

**UNIVERSIDAD PERUANA UNIÓN**  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
Escuela Profesional de Ingeniería de Sistemas



*Una Institución Adventista*

**Desarrollo de un sistema de Identificación por Radiofrecuencia para  
el control de inventarios de Activos Fijos en la Universidad Peruana  
Unión Filial Juliaca**

Tesis para obtener el Título Profesional de Ingeniero de Sistemas

Por:  
Daniel Yeber León Sillo

Asesor:  
Ing. Eder Gutierrez Quispe

**Juliaca, febrero de 2020**

## DECLARACIÓN JURADA DE AUTORÍA DEL INFORME DE TESIS

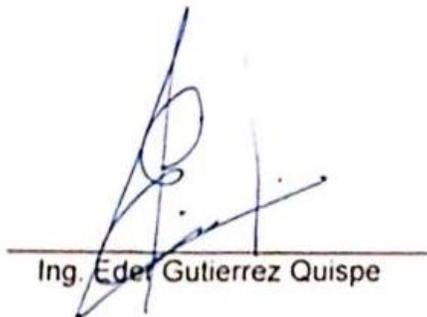
Ing. Eder Gutierrez Quispe, de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que el presente informe de investigación titulado: "DESARROLLO DE UN SISTEMA DE IDENTIFICACIÓN POR RADIO FRECUENCIA PARA EL CONTROL DE INVENTARIOS DE ACTIVOS FIJOS EN LA UNIVERSIDAD PERUANA UNION FILIAL JULIACA" constituye la memoria que presenta el Bachiller Daniel Yeber León Sillo para aspirar al título de Profesional de Ingeniero de Sistemas, ha sido realizada en la Universidad Peruana Unión bajo mi dirección.

Las opiniones y declaraciones en este informe son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente constancia en Juliaca, a los 20 días del mes de febrero del año 2020.



Ing. Eder Gutierrez Quispe

Desarrollo de un Sistema de Identificación por Radiofrecuencia para el control de inventario de Activos Fijos en la Universidad Peruana Unión  
Filial Juliaca

# TESIS

Presentada para optar el título profesional de Ingeniero de Sistemas

## JURADO CALIFICADOR



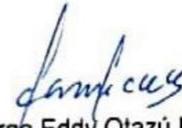
Mg. Lennin Henry Centurión Julca  
Presidente



Ing. Angel Rosendo Condori Coaquira  
Secretario



Ing. David Mamani Parí  
Vocal



Ing. Jorge Eddy Otazú Luque  
Vocal



Ing. Eder Gutierrez Quispe  
Asesor

Juliaca, 04 de febrero de 2020

## **DEDICATORIA**

La presente tesis es dedicada a Dios por ser guía y darme fortaleza para seguir adelante en todo momento. A mis padres Aurelio León Cala y Lilia Rosa Sillo Apaza con mucho amor y cariño por contribuir a mi empeño, por hacer más perfecto aquello en lo que creo. A nuestros maestros por enseñarme el amor al estudio y por su ejemplo de profesionalidad que nunca he de olvidar y todos aquellos que hicieron posible la confección y elaboración de este trabajo.

## **AGRADECIMIENTOS**

A Dios por bendecirme día a día para llegar hasta donde he llegado, porque hizo realidad este sueño anhelado. A mis padres por ser aquellas personas que están conmigo en todo momento a pesar de las dificultades, ya que con sus palabras de aliento me dan animo y fortaleza en las adversidades que se me presentan. A mis compañeros y amigos de clase por su colaboración, comprensión, paciencia y confianza depositada en mí, quienes me han acompañado en esta travesía de aprendizajes y conocimientos.

A mi Escuela Profesional de Ingeniería de Sistemas que me dio la oportunidad de una educación superior de calidad y en la cual he forjado mis conocimientos profesionales día a día. Especial reconocimiento al Ing. Eder Gutierrez y al coordinador Dr. Jorge Alejandro Sánchez Garces, por la orientación, seguimiento y la supervisión continua de la misma y sobre todo por la motivación y el apoyo recibido al largo de estos años. Al área de Activos Fijos por brindar la confianza a los autores para la ejecución del Sistema Informático.

## TABLA DE CONTENIDO

DEDICATORIA .....	iv
AGRADECIMIENTOS.....	v
TABLA DE CONTENIDO.....	vi
ÍNDICE DE TABLAS.....	ix
SIMBOLOS USADOS.....	xiv
RESUMEN .....	xv
ABSTRACT.....	xvi
CAPITULO I.....	17
EL PROBLEMA.....	17
1.1. Descripción de la situación problemática.....	17
1.2. Objetivos de la investigación.....	18
1.2.1. Objetivo general .....	18
1.2.2. Objetivos específicos.....	18
1.3. Justificación .....	19
1.4. Presuposición filosófica .....	19
CAPITULO II.....	20
MARCO TEÓRICO.....	20
2.1. Revisión de la literatura.....	20
2.1.1. Investigaciones acerca de la tecnología RFID .....	20
2.1.2. Investigación acerca de desarrollo.....	21
2.1.3. Investigación acerca de control de inventarios.....	22
2.2. Bases teóricas .....	23
2.2.1. Radiofrecuencia.....	23
2.2.2. Tecnologías de Auto-identificación .....	24
2.2.3. Etiquetas RFID .....	25
2.2.4. Lectores RFID.....	26
2.2.5. Microcontrolador.....	27
2.2.6. Arduino.....	29
2.2.7. Base de datos.....	33

2.2.8. Metodología ágil XP .....	35
2.2.9. Lenguaje de programación (comparativas) .....	37
2.2.10. Activos Fijos.....	38
CAPITULO III.....	42
MATERIALES Y MÉTODOS .....	42
3.1. Descripción del lugar de ejecución .....	42
3.2. Metodología de la investigación.....	42
3.2.1. Tipo de investigación.....	42
3.3. Materiales e insumos.....	42
3.3.1. Hardware .....	43
3.3.2. Software.....	43
3.3.3. Arquitectura de solución.....	44
3.3.4. Metodología XP .....	45
3.4. Fase de planificación.....	47
3.4.1. Historias de usuario.....	47
3.5. Fase de diseño .....	55
3.5.1. Diseño simple - arquitectura tecnológica .....	55
3.6. Fase de codificación.....	64
3.6.1. Programación del microcontrolador .....	64
3.6.2. Desarrollo Web NuxtJs.....	65
3.7. Fase de pruebas.....	69
3.7.1. Pruebas de configuración del hardware .....	69
3.7.2. Pruebas de aceptación del sistema web .....	70
CAPITULO IV.....	75
RESULTADOS Y DISCUSIÓN.....	75
4.1. Resultado del objetivo específico 1.....	75
4.1.1. Discusión .....	76
4.2. Resultado del objetivo específico 2.....	77
4.2.1. Discusión .....	77
4.3. Resultado del objetivo específico 3.....	78

4.3.1. Discusión .....	78
CAPITULO V.....	79
CONCLUSIONES Y RECOMENDACIONES.....	79
5.1. Conclusiones .....	79
5.2. Recomendaciones.....	80
REFERENCIAS .....	81
ANEXOS.....	84

## ÍNDICE DE TABLAS

Tabla 1. Cuadro de comparación de tecnología de auto-identificación .....	24
Tabla 2. Cuadro de comparación de etiquetas RFID .....	25
Tabla 3. Cuadro de comparación de lectores RFID .....	26
Tabla 4. Tabla de comparación de microcontroladores .....	28
Tabla 5. Tabla de comparación de microcontroladores .....	34
Tabla 6. Cuadro de comparación de los lenguajes de programación.....	36
Tabla 7. Cuadro de comparación de los lenguajes de programación.....	37
Tabla 8. Cuadro de términos de un inventario fijo .....	39
Tabla 9. Historia de usuario número 1 .....	48
Tabla 10. Historia de usuario número 2.....	48
Tabla 11. Historia de usuario número 3.....	49
Tabla 12. Historia de usuario número 4.....	49
Tabla 13. Historia de usuario número 5.....	50
Tabla 14. Historia de usuario número 6.....	50
Tabla 15. Requerimientos funcionales.....	51
Tabla 16. Requerimientos no funcionales.....	51
Tabla 17. Tarea 1 HDU-01 .....	52
Tabla 18. Tarea 2 HDU-02 .....	52
Tabla 19. Tarea 3 HDU-03 .....	53
Tabla 20. Tarea 4 HDU-04 .....	53

Tabla 21.Tarea 5 HDU-05 .....	54
Tabla 22.Tarea 6 HDU-01 hasta HDU-06.....	54
Tabla 23.Caso de prueba - Correcto funcionamiento de los dispositivos electrónicos integrados con ESP8266 y MFRC522 .....	69
Tabla 24.Caso de prueba - Validar usuarios .....	71
Tabla 25.Caso de prueba - Visualizar reporte de artículos por departamentos y ambientes .	73
Tabla 26.Caso de prueba – Reporte de los departamentos y ambientes.....	74

## ÍNDICE DE FIGURAS

Figura 1. Victimización a empresas, según tipo de hecho delictivo .....	17
Figura 2. Longitud de ondas según el tipo de frecuencia, 2010 - 2017 .....	23
Figura 3. Arquitectura del microcontrolador, arquitectura Harvard. ....	28
Figura 4. Arquitectura RISC, es la arquitectura de la placa Arduino.....	30
Figura 5. Arquitectura RISC, es la arquitectura de la placa Arduino.....	31
Figura 6. Arduino ESP8266.....	32
Figura 7.Arduino RC522.....	33
Figura 8. Funcionalidad de Firebase. ....	35
Figura 9. Metodología XP.....	36
Figura 10. Arquitectura de solución del proyecto. ....	44
Figura 11. Metodología XP reducida. ....	45
Figura 12. Arquitectura Tecnológica del Sistema. ....	55
Figura 13. Diseño de esquema físico de conexiones en ESP8266. ....	56
Figura 14. Vista de captura de datos de Arduino a Firebase. ....	56
Figura 15. Vista de Acceso al sistema. ....	57
Figura 16. Vista para registro de usuario.....	57
Figura 17.Vista para eliminar Usuarios.....	58
Figura 18. Vista principal al sistema. ....	58
Figura 19 .Configuración de Bienes por Ambiente y artículo .....	59
Figura 20.Vista para agregar Bienes. ....	59

Figura 21. Reporte de Artículos con su característica.....	60
Figura 22. Vista para crear artículos.....	60
Figura 23. Configuración de Departamentos.....	61
Figura 24. Vista para agregar Departamentos .....	61
Figura 25. Vista para agregar Departamentos .....	62
Figura 26. Vista para agregar Ambientes .....	62
Figura 27. Configuración de Inventarios.....	63
Figura 28. Configuración para el control de inventarios por departamentos y ambientes. ....	63
Figura 29. Diagrama de flujo de funcionamiento del código del microcontrolador .....	64
Figura 30. Diagrama de flujo de proyecto en Firebase .....	65
Figura 31. Entorno de desarrollo de Firebase. ....	66
Figura 32. Integración de ESP8266 y MFRC522. ....	70
Figura 33. Correcto funcionamiento de la integración.....	70
Figura 34. Habilitando los servicios de autenticación por correo electrónico/contraseña y google. ....	71
Figura 35. Método de acceso del sistema Web.....	72
Figura 36. Registro de nuevo usuario, almacenamiento en tiempo real Firebase. ....	72
Figura 37. Funcionalidad de reporte de inventario.....	73
Figura 38. Funcionalidad de Departamentos y ambientes.....	74
Figura 39. Resultado de encuesta del sistema AFID. ....	75
Figura 41. Esquema general del procedimiento para el control de inventarios.....	76

## ÍNDICE DE ANEXOS

Anexo 1. Especificaciones técnicas de hardware usado.....	84
Anexo 2. Código de Arduino .....	86
Anexo 3. Código VueJS-NuxtJS .....	88
Anexo 4. Formatos .....	101
Anexo 5. MAPIC .....	106

## **SIMBOLOS USADOS**

ESP8266: Abreviatura en ingles módulo Wifi.

WIFI: Abreviatura en ingles Fidelidad inalámbrica.

MFRC522: Abreviatura en inglés Lector de identificación por Radiofrecuencia

TAG-RFID: Abreviatura en inglés de Etiquetas para contacto con peces y vida silvestre.

XP: Abreviatura en inglés de Programación Extrema

HTML: Abreviatura en inglés de Lenguaje de Marcas de Hipertexto

URL: Abreviatura en inglés Localizador Uniforme de Recursos

HTTP: Abreviatura en inglés Protocolo de Transferencia de Hipertexto

JS: Abreviatura de JavaScript

HDU: Historia de usuario

## **RESUMEN**

Las incidencias de pérdidas e inconsistencia de los bienes en la Universidad Peruana Unión filial-Juliaca han tenido un crecimiento notable en los últimos años, en el año 2019 el área de activos fijos tuvo pérdidas en bienes con un valor residual de S/.50.000 a S/.70.000, frente a este problema el objetivo de esta investigación es desarrollar un sistema Web controlado mediante dispositivos (móvil, Tablet) para el control de activos fijos.

Para el desarrollo de esta investigación se usó la metodología XP que cuenta con 4 fases que son: planificación en la cual se hizo el levantamiento de los requerimientos, diseño en el cual se realizó los prototipos, codificación donde se desarrolló cada requerimiento y pruebas donde se realizó las pruebas de aceptación. Se usó herramientas open source y open hardware, como son Arduino, lector MFRC522, módulo ESP8266 y los tag RFID. Vuejs – Nuxtjs para el desarrollo de la aplicación, y Firebase como base de datos, estas herramientas nos permitieron desarrollar un sistema Web de bajo costo.

Se desarrolló la integración de dispositivos al usando una placa Arduino ESP8266 en el cual se enlazó a un host Wifi y a Data Real Time Firebase, usando el MFRC522 que permitió capturar la información de los Tags RFID en la base de datos y visualización de inventarios en el sistema Web. Contribuyendo así a la sociedad para hacer las empresas e instituciones seguras usando las tecnologías de información que en esta investigación permiten la automatización en tiempo real de inventarios.

### **PALABRAS CLAVE**

MFRC522, ESP8266, Vuejs – Nuxtjs, tag RFID

## **ABSTRACT**

The incidence of robbery in institutions has had a remarkable growth in recent years, until 2016, 79.8% of the institutions were affected by lack of monitoring at the institutional level as mentioned by the INEI, facing this problem the The objective of this research is to develop a web system controlled by mobile devices for the control of fixed assets.

For the development of this research, the XP methodology was used, which has 4 phases that are: planning in which the requirements were lifted, design in which the prototypes were made, coding where each requirement is performed and tests where Performed acceptance tests. Open source tools and open hardware were used, such as arduino, MFRC522 reader, ESP8266 module and RFID tags. Vuejs - Nuxtjs for application development, and firebase as a database, these tools allow us to develop a low cost web system.

Verify the integration of devices by using an Arduino ESP8266 board on which it was linked to a Wifi host and to Data Firebase in real time, using the MFRC522 that captured the information of the RFID tags in the database and inventory records in the system Web. Thus contributing to society to make companies and institutions safe using the information technologies that in this research allows real-time inventory automation.

### **KEYWORDS**

MFRC522, ESP8266, Vuejs – Nuxtjs, tag RF

## CAPITULO I.

### EL PROBLEMA

#### 1.1. Descripción de la situación problemática

El área de Activos Fijos es la comisionada de controlar los movimientos físicos y económicos que cuenta la Universidad Peruana Unión Filial Juliaca entre los cuales se encuentran los bienes, departamentos y proyectos que son aprobados por el consejo universitario. Durante 15 años activos fijos ha ido trabajando con un sistema de información propio de la universidad registrando manualmente así el estado y la vida útil de equipos y materiales, sin embargo debido a una incorrecta gestión de bienes e inventarios se ha manifestado las siguientes necesidades:

- Mostrar la información precisa y completa acerca de los bienes que utiliza la Universidad para llevar a cabo los proyectos y programas.
- Monitoreo de bienes en tiempo real, dado que no coinciden con el inventario físico, reflejando así pérdidas de materiales y equipos.



Figura 1. Victimización a empresas, según tipo de hecho delictivo

Fuente: (INEI, 2016)

Según Poch (2000) “ el control del inventario de activos fijos es uno de los aspectos de la administración que en las empresas públicas y privadas es pocas veces atendida, sin tener registrado los responsables de cada uno de los activos, políticas o sistemas que ayuden a esta fácil pero tediosa tarea”.

Por otro lado en el año 2016 el INEI (2017) “ hizo una encuesta nacional de empresas cuyo objetivo principal fue conocer si las empresas, han sido víctimas de algún hecho delictivo ocurrido durante el periodo 2016. Las empresas víctimas de algún hecho delictivo, el robo o hurto es la forma más común de hecho que afecta a las empresas con 79,8%, seguido del intento de robo con el 12,1%, estafa con el 10,1%, extorsión 9,1% y las amenazas con 6,2%”.

## **1.2. Objetivos de la investigación**

### **1.2.1. Objetivo general**

Desarrollar un sistema de Identificación por Radio Frecuencia para el control de inventario de activos fijos en la Universidad Peruana Unión Filial Juliaca.

### **1.2.2. Objetivos específicos**

- Analizar el funcionamiento de los servicios que realiza el área de activos fijos en la Universidad Peruana Unión filial Juliaca.
- Desarrollar una aplicación Web que permita gestionar la información de los activos fijos.
- Realizar las pruebas de comunicación e integración con los sistemas de manejo RFID, logrando recuperar la información almacenada en los Tags-RFID.

