



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD BIOMÉTRICA
PARA LA UNIDAD DE INNOVACIÓN TECNOLÓGICA DE LA
UNIVERSIDAD DE LAS AMÉRICAS

AUTORES

DIEGO ESTEBAN FLORES MORENO
SANTIAGO ALEJANDRO VILLACÍS FLORES

AÑO

2018



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD BIOMÉTRICA PARA
LA UNIDAD DE INNOVACIÓN TECNOLÓGICA DE LA UNIVERSIDAD DE LAS
AMÉRICAS

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingenieros en Electrónica y Redes de
Información

Profesor Guía

Mdhd. Héctor Fernando Chinchero Villacís

Autores

Diego Esteban Flores Moreno

Santiago Alejandro Villacís Flores

Año

2018

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo, implementación de un sistema de seguridad biométrica para la unidad de innovación tecnológica de la Universidad de las Américas, a través de reuniones periódicas con los estudiantes Diego Esteban Flores Moreno, Santiago Alejandro Villacís Flores, en el semestre 2018-2, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

Héctor Fernando Chinchero Villacís

Máster en Domótica

C.I.: 1715451330

DECLARACIÓN DEL PROFESOR CORRECTOR

"Declaro haber revisado este trabajo, implementación de un sistema de seguridad biométrica para la unidad de innovación tecnológica de la Universidad de las Américas, de Diego Esteban Flores Moreno, Santiago Alejandro Villacís Flores, en el semestre 2018-2, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

Luis Santiago Criollo Caizaguano

Máster en Redes de Comunicación

C.I.:1717112955

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaramos que este trabajo es original, de nuestra autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

Diego Esteban Flores Moreno

C.I.: 1716389117

Santiago Alejandro Villacís Flores

C.I.: 1719798826

AGRADECIMIENTOS

A nuestras familias, por darnos el apoyo de seguir adelante en nuestros estudios, de llegar a ser profesionales de calidad para nuestro país, a nuestro maestros y profesores de carrera que nos han enseñado durante todos estos años y prepararnos para el nuevo mundo del trabajo y principalmente a Dios por la salud, vida y sabiduría para poder llegar a nuestras metas como profesionales.

DEDICATORIA

Dedicamos este trabajo a Dios por darnos salud y conocimiento, a nuestros padres por el apoyo incondicional frente a todas los problemas y adversidades, a nuestros hermano/as por los consejos y conocimientos del caso del trabajo de titulación y a nuestros docentes que nos enseñaron y guiaron para que podamos desarrollar un trabajo de excelencia.

RESUMEN

Los sistemas biométricos actualmente se han desarrollado para diferentes usos, estos se han juntado con diferentes tipos de tecnologías para satisfacer las necesidades los usuarios.

Hoy en día gracias a los avances de la tecnología se pueden añadir varios componentes tanto en hardware como en software y combinarlos dando como resultado un sistema biométrico más eficiente, seguro y fácil de montar. Dependiendo de los requerimientos de los usuarios se asigna las funcionalidades al sistema que en este caso es la autenticación de los usuarios para acceder a una sala.

El sistema realiza la función de dar acceso a usuarios autorizados a la sala principal de la Unidad de Innovación Tecnológica (UIDE) de la Universidad de las Américas, un único administrador puede enrolar nuevas huellas digitales para nuevos usuarios, estos al estar registrados en la base de datos pueden abrir la puerta que tiene una cerradura magnética. Además, el administrador puede manejar los usuarios en una página la cual esta guardada en un Raspberry pi 3.

Con la finalidad de tener un sistema sin errores se realiza el análisis, la creación del sistema biométrico en una placa electrónica con sus respectivas pruebas y el análisis de los resultados obtenidos durante este proceso.

ABSTRACT

Biometric systems are currently developed for different uses, these have been combined with different types of technologies to cover the needs of users.

Nowadays, thanks to advances in technology, many components can be added, both in hardware and software, and combine these, resulting in a more efficient, safer and easier to assemble biometric system. Depending on the requirements of the users, the functionalities are assigned to the system, which in this case is the authentication of the users to access a room.

The system performs the function of giving authorized users access to the main room of the Technological Innovation Unit (UIDE) of the University of the Americas, a single administrator can enroll new fingerprints for new users, these being registered in the base Data can open the door that has a magnetic lock. In addition, the administrator can manage the users on a page which is stored in a Raspberry pi 3.

In order to have a system without errors, the analysis is performed, the creation of the biometric system in an electronic board with its respective tests and the analysis of the results obtained during this process.

ÍNDICE

INTRODUCCIÓN	1
1. CAPÍTULO I. ANÁLISIS DE TECNOLOGÍAS	2
1.1. Introducción Capítulo	2
1.2. Sistemas biométricos	3
1.2.1. Huellas Dactilares	3
1.2.1.1. Ópticos Reflexivos	4
1.2.1.2. Ópticos Transmisivos.....	4
1.2.1.3. Capacitivos	4
1.2.1.4. Térmicos	4
1.2.1.5. Salida Dinámica	5
1.2.2. Rostro	5
1.2.3. Termograma del rostro	5
1.2.4. Geometría de la mano	6
1.2.5. Venas de las manos	6
1.2.6. Iris	6
1.2.7. Patrones de la retina.....	7
1.2.8. Voz.....	7
1.3. Lenguajes de programación	8
1.3.1. Lenguaje C	8
1.3.2. Lenguaje C++	9
1.3.3. Lenguaje C#.....	9
1.4. Manejo de bases de datos	10
1.4.1. Servidores de Bases de Datos.....	10
1.4.1.1. MySQL	11
1.4.1.2. SQL Server	11
1.4.1.3. Oracle	12
1.4.1.4. PostgreSQL	13
1.5. Conectividad	15
1.5.1. WLAN	15

1.5.2.	Bluetooth.....	15
1.5.3.	Serial.....	16
1.5.3.1.	Baudrate	16
1.5.3.2.	Bits de Datos.....	16
1.5.3.3.	Bits de Parada	17
1.5.3.4.	Paridad.....	17
1.6.	Dispositivos	17
1.6.1.	Raspberry Pi 3 Model B	17
1.6.2.	Arduino UNO.....	18
1.7.	Materiales Extras.....	19
1.7.1.	Pcb fibra-vidrio.....	19
1.7.2.	Transistores	20
1.7.3.	Transistor de Unión Bipolar.....	20
1.7.4.	Fototransistor	20
2.	CAPÍTULO II. IMPLEMENTACIÓN	20
2.1.	Requerimientos	20
2.2.	Ubicación	21
2.3.	Arquitectura del sistema	22
2.4.	Diagrama de circuitos.....	23
2.5.	Diagramas de flujo	25
2.5.1.	Diagrama general del sistema	25
2.5.2.	Diagrama creación usuarios	26
2.5.3.	Diagrama base de datos.....	27
2.5.4.	Diagrama acceso UITEC	28
2.6.	Requerimientos funcionales	29
2.6.1.	Hardware	29
2.6.1.1.	Raspberry Pi 3 Modelo B	29
2.6.1.2.	Lector de Huella GT-511C3	30
2.6.1.3.	Arduino UNO.....	31
2.6.2.	Software.....	32
2.6.2.1.	Arduino 1.8.5.....	34

2.7.	Configuración de los prototipos	34
2.7.1.	Configuración Prototipo N°1	34
2.7.2.	Configuración Prototipo N°2 y Prototipo Final.....	36
3.	CAPÍTULO III. ANALISIS DE RESULTADOS	39
3.1.	Análisis de costos.....	39
3.2.	Realizar las pruebas.....	40
3.2.1.	Pruebas Prototipo N°1	40
3.2.2.	Pruebas prototipo N°2.....	43
3.2.3.	Pruebas Prototipo N°3	47
3.3.	Análisis de pruebas	49
3.3.1.	Análisis Prototipo N°1	49
3.3.2.	Análisis Prototipo N°2	50
3.3.3.	Análisis Producto final.....	51
3.4.	Resultados Obtenidos	51
4.	CONCLUSIONES Y RECOMENDACIONES	55
4.1.	Conclusiones.....	55
4.2.	Recomendaciones.....	56
	REFERENCIAS.....	58
	ANEXOS	69

ÍNDICE DE FIGURAS

Figura 1. Huella digital con sus partes	4
Figura 2. Biometría por rostro.....	5
Figura 3. Reconocimiento por venas de las manos.....	6
Figura 4. Procedimiento de extracción de voz.....	8
Figura 5. Bases de datos del sistema	12
Figura 6. Oracle tablespaces.....	13
Figura 7. Raspberry pi 3.....	18
Figura 8. Arduino uno.....	19
Figura 9. Transistor de unión bipolar.....	20
Figura 10. Ubicación uitec universidad de las américas sede queri	22
Figura 11. Arquitectura del sistema.....	22
Figura 12. Circuito esquemático proteus 8.....	23
Figura 13. Placa prototipo n°2.....	24
Figura 14. Placa producto final.....	24
Figura 15. Diagrama de flujo general del sistema	25
Figura 16. Diagrama de flujo creación usuarios	26
Figura 17. Diagrama de flujo base de datos.....	27
Figura 18. Diagrama de flujo acceso uitec	28
Figura 19. Raspberry pi 3.....	29
Figura 20. Lector huella gt-511c3.....	30
Figura 21. Arduino uno.....	31
Figura 22. Prototipo de lector de huellas.....	34
Figura 23. Rasphuella.ddns.net.....	35
Figura 24. Acceso a apache2.....	37
Figura 25. Código configuración página ingreso de usuarios.....	38
Figura 26. Métodos en archivo main.js.....	38
Figura 27. Página web rasphuella	39
Figura 28. Sentencias mysql	39
Figura 29. Módulo en la entrada de la sala interior uitec.....	41
Figura 30. Prototipo N°1.....	41
Figura 31. Enrolar nuevas huellas presionando el botón	41

Figura 32. Identificar los usuarios registrados presionando el segundo botón.....	42
Figura 33. Ip consola raspberry	42
Figura 34. Sistema sin caja acrílica	43
Figura 35. Código arduino (1).....	44
Figura 36. Código arduino (2).....	44
Figura 37. Código arduino (3).....	45
Figura 38. Código arduino (4).....	45
Figura 39. Código arduino (5).....	46
Figura 40. Código arduino (6).....	46
Figura 41. Placa prototipo n°2.....	47
Figura 42. Transformador de voltaje 110 (v) a 12 (v)	47
Figura 43. Caja metálica de 12 x 12 cm	48
Figura 44. Comandos de conectividad con el módulo bluetooth	48
Figura 45. Conexión al módulo bluetooth hc-06 desde raspberry	48
Figura 46. Placa final del control de acceso para uitec	49
Figura 47. Cristal de arduino uno	50
Figura 48. Altura para pantalla y lector de huella	51
Figura 49. Distancia entre lcd y lector de huella.....	52
Figura 50. Altura de la caja.....	52
Figura 51. Dimensiones de la caja	53
Figura 52. Parte delantera del sistema.....	53
Figura 53. Parte trasera del sistema (circuito visible).....	54

ÍNDICE DE TABLAS

Tabla 1. Cuadro comparativo de sistemas biométricos	8
Tabla 2. Cuadro comparativo de lenguajes de programación	10
Tabla 3. Cuadro comparativo de sistemas gestores de bases de datos	14
Tabla 4. Cuadro clases de bluetooth	16
Tabla 5. Cuadro de requerimientos	21
Tabla 6. Cuadro de características raspberry pi 3.....	29
Tabla 7. Cuadro de características lector gt-511c3.....	31
Tabla 8. Cuadro de características arduino uno.....	32
Tabla 9. Cuadro de características raspbian.....	32
Tabla 10. Cuadro de características http apache	33
Tabla 11. Cuadro de características arduino 1.8.5.....	34
Tabla 12. Análisis de costos.....	39

INTRODUCCIÓN

Los sistemas biométricos son utilizados en campos de gran trascendencia, como en medicina general, forense, en campos tecnológicos, científicos o de seguridad, este último satisface necesidades empresariales e institucionales, pues realiza el proceso de identificación, registro y verificación de identidad para los miembros que lo conforman.

Actualmente las tecnologías han ido avanzando muy rápido en estos últimos años, en donde se puede aprovechar para aplicarlas de acuerdo a la necesidad de las empresas ya sean grandes medianas o pequeñas (Gestiopolis, 2007).

Hoy en día estas tecnologías son aplicadas orientadas a la necesidades y requerimientos de los usuarios de tal manera que sea notable un avance de la tecnología para estos. Los usuarios actuales prefieren tener conexiones inalámbricas para evitar cables por todo lado, por lo cual las redes se manejan por medio de redes WLAN manejadas con diferentes estándares los cuales se adaptan al requerimiento del usuario final ya sea desde una red local a una red a gran escala. (Semenov, 2005).

El sistema planteado es posible implementar gracias a las tecnologías existentes, es decir, que el biométrico se va a realizar a partir del software y hardware que se tenga disponible para el manejo y administración de los usuarios y así satisfacer la necesidad de la UITEC mencionadas en este documento.

Uno de los temas principales que las empresas siempre tienen en cuenta es el control de la seguridad física y la seguridad de la información de la compañía. Cada empresa emplea su propio sistema de seguridad según las políticas de esta, para ello la tecnología también se ha acoplado para satisfacer los requerimientos de seguridad implementados por cada institución como son los controles de acceso, respaldos de información, registros de personal (Talens-Oliag, 2011). Actualmente la tecnología basada en los sistemas biométricos es la mejor opción para el control de la seguridad y de accesos en las diferentes compañías en donde necesitan tener el registro de quien entra y sale, como

también tener el control de quien puede ingresar a la empresa. Estos sistemas biométricos han sido aplicados en diferentes campos como la medicina general, forense y en los sistemas electrónicos que ya se mencionó. (Tolosa & Giz, 2010)

Este club tecnológico necesita seguridad a nivel de infraestructura, puesto que el acceso hacia el mismo es libre a cualquier persona, propia o ajena a los intereses de la unidad de innovación, arriesgando materiales o dispositivos de gran valor económico y complejidad. La creación de un módulo de acceso (biométrico) determinará un medio de seguridad vital para contrarrestar este problema.

Como primer punto se planea analizar las tecnologías de biometría, base datos, lenguajes de programación C#, java, python, microprocesadores y protocolos de comunicación WLAN y bluetooth.

Seguido de la implementación del sistema de control de acceso biométrico, creando un biométrico con microcontroladores y tecnologías WLAN con standard IEEE 802.11x y módulos bluetooth para las comunicaciones entre equipos.

Finalmente se pasa al análisis de resultados, costos y pruebas de funcionamiento del sistema biométrico.

1. CAPÍTULO I. ANÁLISIS DE TECNOLOGÍAS

1.1. Introducción Capítulo

Los biométricos internamente almacenan datos dentro de una base la cual es administrada por el encargado de TI. Dentro de este punto se necesita siempre de un administrador que maneje los usuarios y datos dentro de una interfaz gráfica y amigable (Díaz, Harari, & Amadeo, 2013). Las tecnologías implementadas dentro de un biométrico tienen que estar interactuando constantemente para su funcionamiento (Figueroa & Villacreses, 2013).

La arquitectura interna de un biométrico consta de varias partes y dispositivos que interactúan entre sí para su correcto funcionamiento.

Los componentes de estos biométricos son módulos, microcontroladores y microprocesadores que hoy en día han avanzado a la par con la tecnología haciéndolos más poderosos y útiles siendo muy pequeños (Espinosa, 2018).

Dentro de este capítulo se explica a fondo cada tecnología, sistemas, circuitos, software y hardware que se va a utilizar para la implementación del biométrico de tal manera que se cumplan con los objetivos propuestos.

1.2. Sistemas biométricos

Los sistemas biométricos son aquellos que realizan las labores de la biometría es decir que se pueda controlar mediante el reconocimiento de una característica única de cada persona a ser verificada (Cruceña, 2009).

Los diferentes tipos de funcionamiento de los dispositivos biométricos son una de las formas más seguras de identificación en donde se utilizan métodos de identificación basado en la biometría que toman las características únicas de cada usuario, estos son:

1.2.1. Huellas Dactilares

Esta técnica toma las características de los patrones de huella digital que puede ser de cualquier dedo de la mano. Con relación al biométrico en el momento que se coloca el dedo en el lector captura imágenes de la huella para luego usarlo como información personal y única del usuario. (Aguilar, Sánchez, Toscano, Nakano, & Pérez, 2008).

La huella digital está formada por una serie de líneas oscuras que representan las crestas y una serie de espacios blancos que representan los valles. La identificación con huellas digitales está basada principalmente en las minucias la ubicación y dirección de las terminaciones de crestas, bifurcaciones, deltas, valles y crestas, aunque existen muchas otras características como se aprecia en la Figura 1 (Galvis, 2007) .



Figura 1. Huella Digital con sus partes

Tomado de (OverBlog, 2011)

1.2.1.1. **Ópticos Reflexivos**

Esta técnica se basa en colocar el dedo en un cristal iluminado por un diodo led en donde la luz se absorbe y en las crestas existe una reflexión en donde la luz de la reflexión y las zonas oscuras son registradas dando como resultado la impresión de la huella (Gómez, 2002).

1.2.1.2. **Ópticos Transmisivos**

La técnica óptico transmisivo al contrario de la anterior no se necesita tener contacto con el cristal ya que la luz del diodo pasa a través del dedo desde la parte superior de la uña tomando una imagen directamente de la huella (Zambrano, 2012).

1.2.1.3. **Capacitivos**

Los capacitivos contienen un circuito integrado cubierto por pixeles con resolución de 500 dpi, de acuerdo con la capacidad entre electrodos se reducen cuando se pone en contacto con el dedo, no se debe poner un dedo sano sin cortadas ni callos (Martínez, 2010).

1.2.1.4. **Térmicos**

El método térmico para capturar las huellas digitales está compuesto por materiales piro eléctricos que funcionan detectando el calor y tomando una imagen en escala de grises al momento del contacto con el dedo (Lara, 2012).

1.2.1.5. Salida Dinámica

La técnica de salida dinámica registra una secuencia de imágenes tomadas de la huella digital en donde el dedo no va a ser colocado y va a estar estático si no que como su nombre lo indica es dinámico y se va a desplazar a lo largo del lector de la huella para luego ser unidas y tener una imagen completa de la huella para luego ser comparada (Chavarrea, 2013).

1.2.2. Rostro

El análisis por reconocimiento facial toma los rasgos del rostro y comparándolo con fotogramas de la cara del usuario.

La comparación se basa en los rasgos de la nariz como el ancho y también la distancia entre los ojos del usuario tomando en cuenta que se estaría comparando con una foto de vista frontal como se puede apreciar en la Figura 2 (Cruz, 2008).

En la actualidad el reconocimiento paso a ser de 2D a 3D en donde se tiene un mayor reconocimiento ya que se basa en submililitros o microondas para ver aspectos faciales más específicos (UPV, 2009).

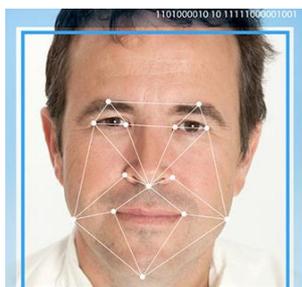


Figura 2. Biometría por Rostro

Tomado de (Cairo Lens, 2017)

1.2.3. Termograma del rostro

Este sistema biométrico utiliza el gráfico del rostro recogido por termografía en donde se analiza el patrón de calor en el rostro y es capturado por cámara de infrarrojo. Estos patrones del rostro tienen como nombre “termogramas” que es una característica única de cada persona, aunque dependiendo de la persona

podría variar estos patrones por el flujo de la sangre en el rostro (Morales V. , 2014).

1.2.4. Geometría de la mano

Estos sistemas se basan en las características de la mano como el ancho y largo de los dedos como también el de la palma y rasgos más definidos específicamente de la mano. Puede variar dependiendo de la condición física del usuario como por ejemplo puede adelgazar como engordar por lo tanto la geometría de la mano cambiara notablemente.(EcuRed, 2012).

1.2.5. Venas de las manos

Este método es considerado como un sistema de seguridad más avanzado que el de la huella dactilar y los anteriores detallados.

El funcionamiento de este sistema se lo realiza por medio de luz infrarroja que puede captar la estructura de las venas de la mano como se puede ver en la Figura 4.

A pesar del paso de los años esta estructura de las venas se mantiene o puede cambiar un poco (Bejerano, 2014).



Figura 3. Reconocimiento por Venas de las Manos
Tomado de (w3layouts, 2016)

1.2.6. Iris

El iris tiene características únicas que frente a los otros métodos de extracción de información única del cuerpo humano este tiene la gran ventaja frente a accidentes al ser resistente gracias a la protección que le otorga la córnea, la característica más importante del iris es que sus características hacen que el

factor de error de un lector sea muy bajo frente al de huella dactilar (Navalpotro, 2014).

Se debe tener en cuenta que existe dos métodos diferentes que están en el ojo del cuerpo humano por lo tanto existe dos tipos de biometría para iris y otro para la retina (González, 2010).

1.2.7. **Patrones de la retina**

Los patrones de la retina son características únicas de cada ser humano que se pueden utilizar como método de autenticación para un sistema biométrico. Este método se basa en los patrones de vasos sanguíneos que están conectados al nervio óptico. El escáner tiene que estar muy cerca de la pupila para poder registrar los patrones. Al igual que el método de venas de las manos el reconocimiento por patrones de la retina es una de las formas más seguras para la implementación de un biométrico (Miroslaba, 2011).

1.2.8. **Voz**

El sistema biométrico por voz maneja factores muy importantes como la acústica, fisiología y ciencia aplicada en la computación.

Para que el sistema pueda autenticar la voz se debe decir una palabra fija y otra que sea diferente ya que con esto se extraen patrones de voz que en todas las personas varía ya sea en su acento, idioma y forma de hablar.

De la voz se puede tener información para la autenticación única, pero se tiene en cuenta ciertas especificaciones como que palabra se menciona siempre, los distintos tonos, acentos de la voz y la más avanzada es la forma de hablar. Este sistema se centra directamente en el tracto vocal, la laringe, boca y nariz (Tolosa & Giz, 2010).

Como se puede apreciar en la Figura 5 el procedimiento de la extracción de voz parte de la captura, extracción de las características, entrenamiento del modelo y se tendrá como resultado un modelo de voz único.

