



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL**

CARRERA DE INGENIERÍA ELECTRÓNICA

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE:
“INGENIERO ELECTRÓNICO”**

PROYECTO TÉCNICO:

**“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE
MARCACIÓN CON CONTROL DACTILAR UTILIZANDO
HARDWARE DE BAJO COSTO Y VISUALIZADO A TRAVÉS DE
UNA APLICACIÓN WEB”**

AUTORES:

Johnny Xavier Escandón Pin
Roberto Carlos Coronel Molina

TUTOR:

Ing. Pablo Fabián Echeverría Ávila.

GUAYAQUIL – ECUADOR

2020

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Johnny Xavier Escandón Pin y Roberto Carlos Coronel Molina, autores del proyecto técnico: "DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE MARCACIÓN CON CONTROL DACTILAR UTILIZANDO HARDWARE DE BAJO COSTO Y VISUALIZADO A TRAVÉS DE UNA APLICACIÓN WEB", declaramos que el contenido de este proyecto técnico aquí descrito es de nuestra exclusiva responsabilidad y autoría. Autorizamos a la UNIVERSIDAD POLITECNICA SALESIANA la publicación total del trabajo de titulación y su reproducción sin beneficio de ganancia.

Guayaquil, julio del 2020



(f) Johnny Xavier Escandón Pin
C.I: 2400251555

(f) Roberto Carlos Coronel Molina
C.I: 0930954649

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTORES

Nosotros, JOHNNY XAVIER ESCANDON PIN, con documento de identificación No 2400251555, y ROBERTO CARLOS CORONEL MOLINA con documento de identificación No 0930954649 manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de grado titulado: "DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE MARCACIÓN CON CONTROL DACTILAR UTILIZANDO HARDWARE DE BAJO COSTO Y VISUALIZADO A TRAVÉS DE UNA APLICACIÓN WEB", mismo que ha sido desarrollado para optar por el título de Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos antes cedidos.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos de los derechos morales de la obra ante citada. En concordancia, suscrito este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil julio del 2020



(f) _____
Johnny Xavier Escandón Pin
C.I.:2400251555



(f) _____
Roberto Carlos Coronel Molina
C.I.:0930954649

CERTIFICADO DE DIRECCIÓN DE TRABAJO DE TITULACIÓN

Yo Msc. Pablo Echeverría certifico que bajo mi dirección y asesoría fue desarrollado el proyecto técnico de titulación “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE MARCACIÓN CON CONTROL DACTILAR UTILIZANDO HARDWARE DE BAJO COSTO Y VISUALIZADO A TRAVÉS DE UNA APLICACIÓN WEB”, realizado por los señores Johnny Xavier Escandón Pin y Roberto Carlos Coronel Molina para la obtención del título de Ingeniero Electrónico.

Por medio de la presente certifico que el documento cumple con los requisitos establecidos en el Instructivo para la Estructura y Desarrollo de Trabajos de Titulación para pregrado de la Universidad Politécnica Salesiana.

En virtud de lo anterior, autorizo su presentación y aceptación como una obra auténtica y de alto valor académico.

Guayaquil, julio del 2020



(f) _____

Ing. Pablo Echeverría

DIRECTOR DE PROYECTO DE TITULACIÓN

DEDICATORIA

Este proyecto va dedicado primero a Dios por brindarme la sabiduría necesaria para lograr esta meta en mi vida, porque sin su bendición nada de esto hubiese sido posible, a mis padres que con amor supieron enseñarme valores esenciales para formarme como persona y a ser perseverante para llegar a cualquier meta propuesta, a mi querida amiga (Carmen Manrique) quien han sido una pieza fundamental, recibiendo un apoyo incondicional para culminar una etapa más de mi vida, a mi tía (Colombia Sáenz) que ha sido de gran ayuda en mi etapa universitaria. Todos ustedes son los más valioso que Dios me ha dado en vida ya que han estado presente brindándome su apoyo y sus consejos, y este logro profesional va en honor a ellos.

Johnny Xavier Escandón Pin

Este proyecto de titulación se lo dedico primero a Dios todo poderoso por todas las bendiciones recibidas, en especial a mis padres que están orgullosos de mí, por ser los pilares fundamentales no solo en conocimientos y cuidados, sino también en valores demostrando como ser un ciudadano de bien, agradezco todo el apoyo que me brindaron desde mis primeros pasos ,a mi hermana (María Coronel) que siempre ha sido mi apoyo fundamental en toda mi carrera universitaria, recibiendo un apoyo incondicional para culminar una etapa más de mi vida. A todos mis amigos y profesores que me impartieron sus conocimientos y me impulsaron a seguir adelante en todo el camino.

Roberto Carlos Coronel Molina

AGRADECIMIENTOS

Agradezco inmensamente a mi Dios todopoderoso por ser el maestro que nos guía en nuestro camino por las bendiciones y por la sabiduría impartida en nosotros, a nuestros queridos padres quienes siempre nos han estado apoyando en cada momento de nuestras vidas nos han brindado su apoyo en cada uno de los aspectos de la misma, creyendo siempre en nosotros y en nuestro potencial; agradecemos a todos nuestros

A nuestro Tutor MSc.. Pablo Fabián Echeverría Ávila quien siempre nos brindó su gran apoyo para salir adelante con nuestro proyecto de titulación.

A nuestros revisores quienes con sus conocimientos y experiencias nos ayudaron de una u otra manera a la realización de este proyecto de titulación.

RESUMEN

Año	Título	Alumnos	Tutor	Tema de titulación
2020	Ingeniero Electrónico	Johnny Xavier Escandón Pin Roberto Carlos Coronel Molina	Ing. Pablo Echeverría	Diseño e implementación de un prototipo de marcación con control dactilar utilizando hardware de bajo costo y visualizado a través de una aplicación web

El presente proyecto de titulación se planteó para el desarrollo de una plataforma de control de marcación de acceso biométrico cuyo objetivo fue optimizar la gestión de acceso de empleados, el cual permitirá mantener información para la institución automáticamente a través del sistema, y vía correo electrónico o a su vez mediante mensaje de texto, se pudo comprobar que con los procedimientos existentes la gestión no era eficaz debido a que los equipos estaban delineados para trabajar de carácter independiente.

Con este proyecto se controla al personal de la institución, y permitirá registrar todas las novedades que se registran al momento de marcar, como; asistencia, puntualidad y compromiso. El fin de este proyecto se orientó para resolver problemáticas que en algunas instituciones desconocen por no contar un control interno, por lo tanto, la implementación de esta herramienta que contiene una aplicación web facilitará visualizar mediante dirección IP dinámica registro de asistencia, una vez que haya marcado con la huella dactilar.

El procesamiento de esta aplicación será por medio de la placa Asus Tinker Board ya que ofrece un rendimiento significativo gracias a sus 2 GB de memoria doble canal LPDDR3, conectado a un biométrico el cual contará con un servidor en la nube donde se instalará varios servidores como es el caso del servidor de correos que permitirá enviar todas las bases de datos por correo al usuario o delegado que lo requiera para su respectivo control de los atrasos en los horarios de entrada y salida, también contará con un servidor web donde se le hará una aplicación multiplataforma para que sea compatible con cualquier sistema operativo (ANDROID, IOS), dentro de la aplicación de administrador contará con una opción para generar reportes de las marcaciones bien detallados y coloridos para el agrado de los usuarios.

Aportará al desarrollo de herramientas virtuales innovadores en la escuela de Manejo CEDUCAR, mejorará la calidad del servicio del registro de marcación,

dispondrá de instrumentos tecnológicos que serán de gran ayuda al personal de la institución, fortaleciendo el área de sistemas, mediante la mejora de los instrumentos de bajo costo y fácil uso que se usará en los equipos que disponen el establecimiento para hacer la respectiva actualización y que se sientan seguros con el nuevo sistema de implementación.

ABSTRACT

Year	Profession	Students	Director of Technical Project	Technical Project
2020	Electronic Engineer	Johnny Xavier Escandón Pin Roberto Carlos Coronel Molina	Ing. Pablo Echeverría.	Design and implementation of a fingerprint control dialing prototype using low-cost hardware and visualized through a web application.

The present degree project was proposed for the development of a biometric access dialing control platform whose objective was to optimize the management of employee access, which will allow maintaining information for the institution automatically through the system, and via email or In turn, through a text message, it was found that with the existing procedures, the management was not effective because the teams were outlined to work independently.

With this project the staff of the institution is controlled, and will allow registering all the news that are registered at the time of marking, such as; assistance, punctuality and commitment. The end of this project was aimed at solving problems that some institutions are unaware of because they do not have internal control, therefore, the implementation of this tool that contains a web application will facilitate viewing the attendance record using a dynamic IP address, once it has been marked with the fingerprint.

The processing of this application will be through the Asus Tinker Board since it offers significant performance thanks to its 2 GB of dual-channel LPDDR3 memory, connected to a biometric which will have a cloud server where several servers will be installed such as This is the case of the mail server that will allow all databases to be sent by mail to the user or delegate who requires it for their respective control of delays in entry and exit times, it will also have a web server where a multiplatform application to be compatible with any operating system (ANDROID, IOS), within the administrator application will have an option to generate reports of the dials well detailed and colorful for the liking of users.

It will contribute to the development of innovative virtual tools in the CEDUCAR Driving School, it will improve the quality of the dial registration service, it will have technological instruments that will be of great help to the staff of the institution,

strengthening the systems area, by improving the Low-cost and easy-to-use instruments that will be used in the teams that have the establishment to make the respective update and feel secure with the new implementation system.

ABREVIATURAS

ADSL: Asymmetric Digital Subscriber Line “Línea de abonado digital asimétrica.

MONGODB: Lenguaje de programación de base de datos No SQL.

JS: Java Script

HTML: Hiper Text Markup Lenguaje, “Lenguaje de marcado hipertexto”.

GPU: Graphics Processing Unit “Unidad de procesamiento grafico”.

HDMI: High Definition Multimedia Interface “Interfaz multimedia en alta definición”.

GPIO: General Purpose Input Output “Salida de entrada de propósito general”.

NPM: Node Package Manager “Gestión de paquetes node”.

VPN: Virtual Private Network “Red privada virtual”.

DSL: Domain Specific Lenguaje “Lenguaje específico de dominio”.

PLUGINS: componente de software que agrega una característica específica a un programa de computo existente.

ETHERNET: es una de las grandes tecnologías de red de área local (LAN) más utilizada.

LAN: Local Área Network “Red de área Local”.

POST: Power-on self-test, envío de información de los formularios válidos.

API: Aplicación Programming Interface “Interfaz de programación de aplicaciones”.

CRUD: Create Read Update delete, “Crear leer Actualizar Eliminar”, se usa para referirse a las funciones básicas de base de datos.

BACK-END: lo que no ve el usuario y está en la parte del servidor.

FRONT-END: lo que ve el usuario en la pantalla del sistema biométrico.

INDICE GENERAL

Contenido

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN	II
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTORES	III
CERTIFICADO DE DIRECCIÓN DE TRABAJO DE TITULACIÓN.....	IV
DEDICATORIA.....	V
AGRADECIMIENTOS.....	VI
RESUMEN.....	VII
ABSTRACT	IX
ABREVIATURAS	XI
INDICE GENERAL	XII
ÍNDICE DE TABLAS.....	XVI
ÍNDICE DE FIGURAS.....	XVII
INTRODUCCION.....	XIX
1. El Problema.....	1
1.1. Planteamiento del problema.....	1
1.2. Importancia y alcances	1
1.3. Delimitación del problema.....	2
1.3.1. Delimitación espacial.....	2
1.3.2. Delimitación temporal.....	2
1.3.3. Delimitación social.....	3
1.4. Justificación.....	3
1.5. Grupo objetivo.....	4
1.5.1. Población.....	4
1.5.2. Muestra.....	4
1.6. Objetivos	4
1.6.1. Objetivo general	4
1.6.2. Objetivos específicos.....	4
1.7. Metodología.....	5
1.7.1. Métodos	5
1.7.2. Método deductivo	5
1.7.3. Método experimental	6
1.7.4. Técnicas.....	6
1.7.5. Técnica de investigación	6
1.7.6. Observación	6

1.7.7.	La entrevista.....	6
1.7.8.	Recolección de información.....	6
1.8.	Descripción de la propuesta	7
1.9.	Beneficiarios.....	7
1.10.	Impacto.....	7
CAPÍTULO II.....		8
2.	Marco Teórico.....	8
2.1.	La biometría en la identificación de personas	8
2.2.	Control de acceso biométrico	8
2.3.	Sensor biométrico.....	9
2.3.1.	Reconocimiento de huellas dactilares.....	10
2.4.	Patrones de huellas dactilares.....	11
2.5.	Tipos de Hardware lectores de huella dactilar	11
2.5.1.	Lectores ópticos	11
2.5.2.	Lectores capacitivos	12
2.5.3.	Lectores de ultrasonido	12
2.5.4.	Lectores térmicos	13
2.6.	Sensor de huella dy50	13
2.7.	Servidores de bases de datos	14
2.8.	Base de datos Oracle	15
2.8.1.	Arquitectura general de aplicación de oracle	16
2.9.	Servidor de Vultr.....	17
2.10.	Mongo DB.....	18
2.10.1.	Ventajas de MongoDB	19
2.11.	Software Python.....	20
2.11.1.	Python-flask	21
2.11.2.	Pymongo.....	21
2.12.	Asus Tinker Board.....	22
2.13.	Servidor Web	23
2.14.	Apache HTTP Server	24
2.15.	Api Rest	25
2.15.1.	Comprender el diseño de API REST	25
2.15.1.1.	de cliente.....	25
2.15.1.2.	Apátrida.....	26
2.15.1.3.	Cache.....	26
2.15.1.4.	Interfaz uniforme.....	27
2.15.1.5.	Sistema de capas.....	27

2.15.1.6.	Código bajo demanda	27
2.16.	Nginx server.....	28
2.17.	App VueJS.....	29
	Ventajas.....	29
2.17.1.	tamaño muy pequeño	29
2.17.2.	Fácil de entender y desarrollar aplicaciones	29
2.17.3.	Integración simple.....	30
2.17.4.	Documentación detallada.....	30
2.17.5.	Flexibilidad.....	30
2.17.6.	Comunicación de dos vías	30
2.18.	Visual Studio Code.....	31
2.19.	Software NodeJS	31
2.20.	Modulo quemador Arduino Pro Mini	32
	CAPÍTULO III.....	33
3.	Diseño del sistema de marcación con control dactilar.....	33
3.1.	Especificaciones del sistema de marcación.....	33
3.2.	Estructura de la red del sistema	33
3.3.	Análisis de comunicación entre los dispositivos.....	33
3.4.	Conexión de la tarjeta asus tinker board a internet.....	34
3.5.	Análisis de la obtención de datos	34
3.6.	Obtención de datos del sensor de huella dactilar.....	34
3.7.	Base de datos.....	36
3.7.1.	Tabla de administradores	37
3.7.2.	Tabla de usuarios	38
3.7.3.	Tabla de Marcaciones	38
3.7.4.	Entrada.....	39
3.7.5.	Salida almuerzo.....	39
3.7.6.	Regreso almuerzo	40
3.7.7.	Salidas	40
3.7.8.	Estructura total de la base de datos.....	40
3.8.	Servidor de infraestructura en la nube vultr.	41
3.9.	Comunicación del api con la base de datos.....	42
3.9.1.	Interfaz de las peticiones del api del servidor Ceducar	43
3.9.2.	Peticiones para administradores.....	43
3.9.3.	Peticiones para usuarios	43
3.9.4.	Peticiones para marcaciones.....	43
3.9.5.	Peticiones para ver todos los reportes.....	44

3.10.	Procesamiento de solicitudes.....	44
3.11.	Interfaz de la aplicación del sensor biométrico-.....	44
3.12.	Desarrollo de aplicaciones de administrador y usuario para el sistema de marcación.	45
3.13.	Front-end (Aplicación administrador) – Nuxt Vue.js.....	45
3.14.	Front-end (Aplicación Administrador) – Nuxt Vue.js	46
3.15.	Interfaz del sensor biométrico	46
3.16.	Front-end (Aplicación de biométrico) – Python-HTML	46
CAPÍTULO IV		48
4.	ANÁLISIS EXPERIMENTAL DEL SISTEMA DE CONTROL DE ACCESO.....	48
4.1.	Análisis experimental del Dispositivo de acceso Biométrico	48
4.2.	Conexión sensor dy50 biométrico.....	49
4.3.	Implementación del proyecto	50
Conclusiones		54
Recomendaciones.....		55
Bibliografía.....		56
ANEXOS		58
ANEXO 1: PRESUPUESTO		58
ANEXO 2: CRONOGRAMA DE ACTIVIDADES.....		58

ÍNDICE DE TABLAS

Tabla 1 Especificaciones del sensor óptico DY50.....	14
Tabla 2 Beneficios y características del servidor Vultr.	17
Tabla 3 Especificaciones de la tarjeta Asus Tinker Board.....	23
Tabla 4 Presupuesto de materiales.....	59
Tabla 5 Cronograma de actividades	58

