracle, 2021).
- Bases de datos orientadas a grafos. Una base de datos orientada a grafos almacena datos relacionados con entidades y las relaciones entre entidades (Oracle, 2021).
- Bases de datos OLTP. Una base de datos OLTP es una base de datos rápida y analítica diseñada para que muchos usuarios realicen un gran número de transacciones (Oracle, 2021).

BACK-END

Spring boot

Spring Boot es una de las tecnologías dentro del mundo de Spring de las que más se está hablando últimamente. Fundamentalmente existen tres pasos a realizar. El primero es crear un proyecto Maven/Gradle y descargar las dependencias necesarias. En segundo lugar, desarrollamos la aplicación y en tercer lugar la desplegamos en un servidor. Si nos ponemos a pensar un poco a detalle en el tema, únicamente el paso dos es una tarea de desarrollo. Los

otros pasos están más orientados a infraestructura. No deberíamos tener que estar eligiendo continuamente las dependencias y el servidor de despliegue (VMware, 2021).

Spring Boot nos proporciona una serie de dependencias, llamadas starters, que podemos añadir a nuestro proyecto dependiendo de lo que necesitemos: crear un controlador REST, acceder a una base de datos usando JDBC, conectar con una cola de mensajes Apache ActiveMQ, etc (VMare, 2021).

Una vez añadimos un starter, éste nos proporciona todas las dependencias que necesitamos, tanto de Spring como de terceros. Además, los starters vienen configurados con valores por defecto, que pretenden minimizar la necesidad de configuración a la hora de desarrollar. Un buen ejemplo es el de spring-boot-starter-actuator: una vez que añadimos la dependencia, nuestra aplicación empezará a generar métricas tanto de la JVM como de la aplicación en sí (latencias, errores, etc) (VMare, 2021).

Al igual que con Spring Framework, cualquier configuración puede ser modificada de ser necesario: desde el puerto en el que la aplicación escucha peticiones, hasta el banner que sale por consola al arrancar la aplicación (VMare, 2021).

FRONT-END

Angular

Angular es una plataforma de desarrollo, construida sobre TypeScript. Como plataforma, Angular incluye (Angular, 2021):

- Un marco basado en componentes para crear aplicaciones web escalables (Angular, 2021).

- Una colección de bibliotecas bien integradas que cubren una amplia variedad de características, que incluyen enrutamiento, administración de formularios, comunicación cliente-servidor y más (Angular, 2021).
- Un conjunto de herramientas para desarrolladores que lo ayudarán a desarrollar, compilar, probar y actualizar su código (Angular, 2021).

Con Angular, está aprovechando una plataforma que puede escalar desde proyectos de un solo desarrollador hasta aplicaciones de nivel empresarial. Angular está diseñado para que la actualización sea lo más fácil posible, para que pueda aprovechar los últimos desarrollos con un mínimo de esfuerzo. Lo mejor de todo es que el ecosistema Angular consta de un grupo diverso de más de 1.7 millones de desarrolladores, autores de bibliotecas y creadores de contenido (Angular, 2021).

IONIC Framework

Ionic es una estructura tecnológica (Framework) de código abierto que se utiliza en el desarrollo de aplicaciones móviles híbridas, es decir, se combinan el HTML5, CSS y JavaScript dando como resultado aplicaciones con una interfaz amigable e intuitiva para el usuario que luego se comercializan o descargan en plataformas como Android o IOs (Ionic, 2021).

La base de Ionic está desarrollada sobre AngularJs y Cordova, vio la luz en 2013 con la única intención de que desarrolladores pudieran crear aplicaciones móviles híbridas con la particularidad y beneficios de los dos framework sobre los que fue construida (Ionic, 2021).

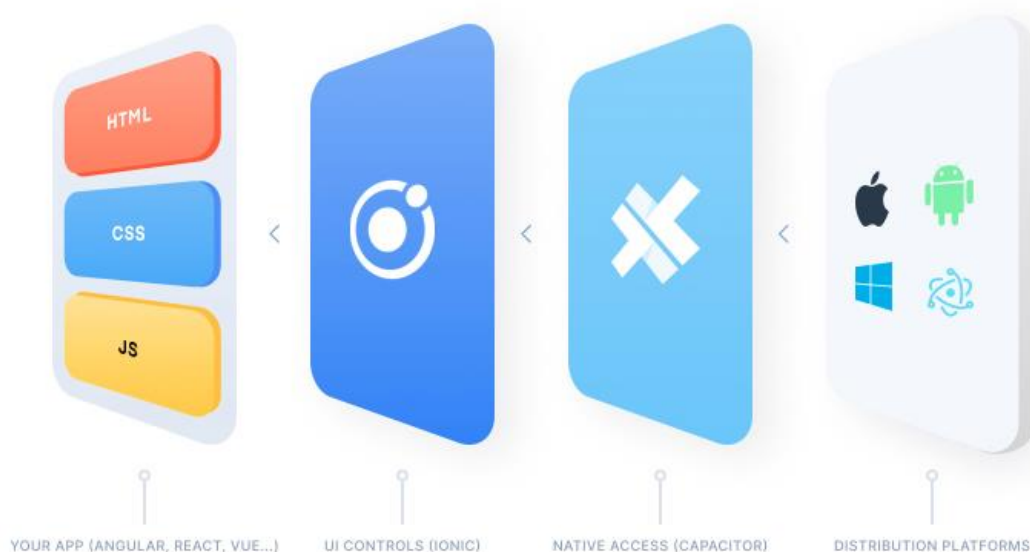
Una de las principales ventajas de trabajar con Ionic es que aprovecha todos los plugins (Hardware, software, imágenes, texto, códigos QR, etc) del marco de desarrollo móvil Cordova. En 2016 se actualiza a la versión Ionic 2, en donde llega la modularización, una de las actualizaciones más completas de este framework, que permite separarlo por partes: core,

angular, native, etc. La actualización de Ionic 3 es prácticamente imperceptible, aunque si trae novedades en su rendimiento. El nuevo cambio importante en la herramienta de desarrollo de aplicaciones móviles es su versión Ionic 4 que llegó a primeros de este año. Ionic 4 empieza con reemplazar AngularJS por Angular moderno. El conjunto de componentes de esta herramienta utiliza elementos personalizados y las API DOM de Shadow disponibles en todos los navegadores modernos para dispositivos móviles y de escritorio (Ionic, 2021).

En la siguiente figura se puede comprender el funcionamiento:

Figura 1

Ionic



Nota: Flujo de IONIC (Ionic, 2021).

METODOLOGÍA

Para realizar un mejor análisis de la metodología que se adapte más al desarrollo del presente proyecto, se presenta en la siguiente figura un cuadro comparativo de las metodologías más comunes:

Figura 2

Metodologías

	XP	SCRUM	RUP
BREVE DESCRIPCION	Modelo en el que se define un plan para desarrollar y liberar software. Y además poder revisarlo para incorporar nuevas funcionalidades.	Modelo en el que se mantiene la participación activa de todos los miembros del proyecto.	Se caracteriza por ocupar el modelo iterativo e incremental. Esta centrado en la arquitectura.
TIPO DE PROYECTO DE SOFTWARE	Aplicaciones móviles	Proyectos pequeños	Grandes empresas
PROGRAMADOR / RELACION CON EL USUARIO	Programador con habilidades blandas y trabajo en equipo	Certificados y/o con conocimientos en Metodologías Ágiles	Certificados y/o con conocimientos en UML
ETAPAS	Definir roles Estimar el esfuerzo Elegir que construir Programar Repetir	Planeamiento Montaje Desarrollo Liberación	Inicio Elaboración Construcción Transición
CARACTERÍSTICAS PROPIAS DEL MODELO	Pone énfasis en la programación	Pone énfasis en la colaboración activa del cliente	Ocupa el modelo incremental y se centra en usar casos de uso

Nota: Cuadro comparativo (JENNIFER, 2018).

Metodología XP

La metodología XP es un conjunto de técnicas que dan agilidad y flexibilidad en la gestión de proyectos. También es conocida como Programación Extrema (Extreme Programming) y se centra en crear un producto según los requisitos exactos del cliente. De ahí, que le involucre al máximo durante el método de gestión del desarrollo del producto (Beck, 2002).

El uso de esta metodología supone, para muchos teóricos, una aproximación a la calidad óptima del producto. Pues durante el ciclo de vida del software, ocurren cambios naturales. Es más, cuantos más cambios, puede que más cerca estemos del mejor resultado que espera nuestro cliente. Por eso, este cambio constante en el proyecto se llega a considerar como favorable. Y si podemos aplicar una manera dinámica de gestionarlos, mejor. Esta forma es conocida como metodología XP (Beck, 2002).

Las planificaciones:

Por una parte, se deben planificar los plazos temporales del proyecto basándose en las exigencias del cliente. En base a las estimaciones de coste y la dificultad del proyecto se marcan las prioridades y las fechas, no siempre de forma precisa, pero sí orientativa (Fuentes, 2015).

Con la entrega de la planificación efectuada, se desarrolla la de la iteración en el que cada dos semanas se marca el rumbo y se entrega el software útil después de cada uno de estos periodos bisemanales. Con esto se consigue que el nivel de precisión sea mucho mayor, las estimaciones sobre los costes sean más exactas y la información mucho más transparente (Fuentes, 2015).

Pruebas

Continuamente se han de efectuar una serie de pruebas automatizadas en base a los requisitos del cliente para comprobar que todo funcione correctamente. Éstas han de hacerse de forma periódica y automática.

Con las planificaciones comentadas anteriormente se incluyen las entregas al final de cada iteración, éstas serán siempre con el software probado y funcionando correctamente y será facilitado al cliente, que puede utilizarlo para cualquier propósito, incluso para el usuario final. Los equipos XP también pueden hacer entregas a otros usuarios finales.

Diseño y programación:

El diseño del programa suele ser simple y basado en la funcionalidad del sistema y se lleva a cabo durante todo el proyecto, tanto durante la planificación de la entrega como en el de la iteración (Fuentes, 2015).

La programación del software se hace siempre en pareja, lo que se llama programar a dos manos. Se asegura con este método que al menos un programador conoce y controla la

labor de otro y queda revisado. La ventaja es que se produce mejor código que en base a un programador, aunque la dificultad de esta sea mayor (Fuentes, 2015).

El código es de todos, con el desarrollo de las pruebas automáticas y la programación a dos manos se incluye también la posibilidad de que cualquiera pueda añadir y retocar parte del código, aunque eso sí, deba ser un estilo común y cuyo resultado sea como si sólo lo hubiera hecho una persona (Fuentes, 2015).

FORMAS DE CONTROL FÍSICO DE INVENTARIO

Código de barras

Un código de barras es una agrupación de líneas de distinto grosor y colocadas a diferente distancia unas de otras que, en conjunto, contienen una cantidad concreta de información. Estas líneas no son otra cosa que una representación de caracteres, en la mayoría de los casos, números (LATIENDEDELASBARRAS, 2019).

Para leer un código de barras es necesario lo siguiente:

- El dispositivo de entrada convierte la energía luminosa en electricidad (Luis, 2021).
- Un decodificador convierte la señal eléctrica en datos (Luis, 2021).
- El decodificador permite ingresar la información del código de barras a algún sistema (Luis, 2021).

Figura 3

Lector Código de Barras



Nota: El gráfico representa el uso de un lector de código de barras (Luis, 2021).

Características del código de barras:

- Densidad: El código de barras se mide por la densidad y no por la longitud que tenga. La densidad se refiere al ancho de la barra o espacio más delgado que hay en el símbolo (Luis, 2021).
- Wide to Narrow Ratio (WNR): Hace referencia a la anchura del elemento más delgado con respecto al más grueso (Luis, 2021).
- Quiet Zone: Son las zonas en blanco al comienzo y al final del código de barras. Es una zona necesaria para que se puede leer el símbolo (Luis, 2021).
- Repetidos por productos, únicos por tipo de producto (Luis, 2021).

Tecnología RFID

RFID son las siglas en inglés Radio Frequency Identification, que significa identificación por radiofrecuencia (Vicent, 2020).

Se trata de un sistema de identificación, almacenamiento y transmisión de datos remotos que utiliza dispositivos llamados etiquetas (tags), tarjetas o transponders RFID activos. Esta tecnología identifica a través de un lector, sin contacto y a distancia, una tarjeta o etiqueta (tag). Su función principal es transmitir la identidad de un objeto por medio de ondas de radio. Las etiquetas RFID (tag) es un chip, pueden ser activas o pasivas (Vicent, 2020).

Las pasivas no requieren alimentación eléctrica interna porque se alimentan de la propia señal, que, mediante la antena, activa el circuito. Mientras que las activas sí traen o una pequeña batería, o van incluidas en un equipo como un teléfono (Vicent, 2020).

La tecnología RFID o Radio Frecuencia, es la forma que tiene de comunicarse los objetos modernos. Los diferentes usos de la tecnología RFID dan respuesta a una amplia gama de procesos empresariales (Gomez, 2021).

Es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID y su propósito fundamental es transmitir la identidad de un objeto mediante ondas de radio. Las tecnologías RFID se agrupan dentro de las denominadas auto ID (identificación automática) (Peris, 2008).

Hoy en día la tecnología más utilizada a nivel mundial para identificar objetos es la de código de barras, pero hay que tomar en cuenta que presenta algunas desventajas, como la imposibilidad de ser reprogramados. El origen de la tecnología RFID consistió en usar chips de silicio que pudieran transferir los datos que almacenaban al lector sin

contacto físico, de forma equivalente a los lectores de infrarrojos utilizados para leer los códigos de barras (Peris, 2008).

Automatizar con fiabilidad la introducción de datos en los ordenadores, en tiempo real y sin dedicar recursos humanos (Philip Kotler, Kevin Lane Keller, 2018).

El código de barras es fiable sólo cuando lo lees, cuando se realiza la acción de lectura. La tecnología RFID viene a garantizar su lectura automática en cualquier momento y en cualquier lugar (Antonio Hernando Grande, Alejandro Úbeda Maeso, Javier López Facal, 2010).

La tecnología RFID puede ser útil allí dónde tengan que realizarse continuados registros de datos, o dónde no se llevan a cabo por imposibilidad humana o coste. No obstante, como un resumen general, el RFID puede ser útil tanto en situaciones internas de las empresas, en procesos propietarios o en procesos de intercambio de información y mercancías entre diferentes agentes de la cadena. Como ejemplo (García, 2008):

- Control de inventarios.
- La trazabilidad.
- Movimiento de mercancías.
- Control de procesos industriales.
- Automatización de procesos.
- Asset tracking.
- Control de accesos y vehículos.
- Optimización en hospitales.
- Logística y almacenamiento.

El RFID tiene como propósito fundamental transmitir la identidad de un objeto o producto cuando es activado por una antena RFID. Hay diferentes frecuencias RF y cada una de ellas conlleva un uso distinto. El uso de las diferentes frecuencias está totalmente generalizado y homologado a nivel mundial por las diferentes entidades (Vicent, 2020).

Principales frecuencias RFID:

- UHF RFID Pasivo (860-960 mhz)
- NFC y HF RFID Pasivo (13,56Mhz)

Tabla 2

Comparación entre código de barras vs RFID

Características	Código de barras	RFID
Capacidad	Espacio limitado	Mayor cantidad de espacio
Tipo de lectura	Sobre la superficie	Puede atravesar varios materiales
Multilecturas	Solo se puede leer una a la vez	Se pueden leer varios objetos a la vez
Durabilidad	Se puede borrar, despegar o romper las etiquetas	Larga durabilidad
Proceso	El proceso es de forma manual	Puede automatizarse

Nota. Se realizó una comparación entre la tecnología de código de barras y RFID para determinar las ventajas y desventajas de estas.

Tipos de dispositivos RFID

Se pueden clasificar los dispositivos RFID en emisores y receptores. Estos últimos se encargan de detectar la señal que emiten los primeros. Los emisores pueden

ser de dos tipos: pasivos o activos, cuya diferencia radica en si cuentan con una fuente de alimentación o no (UIV, 2017).

Las etiquetas pasivas tienen un alcance limitado, de unos 5 metros. En el caso de las que llevan alimentación propia, el alcance puede ser 100 veces mayor. Las de menor tamaño suelen tener un rango bastante menor, de aproximadamente 10 metros (UIV, 2017).

Cómo funciona el sistema RFID

El funcionamiento de esta tecnología es bastante simple. Un receptor envía una señal continua dentro de un radio de alcance concreto. Cuando una etiqueta entra en contacto con esta, envía información que el lector interpreta según esté programado. Dependiendo de las características de la etiqueta, se puede grabar o editar la información. Esto resulta muy útil en aplicaciones como la logística, donde es posible tener un control específico de stock o localización de envíos (UIV, 2017).