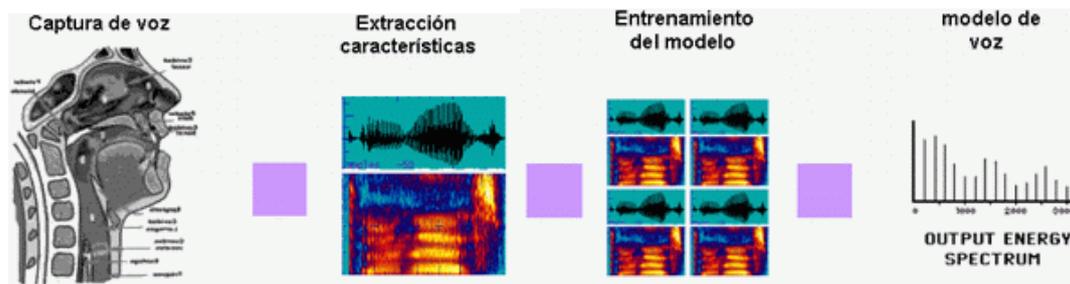


Figura 4. Procedimiento de extracción de voz

Tomado de (Paz, 2009)

Tabla 1.
Cuadro Comparativo de Sistemas Biométricos

Sistema	Fiabilidad	Seguridad	Aceptación	Captación de Datos	Precio
Huella Digital	Alta	Media	Media	Media	Medio
Rostro	Baja	Alta	Alta	Alta	Medio
Termograma de Rostro	Media	Alta	Alta	Alta	Alto
Geometría de la Mano	Media	Media	Media	Alta	Medio
Venas de las Manos	Media	Alta	Media	Media	Alto
Iris	Alta	Alta	Baja	Media	Alto
Patrones de retina	Alta	Alta	Baja	Baja	Alto
Voz	Baja	Media	Alta	Media	Medio

Adaptado de (Cano J. L., 2014)

1.3. Lenguajes de programación

1.3.1. Lenguaje C

El lenguaje de programación en C fue desarrollado en 1972 para Unix, es considerado un lenguaje de alto nivel que simplifica algunas funciones. También se puede programar en bajo nivel para interactuar con registros, memorias las cuales se pueden definir en librerías (Rena, 2015).

Este lenguaje está estructurado en tres partes:

- Librerías: aquellas que deben ser declaradas con # include.
- Compilador: aquel que traduce instrucciones entre el programa y la computadora es decir lo pasa de instrucciones a lenguaje de máquina.
- Preprocesador: traduce instrucciones del programa, incluye el contenido de las librerías declaradas, estos archivos son llamados cabeceras.

Se puede asociar a bases de datos para almacenar información mediante sentencias SQL.

1.3.2. Lenguaje C++

Es un lenguaje creado en 1983 en base a lenguaje C, este lenguaje no soportaba los objetos y las clases. El lenguaje C++ nace por las necesidades de comprensión e interacción con el usuario, a partir de esto nace el paradigma de los objetos y sus características únicas dentro de clases. Al crear C++ se creó también la capacidad de manejar objetos (Wordpress, 2015).

1.3.3. Lenguaje C#

Este es un lenguaje orientado objetos con mejores del lenguaje en JAVA dando como resultado la unión de C++ y las ventajas de JAVA. Maneja el modelo de orientación a objetos de .NET.

Este lenguaje es más visual por lo cual se maneja con distinto software como Visual Basic, Visual Studio con los cuales se puede manejar y crear Querys para realizar consultas desde el mismo programa teniendo una conexión a bases de datos como SqlServer, MySql, Oracle, Postgrest etc. (Microsoft, 2015)

Al utilizar estos programas se pueden crear formularios los cuales van a interactuar directamente con el usuario en donde este podrá agregar, eliminar, buscar y actualizar datos con lo que mencionamos antes el manejo de consultas internamente. Se considera más simple que el lenguaje C++ pero más útil de cierta forma.

Tabla 2.
Cuadro Comparativo de Lenguajes de Programación

Lenguaje	Características	Ventajas	Desventajas
Python	- Creación de sitios web y programas - No necesita compilación	- Código Libre - Multiplataforma - Orientado a Objetos	- Suele ser lento al ser un lenguaje interpretado
C	- Utilizado para la creación de programas de sistema	- Lenguaje utilizado para la implementación de SO como Unix	- No utilizado para la creación de aplicaciones
C++	- Lenguaje Orientado a objetos	- Multiplataforma - Se utiliza para creación de sistemas robustos	- Al igual que el lenguaje C no puede crear aplicaciones Web - Lenguaje Complejo
C#	- Es parte de la plataforma .net de Microsoft	- Su sintaxis es parecida a C y C++ - Puede crear aplicaciones web, móviles y de escritorio	- Se necesita un espacio considerable en memoria para su instalación

Adaptado de (Cano A. , 2014)

1.4. Manejo de bases de datos

1.4.1. Servidores de Bases de Datos

Los servidores de bases de datos se utilizan para administrar, almacenar y en algunos casos en recuperar los datos de una base, estos servidores deben resguardar los datos de manera de segura de tal manera que se garantice la integridad de estos.

Estos servidores se usan para diferentes tipos de uso como gestión de archivos, servidores de correo electrónico entre otros.

Para los servidores web que se usan para alojar páginas web en donde se puede agregar contenido en sitios web dinámicos un claro ejemplo es el ingreso de usuario y contraseña en una página de estas que se guardara en la base de datos.

Otro punto por tomar en cuenta es el tamaño para un servidor de base de datos, esto va a depender de la frecuencia con que se van a realizar las consultas, el número estimado de bases de datos se van a crear en un futuro (Microsoft, 2017).

Los sistemas de gestión de base de datos en la actualidad son:

1.4.1.1. **MySQL**

MySQL es uno de los softwares más famosos dedicado a la gestión de bases de datos relacionales, es de código abierto es decir que es gratuito su uso, su lenguaje de programación es en SQL. Este software soporta varios tipos de datos para las tablas y se puede manejar en diferentes sistemas operativos. Tiene un buen nivel en seguridad de los datos y en cada base que es creada contiene tres archivos, un archivo de datos, de índice y de estructura. La principal característica de MySQL es su rapidez para resolver operaciones tales como transacciones y consultas (Rouse, MySQL, 2013).

Se adapta a diferentes entornos de desarrollo para interactuar con lenguajes de programación como el famoso PHP, Java entre otros.

1.4.1.2. **SQL Server**

Es un sistema de gestión de bases de datos desarrollado por Microsoft, utiliza un procesamiento de transacciones en línea y procedimientos almacenados. Para una correcta administración se tiene un entorno gráfico. (Rouse, TechTarget, 2015)

También tiene una versión en la nube asociada con Azure en donde se paga mensualmente para mantener el servicio activo en la nube ahorrando gastos en equipos físicos. (Rouse, TechTarget, 2015)

Las bases de datos por defecto creadas por los desarrolladores son:

- *Master*: En donde se guardan los procedimientos, tablas del sistema que utilizan todas las bases de datos creadas. (Wikipedia, 2017)

- *MsdB*: En esta base de guardan las tareas, los códigos CLR, paquetes entre otros componentes.
- *Tempdb*: Esta es una base que se crea cada vez que se inicializa el servicio, esta se almacena tablas temporales creadas por el sistema y por los mismos usuarios (Wikipedia, 2017).
- *Model*: Es el modelo de cada base de datos nueva, es decir todas las bases de datos creadas saldrán con este modelo de base de datos.

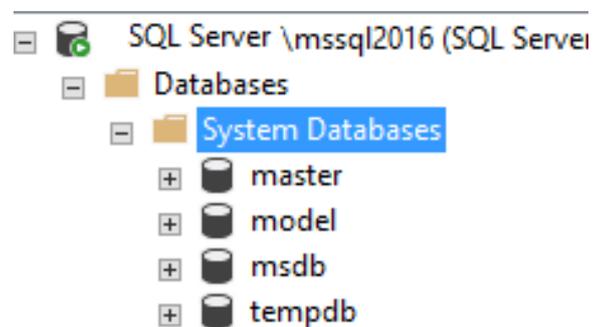


Figura 5. Bases de Datos del Sistema

1.4.1.3. Oracle

Es un sistema de gestión de base de datos relacionales de igual manera basada en cliente – servidor como SQL Server. Oracle como compañía ha desarrollado Oracle DB para el manejo de datos a gran escala para las distintas compañías alrededor del mundo.

Con respecto a la estructura de las bases de datos de Oracle se tiene dos partes, una lógica y una física.

- Estructura Lógica: está conformada por los “Tablespaces” por uno o varios ficheros de datos y los objetos para las bases de datos como tables, procedimientos, vistas e índices (iessanvicente, s.f.).
- Estructura Física: los ficheros de datos dentro de una base de datos son de tamaño fijo y se crean junto con la base y las “Tablespaces”. Estos ficheros son almacenados en una caché

compartida para cada nuevo acceso en los que van a ser leídos rápidamente (iessanvicente, s.f.).

Una de grandes limitaciones que tiene Oracle es que tiene una instalación de gran complejidad ya que a diferencia de MySQL este no tiene un instalador.

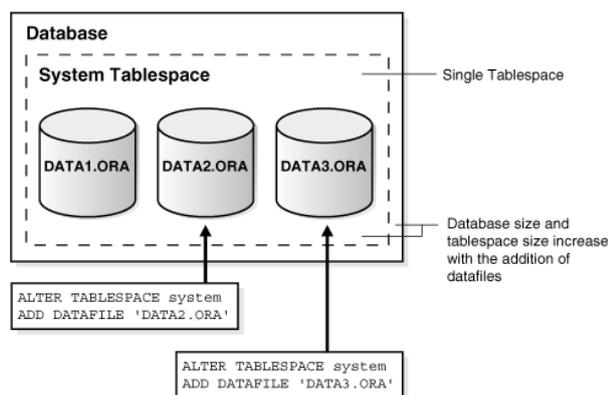


Figura 6. Oracle Tablespaces

Tomado de (Oracle, 2005)

1.4.1.4. PostgreSQL

PostgreSQL al igual que MySQL es de uso gratuito, maneja el control de concurrencias en multiversiones en donde agrega en cada transacción una imagen de lo que es el estado de la base de datos.

Tiene un parecido con MySQL ya que utiliza un administrador llamado *pgAdmin* en donde se manejan y administran todas las bases creadas en PostgreSQL, también se pueden realizar consultas todo esto en un entorno amigable y seguro. Se pueden usar también vistas, índices y reglas dentro de la base (Dorantes, 2015).

Este gestor de base de datos maneja las siguientes características:

- DBMS Objeto-Relacional: PostgreSQL aproxima los datos a un modelo objeto relacional, y es capaz de manejar complejas rutinas y reglas.

- Su avanzada funcionalidad son las consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arrays.
- Herencia: Las tablas pueden ser configuradas para heredar características de una tabla padre. Los datos son compartidos entre las tablas padre e hija(s). Las tuplas insertadas o eliminadas en la tabla hija serán insertadas o eliminadas en la tabla padre respectivamente.
- Altamente Extensible: PostgreSQL soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario.

Tabla 3.
Cuadro Comparativo de Sistemas Gestores de Bases de Datos

Sistema Gestor de Base de Datos	Características	Ventajas	Desventajas
MySQL	<ul style="list-style-type: none"> - Desarrollado en C y C++ - Reserva memoria - Sistema seguro de privilegios y contraseñas 	<ul style="list-style-type: none"> - Portabilidad - Manejo en distintos sistemas operativos. 	<ul style="list-style-type: none"> - No soporta algunas conversiones de datos.
SQL Server	<ul style="list-style-type: none"> - Tiene herramientas muy útiles - Recuperación de los datos rápida - Portabilidad 	<ul style="list-style-type: none"> - Tienes soporte para transacciones - Se pueden utilizar procedimientos almacenados 	<ul style="list-style-type: none"> - Utiliza demasiados recursos de CPU y RAM
Oracle	<ul style="list-style-type: none"> - Sistema Gestor Relacional de Base de datos para distintas plataformas 	<ul style="list-style-type: none"> - Puede utilizar particiones para administrar las bases de datos 	<ul style="list-style-type: none"> - Precio alto
PostgreSQL	<ul style="list-style-type: none"> - Soporta estructuras de arrays - Maneja la herencia entre las tablas 	<ul style="list-style-type: none"> - Puede gestionar diferentes bases de datos 	<ul style="list-style-type: none"> - Para insertar y actualizar datos es lento

Adaptado de (Florencio, 2017)

1.5. Conectividad

1.5.1. WLAN

Es una tecnología de comunicación inalámbrica que es la más utilizada en la actualidad, esta tecnología wireless-lan es llamada también por el estándar de IEEE 802.11 en la que se definen el uso de las capas 1 y 2 del modelo OSI que corresponde a las capas física y de enlace de datos. (US, 2014).

Wlan utiliza ondas de radio que son enviadas de un punto o dispositivo a otro por medio del aire sin necesidad de cables físicos en donde los datos que son enviados se superponen a la portadora de radio para que así el receptor recoja esta información. En el medio existen muchas ondas con distintas frecuencias de radio por lo cual el receptor necesitar estar en una frecuencia específica para recibir los datos enviados (GSITIC, 2018).

1.5.2. Bluetooth

Es un protocolo de comunicación inalámbrico a corta distancia que en comparación con la comunicación de infrarrojo no se necesita estar alineado sino estar a una cierta distancia para poder enlazar dispositivos de forma inalámbrica (Bellido, De la Cruz, Torres, & Gistas, 2004).

Los sistemas Bluetooth tienen un alcance máximo hasta que se pierda el enlace, este alcance puede variar en tres clases detallado en la Tabla 4.

Las clases son asignadas dependiendo de las necesidades como por ejemplo los dispositivos auriculares por bluetooth son de clase 2 o de clase 3 (Morales N. , 2017).

Bluetooth está compuesto por un dispositivo que genera ondas de radio que también modula y envía estas señales, luego un controlador digital procesa las señales para que otro dispositivo reciba estas ondas (Morales N. , 2017).

Actualmente a cualquier dispositivo se le puede integrar Bluetooth para transmitir datos.

Tabla 4.
Cuadro Clases de Bluetooth

Clase	Distancia	Potencia
1	100 (m)	100 mW
2	25 (m)	2,5 mW
3	1 (m)	1 mW

Adaptado de (Morales N. , 2017)

1.5.3. **Serial**

Este tipo de comunicación serial es un protocolo con el cual se comunican 2 dispositivos por medio del envío de datos un bit a la vez sobre un canal, es decir envía y recibe por bit (National Instruments, 2006).

Este tipo de comunicación se utiliza para transmitir datos en formato ASCII que es el Código Estándar Estadounidense para el intercambio de Información en donde se necesitara siempre tres líneas de transmisión:

1. Transmisor (Tx)
2. Receptor (Rx)
3. Tierra (GND)

La comunicación serial es asincrónica por lo cual se puede invitar y recibir al mismo tiempo por distintas líneas de transmisión. Se debe tener en cuenta ciertos parámetros que son importantes como son:

1.5.3.1. **Baudrate**

Es la velocidad de transmisión en donde se indica el número de bit por segundo en el que se transfieren (National Instruments, 2006).

1.5.3.2. **Bits de Datos**

Se refiere a la cantidad de bits de transmisión, estos bits dependen de qué tipo de información se transmita (National Instruments, 2006).

1.5.3.3. Bits de Parada

Los valores de parada de bits se usan para indicar el fin de la comunicación de cada paquete, adicionalmente cada dispositivo tiene diferentes tiempos marcados por reloj con lo cual estos bits de parada marcan un cierto tiempo de tolerancia para los diferentes relojes de estos dispositivos (National Instruments, 2006).

1.5.3.4. Paridad

Con la paridad se puede verificar si hay errores en la transmisión serial. La paridad se puede clasificar en:

- Par e Impar: se fija un bit de paridad que es aquel que va luego de los bits de datos que definimos anteriormente para asegurar un bit par o impar y mantener la transmisión en estado alto lógico.
- Marcada y Espaciada: fija el bit de paridad en estado lógico bajo para la paridad Espaciada y en lógico alto para la Marcada con lo cual se puede dar a conocer si existe o no ruido que afecte a la transmisión (National Instruments, 2006).

1.6. Dispositivos

1.6.1. Raspberry Pi 3 Model B

Raspberry es un minicomputador hecho en placa de 85 x 54 en el cual se pueden implementar tareas como levantar servicios, desarrollar programas, juegos y sin fin de aplicaciones y proyectos sin gastar muchos recursos como en una computadora de escritorio o laptop.

Entre sus características principales es su bajo costo, su reducido tamaño y el gran número de aplicaciones que le puedes dar.

Los componentes del raspberry para conexión son:

- HDMI: se puede conectar mediante un cable Hdmi a un monitor o Tv donde nos reflejara una interfaz gráfica.

- Puertos USB 2.0: tiene 4 puertos USB en donde se puede utilizar para conectar dispositivos externos.
- Puerto 10/100 Ethernet: Tiene un puerto ethernet con el cual se puede conectar a una red por medio de un cable RJ-45.
- Slot Micro SD: Contiene un slot para tarjetas de memoria Micro SD en donde se instalará el sistema operativo y demás características.
- Wifi: puede conectarse vía Wifi, tiene el estándar 802.11n.
- Bluetooth: en el caso del Raspberry pi 3 la versión es la 4.1

El sistema operativo que puede soportar pueden ser DEBIAN que adaptado para raspberry se hace llamar RASBIAN, también PIDORA entre otros. Este sistema operativo debe estar preinstalado con NOOBS.

Cuando se inicia un raspberry se debe especificar qué tipo de S.O es que se quiere manejar.



Figura 7. Raspberry Pi 3

Tomado de (Lestoc, 2016)

1.6.2. **Arduino uno**

El Arduino UNO es una placa basado en microcontrolador Atmega328, tiene 14 pines de entrada como de salida en los cuales se realizan proceso lógicos y matemáticos. Este consta de algunos componentes como son:

- Cámaras VGA
- Sensores de sonido

Se pueden acoplar una gran variedad de dispositivos como también se puede acoplar al Raspberry Pi 3 que se menciona en el punto anterior.

Aparte de ser una placa con componentes eléctricos combina los mismo con el lenguaje de programación para darle mayor funcionalidad.

Este lenguaje para Arduino se llama "Wiring" en el cual se maneja lenguaje de programación C y C++. Tomando como base el lenguaje C de este se han derivado muchos que han ido evolucionando con el pasar de los años como son C#, Java, Python etc.

Para poder interactuar entre el Arduino con el lenguaje de programación se necesita de un software para el sistema que se esté utilizando este se llama Arduino 1.0.3.

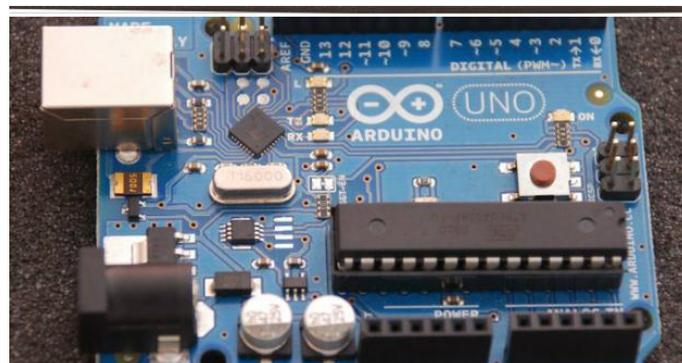


Figura 8. Arduino UNO

Tomado de (Prometec, 2017)

1.7. Materiales Extras

1.7.1. Pcb fibra-vidrio

PCB es una placa que soporta ciertas conexiones eléctricas entre distintos componentes como pueden ser, resistencias, condensadores, transistores, baterías entre otros. Existen dos clases de placas fibra – vidrio, de simple cara y doble cara.

Esto se refiere a que las simples son placas impresas en un solo lado y doble en ambos lados, esto depende mucho del uso que se le va a dar a dicha placa.

1.7.2. Transistores

Son dispositivos electrónicos usados para entregar señales de salida en respuesta a una señal de entrada.

Puede cumplir varias funciones como:

- Conmutador
- Oscilador
- Amplificador

Existen distintos tipos de transistores entre estos:

1.7.3. Transistor de Unión Bipolar

Están fabricados sobre un mono cristal hecho de componentes semiconductores, es decir, es conductor y aislante.

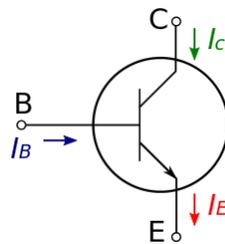


Figura 9. Transistor de Unión Bipolar

Tomado de (TOGELL, 2015)

1.7.4. Fototransistor

Son transistores regulados por la luz, son sensibles a la radiación electromagnética. Puede utilizarse como un transistor normal y un fototransistor en modo de iluminación.

2. CAPÍTULO II. IMPLEMENTACIÓN

2.1. Requerimientos

Los requerimientos del proyecto, por parte de la Unidad de Innovación Tecnológica (UITEC) de la Universidad de las Américas, son presentados a continuación:

Tabla 5.
Cuadro de Requerimientos

REQUERIMIENTO	DESCRIPCIÓN	FUNCIONALIDAD
LECTOR	Lector Huella	Lector de Huella para habilitación de puerta de la sala 1 (sala de reuniones y armario de Robótica).
COMUNICACIÓN	Inalámbrica (Bluetooth, Wi-Fi) LAN servicios	Comunicación entre el módulo de lector de huella y el servidor a través de medio inalámbrico, igualmente para acceso a puerta principal desde sala de reuniones. Conexión LAN del servidor en red UDLA.
USUARIOS	Base de Datos	Administración y gestión de usuarios a través de un servidor de base de datos. Alrededor 4 usuarios con todos los privilegios y 30 usuarios limitados.
SOFTWARE	Aplicación JAVA o HTML	Software control a tiempo real de usuarios y notificación mediante correo o aplicación de las personas que ingresan a la sala de reuniones

2.2. Ubicación

La Unidad de Innovación Tecnológica (UITEC) se encuentra en la Sede Queri, de la Universidad de las Américas, como se muestra en la figura 11. Dentro del bloque 6, último piso. Anteriormente se encontraba en el segundo piso, frente a los laboratorios de biotecnología, actualmente está en el tercer piso junto a la Sala de Docentes:

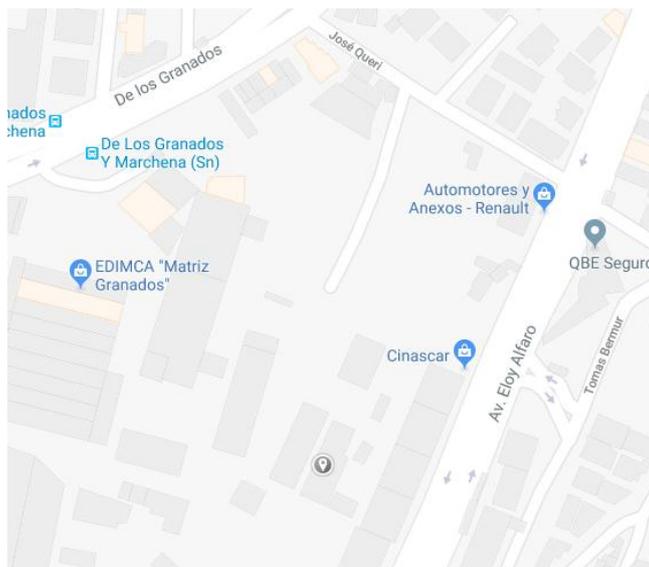


Figura 10. Ubicación UITEC Universidad de las Américas Sede Querí

2.3. Arquitectura del sistema

El sistema biométrico está compuesto por: lector de huella dactilar, microcontrolador Arduino, Raspberry Pi 3 el cual almacena el servidor Web y base de datos, sistema de habilitación de puerta de UITEC, red LAN y WAN, previamente especificados en los requerimientos. Reflejados en el diagrama de la Figura 12, que se muestra a continuación:

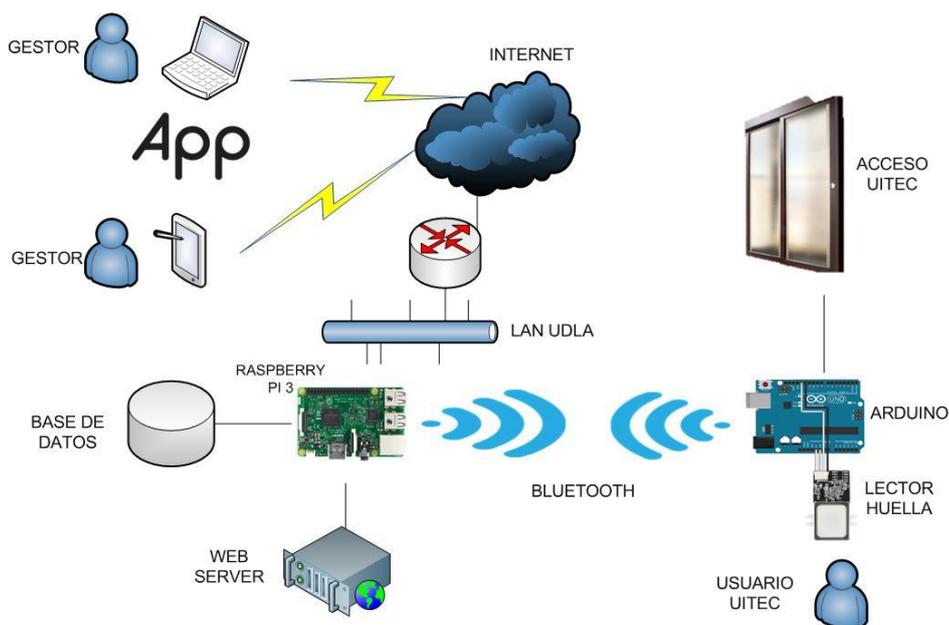


Figura 11. Arquitectura del Sistema. [Diagrama]

En este diagrama se evidencia el sistema, dividido en las siguientes zonas: Web Server, Base de Datos, Lector de Huella y Acceso a UITEC.