ÍNDICE DE FIGURAS

Figura 1.3.1. Ubicación de la Escuela de Manejo 'Ceducar', vista satelital.	2
Figura 2.1 Huella dactilar	8
Figura 2.2 <i>Reconocimiento de huellas</i>	10
Figura 2.3 Lectores ópticos.....	12
Figura 2.4 <i>Lectores capacitivos</i>	12
Figura 2.5 Lectores de ultrasonido.....	13
Figura 2.6 Lectores térmicos	13
Figura 2.7 Sensor de huella DYS50.....	14
Figura 2.8 Servidor de base de datos oracle.....	15
Figura 2.9 Logo del software python.....	20
Figura 2.10 Tarjeta asus tinker board para el sistema de marcación.	23
Figura 2.11 Diseño esquemático del servidor web.....	24
Figura 2.12 Ilustración del servidor apache.....	25
Figura 2.13 Logo del servidor nginx.....	28
Figura 2.14 Logo del servidor web vue.js.....	29
Figura 2.15 Captura de pantalla de kdevelop.....	31
Figura 2.16 Logo del software node js	32
Figura 2.17 Modulo convertidor de datos para la asus tinker board	32
Figura 3.1 <i>Diagrama de conexión de datos de la tarjeta asus tinker board al sistema de marcación del servidor de Ceducar</i>	34
Figura 3.2 Diagrama de obtención del código de huella a través del sensor biométrico.	35
Figura 3.3 Diagrama de obtención de datos del sistema de marcación.	36
Figura 3.4 <i>Interfaz mongodb compass para acceder a la base de datos</i>	37
Figura 3.5 Tablas del sistema ceducar en MongoDB.....	37
Figura 3.6 Estructura de registro para el administrador.....	38
Figura 3.7. <i>Estructura para el registro de usuarios</i>	38
Figura 3.8. Estructura para el registro de marcaciones.....	39
Figura 3.9. Estructura de registro de Marcaciones.....	39
Figura 3.10. Estructura de registro de Salida de Almuerzo.	39
Figura 3.11. Estructura de registro de Salida de Almuerzo.	40
Figura 3.12. Estructura de registro de salida.....	40
Figura 3.13. Estructura final de los datos de sistema de marcación.....	41
Figura 3.14. <i>Servidor en la Nube del sistema de marcación</i>	42
Figura 3.15 Base de datos del api a través de la ip estática.	42

Figura 3.16 Peticiones para administrador del servidor CEDUCAR	43
Figura 3.17 Peticiones para usuarios del servidor Ceducar	43
Figura 3.18 Peticiones de marcaciones del servidor Ceducar.....	44
Figura 3.19 Peticiones para reportes del servidor Ceducar.....	44
Figura 3.20 Interfaz de inicio de sesión de la aplicación de administrador del sistema de marcación Ceducar.....	45
Figura 3.21 Interfaz de inicio de sesión de la aplicación de usuario del sistema de marcación Ceducar.....	46
Figura 3.22 Interfaz final del sistema de marcación Ceducar	47
Figura 4.1 Diagrama General del Proyecto.....	48
Figura 4.2 Montaje de la tarjeta Asus al Tablero del sistema de marcación	50
Figura 4.3 Cableado general de la tarjeta asus tinker board con el sensor de huella.	51
Figura 4.4 Instalación de un tomacorriente para el encendido <i>del sistema de marcación</i>	51
Figura 4.5 Elaboración del acrílico de la estructura general del sistema de marcación.	52
Figura 4.6 Instalación del LCD en el <i>Tablero del sistema de marcación</i>	52
Figura 4.7 Montaje de la tapa de la estructura <i>del sistema de marcación</i>	53
Figura 4.8 Funcionamiento del sistema de marcación en la escuela de manejo CEDUCAR.....	53

INTRODUCCION

Hoy en día es muy común en las empresas tener un registro del ingreso del personal para poder determinar y controlar las horas de las funciones dentro de la empresa, así como también tener un registro de los atrasos de los colaboradores para poder tomar las acciones pertinentes y así como también registro de horas extras para acogerse a los beneficios de ley.

La arquitectura para llevar a cabo este sistema de registro tiene un nivel significativo de robustez ya que la cantidad de empleados es considerablemente numerosa en donde siempre debe existir la posibilidad de agregar un nuevo registro de un empleado nuevo.

Existen en el mercado otras formas de registro de personal o marcación de ingresos como por ejemplo las tarjetas de acceso, pero este método no es muy confiable ya que el usuario la puede extraviar o bien en descuido podría incluso ser clonada, permitiendo de esta manera el ingreso de personas ajenas a la institución en zonas de acceso restringido para cometer delitos dentro de las instalaciones de la institución.

El objetivo de este trabajo de titulación es mejorar el proceso de administración en el control al personal de la escuela de conducción, con el diseño del sistema de marcación con control dactilar utilizando hardware de bajo costo a través de una aplicación web, utilizando la tarjeta Asus Tinker Board en donde se almacenará los datos del sistema de marcación de manera centralizada.

El funcionamiento de la plataforma se basa en la gestión remota y biométrico, con la Asus Tinker Board de dispositivo central de almacenamiento de datos a través de la programación MongoDB que es el lenguaje de la base de datos del sistema, para la programación de la interfaz del biométrico se utilizó los siguientes softwares: VueJS, NodeJS, MongoDB Compass y Python, y dispositivos electrónicos para realizar un mayor control de seguridad y acceso dentro de la institución, desde cualquier tipo de dispositivo electrónico móvil u ordenador, el administrador podrá visualizar los datos de los registros de marcación de forma remota o local, ya que cuenta con aplicación web.

En el Capítulo 1 se detalla los hechos preliminares tales como el planteamiento del problema, objetivos, metodologías, población y muestra, entre otros

En el Capítulo 2 se describe el marco teórico donde se detallan los conceptos generales y características de todos los elementos que se utilizaron en el desarrollo del proyecto de marcación.

En el Capítulo 3 se prepara el estudio y elaboración del proyecto, explicando paso a paso el desarrollo de las interfaces desarrolladas con el software MongoDB.

En el Capítulo 4 se explica el análisis y explicación de los resultados que se obtuvieron en la implementación del proyecto, las conclusiones y recomendaciones.

CAPITULO I

1. El Problema

1.1. Planteamiento del problema.

En la actualidad el avance de la biometría se ha usado para diferentes y distintos fines, entre las principales y más usadas en para la seguridad de las entidades, que por sus labores y trabajos diarios se muestran considerables y altos flujos en el proceso de ingreso y salida del personal.

El sistema a emplear es una biometría estática, la cual por su efectividad tiene mayor aceptación, ya que por más exacta que sea la huella dactilar solo por el hecho de ser única presenta un menor rango de error.

Actualmente en la escuela de manejo “CEDUCAR” no cuenta con un sistema biométrico avanzado, por lo que se vio la necesidad de implementar este sistema actualizado en la institución, en el cual los delegados no sientan inseguridad de su registro de marcación ingresado al portal de la institución.

La escuela contará con un sistema de marcación avanzado que le permitirá verificar su ingreso por medio de huellas dactilares para comprobar si ingreso correctamente su marcación, los datos se almacenan en una base datos la cual estará alojada en un servidor.

1.2. Importancia y alcances

La implementación del presente trabajo de titulación es conveniente y dinámico, porque busca crear una plataforma (software y hardware) la cual sirve gestionar el control de horarios de acceso a la escuela de manejo “Ceducar”, así como tener una base de datos un registro de acceso con reportes bien elaborados y accesibles para los delegados de la institución.

Este proyecto de titulación fue pensado como un prototipo de suma importancia para el usuario, mediante la implementación se brindará un buen registro de las horas laboradas para mejorar el ambiente laboral, siendo sus principales beneficiarios empleados, gerentes y alumnos de la escuela de la institución

La funcionalidad del sistema propuesto se implementará en las Instalaciones de la escuela de manejo “CEDUCAR”, donde se pondrá en operatividad desde cualquier dispositivo inteligente conectado a una red de internet.

1.3. Delimitación del problema.

La delimitación del problema se enfoca en términos concretos en el área en que se presenta el proyecto de titulación, donde la situación en que se expresa va desde una difícil solución a una realidad concreta y fácil de manejar, como se la subdivide a continuación.

1.3.1. Delimitación espacial.

El desarrollo del proyecto se llevará a cabo dentro del departamento de Dirección Administrativa en las instalaciones de la Escuela de Manejo “CEDUCAR” ubicada en la Ciudadela Abel Gilbert Av. Principal frente Escuela CEDI Duran, Guayas. (Véase Figura 1.3.1).

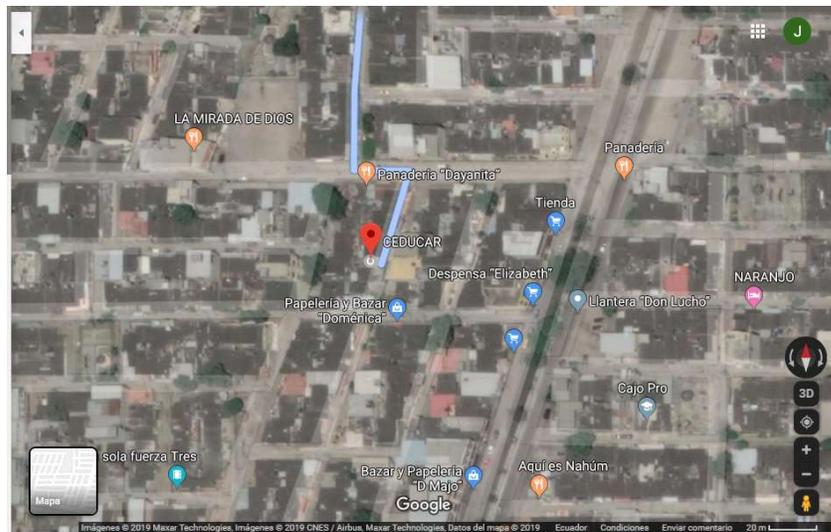


Figura 1.3.1. Ubicación de la Escuela de Manejo 'Ceducar', vista satelital.
Fuente: Véase (Google, 2017)
Elaborado: Los autores.

1.3.2. Delimitación temporal.

El proceso de desarrollo del proyecto tendrá un plazo normal de 12 meses a partir de la aprobación.

En este proyecto de titulación, se pone en práctica todos los conocimientos adquiridos durante el proceso universitario como Ingeniero Electrónico en la

Universidad Politécnica Salesiana, Sede Guayaquil, marcando énfasis en los temas relacionados con redes, electrónica digital, programación, entre otros; asesorándose en la bibliografía en: libros, artículos académicos, revistas científicas, repositorio de tesis y páginas web con información certificada.

1.3.3. Delimitación social.

La presente propuesta busca entregar a la escuela de manejo “CEDUCAR” un prototipo de sistema de marcación con el fin de que el personal pueda acceder al registro mediante una dirección IP previamente programada en el dispositivo con el fin de verificar que haya sido registrado correctamente al sistema de marcación de entrada según su horario establecido.

1.4. Justificación.

Actualmente, es difícil contar con un sistema de control de acceso de tecnología avanzada. Se propone realizar el diseño e implementación del sistema de marcación el cual informará al personal de la institución automáticamente en tiempo real la marcación realizada mediante un módulo GSM que le enviará una notificación de su marcación al momento de realizar su marcación en el biométrico.

Con este proyecto el usuario podrá despejar sus dudas al momento de haber ingresado su hora de marcación ya que nos enfocaremos en resolver esta problemática a través de la implementación de una aplicación web la que nos facilitará visualizar mediante una dirección IP dinámica su registro de asistencia una vez que haya marcado su huella en el biométrico.

Aportará al desarrollo de instrumentos virtuales innovadores en la Escuela de Conducción CEDUCAR, mejorará la calidad del servicio de marcación, dispondrá de instrumentos tecnológicos que serán de gran ayuda al personal docente y administrativo del plantel, fortaleciendo el área de sistemas, mediante la mejora de los instrumentos de bajo costo y fácil uso que se empleara en los equipos que disponen el plantel para hacer su respectiva actualización y que se sientan seguros con la nueva implementación instalada en la institución.

1.5. Grupo objetivo.

A continuación, se describe la población y muestra en donde se valida el sistema desarrollado.

1.5.1. Población.

La población son todas las personas que se encuentran laborando dentro de las instalaciones de la escuela de manejo "CEDUCAR", dentro del departamento de información.

1.5.2. Muestra.

Las muestras son los distintos ingresos de código de huellas dactilares de cada trabajador, en donde una persona asignada es la encargada de manipular el sistema para el control de ingresos y salidas del personal que se encuentra laborando dentro de las instalaciones de la escuela de manejo a través de un aplicativo para administrador donde tendrá el todo lo necesario para el control de reportes diarios.

1.6. Objetivos

1.6.1. Objetivo general

Diseñar e implementar un prototipo de marcación con control dactilar utilizando hardware de bajo costo y visualizado a través de una aplicación web.

1.6.2. Objetivos específicos

- Diseñar el hardware que se acoplará al sensor biométrico mediante componentes pre-fabricados.
- Diseñar la aplicación con la cual se podrá gestionar varias funcionalidades según el usuario.
- Implementar el prototipo de sistema de marcaciones en la escuela de conducción CEDUCAR
- Generar reportes del sistema de marcaciones coloridos para que sean amigables para el usuario.
- Realizar las pruebas del dispositivo para su respectivo funcionamiento con el sistema de marcaciones.

1.7. Metodología

1.7.1. Métodos

La metodología es el procedimiento y la técnica usada para resolver un problema, para el desarrollo del proyecto de titulación se realizó el uso de varios métodos apropiados al tema planteado, los cuales se detallan a continuación:

1.7.2. Método deductivo

El método deductivo consiste en hacer uso de todos los conocimientos adquiridos en el trayecto de nuestra etapa universitaria.

Se hace uso de esta metodología durante el diseño e implementación de la parte electrónica del control de acceso del prototipo de marcación con huella dactilar, ya que se necesitarán de los conocimientos tales como:

- Programación de microcontroladores
- Electrónica digital
- Redes de computadoras.
- Método científico

Este método se encuentra sustentado por dos pilares fundamentales, los cuales son:

La reproducibilidad. - se refiere al experimento que se puede repetir por cualquier persona bajo ciertas condiciones predefinidas y que el resultado de igual.

La refutabilidad. – implica que cualquier resultado logrado puede ser discutido a través de otros experimentos, lo que involucra que los resultados no son incondicionales.

Se necesitó de esta metodología durante el proceso de desarrollo de la interfaz del administrador y usuarios ya que debemos adquirir nuevos conocimientos tales como:

- Diseño de aplicaciones web
- Diseño y administración de base de datos
- Diseño de interfaz de servidores web.

1.7.3. Método experimental

Este método radica en la realización de pruebas constantes a una hipótesis con el propósito de corroborar o refutar. Estas pruebas se producen tanto en medios controlados, como en ambientes de producción donde hay otros elementos externos que pueden variar según los resultados obtenidos.

Esta metodología se utilizó durante las programaciones y la forma del uso adecuado del equipo de marcación que se realizara a lo largo de la elaboración del proyecto técnico y está enfocado en:

- Programación en Asus Tinker Board
- Programación en sensor dactilar.

1.7.4. Técnicas

Las técnicas son el conjunto de acciones usadas para lograr un objetivo, las técnicas que se utilizaron en nuestro proyecto fueron:

1.7.5. Técnica de investigación

Consiste en observar un hecho atentamente para tomar y registrar para su posterior análisis. La técnica de investigación es indispensable en el desarrollo de la investigación científica ya que compone la estructura a través de la cual se organiza la exploración.

1.7.6. Observación

Es una técnica que consiste en observar de manera atenta un hecho o fenómeno para luego registrar la información que posterior análisis.

1.7.7. La entrevista

Es una técnica de recolección de datos a tras de una conversación profesional, con la que además se adquiere información sobre el tema investigado, posee importancia desde el punto de vista educativo; los resultados se obtienen en la misión dependerán de la cantidad de comunicación entre el investigador y el participante.

1.7.8. Recolección de información

Para el presente trabajo de investigación se recolecto la información de las fuentes primarias y secundarias, todos esto se podrá encontrar en la sección de proyectos de investigación vinculados y referentes bibliográficos

1.8. Descripción de la propuesta

Diseñar e implementar una plataforma de control de acceso biométrico y gestión remota que pueda gestionar las entradas y salidas del personal de la institución, donde se instaló un sensor biométrico, el cual lee la información del usuario para enviarla a la Asus Tinker Board mediante el convertidor de comunicación serial a USB.

La Asus Tinker Board recibe los datos con la información de la huella y mediante un script programado en Python, consulta la base de datos MongoDB para determinar si el usuario puede o no registrar su acceso.

Si el usuario puede registrar su acceso, se realiza a guardar los datos del usuario en la base de datos y enviar un correo electrónico informando que se realizó una marcación exitosa, en el correo se detalle la hora y fecha que se efectuó el registro.

La gestión de estos usuarios la realiza el administrador del sistema desde un navegador mediante una aplicación web escrita en VueJS, esta aplicación permite ver, crear, editar y eliminar los usuarios, así como ver el historial de accesos.

1.9. Beneficiarios

Los beneficiarios directos de la interfaz del sistema de control de acceso son las personas que laboran dentro de las instalaciones de la escuela de manejo, el dueño y la persona que estén encargadas de la institución.

