ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA TELECOMUNICACIONES

"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA IOT PARA PAGOS Y CONTROL DE ACCESO VEHICULAR MEDIANTE LA TECNOLOGÍA NFC PARA MEJORAR EL TRÁNSITO EN LA EP-EMMPA"

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES

AUTOR:

CRISTIAN NOE SAEZ SAEZ

Riobamba – Ecuador

2023

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA TELECOMUNICACIONES

"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA IOT PARA PAGOS Y CONTROL DE ACCESO VEHICULAR MEDIANTE LA TECNOLOGÍA NFC PARA MEJORAR EL TRÁNSITO EN LA EP-EMMPA"

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES

AUTOR: CRISTIAN NOE SAEZ SAEZ

DIRECTOR: ING. DIEGO FERNANDO VELOZ CHERREZ MSc.

Riobamba – Ecuador

2023

Yo, Cristian Noe Saez Saez, declaro que el presente Trabajo de Integración Curricular es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 27 de marzo de 2023

Cristian Noe Saez Saez
0605111681

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA TELECOMUNICACIONES

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; tipo: proyecto técnico **DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA IOT PARA PAGOS Y CONTROL DE ACCESO VEHICULAR MEDIANTE LA TECNOLOGÍA NFC PARA MEJORAR EL TRÁNSITO EN LA EP-EMMPA** , realizado por el señor: **CRISTIAN NOE SAEZ SAEZ**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Dra. Silvia Mariana Haro Rivera **PRESIDENTE DEL TRIBUNAL**		2023-03-27
Ing. Diego Fernando Veloz Cherrez **DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR**		2023-03-27
Ing. Verónica Elizabeth Mora Chunllo **ASESORA DEL TRABAJO DE INTEGRACIÓN CURRICULAR**		2023-03-27

DEDICATORIA

Dedico el presente trabajo de titulación a Dios quien me guío en cada paso de este proceso, a mis padres por su esfuerzo y apoyo incondicional, a mi familia por sus consejos en cada momento.

Cristian

AGRADECIMIENTO

Agradezco a Dios quien siempre estuvo conmigo guiándome y cuidándome en cada instante, a mis padres quienes con esfuerzo, dedicación y amor me animaron para seguir adelante a pesar de las adversidades, a mis abuelitos que siempre me aconsejaron y fueron el motor para avanzar, a mis tíos por sus sabios consejos y motivación, a mis compañeros de estudio que entre alegrías y tristezas formamos vínculos grandes de amistad, a Gabriela que se convirtió en una amiga y apoyo incondicional en este proceso. A la Escuela Superior Politécnica de Chimborazo por haberme acogido en sus instalaciones y mediante sus docentes me enseñaron el sacrificio y dedicación que se necesitan para alcanzar el éxito, de manera especial a los ingenieros Diego Veloz y Verónica Mora quienes guiaron la culminación del presente trabajo de integración curricular.

Cristian

TABLA DE CONTENIDO

ÍNDICE DE TABLAS ..... **xii**
ÍNDICE DE ILUSTRACIONES ..... **xiii**
ÍNDICE DE ANEXOS ..... **xvi**
SUMMARY ..... **xviii**
INTRODUCCIÓN **1**

CAPÍTULO I

1 PROBLEMA DE INVESTIGACIÓN **2**
1.1 Problema general de investigación 2
1.2 Problemas específicos de investigación 2
1.3 Objetivos 2
1.3.1 Objetivo general ***2***
1.3.2 Objetivos específicos ***2***
1.4 Justificación 3
1.4.1 Justificación teórica ***3***
1.4.2 Justificación práctica ***4***

CAPITULO II

2 MARCO TEÓRICO **5**
2.1 Internet de las cosas 5
2.1.1 Generalidades ***5***
2.1.2 Características ***6***
2.1.3 Dispositivos ***8***
2.1.3.1 Sensores ***8***
2.1.4 Redes ***9***
2.1.5 Comunicaciones Inalámbricas ***10***
2.1.5.1 Tecnologías WPAN ***10***
2.2 Tecnología NFC 11
2.2.1 Características de NFC ***11***
2.2.2 Arquitectura de un dispositivo NFC ***12***
2.2.3 Modos de operación NFC ***13***
2.2.3.1 Modo lector/escritor ***13***
2.2.3.2 Modo Peer to Peer ***13***
2.2.3.3 Modo de emulación de tarjeta ***14***
2.2.3.4 Modo HCE ***14***
2.2.4 Modos de comunicación ***15***

2.2.4.1 NDEF *15*
2.2.4.2 Registro *15*
2.2.4.3 Estructura de mensaje NDEF *15*
2.2.4.4 LLCP *17*
2.3 Protocolos IoT 18
2.3.1 Nivel de aplicación *19*
2.3.1.1 AMQP Advanced Message Queuing Protocol *19*
2.3.1.2 COAP Protocolo de aplicaciones restringida *19*
2.3.1.3 DDS Servicio de distribución de datos *19*
2.3.1.4 MQTT *20*
2.3.1.5 OPC UA *21*
2.3.1.6 HTTP (REST/JSON) *21*
2.3.2 Nivel de transporte *22*
2.3.2.1 TPC *22*
2.3.2.2 UDP *22*
2.3.3 Nivel de red *22*
2.3.3.1 IP *22*
2.3.3.2 6LoWPAN *23*
2.3.4 Nivel de vinculo de datos *24*
2.3.4.1 LPWAN *24*
2.3.5 Comparación de protocolos IoT *26*
2.3.6 Nivel físico *27*
2.4 Arquitectura IoT 27
2.4.1 Modelo de referencia de IoT *27*
2.4.1.1 Capa de aplicación *27*
2.4.1.2 Capa de apoyo de servicios y aplicaciones *28*
2.4.1.3 Capa de red *28*
2.4.1.4 Capa de dispositivo *28*
2.4.1.5 Capacidades de gestión *29*
2.4.1.6 Capacidades de seguridad *29*
2.4.2 Arquitectura de 3 capas *29*
2.4.3 Arquitectura de 4 capas *30*
2.4.4 Arquitectura de 5 capas *30*
2.4.5 Arquitectura SOA (Arquitectura orientada a servicios) *31*
2.4.6 Arquitectura basada en API *32*
2.4.7 Gestión de la infraestructura *33*
2.4.7.1 Fog computing *33*
2.4.7.2 Cloud computing *33*
2.4.7.3 Edge computing *34*

2.5 Seguridad 34
2.5.1.1 Seguridad en NFC ***35***
2.5.1.2 Seguridad de IoT ***35***
2.5.1.3 Gestión de acceso ***36***
2.5.2 Placas de desarrollo ***36***
2.5.2.1 Arduino ***36***
2.5.2.2 Raspberry ***37***
2.5.2.3 ESP 8266 ..... ***38***
2.5.2.4 ESP 32 ..... ***39***
2.6 Estado del arte 39
2.6.1 Tecnología NFC ***39***
2.6.2 IoT ***41***

CAPÍTULO III

3 MARCO METODOLÓGICO **44**
3.1 Introducción 44
3.2 Estado actual del control de acceso vehicular en la EP-EMMPA 44
3.3 Infraestructura y arquitectura IoT 47
3.4 Desarrollo del prototipo 48
3.4.1 Aplicación móvil ***49***
3.4.2 Lector NFC ***50***
3.4.3 Interfaz web ***50***
3.4.4 Servidor/Base de datos ***50***
3.5 Requerimientos de diseño del prototipo **50**
3.6 Requerimientos de software 50
3.6.1 Software para aplicación móvil *50*
3.6.1.1 Android Studio ***51***
3.7 Requerimientos de hardware **51**
3.7.1 Módulo NFC PN532 ..... ***52***
3.7.2 Node MCU ESP 8266 ..... ***54***
3.8 Diseño lógico del prototipo **55**
3.8.1 Plataforma para el servidor ***55***
3.8.1.1 Hostinger ***56***
3.8.1.2 PHP ***57***
3.8.1.3 Dominio ***57***
3.8.2 MySQL ***59***
3.8.3 Características MySQL ***59***
3.8.4 Sentencias y Funciones MySQL ***60***
3.9 Diseño e implementación del prototipo **61**

3.9.1 Desarrollo e implementación del módulo lector de dispositivos NFC *61*
3.9.1.1 Conexión entre módulos de desarrollo *61*
3.9.1.2 Codificación de lectura de UID de NFC *62*
3.9.1.3 Conexión de Gateway y envío de datos hacia el servidor *63*
3.9.2 Desarrollo e implementación del servidor y base de datos *65*
3.9.2.1 Implementación del servidor *65*
3.9.2.2 Diseño e Implementación de la base de datos en MySQL *66*
3.9.3 Estructura de interfaz web de administrador *67*
3.9.3.1 Control de acceso al sistema de administración *67*
3.9.3.2 Interfaz para el manejo de datos de usuario *68*
3.9.3.3 Interfaz para el ingreso de usuarios que no poseen smartphones con tecnología NFC *71*
3.9.4 Estructura de la aplicación en el dispositivo inteligente *71*
3.9.4.1 Android Manifest *71*
3.9.4.2 Host APDU Service *73*
3.9.4.3 Generador código HCE *73*
3.9.4.4 Registro de usuarios *74*
3.9.4.5 Validación de usuario para inicio de sesión *74*
3.9.4.6 Registro de tarjeta de crédito para realizar el pago *74*
3.9.4.7 Registro de tarjeta de crédito para realizar el pago *75*
3.9.4.8 Visualización de registro de parqueo *75*
3.9.4.9 Pin de seguridad para aprobación de pago *76*
3.9.5 Proceso para el reconocimiento de placas vehiculares *77*

CAPITULO IV

4 RESULTADOS **80**
4.1 Módulo lector de dispositivos NFC 80
4.1.1 Registro de datos en la entrada ***80***
4.1.2 Registro de datos en la salida ***81***
4.1.3 Apertura de la barrera vehicular ***82***

CAPITULO V

5 ANÁLISIS DE RESULTADOS **89**
5.1 Latencia del prototipo **89**
5.1.1 Latencia del dominio del servidor ***89***
5.1.2 Latencia de la API de Google vision para el reconocimiento óptico de caracteres ***90***
5.1.3 Latencia generada por el protocolo MQTT ***92***
5.1.4 Latencia total al ingreso y salida del establecimiento ***92***
5.2 Parámetros de seguridad implementados **92**
5.2.1 Seguridad en la capa de transporte ***93***

5.2.2 Pruebas de autenticación de administrador ***97***
5.2.3 Pruebas de seguridad en la Aplicación móvil ***97***
5.2.3.1 Autenticación del usuario para ingresar a la aplicación móvil *97*
5.2.3.2 Verificación del pin de pago *98*
5.2.4 Seguridad física al ingreso y salida del establecimiento ***99***
5.2.5 Seguridad en el dispositivo lector NFC ***99***
5.3 Análisis de costos **100**
5.4 Comparación entre el sistema implementado actualmente y el prototipo **100**
CONCLUSIONES **102**
RECOMENDACIONES **103**
BIBLIOGRAFÍA
ANEXOS

ÍNDICE DE TABLAS

Tabla 1-2: Tipos de sensores 8
Tabla 2-2: Campos de la estructura NDEF 16
Tabla 3-2: Comparación de protocolos IoT 26
Tabla 4-2: Modelos de Arduino 37
Tabla 5-2: Modelos de Raspberry 37
Tabla 6-2: ESP 8266 38
Tabla 7-2: ESP 32 39
Tabla 1-3: Equipos y funcionamiento del sistema actual 44
Tabla 2-3: Comparación de módulos 52
Tabla 3-3: Escala de Likert de los módulos 52
Tabla 4-3: Módulo PN532 53
Tabla 5-3: Escala de Likert para las placas de desarrollo 54
Tabla 6-3: Comparación de servidores 55
Tabla 7-3: Escala de Likert para los servidores 56
Tabla 8-3: Sentencias y funciones MySql 61
Tabla 1-5: Latencia Total al ingreso y salida del establecimiento 92
Tabla 2-5: Parámetros Hello del cliente 94
Tabla 3-5: Parámetros Hello del servidor 95
Tabla 4-5: Envío de parámetros del servidor 95
Tabla 5-5: Envío de parámetros del cliente 96
Tabla 6-5: Culminación de handshake del servidor 96
Tabla 7-5: Análisis de costos 100
Tabla 8-5: Comparación entre el sistema implementado actualmente y el prototipo 101

ÍNDICE DE ILUSTRACIONES

Ilustración 1-2: Arquitectura NFC 12
Ilustración 2-2: Estructura mensaje NDEF 16
Ilustración 3-2: Protocolo NFC LLCP y modelo OSI 17
Ilustración 4-2: Formato PDU LLCP 18
Ilustración 5-2: Ejemplo de arquitectura MQTT 20
Ilustración 6-2: Ejemplo de arquitectura MQTT 27
Ilustración 7-2: Arquitectura de 3 capas 29
Ilustración 8-2: Arquitectura de 5 capas 31
Ilustración 9-2: Arquitectura basada en SOA para IoT 32
Ilustración 1-3: Interfaz de categorización de vehículos 46
Ilustración 2-3: Ticket generado al ingreso del estacionamiento 46
Ilustración 3-3: Emisión de factura 47
Ilustración 4-3: Infraestructura y arquitectura IoT 48
Ilustración 5-3: Topología del prototipo 49
Ilustración 6-3: Funcionamiento de nombres de dominio 58
Ilustración 7-3: Detalle del dominio 59
Ilustración 8-3: Certificado aplicado al dominio 59
Ilustración 9-3: Librerías y creación objeto NFC 61
Ilustración 10-3: Envío de comando APDU hacia dispositivos NFC y respuesta 62
Ilustración 11-3: Almacenamiento y verificación de UID 62
Ilustración 12-3: Librerías y parámetros de conexión a la red 63
Ilustración 13-3: Envío de datos hacia el servidor 64
Ilustración 14-3: Detalles del servidor 65
Ilustración 15-3: Credenciales para conexión FTP 66
Ilustración 16-3: Archivos subidos al servidor 66
Ilustración 17-3: Base de datos utilizadas en el prototipo 67
Ilustración 18-3: Archivo php para validación de datos de administrador 68
Ilustración 19-3: Archivo php cierre de sesión 68
Ilustración 20-3: Archivo php para visualizar los datos 69
Ilustración 21-3: Archivo php para editar los datos de los usuarios 69
Ilustración 22-3: Archivo php para eliminar usuarios 70
Ilustración 23-3: Archivo PHP para recargar saldo a usuarios 70
Ilustración 24-3: Archivo php para registrar usuarios 71
Ilustración 25-3: Codificación de Android Manifest 72

Ilustración 26-3: Servicios para la emulación de tarjeta inteligente....72
Ilustración 27-3: Servicio HOST APdu....73
Ilustración 28-3: Generación de código HCE....73
Ilustración 29-3: Registro de usuario....74
Ilustración 30-3: Validación inicio de sesión de usuario....74
Ilustración 31-3: Registro de datos de tarjetas a la base de datos....75
Ilustración 32-3: Selección de tarjeta de crédito a utilizar....75
Ilustración 33-3: Visualización de datos de parqueo....76
Ilustración 34-3: Validación de datos para aprobar el pago....76
Ilustración 35-3: Procedimiento en el caso de ser inválido algún dato....77
Ilustración 36-3: API de Google visión para detección óptico de caracteres....78
Ilustración 37-3: Tópicos establecidos para comunicación MQTT....78
Ilustración 38-3: Proceso de comparación de placas....79
Ilustración 1-4: Dispositivo lector NFC....80
Ilustración 2-4: Registro y visualización de datos al ingreso del establecimiento....81
Ilustración 3-4: Registro y visualización de datos al salir del establecimiento....82
Ilustración 4-4: Registro y visualización de datos al salir del establecimiento....83
Ilustración 5-4: Pantalla principal y sistema de login de interfaz web....83
Ilustración 6-4: Pantalla de usuarios registrados....84
Ilustración 7-4: Editar dato de usuario....84
Ilustración 8-4: Recargar usuario....85
Ilustración 9-4: Eliminar usuario....85
Ilustración 10-4: Sistema de Login para usuario....86
Ilustración 11-4: Pantalla de movimientos y balance....86
Ilustración 12-4: Pantalla editar datos de usuario....87
Ilustración 13-4: Registro de tarjetas de crédito....87
Ilustración 14-4: Lista de tarjetas....88
Ilustración 15-4: PIN de seguridad para realizar el pago....88
Ilustración 1-5: URL y método para analizar la latencia del hosting....89
Ilustración 2-5: PRTG tiempo de respuesta del hosting....90
Ilustración 3-5: Mediana de latencia API de Google vision....90
Ilustración 4-5: Reconocimiento de placas en Arduino....91
Ilustración 5-5: Placa del vehículo....91
Ilustración 6-5: Latencia protocolo MQTT....92
Ilustración 7-5: Certificado de seguridad del dominio....93
Ilustración 8-5: Certificado de seguridad del dominio. Proceso de handshake....94
Ilustración 9-5: Análisis de tráfico....97

Ilustración 10-5: Pruebas de autenticación de administrador.....97
Ilustración 11-5: Autenticación 98
Ilustración 12-5: Correo electrónico enviado en caso de falla de PIN de seguridad 98
Ilustración 13-5: Seguridad física al ingreso y salida del establecimiento 99
Ilustración 14-5: Seguridad en el lector NFC 99

ÍNDICE DE ANEXOS

ANEXO A: Código implementado en el ESP8266

ANEXO B: Conexión a la base de datos

ANEXO C: Código para Actualizar usuarios en la interfaz

ANEXO D: Android Manifest

ANEXO E: Código para realizar el reconocimiento de placas vehiculares

RESUMEN

El objetivo del presente trabajo fue diseñar e implementar un sistema de Internet de las Cosas (IoT) para pagos y control de acceso vehicular mediante la tecnología de Comunicación de Campo Cercano (NFC) para mejorar el tránsito en la Empresa Pública Municipal Mercado de Productores Agrícolas San Pedro de Riobamba (EP-EMMPA), por lo que se estudió el estado del arte de las tecnologías. Al analizar el sistema actual del control de acceso al establecimiento se apreciaron tecnologías tradicionales, sin tomar en cuenta la experiencia del usuario, generado congestión vehicular a la salida del lugar; es así que se analizó una infraestructura IoT junto a NFC que permita controlar el acceso y salida del lugar de manera autónoma y eficiente, permitiendo palear el problema del tráfico; es así que, se diseñó un prototipo con HTTP y MQTT como protocolos de comunicación, y el cloud computing, lo que permitió el manejo de datos de manera remota, mediante el diseño de una aplicación móvil, la API de Google vision con el desarrollo de un script en Python, que permitió el reconocimiento de la placa vehicular como método de autenticación. Para el análisis de la latencia se tomó en cuenta el tiempo de respuesta del servidor, el retardo de la API y del protocolo MQTT, resultando 3693 mseg a la entrada y 1640 mseg a la salida, que en comparación con el sistema actual se consideraría muy bueno, los métodos de autenticación implementados y probados generan un grado de seguridad alto ante posibles intentos de vulneración del sistema. Se concluye que el prototipo redujo un 80% los tiempos de espera del usuario en comparación con el sistema actual, y posee mecanismos de seguridad eficientes que permitirían una implementación a gran escala del control de acceso al establecimiento. Se recomienda la contratación del servicio de hosting con mayores capacidades acorde al tráfico que se va a generar.

Palabras clave: <INTERNET DE LA COSAS>, <Comunicación de Campo Cercano (NFC)>, <CONTROL DE ACCESO >, <SEGURIDAD >, <CLOUD COMPUTING>, <PROTOCOLO HTTPS>, <PROTOCOLO MQTT>.

SUMMARY

The objective of this work was to design and implement an Internet of Things (IoT) system for payments and vehicle access control using Near Field Communication (NFC) technology to improve traffic in the Municipal Public Company Mercado de Productores Agrícolas. San Pedro de Riobamba (EP-EMMPA), for which the state of the art of technologies was studied. When analyzing the current access control system to the establishment, traditional technologies were appreciated, without taking into account the user experience, generating vehicular congestion at the exit of the place; Thus, an IoT infrastructure was analyzed together with NFC that allows access to and exit from the place to be controlled autonomously and efficiently, allowing to alleviate the traffic problem; Thus, a prototype was designed with HTTP and MQTT as communication protocols, and cloud computing, which allowed data management remotely, through the design of a mobile application, the Google vision API with the development of a script in Python, which allowed the recognition of the vehicle license plate as an authentication method. For the analysis of the latency, the response time of the server, the delay of the API and the MQTT protocol were taken into account, resulting in 3693 msec at the entrance and 1640 msec at the exit, which in comparison with the current system would be considered very Well, the implemented and tested authentication methods generate a high degree of security against possible attempts to violate the system. It is concluded that the prototype reduced user waiting times by 80% compared to the current system, and has efficient security mechanisms that would allow a large-scale implementation of access control to the establishment. It is recommended to contract the hosting service with greater capacities according to the traffic that is going to be generated.

Keywords: <INTERNET OF THINGS>, <NEAR FIELD COMMUNICATION (NFC)>, <ACCESS CONTROL>, <SECURITY>, <CLOUD COMPUTING>, <HTTPS PROTOCOL>, <MQTT PROTOCOL>.

MSc. Wilson G. Rojas

C.I 0602361842

INTRODUCCIÓN

Los avances tecnológicos tienen como fin facilitar la vida de las personas y edificar empresas sustentables y eficientes, es por eso por lo que en el siguiente trabajo técnico se analiza, desarrolla e implementa un prototipo con la tecnología IoT y NFC que permitan controlar el acceso vehicular al establecimiento, implementando medidas de seguridad confiables para realizar el pago por el servicio.

El desarrollo del presente estudio se lo realiza en 5 capítulos que se detallan a continuación

El primer capítulo enfatiza el marco referencial, donde se detallan la problemática y los objetivos que generan dichos problemas que se van a cumplir en el desarrollo del prototipo, así mismo la justificación aplicativa y practica que sustentan por qué se va a realizar el estudio.

En el segundo capítulo se estudian los diferentes conceptos que van a sustentar la práctica, enfatizando los conceptos del Internet de las Cosas con sus diferentes capas y protocolos, de igual manera el estudio de la tecnología NFC con su arquitectura que permitirá el correcto análisis del protocolo de comunicación, adicional las propiedades de seguridad que puede ayudar en estos sistemas. Así mismo se analiza el estado del arte de IoT y NFC enfatizando en los materiales de construcción y cuestiones de seguridad.

El tercer capítulo analiza el estado actual del control de acceso al establecimiento, así como el análisis de una infraestructura IoT con la tecnología NFC añadiendo seguridad en sus diferentes capas con el fin de que sea eficiente para el desarrollo de esta, estableciendo protocolos como HTTP y MQTT para la comunicación, así mismo el servidor en la nube para el manejo de datos, añadiendo mecanismos de seguridad que garanticen un pago seguro.

En el cuarto capitulo se visualizan los resultados del desarrollo del prototipo, en el que se detallan todos los aspectos que se tomaron en cuenta, tanto como para el usuario, para el administrador y las formas y métodos que se aplicaron para conseguir un sistema seguro y eficiente. El quinto capitulo analiza los resultados de la implementación, tales como la latencia que genera el sistema por separado ya que se consideran diferentes estructuras para luego unirlas y analizarlas en el proceso, verificando su eficiencia en la práctica, de igual manera al considerar factores de pago, se analizan los métodos de seguridad implementados en la parte del usuario, como métodos de autenticación para ingresar a la aplicación y la implementación de un pin de seguridad para habilitar el pago.

CAPÍTULO I

1 PROBLEMA DE INVESTIGACIÓN

1.1 Problema general de investigación

¿Un sistema IoT para el control de acceso vehicular en el mercado mayorista de Riobamba permitirá una mejora en el flujo de tráfico?

1.2 Problemas específicos de investigación

¿Cuál es la situación actual de los sistemas IoT y de la tecnología NFC?
¿Qué tipo de control de acceso vehicular esta implementado en la actualidad?
¿La implementación de una infraestructura NFC e IoT mejorará el tránsito vehicular en la EP-EMMPA?
¿Cuáles son los requerimientos para un sistema de pagos en una infraestructura IoT?
¿Cuáles son la característica que debe cumplir un sistema Iot para el control?

1.3 Objetivos

1.3.1 *Objetivo general*

Diseñar e implementar un sistema IOT para pagos y control de acceso vehicular mediante la tecnología NFC para mejorar el tránsito en la EP-EMMPA

1.3.2 *Objetivos específicos*

- Estudiar el estado del arte de sistemas IoT y la tecnología inalámbrica NFC
- Examinar el estado actual del ingreso y salida de vehículos en la EP-EMMPA
- Analizar la infraestructura IoT y el sistema de pagos que se ajuste a las necesidades del control de acceso para la EP-EMMPA
- Diseñar un prototipo IoT de control de entrada y salida vehicular mediante la tecnología inalámbrica NFC, aplicando mecanismos de seguridad en el sistema de pagos.
- Implementar el prototipo según los requerimientos analizados del sistema
- Evaluar el nivel de seguridad en el sistema de pagos y parámetros de latencia en la comunicación para determinar la eficiencia del sistema

1.4 Justificación

1.4.1 *Justificación teórica*

Existe una gran acogida de dispositivos que permiten la facilidad de pago, a través de dispositivos portátiles como smartphones o tarjetas inteligentes, la mayoría de estas basadas en tecnologías inalámbricas como NFC, ha dado lugar a un crecimiento potencial en base a estos avances tecnológicos. Podemos destacar diversas áreas que necesitan pagos, como es el servicio de transporte público, el servicio de estacionamiento, ingreso a lugares como oficinas, pagos en lugares de comida, ingreso a lugares de entretenimiento y muchos usos más, pero en la actualidad se están desarrollando muchos proyectos que facilitan la vida de las personas y ayuda a optimizar el tiempo. (NFC FORUM 2023)

(Ramos 2020) enuncia que el desarrollo de nuevas tecnologías tiene por objetivo mejorar la calidad de vida de las personas en el sentido de comodidad, agilidad en realizar procesos, optimización del tiempo, etc Pero también se puede considerar beneficios para las empresas o instituciones que aplican estas tecnologías, en el caso específico de la automatización del ingreso de vehículos a cierto establecimiento como es un mercado, dado la alta densidad de tráfico que esta presenta genera muchos beneficios, a nivel usuario, gracias a IoT va a tener a su disposición la información de sus movimientos en todo el tiempo, se va a demorar menos al ingreso del establecimiento, va a evitar el contacto con otras personas y va a tener la seguridad a través de la verificación de datos, que sus movimientos son los correctos. Pero también podemos destacar un alto grado de beneficios a nivel del administrador, si bien al principio va a demandar trabajo hasta que se pueda poner el sistema a punto, hecho esto el no tendrá mayor dificultad de manejo, porque todos los componentes se van a tener a disposición y este se encargaría del monitoreo del proceso, de datos, de actividad, de finanzas todo esto de una forma más organizada y precisa.

Según Pilamunga, (2019, p. 80) a la EP-EMMPA ingresan alrededor de 17340 vehículos semanalmente, estos datos nos dan la idea la gran densidad de tráfico que se genera tanto al ingreso como a la salida del establecimiento, adicional a esto podemos ver que no existe la capacidad suficiente de estacionamientos para dicha densidad de autos, dado que cuenta con 533 espacios para poder estacionarse; un aspecto muy importante es la forma de controlar el acceso y salida de vehículos dado que el tiempo en que se demora un usuario es de aproximadamente 1 minuto, que al relacionar con la cantidad de vehículos es una atenuante para el inconveniente y de igual manera la forma de realizar el cobro por la estancia en el lugar, la intervención de personas es menos eficiente que un sistema autónomo, por lo que también es un agravante para el problema, en este contexto la implementación de un sistema autónomo va a agilitar de gran

manera el ingreso y salida del lugar, adicional a esto el que el usuario y administrador pueda estar pendiente de los movimientos que realiza, para tener un control adecuado.

En (NOTISEG 2019) vemos que el uso de la tecnología NFC complementado con IoT, permitiría dar solución en gran proporción a estos problemas ya que el usuario con un dispositivo inteligente como puede ser un smartphone o a su vez tarjetas inteligentes, pude acceder a estos servicios de manera rápida y sencilla, puesto que en el uso no demanda más de 5 segundos, dado que solo necesita realizar el contacto con el lector y el proceso siguiente depende del sistema implementado. Dado el auge del internet de las cosas, los dispositivos con NFC en la actualidad hay gran cantidad de dispositivos celulares que cuentan con la tecnología inalámbrica y en el caso de no poseer, una tarjeta inteligente puede resolver el problema.

1.4.2 *Justificación práctica*

Según Valverde, (2015) menciona en su estudio, que la implementación de acceso y los sistemas de control por medio de la tecnología NFC son más eficientes en cuanto tiempo, dinero y esfuerzo que los sistemas manuales, ya que brinda el control de entrada y salida de forma automática.

Con un sistema de control vehicular ya sea por medio de acceso remoto o por control biométrico nos da una pérdida de tiempo, que con un acceso automatizado por medio de redes de corto alcance y el pago automático que minimiza la cantidad de tiempo que se toma para realizar el registro en los que en los sistemas ya mencionados y una ventaja que menciona (Dr. Fons J, 2012) en su investigación es que se tiene una aplicación para lo cual servirá de gran ayuda al momento de pagar.

Con lo mencionado anteriormente se propone en realizar un sistema de control para la entrada y salida de vehículos por medio de smartphone, los mismo deben poseer la tecnología NFC, en caso de no tener dicha tecnología se aplicará tarjetas NFC, para la lectura de dichos dispositivos en la entrada y salida se implementará un lector NFC por medio de módulos y unas tarjetas de desarrollo para la conexión.

La información de los usuarios se almacenará en el internet y la conexión desde el lector hacia el internet se realizará por medio de las placas de desarrollo donde se tendrá la una base de datos la cual contendrá la información de los diferentes usuarios ya sea el nombre, ID, placas y su saldo respectivo. Para la consulta y la recarga de saldo se realizar por el método de prepago en un punto estratégico, lo cual estará encargado por un administrador quien tendrá acceso total a la base de datos por medio de una aplicación, por otra parte, el usuario también tendrá acceso a sus datos personales por medio de una aplicación que sería exclusivamente para los usuarios, el administrador se encargará de realizar el ingreso del saldo en la base de datos que se encuentra en el internet.

CAPITULO II

2 MARCO TEÓRICO

2.1 Internet de las cosas

2.1.1 *Generalidades*

La sociedad ha avanzado de manera exponencial en cuanto a tecnología, hace pocos años era impensable que se pueda controlar la iluminación, las persianas, el monitoreo de alimentos en un refrigerador, entre más cosas, pero de manera remota, de igual manera en campos más avanzados que el agricultor pueda controlar el proceso de regado de agua en un sembrío, o que el ingeniero pueda monitorear la calidad del aire o agua, pero desde la comodidad de un escritorio o desde su hogar, desde un dispositivo inteligente, o que incluso estos procesos pasen desapercibidos y se sigan ejecutando en forma automática, facilitan de gran manera la vida de las personas; pero cabe destacar que todo esto demanda recursos proporcionales a la aplicación que se esté implementando, en todo esto el internet juega un papel muy importante ya que es el nexo entre los dispositivos y el usuario, todo este contexto define en gran manera el internet de las cosas, ya que una persona puede tener el control de procesos sin la necesidad de presencia en el lugar.

Según la Unión Internacional de Telecomunicaciones define a IoT como la "Infraestructura mundial para la sociedad de la información que propicia la prestación de servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperabilidad de tecnologías de la información y la comunicación presentes y futuras." IoT Entre sus propiedades identifica, adquiere y procesa datos además de su capacidad de comunicación, lo que lo hace propicio para el uso de cosas con el fin de ofrecer servicios a diferentes tipos de aplicaciones, con la garantía del cumplimiento de requisitos de seguridad y privacidad; adicional puede considerarse como una noción con efectos tecnológicos y sociales (ITU 2012, p. 7)

La implementación de sistemas IoT proyecta grandes avances e importantes consecuencias para programas de salud, educación y subsistencia en países subdesarrollados, en base a esto se detallan tres pilares fundamentales en las cuales se deben fundamentar estos sistemas que se detallan a continuación

- Disponibilidad, el avance de la conectividad móvil con redes 2G con 95% desplazado a nivel mundial y 3G con 89% en zonas urbanas, da una gran proyección para poder implementar

sistemas IoT ya que los dispositivos son habituales, económicos y fáciles de reemplazar en estos mercados, lo que lo pone al alcance de la población.