- Un gestor accede a la base de datos donde se encuentran los diferentes perfiles de usuario, con su información.
- El acceso es a través de una App Web, por ende, a través de la red, desde cualquier sitio de la universidad, con la IP asignada seguido de /rasphuella
- La base de datos y Web Server, se encuentran en el módulo Raspberry Pi 3, el cual se instancia en la red interna (LAN) de la Universidad de las Américas (UDLA).
- El módulo de lector de huella, en Arduino, se localiza en la puerta de acceso a la sala 1 (sala de reuniones), en el interior de la UITEC.
- La comunicación del módulo con el servidor de Raspberry Pi 3 es a través de tecnología inalámbrica, en este caso Bluetooth.
- Una vez que, un usuario registrado de la Unidad de Innovación Tecnológica intente acceder a la locación podrá desbloquear la chapa magnetica.

2.4. Diagrama de circuitos

Por medio del programa Proteus 8 se realizó el circuito esquemático para posteriormente crear las placas que más adelante se explicará más detalladamente en cada prototipo implementado.

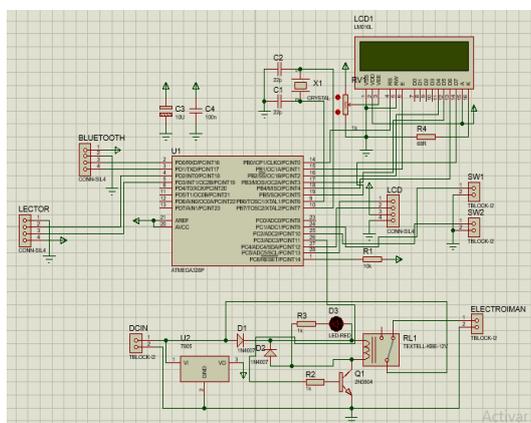


Figura 12. Circuito Esquemático Proteus 8

Como se puede visualizar, en el circuito cada componente está conectado al microcontrolador ATMEGA 328P de tal manera que se podrá realizar la placa de distintas formas para cumplir con los requerimientos.

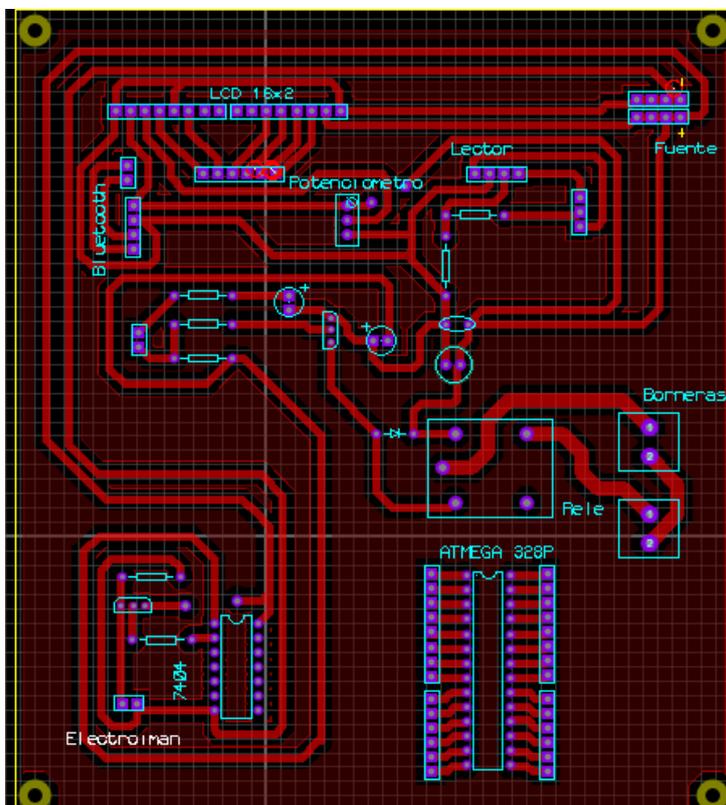


Figura 13. Placa Prototipo N°2

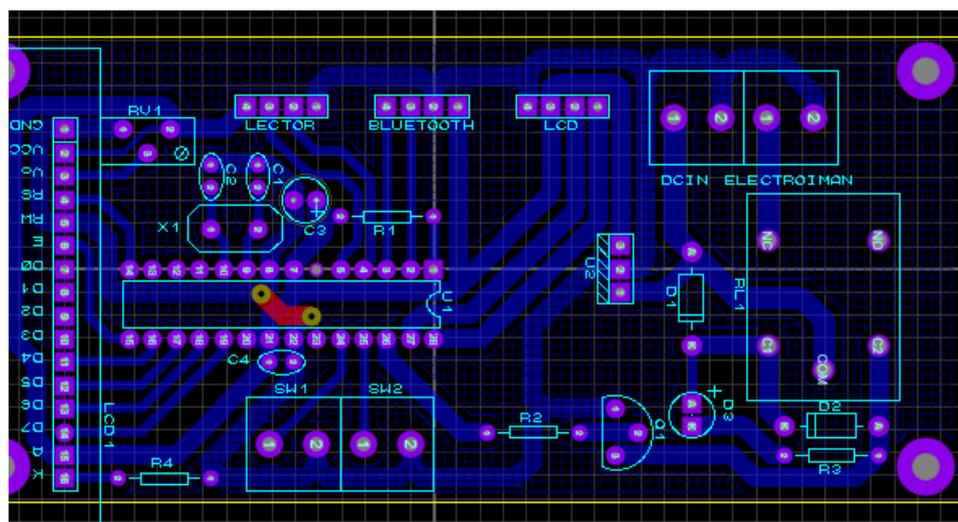


Figura 14. Placa Producto Final

2.5. Diagramas de flujo

2.5.1. Diagrama general del sistema

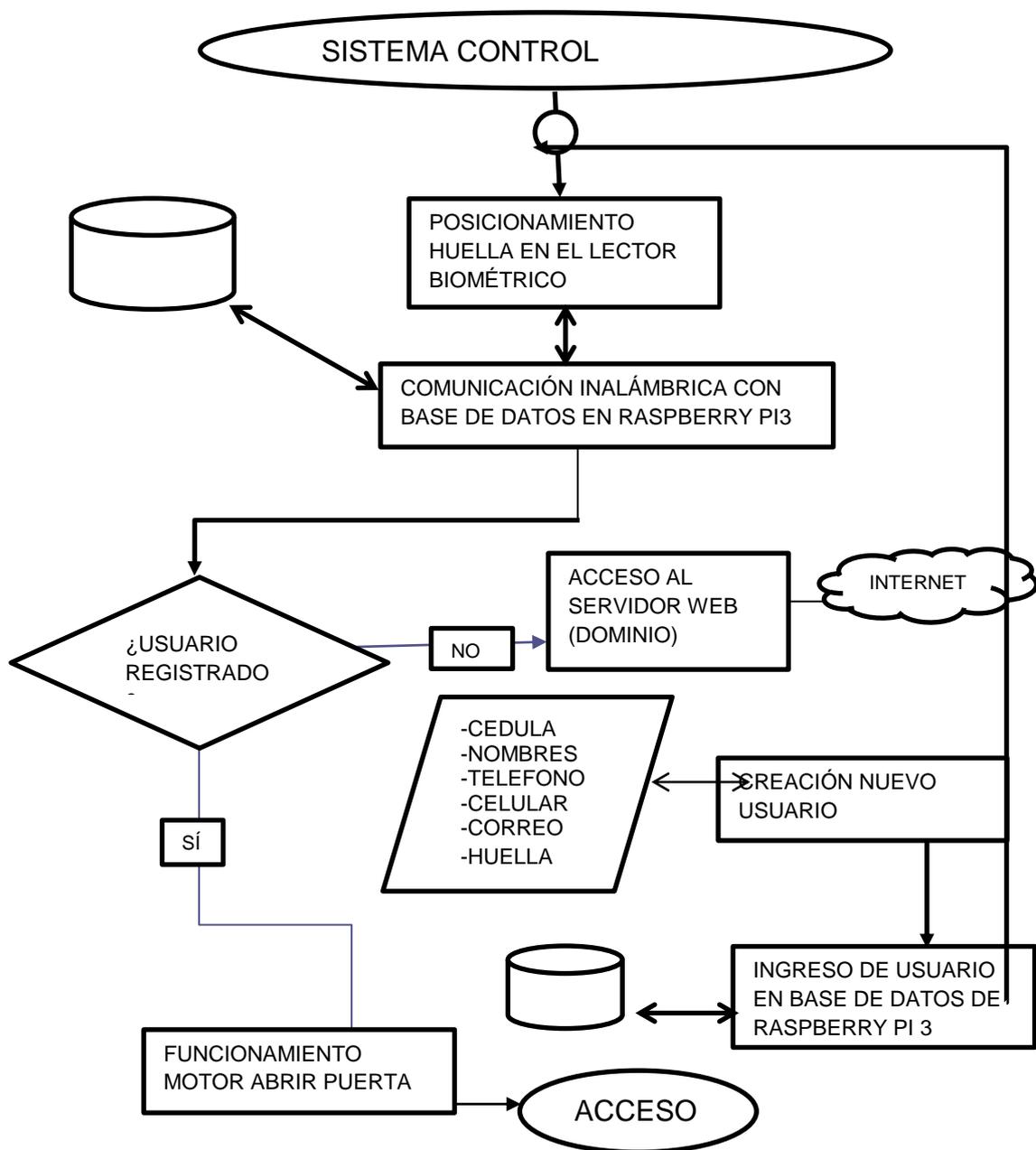


Figura 15. Diagrama de flujo general del sistema

- Un usuario intenta acceder a la Unidad de Innovación Tecnológica, por ende, posiciona su huella dactilar en el lector biométrico.
- El lector recibe la señal y realiza una consulta inalámbricamente al servidor de base de datos del Raspberry.

- Si el usuario no se encuentra registrado, se despliega el mensaje y el gestor creará el usuario correspondiente a través de internet, ingresando datos a la base, posteriormente realizará el mismo proceso de posicionar la huella y si el este se encuentra registrado, el mecanismo de acceso habilita la puerta a la sala de reuniones hacia la sala de la UITEC.

2.5.2. Diagrama creación usuarios

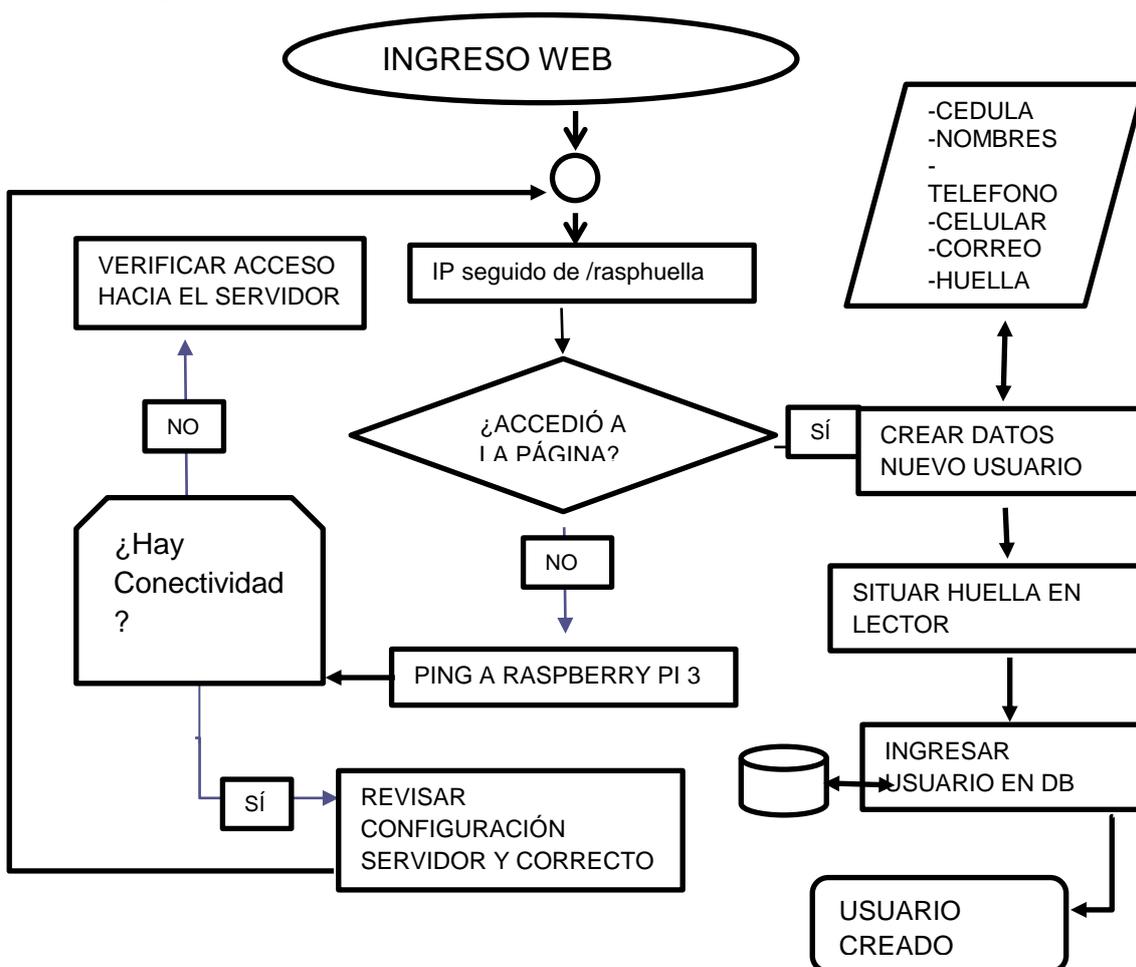


Figura 16. Diagrama de flujo Creación Usuarios

- Para la creación de usuarios, el gestor ingresa al web server a través de la URL: rasphuella.ddns.net por cualquier dispositivo conectado a internet.
- En caso de no poder acceder a la página web, se revisa conectividad al Raspberry Pi 3 desde el interior de la UDLA.

- Si existe conectividad, se valida la configuración del servidor Web y funcionamiento de este, posteriormente probar el ingreso a la URL respectiva y se crea el usuario con los datos específicos.

2.5.3. Diagrama base de datos

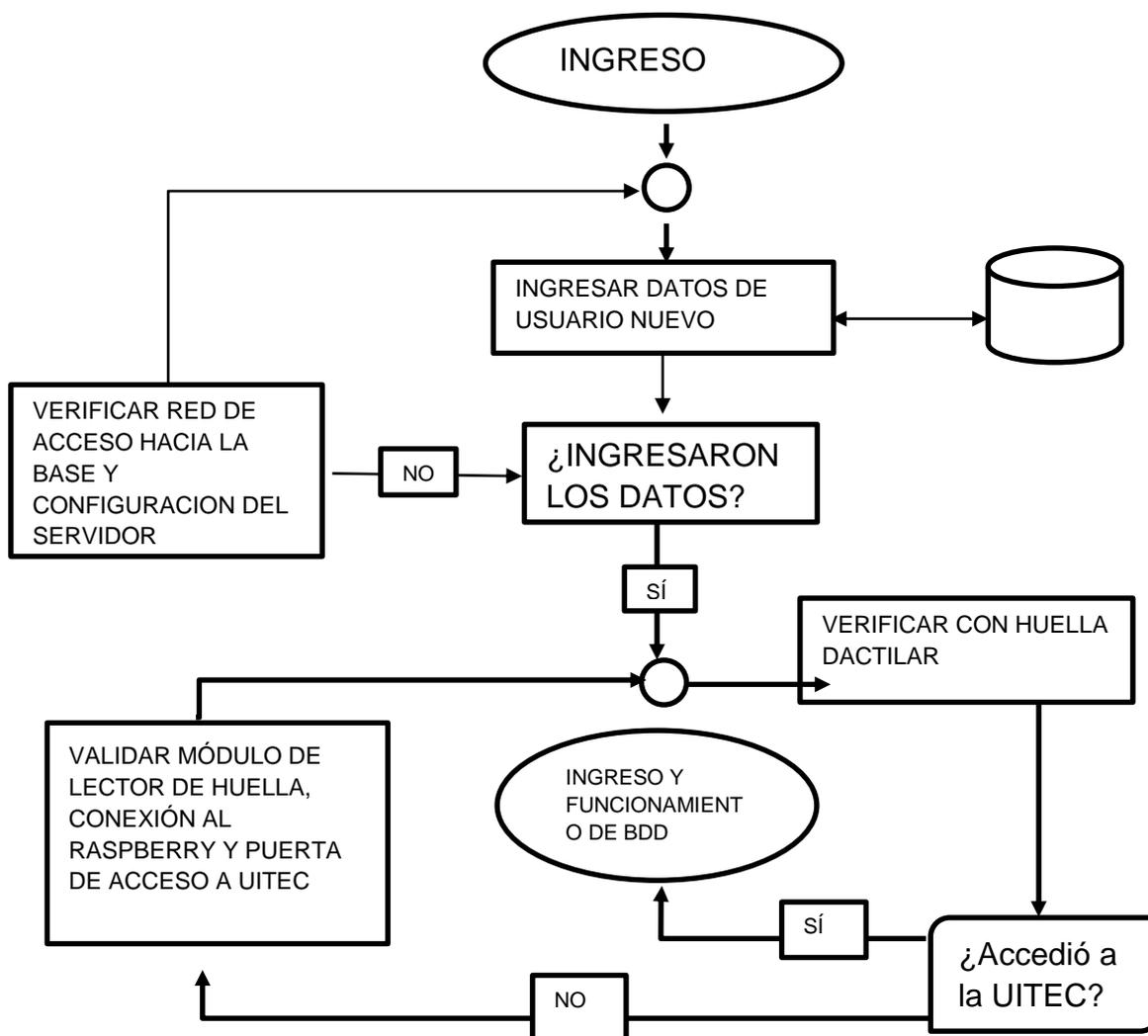


Figura 17. Diagrama de Flujo Base de Datos

- El proceso de validación de base de datos es parecido al Web Server, se inicializa con el acceso a la página web a través de URL y se ingresan datos.
- En caso de no agregarse la información, se verifica el acceso al servidor de base de datos (conectividad) y el proceso de configuración y funcionamiento de este.

- Si es que se visualiza el usuario creado, se procede a acceder a la UITEC, en caso de no habilitarse la puerta, validar la conexión del lector y motor de acceso.

2.5.4. Diagrama acceso UITEC

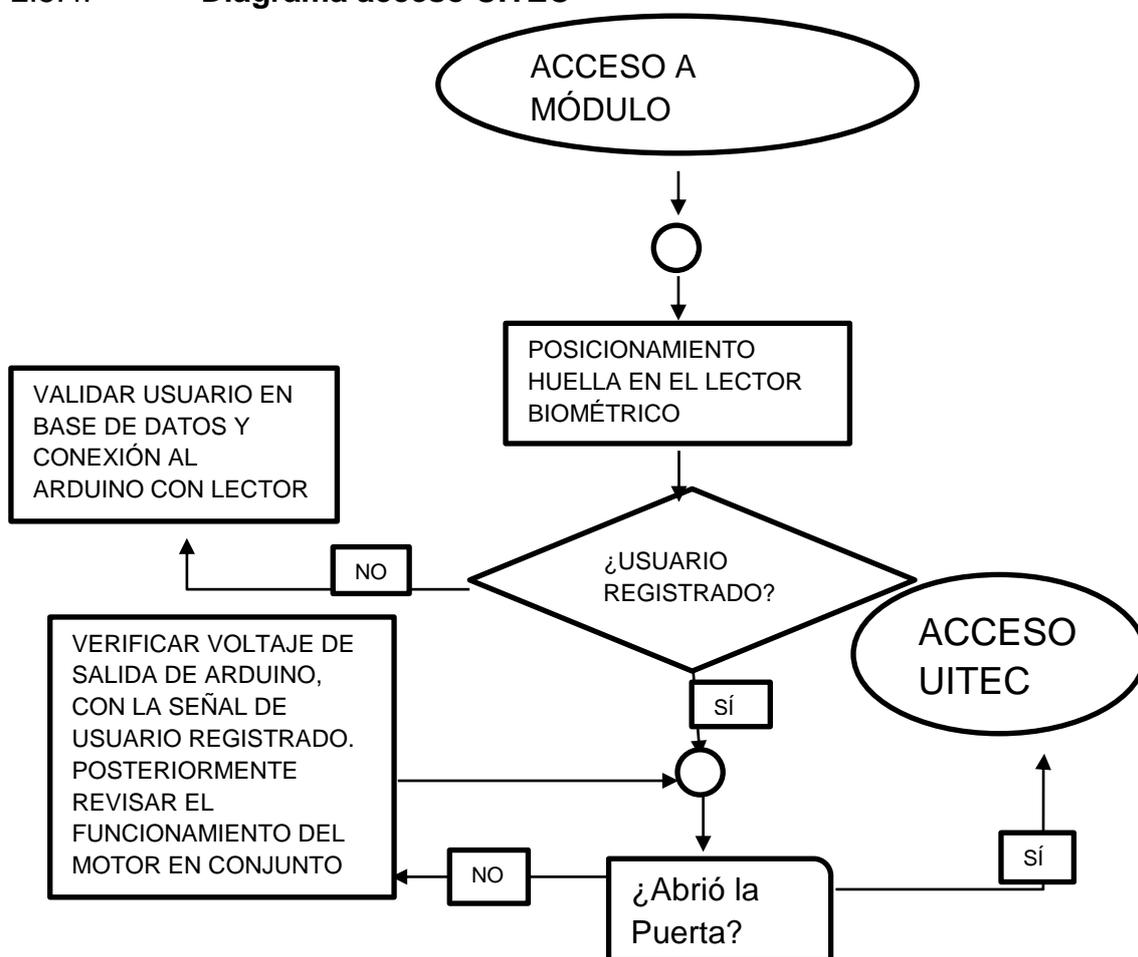


Figura 18. Diagrama de flujo Acceso UITEC

- El método de acceso a través de la puerta corrediza es través de un sistema magnético o por motor, el usuario que intenta ingresar posiciona su huella en el lector biométrico.
- En caso de no existir registro del usuario, se verifica el usuario en el servidor Web y base de datos.
- Si el usuario se encuentra registrado el mecanismo de acceso debe habilitar la puerta y permitir el paso a la sala de reuniones, caso contrario, se valida el voltaje de salida del microcontrolador,

igualmente el funcionamiento del motor o sistema magnético en conjunto con la puerta.