1.10. Impacto

Al realizar la interfaz del sistema de marcación con control dactilar con los distintos dispositivos mostrados y al interconectar y enlazarlos con el miniordenador Asus Tinker Board se incrementó la seguridad de la empresa debido a su tecnología, además de su facilidad de uso para interactuar con los demás usuarios de la institución

Por otro lado, tener una acogida comodidad con el uso de nuevas tecnologías para la contribución y desarrollo de la sociedad, así como su respectivo uso en áreas de seguridad y control.

CAPÍTULO II

2. Marco Teórico

2.1. La biometría en la identificación de personas

La biometría le permite identificar y autenticar a las personas en función de un conjunto de datos que pueden reconocerse y verificarse. Estos datos son únicos para ellos. La autenticación biométrica es el proceso de comparar los datos característicos de una persona con la “plantilla” biométrica de la persona para determinar la similitud. Un modelo de referencia es el primer enlace a una base de datos o elemento portátil. Los datos almacenados se comparan con los datos biométricos de la persona que se autentica. (Castro, 2017).



*Figura 2.1 Huella dactilar
Fuente: Castro (2017)*

La identificación de la autenticación biométrica es para confirmar la identidad del individuo. El propósito es capturar elementos de los datos biométricos de esta persona. Hay fotos de rostros, grabaciones de voz, imágenes de huellas digitales (véase *Figura 2.1*). Estos datos se comparan con los datos biométricos de otras personas almacenados en la base de datos.

2.2. Control de acceso biométrico

En general, el control de acceso biométrico es una tecnología de rápida maduración que se utiliza para vigilar nuestra capacidad de acceder a lugares, nos permite acceder a software y hardware, así como a proteger contra el acceso no autorizado a nuestras cuentas en línea. Es una forma de identificación distinta de lo que es hasta hoy el estándar de la industria: las contraseñas.

En general, el control de acceso biométrico es una tecnología de maduración rápida que supervisa el acceso a ubicaciones, permite el acceso a software y hardware, y protege contra el acceso no autorizado a cuentas en línea. Esta es una forma diferente de identificación de la contraseña estándar actual de la industria.

Sin embargo, los nuevos sistemas, como la autenticación biométrica, se posicionan gradualmente como protección de próxima generación. Los sistemas biométricos analizan las características físicas para garantizar que las personas que utilizan diferentes sistemas y funciones estén autorizadas. Los ejemplos de control de acceso biométrico incluyen escaneo de huellas dactilares, escaneo de iris, reconocimiento de voz e incluso características tales como métodos de caminar y gestos.

2.3. Sensor biométrico

Un sensor biométrico es un transductor que convierte un proceso biométrico humano en una señal eléctrica. La autenticación biométrica incluye principalmente lectores biométricos de huellas digitales, iris, rostros y voces. En general, los sensores leen o miden luz, temperatura, velocidad, capacitancia y otros tipos de energía. Puede usar una combinación sofisticada de redes para aplicar esta tecnología para capturar esta conversación. De sensores y cámaras digitales. Cada dispositivo biométrico requiere un tipo de sensor (Portilla, 2015).

Las aplicaciones de autenticación biométrica se utilizan principalmente con cámaras de alta resolución para reconocimiento facial y micrófonos para captura de voz. Algunos datos biométricos están diseñados específicamente para escanear el patrón de las venas debajo de la piel. Los sensores biométricos son una característica importante de la tecnología de identificación. Hay dos categorías de tecnología biométrica:

Medidas fisiológicas. - Puede ser morfológico o biológico. Consisten principalmente en huellas dactilares, formas de manos, patrones de dedos, venas, ojos (iris y retina) y formas de cara para análisis morfológicos. Los equipos médicos y la policía forense pueden usar ADN, sangre, saliva u orina para análisis biológicos.

Medidas de comportamiento. - Los más comunes son el reconocimiento de voz, la dinámica de la firma (velocidad de movimiento del lápiz, aceleración, presión

aplicada, inclinación), dinámica de pulsación de teclas, uso de objetos, caminar, sonidos de pasos, gestos, etc.

2.3.1. Reconocimiento de huellas dactilares

El reconocimiento de huellas digitales implica tomar una fotografía de la huella digital de una persona y registrar sus características como arcos, remolinos, bucles, bordes, puntos de características y contornos de surco. La coincidencia de huellas digitales se puede lograr de tres maneras: puntos de características, correlaciones y crestas (Calle, 2016).

- La coincidencia de huellas digitales basada en minucias guarda un plano que contiene una serie de puntos, que corresponden a la plantilla y los detalles de *i / p*.
- La correlación de huellas digitales basada en la correlación superpone dos imágenes de la huella digital y calcula la asociación entre píxeles equivalentes.
- La comparación de huellas digitales basada en crestas es una forma innovadora de capturar crestas porque es difícil capturar huellas digitales finas a baja calidad. (véase *Figura 2.2*).



Figura 2.2 Reconocimiento de huellas
Fuente: Calle (2016)

Los métodos actuales emplean sensores ópticos que utilizan sensores de imagen CMOS o CCD para capturar huellas digitales. Los sensores de estado sólido funcionan de acuerdo con los principios de la tecnología de transductores utilizando sensores térmicos, capacitivos, piezoeléctricos o de campo eléctrico. O, un sensor ultrasónico funciona con ondas ultrasónicas donde el sensor transmite una señal

acústica a través de un transmisor cerca del dedo y captura la señal en el receptor. El escaneo de huellas digitales es muy estable y confiable.

2.4. Patrones de huellas dactilares

Los tres patrones básicos de las crestas de huellas digitales son arcos, bucles y espirales. Un arco es un patrón donde el borde entra por un lado del dedo, se eleva en el medio para formar un arco y sale del otro lado del dedo. En el caso de un bucle, la cresta entra por un lado del dedo, luego forma una curva y sale por el mismo lado del dedo que ingresó (Suarez, 2016).

El bucle es el patrón más común para las huellas digitales. Finalmente, la hélice es un patrón cuando la cresta se forma en un círculo alrededor del punto central. Mire las imágenes a la derecha para ver ejemplos de cada patrón. Un punto de característica se refiere a un punto específico en una huella digital. Estos son detalles de huellas digitales que son más importantes para el reconocimiento de huellas digitales.

Hay tres tipos principales de características minuciosas:

- El final de la cresta es el punto donde termina la cresta, como su nombre indica.
- La horquilla es donde la cresta se divide en dos crestas.
- Los puntos son crestas de huellas digitales que son significativamente más cortas que otras crestas.

Existen cuatro tipos principales de hardware de lector de huella digital:

2.5. Tipos de Hardware lectores de huella dactilar

2.5.1. Lectores ópticos

Los lectores ópticos son el tipo más común de lector de huellas digitales. El tipo de sensor del lector óptico es una cámara digital que adquiere una imagen visual de la huella digital. La ventaja es que el lector óptico comienza a un precio muy barato. La desventaja es que la lectura se ve afectada por dedos sucios o marcados, y este tipo de lector de huellas digitales se engaña más fácilmente que otros (Tolosa & Giz, 2016) (véase *Figura 2.3*).



Figura 2.3 Lectores ópticos
Fuente: Tolosa y Giz (2013)

2.5.2. Lectores capacitivos

Los lectores capacitivos, también llamados lectores CMOS, no leen las huellas digitales con luz. En cambio, el lector CMOS usa un condensador y, por lo tanto, usa una corriente para formar una imagen de huella digital. Los lectores CMOS son más caros que los lectores ópticos, pero siguen siendo relativamente baratos a precios muy inferiores a 100 euros. Una ventaja importante de los lectores capacitivos sobre los lectores ópticos es que requieren huellas digitales reales en lugar de solo imágenes visuales. Esto hace que sea más difícil engañar a un lector CMOS. (Sanchez, 2015) (véase Figura 2.4).



Figura 2.4 Lectores capacitivos
Fuente: Sánchez (2015)

2.5.3. Lectores de ultrasonido

El lector ultrasónico es el último lector de huellas digitales que utiliza ondas sonoras de alta frecuencia para penetrar la capa epidermis (externa) de la piel. Lee la huella digital de la capa de piel de la piel. Esto elimina la necesidad de superficies limpias y sin arañazos. Todos los demás tipos de lectores de huellas digitales obtienen una imagen de la superficie exterior, por lo que debe limpiarse las manos y evitar arañazos antes de lavarse las manos. Este tipo de lector de huellas digitales es mucho más costoso que los dos primeros, pero debido a su precisión y al hecho

de que es difícil de engañar, los lectores de ultrasonido ya son muy populares (Hernandez, 2015) (véase *Figura 2.5*).



Figura 2.5 Lectores de ultrasonido
Fuente: Sánchez (2015)

2.5.4. Lectores térmicos

El lector térmico detecta la diferencia de temperatura entre las crestas de las huellas digitales y los valles en la superficie de contacto. Los lectores térmicos de huellas digitales tienen muchos inconvenientes, como el consumo de energía y el rendimiento, dependiendo de la temperatura ambiente (Cueva, 2015) (véase *Figura 2.6*)



Figura 2.6 Lectores térmicos
Fuente: Cueva (2015)

2.6. Sensor de huella dy50

Módulo de sensor óptico de huellas dactilares / biométrico: FPM10A DY50 con interfaz TTL / UART con soporte de firmware completo y un software de captura de huellas dactilares de Windows. (Li, 2016) (véase *Figura 2.7*)

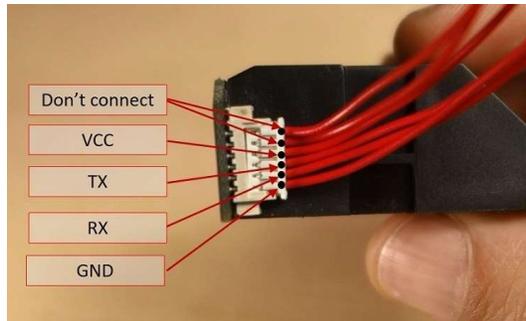


Figura 2.7 Sensor de huella DYS50
Fuente: (PROMETEC, 2019).

Tabla 1
Especificaciones del sensor óptico DY50.

Detalles técnicos del sensor óptico DY50.	
Resolución	500. pixeles por pulgada
Área del sensor(mm)	14.0 x18.0
Tamaño de imagen (pixel)	Crea e implementa instantáneas con el clic de un mouse.
Protocolo de comunicación	UART
Voltaje de alimentación	3.6V DC - 6V DC.
Interfaz	TTL

Fuente: Los autores

2.7. Servidores de bases de datos

Un servidor de base de datos es similar a un almacén de datos donde un sitio web almacena o mantiene sus datos e información. El servidor de bases de datos es una computadora en la LAN dedicada al almacenamiento y recuperación de bases de datos. El servidor de bases de datos contiene un sistema de gestión de bases de datos y una base de datos. Cuando se recibe una solicitud de una máquina cliente, el registro seleccionado se busca en la base de datos y se devuelve a la red (Gonzales & Cervantes, 2016).

Los servidores de bases de datos son útiles para las organizaciones que necesitan procesar una cierta cantidad de datos de manera regular o regular. Si tiene una arquitectura cliente-servidor donde los clientes necesitan procesar datos con

frecuencia, recomendamos utilizar un servidor de base de datos. Algunas organizaciones usan servidores de archivos para almacenar y procesar datos. Los servidores de bases de datos son mucho más eficientes que los servidores de archivos.

Todas las funciones de la base de datos son controladas por el servidor de la base de datos. Cualquier tipo de computadora se puede utilizar como servidor de base de datos. Puede ser una microcomputadora, una minicomputadora o una computadora central. En las redes de grandes organizaciones, se utiliza una computadora central como servidor. Actualmente, existen varios sistemas de gestión de bases de datos, que incluyen:

2.8. Base de datos Oracle

Una base de datos Oracle es una colección de datos que se trata como una unidad. El propósito de la base de datos es almacenar y recuperar información relacionada. El servidor de bases de datos es la clave para resolver los problemas de gestión de la información. En general, el servidor administra de manera confiable grandes cantidades de datos en un entorno multiusuario y permite que muchos usuarios accedan a los mismos datos al mismo tiempo. Todo esto se logra con un alto rendimiento. Los servidores de bases de datos también evitan el acceso no autorizado y proporcionan una solución eficiente para la recuperación ante desastres. (Oracle, 2016) (véase *Figura 2.8*)



Figura 2.8 Servidor de base de datos oracle
Fuente: Oracle (2016)

Oracle Database es la primera base de datos diseñada para la computación grid en una empresa, la forma más flexible y rentable de administrar información y

aplicaciones. La computación grid de la empresa influye en la creación de grandes grupos de servidores y almacenamiento modular estándar de la industria. Con esta arquitectura, cada nuevo sistema se puede aprovisionar rápidamente desde el conjunto de componentes. No hay necesidad de cargas de trabajo máximas, ya que la capacidad se puede agregar o reasignar fácilmente desde los grupos de recursos según sea necesario.

2.8.1. Arquitectura general de aplicación de oracle

Hay dos formas comunes de diseñar una base de datos: cliente / servidor o multinivel. A medida que el entorno de Internet se utiliza con mayor frecuencia en entornos informáticos, muchos sistemas de gestión de bases de datos se están trasladando a entornos de varios niveles.

El procesamiento distribuido reduce la carga en un único procesador y mejora el rendimiento general del sistema y la funcionalidad al permitir que diferentes procesadores se centren en un subconjunto de tareas relacionadas.

Los sistemas de bases de datos Oracle pueden aprovechar fácilmente el procesamiento distribuido utilizando una arquitectura cliente / servidor. En esta arquitectura, el sistema de base de datos se divide en dos partes. Front-end o cliente, y back-end o servidor (Ortiz, 2015).

El cliente. - Es una aplicación de base de datos que inicia una solicitud para que se realice una operación en el servidor de la base de datos. Solicita, procesa y presenta datos gestionados por el servidor. La estación de trabajo cliente se puede optimizar para su trabajo. Por ejemplo, es posible que no necesite una gran capacidad de disco o que se beneficie de las capacidades gráficas. A menudo, el cliente se ejecuta en una computadora diferente a la del servidor de base de datos, generalmente en una PC. Muchos clientes pueden ejecutar simultáneamente contra un servidor.

El servidor. - Es aquel que se ejecuta el software de Oracle y maneja las funciones requeridas para el acceso simultáneo de datos compartidos. El servidor recibe y procesa las sentencias de SQL y PL / SQL que se originan de las aplicaciones cliente. La computadora que administra el servidor puede ser optimizada para sus funciones. Por ejemplo, puede tener gran capacidad de disco y procesadores rápidos.

2.9. Servidor de Vultr

Vultr proporciona servidores de SSD en la nube de alto rendimiento que cuentan con una huella global. Su objetivo es crear el servicio más confiable y eficiente para el lanzamiento de un servidor en la nube SSD de gran velocidad, ofreciendo unidades de estado sólido (SSD) al 100% utilizando las CPU Intel de última generación en un panel de control con numerosas funciones (Comprarhosting, 2019).

Al igual que muchos de sus competidores, como Digital Ocean, Amazon Web Services, Cloudways y Microsoft Azure, Vultr es un proveedor de servidores de SSD en la nube de alto rendimiento que cuenta con una presencia global.

Sin embargo, lo que distingue a Vultr es su enfoque en las opciones flexibles a un nivel de precios más bajo. Estas opciones son atractivas incluso para usuarios de pequeña escala, como empresarios individuales y pequeñas empresas.

El panel de control es un aspecto clave de cualquier servicio en la nube, y el panel de control de Vultr está diseñado pensando en el cliente. Muchas características se pueden instalar con un solo clic. Este tiempo ahorrado permite al desarrollador concentrarse en temas más importantes. El panel de control de la nube de Vultr tiene características útiles que pueden beneficiarlo de varias maneras:

Tabla 2
Beneficios y características del servidor Vultr.

Característica	Beneficio
Implementación instantánea del servidor	Implementar el servidor sin largos procesos con un simple clic, a través de una interfaz intuitiva, ya sea desde el escritorio o el móvil.
Implementación instantánea del servidor	Recibirá análisis instantáneos de la salud general de sus servidores, con datos como el uso de la Banda Ancha y la CPU, o los fondos actuales gastados.
Interfaces fáciles	Diseñar e implementar instantáneamente con solo un clic de un mouse.

2.10. Mongo DB

MongoDB es una herramienta de gestión de base de datos basada en documentos de código abierto que almacena datos en formatos similares a JSON. Es una base de datos NOSQL altamente escalable, flexible y distribuida.

MongoDB es una base de datos de código abierto que utiliza un modelo de datos orientado a documentos y un lenguaje de consulta no estructurado. Es uno de los sistemas y bases de datos NoSQL más poderosos de la actualidad. Ser una herramienta NoSQL significa que no utiliza las filas y columnas habituales que tanto asociamos con la gestión de la base de datos relacional. Es una arquitectura que se construye sobre colecciones y documentos. La unidad básica de datos en esta base de datos consiste en un conjunto de pares clave-valor

Permite que los documentos tengan diferentes campos y estructuras. Esta base de datos utiliza un formato de almacenamiento de documentos llamado BSON, que es un estilo binario de los documentos de estilo JSON.

El modelo de datos que sigue MongoDB es muy flexible y le permite combinar y almacenar datos de tipos multivariados sin tener que comprometer las poderosas opciones de indexación, el acceso a los datos y las reglas de validación. No hay tiempo de inactividad cuando desea modificar dinámicamente los esquemas. Entonces, lo que significa que puede concentrarse más en hacer que sus datos trabajen más duro en lugar de dedicar más tiempo a preparar los datos para la base de datos.