Aplicaciones presentes y futuras

En la actualidad, esta tecnología se utiliza sobre todo para dispositivos contra el robo. Al tratarse de una forma económica de localizar objetos, se pueden colocar etiquetas en prendas de ropa y todo tipo de artículos. Se pueden encontrar en supermercados, bibliotecas e incluso para desbloquear sistemas antirrobo instalados en cerraduras. Los chips que tienen las mascotas también son identificadores RFID. Al igual que algunas tarjetas de identificación y bancarias. También existen chips que se pueden implantar en humanos como sistemas de localización o identificación especial (UIV, 2017).

En cuanto a las aplicaciones potenciales, podemos decir de RFID que es un sustituto ideal de los códigos de barra. Tienen un mayor alcance y se pueden leer con más rapidez. Incluso se pueden recodificar si fuera necesario. Otra propuesta es la señalización inteligente en carreteras, con balizas que indicarían al coche los límites de velocidad (UIV, 2017).

A grandes rasgos, esta es la tecnología inalámbrica conocida como RFID. Algo que suena a nuevo, pero que en realidad es una de las que más tiempo llevan en uso (UIV, 2017).

Arquitectura

El modo de funcionamiento de los sistemas RFID es simple. La etiqueta RFID, que contiene los datos de identificación del objeto al que se encuentra adherida, genera una señal de radiofrecuencia con dichos datos. Esta señal puede ser captada por un lector RFID, el cual se encarga de leer la información y pasarla en formato digital a la aplicación específica que utiliza RFID (Peris, 2008).

Un sistema RFID consta de los siguientes tres componentes:

Etiqueta RFID o transpondedor: compuesta por una antena, un transductor radio y un material encapsulado o chip. El propósito de la antena es permitirle al chip, el cual contiene la información, transmitir la información de identificación de la etiqueta. Existen varios tipos de etiquetas. El chip posee una memoria interna con una capacidad que depende del modelo y varía de una decena a millares de bytes (Peris, 2008). Hay varios tipos de memoria:

Solo lectura: el código de identificación que contiene es único y es personalizado durante la fabricación de la etiqueta (González, 2021).

De lectura y escritura: la información de identificación puede ser modificada por el lector (González, 2021).

- Anticolisión: se trata de etiquetas especiales que permiten que un lector identifique varias al mismo tiempo (habitualmente las etiquetas deben entrar una a una en la zona de cobertura del lector) (Alejandro Medina Santiago, Ciclalli Cabrera, Jorge Luis Camas, 2012).

- Lector de RFID o transceptor: compuesto por una antena, un transceptor y un decodificador. El lector envía periódicamente señales para ver si hay alguna etiqueta en sus inmediaciones. Cuando capta una señal de una etiqueta (la cual contiene la información de identificación de esta), extrae la información y se la pasa al subsistema de procesamiento de datos (Alejandro Medina Santiago, Ciclalli Cabrera, Jorge Luis Camas, 2012).

- Subsistema de procesamiento de datos o middleware RFID: proporciona los medios de proceso y almacenamiento de datos (Alejandro Medina Santiago, Ciclalli Cabrera, Jorge Luis Camas, 2012).

Tipos de Etiqueta RFID

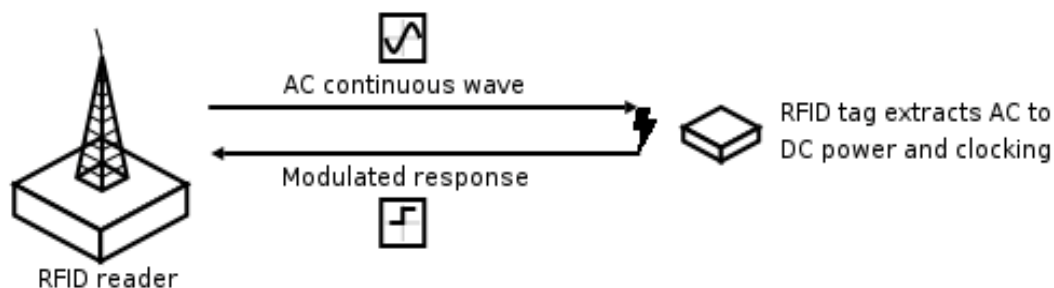
Las etiquetas RFID pueden ser activas, semipasivas (también conocidos como semiactivos o asistidos por batería) o pasivos. Las etiquetas pasivas no requieren ninguna fuente de alimentación interna y son dispositivos puramente pasivos (solo se activan cuando un lector se encuentra cerca para suministrarles la energía necesaria). Los otros dos tipos necesitan alimentación, típicamente una pila pequeña (Peris, 2008).

La gran mayoría de las etiquetas RFID son pasivas, que son mucho más baratas de fabricar y no necesitan batería. En 2004, estas etiquetas tenían un precio desde 0.40 \$, en grandes pedidos, para etiquetas inteligentes, según el formato, y de 0.95 \$ para etiquetas rígidas usados frecuentemente en el sector textil encapsulados en el PP o epoxi. El mercado de RFID universal de productos individuales será comercialmente viable con volúmenes muy grandes de 10 000 millones de unidades al año, llevando el coste de producción a menos de 0.05 \$ según un fabricante. La demanda actual de chips de circuitos integrados con RFID no está cerca de soportar ese coste. Los analistas de las compañías independientes de investigación como Gartner and Forrester Research convienen en que un nivel de precio de menos de 0.10 \$ (con un volumen de producción de 1 000 millones de unidades) solo se puede lograr en unos 6 u 8 años, 6 lo que limita los planes a corto plazo para una adopción extensa de las etiquetas RFID pasivas. Otros analistas creen que esos precios serían alcanzables dentro de 10 a 15 años (Peris, 2008).

A pesar de que las ventajas en cuanto al coste de las etiquetas RFID pasivas con respecto a las activas son significativas, otros factores, incluyendo exactitud, funcionamiento en ciertos ambientes como cerca del agua o metal, y fiabilidad, hacen que el uso de etiquetas activas sea muy común hoy en día (Bonet, 2018).

Figura 4

RFID READER



Nota. Como trabaja el lector de la tecnología de radio frecuencia (Flamarique, 2009).

Para comunicarse, las etiquetas responden a peticiones o preguntas generando señales que a su vez no deben interferir con las transmisiones del lector, ya que las señales que llegan de las etiquetas pueden ser muy débiles y han de poder distinguirse. Además de la reflexión o backscatter, puede manipularse el campo magnético del lector por medio de técnicas de modulación de carga. El backscatter se usa típicamente en el campo lejano y la modulación de carga en el campo próximo (a distancias de unas pocas veces la longitud de onda del lector) (Moya, 2021).

Etiquetas Pasivas

Las etiquetas pasivas no poseen alimentación eléctrica. La señal que les llega de los lectores induce una corriente eléctrica pequeña y suficiente para operar el circuito integrado CMOS de la etiqueta, de forma que puede generar y transmitir una respuesta. La mayoría de las etiquetas pasivas utiliza backscatter sobre la portadora recibida; esto es, la antena ha de estar diseñada para obtener la energía necesaria para funcionar a la vez que para transmitir la respuesta por backscatter. Esta respuesta puede ser cualquier tipo de información, no solo un código identificador. Una etiqueta puede incluir memoria no volátil, posiblemente escribible (por ejemplo, EEPROM) (Peris, 2008).

Las etiquetas pasivas suelen tener distancias de uso práctico comprendidas entre los 10 cm y llegando hasta unos pocos metros, según la frecuencia de funcionamiento y el diseño y tamaño de la antena. Por su sencillez conceptual, son obtenibles por medio de un proceso de impresión de las antenas. Como no precisan de

alimentación energética, el dispositivo puede resultar muy pequeño: pueden incluirse en una pegatina o insertarse bajo la piel (etiquetas de baja frecuencia) (Peris, 2008).

En 2006, Hitachi desarrolló un dispositivo pasivo denominado μ -Chip con un tamaño de 0.15×0.15 mm sin antena, más delgado que una hoja de papel ($7,5 \mu\text{m}$).⁷⁸ Se utiliza SOI (Silicon-on-Insulator) para lograr esta integración. Este chip puede transmitir un identificador único de 128 bits fijado a él en su fabricación, que no puede modificarse y confiere autenticidad al mismo. Tiene un rango máximo de lectura de 30 cm. En febrero de 2007 Hitachi presentó un dispositivo aún menor de 0.05×0.05 mm y lo suficientemente delgado como para poder estar integrado en una hoja de papel.⁹ Estos chips tienen capacidad de almacenamiento y pueden funcionar en distancias de hasta unos pocos cientos de metros. Su principal inconveniente es que su antena debe ser como mínimo 80 veces más grande que el chip (CAPELO BADILLO ALDO GRIVALDY, 2009).

Existen etiquetas fabricadas con semiconductores basados en polímeros desarrollados por compañías de todo el mundo. En 2005 PolyIC y Philips presentaron etiquetas sencillas en el rango de 13.56 MHz que utilizaban esta tecnología. Si se introducen en el mercado con éxito, estas etiquetas serían producibles en imprenta como una revista, con costes de producción mucho menores que las de silicio, sirviendo como alternativa totalmente impresa, como los actuales códigos de barras. Sin embargo, para ello es necesario que superen aspectos técnicos y económicos, teniendo en cuenta que el silicio es una tecnología que lleva décadas disfrutando de inversiones de desarrollo multimillonarias que han resultado en un coste menor que el de la impresión convencional (CAPELO BADILLO ALDO GRIVALDY, 2009).

Debido a las preocupaciones por la energía y el coste, la respuesta de una etiqueta pasiva RFID es necesariamente breve, normalmente apenas un número de identificación (GUID). La falta de una fuente de alimentación propia hace que el dispositivo pueda ser bastante pequeño: existen productos disponibles de forma comercial que pueden ser insertados bajo la piel. En la práctica, las etiquetas pasivas tienen distancias de lectura que varían entre unos 10 milímetros hasta cerca de 6 metros, dependiendo del tamaño de la antena de la etiqueta y de la potencia y frecuencia en la que opera el lector. En 2007, el dispositivo disponible comercialmente más pequeño de este tipo medía 0.05 milímetros × 0.05 milímetros, y más fino que una hoja de papel; estos dispositivos son prácticamente invisibles (CAPELO BADILLO ALDO GRIVALDY, 2009).

Etiquetas Activas

A diferencia de las etiquetas pasivas, las activas poseen su propia fuente autónoma de energía, que utilizan para dar corriente a sus circuitos integrados y propagar su señal al lector. Estas son mucho más fiables (tienen menos errores) que las pasivas debido a su capacidad de establecer sesiones con el lector. Gracias a su fuente de energía son capaces de transmitir señales más potentes que las de las pasivas, lo que los lleva a ser más eficientes en entornos dificultosos para la radiofrecuencia como el agua (incluyendo humanos y ganado, formados en su mayoría por agua), metal (contenedores, vehículos). También son efectivas a distancias mayores pudiendo generar respuestas claras a partir de recepciones débiles (al contrario que las pasivas). Por el contrario, suelen ser mayores y más caras, y su vida útil es en general mucho más corta (Moya, 2021).

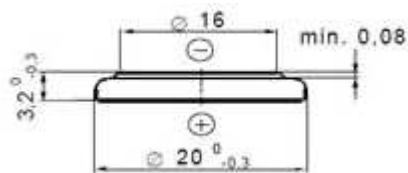
Muchas etiquetas activas tienen rangos efectivos de cientos de metros y una vida útil de sus baterías de hasta 10 años. Algunas de ellas integran sensores de registro de temperatura y otras variables que pueden usarse para monitorizar entornos de alimentación o productos farmacéuticos. Otros sensores asociados con RFID incluyen humedad, vibración, luz, radiación, temperatura y componentes atmosféricos como el etileno. Además de mucho más rango (500 m), tienen capacidades de almacenamiento mayores y la habilidad de guardar información adicional enviada por el transceptor (Moya, 2021).

Actualmente, las etiquetas activas más pequeñas tienen un tamaño aproximado de una moneda. Muchas etiquetas activas tienen rangos prácticos de diez metros, y una duración de batería de hasta varios años (Moya, 2021).

La principal ventaja de las etiquetas RFID activas respecto a las pasivas es el elevado rango de lectura, del orden de decenas de metros. Como desventajas, cabe destacar el precio, que es muy superior que las pasivas y la dependencia de alimentación por baterías. El tiempo de vida de las baterías depende de cada modelo de etiqueta y también de la actividad de este, normalmente es del orden de años. Para facilitar la gestión de las baterías, es habitual que las etiquetas RFID activas envíen al lector información del nivel de batería, lo que permite sustituir con antelación aquellas que están a punto de agotarse. Baterías de larga duración utilizadas en etiquetas RFID activas (CAPELO BADILLO ALDO GRIVALDY, 2009).

Figura 5

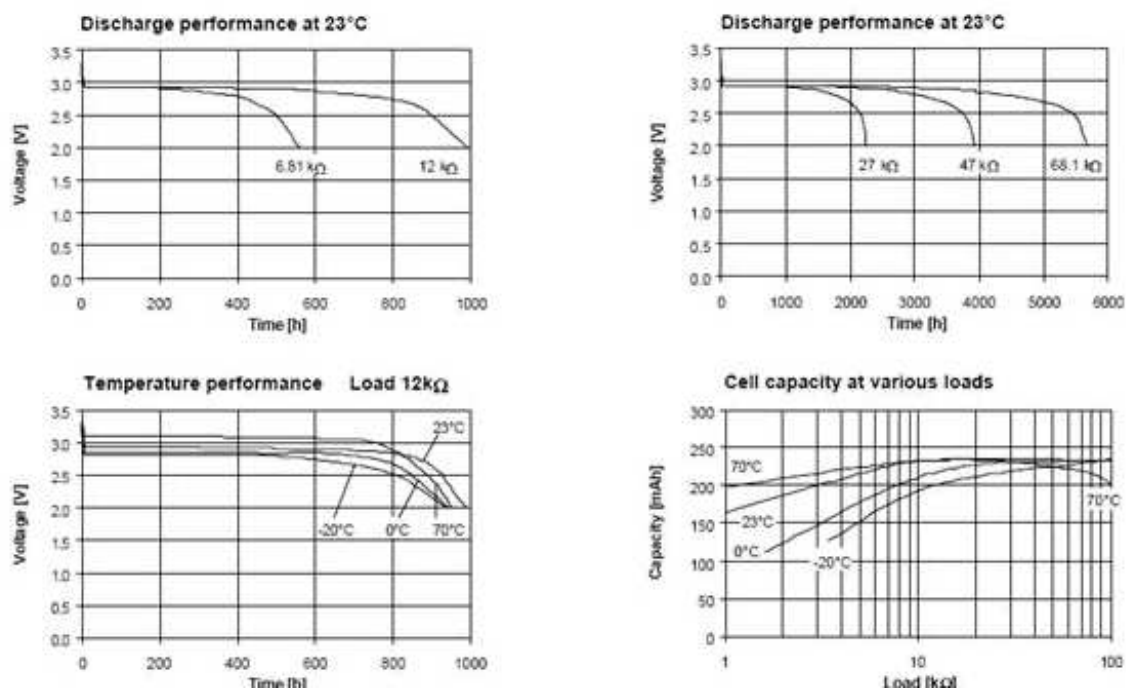
Baterías



Nota. Baterías de larga duración utilizadas en etiquetas RFID activas, tienen las siguientes dimensiones de acuerdo con el IEC 60086 (Moya, 2021).

Figura 6

Gráficos



Nota. Se muestra el desempeño de la descarga en función a la temperatura y a la resistencia de carga. (Moya, 2021).

También hay baterías impresas ultra-finas para el diseño de empaquetado activo. Estas baterías son flexibles, de gran alcance y tienen menos de un milímetro de

grosor, lo que las hacen ideales para las etiquetas activas de los sistemas RFID. Otra alternativa son las baterías de papel, que tienen aplicaciones en dispositivos RFID, smart cards y LED en papel, entre otros. Se trata de una batería que está formada por laminas finas de compuestos químicos incrustados en papel obteniéndose energía eléctrica a partir de reacciones de oxidación-reducción, produciendo en los bornes un voltaje nominal de 1.5 V y una carga de 1.5 mAh aproximadamente (Moya, 2021).

Etiquetas semipasivas

Las etiquetas semipasivas se parecen a las activas en que poseen una fuente de alimentación propia, aunque en este caso se utiliza principalmente para alimentar el microchip y no para transmitir una señal. La energía contenida en la radiofrecuencia se refleja hacia el lector como en una etiqueta pasiva. Un uso alternativo para la batería es almacenar información propagada desde el lector para emitir una respuesta en el futuro, típicamente usando backscatter. Las etiquetas sin batería deben responder reflejando energía de la portadora del lector al vuelo (CAPELO BADILLO ALDO GRIVALDY, 2009).

