



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

“DISEÑO E IMPLEMENTACIÓN DE SISTEMA PARA TOMAR
AUTOMÁTICAMENTE ASISTENCIA A CLASE USANDO
VISIÓN ARTIFICIAL Y TECNOLOGÍAS MÓBILES”

INFORME DE PROYECTO INTEGRADOR

Previo a la obtención del Título de:

INGENIERO EN COMPUTACIÓN

ERICK JOEL ROCAFUERTE VILLÓN
CHRISTIAN ALEJANDRO JARAMILLO ESPINOZA

GUAYAQUIL – ECUADOR

AÑO: 2018

AGRADECIMIENTOS

A la vida.

Christian Alejandro Jaramillo Espinoza

AGRADECIMIENTOS

Agradezco a mis padres y hermanas por su comprensión y apoyo incondicional, a mis sobrinas, por ser mi mayor motivación, a Kaymara por estar siempre a mi lado aún en los momentos difíciles.

Erick Joel Rocafuerte Villón

DEDICATORIA

Para Ana, Anahí, Benjamín (🐾), Carola, Javier, Teresa y Tobías por su infinita paciencia y motivación.

Christian Alejandro Jaramillo Espinoza

DEDICATORIA

El presente proyecto se lo dedico a mis padres, sin ellos este logro no hubiera sido posible.

Erick Joel Rocafuerte Villón

TRIBUNAL DE EVALUACIÓN

.....
PhD. Daniel Ochoa D.
PROFESOR TUTOR

.....
PhD. Daniel Ochoa D.
PROFESOR COLABORADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual".

.....
Christian Jaramillo

.....
Erick Rocafuerte

RESUMEN

La toma de asistencia a clases es un proceso que debe realizar el profesor en cada sesión de clase, el mismo toma mayor tiempo a medida que se incrementa el número de alumnos en clase, se propone dar solución a esta problemática mediante el desarrollo de un sistema que incluya el teléfono del estudiante y del profesor, además de unas matrices led que emiten un código identificador.

La aplicación emisor del profesor se conecta mediante *bluetooth* a la matrices leds para transmitir el código identificador, la aplicación del estudiante decodifica lo que genera las matrices leds empleando técnicas de visión artificial y estima la posición del estudiante dentro del aula de clases y, por último, envía la información recolectada, como información personal, potencias de señales wifi alrededor del aula y el código decodificado a un servidor web que almacena y valida las asistencias.

El profesor puede obtener al instante un listado de todas las asistencias de sus estudiantes han sido aprobadas satisfactoriamente y observar su posición aproximada dentro del aula de clases.

Este proceso toma máximo 2 minutos para el profesor, lo que permite realizar este proceso varias veces en la misma sesión de clases.

ABSTRACT

The attendance of classes is a process that the teacher must perform in each class session, it takes more time as the number of students in class increases, it's proposed to solve this problem by developing a system that includes the student's telephone, the professor's telephone and a led matrix that emit an identifier code.

The teacher's application or transmitter is connected by bluetooth to the LED matrix to transmit the identifier code, the student's application decodes what generates the LED matrix using artificial vision techniques and estimates the student's position in the classroom and, finally, sends the collected information, such as personal information, powers of wifi signals around the classroom and the code decoded to a web server that stores and validates the assistance.

The teacher can instantly get a list of all the students have been satisfactorily approved and observe their approximate position within the classroom.

This process takes maximum 2 minutes for the teacher, which allows this process to be done several times in the same class session.

ÍNDICE GENERAL

RESUMEN	i
ABSTRACT	ii
ÍNDICE DE FIGURAS	v
CAPÍTULO 1	
1.MARCO GENERAL	1
1.1 Análisis de la problemática	1
1.2. Justificación	2
1.3. Objetivos	2
1.3.1. Objetivo principal	2
1.3.2. Objetivos específicos	3
1.4. Alcance	3
1.5. Resultados esperados	4
CAPÍTULO 2	
2. DISEÑO	5
2.1. Arquitectura de la solución	5
2.2. Diagrama de Despliegue	6
2.3. Hardware	7
CAPITULO 3	
3. IMPLEMENTACIÓN	11
3.1. Matrices Led	11
3.1.1. Elementos de la matriz	11
3.1.2. Programación del Microcontrolador	13
3.1.3. Diagrama de esquemático de la matriz led	13
3.2. Software de inicio de asistencia (emisor)	15
3.2.1. Recepción de información desde el servidor remoto	16
3.2.2. Envío de códigos a matriz led	18
3.2.3. Visualización de estudiantes	20
3.3. Software decodificador de asistencia (capturador)	21
3.3.1. Información del estudiante	21
3.3.2. Recepción de la imagen	21
3.3.3. Procesamiento de la imagen	22
3.3.4. Decodificación	24