- Asequibilidad, el desplazamiento de infraestructura es importante para la implementación de sistemas IoT, en países desarrollados existe gran demanda de dispositivos e infraestructura avanzada, pero en países en desarrollo sistemas básicos pueden ser utilizados para implementarse, esto brinda una base fuerte en la cual se puede seguir fomentado la tecnología IoT.
- Adaptabilidad, en muchos casos dependiendo de la aplicación muchos de los dispositivos solo necesitan conectar a energía y a internet para poder ser utilizados de manera inmediata, este proceso lo puede realizar un usuario común, lo que hace que una implementación sea práctica y sencilla, en cuanto a aplicaciones donde no haya carencia de recursos como energía eléctrica, podemos destacar que los dispositivos que recolectan información son por lo general de bajo consumo por lo que fuentes alternativas como la solar son un gran apoyo; al ser tan versátil ofrece soluciones a corto y largo plazo a ritmo de la sociedad (U.ITU 2022)

2.1.2 *Características*

Las redes IoT son distintas y con diferentes propósitos, en ocasiones privadas que actúan por encima de internet, en síntesis, utiliza una comunicación directa al internet; a medida que evolucione y afine, dispondrán de mayores niveles de seguridad, análisis y administración, haciendo de IoT más poderosa e importante, dado que se considera la primera evolución real del internet, ya que brinda una mejoría potencial en la forma en que los usuarios viven, aprenden, trabajan, etc. Dado que la información es un parámetro trascendental en una empresa u hogar, se conoce que cualquier objeto que genere o recopile datos puede ser incorporado a IoT, pero por consecuencia es blanco de vectores de ataque, por esto detalla ciertas características presentes en la tecnología (Bermejo Vera y Guedea Martin 2020, p. 26)

- Comunicación y cooperación, se forman redes entre sensores o sistemas los cuales deben estar interconectados dentro de la misma red o haciendo uso del internet, para poder utilizar datos, servicios y actualizar su estado, en este contexto la tecnología móvil como 3G, LTE o 5G, así como las tecnologías inalámbricas wi-fi, wimax, zegbee u otras que sean privadas de manera preferencial ya sea doméstica o industrial tienen un papel trascendental en redes IoT (Bermejo Vera y Guedea Martin 2020, p. 27)

- Identificación, existen tecnologías que permiten realizar este proceso por lo general son inalámbricas de corto alcance como RFID, NFC o código de barras, los mismos pueden ser

activos en el caso de un Smartphone de NFC o tags en el caso de RFID y código de barras, en estos casos puede haber información que se guarda en estos dispositivos los cuales tiene la capacidad de ser recuperados de una base de datos o la que se encuentre en el mismo dispositivo con el fin de realizar alguna acción. (Bermejo Vera y Guedea Martin 2020, p. 27)

- Direccionamiento, se pueden ubicar y dirigir objetos a través de servicios como investigación, búsqueda o nombres de dominio tales como ONS, EPC Discovery Services, DNS, con la posibilidad de ser consultados o configurados de manera remota (Bermejo Vera y Guedea Martin 2020, p. 27)
- Detección, se puede identificar esta característica a la función que puede realizar un sensor, como recopilación de información, graban, reenvían datos o reaccionan directamente sobre el dispositivo (Bermejo Vera y Guedea Martin 2020, p. 27-28)
- Actuación, esta característica hace referencia a la acción que pueden realizar los objetos, por lo general de manera remota mediante el internet, como el control de una persiana o el encendido de un foco
- Procesamiento de información integrado, después de la etapa de recolección de datos, los mismos que deben tener una capacidad técnica a través de procesos o microcontroladores con espacio de almacenamiento, (Bermejo Vera y Guedea Martin 2020, p. 28)
- Localización y rastreo, dentro de los riesgos que puede tener un sistema IoT está el rastro de información que alguna vez fue utilizada o sigue siendo manipulada, esta característica permite que los objetos conozcan la ubicación, física o geográfica a través de varios medios como la red móvil o el servicio de GPS, existe una gran variedad de aplicaciones que son implementadas en gran afluencia hoy día con muchas ventajas pero de igual manera con riesgos por ejemplo, un crecimiento en servicios basados en localización en exterior o en ambientes interiores, el usuario muchas veces es inconsciente de la información que está siendo enviada ya que esta se realiza de manera silenciosa y dado el alcance de la tecnología existe una creciente interacción con objetos que dejan algún tipo de información que fue recolectada y puede ser útil (Bermejo Vera y Guedea Martin 2020, p. 28)
- Interfaz de usuario, se refiere a las interfaces mediante las cuales el usuario pueda administrar, actuar, controlar procesos inmersos dentro de la estructura IoT, por lo general de manera gráfica a través de ordenadores, smartphones, paneles táctiles, etc, (Bermejo Vera y Guedea Martin 2020, p. 28,29)

2.1.3 *Dispositivos*

2.1.3.1 *Sensores*

Este tipo de dispositivos son trascendentes en la recolección de datos de diferentes situaciones, el desafío es implementarlos de manera trasparente en cierto modo, es así que se define a los sensores como objetos que permiten crear información a partir de la acción de dispositivos destinados a actuar mediante la conversión de una entrada no eléctrica en una señal eléctrica que puede transmitirse a un circuito electrónico, el objeto por sí solo no podrá realizar su propósito, necesita un complemento en este caso un actuador que convierte la señal eléctrica en acción. Los sensores y actuadores forman parte de los transductores más amplios, estos convierten energía eléctrica o no en otra de la misma manera; se pueden clasificar a los sensores por su fuente de energía en activos y pasivos, los dispositivos activos son los que emiten una fuente de energía propia para detectar la respuesta en torno a esa energía, un ejemplo puede ser los radares que emiten señales electromagnéticas que rebotan en objetos y son detectadas por el sistema, en cambio los dispositivos pasivos son los que reciben energía de manera externa, otra diferencia es el consumo de energía un activo demanda más recursos que un pasivo por el simple hecho de su funcionamiento, en la tabla 1-2 se detallan tipos de sensores y su descripción (Holdowsky 2015, p. 6)

Tabla 1-2: Tipos de sensores

Tipo de sensores	**Descripción**	**Ejemplos**
Posición	Mide la posición de un objeto de forma absoluta o relativa, fijos o en movimiento, estos pueden ser lineales, angulares o multieje	Potenciómetros, sensores de proximidad e inclinómetros
Ocupación y movimiento	El sensor de ocupación detecta presencia de personas o animales estén en movimiento o estáticas, en cambio los de movimiento se activan solo con el movimiento de las personas o animales	Radares, ojo eléctrico
Velocidad y aceleración	Los sensores de velocidad miden la rapidez con la que se mueve el objeto sea lineal o giratoria, en cambio los de aceleración detectan los cambios de velocidad	Giroscopio, acelerómetro
Fuerza	Detecta si se ha aplicado una fuerza física y si esta supera un umbral establecido	Medidor de fuerza, viscosímetro

Presión	Se relacionan con los sensores de fuerza porque miden este parámetro aplicado por liquido o gases	Barómetros
Flujo	Detectan el volumen o velocidad de fluidos que pasan por estos sensores en determinados tiempos	Anemómetro, piezómetro
Acústicos	Miden y convierten las señales digitales o analógicas de sonido	Micrófonos, geófono, hidrófono
Humedad	Detectan la cantidad de vapor de agua en ambientes u objetos, estos pueden ser de manera absoluta, relativa o en relación de masa	Higrómetro, sensor de humedad
Luz	Su función es detectar luz visible o invisible	Infrarrojo, Foto detector
Radiación	Sirven para medir niveles de radiación en el ambiente ya sea por centelleo o ionización	Contador Geiger-Muller, centelleador, detector de neutrones
Temperatura	Miden niveles de calor o frío presentes en el ambiente u objetos, con o sin contacto, los de contacto obligatoriamente deben estar físicamente ligados para poder medir, en cambio los que son sin contacto lo hacen por convección y radiación	Termómetro, calorímetro
Químico	Son sensores selectivos de sustancias químicas ya que miden la concentración de químicos en un sistema	Alcoholímetro, olfatómetro, sensor detector de humo
Biosensores	Muestran organismos, tejidos, células, enzimas, anticuerpos y ácidos nucleicos	Medidor de glucosa en la sangre, oxímetro, electrocardiografía

Fuente: (Holdowsky 2015, p. 7)

Realizado por: Sáez, Cristian,2022

2.1.4 *Redes*

La información recolectada por los sensores muchas veces no es suficiente, por lo general necesita procesamiento para ser utilizada, usualmente esto no se realiza en el mismo lugar donde interactuó el dispositivo, por lo que necesita comunicarse con otras ubicaciones para ser añadida ya analizada; esta comunicación se basa en la red mediante diversos dispositivos como gateways, routers, puentes de red y switch dependiendo el sistema, como primer paso para transferir datos de una maquina a otra mediante la red es la identificación única de cada una de ellas, para realizar

esto IoT utiliza protocolos que son reglas que definen como los dispositivos se identifican entre sí; estos pueden ser propietarios o abiertos, los que son de propiedad interactúan con hardware y software específicos, mientras que los de código abierto facilitan la interoperabilidad entre sistemas diferentes, haciéndolo escalable (Holdowsky 2015, p. 10)

2.1.5 *Comunicaciones Inalámbricas*

Se denomina comunicación inalámbrica a la que se realiza sin la necesidad de medios de propagación físico entre el emisor y receptor, en cambio utiliza la modulación de ondas electromagnéticas en el espacio para poder establecer la comunicación. (Federación de Enseñanza de CC. OO 2010, p. 1) Se puede considerar una clasificación de las redes inalámbricas en base a la distancia máxima que las dos partes puedan establecer la comunicación, por lo que se pueden clasificar de la siguiente manera:

- Wireless Personal Area Network (WPAN): diseñado para la interconexión de dispositivos en un área centrada en el trabajo del usuario, basado en el estándar IEEE 802.15 siendo bluetooth e infrarrojos las tecnologías más representativas (Gonzáles 2018, p. 7)
- Wireless Local Area Network (WLAN): apoyado en la interconexión de dos o más dispositivos dentro de un área limitada o local, por eso su nombre, brindando la capacidad de interconexión de dicha red dentro de un rango de cobertura como un hogar, edificio, escuela, entre otros, basado en el estándar IEEE 802.11 comercializado bajo la tecnología Wifi, en gran uso hoy en día (Gonzáles 2018, p. 7)
- Wireless Metropolitan Area Network (WMAN): Pensada mayormente en una conectividad punto multipunto, con una velocidad alta para áreas metropolitanas, basado en el estándar IEE 802.16, con la tecnología representativa WiMAX que es parecida a Wi-Fi con la diferencia de una cobertura muchísimo mayor como 50 km, desde una única estación central (Salazar. Jordi 2016, p. 14)
- Wireless Wide Area Network (WWAN): Son redes inalámbricas de alcance superior a los 50 km, tales como redes móviles o satelitales, implementados en ciudades grandes o países (Salazar. Jordi 2016, p. 15)

2.1.5.1 *Tecnologías WPAN*

- Tecnología Bluetooth: Es una estándar abierto de uso fácil y abierto, pensado para dispositivos en áreas pequeñas de área local, para aplicaciones de corto alcance, que opera en la banda de 2,4 y 2,4835 Ghz, utilizando la técnica de espectro ensanchado, con 79 canales y

ancho de banda de 1 MHz, con velocidades de transferencia de datos entre 2 y 3 Mbps (Corredor Camargo, Pedraza Martínez y Hernández 2009, p. 74)

- Tecnología RFID: es una tecnología que permite una interacción rápida de objetos con el lector para la lectura de datos de manera automática mediante radiofrecuencia, se considera como un método de almacenamiento y recuperación de estos de forma remota, a través de dispositivos denominados como tag, que son de tamaño reducido lo que lo hace versátil y accesible (Ramírez 2006, p. 23)
- Tecnología Zigbee: considerada de corto alcance en rangos de 10 a 75 metros y trasferencia de datos baja de 20 a 250 kbps, por lo que es de bajo consumo, lo que lo hace ideal para funciones de seguridad y automatización de dispositivos del hogar, como domótica, automatización industrial, medicina, etc, lo interesante de esta tecnología es su bajísimo consumo de energía ya pasan en un estado latente (Moreno y Ruiz 2007, p. 4)

2.2 Tecnología NFC

Tecnología inalámbrica que tiene como objetivo facilitar la vida y conveniente para consumidores, en base a estándares de corto alcance, que permiten realizar transacciones, interacción de contenido digital y la conexión de dispositivos mediante tags, compatibles con gran cantidad de lectores y tags, utiliza un campo de radio frecuencia en 13,56 MHz (NFC FORUM 2022b)

2.2.1 *Ca*racterísticas de NFC

Para establecer una comunicación entre 2 dispositivos se debe cumplir tres tareas
La primera es la energía que necesita para la realizar la interacción, los tags no necesitan de fuente de energía eterna ya que el campo de Radio Frecuencia proporciona dicha energía. Para la carga inalámbrica, la comunicación regula la transferencia de energía, cuando este se encuentra activo permite una transferencia de hasta 1 W; el tag recibe información mediante la modulación de la señal de Radio Frecuencia; y a su vez el dispositivo NFC recibe la información mediante la modulación de la carga (NFC FORUM 2022a)

De igual manera se destaca que una distancia promedio para que dos dispositivos NFC puedan interactuar es de 4 centímetros aproximadamente, dicha operación la realiza en la frecuencia central $fc = 13{,}56$ Mhz que es una frecuencia no licenciada, con un ancho de banda de 14 khz; también utiliza modulación ASK y la codificación Manchester con velocidades de transmisión de 106, 212 y 404 kbps, los mismos que son fijados por el dispositivo que inicia la conexión que es conocido como iniciador y el que responde es nombrado como objetivo, la comunicación puede establecerse de manera activa o pasiva de tipo half o full dúplex, definiéndose dos modos de

operación. Peer to peer que es utilizada si la tasa de transmisión es baja en el orden de los kbps y el modo de lectura/ escritura que lee y escribe 4 tipos de etiquetas que son definidas en NFC forum; adicional se puede emular una tarjeta inteligente mediante dispositivos móviles como smartphones que cuenten con esta tecnología, utilizada comúnmente para realizar pagos de forma rápida, transacciones bancarias y control de acceso (AG Electrónica S.A 2015)

2.2.2 *Arquitectura de un dispositivo NFC*

A continuación, especifica una arquitectura de alto nivel de la pila NFC:

- Proximidad de campo cercano el cual publica y suscribe el paso de mensajes de un mismo nivel cuando hay proximidad y se presenta la interacción
- Elemento seguro el mismo que permite enumerar y exponer elementos seguros que son conectados al controlador NFC que pueden ser lectores externos, permitiendo el reenvío de eventos de NFCC y applets a capas superiores, proveyendo el ingreso para configuración y administración de la configuración del enrutamiento.
- Tarjeta inteligente es de bajo nivel que transmite la solicitud al Smart tag y permite la recuperación de la información
- Administración de radio, permite el acceso al panel de control que va a establecer estados de radio de proximidad en los diferentes modos de operación de NFC (Microsoft 2022a)

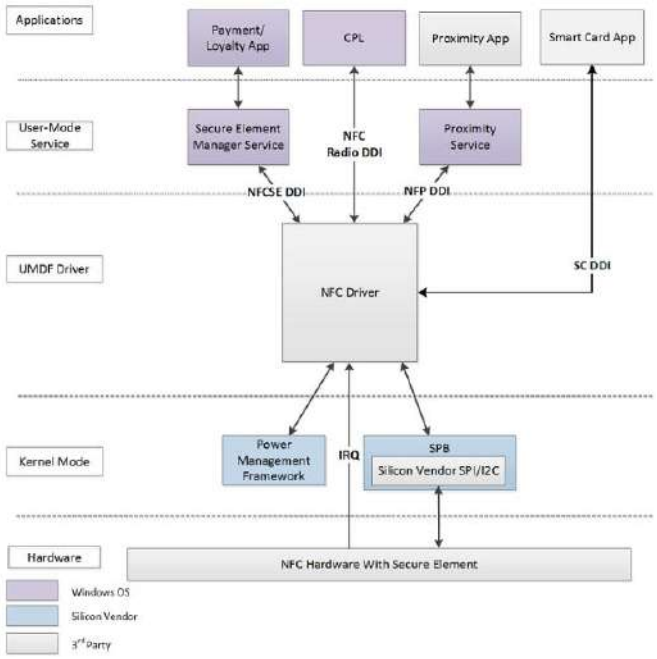


Ilustración 1-2: Arquitectura NFC

Fuente: (Microsoft 2022a)

En la Ilustración 1-2 se observan los diferentes elementos de la arquitectura NFC en la cual diferenciamos diferentes capas como: la de Aplicación, Servicio de modo usuario, UMDF Driver, modo de núcleo, y el hardware.(Microsoft 2022a)

2.2.3 *Modos de operación NFC*

Existen 3 tipos de dispositivos que interactúan para llevar a cabo el proceso NFC, un teléfono inteligente, lector y etiquetas que soporten la tecnología NFC, la comunicación se produce entre 2 dispositivos con combinaciones válidas como Lector-Tarjeta, Smartphone-Smartphone, Smartphone-Tarjeta, entre otros; estos procesos son de corto alcance por lo que es común tocar dispositivos entre sí, denominando este proceso como paradigma; dado esta interacción el usuario está en la capacidad de obtener aplicaciones como abrir página web, conectarse a servicio web, transferencia de información, etc. (Coskun, Ok y Ozdenizci 2011, p. 41)
Se consideran tres modos de operación de NFC los cuales se detallan a continuación:

2.2.3.1 *Modo lector/escritor*

El modo establece una comunicación mediante la interacción entre un dispositivo móvil NFC y una etiqueta, dentro de este modo podemos ver 2 sub-modos, modo lector y otro escritor. Dentro de este modo se considera que un teléfono móvil puede interactuar con cuatro tipos de tarjetas 1,2,3 y 4 cada uno con capacidades diferentes basados en normas ISO 14443 Tipo A, ISO 14443 Tipo B o Sony FeliCa; también tenemos otro estándar considerado de importancia el NDEF ya que permite el intercambio de información entre los 2 dispositivos; como aplicaciones de este modo tenemos la interacción con carteles inteligentes, URI, firma digital y texto (Coskun, Ok y Ozdenizci 2011, p. 42-43)

2.2.3.2 *Modo Peer to Peer*

Este modo tiene una conexión de tipo bidireccional semidúplex lo que quiere decir que cuando un dispositivo transmite el otro escucha, hasta tener espacio para transmitir. El estándar ISO/IEC 118092 define la interfaz de comunicación de radio frecuencia del modo operativo de igual a igual, de igual manera se usa el mensaje NDEF que se recibe a través del protocolo de capa de enlace de datos LLCP que permite la comunicación entre 2 pares, definiendo 5 servicios principales los cuales son: transporte orientado a la conexión y sin conexión, la activación - supervisión y desactivación de enlaces, comunicación equilibrada asíncrona y multiplexación de protocolos (Coskun, Ok y Ozdenizci 2011, p. 43)

2.2.3.3 *Modo de emulación de tarjeta*

Es un modo con más aplicabilidad ya que se define en los smartphones con tecnología NFC, dando la capacidad de emular varias tarjetas con diferentes utilidades en uno solo, el dispositivo emula un smart tag ISO 14443 o a su vez está conectado directamente a la antena del módulo NFC, utilizado comúnmente para pagos de forma segura, control de acceso, boletería, etc. Podemos destacar que solo este modo utiliza un SE de manera eficiente y funciones seguras (Coskun, Ok y Ozdenizci 2011, p. 43)

2.2.3.4 *Modo HCE*

Es la modulación de tarjeta basado en host considerado como un modo especial de la emulación de tarjetas, el servicio SE es reemplazado por un servicio que se ejecuta en segundo plano y en el sistema operativo del Smartphone con capacidad de administrar APDU, la ventaja de este modo es que accede a funciones avanzadas del smatphone y su tiempo de vida es igual a la de una aplicación habitual, lo que la hace más sencilla sin interfaces de terceros (Lesas, Anne y Miranda 2017, p. 144-145)

- ARQUITECTURA HCE

Se basa en los servicios que son componentes de Android, por las cuales tienen la capacidad de ejecutarse en segundo plano, las cuales ayudarían a aplicaciones de autenticación, tránsito u otras en las que no sea estrictamente necesario el ingreso a la interfaz de la aplicación, adicional se puede implementar notificaciones en el caso de ser necesarias, al tener la capacidad de ser implementado en cualquier aplicación, este modo mediante la especificación ISO/IEC 7816-4 permite elegir la que realmente necesita, basado en un AID que es una ID de aplicación la cual se forma de 16 bytes, que ya es conocido para redes de pago conocida, en el caso de ser una aplicación nueve esta debe ser definida y registrada. (Developers 2019)

- SEGURIDAD EN EMULACIÓN DE TARJETAS BASADAS EN HOST (HCE)

La seguridad radica en que la arquitectura brinda un parámetro básico pero efectivo, donde solo el sistema operativo está en la capacidad de vincularse y comunicarse, esto funciona gracias a que el APDU que se recibe sea uno también los hizo el sistema operativo desde el controlador NFC y en el caso de transmisión se envíe solo al sistema operativo, el que reenviará al controlador, adicional brinda seguridad en el transporte, ya que cuando se envía la información de la aplicación hacia el lector se asegura de que llegue hacia el controlador y el lector NFC. (Google Developers 2019)

2.2.4 *Modos de comunicación*

Considera dos tipos de modos el Pasivo y Activo definidos en los estándares NFC IP-1 (ISO 18092) e IP2 (ISO 21481)

- Modo de comunicación Pasiva, el que inicia la interacción lo hace a través de un campo de radio frecuencia que envía comandos mediante modulación, y las respuestas por modulación de carga
- Modo de comunicación Activa, en este caso los dispositivos que se van a emparejar generan los campos propios para enviar comandos, uno a la vez, deteniéndose por completo para que el otro dispositivo pueda modular y enviar sus respuestas (Paret 2016, p. 15-16)

2.2.4.1 *NDEF*

Se conoce como NDEF al formato de datos que tiene como fin el intercambio de información entre dispositivos ya sean un activo / etiqueta pasiva o dos dispositivos NFC activos; este mensaje se intercambia cuando hay cercanía entre los dispositivos y el mismo se recibe desde la etiqueta o del protocolo control de enlace lógico LLCP. También se puede destacar que encapsula una o más cargas útiles en un solo mensaje ya que es formato de mensaje binario. Se destaca que cada mensaje NDEF contiene uno o varios registros NDEF, cada uno de los cuales posee una carga útil de hasta 2^{32} -1 octetos de tamaño, dicho esto podemos decir que el número máximo de registros NDEF que se pueden transmitir se limita(Coskun 2013, p. 38)

2.2.4.2 *Registro*

Se define a registro como el transporte de una carga útil dentro de un mensaje a través de una unidad, cada registro NDEF describe su carga útil mediante tres parámetros:

- Longitud de carga útil, que denota el número total de octetos
- Identificador de tipo de carga útil, indica el tipo de carga que permite el envío de la misma a la aplicación adecuada, NDEF admite identificadores de tipo URI, MIME (construcciones de tipos de medios) y NFC (formato de tipo específico)
- Identificador de carga útil opcional, cuya función es identificar la carga útil que se transporta dentro de un registro NDEF.(Coskun 2013, p. 39)

2.2.4.3 *Estructura de mensaje NDEF*

Dentro de la estructura, el primer registro es marcado con el conjunto que indican el mensaje inicial (MB), y el último es marcado con el conjunto que indican el mensaje final (ME), se puede configurar los indicadores MB y ME en un mismo registro para definir la longitud mínima de un mensaje que es un registro. La lectura del encabezado se realiza de cabeza a cola de izquierda a derecha, en la ilustración 2-2 se muestra el índice 1 que nota al primer conjunto de banderas MB y el índice t nota al último conjunto de bandera ME.(Coskun 2013, p. 39)

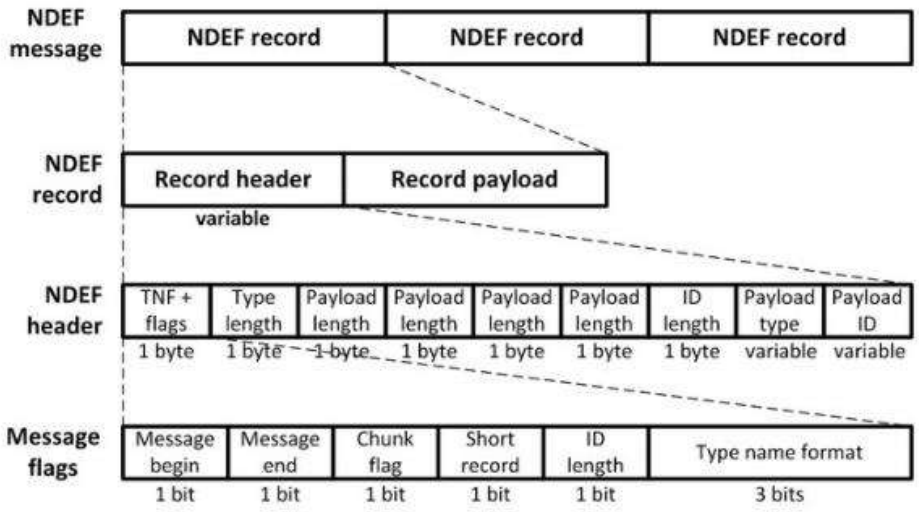


Ilustración 2-2: Estructura mensaje NDEF

Fuente: (Boada et al. 2021, p. 5)

Los registros tienen un formato común y son de longitud variable, en la Tabla 2-2 se puede apreciar campos individuales cada uno con características

Tabla 2-2: Campos de la estructura NDEF

Campo	Descripción
MB Mensaje incial	Indica el inicio del mensaje NDEF con longitud de 1 bit
ME Mensaje Final	Indica el final del mensaje NDEF con longitud de 1 bit
CF Indicador fragmento	Muestra que es el fragmento del primer o registro intermedio de carga con longitud de 1 bit
SR Registro Corto	Indica que PAYLOAD_LENGTH es un solo octeto con longitud de 1 bit
IL	Indica que el campo ID_LENGTH está en el encabezado con un solo octeto con longitud de 1 bit
TNF Formato de tipo de nombre	Muestra la estructura del valor del campo TIPO con longitud de 3 bits

TYPE_LENGTH	Indica la longitud en octetos del campo TYPE con longitud de 8 bits sin signo, este campo siempre es cero para ciertos valores de TNF
ID_LENGTH	Muestra la longitud de ID en octetos con longitud de 8 bits
PAYLOAD LENGTH	Muestra la longitud del PAYLOAD, con tamaño determinado por la bandera SR
TYPE	Describe el tipo de la carga útil
ID	Identifica en forma de referencia URI
PAYLOAD	Especifica la carga útil de NDEF

Fuente (Coskun 2013, p. 40)

Realizado por: Sáez, Cristian, 2022

2.2.4.4 *LLCP*

Es un protocolo de enlace de datos del modelo OSI que permite la comunicación punto a punto, trascendental para aplicaciones que necesiten comunicación bidireccional; se puede destacar que garantiza un base sólida y mejora funcionalidades del protocolo NFCIP-1 el cual brinda capacidad de segmentación y reensamblaje, flujo de datos, adicional el manejo de errores a través de reconocimiento ACK y NACK, facilita una capa de enlace confiable y libre de errores.(Coskun, Ok y Ozdenizci 2011, p. 46). Se basa en la capa 2 del estándar IEE 802.2 que se observa en la ilustración 3-2

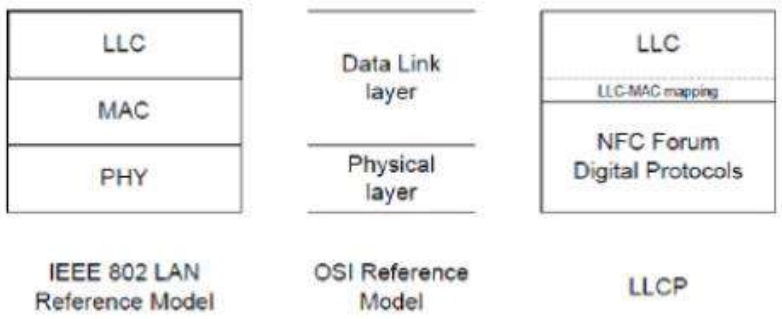


Ilustración 3-2: Protocolo NFC LLCP y modelo OSI

Fuente: (Lesas, Anne y Miranda 2017, p. 36)

En la Ilustración 4-2 se pude apreciar el formato del PDU de LLCP en la cual se detalla los siguientes campos:

- DSAP: indica el punto de acceso al servicio de destino con longitud de 6 bits
- PTYPE: tipo de carga útil con longitud de 4 bits
- SSAP: punto de acceso al servicio de origen con longitud de 6 bits
- PDU SEQUENCE: considera una longitud de 0 u 8 bits
- Payload: Carga útil netamente con longitudes M*8 bits (Lesas, Anne y Miranda 2017, p. 37)

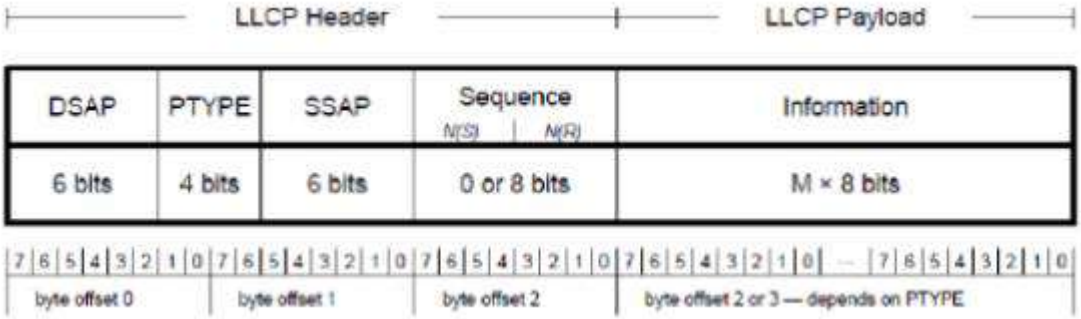


Ilustración 4-2: Formato PDU LLCP

Fuente (Lesas, Anne-Marie. y Miranda 2017, p. 37)

En la Ilustración 4-2 se puede distinguir dos partes, la cabecera LLCP con una longitud de 16 o 24 bits y el Payload con una longitud de M*bits

2.3 Protocolos IoT

Uno de los desafíos de los sistemas IoT es la interoperabilidad de los sistemas por lo cual existen los protocolos propios de cada función, sistema, tecnología, etc; mediante los cuales se trata de que sistemas distintos se puedan conectar en el caso de que así lo dispongan, al tomar en cuenta los sistemas IoT sabemos que en casos básicos existe mínimo 2 protocolos que deben coexistir, en aplicaciones más grandes como industria, se conoce que deben coexistir varios tipos de protocolos propios o complementarios de IoT. Se puede clasificar o agrupar los protocolos en 2 categorías, cliente-servidor y publicar-suscribir. (Aron Semle 2016, p. 32)

En el modo cliente-servidor el cliente realiza peticiones o consultas hacia el servidor, al conectarse el servidor envía en respuesta los pedidos que el cliente necesita, estos protocolos funcionan de mejor manera cuando se conoce la infraestructura o es propia, al estar basado en conexiones punto a punto, los hace más compatibles y seguros, pero al mismo tiempo representa una desventaja que son complicados de implementarse a gran escala ya que demandan mayor complejidad en el manejo y mayores recursos. En el caso de los protocolos publicar-suscribir los objetos se deben conectar a un gestor que trabaje como intermediario el cual publique la información, en este caso los clientes tienen la posibilidad de conectarse al gestor y suscribirse a los datos, como ejemplo se puede tomar un sensor que recopile datos en forma periódica y lo publique en cierto horario, el cliente podrá acceder a esta información cuando esta esté disponible, estos protocolos son mejores cuando no se tiene conocimiento de la infraestructura, como ventaja se destaca que son más escalables ya que permite agregación o eliminación independiente de productores o

consumidores, en cambio como desventaja es más complejo asegurar tantos dispositivos.(Aron Semle 2016, p. 32)

Dado la extensión de los sistemas IoT y de sus protocolos se detallan a continuación algunos agrupados de acuerdo con el nivel de capas mediante el modelo OSI

2.3.1 *Nivel de aplicación*

2.3.1.1 *AMQP Advanced Message Queuing Protocol*

Se considera un protocolo que publica y suscribe, surgido en el sector financiero ya que permite transacciones completas, porque posee un modelo de comunicaciones robusto lo que le lleva a gran demanda de recursos (Aron Semle 2016, p. 35) también se considera como un protocolo de almacenamiento y reenvío ya que encola mensajes y brinda servicios como, encolamiento, enrutamiento, fiabilidad y seguridad, por lo que se considera un protocolo middleware orientado a mensajes, a través del protocolo TCP (González et al. 2020, p. 118)

2.3.1.2 *COAP Protocolo de aplicaciones restringida*

Es un protocolo que fue creado para establecer compatibilidad de HTTP con carga mínima, además de tener similitud con la diferencia de que utiliza UDP, simplificando el encabezado y reduciendo el tamaño de cada requerimiento, COAP es utilizado cuando HTTP demanda muchos recursos por lo general en el borde, es utilizado por sistemas IoT como un protocolo de prioridad, después de HTTP y MQTT (Aron Semle 2016, p. 35).También conocida por usar nodos restringidos del mismo modo, de manera general constituido por un microcontrolador de 8 bits con Rom y Ram limitadas, pensado para sensores que no demanden mucha potencia y aplicaciones M2M. Entra en el grupo cliente-servidor entre la aplicación y puntos finales, entre sus principales características se puede mencionar que; soporte tipo de contenidos, descubrimiento de recursos, suscripción de recursos, almacenamiento simple en caché (Cornejo, Godoy y Roca 2018, p. 3)

2.3.1.3 *DDS Servicio de distribución de datos*

Este protocolo se basa en conectividad cendrada de datos de baja latencia, muy confiable y arquitectura escalable, lo que lo hace óptimo para aplicaciones industriales. La idea de centralidad se basa en la comprensión de datos que controla y administra para poder compartirlos, la programación de un middleware centralizado en datos debe especificar como y cuando se comparten los datos para poder hacerlo directamente, en lugar de realizar todo este proceso DDS implementa el intercambio de datos controlado, administrado y seguro de manera directa. Este

protocolo ha sido probado en aplicaciones reales en el campo industrial por lo que cada vez es más confiable y segura una implementación masiva donde exista gran cantidad de información en tiempo real, procesando y actuando de manera rápida sobre los eventos, el resultado son nuevos servicios, ingresos adicionales y menores costos. DDS ofrece las siguientes ventajas: una integración fácil, un rendimiento eficiente, permite escalabilidad, sistemas seguros, estándar abierto, QoS, compatibilidad amplia y varias aplicaciones (OMG 2017, p. 2)

2.3.1.4 *MQTT*

Este es un protocolo del grupo publicar-suscribir diseñado para objetos ligeros M2M, diseñado como cliente-servidor orientado a mensajes donde los sensores son los clientes que se conectan a un bróker que es el servidor mediante el protocolo TCP, cada uno de estos mensajes son conocidos como tópicos los mismo que son publicados a una dirección, teniendo la capacidad de acceder a varios tópicos mientras hayan sido suscritos, en la ilustración 5-2 se observa un ejemplo de la arquitectura MQTT (Bliznakoff del Valle 2014, p. 49)

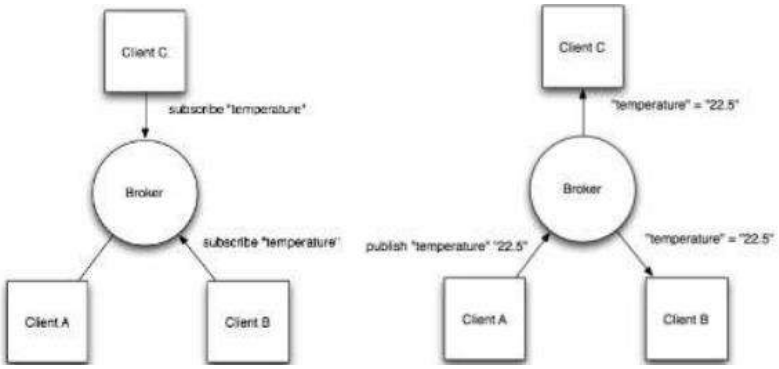


Ilustración 5-2: Ejemplo de arquitectura MQTT

Fuente (Bliznakoff del Valle 2014, p. 49)

La arquitectura permite establecer diferentes tipos de comunicación de 1 a 1, de varios a 1 y 1 a varios, de esta manera podemos detallar las siguientes características:

- Selección de tópicos, estos son de tipo jerárquico y permiten una suscripción a canales, en el caso de hacerlo a un directorio se utiliza el operador + y en caso de hacerlo a otro se usa #
- QoS a nivel de aplicación, el protocolo MQTT se establece en tres niveles que pueden ser utilizados de acuerdo con necesidades propias del sistema los cuales son: modo disparar-olvidar sin garantizar QoS; entregar al menos 1 vez como lo dice se debe garantizar la entrega de mensajes mínimo 1 vez; el entregar solo 1 vez este caso es estricto a entregar el mensaje solo 1 vez por lo cual la red exige que se reduzca el over-head

- Ultima voluntad y testamento, este tipo de mensajes se envía cuando un dispositivo se ha desconectado, esto se realiza mediante un mensaje de despedida que se envía hacia el servidor
- Persistencia, el protocolo brinda soporte para el almacenamiento de mensajes en el servidor, así los clientes tienen la capacidad de decir que los mensajes se mantengan para que esté disponible para los próximos clientes, se recalca que el mensaje que se guarda es el último
- Seguridad, en los servidores se puede configurar un usuario y contraseña para poder autenticar a los clientes y la conexión TCP se puede encriptar mediante SSL y TLS
- MQTT-SN como todo protocolo tiene debilidades este no es la excepción, esto más por los objetos que van a interactúa, ya que algunos no tienen la potencia suficiente para implementar TCP o a su vez existe pérdida de paquetes elevada, por lo que se ha diseñado un mapeado UDP añadiendo en el bróker soporte para que se pueda indexar los topics(Bliznakoff del Valle 2014, p. 50)

2.3.1.5 *OPC UA*

Conocido como arquitectura unificada, es el estándar de nueva generación de OPC, es uno de los más utilizados en la industria ya que brinda una interfaz para conectarse con los PCL, se define como un protocolo cliente-servidor ya que el usuario conecta, navega, escribe y lee en los equipos industriales, este protocolo tiene como fin brindar compatibilidad de OPC al nivel de dispositivos y empresas. Este protocolo comunica la capa de aplicación con la capa de transporte, lo que brinda compatibilidad entre vendedores, en cuanto a seguridad podemos decir que es muy confiable ya que usa mensajes bidireccionales firmados y el transporte es encriptado, se considera muy bueno para conectar sensores y PCL con aplicaciones industriales ya existentes como MES y SCADA, si bien brinda muchas ventajas tiene como contra que es muy complejo por lo que su programación se hace en casos complicadas para usuarios nuevos, por lo que plataformas IoT y la comunidad de código abierto no lo adoptan de manera masiva.(Aron Semle 2016, p. 33)

2.3.1.6 *HTTP (REST/JSON)*

Entra en el grupo de cliente-servidor sin conexión, vigente en las TIC y web, de código abierto y con numerosas librerías, ideal para enviar grandes cantidades de información como datos de sensores que necesitan transferirse cada minuto, pero no para aplicaciones que se deban monitorear cada segundo o milisegundos y flujo de datos de video, en cuanto a seguridad es importante aplicar protocolos de criptografía como SSL/TSL que corra sobre HTTP, aunque un método más seguro sería incluir en el objeto IoT solamente un cliente no el servidor, de tal manera que el objeto pueda iniciar conexión hacia el servidor pero no reciba solicitudes para conectarse (Porro 2019a)

2.3.2 *Nivel de transporte*

2.3.2.1 *TPC*

Considerado como un protocolo confiable orientado a conexión mediante el flujo de bytes, esta confianza se define en la etapa previa a conectarse y posterior a desconectarse; se define como segmento a la unidad de información, cuando se emite se espera mediante un temporizador al establecimiento para el otro extremo, si este tiempo caduca TCP reenvía la información hasta que el otro extremo envía una notificación de asentamiento, estos segmento pueden llegar desordenados dado que son datagramas IP, el protocolo los reordena en el destino, en el caso de duplicación se elimina el repetido, adicional se define un control de flujo cuando un emisor es rápido pero el receptor es lento, al inicio se debe establecer la conexión, para lo cual se establecen mensajes Sync que pide sincronización y ACK que permite reconocimiento, dicho esto se puede decir que una conexión TCP se define por un cuarteto, la ip y puerto de origen y la ip y puerto de destino, se encuentran 65536 puertos en cada ordenado los mismo que son utilizados para enviar y recibir información de todos estos se eligen de acuerdo a criterios de diseño, entre los más conocidos o utilizados tenemos a los siguientes: ftp data 20/tcp, ftp 21/tcp, telnet 23/tcp, smtp 25/tcp, domain 53/tcp/udp, http 80/tcp, pop3 110/tcp, (Anzar 2004, p. 34-35)