La batería puede permitir al circuito integrado de la etiqueta estar constantemente alimentado y eliminar la necesidad de diseñar una antena para recoger potencia de una señal entrante. Por ello, las antenas pueden ser optimizadas para utilizar métodos de backscattering. Las etiquetas RFID semipasivas responden más rápidamente, por lo que son más fuertes en la ratio de lectura que las pasivas (CAPELO BADILLO ALDO GRIVALDY, 2009).

Este tipo de etiqueta tiene una fiabilidad comparable a la de las activas, a la vez que pueden mantener el rango operativo de una pasiva. También suelen durar más tiempo que las activas (CAPELO BADILLO ALDO GRIVALDY, 2009).

Tipos de antena

El tipo de antena utilizado en una etiqueta depende de la aplicación para la que está diseñado y de la frecuencia de operación. Las etiquetas de baja frecuencia o LF (del inglés low frequency) normalmente se sirven de la inducción electromagnética. Como el voltaje inducido es proporcional a la frecuencia, se puede producir el necesario para alimentar un circuito integrado utilizando un número suficiente de espiras. Existen etiquetas LF compactas (como las encapsuladas en vidrio, utilizadas para identificación humana y animal) que utilizan una antena en varios niveles (tres de 100-150 espiras cada uno) alrededor de un núcleo de ferrita (CAPELO BADILLO ALDO GRIVALDY, 2009).

En alta frecuencia (HF, 13.56 MHz) se utiliza una espiral plana con 5 a 7 vueltas y un factor de forma parecido al de una tarjeta de crédito para lograr distancias de decenas de centímetros. Estas antenas son más baratas que las LF ya que pueden producirse por medio de litografía en lugar de espiración, aunque son necesarias dos superficies de metal y una aislante para realizar la conexión cruzada del nivel exterior al interior de la espiral, donde se encuentran el condensador de resonancia y el circuito integrado (CAPELO BADILLO ALDO GRIVALDY, 2009).

Las etiquetas pasivas en frecuencias ultraalta (UHF) y de microondas suelen acoplarse por radio a la antena del lector y utilizar antenas clásicas de dipolo. Solo es necesaria una capa de metal, lo que reduce el coste. Las antenas de dipolo, no obstante, no se ajustan muy bien a las características de los circuitos integrados típicos (con alta

impedancia de entrada, ligeramente capacitiva). Se pueden utilizar dipolos plegados o bucles cortos como estructuras inductivas complementarias para mejorar la alimentación. Los dipolos de media onda (16 cm a 900 MHz) son demasiado grandes para la mayoría de las aplicaciones (por ejemplo, las etiquetas RFID para uso en etiquetas no pueden medir más de 10 cm), por lo que hay que doblar las antenas para satisfacer las necesidades de tamaño. También pueden usarse estructuras de banda ancha. La ganancia de las antenas compactas suele ser menor que la de un dipolo (menos de 2 dB) y pueden considerarse isótropas en el plano perpendicular a su eje (Moya, 2021).

Pueden usarse antenas de parche para dar servicio en las cercanías de superficies metálicas, aunque es necesario un grosor de 3 a 6 mm para lograr un buen ancho de banda, además de que es necesario tener una conexión a tierra que incrementa el coste comparado con estructuras de una capa más sencillas. Las antenas HF y UHF suelen ser de cobre o aluminio. Se han probado tintas conductoras en algunas antenas encontrando problemas con la adhesión al circuito integrado y la estabilidad del entorno (Moya, 2021).

Asociación de etiquetas

Existen tres tipos básicos de etiquetas por su relación con los objetos que identifican: asociable, implantable e insertable (attachable, implantable, insertion). Además de estos tipos de etiquetas, Eastman Kodak ha presentado dos solicitudes de patente que tratan de la monitorización del consumo de medicina en forma de una etiqueta “digerible” (Moya, 2021).

Entornos de etiquetas

Para leer los datos de las etiquetas, los lectores utilizan un algoritmo basado en el recorrido de árboles, resolviendo las colisiones que puedan darse y procesando secuencialmente las respuestas. Existen etiquetas bloqueantes que pueden usarse para evitar que haya lectores que accedan a las etiquetas de un área sin necesidad de recurrir a comandos de suicidio para inhabilitarlas. Estas se hacen pasar por etiquetas normales, pero poseen ciertas características específicas; en concreto, pueden tomar cualquier código de identificación como propio, y pueden responder a toda pregunta que escuchen, asegurando el entorno al anular la utilidad de estas preguntas (CAPELO BADILLO ALDO GRIVALDY, 2009).

En general, puede emitirse una señal espuria si se detecta actividad de etiquetas para bloquear las transmisiones débiles producidas por estas. En caso de que las etiquetas sean prescindibles o no sean necesarias de nuevo, pueden inutilizarse induciendo en ellos corrientes elevadas que inutilicen sus circuitos (CAPELO BADILLO ALDO GRIVALDY, 2009).

Aparte de esto, una etiqueta puede ser promiscua, si responde a todas las peticiones sin excepción, o segura, si requiere autenticación (esto conlleva los aspectos típicos de gestión de claves criptográficas y de acceso). Una etiqueta puede estar preparada para activarse o desactivarse como respuesta a comandos del lector (CAPELO BADILLO ALDO GRIVALDY, 2009).

Los lectores encargados de un grupo de etiquetas en un área pueden operar en modo autónomo, en contraposición al modo interactivo. Si trabajan de esta forma, realizan una identificación periódica de todas las etiquetas en su entorno y mantienen

una lista de presencia con tiempos de persistencia (timeouts) e información de control.



Si una entrada expira, se elimina de la tabla (CAPELO BADILLO ALDO GRIVALDY, 2009).

Con frecuencia una aplicación distribuida requiere el uso de ambos tipos extremos de etiquetas. Las pasivas no pueden realizar labores de monitorización continua, sino que realizan tareas bajo demanda cuando los lectores se las solicitan. Son útiles para realizar actividades regulares y bien definidas con necesidades de almacenamiento y seguridad acotadas. Si hay accesos frecuentes, continuos o impredecibles, o bien existen requerimientos de tiempo real o procesamiento de datos (como búsqueda en tablas internas) suele ser conveniente utilizar etiquetas activas (CAPELO BADILLO ALDO GRIVALDY, 2009).

A fecha de hoy, la mayoría de las empresas que venden etiquetas RFID no indican los precios porque el precio se basa en el volumen, la memoria, el embalaje de la propia etiqueta, y en si la etiqueta es activa, pasiva o muchas otras variables (CAPELO BADILLO ALDO GRIVALDY, 2009).

Tabla 3

Dispositivos

N.º	Descripción	Costo	Imagen
1	Etiquetas pasivas UHF	\$0,08 - \$0,15	
2	Etiquetas pasivas imprimibles	\$0,2 aprox	

N.º	Descripción	Costo	Imagen
3	Etiquetas activas	\$30,00 aprox.	
4	Etiquetas activas AP	\$60,00 aprox.	
5	Cables RFID	\$3,00 el metro	
6	Antenas	\$200,00	
7	Lectores RFID UHF	\$500 - \$200	
8	Lectores de baja frecuencia	\$100 aprox.	

Nota. Tabla de precios aproximados

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

Especificaciones de requerimientos de software

Introducción

Es una especificación de requerimientos de software del sistema de control de inventarios utilizando la tecnología RFID para la gestión de los bienes pertenecientes a la Universidad de las Fuerzas Armadas – ESPE, el mismo que ayudará en el proceso de la constatación física.

Propósito

Definir los requisitos, requerimientos funcionales y requerimientos no funcionales que debe cumplir el sistema de software con tecnología RFID para el control de inventario de los bienes de la Universidad de la Fuerzas Armadas – ESPE.

Alcance

El presente proyecto va a ser utilizado por la Universidad de las Fuerzas Armadas – ESPE, sede Sangolquí. En el presente documento consta todos los procesos que se van a llevar a cabo para el desarrollo del sistema. Lo que se esperar lograr por medio del sistema son las siguientes actividades:

- Lectura del inventario
- Comparativa de bienes
- Generar actas
- Migración de datos entre sistemas
- Agilizar la toma física

Historias de usuarios

Concepto

Las historias de usuario son descripciones cortas y simples de una característica contada desde la perspectiva de la persona que desea la nueva capacidad, generalmente un usuario o cliente del sistema. Por lo general, siguen una plantilla simple:

- Como <Usuario>
- Quiero <algún objetivo>
- Para que <motivo>

Las historias de los usuarios a menudo se escriben en fichas o notas adhesivas, se almacenan en una caja y se organizan en paredes o mesas para facilitar la planificación y el debate. Como tal, cambian fuertemente el enfoque de escribir sobre las características a discutir. De hecho, estas discusiones son más importantes que cualquier texto que se escriba.

Historias de usuario

En las siguientes tablas se detallarán cada una de las historias de usuario obtenidas durante el levantamiento de requerimientos para realizar el desarrollo del sistema de control de inventario utilizando tecnología RFID para la Universidad de las Fuerzas Armadas – ESPE, sede Sangolquí.

Historia de usuario 1

Tabla 4

Historia 1

Historias de usuario	
Número: 1	Usuario: Personal de bienes
Nombre historia: Autenticar usuario	
Prioridad en negocio: .	Riesgo en desarrollo: .
Puntos estimados:	Iteración asignada:
Programador responsable: Stephen Sampedro	
Descripción: El personal de bienes serán los únicos que tendrán la opción de ingresar al sistema con el usuario y contraseña que utilicen para ingresar a miespe.	
Observaciones: Deberá estar registrado el personal previamente en miespe	

Nota. Se ingresa con el nombre de usuario de miespe

Historia de usuario 2

Tabla 5

Historia 2

Historias de usuario	
Número: 2	Usuario: Personal de bienes
Nombre historia: Obtener bienes registrados	
Prioridad en negocio: .	Riesgo en desarrollo: .
Puntos estimados:	Iteración asignada:
Programador responsable: Stephen Sampedro	
Descripción: El personal de bienes podrá obtener un listado de todos los bienes registrados anteriormente.	
Observaciones: Los bienes se podrá filtrar por custodio, área o departamento	

Nota. Personal de bienes

Historia de usuario 3

Tabla 6

Historia 3

Historias de usuario	
Número: 3	Usuario: Personal de bienes
Nombre historia: Leer bienes	
Prioridad en negocio:	Riesgo en desarrollo:
.	.
Puntos estimados:	Iteración asignada:
Programador responsable: Stephen Sampedro	
<p>Descripción:</p> <p>El personal de bienes podrá leer los activos de un puesto de trabajo, utilizando la tecnología de RFID.</p>	
<p>Observaciones:</p> <p>Por medio del lector y los tags de RFID se podrá identificar cada bien</p>	

Historia de usuario 4

Tabla 7

Historia 4

Historias de usuario	
Número: 4	Usuario: Personal de bienes
Nombre historia: Etiquetar	
Prioridad en negocio: .	Riesgo en desarrollo: .
Puntos estimados:	Iteración asignada:
Programador responsable: Stephen Sampedro	
Descripción: Se debe etiquetar los bienes utilizando los tags de RFID.	
Observaciones: Los activos que se van a etiquetar son los pertenecientes al departamento de bienes.	

Historia de usuario 5

Tabla 8

Historia 5

Historias de usuario	
Número: 5	Usuario: Personal de bienes
Nombre historia: Verificar bienes	
Prioridad en negocio: .	Riesgo en desarrollo: .
Puntos estimados:	Iteración asignada:
Programador responsable: Stephen Sampedro	
Descripción: Se valida que el activo esté bien vinculado con el código del tag RFID.	
Observaciones:	

Historia de usuario 6

Tabla 9

Historia 6

Historias de usuario	
Número: 6	Usuario: Personal de bienes
Nombre historia: Comparar bienes	
Prioridad en negocio: .	Riesgo en desarrollo: .
Puntos estimados:	Iteración asignada:
Programador responsable: Stephen Sampedro	
Descripción: Se realiza la comparación entre los bienes de la base de datos versus los bienes leídos, este proceso se hace por custodio.	
Observaciones:	

Historia de usuario 7

Tabla 10

Historia 7

Historias de usuario	
Número: 7	Usuario: Personal de bienes
Nombre historia: Generar acta	
Prioridad en negocio: .	Riesgo en desarrollo: .
Puntos estimados:	Iteración asignada:
Programador responsable: Stephen Sampedro	
Descripción: Se genera un PDF con los bienes sobrantes, faltantes o con ninguna novedad.	
Observaciones:	

Historia de usuario 8

Tabla 11

Historia 8

Historias de usuario	
Número: 8	Usuario: Personal de bienes
Nombre historia: Migrar datos	
Prioridad en negocio: .	Riesgo en desarrollo: .
Puntos estimados:	Iteración asignada:
Programador responsable: Stephen Sampedro	
Descripción: El personal de bienes tendrá la opción de pasar los datos del sistema eSByE al sistema OLYMPO.	
Observaciones:	

Diagramas entidad relación

Definición

El modelo de entidad relación nos permite conocer los diferentes componentes que interviene en los procesos del sistema a desarrollar.

Entidades nuevas

Para la correcta funcionalidad del sistema a desarrollar, se van a crear nuevas entidades dentro de la base de datos que actualmente se utiliza por los sistemas de inventario para la toma física. Las nuevas entidades son las que se muestran en las siguientes figuras:

Figura 7

Funcionarios

OLYPRUEBA.AUDITORIAAPP	
P *	CODIGO NUMBER (10)
	USUARIO VARCHAR2 (255 BYTE)
	OPCION VARCHAR2 (300 BYTE)
	CODIGOBIEN VARCHAR2 (100 BYTE)
	DESCRIPCION VARCHAR2 (300 BYTE)
	FECHA_AUDIT DATE
	AUDITORIAAPP_PK (CODIGO)
	AUDITORIAAPP_PK (CODIGO)

Figura 8

Activos RFID

OLYPRUEBA.ACTIVOSRFID	
P *	CODIGO NUMBER (10)
	CODIGO_RFID VARCHAR2 (100 BYTE)
	CUSTODIO1 VARCHAR2 (300 BYTE)
	CUSTODIO2 VARCHAR2 (300 BYTE)
	DESCRIPCION VARCHAR2 (300 BYTE)
	USUARIO VARCHAR2 (255 BYTE)
	ESTADJUSTIFICADO VARCHAR2 (15 BYTE)
	ESTADOTRANSACCION VARCHAR2 (15 BYTE)
	FECHA_LECTURA DATE
	ACTIVOSRFID_PK (CODIGO)
	ACTIVOSRFID_PK (CODIGO)

Figura 9

Archivos

OLYPRUEBA.ARCHIVOSJUSTIF	
P *	CODIGO NUMBER (10)
	CODIGOBIEN VARCHAR2 (100 BYTE)
	CUSTODIO1 VARCHAR2 (300 BYTE)
	CUSTODIO2 VARCHAR2 (300 BYTE)
	DESCRIPCION VARCHAR2 (300 BYTE)
	DOCUMENTO CLOB
	NOMBREARCHIVO VARCHAR2 (300 BYTE)
	TIPOARCHIVO VARCHAR2 (300 BYTE)
	FECHA_INGRESO DATE
	ARCHIVOSJUSTIF_PK (CODIGO)
	ARCHIVOSJUSTIF_PK (CODIGO)

Entidades

También se utilizan las siguientes entidades que forman parte de la base de datos de los sistemas que la ESPE utiliza para la toma física. Las cuales se detallan en las siguientes figuras:

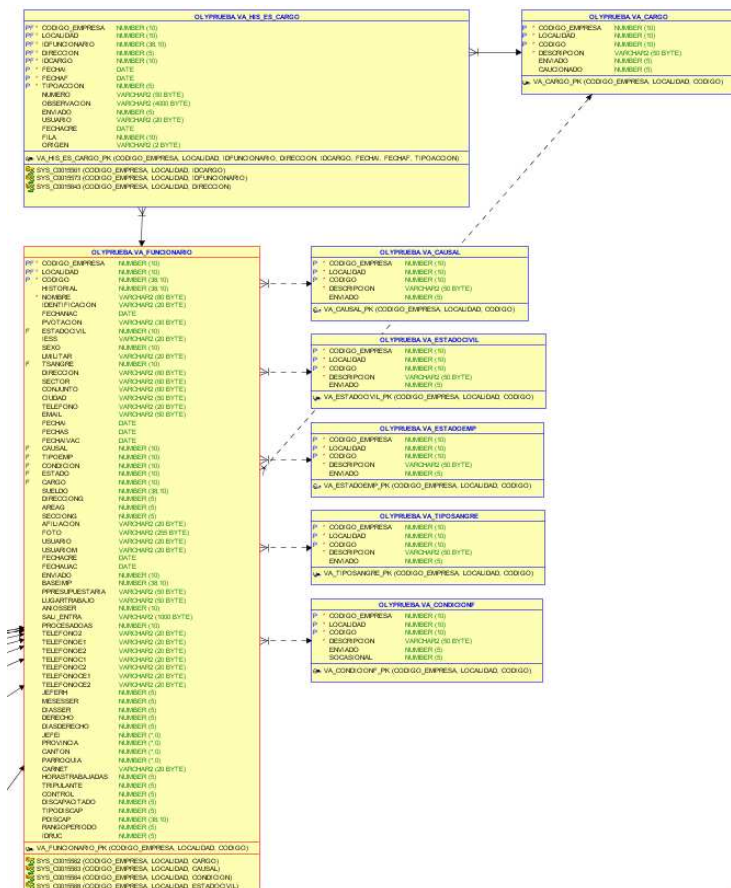
Figura 10

Provee

OLYPRUEBA.PPPOVEE	
PF *	CODIGO_EMPRESA NUMBER (10)
P *	CODIGO_PROVEEDOR VARCHAR2 (15 BYTE)
	NOMBRE VARCHAR2 (150 BYTE)
	NOMBRECMR VARCHAR2 (150 BYTE)
	CODIGO_BANCO NUMBER (5)
	IDENTIFICACION VARCHAR2 (20 BYTE)
	PRODUCTOR NUMBER (5)
	DIRECCION VARCHAR2 (80 BYTE)
	TELEFONO VARCHAR2 (30 BYTE)
	FAX VARCHAR2 (30 BYTE)
	CIUDAD VARCHAR2 (30 BYTE)
	REPRESENTANTE VARCHAR2 (80 BYTE)
	CUENTA_AHORROS VARCHAR2 (20 BYTE)
	CUENTA_CORRIENTE VARCHAR2 (50 BYTE)
	AGENCIA VARCHAR2 (10 BYTE)
	CUENTA_NUM VARCHAR2 (25 BYTE)
F	TIPO_MONEDA NUMBER (5)
	TIPO_COTIZACION VARCHAR2 (10 BYTE)
	ESTADO_CREDITO NUMBER (5)
	PERIODO VARCHAR2 (20 BYTE)
	TRANSPORTE NUMBER (5)
	TRANSPORTEIV NUMBER (5)
	LIMITE_CR_SUCRES NUMBER (38,10)
	LIMITE_CR_ME NUMBER (38,10)
	MONTO NUMBER (38,10)
	CONTRATO VARCHAR2 (20 BYTE)
	FECHANAC DATE
	SEXO NUMBER (5)
	ACUMULADO NUMBER (38,10)
	PROCEDENCIA VARCHAR2 (30 BYTE)
	FECHAOBJ DATE
	CONTRIBUYENTE NUMBER (5)
	CUENTA_CONTABLE VARCHAR2 (50 BYTE)
	CUENTA_CONTABLEON VARCHAR2 (50 BYTE)
	CUENTA_CONTABLEA VARCHAR2 (50 BYTE)
	CUENTA_CONTABLEAON VARCHAR2 (50 BYTE)
	COPRATICO VARCHAR2 (50 BYTE)
	NIVEL NUMBER (5)
	GALFICADO NUMBER (5)
	VALORESTRANJERO NUMBER (38,10)
	IMPACTIVOS NUMBER (10)
	NLGENCIA VARCHAR2 (50 BYTE)
	TIPOCHV VARCHAR2 (4 BYTE)
	CADUCACHV DATE
	VEHICULO VARCHAR2 (50 BYTE)
	ACTIVOCHV NUMBER (10)
	TIPOEMP NUMBER (5)
	CARGO NUMBER (5)
	CONDUCTOR NUMBER (10)
	MONTOCRD NUMBER (38,10)
	PLAZOCRD NUMBER (5)
	DSCTOCDR NUMBER (38,10)
	OBSERVACION VARCHAR2 (255 BYTE)
	ENVIADO NUMBER (5)
	USUARIO VARCHAR2 (20 BYTE)

Figura 11

Funcionarios



Nota. Se muestra el modelo entidad relación de la tabla de funcionarios.

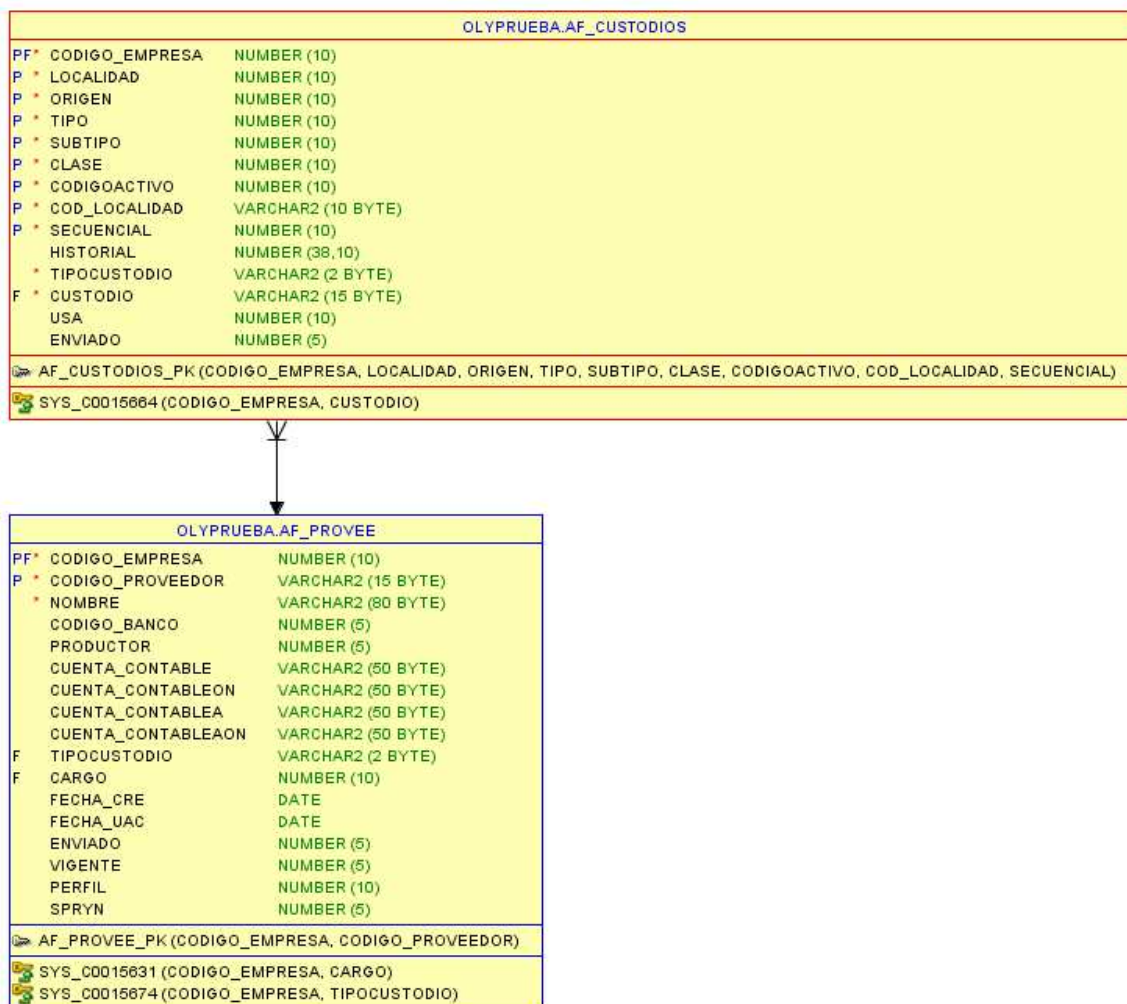
Figura 12

Activos

P *	CODIGO_EMPRESA	NUMBER (10)
P *	LOCALIDAD	NUMBER (10)
P *	ORIGEN	NUMBER (10)
P *	TIPO	NUMBER (10)
P *	SUBTIPO	NUMBER (10)
P *	CLASE	NUMBER (10)
P *	CODIGO	NUMBER (10)
P *	COD_LOCALIDAD	VARCHAR2 (10 BYTE)
	HISTORIAL	NUMBER (38,10)
	TIPOTRANSACCION	VARCHAR2 (4 BYTE)
	CONCEPTO_TRANSACCION	NUMBER (10)
	NUMERO_TRANSACCION	NUMBER (38,10)
	FECHA_TRANSACCION	DATE
	LOTE	NUMBER (38,10)
	SUBLOTE	NUMBER (38,10)
	PRINCIPAL	NUMBER (38,10)
	SINUSO	NUMBER (10)
	ACTIVIDAD	NUMBER (10)
	DESCRIPCION	VARCHAR2 (1000 BYTE)
	OBSERVACION	VARCHAR2 (1000 BYTE)
	CANTIDADORIGINAL	NUMBER (10)
	CANTIDAD	NUMBER (10)
	AVALUOTECNICO	NUMBER (10)
	VALOR	NUMBER (38,10)
	VALORME	NUMBER (38,10)
	VALORAVALUO	NUMBER (38,10)
	VALORAVALLIOME	NUMBER (38,10)
	DEPRECH	NUMBER (38,10)
	DEPRECHME	NUMBER (38,10)
	VIDAUTIL	NUMBER (10)
	PORCDEPRE	NUMBER (38,10)
	PORCDEPREMENSUAL	NUMBER (38,10)
	FECHAADQUISICION	DATE
	FECHAAPLICACION	DATE
	ESTADOCONS	NUMBER (10)
	ESTRUCTURABIEN	NUMBER (10)
	COLOR	NUMBER (10)
	DIMENSIONES	VARCHAR2 (150 BYTE)
	MARCA	VARCHAR2 (150 BYTE)
	MODELO	VARCHAR2 (150 BYTE)
	NSERIE	VARCHAR2 (150 BYTE)
	PLACA	VARCHAR2 (150 BYTE)
	MOTOR	VARCHAR2 (150 BYTE)
	CHASIS	VARCHAR2 (150 BYTE)
	CLASEVEHICULO	VARCHAR2 (150 BYTE)
	TIPOVEHICULO	VARCHAR2 (150 BYTE)
	ANIOFABRICA	NUMBER (10)
	OBRA	VARCHAR2 (150 BYTE)
	ALITOR	VARCHAR2 (150 BYTE)
	TIPOOBRA	VARCHAR2 (150 BYTE)
	NOMBRE	VARCHAR2 (150 BYTE)
	ARETE	VARCHAR2 (150 BYTE)
	FECHA_NACIMIENTO	DATE
	RAZA	VARCHAR2 (50 BYTE)
	SEXO	VARCHAR2 (1 BYTE)
	AREACONS	VARCHAR2 (150 BYTE)
	VALORCOMERCIAL	NUMBER (38,10)
	INFRAESTRUCTURA	VARCHAR2 (150 BYTE)
	REGISTROPROPIEDAD	DATE
	LINDEROS	VARCHAR2 (150 BYTE)
	ORIGENCOMPRA	NUMBER (10)
	ORIGENFONDO	NUMBER (10)
	TIPOCOMODATO	NUMBER (10)
	NPROPIETARIO	VARCHAR2 (15 BYTE)
	CUSTODIO1	VARCHAR2 (15 BYTE)
	CUSTODIO2	VARCHAR2 (15 BYTE)
	UNIOPERATIVA	NUMBER (10)
	SUCURSAL	NUMBER (10)
	PROVINCIA	NUMBER (5)
	CANTON	NUMBER (5)
	PARROQUIA	NUMBER (5)
	DIRECCION	NUMBER (5)
	AREA	NUMBER (5)
	SECCION	NUMBER (5)
	SUBSECCION	NUMBER (5)
	FECHAGARANTIA	DATE
	DEPREANT	NUMBER (38,10)
	DEPREANTME	NUMBER (38,10)
	DEPREACTUAL	NUMBER (38,10)
	DEPREACTUALME	NUMBER (38,10)

Nota. Se muestra el modelo entidad relación de la tabla de funcionarios.

Figura 13

Custodios

Nota. Se muestra el modelo entidad relación de la tabla de custodios.

Diagramas de clases***Definición***

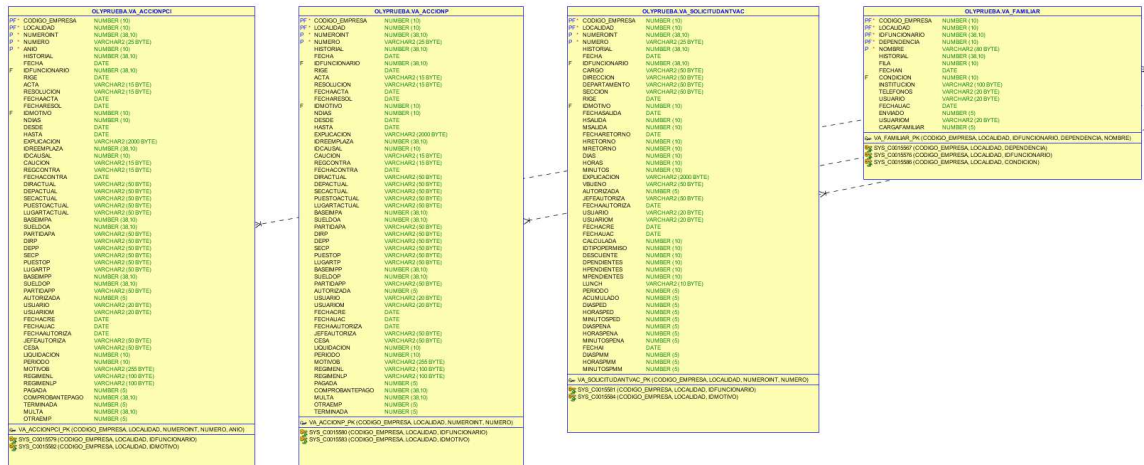
Los diagramas de clases nos permiten dar a conocer el funcionamiento de un programa en bases a las relaciones entre sus diferentes entidades.

Clases

Las clases que se van a diagraman son de las entidades que se utilizan actualmente en los sistemas para el proceso de constatación física de la universidad.

Figura 14

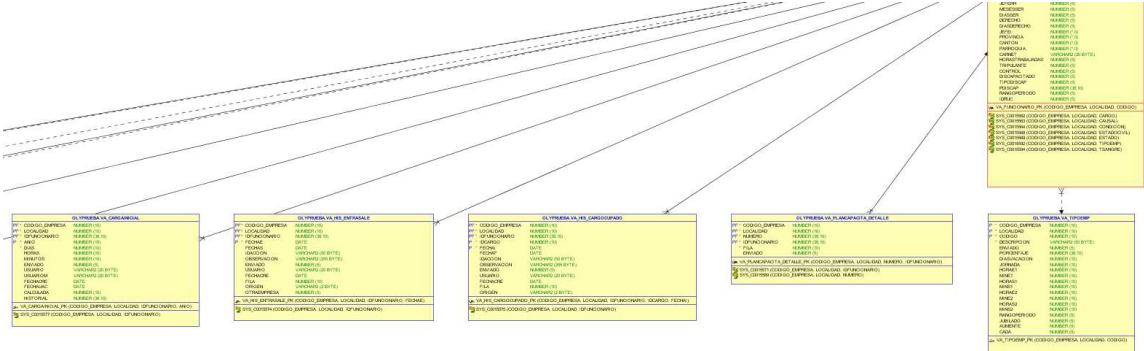
Funcionarios



Nota. Se muestra el modelo entidad relación de la tabla de funcionarios.

Figura 15

Funcionarios



Nota. Continuación de la tabla de funcionarios.

CAPÍTULO IV

DESARROLLO Y CODIFICACIÓN

Introducción

El presente proyecto de investigación se basa en la necesidad de agilizar el procedimiento que actualmente realiza la Unidad de Logística al momento de realizar la toma física de los bienes. Es por esto por lo que el sistema web y la aplicación móvil debe contar con una interfaz amigable, y que cumpla con los requerimientos solicitados.

Para llevar a cabo el desarrollo del sistema web y la app se utilizó varias tecnologías las cuales están en auge en los últimos años, las mismas que se describirán a continuación:

IONIC Framework: Ionic Framework es un conjunto de herramientas de interfaz de usuario de código abierto para crear aplicaciones móviles y de escritorio de alta calidad y rendimiento utilizando tecnologías web (HTML, CSS y JavaScript) con integraciones para marcos populares como Angular, React y Vue (community, 2021).

Angular: Angular es un framework opensource desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (Single Page Application) (Community, 2021).

Spring Boot: Spring Boot es una infraestructura ligera que elimina la mayor parte del trabajo de configurar las aplicaciones basadas en Spring (Perry, 2017).

Desarrollo y configuración del servidor (BACKEND)

Creación de proyecto

Se realiza la creación de un proyecto Maven java, en el cual se va a tener toda la lógica de la aplicación, así como también contará con la conexión a la base de datos que actualmente la ESPE está manejando para el control de bienes de la universidad.