2.6. Requerimientos funcionales

2.6.1. Hardware

2.6.1.1. Raspberry Pi 3 Modelo B



Figura 19. Raspberry Pi 3

Tomado de (Feed RSS Reader, 2017)

Tabla 6.

Cuadro de Características Raspberry pi 3

Característica	Aplicación
<ul style="list-style-type: none"> • Chipset Broadcom BCM2387 • Arquitectura CPU de 64 bits • 1 GB de memoria RAM. 	<p>El servidor WEB y base de Datos requerido, es gestionado mediante el computador Raspberry Pi 3, puesto que posee características óptimas para realizar proyectos a la altura de procesos micro- empresariales. Esta versión de Raspberry es 10 veces mejor que sus predecesores Raspberry Pi Modelo B+ y Raspberry Pi 2 Modelo B.</p>

<ul style="list-style-type: none"> • 802.11 bgn Wireless LAN y Bluetooth 4.1 	<p>Conectividad LAN inalámbrica y Bluetooth, los cuales son vitales para la elaboración de proyectos de Tesis sobre conectividad entre módulos, en este caso Biométrico.</p>
<ul style="list-style-type: none"> • Puerto microSD para cargar su sistema operativo y almacenar datos 	<p>Posee memoria considerablemente amplia, a través de microSD, por lo cual almacena datos de gran importancia, como bases de datos.</p>
<ul style="list-style-type: none"> • Procesador 1.2GHz Quad-Core ARM Cortex-A53 • 4 puertos USB • Conector Ethernet 10/100 BaseT <p>(BangGood, 2018)</p>	<p>Existen estándares de conexión que facilitan la comunicación y transferencia de datos, tales como puertos USB o tecnología Ethernet. Al ser un procesador de 1.2 GHz de cuatro núcleos individuales (Quad-core) evidencia potencia suficiente para la elaboración del proyecto.</p>

Adaptado de (alaMaula, 2017)

2.6.1.2. Lector de Huella GT-511C3



Figura 20. Lector Huella GT-511C3

Tomado de (Generation Robots, 2017)

Tabla 7.
Cuadro de Características Lector GT-511C3

Característica	Aplicación
<ul style="list-style-type: none"> • Lee y escribe plantillas de huellas dactilares y maneja la identificación de huellas dactilares de alta precisión y alta velocidad 	Se puede administrar hasta 200 plantillas de huellas para su uso dentro de una aplicación.
<ul style="list-style-type: none"> • Protocolo simple de comunicación UART y USB 	Maneja protocolos y estándares UART y USB, serial hacia cualquier tipo de microcontrolador u ordenador para el procesamiento de datos.
<ul style="list-style-type: none"> • Módulo de chip, con algoritmo de huella digital y sensor óptico 	Posee un sensor óptico y una pequeña CPU de arquitectura de 32 bits, que simplifica el funcionamiento de plantillas dactilares y su transferencia vía serial al microcontrolador deseado

Adaptado de (SparkFun, 2016)

2.6.1.3. Arduino UNO



Figura 21. Arduino UNO

Tomado de (Prometec, 2017)

Tabla 8.
Cuadro de Características Arduino UNO

Característica	Aplicación
<ul style="list-style-type: none"> Tiene 14 pines digitales de entrada y salida 	Gran recepción de una señal, para procesarla y enviarla a través de medios inalámbricos y por cable, a un servidor de cualquier índole, posee gran compatibilidad, con el lector de huella GT-511C3, el cual recibe una plantilla por parte de la misma.
<ul style="list-style-type: none"> 6 pines se pueden usar como salidas PWM Una conexión USB 	Posee pines para una gran variedad de aplicaciones, y estándares a nivel de conexión como USB o salidas PWM.
<ul style="list-style-type: none"> Velocidad de CPU 	La velocidad de procesamiento no es de gran potencia, pero satisface procesos de recepción, procesamiento y envío de datos.

Adaptado de (DescubreArduino), s.f.

2.6.2. Software

Tabla 9.
Cuadro de Características RASPBIAN

	CARACTERÍSTICAS
SISTEMA OPERATIVO RASPBIAN	Es una distribución de GNU/Linux, por ende es software libre basado en Debian 8.0 específicamente diseñado para placas de computadora (SBC) Raspberry Pi. Esta distribución usa LXDE, que no es más que un entorno de escritorio libre para UNIX y algunas otras

	plataformas. Utiliza Midori para el caso de navegar sobre la Web.
APLICACIONES	Para el caso de desarrollo de proyectos, contiene varias herramientas de desarrollo, como IDLE, para el lenguaje de programación Python o Scratch.
COMPATIBILIDAD	Es totalmente efectivo y compatible con Raspberry Pi 3, ya sea con interfaz gráfica o únicamente por consola, se pueden montar varios servicios de gran utilidad empresarial o a nivel de proyectos sustentables, como el caso de servidor Web o de Base de Datos.

Adaptado de (Espberry,2016)

Tabla 10.
Cuadro de Características HTTP Apache

	CARACTERÍSTICAS
FUNCIONALIDAD	Es un servidor web HTTP de código abierto, que sirve para las diferentes plataformas de UNIX, por ejemplo, para BSD, GNU/Linux, otras plataformas como Microsoft Windows, Macintosh, entre otras.
USO Y APLICACIÓN	Es usado principalmente como envío de páginas web, ya sean estáticas o dinámicas, en la World Wide Web (WWW). Por ende, es utilizado para una diversidad de proyectos, los cuales utilizan su respectivo diseño de aplicación Web. Logrando así, obtener datos o enviarlos a través de internet y procesarlos según lo deseado.

Adaptado de (Ecured), s.f.

2.6.2.1. **Arduino 1.8.5**

Tabla 11.

Cuadro de Características Arduino 1.8.5

El software de código abierto Arduino (IDE), permite que sea sencillo escribir código y subirlo a la pizarra (board). Se ejecuta en Windows, Mac OS X y Linux. El entorno está escrito en Java y está basado en Processing y otro software de código abierto. Este software se puede usar con cualquier placa perteneciente a la marca Arduino.

A nivel de programación, es importante destacar que, si se desea realizar un proyecto, de recepción, procesamiento y envío de datos, debe estar totalmente configurado a través de este software, de esa manera, se logra obtener los resultados deseados, en el caso de un sistema biométrico, recibe la señal de la huella, la procesa, y la envía hacia cualquier otro servidor que tenga alojados los datos del usuario, de igual manera, envía una petición al servidor para obtener información y comparar con la señal de entrada.

Adaptado de (DescubreArduino) s.f

2.7. Configuración de los prototipos

2.7.1. Configuración Prototipo N°1

A nivel de hardware, se crea el prototipo, que consta del lector de huella GT-511C3, con conexión serial, hacia el arduino, el cual recibe la señal, la procesa, y se encarga de enviar la información hacia el servidor.

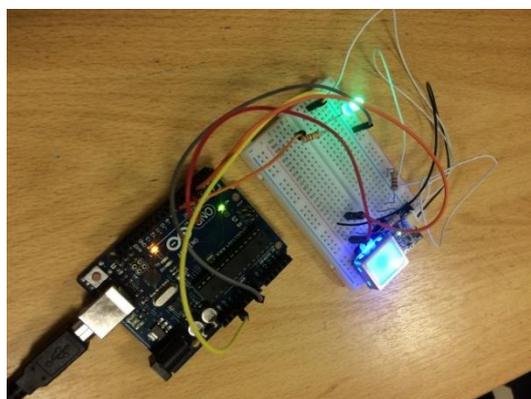
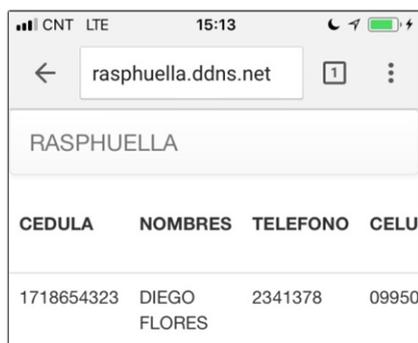


Figura 22. Prototipo de lector de Huellas. [Foto]

A través del software (Arduino versión 1.8.5) se genera el programa, el cual consta de varias funciones específicas, entre ellas se encuentra el enrolamiento de usuarios, los cuales se deben instanciar en la base de datos del Raspberry Pi 3.

De igual manera, existe la función de identificar los usuarios ya registrados en el sistema, en el servidor mencionado. A través de la pizarra (board), del software, se puede evidenciar el funcionamiento de los procesos mencionados anteriormente, identificando el usuario que anteriormente ya fue instanciado, y creando nuevos usuarios con su respectivo ID.

El nombre del servidor Web se estableció con el nombre de dominio "rasphuella.ddns.net" o a su vez la IP del raspberry, como se puede ver en la siguiente figura.



The screenshot shows a mobile browser interface. At the top, the status bar displays 'CNT LTE', the time '15:13', and battery level. The browser's address bar contains 'rasphuella.ddns.net'. Below the address bar, the page title is 'RASPHUELLA'. The main content area features a table with the following data:

CEDULA	NOMBRES	TELEFONO	CELU
1718654323	DIEGO FLORES	2341378	09950

Figura 23. rasphuella.ddns.net

Al usar un servidor web, puede accederse al mismo a través de cualquier dispositivo electrónico con acceso a la red, una vez ingresado el dominio, se puede visualizar la administración de los usuarios ingresados, que se encuentran instanciados en la base de datos previamente especificada.

Tanto el servidor Web, como la base de datos, se encuentran alojadas en el Raspberry Pi 3, el cual se comunica inalámbricamente hacia el módulo arduino, y se encuentra de manera local (LAN) en la red de la Universidad de las Américas.

2.7.2. Configuración Prototipo N°2 y Prototipo Final

El manual de usuario, en el cual consta todos los pasos para el correcto funcionamiento del biométrico, se encuentra en la sección de “Anexos” del presente documento.

Para la configuración de este prototipo se debe tomar los siguientes parámetros que maneja el Raspberry pi 3:

En la dirección “var/www/html/rasphuella” se encuentra la siguiente estructura:

- **Css:** Carpeta que contiene el diseño de la página web
- **Fonts:** Contiene las fuentes para la página web e iconos
- **Js:** Se aloja el archivo *Main.js* que es el javascript que maneja la función e información tanto de base de datos como de la lectura de las huellas para mostrarlas en la web
- **Php:** Dentro de este directorio se encuentran los diferentes archivos de manejo para el manejo de los usuarios para la página.
 - **Conexión.php:** Realiza la conexión a la base de datos Mysql.
 - **Config.php:** Contiene las credenciales para la base de datos.
 - **Data.txt:** Archivo donde se guarda temporalmente el id leído desde el Arduino.
 - **Eliminar_usuarios.php:** script para eliminar usuarios en la base de datos desde la web.
 - **Guardar_usuario.php:** Script para crear usuario en la base de datos desde la web.
 - **Lectura.py:** Script que se ejecuta infinitamente para la lectura de datos enviados desde el Arduino a la Raspberry por bluetooth. Si envía “R:num” se registra un nuevo id o huella, si envía “A:num” se consulta con la base de datos para verificar si existe el usuario, si envía otros datos no se toma en cuenta. Si encuentra el usuario envía un HIGH

por el GPIO 26 o pin 37 de la Raspberry además de enviar la letra “E” al Arduino. Si no lo encuentra envía la letra “N”.

- **Lista_usuarios.php:** Script para mostrar todos los usuarios existentes en la base de datos en la web.
- **Meta.php:** Realiza las consultas o peticiones en la base de datos Mysq.
- **Modificar_usuario.php:** Script para modificar un usuario especifico en la base de datos desde la web.
- **Obtener_usuario.php:** Obtiene la información del usuario seleccionado para mostrar en la web.
- **Index.html:** Es la página principal de la web.

Ya que el dominio rasphuella.ddns.net era temporal para las pruebas de conexión y registro de huellas se creo en el Raspberry la base de datos utilizando MySQL para almacenar los usuarios y la ID de la huella digital que va a ser recibida desde el microcontrolador ATMEGA 328P, adicional a esto se crea una página web dentro del web server Apache.

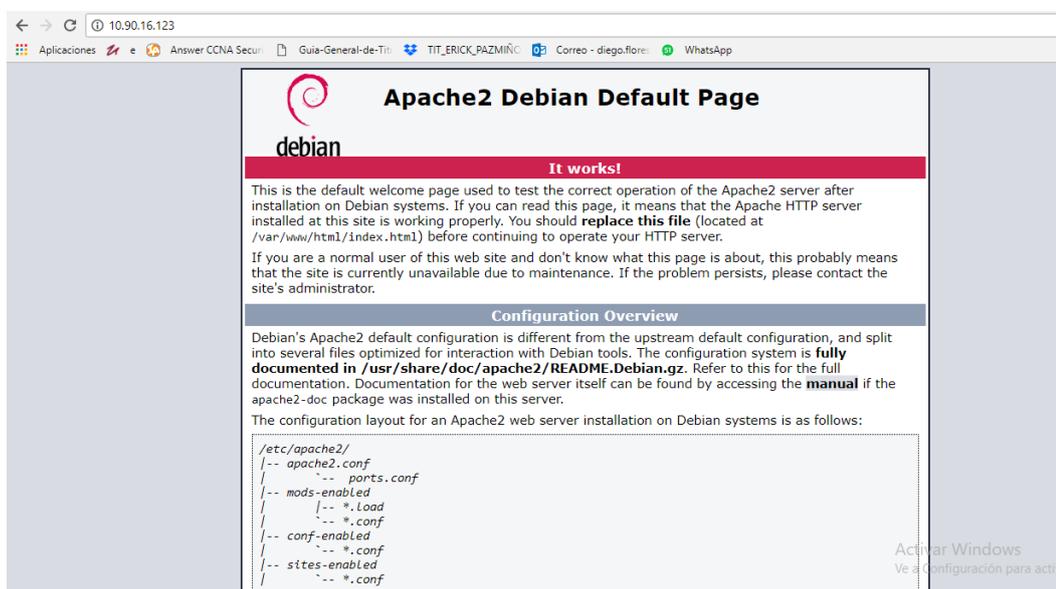


Figura 24. Acceso a Apache2



Figura 27. Página Web Rasphuella

Como se puede apreciar no se están grabando los usuarios y sus datos ya que no esta configurada la base de datos ni creada la tabla correspondiente.

Para la creación de la base de datos se debe ingresar consola de la raspberry dentro de los comandos de MySQL la siguiente sentencias :

```
create database rasphuella; use rasphuella;

CREATE TABLE usuarios (id_usuario int(11) NOT NULL AUTO_INCREMENT, cedula varchar(10) DEFAULT NULL, nombres varchar(255) DEFAULT NULL, telefono varchar(20) DEFAULT NULL, celular varchar(20) DEFAULT NULL, correo varchar(100) DEFAULT NULL, huella varchar(50) DEFAULT NULL, estado int(1) NOT NULL DEFAULT 0, PRIMARY KEY (id_usuario)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

grant all privileges on *.* to 'root'@'localhost' identified by 'root'; flush privileges;

exit
|
```

Figura 28. Sentencias MySQL

3. CAPÍTULO III. ANALISIS DE RESULTADOS

3.1. Análisis de costos

Tabla 12.

Análisis de costos

Componentes	Cantidad	Valor Unitario (\$)	Valor Total (\$)
Raspberry pi 3	2	65	130
Arduino UNO	1	12	12
Lector Huella Digital GT511C3	2	40	80
PCB Fibra Vidrio	3	7,9	23,7
Borneras	8	0,28	2,24
Módulo Bluetooth HC06	2	9	18
Regleta Macho	1	0,5	0,5
Regleta Hembra	2	0,5	1
Transistores	3	0,08	0,24
Diodo	4	0,08	0,32
Caja Acrílica	1	25	25

Pantalla LCD 16x2	1	5	5
Caja 12 x 12	2	1,43	2,86
Cables	40	0,1	4
Leds	5	0,03	0,15
Resistencias	10	0,03	0,3
Potenciómetro	2	1	2
Relay	3	2	6
		TOTAL	<u>313,31</u>

De acuerdo con el presupuesto inicial, se predice un costo de \$ 300 en todo el sistema, según la lista de costos tangibles se sobrepasó el valor estimado, cabe recalcar que solo se tomó en cuenta los costos de los equipos que se utilizó para el desarrollo del sistema.

Al ser implementado en las instalaciones de la Universidad de las Américas, se toma en cuenta el costo de instalación, debido al ser un trabajo de titulación por parte de estudiantes de la carrera. Tomando en cuenta esto, el tiempo y la mano de obra no son valores que se tomen en consideración como costos ya que todo se realiza por los desarrolladores de este proyecto.

Nota: El manual de usuario, en el cual consta todos los pasos para el correcto funcionamiento del biométrico, se encuentra en la sección de “Anexos” del presente documento.

3.2. Realizar las pruebas

3.2.1. Pruebas Prototipo N°1

Se realizaron las pruebas con el primer prototipo que constaba de 4 partes dentro de una caja acrílica de color negro impresa en 3D de tal manera que se podía pegar a la pared con tornillos como se puede ver las siguientes figuras.



Figura 29. Módulo en la entrada de la sala interior UITEC



Figura 30. Prototipo N°1

En la *Figura 47* se puede apreciar el menú principal en la pantalla LCD de 16x2, el lector de huella digital GT-511C3 de 5V y dos pulsadores los cuales cumplían con las funciones de:

- a) Enrolar nuevas huellas para el registro de nuevos usuarios de la UITEC.



Figura 31. Enrolar nuevas huellas presionando el botón

- b) Verificar si los usuarios están registrados y pueden acceder a la sala principal desbloqueando la chapa magnética.

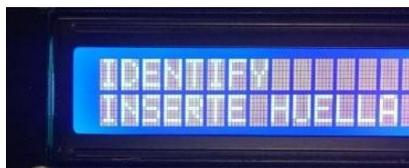


Figura 32. Identificar los usuarios registrados presionando el segundo botón

En este desarrollo de pruebas con este prototipo se debe tener una conexión vía bluetooth desde el módulo HC-06 hacia el Raspberry pi 3 que es donde se encuentran los usuarios nuevos y enrolados anteriormente.

Se debe realizar la conexión bluetooth desde el Raspberry pi 3 siendo este el maestro y el módulo HC-06 incluido en el sistema el esclavo, pero para ello se necesita primero tener una conexión a la red de la universidad.

Como podemos apreciar en la Figura 50 nos da la IP 10.90.4.215, con esto podemos conectarnos con el programa Putty a otra ventana de comandos del Raspberry.

```
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.90.4.215 netmask 255.255.252.0 broadcast 10.90.7.255
    inet6 fe80::d13b:f8f6:841c:dd31 prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:a6:65:21 txqueuelen 1000 (Ethernet)
    RX packets 233 bytes 30701 (29.9 KiB)
    RX errors 0 dropped 223 overruns 0 frame 0
    TX packets 912 bytes 64668 (63.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1 (Local Loopback)
    RX packets 524 bytes 49780 (48.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 524 bytes 49780 (48.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether b8:27:eb:f3:30:74 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$
```

Figura 33. IP Consola Raspberry

3.2.2. Pruebas prototipo N°2

A partir del prototipo N°1 se decidió realizar una mejora en cuanto a lo estético por lo cual se eliminó la caja acrílica negra y solo se tenían los circuitos y el Arduino Uno para poder ubicar en la pared de entrada a la sala interior de la UITEC.

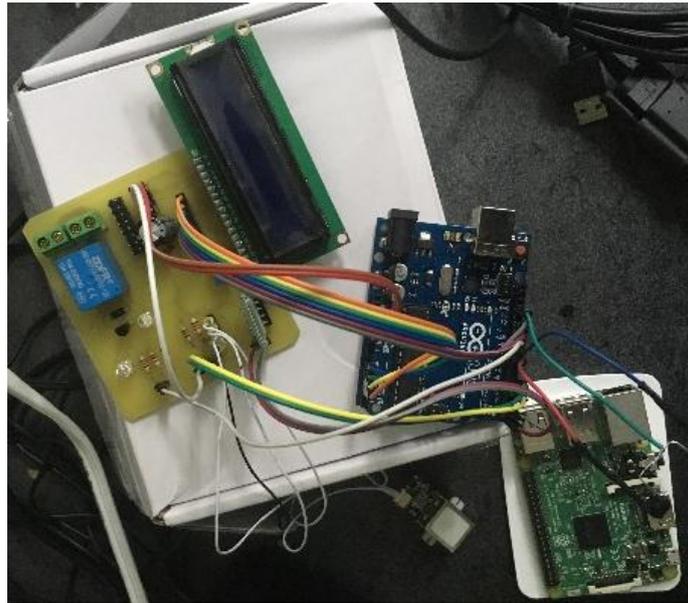


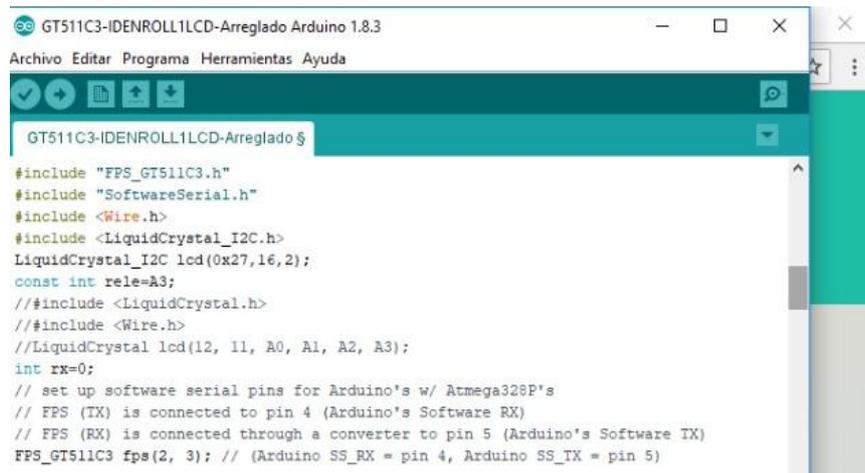
Figura 34. Sistema sin Caja Acrílica

Estos módulos se tuvieron que tratar de ubicar en una pared de 12,5 cm de ancho y 17,5 cm de largo a una altura de 1,27 m, en esta pared se ubicarían los orificios para la pantalla LCD de 16x2 que tienen una dimensión de 8 x 3,5 x 1,1 cm en total.

Al realizar los orificios en la pared de Gypsum se realiza solo los orificios para el display de 6,45 x 1,6 x 1,1 cm y para el lector de huellas de digitales GT-511C3 de 3,7 x 1,7 x 0,95 cm.