### **1.3. Justificación**

El personal del área de activos fijos de la Universidad Peruana Unión filial Juliaca está alarmado por las incidencias de pérdidas desconocidas con los equipos para procesos informáticos generando pérdidas financieras en la institución. El desarrollo de la presente investigación tiene como finalidad proponer la mejora en la gestión de Activos Fijos, desarrollando un sistema de Identificación por Radio Frecuencia en la Universidad Peruana Unión Filial Juliaca, lo cual ayudará a tener un mejor control sobre los equipos, disminuyendo las pérdidas desconocidas por hurto de bienes.

Sachon y Almirall (2016) en su investigación “La tecnología RFID: Usos y oportunidades”, da a conocer la experiencia en la empresa pública Hospital Costa del Sol de España, donde se implementó un sistema RFID para la Identificación de pacientes, mejora y verificación de procesos, localización de instrumental, seguridad, logrando así mejorar en la identificación de los pacientes evitando errores en procesos oncológicos, quirúrgicos y de farmacia, entre otros, mejoras de seguridad para el personal médico, localización en tiempo real de equipamiento médico. Para esto se utilizó tecnología HF y WI FI. Etiquetas RFID pasivas para identificación y activas para localización, terminales de mano RFID con Bluetooth.

### **1.4. Presuposición filosófica**

Con este proyecto aportaría a la mejora de la calidad de vida de las personas que hagan uso de este sistema, para prevenir las incidencias de robo en las instituciones, de esta manera se cumple con los principios de la iglesia, la cual sigue el ejemplo de Cristo, el cual siempre vela por nuestra seguridad e integridad física, también agradecer a Él por la sabiduría que nos brinda para poder aprovechar los recursos tecnológicos para ayudar a la sociedad.

## CAPITULO II.

### MARCO TEÓRICO

#### 2.1. Revisión de la literatura

##### 2.1.1. Investigaciones acerca de la tecnología RFID

En la Pontificia Universidad Católica del Perú, Alejandro (2013), realizó un estudio de las tecnologías de identificación por radio frecuencia titulada "Diseño de un sistema de control de activos para el almacén de electrónica de la Pontificia Universidad Católica del Perú utilizando RFID", con el objetivo de controlar el almacén y agilizar el control de inventarios. Se alcanzó el objetivo cubriendo la necesidad de control de activos y mejorando la gestión de préstamos lo cual agilizó el procedimiento de préstamos y devoluciones.

En la Universidad Nacional Mayor de San Marcos, Quispe (2017), realizó una investigación titulada "Implementación de un sistema RFID para mejorar la productividad de una planta de producción de vidrio templado" con el objetivo de ahorrar el costo por pérdidas, robos y demora en la fabricación de vidrios, que permita disminuir el costo por inventarios y reducir las horas extra del personal. Se llegó al objetivo de mejorar la productividad y reducir los costos operativos por retrasos pérdidas y robos, también se redujo las horas extras por daños y errores causados por la debilidad de los actuales métodos de ubicación de materia prima, mercadería en proceso y producto terminado.

En la Escuela Técnica Superior de ingeniería Industrial de Barcelona, Guizar ( 2017), realizó una investigación titulada "Implementación de RFID en un almacén Logístico" con el objetivo de definir los procesos productivos del almacén logístico para su correcta integración y la implementación de un sistema basado en RFID para controlar los activos fijos del almacén, para lo cual usó una metodología que se divide en dos procesos: Definición de los procesos productivos y alcance de la implementación del RFID. Los resultados en cuanto al uso de RFID

son los siguientes: disminución del tiempo para dar entrada al sistema del almacén, lecturas más rápidas con información precisa en los activos fijos, reducción de roturas de stock.

En la Escuela Politécnica de Chimborazo (Ecuador), Quilligana (2018) , realizó una investigación titulada “Implementación de un prototipo: Sistema Automático de Censado Inalámbrico en RFID para estaciones de control de buses urbanos de Riobamba” con el objetivo de implementar un prototipo basado en RFID para control de buses transmitiendo un reporte de censado de los buses a través de la red celular, para ser almacenados y mostrados en la central de control. Los resultados del prototipo para la implementación específicamente de la comunicación entre el tag y el lector UHF-RFID, se determinó que la distancia de la lectura en el rango de 2 a 4 metros no presenta error de lectura con vehículo detenido y en circulación a velocidades de 20 a 40 km/h. Se concluye que el prototipo es aceptable y se acopla a los requerimientos y a los límites de velocidad vigentes en la ley de tránsito del país.

### **2.1.2. Investigación acerca de desarrollo**

En la Universidad de Ciencias y Humanidades, Sanchez (2017), realizó una investigación titulada “Implementación de un sistema de gestión del activo fijo para el instituto Arzobispo Loayza Los Olivos” con el objetivo de implementar un sistema que permita visualizar la ubicación física y la persona responsable del cuidado de los activos, para lo cual usó la metodología Scrum: Revisión de planes de Release, Distribución, revisión y ajustes de estándares de producto, sprint, revisión de sprint, cierre. Llegó a las siguientes conclusiones: La herramienta Microsoft Visual Studio 2010 con lenguaje de programación en visual Basic, fue de fundamental importancia, ya que se pudo llevar un mejor control de los activos por áreas y locales para el balance con el área de contabilidad, así también hacer inventarios anuales identificando activos faltantes por área además del correcto uso de los equipos asignados a las áreas.

En la Pontificia Universidad Javeriana Cali, Cifuentes (2019), realizó una investigación titulada “Desarrollo de sistema multifuncional para la Pontificia Universidad Javeriana Cali con manilla RFID” con el objetivo de mejorar el control de libros en una biblioteca, control de alimentos en cafeterías y control de usuario a la entrada de la universidad, para ello se realizaron encuestas a usuarios potenciales, estudiantes, profesores y colaboradores pertenecientes a las diferentes facultades de la Universidad Javeriana Cali. Llegó a las siguientes conclusiones: El impacto de las pruebas de funcionamiento del sistema RFID en los usuarios fue positiva en todos los aspectos, entre los más importantes la calificación, preferencia entre método de identificación, realimentación y sugerencias para mejorar el producto.

### **2.1.3. Investigación acerca de control de inventarios**

Saldaña (2016) menciona que “ el control del inventario de activos fijos es uno de los aspectos de la administración que en las empresas públicas y privadas es pocas veces atendida, sin tener registrado los responsables de cada uno de los activos, políticas o sistemas que ayuden a esta fácil pero tediosa tarea”.

González y Salcedo (2017) en su investigación titulada “Sistema de control de inventario de activos fijos en la dirección distrital de salud N°13D06” mencionan que “Aunque con coste mayor la tecnología RFID resuelve problemas inherentes de la tecnología código de barras. Con ello mejoraron las actividades del proceso, en cuanto a tiempos y a un menor requerimiento de fuerza laboral así como: eliminación de los errores en el número de serie, reducción del tiempo dedicado a la captura de los números de serie, aumento de calidad en la preparación de la mercancía despachada, reducción de tiempo en los ciclos de inventariado, documentación de la visibilidad en tiempo real.”

## 2.2. Bases teóricas

### 2.2.1. Radiofrecuencia

Según Ruiz (2017) “La distribución de energía por medio de las ondas electromagnéticas. Este espectro queda determinado por las radiaciones que se emiten. Las ondas, por su parte suponen la propagación de esta radiación, acarreando energía. El espectro electromagnético abarca desde las radiaciones con longitud de onda más pequeña hasta las radiaciones con longitud de onda más amplia.”

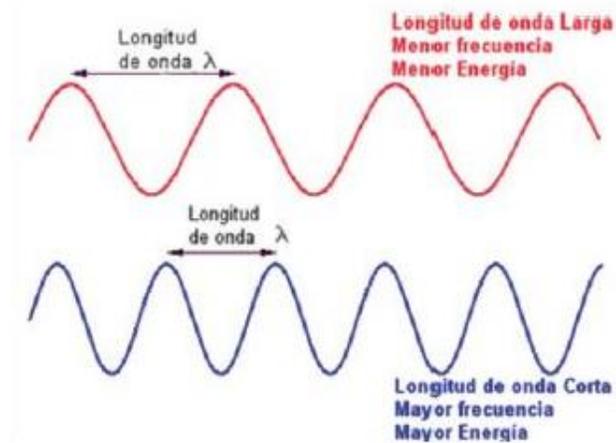


Figura 2. Longitud de ondas según el tipo de frecuencia, 2010 - 2017

Fuente: (Ruiz Sanchez, 2017)

En la figura 2 observamos que la distribución de ondas electromagnéticas tiene dos elementos principales, longitud de onda larga y longitud de onda corta. En la actualidad la tecnología RFID está bajo esta distribución sin embargo, trabajar con distribución de longitud larga mejora la velocidad de identificación entre etiquetas y lectores RFID.

### 2.2.2. Tecnologías de Auto-identificación

Sánchez (2008) refiere: “La auto-identificación es la habilidad de una máquina para leer una identidad que se aplica normalmente cuando es un objeto o persona. Dentro del ámbito de la tecnología de identificación, aplicando al control de acceso, se pueden encontrar diversas tecnologías como: sistemas biométricos, tarjetas magnéticas, código de barras, tecnología RFID y memorias de contacto.”

*Por su parte Cáceres (2018) “ La tecnología de almacenamiento y recuperación remota aborda principalmente al uso de etiquetas o también llamados tags que poseen un chip en las que reside la información y genera la comunicación en distancia por medio de ondas electromagnéticas entre 2 hasta 100 metros aproximadamente. RFID se basa en un concepto similar, la diferencia es que usa señales de radiofrecuencia para transmitir los datos.”*

Tabla 1.

*Cuadro de comparación de tecnología de auto-identificación*

TECNOLOGÍAS	CARACTERÍSTICAS
Sistemas biométricos	El sistema biométrico, trabaja con accesos, por medio de características físicas.
Tarjetas magnéticas	Utiliza señales electromagnéticas para registrar y codificar información en una banda que puede ser leída por una máquina.
Código de barras	Realiza codificación de imágenes en forma de barras y espacios.
Tecnología RFID	Captura la información sin necesidad de tener un contacto visual entre el dispositivo lector y las etiquetas.

Fuente: Elaboración propia. Según tecnologías de auto-identificación.

Para este proyecto se hará uso de la tecnología RFID, por la captura datos es más precisa la recopilación y suficiente para abarcar el proyecto.

### 2.2.3. Etiquetas RFID

(Sepúlveda, 2018) En su investigación menciona que “es un chip embebido en una antena a modo de bobinado laminado también llamada tags (transmisor y receptor), y dependiendo de su aplicación o el ambiente donde van a ser instalados o el tipo de material que se van a adherir pueden variar en la composición de su material externo. La etiqueta hace función de transmisión y recepción por lo que también se le da el nombre de transponder. Actualmente existe una extensa variedad de tags en el mercado, y el costo varía con el grado de complejidad en la fabricación y característica que difiere. Los tags se pueden dividir en dos grandes grupos”.

Tabla 2

*Cuadro de comparación de etiquetas RFID*

HERRAMIENTAS	CARACTERÍSTICAS
Tags activos	Son más eficientes en medios ruidosos, además son más eficaces logrando alcanzar mayores distancias. La vida útil de la batería es de unos 10 años. Sin embargo, el costo de estos tags es 10 y 20 veces el valor de un tag pasivo.

---

Tags pasivos	Estas etiquetas pueden llegar a ser muy económicas y pequeñas, pero su rango de lectura es muy limitado. Sólo funcionan cuando reciben potencia desde un emisor de radiación.
--------------	---

---

Fuente: Elaboración propia. Según Tags activos & Tags pasivos.

Para esta investigación se hará uso de tags pasivos, ya que los tags activas RFID tienen un elevado costo en el mercado.

#### 2.2.4. Lectores RFID

(Ruiz, 2013) “Los lectores RFID tienen la función de alimentar los tags y etiquetas a través de las antenas por medio de la emisión de una señal de radio frecuencia, y al mismo tiempo capturan los datos que envían los tags para luego ser decodificados e interpretados por el software correspondiente”.

(Jung, 2011) “Los lectores fijos necesitan de las antenas para poder generar la onda que llega a los tags, de la misma forma la onda de respuesta es leída a través de estas antenas. Los lectores portátiles brindan de forma manual una captura de datos de una sola vez y con gran rapidez comparados con un tradicional código de barras”.

Tabla 3

*Cuadro de comparación de lectores RFID*

HERRAMIENTAS	CARACTERÍSTICAS
Lectores Fijos	Requiere de antenas para poder generar ondas de frecuencia que llega a los Tags, de la misma manera la onda de frecuencia de respuesta es leída a través de estas antenas.

---

Lectores portátiles	Brindan de forma manual una captura de datos con gran rapidez, se basa en la lectura a partir de un láser.
Lectores de mesa USB	Diseñados para aplicaciones más sencillas el cual no se necesita grandes lecturas de gran alcance.
Lectores de carretilla	Obtiene automáticamente la información de los ítems controlando en todo momento el stock del almacén y su localización.

---

Fuente: Elaboración propia. Según lectores RFID.

Para este proyecto se hará uso de los lectores de mesa USB, ya que el costo de otros lectores es más elevado.

### **2.2.5. Microcontrolador**

Henao y Cardona (2009) “Un microcontrolador es un circuito integrado que está diseñado para distintas funciones, los microcontroladores están formados por al menos tres elementos: procesador, memoria y dispositivos de entrada y salida, son también considerados como ordenadores de menor instrucción, son económicos y sirven para implementar proyectos pequeños o grandes, cada microcontrolador se diseña de acuerdo a los usos que se le va a dar por lo cual están en un punto entre un simple dispositivo y un potente ordenador. Anteriormente los microcontroladores podían ser usados como un dispositivo simple, ideal para usar en grandes y pequeñas empresas para experimentar con pequeños proyectos, al pasar los años estos han ido evolucionando, ahora en el mercado se puede encontrar todo

tipo de microcontroladores en las placas pequeñas, todos estos diseñados de distinta forma y con distintas funcionalidades, dentro de estas placas tenemos Arduino, Raspberry Pi, etc”.

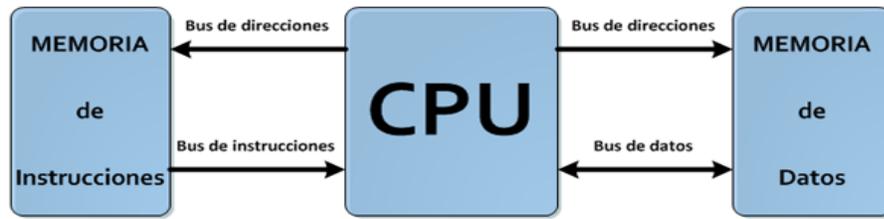


Figura 3. Arquitectura del microcontrolador, arquitectura Harvard.

Fuente: (SHERIN, 2018)

En la Figura 3 estimamos que la arquitectura Harvard posee tres elementos principales, memoria de programa, CPU y memoria de datos. En la actualidad la mayoría de los microcontroladores están bajo esta arquitectura porque tiene la ventaja de trabajar con dos memorias simultáneamente, esto aumenta la rapidez de ejecución de los programas que se cargan en los microcontroladores.

Tabla 4

Tabla de comparación de microcontroladores

HERRAMIENTAS	CARACTERÍSTICAS
Arduino Uno	Permite la captura de datos en tiempo real, voltaje de entrada 7-12v, 14 pines digitales, 32k de memoria flash.
Raspberry Pi 3 Modelo B	Socket ethernet, salida de video, 27 pines digitales, LAN inalámbrica, pantalla de visualización.

Fuente: Elaboración propia. Según microcontroladores.

Para este proyecto se usará la placa Arduino Uno, para la captura de información en tiempo real, ya que es económico.

### 2.2.6. Arduino

Vargas (2015) “Los Arduinos se codifican con el lenguaje C o C++ en el entorno del desarrollo integrado (IDE) de Arduino. Además, existen paquetes de programación alternativos, por ejemplo, el CodeBlocks for Arduino. También se puede programar de manera gráfica usando Ardublocks o el paquete QP Framework diseñado para programar máquinas de estado complejas a partir de los SateChart. La utilización de software para el procesamiento numérico o computación científica, tal como Scilab 5.5, es posible gracias al lenguaje de programación gráfico de Xcos y una biblioteca desarrollada por Bruno Jofret. Cabe destacar que los paquetes mencionados y sus bibliotecas son gratuitos y de uso público”.

Murillo (2015) “Arduino es un proyecto de Código Abierto (Open Source) que posee una plataforma de hardware y un IDE (Integrated Development Enviroment). Esta plataforma de hardware está integrada por una variedad de tarjetas programables, de las cuales la más básica y accesible es Arduino Uno, la cual no solo es la tarjeta más accesible desde el punto de vista económico, también es la más usada para fines diversos, edades e intereses”.

Vega (2014) hacen mención que “El Arduino, es un elemento de fácil conectividad a una red y adicionalmente permite implementar un servidor de protocolos de alto nivel, como el Hypertext Transfer Protocol (HTTP) tiene memoria, capacidad de procesamiento autónomo, compiladores de lenguajes de programación como C y puertos físicos para interconectar con dispositivos. El Arduino Mega 2560, es una placa electrónica basada en el ATmega2560 (hoja de datos). Tiene 54 pines digitales de entrada/salida, 16 entradas analógicas, cuatro UARTs (puertos seriales), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de reinicio. La Mega es compatible con la mayoría de los protectores diseñados para el Arduino Diecimila. La interfaz del Arduino y la tarjeta de conexión Ethernet Shield que se requieren para el desarrollo del prototipo de gestión domótica, se vinculan a través de los puertos de conexión o pines los cuales se ensamblan de manera

precisa. Es así como se muestra la unión que se efectuó para el desarrollo del prototipo presentado en este artículo”.