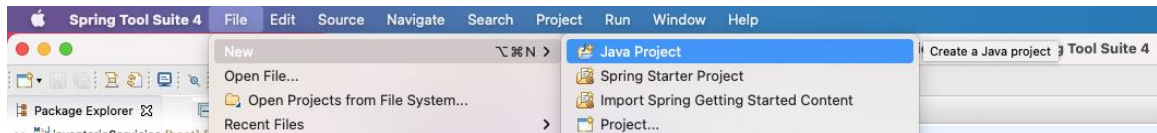
Procesos

Los procesos que se van a desarrollar en el backend son:

- Consulta de activos
- Consulta de funcionarios
- Gestión de activos RFID
- Ingreso de pistas de auditorías
- Consulta de áreas
- Consulta de subsección y sección
- Consulta de usuarios

Figura 16

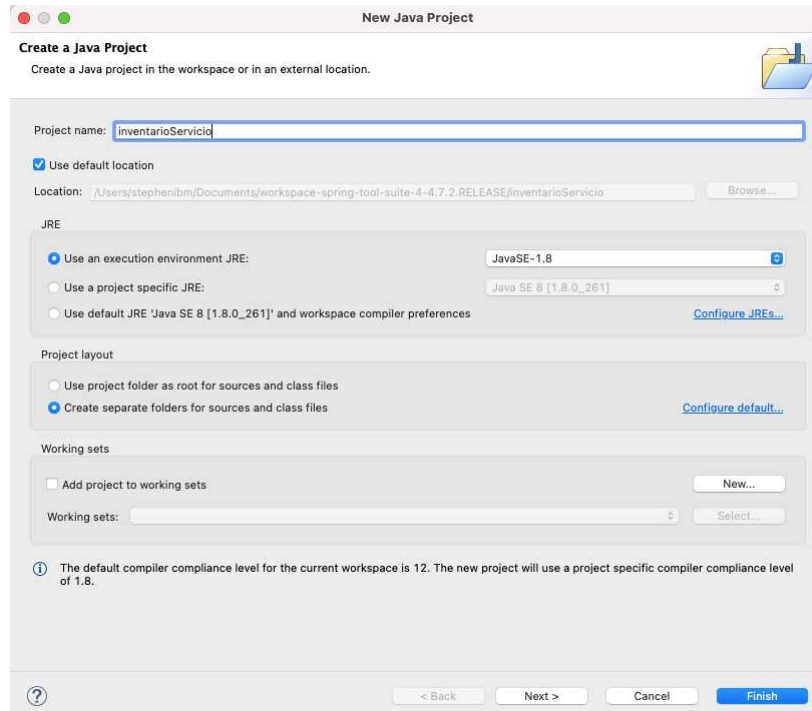
Crear



Nota. Se crea un proyecto java.

Figura 17

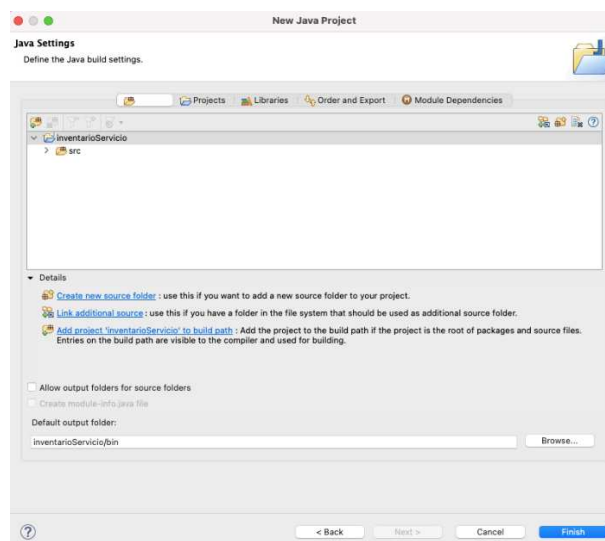
Crear



Nota. Se crea un proyecto java.

Figura 18

Crear



Nota. Se crea un proyecto java.

Figura 19

Crear

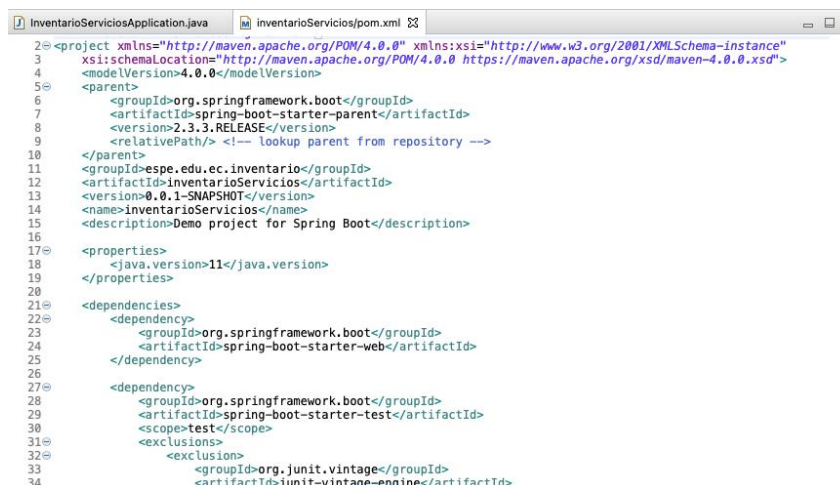


Nota. Se crea un proyecto java.

Pom.xml: En este fichero se configuran las diferentes dependencias que se utilizaran para el correcto funcionamiento del proyecto.

Figura 20

Fichero



Nota. pom.xml.

En este fichero se puede añadir la dependencia para seleccionar la base de datos a la que se va a conectar el proyecto.

Figura 21

Fichero



Nota. Source.

Application.properties: En este archivo se realiza la conexión a la base de datos donde se puede configura la url, usuario y contraseña de ORACLE.

Figura 22

Fichero

```

application.properties
6 # OracleDB connection settings
7 spring.datasource.url=jdbc:oracle:thin:@//10.1.0.188:1521/RELOJ
8 spring.datasource.username=ORACLE
9 spring.datasource.password=ORACLE
10 spring.datasource.driver-class-name=oracle.jdbc.OracleDriver
11 #spring.jpa.database-platform = org.hibernate.dialect.Oracle10gDialect
12
13 #spring.datasource.url = jdbc:mysql://192.168.4.2/maha?useSSL=false
14 #spring.datasource.username = test
15 #spring.datasource.password = test@123
16 # ### Hibernate Properties
17 ## The SQL dialect makes Hibernate generate better SQL for the chosen database
18 #spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect
19 ## Hibernate ddl auto (create, create-drop, validate, update)
20 # spring.mvc.dispatch-options-request=true
21
22 # HikariCP settings
23 spring.datasource.hikari.minimumIdle=5
24 spring.datasource.hikari.maximumPoolSize=20
25 spring.datasource.hikari.idleTimeout=30000
26 spring.datasource.hikari.maxLifetime=2000000
27 spring.datasource.hikari.connectionTimeout=30000
28 spring.datasource.hikari.poolName=HikariPoolBooks
29
30 # JPA settings
31 spring.jpa.database-platform=org.hibernate.dialect.Oracle12cDialect
32 #spring.jpa.hibernate.use-new-id-generator-mappings=false
33 #spring.jpa.hibernate.ddl-auto=create
34
35 # Subir archivos
36 spring.servlet.multipart.max-file-size=2MB
37 spring.servlet.multipart.max-request-size=2MB
38
39 server.tomcat.max-swallow-size=-1

```

Nota. Source.

Anotaciones

Spring boot maneja diferentes tipos de anotaciones para el correcto funcionamiento del proyecto dentro de las cuales tenemos: @Autowired, @Component, @Controller, @Repository, @Service, @RequestMapping, @Entity, @Table, entre otros.

En el presente proyecto se va a utilizar las anotaciones @Entity, @Table, @Id, @Column para las clases Modelo como se muestra en las siguientes figuras.

Figura 23

Fichero



Nota. Source.

Figura 24

Activo

```

Activo.java
1 package espe.edu.ec.inventario.modelo;
2
3 import java.io.Serializable;
4
5 @Entity
6 @Table ( name = "AF_ACTIVOFIJOB" )
7 public class Activo implements Serializable{
8
9     private static final long serialVersionUID = 1L;
10
11     @Id
12     @Column ( name = "CODIGO" )
13     private int codigo;
14     @Column ( name = "CODIGO_EMPRESA" )
15     private int codigoEmpresa;
16     @Column ( name = "LOCALIDAD" )
17     private int localidad;
18     @Column ( name = "ORIGEN" )
19     private int origen;
20     @Column ( name = "TIPO" )
21     private int tipo;
22     @Column ( name = "SUBTIPO" )
23     private int subtipo;
24     @Column ( name = "CLASE" )
25     private int clase;
26     @Column ( name = "COD_LOCALIDAD" )
27     private int codLocalidad;
28     @Column ( name = "CONCEPTO_TRANSACCION" )
29     private int conceptoTransaccion;
30     @Column ( name = "LOTE" )
31     private int lote;
32     @Column ( name = "SUBLOTE" )
33     private int subLote;
34     @Column ( name = "DESCRIPCION" )

```

Nota. Source.

Figura 25

ActivoRfid

```

ActivoRfid.java
1 package espe.edu.ec.inventario.modelo;
2
3 import java.io.Serializable;
4
5 @Entity
6 @Table ( name = "ACTIVOSRFID" )
7 public class ActivoRfid implements Serializable{
8
9     private static final long serialVersionUID = 1L;
10
11     @Id
12     @Column ( name = "CODIGO" )
13     @GeneratedValue(strategy=GenerationType.SEQUENCE)
14     private int codigo;
15     @Column ( name = "CODIGO_RFID" )
16     private String codigoRfid;
17     @Column ( name = "CUSTODIO1" )
18     private String custodio1;
19     @Column ( name = "CUSTODIO2" )
20     private String custodio2;
21     @Column ( name = "DESCRIPCION" )
22     private String descripcion;
23     @Column ( name = "USUARIO" )
24     private String usuario;
25     @Column ( name = "ESTADOJUSTIFICADO" )
26     private String estadoJustificado;
27     @Column ( name = "ESTADOTRANSACCION" )
28     private String estadoTransaccion;
29     @Column ( name = "FECHA_LECTURA" )
30     private Date fechaLectura;
31     public int getCodigo() {
32         return codigo;
33     }
34     public void setCodigo(int codigo) {

```

Nota. Source.

Figura 26

ArchivoJustif

```

1 package espe.edu.ec.inventario.modelo;
2
3 import java.io.Serializable;
4
5 @Entity
6 @Table ( name = "ARCHIVOSJUSTIF" )
7 public class ArchivoJustif implements Serializable{
8
9     private static final long serialVersionUID = 1L;
10
11     @Id
12     @Column ( name = "CODIGO" )
13     @GeneratedValue(strategy=GenerationType.SEQUENCE)
14     private int codigo;
15     @Column ( name = "CODIGOBLEN" )
16     private String codigoBlen;
17     @Column ( name = "CUSTODIO1" )
18     private String custodio1;
19     @Column ( name = "CUSTODIO2" )
20     private String custodio2;
21     @Column ( name = "DESCRIPCION" )
22     private String descripcion;
23     @Column ( name = "DOCUMENTO" )
24     private String documento;
25     @Column ( name = "NOMBREARCHIVO" )
26     private String nombreArchivo;
27     @Column ( name = "TIPOARCHIVO" )
28     private String tipoArchivo;
29     @Column ( name = "FECHA_INGRESO" )
30     private Date fechaIngreso;
31     public int getCodigo() {
32         return codigo;
33     }
34     public void setCodigo(int codigo) {

```

Nota. Source.

Figura 27

Área

```

1 package espe.edu.ec.inventario.modelo;
2
3 import java.io.Serializable;
4
5 @Entity
6 @Table ( name = "AF_AREA" )
7 public class Area implements Serializable{
8
9     /**
10      *
11      */
12     private static final long serialVersionUID = 1L;
13
14     @Id
15     @Column ( name = "CODIGO_AREA" )
16     private int codigoArea;
17     @Column ( name = "COD_DIRECCION" )
18     private int codigoDireccion;
19     @Column ( name = "DESC_AREA" )
20     private String descArea;
21
22     public int getCodigoArea() {
23         return codigoArea;
24     }
25     public void setCodigoArea(int codigoArea) {
26         this.codigoArea = codigoArea;
27     }
28     public int getCodigoDireccion() {
29         return codigoDireccion;
30     }
31     public void setCodigoDireccion(int codigoDireccion) {
32         this.codigoDireccion = codigoDireccion;
33     }
34     public String getDescArea() {

```

Nota. Source.

Figura 28

Auditoria

```

AuditoriaApp.java
1 package espe.edu.ec.inventario.modelo;
2
3 import java.io.Serializable;
4
5 @Entity
6 @Table ( name = "AUDITORIAAPP" )
7 public class AuditoriaApp implements Serializable{
8
9     private static final long serialVersionUID = 1L;
10
11     @Id
12     @Column ( name = "CODIGO" )
13     @GeneratedValue(strategy=GenerationType.SEQUENCE)
14     private int codigo;
15     @Column ( name = "USUARIO" )
16     private String usuario;
17     @Column ( name = "OPCION" )
18     private String opcion;
19     @Column ( name = "CODIGOBLEN" )
20     private String codigoBien;
21     @Column ( name = "DESCRIPCION" )
22     private String descripcion;
23     @Column ( name = "FECHA_AUDIT" )
24     private Date fechaAudit;
25     public int getCodigo() {
26         return codigo;
27     }
28     public void setCodigo(int codigo) {
29         this.codigo = codigo;
30     }
31     public String getUsuario() {
32         return usuario;
33     }
34     public void setUsuario(String usuario) {

```

Nota. Source.

Figura 29

Dirección

```

Direccion.java
1 package espe.edu.ec.inventario.modelo;
2
3 import java.io.Serializable;
4
5 @Entity
6 @Table ( name = "AF_DIRECCION" )
7 public class Direccion implements Serializable{
8
9     /**
10      *
11      */
12     private static final long serialVersionUID = 1L;
13
14     @Id
15     @Column ( name = "CODIGO_DIRECCION" )
16     private int codigoDireccion;
17     @Column ( name = "DESC_DIRECCION" )
18     private String descDireccion;
19
20     public int getCodigoDireccion() {
21         return codigoDireccion;
22     }
23     public void setCodigoDireccion(int codigoDireccion) {
24         this.codigoDireccion = codigoDireccion;
25     }
26     public String getDescDireccion() {
27         return descDireccion;
28     }
29     public void setDescDireccion(String descDireccion) {
30         this.descDireccion = descDireccion;
31     }
32 }
33
34 }
35
36 }
37
38 }
39

```

Nota. Source.

Figura 30

Funcionario

```

Funcionario.java
1 package espe.edu.ec.inventario.modelo;
2
3 import java.io.Serializable;
4
5
6
7
8
9
10 @Entity
11 @Table ( name = "VA_FUNCIONARIO" )
12 public class Funcionario implements Serializable{
13
14     /**
15      *
16      */
17     private static final long serialVersionUID = 1L;
18
19     @Id
20     @Column ( name = "CODIGO" )
21     private int codigo;
22     @Column ( name = "CODIGO_EMPRESA" )
23     private int codigoEmpresa;
24     @Column ( name = "LOCALIDAD" )
25     private int localidad;
26     @Column ( name = "NOMBRE" )
27     private String nombre;
28     @Column ( name = "IDENTIFICACION" )
29     private String identificacion;
30
31     public int getCodigo() {
32         return codigo;
33     }
34     public void setCodigo(int codigo) {
35         this.codigo = codigo;
36     }
37     public int getCodigoEmpresa() {
38         return codigoEmpresa;
39     }

```

Nota. Source.

Figura 31

Sección

```

Seccion.java
1 package espe.edu.ec.inventario.modelo;
2
3 import java.io.Serializable;
4
5
6
7
8
9
10 @Entity
11 @Table ( name = "AF_SECCION" )
12 public class Seccion implements Serializable{
13
14     /**
15      *
16      */
17     private static final long serialVersionUID = 1L;
18
19     @Id
20     @Column ( name = "CODIGO_SECCION" )
21     private int codigoSeccion;
22     @Column ( name = "COD_AREA" )
23     private int codigoArea;
24     @Column ( name = "COD_DIRECCION" )
25     private int codigoDireccion;
26     @Column ( name = "DESC_SECCION" )
27     private String descSeccion;
28
29     public int getCodigoSeccion() {
30         return codigoSeccion;
31     }
32     public void setCodigoSeccion(int codigoSeccion) {
33         this.codigoSeccion = codigoSeccion;
34     }
35     public int getCodigoArea() {
36         return codigoArea;
37     }
38     public void setCodigoArea(int codigoArea) {
39         this.codigoArea = codigoArea;

```

Nota. Source.