La arquitectura de la base de datos NoSQL de MongoDB se conforma de la siguiente manera:

La base de datos: en palabras simples, se puede llamar como el contenedor físico para los datos. Cada una de las bases de datos tiene su propio conjunto de archivos en el sistema de archivos con varias bases de datos existentes en un único servidor MongoDB.

La Colección: Un grupo de documentos de base de datos puede ser llamado como una colección. El equivalente RDBMS de la colección es una tabla. Toda la colección existe dentro de una sola base de datos. No hay esquemas cuando se trata de colecciones. Dentro de la colección, los diversos documentos pueden tener campos

variados, pero la mayoría de los documentos dentro de una colección están diseñados para el mismo propósito o con el mismo objetivo final.

El documento: Un conjunto de pares clave-valor puede ser designado como un documento. Los documentos están asociados con esquemas dinámicos. La ventaja de tener esquemas dinámicos es que el documento en una sola colección no tiene que tener la misma estructura o campos. También los campos comunes en el documento de una colección pueden tener diversos tipos de datos.

2.10.1. Ventajas de MongoDB

MongoDB es una de las numerosas tecnologías de bases de datos no relacionales que surgieron a mediados de la década de 2000 bajo el banner NoSQL para su uso en aplicaciones de big data y otros trabajos de procesamiento que involucran datos que no encajan bien en un modelo relacional rígido. Sus principales ventajas son:

Plataforma de datos distribuidos: a lo largo de los centros de datos distribuidos geográficamente y las regiones de la nube, MongoDB puede ejecutarse garantizando nuevos niveles de disponibilidad y escalabilidad. Sin tiempo de inactividad y sin cambiar su aplicación, MongoDB se amplía elásticamente en términos de volumen de datos y rendimiento. La tecnología le brinda suficiente flexibilidad en varios centros de datos con buena consistencia.

Desarrollo rápido e iterativo: los requisitos comerciales cambiantes ya no afectarán la entrega exitosa de proyectos en su empresa. Un modelo de datos flexible con esquema dinámico, con poderosas GUI y herramientas de línea de comandos hacen que los desarrolladores puedan desarrollar y desarrollar aplicaciones rápidamente. El aprovisionamiento automatizado permite la integración y entrega continua para operaciones productivas. Los esquemas relacionales estáticos y las operaciones complejas de RDBMS son ahora cosa del pasado.

Modelo de datos flexible: MongoDB almacena los datos en documentos similares a JSON flexibles, lo que hace que la persistencia y la combinación de los datos sean fáciles. Los objetos en el código de su aplicación se asignan al modelo del documento, por lo que el trabajo con datos se ha vuelto fácil. No hace falta decir que los controles de gobernanza del esquema, el acceso a los datos, las agregaciones complejas y la funcionalidad de indexación enriquecida no se ven comprometidos de ninguna manera. Sin tiempo de inactividad se puede modificar el esquema

dinámicamente. Debido a esta flexibilidad, un desarrollador debe preocuparse menos por la manipulación de datos y más sobre el uso de esos datos.

Reducción de costos: los desarrolladores de aplicaciones pueden hacer su trabajo mejor cuando se usa MongoDB. El equipo de operaciones también puede realizar su trabajo bien gracias al servicio en la nube Atlas. Los costos se reducen significativamente a medida que MongoDB se ejecuta en hardware básico. La tecnología ofrece precios a la medida, de pago por uso con suscripciones anuales junto con soporte global 24/7.

2.11. Software Python

Este programa no es más que un lenguaje de programación, el mismo que se basa en interpretar un código legible por medio de la sintaxis, el código del programador es interpretado por medio de Python o algún otro interprete, este posee ventajas, tal como se expresa a continuación:

Por ser un programa que interpreta, este no necesita compilar, permitiendo ahorrar tanto tiempo como la prueba del mismo.

El código fuente puede ejecutarse en cualquier software, siempre que dicho software tenga un intérprete como Windows, Linux, Mac, Android, web.

Este tipo de software interpretativo se diseñó hacia objetos con el fin de obtener resultados, esto permitirá agrupar la información en las bibliotecas de información, tan igual como ocurre con la creación de nuevas librerías. (Van Rossum & Drake, 2011) (véase *Figura 2.9*).



Figura 2.9 Logo del software python
Fuente:emanuelpeg.blogspot.com

2.11.1. Python-flask

Flask es un framework web, lo cual significa que proporciona herramientas, bibliotecas y tecnologías que le permiten crear una aplicación web. Esta aplicación web puede ser algunas páginas web, un blog, un wiki o ir tan grande como una aplicación de calendario basada en web o un sitio web comercial. Este framework es igual de bueno para aplicaciones más grandes como para aplicaciones más pequeñas. (Cuervo, lineadecodigo.com, 2016)

Forma parte de las categorías del micro-framework son normalmente flask con pocas o ninguna dependencia de bibliotecas externas. Esto presenta pros y sus contras. Las ventajas se orientan a que el flask es ligero, existiendo poca dependencia para actualizar y vigilar los errores de seguridad, sin embargo, la desventaja radica en lo contrario es que en algún momento tendrá que aumentar la lista de dependencias agregando complementos. Su estructura y funcionalidad radica en:

Flask tiene que ver con el minimalismo y la simplicidad. No hay restricciones y puedes implementar lo que quieras.

El motor de plantillas funciona en gran medida para Flask, proporcionando una funcionalidad fácil de usar.

Flask permite usar diversos componentes, y ninguno de los frameworks anteriores permitía la modificación de sus módulos en tal medida.

Es ideal para crear sitios simples con contenido estático, como blogs. Proporciona toda la funcionalidad que necesita y permite su personalización en gran medida.

2.11.2. Pymongo

Pymongo es una distribución de Python que contiene herramientas para trabajar con MongoDB y es la forma recomendada de trabajar con MongoDB desde Python.

En la herramienta denominada Pymongo se utilizan diccionarios para representar documentos. Los documentos pueden contener tipos de Python nativos (como `datetime.datetime`) que se convertirán automáticamente desde y hacia los tipos de BSON apropiados. (Cuervo, 2016)

En el núcleo de Pymongo se encuentra el objeto MongoClient, que se utiliza para realizar conexiones y consultas a un grupo de bases de datos MongoDB. Se puede utilizar para conectarse a una instancia de MongoDB independiente, a un conjunto de réplicas o a instancias de mongos. Para su utilización se debe mantener a consideración lo siguiente:

Establecer una conexión. - Para establecer una conexión se debe usar el MongoClientobjeto. Lo primero que se debe hacer para establecer una conexión es importar la MongoClientclase. Utilizado para la comunicación con la instancia de la base de datos en ejecución.

Acceso a las bases de datos. - Una vez establecida la conexión con MongoClient, puede acceder a cualquiera de las bases de datos dentro de ese servidor Mongo. Para especificar qué base de datos realmente se necesita utilizar, debe acceder a ella como un atributo.

Insertando Documentos. - Almacenar datos en la base de datos es de fácil realización con la ejecución de scripts. La primera línea especifica qué colección utilizará (postsen el siguiente ejemplo). En la terminología de MongoDB, una colección es un grupo de documentos que se almacenan juntos dentro de la base de datos. Las colecciones y los documentos son similares a las tablas y filas de SQL, respectivamente.

2.12. Asus Tinker Board

La placa Tinker es una computadora de placa única de tamaño muy pequeño que ofrece un rendimiento inigualable y también se beneficia de la máxima compatibilidad de los componentes. Los creadores de Tinker Board ofrecen entusiastas, fanáticos y cualquier persona que quiera hacer algo por su cuenta, una plataforma que hace realidad sus ideas.

Con un potente y moderno procesador de cuatro núcleos basado en ARM, la tarjeta Tinker RK3288 de Rockchip ofrece un rendimiento dramáticamente mejorado con respecto a otras tarjetas SBC populares.

Cuando notamos la creciente demanda de diferentes estructuras y proyectos, ofrece memoria LPDDR3 de doble canal de hasta 2 GB. La Tinker Board está equipada con una interfaz SDIO 3.0 que brinda velocidades de lectura y escritura

más rápidas que las tarjetas microSD expandibles utilizadas para el sistema operativo, las aplicaciones y el almacenamiento de archivos (ServidorWeb, 2017).



Figura 2.10 Tarjeta asus tinker board para el sistema de marcación.
Fuente: (ServidorWeb, 2017).

Tabla 3
Especificaciones de la tarjeta Asus Tinker Board

Detalles técnicos de la Tarjeta Asus Tinker Board.	
Memoria	2GB doble canal DDR3
LAN	RTL GB LAN
Almacenamiento	Ranura para tarjeta Micro SD (TF).
Protocolo de comunicación	UART
Voltaje de alimentación	3.5V DC - 5V DC.
Interfaz	TTL
Inalámbrico	Bluetooth v4.0+ EDR, 802.11 b/g/n, WIFI con encabezado de antena Wipex

Fuente: Asus (2017)

2.13. Servidor Web

Se denomina como un programa que se realiza especialmente para diseñar o transferir los datos denominados como hipertexto, también conocido como página web que posee todos los elementos que conforman una página web, estos servidores emplean un protocolo denominado http. Los datos que estos manejan se alojan en un ordenador con conexión a internet, recibiendo peticiones de clientes y respondiendo con el envío de ficheros que se soliciten, bien sea de un texto plano denominado html o php y binarios como gif, jpeg (Gudiño de la A & Chavarria Neira, 2017) (véase Figura 2.10).



Figura 2.11 Diseño esquemático del servidor web.
Fuente: (ServidorWeb, 2017).

2.14. Apache HTTP Server

Se denomina como uno de varios tipos de servidores web o protocolo HTTP desarrollado con un formato libre, estos poseen la capacidad de ser instalada y a la vez ejecutada en diferentes sistemas operativos, sin embargo, es muy complejo y su seguridad es la que más destaca, al igual que el gran rendimiento. (véase *Figura 2.11*)

Este tipo de servidor permite almacenar no solo páginas web estáticas, sino también los que se desarrollan en formato HTML, que conforman la web dinámica, con el fin de tener mayor capacidad para agregar módulos empleados con lenguajes PHP (Barbella, 2019).

Dentro de las ventajas que se este servidor posee se tiene las siguientes:

- Se califica como un software de código abierto empleada para la instalación o configuración.
- Es un software de costo gratuito.
- De carácter funcional y de soporte por medio de personas que adicionan mejoras.
- Puede instalarse y ejecutarse en diferentes sistemas operativos.
- Rendimiento alto para las diferentes gestiones superiores a un millón de visitas diarias.
- Tiene un soporte de seguridad SSL y TLS.

APACHE HTTP SERVER



*Figura 2.12 Ilustración del servidor apache.
Fuente:Apermuy.blogspot.com*

2.15. Api Rest

REST es básicamente un estilo arquitectónico de los servicios web que funcionan como un canal de comunicación entre diferentes computadoras o sistemas en Internet. El término API REST es otra cosa.

Las interfaces de programación de aplicaciones respaldadas por el estilo arquitectónico del sistema arquitectónico REST se denominan API REST. Los servicios web, los sistemas de bases de datos y los sistemas informáticos que cumplen con la API REST permiten que los sistemas solicitantes obtengan un acceso robusto y redefinan las representaciones de los recursos basados en la web mediante la implementación de un conjunto predefinido de protocolos sin estado y operaciones estándar (Richards, 2015).

Mediante estos protocolos y operaciones, y la redistribución de los componentes manejables y actualizables sin causar el efecto en el sistema, los sistemas REST API ofrecen un rendimiento rápido, confiabilidad y más progresión.

2.15.1. Comprender el diseño de API REST

Hay seis restricciones clave para el diseño de la API REST que debe tener en cuenta al decidir si este es el tipo de API adecuado para su proyecto. De acuerdo con

2.15.1.1. de cliente

La restricción cliente-servidor funciona sobre el concepto de que el cliente y el servidor deben estar separados entre sí y permitir que evolucionen de forma

individual e independiente. En otras palabras, debería poder hacer cambios en mi aplicación móvil sin afectar la estructura de datos o el diseño de la base de datos en el servidor. Al mismo tiempo, debería poder modificar la base de datos o hacer cambios en la aplicación de mi servidor sin afectar al cliente móvil. Esto crea una separación de preocupaciones, permitiendo que cada aplicación crezca y escale independientemente de la otra y que su organización crezca de manera rápida y eficiente.

2.15.1.2. Apátrida

Las API REST no tienen estado, lo que significa que las llamadas se pueden hacer independientemente una de la otra, y cada llamada contiene todos los datos necesarios para completarse con éxito. Una API REST no debe confiar en los datos que se almacenan en el servidor o las sesiones para determinar qué hacer con una llamada, sino que solo debe confiar en los datos que se proporcionan en esa llamada. La información de identificación no se almacena en el servidor al hacer llamadas. En cambio, cada llamada tiene los datos necesarios en sí mismos, como la clave API, el token de acceso, la identificación de usuario, etc. Esto también ayuda a aumentar la confiabilidad de la API al tener todos los datos necesarios para realizar la llamada, en lugar de depender de una serie de llamadas con estado de servidor para crear un objeto, lo que puede provocar fallas parciales. En cambio, para reducir los requisitos de memoria y mantener su aplicación lo más escalable posible, una API RESTful requiere que cualquier estado se almacene en el cliente, no en el servidor.

2.15.1.3. Cache

Debido a que una API sin estado puede aumentar la sobrecarga de la solicitud al manejar grandes cargas de llamadas entrantes y salientes, una API REST debe diseñarse para alentar el almacenamiento de datos almacenables en caché. Esto significa que cuando los datos se pueden almacenar en caché, la respuesta debe indicar que los datos se pueden almacenar hasta cierto tiempo (caduca a), o en los casos en que los datos deben ser en tiempo real, que la respuesta no debe ser almacenada en caché cliente. Al habilitar esta restricción crítica, no solo reducirá en gran medida el número de interacciones con su API, reduciendo el uso interno del servidor, sino que también proporcionará a sus usuarios de API las herramientas necesarias para proporcionar las aplicaciones más rápidas y eficientes posibles. Tenga en cuenta que el almacenamiento en caché se realiza en el lado del cliente.

Si bien es posible que pueda almacenar en caché algunos datos dentro de su arquitectura para realizar un rendimiento general, la intención es instruir al cliente sobre cómo debe proceder y si el cliente puede o no almacenar los datos temporalmente.

2.15.1.4. Interfaz uniforme

La clave para el cliente de desacoplamiento del servidor es tener una interfaz uniforme que permita la evolución independiente de la aplicación sin tener los servicios, modelos o acciones de la aplicación estrechamente vinculados a la capa API. La interfaz uniforme permite al cliente hablar con el servidor en un solo idioma, independientemente del back-end arquitectónico de cualquiera de ellos. Esta interfaz debe proporcionar un medio estandarizado e inmutable de comunicación entre el cliente y el servidor, como el uso de HTTP con recursos URI, CRUD (Crear, Leer, Actualizar, Eliminar) y JSON.

2.15.1.5. Sistema de capas

Como su nombre lo indica, un sistema en capas es un sistema compuesto por capas, y cada capa tiene una funcionalidad y responsabilidad específicas. Si pensamos en un marco de Model View Controller, cada capa tiene sus propias responsabilidades, y los modelos comprenden cómo se deben formar los datos, el controlador se centra en las acciones entrantes y la vista se centra en la salida. Cada capa está separada pero también interactúa con la otra. En el diseño de REST API, el mismo principio es válido, con diferentes capas de la arquitectura trabajando juntas para construir una jerarquía que ayuda a crear una aplicación más escalable y modular.

2.15.1.6. Código bajo demanda

Quizás la menos conocida de las seis restricciones, y la única restricción opcional, Code on Demand permite que el código o los applets se transmitan a través de la API para su uso dentro de la aplicación. En esencia, crea una aplicación inteligente que ya no depende únicamente de su propia estructura de código. Sin embargo, tal vez porque está adelantado a su tiempo, Code on Demand ha tenido problemas para su adopción, ya que las API web se consumen en varios idiomas y la transmisión del código plantea preguntas y preocupaciones de seguridad. (Por ejemplo, el directorio tendría que ser grabable y los cortafuegos debería dejar pasar lo que normalmente puede ser contenido restringido).

Juntas, estas restricciones conforman la teoría de la transferencia de estado representacional, o REST. Al mirar hacia atrás, puede ver cómo cada restricción sucesiva se suma a la anterior, y finalmente crea una interfaz de programa de aplicación bastante compleja, pero potente y flexible. Pero lo más importante, estas limitaciones conforman un diseño que funciona de manera similar a cómo accedemos a las páginas de nuestros navegadores en la World Wide Web. Crea una API que no está dictada por su arquitectura, sino por las representaciones que devuelve, y una API que, aunque sin estado arquitectónico, se basa en la representación para dictar el estado de la aplicación.

2.16. Nginx server

Se denomina como un servidor web de código abierto, actualmente utilizado como proxy inverso, cache de HTTP y también balanceador de carga, Nginx fue diseñado para ofrecer un uso menor de la memoria y por ende la alta recurrencia, este utiliza un enfoque asíncronico que se basa en los diferentes eventos que manejan un solo hilo (Nedelcu, 2010)..