### **Arquitectura RISC de Arduino**

Morales (2015) define que “La arquitectura RISC reduce el tiempo de ejecución simplificando las instrucciones que se dan a una computadora, su CPU tiene instrucciones sencillas, su operación es sencilla, y son procesados a una alta velocidad.”

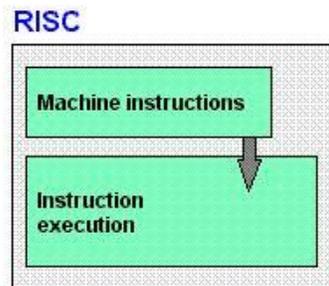


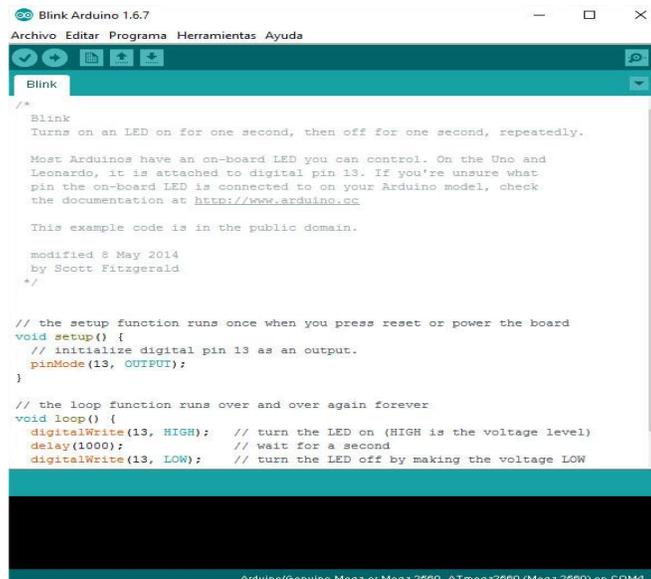
Figura 4. Arquitectura RISC, es la arquitectura de la placa Arduino.

Fuente: (4BP, n.d.)

### **IDE Arduino**

“Arduino un programa hecho para microcontroladores de una sola placa que fue desarrollado en 2005 y se desarrolló como un proyecto estudiantil con un costo de solo. Arduino se utiliza para hacer que la creación de aplicaciones interactivas y otros programas sea mucho más accesible para los usuarios. Arduino permite que los usuarios carguen un programa a un microcontrolador sin la necesidad de un chip u otro dispositivo. Arduino incluye una gama de herramientas y funciones y viene equipado en una plataforma de código abierto con placa que se basó en microcontrolador de 8 bit Atmel AVR. La placa de hardware de Arduino puede venir pre ensamblada o puede ser ensamblada manualmente por el usuario. La última opción es la opción más económica y también es una opción popular para los usuarios que les gusta jugar con la electrónica y les gusta construir cosas ellos mismos. El

programa hazlo-tu-mismo viene con información comprensiva la cual puede guiar al usuario a través del proceso de creación y ayudarlos si se traban en algún punto.” (Arduino, 2016)

A screenshot of the Arduino IDE interface. The title bar reads "Blink Arduino 1.6.7". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar contains icons for file operations and execution. The main text area displays the "Blink" program code, which includes a multi-line comment explaining the program's purpose and a C++ code block with setup and loop functions. The status bar at the bottom indicates the board and port: "Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM4".

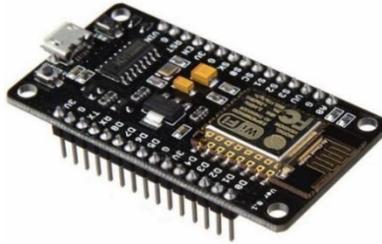
```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. On the Uno and
 * Leonardo, it is attached to digital pin 13. If you're unsure what
 * pin the on-board LED is connected to on your Arduino model, check
 * the documentation at http://www.arduino.cc
 *
 * This example code is in the public domain.
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 */
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
}
```

Figura 5. Arquitectura RISC, es la arquitectura de la placa Arduino.

Fuente: (4BP, n.d.)

“Cuando varios estudiantes del Instituto de Diseño Interactivo de Ivrea, Italia, con el fin de facilitar el acceso y uso de la electrónico y programación. Lo hicieron para que los estudiantes de electrónica tuviesen una alternativa más económica a las populares BASIC Stamp, unas placas que por aquel entonces valían más de cien dólares, y que no todos se podían permitir. El resultado fue Arduino, una placa con todos los elementos necesarios para conectar periféricos a las entradas y salidas de un microcontrolador, y que puede ser programada tanto en Windows como macOS y GNU/Linux. Un proyecto que promueve la filosofía learning by doing, que viene a querer decir que la mejor manera de aprender es cacharreando.” (Xataca, 2018)

En esta investigación se usará la placa node MCU, como mencionan los anteriores investigadores la comunidad Arduino es de open hardware y open software esto nos brinda la seguridad de que atravesemos por una importación legal.



*Figura 6. Arduino ESP8266*

Fuente: Elaboración Propia, 2019.

### **Arduino RC522**

“El Módulo RC522 es Lector-Grabador RFID, posee comunicación SPI permitiéndole trabajar fácilmente con la mayoría de microcontroladores. Utiliza un sistema de modulación y demodulación para todo tipo de dispositivos pasivos de 13.56MHz. El dispositivo maneja el ISO14443A y soporta el algoritmo de encriptación Quick CRYPTO1 y MIFARE. El rango de detección de tags RFID es de aprox. 5-7cm.” (RC522, 2018)

Baltzar(2016) menciona que “ RC522 utiliza un sistema avanzado para la modulación y demodulación en todo tipo de dispositivos pasivos de 13.56Mhz. Como se hará una lectura y escritura de la tarjeta, es necesario conocer las características de los bloques de memoria una tarjeta: La tarjeta que viene con el módulo RFID cuenta con 64 bloques de memoria (0-63) donde se hace lectura y/o escritura. Cada bloque de memoria tiene la capacidad de almacenar hasta 16 Bytes. El número de serie consiste de 5 valores hexadecimales, se podría utilizar esto para hacer una operación dependiendo del número de serie.”

En anteriores investigaciones mencionadas vemos que RFID RC522 nos permite un acceso de lectura y escritura de las tarjetas, mediante la conexión a Node MCU en el cual podemos realizar pruebas de reconocimiento con las tarjetas o tags Rfid.



Figura 7.Arduino RC522.

Fuente: Propia, 2019.

### 2.2.7. Base de datos

Una base de datos una colección organizada de información. Generalmente, cuando nos referimos a una base de datos, significa uno que indexa artículos en revistas. En informática la base de datos es un sistema que está formado por una cantidad de información y a la vez está esta almacenada en discos, a los cuales se puede acceder directamente, estos son manipulados por un software que los administra. Las bases de datos se conforman por tablas o entidades que tiene sus respectivos atributos y sus valores que se distribuyen en filas y columnas respectivamente.

“Colección o depósito de datos integrados, almacenados en soporte secundario (no volátil) y con redundancia controlada los datos que han de ser compartidos por diferentes usuarios y aplicaciones deben mantenerse libres de ellos y su definición (estructura de la base de datos)

única y almacenada junto con los datos el cual ha de permitir captar las interrelaciones y restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinado facilitaran la seguridad del conjunto de los datos.” (Cáceres, 2016)

Tabla 5

*Tabla de comparación de microcontroladores*

HERRAMIENTAS	CARACTERÍSTICAS
Base de datos relacional	Se usa una relación de identificadores, son más robustos y menos vulnerables.
Base de datos no relacional	No hay necesidad de un identificador, la información se organiza en documentos, son más flexibles, manejan los datos en tiempo real.

Fuente: Elaboración Propia.

### **Firestore**

Garcia (2017) menciona que “ Firestore proporciona una solución eficaz frente no solo a problemas de desarrollo, sino también de escalabilidad a medida que la base de usuarios de la aplicación crece, ya que los servidores son proporcionados por Google. Entre sus funcionalidades se encuentra un servicio de autenticación, base de datos en tiempo real, almacenamiento de archivos, solución de errores, funciones backend, testeo, y medida de estadísticas recogidas de los usuarios. Aunque al realizar MyBets no se ha hecho uso de todas estas funciones, sí se ha usado una gran parte. A continuación se explica el papel que ha tenido cada una de ellas en la aplicación.”



Figura 8. Funcionalidad de Firebase.

Fuente: (Cyron, 2018)

Garcia (2017) “Firebase proporciona una base de datos noSQL que almacena datos y los sincroniza en tiempo real. Esto ha sido utilizado para almacenar todos los datos de las apuestas realizadas, los resultados, datos de los usuarios como los puntos de los que disponen, y en general, cualquier dato de la aplicación que necesite ser guardado.”

### 2.2.8. Metodología ágil XP

Garcia (2017) Para el desarrollo de software “la metodología de programación extrema es una metodología ágil y flexible utilizada para la gestión de proyectos. La programación extrema se enfoca en mejorar las relaciones interpersonales del equipo de desarrollo como clave para el éxito a través del trabajo en equipo, el aprendizaje continuo y el buen clima de trabajo. Esta metodología pone énfasis en la retroalimentación continua entre el cliente y el equipo de desarrollo y es ideal para proyectos con requisitos imprecisos y altamente cambiantes.”

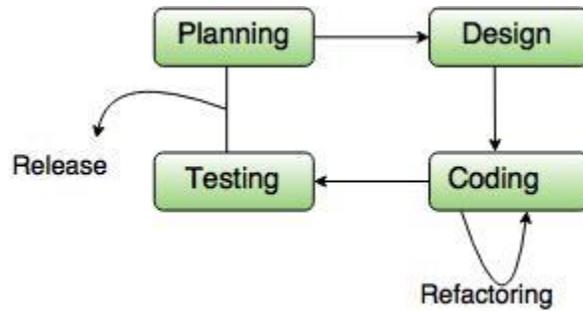


Figura 9. Metodología XP.

Fuente: (Ramírez, 2017)

Está conformada por cuatro pasos, dentro de cada paso consta de actividades, los cuatro pasos de la metodología XP son los siguientes:

- Planificación.
- Diseño.
- Codificación.
- Pruebas.

Tabla 6

Cuadro de comparación de los lenguajes de programación

Metodología SCRUM	Metodología XP
Individuos e iteraciones sobre procesos y herramientas.	Las tareas se toman en un orden de prioridad estricta.
Software de trabajo sobre documentación completa.	Los desarrolladores pueden modificar o re factorizar partes del código cuando sea necesario.
Colaboración del cliente sobre negociación de contrato.	El software debe validarse en todo momento, en la medida en que las pruebas se escriban antes del software.

---

Responde al cambio sobre el siguiente plan.	Siempre y cuando el equipo no haya comenzado a trabajar en una característica particular, puede intercambiarse en la interacción a cambio de una característica no
---	--

---

### 2.2.9. Lenguaje de programación (comparativas)

Es el idioma artificial de las computadoras, se usa para la creación de software y estos pueden controlar la parte física y lógica de una computadora o una máquina, para dar instrucciones de pasos que debe seguir para poder comprenderse con el ser humano. Está formado por un conjunto de expresiones, símbolos y reglas sintácticas, estas dan sentido y significado a su estructura.

Tabla 7

*Cuadro de comparación de los lenguajes de programación*

HERRAMIENTAS	CARACTERÍSTICAS
JAVA	Orientado a objetos, diseñado para tener pocas dependencias de implementación como fuera posible.
PYTHON	Es un lenguaje de programación multiparadigma, soporta orientación a objetos, programación imperativa y programación funcional.

---

PHP	Originalmente diseñado para el desarrollo web de contenido dinámico. Puede ser usada en aplicaciones gráficas independiente.
C	Orientado a la implementación de sistemas operativos, apreciado por la eficiencia de código que produce.
JavaScript	Maneja paradigmas de programación, como programación orientada a objetos, funcional e imperativa, menos programación procedimental.

---

Fuente: Elaboración propia. Según Java, Python, Php, C & JavaScript.

Las ventajas más resaltantes de JavaScript son: soporta todos los navegadores, es multiplataforma y es un lenguaje sencillo de aprender.

### **2.2.10. Activos Fijos**

“Son aquellos bienes tangibles que se utilizan en la actividad de la empresa, que tengan una vida útil superior a un año y que no estén destinados a la venta, tales como: Terrenos, edificios, muebles, vehículos, maquinaria, herramienta y equipos de computación. Los bienes de uso (activos fijos) a diferencia de los activos corrientes, no entran en la rotación comercial o industrial del negocio y por su naturaleza constituyen inversiones de carácter permanente que se encuentran al servicio de la empresa.” (Real, 2014)

### ***Términos en un Inventario de Activos***

Según el reglamento Interno para la Administración y Control de Activos Fijos del Ministerio de Salud Pública (2009) en su Art. 1 que los términos utilizados en un inventario de activo fijo, deben interpretarse de acuerdo a las siguientes definiciones:

Tabla 8

*Cuadro de términos de un inventario fijo*

TÉRMINOS	CONCEPTOS
Baja de activos	Eliminación de los activos de los registros contables y administrativos por motivo de desecho, donación, venta, permuta o sustracción, siempre que se sigan los procedimientos establecidos en la ley y este reglamento.
Bienes	Son artículos materiales o inmateriales de propiedad de la institución, que requiere para el desarrollo de sus actividades y cumplimiento de sus fines.
Bien inservible	Que no está en condiciones de servir, muy estropeado.
Bien obsoleto	Que el bien es anticuado, inadecuado a las circunstancias actuales

---

Bien en mal estado	Aquellos que en virtud de su estado físico no tengan utilidad alguna, con lo que sería objeto de destrucción y para tal efecto se levantará el acta respectiva.
Comodato	Es el contrato por el cual uno del contratante llamado comodante, se obliga a prestar gratuitamente el uso de una cosa no fungible, pero no los frutos de ella; y el otro, llamado comodatario, se obliga a restituirla íntegramente a su vencimiento.
Constatación física de activos fijos	Comprobación de la existencia de los bienes de propiedad de la entidad, tomando en cuenta la actualización y ubicación de los bienes.
Egreso de bienes	Es la salida de bienes de la entidad, mediante la enajenación a través de cualquiera de las modalidades: remate al martillo, remate en sobre cerrado y venta directa, también cuando se produce la baja o transferencia gratuita de los mismos.

---

---

Hurto

Es el acto de apoderarse ilegítimamente de cosa mueble ajena, substrayéndosela a quien la retiene, sin violencia o intimidación en las personas, ni fuerza en las cosas. El hecho de llevarse a cabo sin violencia o intimidación en las personas, ni fuerza en las cosas, se diferencia radicalmente del delito de robo.

Robo

Es el acto de apoderarse ilegítimamente de cosa mueble ajena, substrayéndosela a quien la retiene con intimidación en las personas y fuerza en las cosas.

---

## CAPITULO III.

### MATERIALES Y MÉTODOS

#### 3.1. Descripción del lugar de ejecución

El desarrollo de esta investigación se sitúa en el departamento de Activos Fijos de la Universidad Peruana Unión- Filial Juliaca ubicada en la Salida Arequipa Km 6, la cual cuenta con instalaciones eléctricas, equipos para procesos informáticos, y más objetos de valor.

#### 3.2. Metodología de la investigación

##### 3.2.1. Tipo de investigación

Tipo de investigación propositiva y tecnológica. La investigación propositiva se sustenta con la siguiente fórmula:

$$\textit{Teoría} + \textit{Hecho} + \textit{Solución} = \textit{Investigación Propositiva}$$

Según Huamaní ( n.d.) “La investigación tecnológica resalta la presencia de un estado cognitivo propio de la ingeniería donde se destaca una potencial actitud innovadora de los involucrados en el proceso de la investigación. El diseñador que reúne elementos en combinaciones nuevas, es capaz de integrar y de manipular en su mente dispositivos, sistemas y aparatos que todavía no existen.”

Según Roget (2018) “La investigación propositiva se fundamenta en una necesidad o vacío. Cuando se tiene la información descrita, se realiza una propuesta para contrarrestar la problemática actual y las deficiencias halladas.”

#### 3.3. Materiales e insumos

A continuación, se mencionan los principales materiales que se usaron para esta investigación.