2.3.2.2 *UDP*

Este es un protocolo que no es orientado a conexión, ya que se emite un datagrama udp único, es por esto que no garantiza secuencia, tampoco fiabilidad por lo que los datagramas pueden ser incorrectos o se pueden perder, también no posee un sistema de control de flujo, a pesar de eso se siguen utilizando ya que las redes actuales son más confiables y garantizan menos pérdidas, es muy utilizado en aplicaciones donde se requiera el flujo de datos en tiempo real como streaming, video en vivo, audio en directo, etc, dentro de estos protocolos tenemos a SNMP que gestiona la red, BOTTP que es arranque remoto, DNS etiqueta de dominios, RPC la ejecución de proceso remotos (Anzar 2004, p. 36)

2.3.3 *Nivel de red*

2.3.3.1 *IP*

Es considerado dentro de internet como el principal protocolo de red, por ende, es el más utilizado para el envío y recepción de datagramas a través del borde de la red, actualmente existen 2 versiones IPv4 e IPv6, cada una con características diferentes, actualmente la más utilizada es la versión 4 con 5 rangos disponibles, clase A,B,C,D, y E las 3 primeras las más utilizadas, el

contraste notable entre las 2 versiones es la cantidad direcciones que proporcionan, ipv4 brinda 4300 millones que están casi agotadas, por lo que ipv6 dispone de 85000 billones de direcciones a las cuales se está migrando (Madakam, Ramaswamy y Tripathi 2015, p. 170)

2.3.3.2 *6LoWPAN*

Se define como redes de área personal inalámbrica de bajo consumo, principalmente utilizado para aplicaciones de objetos que necesitan un consumo reducido de potencia, por esta razón se adapta al estándar 802.15.4 de IEEE que son objetos con rango, tasa de bits, potencia y costo bajos, por lo general limitados en potencia, memoria y disponibilidad, entre las características podemos destacar las siguientes:

- Tamaño reducido de paquetes, la trama máxima es de 102 octetos adicionando sobrecarga en el caso de que se adicione seguridad siendo 21 octetos el máximo en el caso de AES-128, dejando 81 octetos para datos
- Compatibilidad, 16 bits para medios cortos o 64 bits para medios extendidos
- Ancho de banda reducido, velocidades de datos de 250 kbps para 2,4 GHz, 40 kbps para 915 MHz, 20 kbps para 866 MHz
- Topología, en estrella y malla
- Potencia baja, demanda de recursos reducida por lo general los objetos funcionan con baterías
- Bajo costo, los objetos que se manejan son sensores, interruptores que nos son de costo elevado
- Escalabilidad, se espera una implementación a gran escala por las características antes mencionadas
- Ubicación no predefinida, por lo general se implementa en modo ad-hoc por lo que en ocasiones es difícil de ubicarlos ya que se pueden trasladar de ubicación
- Poca confiabilidad, al ser de recursos reducidos trae consigo algunas desventajas como radio de conectividad, duración de baterías, bloqueo de dispositivos, manipulación involuntaria, etc lo que la hace no tan confiable
- Periodos de inactividad, al tratar de optimizar la energía algunos dispositivos entran en un modo de reposo, periodo en los cuales no se enviará información (Montenegro, Schumacher y Kushalnagar 2007)

2.3.4 *Nivel de vinculo de datos*

2.3.4.1 *LPWAN*

Son redes de área amplia y baja potencia, práctico e ideal para aplicaciones donde los sensores no demandan mayor consumo de potencia, aplicado de manera correcta podría reemplazar a redes móviles como 3G o 4G, por su consumo mínimo permite ahorro de energía, potencia y recursos económicos, haciendo referencia a las redes móviles el ancho de banda de144 kbps que brinda 2G es más que suficiente para aplicaciones con tecnología LPWAN. Esta tecnología trabaja dependiendo la región en la cual va a ser implementado, en Europa se utiliza el espacio de 867 y 869 MHz, en Estados Unidos 902 y 928 MHz en Asia varía dependiendo el país por ejemplo en Japón se usa 920 y 925 MHz, también se puede destacar la distancia en la que pueden conectarse estos dispositivos, alcanzando hasta los 800 km, en síntesis podemos destacar 3 características importantes, el gran radio de alcance para poder conectarse de manera inalámbrica, consumo de energía reducido, la regulación de transporte de datos de flujo de datos pequeños en el rango de 0,3 kbps a 50 kbps por canal, a pesar de parecer una ventaja también sería una desventaja ya que en caso de tratar de transferir mayor volumen de datos no sería factible. (Sampaulo 2021, p. 1-5)

Dentro de las redes LPWAN existes protocolos que se detallan a continuación que con el paso del tiempo se espera que se adopte como un estándar para soluciones IoT

- LTE-M

Este protocolo de comunicación celular utiliza la rede LTE, mediante sus antenas instaladas y optimiza un ancho de banda mayor de hasta 1 Mbps, utilizado por dispositivos que necesiten periodos de vida útil largo y bajo consumo de potencia, el ancho de banda permite mayor velocidad de transferencia de datos, latencia menor y ubicación de los objetos más exactos, entre sus características se conoce que VoLTE permite transferencia de datos de vos, dado que se basa en la red móvil es ideal para aplicaciones en movimiento, se podría expandir la batería hasta 10 años dependiendo el uso, dado que ofrece un ancho de banda mayor permite la transferencia de contenido multimedia como imágenes, datos y audio, como desventaja se puede acotar que el alcance de conectividad depende del proveedor de servicio de telefonía. En campos aplicativos tiene alcance a conexión de vehículos, objetos portátiles, rastreo y sistemas de alarma, en síntesis, sería ideal para proyectos que no requieran transferencia alta de velocidad si no que necesitan duración de baterías muy largas y conectividad en lugares donde la red móvil llegaría con facilidad como interiores, edificios, subterráneos.(Sampaulo 2021, p. 5-6)

- NARROWBAND (NB-IoT)

Se funda en LTE para comunicación, modifica el software de las antenas, así como un servicio que sea comercial, en esencia tiene como fin proporcionar conectividad de largo alcance y consumo de potencia bajo, al operar bajo LTE supone ventajas grandes como el aprovechamiento de las capacidades de la red en cuanto a hardware y software, lo que supone coberturas mejoradas, costos menores y consumo de potencia reducido. Trabaja en la banda de 790 Y 862 MHz, enfocado en mantener la simplicidad de la interfaz de comunicación, costos reducidos en fabricación de dispositivos y ahorro de energía, en cuanto a estructura se ha eliminado elementos de LTE que para dar el servicio de IoT no son trascendentales como Handover que no permite reconexión automática en modo conectado. Está fundada en EPS redefinido por 3GPP, las cuales detallan 2 optimizaciones enfocada en IoT celular, esquema de usuario y control, las 2 con el fin de buscar camino óptimo para trasferencia de datos. Esta tecnología utiliza el protocolo UDP como transporte, ya que transfiere datos a través de puerto definido ambas maquinas, dicho esto no puede utilizar protocolos de TCP, como un mensaje MQTT, en cambio puede utilizar COAP que es fundamentado en UDP, en cuanto a detalles técnicos se menciona que utiliza modulación QPSK, las frecuencias de LTE, ancho de banda de 200khz, data rate de 200 kbps, comunicación bidireccional, longitud máxima de carga útil de 1600 bytes, con alcance en entornos urbanos de 1 km y 10km en entornos rurales, encriptación propia de LTE. (Rosado 2019)

- LoRaWAN

La capa física trabaja bajo LoRa transmitiendo datos y al añadir un protocolo MAC que configura red de dispositivos se denomina LoRaWAN, LoRa trabaja con la modulación de espectro ensanchado Chrip Spread Spectrum, la cual produce una señal de chrip que cambia de frecuencia de fo y f1 en un tiempo T, definiendo 2 tipos up-chirp donde la frecuencia aumenta desde un mínimo a un máximo y down-chirp donde la frecuencia va desde un máximo a mínimo, siendo más robusto a interferencias, la cantidad de chirp se determina con la fórmula 2^{SF} para representar un símbolo, donde el spreading factor SF que puede tomar valores entre 7 y 12, el valor de SF es proporcional al SNR, sensibilidad de transmisión y Time on Air de la transmisión, en consecuencia un valor alto de SF disminuye la cantidad de información, consumir mayor potencia, pero aumenta el radio de transmisión. La frecuencia en la cual opera LoRa es 433, 866 y 915 MHz dependiendo de la norma, en lo que respecta a la arquitectura utiliza topología estrella para ser escalable en una red masiva, cada objeto o sensor se enlaza a un Gateway que a su vez se comunica con el servidor, de igual manera se definen 3 clases para cada terminal, la clase A con comunicación dual es la definida por defecto y soportada, es la más utilizada ya que inicia la comunicación, la clase B receptan mensajes en periodos de tiempo asignados y sincronizados y la clase C que inicia la comunicación en tiempos elegibles lo que tiene como consecuencia mayor

consumo de potencia (Carrasco 2020, p. 6-7). La ventaja más importante es que en aplicaciones la implementación se la puede realizar con infraestructura propia ya que la tecnología posee dispositivos propios, y otro punto importante es el alcance de radio de hasta 15 km.

- SIGFOX

Es una de las redes más utilizada y grande del mundo en la banda de 868 o 902 MHz sin licencia, con características como: ancho de banda ultra estrecha, cobertura amplia y tasa de transferencia baja, plataforma que envía 12 y recibe 8 bytes, aparenta ser una tasa muy baja pero para e transferencia de datos de temperatura, geolocalización, sensores es suficiente, trata de que todo sea bajo en costo consumo de potencia y ancho de banda, se destaca características como; comunicación no bidireccional, frecuencia baja, radio de hasta 50 km en zonas rurales, envío de mensajes cada 30 minutos, es de carácter propietario por lo que es obligatorio su contratación, requiere menos potencia porque no utiliza circuitos en el receptor, como ventajas podemos destacar su alcance, bajo consumo de energía, tiempo de batería útil extendido, costo de servicios bajos 0,83 centavos de dólar por dispositivo, como desventaja se pude acotar que no tiene cobertura en muchos países por lo que se dificultaría la implementación, también la comunicación que no es bidireccional y menor ancho de banda (Sampaulo 2021, p. 11-14)

2.3.5 *Comparación de protocolos IoT*

A continuación, la Tabla 3-2 realiza una comparativa de la compatibilidad de los protocolos que ayudarán en el estudio del prototipo

Tabla 3-2: Comparación de protocolos IoT

Protocolo	Compatibilidad	Formato	Arquitectura	Seguridad
AMQP	MQTT, HTTP	pesado	Publicación/suscripción	TLS/SSL
COAP	MQTT, HTTP, AMQP	Liviano	Petición/respuesta Publicación/suscripción	DTLS
DDS	HTTP	Liviano	Publicación/suscripción Petición/respuesta	TLS/DTLS/DDS
MQTT	AMQP	Liviano	Publicación/suscripción	TLS/SSL
OPC UA	MQTT	Liviano	Cliente/servidor	Nativa
HTTP	COAP	Pesado	Petición/respuesta Cliente/servidor	SSL/TLS

Fuente: (Inchaurza 2018, p. 24)(krypton Solid 2022) (Kepware 2016)(CEPRA 2022)(versys 2022) (Porro 2019b)
Realizado por: Sáez, Cristian, 2022

2.3.6 *Nivel físico*

En este nivel se especifica el canal de comunicación entre dispositivos de un entorno físico, dentro de las cuales tenemos distintos protocolos alámbricos e inalámbricos, entre los medio alámbricos tenemos Ethernet que puede ser menos costosa, transferencia de datos rápida con baja latencia, la limitante sería que no es aplicable a distancias grandes, PLC (power line communication) esta tecnología utiliza los cables eléctricos para transferir datos, además de alimentar y controla el dispositivo, en cuanto a medios inalámbricos podemos destacar la red móvil LTE que es de banda ancha configurado inicialmente para dispositivos móviles pero que puede ser aplicado a IoT y también tenemos a tecnologías como Bluetooth, NFC, RFID, WIFI, Zigbee, entre otras que fueron detalladas con anterioridad (Microsoft 2022b)

2.4 Arquitectura IoT

2.4.1 *Modelo de referencia de IoT*

Según la Union Internacional de Telecomunicaciones define cuatro capas con capacidades de gestión y seguridad las cuales se detallan en la ilustración 6-2.

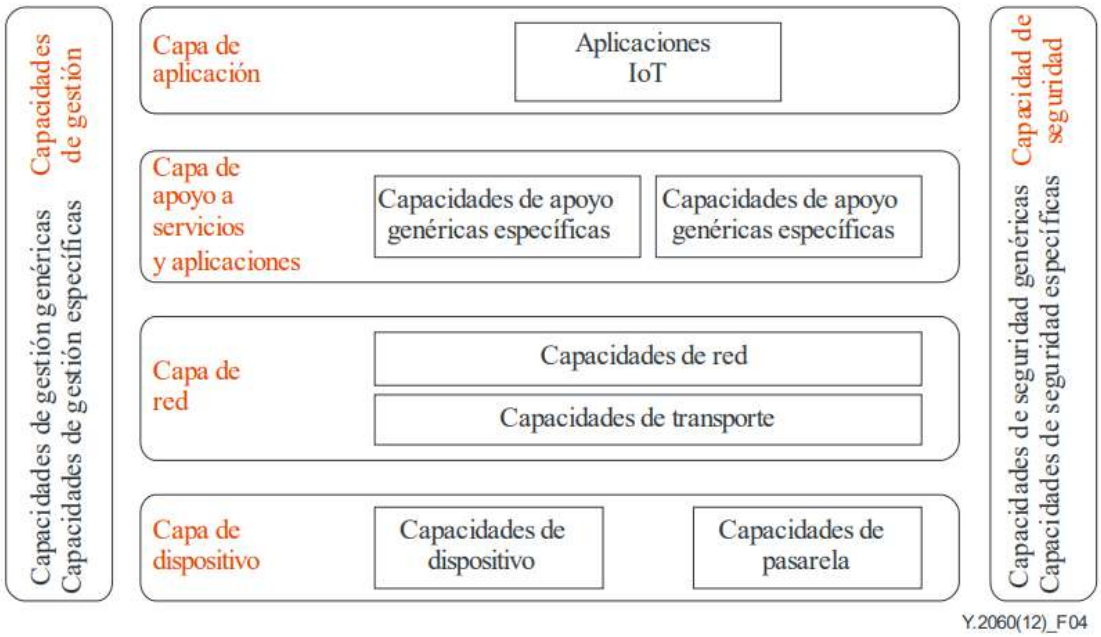


Ilustración 6-2: Ejemplo de arquitectura MQTT

Fuente: (ITU 2012, p. 13)

2.4.1.1 *Capa de aplicación*

En esta capa se encuentran las aplicaciones IoT

2.4.1.2 *Capa de apoyo de servicios y aplicaciones*

Se agrupa en 2 grupos de acuerdo con las capacidades las cuales son:

- Capacidad de soporte genéricas, son capacidades como procesamiento o almacenamiento de datos que pueden ser utilizados por aplicaciones IoT por lo general son comunes, o pueden ser específicas para crear otras del mismo tipo
- Capacidades de soporte específicas, atienden aplicaciones específicas, estas pueden ser grupos de capacidades exactas que brindan diversas funciones de apoyo a diferentes aplicaciones (ITU 2012, p. 14)

2.4.1.3 *Capa de red*

Se basa en 2 tipos de capacidades

- Capacidad de red, controlan la conexión en la red, funciones como control de acceso y de recursos de transporte, gestión de movilidad, y funciones de AAA autenticación, autorización y contabilidad
- Capacidad de transporte, tiene prioridad para transportar información y datos específicos de servicios y aplicaciones, también de transporte de información que permite control y gestión (ITU 2012, p. 14)

2.4.1.4 *Capa de dispositivo*

Se define en 2 tipos de capacidades

- Capacidades de dispositivos, son las siguientes: interacción directa, con la red de comunicación en esto los objetos pueden obtener y subir información de forma directa y de la misma manera puede recibir la información de forma directa de la red; en interacción indirecta, se obtiene y envía información de manera indirecta a la red de comunicación haciendo uso de capacidades de pasarela, y de igual manera lo hace para la recepción; redes ad-hoc en ciertas aplicaciones se pueden construir redes de este tipo para aumentar capacidades y velocidades de despliegue; modo reposo y activo, los objetos deben tener la capacidad de pasar de estado de reposo a activo con el fin de optimizar recursos energéticos

- Capacidad de pasarela, son las siguientes: soporte de interfaces múltiples, en este caso los objetos soportan conexión de diferentes tecnologías alámbricas e inalámbricas mediante capacidades de pasarela, en la capa de red pueden comunicarse con diferentes tecnologías como la red móvil; conversión de protocolo, cuando en la capa de dispositivos se utiliza

diferentes protocolos, y otra es cuando interactúan la capa de red y de dispositivo con protocolos distintos (ITU 2012, p. 14)

2.4.1.5 *Capacidades de gestión*

En esta se analizan los fallos, configuración, contabilidad, rendimiento y seguridad, pueden clasificarse en: genéricas como gestión de dispositivos, diagnóstico, actualizaciones, gestión de trabajo, de topología de red, gestión de tráfico y congestión; el otro grupo es capacidades de gestión especifica que se relacionan con aplicaciones determinadas (ITU 2012, p. 15)

2.4.1.6 *Capacidades de seguridad*

Se agrupan en capacidades genéricas y específicas. Las genéricas son independiente de la aplicación y son: autorización, autenticación, confidencialidad, protección de integridad, protección de privacidad, auditorias de seguridad y antivirus todo esto en la capa de aplicación; en la capa de red se determina autorización, confidencialidad e integridad; en la capa de dispositivos autentificación, autorización, e integridad, control de acceso, confidencialidad. En el caso de capacidades especificas es para aplicaciones determinadas (ITU 2012, p. 16)

2.4.2 *Arquitectura de 3 capas*

Consta de 3 capas, la de percepción la que se encarga de captar y recolectar información mediante sensores o actuadores, esto mediante la detección de parámetros físicos o identificación de objetos en el medio; la capa de red es la encargada de conectar objetos inteligentes con servidores y dispositivos de red; la tercera capa es la de aplicación, que se enlaza directamente con el usuario mediante aplicaciones específicas como hogares, ciudades, salud inteligente, la Ilustración 7-2 detalla esta arquitectura que se muestra a continuación.(Mouha 2021, p. 80)

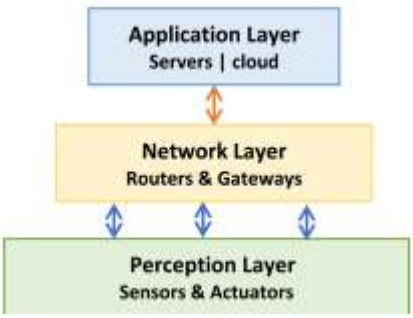


Ilustración 7-2: Arquitectura de 3 capas

Fuente (Mouha 2021, p. 80)

Básicamente esta arquitectura detalla en cada una de sus capas un modelo simple en el cual intervienen sensores y actuadores para recolectar datos, los mismo que son transmitidos mediante gateways o routers hacia servidores donde será tratada de acuerdo con las aplicaciones que se requieran, resultados que podrán ser visualizados o modificados por el usuario

2.4.3 *Arquitectura de 4 capas*

En esta arquitectura se presentan 4 etapas, la primera en la cual se obtiene la información mediante sensores y actuadores, desde la perspectiva de inmediatez se nota mayor o menor procesamiento en esta o capas cercanas; la segunda etapa detalla la puerta de enlace la cual se encarga de la conversión de los datos recolectados por los sensores a flujos digitales para ser procesados, por lo general se encuentran cerca de los sensores y actuadores; a continuación se tiene una etapa Edge IT que es el procesamiento antes de ingresar al centro de datos, dicho proceso se realiza por lo general donde se encuentran los sensores, esta etapa es importante en el sentido de que se alivia el manejo de ancho de banda y recursos en el centro de datos; por último se tiene el centro de datos y la nube, en el cual la información tiene un procesamiento más profundo los mismos que son enviados a un centro de datos físico o en la nube, los cuales analizan, administran y almacenan de manera segura estos datos. (Fuller 2016)

2.4.4 *Arquitectura de 5 capas*

Esta es una arquitectura más compleja, en esta se definen conceptos completos para comprender el funcionamiento y desarrollo de dispositivos, las 5 capas son las siguientes, las capas de percepción y aplicación tiene el mismo funcionamiento que el modelo de 3 capas, aparece la capa de procesamiento que procesa los datos obtenidos por los sensores en la capa de red con el fin de tomar decisiones mediante el análisis de computador, también interviene la capa de transporte que se encarga de transferir la información obtenida por los sensores hacia la etapa de procesamiento y de la misma manera el proceso inverso, esto a través de redes inalámbricas como LTE, RFID, NFC, etc, por último se aprecia la capa de negocios que analiza datos estadísticos de la capa de aplicación con el fin de planificar metas y futuras estrategias, (Mouha 2021, p. 81) estas capas podemos observarlas en la Ilustración 8-2.

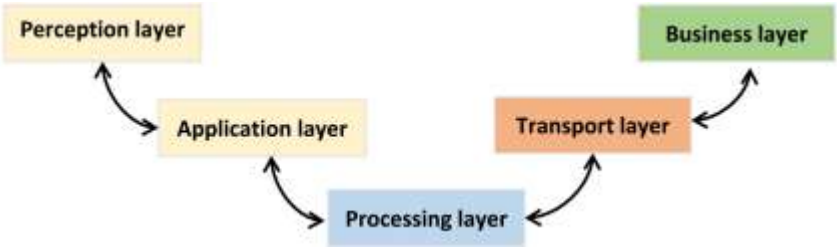


Ilustración 8-2: Arquitectura de 5 capas

Fuente (Mouha 2021, p. 81)

2.4.5 *Arquitectura SOA (Arquitectura orientada a servicios)*

Se define como un servicio a la autonomía de una unidad u otras funciones del software que realizan una tarea específica, integrando datos y códigos necesarios para cumplir una función, la arquitectura que se orienta a servicios permite utilizar varias veces los elementos, porque utilizan interfaces de servicios que se comunican mediante red de lenguaje común, pudiendo interactuar o actualizarlo de forma remota, resumiendo integran softwares, elementos que se han manejado de manera separada permitiendo interoperabilidad entre ellos para que puedan operar en conjunto(RED HAT 2020)

El acoplamiento es débil por lo que los servicios se demandan sin complejidad, estos se exponen mediante protocolos de red como SOAP, HTTP, JSON/HTTP, para enviar, leer o cambiar información, estos se publican de manera que sean fáciles, rápidos y reutilizables para otras aplicaciones. Dichos servicios se pueden originar desde cero, pero por lo general se generan exponiendo funciones de sistemas existentes. Esta arquitectura tiene algunas ventajas como: agilidad del negocio, comercialización rápida, aprovechamiento de funcionalidad existente en mercados nuevos, colaboración mejorada entre negocio y TI (IBM Cloud Education 2019)
Dentro de la arquitectura de SOA para sistemas IoT, garantizan interconexión y operación de objetos permitiendo establecer conexión entre las partes físicas y el mundo virtual, se conforma de 4 capas que se observan en la Ilustración 9-2.

Ilustración 9-2: Arquitectura basada en SOA para IoT

Fuente: (Sosa, Tello y Lara 2016, p. 3)

Las 4 capas de la arquitectura son las siguientes:

- Capa de objetos, son los sensores, tarjetas inteligentes y todo tipo de dispositivos hardware que captan datos del medio y el intercambio de datos entre dispositivos, las cosas se identifican con una identidad digital denominado UUID (identificador único universal) que permite el rastreo de la cosa en el dominio digital
- Capa de red, es la infraestructura en la que forjan las conexiones cableadas, inalámbricas o móviles hacia la capa de servicios o entre dispositivos, esta capa es fundamental en entornos IoT ya que brinda QoS, gestión de energía en la red y objetos, procesamiento de señales y datos, seguridad o privacidad
- Capa de Servicios, basada en la tecnología de middleware donde se crean y gestionan los servicios que peticionan los usuarios, la que es fundamental para consumo de servicios y ejecución de aplicaciones
- Capa de aplicaciones, es la capa que se encarga de entregar las aplicaciones entre los diferentes usuarios de IoT. (Sosa, Tello y Lara 2016, p. 4)

2.4.6 *Arquitectura basada en API*

En si esta arquitectura se centra en la conexión de servicios entre sí, una API significa una interfaz de programación de aplicaciones que contiene subrutinas, funciones y procesos que brinda alguna biblioteca la misma que puede ser usada por otro software como capa de percepción, básicamente se considera el nexo para que aplicaciones se puedan conectar entre si con el fin de obtener fuentes de datos, servicios u otras aplicaciones de corporaciones como Google, Facebook, Twitter, entre otras, sin que el lenguaje en el cual fueron desarrollados sea un problema ya que la API interactúa con un formato común que puede ser JSON o XML.

Dentro de esta arquitectura juega un papel importante REST que es un tipo de arquitectura de desarrollo web fundado en HTTP, el cual ayuda en la creación de servicios y aplicaciones que puedan ser utilizados por dispositivos o clientes que puedan concebir HTTP, también tenemos restricciones en REST que son: la del cliente-servidor la cual las mantiene débilmente acoplados, por lo que el cliente no necesita necesariamente saber de la implementación del servidor y este a su vez se desinteresa del uso de datos que el genera, sin estado que dice que el mantener sesiones no es necesario, cacheable, la implementación de caches en varios niveles para no repetir conexiones, interfaz uniforme dice que los recursos de servicio REST deben tener una dirección única denominada URI de manera individual, el sistema de capas para brindar escalabilidad, rendimiento y seguridad. La arquitectura se basa en funciones como la creación, lectura, actualización y eliminación de datos y operaciones GET, POST, PUT y DELETE las cuales consultan y leen, crean, edita y eliminan respectivamente. (Crespo 2019)

2.4.7 *Gestión de la infraestructura*

Una de las partes más importantes o donde se realiza la mayor parte del procesamiento se considera en los centros de datos los cuales pueden ubicarse en diferentes niveles, de acuerdo al nivel de proceso que necesiten los datos, en la cual tenemos el fog, cloud y edge computing

2.4.7.1 *Fog computing*

Es un proceso intermedio en el cual los datos que se generan o son recolectados por los sensores no se cargan directamente en el servicio de la nube si no que tienen una etapa previa de procesamiento o preparación en centros de datos pequeños que se manejan de manera descentralizada; estas instancias se acumulan en los fog nodes que se encargan de este procesamiento previo en partes pequeñas hasta acumular un todo, así mismo ayuda a decidir qué datos se dirigen a un procesamiento local o cuales se envían a la nube o centro de datos; el propósito del fog computing es minimizar la comunicación entre la nube y los objetos; de igual manera se pueden analizar 3 capas las cuales son: el Edge layer que es la capa de borde donde se encuentran los objetos inteligentes, los mismo que generan datos los cuales van a a ser procesados en el mismo dispositivo o pueden ser enviados a los nodos; el Fog layer es la capa donde se procesan los datos recolectados mediante un conjunto de servidores de mayor eficacia, loos cuales en caso de ser necesario se envían a la nube, el Cloud layer que se considera como la capa final de la arquitectura (IONOS 2019)

2.4.7.2 *Cloud computing*

Tiene como finalidad que los usuarios tengan disponibilidad y accesibilidad a recursos y aplicaciones en tiempo real, sin la necesidad de tener implementado una infraestructura ni hardware interno, esto gracias a que las empresas que brindan estos servicios alojan sus servidores en línea, entendiendo de esa manera a las cosas que pueden obtenerse de manera remota a través del servicio de internet. Brindando algunas ventajas como flexibilidad, ya que los trabajadores tienen la facilidad de acceder a los datos desde cualquier lugar donde tengan internet; eficiencia, varios usuarios tienen la posibilidad de acceder hacia un mismo recurso y trabajarlo de manera conjunta; reducción de costos, el mantenimiento de servidores físicos demanda mayores recursos que los virtuales; escalabilidad, el almacenamiento depende de las necesidades de la empresa sin que esta sea un limitante ya que los proveedores están en la capacidad de expandirlo en caso de ser necesario, acceso a actualizaciones, depende del servicio contratado y del proveedor este mantendrá el servicio actualizado garantizando la seguridad de la información (McKenna 2021)

2.4.7.3 *Edge computing*

Es una etapa con arquitectura distribuida que procesan datos cuando estos son recopilados, con propósito claro el menorar el ancho de banda y el tiempo de respuesta de procesos IoT, en este contexto se desea minimizar la latencia para evitar posibles saturaciones de la red cuando existe un procesamiento alto en infraestructuras, relacionado de gran manera con el fog computing la cual utiliza objetos de borde para lograr un procesamiento computacional elevado, almacenamiento y comunicación, esto mediante los fog nodes que dicen si se debe procesar los datos de forma local o deben ser enviados a la nube. Se puede destacar 3 elementos básicos en Edge Computing, la fuente de datos, que es la recopilación de la información mediante diferentes medios como sensores, la inteligencia artificial, es el procesamiento de la información en el cual se deben analizar prácticas, localizar patrones y manejar tendencias, para brindar recomendaciones, aprendizaje autónomo y análisis de datos mediante modelos; como tercera etapa tenemos una información práctica, la cual es la actuación y selección informada de los datos que fueron procesados en etapas anteriores (Alrowaily y Lu 2018, p. 441)

2.5 Seguridad

Se define a la seguridad como un conjunto de herramientas diseñadas para la protección de información, sistemas o redes a un acceso, uso, divulgación, modificación, o destrucción de entes no autorizados (Soriano 2013, p. 7)

- Integridad: la información que se transmite y la que se recibe posee fidelidad y debe ser enviados por un ente autorizado.

- Disponibilidad: hace referencia a que la información debe permanecer disponible en cualquier momento al ente autorizado
- Privacidad: Protección de la información para que solo pueda ser accedida por entes autorizadas.(Calderón 2015)
- Control: prevención de uso no autorizado de la información
- Autenticidad: verificación que el ente que se comunica es quien dice ser. La autenticación evita la suplantación de identidad
- No Repudio: protección contra las negaciones de una de las entidades en alguna parte de la comunicación, es por esto, que el no repudio utiliza evidencias.(Soriano 2013, p. 37)
- Auditoría: monitoreo constante de los servicios de los cuales se recopila y analiza información. Esto ayuda a verificar que los métodos de autorización y autenticación cumplen con las normas establecidas por la institución (Calderón 2015)

2.5.1.1 *Seguridad en NFC*

La tecnología NFC está basada en comunicación por radiofrecuencia, esta puede ser interceptada por otras personas en el medio; sin embargo, esto se puede contrarrestar por los métodos simples y sencillos de autenticación que se emplean en la tecnología ya que la interacción de los dispositivos de be ser a una distancia menor a 20cm lo que hace segura al momento de una posible intervención para robar información, dado que el atacante debería aproximarse lo suficientemente cerca para lograr su cometido lo cual es muy arriesgado incluso para él ya que el usuario se daría cuenta de la acción que quiera cometer(Bricio y Chisag 2019, p. 15)

2.5.1.2 *Seguridad de IoT*

Los dispositivos que se conectan tienen un amplio número de posibles áreas y patrones de interacción los cuales se consideran al momento de proporcionar un marco para proteger el acceso digital a los dispositivos, defiendo este término en (Tapia 2022, p. 2,3). Para optimizar los procedimientos de seguridad, se recomienda dividir las arquitecturas de IoT en zonas como parte del ejercicio de modelado de riesgos las cuales incluyen:

- Dispositivo: define como la parte física más cercano que envuelve al dispositivo.
- Puerta de enlace de campo: dispositivo o software de equipo servidor de uso general, que actúa como habilitador de comunicaciones.
- Puertas de enlace en la nube: sistema que permite la comunicación desde y hacia las cosas o gateways de varios sitios de manera remota, mediante la red pública

- Zona de servicios: son componentes de software o módulos que tienen la capacidad de interactuar con cosas a través de puertas de enlace para recopilar y analizar los datos. (Lozano y Alaberto 2016, p. 5,6)
- Ejecuciones en ambiente seguro (TEE) las Ejecuciones en ambiente seguro (TEE) se dan cuando existe un área segura en el procesador del dispositivo móvil o en su coprocesador en donde puede procesarse la información y almacenarse
- Tokenización Es el proceso en el cual se sustituye un valor al azar por una credencial de valor alto, creando un valor equivalente bajo, es usada para enmascarar la identidad de una tarjeta y solo puede utilizarse una vez ya que es generado al momento de recibir información en la tarjeta. La Tokenización es un mecanismo que en la actualidad es necesario para proteger la identidad y las credenciales de pago contra fraudes y falsificación (Suarez 2018, p. 28)

2.5.1.3 *Gestión de acceso*

- OAuth 2.0.- Considerado como un framework de autorización que permite al usuario compartir la información de un punto A considerado el proveedor de servicio a un punto B que es el consumidor, en el cual el consumidor o usuario limita recursos para no exponer sus datos confidenciales, se considera una interacción con información protegida sin compartir datos de contraseñas, esta utiliza tokens de autorización para probar una identidad entre los consumidores y proveedores de servicios. Podemos destacar que permite flujos de autorización para aplicaciones web, de escritorio y móviles, utilizado para compartir información sobre sus cuentas con terceros de los usuarios(Valdivieso 2019, p. 11)

2.5.2 ***Placas de desarrollo***

También denominadas computadora de placa única (SBC) ya que es un ordenador fundamentado sobre una placa de circuito único, con elementos como: microprocesador, memoria, pines de entrada y salida y opciones básicas de una computadora, el desarrollo de estas placas tienen como fin aplicaciones demostrativas, educativas o controladores computacionales integrados, existe una gran variedad y cantidad de estos dispositivos pero se destacan los más conocidos y utilizados en el medio (Solectro 2022)

2.5.2.1 *Arduino*

Se considera como una plataforma en la cual podemos crear electrónica de código abierto, ya que el hardware y software son libres y fáciles de utilizar, con la capacidad de crear diferentes tipos

de microordenadores de placa única para diversos usos, basada en micro contralores ATMEL, en los cuales se pueden grabar instrucciones mediante lenguajes de programación compatibles con Arduino IDE, con diferentes tipos y características como en la Tabla 4-2, Arduino:

Tabla 4-2: Modelos de Arduino

ARDUINO								
Características	UNO		MEGA		NANO		LEONARDO	
	Valor	Unidad	Valor	Unidad	Valor	Unidad	Valor	Unidad
Velocidad reloj	16	MHz	16	MHz	16	MHz	16	MHz
Memoria	32	KB Flash	256	KB Flash	32	KB Flash	32	KB Flash
Ram	2	KB	8	KB	2	KB	2.5	KB
Eeprom	1	KB	4	KB	1	KB	1	KB
Pines	20		70		22		32	
	14 digitales 6 analógicos		54 digitales 16 analógicos		14 digitales. 8 analógicos		20 Digitales 12Analógicos	

Fuente: (Suárez 2022)

Realizado por: Sáez, Cristian, 2022

2.5.2.2 *Raspberry*

Se considera como un ordenador de tamaño reducido, flexible, simple y de alta compatibilidad, con una placa única de circuito integrado impreso, con la capacidad de cargar un sistema operativo básico y utilizarlo como PC de bajo consumo, se tiene diferentes modelos con características iguales como 8 x GPIO, SPI, I²C y UART, y características diferentes entre los cuales se destacan los siguientes Raspberry como en la Tabla 5-2:

Tabla 5-2: Modelos de Raspberry

Raspberry			
Características	**Pi 1**	**Pi 2**	**Pi3**
RAM	512 Mb	1Gb	1 Gb
Puerto ethernet	10/100 Mbits	10/100 Mbits	10/100 Mbits
PINES	26 GPIO	17 GPIO	40 GPIO
CPU	ARM1 700Mhz	ARM Cortex A7, 4 núcleos 900Mhz	ARMv8 4 núcleos 1.2GHz de 64 bits

SOC Broadcom	BCM2835	BCM2836	BCM287
salidas de video	RCA, HDMI, y DSI	HDMI 1.4	HDMI 1.4
Puerto USB	2	4	4
Almacenamiento	microSD	microSD	microSD

Fuente: (Llamas 2017)

Realizado por: Sáez, Cristian, 2022

2.5.2.3 *ESP 8266*

Se define como un chip integrado de conectividad WiFi compatible con TCP/IP, con el objetivo principal de dar acceso a la red a cualquier microcontrolador compatible, la plataforma permite el desarrollo de aplicaciones en lenguajes como Lua, MicroPython, C/C++, Scratch, entre las características más importantes se puede mencionar en la Tabla 6-2:

Tabla 6-2: ESP 8266

Características/equipo ESP 8266		
Características	Valor	unidad
Frecuencia de Reloj	80/160	MHz
RAM:	32	KB
Data RAM:	96	KB
Memoria Flash Externa:	4	MB
CPU: Tensilica Xtensa LX3	32	Bit
Pin Analógico ADC:	1(0-1)	V
Placa	NodeMCU v2 (Amica)	
Chip conversor USB-serial:	CP2102	
Pines Digitales GPIO:	17 (4 como PWM a 3.3V)	
Puerto Serial UART:	2	
Antena	PCB	
Estándar	802.11 b/g/n	
Stack de Protocolo	TCP/IP integrado	
Protocolo inalámbrico	Wi-Fi Direct (P2P)	

Fuente:(Naylamp Mechatronics 2021)

Realizado por: Sáez, Cristian, 2022

2.5.2.4 *ESP 32*

Tiene bastante similitud con el esp8266, pero con más capacidades y potencialidades con las características observadas en la Tabla 7-2:

Tabla 7-2: ESP 32

Características/equipo ESP 32		
Características	Valor	unidad
Procesador	240	MHz
SRAM	520	KB
voltaje de funcionamiento	2.2-3-6	V
Velocidad de datos máxima	150	Mbps
802.11 n (2.4 GHz)	150	Mbps
Bluetooth	modo clásico y BLE	
Estandar	Wi-fi 802.11 b/g/n/e/i	
P2P Descubrimiento	Propietario Del Grupo modo de Administración De Energía	

Fuente: (Prometec 2022)

Realizado por: Sáez, Cristian, 2022

2.6 Estado del arte

2.6.1 *Tecnología NFC*

El crecimiento del uso de dispositivos tecnológicos para diferentes aplicaciones trae consigo grandes beneficios pero a su vez demanda mecanismos que protejan el bien principal que manejan la información, uno de los campos en los que existe gran demanda es el pago mediante dispositivos portátiles, ligeros y seguros, es así que (Yang et al., 2022) define la solución y una de las más utilizadas son las transacciones que se pueden realizar mediante teléfonos inteligentes que posean la tecnología NFC, la cual permite emular tarjetas de crédito mediante el uso de códigos de respuesta rápida, esto gracias a los grandes saltos en cuanto a capacidad que han desarrollado para los smartphones. Para el desarrollo de estos procesos se basan en el protocolo EMV (Europa

MasterCard Visa) que presenta algunos inconvenientes de seguridad, tales como que: un atacante pueda realizar el ataque de retransmisión a distancia, para minimizar estos problemas el autor propone un protocolo compatible con los EMV con la capacidad de autenticarse mutuamente y ambiental en dispositivos móviles inteligentes con NFC, para asegurar que la transacción sea legítima estableciendo claves para proteger los mensajes que vengan después, evitando que ataques man in the middle, skimming y clonación de tarjetas cumplan su propósito, los factores ambientales sirven para verificar la ubicación de ambas partes de la transacción, para que sea correcta deben estar en el mismo entorno lo que evita ataques de retransmisión .