- Dentro de las características que tienen consigo el Nginx, posee lo siguiente:
- Proxy inverso con caché.
- IPv6.
- Balanceo de carga.
- Soporte FastCGI con almacenamiento en caché.
- Websockets.
- Manejo de archivos estáticos, archivos de índice y auto indexación.
- TLS/SSL con SIN. (véase Figura 2.12).



Figura 2.13 Logo del servidor nginx
Fuente: (Nedelcu, 2010).

2.17. App VueJS

El desarrollo de software de aplicaciones es una de las empresas más populares que se practica tanto a nivel individual como empresarial. Los desarrolladores están utilizando diferentes herramientas y técnicas para lanzar aplicaciones exitosas. Los desarrolladores utilizan muchas tecnologías de software para hacer que las aplicaciones sean más rápidas, más atractivas y fáciles de usar. Vue.js es una de esas nuevas tecnologías de software que se utilizan ampliamente en todo el mundo para el desarrollo web. Vue.js es en realidad un marco de JavaScript con varias herramientas opcionales para construir interfaces de usuario. (Filipova, 2016) (véase *Figura 2.13*)



Welcome to your Vue.js app!

Figura 2.14 Logo del servidor web vue.js
Fuente: <https://vuejs.org/v2/guide/>

Ventajas

Las siguientes son las ventajas de usar la tecnología Vue.js en el desarrollo web.

2.17.1. tamaño muy pequeño

El éxito del framework JavaScript depende de su tamaño. Cuanto más pequeño sea el tamaño, más se usará. Una de las mayores ventajas de Vue.js es su pequeño tamaño. El tamaño de este marco es de 18 a 21 KB y el usuario no necesita tiempo para descargarlo y usarlo. Esto no significa que tenga baja velocidad debido a su pequeño tamaño. En cambio, supera a todos los frameworks voluminosos como React.js, Angular.js y Ember.js.

2.17.2. Fácil de entender y desarrollar aplicaciones

Una de las razones de la popularidad de este marco es que es bastante fácil de entender. El usuario puede agregar fácilmente Vue.js a su proyecto web debido a su estructura simple. Tanto las plantillas de escalas pequeñas como las grandes se

pueden desarrollar a través de este marco que ahorra mucho tiempo. En caso de cualquier problema, el usuario puede rastrear fácilmente los bloques con errores. Todo esto se debe a su estructura simple.

2.17.3. Integración simple

Vue.js también es popular entre los desarrolladores web porque les facilita la integración con las aplicaciones existentes. Esto se debe a que se basa en el marco de JavaScript y se puede integrar en otras aplicaciones creadas en JavaScript. Esto significa que es útil para desarrollar nuevas aplicaciones web, así como para alterar las aplicaciones preexistentes. Esta integración es posible porque Vue.js tiene componentes para todo.

2.17.4. Documentación detallada

A los desarrolladores siempre les gusta usar el marco con documentación detallada porque siempre les resulta fácil escribir su primera aplicación. La documentación con Vue.js es tan completa que cualquier usuario que sepa un poco sobre JavaScript y HTML puede desarrollar su propia aplicación o página web.

2.17.5. Flexibilidad

Una gran flexibilidad es otra ventaja de Vue.js. Permite al usuario escribir su plantilla en un archivo HTML, un archivo JavaScript y un archivo JavaScript puro utilizando nodos virtuales. Esta flexibilidad también facilita la comprensión para los desarrolladores de React.js, Angular.js y cualquier otro marco JavaScript nuevo. Vue.js ha demostrado ser muy beneficioso en el desarrollo de esas aplicaciones simples que se ejecutan directamente desde los navegadores.

2.17.6. Comunicación de dos vías

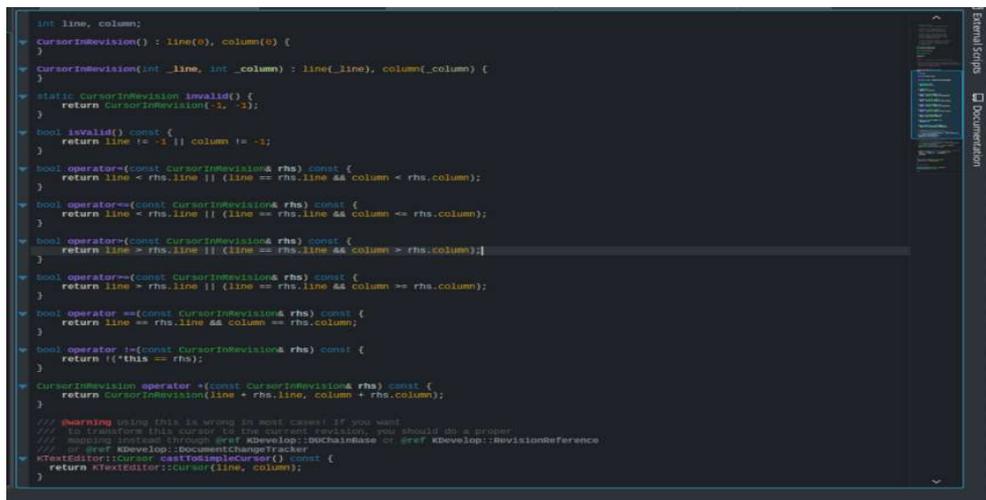
Y, por último, pero no menos importante, Vue.js también facilita las comunicaciones bidireccionales debido a su arquitectura MVVM que hace que sea bastante fácil manejar bloques HTML. A este respecto, parece muy cercano a Angular.js, que también acelera los bloques HTML.

En conclusión, Vue.js tiene claras ventajas sobre todos los marcos anteriores como Angular.js y React.js. En resumen, combina las características peculiares de todos los marcos más antiguos.

2.18. Visual Studio Code

Se conoce como un editor de texto creado por Microsoft, con el fin de satisfacer las diferentes necesidades que los programadores solicitan, sin considerar el lenguaje que empleen, de ese modo se puede utilizar extensiones por medio de Chrome, a través de los archivos y carpetas de forma rápida (Barbella, 2019).

Denominada como una herramienta que forma parte de la función del editor de código, con una fuente ligera pero capaz de conformar un ciclo en el cual se pueda editar, construir y depurar, permitiendo ejecutarse desde el escritorio y por ende de diferentes sistemas operativos. (véase Figura 2.14).



```
int line, column;
CursorInRevision() : line(0), column(0) {
}
CursorInRevision(int _line, int _column) : line(_line), column(_column) {
}
static CursorInRevision invalid() {
    return CursorInRevision(-1, -1);
}
bool isValid() const {
    return line != -1 && column != -1;
}
bool operator<(const CursorInRevision& rhs) const {
    return line < rhs.line || (line == rhs.line && column < rhs.column);
}
bool operator<=(const CursorInRevision& rhs) const {
    return line < rhs.line || (line == rhs.line && column == rhs.column);
}
bool operator>(const CursorInRevision& rhs) const {
    return line > rhs.line || (line == rhs.line && column > rhs.column);
}
bool operator>=(const CursorInRevision& rhs) const {
    return line > rhs.line || (line == rhs.line && column == rhs.column);
}
bool operator==(const CursorInRevision& rhs) const {
    return line == rhs.line && column == rhs.column;
}
bool operator!=(const CursorInRevision& rhs) const {
    return !(*this == rhs);
}
CursorInRevision operator+(const CursorInRevision& rhs) const {
    return CursorInRevision(line + rhs.line, column + rhs.column);
}
// Warning: using this is wrong in most cases! If you want
// to transform this cursor to the other revision, you should do it before
// moving around through pref KDevelop::DBChainedBase or pref KDevelop::RevisionReference
pref KDevelop::DocumentChangeTracker
KTextEditor::Cursor::castToSameRevision() const {
    return KTextEditor::Cursor(line, column);
}
```

Figura 2.15 Captura de pantalla de kdevelop
Fuente: (Lorenzo, 2018)

2.19. Software NodeJS

Este software se caracteriza por ser de código abierto perteneciente a JavaScript, también denominado como un lenguaje de programación, este se diseñó con el fin de desarrollar aplicaciones web con buena optimización, este software se generó para ir avanzando poco a poco y de ese modo escalar para crear más conexiones de manera simultánea con el servidor, dando mayor rendimiento y velocidad de las aplicaciones.

Este servidor se caracteriza en realizar diferentes asignaciones, las mismas que permiten facilitar la comunicación existente entre los diferentes clientes, cuando las tareas se conglomeran se procesan por medio de la entrada (input) o la salida (output) es toda la información por supuesto (jgtecnologias, 2019) (véase Figura 2.15)

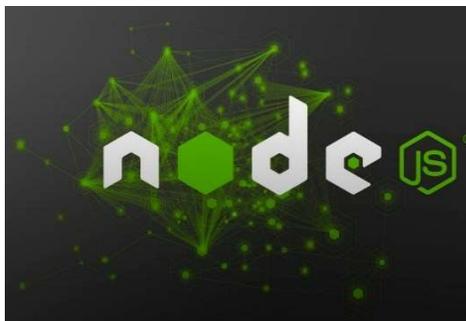


Figura 2.16 Logo del software node js
Fuente: (jgtecnologias, 2019).

2.20. Modulo quemador Arduino Pro Mini

Se conoce como una tarjeta de desarrollo, con una calidad alta que permite convertir los datos de USB a TTL, estos se diseñaron para la realización de proyectos que emplean un puerto serial USB cuando es utilizado en un ordenador y microcontroladores, por lo que se adapta más fácil, con el objetivo de generar comunicación, conexión y por ende compilación de código abierto para el módulo Arduino pro mini, por medio de un puerto USB (Tite, Fabián, Caiza, & Angélica, 2017) (véase *Figura 2.16*).



*Figura 2.17*Modulo convertidor de datos para la asus tinker board
Fuente: naylorlampmechatronics.com

CAPÍTULO III

3. Diseño del prototipo de marcación con control dactilar

3.1. Especificaciones del prototipo de marcación

El prototipo de marcación con control dactilar utilizando hardware de bajo costo para su visualización a través de una aplicación web consta de las siguientes etapas: desarrollo de la aplicación para el sistema de marcación, la comunicación entre los dispositivos, obtención de datos del dispositivo biométrico, la obtención y procesamiento de huellas, obtención de los reportes, envío de mensaje de confirmación por correo.

Todas estas funciones corresponden al desarrollo del sistema de marcación, ya que cada una de ellas es de suma importancia porque todas conforman y una depende de la otra

3.2. Estructura de la red del sistema

El dispositivo central de almacenamiento de datos para el sistema de marcación es la tarjeta Asus Tinker Board debido a su bajo consumo energético, de bajo costo y óptimo rendimiento, el dispositivo se conecta a la red mediante un cable de red o vía inalámbrica por wifi ya que la Asus Tinker Board dispone de ambas opciones, además es estable en la forma funcional en comparación con la raspberry.

La conexión siempre es más segura cableada que por wifi, pero este criterio queda a opción de la elección del administrador de la escuela de manejo. Para poder acceder a la interfaz web del servidor en la nube donde se visualizará las peticiones del api del sistema programadas con el software Python a través de la base de datos MongoDB, el administrador de la escuela de manejo es el único que tendrá acceso a la información de los registros almacenados dentro del sistema. La ip estática que proporciona el servidor en la nube es: "45.77.167.116"

3.3. Análisis de comunicación entre los dispositivos

En el proceso de la comunicación entre los dispositivos utilizados para el sistema de marcación se realiza un análisis teniendo en cuenta la conexión de todos los dispositivos entre sí, para dar paso a la aceptación y emisión de información, los datos son transmitidos en tiempo real, donde los equipos que lo conforman se enlazan entre sí, cada uno de ellos tienen una interdependencia para su tarea de acceso y administración del almacenamiento de información, cada comunicación tiene su propia secuencia dependiendo de las funciones que realicen.

3.4. Conexión de la tarjeta asus tinker board a internet.

La tarjeta Asus Tinker Board es un dispositivo que cuenta con acceso a wifi para conectarse a los proveedores de servicios de internet sea por cable o inalámbricamente. Para conectarse a internet se procede a visualizar a través de la pantalla LCD el cual nos muestra la interfaz de inicio de la tarjeta como cualquier equipo para acceder a internet.

Se accede mediante: Conexión de una red inalámbrica o local > se busca las redes disponibles y se selecciona la red netlife Ceducar, proveedor que brinda internet a la escuela de manejo para poder inicializar en el servidor que nos da los servicios para el api del sistema de marcación el cual se la programa previamente la interfaz remota del sistema de marcación para que pueda mostrar sus respectivas funciones entorno a las marcaciones. (véase Figura 3.1)



Figura 3.1 *Diagrama de conexión de datos de la tarjeta asus tinker board al sistema de marcación del servidor de Ceducar*
Fuente: Los Autores.

3.5. Análisis de la obtención de datos

La obtención de datos se realiza a través de la adquisición del código dactilar a través del sensor biométrico, este código es enviado a la tarjeta Asus Tinker Board, después según la función que se desea realizar, se puede ingresar la clave como código nuevo de usuario a la base de datos o compararlos con los datos ya existentes en la Base de datos.

3.6. Obtención de datos del sensor de huella dactilar

La obtención de datos se realiza a través de la adquisición del código dactilar a través del sensor biométrico. (véase Figura 3.2)

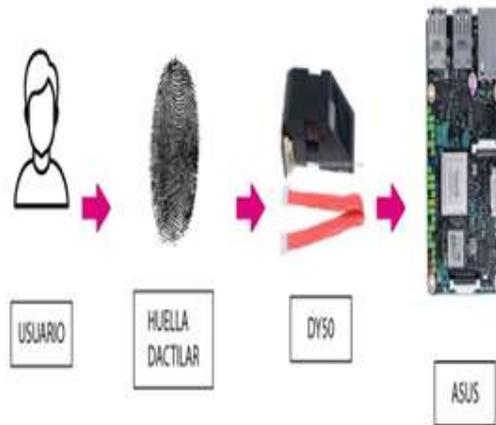


Figura 3.2 Diagrama de obtención del código de huella a través del sensor biométrico.
Fuente: Los Autores.

Este código es enviado a la tarjeta Asus Tinker Board donde posteriormente ese código único es enviado a la tarjeta central que es la Asus del equipo biométrico, cuando llega el dato a la tarjeta lo que hace es un llamado de una petición a la api del servidor instalada mediante post, el equipo sabe que se hará una petición de la opción señalada por el usuario con el código de huella que se envió desde el sensor hacia la Asus Tinker Board, con la petición que se realiza la api empieza a ejecutar los servicios del llamado de la petición de cada usuario en el momento de realizar su marcación.

La api empieza a ejecutar las funciones instaladas internamente en este caso la petición de entrada la api coge el código de huella que llego por los parámetros que se hizo en la petición y lo busca en la base de datos para ver si existe el usuario, si el usuario existe la api procede a realizar la marcación y se registra en la tabla de marcaciones y el de entradas y posteriormente cuando se hizo el registro desde la api mediante el servidor se envía un correo electrónico al usuario notificándole que se realizó correctamente su marcación donde se detalla su registro de ingresos con especificaciones de hora fecha de su registro con esto se evitara dudas al momento de su marcación. (véase Figura 3.3).

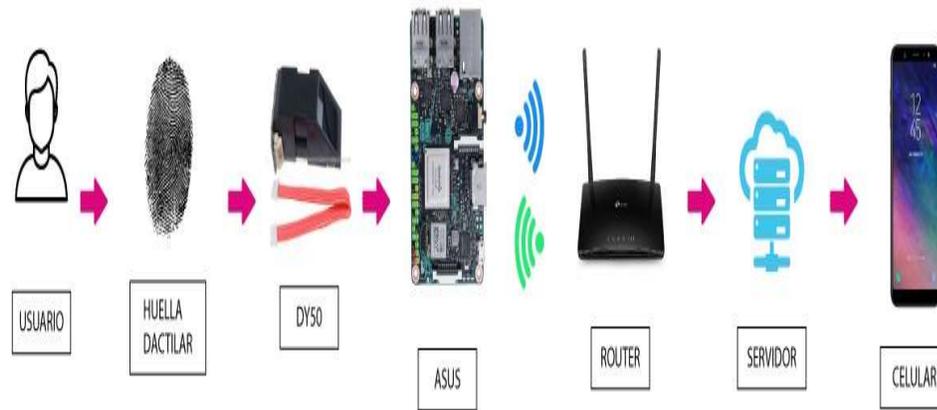


Figura 3.3 Diagrama de obtención de datos del sistema de marcación.
Fuente: Los Autores.

3.7. Base de datos

Se optó por Mongo DB para este sistema ya que en la actualidad es el lenguaje de programación del lado del servidor más accesible y compatible con los demás sistemas operativos y este lenguaje de base de datos esta descrito en Python.

Para acceder a la base de datos del sistema, MongoDB Compass nos pedirá varios campos requeridos tales como: (véase Figura 3.4)

- Hostname: 127.0.0.1
- Port: 27017
- Authentication: Username / Password
- Username: admin
- Password: admin1234
- Authentication Database: admin
- SSH Tunnel: Use Password
- SSH Hostname: 45.77.167.116
- SSH Tunnel Port: 22
- SSH Username: root
- SSH Password: b\$2W%z_R9Sz (tDtU)

Hostname
 Port
 SRV Record

Authentication
 Username
 Password
 Authentication Database

Read Preference

SSL

SSH Tunnel
 SSH Hostname
 SSH Tunnel Port
 SSH Username
 SSH Password

Figura 3.4 Interfaz mongoddb compass para acceder a la base de datos.
Fuente: Los autores.