Los circuitos tienen que ir dentro de la pared de Gypsum los cuales tienen que entrar por la parte de atrás donde se tiene una un interruptor de luz.

Al final de las pruebas de este prototipo nuevamente se realizó la conexión por medio del módulo de bluetooth hacia el Raspberry Pi 3 que es en donde se encuentra alojado los usuarios y huellas registrados.



```

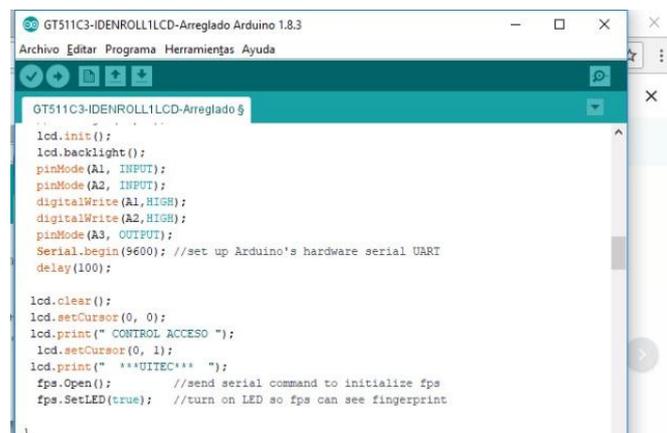
GT511C3-IDENROLL1LCD-Arreglado $
#include "FPS_GT511C3.h"
#include "SoftwareSerial.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
const int rele=A3;
//#include <LiquidCrystal.h>
//#include <Wire.h>
//LiquidCrystal lcd(12, 11, A0, A1, A2, A3);
int rx=0;
// set up software serial pins for Arduino's w/ Atmega328P's
// FPS (TX) is connected to pin 4 (Arduino's Software RX)
// FPS (RX) is connected through a converter to pin 5 (Arduino's Software TX)
FPS_GT511C3 fps(2, 3); // (Arduino SS_RX = pin 4, Arduino SS_TX = pin 5)

```

Figura 35. Código Arduino (1)

Incluimos las librerías que vamos a utilizar, entre ellas se encuentra “FPS_GT511C3.h” para administrar el lector de huella previamente explicado, la librería “SoftwareSerial.h” la cual comunica a través del módulo Bluetooth hacia el Raspberry Pi 3, la librería “Wire.h” para gestionar el LCD junto con la librería del módulo “LiquidCrystal_I2C.h” el cual es un dispositivo para ahorrar cableado en el LCD.

Creamos constantes, para el relé, para el módulo que utilizará varios pines.



```

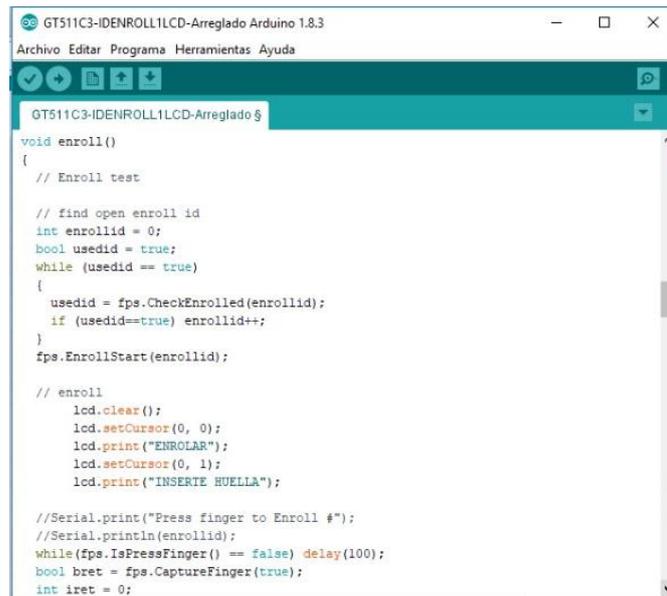
GT511C3-IDENROLL1LCD-Arreglado $
...
lcd.init();
lcd.backlight();
pinMode(A1, INPUT);
pinMode(A2, INPUT);
digitalWrite(A1, HIGH);
digitalWrite(A2, HIGH);
pinMode(A3, OUTPUT);
Serial.begin(9600); //set up Arduino's hardware serial UART
delay(100);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print(" CONTROL ACCESO ");
lcd.setCursor(0, 1);
lcd.print(" ***UIITEC*** ");
fps.Open(); //send serial command to initialize fps
fps.SetLED(true); //turn on LED so fps can see fingerprint
}

```

Figura 36. Código Arduino (2)

En la función setup(), especificamos las variables de ingreso, como los pulsadores, variables de salida, como lo es el relé, inicializar el LCD, e imprimir texto acorde a nuestra necesidad.



```

GT511C3-IDENROLL1LCD-Arreglado Arduino 1.8.3
Archivo Editar Programa Herramientas Ayuda
GT511C3-IDENROLL1LCD-Arreglado $
void enroll()
{
  // Enroll test

  // find open enroll id
  int enrollid = 0;
  bool usedid = true;
  while (usedid == true)
  {
    usedid = fps.CheckEnrolled(enrollid);
    if (usedid==true) enrollid++;
  }
  fps.EnrollStart(enrollid);

  // enroll
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("ENROLAR");
  lcd.setCursor(0, 1);
  lcd.print("INSERTE HUELLA");

  //Serial.print("Press finger to Enroll #");
  //Serial.println(enrollid);
  while (fps.IsPressFinger() == false) delay(100);
  bool bret = fps.CaptureFinger(true);
  int iret = 0;

```

Figura 37. Código Arduino (3)



```

while (fps.IsPressFinger() == false) delay(100);
bret = fps.CaptureFinger(true);
if (bret != false)
{
  lcd.setCursor(0, 1);
  lcd.print("RETIRE ");
  //Serial.println("Remove finger");
  fps.Enroll2();
  while (fps.IsPressFinger() == true) delay(100);
  lcd.setCursor(0, 1);
  lcd.print("INSERTE HUELLA ");
  //Serial.println("Press same finger yet again");
  while (fps.IsPressFinger() == false) delay(100);
  bret = fps.CaptureFinger(true);
  if (bret != false)
  {
    lcd.setCursor(0, 1);
    lcd.print("RETIRE ");

```

Figura 38. Código Arduino (4)

Especificamos la función enrollar, la cual consta de una variable entera (int), esta variable recibe la señal del lector, puede ser la señal de un usuario ya ha sido ingresado, o por enrollar.

Al iniciar el método de enrollar, se despliega en el LCD “Ingresa Huella”, en la cual, al presionar el lector, se esperará 3 segundos para leer y almacenar esta huella en el lector con su correspondiente ID. Esta ID se utilizará en web server para ingresarla a la base de datos.

```

GT511C3-IDENROLL1LCD-Arreglado $
}

void identify() {
  lcd.clear();
  //lcd.setCursor(0, 0);
  //lcd.print("CONTROL ACCESO");
  lcd.setCursor(0, 1);
  lcd.print("LEYENDO HUELLA");
  int st=0;
  // Identify fingerprint test
  while(st==0) {
    if (fps.IsPressFinger())
    {
      fps.CaptureFinger(false);
      int id = fps.IdentifyI_N();

      /*Note : GT-511C3 can hold 200 fingerprint templates.
      GT-511C1R can hold 20 fingerprint templates.
      Make sure to change the id depending on what
      model you are using */
      if (id < 200)
      {
        //if the fingerprint matches, provide the matching template ID
        //Serial.print("Verified ID:");
        //Serial.println(id);
        st=1;
        lcd.clear();
      }
    }
  }
}

```

Figura 39. Código Arduino (5)

```

void loop()
{
  delay(500);
  //lcd.clear();
  // lcd.setCursor(0, 1);
  // lcd.print(" ***UIITEC*** ");

  if(digitalRead(A2)==LOW){
    enroll();
  }

  if(fps.IsPressFinger() == true){
    identify();
  }
}

```

Figura 40. Código Arduino (6)

Por último, para el caso de identificar una huella, se despliega en el LCD “Leyendo Huella” en el momento en que una huella es colocada, dentro de un loop infinito, hasta que uno de estos pulsadores interrumpa este LOOP.

La señal ingresada se va a verificar en el lector de huella, y posteriormente se comunica con el Raspberry para determinar que se ha ingresado en la base de datos, al recibir confirmación, se procederá a abrir el relé.

En el caso del prototipo N°2 se realizó el cambio de componentes hacia una sola placa para poder ahorrar espacio y que se vea mejor de tal manera se pueda apreciar desde una caja de 12 x 12 el circuito por la parte posterior de la pared donde se instalará el producto final.

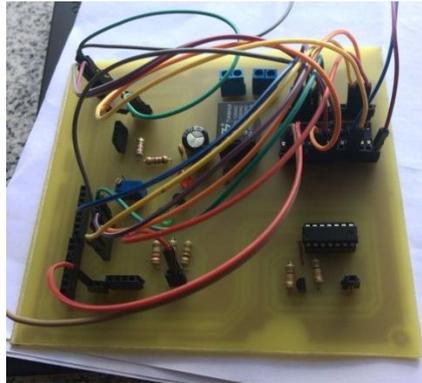


Figura 41. Placa Prototipo N°2

3.2.3. Pruebas Prototipo N°3

Durante las pruebas con los anteriores prototipos en el software no se realizó cambios de tal manera que solo la placa se realizó cambios y se procedió a realizar las pruebas pertinentes.

Para la alimentación de nuestra placa se tuvo que comprar un transformador de voltaje de 110 v a 12 v ya que todo el circuito esta implementado para funcionar con este voltaje.



Figura 42. Transformador de voltaje 110 (v) a 12 (v)

Para que el circuito final quede a la vista dentro de la caja que se mencionó anteriormente se procedió a realizar el agujero en la pared de Gypsum en donde quedaría el case.



Figura 43. Caja metálica de 12 x 12 cm

Se probó las conexiones pertinentes con el prototipo N°3 con el módulo bluetooth hacia la Raspberry pi 3 con los comandos correspondientes ingresados en consola.

```
sudo bluetoothctl
power on
agent on
scan on
scan off
devices (deberá salir el módulo bluetooth del arduino) pair DIRECCION_MAC (si pide pin ingresarlo)
paired-devices (muestra los dispositivos pareados o linkeados)
trust DIRECCION_MAC (lo asigna como de confianza)
info DIRECCION_MAC (muestra información del bluetooth)

exit
```

Figura 44. Comandos de conectividad con el módulo Bluetooth

```
sudo rfcomm connect hci0 00:21:13:00:F0:94
cd /var/www/html/rasphuella
cd php
sudo python lectura.py
```

Figura 45. Conexión al módulo bluetooth HC-06 desde Raspberry

En la *Figura 45* se realiza la conexión en la consola de la Raspberry pi 3 hacia el módulo bluetooth que se encuentra en la placa, esta conexión se logra mediante la línea de comandos y la dirección MAC del módulo HC-06 que es único.

Al ingresar el comando “sudo Python lectura.py” podemos apreciar las sentencias y datos leídos por la placa, es decir que al colocar una huella digital

ya registrada en el servidor enviará un dato que será visualizado en esta consola mientras se esté corriendo el programa.

Finalmente, en la siguiente figura se puede apreciar la placa final la cual será colocada en la pared de la sala interior de la UITEC y tapada con una tapa de plástico y cerrada con tornillos para poder realizar futuros cambios.



Figura 46. Placa Final del control de acceso para UITEC

3.3. Análisis de pruebas

3.3.1. Análisis Prototipo N°1

En el desarrollo de la prueba se produjo dos inconvenientes, el primer problema fue la falta de señalización con respecto a los botones de enrolamiento y al de verificación que no tenían ninguna diferencia y los usuarios se podrían confundir. El segundo fue que siempre se tenía que presionar el botón para poder leer la huella y verificar si el usuario esta registrado en la base de datos lo cual debería siempre estar habilitado desde el momento que cualquier usuario coloque la huella en el lector.

Para salir de la sala no se había integrado un botón por lo cual eso se añade a los requerimientos de la UITEC frente a nuestro sistema.

Otro de los inconvenientes que se toma en cuenta es que en este prototipo los puntos están pegados con silicona lo cual es antiestético y por eso se requiere realizar una sola placa.

Frente a este nuevo requerimiento también se solicitó que los pulsadores sean más grandes ya que los integrados en la caja acrílica eran muy pequeños.

3.3.2. Análisis Prototipo N°2

Dentro del análisis de este prototipo se encontraron tres problemas la placa era muy grande para poder meter en la pared por lo cual se tendría que minimizar el tamaño, también el funcionamiento se vio afectado al unir los componentes en uno por el manejo de los elementos al momento de soldar en la placa y faltó colocar en la placa el cristal de 16 Mhz que se utiliza como reloj del Arduino Uno.

Otro de los problemas es el cableado ya que pese a que se unió los tres componentes en una sola placa aún se puede observar los cables por encima de la placa para conectar el microcontrolador a los componentes que no se pudo realizar en el mapeo de la placa.

Aparte de los problemas mencionados se requiere agregar el botón de *reset* del Arduino y agregarlo a la placa que se tenía.



Figura 47. Cristal de Arduino UNO

Adaptado de (SiguePasos.blogspot, 2015)

3.3.3. Análisis Producto final

Después de varios cambios a partir de los análisis de los prototipos N°1 y 2 se agregó la caja en la pared de la sala interna de la UITEC con el fin de ubicar la placa final en esta y cerrarla con una tapa de plástico transparente para que se pueda apreciar la implementación del producto final propuesto.

Se tuvo un problema con el módulo bluetooth y su conexión al enviar los datos hacia la Raspberry pi 3 ya que no graba el dato de los usuarios enrolados hasta el momento por lo cual se procedió a realizar los cambios para poder solucionar este inconveniente.

3.4. Resultados Obtenidos

Para realizar los huecos en la pared se tomó las siguientes medidas tomando en cuenta una altura promedio de los usuarios entre hombres y mujeres de 1,67 (m).



Figura 48. Altura para pantalla y lector de huella

Entre los dos componentes se tiene una separación considerable para que el usuario pueda visualizar en la pantalla mientras coloca el dedo en el lector de huella digital.

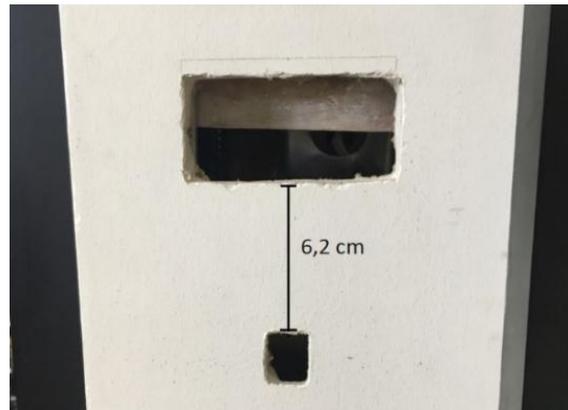


Figura 49. Distancia entre LCD y lector de huella

Con respecto a la caja donde va a ir ubicada la placa y conectados los demás componentes se ubicó de igual manera para que pueda ser visualizado por una persona de alrededor de 1,60 m.



Figura 50. Altura de la caja

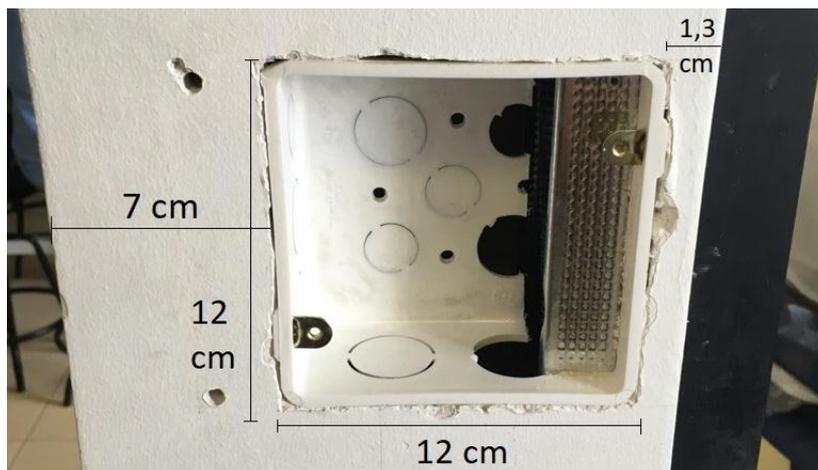


Figura 51. Dimensiones de la caja

Para poder colocar la caja dentro de la pared se tuvo un problema ya que a 1,5 cm de cada lado de la pared se encuentra una varilla de metal la cual impedía que entre la caja de 12 x 12. Para solucionar este pequeño inconveniente se procedió a cortar la parte trasera de la caja a 1,5 cm de ancho a partir de la parte derecha y a 1 cm del borde frontal de la caja.

Luego de la implementación el sistema quedó ya activo para su funcionamiento inmediato.



Figura 52. Parte delantera del sistema



Figura 53. Parte trasera del sistema (Circuito Visible)

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

Cuando el baudrate del serial es diferente, entre el módulo bluetooth en el arduino, y el módulo bluetooth del Raspberry, no se va a efectivizar la transmisión de datos.

Al declarar variables locales, es más fácil administrar los objetos en el script de arduino, tales como LCD y el relé, para cumplir funciones como activar y desactivar.

Especificamos los métodos de nuestras funciones en la página Web (Html) en el directorio `/var/www/html/rasphuella/js/main.js` con todas las funciones necesarias para los requerimientos.

Usamos base de datos MySQL para la creación de usuarios, en la cual se especifica el acceso por root y la contraseña por defecto.

Se determinó que la salida de corriente del relé no es suficiente para el acceso magnético, por ende, mediante un transistor BJT obtuvimos la corriente necesaria de ingreso al sistema, en modo switch con la señal de entrada.

Un estilo ordenado de placa logró satisfacer y organizar nuestro sistema, optimizando 3 placas de 10 cm, en una sola de 5 cm usando los caminos de voltaje correctos.

Al usar nuestro sistema operativo Raspbian, en modo consola, consume menos recursos que modo gráfico, con ello es más factible crear servidores de distintos servicios, sin consumir RAM innecesaria.

Logramos obtener los resultados deseados mediante el lector de huella GT511C3, el cual posee su propia base de datos interna y desplegar huellas digitales únicamente a través de una variable entera.

El script generado para cumplir nuestras funciones, siempre van a ser utilizadas mediante el acceso en modo root, mediante el comando "sudo" caso contrario toca brindar permisos totales al archivo python.

Para acceder a nuestro Raspberry, utilizamos un módem para identificar la IP, caso contrario necesitamos una pantalla y teclado para correr el script y conexión al módulo bluetooth.

Al ingresar una huella digital, nos es imposible utilizar la misma para otro ingreso, evitando duplicados por parte del lector GT511C3.

Todos los requerimientos por parte de UITEC fueron cumplidos satisfactoriamente de acuerdo con el plazo establecido.

4.2. Recomendaciones

Debemos determinar módulos Bluetooth esclavos para la conexión externa desde el Raspberry, caso contrario, como con un módulo HC05, se debe programar para cumplir esta función.

Es posible la optimización de cables, por ejemplo usando una placa óptima, o mediante distintos módulos como el I2C para la pantalla LCD.

Debemos usar pasta térmica para ciertos elementos que siempre van a crear calor excesivo, de esa manera evitamos un sobrecalentamiento del sistema.

Hay que analizar las distintas tecnologías inalámbricas disponibles, para poder crear módulos óptimos sin cables en nuestros sistemas.

Necesitamos optimizar líneas de código, tanto en el arduino como en el raspberry, de esa manera ocupan menor cantidad de memoria RAM y se obtiene menos procesos innecesarios.

Al crear una IP estática, nos facilita tener que acceder al Raspberry mediante una pantalla y teclado, puesto que iniciamos sesión a través de SSH o telnet.

Es importante proteger nuestro dispositivo con "case" propio de la empresa fabricante, de esa manera evitamos posibles coaliciones con el sistema.

Es vital la prevención de posibles desastres naturales en nuestro sistema, tales como temblores que causen ausencia de luz y que puedan afectar nuestro sistema.

Al utilizar servidores ya conocidos, como para páginas web o base de datos, evitamos posibles fallas en el sistema, de esa manera ahorramos costos económicos y de logística.

Siempre podemos establecer la base de proyectos, en varios repositorios ya implementados previamente, de esa manera es más factible crear nuevos programas partiendo de algo ya instalado previamente.