### **3.3.1. Hardware**

- Etiquetas RFID
- Lector RFID portátil
- Sensor NodeMCU
- Servicio de Internet

Las especificaciones de estos se pueden apreciar en el Anexo 1

### **3.3.2. Software**

- IDE Arduino versión 1.8.8
- IDE Visual Studio Code versión 1.33.1
- Librería ESP8266
- Librería Arduino Json

### 3.3.3. Arquitectura de solución

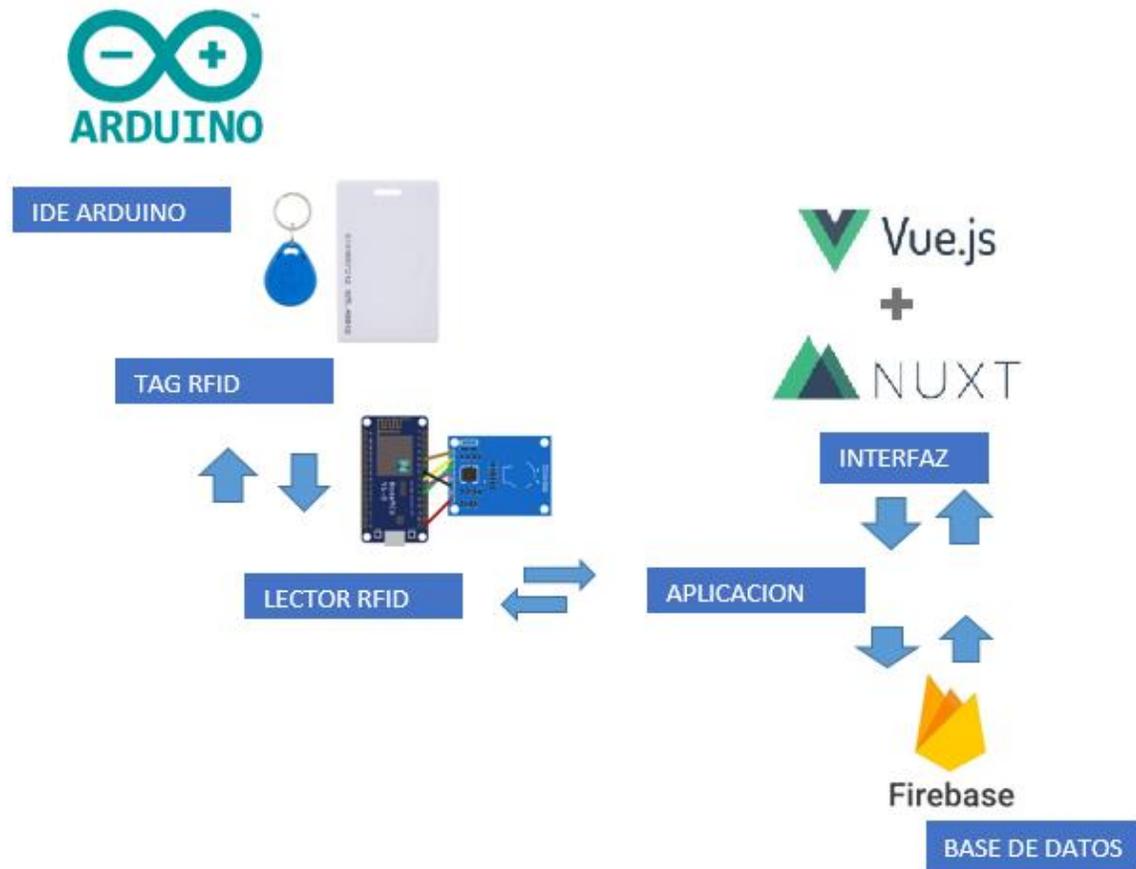


Figura 10. Arquitectura de solución del proyecto.

Fuente: Elaboración propia, 2019.

En la *figura 10* representa a la arquitectura de solución del proyecto, se muestra que la solución al control de activos da inicio en el dispositivo Tag-RFID que estará enlazado con la información de un activo, el lector RFID-RC522 que detecta el Tag-RFID en un radio de 5cm y el módulo Wifi ESP8266 que estará enlazado a un host y sincronizado con Firebase para la visualización de los datos en tiempo real , luego la aplicación Web que notifica la detección del Tag-RFID y emite una alarma, el usuario tendrá un control de los activos visualizando en tiempo real los equipos que el lector RFID-RC522 detecte.

### 3.3.4. Metodología XP

En esta parte el investigador hace uso de una metodología ágil, hace uso de las actividades necesarias para cada fase de la metodología, esta personalización se hizo de acuerdo a las necesidades para el desarrollo de la solución.

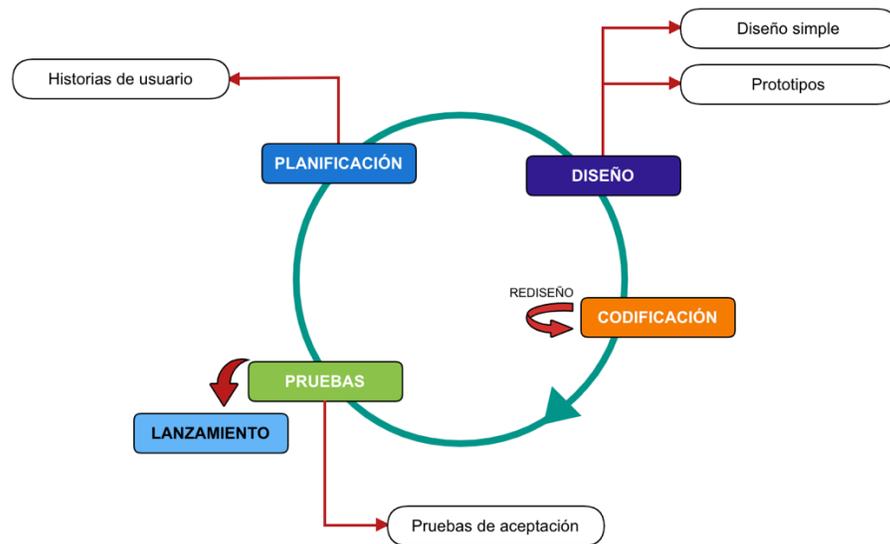


Figura 11. Metodología XP reducida.

Fuente: Ramírez, 2017.

### ***Planificación***

En esta fase se define las historias de usuario con los clientes, estos constan de 2 o 3 líneas escritas por el cliente en un lenguaje que no es técnico, estas mismas se utiliza para la fase de las pruebas, para corroborar si el sistema cumple con lo que especifica la historia de usuario.

### ***Diseño***

En esta fase se define un diseño simple y sencillo del sistema, como por ejemplo la arquitectura tecnológica que se usará para el desarrollo, esto debe ser entendible y fácil de implementar para que demande menos tiempo y esfuerzo para desarrollar. Después de esto se diseña un prototipo del sistema, tanto la parte del hardware y software.

### ***Codificación***

En esta fase a la hora de la codificación de una historia de usuario, que es un requerimiento, la presencia del cliente es muy importante. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente que hará y que tendrá que contemplar. Para que a la hora de realizar las pruebas se compruebe si cumple la funcionalidad especificada.

### ***Pruebas***

En esta fase se hace la prueba de funcionalidad por cada historia de usuario, estas pruebas de funcionalidad se valida con la presencia de los clientes quienes definen si la prueba fue satisfactoria o deficiente.

### **3.4. Fase de planificación**

En esta etapa el investigador desarrolló un formato para el levantamiento de las historias de usuario, lo cual en detalle es explicado en el Anexo 4, esto permitió al investigador recopilar los requerimientos funcionales para el sistema y posteriormente la asignación de las tareas.

#### **3.4.1. Historias de usuario**

Mediante la técnica de entrevista realizada al jefe del área de Activos Fijos, el investigador recopiló las siguientes historias de usuario:

- HDU-01: Quiero un sistema que me permita tener el control de mis activos fijos por departamentos y ambientes.
- HDU-02: Quiero ordenar los activos por bienes y artículos.
- HDU-03: Nuestros equipos están registrados en un Excel y no hay un orden en control inventarios.
- HDU-04: Quisiera un sistema solo para las personas autorizadas por el área.
- HDU-05: Quisiera ver un reporte de los inventarios por departamentos, ambientes que tengan equipos completos, incompletos y sobrantes.
- HDU-06: Quiero realizar inventarios de manera ágil y rápida, con la ayuda de un sensor para identificar cada activo, sin la necesidad de buscar el código CAF que se le asigna a cada equipo.

## ***Descripción de Historias de usuario para la aplicación Web***

Tabla 9

### *Historia de usuario número 1*

---

Historia de Usuario	
Número: <b>HDU-01</b>	Usuario: Jefe de área
Nombre Historia: Quiero un sistema que me permita tener el control de mis activos fijos por departamentos y ambientes.	
Prioridad: Alta	Puntos Estimados: 2
Riesgo: Bajo	Iteración Asignada: 1
Programador Responsable: Daniel León	
Descripción: Configuración de departamentos y ambientes.	

---

Fuente: Elaboración propia, 2019.

Tabla 10

### *Historia de usuario número 2*

---

Historia de Usuario	
Número: <b>HDU-02</b>	Usuario: Jefe de área
Nombre Historia: Quiero ordenar mis activos por bienes y artículos	
Prioridad: Alta	Puntos Estimados: 1
Riesgo: Bajo	Iteración Asignada: 1
Programador Responsable: Daniel León	
Descripción: Asignación de activos por conjunto de bienes y el detallado de sus artículos en cada ambiente y departamento.	

---

Fuente: Elaboración propia, 2019.

Tabla 11

*Historia de usuario número 3*

---

Historia de Usuario

---

Número: **HDU-03** Usuario: Jefe de área

Nombre Historia: Nuestros equipos están registrados en un Excel y no hay un orden en control inventarios.

Prioridad: Media

Puntos Estimados: 1

Riesgo: Bajo

Iteración Asignada: 1

Programador Responsable: Daniel León

Descripción: Desarrollo del módulo de Inventario de activos del campus para el control por departamentos y ambientes.

---

Fuente: Elaboración propia, 2019.

Tabla 12

*Historia de usuario número 4*

---

Historia de Usuario

---

Número: **HDU-04** Usuario: Jefe de área

Nombre Historia: Quisiera un sistema solo para las personas autorizadas por el área.

Prioridad: Alta

Puntos Estimados: 2

Riesgo en Desarrollo: Medio

Iteración Asignada: 1

---

Programador Responsable: Daniel León

Descripción: Mediante un sistema se debe controlar los roles de cada personal.

---

Fuente: Elaboración propia, 2019.

Tabla 13

*Historia de usuario número 5*

---

Historia de Usuario

---

Número: **HDU-05** Usuario: Jefe de área

Nombre Historia: Quisiera ver un reporte de los inventarios por Departamento ambientes que tengan equipos completos, incompletos y sobrantes.

Prioridad: Alta

Puntos Estimados: 3

Riesgo en Desarrollo: Alto

Iteración Asignada: 1

Programador Responsable: Daniel León

Descripción: Realización de reportes de inventario por estados de cada departamento y ambiente.

---

Fuente: Elaboración propia, 2019.

Tabla 14

*Historia de usuario número 6*

---

Historia de Usuario

---

Número: **HDU-06** Usuario: Jefe de área

Nombre Historia: Quiero realizar inventarios de manera ágil y rápida con la ayuda de un sensor para identificar cada activo, sin la necesidad de buscar el código CAF que se le asigna a cada equipo.

Prioridad: Alta

Puntos Estimados: 3

Riesgo en Desarrollo: Alto

Iteración Asignada: 1

Programador Responsable: Daniel León

Descripción: Mediante dos placas Arduino ESP8266 y MFRC522 se deberá controlar los inventarios, capturando la información del activo en un TAG RFID y almacenándola en Firebase.

---

Fuente: Elaboración propia, 2019.

### ***Requerimientos para el sistema RFID***

Tabla 15

#### *Requerimientos funcionales*

<b>CÓDIGO</b>	<b>REQUERIMIENTOS FUNCIONALES</b>	<b>ASIGNADO</b>
<b><i>REQF-001</i></b>	Validación de usuarios.	Daniel León
<b><i>REQF-002</i></b>	Control de activos fijos por departamentos y ambientes.	Daniel León
<b><i>REQF-003</i></b>	Control inventarios por bienes y artículos.	Daniel León

---

Fuente: Elaboración propia, 2019.

Tabla 16

#### *Requerimientos no funcionales*

<b>CÓDIGO</b>	<b>REQUERIMIENTOS NO FUNCIONALES</b>	<b>ASIGNADO</b>
<b><i>REQNF-001</i></b>	Vistas con funcionalidad	Daniel León
<b><i>REQNF-002</i></b>	Autenticación por usuario/email, contraseña y cuentas de google.	Daniel León

---

Fuente: Elaboración propia, 2019.

**Tareas asignadas para el sistema RFID**

Tabla 17

*Tarea 1 HDU-01*

---

Tarea	
Número: <b>1</b>	Numero de Historia: HDU-01
Nombre: Desarrollo del módulo departamentos y ambientes.	
Tipo de tarea: Configuración	Puntos Estimados: 1
Fecha Inicio: 01 de Octubre de 2019	Fecha fin: 11 de Octubre de 2019
Programador Responsable: Daniel León	
Descripción: Se hace el desarrollo de los módulos configurables departamento y ambientes.	

---

Fuente: Elaboración propia, 2019.

Tabla 18

*Tarea 2 HDU-02*

---

Tarea	
Número: <b>2</b>	Numero de Historia: HDU-02
Nombre: Desarrollo de los módulo artículos y bienes	
Tipo de tarea: Instalación	Puntos Estimados: 1
Fecha Inicio: 14 de Octubre de 2019	Fecha fin: 15 de Octubre de 2019
Programador Responsable: Daniel León	
Descripción: Se hace el desarrollo de los módulos artículos y bienes.	

---

Fuente: Elaboración propia, 2019.

Tabla 19

*Tarea 3 HDU-03*

---

Tarea

---

Número: **3**                  Numero de Historia: HDU-03

Nombre: Llenar nuestro Firebase de la información del Excel de los activos fijos.

---

Tipo de tarea: Desarrollo                  Puntos Estimados: 2

Fecha Inicio: 14 de octubre de 2019          Fecha fin: 18 de octubre de 2019

Programador Responsable: Daniel León

Descripción: Mediante un archivo Excel se hará la inserción de activos en un Firebase

---

Fuente: Elaboración propia, 2019.

Tabla 20

*Tarea 4 HDU-04*

---

Tarea

---

Número: **4**                  Numero de Historia: HDU-04

Nombre: Acceso a aplicación Web

Tipo de tarea: Desarrollo                  Puntos Estimados: 2

Fecha Inicio: 04 de marzo 2019                  Fecha fin: 13 de marzo de 2019

Programador Responsable: Daniel León

Descripción: Vista de acceso al aplicación Web, autenticación mediante usuario/email, contraseña y cuentas de google.

---

Fuente: Elaboración propia, 2019.

Tabla 21

*Tarea 5 HDU-05*

---

Tarea	
Número: <b>5</b>	Numero de Historia: HDU-05
Nombre: Visualización de datos en tiempo real.	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 21 de octubre de 2019	Fecha fin: 25 de octubre 2019
Programador Responsable: Daniel León	
Descripción: Reporte de activos fijos por departamento y ambiente que tendrán un estado de equipos completos, incompletos y sobrantes.	

---

Fuente: Elaboración propia, 2019.

Tabla 22

*Tarea 6 HDU-01 hasta HDU-06*

---

Tarea	
Número: <b>6</b>	Numero de Historia: HDU-01 hasta HDU-06
Nombre: Integración del sistema	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 28 de octubre de 2019	Fecha fin: 31 de octubre 2019
Programador Responsable: Daniel León	
Descripción: En esta tarea se hace la integración tanto del hardware (Arduino) y software (aplicación Web), para realizar el control de inventarios.	

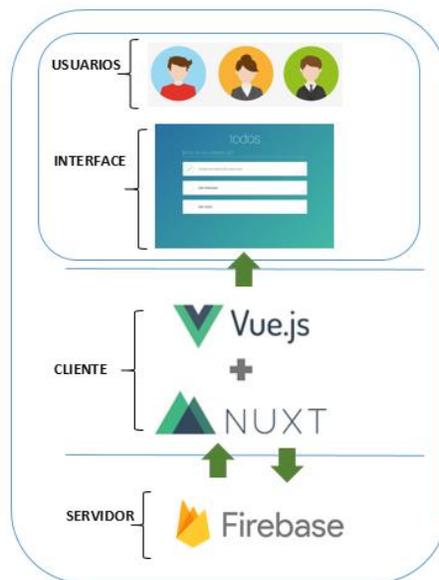
---

Fuente: Elaboración propia, 2019.

### 3.5. Fase de diseño

#### 3.5.1. Diseño simple - arquitectura tecnológica

En la *Figura 12*, se muestra arquitectura de desarrollo es de tipo Cliente-Servidor. Codificado en Vuejs con el Framework Nuxt js y con lenguaje de programación JavaScript de la aplicación, juntamente con documentos HTML como estilo para la interface(vista) de usuario. mediante una Authentication recuperaremos para el consumo y almacenamiento de datos utilizaremos el servidor de Firebase.



*Figura 12.* Arquitectura Tecnológica del Sistema.

Fuente: Elaboración propia, 2019.

#### ***Prototipo del esquema físico de las conexiones en Arduino***

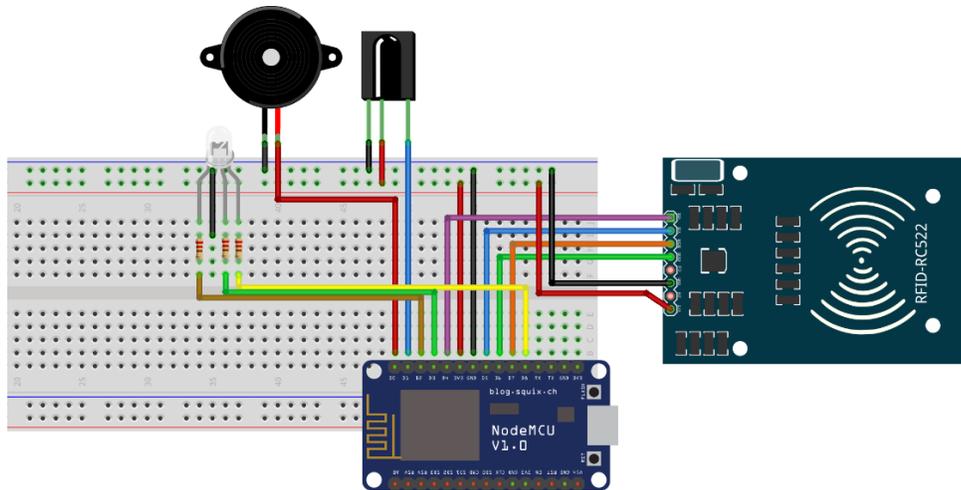


Figura 13. Diseño de esquema físico de conexiones en ESP8266.