Al momento de acceder al MongoDB Compass se podrá visualizar las tablas de la base de datos creadas del sistema de marcación Ceducar (véase Figura 3.5)

Collection Name *	Documents	Avg Document Size	Total Document Size	Num. Indexes	Total Index Size
administradores	2	89.5 B	179.0 B	1	32.0 KB
entradas	41	99.1 B	4.0 KB	1	36.0 KB
marcaciones	97	120.3 B	11.4 KB	1	36.0 KB
regresos_almuerzo	17	99.1 B	1.6 KB	1	36.0 KB
salidas	17	99.1 B	1.6 KB	1	36.0 KB
salidas_almuerzo	22	99.1 B	2.1 KB	1	36.0 KB
usuarios	7	226.7 B	1.5 KB	1	36.0 KB

Figura 3.5 Tablas del sistema ceducar en MongoDB.
Fuente: Los autores.

3.7.1. Tabla de administradores

La estructura de la tabla de administrador está definida de la siguiente manera: Véase Figura 3.6)

ADMINISTRADORES
<pre> +_id +correo: String +clave: String +activo: Boolean </pre>

Figura 3.6 Estructura de registro para el administrador.
Fuente: Los autores.

Como podemos observar en la tabla de administrador el dueño de la empresa ingresa al sistema con su correo y clave para ver toda la información de los usuarios de la escuela de conducción.

3.7.2. Tabla de usuarios

La estructura de la tabla de usuarios está definida de la siguiente manera: (véase Figura 3.7)

USUARIOS
<pre> +_id +nombre: String +correo: String +cedula: String +telefono: String +clave: String +huella_id: String +activo: Boolean +actualizado_en: Datetime +creado_en: Datetime </pre>

Figura 3.7. Estructura para el registro de usuarios.
Fuente: Los autores.

En la tabla se almacena toda la información del usuario con los campos detallados como nombre, email, correo, cedula, teléfono, clave, huella id, donde el empleado acceso al sistema para ver su registro de marcación.

3.7.3. Tabla de Marcaciones

La estructura de la tabla de registro de marcaciones está definida como podemos observar en la imagen descrita a continuación (véase Figura 3.8)

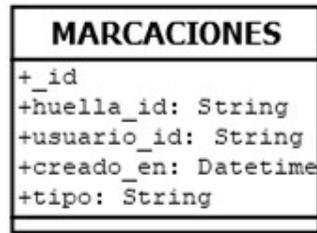


Figura 3.8. Estructura para el registro de marcaciones.
Fuente: Los autores.

En el registro de marcaciones cada vez que se haga una marcación se va registrar en la base de datos de la tabla de marcación.

3.7.4. Entrada

La estructura de la tabla de registro de entradas está definida como podemos observar en la siguiente imagen (véase Figura 3.9).

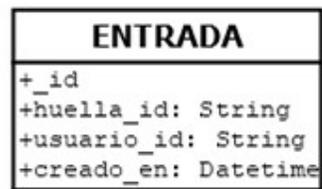


Figura 3.9. Estructura de registro de Marcaciones.
Fuente: Los autores.

En registro de entradas cada vez que un usuario con su huella dactilar haga una marcación de entrada se va registrar automáticamente en la tabla de entradas.

3.7.5. Salida almuerzo

La estructura de la tabla de registro de salidas de almuerzo está definida como podemos observar en la siguiente imagen (véase Figura 3.10).

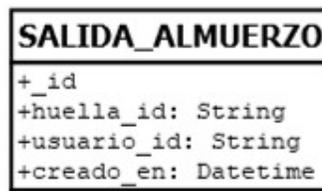


Figura 3.10. Estructura de registro de Salida de Almuerzo.
Fuente: Los autores.

En registro de salidas de almuerzo cada vez que un usuario con su huella dactilar en el biométrico seleccione la opción de salida de almuerzo se va registrar automáticamente en la tabla de salida de almuerzo.

3.7.6. Regreso almuerzo

La estructura de la tabla de registro de regreso de almuerzo está definida como podemos observar en la siguiente imagen (véase Figura 3.11).

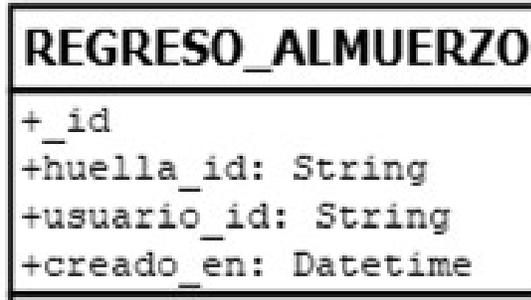


Figura 3.11. Estructura de registro de Salida de Almuerzo.
Fuente: Los autores.

En registro de regreso de almuerzo cada vez que un usuario con su huella dactilar en el biométrico seleccione la opción de regreso de almuerzo se va registrar automáticamente en la tabla de regreso de almuerzo.

3.7.7. Salidas

La estructura de la tabla de registro de salidas está definida como podemos observar en la siguiente imagen (véase Figura 3.12).

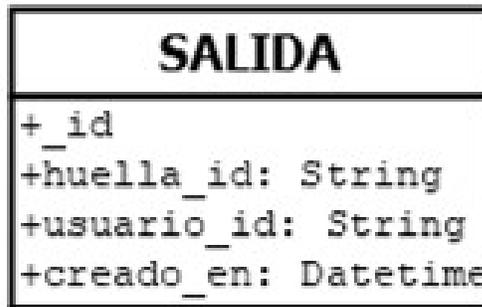


Figura 3.12. Estructura de registro de salida.
Fuente: Los autores.

En registro de salidas cada vez que un usuario con su huella dactilar en el biométrico seleccione la opción de salida se va registrar automáticamente en la tabla de salida.

3.7.8. Estructura total de la base de datos

La estructura final de la base de datos con todas sus funciones se ve como se demuestra en la siguiente figura. (véase Figura 3.13)



Figura 3.13. Estructura final de los datos de sistema de marcación.
Fuente: Los autores.

3.8. Servidor de infraestructura en la nube vultr.

El servidor web es un software que permite gestionar las solicitudes HTTP y se ejecuta del lado del servidor, es decir un cliente que desee realiza la petición solo necesitara de un navegador web, el cual en la actualidad es básico de todos los dispositivos tecnológicos.

Como software de servidor se va a utilizar Vultr por ser el software de infraestructura de la nube donde se enlazará todos los datos de las peticiones del api creadas en Python con la librería flask rest plus donde se conectará con el Host cargado de la base de datos Mongo DB, donde nos proporcionará una Ip estática para el control del sistema de marcación. (véase Figura 3.14)

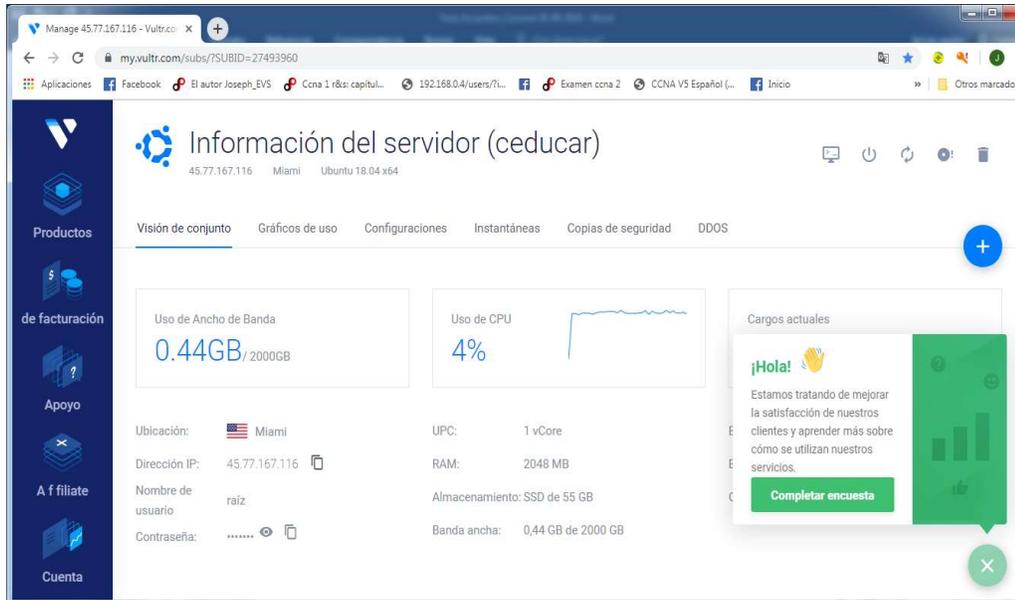


Figura 3.14. Servidor en la Nube del sistema de marcación.
Fuente: Los autores.

3.9. Comunicación del api con la base de datos

El API se la programo en Python con la librería Flask-Restplus, la cual nos ayuda a programar de una manera más fácil la API para todas las peticiones que requiera el sistema Biométrico de Ceducar tales como los CRUD para usuarios donde se almacenara todos los registros del personal en la plataforma.

Para acceder a la documentación del API será mediante la dirección Ip estática proporcionada por el servidor la cual es <http://45.77.167.116/>. (véase Figura 3.15)



Figura 3.15 Base de datos del api a través de la ip estática.
Fuente: Los autores

3.9.1. Interfaz de las peticiones del api del servidor Ceducar

A través de la lp estática mencionada anteriormente se selecciona la opción de las peticiones del api en donde se muestran las funciones para el envío de datos del sistema de marcación. (véase Figura 3.16)

3.9.2. Peticiones para administradores

En la siguiente ilustración se detalla las peticiones del administrador a través del api del sistema de marcación-

administradores	
Funciones de sesión para el administrador	
POST	/administradores/actualizartoken Actualizar token para administrador
POST	/administradores/login Acceso login para administrador
POST	/administradores/logout Logout para administrador
GET	/administradores/{administrador_id} Obtener un administrador buscado por ID

Figura 3.16 Peticiones para administrador del servidor CEDUCAR
Fuente: Los autores

3.9.3. Peticiones para usuarios

En la siguiente ilustración se detalla las peticiones de los usuarios a través del api del sistema de marcación donde el administrador tiene los permisos para acceder a los registros por cada usuario de la escuela de manejo. (véase Figura 3.17)

usuarios	
Funciones para los usuarios de la empresa.	
POST	/usuarios Crear un nuevo usuario
GET	/usuarios Obtener todos los usuarios activos
POST	/usuarios/login Acceso login para administrador
GET	/usuarios/{usuario_id} Obtener un departamento buscado por ID
DELETE	/usuarios/{usuario_id} Eliminar un usuario buscado por ID
PUT	/usuarios/{usuario_id} Actualizar un departamento buscado por ID
GET	/usuarios/{usuario_id}/marcaciones Obtener las marcaciones del usuario

Figura 3.17 Peticiones para usuarios del servidor CEDUCAR
Fuente: Los autores

3.9.4. Peticiones para marcaciones

En la siguiente ilustración se detalla las peticiones de marcaciones a través del api del sistema de marcación donde se detalla las marcaciones por cada usuario de la escuela de manejo. (véase Figura 3.18)

marcaciones	
GET	/marcaciones Obtener todas las marcaciones
POST	/marcaciones/entrada Registrar entrada
GET	/marcaciones/entrada Obtener todas las marcaciones de entrada
POST	/marcaciones/regreso_almuerzo Registrar entrada
GET	/marcaciones/regreso_almuerzo Obtener todas las marcaciones de regreso de almuerzo
POST	/marcaciones/salida Registrar salida
GET	/marcaciones/salida Obtener todas las marcaciones de salida
POST	/marcaciones/salida_almuerzo Registrar entrada
GET	/marcaciones/salida_almuerzo Obtener todas las marcaciones de salida de almuerzo

Figura 3.18 Peticiones de marcaciones del servidor Ceducar
Fuente: Los autores

3.9.5. Peticiones para ver todos los reportes

En la siguiente ilustración se detalla las peticiones de los reportes a través del api del sistema de marcación donde el administrador es el encargado de generarlos a través de la opción enviar reportes a través de la aplicación de administrador que le llegara por medio del correo personal. (véase Figura 3.19)

reportes	
POST	/reportes/entrada Enviar reporte de entrada al correo
POST	/reportes/marcaciones Enviar reporte de marcaciones al correo
POST	/reportes/regreso_almuerzo Enviar reporte de regreso de almuerzo al correo
POST	/reportes/salida Enviar reporte de salida al correo
POST	/reportes/salida_almuerzo Enviar reporte de salida de almuerzo al correo
POST	/reportes/usuarios Enviar reporte de usuarios al correo

Figura 3.19 Peticiones para reportes del servidor CEDUCAR
Fuente: Los autores

3.10. Procesamiento de solicitudes.

El api procesa las solicitudes a través de la librería flask rest plus creada en Python lidiando con las cabeceras HTTP además de controlar los sockets TCP/UDP y gestionar los recursos físicos del servidor vultr, la instancia que está alojada en vultr es la cual le llega las peticiones, pero el api es el motor del todo el back-end es la encargada de resolver la petición solicitada.

3.11. Interfaz de la aplicación del sensor biométrico-

La interfaz del prototipo de marcación para la aplicación web contara con un dos aplicativos uno para usuario y el otro para administrador cada uno de ellos contara con un inicio de sesión en donde se debe escribir el usuario y la contraseña para acceder. Al ingresar correctamente los datos, esta interfaz le va permitir solamente al usuario ver sus registros de marcaciones, en cambio el administrador es el único

que puede ver crear modificar y eliminar usuarios, además de obtener los reportes diarios de las marcaciones registradas por los usuarios de la escuela de manejo.

3.12. Desarrollo de aplicaciones de administrador y usuario para el prototipo de marcación.

Se va a realizar una aplicación web para almacenarla en la placa Asus Tinker Board utilizando Vue.js el cual es un framework de Node.js, este programa le va permitir al usuario acceder a sus registros de marcación en el cual tendrá a su disposición las peticiones de entrada, salida de almuerzo, regreso de almuerzo, salida, además tiene un botón de registro o verificación de huella, lo que permite obtener un código de huella dactilar por cada usuario, el cual es mostrado en la pantalla LCD 7", para que pueda presionar cualquier botón de registro de la marcación que desee realizar en las instalaciones de la escuela de manejo.

3.13. Front-end (Aplicación administrador) – Nuxt Vue.js

Para acceder a la aplicación del administrador será mediante el siguiente link

<http://45.77.167.116/app/administrador/>

En la aplicación el administrador podrá crear, leer, editar y eliminar usuarios. También se podrá ver todos los registros y marcaciones de todos los usuarios al igual que enviar a sus correos un reporte de las marcaciones en Excel. (véase Figura 3.20)

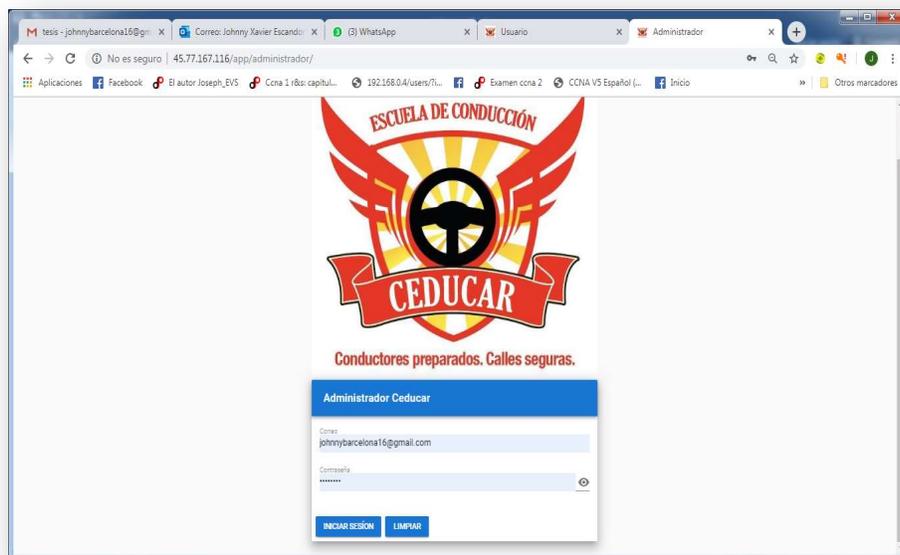


Figura 3.20 Interfaz de inicio de sesión de la aplicación de administrador del sistema de marcación Ceducar

Fuente: Los Autores

3.14. Front-end (Aplicación Administrador) – Nuxt Vue.js

Para acceder a la aplicación del administrador será mediante el siguiente link

<http://45.77.167.116/app/usuario/>

En la aplicación se podrá ver todos los registros y marcaciones del mismo usuario.