El gran avance tecnológico demanda la fabricación de dispositivos portables para usos de comunicación inalámbricas hasta objetos dentro de IoT, pero a pesar de estar en auge estos todavía son costosos, rígidos y de tamaño amplio, a pesar de esto los dispositivos NFC se destacan por ser de bajo costo, fácil de fabricar con varias aplicaciones, al ser capaces de transferir señales inalámbricas e identificación por frecuencia de radio, es por esto que (Sun et al., 2022) estudia la composición de sensores con características diferentes a las habituales. Dentro de estas características se mencionan: el parileno-C como material de fabricación gracias a que es muy flexible, delgados y fáciles de cortar, demostrando alta estabilidad bajo flexión o tensión, por lo cual se puede recoger y laminar en superficies 3D como la piel sin desprenderse; en cuanto a consumo de energía es un aspecto muy importante y se destaca como una de las grandes ventajas es que no necesitan de fuentes de energía externa como batería para funcionar, destacando 2 modos semi pasivos y pasivos los cuales pueden recolectar energía para funcionar de campos magnéticos generados por vibración, luz, sonido, calor, cambios de temperatura, lo que lo hace potencialmente aplicables; en cuanto al diseño de la antena tomando el reto de que debe ser imperceptible en el lugar donde es instalado, todos estos aspectos son reunidos en la fabricación la cual se realiza sobre sustratos flexibles siendo los procesos litográficos el principal método para hacerlo pero con la desventaja de ser muy caros, pero puede ser reemplazado por otras tecnologías como la impresión por inyección de tinta.

Entre las aplicaciones podemos destacar los sensores portátiles para aplicaciones médicas las cuales tienen como fin el crecimiento económico y una calidad de vida mejor, entre estas podemos destacar el monitoreo de señales biofísicas y bioquímicas, entre las biofísicas tenemos el pulso, frecuencia cardiaca, temperatura, entre otros por lo que los sensores NFC se centran su investigación en el monitoreo de estos parámetros, casos específicos como el monitoreo de bebés prematuros es un ejemplo claro ya que requieren sensores para poder recopilar datos los cuales pueden ser implantados en su frágil piel, en cambio el monitoreo de señales bioquímicas como las secreciones del cuerpo como sudor también son monitorearles.

La energía juega un papel muy importante en los dispositivos electrónicos en el caso de los sensores al ser la tendencia diseñarlos cada vez más pequeños, la carga de los mismos representa un problema ya que se hace difícil integrar conectores para poder cargarlos o implementar algún tipo de batería, es así que la carga de manera inalámbrica es una solución rentable y viable, en el caso de NFC ya que la comunicación se activa cuando esta es necesaria no necesariamente necesitan de una fuente externa ya que algunos casos funciona con el campo magnético del lector, pero en otros modos de operación como semi pasivo o activo, necesita alguna fuente que pueden ser obtenidas durante el tiempo en la cual está interactuando con el receptor, en este contexto (Buchmeier et al., 2021) estudia el diseño de un circuito de bobina NFC optimizado para la transferencia de energía de manera inalámbrica con posicionamiento libre 2D y sensibilidad de carga baja, mediante la cual se busca que los dispositivos no se limiten a la comunicación sino también a la carga, esto a través de procesos óptimos con los cuales se calculen las dimensiones del transmisor y receptor, para facilitar y automatizar el diseño de bobinas en forma rectangular contrario a las establecidas en forma cuadrada.

NFC se ha conocido como una tecnología que permite la comunicación en campo cercano mediante bobinas que interactúan cuando sus campos entran funcionamiento, estos campos generan energía la cual pudiera ser utilizada para otras aplicaciones adicionales como la carga inalámbrica. Principalmente utilizado para cargar dispositivos de baja potencia como auriculares, lápices ópticos, rastreadores de actividad física entre otros con capacidad de hasta 1 watt hasta 2 cm de distancia, con un campo negociado de radio frecuencia con 4 clases de transferencia de potencia en el rango de 250,500,750 y 1000 milivatios, caracterizada por antenas de alrededor de 10 cm^2 o menos permitiendo una fácil integración y potencia adecuado para dispositivos pequeños. Se basa básicamente en un transmisor denominado Poller y un receptor denominado Listener (STMicroelectronics 2022; NFC FORUM 2022c)

2.6.2 *IoT*

Dentro de los sistemas IoT los sensores juegan un papel muy importante ya que son los encargados de recopilar datos que van a ser procesados con algún fin, dicho esto es importante destacar que dependiendo de la magnitud de la implementación estos pueden ir de unidades hasta centenas o miles, si bien son pequeños y no consumen mucha potencia en la mayoría de casos, afectan de alguna manera al medio ambiente ya que al ser masivos y funcionar con baterías al momento de terminar su vida útil estos deben ser desechados, lo que generan desechos que perjudican al entorno.

En este contexto, en el campo de la agricultura (Gopalakrishnan et al., 2022), describe el desarrollo de un sensor biodegradable sin el uso de chip con el fin de monitorear de manera inalámbrica la salud del subsuelo, este sensor e inteligente de transmisión de radio que ayuda en la detención de remota de niveles de agua volumétrica del subsuelo, el dispositivo en si cuenta con una antena muy pequeña, simple que se encuentra encapsulada en material polimétrico biodegradable que resuena en base a las propiedades dieléctricas del suelo que lo rodea, siendo esta señal detectada y asistida por drones que vuelan a 40 cm del suelo, en cuanto a fabricación destacamos que la estructura minimizada, simple y biodegradable permite una producción en masa lo que lo hace económica para la agricultura de precisión, adicionando que el sensor puede generar lecturas estables sin problema durante 1 y 4% de variación de sensibilidad cuando el dispositivo comienza a degradarse.

En ciudades inteligentes el desarrollo de sistemas de transporte inteligente muestran avances muy importante en el desarrollo de las ciudades, es así que aparece VANET (Vehicular Ad Hoc Networks) como campo que ayuda a la comunicación entre vehículos en tiempo real para garantizar comodidad y seguridad, sin embargo los parámetros de autenticación robustos en entornos con recursos limitados puede presentar problemas, en este contexto (Damaševičius et al., 2022) proponen una autenticación mutua anónima y establecimiento de claves denominado SELWAK el cual se considera liviano, eficiente y seguro basado en IoT, este se conforma de 2 tipos de autenticación mutua V2V y V2R, adicionalmente conserva claves de forma secreta para garantizar comunicación de manera segura entre las RSU (unidades de carretera). Dando como resultados costo computacional liviano y sobrecarga de comunicación ya que se basa en operaciones XOR bit a bit y funciones hash unidireccional, además tiene rendimiento sólido frente a ataques man in the middle, replays, verificadores robados, imposibilidad de rastrear, ataques de suplantación de identidad.

Dentro de IoT la tecnología LoRaWAN es un protocolo de comunicación inalámbrica IoT interoperable para redes LoRa gracias a sus características como alcances grandes en el orden de los kilómetros, batería de larga duración, capacidad de la red, QoS, seguridad y bajo consumo de potencia , la gran demanda de dispositivos IoT hace que en el fututo el espectro gratuito sin licencia en el cual operan se llegue a sobrecargar, por lo que (Salika et al., 2022) propone LoRaCog un nuevo protocolo basado en las redes LoRa y radio cognitiva, utilizando el espectro no utilizado para permitir el acceso a un espectro más grande que el de LoRaWAN con el fin de ser más eficiente sin afectar el consumo de energía en los sensores finales, el uso de los espectros con licencia tiene como objetivo equilibrar la carga que se centra en el no licenciado. LoRaCog no pretende reemplazar a LoRaWAN pero cuando el espectro sin licencia se encuentre saturado se

consideraría como un asistente para el cambio de espectro de los dispositivos finales, detectando el Gateway para no aumentar carga al dispositivo final. El funcionamiento se basa en la topología en estrella, y de la arquitectura podemos destacar que cada Gateway detecta espacios en el espectro, para posteriormente enviar resultados de haberlos detectado a la red de sensores, el cual decide cual sería el mejor espectro, para que la puerta de enlace reenvíe esta información hacia los dispositivos finales, los mimos que aceptan o rechazan el espectro y proceden o no a transmitir los datos.

En el mundo de IoT los botnets maliciosos figuran como potenciales amenazas para la seguridad de las redes, para lo cual se tiene a Botnet Defense System (BDS) como un mecanismo de defensa el cual usa botnets de sombrero blanco para contrarrestar los maliciosos, el cual toma en cuenta la cantidad pero no la ubicación de los mismos; es así que (Pan y Yamaguchi 2022) propone identificar y definir zonas donde van a actuar los botnets mediante machine learning, el mismo que identifica los comportamientos maliciosos y actúa de forma táctica a los resultados específicos, luego de esto se pasa al modelado del BDS utilizando redes que se orientan a agentes y se evalúa el lanzador que se propuso. Para dicho caso se utiliza al botnet Mirai como botnet malicioso, identificando de esta manera 3 principales contribuciones como son: la comprensión de la propagación de botnets maliciosos en redes IoT, respuestas tácticas de acuerdo con cada zona, reducción de aproximadamente 30% de los dispositivos IoT infectados mediante la evaluación de redes Petri encaminadas a agentes.

La seguridad en internet es muy importante y en base a las propiedades de la seguridad tenemos mecanismo que garanticen dichos aspectos como: encriptación AES para la confidencialidad, RSA para autenticidad, SHA-256 para integridad y el no repudio, pero se destaca que los dispositivos IoT deben ser ligeros y de bajo consumo por lo que estos mecanismos criptográficos no podrían ser aplicables, en este marco (Morge-Rollet et al. 2022) proponen una huella dactilar de Radio Frecuencia para la autenticación de las cosas inteligentes como mecanismo no criptográfico; este se basa en el análisis de las propiedades de degradación de los componentes, inicialmente este proceso se lo realizaba de forma manual pero por el crecimiento exponencial de dispositivos estos se realizan mediante modelos de aprendizaje profundo. Los autores proponen un método de huellas dactilares de RF con baja complejidad de cálculo y baja demanda de datos para el aprendizaje, descomponiendo valores singulares para el reconocimiento de características, luego se selecciona las más destacadas mediante prueba de hipótesis para posterior toma de decisiones en base a modelos estadísticos.

CAPÍTULO III

3 MARCO METODOLÓGICO

3.1 Introducción

Este capítulo detalla el proceso mediante el cual se desarrollaron los prototipos de lectura de dispositivos NFC, el diseño e implementación de las interfaces de interacción para administradores y operarios, así como la aplicación a través de la cual el usuario va a poder ingresar, salir del establecimiento y consultar sus registros de actividad, todo esto mediante el alojamiento en un servidor hosting.

3.2 Estado actual del control de acceso vehicular en la EP-EMMPA

La Empresa Pública Municipal Mercado de Productores "San Pedro de Riobamba", en la actualidad cuenta con un control de acceso vehicular denominado SCE (Sistema de control de estacionamientos) implementado y en funcionamiento desde el año 2007. El mismo que se basa en tecnología láser para la lectura de tickets que almacenan un código mediante barras; que necesita obligatoriamente la interacción de un operario, al ingreso para su registro y a la salida para el cobro respectivo.

La capa visual se presenta en Visual Basic, donde permite: en el ingreso la categorización de los vehículos para asignar una tarifa y en la salida la respectiva verificación, cobro y facturación acorde al tiempo que estuvo en el establecimiento. Toda esta información se almacena en la base de datos de la marca comercial María DB.

El sistema se ha desarrollado en el principio cliente-servidor en el cual el cliente no guarda ningún tipo de sesión o recurso, netamente se encarga de hacer peticiones al servidor en busca de respuesta y este responde con la información pertinente al requerimiento. El funcionamiento de la infraestructura se basa en una red local con los equipos detallados en la tabla 1-3:

Tabla 1-3: Equipos y funcionamiento del sistema actual

Equipo	Función	Características
Servidor DELL Power Edge T620	Servidor que gestiona todas las peticiones de manera local, actualmente cuenta con 3 discos duros de 300gb de capacidad de almacenamiento	Torre de servidor de 16 bahías de 2,5" Procesador: 2 x Intel Xeon 2697 V2 a 2,7 GHz 10 núcleos. Memoria registrada PC3-12800R DDR3 de 256 GB.

		Tarjeta RAID: Dell PCIe H170 RAID – caché de 512 MB. 2 fuentes alimentación 750 W. Acceso remoto mediante iDRAC7 Express. 16 bandejas de disco vacías Dell de 2,5 pulgadas.
Impresora Epson TM T20II	En la entrada se encarga de emitir los tickets para los diferentes usuarios y en la salida para emitir los comprobantes que se cancelan por el servicio prestado	Impresora monocromática térmica de línea Tamaño compacto de recibos 8cm de ancho Conexión USB
Lector código de barras Honeywell 1250g	Utilizado en la salida para poder escanear los códigos de barras que le fue entregado al usuario en la entrada	Lectura Uni-direccional Puerto USB Rendimiento de lectura Lectura de una sola línea Velocidad de hasta 100 líneas de lectura por segundo

Realizado por: Sáez, Cristian, 2022

De forma detallada, el sistema actual funciona de la siguiente manera: en el ingreso se realiza una clasificación manual para emitir un ticket de acuerdo al tipo de vehículo teniendo los siguientes y tarifas:

- Pequeño de 0 a 2 toneladas, $ 0,50 por hora o fracción
- Mediano 2,1 a 5 toneladas, $ 0,80 por hora o fracción
- Grande de 5,1 en adelante, $ 1,40 por hora o fracción
- Mula, $ 2,00 por hora o fracción
- Tráiler, $ 3,00 por hora o fracción
- Frutas tropicales, si ingresan y salen hasta las 10:00 cancela $2,50, si se excede de ese tiempo y sale antes hasta el cierre del establecimiento se aumenta $2,50. Si permanece hasta el siguiente día antes de las 10:00 sigue el aumento de $2,50 y sigue el ciclo
- Triciclos, $0,20 sin límite en el día
- Frecuente post pago (vehículos cooperados de la empresa), $0,10 sin límite en el día

Este proceso lo realiza un operario de manera manual en la interfaz detalla en la ilustración 1-3.

Ilustración 1-3: Interfaz de categorización de vehículos

Realizado por: Sáez, Cristian, 2022

Posterior a la selección se emite un ticket donde se detalla el nombre de la empresa, el tipo de vehículo, la fecha, hora de ingreso, la numeración de ticket y el código de barras enumerado, como se muestra en la ilustración 2-3.

Ilustración 2-3: Ticket generado al ingreso del estacionamiento

Realizado por: Sáez, Cristian, 2022

Posteriormente en la salida se tiene un sistema en el cual el lector va a interpretar el código emitido en el ingreso, el mismo que finalizará el ciclo asignando a través del sistema la fecha y hora de salida, calculando el tiempo de permanencia y el valor a cancelar, en el caso de presentar

inconsistencia en el tipo de vehículo puede recategorizarse para su correcto cobro, finalizado el proceso se emite el comprobante, documentos detallados en la ilustración 3-3


EP MUNICIPAL MERCADO DE
PRODUCTORES AGRICOLAS SAN
PEDRO DE RIOBA
AV. LEOPOLDO FREIRE S/N

RUC:0660835430001
Riobamba
COMPROBANTE:001-017-000510439

Usuario:CONSUMIDOR FINAL
Ced.o Ruc:9999999999999
Dirección:
Telefono:
Ingreso:2022-07-27 08:15:48
Salida: 2022-07-27 08:36:48
Nro. Tarjeta:11265985

<C> <Descripcion> <Valor>
3 Parqueadero Vehiculos Pequeños 1.50
TOTAL $1.50
Pago Efectivo
Fecha de Emision:27-07-2022
Recaudador:ibernal
Control de Factura: 184

Clave de acceso SRI
2707202201066083543000120010170005104390651043918
Para revisar su factura electronica ingrese al sitio web http://www.sri.gob.ec y se registra con su numero de cedula/RUC/pasaporte. Su factura estara disponible dentro de las proximas 24 horas Dudas: 032994030

Ilustración 3-3: Emisión de factura

Realizado por: Sáez, Cristian, 2022

A pesar de contar con un sistema de control de acceso vehicular, la gran afluencia de vehículos y la manera en la que se maneja el servicio de estacionamiento genera algunos problemas, destacando el tiempo de ingreso y salida de vehículos, mediante la técnica de observación se aprecia que: al ingresar el usuario se demora en promedio 1 minuto sin presencia de tráfico caso contrario hasta 15 minutos, a la salida unos 2 minutos sin presencia de vehículos y con afluencia de los mismos hasta 18 minutos. Esto debido en gran manera a que no cuentan con procesos automatizados, ni registros de los usuarios que puedan acelerar dicha actividad.

3.3 Infraestructura y arquitectura IoT

Posterior a realizar el análisis por el método deductivo, se considera que la arquitectura basada en REST (Transferencia de estados representacionales) es la idónea para el proyecto ya que este tipo se basa en HTTP, el mismo que crea servicios y aplicaciones que pueden ser accedidos por clientes. Dentro de esta arquitectura se destaca el principio de restricción cliente-servidor el cual es uno de los más utilizados gracias a que el servidor brinda varios servicios, el mismo que escucha solicitudes de los clientes mediante conectores, al tiempo que realiza o rechaza la solicitud enviando respuestas, en este caso los clientes van a interactuar con el servidor mediante solicitudes. Adicional se destaca que el conjunto de funciones CRUD son de suma importancia

ya que en estos se fundan gran parte del sistema por sus características de crear, leer, actualizar y eliminar.

De igual manera se utiliza el protocolo MQTT ya que su comunicación es ligera y responde al principio publicación/suscripción necesarios para activar el procesamiento de imágenes mediante el lenguaje de programación Python cada que sea requerido el reconocimiento de placas vehiculares, juntamente con el uso de la API de Google vision que permite el procesamiento de la imagen de forma rápida y sencilla.

Adicional al considerar sistema de pago y por la seguridad misma que necesita el sistema se desarrollan diferentes medidas de seguridad en las diferentes capas, tales como en la capa física, la seguridad intrínseca que ofrece NFC y una barrera vehicular, en la capa de red el protocolo SSL que junto a http proporcionan comunicaciones seguras en la red y en la capa de aplicación sistemas de autenticación y verificación de datos mediante logueos a la interfaz web y aplicación, así mismo ingreso de PIN de seguridad para autorizar el pago por el servicio.

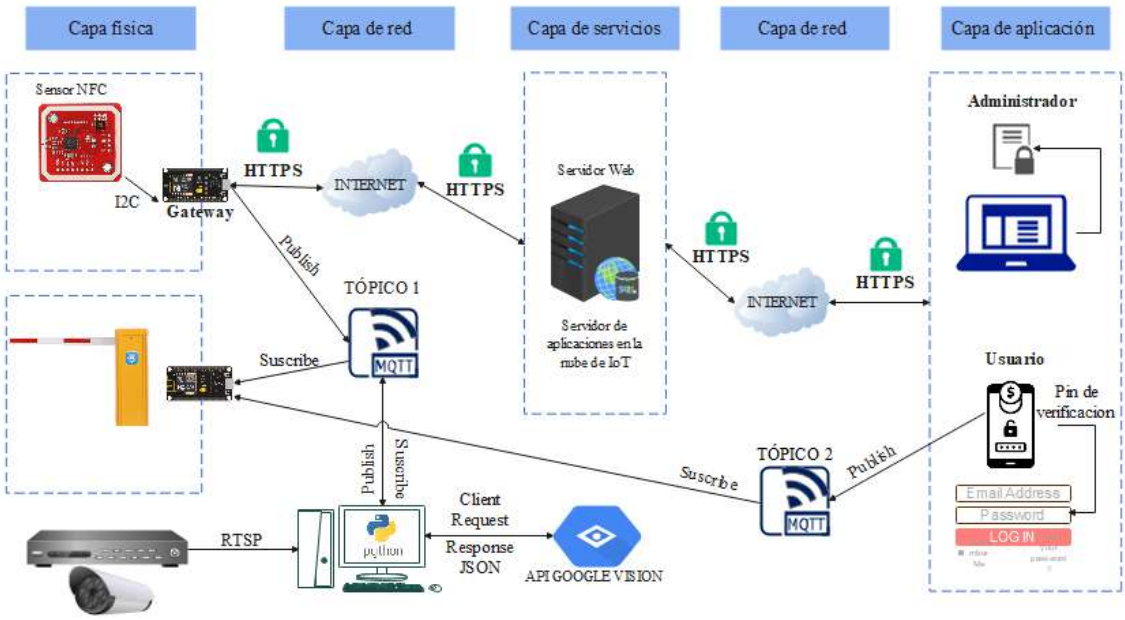


Ilustración 4-3: Infraestructura y arquitectura IoT

Realizado por: Sáez, Cristian, 2022

3.4 Desarrollo del prototipo

Para poder realizar el prototipo se usa una metodología basada en una investigación experimental, donde el método deductivo se usa para recopilar información que permita establecer la arquitectura de red que se pueda adecuar al sistema para su correcto funcionamiento en todos los niveles y aplicaciones.

En la ilustración 4-3 se pude visualizar la topología en la que se funda para el prototipo, de acuerdo con esto, al ingreso el usuario debe tener un dispositivo inteligente con la tecnología NFC, que mediante un tag pueda verificar datos del usuario que se encuentran en la base de datos del servidor web, en este punto dado que existe una categorización se autentica que el vehículo es el registrado mediante el reconocimiento de placas del mismo, si los datos son correctos se registrará la hora de ingreso.

Al momento de salir el usuario deberá preparar su dispositivo para poder cancelar por el servicio de estacionamiento, este se realizará mediante tarjetas de crédito o débito que se registraron previamente en el dispositivo, se considera un pin de pago para garantizar seguridad en la transacción. La aplicación móvil permitirá el registro de datos el usuario y el control de su actividad en el establecimiento.

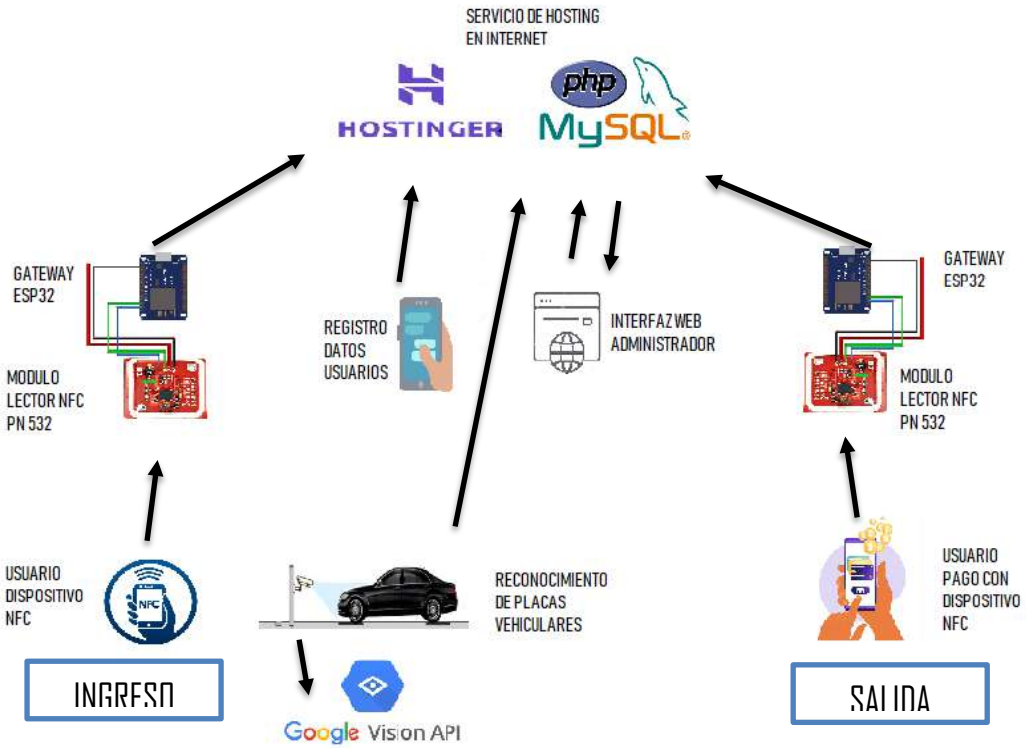


Ilustración 5-3: Topología del prototipo

Realizado por: Sáez, Cristian, 2022

3.4.1 *Aplicación móvil*

En el diseño es el que accionará el sistema, tanto para el ingreso como para la salida, mediante esta aplicación, el dispositivo realizará el tag con el prototipo del módulo lector NFC, adicional implementa funciones como registro, login y visualización de datos y reportes del usuario, considerando también el registro de tarjetas de débito o crédito para el pago del servicio.

3.4.2 *Lector NFC*

El prototipo tiene la función de leer el uid que se encuentra en el dispositivo inteligente, mediante el tag, los datos obtenidos se envían hacia el servidor mediante el módulo que hace la función de gateway que accionará el sistema.

3.4.3 *Interfaz web*

En esta etapa se podrá visualizar los datos de usuarios registrados, la capacidad de registrar usuarios mediante tarjetas o llaveros, visualizar los datos del usuario y tiempo de ingreso y salida del establecimiento, adicional ver reportes

3.4.4 *Servidor/Base de datos*

Es una de las etapas más importantes del sistema IoT ya que en la base de datos se va a almacenar la información de los usuarios, tiempos de ingreso, salida y reportes de cada usuario como saldo y cantidad de ingreso y salida del establecimiento

3.5 Requerimientos de diseño del prototipo

Mediante el análisis realizado se puede plantear requerimientos que demanda el prototipo para controlar el acceso a la EP EMMPA con sistema IoT y NFC implementado en smartphones con la tecnología o a su vez con otros dispositivos como llaves o tarjetas, entre dichas demandas podemos mencionar que:

- La operación debe ser ágil y adaptarse a los parámetros establecidos en cuanto a costos y horarios del lugar
- Se debe garantizar una comunicación fluida y segura al momento de accionar el sistema IoT
- La información debe ser protegida para evitar posibles ataques que tengan como objetivo el robo de información
- El uso del sistema por parte del usuario debe ser intuitivo y en el momento de ingresar o salir este no debería abrir la aplicación para poder accionar el sistema

3.6 Requerimientos de software

3.6.1 *Software para aplicación móvil*

Para el desarrollo de la aplicación móvil se escoge Android Studio, ya que cuanta con información variada y extensa sobre las librerías y el modo HCE que se necesita para accionar el sistema, adicional podemos destacar que cuenta con información y soporte del sistema operativo Android que se está utilizando en el smartphone

3.6.1.1 *Android Studio*

Se define como el entorno de desarrollo integrado basado en IntellJ IDEA, que es oficial para el desarrollo de aplicaciones con el sistema operativo Android, brindando variadas funciones que tienen como fin la mejora de productividad al momento de la creación de aplicaciones entre las cuales tenemos

- Compilación flexible fundado en Gradle
- Emulador ligero, extenso y variado de funciones
- Entorno único para todos los equipos Android
- Modificación de código y recursos de la aplicación sin la necesidad de reinicio
- Importación y ayuda de recursos alojados en GitHub que permiten crear funciones comunes
- Herramientas variadas y módulo de prueba
- Detección de problemas de rendimiento, uso, compatibilidad, etc mediante Lint
- Compatibilidad con lenguaje de programación C++ y NDK, Java y Kotlin
- Compatible con Google Cloud Platform, que permite una integración más fácil de Google Cloud Messaging y App Engine (Developers 2021)

Entre sus especificaciones se puede encontrar los requerimientos para poder utilizar el programa entre los cuales se destacan:

- Conocimiento de programación en los lenguajes ya mencionados
- Características mínimas de hardware y software para la instalación y ejecución del programa
    - Sistemas operativos que soportan: windows 7, 8 o 10 de 32 y 64 bits, MacOS X desde la versión 10.10 hasta la 10.13, GNOME o KDE Linux.
    - Memoria RAM mínima de 4 gb, recomendación de 8 gb.
    - Memoria para almacenamiento mínimo de 2 gb, recomendado superior o igual a 4 gb
    - Java Development Kit (JDK) 8.
    - Resolución 1280 x 800 mínima, 1440 x 900 recomendada (Santaella 2022)

3.7 Requerimientos de hardware

Los dispositivos físicos juegan un papel muy importante ya que permite, ya que son los equipos que van a interactuar para recibir la información que va a activar el sistema, al ser el desarrollo de un prototipo este debe recopilar las características necesarias que permitan el correcto funcionamiento de lectura de los dispositivos NFC

3.7.1 *Módulo NFC PN532*

Se realiza una comparación de diferentes módulos comunes que existen en el mercado, en este caso el NFC PN532, de adafruit el RC522 y finalmente el RFID RCRC522 para escoger el que mejor se adapte, esta se puede apreciar en la tabla 2-3

Tabla 2-3: Comparación de módulos

Características	**RFID/NFC PN532**	**Adafruit RC522**	**RFID RC522**
Voltaje de operación	3.3-5	3.3	3.3
Frecuencia operación	13.56	13.56	13.56
Transferencia de datos	10	424	100
Interfaces	SPI, l2C, UART	SPI, l2C, UART	SPI
Precio	$11.93	$40	$ 5.20

Fuente: (Adafruit 2023)(Mechatronics Naylamp 2023)(Mechatronics Naylamp 2022)

Realizado por: Sáez, Cristian, 2022

Una vez realizado la comparación de los módulos, se procede a realizar la escala de Likert de los diferentes módulos en la tabla 3-3

Tabla 3-3: Escala de Likert de los módulos

Características	**RFID/NFC PN532**	**Adafruit RC522**	**RFID RC522**
Voltaje de operación	5	5	5
Frecuencia operación	5	5	5
Transferencia datos	3	1	4
Interfaces	5	5	1
Precio	4	2	5
Total	**22**	**18**	**20**
Porcentaje	**88%**	**72%**	**80%**

Realizado por: Sáez, Cristian, 2022

La escala de Likert para los servidores el número máximo que se obtiene es de 25 puntos debido a las características multiplicadas por el valor máximo que se puede obtener por cada uno de ellos, en este caso el valor máximo es 5 ya que define que está totalmente de acuerdo y el valor mínimo que es 1 y está totalmente en desacuerdo, el valor 2 representa que esa en desacuerdo, el 3 implica indecisión, finalmente el número 4 está de acuerdo, de tabla 3-3 se establecen rangos aceptables que se definen que del 0 al 6 y presenta una debilidad del producto, del 7-13 puede mejorar, del 14-20 implica fortaleza y del 21-25 es una ventaja que posee el producto frente a los demás.
Adafruit RC522 tiene un puntaje de 18 que está dentro del rango que puede mejorar, los aspectos que ese modulo tienen falencias son principalmente en el precio debido a que este no se encuentra disponible en el país y adicional se tiene que pagar por el envío del producto, el siguiente es RFID

RC522 este módulo suma 20 puntos y también pueden mejorar, a pesar que tienen un precio accesible, este módulo no funciona como NFC lo que para este proyecto no es óptimo además trabaja solo con 1 interfaz. Finalmente, el módulo RFID/NFC PN532suma 22 puntos y es el óptimo para este proyecto debido a que posee las funciones de RFID Y NFC teniendo 3 interfaces como son SPI, l2C, UART, contando con un precio accesible para el usuario.

Dado que se va a implementar el modo HCE de NFC se elige el módulo NFC PN532 que, por sus capacidades, precio, modos de operación pueden facilitar la lectura de los diferentes tipos de tarjetas NFC, tanto la emulada como la del llavero. (Naylamp Mecatronics SAC 2021)

El análisis establece que el módulo PN532 es el idóneo para el proyecto ya que permite leer y escribir en tags y en celulares que soportan NFC como los celulares, se desarrolla sobre dispositivos y tarjetas NFC que trabajen a la frecuencia de 13.56Mhz, sus interfaces de programación son SPI, I2C y UART se elige la requerida usando correctamente el DIP-SWITCH o un cable USB-Serial. (Mechatronics Naylamp 2023) detalles que se observan en la tabla 4-3

Tabla 4-3: Módulo PN532

Especificaciones técnicas		
Características	**Valor**	**unidad**
Voltaje de operación	3.3-5	V
Frecuencia de operación	13.56	Mhz
Transferencia de datos	10	Mbit/s
Interfaces	SPI, l2C, UART	
Estándar RFID	ISO/IEC 14443-A	
Estándar NFC	ISO/IEC 18092	

Fuente: (Mechatronics Naylamp 2023)

Realizado por: Sáez, Cristian, 2022

3.7.2 *Node MCU ESP 8266*

Para la selección del Gateway se consideraron diferentes tarjetas de desarrollo que se presentan en la tabla 5-3

Tabla 5-3: Escala de Likert para las placas de desarrollo

Características	**ARDUINO**				**RASPBERRY**			**ESP8266**	**ESP32**
	UNO	**MEGA**	**NANO**	**LEO**	**Pi1**	**Pi2**	**Pi3**		
RAM	1	1	1	1	3	5	5	2	3
Consumo energía	4	4	2	3	1	5	5	5	4
Pines digitales IO	3	5	3	4	0	0	0	4	4
Pines entrada analógica	4	5	4	4	0	0	0	3	5
wifi	0	0	0	0	0	0	5	5	5
costo	5	3	4	4	0	2	1	5	4
CompatibilidadPN532	2	2	2	2	5	5	5	5	2
TOTAL	19	20	16	18	9	17	21	29	27
Porcentaje [%]	54.28	57.14	45.71	51.42	25.8	48.57	60	82.85	77.14

Realizado por: Sáez, Cristian, 2022

Para el análisis de las placas de desarrollo, como se observa en la Tabla 5-3, el número máximo que se obtiene es de 35 puntos debido a las características multiplicadas por el valor máximo que se puede obtener por cada uno de ellos, en este caso el valor máximo es 5 ya que define que está totalmente de acuerdo y el valor mínimo que es 1 y está totalmente en desacuerdo, el valor 2 representa que esa en desacuerdo, el 3 implica indecisión, finalmente el número 4 está de acuerdo, de la tabla se establecen rangos aceptables que se definen que del 0 al 8 y presenta una debilidad del producto, del 9-17 puede mejorar, del 18-26 implica fortaleza y del 27-35 es una ventaja que posee el producto frente a los demás,

De esta manera se puede observar en la tabla que raspberry en sus 2 primeras versiones y Arduino nano presentan debilidades pero están en la capacidad de mejorar, así mismo, dentro del siguiente rango que es de 18-26 se encuentran el raspberry pi 3 y Arduino uno y leo, finalmente los productos que demuestran fortaleza con los valores dentro de los rangos del 27-35 son los ESP8266 y el ESP32, demostrando que el ESP8266 se encuentra por encima de los demás productos en diferentes aspectos como el bajo consumo de energía, la compatibilidad con el módulo PN532, su wifi integrado y su bajo costo que es asequible con el usuario final.