Fuente: Elaboración propia, 2019.

### Prototipo del servicio web Arduino

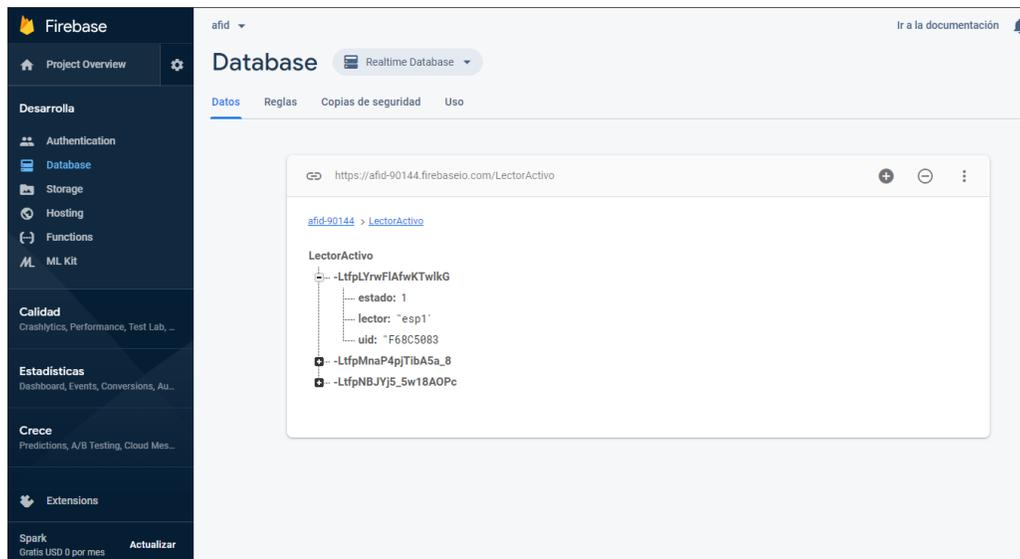


Figura 14. Vista de captura de datos de Arduino a Firebase.

Fuente: Elaboración propia, 2019.

### Prototipo del sistema web



Figura 15. Vista de Acceso al sistema.

Fuente: Elaboración propia, 2019.



Figura 16. Vista para registro de usuario.

Fuente: Elaboración propia, 2019.

REPORTE USUARIOS			
NOMBRE	CORREO	DNI	OPCIONES
Daniel León	linn@upeu.pe	72114589	[Edit] [Delete]
Anderson Choque		69356689	[Edit] [Delete]
Tony Aquino	SI NO	02458964	[Edit] [Delete]
David Huaricallo	david.huaricallo@gmail.com	03265908	[Edit] [Delete]
Eder Gutierrez	eder.gutierrez@upeu.pe	03656556	[Edit] [Delete]

Item per page: 1 1-5 de 5 < >

Figura 17. Vista para eliminar Usuarios.

Fuente: Elaboración propia, 2019.



Figura 18. Vista principal al sistema.

Fuente: Elaboración propia, 2019.

REPORTE BIENES			
NOMBRE	AMBIENTE	ARTICULO	OPCIONES
Muebles y útiles	REDES	MONITOR	[Edit] [Delete]
Muebles y útiles	REDES	MONITOR	[Edit] [Delete]
Muebles y útiles	AUDITORIO DTI	IMPRESORA	[Edit] [Delete]
Equipos informaticos	DESARROLLO	EQUIPO ELECTRONICO	[Edit] [Delete]
Equipos informaticos	SERVICIOS COMPUTACIONALES	LAPTOP	[Edit] [Delete]

Item per page: 1 1-5 de 5 < >

Figura 19 .Configuración de Bienes por Ambiente y artículo

Fuente: Elaboración propia, 2019.

### AGREGAR BIENES

Nombre

Artículo

Ambiente

Figura 20.Vista para agregar Bienes.

Fuente: Elaboración propia, 2019.

REPORTE ARTICULOS		
NOMBRE	CARACTERISTICA	OPCIONES
MONITOR	22 PULGADAS	[Edit] [Delete]
TECLADO	MEDIANO	[Edit] [Delete]
ESCRITORIO	CAFE	[Edit] [Delete]
SILLA	de oficina	[Edit] [Delete]
CPU	I7	[Edit] [Delete]

Item per page: 1 1-5 de 5 < >

Figura 21. Reporte de Artículos con su característica.

Fuente: Elaboración propia, 2019.

Cerrar Sesión →

**inventario**

SETUP

BIENES

ARTICULOS

AREAS

AMBIENTES

INVENTARIO

### AGREGAR ARTICULO

Nombre

Característica

Cantidad

Figura 22. Vista para crear artículos.

Fuente: Elaboración propia, 2019.

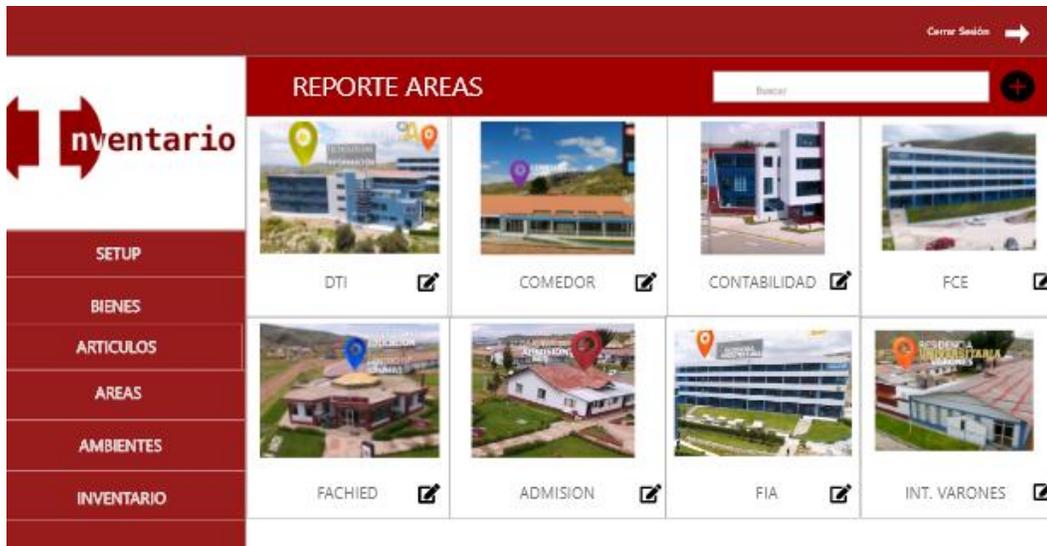


Figura 23. Configuración de Departamentos

Fuente: Elaboración propia, 2019.

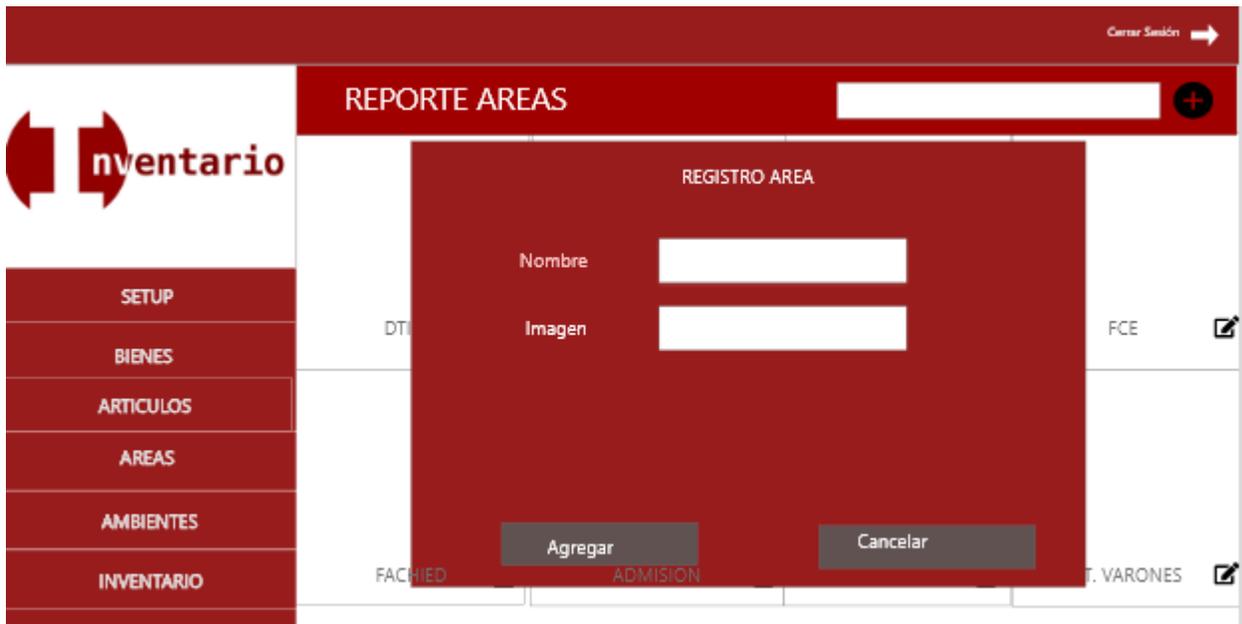


Figura 24. Vista para agregar Departamentos

Fuente: Elaboración propia, 2019.

REPORTE AMBIENTES		
AMBIENTE	ESTADO	OPCIONES
REDES	ACTIVO	[Icono de edición]
DESARROLLO	ACTIVO	[Icono de edición]
SERVICIOS COMPUTACIONALES	ACTIVO	[Icono de edición]
LAB-3	ACTIVO	[Icono de edición]
AUDITORIO DTI	ACTIVO	[Icono de edición]

Item per page: 1 1-5 de 5 < >

Figura 25. Vista para agregar Departamentos

Fuente: Elaboración propia, 2019.

Cerrar Sesión →

### REGISTRO AMBIENTE

Nombre Ambiente

Nombre Area

Descripcion

Figura 26. Vista para agregar Ambientes

Fuente: Elaboración propia, 2019.

AMBIENTE	ESTADO	FECHA	DETALLE
REDES	COMPLETO	26/10/19	▼
DESARROLLO	INCOMPLETO	26/10/19	▼
SERVICIOS COMPUTACIONALES	COMPLETO	26/10/19	▼
LAB-3	COMPLETO	26/10/19	▼
AUDITORIO DTI	INCOMPLETO	26/10/19	▼

Item per page: 1 - 1-5 de 5 < >

Figura 27. Configuración de Inventarios

Fuente: Elaboración propia, 2019

DETALLE INVENTARIO

Nombre Area: DTI

Nombre Ambiente: Redes

Nombre Articulo: Monitor

Estado: Incompleto ▼

Observación: El monitor faltante esta en desarrollo

Stock: 5

Cantidad: 4

Botones: Agregar, Cancelar

Figura 28. Configuración para el control de inventarios por departamentos y ambientes.

Fuente: Elaboración propia, 2019

### 3.6. Fase de codificación

#### 3.6.1. Programación del microcontrolador

En la figura 29 se puede observar un diagrama de flujo que describe la lógica del programa del microcontrolador en el lector MFRC522 que permite el funcionamiento del sistema general.

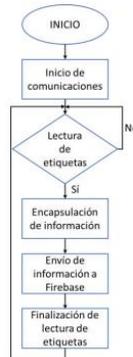


Figura 29. Diagrama de flujo de funcionamiento del código del microcontrolador

Fuente: Elaboración propia, 2019

Para la configuración del servidor web en Arduino se requirió incluir la librería *SPI.h* que permitió la comunicación entre ESP8266 y MFRC522. También se requirió incluir la librería *ESP8266.h* que es la que conectó el Arduino a internet. Después se configuró la red Wifi para tener el acceso a Internet.

```
#include <SPI.h>
#include <MFRC522.h>
#include <ESP8266WiFi.h>
#include <ArduinoJson.h>
#include <FirebaseArduino.h>

#define FIREBASE_HOST "afid-90144.firebaseio.com"
#define FIREBASE_AUTH "sf7S104Q4NscPsm3XZgtvB0qciWeIRbNa1TsIIM"
#define WIFI_SSID "daniel"
#define WIFI_PASSWORD "12345678"
#define RST_PIN D2
#define SS_PIN D4
MFRC522 mfrc522 (SS_PIN, RST_PIN);
```

En este sector del código se muestra la conexión a Firebase, a una red Wifi y al lector MRC52 en un archivo .ino.

```

void setup () {
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Conectado");
  while (WiFi.status() !=WL_CONNECTED){
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("Conectado: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

```

Para ver todo el código de Arduino, revise el Anexo 2.

### 3.6.2. Desarrollo Web NuxtJs

Para el desarrollo de la aplicación web de forma ágil se usó el framework NuxtJs, y con la utilización de la base de datos Firebase se permitió acceder al sistema. En la figura 30 se puede observar un diagrama de flujo que enseña las funciones de Firebase más relevantes para el proyecto:



Figura 30. Diagrama de flujo de proyecto en Firebase

Fuente: Elaboración propia, 2019

Para crear un proyecto en Firebase se debe tener una cuenta en Google. Al crear o añadir un nuevo proyecto, Firebase asigna un entorno de desarrollo y funciones para la creación de aplicaciones de distintas plataformas basadas en la base de datos, como se ve en la Figura 31.

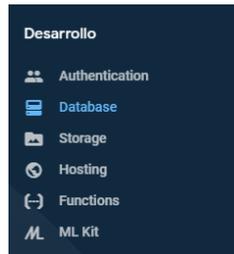


Figura 31. Entorno de desarrollo de Firebase.

Fuente: Elaboración propia, 2019

Tal como se muestra en el siguiente código. Componente LoginPage.vue, ver en anexo 3.

### Correo electrónico y contraseña

```
<vs-col vs-type="flex" vs-justify="flex-start" vs-align="center" vs-w="12">
  <vs-input
    class="ma-1"
    icon="person"
    label="Email"
    placeholder="example@gamil.com"
    v-model="user.email"
  />
</vs-col>
<vs-col
  v-if="register"
  vs-type="flex"
  vs-justify="flex-start"
  vs-align="center"
  vs-w="12"
>
  <vs-input
    class="ma-1"
    icon="person"
    label="Nombre completo"
    placeholder="Joel ..."
    v-model="user.nombres"
  />
</vs-col>
</vs-col>
```

```

vs-type="flex"
vs-justify="flex-start"
vs-align="center"
vs-w="12"
class="password__container"

```

>

## Cuenta existente de Google

```

methods: {
  async Login() {
    this.$vs.loading({ color: this.colorLoading });

    await firebase
      .auth()
      .signInWithEmailAndPassword(this.user.email, this.user.password)
      .then(async x => {
        await this.$store.commit("LOGIN", x.user.uid);
        //await this.$store.dispatch("getUser", x.user.uid);
        this.$router.push("/");
      })
      .catch(x => {
        switch (x.code) {
          case "auth/invalid-email":
            this.openNotification({
              text: `El correo electrónico no es válido`
            });
            break;
          case "auth/user-not-found":
            this.openNotification({
              text: `No hay registro de usuario correspondiente a este
identificador. El usuario puede haber sido eliminado.`
            });
            break;
          case "auth/wrong-password":
            this.openNotification({
              text: `La contraseña no es válida o el usuario no tiene una
contraseña.`
            });
            break;
          default:
            this.openNotification({
              text: `${x.message}`
            });
            break;
        }
      });
    this.$vs.loading.close();
  },

```

Se hizo la integración del servidor de Arduino en un componente <WebView/> y para poder controlar la cámara se usó un componente <Button/>. Componente Player.vue, ver en Anexo

3.

```
<template>
  <Page class="page">
    <ActionBar title="Video" class="action-bar" />
    <ScrollView>
      <StackLayout class="home-panel">
        <WebView height="1000px" src="http://192.168.1.70:5300/">
        <Button text="Ver Cámara" @tap="submit" class="buttoncam" />
      </StackLayout>
    </ScrollView>
  </Page>
</template>
```

Para acceder a la aplicación de la cámara mediante nuestra aplicación se usó el plugin “*nativescript-open-app*”, brindando el ID único “*com.xml.csee*”. Tal como se muestra en las siguientes líneas de código:

```
<script>
var openApp = require("nativescript-open-app").
openApp;

export default {
  methods: {
    submit() {
      var installed = openApp("com.xml.csee",false);
      console.log("Is it installed? " + installed);
    }
  },
  data () {
    return {
    };
  },
}
}</script>
```

### 3.7. Fase de pruebas.

#### 3.7.1. Pruebas de configuración del hardware

Tabla 23

*Caso de prueba - Correcto funcionamiento de los dispositivos electrónicos integrados con ESP8266 y MFRC522*

---

Caso de Prueba	
Número de Caso de Prueba:1	Número Historia de Usuario: HDU-01
Nombre Caso de Prueba: Correcto funcionamiento de los dispositivos electrónicos integrados con ESP8266 y MFRC522	
Descripción: prueba para el correcto funcionamiento de los dispositivos electrónicos integrados con la placa ESP8266 y MFRC522	
Condiciones de ejecución: se debe tener las conexiones físicas realizadas, entre dispositivos electrónicos, placa ESP8266 y MFRC522	
Entradas: Una vez compilado el programa en la placa Arduino.	
Resultado esperado: Se logra establecer las conexiones y tener la integración de toda la parte de hardware del sistema.	
Evaluación: prueba satisfactoria	

---

Fuente: Elaboración propia, 2019.