REFERENCIAS

- 3nStar. (2017). Lector biométrico de huellas dactilares. Recuperado el 18 de Mayo de 2018 de <http://3nstar.com.ar/producto/3nstar-lector-biometrico-de-huellas-dactilares-ta010/>
- Aguilar, G., Sánchez, G., Toscano, K., Nakano, M., & Pérez, H. (2008). Reconocimiento de huellas dactilares usando características locales. Recuperado el 16 de Febrero de 2018 de <http://www.scielo.org.co/pdf/rfiua/n46/n46a10.pdf>
- alaMaula. (2017). Raspberry Pi 3 Modelo B. Recuperado el 20 de Febrero de 2018 de <https://www.alamaula.com/a-pc/mendoza/raspberry-pi-3-modelo-b/1001690827990910557887309>
- Aprendiendo Arduino. (2015). Lenguaje C++. Recuperado el 3 de Abril de 2018 de <https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-c/>
- BangGood. (2018). Raspberry Pi 3 Modelo B. Recuperado el 17 de Enero de 2018 de https://www.banggood.com/es/Raspberry-Pi-3-Model-B-ARM-Cortex-A53-CPU-1_2GHz-64-Bit-Quad-Core-1GB-RAM-10-Times-B-p-1041862.html
- Bejerano, P. (2014). De la huella dactilar a nuestras venas, el futuro de la biometría. Recuperado el 20 de Abril de 2018 de <https://blogthinkbig.com/biometria-de-las-venas-de-la-mano>
- Bellido, F., De la Cruz, J., Torres, M., & Gistas, J. (2004). Comunicación inalámbrica con Bluetooth. Recuperado el 27 de Enero de 2018 de <http://www.tecnicaindustrial.es/tiadmin/numeros/15/06/a06.pdf>.
- Caffelli, P. (2010). Qué es y cómo funciona el sistema Bluetooth. Recuperado el 11 de Enero de 2018 de <http://etecnologia.com/gadgets/funcionamiento-bluetooth>

- Cano, A. (s.f.). Cuadro Comparativo de los Lenguajes de Programación. Recuperado el 27 de Abril de 2018 de <http://www.youblisher.com/p/1068622-cuadro-comparativo-de-los-lenguajes-de-programacion/>
- Cano, J. L. (2014). Adopción de pautas de seguridad informática. Recuperado el 02 de Marzo de 2018 de <https://slideplayer.es/slide/1041716/>
- Chavarrea, G. (2013). Construcción e implementación de un circuito electrónico mediante sensor de huellas dactilares para el control de ingreso y salida del personal autorizado al cuarto de equipos de computación ubicado en el Instituto de estudios del Petróleo- Quito. Escuela Politécnica Nacional, 156. Recuperado el 8 de Marzo de 2018 de <http://bibdigital.epn.edu.ec/bitstream/15000/6359/1/CD-4882.pdf>
- Cruceña, B. (2009). Que son los Sistemas Biométricos. Recuperado el 22 de Junio de 2018 de <http://eju.tv/2009/04/que-son-los-sistemas-biometricos/>
- Cruz, C. (2008). Reconocimiento de rostros basado en características invariantes. INAOE, 126.
- DescubreArduino. (s.f.). Arduino Uno, partes, componentes, para qué sirve y donde comprar. Recuperado el 11 de Febrero de 2018 de <https://descubrearduino.com/arduino-uno/>
- Díaz, J., Harari, I., & Amadeo, A. (2013). Guía de recomendaciones para diseño de software centrado en el usuario. La Plata.
- DIYGEEKS. (2017). *Setting up Arduino IDE for NodeMCU*. Recuperado el 03 de Marzo de 2018 de <https://diygeeks.org/learn/setting-up-arduino-ide-for-nodemcu/>
- DOCBOX. (2008). *Digital Image Processing*. Recuperado el 29 de Diciembre de 2017 de <http://hobbydocbox.com/Photography/71275683-Digital-image-processing.html>

- Dorantes, C. A. (2015). Qué es PostgreSQL y cuáles son sus ventajas. Recuperado el 09 de Mayo de 2018 de <https://platzi.com/blog/que-es-postgresql/>
- EcuRed. (2012). Geometría de la mano. Recuperado el 03 de Junio de 2018 de https://www.ecured.cu/Geometr%C3%ADa_de_la_mano
- Eju.tv. (2009). Que son los Sistemas Biométricos. Recuperado el 12 de Abril de 2018 de <http://eju.tv/2009/04/que-son-los-sistemas-biometricos/>
- Espinosa, F. (2018). Introducción a los Microcontroladores. Recuperado el 14 de Mayo de 2018 de http://www.utm.mx/~fsantiag/Micros/1_Introduccion_MCUs.pdf
- Feed RSS Reader. (2017). Raspberry Pi3. Recuperado el 02 de Febrero de 2018 de http://www.feedrssreader.com/aggregator/p/i/?a=reader&next=1507798837536042&ajax=1&get=f_8&order=ASC
- Figuerola, K., & Villacreses, A. (2013). Comercialización de sistemas de acceso vehicular y control peatonal para las ciudades privadas, basada en sistemas biométricos de reconocimiento facial. ESPOL, 133.
- Florencio, A. (2017). Comparativa de los principales sistemas gestores de Bases de Datos (SGBD). Recuperado el 10 de Marzo de 2018 de <https://www.cursosgis.com/comparativa-de-los-principales-sistemas-gestores-de-bases-de-datos-sgbd/>
- FuenteLibre. (2016). RASPEBERRY PI. Recuperado el 05 de Mayo de 2018 de <https://fuentelibre.blog/tag/raspeberry-pi/>
- Gaensslen, H. C. (1991). *Advances in Fingerprint Technology*. New York: CRC Press. 38-59
- Galvis, C. (2007). Estado del arte, diseño y programación de una aplicación que utilice tecnologías de autenticación. Bogota D.C. Recuperado el 20 de Enero de 2018 de <https://kipdf.com/estado-del-arte-diseo-y->

programacion-de-una-aplicacion-que-utilice-tecnologias-d_5ab49d861723dd419ce53feb.html

Generation Robots. (2017). Fingerabdruckscanner. Recuperado el 30 de Marzo de 2018 de <https://www.generationrobots.com/de/401607-fingerabdruckscanner.html>

Gestiopolis. (2007). Evaluación de la implementación de nuevas tecnologías en las empresas. Recuperado el 12 de Febrero de 2018 de <https://www.gestiopolis.com/evaluacion-implementacion-nuevas-tecnologias-empresas/>

Gómez, R. (2002). Los Diodos Led. Recuperado el 02 de Marzo de 2018 de http://www.iearobotics.com/personal/ricardo/articulos/diodos_led/index.html

Gomez, R. (2006). Identificación de personas mediante el reconocimiento dactilar y su aplicación a la seguridad organizacional. (tesis universidad nacional mayor de san marcos). Recuperado el 03 de Abril de 2018 de http://cybertesis.unmsm.edu.pe/bitstream/cybertesis/3233/1/Cernandez_gh.pdf

González, R. (2013). Reconocimiento del Iris. Recuperado el 12 de Febrero de 2018 de <http://ramon-gzz.blogspot.com/2013/01/reconocimiento-del-iris.html>

González, J. (2003). CONTROL DE ACCESO BASADO EN RECONOCIMIENTO DE IRIS . 50. Recuperado el 14 de Diciembre de 2017 de <http://biblioteca.unitecnologica.edu.co/notas/tesis/0024820.pdf>

GSITIC. (2018). Redes inalámbricas. Protocolos Características funcionales y técnicas. Sistemas de expansión del espectro. Sistemas de acceso. Modos de Operación. Seguridad. Normativa reguladora. Recuperado el 22 de Mayo de 2018 de <https://gsitic.wordpress.com/2018/03/19/biv18-redes-inalambricas-protocolos-caracteristicas-funcionales-y-tecnicas-sistemas-de->

expansion-del-espectro-sistemas-de-acceso-modos-de-operacion-seguridad-normativa-reguladora/

lessanvicente. (s.f.). ¿Que es Oracle?. Recuperado el 11 de Febrero de 2018 de <https://iessanvicente.com/colaboraciones/oracle.pdf>

IMO. (2015). Insituto de Molestias Oculares. Recuperado el 20 de Marzo de 2018 de <http://imo.com.br/laser/>

iWeb, C. d. (2014). Una introducción a servidores de bases de datos. Recuperado el 20 de Enero de 2018 de <http://blog.iweb.com/es/2014/04/servidores-de-bases-de-datos/2487.html>

J.E.Nivetha. (2016). *Global Wireless E-Voting system*. Recuperado el 10 de Marzo de 2018 de <http://www.ijcst.com/vol73/1/17-k-kiruthika.pdf>

Lara, P. (2012). Sensores y Transductores. Recuperado el 29 de Marzo de 2018 de <http://1538445.blogspot.com/2012/11/21-sensores-y-transductores.html>

Lestoc, C. (2016). *Windows 10 on ARM gets tested on Raspberry Pi 3B*. Recuperado el 20 de Febrero de 2018 de <https://windowsreport.com/windows-10-arm-raspberry-pi-3/>

Lopez, Q. (2014). Herramientas de seguridad. Recuperado el 15 de Abril de 2018 de http://dis.um.es/~lopezquesada/documentos/IES_1213/SAD/curso/UT3/ActividadesAlumnos/1/index.html

Luis, C. (2014). Diseño e implementación de un sistema de control de asistencia de personal, mediante el uso de tecnología biométrica de huella dactilar. Quito, Pichincha, Ecuador: EPN. Recuperado el 24 de Abril de 2018 de <http://bibdigital.epn.edu.ec/bitstream/15000/7731/1/CD-5637.pdf>

- Martínez, T. (2010). Análisis de factibilidad de la implementación de un sistema de asistencia a clase mediante huella dactilar en la Facultad de Ingeniería "Arturo Narro Siller". UAT, 28.
- Microsoft. (2015). Introducción al lenguaje C# y .NET Framework. Recuperado el 11 de Enero de 2018 de <https://docs.microsoft.com/es-es/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>
- Microsoft. (2017). Servidores vinculados (motor de base de datos). Recuperado el 29 de Febrero de 2018 de <https://docs.microsoft.com/es-es/sql/relational-databases/linked-servers/linked-servers-database-engine?view=sql-server-2017>
- Mirolaba, F. (2011). Uso de la red vascular de la retina como medio biométrico de identificación. 56. Recuperado el 01 de Marzo de 2018 de <http://148.204.63.111/SABERv3/Repositorios/webVerArchivo/26203>
- Morales, N. (2017). Bluetooth, clases y versiones desde v1.0 hasta v5.0. Recuperado el 03 de Marzo de 2018 de <https://blog.330ohms.com/2017/02/02/bluetooth-clases-y-versiones-desde-v1-0-hasta-v5-0/>
- Morales, V. (2014). Implantación de sistemas biométricos en el aula. Recuperado el 10 de Febrero de 2018 de <https://vicenmorales.files.wordpress.com/2011/11/implantacion-de-sistemas-biometricos-en-el-aula.pdf>
- National Instruments. (2006). Comunicación Serial: Conceptos Generales. Recuperado el 09 de Febrero de 2018 de <http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>
- Navalpotro, M. (2014). Estudio y análisis comparativo de las actuales técnicas biométricas. 250. Recuperado el 16 de Abril de 2018 de http://oa.upm.es/32372/1/tesis_master_mario_navalpotro_molina.pdf

- nchsbands. (2017). *Killer Features About Iphone*. Recuperado el 06 de Junio de 2018 de Killer Features About Iphone: <http://nchsbands.info/2017/killer-features-about-iphone.html>
- Oracle. (2005). *Database Oracle*. Recuperado el 20 de Marzo de 2018 de http://docs.oracle.com/cd/B19306_01/server.102/b14220/physical.htm
- Ortega, A. (2008). *Biometría y Seguridad*. Madrid: Fundación Rogelio Segovia para el Desarrollo de las Telecomunicaciones.
- OverBlog. (2011). *Huella Digital con sus partes*. Recuperado el 20 de Febrero de 2018 de <http://algoquedecir.over-blog.es/article-que-funcion-tienen-dispositivos-biometricos-86149233.html>
- Patiño, S. (2012). *Sistema de Registro Biométrico de Asistencia para Empleados*. Recuperado el 01 de Abril de 2018 de <http://dspace.uazuay.edu.ec/bitstream/datos/2688/1/09224.pdf>
- Paz, N. (2009). *Revelado de huellas lofoscópicas en papel*. Recuperado el 22 de Abril de 2018 de <http://www.monografias.com/trabajos56/huellas-lofoscopicas/huellas-lofoscopicas3.shtml>
- Professional, F. (2016). *Sistemas de control de accesos para empresas*. Recuperado el 14 de Marzo de 2018 de <https://fermax.com/esp/sistemas-de-control-de-accesos-para-empresas-cuándo-es-necesario>
- Prometec. (2017). *Arduino UNO Rev 3*. Recuperado el 05 de Junio de 2018 de <https://www.prometec.net/producto/arduino-uno/>
- Raspberry. (2017). *RASPBERRY PI 3 MODEL B*. Recuperado el 12 de Abril de 2018 de <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- Rena, F. (2015). *Manual de Programación en C para principiantes y avanzados*. Recuperado el 01 de Marzo de 2018 de <https://www.fiuxy.org/threads/programacion-en-c-para-principiantes-y-avanzados-manual-pdf-mg.3582090/>

- Rosario, M. d. (2016). CONCEPTOS BÁSICOS DE ANATOMÍA. Recuperado el 09 de Febrero de 2018 de http://tanatopraxiablogspot.blogspot.com/2016/10/conceptos-basicos-de-anatomia-capitulo_31.html
- Rouse, M. (2013). MySQL. Recuperado el 11 de Marzo de 2018 de <https://searchdatacenter.techtarget.com/es/definicion/MySQL>
- Rouse, M. (2015). TechTarget. Recuperado el 08 de Marzo de 2018 de <https://searchdatacenter.techtarget.com/es/definicion/SQL-Server>
- SearchDataCenter. (2016). Requerimientos de una LAN inalámbrica (WLAN). Recuperado el 14 de Abril de 2018 de <https://searchdatacenter.techtarget.com/es/consejo/Requerimientos-de-una-LAN-inalambrica-WLAN>
- Semenov, A. (2005). Las tecnologías de la información y la comunicación en la enseñanza . Paris. 59-66. Recuperado el 25 de Febrero de 2018 de <http://unesdoc.unesco.org/images/0013/001390/139028s.pdf>
- SiguePasos.blogspot. (2015). Introducción a Arduino I: ¿Qué es?, ¿Para que sirve? y ¿Cuántos modelos existen?. Recuperado el 28 de Marzo de 2018 de <http://siguepasos.blogspot.com/2015/07/IntroduccionArduino-I.html>
- SparkFun. (2016). *Fingerprint Scanner - TTL (GT-511C3)*. Recuperado el 23 de Febrero de 2018 de <https://www.sparkfun.com/products/retired/11792>
- Talens-Oliag, S. (2011). Seguridad física. Recuperado el 31 de Marzo de 2018 de <https://www.uv.es/sto/cursos/icssu/html/ar01s04.html>
- TOGELL. (2015). FILEDIAGRAMA DE TRANSISTOR NPN SVG. Recuperado el 07 de Enero de 2018 de <http://togell.com/yl/u1113w6/transistor-svg/263ik1/>
- Tolosa, C. (2016). Sistemas Biométricos. Recuperado el 14 de Abril de 2018 de <https://www.dsi.uclm.es/personal/MiguelFGraciani/mikicurri/Docencia>

/Bioinformatica/web_BIO/Documentacion/Trabajos/Biometria/Trabajo
%20Biometria.pdf

UPV. (2009). El reconocimiento facial en 3D da nuevos pasos . Recuperado el 15 de Febrero de 2018 de <https://www.agenciasinc.es/Noticias/El-reconocimiento-facial-en-3D-da-nuevos-pasos>

US. (2014). Redes inalámbricas de área local (WLAN). 98. Recuperado el 31 de Marzo de 2018 de <http://bibing.us.es/proyectos/abreproy/11761/fichero/Volumen1%252F7-Cap%C3%ADtulo3+-+Redes+inal%C3%A1mbricas+de+%C3%A1rea+local+%28WLAN%29.pdf>

Visual Studio. (2018). Visual Studio. Recuperado el 07 de Marzo de 2018 de <https://code.visualstudio.com/docs/languages/csharp>

w3layouts. (2016). Que es infrarrojo. Recuperado el 02 de Febrero de 2018 de <https://www.black-thursday.com/foto/que-es-infrarrojo.html>

Web Hosting. (2018). *Tuning Apache* (httpd). Recuperado el 20 de Febrero de 2018 de http://www.webhostingchennai.co.in/blog/vps/tunning-apache-httpd-mpm_worker-cpanel/

Welivesecurity. (2015). ¿Cómo funciona la tecnología de reconocimiento facial?. Recuperado el 16 de Junio de 2018 de <https://www.welivesecurity.com/la-es/2015/08/27/tecnologia-de-reconocimiento-facial/>

Wordpress. (2015). Lenguaje de programación C++. Recuperado el 08 de Febrero de 2018 de <https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-c/>

Zambrano, M. (2012). Construcción de un sistema electrónico de control de acceso y registro de entrada-salida mediante lectores biométricos.

ESPN, 115. Recuperado el 18 de Enero de 2018 de <http://bibdigital.epn.edu.ec/bitstream/15000/4572/1/CD-4199.pdf>

Zúniga, C. (s.f.). Raspberry Pi 3 Características Técnicas. Recuperado el 28 de Marzo de 2018 de <http://www.proyectobyte.com/android/raspberry-pi-3-caracteristicas-tecnicas>

ANEXOS

```

from serial import SerialException
import time
import RPi.GPIO as GPIO # para manejar la gpio de la rasp
import MySQLdb

conexion = True
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(26, GPIO.OUT)

db = MySQLdb.connect(host="localhost", # your host
                    user="root",      # username
                    passwd="root",    # password
                    db="rasphuella") # name of the databa

# Create a Cursor object to execute queries.
cur = db.cursor()

# INTENTA CONECTARSE AL BLUETOOTH
while conexion:
    try:
        puerto_serial = serial.Serial("/dev/rfcomm0",
        conexion = False
        print '\nConectado correctamente.\n\n'
    except SerialException:
        print 'Error conexion a bluetooth. Intentando en
        time.sleep(10)
        conexion = True

# SI ESTA CONECTADO REALIZA LA LECTURA
if conexion==False:
    #estado = True;
    #contador = 0;
    print '\nLegendo datos por bluetooth...\n\n'

    while True:
        dato = puerto_serial.read(512)
        if dato:
            print 'Dato leido: ', dato
            vec = dato.split(':')

            if (vec[0]=="R"):
                # registro de nuevo id de usuario
                f = open("data.txt", "w")

"lectura.py" 80 lines, 1944 characters

```

```

conexion==False:
#estado = True;
#contador = 0;
print '\nLeyendo datos por bluetooth...\n\n'

while True:
    dato = puerto_serial.read(512)
    if dato:
        print 'Dato leído: ', dato
        vec = dato.split(':')

        if (vec[0]=="R"):
            # registro de nuevo id de usuario
            f = open("data.txt", "w")
            f.write(vec[1])
            f.close()
            print "Registro ID correcto."

        if (vec[0]=="A"):
            # acceso de usuario y comparacion con base de datos existente
            sql = "SELECT * FROM usuarios WHERE huella=" + vec[1]
            try:
                cur.execute(sql)
            except:
                print ""
            user=''
            for row in cur.fetchall():
                user = row[0]

            if (user!=''):
                puerto_serial.write("E\n") #envia al arduino E de Existe
                GPIO.output(26, True)
                print "Usuario registrado."
                time.sleep(2)
                GPIO.output(26, False)
                print "Ok!"
            else:
                puerto_serial.write("N\n") #envia al arduino N de No Existe
                print "Usuario no encontrado."
                time.sleep(1)

        vec=[]
        time.sleep(1)

    puerto_serial.close()
    print 'Proceso terminado.'

```

```

from serial import SerialException
import time
import RPi.GPIO as GPIO # para manejar la gpio de la rasp
import MySQLdb

conexion = True
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(26, GPIO.OUT)

db = MySQLdb.connect(host="localhost", # your host
                     user="root",      # username
                     passwd="root",    # password
                     db="rasphuella")  # name of the databa

# Create a Cursor object to execute queries.
cur = db.cursor()

# INTENTA CONECTARSE AL BLUETOOTH
while conexion:
    try:
        puerto_serial = serial.Serial("/dev/rfcomm0",
        conexion = False
        print '\nConectado correctamente.\n\n'
    except SerialException:
        print 'Error conexion a bluetooth. Intentando en
        time.sleep(10)
        conexion = True

# SI ESTA CONECTADO REALIZA LA LECTURA
if conexion==False:
    #estado = True:
    #contador = 0;
    print '\nLeyendo datos por bluetooth...\n\n'

    while True:
        dato = puerto_serial.read(512)
        if dato:
            print 'Dato leído: ', dato
            vec = dato.split(':')

            if (vec[0]=="R"):
                # registro de nuevo id de usuario
                f = open("data.txt", "w")
                f.write(vec[1])
                f.close()
                print "Registro ID correcto."

            if (vec[0]=="A"):
                # acceso de usuario y comparacion con base de datos existente
                sql = "SELECT * FROM usuarios WHERE huella=" + vec[1]
                try:
                    cur.execute(sql)
                except:
                    print ""
                    user=""
                    for row in cur.fetchall():
                        user = row[0]

                if (user!=""):
                    puerto_serial.write("E\n") #envia al arduino E de Existe
                    GPIO.output(26, True)
                    print "Usuario registrado."
                    time.sleep(2)
                    GPIO.output(26, False)
                    print "Ok!"
                else:
                    puerto_serial.write("N\n") #envia al arduino N de No Existe
                    print "Usuario no encontrado."
                    time.sleep(1)

            vec=""
            time.sleep(1)
        puerto_serial.close()
    print 'Proceso terminado.'

```

```

conexion=False:
#estado = True:
#contador = 0;
print '\nLeyendo datos por bluetooth...\n\n'

while True:
    dato = puerto_serial.read(512)
    if dato:
        print 'Dato leído: ', dato
        vec = dato.split(':')

        if (vec[0]=="R"):
            # registro de nuevo id de usuario
            f = open("data.txt", "w")
            f.write(vec[1])
            f.close()
            print "Registro ID correcto."

        if (vec[0]=="A"):
            # acceso de usuario y comparacion con base de datos existente
            sql = "SELECT * FROM usuarios WHERE huella=" + vec[1]
            try:
                cur.execute(sql)
            except:
                print ""
                user=""
                for row in cur.fetchall():
                    user = row[0]

            if (user!=""):
                puerto_serial.write("E\n") #envia al arduino E de Existe
                GPIO.output(26, True)
                print "Usuario registrado."
                time.sleep(2)
                GPIO.output(26, False)
                print "Ok!"
            else:
                puerto_serial.write("N\n") #envia al arduino N de No Existe
                print "Usuario no encontrado."
                time.sleep(1)

        vec=""
        time.sleep(1)
    puerto_serial.close()
print 'Proceso terminado.'