Una de las razones que llevaron a elegir este dispositivo para el desarrollo del prototipo es la extensa y variada cantidad de recursos y ayuda que se puede encontrar en la web, además que sus características de compatibilidad y versatilidad ayudan de gran manera el desarrollo del módulo lector, ya que es compatible con el lector antes mencionado

3.8 Diseño lógico del prototipo

En este punto se desarrollan la aplicación, interfaces web, adicional del servidor Hostinger con bases de datos MySQL

3.8.1 *Plataforma para el servidor*

Se realiza una comparación de los servidores en el mercado más utilizados como son HostGator, A2Hosting y Hostinger, del cual se elige el que presenta con las mejores características para el proyecto, por lo que en la tabla 6-3 se realiza una comparativa de características

Tabla 6-3: Comparación de servidores

Características	**HostGator**	**A2Hosting**	**Hostinger**
Almacenamiento	50Gb	50 Gb	100 Gb
Bases de datos	Ilimitado	5 sitios	Ilimitado
Ancho de banda	Ilimitado	ilimitado	ilimitado
Confiabilidad	99,9%	99,9%	99,90%
Seguridad	SSL	SSL	SSL
Servidores protegidos	-----	Imunify360	Cloudflare
Copias de seguridad	-----	Diarias	semanales
Wordpress	Administrado	Administrado	Administrado
Servicio	devolución de dinero 45 días	------	devolución de dinero 30 días
soporte	24/7	24/7	24/7
Subdominios	-----	Ilimitado	100

Fuente: (Hostinger 2023; A2 Hosting 2023; Hostgator.com 2023)

Realizado por: Sáez, Cristian, 2022

Se procede a realizar la escala de Likert de los diferentes servidores en la tabla 7-3.

Tabla 7-3: Escala de Likert para los servidores

Características	**HostGator**	**A2Hosting**	**Hostinger**
Almacenamiento	3	3	5
Bases de datos	5	1	5
Ancho de banda	5	5	5
Confiabilidad	4	4	3
Seguridad	5	5	5
Servidores protegidos	0	4	5
Copias de seguridad	0	5	4
Wordpress	5	5	5
Servicio	4	0	5
Soporte	5	5	5
Subdominios	0	5	4
Total	**36**	**42**	**51**
Porcentaje	**65.45%**	**76.36%**	**92.72%**

Realizado por: Sáez, Cristian, 2022

La escala de Likert para los servidores el número máximo que se obtiene es de 55 puntos debido a las características multiplicadas por el valor máximo que se puede obtener por cada uno de ellos, en este caso el valor máximo es 5 ya que define que está totalmente de acuerdo y el valor mínimo que es 1 y está totalmente en desacuerdo, el valor 2 representa que esa en desacuerdo, el 3 implica indecisión, finalmente el número 4 está de acuerdo, de la tabla se establecen rangos aceptables que se definen que del 0 al 13 y presenta una debilidad del producto, del 14-27 puede mejorar, del 28-41 implica fortaleza y del 42-55 es una ventaja que posee el producto frente a los demás.

Hostgator suma un total de 36 puntos, lo que significa que el servidor tiene fortaleza en la mayoría de aspectos, pero le faltan características tales como los servidores protegidos o los subdominios, sin embargo, A2 Hosting y Hostinger sumando 42 y 51 respectivamente, el mejor es Hostinger debido a que tiene muchas ventajas sobre los demás debido a que posee un mayor almacenamiento para los datos, tiene seguridad además de servicio y soporte por técnicos certificados y una devolución de dinero en el caso que al usuario final no le guste el producto.

3.8.1.1 *Hostinger*

Se considera como un servidor virtual privado, con funciones integradas como optimización de sitios web que ayudan a la creación, monitorización de vulnerabilidades, copias de seguridad de sitios web, configuración de parámetros para el protocolo FTP, que permite transferencia de archivos de forma fácil, en cuestión de seguridad brinda certificados SSL y motor de caché. Posee

una extensa lista de funciones que se encuentran en el dashboard, de las cuales se destacan información de la cuenta, dominios, manejo de archivos, base de datos, seguridad, entre muchas otras más. Se destaca que ayuda en la configuración de direcciones IP dedicadas, acceso completo a la raíz y configuración del almacenamiento. Posee 21 funcionalidades entre las cuales destacamos las ya mencionadas y adicional las siguientes, API, Controles o permisos de accesos, creación de informes y datos estadísticos, gestión de archivos, canales, comercio electrónico, contenidos, datos del producto entre otras más (GetApp 2022)

3.8.1.2 *PHP*

En primer lugar, cabe destacar que el procesador de hipertexto PHP es un lenguaje de scripting que se implementa en el servidor, gratuito de código abierto utilizado por el 77,6% de todos los sitios web, este tipo de lenguajes se diferencia de los lenguajes de programación ya que estos incorporan funcionalidades para realizar una acción o función específica, brindando la capacidad de interpretar scripts en otros softwares.

Los lenguajes de scripting se diferencian entre los dedicados al back-end para el servidor y front-end para el usuario. PHP utiliza el lenguaje del lado del servidor para ejecutar instrucciones del script, para que el servidor responda datos a petición, canaliza las mismas y organiza la información en una base de datos. Cuando el servidor web recibe el script la procesa y responde a un navegador en formato HTML, almacenando la información en bases de datos para asegurar que no se puedan acceder a estos datos o al código fuente. (Deyimar 2022)

3.8.1.3 *Dominio*

Un dominio es una dirección digital única, donde las personas usan para encontrar una página web en internet, una dirección IP está vinculada a cada dominio.

“Hostinger.com es un nombre de dominio, así como Google.com o Facebook.com. Lo ideal es que tu nombre de dominio sea registrado bajo el mismo nombre que tu sitio web o marca.” (Hostinger 2022a)

Hostinger proporciona protección de privacidad denominada WHOIS, misma que oculta información tal como nombre, dirección, correo electrónico, entre otras para que sea genérico y no se pueda identificar fácilmente. Se siguen los siguientes pasos para saber cómo funcionan los nombres de dominio como se observa en la ilustración 5-3.

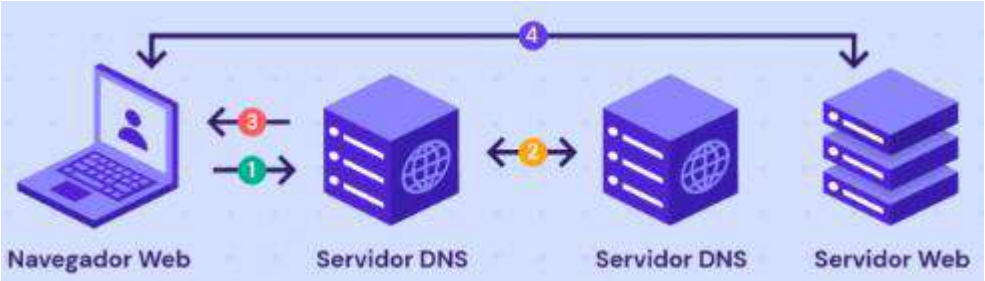


Ilustración 6-3: Funcionamiento de nombres de dominio

Fuente: (Hostinger 2022b)

1. Cuando se desea acceder con un dominio en el navegador web, el servidor realiza una búsqueda en los servidores que conforman el sistema de nombres de dominio (DNS).
2. El sistema de nombres de dominio busca la dirección IP que se encuentra asociada al nombre del dominio.
3. El navegador muestra la información que el servidor DNS asocie a la IP introducida
4. El navegador solicita datos sobre el sitio al servidor de hosting del dominio y una vez que el servidor de hosting regresa los datos, se observa la página web. (Hostinger 2022b)

Existen diferentes tipos de dominios

Dominios de nivel superior (TLD)

estos son asignados por una organización llamada Autoridad de números asignados de internet, más conocida como IANA y como su nombre lo dice se encuentran en el nivel superior de dominios de internet, dentro de estos se encuentran los .com .org .net y .edu

- Dominios de nivel superior de código de país (ccTDL): estos dominios poseen únicamente dos letras y se basan en los códigos internacionales de países, tales como .es para España, .ec para Ecuador, .us para los Estados Unidos, entre otros.

- Dominios de nivel superior genérico (gTDL): estos dominios son específicamente para un caso como, por ejemplo .edu que se usa para las instituciones educativas, .gov que es utilizado para el gobierno, .org para organizaciones sin fines de lucro, entre otros

Para el proyecto se utiliza un dominio de nivel superior, detallado por la siguiente URI http://espochemmpa.com información detallada en la ilustración 6-3

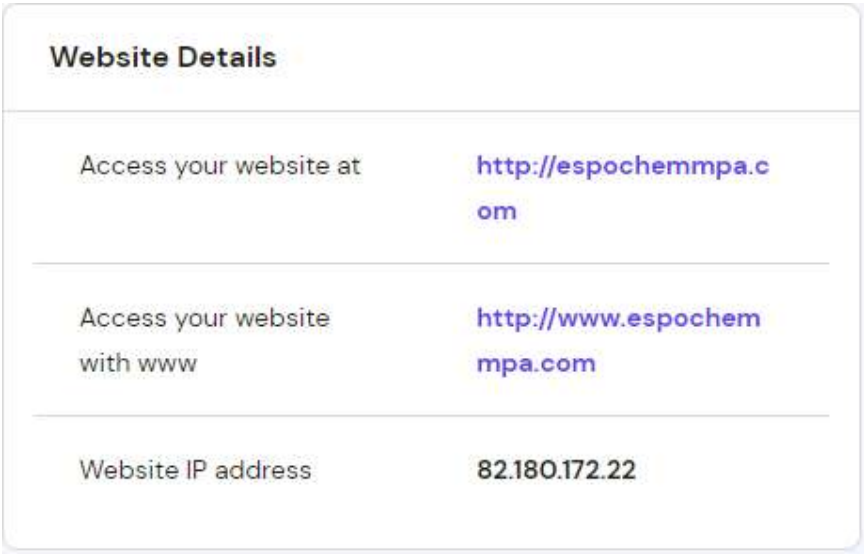


Ilustración 7-3: Detalle del dominio

Realizado por: Sáez, Cristian, 2022

Al dominio establecido se aplica el certificado SSL que ayuda a establecer conexiones de forma segura a Internet, esto codificando los datos que se transmiten entre 2 sistemas mediante algoritmos de cifrado(DigiCert 2023). En la ilustración 8-3 se puede observar que el servidor Hostinger activa de forma automática el protocolo SSL mediante Lets's Encrypt a espochemmpa.com que es el dominio del prototipo

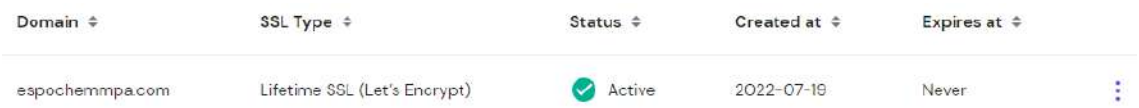


Ilustración 8-3: Certificado aplicado al dominio

Realizado por: Sáez, Cristian, 2022

3.8.2 *MySQL*

Es un sistema de gestión de bases de datos, Trabaja con bases de datos relacionales, es decir, interconecta diferentes tablas para posteriormente almacenar información, posee una licencia abierta y otra de versión comercial otorgada por ORACLE (Robledano 2019b)

3.8.3 *Características MySQL*

MySQL presenta características como las siguientes

- Cliente-servidor: Uno o varios dispositivos se conectan al servidor a través de una red específica, y cada uno de ellos realizan una solicitud desde la interfaz gráfica de usuario, posteriormente el servidor dará respuesta siempre y cuando ambas partes entiendan las instrucciones.

En MySQL se crea la base de datos con las respectivas relaciones de las tablas, posteriormente los clientes realizan solicitudes escribiendo instrucciones SQL en MySQL y finalmente el servidor dará respuesta con la información solicitada en la pantalla de los clientes. (Hostinger tutoriales 2022)

- Compatibilidad con SQL: si se desea pasar de otro motor de base de datos a MySQL no será ningún problema debido a que esta ofrece compatibilidad.

- Desencadenantes: MySQL brinda la posibilidad de automatizar algunas tareas en la base de datos, por ejemplo, se activan diferentes eventos que permiten actualizar en tiempo real registros de esta manera optimiza recursos y tiempo.

- Transacciones: MySQL permite la integridad de la base de datos debido a que se asegura que todas las instrucciones sean ejecutadas correctamente, de esta manera al fallar el sistema se resguarda la información

- Flexible y fácil de usar: el código fuente es manejable para satisfacer las necesidades del desarrollador

- Alto rendimiento: MySQL puede trabajar a grandes velocidades sin importar la cantidad de datos que se estén almacenando.

- Un estándar en la industria: debido a que la primera versión de MySQL es realizada en mayo de 1995 posee diferentes recursos que han sido proporcionado por distintos desarrolladores calificados.

- Seguro: la verificación de MySQL esa basada en el host y cifrado de contraseña además de utilizar privilegios de acceso y administración de cuentas, todo esto hace que proporcione la seguridad esperada a los clientes. (Robledano 2019b)

3.8.4 *Sentencias y Funciones MySQL*

En la tabla 8-3 se muestran algunas de las sentencias básicas que se usa en MySQL debido a que posee compatibilidad con diferentes aplicaciones web debido especialmente cuando se habla del backend realizadas en algún lenguaje de programación como PHP.

Tabla 8-3: Sentencias y funciones MySql

Sentencia o función	Descripción
Select	Consulta datos
Where	Incluye condiciones en la consulta
Insert	Inserta datos
Update	Actualiza o modifica datos existentes
Delete	Elimina datos
Order By	Ordena los resultados de una consulta
Distinct	Elimina datos duplicados de las consultas

Fuente: (Robledano 2019a)

3.9 Diseño e implementación del prototipo

3.9.1 *Desarrollo e implementación del módulo lector de dispositivos NFC*

3.9.1.1 *Conexión entre módulos de desarrollo*

La conexión entre los módulos de desarrollo NFC PN532 y el ESP 8266 se lo realiza mediante la inclusión de librerías, también se crea el objeto pn532 i2c para establecer la comunicación entre los dispositivos de forma serial, estableciendo un SDA para la línea de datos y SCL/CLK para el reloj, siendo estas bidireccionales, de tipo Half Duplex, dicho proceso podemos observar en la ilustración 9-3

```
// NFC
#include <Wire.h>
#include <PN532_I2C.h>
#include "PN532.h"
#include <NfcAdapter.h>

PN532_I2C pn532i2c(Wire);
PN532 nfc(pn532i2c);
```

Ilustración 9-3: Librerías y creación objeto NFC

Realizado por: Sáez, Cristian, 2022

3.9.1.2 *Codificación de lectura de UID de NFC*

```
if (success) {
  //   Serial.println("Found something!");

  uint8_t selectApdu[] = { 0x00, /* CLA */
                           0xA4, /* INS */
                           0x04, /* P1  */
                           0x00, /* P2  */
                           0x07, /* Length of AID  */
                           0xF0, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, /* AID defined on Android App */
                           0x00  /* Le  */
                         };
  uint8_t response[32];

  success = nfc.inDataExchange(selectApdu, sizeof(selectApdu), response, &responseLength);

  if (success) {

   // hex_value = "";
   // hex_value =   printResponse(response, responseLength);


    uint8_t apdu[] = "ENTRADA";
    uint8_t back[32];
    uint8_t length = 32;

    success = nfc.inDataExchange(apdu, sizeof(apdu), back, &length);
```

Ilustración 10-3: Envío de comando APDU hacia dispositivos NFC y respuesta

Realizado por: Sáez, Cristian, 2022

El código detallado en la ilustración 10-3, permite el envío un comando APDU hacia la tarjeta que en este caso se puede encontrar en el teléfono inteligente o en un llavero con la tecnología NFC, definiendo el AID que se encuentra en el dispositivo Android como se muestra en la ilustración 10-3, el mismo que almacenará en la variable success la longitud de la respuesta del otro dispositivo, posterior a este proceso se almacenará en la variable hex_value el UID que emitirá el dispositivo NFC, el cual en caso de ser mayor a 6 en su longitud se tomará como válido para su procesamiento, lo cual se puede verificar en la ilustración 11-3. Para que este proceso se cumpla se debe cumplir con el intercambio de un mensaje para confirmación de los 2 dispositivos que desean interactuar, en este caso se definió la palabra “ENTRADA” como reservada para intercambiar el UID

```
323         hex_value = "";
324         hex_value =   printResponse(back, length);
325
326
327         if (hex_value.length() == 6) {
328           isPhone = true;
329         }
330
331         datoNFC = false;
```

Ilustración 11-3: Almacenamiento y verificación de UID

Realizado por: Sáez, Cristian, 2022

3.9.1.3 *Conexión de Gateway y envío de datos hacia el servidor*

Para enviar datos hacia el servidor, se deben establecer la librería <ESP8266WiFi.h> definido en el anexo A en la línea 9 y parámetros de la red para poder conectar el módulo ESP8266 hacia internet, en las líneas 166 a la 172 del mismo anexo podemos ver el proceso para poder conectar al router utilizado como punto de acceso

```
38 /* 1. Define the WiFi credentials */
39 #define WIFI_SSID "AccesoIoT"
40 #define WIFI PASSWORD "Family.Grab"
166   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
167   Serial.print("Connecting to Wi-Fi");
168   while (WiFi.status() != WL_CONNECTED)
169   {
170     Serial.print(".");
171     delay(300);
172   }
```

Ilustración 12-3: Librerías y parámetros de conexión a la red

Realizado por: Sáez, Cristian, 2022

En el caso de aprobar un UID válido se procede al envío de la información hacia el servidor mediante el protocolo HTTP, para realizar este procedimiento primero se definen la librería <ESP8266HTTPClient.h>, definido en el anexo A en las líneas 3.

Se realizan diferentes envíos hacia el servidor mediante el método POST que permite que se ejecute la acción hacia diferentes direcciones de acuerdo con la necesidad, en este marco en la ilustración 13-3 detalla los parámetros de conexión y envío.

```
451 String consultaTicket(String uid) {
452
453   String ticket = "0";
454   WiFiClient client;
455   HTTPClient http;    //Declare object of class HTTPClient
456
457   String postData;
458
459   postData = "UID=" + uid;
460
461   if (http.begin(client, "http://www.espochemmpa.com/acceso/android/consultaTicket.php")) {
462     http.addHeader("Content-Type", "application/x-www-form-urlencoded");
463     int httpCode = http.POST(postData);   //Send the request
464
465     if (httpCode > 0) {
466
467       if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY) {
468         String payload = http.getString();
469         Serial.println("Longitud: " + String(payload.length()));
470         Serial.println(payload);
471         Serial.println("Codigo HTTP: " + String(httpCode)); //Print HTTP return code
```

Ilustración 13-3: Envío de datos hacia el servidor

Realizado por: Sáez, Cristian, 2022

Como primer paso se conecta hacia la red wifi y se procede a declarar el objeto de clase HTTP, se define un variable en la cual se va a almacenar la información, la cual se desea enviar a la URI elegida previa respuesta favorable de conexión hacia dicho servidor.
En el proyecto se consideran 3 envíos de información al servidor mediante este protocolo que se detallan a continuación:
En el Anexo A desde la línea 451 hasta la 508 se procede a consultar si el tag que lee se encuentra registrado en la base datos, en el caso de existir devuelve el UID caso contrario da como respuesta el mensaje TAG NO REGISTRADA.

En la línea 511 hasta la 546 del anexo A se considera el registro de entrada del usuario hacia el establecimiento, asignándole un id al tag que se ha registrado, dando como resultado en el caso de haberse conectado correctamente, ¡un mensaje de “TICKET INGRESADO!" y en el caso de concretarse la conexión "ALGO SALIO MAL TICKET"

Siguiendo el proceso de ingreso se procede a ingresar los datos de ingreso al establecimiento como el UID y la hora de ingreso para posterior cálculo del monto a pagar con la hora de salida, dicho procedimiento se observa en el anexo A en las líneas 522 hasta la 590

Dado que el prototipo también considera un sistema de pago se agrega un historial de datos de usuario que ingresa al establecimiento como fecha, horario, tipo de vehículo, proceso detallado en las líneas 595 hasta la 633

Por último, se realiza una verificación del usuario para su ingreso, comparando el valor que tiene acreditado, el cual debe ser mayor al mínimo para poder ingresar al establecimiento, el cual se detalla en las líneas 638 hasta la 729

3.9.2 *Desarrollo e implementación del servidor y base de datos*

3.9.2.1 *Implementación del servidor*

Como se mencionó anteriormente el servidor elegido es Hostinger en el cual levantamos el servicio de hosting con los beneficios detallados en la ilustración 14-3

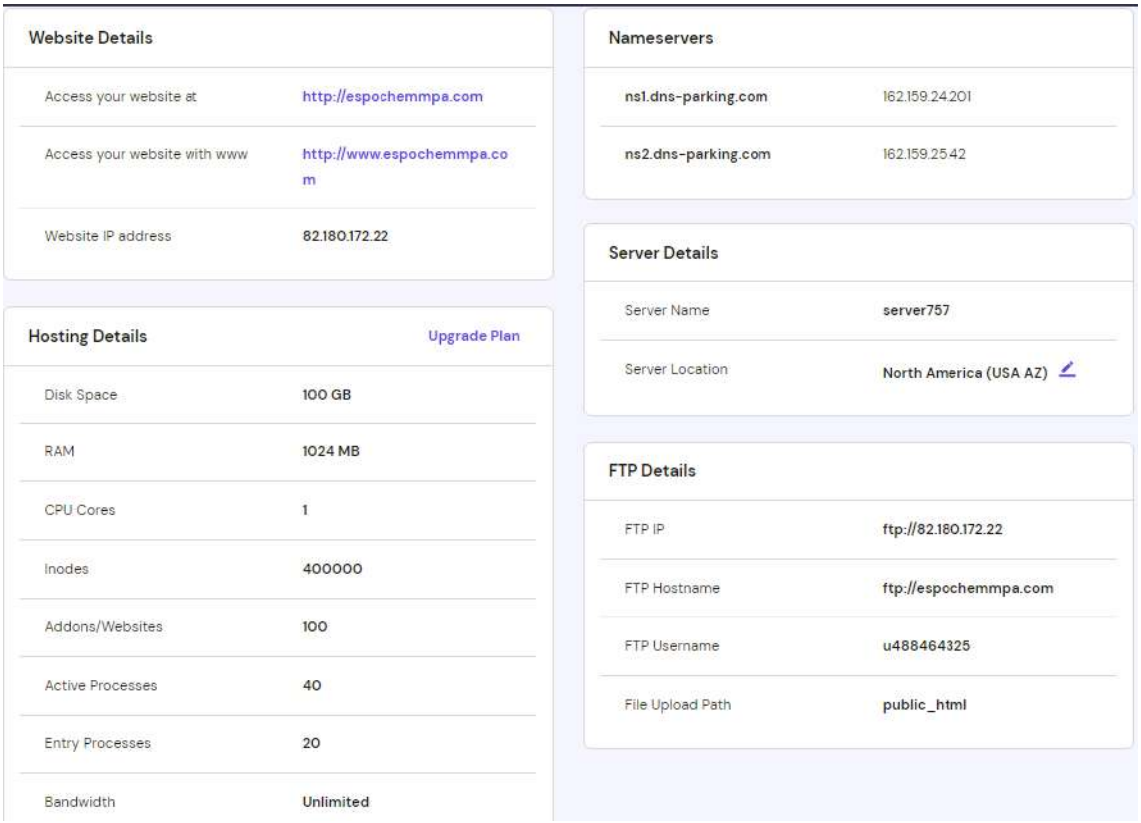


Ilustración 14-3: Detalles del servidor

Realizado por: Sáez, Cristian, 2022

Entre los aspectos a destacar tenemos el dominio principal http://espochemmpa.com, 82.180.172.22 como dirección IP, la localización del servidor en este caso América del Norte, 100 GB como espacio del disco, 1024 MB de memoria RAM y los datos del protocolo FTP para subir los archivos hacia el servidor.

En este marco, Hostinger permite subir diferentes tipos de archivos como .PHP a través de su panel, el cual permite administrar los archivos, esto se lo puede realizar con la ayuda del protocolo FTP con el programa FileZilla el cual permite subir los archivos, mediante credenciales que se muestran en la ilustración 15-3:

FTP Access	
FTP IP	ftp://82.180.172.22
FTP hostname	ftp://espochemmpa.com
FTP username	u488464325
FTP port	21
Folder to upload files	public_html

Ilustración 15-3: Credenciales para conexión FTP

Realizado por: Sáez, Cristian, 2022

En la ilustración 15-3 se observa los archivos .PHP que permiten visualizar las interfaces web, interpretar las acciones o peticiones de otro software como el de la aplicación Android Studio, o el de la misma interfaz de administración, incluyendo librerías o complementos, que manejan la parte gráfica de la web

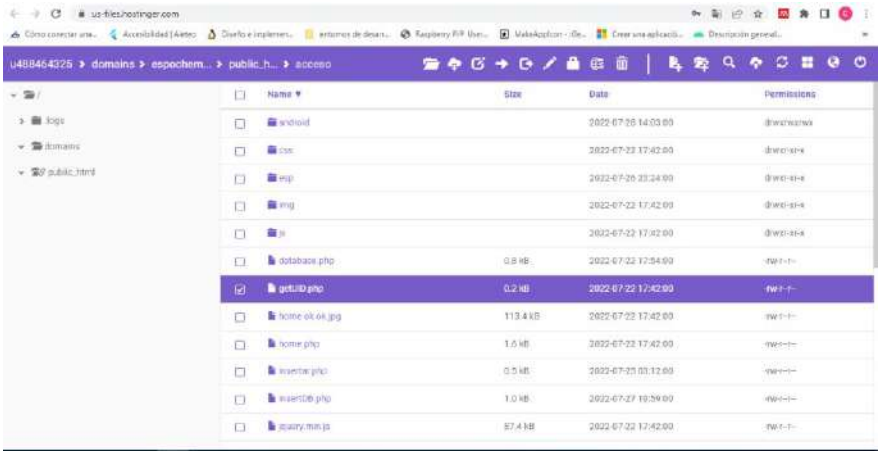


Ilustración 16-3: Archivos subidos al servidor

Realizado por: Sáez, Cristian, 2022

3.9.2.2 *Diseño e Implementación de la base de datos en MySQL*

El prototipo se conforma de algunas bases de datos en las cuales se almacena la siguiente información

Administrador: información del administrador y operario para ingresar al sistema

Parqueo: registro de la hora de ingreso, salida del establecimiento, así mismo del tiempo que utilizó el establecimiento y el valor a cancelar o debitar

Placa: en esta tabla se almacenará la información de la placa vehicular que se visualice al ingreso del establecimiento para autenticar al usuario

Tarjeta: en esta tabla se registrará la información de las tarjetas de crédito o débito que almacene el usuario

Usuario: en dicha tabla se registra toda la información de los usuarios que se ingresan desde la aplicación móvil

Usuarios: Se almacena la información mediante el cual el administrador va a poder ingresar al sistema

	Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
☐	administrador	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
☐	cargo	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
☐	parqueo	Examinar Estructura Buscar Insertar Vaciar Eliminar	26	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
☐	placa	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
☐	recarga	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
☐	registros	Examinar Estructura Buscar Insertar Vaciar Eliminar	26	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
☐	tarjeta	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
☐	usuario	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8mb4_general_ci	16.0 KB	-
☐	usuarios	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8mb4_general_ci	32.0 KB	-
	9 tablas	Número de filas	66	InnoDB	utf8mb4_unicode_ci	192.0 KB	0 B

Ilustración 17-3: Base de datos utilizadas en el prototipo

Realizado por: Sáez, Cristian, 2022

3.9.3 *Estructura de interfaz web de administrador*

El prototipo cuenta con una interfaz que permite al administrador manejar la información de los usuarios, con funciones tales como, editar, borrar o recargar saldo, dichos procesos se manejan con archivos php, con los que podemos manejar las bases de datos del servidor mediante diferentes sentencias o métodos que se detallarán en secciones posteriores, para lo cual como primer procedimiento se considera la conexión a la base de datos, para realizar esto se consideran las credenciales de la base de datos, detallas en el ANEXO B en las líneas 4 hasta la 7, posterior en la 22 dentro de la función pública denominada connect( ) se realiza la conexión a la base de datos mediante el controlador de PDO

3.9.3.1 *Control de acceso al sistema de administración*

El sistema al tener datos de usuarios y manejar sistema de recargas, debe poseer parámetros de seguridad que permita el ingreso de solo personal autorizado a dicha plataforma, para lo cual se realiza un sistema de login hacia la página principal, estas restricciones se basan en el uso de sesiones para cada archivo php, en la ilustración 18-4podemos observar la consulta que se realiza a la base de datos, para verificar que los datos ingresados en la página de login sean los mismo que se almacenaron con el id para el administrador, el único que tendrá las credenciales y el permiso para poder ingresar.

```php
session_start();
//$_SESSION['usuario']=$usuario;

$conexion=mysqli_connect("localhost","u488464325_admin2","Grab280895","u488464325_acceso");

$consulta="SELECT*FROM usuarios where usuario='$usuario' and contraseña='$contraseña'";
$resultado=mysqli_query($conexion,$consulta);

$filas=mysqli_fetch_array($resultado);

if($filas['id_cargo']==1){ //administrador
    $_SESSION['username']= $usuario;
    header("location:http://espochemmpa.com/acceso/login.php");
```

Ilustración 18-3: Archivo php para validación de datos de administrador

Realizado por: Sáez, Cristian, 2022

Cuando se ha abierto la sesión es necesario cerrar la misma para evitar que personas ajenas puedan manipular los datos, para lo cual se diseña un archivo que manejará los cierres de sesión, el mismo que podrá ser llamado cuando sea necesario, para realizar este proceso se utiliza el inicio de sesión con ssession_star y session_destroy par terminar la misma, la que redigirá a la página de autenticación

```php
<?php
session_start();
session_destroy();
header("location: index.html");
exit();
?>
```

Ilustración 19-3: Archivo php cierre de sesión

Realizado por: Sáez, Cristian, 2022

3.9.3.2 *Interfaz para el manejo de datos de usuario*

El administrador tiene acceso a diferentes interfaces, entre ellas la que permite la visualización y el manejo de datos, dando paso a editar, borrar y recargar saldo a los usuarios, para esto se construye la interfaz gráfica mediante html y el manejo de la misma mediante archivos php que manejan las bases de datos. La gráfica 20-3 muestra la sentencia para la selección de datos desde la base datos usuario, la que será impresa en la interfaz.

```php
<?php
 include 'database.php';
 $pdo = Database::connect();
 $sql = 'SELECT * FROM usuario ORDER BY nombre ASC';
 foreach ($pdo->query($sql) as $row) {
         echo '<tr>';
         echo '<td>'. $row['uid'] . '</td>';
         echo '<td>'. $row['nombre'] . '</td>';
         echo '<td>'. $row['apellido'] . '</td>';
         echo '<td>'. $row['email'] . '</td>';
         echo '<td>'. $row['vehiculo'] . '</td>';
         echo '<td>'. $row['placa'] . '</td>';
         echo '<td>'. $row['saldo'] . '</td>';

         echo '<td><a class="btn btn-success" href="user data edit page.php?uid='.$row['uid'].'">Editar</a>';
         echo ' ';
         echo '<a class="btn btn-danger" href="user data delete page.php?uid='.$row['uid'].'">Borrar</a>';
         echo ' ';
         echo '<a class="btn btn-info" href="recargar.php?uid='.$row['uid'].'">Recargar</a>';
         echo '</td>';
         echo '</tr>';
 }
 Database::disconnect();
 ?>
```

Ilustración 20-3: Archivo php para visualizar los datos

Realizado por: Sáez, Cristian, 2022

Para la editar la información de los usuarios se utiliza la sentencia $sql = "SELECT * FROM usuario where uid = ?"; para llamar la información de forma específica, la misma que puede ser visualizada y modificada, el anexo C detalla las sentencias y parámetros para la capa visual

La ilustración 21-3 muestra las sentencias que se utilizan para editar la información de los usuarios, primero se extrae la información que se ingresó en la interfaz gráfica mediante el método POST, los cuales a través de UPDATE van a ser modificados en la base de datos, devolviendo los resultados a la página principal

```php
    if ( !empty($_POST)) {
        // keep track post values
        $uid = $_POST['uid'];
        $nombre = $_POST['nombre'];
        $apellido = $_POST['apellido'];
        $email = $_POST['email'];
        $vehiculo = $_POST['vehiculo'];
        $saldo = $_POST['saldo'];
        $placa = $_POST['placa'];

        $pdo = Database::connect();
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $sql = "UPDATE usuario set nombre = ?, apellido = ?, email =?, vehiculo = ?,
            saldo =?, placa = ? WHERE uid = ?";
        $q = $pdo->prepare($sql);
        $q->execute(array($nombre,$apellido,$email,$vehiculo, $saldo,$placa,$uid));
        Database::disconnect();
        header("Location: user_data.php");
    }
?>
```

Ilustración 21-3: Archivo php para editar los datos de los usuarios

Realizado por: Sáez, Cristian, 2022

Para la eliminación de usuarios se utiliza la sentencia $sql = "DELETE FROM usuario WHERE uid = ?"; antes de realizar esta acción es necesario llamar a la base de datos, por lo que mediante la sentencia GET se llama al UID perteneciente a dicho usuario

```php
<?php
    require 'database.php';
    $uid = 0;

    if ( !empty($_GET['uid'])) {
        $uid = $_REQUEST['uid'];
    }

    if ( !empty($_POST)) {
        // keep track post values
        $uid = $_POST['uid'];

        // delete data
        $pdo = Database::connect();
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $sql = "DELETE FROM usuario  WHERE uid = ?";
        $q = $pdo->prepare($sql);
        $q->execute(array($uid));
        Database::disconnect();
        header("Location: user_data.php");

    }
?>
```

Ilustración 22-3: Archivo php para eliminar usuarios

Realizado por: Sáez, Cristian, 2022

Para realizar las recargas a los usuarios, se considera una columna adicional con el valor de la recarga el mismo que será añadido al valor actual de saldo existente, actualizando mediante el método UPDATE dicho valor en la tabla principal

```php
<?php
    require 'database.php';

    $uid = null;
    if ( !empty($_GET['uid'])) {
        $uid = $_REQUEST['uid'];
    }

    if ( !empty($_POST)) {
        // keep track post values
        $saldo = $_POST['saldo'];
        $recarga = $_POST['recarga'];
        $sal= $saldo + $recarga;


        $pdo = Database::connect();
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $sql = "UPDATE usuario set saldo =? WHERE uid = ?";
        $q = $pdo->prepare($sql);
        $q->execute(array($sal,$uid));
        Database::disconnect();
        header("Location: user_data.php");
    }
?>
```

Ilustración 23-3: Archivo PHP para recargar saldo a usuarios

Realizado por: Sáez, Cristian, 2022

3.9.3.3 *Interfaz para el ingreso de usuarios que no poseen smartphones con tecnología NFC*

En el caso de de que los usuarios no posean dispositivos ingeligentes con la tecnologia NFC, se utilizan tarjetas o llaveros que tengan dicha tecnología, para poder registrarlos en la base de datos y puedan acceder a los servicios para dicho proceso se utiliza el metodo POST que recogerá la información que se ingresa a la plataforma , la misma que mediante la sentencia INSERT INTO ingresará la infromación en la base de datos

```php
<?php

    require 'database.php';

    if ( !empty($_POST)) {
        // keep track post values
        $uid = $_POST['uid'];
        $nombre = $_POST['nombre'];
        $email = $_POST['email'];
        $vehiculo = $_POST['vehiculo'];
        $saldo = $_POST['saldo'];


        // insert data
        $pdo = Database::connect();
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $sql = "INSERT INTO usuario (uid,nombre,email,vehiculo,saldo) values (?, ?, ?, ?, ?)";
        $q = $pdo->prepare($sql);
        $q->execute(array($uid,$nombre,$email,$vehiculo,$saldo));
        Database::disconnect();
        header("Location: user data.php");
    }

?>
```

Ilustración 24-3: Archivo php para registrar usuarios

Realizado por: Sáez, Cristian, 2022

3.9.4 *Estructura de la aplicación en el dispositivo inteligente*

Para la aplicación móvil se considera la emulación de tarjeta inteligente basada en host, la misma que considera diferentes capas, para el registro del usuario, la autenticación para ingresar a la plataforma que le permite al usuario registrar el ingreso, estancia y salida del establecimiento, adicional se considera la emulación de una tarjeta de crédito para realizar el pago, para lo que se establece una capa que permita el registro de los datos de dichas tarjetas implementado un pin de seguridad la misma que se utilizará para aprobar el pago, todos estos procesos dependen de conexión activa a internet ya que se conecta al servidor que interactúa con la otra plataforma.