Figura 32

Subsección

```

Subseccion.java
1 package espe.edu.ec.inventario.modelo;
2
3 import java.io.Serializable;
9
10 @Entity
11 @Table ( name = "AF_SUBSECCION" )
12 public class Subseccion implements Serializable{
13
14     /**
15      *
16      */
17     private static final long serialVersionUID = 1L;
18
19     @Id
20     @Column ( name = "CODIGO_SUBSECCION" )
21     private int codigoSubseccion;
22     @Column ( name = "COD_SECCION" )
23     private int codigoSeccion;
24     @Column ( name = "COD_AREA" )
25     private int codigoArea;
26     @Column ( name = "COD_DIRECCION" )
27     private int codigoDireccion;
28     @Column ( name = "DESC_SUBSECCION" )
29     private String descSubseccion;
30
31     public int getCodigoSubseccion() {
32         return codigoSubseccion;
33     }
34     public void setCodigoSubseccion(int codigoSubseccion) {
35         this.codigoSubseccion = codigoSubseccion;
36     }
37     public int getCodigoSeccion() {
38         return codigoSeccion;
39     }

```

Nota. Source.

También se va a utilizar las anotaciones `@Repository`, `@Query`, `@Param` para las interfaces de repositorios y servicios de cada una de las entidades anteriormente mencionadas como se muestra en las siguientes figuras.

Figura 33

Fichero



Nota. Source.

Figura 34

ActivoRfid

```

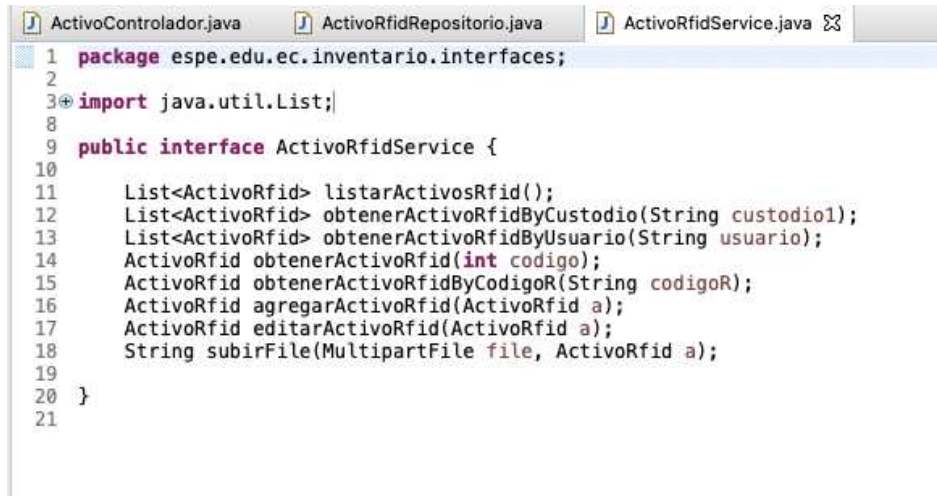
1 package espe.edu.ec.inventario.interfaces;
2
3 import java.util.List;
11
12 @Repository
13 public interface ActivoRfidRepositorio extends JpaRepository<ActivoRfid, Integer>{
14
15     @Query(nativeQuery = true, value = "INSERT INTO ACTIVOSRFID (CODIGO_RFID,"
16         + "CUSTODIO1,CUSTODIO2,DESCRIPCION) VALUES (:A, :B , :C, :D, :E, :F, :G)")
17     ActivoRfid save(@Param("A") String codigoR,@Param("B") String custodio1,
18         @Param("C") String custodio2,@Param("D") String descripcion,
19         @Param("E") String usuario,@Param("F") String estadoJustificado,
20         @Param("G") String estadoTransaccion);
21
22     // @Query(nativeQuery = true, value = "SELECT * FROM ACTIVOSRFID")
23     List<ActivoRfid> findAll();
24     @Query(nativeQuery = true, value = "SELECT * FROM ACTIVOSRFID a WHERE a.CODIGO_RFID = :C")
25     ActivoRfid obtenerActivosRfidByCodigoR(@Param("C") String codigoR);
26
27     @Query(nativeQuery = true, value = "SELECT * FROM ACTIVOSRFID a WHERE a.CUSTODIO1 = :C")
28     List<ActivoRfid> obtenerActivoRfidByCustodio(@Param("C") String custodio1);
29
30     @Query(nativeQuery = true, value = "SELECT * FROM ACTIVOSRFID a WHERE a.CODIGO = :C")
31     ActivoRfid obtenerActivosRfidByCodigo(@Param("C") int codigo);
32
33     @Query(nativeQuery = true, value = "SELECT * FROM ACTIVOSRFID a WHERE a.USUARIO = :C")
34     List<ActivoRfid> obtenerActivosRfidByUsuario(@Param("C") String usuario);
35 }
36

```

Nota. Source.

Figura 35

ActivoRfid



```

1 package espe.edu.ec.inventario.interfaces;
2
3 import java.util.List;
4
5
6
7
8
9 public interface ActivoRfidService {
10
11     List<ActivoRfid> listarActivosRfid();
12     List<ActivoRfid> obtenerActivoRfidByCustodio(String custodio1);
13     List<ActivoRfid> obtenerActivoRfidByUsuario(String usuario);
14     ActivoRfid obtenerActivoRfid(int codigo);
15     ActivoRfid obtenerActivoRfidByCodigoR(String codigoR);
16     ActivoRfid agregarActivoRfid(ActivoRfid a);
17     ActivoRfid editarActivoRfid(ActivoRfid a);
18     String subirFile(MultipartFile file, ActivoRfid a);
19
20 }
21

```

Nota. Source.

Parte del proyecto también va a usar las anotaciones @CrossOrigin, @RestController, @RequestMapping, @Autowired, @GetMapping, @PostMapping para los controladores de cada una de las entidades anteriormente mencionadas como se muestra en las siguientes figuras.

Figura 36

Fichero



```

espe.edu.ec.inventario.controlador
├── ActivoControlador.java
├── ActivoRfidControlador.java
├── ArchivoJustifControlador.java
├── AreaControlador.java
├── AuditoriaAppControlador.java
├── DireccionControlador.java
├── FuncionarioControlador.java
├── GlobalExceptionHandler.java
├── SeccionControlador.java
├── SubseccionControlador.java
└── UsuarioControlador.java

```

Nota. Source.

Figura 37

ActivoRfid

```

ActivoRfidControlador.java
28 @CrossOrigin(origins = "*", maxAge = 3600, allowCredentials = "true")
29 @RestController
30 @RequestMapping({"/activosRfid"})
31 public class ActivoRfidControlador {
32
33     @Autowired
34     ActivoRfidService service;
35     @Autowired
36     ActivoService service2;
37
38     @Autowired
39     Rfid r;
40
41     @GetMapping("/listarActivosRfid")
42     public List<ActivoRfid> listarActRfid(){
43         return service.listarActivosRfid();
44     }
45
46     @PostMapping("/registrarActivoRfid")
47     // @RequestMapping(value = "/", method = RequestMethod.POST, consumes = MediaType
48     public ActivoRfid registrarActivo(@RequestBody ActivoRfid activoR) {
49         return service.agregarActivoRfid(activoR);
50     }
51
52     @PostMapping("/actualizarActivoRfid")
53     public ActivoRfid actualizarActivo(@RequestBody ActivoRfid activoR) {
54         return service.editarActivoRfid(activoR);
55     }
56
57     @GetMapping("/obtenerActivoRfid")
58     public ActivoRfid actRfid(String codigoR){
59         return service.obtenerActivoRfidByCodigoR(codigoR);
60     }
61

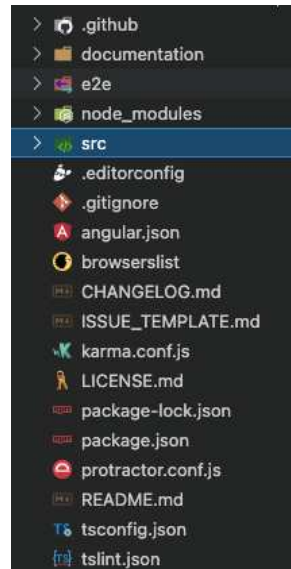
```

Nota. Source.

Desarrollo de la aplicación web

Creación de proyecto

Para realizar el FRONT-END del sistema web del presente proyecto se va a realizar la creación de un proyecto angular el cual tiene la siguiente estructura.

Figura 38*Fichero**Nota. Source.*

Node_modules: En esta carpeta se encuentran todas las dependencias que utiliza nuestro proyecto.

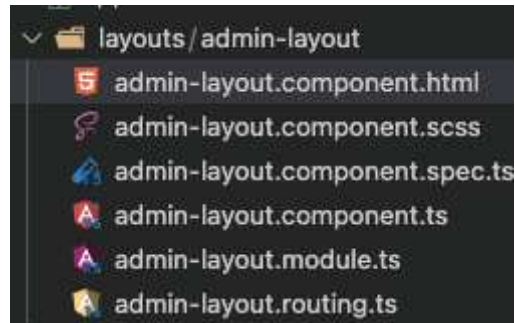
Src: En esta carpeta encontraremos los diferentes componentes y módulos; también contiene el código fuente de nuestro proyecto.

E2e: En esta carpeta se crean ficheros que permite realizar pruebas de la aplicación.

Para el desarrollo de nuestro proyecto se crea un componente llamado “admin-layout”, el cual contiene la validación para saber si un usuario ha realizado el login correctamente. También es el encargado de mapear las diferentes opciones del menú de nuestro proyecto.

Figura 39

Fichero



Nota. Source.

Figura 40

Fichero

```
admin-layout.component.html x admin-layout.module.ts
src > app > layouts > admin-layout > admin-layout.component.html > div.wrapper > div
1 <div class="wrapper" *ngIf="service.usuarioLogeado == true">
2   <div class="sidebar" data-color="white" data-active-color="danger">
3     <app-sidebar-cmp></app-sidebar-cmp>
4   </div>
5   <div class="main-panel">
6     <navbar-cmp></navbar-cmp>
7     <div class="content">
8       <router-outlet></router-outlet>
9     </div>
10    <footer-cmp></footer-cmp>
11  </div>
12  <div *ngIf="service.pendienteTransaccion == true">
13    <fixedplugin-cmp></fixedplugin-cmp>
14  </div>
15 </div>
16 <div *ngIf="service.usuarioLogeado == false">
17   <app-login></app-login>
18 </div>
19
```

Nota. Source.

Figura 41

Fichero

```

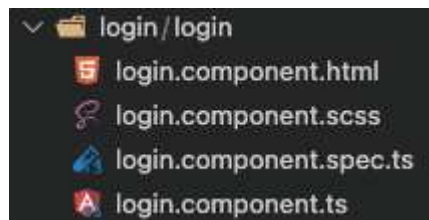
admin-layout.component.html  admin-layout.module.ts x
src > app > layouts > admin-layout > admin-layout.module.ts > ...
7  import { AdminLayoutRoutes } from './admin-layout.routing';
8  import { PaginatePipe } from '../pipes/paginate.pipe';
9  import { FilterPipe } from '../pipes/filter.pipe';
10
11 import { DashboardComponent } from '../pages/dashboard/dashboard.component';
12 import { UserComponent } from '../pages/user/user.component';
13 import { TableComponent } from '../pages/table/table.component';
14 import { TypographyComponent } from '../pages/typography/typography.component';
15 import { IconsComponent } from '../pages/icons/icons.component';
16 import { MapsComponent } from '../pages/maps/maps.component';
17 import { NotificationsComponent } from '../pages/notifications/notifications.component';
18 import { UpgradeComponent } from '../pages/upgrade/upgrade.component';
19
20 import { NgbModule } from '@ng-bootstrap/ng-bootstrap';
21
22 @NgModule({
23   imports: [
24     CommonModule,
25     RouterModule.forChild(AdminLayoutRoutes),
26     FormsModule,
27     NgbModule,
28     MatPaginatorModule
29   ],
30   declarations: [
31     DashboardComponent,
32     UserComponent,
33     TableComponent,
34     UpgradeComponent,
35     TypographyComponent,
36     IconsComponent,
37     MapsComponent,
38     NotificationsComponent,
39     PaginatePipe,
40     FilterPipe
41 ]

```

Nota. Source.

Figura 42

Fichero



Nota. Source.

Figura 43

Fichero

```
login.component.html x
src > app > login > login > login.component.html > div.card.card-user > div.card-body > div.card.card-user > div.card-body >
1 <div class="card card-user">
2   <div class="image">
3     
4   </div>
5   <div class="card-body">
6     <div class="author">
7       <a href="#">
8         
9         <h5 class="title">LOGIN</h5>
10      </a>
11      <p class="description">
12        @espe
13      </p>
14    </div>
15    <p class="description text-center">
16      Unidad de Bienes
17      <br> Programa
18      <br> Para realizar tona fisica
19    </p>
20  </div>
21 </div>
22 <div class="row">
23   <div class="col-md-4"></div>
24   <div class="card-header col-md-4 text-center">
25     <h5 class="card-title">Ingresar Datos</h5>
26   </div>
27 </div>
28 <div class="card-body">
29   <form>
30     <div class="row">
31       <div class="col-md-4"></div>
32       <div class="col-md-4">
33         <div class="form-group">
34           <label-Usuario</label>
35           <input type="text" class="form-control" placeholder="Usuario" [(ngModel)]="usuario" name="usuario">
36         </div>
37       </div>
38     </div>
39   </form>
40 </div>
41 </div>
42 </div>
```

Nota. Source.

Figura 44

Fichero

```
login.component.html login.component.ts x
src > app > login > login > login.component.ts > LoginComponent > obtenerEstado > subscribe()
31   <!-- @ts-ignore -->
32   icon: 'success',
33   confirmButtonText: 'OK'
34 });
35   this.service.nombreUsuario = this.usuario.toUpperCase();
36   this.service.usuarioLogeado = true;
37   this.audit.descripcion = 'El usuario ' + this.usuario + ' ingreso correctamente';
38   this.audit.usuario = this.usuario;
39   this.audit.opcion = 'LOGIN';
40   this.service.ingresarRegistroAuditoria(this.audit).subscribe();
41
42   this.obtenerEstado();
43   this.router.navigate(['dashboard']);
44 } else {
45   Swal.fire({
46     title: 'Error',
47     text: 'Fallo el ingreso',
48     icon: 'error',
49     confirmButtonText: 'OK'
50   });
51 }
52 }
53
54 obtenerEstado() {
55   this.service.activosPendientes = [];
56   this.service.getActivosRfidByUsuario(this.service.nombreUsuario.toUpperCase())
57   .subscribe( data => {
58     console.log(data.length);
59     data.forEach(element => {
60       if (element.estadoTransaccion.toUpperCase() === 'PENDIENTE') {
61         this.service.pendienteTransaccion = true;
62         this.listaAct.push(element);
63       }
64     });
65   this.service.activosPendientes = this.listaAct;
66 });
67 }
```

Nota. Source.

Figura 45

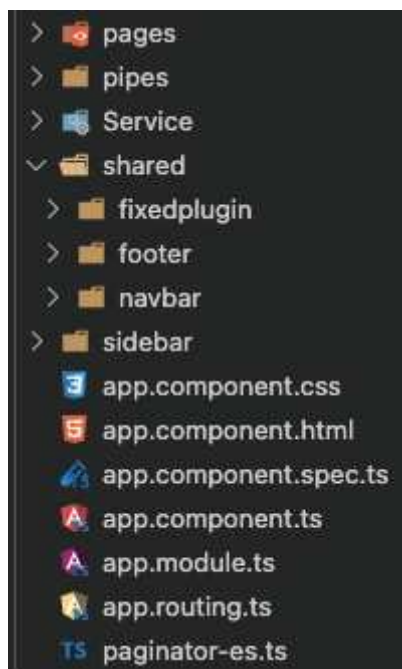
Modelo



Nota. Modelo de cada entidad.

Figura 46

Fichero



Nota. Source.

pages: En esta carpeta las diferentes páginas con su lógica desarrollada en typescript y html.

pipe: Esto es un componente que nos permite utilizar angular para realizar funciones específicas, como por ejemplo la paginación de las tablas.

services: Aquí se encuentra el archivo que contiene las urls para el consumo de los servicios realizados en el back-end.