Figura 32. Integración de ESP8266 y MFRC522.

Fuente: Elaboración propia, 2019.

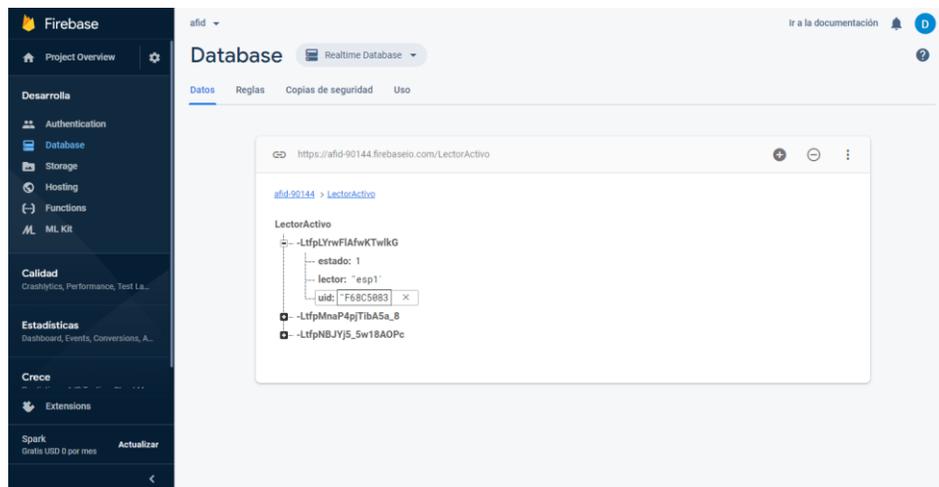


Figura 33. Correcto funcionamiento de la integración.

Fuente: Elaboración propia, 2019.

### 3.7.2. Pruebas de aceptación del sistema web

Para la realización de las pruebas de funcionalidad del sistema web, se definió las características mínimas con las que debe contar una Tablet o celular. Ver Anexo 1.

Tabla 24

Caso de prueba - Validar usuarios

---

Caso de Prueba	
Número de Caso de Prueba:2	Número Historia de Usuario: <b>REQF-001, REQNF-001, REQNF-002</b>
Nombre Caso de Prueba: Validar usuarios	
Descripción: prueba para la funcionalidad validar usuarios	
Condiciones de ejecución: el usuario tiene que estar previamente registrado y si no se otorga acceso desde el usuario administrador	
Entradas: el usuario accede al sistema mediante un email y contraseña o con una cuenta de existente de google validado por la UPeU	
Resultado esperado: muestra la pantalla principal de la aplicación	
Evaluación: prueba satisfactoria	

---

Fuente: Elaboración propia, 2019.

Se desarrolló dos métodos de acceso al sistema, el número 1 es usando un correo electrónico y contraseña validado por la UPeU, como se muestra en la *figura 34*, el método de autenticación de usuarios es proveídos por Firebase, como se puede apreciar en la *figura 35*. Las pruebas de aceptación se pueden ver en la parte de la metodología (fase de pruebas).

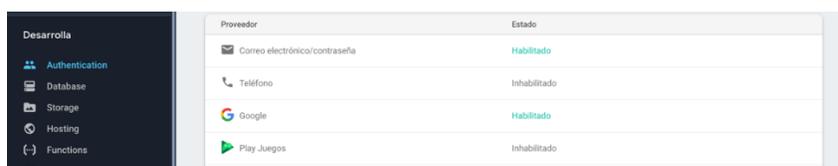


Figura 34. Habilitando los servicios de autenticación por correo electrónico/contraseña y google.

Fuente: Elaboración propia, 2019.

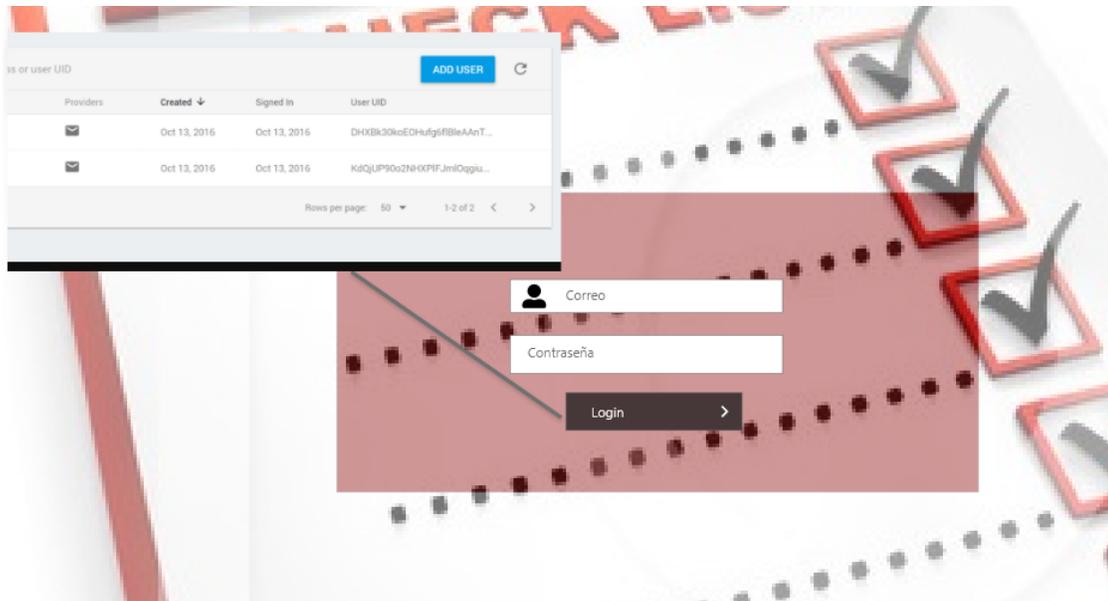


Figura 35. Método de acceso del sistema Web.

Fuente: Elaboración propia, 2019.

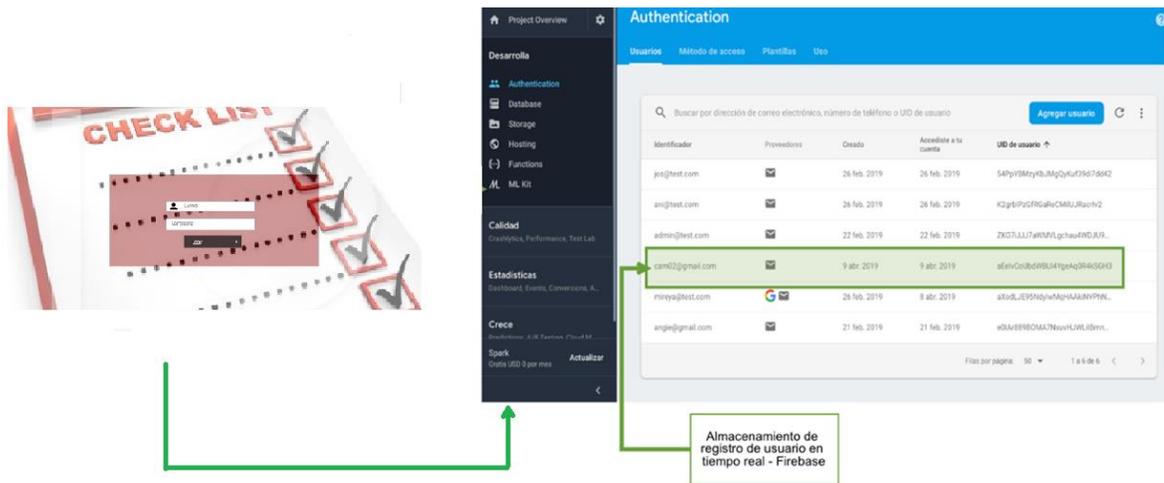


Figura 36. Registro de nuevo usuario, almacenamiento en tiempo real Firebase.

Fuente: Elaboración propia, 2019.

Tabla 25

Caso de prueba - Visualizar reporte de artículos por departamentos y ambientes

Caso de Prueba

Número de Caso de Prueba:3

Número Historia de Usuario: **REQF-002, REQNF-002**

Nombre Caso de Prueba: Visualizar reporte de artículos por departamento y ambientes en tiempo real.

Descripción: prueba para la funcionalidad de visualización de artículos por departamento y ambientes en tiempo real.

Condiciones de ejecución: el usuario tiene que ingresar al sistema

Entradas: una vez que el usuario ingrese al sistema presiona en la opción Inventario.

Resultado esperado: muestra el reporte en tiempo real.

Evaluación: prueba satisfactoria

Fuente: Elaboración propia, 2019.

AMBIENTE	ESTADO	FECHA	DETALLE
REDES	INCOMPLETO	26/11/19	▼
DESARROLLO	INCOMPLETO	26/11/19	▼
SERVICIOS COMPUTACIONALES	COMPLETO	26/11/19	▼
LAB-3	COMPLETO	26/11/19	▼
AUDITORIO DTI	INCOMPLETO	26/11/19	▼

Figura 37. Funcionalidad de reporte de inventario.

Fuente: Elaboración propia, 2019.

Tabla 26

Caso de prueba – Reporte de los departamentos y ambientes

---

Caso de Prueba	
Número de Caso de Prueba: 4	Número Historia de Usuario: <b>REQF-003,</b> <b>REQNF-002</b>
Nombre Caso de Prueba: Visualizar el reporte de los departamentos y ambientes.	
Descripción: prueba para la funcionalidad de la visualización de los departamentos y ambientes.	
Condiciones de ejecución: el usuario tiene que ingresar al sistema Web	
Entradas: una vez que el usuario ingrese al sistema presiona la opción de departamentos para visualizar el estado y ambientes que contiene.	
Resultado esperado: los ambientes deben estar ordenados por departamentos	
Evaluación: prueba satisfactoria	

---

Fuente: Elaboración propia, 2019.



Figura 38. Funcionalidad de Departamentos y ambientes.

Fuente: Elaboración propia, 2019

## CAPITULO IV. RESULTADOS Y DISCUSIÓN

### 4.1. Resultado del objetivo específico 1

Para el resultado 1 se usó el método de las historias de usuario, a través de esto se logró concretar el seguimiento para el procedimiento de control de inventarios, posteriormente se logró diseñar un nuevo proceso para el uso del sistema descrito en el capítulo IV. Para la corroboración de los resultados se realizó un test con el cual se recogió información (los datos recopilados se visualizan en el Anexo 7). En el cual un 14.3% de los trabajadores están **muy de acuerdo** con el aplicativo, así mismo 64.3% estuvieron **de acuerdo** con el sistema y sus funcionalidades y el otro porcentaje de 14.3% fue aprobado en término medio **ni en acuerdo tampoco en desacuerdo** y el porcentaje restante de 7.1% estuvo en **desacuerdo**. En cuanto a la funcionalidad que se detalla en anexo 7

El tiempo que toma al realizar un proceso en el sistema AFID es aceptable.

14 respuestas

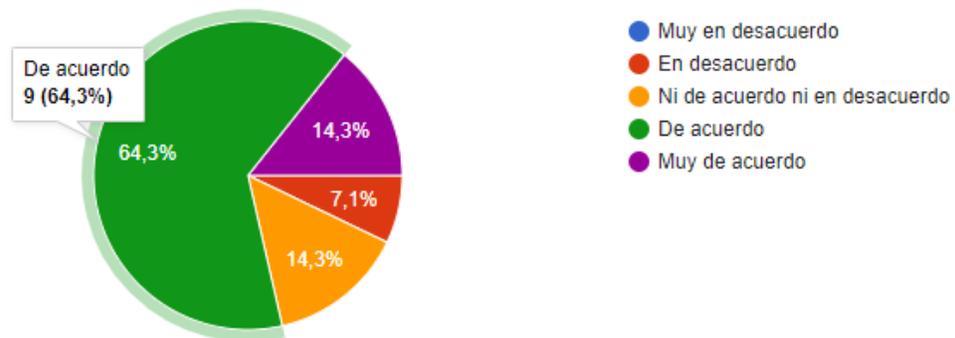


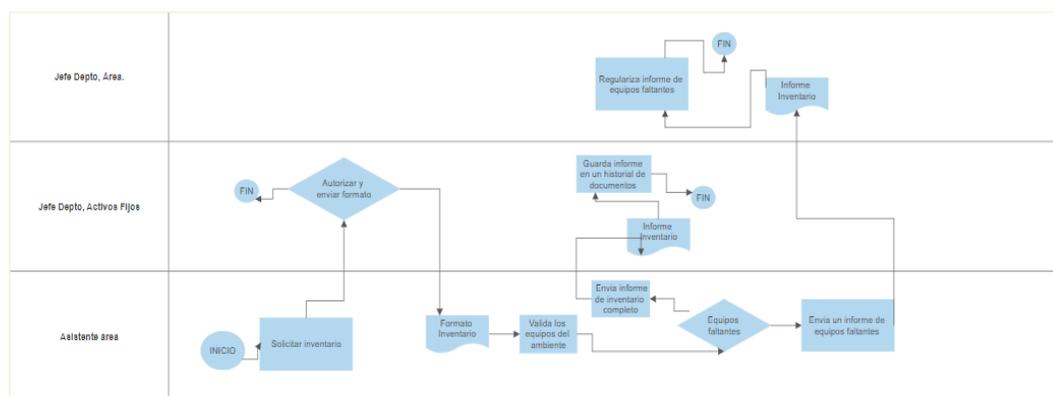
Figura 39. Resultado de encuesta del sistema AFID.

Fuente: Elaboración propia, 2019.

Por otra parte, en la encuesta se contempló un punto muy importante y fue formulada de la siguiente manera: ¿El tiempo de respuesta al realizar un proceso en el sistema AFID es aceptable? Y la respuesta de un total de 14 trabajadores, 9 consideran que el tiempo de respuesta es aceptable comparado con el anterior control que se realizaba. Esto equivale al 64.3% que se puede visualizar en la *figura 39*.

#### 4.1.1. Discusión

Uno de los propósitos de la presente investigación fue analizar el funcionamiento de los servicios que realiza el área de activos fijos en la Universidad Peruana Unión filial Juliaca, el área de activos fijos tiene como modelos políticas para el control de inventarios la cual usan un formato Excel para tener control de todos los equipos por departamentos y ambientes. En investigaciones anteriores González & Salcedo(2017) implementó un sistema de Inventarios sin embargo no diseño su proceso de mejora para medir la funcionalidad en el desarrollo del sistema. En esta investigación se usó la entrevista y se dio seguimiento al procedimiento de inventarios. En la *figura 40* se muestra el esquema general del procedimiento para el control de inventarios.



*Figura 40.* Esquema general del procedimiento para el control de inventarios.

Fuente: Elaboración propia, 2019.

## **4.2. Resultado del objetivo específico 2**

En esta parte se utilizó la técnica de la entrevista a los jefes de área para levantar las historias de usuario, a través de esto se concretaron los requerimientos funcionales y no funcionales del sistema Web, ver tablas 14 y 15, como resultado se obtuvo la integración del hardware que va enlazado a un sistema Web y base de datos Firebase para el control de inventarios en tiempo real, que cumplió expectativas para el personal del área de Activos Fijos.

### **4.2.1. Discusión**

Otro de los propósitos para realizar el sistema Web de la presente investigación, primero fue integrar las tecnologías de Identificación por Radiofrecuencia. Sanchez (2017) quien desarrolló un sistema entorno escritorio basado en Microsoft Visual Studio 2010 con lenguaje de programación Visual Basic, también Guizar (2017) desarrolló un sistema de RFID integrando toda la administración y operación logística de la empresa. Tomando en cuenta los antecedentes mencionados, en esta investigación se realizó la integración de las tecnologías para el control de Activos Fijos con el uso de un ESP8266 enlazada con la placa MFRC522 y Firebase para el almacenamiento de datos, una aplicación Web desarrollada con Nuxt.js-Vuejs como puente para gestionar los datos en tiempo real.

### **4.3. Resultado del objetivo específico 3**

Para la realización de las pruebas se consideró un indicador importante las notificaciones enviadas al sistema Web desde un MFRC522 almacenadas en una base de datos Firebase, notificando la presencia del activo mediante un TAG RFID, en cuanto a las funcionalidades del sistema se obtuvo un resultado satisfactorio, como se puede ver en la fase de pruebas descrito en el capítulo IV. Dando como resultado una versión de producción del sistema Web.

#### **4.3.1. Discusión**

Otro de los propósitos de la presente investigación fue realizar pruebas del sistema Web y las placas Arduino Cifuentes (2019) realizó sus pruebas de aceptación mediante la aceptación de los usuarios quienes le indicaron si su sistema cumplía con sus requerimientos, no desarrolló ningún formato, a la vez Quispe (2017) quién realizó las pruebas de la implementación mediante el equipo de Sensores RFID y esto es reflejado en la aplicación de PC. En ambos casos no se desarrolló un aplicativo Web sino hicieron uso de otras tecnologías como C#, Visual Basic, en esta investigación se codificó una aplicación Web, se usó conexiones a Wifi, y para la realización de las pruebas.

## **CAPITULO V.**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1. Conclusiones**

Conforme a los resultados de esta investigación de desarrollo de un sistema enlazado con Tecnología Arduino RFID y ESP8266 para el control de activos fijos en tiempo real, se concluyó lo siguiente:

Con respecto al objetivo general, se desarrolló un sistema Web enlazado con Tecnología Arduino RFID y ESP8266 para el control de activos fijos, validado con las pruebas de aceptación por parte del jefe de área y el personal de Activos Fijos tanto en hardware y software.