3.9.4.1 *Android Manifest*

En el archivo Android Manifest se consideran los permisos que permiten el correcto funcionamiento de la aplicación y puedan acceder a servicios como internet y NFC, adicional se establecen los parámetros para el funcionamiento de la aplicación como la referencia de las

actividades que cuenta la aplicación, restricciones o permisos para que se pueda instalar solo en dispositivos que posean la tecnología NFC, la ilustración 25-3 muestra el código de la aplicación.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.nfc.hcefinal">

    <uses-permission android:name="android.permission.NFC" /> <!-- Requiremen
    <uses-feature
        android:name="android.hardware.nfc.hce"
        android:required="true" />

    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="hceFInal"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.HceFInal"
        android:usesCleartextTraffic="true"
        tools:targetApi="31">
        <activity
```

Ilustración 25-3: Codificación de Android Manifest

Realizado por: Sáez, Cristian, 2022

Dentro de este mismo archivo se deben considerar declarar servicios, para la aplicación se establece el de emulación de tarjeta inteligente, que entre sus parámetros se detallan, nombre de la clase, exported para que puede interactuar con otras aplicaciones, permission para poder vincular al servicio de NFC, el intent-filter que activa la acción HOST_APDU_SERVICE cuando un lector externo trata de leer la tarjeta del dispositivo y por último el meta-data que establece los servicios a los cuales llamar de acuerdo al AID con que el lector trata de comunicarse

```
        <service
            android:name=".HostCardEmulatorService"
            android:exported="true"
            android:permission="android.permission.BIND_NFC_SERVICE">
            <intent-filter>
                <action android:name="android.nfc.cardemulation.action.HOST_APDU_SERVICE" />
            </intent-filter>

            <meta-data
                android:name="android.nfc.cardemulation.host_apdu_service"
                android:resource="@xml/apduservice" />
        </service>
    </application>
```

Ilustración 26-3: Servicios para la emulación de tarjeta inteligente

Realizado por: Sáez, Cristian, 2022

3.9.4.2 *Host APDU Service*

El archivo Host APDU Service es muy importante ya que mediante el fitlrado de AID que se define en este apartado, el que podrá ser disparado si este AID tiene llamado o está siendo seleccionado por el lector de tarjetas de este tipo, en este caso se define por F0010203040506, el mismo que se establece de igual manera en el ESP en secciones anteriores

```
<?xml version="1.0" encoding="utf-8"?>

<host-apdu-service xmlns:android="http://schemas.android.com/apk/res/android"
    android:description="HCE Service"
    android:requireDeviceUnlock="false"

    >
    <aid-group android:description="Aid Description"
        android:category="other"

        >
        <aid-filter android:name="F0010203040506"/>

    </aid-group>
</host-apdu-service>
```

Ilustración 27-3: Servicio HOST APdu

Realizado por: Sáez, Cristian, 2022

3.9.4.3 *Generador código HCE*

Dentro del modelaje del prototipo se considera generar un UID estático para cada usuario con el fin de que cada usuario se pueda autenticar tanto en el dispositivo como en el vehículo, para generar dicho UID se consideran caracteres hexadecimales que de forma aleatoria se generaran cuando sea la etapa de registro , definiendo una longitud de 6 dígitos para la misma

```
private final String [] ALLOWED_CHARACTERS = {"0","1","2","3","4","5","6","7","8","9","A","B","C","D","E","F"};
    Random random = new Random();
    StringBuilder sb = new StringBuilder(longitud);

    for (int i=0; i<longitud;i++){

        sb.append(ALLOWED_CHARACTERS[random.nextInt(ALLOWED_CHARACTERS.length)]);
    }

    return sb.toString();
```

Ilustración 28-3: Generación de código HCE

Realizado por: Sáez, Cristian, 2022

3.9.4.4 *Registro de usuarios*

```
if(TextUtils.isEmpty(nombres.getText())|| TextUtils.isEmpty(apellidos.getText())||
        TextUtils.isEmpty(UID.getText())||TextUtils.isEmpty(correo.getText())||TextUtils.isEmpty(pass.getText()) ||
        TextUtils.isEmpty(placa.getText())) {
    Toast.makeText( context: registro_usuario.this, text: "Primero llene los campos por favor",Toast.LENGTH_SHORT).show();
}
else {

    registrarUsuario( URL: "https://www." + direccionIP + urlServidor + "insertarUsuario.php");
        StringRequest stringRequest1 = new StringRequest(Request.Method.POST, URL,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {

                        if(response.equals("ok")){

                            showToast( msg: "Usuario Registrado correctamente");

$sentencia = $conexion->prepare("INSERT INTO usuario (uid, nombre, apellido, email, vehiculo, contrasena, saldo, placa) VALUES (?,?,?,?,?,?,?,?)");
```

Ilustración 29-3: Registro de usuario

Realizado por: Sáez, Cristian, 2022

Para el registro de usuarios se utiliza el método POST del protocolo HTTP, que previa verificación de que todos los campos de registro estén ingresados correctamente se procede a ingresar en la base de datos esto mediante archivos php que se suben al servidor, proceso que se puede visualizar en la ilustración 29-3

3.9.4.5 *Validación de usuario para inicio de sesión*

Al ser un prototipo que maneja datos confidenciales y dinero, se considera el inicio de sesión para poder acceder al servicio, para lo cual se ingresan los datos en la interfaz y se procede a realizar una verificación de la información que registró en etapas anteriores, para determinar que es el mismo usuario el que desea ingresar, para esto en una archivo php se selecciona el usuario y la contraseña almacenados y se devuelve en un mensaje json para que en el dispositivo móvil estos puedan ser igualados y permitir o denegar el ingreso del usuario

```
if(!etPass.getText().toString().isEmpty() && !etEmail.getText().toString().isEmpty() ){
    validarUsuario( URL: "http://www."+direccionIP+"validar_usuario.php",etEmail.getText().toString().trim(),etPass.getText().toString().trim());
$sentencia=$conexion->prepare("SELECT * FROM usuario WHERE email=? AND contrasena=?");
```

Ilustración 30-3: Validación inicio de sesión de usuario

Realizado por: Sáez, Cristian, 2022

3.9.4.6 *Registro de tarjeta de crédito para realizar el pago*

Para realizar el pago por el servicio se plantea el débito del saldo de una tarjeta de crédito, por lo que se plantea el ingreso de los datos de la misma a la plataforma, los datos que se consideran son un id de usuario, el número de la tarjeta, el mes, el año, el CVV (código valor de validación), el

nombre del usuario de la tarjeta y como mecanismo de seguridad se implementa un PIN de 4 dígitos que permitirá confirmar el pago del servicio, para este registro se utiliza el método POST en el programa e INSERT INTO en el archivo php para el ingreso de información a la base de datos. Adicional se establece la creación de varias tarjetas en caso de que el usuario así lo disponga, la ilustración 31-3 muestra las consultas y métodos utilizados para este procedimiento

```
StringRequest stringRequest1 = new StringRequest(Request.Method.POST, URL,
params.put( k: "idUsuario", stridUsuario);
params.put( k: "numero", strnumeroTarjeta);
params.put( k: "mes", strMes);
params.put( k: "anio", strAnio);
params.put( k: "cvv", strCVV);
params.put( k: "nombre", strnombreTarjeta);
params.put( k: "pin", strPin);

$sentencia = $conexion->prepare("INSERT INTO tarjeta (id_usuario, numero, mes, anio, cvv, nombre, pin,seleccion) VALUES (?,?,?,?,?,?,?,?,?)");
```

Ilustración 31-3: Registro de datos de tarjetas a la base de datos

Realizado por: Sáez, Cristian, 2022

3.9.4.7 *Registro de tarjeta de crédito para realizar el pago*

Ya que el usuario puede seleccionar la tarjeta de su preferencia se realiza una actualización en la base de datos de la tarjeta seleccionada por el usuario para que pueda ser utilizada al momento de cancelar por el servicio de estacionamiento, para esto primero se realiza una consulta mediante el método POST al archivo consultaTarjetas.php el cual devuelve todas las tarjetas registradas por el usuario, estos datos son devueltos en archivos json y visualizados en la aplicación, luego de que se selecciona una, esta se actualiza mediante el método POST a través del archivo editarSeleccionTarjeta.php que realiza un UPDATE en la base de datos

```
StringRequest stringRequest1 = new StringRequest(Request.Method.POST, URL ,
mostrarTarjetas( URL: "http://www."+direccionIP+"consultaTarjetas.php", String.valueOf(idUsuario));
$sentencia=$conexion->prepare("SELECT * FROM tarjeta WHERE id_usuario=? ORDER BY id DESC");
StringRequest stringRequest1 = new StringRequest(Request.Method.POST, URLActualizar

actualizarSeleccionTarjeta( URLActualizar: "https://www." + direccionIP + "editarSeleccionTarjeta.php",id, seleccion: 1);
$sentencia=$conexion->prepare("UPDATE tarjeta SET seleccion=? WHERE id=?");
```

Ilustración 32-3: Selección de tarjeta de crédito a utilizar

Realizado por: Sáez, Cristian, 2022

3.9.4.8 *Visualización de registro de parqueo*

En la sección 3.8.1.3 se detalla el registro y envío de datos cuando el usuario ingresa al establecimiento, en la parte de la aplicación móvil se realiza una consulta del parqueo del usuario

para poder visualizar en la interfaz gráfica, para esto se realiza una consulta a la lista de parqueos tanto en la aplicación como en el archivo consultaListasParqueos,php que mediante la sentencia SELECT consulta los parqueos registrados, devolviendo los datos de parqueos completados o los que se cursan en el proceso

```
mostrarParqueos( URL: "http://www."+direccionIP+"consultaListasParqueos.php", strUID);

$sentencia=$conexion->prepare("SELECT * FROM parqueo WHERE uid=? ORDER BY id DESC");
```

Ilustración 33-3: Visualización de datos de parqueo

Realizado por: Sáez, Cristian, 2022

3.9.4.9 *Pin de seguridad para aprobación de pago*

El pago es uno de los aspectos más importantes del prototipo, el usuario cuando va a salir del establecimiento necesita finalizar el proceso, por lo que debe cancelar por el servicio, al intervenir recursos económicos se debe hacer validación de los datos de la tarjeta del usuario, para esto se realiza una consulta hacia la base de datos mediante validar_usuario.php el cual devolverá los datos del usuario y de la tarjeta, esta consulta se hace en base a el email y contraseña guardados. En el caso de que los datos sean correctos y haya saldo disponible se publica un mensaje en el tópico NFC_OUT mediante el protocolo MQTT que permitirá la apertura de la barrera, procedimiento que se puede visualizar en la ilustración 34-3

```
private void validarUsuario(String URL, String emailGuardado, String passGuardado){
    StringRequest stringRequest = new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
  client = new MqttAndroidClient(this.getApplicationContext(), serverURI: "tcp://broker.emqx.io:1883", clientId);

 SharedPreferences preferences = getSharedPreferences( name: "APPHCE", Context.MODE_PRIVATE);

 pinSeguridad   = preferences.getString( s: "pin", s1: null);
 strcorreo    = preferences.getString( s: "correo", s1: null);
 strpass      = preferences.getString( s: "pass", s1: null);
 direccionIP = preferences.getString( s: "ip", s1: null);
 numeroTarjeta = preferences.getString( s: "tarjeta", s1: null);
 placa = preferences.getString( s: "placa", s1: null);
 strnombre    = preferences.getString( s: "nombre", s1: null);
 strapellido = preferences.getString( s: "apellido", s1: null);
 uid = preferences.getString( s: "UID", s1: null);

String saldo =  json_data.getString( name: "saldo");

 validarUsuario( URL: "http://www."+direccionIP+"validar_usuario.php",strcorreo,strpass);

 enviarSMSMQTT( topic: "NFC_OUT", payload: "ON");
```

Ilustración 34-3: Validación de datos para aprobar el pago

Realizado por: Sáez, Cristian, 2022

En el caso de encontrar alguna falla en la validación de usuarios o cuando el PIN ingresad no sea el correcto durante 3 intentos, se envía un correo electrónico al administrador informando del defecto, para este procedimiento se realiza una consulta POST hacia el servidor que vincula con los datos del usuario que genera el error para enviar con la cabecera que se observa en la ilustración 35-3

```
StringRequest stringRequest = new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {

if(contadorIntentos  < 3){
    showToast( msg: "Tiene " + String.valueOf(contadorIntentos) + " intentos fallidos" );
}

else{

    showToast( msg: "ALERTA DE SEGURIDAD");
    enviarCorreo( URL: "http://www."+direccionIP+"correo.php");
```

```
$to      = 'cristiansaez2871@gmail.com';
$subject = 'TESIS HCE NFC SEGURIDAD';
$message = "Se detecto un problema de seguridad del siguiente usuario -> "."\n\n".

"DATOS"."\n".
"UID NFC: ".$uid."\n".
"Nombre: ".$nombre."\n".
"Apellido :".$apellido."\n".
"Placa: ".$placa."\n".
"Tarjeta de Crédito: ".$tarjeta;


$headers = 'From: hceAndroid@espoch.edu.com'       . "\r\n" .
           'Reply-To: hceAndroid@espoch.edu.com' . "\r\n" .
           'X-Mailer: PHP/' . phpversion();

mail($to, $subject, $message, $headers);
```

Ilustración 35-3: Procedimiento en el caso de ser inválido algún dato

Realizado por: Sáez, Cristian, 2022

3.9.5 *Proceso para el reconocimiento de placas vehiculares*

El sistema considera un método de autenticación para asegurar que el vehículo que va a ingresar sea el mismo que se registró en la plataforma, para dicho proceso se diseña un sistema que permita reconocer la placa vehicular que identifica al usuario que realiza la petición de ingreso.

El reconocimiento de placas se realiza con el lenguaje de programación Python, en el cual mediante el protocolo RTSP (Protocolo de transmisión en tiempo real) se toma el video de un dvr que procesa la imagen de una una cámara externa en este caso se opta por una de tipo varifocal de la marca Hickvision, para detectar los caracteres se debe extraer un fotograma de dicha señal de entrada que deberá guardarse, que posteriormente mediante el uso de una la api de google vision permitirá el reconocimiento de texto que se encuentra en la imagen, realizando un filtrado que permita obtener el valor de la placa vehicular, el anexo E muestra las configuraciones

necesarias para que se cumpla este proceso, como requisito se crea el archivo .json en la consola de Google cloud, que posee la información de autenticación para acceder a los recursos, el mismo que devolverá una cadena de texto con su ubicación. La ilustración 36-3 muestra la autenticación al api, la creación del cliente y el texto de respuesta con lo detectado

```
37  # Setup google authen client key
38  os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = 'placa-emmpa-3c44932c28ec.json'

67      # Create google vision client
68      client1 = vision.ImageAnnotatorClient()

76      # Recognize text
77      response = client1.text_detection(image=image)
78      texts = response.text_annotations
```

Ilustración 36-3: API de Google visión para detección óptico de caracteres

Realizado por: Sáez, Cristian, 2022

El prototipo necesita estar activo todo el tiempo, por lo que se implementa el protocolo MQTT por su comunicación efectiva y liviana ya que la cantidad de datos es mínima, para lo cual se crean tópicos dentro de bróker, en la cual al leer un UID válido en el módulo lector ESP8266 éste publique ese dato en el tópico "SALIDA1", a su vez el script de Python se suscribirá para que active el proceso para el reconocimiento de placas, en el caso de ser iguales este publicará en el tópico "ENTRADA 1" un mensaje de aprobación el mismo que será suscrito en el módulo lector con el fin de transformar en un pulso que activará un relé para levantar la barrera vehicular que permitirá el acceso al establecimiento, la ilustración 37-3 observar los tópicos definidos en el servidor MQTT y los datos que se envían y suscriben en los mismos, para este prototipo se utiliza el corredor broker.emqx.io en el puerto TCP 1883,

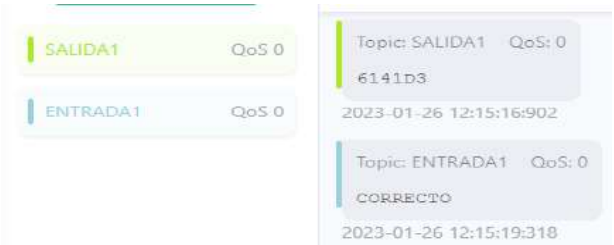


Ilustración 37-3: Tópicos establecidos para comunicación MQTT

Realizado por: Sáez, Cristian, 2022

Para poder verificar que la placa detectada es la misma que la que se encuentra almacenada en la base de datos del usuario, se realiza una consulta hacia el servidor para extraer el dato requerido mediante la sintaxis SELECT, programación que se puede observar en el anexo E en las líneas

106 a la 114, posterior a esto sí coinciden los caracteres, se levantará la barrera, caso contrario no permitirá el acceso.

```
114         sql = "SELECT placa FROM usuario WHERE uid= '%s'" % strmqtt
123             if(str(x) == strplaca):
124                 client.publish(topic1, "CORRECTO")
125                 print("LAS PLACAS COINCIDEN")
126
153     client.publish("ENTRADA1", "CONEXION LISTA")
154     client.subscribe("SALIDA1")
```

Ilustración 38-3: Proceso de comparación de placas

Realizado por: Sáez, Cristian, 2022

CAPITULO IV

4 RESULTADOS

En base a la documentación detallada en el capítulo anterior se tiene como resultado la implementación del dispositivo lector NFC construido con tarjetas de desarrollo, los mecanismos de seguridad, la interfaz del administrador, la aplicación para el usuario para el registro y pago del servicio, el reconocimiento de placas vehiculares como mecanismo de autenticación. El capítulo siguiente muestra los resultados de la implementación de todo el mecanismo implementado

4.1 Módulo lector de dispositivos NFC

La etapa inicial para el funcionamiento del prototipo es la lectura del dispositivo NFC, mediante la programación e implementación de los módulos de desarrollo se obtiene el lector que se muestra en la ilustración 1-4

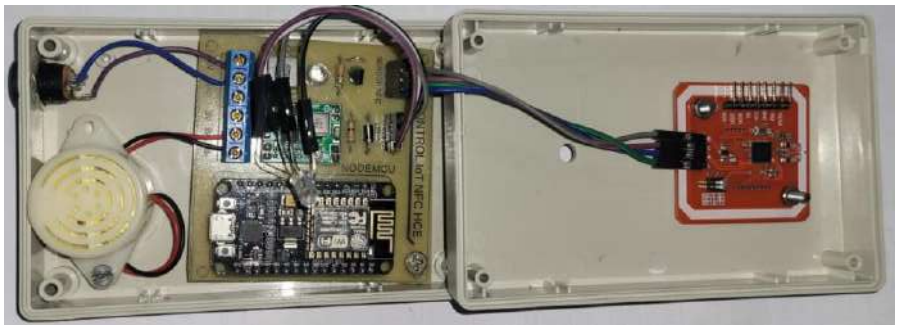


Ilustración 1-4: Dispositivo lector NFC

Realizado por: Sáez, Cristian, 2022

4.1.1 *Registro de datos en la entrada*

Al realizar el tag con el módulo lector y enviar los mismos por el Gateway ESP8266, se realiza una consulta para poder visualizar el correcto registro de entrada del usuario por lo que en la ilustración 2-4 se pude observar, el UID detectado la hora de ingreso y los datos pertenecientes a dicho dispositivo, de igual manera verificamos que existe una respuesta satisfactoria de la petición HTTP esto mediante la respuesta de estado 200 que indica satisfactorio, de igual manera se realiza para confirmar el ingreso del ticket que se asigna mediante un id para identificar el proceso de estacionamiento del usuario, en la ilustración 2-4 se puede verificar el ingreso de manera correcta a la base de datos, el ingreso del id 127 que se asigna al usuario que tiene el UID 6141D3

```
Connected with IP: 192.168.116.16

Listo Conexiones de red
LISTOOOO
Esperando por tarjeta ISO14443A
The client mqttx_4a2e027c5C:CF:7F:A4:31:91 connects to the public mqtt broker
Message arrived in topic: ENTRADA1
Message:CONEXION LISTA

-----------------------
UID: 6141D3  HORA: 1675137696
CONEXION EXITOSA!!!
Longitud: 212
[{"id":3,"uid":"6141D3","nombre":"laurita","apellido":"gomez","email":"laura614@gmail.com","vehicul
Codigo HTTP: 200
ID: 3
DATO: 19.00
CARRO: Pequeu00flo 0-2 tn
Payload: ok
Codigo HTTP: 200
REGISTRO PARQUEO SUBIDOS!
Longitud: 101
[{"id":133,"uid":"6141D3","ingreso":"1675137696","salida":"------","tiempo":"------","costo":"0.00"
Codigo HTTP: 200
ID ticket: 133
Payload: ok
Codigo HTTP: 200
TICKET INGRESADO!
```

id	uid	ingreso	salida	tiempo	costo
155	21523617179	1675245881	------	------	0.00
156	18814617022	1675245889	------	------	0.00
157	92125233169	1675245898	------	------	0.00
158	2439216945	1675245905	------	------	0.00
159	13B17103	1675245925	------	------	0.00
160	6141D3	1675245972	------	------	0.00

Ilustración 2-4: Registro y visualización de datos al ingreso del establecimiento

Realizado por: Sáez, Cristian, 2022

4.1.2 *Registro de datos en la salida*

Cuando el usuario desea salir del establecimiento se debe realizar la verificación que se detalló en secciones anteriores, ya culminado el servicio de parqueo, se registra la hora de salida y en base a la diferencia de tiempo y de acuerdo con la categorización que se detalló, se obtiene el valor a cancelar, datos que se pueden observar en la ilustración 3-4

```
UID: 6141D3  HORA: 1675139033
CONEXION EXITOSA!!!
Longitud: 214
[{"id":3,"uid":"6141D3","nombre":"laurita","apellido":"gomez","email":"laura614@gmail.com","vehi
Codigo HTTP: 200
ID: 3
DATO: 19.00
CARRO: Pequeu00flo 0-2 tn
TICKET: 133
SALDO: 19.00
0
si es pequeño
Longitud: 101
[{"id":133,"uid":"6141D3","ingreso":"1675137696","salida":"------","tiempo":"------","costo":"0.
Codigo HTTP: 200
hora ingreso: 1675137696
hora salida: 1675139033
diasTranscurridos: 0 , horasTranscurridos: 0 , minutosTranscurridos: 22 , segsTranscurridos: 17
carro antes de comparar: PEQUEU00FlO 0-2 TN
SI ES PEQUENO!!!
SALDO: 18.50
DESCUENTO: 0.50
Payload: ok
Codigo HTTP: 200
PARQUEO EDITADO!
Payload: ok
Codigo HTTP: 200
TICKET INGRESADO!
Payload: ok
Codigo HTTP: 200
SALDO ACTUALIZADO!
Payload: ok
Codigo HTTP: 200
DATOS SUBIDOS!
YA PUBLIQUE TODO!
```

id	uid	ingreso	salida	tiempo	costo
133	6141D3	1675137696	1675139033	0:22:17	0.50

Ilustración 3-4: Registro y visualización de datos al salir del establecimiento

Realizado por: Sáez, Cristian, 2022

4.1.3 *Apertura de la barrera vehicular*

Para la apertura de la barrera vehicular se considera el pago por el servicio de estacionamiento, en el caso de que se cumpla con este procedimiento de manera correcta, cuando se debite del saldo de usuario la cantidad que corresponda, se publicará a un módulo ESP8266 un mensaje de publicación con la palabra “ON” el cual mediante un relé activa el brazo que controla el acceso al establecimiento.

```
.....................................................................................
Connected with IP: 192.168.116.101

Listo Conexiones de red
The client mqttx_4a2e027cC8:C9:A3:64:D0:FD connects to the public mqtt broker
Message arrived in topic: NFC_OUT
Message:ON

-----------------------
```

Ilustración 4-4: Registro y visualización de datos al salir del establecimiento

Realizado por: Sáez, Cristian, 2022

1.1. Funcionamiento de interfaz web del administrador

La interfaz web del administrador se encuentra protegido por un sistema de autenticación con un login de usuario y contraseña, lo que brinda seguridad al usuario y al administrador como se observa en la ilustración 5-4.

Ilustración 5-4: Pantalla principal y sistema de login de interfaz web

Realizado por: Sáez, Cristian, 2022

Una vez ingresado en la interfaz web del administrador se encuentran diferentes pantallas contando con privilegios, la primera que despliega los usuarios registrados, donde se podrán realizar diferentes acciones como editar, eliminar, recargar y en la segunda pantalla permite registrar nuevos usuarios que no posean dispositivos inteligentes compatibles con la tecnología NFC, en este caso se registran llaveros o tarjetas RFID

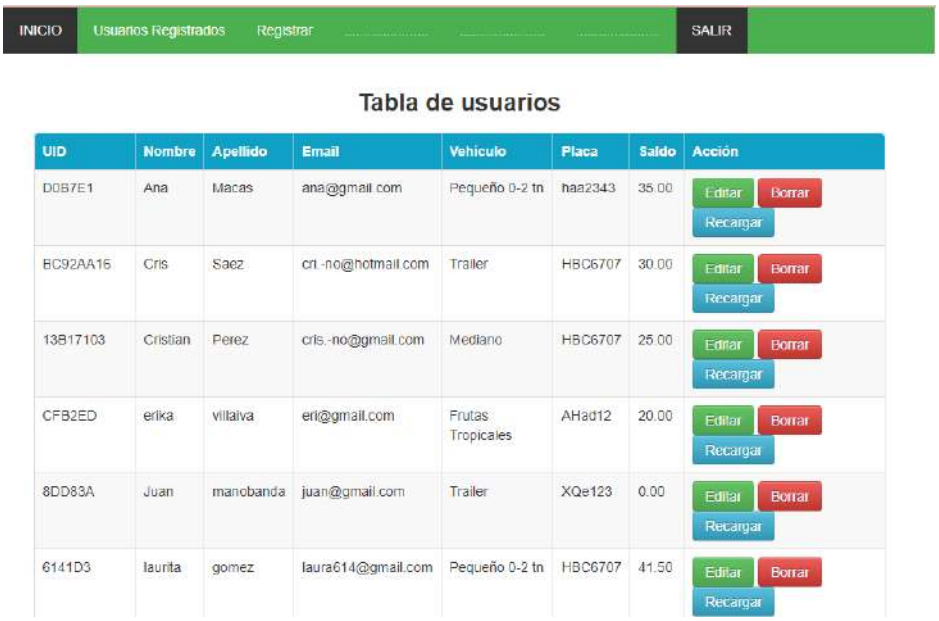


Ilustración 6-4: Pantalla de usuarios registrados

Realizado por: Sáez, Cristian, 2022

Dentro de las opciones del manejo de usuarios existe el botón editar que se encuentra al lado derecho de la pantalla principal, que al dar click se despliega otra interfaz como se observa en la ilustración 7-4 donde se puede editar los datos de los usuarios como Nombre, Apellido, Email y el tipo de vehículo que esta persona tiene porqué de acuerdo a eso se paga un valor diferente, en el caso de desear el cambio de tipo de vehículo este está sujeto a verificación por parte del administrador y esta interfaz es la única que ofrece dicha función, los campos del UID y el saldo no son editables ya que en el caso del UID este funciona como el identificador único del usuario y en caso del saldo se bloquea esta posibilidad porque no se encuentra en la interfaz que tienen esta función

Ilustración 7-4: Editar dato de usuario

Realizado por: Sáez, Cristian, 2022

En la interfaz de recargar saldo al usuario, se encuentran los datos más importantes del usuario seleccionado y permite la opción de acreditar saldo al usuario, estos valores son de libre ingreso y al aceptar la recarga esta se sumará al valor ya existente y devolverá a la interfaz principal.

Ilustración 8-4: Recargar usuario

Realizado por: Sáez, Cristian, 2022

Finalmente se encuentra la opción que permite eliminar usuarios, donde al seleccionar aparece un mensaje de advertencia donde nos peguntará si estamos seguros de que se desea eliminar y se procederá a aceptar o rechazar, en el caso de aceptar la eliminación será permanente de toda la base de datos.

Ilustración 9-4: Eliminar usuario

Realizado por: Sáez, Cristian, 2022

1.2. Funcionamiento de la interfaz de la aplicación móvil para usuario

La instalación de la aplicación en el teléfono inteligente se debe hacer en un sistema con un sistema operativo superior a la versión debido a que se usa la emulación de tarjeta con Host (HCE), en la ilustración 10-4 se observa la pantalla inicial de la aplicación, donde el primer paso es que el cliente se registre, los que ya han realizado este paso deben ingresar el correo electrónico y su respectiva contraseña para ingresar a la aplicación.

Ilustración 10-4: Sistema de Login para usuario

Realizado por: Sáez, Cristian, 2022

Una vez que el usuario realice el respectivo login, aparece la pantalla del balance y los movimientos respectivos que permiten observar el estado del parqueo y el historial de estos. En la parte superior, en el icono se puede editar los datos de la persona y realizar otras acciones como agregar el método de pago.

Ilustración 11-4: Pantalla de movimientos y balance

Realizado por: Sáez, Cristian, 2022

En la pantalla de editar datos de usuario se pueden modificar ciertos campos como el nombre, apellido, correo electrónico, la placa del vehículo y la contraseña, los campos como UID y el tipo de vehículo no son modificables, el tipo de vehículo solo puede modificar el administrador. En los 3 puntos en la parte superior derecha se encuentra el método de pago

Ilustración 12-4: Pantalla editar datos de usuario

Realizado por: Sáez, Cristian, 2022

Para el registro de la tarjeta de crédito se deben ingresar datos de esta, como el número, la fecha de expiración el, CVV (Código de valor de verificación), el nombre de la tarjeta y para agregarle una mayor seguridad se debe registrar un pin de verificación.

Ilustración 13-4: Registro de tarjetas de crédito

Realizado por: Sáez, Cristian, 2022

Se pueden agregar diferentes tarjetas válidas para que se realicen las transacciones en la ilustración 14-4 se observa que no tiene restricción para un número de tarjetas validas, se debe seleccionar una tarjeta para poder cancelar del servicio

Ilustración 14-4: Lista de tarjetas

Realizado por: Sáez, Cristian, 2022

La ilustración 15-4 muestra la interfaz que se despliega para realizar el pago del servicio, en esta se muestra que el usuario debe ingresar el PIN de seguridad que se registró previamente con su tarjeta de crédito, si es correcto se culmina el servicio de estacionamiento y se debita de la cuenta la cantidad calculada

Ilustración 15-4: PIN de seguridad para realizar el pago

Realizado por: Sáez, Cristian, 2022

CAPITULO V

5 ANÁLISIS DE RESULTADOS

El capítulo 5 analiza parámetros como la latencia y el nivel de seguridad del prototipo, con el fin de medir la eficiencia de la implementación

5.1 Latencia del prototipo

Para este apartado se toman en cuenta diferentes aspectos, dado que el prototipo utiliza algunos sistemas y servicios, se considera la latencia que causan por separado

5.1.1 *Latencia del dominio del servidor*

Al ser un sistema IoT la infraestructura del servicio en la nube juega un papel muy importante ya que los datos se procesan en sus servidores, en este marco se utiliza la herramienta PRTG Network Monitor que es un software de monitoreo de red; para analizar la latencia del prototipo se considera el tiempo de respuesta del servidor web por lo que se establece el monitoreo del dominio en general, en este caso el que se muestra en la ilustración 1-5, se establece el tiempo de fallo de 60 segundos, es decir que si no obtiene un respuesta en ese tiempo se considera el servidor en inactividad; otro de los parámetros a considerar es la URL a monitorear, en este caso se utiliza a "https://espochemmpa.com/" que es el dominio general del servidor, adicional como método de petición se usa GET que permite realizar peticiones hacia el servidor.

Ilustración 1-5: URL y método para analizar la latencia del hosting

Realizado por: Sáez, Cristian, 2022

Luego de haber definido los parámetros se tienen los siguientes resultados para la URL especificada, se toma en cuenta esos lapsos de monitoreo porque es donde más se pueden notar los valores, teniendo como resultado de disponibilidad del servidor en un 96,127% y un fallo de 3,873 % que es menor al 99,90 % que ofrece como característica el servidor.

Como resultado de tiempo de respuesta se obtiene un máximo de 6,365 mseg como pico, esto debido a que en este momento se conectaron los dispositivos que requieren del hosting y se pone

en marcha todo el prototipo para realizar pruebas, en el caso de tiempo de respuesta mínimo se obtiene 655 mseg que es valor considerablemente bueno, ya que como se mencionó se está monitoreando todo el dominio, tomando en cuenta todos los valores se tiene como promedio 1.252 mseg como promedio, que en pruebas realizadas con escenarios lo más parecido a los reales no es perceptible para el usuario, estos datos se pueden observar en la ilustración 2-5

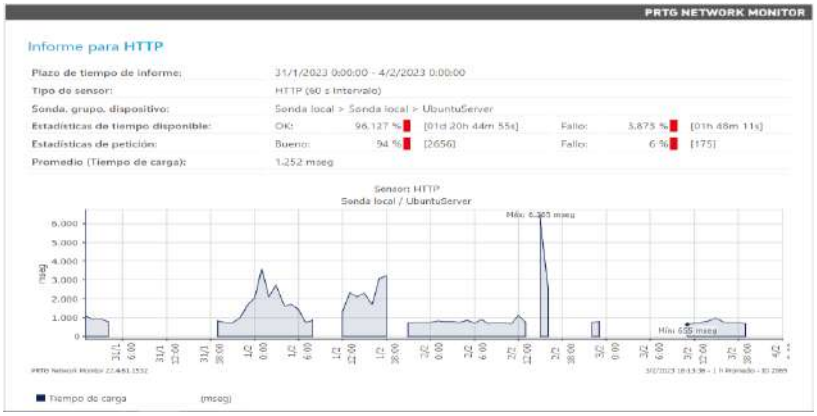


Ilustración 2-5: PRTG tiempo de respuesta del hosting

Realizado por: Sáez, Cristian, 2022

5.1.2 *Latencia de la API de Google vision para el reconocimiento óptico de caracteres*

Otro de los parámetros a considerar del prototipo es la latencia que se genera al llamar a la API de Google vision que es la encargada de realizar el reconocimiento óptico de caracteres para detectar la placa vehicular del vehículo que desea ingresar, según la ilustración 3-5 se obtiene una latencia máxima de 1683 mseg y una mínima de 0,12 mseg, dando una latencia media de 259 mseg, que es muy buena ya la API procesa imágenes y devuelve el texto en formato JSON

Ilustración 3-5: Mediana de latencia API de Google vision

Realizado por: Sáez, Cristian, 2022

Otro de los aspectos a considerar es la latencia generada por el procesamiento que realiza el computador, el cual considera los aspectos analizados en el capítulo 3, para analizar este valor se implementa en el código el módulo datetime para determinar el tiempo que se demora en procesar la imagen y extraer el valor de la placa vehicular, en la ilustración 4-5 se observa 2.053 segundos como el lapso que tomó el procesamiento de la imagen