CAPÍTULO V

RESULTADOS OBTENIDOS Y PRUEBAS

Pruebas de rendimiento

Definición

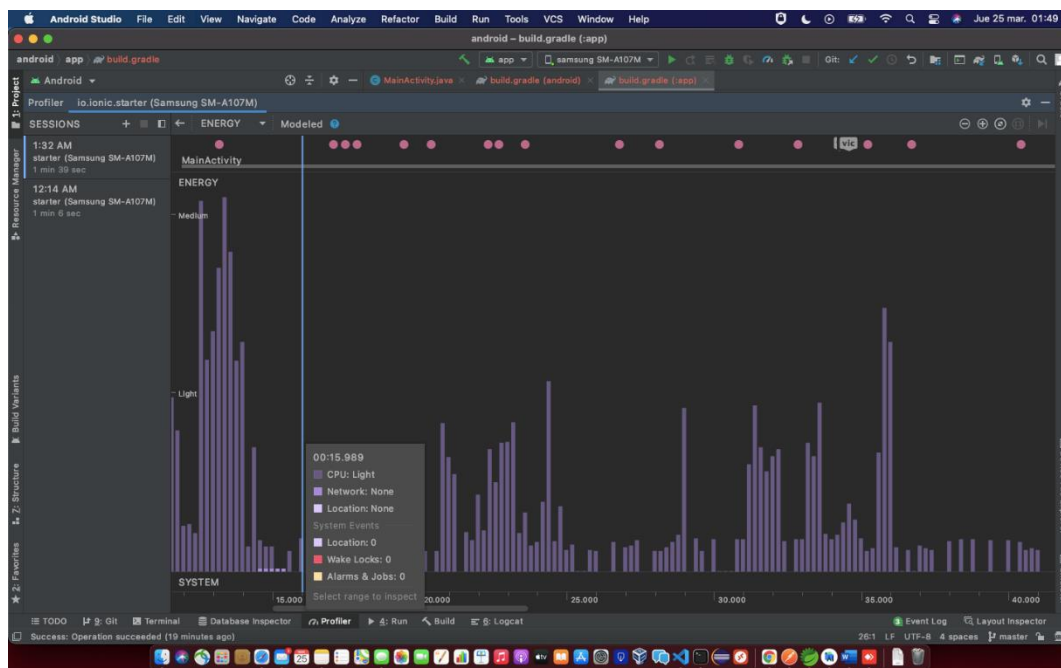
Las pruebas de rendimiento son aquellas pruebas que someten a un sistema a una carga de trabajo con el fin de medir su velocidad, fiabilidad y estabilidad en esas condiciones de trabajo (Sommerville, 2005).

Uso de batería

Se realizó una prueba del consumo de batería de dispositivo móvil Android al momento de correr y navegar por las diferentes opciones que tiene la aplicación móvil.

Figura 47

Energía



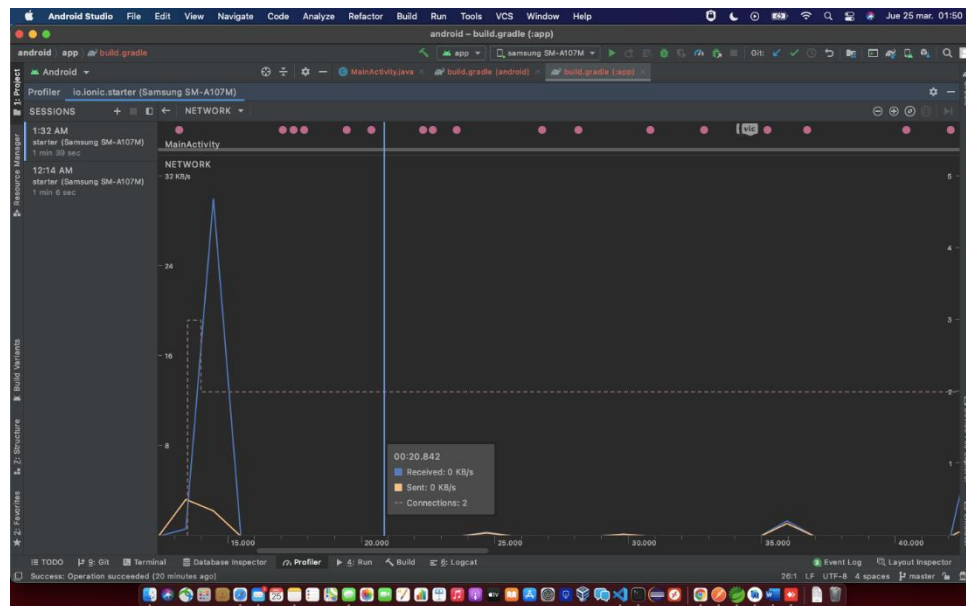
Nota. Consumo de energía.

Uso de red

Se realizó una prueba del consumo de datos de un dispositivo móvil Android al momento correr y navegar por las diferentes opciones dentro de la aplicación móvil.

Figura 48

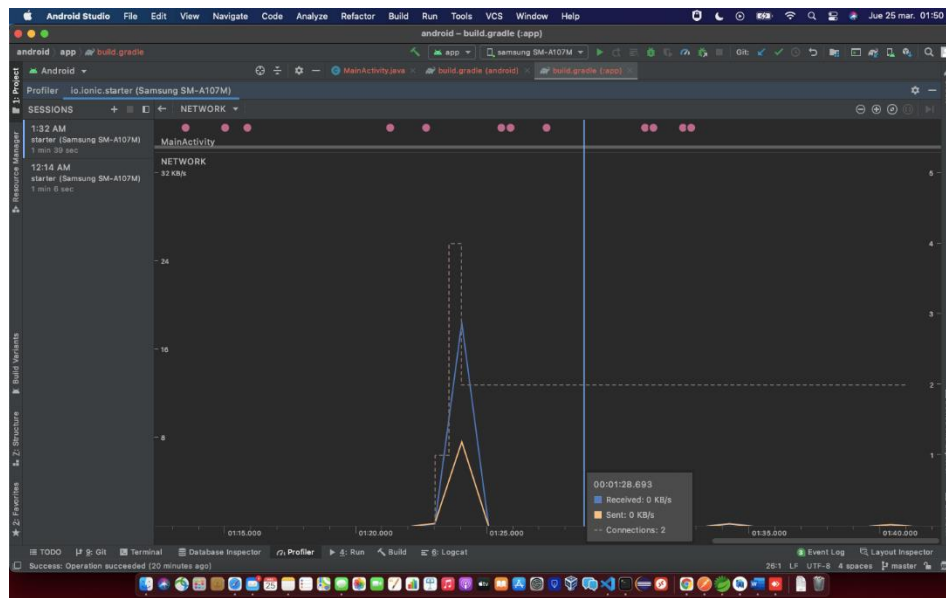
Red



Nota. Consumo de red inicial.

Figura 49

Red



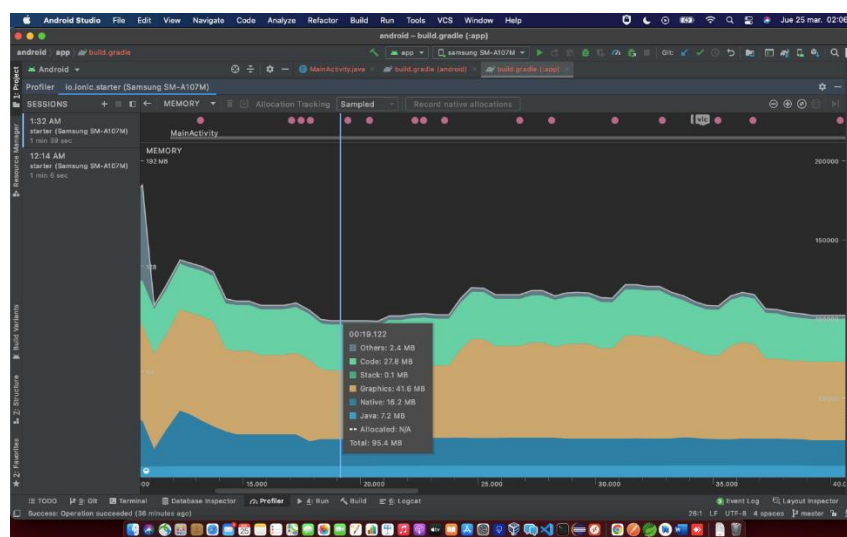
Nota. Consumo de red.

Uso de memoria

Se realizó una prueba del consumo de memoria de un dispositivo móvil Android al momento correr y navegar por las diferentes opciones dentro de la aplicación móvil.

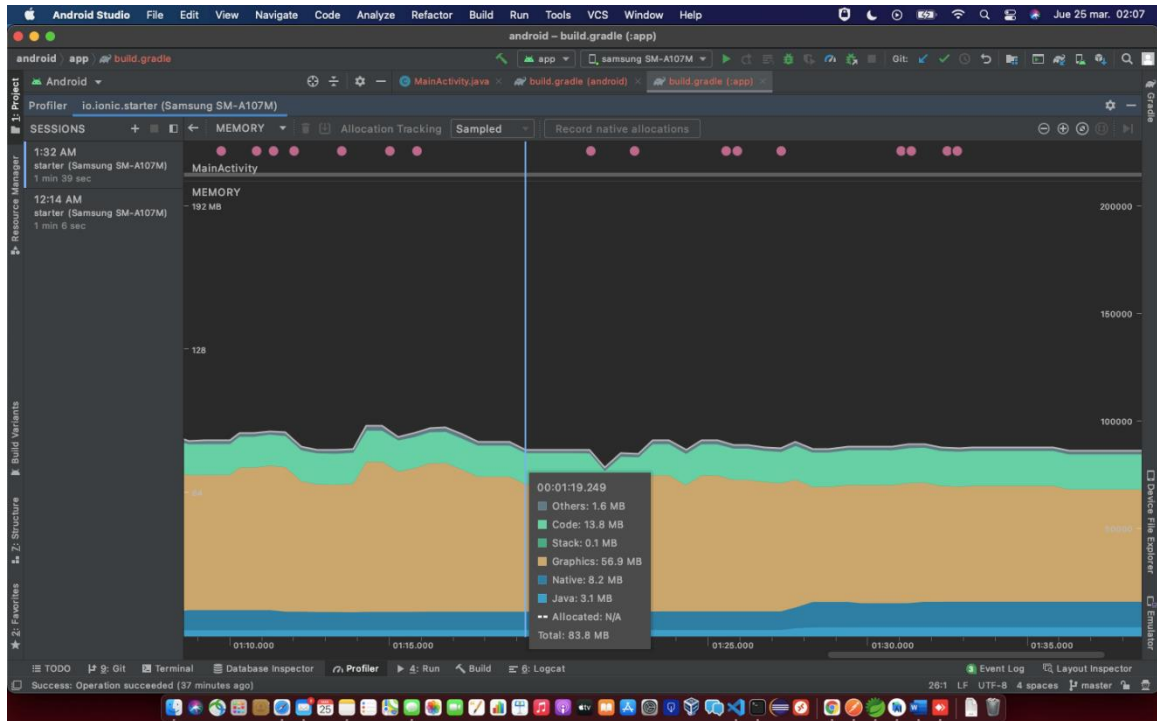
Figura 50

Memoria



Nota. Consumo de memoria inicial.

Figura 51

Memoria

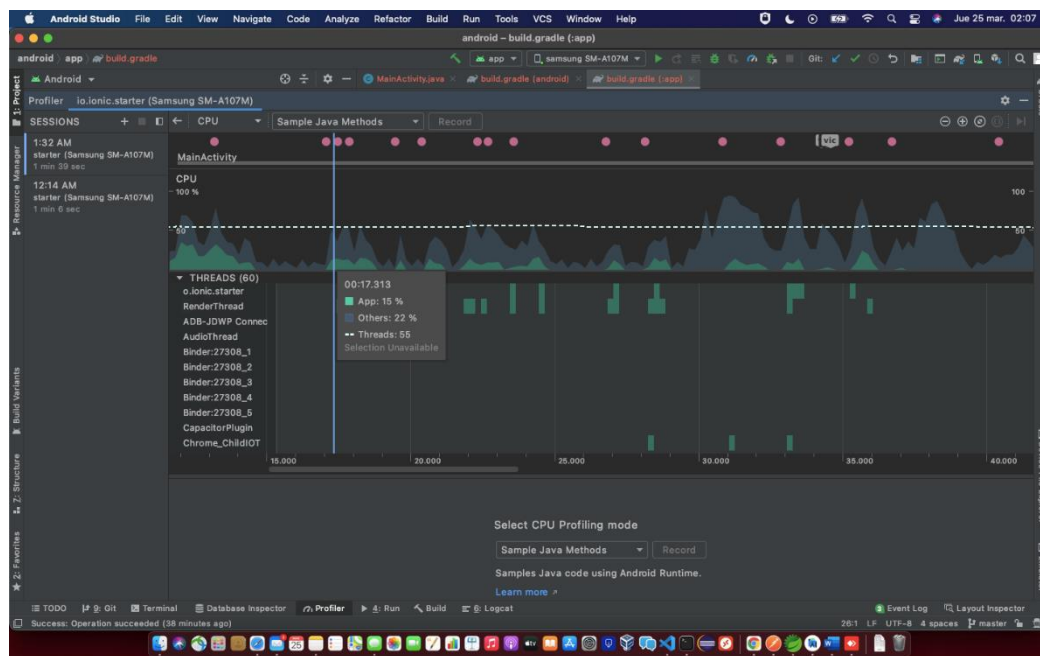
Nota. Consumo de memoria.

Uso de CPU

Se realizó una prueba del consumo de CPU de un dispositivo móvil Android al momento correr y navegar por las diferentes opciones dentro de la aplicación móvil.

Figura 52

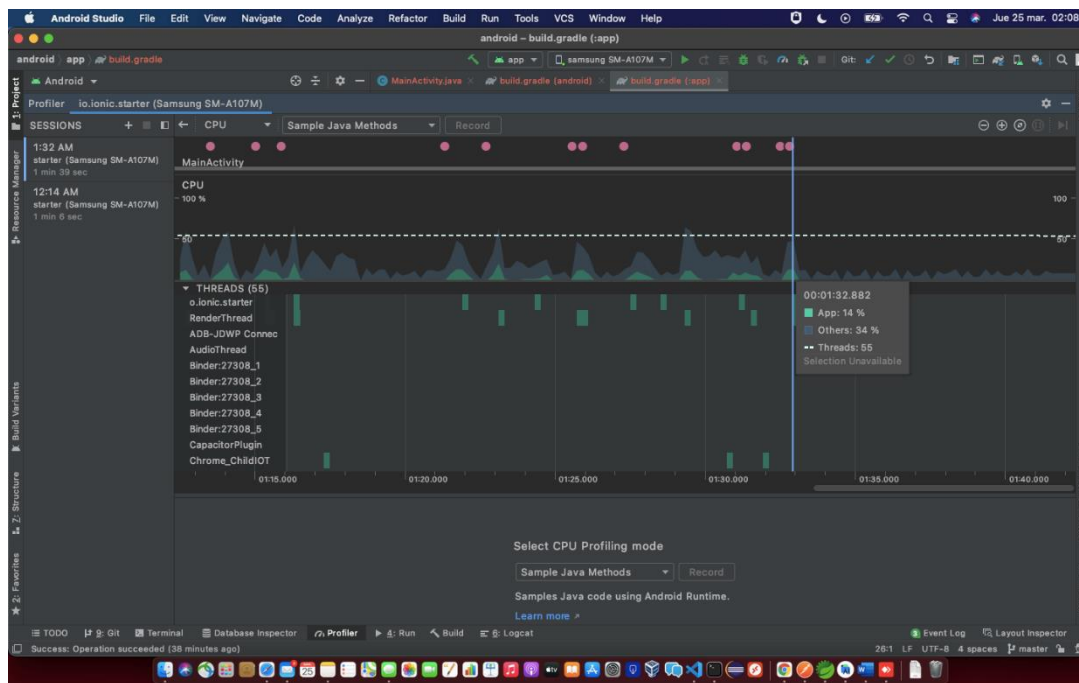
CPU



Nota. Consumo de CPU inicial.

Figura 53

CPU



Nota. Consumo de CPU.

Pruebas de aceptación

Definición

Las Pruebas de aceptación son de vital importancia para el éxito de una iteración y el comienzo de la siguiente, con lo cual el cliente puede conocer el avance en el desarrollo del sistema y a los programadores lo que les resta por hacer. Además, permite una retroalimentación para el desarrollo de las próximas historias de usuarios a ser entregadas. Estas son comúnmente llamadas pruebas del cliente, por lo que son realizadas por el encargado de verificar si las historias de usuarios de cada iteración cumplen con la funcionalidad esperada (SINTYA MILENA MELÉNDEZ VALLADAREZ, 2016).