Con respecto al primer objetivo específico se analizó el procedimiento del control de activos que tiene el área realizando visitas al personal, siendo el sistema RFID totalmente portable y gracias a que el lector RFID puede conectarse a un cargador portable para encenderlo, se redujo las horas extra del personal.

Con respecto al segundo objetivo se analizó y diseñó los requerimientos levantados para el sistema Web, mediante las historias de usuario que vienen siendo los requerimientos, y los prototipos tanto para la parte del hardware y software del sistema Web.

Con respecto al tercer objetivo se concluyó que, si es factible integrar hardware con software tratándose de tecnologías de un servicio en Arduino, se realizó pruebas al sistema Web permitiendo detectar los tag RFID desde un Sensor MFRC522 conectado a un módulo ESP8266 para la conexión de internet y el almacenamiento de datos en un Firebase, las cuales tuvieron resultados satisfactorios.

## **5.2. Recomendaciones**

Se recomienda aumentar el rango de lectura del Sensor RFID para la mejor localización de las etiquetas RFID.

Se recomienda construir una presentación (armazón) del sensor RFID para un mejor uso, lo cual no se realizó por motivos de presupuesto.

Continuar con el soporte y actualizaciones de la aplicación Web, agregando nuevas características para versiones futuras: como trazado de rutas de activos para saber en qué lugares de destino estuvo.

Se recomienda agregar un sensor de huella digital para identificar a cada encargado de Inventario y controlar el trabajo que realizan.

## REFERENCIAS

- Alcon Baltzar, M. A. (2016). *Sistema De De Control Y Monitoreo De Asistencia De Personal Mediante Radiofrecuencia Y Nfc*.
- Alejandro Menese, C. P. (2013). *Diseño de un sistema de Control de sctivos para el almacen de electronica de la Pontificia Universidad Catolica del Perú utilizando RFIED* (Pontificia Universidad Católica del Perú). Retrieved from [http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/1507/ALEJANDRO\\_MENESES\\_CRISTHIAN\\_CONTROL\\_ACTIVOS.pdf?sequence=1&isAllowed=y](http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/1507/ALEJANDRO_MENESES_CRISTHIAN_CONTROL_ACTIVOS.pdf?sequence=1&isAllowed=y)
- Arduino. (2016). Arduino IDE.
- Cáceres Álvarez, L., & Carpio, A. O. (2018). *Desarrollo de una aplicación móvil para reconocimiento de personas después de una catástrofe natural utilizando la tecnología NFC* *Development of a mobile application for recognition of people after a natural disaster using NFC technology*. 26, 28–44. Retrieved from <https://scielo.conicyt.cl/pdf/ingeniare/v26s1/0718-3305-ingeniare-26-00028.pdf>
- Castellote Garcia, M. (2017). Desarrollo de una aplicación Android de apuestas utilizando Firebase para la sincronización de datos. *UNIVERSITAT JAUME L*, (1), 1–92.
- Cifuentes, S. C. (2019). Desarrollo de sistema multifuncional para la Pontificia Universidad Javeriana Cali con manilla RFID. *Pontificia Universidad Javeriana Cali*. Retrieved from [http://www.ghbook.ir/index.php?name=فرهنگ و رسانه های نوین&option=com\\_dbook&task=readonline&book\\_id=13650&page=73&chckhashk=ED9C9491B4&Itemid=218&lang=fa&tmpl=component](http://www.ghbook.ir/index.php?name=فرهنگ و رسانه های نوین&option=com_dbook&task=readonline&book_id=13650&page=73&chckhashk=ED9C9491B4&Itemid=218&lang=fa&tmpl=component)
- González, C., & Salcedo, O. (2017). Sistema de seguridad para locales comerciales mediante Raspberry Pi, cámara y sensor PIR. *Revista Virtual*, 175–193.
- Guizar Sepúlveda, E. A. (2017). *Implementación de RFID en un almacén logístico* (Escola Tècnica Superior d'Enginyeria Industrial de Barcelona). Retrieved from <https://upcommons.upc.edu/bitstream/handle/2117/114261/memoria-tfm-implementacio-n-rfid.pdf?sequence=1&isAllowed=y>

- Henao, C., & Cardona, E. (2009). Programando Microcontroladores Pic En Lenguaje C. *Scientia et Technica*, 3(43), 37–42. <https://doi.org/10.22517/23447214.2225>
- Huamaní Navarrete, P. F. (n.d.). *La investigación Tecnológica*.
- INEI. (2017). *La victimización en las empresas 2015-2016*. Retrieved from [https://www.inei.gob.pe/media/MenuRecursivo/publicaciones\\_digitales/Est/Lib1434/libro.pdf](https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1434/libro.pdf)
- Morales, M. (2015). *Ingeniería en Informática*.
- Murillo Soto, L. D. (2015). Automatización de pequeña escala con Open Hardware Automation of small-scale with Open Hardware Palabras clave. *Tecnología En Marcha. Tecnología En Marcha*, 28(1), 15–23.
- Poch Torres, R. (2000). *Manual de Control Interno* (E. G. 2000, Ed.). Retrieved from <https://www.casadellibro.com/libro-manual-de-control-interno/9788486582128/172158>
- Quilligana, D. D. Q. (2018). *Implementación De Un Prototipo: Sistema Automático De Censado Inalámbrico Basado En Rfid Para Estaciones De Control De Buses Urbanos De Riobamba*.
- Quispe Vega, K. (2017). *Implementación de un sistema RFID para mejorar la productividad de una planta de producción de vidrio templado Para optar el Título Profesional de Ingeniero Industrial* (Universidad Nacional Mayor de San Marcos). Retrieved from [http://cybertesis.unmsm.edu.pe/bitstream/handle/cybertesis/6290/Quispe\\_vk.pdf?sequence=1](http://cybertesis.unmsm.edu.pe/bitstream/handle/cybertesis/6290/Quispe_vk.pdf?sequence=1)
- RC522. (2018). rc522 rfid.
- Roget, A. D. (2018). *La práctica reflexiva en la formación inicial de maestros/as. Evaluación de un modelo*.
- Ruiz Sanchez, W. A. (2017). *Implementación De Un Sistema De Gestión Del Activo Fijo Para El Instituto Arzobispo Loayza Los Olivos, 2013*. 1–105.
- Sachon, M., & Almirall, E. (2009). *tecnología RFID*. 1–8. Retrieved from

[www.spri.eus/euskadinnova/documentos/352.aspx](http://www.spri.eus/euskadinnova/documentos/352.aspx)

Sánchez Alvarado, J. A. (2014). *Sistema de Control de Acceso con RFID* (Instituto Politécnico Nacional). Retrieved from <http://www.cs.cinvestav.mx/TesisGraduados/2008/tesisJorgeAlvarado.pdf>

Tarrillo Saldaña, E. (2016). Influencia de la Gestión de Riesgo en la seguridad de Activos de Información de la zona Registral III Sede Moyobamba, 2015. *Universidad César Vallejo*.

Vargas Manuel, Castillo-Georgina, S.-J. &, & Brambila-Alfredo. (2015). Arduino una Herramienta Accesible para el Aprendizaje de Programación. *Artículo Revista de Tecnología e Innovación Septiembre, 2(4)*, 810–815. Retrieved from [www.ecorfan.org/bolivia](http://www.ecorfan.org/bolivia)

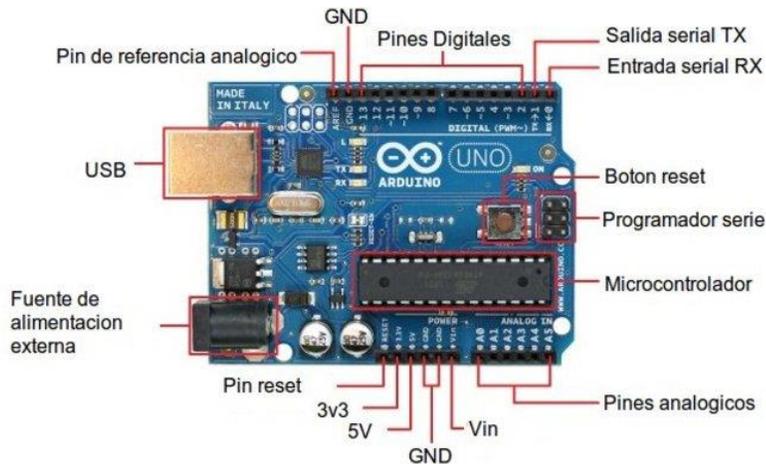
Vega E., A. M., Santamaría P., F., & Rivas T., E. (2014). Internet de los objetos empleando arduino para la gestión eléctrica domiciliaria. *Revista EAN, (77)*, 24. <https://doi.org/10.21158/01208160.n77.2014.813>

Xataca. (2018). xatana arduino.

## ANEXOS

### Anexo 1. Especificaciones técnicas de hardware usado

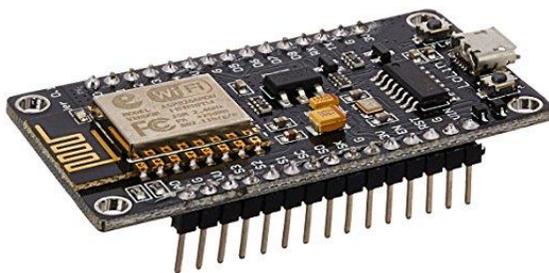
#### Arduino Uno R3



#### Especificaciones técnicas:

- Microcontrolador ATmega328.
- Voltaje de entrada 7-12V.
- 14 pines digitales de I/O (6 salidas PWM).
- 6 entradas análogas.
- 32k de memoria Flash.
- Reloj de 16MHz de velocidad.

#### Arduino ESP8266



#### Especificaciones técnicas:

- 32-bit RISC CPU: Tensilica Xtensa LX106 corriendo a 80 MHz (que puede ser overclokeado a 160MHz si se requiere)
- 64 KiB de RAM para instrucciones y 96 KiB de RAM para datos
- IEEE 802.11 b/g/n Wi-Fi
- 16 pines GPIO

- SPI e I2C
- UART en los pines dedicados (usada para la programación del chip)
- Un convertidor Analógico-Digital (ADC) de 10 bit.

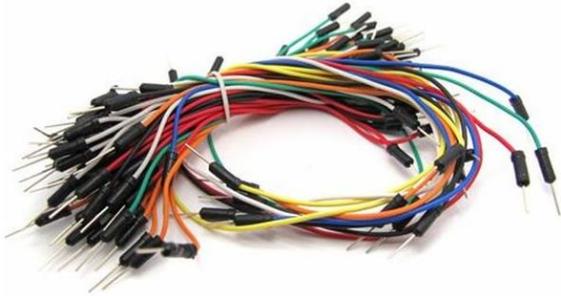
## RFID



### Especificaciones técnicas:

- Voltaje de Operación: 3.3V DC
- Corriente de Operación: 13-26mA/3.3V DC
- Corriente de Standby: 10-13mA/3.3V DC
- Corriente de Sleep: <80uA
- Corriente pico: <30mA
- Frecuencia de operación: 13.56 MHz
- Transferencia de datos: Max. 10Mbit/s
- Tipos de tarjetas compatibles: Mifare1 S50, S70 Mifare1, MIFARE Ultralight, Mifare Pro, Mifare DESFire.
- Dimensiones RFID-RC522: 40 mm x 60 mm
- Dimensiones Tarjeta: 85 mm x 54 mm
- Temperatura de funcionamiento: -20 a 80 grados centígrados
- Temperatura de almacenamiento: -40 a 85 grados centígrados
- Humedad relativa: 5% hasta 95 %
- La tasa de transmisión por defecto: 9600bps, velocidad de transferencia máxima : 1228800bps

## Cables Dupont



### Especificaciones técnicas:

- Tipo: Hembra a hembra, hembra a macho
- Espaciado: 2.54mm (0.1 pulgada)
- Longitud: 20cm, 10cm
- Cantidad: 6 unidades

## Smartphone para prueba

### Características:

- Procesador: Snapdragon 435 1.4GHz
- RAM: 2GB
- Almacenamiento: 16GB
- OS: Android 7.0
- Pantalla: 5.5", 720 x 1280 pixels

## Anexo 2. Código de Arduino

```
#include <SPI.h>
#include <MFRC522.h>
#include <ESP8266WiFi.h>
#include <ArduinoJson.h>
#include <FirebaseArduino.h>

#define FIREBASE_HOST "afid-90144.firebaseio.com"
#define FIREBASE_AUTH "sf7S104Q4NscPsnm3XZgtvB0qcIWeIRbNa1TslIM"
#define WIFI_SSID "Desarrollo"
#define WIFI_PASSWORD "sonriesiempre"
#define RST_PIN D2
#define SS_PIN D4
MFRC522 mfrc522 (SS_PIN, RST_PIN);
```

```

void setup () {
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Conectado");
  while (WiFi.status() !=WL_CONNECTED){
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("Conectado: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

int n = 0;
byte actualuid[4];
char uid[32] = "";
String lectorid = "esp1";

void loop(){
  if(mfrc522.PICC_IsNewCardPresent()){
    if( mfrc522.PICC_ReadCardSerial()){
      Serial.print("Card UID:");
      for (byte i = 0; i < mfrc522.uid.size; i++){
        actualuid[i]=mfrc522.uid.uidByte[i];
      }
      array_to_string(actualuid, 4, uid);
      String uid2(uid);
      StaticJsonBuffer<200> jsonBuffer;
      JsonObject& root = jsonBuffer.createObject();
      root["estado"] = 1;
      root ["lector"] = lectorid;
      root ["uid"] = uid2;

      Firebase.push("LectorActivo/", root);
      mfrc522.PICC_HaltA();
    }}

void array_to_string(byte array[], unsigned int len, char buffer[])
{

```

```

for (unsigned int i = 0; i < len; i++)
{
    byte nib1 = (array[i] >> 4) & 0x0F;
    byte nib2 = (array[i] >> 0) & 0x0F;
    buffer[i*2+0] = nib1 < 0xA ? '0' + nib1 : 'A' + nib1 - 0xA;
    buffer[i*2+1] = nib2 < 0xA ? '0' + nib2 : 'A' + nib2 - 0xA;
}
buffer[len*2] = '\0';
}

```

### Anexo 3. Código VueJS-NuxtJS

#### LoginPage.vue

```

<template>
  <vs-col vs-type="flex" vs-justify="center" vs-align="center" vs-w="12"
  class="login__page">
    <vs-col vs-type="flex" vs-justify="center" vs-align="center" vs-lg="7" vs-sm="11" vs-
  xs="11">
      <vs-card class="card">
        <vs-row>
          <vs-col
            vs-type="flex"
            vs-justify="center"
            vs-align="center"
            vs-lg="6"
            vs-sm="6"
            vs-xs="12"
            class="logo_container light"
          >
            
          </vs-col>
          <vs-col
            vs-type="flex"
            vs-justify="center"
            vs-align="center"
            vs-lg="6"
            vs-sm="6"
            vs-xs="12"
          >
            <vs-row class="ma-3">
              <vs-col
                vs-type="flex"
                vs-justify="center"

```

```

    vs-align="center"
    vs-w="12"
    class="text-xs-center"
  >
  <h1 class="ma-4 display-1 primary-text font-weight-bold">{{title}}</h1>
</vs-col>
<vs-col
  vs-type="flex"
  vs-justify="flex-start"
  vs-align="center"
  vs-w="12"
  v-if="!register"
>
  <span
    class="ml-2 mr-2 mb-2 title font-weight-thin text-xs-center"
  >Bienvenido, por favor inicie sesión en su cuenta.</span>
</vs-col>
<vs-col vs-type="flex" vs-justify="flex-start" vs-align="center" vs-w="12" v-
else>
  <span
    class="ml-2 mr-2 mb-2 title font-weight-thin text-xs-center"
  >Por favor complete los siguientes campos para su registro.</span>
</vs-col>
<vs-col vs-type="flex" vs-justify="flex-start" vs-align="center" vs-w="12">
  <vs-input
    class="ma-1"
    icon="person"
    label="Email"
    placeholder="example@gamil.com"
    v-model="user.email"
  />
</vs-col>
<vs-col
  v-if="register"
  vs-type="flex"
  vs-justify="flex-start"
  vs-align="center"
  vs-w="12"
>
  <vs-input
    class="ma-1"
    icon="person"
    label="Nombre completo"
    placeholder="Joel ..."
    v-model="user.nombres"
  />

```

```

</vs-col>
<vs-col
  vs-type="flex"
  vs-justify="flex-start"
  vs-align="center"
  vs-w="12"
  class="password__container"
>
  <vs-input
    class="ma-1"
    icon="lock"
    label="Contraseña"
    placeholder="*****"
    :type="mode_password?'password':'text'"
    v-model="user.password"
  />
  <vs-button
    color="primary"
    type="flat"
    :icon="mode_password?'visibility':'visibility_off'"
    radius
    size="small"
    class="button_visivility"
    @click="mode_password=!mode_password"
  ></vs-button>
</vs-col>

<vs-col
  vs-type="flex"
  vs-justify="flex-end"
  vs-align="center"
  vs-w="12"
  class="ma-2 mb-3"
>
  <vs-button
    :disabled="!user.email || !user.password"
    color="primary"
    class="btn-login"
    size="large"
    type="filled"
    @click="Login"
  >INGRESAR</vs-button>
</vs-col>
</vs-row>
</vs-col>
</vs-row>

```

```

    </vs-card>
  </vs-col>
</vs-col>
</template>

```

```

<script>

```

```

import { mapState } from "vuex";
import firebase from "@plugins/firebase.js";