```
---------- Start recognize license palate --------
['TBJ']
[9403]
PLACA: TBJ9403
BASE DE DATOS: TBJ9403
LAS PLACAS COINCIDEN
Total_time : 0:00:02.053525
---------- End ----------
```

Ilustración 4-5: Reconocimiento de placas en Arduino

Realizado por: Sáez, Cristian, 2022

En la ilustración 5-5 se pude observar la imagen captada por la cámara y el valor de la placa extraída, se puede decir que la cámara considerada para el prototipo cumple con los requisitos para obtener la imagen de manera nítida y estable para que el script diseñado en Python pueda procesar la imagen y devolver la placa del vehículo que permitirá o denegará el acceso al establecimiento

Ilustración 5-5: Placa del vehículo

Realizado por: Sáez, Cristian, 2022

5.1.3 *Latencia generada por el protocolo MQTT*

Por último, se toma en cuenta la latencia generada por el protocolo MQTT que se utiliza para la comunicación entre los ESP y el programa que procesa la imagen tanto como publicación y suscripción, para encontrar este valor se observa en la ilustración 6-5 la hora en la que se tomó el dato en el ESP y el que llega al servidor donde se encuentra el bróker. La hora que se visualiza en formatos que conocemos se traduce a 03:11:46:928 horas y la hora de llegada es 03:11:47:266 que al realizar la diferencia da como resultado 388 mseg de retardo

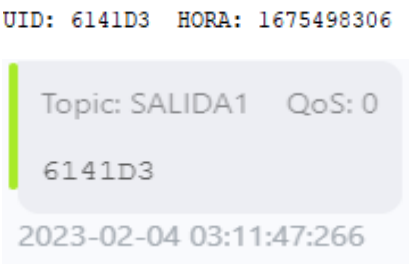


Ilustración 6-5: Latencia protocolo MQTT

Realizado por: Sáez, Cristian, 2022

5.1.4 *Latencia total al ingreso y salida del establecimiento*

Para la latencia de todo el prototipo se realiza una adición de todos los valores encontrados con anterioridad los cuales se observan en la tabla 1-5, teniendo como resultado un retardo de 3693 mseg en el ingreso y 1640 mseg para la salida del establecimiento.

Tabla 1-5: Latencia Total al ingreso y salida del establecimiento

Parámetro	Latencia Ingreso	Latencia Salida
Tiempo de respuesta del servidor	1 252 mseg	1 252 mseg
Procesamiento de imagen para el reconocimiento de la placa vehicular	2 053 mseg	----------
Protocolo MQTT	388 mseg	388 mseg
Total	3 693 mseg	1 640 mseg

Realizado por: Sáez, Cristian, 2022

5.2 Parámetros de seguridad implementados

Otro de los aspectos a analizar es la seguridad que se implementó en el prototipo, esta fue implementada en las diferentes capas dentro del modelo OSI, así se tiene

5.2.1 *Seguridad en la capa de transporte*

El protocolo SSL (Secure Socket Layer) se establece entre la capa de aplicación y la de trasporte, esta se considera en el prototipo que ocupa el protocolo HTTP que por sí solo deja las brechas de seguridad abiertas en un gran porcentaje ya que la misma puede ser intervenida, al unir estos 2 protocolos da lugar a HTTPS, consiguiendo que la transferencia de datos entre el sitio web y el usuarios se lo realice de manera segura en ambos sentidos, esta se considera esencial ya que se manejan datos personales y confidenciales, como información del usuario y sus tarjetas de crédito para realizar el pago.(Redalia 2023)

En la ilustración 7-5 se observa que está SSL con HTTP está implementado a nivel de conexión el protocolo TLS 1.3 que es propio del navegador, el intercambio de clave X25519 que se basa en Diffie-Hellman de curva elíptica que permite intercambiar de manera segura claves criptográficas mediante un canal público, la firma del servidor RSA-PSS with SHA-256 y cifrado AES_128_GCM que en detalle es de 128 bits con cifrado por bloque simétrico.
En el caso del certificado se observa el dominio propietario espochemmpa.com en el cual se aplica el protocolo, la fecha de validación, se observa que todavía sigue vigente.

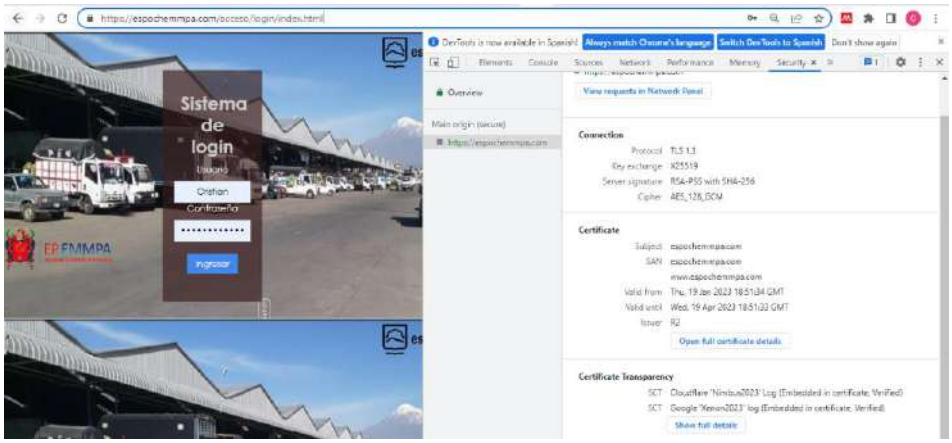
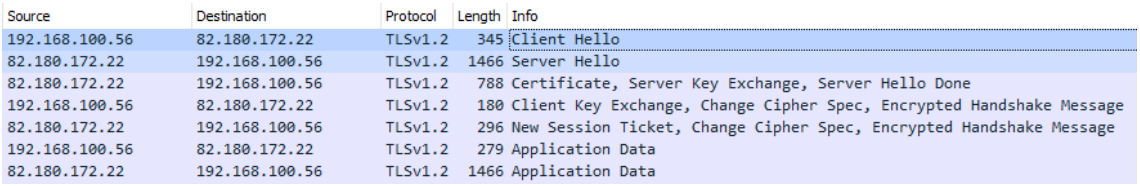


Ilustración 7-5: Certificado de seguridad del dominio

Realizado por: Sáez, Cristian, 2022

Para verificar que existe seguridad en la capa de transporte se realizó un análisis del tráfico generado en la red mediante el software wireshark, en la gráfica 8.5 se observa el proceso de handshake del protocolo TLS para esto se toma en cuenta la dirección de origen en este caso del ordenador del usuario 192.168.100.56, la dirección de destino 82.100.172.22 perteneciente al servidor, el protocolo TLS v1.2 la longitud de datos y el detalle del proceso. En la gráfica se notó que el cliente envía un mensaje “hello” y el servidor le responde con uno igual y datos como el certificado, luego el cliente verificará estos datos y devolverá una cadena aleatoria de bytes y la

cifra, cuando el servidor ha recibido esta cadena, las 2 partes generan una clave junto con las de sesión las cuales se usan para el cifrado simétrico de la información



Source	Destination	Protocol	Length	Info
192.168.100.56	82.180.172.22	TLSv1.2	345	Client Hello
82.180.172.22	192.168.100.56	TLSv1.2	1466	Server Hello
82.180.172.22	192.168.100.56	TLSv1.2	788	Certificate, Server Key Exchange, Server Hello Done
192.168.100.56	82.180.172.22	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
82.180.172.22	192.168.100.56	TLSv1.2	296	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
192.168.100.56	82.180.172.22	TLSv1.2	279	Application Data
82.180.172.22	192.168.100.56	TLSv1.2	1466	Application Data

Ilustración 8-5: Certificado de seguridad del dominio. Proceso de handshake

Realizado por: Sáez, Cristian, 2022

Para analizar de mejor manera el proceso de handshake se extrajeron los resultados de cada fase, teniendo como primer paso el envío del mensaje hola del cliente, que se compone de la versión de TLS que admite el cliente, la hora, una cadena randómica de bytes de 28, el ID de sesión, el conjunto de cifrados que soporta el cliente y la URL del servidor, detalles que se observan en la Tabla 2.5

Tabla 2-5: Parámetros Hello del cliente

Parámetro	Valor
Version de TLS soportada	Handshake Type: Client Hello (1) Length: 282 Version: TLS 1.2 (0x0303)
UTC time	08:47:37.000000000 Hora est. Pacífico, Sudamérica
Random number	934e3094099107179aefba4d5ad31977beb570143738523a5f36fdb0
ID de sesión	Session ID Length: 0
Conjunto de cifrado	Cipher Suites Length: 132 Cipher Suites (66 suites) Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c) Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
URL del servido	Server Name Indication extension Server Name list length: 18 Server Name Type: host_name (0) Server Name length: 15 Server Name: espochemmpa.com

Realizado por: Sáez, Cristian, 2022

El servidor que recibió el mensaje hola del cliente, respondió con la versión de TLS que concuerdan tanto el servidor como cliente en este caso TLS 1.2, el ID de sesión, el tiempo UTC otra cadena aleatoria de 28 bytes y el conjunto de cifrado que seleccionó que dice que utiliza el intercambio de llaves de curva elíptica Diffie-Hellman Efímera (ECDHE), autenticación mediante el algoritmo RSA, (AES 128 GCM) como estándar de cifrado, y SHA 256 para el código de autenticación de mensajes con los parámetros de la tabla 3-5

Tabla 3-5: Parámetros Hello del servidor

Parámetro	Valor
TLS admitida por el cliente y servidor	Handshake Protocol: Server Hello Handshake Type: Server Hello (2) Length: 55 Version: TLS 1.2 (0x0303)
ID de la sesión	Session ID Length: 0
Tiempo UTC	Feb 9, 2023 09:54:29.000000000 Hora est. Pacífico, Sudamérica
Número aleatorio	63e50925c4de55c4b5df8ff4f1bf46aa7f83d77dc85d795c444f574e47524401
Conjunto cifrado seleccionado	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)

Realizado por: Sáez, Cristian, 2022

A la par del envío del mensaje "hello" del servidor, también envía los certificados que posee la página, se observaron el del dominio, el de la organización, los cuales tienen llaves públicas para intercambiar claves, mediante Diffie-Hellman que se visualiza en la clave de cifrado, culminando con un saludo hacia el cliente. Los parámetros en detalle se visualizan en la tabla 4-5

Tabla 4-5: Envío de parámetros del servidor

Parámetro	Valor
Certificado	Certificates (4291 bytes) Certificate Length: 1596 Certificate: 308206383082052… Certificate Length: 1306 Certificate: 3082051630820… Certificate Length: 1380 Certificate: 308205603082044…
Clave de cifrado del servidor	EC Diffie-Hellman Server Params Curve Type: named_curve (0x03) Named Curve: secp256r1 (0x0017) Pubkey Length: 65 Pubkey: 04a9757fe16e36b096c700b8e7b150218d74f5d4ce733eea14c4fd910c3d8ec69479d7f8… Signature Algorithm: rsa_pkcs1_sha256 (0x0401) Signature Hash Algorithm Hash: SHA256 (4) Signature Hash Algorithm Signature: RSA (1) Signature Length: 512 Signature: 776aa9ec1264aa89e8001dcd4f7d30deaf79c31312b593d0a5039a707e47421577ca272b…
Saludo del servidor hecho al cliente	TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done Content Type: Handshake (22) Version: TLS 1.2 (0x0303) Length: 4 Handshake Protocol: Server Hello Done Handshake Type: Server Hello Done (14) Length: 0

Realizado por: Sáez, Cristian, 2022

Para seguir con el hanshake, el cliente respondió con su clave pública y cambio de cifrado, que da a conocer que se puede iniciar la comunicación encriptada ya que tiene la información necesaria para hacerlo, quedando a la espera de datos cifrados. En la tabla 5-5 se detallan todos los parámetros que intervienen en este proceso

Tabla 5-5: Envío de parámetros del cliente

Parámetro	Valor
Llave de encriptación del cliente	Handshake Protocol: Client Key Exchange Handshake Type: Client Key Exchange (16) Length: 66 EC Diffie-Hellman Client Params Pubkey Length: 65 Pubkey: 042a18da1e45194c992b872cb67c69db88a68eae10aad9f95526a9b5dc70aa2b74899ac6…
Especificaciones de cambio de cifrado	TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec Content Type: Change Cipher Spec (20) Version: TLS 1.2 (0x0303) Length: 1 Change Cipher Spec Message
Finalización	TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message Content Type: Handshake (22) Version: TLS 1.2 (0x0303) Length: 40 Handshake Protocol: Encrypted Handshake Message

Realizado por: Sáez, Cristian, 2022

Finalmente, el servidor emite un mensaje de cambio de cifrado y que ha finalizado el handshake de manera exitosa, esto se observa en la tabla 6-5

Tabla 6-5: Culminación de handshake del servidor

Parámetro	Valor
Especificaciones de cambio de cifrado	TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec Content Type: Change Cipher Spec (20) Version: TLS 1.2 (0x0303) Length: 1 Change Cipher Spec Message
Finalización	TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message Content Type: Handshake (22) Version: TLS 1.2 (0x0303) Length: 40 Handshake Protocol: Encrypted Handshake Message

Realizado por: Sáez, Cristian, 2022

Culminado estas fases el trasporte de paquetes se puede considerar seguro y esto se puede visualizar en la ilustración 9-5, el mensaje resaltado es ilegible y tiene la etiqueta de datos encriptados, igualmente que el protocolo utilizado para la transferencia de datos es Hypertext Transfer Protocol (HTTP) pero ya de manera segura HTTPS, evitando así que otras personas ajenas al establecimiento puedan ver la información que se está manejando.

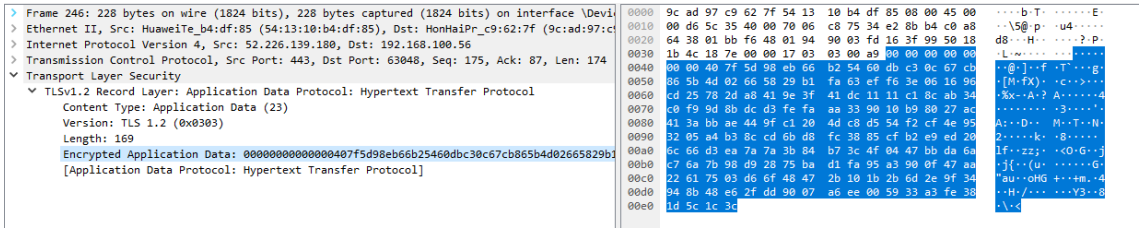


Ilustración 9-5: Análisis de tráfico

Realizado por: Sáez, Cristian, 2022

5.2.2 *Pruebas de autenticación de administrador*

El manejo de datos por parte del administrador debe ser restringido solo a dicha persona, por lo que, en el caso del ingreso incorrecto de las credenciales, el sistema no le permitirá ingresar a la interfaz principal, protegiendo así la propiedad de autenticación de la seguridad, cuando esto suceda se visualiza en la ilustración 10-5 el mensaje ERROR DE LA AUTENTICACION

Ilustración 10-5: Pruebas de autenticación de administrador

Realizado por: Sáez, Cristian, 2022

5.2.3 *Pruebas de seguridad en la Aplicación móvil*

5.2.3.1 *Autenticación del usuario para ingresar a la aplicación móvil*

Para poder acceder a servicio del estacionamiento el usuario debe ingresar a su app, ya que se manejan datos y funciones confidenciales y delicadas como es el uso de tarjetas de crédito, esta se controla por un sistema de autenticación al iniciar la app, en el caso del incorrecto ingreso de datos, esta no le permitirá el acceso a la misma, limpiando los espacios para ingresar nuevamente los campos y generando un mensaje de Credenciales Incorrectas, interfaz que se observa a continuación

Ilustración 11-5: Autenticación de usuario para aplicación móvil

Realizado por: Sáez, Cristian, 2022

5.2.3.2 *Verificación del pin de pago*

En el caso de fallar por 3 ocasiones el ingreso del PIN de seguridad para realizar el pago se envía un correo electrónico a la persona designada como administrador para que pueda tomar correctivos ante la situación, en la ilustración 10-5 se puede apreciar el UID, Nombre, Apellido la Placa del vehículo y el número de tarjeta que ha cometido el error al momento de cancelar por el servicio del estacionamiento

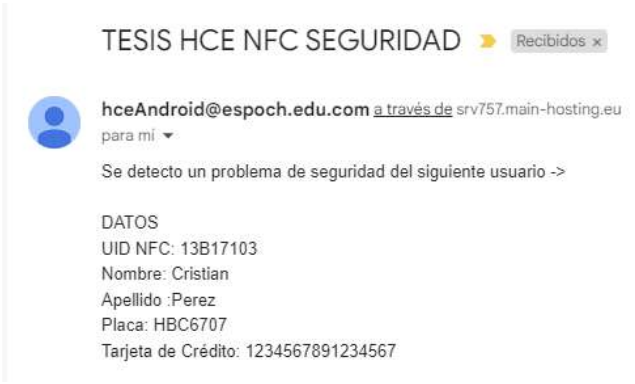


Ilustración 12-5: Correo electrónico enviado en caso de falla de PIN de seguridad

Realizado por: Sáez, Cristian, 2022

5.2.4 *Seguridad física al ingreso y salida del establecimiento*

Al ser un control de acceso, se busca la manera de restringir el libre tránsito al establecimiento, por lo que se opta por una barrera vehicular, la cual controla el ingreso y salida de los usuarios, el mismo que concederá el paso a aquellos usuarios que cumplan con las reglas establecidas

Ilustración 13-5: Seguridad física al ingreso y salida del establecimiento

Realizado por: Sáez, Cristian, 2022

5.2.5 *Seguridad en el dispositivo lector NFC*

La tecnología NFC permite una seguridad intrínseca ya que es el usuario el interactúa directamente con el sensor detector de la tecnología, esto a muy poca distancia, adicional a este aspecto, el sistema mediante el lector realiza una consulta del UID que está queriendo ingresar, en el caso de no estar registrado, no se permitirá el acceso.

Ilustración 14-5: Seguridad en el lector NFC

Realizado por: Sáez, Cristian, 2022

5.3 Análisis de costos

En la tabla 7-5 se consideran todos los gastos incurridos tanto en la parte de software como hardware para tener una idea de cuál sería el valor del prototipo

Tabla 7-5: Análisis de costos

CANTIDAD	DESCRIPCIÓN	COSTO UNITARIO	COSTO TOTAL
12 meses	Servicio de hosting alojamiento compartido premium	$ 35,88	$ 35,88
4	Node MCU ESP 8266	$ 9,00	$ 36,00
2	Módulo NFC P532	$ 15,00	$ 30,00
4	Cajas y cables para prototipos	$ 6,00	$ 24,00
2	Barreras vehiculares	$ 850,00	$ 1700,00
1	Cámara Hikvision Tubo Vari focal 1080p Modelo DS-2CE665D0T-VFIR3F	$ 65,00	$ 65,00
1	Dvr Hikvision 04 Cámaras 1080p Modelo DS-7104HGHI-M1/S	$ 75,00	$ 75,00
1	Pareja de Balun HD	$ 8,00	$ 8,00
1	Fuente de 12v para cámara	$ 7,90	$ 7,90
20 m	Cable UTP para conexión de cámaras	$ 0,50	$ 10,00
1	Router Tplink WR841HP N300mbps High Power 2ant. 9dbi 4 Puertos LAN	$ 49,00	$ 49,00
1	Ordenador de Escritorio	$ 850,00	$ 850,00
		TOTAL	$ 2 890, 78

Realizado por: Sáez, Cristian, 2022

Conforme al análisis de costos realizado se puede notar que los valores más elevados corresponden a las barreras vehiculares y al ordenador, cada uno con un valor de $ 850,00, pero si se hace compara con tener a 2 operadores durante 1 año el ahorro para la empresa va a ser considerable, los demás dispositivos son de bajo medio y bajo que fácilmente ayudarían a una escalabilidad futura. Se destaca que no se consideran valores par operadores ya que al considerar barreras vehiculares estos reemplazaran al personal, generando mayores réditos para la empresa.

5.4 Comparación entre el sistema implementado actualmente y el prototipo

El funcionamiento de los sistemas tiene un fin similar, sin embargo, se destacan diferencia marcadas que se visualizan en la tabla 8-5, tales como disponibilidad, experiencia de usuario, tiempo de espera, forma de pago por el servicio

Tabla 8-5: Comparación entre el sistema implementado actualmente y el prototipo

Sistema Actual	**Prototipo**
Disponibilidad de información dentro del establecimiento	Disponibilidad de información en todo tiempo y lugar mientras se tenga internet
Control de acceso con operadores en ingreso y salida del establecimiento	Control de acceso automatizado mediante verificación de datos y barreras vehiculares
Uso de tickets de papel para registrar el ingreso y calcular el valor a cancelar en la salida	Uso de aplicación móvil para registro de entrada y cálculo de valor a cancelar en la salida
Pago del servicio de estacionamiento con dinero físico	Débito de recargas o de tarjetas de crédito registradas por el usuario
El usuario solo conoce el valor que cancela al salir del establecimiento	Usuario informado mediante la app móvil de sus parqueos completados, información personal, saldo disponible
Tiempo de espera: Ingreso 1 minuto sin tráfico, 15 minutos con tráfico Salida 2 minuto sin tráfico, 18 minutos con tráfico	Tiempo de espera Ingreso 8 segundos sin tráfico, 2 minutos con tráfico Salida 12 segundos sin tráfico, 3 minutos con tráfico

Realizado por: Sáez, Cristian, 2022

Se destaca que el sistema actual es uno más tradicional, ya que considera elementos físicos como tickets, lectores, operadores para su funcionamiento, en cambio el prototipo presentado es un sistema tecnificado en el cual la infraestructura IoT controla el acceso y mantiene un registro por cada usuario que hacen uso del servicio de estacionamiento en la EPEMMPA, obteniendo un mayor control y eficiencia del servicio en cuanto a información, recaudación y movilidad. Se estima una reducción de alrededor del 80% en tiempos de espera con y sin tráfico vehicular, ya que se reducen proceso como entrega de tickets, cobros manuales, recategorizaciones que realiza el operador.

CONCLUSIONES

El prototipo ayuda a mantener un control eficiente de la recaudación y el manejo de usuarios que hacen uso del servicio de estacionamiento de la EP-EMMPA, ya que la información se encuentra disponible en internet y puede ser analizada desde cualquier ubicación y en todo tiempo.

La inversión de $ 2 890, 78 no es elevada, tomando en cuenta la gran afluencia de usuarios y que no se incurrirá en gastos posteriores a la implementación como pago de operarios y compra de papelería para la impresión de tickets

Los sistemas de autenticación en la capa de aplicación del administrador y usuario son importantes, ya que el manejo de datos se maneja en base a privilegios, permitiendo o denegando el uso de sus datos ya sea de control en caso de administración, registro o pago para los usuarios.

Los 3793 mseg que le toma al prototipo en registrar el ingreso del usuario son eficientes por todo el proceso que realiza para validar la información del cliente, así mismo los 1640 mseg al salir del establecimiento no generan problemas de atascamiento.

El cifrado asimétrico que implementa TLS con AES de 128 bits, RSA y SHA 256 garantizan confidencialidad, autenticidad e integridad en la comunicación del medio ya que un atacante no podrá leer la información ni conocerá cuales son las llaves públicas o privadas.

El usuario tiene una experiencia confortable e eficiente ya que todos sus movimientos e información puede ser visualizada en su dispositivo móvil, adicional es el quien permite el pago mediante el PIN de seguridad, generando confianza en el entorno.

La reducción aproximada del 80% en los tiempos de espera tanto a la entrada como salida, son una solución a los problemas de tráfico que se generan con el sistema actual, ya que con el sistema tradicional se demora de 2 hasta 15 minutos en cada proceso reduciendo estos tiempos de 8 segundos hasta 2 minutos.

La emulación de tarjetas mediante la aplicación móvil y la tecnología NFC es la que mejor se acomoda al prototipo ya que se mantiene un UID fijo para cada usuario con el fin de identificarlos y autenticarlos para evitar fraudes.

RECOMENDACIONES

Para los usuarios que no posean dispositivos inteligentes con tecnología NFC, la compra de llaveros o tarjetas se lo realice de forma masiva para abaratar costos.

Contratar servicios de hosting con mayor capacidad de procesamiento y almacenamiento, por la alta demanda de usuarios.

Tener conectividad inalámbrica estable y rápida en las entradas y salidas para menorar tiempos de latencia en la comunicación.

Considerar barreras vehiculares robustas e industriales que soporten la alta demanda que va a tener que soportar el sistema.

BIBLIOGRAFÍA

A2 HOSTING. *The Best Web Hosting Services at 20x Speeds.* [en línea]. [Consulta: 21 enero 2023]. Disponible en: https://www.a2hosting.com/?aid=582adb16d72a0&bid=11a44674&data1=w59Z20wNea.

ADAFRUIT. *Adafruit PN532 NFC/RFID Controller Shield para Arduino + Extras : ID 789 : $39.95 : Adafruit Industries, Electrónica y kits de bricolaje únicos y divertidos*. [en línea]. [Consulta: 19 enero 2023]. Disponible en: https://www.adafruit.com/product/789#technical-details.

AG ELECTRÓNICA S.A. *Guía rápida: Escudo NFC/RFID para arduino (PN532) Número de parte: ADA-789. 1library* [en línea]. [Consulta: 21 abril 2022]. Disponible en: https://1library.co/document/yj8vd02q-guia-rapida-escudo-nfc-rfid-arduino-numero-parte.html.

ALROWAILY, M. & LU, Z. "Secure edge computing in IoT systems: Review and case studies." *Proceedings - 2018 3rd ACM/IEEE Symposium on Edge Computing, SEC 2018*, (2018) pp. 440-444. DOI 10.1109/SEC.2018.00060.

ANZAR, A. *La red Internet. El modelo TCP/IP*. España: Grupo Abantos Formación y Consultoría, 2004. ISBN 8495679493.

ARON SEMLE, K. "Protocolos IoT para considerar". *Aadeca Revista*, (2016), pp. 34.

BERMEJO VERA, J. & GUEDEA MARTIN, M. *Internet de las cosas.* 2a. ed. 2. Madrid: Editorial Reus, 2020. ISBN 9788429022001.

BLIZNAKOFF DEL VALLE, D. Iot: tecnologías, usos, tendencias y desarrollo futuro. 2014. S.l.: s.n.

BOADA, M., LAZARO, A., VILLARINO, R., GIL-DOLCET, E. y GIRBAU, D., 2021. Battery-Less NFC Bicycle Tire Pressure Sensor Based on a Force-Sensing Resistor. *IEEE Access*, vol. 9, pp. 103975-103987. ISSN 21693536. DOI 10.1109/ACCESS.2021.3099946.

BRICIO, A. y CHISAG, A. "Prototipo de un sistema de control de pago de pasajeros enel transporte urbano de la ciudad de guayaquil utilizando tecnología nfc" [en línea], 2019.

Guayaquil: UNIVERSIDAD DE GUAYAQUIL. [Consulta: 6 febrero 2023]. Disponible en: https://docplayer.es/159952503-Universidad-de-guayaquil-facultad-de-ciencias-matematicas-y-fisicas-carrera-de-ingenieria-en-networking-y-telecomunicaciones-proyecto-de-titulacion.html.

BUCHMEIER, G.G., TAKACS, A., DRAGOMIRESCU, D., RAMOS, J.A. y MONTILLA, A.F. "Optimized NFC Circuit and Coil Design for Wireless Power Transfer with 2D Free-Positioning and Low Load Sensibility". *Sensors 2021, Vol. 21, Page 8074*, vol. 21, no. 23, pp. 8074 (2021). ISSN 1424-8220. DOI 10.3390/S21238074.

CALDERÓN, L. Seguridad informática y seguridad de la información. 2015. . S.l.:

CARRASCO, E. Metodología para selección de tecnologías lpwan para diversas aplicaciones de internet de las cosas (Trabajo de titulación). Universidad de Chile, Santiago, Chile. 2020.

CEPRA. "Servicio de Distribución de Datos (Data Distribution Service), DDS | PROYECTO CEPRA". [en línea], 2022. [Consulta: 8 diciembre 2022]. Disponible en: https://www.utpl.edu.ec/proyectomiddleware/?q=tutorial-dds.

CORNEJO, B., GODOY, C. & ROCA, E. Constrained Application Protocol. 2018. S.l.:

CORREDOR CAMARGO, Ó.F., PEDRAZA MARTÍNEZ, L.F. & HERNÁNDEZ, C.A. "Bluetooth Technology: Alternative to Cellular Networks of Voice and Data". *Revista Visión Electrónica*, vol. 1, no. 1,(2009), pp. 73-84.

COSKUN, V., OK, K. & OZDENIZCI, B. *Near Field Communication (NFC)* [electronic resource]*: From Theory to Practice*. WILEY. 2011. S.l.: Wiley. ISBN 9781119965787.

COSKUN, Vedat. *Professional NFC application development for Android*. 2013, S.l.: Wiley. ISBN 9781118380543.

CRESPO, E. *arquitectura API | Aprendiendo Arduino*. [en línea]. [Consulta: 31 mayo 2022]. Disponible en: https://aprendiendoarduino.wordpress.com/tag/arquitectura-api/.

DAMAŠEVIČIUS, R., POONGODI, M., TAYYAB RAUF, H., ALI KHATTAK, H., AHMED JAN, S., UL AMIN, N., SHUJA, J., ABBAS, A., MARAY, M. & ALI, M., "SELWAK: A Secure and Efficient Lightweight and Anonymous Authentication and Key

Establishment Scheme for IoT Based Vehicular Ad hoc Networks". *Sensors 2022, Vol. 22, Page 4019*, vol. 22, no. 11, (2022), pp. 4019. ISSN 1424-8220. DOI 10.3390/S22114019.

DEVELOPERS. *Host-based Card Emulation | Android Developers.* [en línea]. [Consulta: 5 junio 2022]. Disponible en: https://android-doc.github.io/guide/topics/connectivity/nfc/hce.html.

DEVELOPERS, A. *Meet Android Studio | Android Developers.* [en línea]. [Consulta: 25 julio 2022]. Disponible en: https://developer.android.com/studio/intro.

DEYIMAR, A. *¿Qué Es PHP? Una Introducción Para Principiantes*. [en línea]. [Consulta: 28 julio 2022]. Disponible en: https://www.hostinger.es/tutoriales/que-es-php.

DIGICERT, I. *¿Qué es SSL, TLS y HTTPS? | DigiCert*. [en línea]. [Consulta: 6 febrero 2023]. Disponible en: https://www.websecurity.digicert.com/es/es/security-topics/what-is-ssl-tls-https.

FEDERACIÓN DE ENSEÑANZA DE CC. OO. "La conectividad Inalámbrica: un enfoque para el alumno". *Temas para la Eduación*, vol. 1, (2010), (Colombia) pp. 1-8.

FULLER, J. *Cómo diseñar una infraestructura preparada para IoT: la arquitectura de 4 etapas.* [en línea]. [Consulta: 31 mayo 2022]. Disponible en: https://techbeacon.com/enterprise-it/4-stages-iot-architecture.

GETAPP. *Hostinger: precios, funciones y opiniones | GetApp España 2022*. [en línea]. [Consulta: 25 julio 2022]. Disponible en: https://www.getapp.es/software/2052421/hostinger.

GONZÁLES, A. Diseño de una Red Inalámbrica de Banda Ancha. Catalunya: (Trabajo de titulación) Universidad Oberta de Catalunya, Catalunya, España. 2018.

GONZÁLEZ, L., SOFÍA, O., LAGUÍA, D., GESTO, E. & HALLAR, K. "Internet del Futuro – Estudio de tecnologías IoT". *Informes Científicos Técnicos - UNPA*, vol. 12, no. 3, pp. 105-137, (2020). ISSN 1852-4516. DOI 10.22305/ict-unpa.v12.n3.744.

GOOGLE DEVELOPERS. *Descripción general de la emulación de tarjetas basada en el host | Desarrolladores de Android | Android Developers*. [en línea]. [Consulta: 5 junio 2022]. Disponible en: https://developer.android.com/guide/topics/connectivity/nfc/hce?hl=es-419#HceSecurity.

GOPALAKRISHNAN, S., WAIMIN, J., ZAREEI, A., SEDAGHAT, S., RAGHUNATHAN, N., SHAKOURI, A. & RAHIMI, R. A biodegradable chipless sensor for wireless subsoil health monitoring. *Scientific Reports 2022 12:1*, vol. 12, no. 1, pp. 1-14. (2022), ISSN 2045-2322. DOI 10.1038/s41598-022-12162-z.

HOLDOWSKY, J.,. "Inside the Internet of Things (IoT) A primer on the technologies building the IoT". *Deloitte*, (2015), (Estados Unidos).

HOSTGATOR.COM, *Hostgator*. [en línea]. 2023. [Consulta: 21 enero 2023]. Disponible en: https://www.hostgator.com/promo/top10best?clickid=0UFzxDyKXxyNWoOzlTXDUSa%3AUkAwTuwHF3mQ3I0&irgwc=1&affpat=1&mpid=34020.

HOSTINGER, *Comprar Y Registrar Dominios Web* . [en línea]. 2022a. [Consulta: 13 diciembre 2022]. Disponible en: https://www.hostinger.es/comprar-dominio.

HOSTINGER, *¿Qué es un dominio web?*. [en línea]. 2022b. [Consulta: 13 diciembre 2022]. Disponible en: https://www.hostinger.es/tutoriales/que-es-un-dominio-web.

HOSTINGER, *Web Hosting*. [en línea]. 2023. [Consulta: 21 enero 2023]. Disponible en: https://www.hostinger.com/web-hosting?utm_medium=affiliate&utm_source=aff1015&utm_campaign=14&session=102369084840d1a9c3f9b19368c3be.

HOSTINGER TUTORIALES, ¿Qué Es MySQL? . [en línea]. 2022. [Consulta: 14 diciembre 2022]. Disponible en: https://www.hostinger.es/tutoriales/que-es-mysql.

IBM CLOUD EDUCATION, *¿Qué es SOA (arquitectura orientada a servicios)?* - [en línea]. España | IBM. *IBM*, 2019. [Consulta: 4 mayo 2022]. Disponible en: https://www.ibm.com/es-es/cloud/learn/soa.

INCHAURZA, G. *COMPARATIVA TEÓRICA Y PRÁCTICA DE MIDDLEWARES MQTT*. 2018. S.l.: s.n.

IONOS, *Fog computing: nuevo paradigma para las nubes del IoT*. [en línea]. 2019. [Consulta: 31 mayo 2022]. Disponible en: https://www.ionos.es/digitalguide/servidores/know-how/fog-computing/.

ITU, U.I. de T., 2012. *Descripción General de Internet de los Objetos* Y.2060- Y.4000. *Y.2060 Y.4000*, pp. 20.

KEPWARE, S. "Protocolos IIoT para considerar." *AADECA REVISTA*, vol. Segunda, (2016).pp. 1-4.

KRYPTON SOLID, *CoAP es el protocolo «moderno» de IoT*. [en línea]. 2022. [Consulta: 7 diciembre 2022]. Disponible en: https://kryptonsolid.com/coap-es-el-protocolo-moderno-de-iot/.

LESAS, Anne & MIRANDA, S. *The Art and Science of NFC Programming.* 1. New York: ISTE Ltda. 2017.ISBN 978-1-78630-057-7.

LESAS, Anne-Marie. & MIRANDA, Serge.. *The Art and Science of NFC Programming.* 1. S.l.: John Wiley & Sons, Incorporate 2017.d. ISBN 9781119379058.

LLAMAS, L.,*Modelos y características de Raspberry Pi*. [en línea] 2017. [Consulta: 23 julio 2022]. Disponible en: https://www.luisllamas.es/modelos-de-raspberry-pi/.

LOZANO, V. & ALABERTO, J. "IoT: La Evolución de la Seguridad en el Internet de las Cosas".

MADAKAM, S., RAMASWAMY, R. & TRIPATHI, S. "Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications*, vol. 03, no. 05, pp. 164-173. ISSN 2327-5219**,** 2015 DOI 10.4236/jcc.2015.35021.

MCKENNA, N. *What is Cloud Computing and How Does It Work? McKenna Consultants* [en línea]. [Consulta: 31 mayo 2022]. Disponible en: https://www.mckennaconsultants.com/how-cloud-computing-works/.

MECHATRONICS NAYLAMP. *Módulo Lector RFID 13.56MHz RC522.* [en línea]. [Consulta: 13 enero 2023]. Disponible en: https://naylampmechatronics.com/rfid-nfc/80-modulo-lector-rfid-1356mhz-rc522.html.

MICROSOFT. *Arquitectura NFC - Windows drivers | Microsoft Docs. Microsoft* [en línea]. [Consulta: 21 abril 2022]. Disponible en: https://docs.microsoft.com/es-es/windows-hardware/drivers/nfc/nfc-architecture.

MICROSOFT. *Protocolos y tecnologías de IoT | Microsoft Azure.* [en línea]. [Consulta: 1 mayo 2022]. Disponible en: https://azure.microsoft.com/es-es/overview/internet-of-things-iot/iot-technology-protocols/?fbclid=IwAR3q3bNMWLbwTZNwi8rNVvbH6-JMi7LUXQNP6FUBwUP00ZpmLc7ejJMnspQ.

MONTENEGRO, G., SCHUMACHER, C. y KUSHALNAGAR, N. "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals". En: Num, (2007) Pages: 12. S.l.:

MORENO, J. y RUIZ, D., (2007). Informe Técnico: *Protocolo ZigBee (IEEE 802.15.4)*. . S.l.:

MORGE-ROLLET, L., LE ROY, F., LE JEUNE, D., CANAFF, C. y GAUTIER, R. "RF eigenfingerprints, an Efficient RF Fingerprinting Method in IoT Context". *Sensors 2022, Vol. 22, Page 4291*, vol. 22, no. 11, (2022) pp. 4291. ISSN 1424-8220. DOI 10.3390/S22114291.

MOUHA, R.A. "Internet of Things (IoT)". *Journal of Data Analysis and Information Processing*, vol. 9, (2021) pp. 77-101. DOI 10.4236/jdaip.2021.92006.

NAYLAMP MECATRONICS SAC. *Módulo Lector RFID/NFC 13.56MHz PN532.* [en línea]. [Consulta: 25 julio 2022]. Disponible en: https://naylampmechatronics.com/rfid-nfc/182-modulo-lector-rfid-nfc-1356mhz-pn532.html.

NAYLAMP MECHATRONICS. *NodeMCU v2 ESP8266 WiFi.* [en línea]. [Consulta: 23 julio 2022]. Disponible en: https://naylampmechatronics.com/espressif-esp/153-nodemcu-v2-esp8266-wifi.html.

NFC FORUM. *Acerca de la tecnología - Foro NFC.* [en línea]. [Consulta: 20 abril 2022]. Disponible en: https://nfc-forum.org/what-is-nfc/about-the-technology/.

NFC FORUM. *¿Qué es NFC? - Foro NFC.* [en línea]. [Consulta: 20 abril 2022]. Disponible en: https://nfc-forum.org/what-is-nfc/.

NFC FORUM. *Wireless Charging with NFC.* [en línea]. [Consulta: 16 junio 2022]. Disponible en: https://nfc-forum.org/learn/use-cases/wireless-charging/.

NFC FORUM. *NFC Forum: NFC in Action.* [en línea]. [Consulta: 6 febrero 2023]. Disponible en: https://members.nfc-forum.org//aboutnfc/nfc_in_action/.

NOTISEG. *Notiseg - Sistemas de Identificación Nedap*. *Notiseg. S. A* [en línea]. [Consulta: 6 febrero 2023]. Disponible en: http://notiseg.com/index.php/productos/control-de-acceso/sistemas-de-identificacion-nedap.

OMG, 2017. "DATA-DISTRIBUTION SERVICE SPECIFICATION AT OMG". [en línea], 2017, pp. 1-2. [Consulta: 28 abril 2022]. Disponible en: www.omg.org.
PAN, X. y YAMAGUCHI, S., 2022. Machine Learning White-Hat Worm Launcher for Tactical Response by Zoning in Botnet Defense System. *Sensors 2022, Vol. 22, Page 4666*, vol. 22, no. 13, pp. 4666. ISSN 1424-8220. DOI 10.3390/S22134666.

PARET, Dominique. *Antenna designs for NFC devices*. 1. Hoboken: ISTE Ltd. 2016, ISBN 9781848218413.

PILAMUNGA, E. DETERMINACIÓN DE INDICADORES PARA EL PLAN DE MOVILIDAD DE LA EMPRESA PÚBLICA MUNICIPAL - MERCADO DE PRODUCTORES AGRÍCOLAS SAN PEDRO DE RIOBAMBA [en línea]. (Trabajo de titulación). ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, Riobamba. 2019. [Consulta: 6 febrero 2023]. Disponible en: http://dspace.espoch.edu.ec/bitstream/123456789/13282/1/20T01280.pdf.

PORRO, I. *IoT: protocolos de comunicación, ataques y recomendaciones | INCIBE-CERT*. [en línea]. [Consulta: 1 mayo 2022]. Disponible en: https://www.incibe-cert.es/blog/iot-protocolos-comunicacion-ataques-y-recomendaciones.

PORRO, I. *IoT: protocolos de comunicación, ataques y recomendaciones | INCIBE-CERT*. [en línea]. [Consulta: 8 diciembre 2022]. Disponible en: https://www.incibe-cert.es/blog/iot-protocolos-comunicacion-ataques-y-recomendaciones.

PROMETEC. *ESP32 WiFi Bluetooth Wroom-32D - Tienda Prometec.* [en línea]. [Consulta: 24 julio 2022]. Disponible en: https://store.prometec.net/producto/esp32-wifi-bluetooth/.

RAMÍREZ, R. Aplicaciones del RFID como herramienta para el proceso de Marketing (Trabajo de titulación) UNIVERSIDAD DE CHILE, Santiago. 2006.

RAMOS, F. Mecanismo de control de acceso para soluciones IoT de FIWARE en escenarios sanitarios [En línea]. (Trabajo de titulación) Sevilla: Universidad de Sevilla. 2020 [Consulta: 6

febrero 2023]. Disponible en: https://idus.us.es/bitstream/handle/11441/104567/TFG-3071-RAMOS%20ROJAS.pdf?sequence=1&isAllowed=y.

RED HAT, *¿Qué es la arquitectura orientada a los servicios?* [en línea]. [Consulta: 4 mayo 2022]. Disponible en: https://www.redhat.com/es/topics/cloud-native-apps/what-is-service-oriented-architecture.

REDALIA, *Qué es el protocolo SSL/TLS | Redalia*. [en línea]. [Consulta: 6 febrero 2023]. Disponible en: https://www.redalia.es/ssl/protocolo-ssl/.

ROBLEDANO, A. *Qué es MySQL: Características y ventajas*. [en línea]. [Consulta: 13 diciembre 2022]. Disponible en: https://openwebinars.net/blog/que-es-mysql/.

ROBLEDANO, A. *Qué es MySQL: Características y ventajas | OpenWebinars.* [en línea]. [Consulta: 6 febrero 2023]. Disponible en: https://openwebinars.net/blog/que-es-mysql/.

ROSADO, D. NB-IoT Tecnologías celulares narrow-band. (Trabajo de titulación) Universidad Complutense de Madrid, Madrid. 2019.

SALAZAR. JORDI. "Redes Inalámbricas". *TECHpedia*, Prueba, (2016). pp. 40.

SALIKA, F., NASSER, A., MROUE, M., PARREIN, B. & MANSOUR, A. "LoRaCog: A Protocol for Cognitive Radio-Based LoRa Network". *Sensors 2022, Vol. 22, Page 3885*, vol. 22, no. 10, (2022). pp. 3885. ISSN 1424-8220. DOI 10.3390/S22103885.

SAMPAULO, P. "Redes LPWAN: Una guía completa de inicio a fin". (2021), pp. 1-21.

SANTAELLA, J.*¿Qué es Android Studio? - Talently | Talently*. [Blog]. [Consulta: 25 julio 2022]. Disponible en: https://talently.tech/blog/que-es-android-studio/.

SOLECTRO. *Placas de desarrollo.* [en línea]. [Consulta: 24 julio 2022]. Disponible en: https://solectroshop.com/es/10-placas-de-desarrollo.

SORIANO, M. *Seguridad en redes y seguridad de la información*. 2013., pp. 1-80.

SOSA, C., TELLO, E. & LARA, D. "Enfoque para Generar Aplicaciones Orientadas a Servicios para IoT mediante el Desarrollo Dirigido por Modelos". (2016)

STMICROELECTRONICS. *NFC for wireless charging - STMicroelectronics*. [en línea]. [Consulta: 16 junio 2022]. Disponible en: https://www.st.com/content/st_com/en/support/learning/essentials-and-insights/connectivity/nfc/nfc-for-wireless-charging.html.

SUAREZ, J. DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA PAGO DE SERVICIO DE TRANSPORTE PÚBLICO EN ESTACIONES A TRAVÉS DE UN TELÉFONO INTELIGENTE CON TECNOLOGÍA NFC [en línea]. (Trabajo de titulación) ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, Riobamba, 2018 [Consulta: 6 diciembre 2022]. Disponible en: http://dspace.espoch.edu.ec/bitstream/123456789/9158/5/98T00209.pdf.

SUÁREZ, J. *ARDUINO – Blog de Tecnologías.* [blog]. [Consulta: 24 julio 2022]. Disponible en: https://www3.gobiernodecanarias.org/medusa/ecoblog/rsuagued/arduino/.

SUN, X., ZHAO, C., LI, H., YU, H., ZHANG, J., QIU, H., LIANG, J., WU, J., SU, M., SHI, Y. & PAN, L. "Wearable Near-Field Communication Sensors for Healthcare: Materials, Fabrication and Application". *Micromachines 2022, Vol. 13, Page 784*, vol. 13, no. 5, (2022) pp. 784. ISSN 2072-666X. DOI 10.3390/MI13050784.

TAPIA, J. "Mecanismos de seguridad del internet de las cosas Internet of things security mechanisms 2022". *Researchgate* [en línea], 2022, [Consulta: 6 febrero 2023]. Disponible en: https://www.researchgate.net/publication/362876118_Mecanismos_de_seguridad_del_internet_de_las_cosas_Internet_of_things_security_mechanisms_2022.

U.ITU. *Internet de las cosas podría ser la «clave de conectividad» de bajo coste que transforme las vidas en los países en desarrollo*. [en línea]. [Consulta: 25 abril 2022]. Disponible en: https://www.itu.int/net/pressoffice/press_releases/2016/02-es.aspx.

VALDIVIESO, E. FRAMEWORK DE SEGURIDAD PARA DISPOSITIVOS IOT, BASADO EN EL PROTOCOLO OAUTH [en línea] (Trabajo de titulación). UNIVERSIDAD DE LAS FUERZAS ARMADAS, Sangolquí, 2019 [Consulta: 6 febrero 2023]. Disponible en: http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/20830/T-ESPE-039713.pdf?sequence=1&isAllowed=y.

VALVERDE, C. SISTEMA DE CONTROL DE ACCESO DE PERSONAS PARA LOS LABORATORIOS DE LA CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES DE LA UNIVERSIDAD DE GUAYAQUIL UTILIZANDO TECNOLOGÍA NFC (Trabajo de titulación). Universidad de Guayaquil, Guayaquil. 2015

VERSYS. *OPC UA & MQTT, ventajas y casos de uso.* [en línea]. [Consulta: 8 diciembre 2022]. Disponible en: https://www.automaticaeinstrumentacion.com/texto-diario/mostrar/3474577/opc-ua-mqtt-ventajas-casos.

YANG, M.H., LUO, J.N., VIJAYALAKSHMI, M. & SHALINIE, S.M. "Contactless Credit Cards Payment Fraud Protection by Ambient Authentication". *Sensors 2022, Vol. 22, Page 1989*, vol. 22, no. 5, (2022) pp. 1989. ISSN 1424-8220. DOI 10.3390/S22051989.

ANEXOS

ANEXO A: Código implementado en el ESP8266