```

```
<?php
/**
 * Representa el la estructura de las metas
 * almacenadas en la base de datos
 */
require 'conexion.php';

class Meta
{
    function __construct()
    {
    }

    /**
     * Retorna en la fila especificada de la tabla 'meta'
     *
     * @param $idMeta Identificador del registro
     * @return array Datos del registro
     */
    public static function Consulta($sql)
    {
        $consulta = $sql;
        try {
            // Preparar sentencia
            $comando = Database::getInstance()->getDb()->prepare($consulta);
            // Ejecutar sentencia preparada
            $comando->execute();

            return $comando->fetchAll(PDO::FETCH_ASSOC);
        } catch (PDOException $e) {
            return false;
        }
    }

    /**
     * Obtiene los campos de una meta con un identificador
     * determinado
     *
     * @param $id_test Identificador de la meta
     * @return mixed
     */
    public static function Ejecutar($sql) {
        // Consulta de la meta
        $consulta = $sql;


```

```

        $comando = Database::getInstance()->getDb()->prepare($consulta);
        // Ejecutar sentencia preparada
        $comando->execute();

        return 1;
    } catch (PDOException $e) {
        // Aquí puedes clasificar el error dependiendo de la excepción
        // para presentarlo en la respuesta Json
        return -1;
    }
}

/**
 * Obtiene los campos de una meta con un identificador
 * determinado
 *
 * @param $id_test Identificador de la meta
 * @return mixed
 */
public static function Consulta_Unico($sql) {
    // Consulta de la meta
    $consulta = $sql;

    try {
        // Preparar sentencia
        $comando = Database::getInstance()->getDb()->prepare($consulta);
        // Ejecutar sentencia preparada
        $comando->execute();
        // Capturar primera fila del resultado
        $row = $comando->fetch(PDO::FETCH_ASSOC);
        return $row;
    } catch (PDOException $e) {
        // Aquí puedes clasificar el error dependiendo de la excepción
        // para presentarlo en la respuesta Json
        return -1;
    }
}

/**
 * Insertar una nueva meta
 *
 * @param $id_test Identificador
 * @param $nombre nuevo titulo
 * @param $edad nueva description
 * @return PreparedStatement
 */

```

```

*/
public static function Ejecutar_Sentencia($sql)
{
    try{

        // Sentencia INSERT
        $comando = $sql;

        // Preparar la sentencia
        $sentencia = Database::getInstance()->getDb()->prepare($comando);

        return $sentencia->execute();

        return 1;

    } catch (PDOException $e) {
        // Aquí puedes clasificar el error dependiendo de la excepción
        // para presentarlo en la respuesta Json
        return -1;
    }

}

/**
 * Insertar una nueva meta
 * @param $id_test      identificador
 * @param $nombre      nuevo título
 * @param $edad        nueva descripción
 * @return PDOStatement
 */
public static function Nuevo_Usuario($cedula, $nombres, $telefono, $celular, $
{
    // Sentencia INSERT
    $comando = "INSERT INTO usuarios (cedula, nombres, telefono, celular, corre
    // Preparar la sentencia
    $sentencia = Database::getInstance()->getDb()->prepare($comando);

    return $sentencia->execute(array($cedula, $nombres, $telefono, $celular, $cor

}

/**
 * Eliminar el registro con el identificador especificado
 * @param $idmeta identificador de la meta

```

```

// para presentarlo en la respuesta Json
return -1;
}

/**
 * Insertar una nueva meta
 * @param $id_test      identificador
 * @param $nombre      nuevo título
 * @param $edad        nueva descripción
 * @return PDOStatement
 */
public static function Nuevo_Usuario($cedula, $nombres, $telefono, $celular, $correo, $huella)
{
    // Sentencia INSERT
    $comando = "INSERT INTO usuarios (cedula, nombres, telefono, celular, correo, huella) VALUES (?,?,?,?,?,?)";
    // Preparar la sentencia
    $sentencia = Database::getInstance()->getDb()->prepare($comando);

    return $sentencia->execute(array($cedula, $nombres, $telefono, $celular, $correo, $huella));
}

/**
 * Eliminar el registro con el identificador especificado
 * @param $idmeta identificador de la meta
 * @return bool Respuesta de la eliminación
 */
public static function Actualizar_Campo($tabla, $campo, $valor, $id, $id_valida)
{
    // Sentencia DELETE
    $comando = "UPDATE ".$tabla." SET ".$campo."=? WHERE ".$id."=?";
    // Preparar la sentencia
    $sentencia = Database::getInstance()->getDb()->prepare($comando);

    return $sentencia->execute(array($valor, $id_valida));
}

```

```

<?php
// Obtiene las credenciales para la conexión a la bd
require_once "config.php";

class Databasef

/**
 * Unica instancia de la clase
 */
private static $db = null;

/**
 * Instancia de PDO
 */
private static $pdo;

final private function __construct(){
    try {
        // Crear nueva conexión PDO
        self::getDb();
    } catch (PDOException $e) {
        // Manejo de excepciones
        print 'Error! : ' . $e->getMessage(). '<br/>';
        die();
    }
}

/**
 * Retorna en la única instancia de la clase
 * @return Databasefnull
 */
public static function getInstance()
{
    if (self::$db == null) {
        self::$db = new self();
    }
    return self::$db;
}

/**
 * Crear una nueva conexión PDO basada
 * en los datos de conexión
 * @return PDO Objeto PDO
 */
public function getDb()
{
    if (self::$pdo == null) {

```

```

*/
public static function getInstance()
{
    if (self::$db == null) {
        self::$db = new self();
    }
    return self::$db;
}

/**
 * Crear una nueva conexión PDO basada
 * en los datos de conexión
 * @return PDO Objeto PDO
 */
public function getDb()
{
    if (self::$pdo == null) {
        self::$pdo = new PDO(
            'mysql:dbname=' . DATABASE .
            ';host=' . HOSTNAME .
            ';port:63343;',
            USERNAME,
            PASSWORD,
            array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8")
        );

        // Habilitar excepciones
        self::$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    }

    return self::$pdo;
}

/**
 * Evita la clonación del objeto
 */
final protected function __clone()
{
}

function __destruct()
{
    self::$pdo = null;
}
}

```



```

<table class="table table-hover table-striped table-border">
  <thead>
    <tr>
      <td width="10%"><strong>CEDULA</strong></td>
      <td><strong>NOMBRES</strong></td>
      <td width="10%"><strong>TELEFONO</strong></td>
      <td width="10%"><strong>CELULAR</strong></td>
      <td width="20%"><strong>CORREO ELECTRONICO</strong></td>
      <td width="20%"><strong>HUELLA</strong></td>
    </tr>
  </thead>
  <tbody id="listado">
</tbody>
</table>


```

```

<label for="exampleInputEmail1">C&eacute;dula</label>
<input type="number" class="form-control" id="cedula_mod" required="true">
</div>
<div class="form-group col-md-8">
<label for="exampleInputEmail1">Nombres Completos</label>
<input type="text" class="form-control" id="nombres_mod" required="true">
</div>
</div>
<div class="row">
<div class="form-group col-md-4">
<label for="exampleInputEmail1">Tel&eacute;fono</label>
<input type="text" class="form-control" id="telefono_mod">
</div>
<div class="form-group col-md-4">
<label for="exampleInputEmail1">Celular</label>
<input type="text" class="form-control" id="celular_mod">
</div>
<div class="form-group col-md-4">
<label for="exampleInputEmail1">Correo Electr&eacute;nico</label>
<input type="text" class="form-control" id="correo_mod">
</div>
</div>
<div class="row">
<div class="form-group col-md-12">
<label for="exampleInputEmail1">C&oacute;digo Huella</label>
<input type="text" class="form-control" id="huella_mod" required="true" readonly="true">
</div>
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-default" data-dismiss="modal">Cerrar</button>
<button type="submit" class="btn btn-info" onclick="Leer_archivo(1); return false;">Leer Huella Digital</button>
<button type="submit" class="btn btn-primary">Modificar Usuario</button>
</div>
</form>
</div><!-- /.modal-content -->
</div><!-- /.modal-dialog -->

```

```

</div>
<div class="modal-footer">
<button type="button" class="btn btn-default" data-dismiss="modal">Cerrar</button>
<button type="submit" class="btn btn-info" onclick="Leer_archivo(1); return false;">Leer Huella Digital</button>
<button type="submit" class="btn btn-primary">Modificar Usuario</button>
</div>
</form>
</div><!-- /.modal-content -->
</div><!-- /.modal-dialog -->
</div><!-- /.modal -->
<!-- Modal -->
<div class="modal fade" id="eliminar_usuario" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
<h4 class="modal-title" id="myModalLabel">Eliminar Usuario</h4>
</div>
<div class="modal-body">
<p>Desea eliminar el usuario seleccionado?</p>
<input type="hidden" value="0" id="id_eliminar">
</div>
<div class="modal-footer">
<button type="button" class="btn btn-default" data-dismiss="modal">Cerrar</button>
<button type="button" class="btn btn-danger" onclick="Eliminar_usuario(); return false;">Eliminar Usuario</button>
</div>
</div><!-- /.modal-content -->
</div><!-- /.modal-dialog -->
</div><!-- /.modal -->
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="js/vendor/jquery-1.11.2.min.js"></script>')</script>
<script src="js/vendor/bootstrap.min.js"></script>
<script src="js/main.js"></script>
</body>
</html>

```

```

    Usuarios();
  })

  function Usuarios(){
    $.ajax({
      url: 'php/lista_usuarios.php',
      dataType: 'html',
      success: function(datos) {
        $('#listado').html(datos);
      },
      error: function(e){
        console.log(e.responseText);
      }
    });
  }

  function Nuevo_Usuario(){
    $('#modal_usuario').modal('show');
  }

  $('#nuevo_usuario').submit(function(event) {
    event.preventDefault();
    var datos_form= {
      cedula: document.getElementById('cedula').value,
      nombres: document.getElementById('nombres').value,
      telefono: document.getElementById('telefono').value,
      celular: document.getElementById('celular').value,
      correo: document.getElementById('correo').value,
      huella: document.getElementById('huella').value
    };
    $.ajax({
      url: 'php/guardar_usuario.php',
      data: {
        formulario: datos_form
      },
      type: 'POST',
      dataType: 'json',
      success: function(datos) {
        if (datos['estado']){
          $('#nuevo_usuario')[0].reset();
          $('#modal_usuario').modal('hide');
          Usuarios();
        }
      },
      error: function(e){
        console.log(e.responseText);
      }
    });
  });
}

```

```

    $('#eliminar_usuario').modal('show');
  }

  function Eliminar_Usuario(){
    var id = document.getElementById('id_eliminar').value;
    if (id!=0){
      $.ajax({
        url: 'php/eliminar_usuarios.php',
        data: {
          id_usuario: id
        },
        type: 'POST',
        dataType: 'json',
        success: function(datos) {
          if (datos['estado']){
            $('#eliminar_usuario').modal('hide');
            document.getElementById('id_eliminar').value = 0;
            Usuarios();
          }
        },
        error: function(e){
          console.log(e.responseText);
        }
      });
    }
  }

  function Editar_Usuario(id){
    $.ajax({
      url: 'php/obtener_usuario.php',
      data: {
        id_usuario: id
      },
      type: 'POST',
      dataType: 'json',
      success: function(datos) {
        if (datos['estado']){
          document.getElementById('id_modifica').value = datos['id_usuario'];
          document.getElementById('cedula_mod').value = datos['cedula'];
          document.getElementById('nombres_mod').value = datos['nombres'];
          document.getElementById('telefono_mod').value = datos['telefono'];
          document.getElementById('celular_mod').value = datos['celular'];
          document.getElementById('correo_mod').value = datos['correo'];
          document.getElementById('huella_mod').value = datos['huella'];
          $('#modal_usuario_existe').modal('show');
        }
      }
    });
  }
}

```

```
        console.log(e.responseText);
    });
}

$('#editar_usuario').submit(function(event) {
    event.preventDefault();
    var datos_form = {
        id_usuario: document.getElementById('id_modifica').value,
        cedula: document.getElementById('cedula_mod').value,
        nombres: document.getElementById('nombres_mod').value,
        telefono: document.getElementById('telefono_mod').value,
        celular: document.getElementById('celular_mod').value,
        correo: document.getElementById('correo_mod').value,
        huella: document.getElementById('huella_mod').value
    };
    $.ajax({
        url: 'php/modificar_usuario.php',
        data: {
            formulario: datos_form
        },
        type: 'POST',
        dataType: 'json',
        success: function(datos) {
            if (datos['estado']){
                $('#editar_usuario')[0].reset();
                $('#modal_usuario_existe').modal('hide');
                Usuarios();
            }
        },
        error: function(e){
            console.log(e.responseText);
        }
    });
});

function Leer_Archivo(cuadro){
    $.ajax({
        url: 'php/data.txt',
        dataType: 'text',
        success: function(datos) {
            if (cuadro==0){
                document.getElementById('huella').value = datos;
            }else{
                document.getElementById('huella_mod').value = datos;
            }
        }
    });
}
```

```

GTS11C3-IDENROLLLCD-Arreglado Arduino 1.8.3
Archivo Editar Programa Herramientas Ayuda
GTS11C3-IDENROLLLCD-Arreglado $
#include "FFS_GTS11C3.h"
#include "SoftwareSerial.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
const int rele=A3;
//#include <LiquidCrystal.h>
//#include <Wire.h>
//LiquidCrystal lcd(12, 11, A0, A1, A2, A3);
int rx=0;
// set up software serial pins for Arduino's w/ Atmega328P's
// FFS (TX) is connected to pin 4 (Arduino's Software RX)
// FFS (RX) is connected through a converter to pin 5 (Arduino's Software TX)
FFS_GTS11C3 ffs(2, 3); // (Arduino SS_RX = pin 4, Arduino SS_TX = pin 5)

// FFS (TX) is connected to pin 10 (Arduino's Software RX)
// FFS (RX) is connected through a converter to pin 11 (Arduino's Software TX)
//FFS_GTS11C3 ffs(10, 11); // (Arduino SS_RX = pin 10, Arduino SS_TX = pin 11)

void setup()
{
    //lcd.begin(16, 2);
    lcd.init();
    lcd.backlight();
}
314 Arduino Nano, ATmega328 en COM10
12:20
8/6/2018
```

```
GT511C3-IDENROLL1LCD-Arreglado Arduino 1.8.3
Archivo Editar Programa Herramientas Ayuda

GT511C3-IDENROLL1LCD-Arreglado $
...
lcd.init();
lcd.backlight();
pinMode(A1, INPUT);
pinMode(A2, INPUT);
digitalWrite(A1, HIGH);
digitalWrite(A2, HIGH);
pinMode(A3, OUTPUT);
Serial.begin(9600); //set up Arduino's hardware serial UART
delay(100);

...

lcd.clear();
lcd.setCursor(0, 0);
lcd.print(" CONTROL ACCESO ");
lcd.setCursor(0, 1);
lcd.print(" ***UIITEC*** ");
fps.Open(); //send serial command to initialize fps
fps.SetLED(true); //turn on LED so fps can see fingerprint
}

void enroll()
{
  // Enroll test

  // find open enroll id
  int enrollid = 0;
  <
  >
}

314 Arduino Nano, ATmega328 en COM10 12:21 8/6/2018
```

```
GT511C3-IDENROLL1LCD-Arreglado Arduino 1.8.3
Archivo Editar Programa Herramientas Ayuda

GT511C3-IDENROLL1LCD-Arreglado $
...
void enroll()
{
  // Enroll test

  // find open enroll id
  int enrollid = 0;
  bool usedid = true;
  while (usedid == true)
  {
    usedid = fps.CheckEnrolled(enrollid);
    if (usedid==true) enrollid++;
  }
  fps.EnrollStart(enrollid);

  // enroll
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("ENROLAR");
  lcd.setCursor(0, 1);
  lcd.print("INSERTE HUELLA");

  //Serial.print("Press finger to Enroll #");
  //Serial.println(enrollid);
  while(fps.IsPressFinger() == false) delay(100);
  bool bret = fps.CaptureFinger(true);
  int iret = 0;
  <
  >
}

314 Arduino Nano, ATmega328 en COM10 12:21 8/6/2018
```

```
GT511C3-IDENROLL1LCD-Arreglado Arduino 1.8.3
Archivo Editar Programa Herramientas Ayuda

GT511C3-IDENROLL1LCD-Arreglado $

//Serial.println("Remove finger");
fps.Enroll1();
while(fps.IsPressFinger() == true) delay(100);
  lcd.setCursor(0, 1);
  lcd.print("INSERTE HUELLA");

//Serial.println("Press same finger again");
while(fps.IsPressFinger() == false) delay(100);
bret = fps.CaptureFinger(true);
if (bret != false)
{
  lcd.setCursor(0, 1);
  lcd.print("RETIRE ");
  //Serial.println("Remove finger");
  fps.Enroll2();
  while(fps.IsPressFinger() == true) delay(100);
}
lcd.setCursor(0, 1);
  lcd.print("INSERTE HUELLA ");
//Serial.println("Press same finger yet again");
while(fps.IsPressFinger() == false) delay(100);
bret = fps.CaptureFinger(true);
if (bret != false)
{
  lcd.setCursor(0, 1);
  lcd.print("RETIRE ");
}

314 Arduino Nano, ATmega328 en COM10 12:21 8/6/2018
```

```
GT511C3-IDENROLL1LCD-Arreglado Arduino 1.8.3
Archivo Editar Programa Herramientas Ayuda

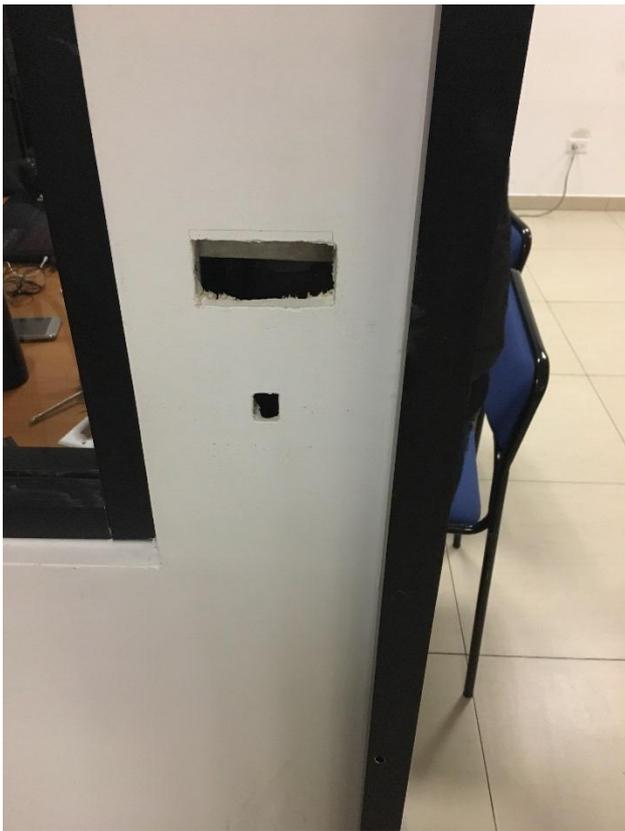
GT511C3-IDENROLL1LCD-Arreglado $

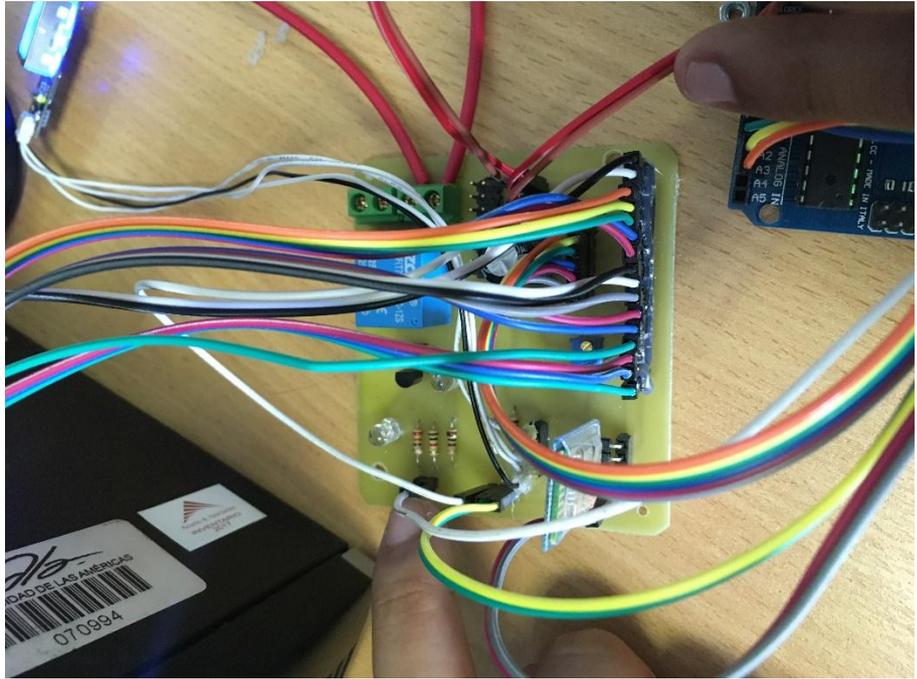
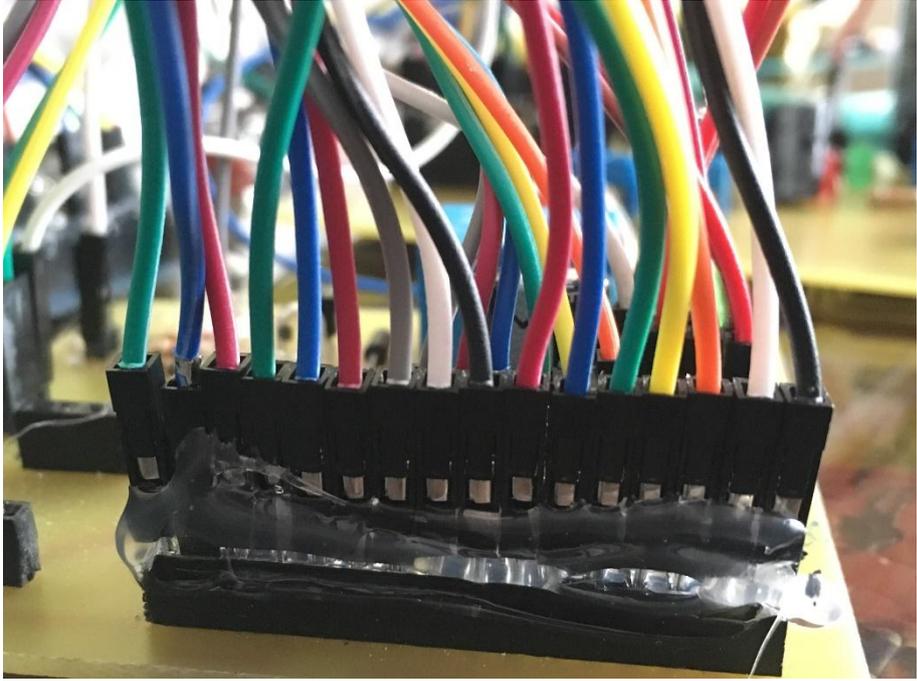
}

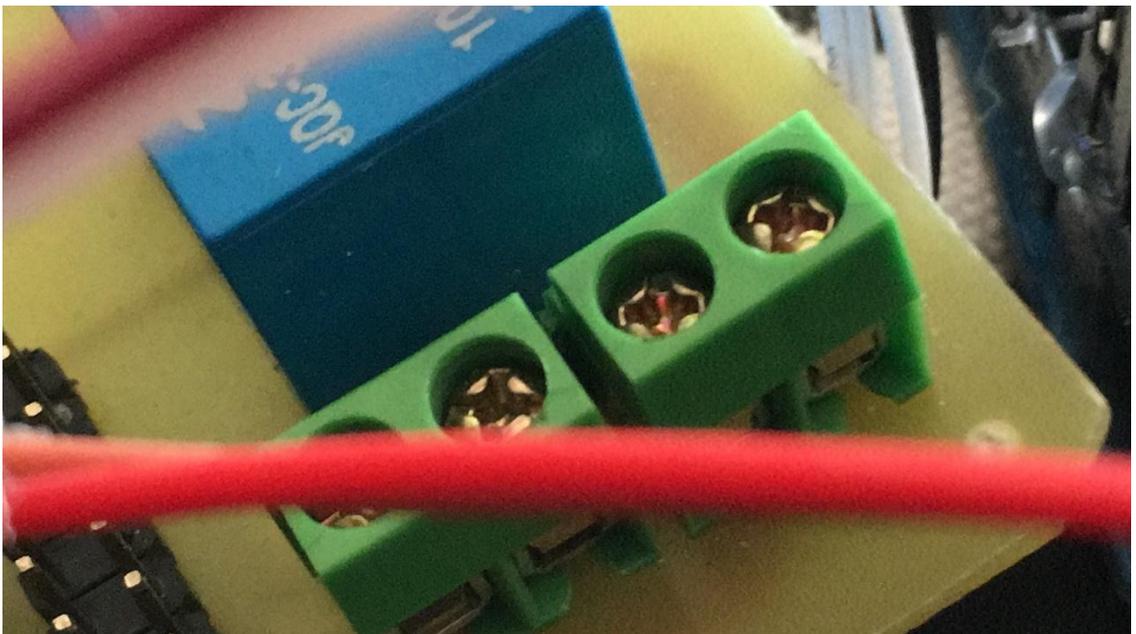
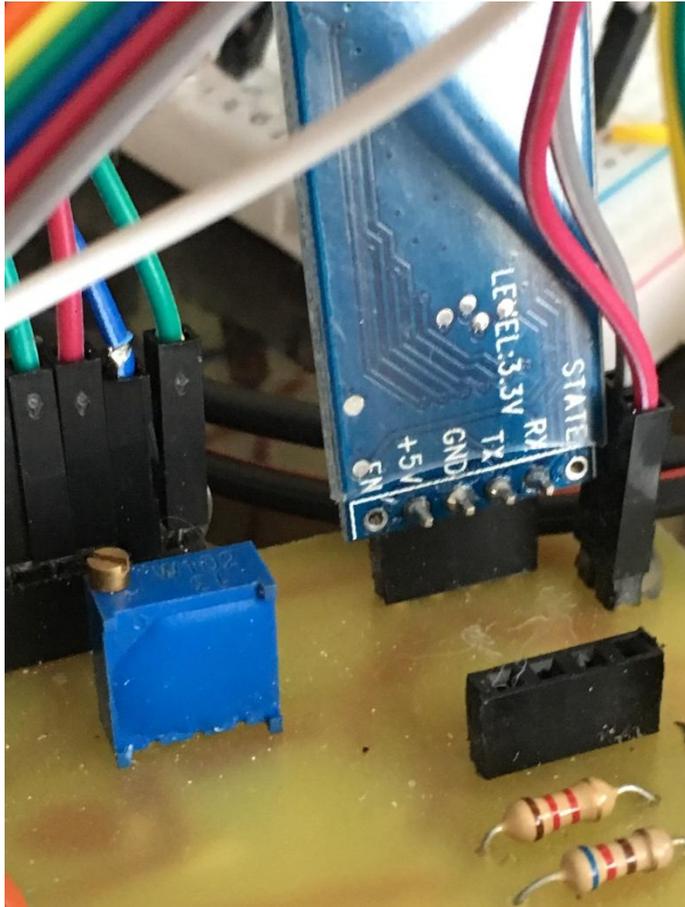
void identify(){
  lcd.clear();
  //lcd.setCursor(0, 0);
  //lcd.print("CONTROL ACCESO");
  lcd.setCursor(0, 1);
  lcd.print("LEYENDO HUELLA");
  int st=0;
  // Identify fingerprint test
  while(st==0){
    if (fps.IsPressFinger())
    {
      fps.CaptureFinger(false);
      int id = fps.Identifyf_N();

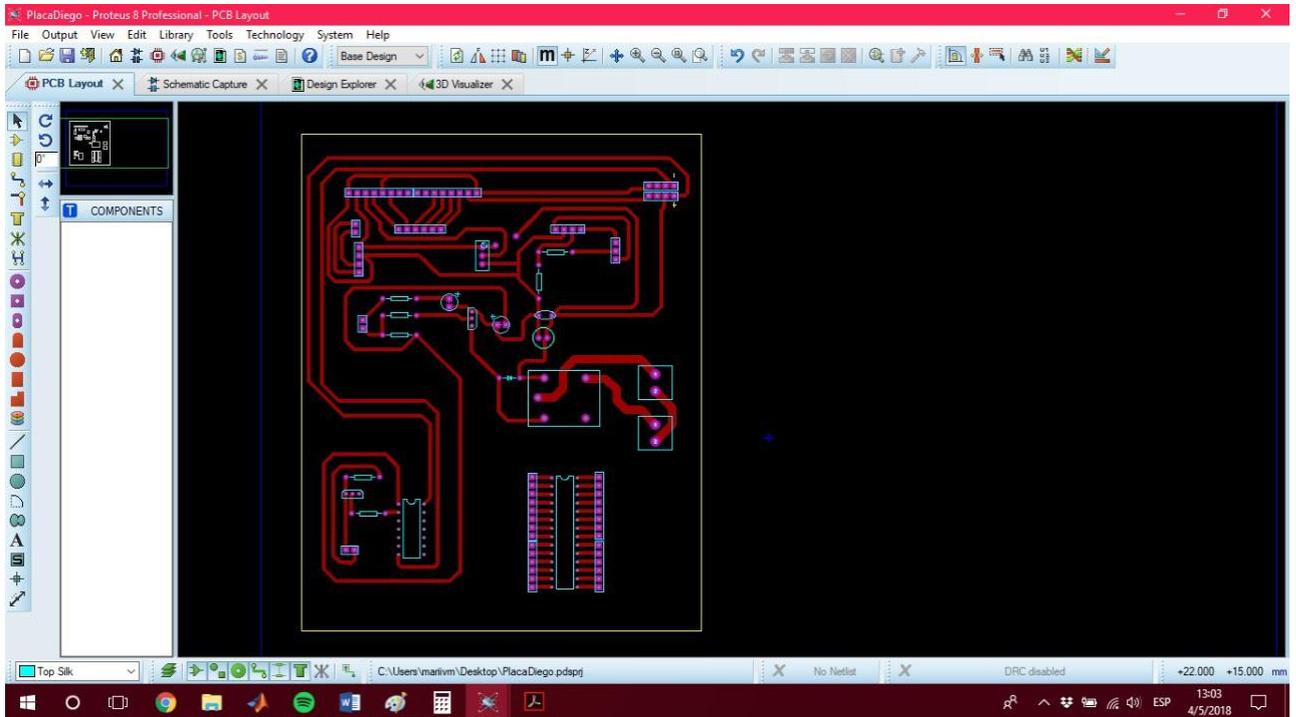
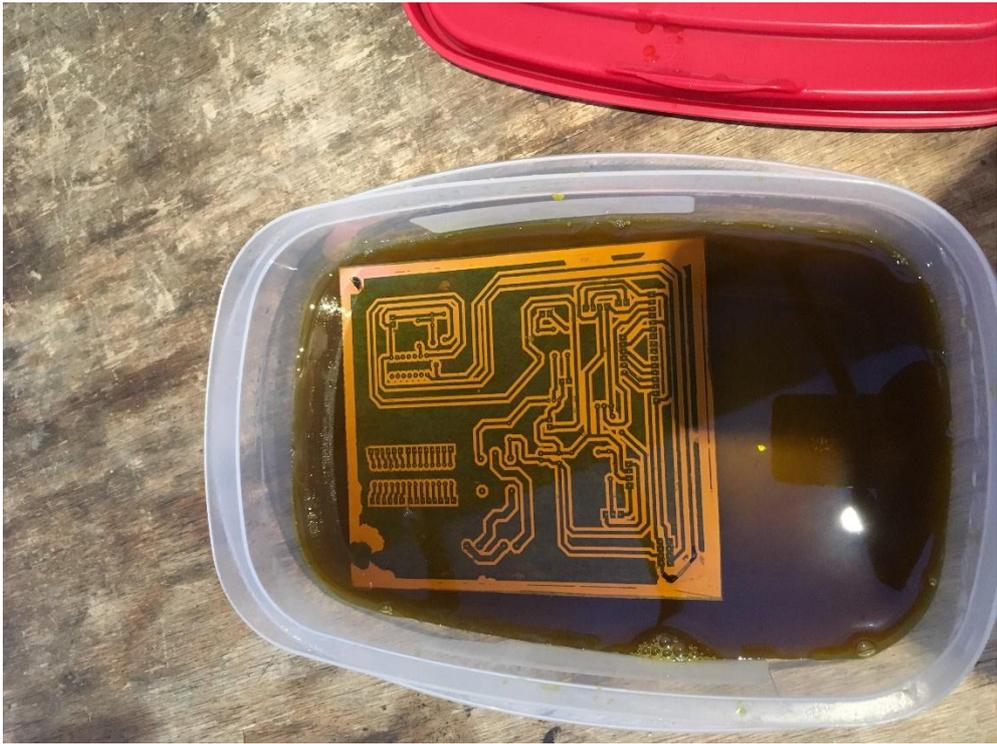
      /*Note : GT-511C3 can hold 200 fingerprint templates.
      GT-511C1R can hold 20 fingerprint templates.
      Make sure to change the id depending on what
      model you are using */
      if (id <200)
      {
        //If the fingerprint matches, provide the matching template ID
        //Serial.print("Verified ID:");
        //Serial.println(id);
        st=1;
        lcd.clear();
      }
    }
  }
}

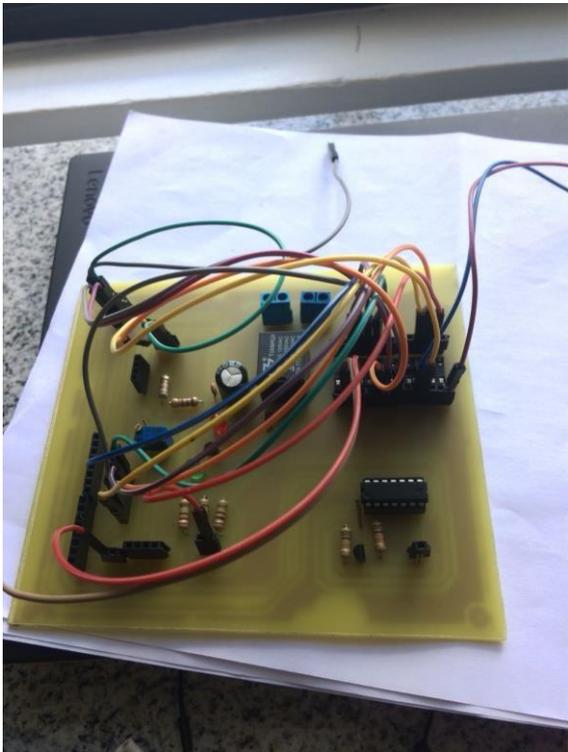
314 Arduino Nano, ATmega328 en COM10 12:22 8/6/2018
```