```

```

export default {
  layout: "login",
  middleware: "login",
  data: () => ({
    user: {
      email: "activos-admin@upeu.edu.pe",
      password: "@ct1v0s2o19"
    },
    mode_password: true,
    filiales: [],
    servidor: true,
    register: false
  }),

```

```

  computed: {
    ...mapState(["title", "auth"])
  },

```

```

  methods: {
    async Login() {
      this.$vs.loading({ color: this.colorLoading });

```

```

      await firebase
        .auth()
        .signInWithEmailAndPassword(this.user.email, this.user.password)
        .then(async x => {
          await this.$store.commit("LOGIN", x.user.uid);
          //await this.$store.dispatch("getUser", x.user.uid);
          this.$router.push("/");
        })
        .catch(x => {
          switch (x.code) {
            case "auth/invalid-email":
              this.openNotification({
                text: `El correo electrónico no es válido`
              });
              break;
            case "auth/user-not-found":

```

```

        this.openNotification({
            text: `No hay registro de usuario correspondiente a este identificador. El
usuario puede haber sido eliminado.`
        });
        break;
        case "auth/wrong-password":
            this.openNotification({
                text: `La contraseña no es válida o el usuario no tiene una contraseña.`
            });
            break;
        default:
            this.openNotification({
                text: `${x.message}`
            });
            break;
    }
    });
    this.$vs.loading.close();
},
openNotification(payload) {
    this.$vs.notify({
        color: payload.type ? payload.type : "danger",
        position: "top-right",
        icon: payload.icon ? payload.icon : "error",
        time: payload.time ? payload.time : 5000,
        title: payload.title ? payload.title : "Error!",
        text: payload.text
    });
}
}
};
</script>

```

```

<style lang="scss" >
.login__page {
    height: 100vh;
    .card {
        padding: 0;
        margin: 0;
        .vs-card--content {
            margin: 0;
            padding: 0;
        }
    }
}
.logo_container {
    background: #f7f7f7;
}

```

```

margin: 0;
padding: 0;
border-radius: 8px 8px 0 0;
}
.vs-input {
width: 100%;
.vs-input--input {
width: 100%;
height: 40px;
}
}
.con-select {
width: 100%;
.vs-select--input {
height: 40px;
}
}
}

.btn-login {
min-width: 80px;
}
@media screen and (min-width: 600px) {
.login-logo {
height: 200px !important;
}
.logo_container {
border-radius: 8px 0 0 8px !important;
}
}

.password__container {
position: relative;
.button_visivility {
position: absolute;
right: 10px;
top: 25px;
}
}
</style>

```

**Index.vue**

```

<template>
  <vs-row class="container fluid">
    <vs-col>
      <vs-table
        pagination
        max-items="7"
        search
        :data="users"
        class="pa-3"
        no-data-text="El Sistema no tiene Usuarios"
      >
        <template slot="header">
          <vs-col      vs-align="center"      vs-type="flex"      vs-justify="center"
class="table__header">
            <span class="title">Usuarios del Sistema</span>
            <vs-button
              class="add_button"
              radius
              size="large"
              color="primary"
              type="filled"
              icon="add"
              @click="popupUsers=true,user={}"
            ></vs-button>
          </vs-col>
        </template>
        <template slot="thead">
          <vs-th sort-key="email">Email</vs-th>
          <vs-th sort-key="nombres">Nombres</vs-th>
          <vs-th sort-key="apellidos">Apellidos</vs-th>
          <vs-th sort-key="dni">DNI</vs-th>
          <vs-th>
            <div class="text-xs-center" style="width:100%">Opciones</div>
          </vs-th>
        </template>

        <template slot-scope="{data}">
          <vs-tr :data="tr" :key="indextr" v-for="(tr, indextr) in data">
            <vs-td :data="data[indextr].email">{{data[indextr].email}}</vs-td>

            <vs-td :data="data[indextr].nombres">{{data[indextr].nombres}}</vs-td>

            <vs-td :data="data[indextr].apellidos">{{data[indextr].apellidos}}</vs-td>

            <vs-td :data="data[indextr].dni">{{data[indextr].dni}}</vs-td>
            <vs-td>

```

```

<vs-col vs-align="flex-start" vs-type="flex" vs-justify="space-around">
  <vs-button
    color="warning"
    type="filled"
    icon="edit"
    radius
    @click="user=data[indextr],popupUsers=true"
  ></vs-button>
  <vs-button
    color="danger"
    type="filled"
    icon="delete"
    radius
    @click="deleteUser(user=data[indextr].id)"
  ></vs-button>
</vs-col>
</vs-td>
</vs-tr>
</template>
</vs-table>
</vs-col>
<vs-popup class="holamundo" title="Información del usuario"
:active.sync="popupUsers">
  <vs-row>
    <vs-col
      vs-align="center"
      vs-type="flex"
      vs-justify="center"
      vs-xs="12"
      vs-sm="6"
      vs-lg="6"
      class="mt-2"
    >
      <vs-input size="large" label-placeholder="Nombres" v-model="user.nombres" />
    </vs-col>
    <vs-col
      vs-align="center"
      vs-type="flex"
      vs-justify="center"
      vs-xs="12"
      vs-sm="6"
      vs-lg="6"
      class="mt-2"
    >
      <vs-input size="large" label-placeholder="Apellidos" v-model="user.apellidos" />
    </vs-col>

```

```

<vs-col
  vs-align="center"
  vs-type="flex"
  vs-justify="center"
  vs-xs="12"
  vs-sm="6"
  vs-lg="6"
  class="mt-2"
>
  <vs-input
    size="large"
    label-placeholder="Email"
    v-model="user.email"
    type="email"
    :disabled="user.id"
  />
</vs-col>
<vs-col
  vs-align="center"
  vs-type="flex"
  vs-justify="center"
  vs-xs="12"
  vs-sm="6"
  vs-lg="6"
  class="mt-2"
>
  <vs-input size="large" label-placeholder="Documento de identidad" v-
model="user.dni" />
</vs-col>
<vs-col vs-align="center" vs-type="flex" vs-justify="flex-end" class="mt-4 mr-3 mb-
2">
  <vs-button
    size="large"
    type="filled"
    color="danger"
    @click="popupUsers=false, user={}"
    class="mr-2"
  >Cancelar</vs-button>
  <vs-button
    size="large"
    type="filled"
    color="success"
    @click="saveUser"
    :disabled="!user.email || !user.dni || !user.nombres ||!user.apellidos"
  >Guardar</vs-button>
</vs-col>

```

```
    </vs-row>
  </vs-popup>
</vs-row>
</template>
```

```
<script>
```

```
import firebase from "@plugins/firebase.js";
const userInstance = firebase.firestore().collection("User");
```

```
export default {
```

```
  data: () => ({
    selected: [],
    popupUsers: false,
    users: [],
    user: {}
```

```
  }),
```

```
  async created() {
    await this.getUsers();
```

```
  },
```

```
  methods: {
```

```
    async deleteUser(id) {
```

```
      this.$vs.dialog({
        type: "confirm",
        color: "danger",
        title: `Eliminar`,
        acceptText: "Eliminar",
        cancelText: "Cancelar",
        text:
```

```
        "El usuario será puesto en estado: ELIMINADO, por lo que ya no tendrá acceso al sistema",
```

```
        accept: async () => {
          await this.confirmDeleteUser(id);
        }
      });
```

```
    });
```

```
  },
```

```
  async confirmDeleteUser(id) {
```

```
    this.$vs.loading({ color: "#5b3cc4", type: "material" });
```

```
    await userInstance
```

```
      .doc(id)
```

```
      .update({ state: "ELIMINADO" })
```

```
      .then(result => {
```

```
        this.openNotification({
```

```
          type: "success",
```

```
          title: "Success!!!",
```

```
          text: `El usuario se eliminó con éxito...`
```

```
        });
```

```

    this.popupUsers = false;
    this.user = {};
  })
  .catch(err => {
    this.openNotification({ text: `${err}` });
  });
  this.$vs.loading.close();
},
async saveUser() {
  this.$vs.loading({ color: "#5b3cc4", type: "material" });
  if (this.user.id) {
    const temp = { ...this.user };
    delete temp.id;
    await userInstance
      .doc(this.user.id)
      .update({ ...temp })
      .then(result => {
        this.openNotification({
          type: "success",
          title: "Success!!!",
          text: `El registro se actualizó con éxito...`
        });
        this.popupUsers = false;
        this.user = {};
      })
      .catch(err => {
        this.openNotification({ text: `${err}` });
      });
  } else {
    await firebase
      .auth()
      .createUserWithEmailAndPassword(this.user.email, this.user.dni)
      .then(async result => {
        await userInstance
          .doc(result.user.uid)
          .set({ ...this.user, state: "ACTIVE" })
          .then(result => {
            this.openNotification({
              type: "success",
              title: "Success!!!",
              text: `El registro se reazlizó con éxito...`
            });
            this.popupUsers = false;
            this.user = {};
          })
          .catch(err => {

```

```

        this.openNotification({ text: `${err}` });
    });
}
.catch(x => {
    switch (x.code) {
        case "auth/invalid-email":
            this.openNotification({
                text: `El correo electrónico no es válido`
            });
            break;
        case "auth/user-not-found":
            this.openNotification({
                text: `No hay registro de usuario correspondiente a este identificador. El
usuario puede haber sido eliminado.`
            });
            break;
        case "auth/wrong-password":
            this.openNotification({
                text: `La contraseña no es válida o el usuario no tiene una contraseña.`
            });
            break;
        default:
            this.openNotification({
                text: `${x.message}`
            });
            break;
    }
});
}

this.$vs.loading.close();
},
async getUsers() {
    this.$vs.loading({ color: "#5b3cc4", type: "material" });
    await userInstance.where("state", "==", "ACTIVE").onSnapshot(
        querySnapshot => {
            this.users = [];
            querySnapshot.forEach(doc => {
                const user = doc.data();
                user["id"] = doc.id;
                this.users.push(user);
            });
            this.$vs.loading.close();
        },
        error => {
            this.$vs.loading.close();
        }
    );
}

```

```

    }
  );
},
openNotification(payload) {
  this.$vs.notify({
    color: payload.type ? payload.type : "danger",
    position: "top-right",
    icon: payload.icon ? payload.icon : "error",
    time: payload.time ? payload.time : 5000,
    title: payload.title ? payload.title : "Error!",
    text: payload.text
  });
}
}
};
</script>

```

```

<style lang="scss" scoped>
.table__header {
  min-height: 80px;
  position: relative;
  display: flex;
  justify-content: center;
  align-items: center;
  .add_button {
    position: absolute;
    right: 10px;
  }
}
</style>

```

## Firestore.js

```

import * as firebase from 'firebase/app';import 'firebase/firestore';
import 'firebase/database';
import 'firebase/auth';
import 'firebase/storage';const config = {
  apiKey: 'AlzaSyC5VTI_J48K5NxQDLtXEoN-bOTmsy_ZpL0',
  authDomain: 'finka-444ec.firebaseio.com',
  databaseURL: 'https://finka-444ec.firebaseio.com',
  projectId: 'finka-444ec',
  storageBucket: 'finka-444ec.appspot.com',
  messagingSenderId: '1077142970772',
  appId: '1:1077142970772:web:a78ff850eca454f5',
};!firebase.apps.length ? firebase.initializeApp(config) : ";

//firebase.firestore().settings({ timestampsInSnapshots: true })
export default firebase;

```

## Anexo 4. Formatos

### Formato de historias de usuario

Historia de Usuario	
Número:	Nombre:
Usuario:	
Modificación de Historia Número:	Iteración Asignado:
Prioridad en Negocio: (Alta/Media/Baja)	Puntos Estimados:
Riesgo en Desarrollo: (Alto/Medio/Bajo)	Puntos Reales:
Descripción:	
Observaciones:	

### Formato de asignación de tareas

Tarea	
Número:	Numero de Historia:
Nombre:	
Tipo de tarea:	Puntos Estimados:
Fecha Inicio:	Fecha fin:
Programador Responsable:	

### Anexo 5. Formato de Encuesta y escala de Baremos

Datos de Software:	Datos Evaluador
Nombre: AFID	Nombre: Personal – Activos Fijos
Fabricante: Daniel Yeber León Sillo	
Área de reconocimiento por el uso: DTI	

	ESCALA DE PREGUNTAS	PUNTACIÓN				
CARACTERÍSTICAS	ASPECTOS	Muy en descuerdo	En descuerdo	Ni de acuerdo ni en De acuerdo	De acuerdo	Muy de acuerdo
FUNCIONALIDAD	1. El sistema AFID permite la identificación mediante una clave y usuario.					
	2. Le resulta fácil interactuar con el sistema AFID.					
	3. Se muestra información sobre los bienes de la UPeU – Juliaca en el sistema AFID.					

EFICIENCIA	5. El tiempo de respuesta del sistema AFID a sus acciones es aceptable.					
	6. El contenido abordado en el sistema AFID es confiable.					
	7. El tiempo que toma al realizar un proceso en el sistema AFID es aceptable.					
USABILIDAD	8. Sistema AFID usa vocabulario de nivel técnico adecuado.					
	9. El funcionamiento de los enlaces y/o botones no presenta inconvenientes en el sistema AFID.					
	10. No es necesario conocimiento o entrenamiento previo para utilizar el sistema AFID.					
	11. El diseño de la interfaz del sistema AFID es claro y atractivo.					
	12. El diseño de la Interfaz del sistema AFID es intuitivo.					
	13. ¿Considera que es importante la implementación del sistema AFID en Activos Fijos?					

## Anexo 6. Test: Sistema AFID como propuesta de implementación en el área de activos fijos.

### Cuestionario - Sistema AFID como Propuesta de implementación en el Area de Activos Fijos

(1) Muy en desacuerdo. (2) En desacuerdo. (3) Ni de acuerdo ni en desacuerdo. (4) De acuerdo. (5) Muy de acuerdo.

---

El sistema AFID permite la identificación mediante una clave y usuario \_\_\_\_\_ / 0

Muy en desacuerdo  
 En desacuerdo  
 Ni de acuerdo ni en desacuerdo  
 De acuerdo  
 Muy de acuerdo

## Anexo 7. Datos recolectados de la encuesta

El sistema AFID permite	Le resulta fácil interactuar	Se muestra información s	El tiempo de respuesta s	El contenido abordado en	El tiempo que toma al rea	Sistema AFID usa vocabab	El funcionamiento de los	No es necesario conocin
De acuerdo	Muy de acuerdo	Muy de acuerdo	Muy de acuerdo	De acuerdo	Muy de acuerdo	De acuerdo	Muy de acuerdo	Muy de acuerdo
De acuerdo	De acuerdo	De acuerdo		De acuerdo	Ni de acuerdo ni en desa	Ni de acuerdo ni en desa	De acuerdo	De acuerdo
Muy de acuerdo	De acuerdo	De acuerdo	Ni de acuerdo ni en desa	Muy de acuerdo	Muy de acuerdo	Muy de acuerdo	De acuerdo	De acuerdo
Muy de acuerdo	De acuerdo	De acuerdo	De acuerdo	Muy de acuerdo	De acuerdo	Muy de acuerdo	De acuerdo	Muy de acuerdo
De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	Ni de acuerdo ni en desa	De acuerdo	Ni de acuerdo ni en desa	De acuerdo
De acuerdo	Ni de acuerdo ni en desa	De acuerdo	Ni de acuerdo ni en desa	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo
De acuerdo	Ni de acuerdo ni en desa	De acuerdo	Ni de acuerdo ni en desa	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo
De acuerdo	En desacuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo
De acuerdo			En desacuerdo	En desacuerdo	En desacuerdo	En desacuerdo	En desacuerdo	En desacuerdo
De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo
Ni de acuerdo ni en desa	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo
De acuerdo	En desacuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	Muy de acuerdo	De acuerdo
De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo
	De acuerdo	Ni de acuerdo ni en desa	En desacuerdo					
De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo

### Formato de pruebas de aceptación

Caso de Prueba	
Número de Caso de Prueba:	Número Historia de Usuario:
Nombre Caso de Prueba:	
Descripción:	
Condiciones de ejecución:	
Entradas:	
Resultado esperado:	
Evaluación:	

## Anexo 5. MAPIC

VARIABLE FÁCTICA	DIMENSIONES	INDICADORES
1. Control de inventario (desde un lector de Radiofrecuencia)	1.1. Módulo ESP8266 1.2. Lector de Tags 1.3. Sistema Web	Notificaciones
VARIABLE TEMÁTICA	EJES TEMÁTICOS	SUBEJES TEMÁTICOS
2. Identificación por Radiofrecuencia. 3. Desarrollo Web.	2.1. Radiofrecuencia  3.1. Tipos de desarrollo Web.  3.2. Framework  3.3. Base de datos.	2.1.1. Identificación por radiofrecuencia. 2.1.2. Lector ESP8266 2.1.3. MFRC522 2.1.4. TAG RFID 2.1.5. Arduino 3.1.1. Desarrollo VueJs 3.1.2. Desarrollo Nuxtjs 3.3.1. Base de datos relacionales 3.3.2. Base de datos no relacionales 3.3.2.1. Firebase
VARIABLE PROPOSITIVA	EJES PROPOSITIVOS	SUBEJES PROPOSITIVOS
4. Desarrollo de un sistema de Identificación por Radio Frecuencia para el control de inventarios de Activos Fijos.	3.1. Metodología.	3.1.1. Tipo de la investigación 3.1.2. Arquitectura de solución. 3.1.3. Metodología XP.