```
#include <Arduino.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecureBearSSL.h>
#include <NTPClient.h>       // Para hora actual
#include <WiFiUdp.h>
#include <ArduinoJson.h>
#include <PubSubClient.h>
#include <ESP8266WiFi.h>

const long utcOffsetInSeconds = -18000;
char daysOfTheWeek[7][12] = {"DOMINGO", "LUNES",
"MARTES","MIERCOLES", "JUEVES", "VIERNES", "SABADO"};

WiFiUDP ntpUDP;      //Define NTP Client to get time
NTPClient timeClient(ntpUDP,
"pool.ntp.org", utcOffsetInSeconds);
ESP8266WebServer server(80);  //Server on port 80

#include <Wire.h>          //I2C library
#include <PN532_I2C.h>     //Libreria NFC I2C
#include "PN532.h"         //Librería NFC
#include <NfcAdapter.h>    //Libreria NFC
PN532_I2C pn532i2c(Wire);  //Crea objeto del NFC
PN532 nfc(pn532i2c);
#if defined(ESP32)
#include <WiFi.h>
#include <FirebaseESP32.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>
#endif

// Provide the RTDB payload printing info
//and other helper functions.
#include <addons/RTDBHelper.h>

/* 1. Define the WiFi credentials */
#define WIFI_SSID "AccesoIoT"
#define WIFI_PASSWORD "Family.Grab"

/* 2. Define the RTDB URL */
#define DATABASE_URL
"alarma-428b3-default-rtdb.firebaseio.com"
#define saldoMinimo 0.05
#define buzzer D8
#define verde  D6
#define azul   D7
#define rojo   D5
#define ON_Board_LED 2

boolean registro = false; //false lee
String tarjetaActual;
String tarjetaSiguiente;

//FIREBASE
boolean inicioNFC = false;
String stringActual;
String stringSiguiente;
String nombreUsuario;
int contadorConsulta = 0;
unsigned long previousMillis = 0;
const long interval = 1200;

boolean isPhone = true;
boolean datoNFC = false;
boolean publicar = false;

String hex_value;
int epoch_time;

String idUsuario;
float saldoUsuario;
String vehiculoUsuario;
String idTicket;

//get current epoch time
unsigned long Get_Epoch_Time() {
  timeClient.update();
  unsigned long now = timeClient.getEpochTime();
  return now;
}

// MQTT Broker
const char *mqtt_broker = "broker.emqx.io";
const char *topic = "SALIDA1";
const char *topic1 = "ENTRADA1";
const char *mqtt_username = "";
const char *mqtt_password = "";
const int mqtt_port = 1883;

WiFiClient espClient;
PubSubClient client(espClient);

void callback(char *topic1, byte *payload,
  unsigned int length) {
  Serial.print("Message arrived in topic: ");
  Serial.println(topic1);
  Serial.print("Message:");

  String dato = "";
  for (int i = 0; i < length; i++) {
    //Serial.print((char) payload[i]);

    dato += (char) payload[i];
  }
  if (dato == "CORRECTO") {
    digitalWrite(ON_Board_LED, LOW);
  }

  if (dato == "MAL") {
    digitalWrite(ON_Board_LED, HIGH);
  }
  Serial.println(dato);
  Serial.println();
  Serial.println("-----------------------");
```

```
void setup() {
  nfc.begin();
  uint32_t versiondata = nfc.getFirmwareVersion();
  nfc.SAMConfig();

  Serial.begin(115200);
  Serial.println(versiondata);
  nfc.setPassiveActivationRetries(0x01);
  // configure board to read RFID tags
  Serial.println();

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }

  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  Serial.println("Listo Conexiones de red");

  pinMode(buzzer, OUTPUT);
  pinMode(rojo, OUTPUT);
  pinMode(verde, OUTPUT);
  pinMode(azul, OUTPUT);

  digitalWrite(buzzer, LOW);
  digitalWrite(rojo, LOW);
  digitalWrite(verde, HIGH);
  digitalWrite(azul, HIGH);
  delay(500);

  digitalWrite(rojo, HIGH);
  digitalWrite(verde, LOW);
  digitalWrite(azul, HIGH);
  delay(500);

  digitalWrite(rojo, HIGH);
  digitalWrite(verde, HIGH);
  digitalWrite(azul, LOW);
  delay(500);

  digitalWrite(rojo, HIGH);
  digitalWrite(verde, HIGH);
  digitalWrite(azul, LOW);

  timeClient.begin();
  timeClient.setTimeOffset(0);

  Serial.println("LISTOOOO");
  Serial.println("Esperando por tarjeta ISO14443A");

  tono(3);
  client.setServer(mqtt_broker, mqtt_port);
  client.setCallback(callback);
  while (!client.connected()) {
    String client_id = "mqttx_4a2e027c";
    client_id += String(WiFi.macAddress());
    Serial.printf("The client %s connects
    to the public mqtt broker\n", client_id.c_str());
    if (client.connect(client_id.c_str(), mqtt_username,
    mqtt_password)) {
    } else {
      Serial.print("failed with state ");
      Serial.print(client.state());
      delay(2000);
    }
  }
  client.subscribe(topic1);
}
void loop() {
  //-----MQTT----
  client.loop();
  if (WiFi.status() != WL_CONNECTED) {
    digitalWrite(rojo, HIGH);
    digitalWrite(verde, HIGH);
    digitalWrite(azul, HIGH);
  }
  else
  {
    lectura_nfc();
  }
}

//FUNCIONES
void lectura_nfc()
{
  if (isPhone) {
    bool success;
    uint8_t responseLength = 32;
    success = nfc.inListPassiveTarget();
    if (success) {
      uint8_t selectApdu[] = { 0x00, /* CLA */
                               0xA4, /* INS */
                               0x04, /* P1  */
                               0x00, /* P2  */
                               0x07, /* Length of AID  */
                               0xF0, 0x01, 0x02, 0x03, 0x04
                               0x05, 0x06,
                               0x00  /* Le  */
                             };
      uint8_t response[32];
      success = nfc.inDataExchange(selectApdu,
            sizeof(selectApdu,response, &responseLength);
      if (success) {
        uint8_t apdu[] = "ENTRADA";
        uint8_t back[32];
        uint8_t length = 32;
        success = nfc.inDataExchange(apdu, sizeof(apdu),
        back, &length);
        if (success) {
          //   Serial.print("responseLength: ");
          //nfc.PrintHexChar(back, length);
```

```
        }
        else {
          //  Serial.println("Broken connection?");
        }
        hex_value = "";
        hex_value =   printResponse(back, length);



        if (hex_value.length() == 6) {
          isPhone = true;
        }

        datoNFC = false;

        //   Serial.print("VARIABLE: ");
        //Serial.println(hex_value);
        //    Serial.print("responseLength: ");
      }
      else {
        // Serial.println("FALLO SELECT AID");
        isPhone = false;
      }
    }
  }
  else {
    boolean success2;
    uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0 };
    uint8_t uidLength;
  //^********** NFC **********************
  success2 = nfc.readPassiveTargetID
  (PN532_MIFARE_ISO14443A,
  &uid[0], &uidLength);

  while (success2)
  {
    //  Serial.print("responseLength 2: ");

    hex_value = "";
    for (uint8_t i = 0; i < uidLength; i++)
    {
      hex_value += (String)uid[i];
    }

    if (hex_value.length() >= 10) {
      isPhone = false;
    }

    else {

      isPhone = true;
      break;
    }
    success2 = false;
    datoNFC = false;
    delay(1000);
  }
}
  if (!datoNFC) {
    if (hex_value.length() > 0) {
      tono(4);
      epoch_time = Get_Epoch_Time();
      Serial.print("UID: ");
      Serial.print(hex_value);
      Serial.print("  HORA: ");
      Serial.println(epoch_time);
      funcionConsulta(hex_value);
      datoNFC = true;
    }
  }
  if (publicar) {
    funcionAgregarParqueo(hex_value,  String(epoch_time),
    "--", "--", "0.00");
    idTicket = consultaTicket(hex_value);
    ingresoTicket(idUsuario, idTicket);
    client.publish(topic, hex_value.c_str()); //MQTT
    publicar = false;
    digitalWrite(rojo, HIGH);
    digitalWrite(verde, HIGH);
    digitalWrite(azul, LOW);
  }
}
String printResponse(uint8_t *response,
uint8_t responseLength) {
  String respBuffer;
  for (int i = 0; i < responseLength; i++) {
    if (response[i] < 0x10)
  if (response[i] < 0x10)
    respBuffer = respBuffer + "0"; //Adds leading
  respBuffer = respBuffer + String(response[i], HEX);
}
//Serial.print("response salida: "); Serial.println(respBuffer);
respBuffer.toUpperCase();
return respBuffer;

tring consultaTicket(String uid) {
String ticket = "0";
WiFiClient client;
HTTPClient http;    //Declare object of class HTTPClient
String postData;
postData = "UID=" + uid;
if (http.begin(client,
"http://www.espochemmpa.com/acceso/android/consultaTicket.php")) {
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");
  int httpCode = http.POST(postData);   //Send the request
  if (httpCode > 0) {
    if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY){
      String payload = http.getString();
      Serial.println("Longitud: " + String(payload.length()));
      Serial.println(payload);
      Serial.println("Codigo HTTP: " + String(httpCode)); //Print HTTP return
      if (payload.length() > 2) {
        DynamicJsonBuffer jsonBuffer;
        JsonArray& arrayl = jsonBuffer.parseArray(payload);
        //ticket = String(arrayl[0]["id"]);
        ticket = arrayl[0]["id"].asString();
```

```
          ticket = arrayl[0]["id"].asString();
          Serial.println("ID ticket: " + ticket);
        }
        else {
          Serial.println("TAG NO REGISTRADA");
        }
      }
    }
    http.end();  //Close connection
  }
  else {
    Serial.printf("[HTTPS] Unable to connect\n");
  }
  return ticket;
}
void ingresoTicket(String id, String ticket) {
  WiFiClient client;
  HTTPClient http;    //Declare object of class HTTPClient
  String postData;
  postData = "id=" + id + "&ticket=" + ticket;
  if (http.begin(client,
"http://www.espochemmpa.com/acceso/android/ingresoUsuarioParqueo.php")){
 http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    int httpCode = http.POST(postData);   //Send the request
    if (httpCode > 0) {
      if (httpCode == HTTP_CODE_OK
      || httpCode == HTTP_CODE_MOVED_PERMANENTLY){
        String payload = http.getString();
        Serial.println("Payload: " + payload);
       Serial.println("Codigo HTTP: " + String(httpCode)); //Print HTTP return
       if (payload.equals("ok")) {
         Serial.println("TICKET INGRESADO!");
       }
       else {
         Serial.println("ALGO SALIO MAL TICKET");
       }
      }
    }
    http.end();  //Close connection
  }

  else {
    Serial.printf("[HTTPS] Unable to connect\n");
  }
}
void funcionAgregarParqueo(String uid, String ingreso, String salida, String tiempo, Stri
  WiFiClient client;
  HTTPClient http;    //Declare object of class HTTPClient
  String postData;
  postData = "uid=" + uid + "&ingreso=" + ingreso + "&salida=" + salida + "&tiempo=" + ti
  if (http.begin(client, "http://www.espochemmpa.com/acceso/android/insertarParqueo.php")
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    int httpCode = http.POST(postData);   //Send the request
    if (httpCode > 0) {
      if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY) {
        String payload = http.getString();
        Serial.println("Payload: " + payload);
        Serial.println("Codigo HTTP: " + String(httpCode)); //Print HTTP return code
```

```
412         if (payload.equals("ok")) {
413           Serial.println("REGISTRO PARQUEO SUBIDOS!");
414         }
415         else {
416           Serial.println("ALGO SALIO MAL");
417         }
418       }
419     }
420     http.end();  //Close connection
421   }
422   else {
423     Serial.printf("[HTTPS] Unable to connect\n");
424   }
425 }
426 void funcionAgregarHistorial(String id, String debito, String credito, String tipo, String fecha) {
427   WiFiClient client;
428   HTTPClient http;    //Declare object of class HTTPClient
429   String postData;
430   postData = "id=" + id + "&debito=" + debito + "&credito=" + credito + "&tipo=" + tipo + "&horario=" + fecha;
431   if (http.begin(client, "http://www.espochemmpa.com/acceso/android/insertarRegistro.php")) {
432     http.addHeader("Content-Type", "application/x-www-form-urlencoded");
433     int httpCode = http.POST(postData);   //Send the request
434     if (httpCode > 0) {
435       if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY) {
436         String payload = http.getString();
437         Serial.println("Payload: " + payload);
438         Serial.println("Codigo HTTP: " + String(httpCode)); //Print HTTP return code
439         if (payload.equals("ok")) {
440           Serial.println("DATOS SUBIDOS!");
441         }
442         else {
443           Serial.println("ALGO SALIO MAL");
444         }
445       }
446     }
447     http.end();  //Close connection
448   }
449   else {
450     Serial.printf("[HTTPS] Unable to connect\n");
451   }
452 }
453 void funcionConsulta(String uid) {
454   WiFiClient client;
455   HTTPClient http;    //Declare object of class HTTPClient
456   String postData;
457   postData = "UID=" + uid;
458   if (http.begin(client, "http://www.espochemmpa.com/acceso/android/consultaUID.php")) {
459     http.addHeader("Content-Type", "application/x-www-form-urlencoded");
460     int httpCode = http.POST(postData);   //Send the request
461     if (httpCode > 0) {
462       if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY) {
463         String payload = http.getString();
464         Serial.println("CONEXION EXITOSA!!!");
465         Serial.println("Longitud: " + String(payload.length()));
466         Serial.println(payload);
467         Serial.println("Codigo HTTP: " + String(httpCode)); //Print HTTP return code
468         if (payload.length() > 2) {
469           digitalWrite(rojo, HIGH);
```

```
            digitalWrite(verde, LOW);
            digitalWrite(azul, HIGH);
            DynamicJsonBuffer jsonBuffer;
            JsonArray& arrayl = jsonBuffer.parseArray(payload);
            //String strSaldo = String(arrayl[0]["saldo"]);
            String strSaldo = arrayl[0]["saldo"].asString();
            float saldo = strSaldo.toFloat();
            //idUsuario = String(arrayl[0]["id"]);
            idUsuario = arrayl[0]["id"].asString();
            //vehiculoUsuario = String(arrayl[0]["vehiculo"]);
            vehiculoUsuario = arrayl[0]["vehiculo"].asString();
            saldoUsuario = saldo;
            //Serial.println("ID: " + String(arrayl[0]["id"]));
            //Serial.println("DATO: " + String(arrayl[0]["saldo"]));
            //Serial.println("CARRO: " + vehiculoUsuario);
            if (saldo < saldoMinimo) {
              Serial.println("SIN SALDO");
            }
            else {
              publicar = true;
              if (vehiculoUsuario.indexOf("Peque") > 0) {
                Serial.println("si es pequeño");
              }
            }
          }
          else {
            digitalWrite(rojo, LOW);
            digitalWrite(verde, HIGH);
            digitalWrite(azul, HIGH);
          delay(2000);
          digitalWrite(rojo, HIGH);
          digitalWrite(verde, HIGH);
          digitalWrite(azul, LOW);
          Serial.println("TAG NO REGISTRADA");
        }
      }
    }
    http.end();  //Close connection
  }

  else {
    Serial.printf("[HTTPS] Unable to connect\n");
  }

}
//***FUNCION MUSICAS*****************************
//***********************************************
void tono(int tipo) {    // Tonos  1->Error  2->C
  switch (tipo) {
    case 1:
      break;
    case 2:
      tone(buzzer, 1000);  delay(50);
      tone(buzzer, 4000);  delay(50);
      tone(buzzer, 1000);  delay(50);
      noTone(buzzer);
      break;
    case 3: //inicio del sistema
```

ANEXO B: Conexión a la base de datos

```php
<?php
    class Database
    {
        private static $dbName = 'u488464325_acceso' ;
        private static $dbHost = 'localhost' ;
        private static $dbUsername = 'u488464325_admin2';
        private static $dbUserPassword = 'Grab280895';

        private static $cont  = null;

        public function __construct() {
            die('Init function is not allowed');
        }

        public static function connect()
        {
           // One connection through whole application
           if ( null == self::$cont )
           {
            try
            {
              self::$cont =  new PDO( "mysql:host=".self::$dbHost.";"."dbname=".self::$dbName, self::$dbUsername, self::$dbUserPassword);
            }
            catch(PDOException $e)
            {
              die($e->getMessage());
            }
           }
           return self::$cont;
        }

        public static function disconnect()
        {
            self::$cont = null;
        }
    }
?>
```

ANEXO C: Código para Actualizar usuarios en la interfaz

```php
<?php
    require 'database.php';
    $uid = null;
    if ( !empty($_GET['uid'])) {
        $uid = $_REQUEST['uid'];
    }

    $pdo = Database::connect();
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "SELECT * FROM usuario where uid = ?";
    $q = $pdo->prepare($sql);
    $q->execute(array($uid));
    $data = $q->fetch(PDO::FETCH_ASSOC);
    Database::disconnect();
?>
```

```php
<form class="form-horizontal" action="user_dataedit_tb.php?uid=<?php echo $uid?>" method="post">
    <div class="control-group">
        <label class="control-label">UID</label>
        <div class="controls">
            <input name="uid" type="text"  placeholder="" value="<?php echo $data['uid'];?>" readonly>
        </div>
    </div>

    <div class="control-group">
        <label class="control-label">Nombre</label>
        <div class="controls">
            <input name="nombre" type="text"  placeholder="" value="<?php echo $data['nombre'];?>" required>
        </div>
    </div>

    <div class="control-group">
        <label class="control-label">Apellido</label>
        <div class="controls">
            <input name="apellido" type="text"  placeholder="" value="<?php echo $data['apellido'];?>" required>
        </div>
    </div>


    <div class="control-group">
        <label class="control-label">Email</label>
        <div class="controls">
            <input name="email" type="text" placeholder="" value="<?php echo $data['email'];?>" required>
        </div>
    </div>
```

```
                    <div class="control-group">
                        <label class="control-label">Tipo de vehículo</label>
                        <div class="controls">
                            <select name="vehiculo" id="mySelect">
                                <option value="Pequeño 0-2 tn">Pequeño 0-2 tn</option>
                                <option value="Mediano 2.1-5 tn">Mediano 2.1-5 tn</option>
                                <option value="Grande 5.1 tn">Grande 5.1 tn </option>
                                <option value="Mula">Mula</option>
                                <option value="Trailer">Trailer</option>
                                <option value="Frutas_Tropicales">Frutas Tropicales</option>
                                <option value="Triciclos">Triciclos </option>
                                <option value="Frecuente">Frecuente</option>
                                </select>
                        </div>
                    </div>

                    <div class="control-group">
                        <label class="control-label">Saldo</label>
                        <div class="controls">
                            <input name="saldo" type="text" placeholder="" value="<?php echo $data['saldo'];?>" readonly>
                        </div>
                    </div>

                    <div class="control-group">
                        <label class="control-label">PLACA</label>
                        <div class="controls">
                            <input name="placa" type="text" placeholder="" value="<?php echo $data['placa'];?>" required>
                        </div>
                    </div>


                    <div class="form-actions">
                        <button type="submit" class="btn btn-success">Actualizar</button>
                        <a class="btn" href="user_data.php">Salir</a>
                    </div>
                </form>
```

ANEXO D: Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.nfc.hcefinal">

    <uses-permission android:name="android.permission.NFC" /> <!-- Require
    <uses-feature
        android:name="android.hardware.nfc.hce"
        android:required="true" />

    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="hceFInal"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.HceFInal"
        android:usesCleartextTraffic="true"
        tools:targetApi="31">
        <activity
```

```
        <activity
            android:name=".activityPin"
            android:exported="false"
            android:theme="@style/AppTheme.NoActionBar"
            >
            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>

        <activity
            android:name=".listaTarjetas"
            android:theme="@style/AppTheme.NoActionBar">
            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>
        <activity
            android:name=".activityIntro"
            android:exported="true"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
```

```
        <uses-library
            android:name="org.apache.http.legacy"
            android:required="false" />

        <activity
            android:name=".registro_tarjeta"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".editar_usuario"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".registro_usuario"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".activityLogin"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".MainActivity"
            android:screenOrientation="portrait"
            android:theme="@style/AppTheme.NoActionBar" />
```

```
    <service
        android:name=".HostCardEmulatorService"
        android:exported="true"
        android:permission="android.permission.BIND_NFC_SERVICE">
        <intent-filter>
            <action android:name="android.nfc.cardemulation.action.HOST_APDU_SERVICE" />
        </intent-filter>

        <meta-data
            android:name="android.nfc.cardemulation.host_apdu_service"
            android:resource="@xml/apduservice" />
    </service>
</application>

</manifest>
```

ANEXO E: Código para realizar el reconocimiento de placas vehiculares

```
broker = 'broker.emqx.io'
port = 1883
topic = "SALIDA1"
topic1 = "ENTRADA1"
client_id = f'python-mqtt-{random.randint(0, 1000)}'
username = 'admin'
password = '11111'


# Setup google authen client key
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = 'placa-emmpa-3c44932c28ec.json'

# Source path content all images
SOURCE_PATH = r"C:/Plate/"

def recognize_license_plate(img_path):

    global mydb,mycursor, strplaca, strmqtt

    start_time = datetime.now()

    # Read image with opencv
    img = cv2.imread(img_path)

    # Get image size
    height, width = img.shape[:2]

    # Scale image
    img = cv2.resize(img, (600, int((height * 600) / width)))

    # Show the origin image
    #cv2.imshow('Origin image', img)

    # Save the image to temp file
    cv2.imwrite(SOURCE_PATH + "output.jpg", img)

    # Create new img path for google vision
    img_path = SOURCE_PATH + "output.jpg"

    # Create google vision client
    client1 = vision.ImageAnnotatorClient()

    # Read image file
    with io.open(img_path, 'rb') as image_file:
        content = image_file.read()

    image = vision.types.Image(content=content)

    # Recognize text
    response = client1.text_detection(image=image)
    texts = response.text_annotations
    #print(texts)

    for text in texts:
        if len(text.description) == 3 :
            license_plate = text.description
            a=[str(license_plate)]
            print(a)
        if len(text.description) == 4 :
            license_plate1 = text.description
            num=str(license_plate1)
            s = [int(s) for s in str.split(num) if s.isdigit()]
            #print(s)
            sin_strings = []
            for string in s:
                if string != "" and string != " ":
                    sin_strings.append(string)
                    print(sin_strings)

            lic = a + sin_strings
            global lic1
            lic1= "".join(map(str, lic))
            print("PLACA: " + lic1)
            strplaca = lic1



            mydb = mysql.connector.connect(
            host="sql1757.main-hosting.eu",
            user="u488464325_admin2",
            password="Grab280895",
            database="u488464325_acceso"
            )
            #global mydb,mycursor
            mycursor = mydb.cursor()
            sql = "SELECT placa FROM usuario WHERE uid= '%s'" % strmqtt
                #uid = (a)
            mycursor.execute(sql)
            myresult = mycursor.fetchone()
            #print(myresult)
            global x
```

```
            for x in myresult:
                print("BASE DE DATOS: " +str(x))

                if(str(x) == strplaca):
                    client.publish(topic1, "CORRECTO")
                    print("LAS PLACAS COINCIDEN")

                else:

                    client.publish(topic1, "MAL")
                    print("NO HAY COINCIDENCIA")



            mycursor.close()
            #con.close()



# The callback for when the client receives a CONNACK response from the server.
def on_connect(client, userdata, flags, rc):
    #print("Connected with result code "+str(rc))

    if rc == 0:
            print("Connected to MQTT Broker!")
    else:
        print("Failed to connect, return code %d\n", rc)

    # Subscribing in on_connect() means that if we lose the connection and
    # reconnect then subscriptions will be renewed.

    client.publish("ENTRADA1", "CONEXION LISTA")
    client.subscribe("SALIDA1")

# The callback for when a PUBLISH message is received from the server.
def on_message(client, userdata, msg):
    #print(msg.topic+" "+str(msg.payload))
    global strmqtt

    print("LLEGA: " + str(msg.payload.decode()))

    a = msg.payload.decode()

    strmqtt = a
```

```
  cap = cv2.VideoCapture( "rtsp://admin:GRAB.230895@192.168.0.180:554/Streaming/Channels/101")
  #cap = cv2.VideoCapture( "rtsp://admin:GRAB.230895@175.25.204.110:554/Streaming/Channels/101",cv2.CAP_FFMPEG)
  cap.set(cv2.CAP_PROP_FRAME_WIDTH,720) # ancho
  cap.set(cv2.CAP_PROP_FRAME_HEIGHT,480) # alto

  # Tomar una imagen
  ret, frame = cap.read()
  # Guardamos la imagen en un archivo
  cv2.imwrite(r'C:/Data/placa.jpg',frame)
  #Liberamos la camara
  cap.release()

  print('---------- Start recognize license palate --------')
  path = SOURCE_PATH + 'placa.jpg'
  recognize_license_plate(path)

  print('---------- End ----------')

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

client.connect("broker.emqx.io", 1883, 60)
```

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL APRENDIZAJE

UNIDAD DE PROCESOS TÉCNICOS

REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 27/ 03 / 2023

INFORMACIÓN DEL AUTOR
Nombres – Apellidos: Cristian Noe Saez Saez
INFORMACIÓN INSTITUCIONAL
Facultad: Informática y Electrónica
Carrera: Telecomunicaciones
Título a optar: Ingeniero En Electrónica, Telecomunicaciones Y Redes
f. Analista de Biblioteca responsable: Ing. Fernanda Arévalo M.