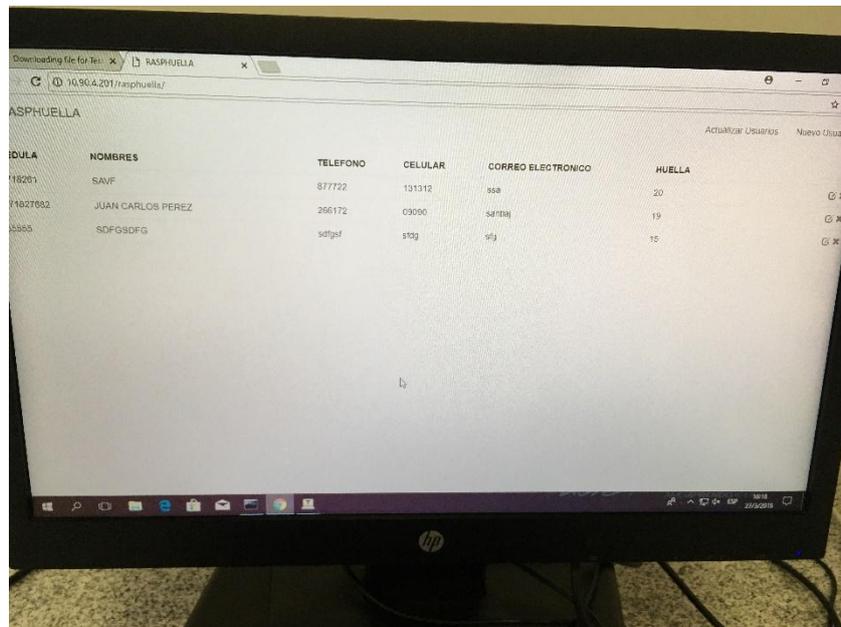
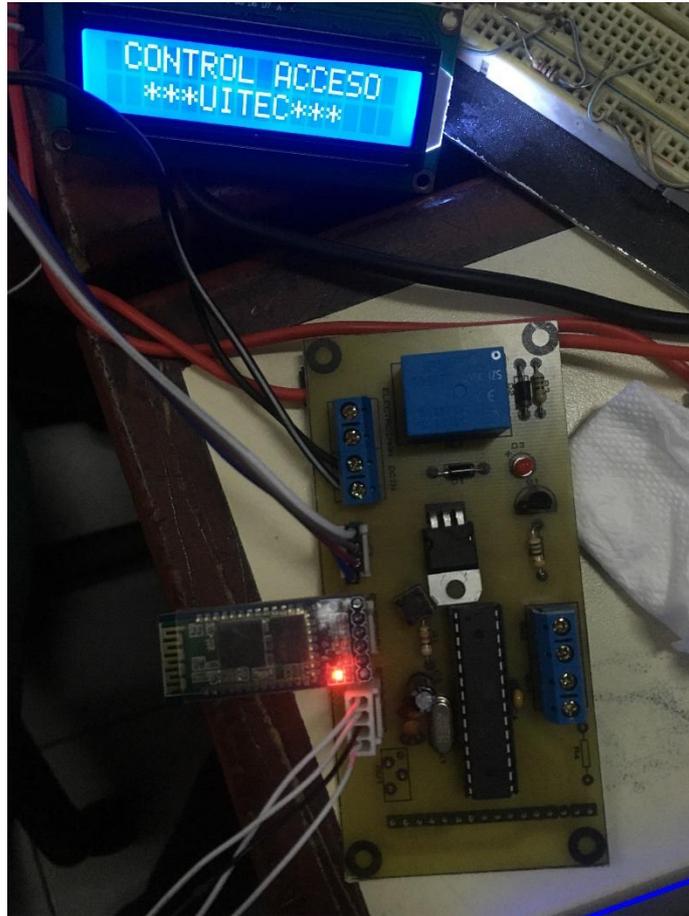










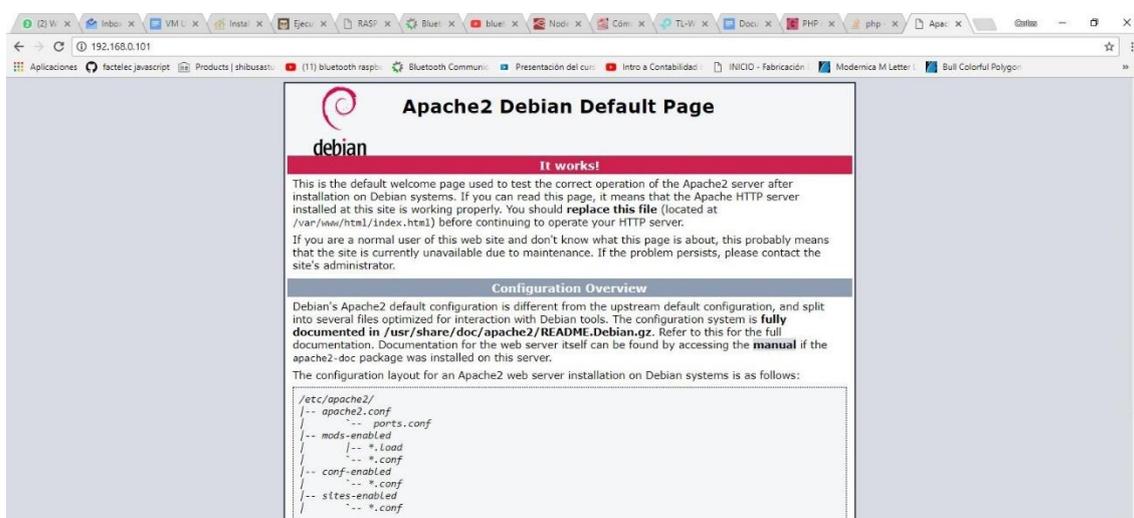


ANEXO MANUAL BIOMÉTRICO UITEC

INGRESO AL RASPBERRY	22
A TRAVÉS DE IP	22
COMANDOS PARA FUNCIONAMIENTO DE BIOMETRICO	24
CONEXIÓN AL MÓDULO BLUETOOTH	24
SCRIPT PYTHON PARA FUNCIONAMIENTO.....	24
ENROLAR USUARIOS	25
IDENTIFICACIÓN USUARIO	27

INGRESO AL RASPBERRY

1. Entramos desde cualquier navegador a la consola de administración del router, es nuestro caso, la dirección IP es **192.168.1.2** por defecto.
2. Una manera de determinar que la IP es la correcta, es entrando en su LAN y colocando la IP en cualquier navegador, se visualizará la página por defecto del servidor Apache.



- a. En caso de no responder a esta IP, se debe ingresar al módem, cuya IP es la **192.168.1.1** y usamos las credenciales: Usuario: **udla**
Contraseña: **udla2018**.

Iniciar sesión

http://192.168.1.1

Tu conexión con este sitio web no es privada

Nombre de usuario

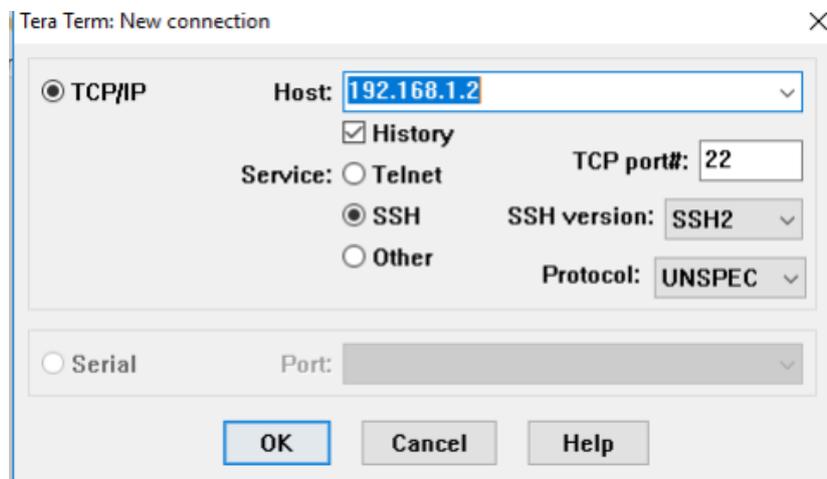
Contraseña

Debe existir la opción de LISTA DE CLIENTES / DHCP. En esta lista muestra los equipos que están conectados a la red. Deberá salir el nombre raspberrypi y debemos copiar su dirección IP.

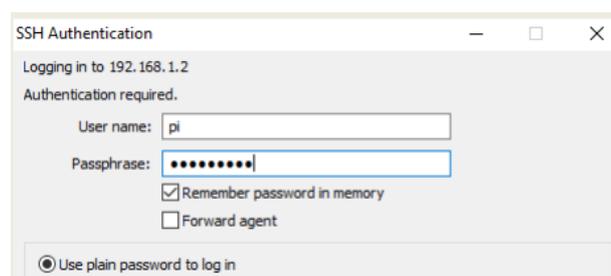
DHCP Table		
Host Name	IP Address	MAC Address
raspberrypi	192.168.1.2	B8:27:EB:A6:65:21
SAINTYAGO	192.168.1.3	34:17:EB:75:98:87

Submit

- Una vez que sabemos la IP del Raspberry, usamos el programa PUTTY, Tera Term, o cualquier otro parecido. Al ejecutar este software se desplegará una ventana donde pondremos la IP de la raspberry.
- Seleccionamos nueva conexión a través de SSH, y nos saldrá un mensaje de autorización para la conexión ssh, presionamos OK.



- Solicitará autorización a la consola de la raspberry, pidiendo credenciales: el usuario es: pi y la contraseña es: raspberry.



COMANDOS PARA FUNCIONAMIENTO DE BIOMETRICO

CONEXIÓN AL MÓDULO BLUETOOTH

6. Una vez ingresado al Raspberry con las credenciales mencionadas, procedemos a conectarnos al módulo Bluetooth.
7. Conectamos nuestro Raspberry Pi3 al módulo Bluetooth a través de su MAC, con el siguiente comando:

sudo rfcomm connect hci0 98:D3:31:FD:77:

```
pi@raspberrypi:~$ sudo rfcomm connect hci0 98:D3:31:FD:77:30
Connected /dev/rfcomm0 to 98:D3:31:FD:77:30 on channel 1
Press CTRL-C for hangup
```

SCRIPT PYTHON PARA FUNCIONAMIENTO

8. Abrimos una nueva sesión en Putty o Tera Term, y procedemos a instanciarnos en el directorio en el cual se encuentra nuestro script, mediante el comando:

cd /var/www/html/rasphuella/php

9. Por último, corremos nuestro programa, mediante el comando:

sudo python lectura.py

```
pi@raspberrypi:~$ cd /var/www/html/rasphuella/php
pi@raspberrypi:/var/www/html/rasphuella/php$ sudo python lectura.py
Conectado correctamente.

Leyendo datos por bluetooth...
```

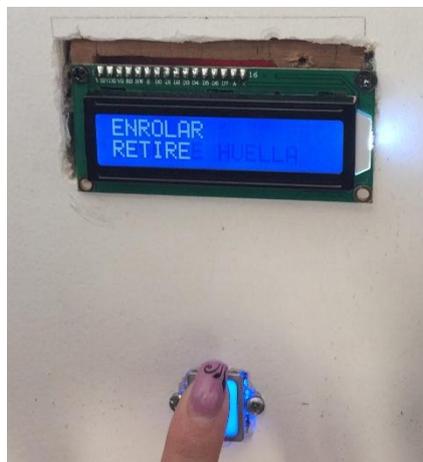
10. Con ambos procesos nuestro sistema funcionará correctamente.

ENROLAR USUARIOS

- Una vez realizado satisfactoriamente los pasos anteriores, podremos visualizar el LCD con el mensaje de inicio.



- Presionamos uno de los pulsadores que se encuentran en la pared de la UITEC, de esa manera se nos desplegará el mensaje de enrollar e insertar huella, a la cual debemos presionar nuestra huella dactilar 3 veces, la pantalla muestra el mensaje de retirar la huella.



- Al presionar nuestra huella 3 veces sin error, el módulo se encargará de almacenarla en el lector y nos proporcionará una ID, que es un número entero el cual se ingresará posteriormente al Raspberry, es decir a su base de datos.



- Es momento de ingresar a la página Web del sistema, a través de la IP de nuestro Raspberry, seguido de */rasphuella* tal y como se puede apreciar en la imagen.
- La interfaz de nuestra página Web es como la imagen a continuación, en la cual presionamos el botón Nuevo Usuario.

CEDULA	NOMBRES	TELEFONO	CELULAR	CORREO ELECTRONICO	HUELLA
1803610185	JORGE ROSERO				38
1717340143	DAVID FERNANDO POZO ESPIN	3111151	0998142421	david.pozo@udla.edu.ec	37
1723667919	SANTIAGO LEONARDO SOLORZANO LESCANO	0984139098	0984139098	santiago.solorzano@udla.edu.ec	35

- Llenamos los campos con nuestros datos personales.

Modificar Usuario

Cédula: 1717340143

Nombres Completos: DAVID FERNANDO POZO ESPIN

Teléfono: 3111151

Celular: 0998142421

Correo Electrónico: david.pozo@udla.edu.ec

Código Huella: 37

Cerrar Leer Huella Digital Modificar Usuario

- Una vez ingresado nuestros datos personales, procedemos a hacer clic en el botón “Leer Huella Digital” de esa manera, en el campo “Código Huella” nos aparecerá un número entero, debemos verificar que ese número sea el mismo que nos proporcionó el módulo Arduino al enrolar nuestra Huella.
- Por último presionamos Guardar Usuario y se ingresará la huella al sistema y base de datos.

NOTA: Por precaución es necesario volver a correr el script python con el comando

```
cd /var/www/html/rasphuella/php
```

```
sudo python lectura.py
```

IDENTIFICACIÓN USUARIO

- El sistema biométrico posee un loop o método en lazo que se ejecuta constantemente, por ello, si es que no se presiona ningún pulsador, el sistema estará siempre disponible para leer e identificar huellas dactilares, por lo tanto solo debemos presionar nuestra huella hasta que salga el mensaje:



- En caso de no existir la huella en la base, existirá un mensaje de “No Registrado”, y si la huella existe y se encuentra almacenada satisfactoriamente, se desplegará el siguiente mensaje en la pantalla LCD.



Por ende, el circuito activará el relé, y la chapa magnética se abrirá.



NOTA: En caso de querer abrir la puerta sin necesidad de ingresar la huella, se puede presionar el pulsador “Abrir”, el cual abre directamente la chapa magnética.

ADVERTENCIA: Si es que llegará a existir falla de energía eléctrica, va a ser necesario correr los 2 códigos mencionados en el manual.