(véase Figura 3.21)

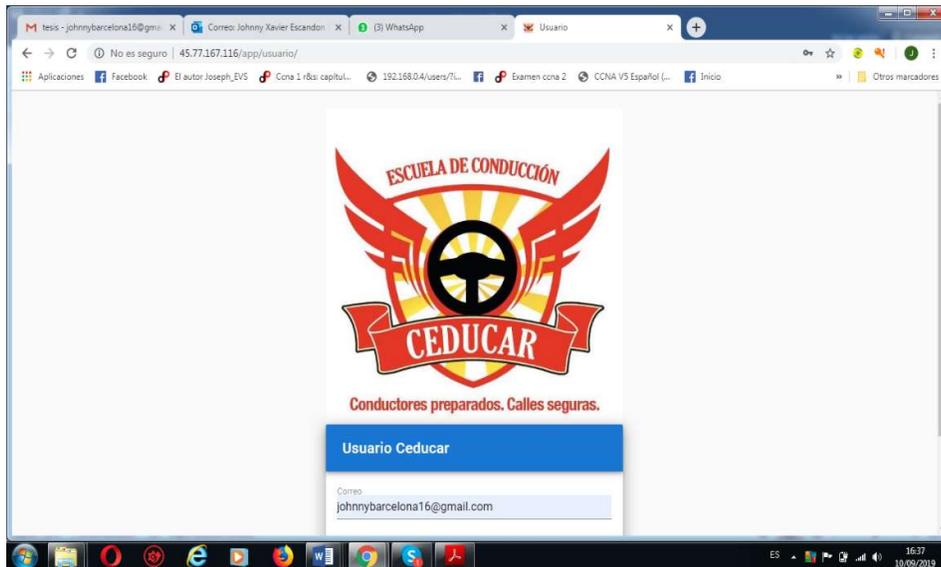


Figura 3.21 Interfaz de inicio de sesión de la aplicación de usuario del sistema de marcación Ceducar
Fuente: Los Autores

3.15. Interfaz del sensor biométrico

El sensor biométrico posee una interfaz web que puede ser visualizada únicamente de manera local en el equipo, es decir que no se puede acceder y ver desde cualquier lugar, sólo se puede visualizar desde el propio sensor biométrico, esta interfaz se podrá visualizar desde la pantalla LCD 7" que está conectada la Asus Tinker Board la cual recibe y envía datos correspondientes al sensor biométrico a través del dispositivo USB TTL

3.16. Front-end (Aplicación de biométrico) – Python-HTML

Esta aplicación es la interfaz que se muestra en el equipo biométrico, la aplicación se encuentra instalada directamente en el equipo, la cual ayuda al usuario a realizar su registro de marcaciones como entra, salida de almuerzo, regreso de almuerzo y salidas al momento de realizar la marcación se hace uso del API mediante la petición POST /marcaciones/entrada, /marcaciones/salida almuerzo, /marcaciones/regreso almuerzo y /marcaciones/salida; cuando se realiza el registro exitoso el servidor procede a enviar al correo la marcación de usuario con fecha y hora. Adicionalmente

cuanto con un botón para registrar las huellas o verificar la huella de algún usuario.
(véase Figura 3.22)



Figura 3.22 Interfaz final del sistema de marcación Ceducar
Fuente: Los Autores I

CAPÍTULO IV

4. ANÁLISIS EXPERIMENTAL DEL PROTOTIPO DE CONTROL DE ACCESO.

4.1. Análisis experimental del Dispositivo de acceso Biométrico

Al dar comienzo al propósito del proyecto se manifestó como meta inicial el diseño e implementación de un prototipo de control con acceso biométrico visualizado a través desde una app Web el mismo que consta de varios bloques de realización los cuales se especifican a continuación: (véase Figura 4.1)

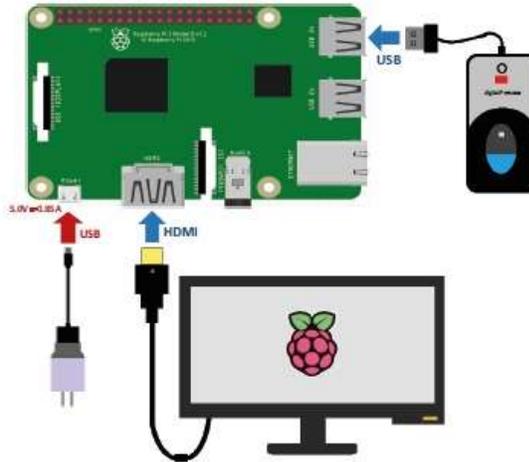


Figura 4.1 Diagrama General del Proyecto
Fuente: Los Autores.

El sensor de huella DY50 tiene una comunicación de salida serial (RS232) por lo cual necesitamos un convertidor de comunicación serial a USB para conectarlo a la tarjeta ASUS Tinker Board, entonces se lo conecta a la ASUS por el puerto USB, para poder visualizar la aplicación del biométrico se conectada la pantalla a la tarjeta mediante un cable HDMI y para poder energizar y utilizar el táctil de la pantalla se debe conectar el cable USB.

Para alimentar nuestro prototipo de marcación es necesario una fuente de 5V-3A, ya que la tarjeta Asus Tinker Board admite dichas características para su funcionamiento. La pantalla de 7 pulgadas táctil se conectó a través de un cable de energía de cargador de 5A-1.5A.

4.2. Conexión sensor dy50 biométrico

El sensor biométrico se conectó al dispositivo Arduino pro mini a través del puerto serie por software que se programó para la comunicación con el biométrico, a través de conectores para la comunicación con el sensor de huella con la Asus Tinker Board.

Para la captura de código de huella dactilar a través del sensor dy50 el dispositivo USB-TTL es el encargado de conectar al sensor biométrico el cual brinda la información de la huella de cada usuario a través de la tarjeta Asus que es la encargada de recibir y gestionar los datos para la ejecución del api de administrador y usuarios para que realicen sus registros de marcaciones.

Las conexiones realizadas fueron la alimentación del lector de huellas 5 voltios, tierra, transmisión tx, recepción rx a los pines ubicados y señalados por medio de serigrafía en el dispositivo USB-TTL.

La placa principal Asus Tinker Board necesita una conexión hacia internet debido a lo propuesto en el proyecto de titulación con el fin de visualizar la interfaz del registro de marcaciones para los usuarios a través del servidor vultr que nos proporcionara los servicios para la operatividad del sistema, la placa consta con un puerto Ethernet para conexión a internet por medio de cables físicos y también es inalámbricamente el acceso para conectarse con la red que brinda los servicios de internet a la escuela de manejo.

El equipo biométrico se lo construyo con acrílico en todo lo que se refiera a la estructura, tiene una tapa desmontable. De pantalla -> Asus, Sensor de huella -> Asus, Cargador -> Asus

El lector de huellas ubicado en la parte lateral del acrílico para mayor comodidad para la captura de la huella digital, procesar la huella registrarla en la base de datos y así conjuntamente con los demás equipos realizar procesos asíncronos.

Todos estos procesos en conjunto funcionamiento son de vital importancia debido a que la comunicación del biométrico debe ser reflejada en procesos de reconocimiento y siendo visualizado en la pantalla para interactuar con el usuario y posterior comunicación con su conexión inalámbrica para servidor.

La validación de huellas ingresadas al lector de huellas dependerá del registro por medio de interfaz de pantalla y su lista de huella ya almacenadas en memoria para reconocimiento y activación del api del servidor para su debida verificación.

En la interfaz gráfica por medio de la pantalla se encuentran involucrados la Asus Tinker Board y el sensor biométrico debido a que se mostrara en pantalla un índice que posee 5 cuadros para su registro de marcaciones y además el registro de huella para obtener su código dactilar para su debido registro con el administrador de la escuela de manejo.

Para el registro de huellas es necesario colocar la huella en el sensor por 5 seg levantar la huella después de otros 5 seg colocarla para que el sensor registre y procese la huella dactilar para que esos datos sean enviados a la Asus Tinker Board donde le mostrara automáticamente su código de huella dactilar.

Cada usuario que es agregado puede ser modificado tanto para agregar o eliminar huellas existentes así mismo como el cambio de nombres por medio de un teclado inalámbrico a través de los puertos USB de la Asus.

4.3. Implementación del proyecto

La implementación del proyecto de tesis se ejecutó en las instalaciones de la escuela de manejo ceducar junto a la oficina de información. (véase Figura 4.2)



Figura 4.2 Montaje de la tarjeta Asus al Tablero del prototipo de marcación
Fuente: Los autores.

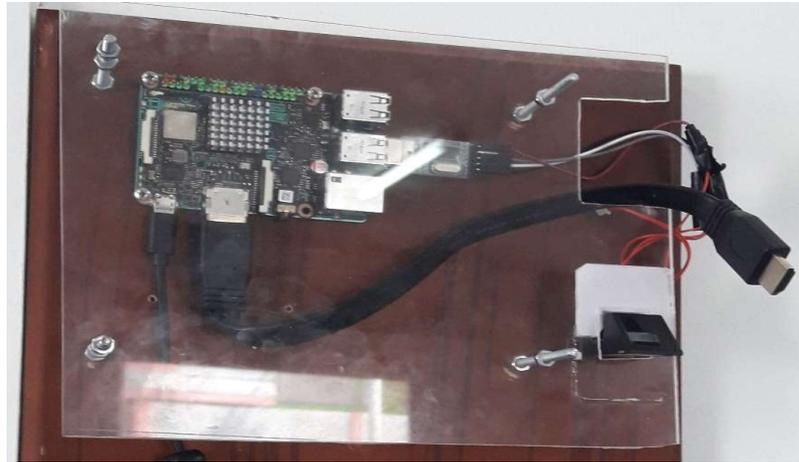


Figura 4.3 Cableado general de la tarjeta asus tinker board con el sensor de huella.
Fuente: Los autores.



Figura 4.4 Instalación de un tomacorriente para el encendido del prototipo de marcación
Fuente: Los autores



Figura 4.5 Elaboración del acrílico de la estructura general del prototipo de marcación.
Fuente: Los autores.



Figura 4.6 Instalación del LCD en el *Tablero del prototipo de marcación*
Fuente: Los autores



Figura 4.7 Montaje de la tapa de la estructura del sistema de marcación
Fuente: Los autores



Figura 4.8 Funcionamiento del prototipo de marcación en la escuela de manejo CEDUCAR
Fuente: Los autores

Conclusiones

Se comprobó que la Asus Tinker Board es un minicomputador con un gran campo de prestaciones para su comunicación con aplicaciones web que utilizan bases de datos y permiten la conexión de varios clientes de manera estable y con un bajo consumo de energía y de bajo costo en adquisición de la misma.

El software Vultr fue usado como servidor web el cual nos brinda los servicios para la comunicación con los demás programas usados en nuestro sistema, posee un gran rendimiento y permite la múltiple conexión de varios hosts al servidor, es muy segura y de fácil uso para el administrador.

El uso de un api basada en JSON hace que el consumo de datos de cualquier dispositivo sea muy reducido dotando de nuevas funcionalidades y avances tecnológicos a la plataforma para brindar una alta escalabilidad para su acceso rápido y sencillo para los usuarios.

En este proyecto de titulación existe un entorno seguro en la parte de comunicación entre el prototipo de marcación implementado en la escuela de manejo y el dueño del mismo, ya que se pueden enviar mensajes de email con los reportes bien detallados de cada registro de marcación de los demás usuarios de la escuela de conducción, además se registra automáticamente en el aplicativo web donde el dueño puede acceder para registrar, ver, editar y eliminar de una manera más fácil. Esto demuestra que el sistema organizado es dinámico e innovador.

Se concluyó como resultado que las pruebas realizadas en la implementación del prototipo de marcación fueron un principal aporte para las mejoras de la misma, para el desarrollo del sistema en el software y la culminación total del proyecto con resultados exitosos.

Recomendaciones

Evitar el desmontaje de la base del prototipo de marcación instalada y todos los componentes que lo conforman debido a que contienen placas electrónicas y conexiones eléctricas lo que puede ocasionar inconvenientes y fallos en el sistema.

Prevenir el contacto directo de los implementos con medios como líquidos o exponerlo a altas temperaturas debido a que acortan su vida útil produciendo a futuro un fallo total entorno al sistema implementado.

Es muy importante por seguridad y estabilidad dar acceso a la red para el dispositivo central del sistema de marcación por medio de cable en lugar de wifi,

Para brindar una óptima precisión del sensor biométrico se recomienda limpiar la lente del sensor diariamente al término de la jornada laboral.

Es conveniente realizar un mantenimiento preventivo de la base de datos del servidor de infraestructura de la nube cada 6 meses con el fin de reducir las tareas al servidor y brindar un manejo eficiente

Se recomienda hacer un respaldo de las Backup de la base de datos semanalmente para evitar la pérdida de información en caso de algún daño en cualquiera de los dispositivos que conforman el prototipo de marcación.

El presente proyecto de sistema de marcación permite reforzar la validación de la eficiencia de un correcto análisis organizacional estratégico en beneficio de los colaboradores de la escuela de manejo Ceducar.

Bibliografía

- Alegsa, L. (05 de Diciembre de 2010). *alegsa.com.ar*. Obtenido de *alegsa.com.ar*: http://www.alegsa.com.ar/m/Dic/servidor_de_base_de_datos.php
- Asus. (2017). *Tablero Tinker*. Asustek.
- Barbella, R. (19 de Febrero de 2019). *Udemy.com*. Obtenido de Udemy.com: <https://www.udemy.com/course/se-un-programador-mas-productivo-con-visual-studio-code/>
- Boicea, A., Radulescu, F., & Agapin, L. I. (2012). MongoDB vs Oracle--database comparison. *third international conference on emerging intelligent data and web technologies* (págs. 330-335). IEEE.
- Calle, V. (2016). *Control de asistencia mediante lector de huella dactilar*. Obtenido de http://dehesa.unex.es/bitstream/handle/10662/5066/TFGUEX_2016_Calle_Sanchez.pdf?sequence=1&isAllowed=y
- Cañas, D. C., Daza, B. S., & Parra, O. J. (2015). Wi-Fi channels saturation using standard Wi-Fi gateway. *International Conference on Mobile, Secure and Programmable Networking* (págs. 101-108). Cham: Springer.
- Castro, S. (2017). Biometría: conveniente y segura. *Revista de seguridad tecnologica*.
- Comprarthosting. (03 de Marzo de 2019). *comprarhosting.co*. Obtenido de *comprarhosting.co*: <https://www.comprarhosting.co/opiniones-de-vultr-en-espanol-cloud-hosting-vps/>
- Cuervo, V. (16 de Enero de 2016). *lineadecodigo.com*. Obtenido de *lineadecodigo.com*: <http://lineadecodigo.com/python/hola-mundo-con-python-flask/>
- Cuervo, V. (09 de Marzo de 2016). *lineadecodigo.com*. Obtenido de *lineadecodigo.com*: <http://lineadecodigo.com/python/hola-mundo-pymongo/>
- Cueva, C. (2015). *Estudio de un sistema biometrico*. Lanus.
- Filipova, O. (2016). *Learning Vue. js 2*. London: Packt Publishing Ltd.
- Goette, E. (10 de Septiembre de 2019). *emanuelpeg.blogspot*. Obtenido de *emanuelpeg.blogspot*: emanuelpeg.blogspot.com
- Gonzales, O., & Cervantes, F. (2016). Implementación de aplicaciones con acceso a base de datos. *Revista de Estrategias del Desarrollo Empresarial*.
- Gudiño de la A, E., & Chavarria Neira, B. (2017). *Implementación de un servidor web y un diseño de una página utilizando herramientas de software libre para el dispensario "Sagrada Familia" de la ciudad de Guayaquil*. Guayaquil: Universidad Politécnica Salesiana.
- Hernandez, C. (2015). *Estudio del rendimiento biometrico de dispositivos de huella dactilar*. Obtenido de <https://core.ac.uk/download/pdf/79176660.pdf>

jgtecnologias. (12 de Septiembre de 2019). *Node Js*. Obtenido de jgtecnologias: <http://www.jgtecnologias.com.ve/course/node-js/>

Li, G. (2016). *Innovative methods for large-scale fingerprint identification systems-Facilitating searching in a large-scale database*. Trondheim, Noruega: Norwegian University of Science and Technology.

Lobby Fix. (16 de Noviembre de 2018). *Lobbyfix.com*. Obtenido de Lobbyfix.com: <https://lobbyfix.com/que-es-un-sistema-de-control-de-acceso/>

Lorenzo, I. (18 de Mayo de 2018). *VISUAL STUDIO CODE*. Obtenido de El Atareao: <https://www.atareao.es/software/programacion/visual-studio-code/>

Nedelcu, C. (2010). *Nginx HTTP Server: Adopt Nginx for Your Web Applications to Make the Most of Your Infrastructure and Serve Pages Faster Than Ever*. Birmingham: Packt Publishing Ltd.

Nolle, T., & Li, T. (Octubre de 2018). *searchdatacenter.techtarget.com*. Obtenido de searchdatacenter.techtarget.com: <https://searchdatacenter.techtarget.com/es/definicion/Interfaz-de-programacion-de-aplicaciones-API>

Oracle. (2016). *Guía de administración de los sistemas Oracle*. Superclus.

Ortiz, J. (2015). *Arquitectura de referencia para Oracle*. TIC.

Portilla, O. (2015). *Funcion de sensor biometrico*. Tecnosiner.

PROMETEC. (12 de Septiembre de 2019). *Sensor de huellas dactilares*. Obtenido de PROMETEC: <https://www.prometec.net/lector-de-huellas/>

Richards, M. (2015). *Software Architecture Patterns*. O'Reilly Media, Inc.