Tabla 12

Pruebas aceptación 1

Pruebas de aceptación web	
ID	P0001
Descripción	Sistema de control
Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente
Instrucciones	Ingresar a la url http://localhost:8080/login
Resultados Esperados	Se muestra la pantalla de inicio para que el usuario pueda ingresar su usuario y contraseña
Fecha de ejecución	26 de marzo del 2021
Ejecutado por	Victor Apunte
Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

Tabla 13*Pruebas aceptación 2*

Pruebas de aceptación web	
ID	P0002
Descripción	Sistema de control
Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente
Instrucciones	Una vez ingresado en el sistema, ir a la opción de menú para de Búsqueda: - Seleccionar el tipo de búsqueda - Ingresar los datos solicitados
Resultados Esperados	Se muestra en una tabla los datos de los bienes conforme a los datos ingresados en los criterios de búsqueda.
Fecha de ejecución	26 de marzo del 2021
Ejecutado por	Victor Apunte
Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

Tabla 14*Pruebas aceptación 3*

Pruebas de aceptación web	
ID	P0003
Descripción	Sistema de control

Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente
Instrucciones	Una vez ingresado en el sistema, ir a la opción de menú para de Búsqueda: - Seleccionar el tipo de búsqueda - Ingresar los datos solicitados
Resultados Esperados	Se muestra en una tabla los datos de los bienes conforme a los datos ingresados en los criterios de búsqueda.
Fecha de ejecución	26 de marzo del 2021
Ejecutado por	Victor Apunte
Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

Tabla 15*Pruebas aceptación 4*

Pruebas de aceptación web	
ID	P0004
Descripción	Sistema de control
Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente
Instrucciones	Procesos pendientes: - Ingresar a la opción de menú Toma Física

	- Seleccione un proceso pendiente (aquí se muestra la identificación de los custodios)
Resultados Esperados	Se muestra el nombre del custodio seleccionado con los respectivos bienes constatados físicamente.
Fecha de ejecución	26 de marzo del 2021
Ejecutado por	Víctor Apunte
Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

Tabla 16*Pruebas aceptación 5*

Pruebas de aceptación web	
ID	P0005
Descripción	Sistema de control
Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente
Instrucciones	Procesos comparativos: Una vez seleccionado el proceso pendiente puede ingresar a la opción de menú Comparar.
Resultados Esperados	Se muestra una tabla con los diferentes bienes sobrantes. Se muestra otra tabla con los bienes no constatados físicamente.
Fecha de ejecución	26 de marzo del 2021

Ejecutado por	Victor Apunte
Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

Tabla 17*Pruebas aceptación 6*

Pruebas de aceptación web	
ID	P0006
Descripción	Sistema de control
Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente
Instrucciones	Procesos comparativos: Una vez se muestran los bienes, se puede modificar los bienes sobrantes para que se muestren o no en el acta que se va a generar posteriormente.
Resultados Esperados	Se actualiza la tabla de sobrantes cambiando el estado de los bienes que no se van a mostrar en el acta.
Fecha de ejecución	26 de marzo del 2021
Ejecutado por	Victor Apunte
Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

Tabla 18*Pruebas aceptación 7*

Pruebas de aceptación web	
ID	P0007
Descripción	Sistema de control
Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente
Instrucciones	<p>Generar acta:</p> <ul style="list-style-type: none"> - Ingrese a la opción de menú Acta - Ingresar los datos solicitados - Dar clic en guardar - Dar clic en generar acta
Resultados Esperados	<p>Primero se guarda los campos ingresados que forman parte del acta.</p> <p>Se genera un archivo en PDF con el formato entregado por la Unidad de Bienes, el cual se puede imprimir o guardar en el directorio del equipo.</p>
Fecha de ejecución	26 de marzo del 2021
Ejecutado por	Victor Apunte
Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

Tabla 19*Pruebas aceptación 8*

Pruebas de aceptación móvil	
ID	P0001
Descripción	Aplicación móvil
Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente
Instrucciones	Se instala la aplicación en el teléfono y posteriormente se abre la app
Resultados Esperados	Se muestra la pantalla de inicio para que el usuario pueda ingresar su usuario para validar que exista
Fecha de ejecución	26 de marzo del 2021
Ejecutado por	Victor Apunte
Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

Tabla 20*Pruebas aceptación 9*

Pruebas de aceptación móvil	
ID	P0002
Descripción	Aplicación móvil
Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente

Instrucciones	Una vez ingresado en la aplicación, en la página de inicio puede conectarse por bluetooth al lector de RFID
Resultados Esperados	Se muestra una lista de dispositivos bluetooth y se muestra un botón para conectarse al dispositivo
Fecha de ejecución	26 de marzo del 2021
Ejecutado por	Victor Apunte
Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

Tabla 21*Pruebas aceptación 10*

Pruebas de aceptación móvil	
ID	P0003
Descripción	Aplicación móvil
Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente
Instrucciones	Puede ir a la opción de menú para de Búsqueda: - Seleccionar el tipo de búsqueda - Ingresar los datos solicitados
Resultados Esperados	Se muestra en una tabla los datos de los bienes conforme a los datos ingresados en los criterios de búsqueda.
Fecha de ejecución	26 de marzo del 2021
Ejecutado por	Victor Apunte

Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

Tabla 22*Pruebas aceptación 11*

Pruebas de aceptación móvil	
ID	P0004
Descripción	Aplicación móvil
Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente
Instrucciones	Proceso de toma física: - Ingresar a la opción de menú toma física - Pulsa el botón del lector - Da clic en actualizar
Resultados Esperados	Se muestra una lista de todas las etiquetas puesta en los bienes que alcanza a leer el lector de RFID.
Fecha de ejecución	26 de marzo del 2021
Ejecutado por	Victor Apunte
Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

Tabla 23*Pruebas aceptación 12*

Pruebas de aceptación móvil	
ID	P0005
Descripción	Aplicación móvil
Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente
Instrucciones	Procesos comparativos: Una vez realizado el proceso de lectura puede ingresar a la opción de menú Comparar.
Resultados Esperados	Se muestra una lista con los diferentes bienes sobrantes, una lista con los bienes constatados físicamente. Se muestra otra tabla con los bienes no constatados físicamente.
Fecha de ejecución	26 de marzo del 2021
Ejecutado por	Victor Apunte
Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

Tabla 24*Pruebas aceptación 13*

Pruebas de aceptación móvil	
ID	P0006
Descripción	Aplicación móvil

Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente
Instrucciones	Procesos comparativos: Una vez se muestran los bienes, se puede modificar los bienes sobrantes para que se muestren o no en el acta que se va a generar posteriormente.
Resultados Esperados	Se actualiza la tabla de sobrantes cambiando el estado de los bienes que no se van a mostrar en el acta.
Fecha de ejecución	26 de marzo del 2021
Ejecutado por	Victor Apunte
Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

Tabla 25*Pruebas aceptación 14*

Pruebas de aceptación móvil	
ID	P0007
Descripción	Aplicación móvil
Inicialización	Validar que la base de datos y el servidor de aplicaciones funcionen correctamente
Instrucciones	Generar acta: - Ingrese a la opción de menú Acta - Ingresar los datos solicitados

	<p>- Dar clic en guardar</p> <p>- Dar clic en generar acta</p>
Resultados Esperados	<p>Primero se guarda los campos ingresados que forman parte del acta.</p> <p>Se genera un archivo en PDF con el formato entregado por la Unidad de Bienes, el cual se puede imprimir o guardar en el directorio del equipo.</p>
Fecha de ejecución	26 de marzo del 2021
Ejecutado por	Victor Apunte
Lugar de ejecución	Universidad de las Fuerzas Armadas
Estado de prueba	Finalizada

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Una vez realizado el análisis técnico de los sistemas: eSByE y OLYMPO, los mismo que intervienen el proceso de la toma física de bienes en la Universidad de las Fuerzas Armadas ESPE, se concluyó que los datos de los bienes y custodios del sistema OLYMPO se pudo utilizar para la integración con el proyecto propuesto.
- También se pudo concluir que el presente proyecto puede automatizar y dinamizar la consulta de bienes por custodio, así como la generación del acta de constatación física que forman parte del proceso de control de activos de la Universidad.
- Una vez investigado sobre la tecnología RFID, se puede concluir que esta tiene ventaja sobre el uso de código de barras porque no solo permite identificar un solo bien, sino varios simultáneamente, ya que las etiquetas RFID emiten una frecuencia que se puede leer en un radio de más/menos 80 centímetros, que varía de acuerdo con el modelo del lector RFID que se utilice. Dando como resultado un ahorro de aproximadamente 50% del tiempo invertido en la identificación de un bien.
- Para el desarrollo del presente proyecto se utilizó la metodología de desarrollo de Software XP, porque esta permite realizar la aplicación web y móvil del presente proyecto por iteraciones, respondiendo a las necesidades específicas que tenía el usuario, lo que se puede ver reflejado en la aceptación certificada por parte de la Unidad de Logística de la universidad y su personal administrativo (Anexo: Acta de aceptación).

- La aplicación móvil evita la dependencia de un computador, puesto que, en esta se puede realizar tanto la consulta como la constatación física de los bienes y la generación de su respectiva acta.
- Se concluyó que para la integración de los sistemas: eSByE y OLYMPO que la ESPE utiliza actualmente en el proceso de la toma física y los sistemas desarrollados en este proyecto, se debe hacerlo utilizando la base de datos del sistema OLYMPO, ya que este cuenta con la información histórica y actualizada de los bienes que pertenecen a la universidad.
- Al realizar la capacitación al personal de la Unidad de Logística sobre el manejo de las aplicaciones, se pudo evidenciar que las mismas son amigables con el usuario, de fácil manejo y útil para el proceso de toma física que actualmente se lleva a cabo.

RECOMENDACIONES

- Para realizar más de una aplicación se recomienda utilizar el framework Spring Boot en el cual se haga el BACK – END que contiene las conexiones y manipulaciones de los datos y por otro lado se debe desarrollar el FRONT – END de cada aplicación con un framework de acuerdo con el aplicativo que se quiera desarrollar.
- Se recomienda utilizar la tecnología RFID con lectores, etiquetas y antenas RFID con una frecuencia mayores a 900 mhz para tener un mejor control y mayor seguridad sobre la ubicación y movimiento de los bienes.
- Se recomienda adquirir etiquetas tipo Flag(ZT400R) para dispositivos electrónicos, etiquetas tipo Jewelry(ZD500R) para bienes de alto costo, etiquetas tipo On-Meta(ZT410) para bienes metálicos, entre los principales, ya que existen etiquetas RFID para los diferentes tipos de superficies y áreas.

- Para mejorar el uso de las aplicaciones desarrolladas, en el primer año se debe etiquetar todos los bienes pertenecientes a la Universidad de las Fuerzas Armadas – ESPE, tomando en cuenta que al momento de etiquetar los mismo se los debe colocar en una parte no visible para evitar su manipulación por parte de usuarios no autorizados.

GLOSARIOS DE TÉRMINOS

- Framework: Es el conjunto de herramientas que ayudan al programador a facilitar el desarrollo de una aplicación-
- RFID: Identificación de Radio Frecuencia
- Etiquetas RFID: Los dispositivos plegables que cuentan con la tecnología de radio frecuencia para responder a peticiones de los lectores de RFID.
- esByE: Sistema impuesto por el estado para que la Universidad de Las Fuerzas Armadas ESPE lleve un control de los bienes de esta.
- OLYMPO: Sistema utilizado por la universidad para llevar el histórico de los bienes de la misma.
- Requerimientos: Son las necesidades que el cliente solicita que los aplicativos deben cumplir al momento de utilizar los sistemas.
- BACK-END: Se lo conoce a toda la parte lógica de cualquier sistema sea web o móvil.
- FRONT-END: Es la parte de un aplicativo web o móvil que interactúa con el usuario que lo utilice.

REFERENCIAS BIBLIOGRÁFICAS

- Alejandro Medina Santiago, Ciclalli Cabrera, Jorge Luis Camas. (2012). *Sistemas de Control de Inventario de Activos Fijos Por Rfid*. Madrid: EAE Editorial.
- Angular. (08 de Marzo de 2021). *Angular Docs*. Obtenido de Angular Docs:
<https://angular.io/guide/what-is-angular>
- Antonio Hernando Grande, Alejandro Úbeda Maeso, Javier López Facal. (25 de Junio de 2010). *Radiofrecuencias*. Madrid: CSIC.
- Beck, K. (2002). *Una explicación de la programación extrema*. Boston: Addison Wesley.
- Bonet, F. P. (2018). *Desarrollo de una tarjeta RFID*. Barcelona: Universitat de Barcelona.
- CAPELO BADILLO ALDO GRIVALDY, C. G. (19 de Octubre de 2009). DISEÑO Y ELABORACIÓN DE UN SISTEMA DE SEGURIDAD VEHICULAR CON TECNOLOGIA RFID. *Tesis*. Universidad de las Fuerzas Armadas, Latacunga. Obtenido de DISEÑO Y ELABORACIÓN DE UN SISTEMA DE SEGURIDAD VEHICULAR CON TECNOLOGIA RFID:
<https://repositorio.espe.edu.ec/bitstream/21000/3989/1/T-ESPEL-0646.pdf>
- Community, A. (16 de Marzo de 2021). *Angular*. Obtenido de Angular: <https://angular.io/docs>
- community, I. (09 de Marzo de 2021). *Ionic*. Obtenido de Ionic doc:
<https://ionicframework.com/docs>
- Edu. (19 de Febrero de 2021). *Educalingo*. Obtenido de Educalingo:
<https://educalingo.com/es/dic-es/factibilidad>
- ESPE. (01 de Febrero de 2021). *Universidad de las Fuerzas Armadas*. Obtenido de Universidad de las Fuerzas Armadas: <https://espe-el.espe.edu.ec/filosofia/>
- Finanzas, M. d. (14 de 12 de 2009). *Capacitación finanzas*. Obtenido de Capacitación finanzas:
https://capacitacion.finanzas.gob.ec/sga/adjuntos_curso/SYLLABUS_BIENES%20Y%20EXISTENCIAS.pdf

- Flamarique, S. (2009). *Gestión de operaciones de manejo*. Barcelona: Marge Books.
- Fuentes, J. R. (2015). *Desarrollo de Software Ágil*. Barcelona: IT Campus Academy.
- García, J. I. (2008). *Tecnología de identificación por radiofrecuencia (RFID)*. Madrid: ISBN Editorial.
- Gomez, F. L. (2021). *Marketing en el punto de venta*. Madrid: Paraninfo.
- González, M. J. (2021). *Optimización de la cadena de logística*. Málaga: IC Editorial.
- Heurtel, O. (2015). *Oracle 12c Administración*. Barcelona: Editions ENI.
- Ionic, C. (24 de Marzo de 2021). *Ionic Framework*. Obtenido de Ionic Framework:
<https://ionicframework.com/docs>
- JENNIFER, M. R. (2018). Docplayer. (Tesis). Universidad César Vallejo, Lima. Obtenido de Docplayer: <http://docplayer.es/133694127-Facultad-de-ingenieria.html>
- LATIENDEDELASBARRAS. (12 de Julio de 2019). *LATIENDEDELASBARRAS*. Obtenido de LATIENDEDELASBARRAS: <https://www.latiendadelasbarras.com/barras/informacion-codigo-de-barras/>
- Lázaro Guillén Lázaro, S. E. (2011). *Avances en la investigación de la tecnología RFID y sus aplicaciones*. España: Sociedad Española de Trazabilidad.
- Luis, J. (01 de Enero de 2021). *Como funciona*. Obtenido de Como funciona: <https://como-funciona.co/un-codigo-de-barras/>
- Moya, J. M. (2021). *Radiocomunicaciones*. España: Creaciones Copyright.
- Oracle. (01 de Enero de 2021). *Oracle*. Obtenido de Oracle:
<https://www.oracle.com/es/database/what-is-database/>
- Peris, S. M. (2008). *Distribución comercial*. España: ESIC EDITORIAL.

Perry, J. S. (11 de Mayo de 2017). *IBM Developer*. Obtenido de IBM:

<https://developer.ibm.com/es/languages/java/tutorials/j-spring-boot-basics-perry/>

Philip Kotler, Kevin Lane Keller. (4 de Abril de 2018). *Dirección de Marketing*. Madrid: Prentice Hall.

Salas, H. G. (2009). *Inventarios manejo y control*. Bogota: ECOE Ediciones.

SINTYA MILENA MELÉNDEZ VALLADAREZ, M. E. (28 de Enero de 2016). METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION. (*Tesis*). UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA, Managua. Obtenido de Repositorio:

<https://repositorio.unan.edu.ni/1365/1/62161.pdf>

Sommerville, I. (2005). Ingeniería del Software. En I. Sommerville, *Ingeniería del Software* (pág. 500). Madrid: Pearson Addison Wersley.

UIV, E. (12 de Octubre de 2017). *Universidad Internacional de Valencia*. Obtenido de Universidad Internacional de Valencia: <https://www.universidadviu.com/pe/actualidad/nuestros-expertos/rfid-que-es-y-como-funciona>

Vicent. (31 de Diciembre de 2020). *Vicent Ferrer*. Obtenido de Vicent Ferrer:

<https://vicentferrer.com/que-es-rfid-y-como-funciona/>

VMare. (31 de Marzo de 2021). *Spring*. Obtenido de Spring: <https://spring.io/projects/spring-framework>

VMware. (8 de Marzo de 2021). *Spring Boot*. Obtenido de Spring Boot:

<https://spring.io/projects/spring-boot>

ANEXOS