Sanchez, S. (2015). *Rendimiento biometrico de sistemas de huella dactilar de diferentes sensores y algoritmos*. Obtenido de https://e-archivo.uc3m.es/bitstream/handle/10016/23621/TFG_Sergio_Sanchez_Martin.pdf

Suarez, D. (2016). *Protocolo de identificacion de huellas dactilares*. Extravi.

Tite, T., Fabián, G., Caiza, L., & Angélica, M. (2017). *Sistema de monitoreo personal inalámbrico para atletas de fondo*. Riobamba: Tesis pregrado, Universidad Nacional de Chimborazo).

Tolosa, C., & Giz, A. (2016). *Sistemas biometricos*. File.

Van Rossum, G., & Drake, F. L. (2011). *The python language reference manual*. London: Network Theory Ltd.

vuejs. (12 de Septiembre de 2019). *Introduction: What is Vue.js?* Obtenido de vuejs.org: <https://vuejs.org/v2/guide/>

CRONOGRAMA DE ACTIVIDADES

Tabla 4
Cronograma de actividades

Actividades	MESES											
	1	2	3	4	5	6	7	8	9	10	11	12
Análisis y determinación de equipos y materiales a utilizar.	■											
Compra de equipos y materiales.		■	■									
Elaboración de diagramas esquemáticos para comunicación de equipos.			■									
Diseño y elaboración de aplicación para el sensor biométrico.				■								
Diseño de la aplicación web.					■	■						
Diseño y elaboración de la base de datos.						■	■					
Elaboración de la estructura para el prototipo de marcación.							■	■				
Etapa de pruebas.				■			■	■		■		
Instalación del prototipo de marcación en las Instalaciones de Ceducar											■	■

Fuente: Los autores

PRESUPUESTO

Tabla 5
Presupuesto de materiales

Materiales	Cantidad	Precio unitario	Precio total
Curso de Programación Python	1	\$ 250.00	\$ 250.00
Curso de Linux	1	\$ 250.00	\$ 250.00
Tarjeta Asus Tinker Board	1	\$ 200.00	\$ 200.00
Pantalla 7" touch screen LCD	1	\$ 120.00	\$ 120.00
Sensor de Huella DYS 50	2	\$ 30.00	\$ 60.00
Cargador para Pantalla LCD 5V-3A	1	\$20.00	\$ 20.00
Arduino Uno R3	1	\$ 20.00	\$ 20.00
Case Acrylic for late panda	1	\$ 25.00	\$ 25.00
Servidor web Vultr	1	\$ 20.00	\$ 20.00
Estructura para prototipo de marcación	1	\$ 200.00	\$ 200.00
Cable de datos para tarjeta Asus Tinker Board	1	\$ 25.00	\$ 25.00
Conectores para sensor de huella	4	\$ 5.00	\$ 20.00
Mouse Inalámbrico	1	\$15.00	\$15.00
Cable para conector RJ45	10 metros	\$ 0.60	\$ 6.00
Ácido férrico	5 sobres	\$ 0.75	\$ 3.75
Canaletas	2	\$ 5.00	\$ 10.00
Borneras plásticas	6 tiras	\$ 1.00	\$ 6.00
Gastos de Ingeniería	1	\$ 250	\$ 250
Total			\$ 1500.75

Fuente: Los autores

ANEXOS

Codigos para el prototipo de marcación.

Codigo Principal

A continuación, se detalla los archivos principales que se ejecutan en la tarjeta principal en los dispositivos que conforman el prototipo de marcación.

Archivo run.py

```
from flask import Flask, render_template, jsonify, redirect, url_for
from time import sleep
from flask_socketio import SocketIO, send, emit
from threading import Lock
import Finger
import requests
¶
asvnc_mode = None
¶
app = Flask(__name__, static_url_path='', static_folder='static', template_folder='templates')
app.config['SECRET_KEY'] = 'secret!'
socketio = SocketIO(app, asvnc_mode=asvnc_mode)
thread = None
thread_lock = Lock()
¶
.....¶
@socketio.on('marcacion')
def handle_marcacion(msg):
    print('Iniciar busqueda')
    Finger.desactivar()
    sleep(1)
    Finger.activar()
    huella = Finger.encontra_huella()
    if huella['error'] == False:
```

Figura A.1 Código fichero principal para la interfaz del prototipo de marcación
Fuente: Los Autores

```

-----url="http://45.77.167.116/marcaciones/" +msg
-----print(huella)
#
#
-----payload="-----011000010111000001101001\r\nContent-Disposition:form-data;
name=\"huella_id\"\r\n\r\n"+str(huella['huella_id'])+"\r\n-----011000010111000001101001--\r\n"
-----headers={'content-type':'multipart/form-data;boundary=---011000010111000001101001'}
#
-----response=requests.request("POST",url,data=payload,headers=headers)
-----if response.status_code==200:
-----response=response.json()
-----print(response)
-----emit('huella_encontrada',response['data'])
-----return
---
---emit('error','Huella-no encontrada')
#
@socketio.on('registrar')
def handle_registra(registrar):
---print('Iniciar busqueda')
---Finger.desactivar()
---sleep(1)
---Finger.activar()
---huella_id=Finger.registrar_huella()
---emit('huella_registrada',huella_id)

```

Figura A.2 Código fichero principal para la interfaz del sistema de marcación
Fuente: Los Autores

```

# Ruta raiz
@app.route("/dashboard")
def dashboard():
---return render_template('dashboard.html')# Renderiza html
#
#
if __name__ == '__main__':
---#app.run(debug=True,host='192.168.10.51')
---app.run(debug=True,port=5050)
---socketio.run(app)

```

Figura A.3 Código fichero principal para la interfaz del sistema de marcación
Fuente: Los Autores

Código Escaneo de Huella

Archivo “Finger.py”

```
#!/usr/bin/env python
#-*-coding:utf-8-*-
"""
PyFingerprint
Copyright (C) 2015 Bastian Raschke <bastian.raschke@posteo.de>
All-rights-reserved.
"""
import hashlib
import time
from pyfingerprint.pyfingerprint import PyFingerprint
flag = False
##-Search-for-a-finger
##
##-Tries-to-initialize-the-sensor
try:
    f = PyFingerprint('/dev/ttyUSB0', 57600, 0xFFFFFFFF, 0x00000000)

```

Figura A.4 Código escaneo de huella
Fuente: Los Autores

```
    if (f.verifyPassword() == False):
        raise ValueError("The-given-fingerprint-sensor-password-is-wrong!")
    except Exception as e:
        print("The-fingerprint-sensor-could-not-be-initialized!")
        print("Exception-message: "+str(e))
        exit(1)
    ##-Gets-some-sensor-information
    print("Currently-used-templates: "+str(f.getTemplateCount())+" / "+str(f.getStorageCapacity()))
    def activar():
        global flag
        flag = True
        print("Sensor-activado")
    def desactivar():
        global flag
        flag = False
        print("Sensor-desactivado")
    def encontra_huella():
        ##-Tries-to-search-the-finger-and-calculate-hash
        try:
            print("Waiting-for-finger...")

```

Figura A.5 Código escaneo de huella
Fuente: Los Autores

```

.....##-Wait-that-finger-is-read¶
.....while-(!f.readImage()==False-and-flag==True):¶
.....pass¶

¶
.....##-Converts-read-image-to-characteristics-and-stores-it-in-charbuffer-1¶
.....f.convertImage(0x01)¶

¶
.....##-Searchs-template¶
.....result=f.searchTemplate()¶

¶
.....positionNumber=result[0]¶
.....accuracyScore=result[1]¶

¶
.....if-(!positionNumber==-1):¶
.....print('No-match-found!')¶
.....return({'error':True,'msg':'Huella-no-encontrada'})¶
.....else:¶
.....print('Found-template-at-position-#'+str(positionNumber))¶
.....print('The-accuracy-score-is:'+str(accuracyScore))¶
.....return({'error':False,'msg':'Huella-encontrada','huella_id':positionNumber})¶

¶
.....###-OPTIONAL-stuff¶
.....###¶
.....###-Loads-the-found-template-to-charbuffer-1¶

```

Figura A.6 Código escaneo de huella
Fuente: Los Autores

```

.....###-Loads-the-found-template-to-charbuffer-1¶
.....f.loadTemplate(positionNumber,0x01)¶

¶
.....###-Downloads-the-characteristics-of-template-loaded-in-charbuffer-1¶
.....characterics=str(f.downloadCharacteristics(0x01)).encode('utf-8')¶

¶
.....###-Hashes-characteristics-of-template¶
.....# print('SHA-2-hash-of-template:'+hashlib.sha256(characterics).hexdigest())¶

¶
.....except-Exception-as-e:¶
.....print('Operation-failed!')¶
.....print('Exception-message:'+str(e))¶
.....exit(1)¶

¶
def registrar_huella():¶
.....##-Tries-to-enroll-new-finger¶
.....try:¶
.....print('Waiting-for-finger...')¶

¶
.....##-Wait-that-finger-is-read¶
.....while-(!f.readImage()==False-and-flag==True):¶
.....pass¶

```

Figura A.7 Código escaneo de huella
Fuente: Los Autores

```

"
.....## Converts read image to characteristics and stores it in charbuffer-1¶
.....f.convertImage(0x01)¶
¶
.....## Checks if finger is already enrolled¶
.....result = f.searchTemplate()¶
.....positionNumber = result[0]¶
¶
.....if (-positionNumber >= 0):¶
.....    print('Template already exists at position #' + str(positionNumber))¶
.....    return positionNumber¶
¶
.....print('Remove finger...')¶
.....time.sleep(2)¶
¶
.....print('Waiting for same finger again...')¶
¶
.....## Wait that finger is read again¶
.....while (-f.readImage() != False):¶
.....    pass¶
¶
.....## Converts read image to characteristics and stores it in charbuffer-2¶
.....f.convertImage(0x02)¶

```

Figura A.8 Código escaneo de huella
Fuente: Los Autores

```

.....## Compares the charbuffers¶
.....if (-f.compareCharacteristics() != 0):¶
.....    raise Exception('Fingers do not match')¶
¶
.....## Creates a template¶
.....f.createTemplate()¶
¶
.....## Saves template at new position number¶
.....positionNumber = f.storeTemplate()¶
.....print('Finger enrolled successfully!')¶
.....print('New template position #' + str(positionNumber))¶
.....return positionNumber¶
¶
except Exception as e:¶
.....print('Operation failed!')¶
.....print('Exception message: ' + str(e))¶
.....exit(1)¶
¶

```

Figura A.9 Código escaneo de huella
Fuente: Los Autores

Página web del biométrico

Archivo “dashboard.html”

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

"""
PyFingerprint
Copyright (C) 2015 Bastian Raschke <bastian.raschke@posteo.de>
All rights reserved.

"""

import hashlib
import time

from pyfingerprint.pyfingerprint import PyFingerprint

flag = False

|
## Search for a finger
##

|
```

Figura A.10 Código página web del fichero del registro de marcación
Fuente: Los Autores

```
|
## Tries to initialize the sensor

try:

... f = PyFingerprint('/dev/ttyUSB0', 57600, 0xFFFFFFFF, 0x00000000)

|
... if (f.verifyPassword() == False):

..... raise ValueError('The given fingerprint sensor password is wrong!')

|
except Exception as e:

... print('The fingerprint sensor could not be initialized!')

... print('Exception message: ' + str(e))

... exit(1)
```

Figura A.11 Código página web del fichero del registro de marcación
Fuente: Los Autores

```

¶
##-Gets-some-sensor-information¶
print('Currently-used-templates-'+str(f.getTemplateCount())+'/'+str(f.getStorageCapacity()))¶
¶
def activar():¶
    global flag¶
    flag=True¶
    print("Sensor-activado")¶
¶
def desactivar():¶
    global flag¶
    flag=False¶
    print("Sensor-desactivado")¶
¶
¶
def encontra_huella():¶
    ##-Tries-to-search-the-finger-and-calculate-hash¶
    try:¶
        print("Waiting-for-finger...")¶
    ¶
    ##-Wait-that-finger-is-read¶
    while (f.readImage()==False and flag==True):¶
        pass¶

```

Figura A.12 Código página web del fichero del registro de marcación
Fuente: Los Autores

```

.....##-Converts-read-image-to-characteristics-and-stores-it-in-charbuffer-1¶
.....f.convertImage(0x01)¶
¶
.....##-Searchs-template¶
.....result=f.searchTemplate()¶
¶
.....positionNumber=result[0]¶
.....accuracyScore=result[1]¶
¶
.....if (positionNumber==1):¶
        print("No-match-found!")¶
        return {'error': True, 'msg': 'Huella-no-encontrada'}¶
    else:¶
        print("Found-template-at-position-#"+str(positionNumber))¶
        print("The-accuracy-score-is:"+str(accuracyScore))¶
        return {'error': False, 'msg': 'Huella-encontrada', 'huella_id': positionNumber}¶

```

Figura A.13 Código página web del fichero del registro de marcación
Fuente: Los Autores

```

-----###OPTIONAL-stuff¶
-----# #¶
-----###Loads-the-found-template-to-charbuffer:1¶
-----# f.loadTemplate(positionNumber,-0x01)¶
¶
-----###Downloads-the-characteristics-of-template-loaded-in-charbuffer:1¶
-----# characteristics = str(f.downloadCharacteristics(0x01)).encode('utf-8')¶
¶
-----###Hashes-characteristics-of-template¶
-----# print('SHA-2-hash-of-template:' + hashlib.sha256(characteristics).hexdigest())¶
¶
-----except-Exception-as-e:¶
-----print('Operation-failed!')¶
-----print('Exception-message:' + str(e))¶
-----exit(1)¶

```

Figura A.14 Código página web del fichero del registro de marcación
Fuente: Los Autores

```

def registrar_huella():¶
    ---##Tries-to-enroll-new-finger¶
    ---try:¶
    -----print('Waiting-for-finger...')¶
    ¶
    -----##Wait-that-finger-is-read¶
    -----while-(f.readImage()==False-and-flag==True):¶
    -----pass¶
    ¶
    -----##Converts-read-image-to-characteristics-and-stores-it-in-charbuffer:1¶
    -----f.convertImage(0x01)¶
    ¶
    -----##Checks-if-finger-is-already-enrolled¶
    -----result=f.searchTemplate()¶
    -----positionNumber=result[0]¶
    ¶
    -----if-(positionNumber >= 0):¶
    -----print('Template-already-exists-at-position-' + str(positionNumber))¶
    -----return positionNumber¶

```

Figura A.15 Código página web del fichero del registro de marcación
Fuente: Los Autores

```

}|
|-----print('Remove-finger...')|
|-----time.sleep(2)|
}|
|-----print('Waiting-for-same-finger-again...')|
}|
|-----##-Wait-that-finger-is-read-again|
|-----while(-f.readImage())==False-:|
|-----pass|
}|
|-----##-Converts-read-image-to-characteristics-and-stores-it-in-charbuffer-2|
|-----f.convertImage(0x02)|
}|

```

Figura A.16 Código página web del fichero del registro de marcación
Fuente: Los Autores

```

|-----##-Compares-the-charbuffers|
|-----if(-f.compareCharacteristics())==0-:|
|-----raise-Exception('Fingers-do-not-match')|
}|
|-----##-Creates-a-template|
|-----f.createTemplate()|
}|
|-----##-Saves-template-at-new-position-number|
|-----positionNumber=f.storeTemplate()|
|-----print('Finger-enrolled-successfully')|
|-----print('New-template-position-#'+str(positionNumber))|
|-----return-positionNumber|
}|
|-----except-Exception-as-e-|
|-----print('Operation-failed')|
|-----print('Exception-message-:'+str(e))|
|-----exit(1)|
}

```

Figura A.17 Código página web del fichero del registro de marcación
Fuente: Los Autores

Fotografías de la implementación del proyecto en ejecución



Figura A.18 Montaje de la tarjeta Asus al Tablero del prototipo de marcación
Fuente: Los Autores



Figura A.19 Cableado general de la tarjeta asus tinker board con el sensor de huella.
Fuente: Los Autores



Figura A.20 Instalación de un tomacorriente para el encendido *del prototipo de marcación*.
Fuente: Los Autores



Figura A.21 Elaboración del acrílico de la estructura general del prototipo de marcación.
Fuente: Los Autores



Figura A.22 Instalación del LCD en el *Tablero del sistema de marcación*
Fuente: Los Autores



Figura A.23 Instalación del LCD en el *Tablero del prototipo de marcación*
Fuente: Los Autores



Figura A.24 Funcionamiento del prototipo de marcación en la escuela de manejo CEDUCAR
Fuente: Los Autores

REPORTE USUARIOS						
ID	Nombre	Correo	Cedula	Teléfono	Clave	Huella ID
5e4e0e8df8eb50aad1b1d6bcf	Johnny	jescandon170393@gmail.com	0987654321	0982904635	170393	4
5e4f0d843e758b55c9147e49	Xavier	xavlirescandonp1993@gmail.com	0906012893	042573255	230298	6
5e5067c094777d47961e329f	PlanB	xavilixitobarcelona20@gmail.com	098234567	0906122456	Mariuxi	8
5e5069ea191a76f5d0976781	Jose Arizaga	jescandon@est.ups.edu.ec	0987686578	0983272729	Ups2020	16

Figura A.25 Elaboración de los reportes coloridos de la plataforma del sistema de marcación.
Fuente: Los Autores