3.3.5. Particionamiento	24
3.3.6. Fases de decodificación	25
3.3.7. Traducción	27
3.3.8. Estimación de la posición	27
3.3.9. Estimación de ángulos de rotación de la cámara	29
3.3.10. Corrección de perspectiva	32
3.3.11. Recopilación de información de la asistencia	34
3.4. Servidor web remoto	35
CAPÍTULO 4	
4. RESULTADOS OBTENIDOS	39
4.1. Resultados de la construcción de las matrices led	39
4.2. Resultados de decodificación	40
4.3. Resultados de estimación de posición	41

ÍNDICE DE FIGURAS

Figura 2.1: Arquitectura de la solución.	5
Figura 2.2: Diagrama de despliegue de la solución.	6
Figura 2.3: Diseño una matriz de led	7
Figura 2.4: Patrón de radiación ofrecido por los diodos led de 940nm	8
Figura 2.5: Diseño final de la matriz led con sus dimensiones	9
Figura 2.6: Representación de datos en la matriz led (primer encendido)	9
Figura 2.7: Representación de datos en la matriz led (segundo encendido)	10
Figura 2.8: Representación de datos en la matriz led (tercer encendido)	10
Figura 3.1: Diagrama del SN74HC595N	11
Figura 3.2: Diodo led rojo 625nm.	12
Figura 3.3: Módulo Bluetooth HC-06[6]	12
Figura 3.4: Flujograma de proceso presentación de datos en la matriz led.	13
Figura 3.5: Diagrama esquemático del circuito para un bloque de leds	14
Figura 3.6: Pantalla de Inicio de Sesión de la aplicación EMISOR	15
Figura 3.7: Listado de paralelo donde el profesor tiene identificador 1	16
Figura 3.8: Creación de tabla paralelo en la base de datos local	17
Figura 3.9: Diagrama de Flujo principal de la aplicación EMISOR	18
Figura 3.10: Mensaje a enviar a la matriz led.	19
Figura 3.11: Lista de estudiantes JSON y organizado por filas en la aplicación Emisor.	20
Figura 3.12: Ícono y pantalla donde ingresa su información personal.	21
Figura. 3.13: Ícono y pantalla del decodificador.	22
Figura 3.14: Diagrama de flujo del procesamiento de la imagen	22
Figura 3.15: Particionamiento de las matrices	25
Figura 3.16: Fase 1 de la decodificación.	25
Figura 3.17: Fase 2 de la decodificación.	25
Figura 3.18: Fase 3 de la decodificación.	26
Figura 3.19: Fase 4 de la decodificación.	26
Figura 3.20: Vista superior del modelo 'thin lens' usado en la solución.	27
Figura 3.21: Ángulos de rotación de la cámara.	29
Figura 3.22: Vanishing points estimados en imagen 640 x 480 pixeles.	30

Figura 3.23: Matrices sin corrección 45° con respecto al eje Z	33
Figura 3.24: Matrices con vista corregida	33
Figura 3.25. Diagrama de entidad relación de la base de datos.	36
Figura 3.26: Información enviada en Formato JSON.	37
Figura 4.1. Luz emitida por un diodo led de 940nm vista de frente (d), vista sesgada 45°	39
Figura 4.2. Matriz led con atenuación en el primer y tercer bloque de leds	39
Figura 4.3: Vista superior al aula de clases	40
Figura 4.4: Resultados de decodificación (% error)	40
Figura 4.5: Resultados de la coordenada 'Y' de la posición	41
Figura 4.6: Resultados de la coordenada 'X' de la posición	42

CAPÍTULO 1

1. MARCO GENERAL

1.1 Análisis de la problemática

En la actualidad la cantidad de alumnos en las materias denominadas básicas o comunes en las universidades ha venido incrementándose con el tiempo, pese a que las universidades no cuentan con los recursos apropiados la manera de satisfacer esta demanda es incrementar la cantidad de alumnos por salón de clases.

El incremento en la cantidad de alumnos por salón y la disminución de las horas clase por semana, convierten cada sesión de clase en un recurso valioso para profesor y alumnos y que no debe ser malgastado, a pesar de esto, en los reglamentos internos de los centros de educación superior del país tanto públicos como privados existen artículos referentes a la asistencia a clases y su respectivo control, el artículo 26 del reglamento de pregrado de la Escuela Superior Politécnica del Litoral ESPOL [1] indica que un estudiante reprueba automáticamente la materia cuando falta al 40% de las horas programadas del curso, y la ejecución de este artículo recae en el profesor como lo indica el artículo 27: *“El profesor de la materia es el responsable del ingreso en el sistema académico de los estudiantes incursos en el artículo inmediato anterior, hasta cinco días hábiles posteriores a la finalización del dictado de clases, según el calendario de Actividades académicas del término respectivo. El reporte escrito y el sustento respectivo...”*.

El tiempo promedio usado en la toma de asistencia en las aulas de clases varía entre 8 y 10 minutos, tiempo calculado en base a mediciones realizadas en varios salones de clase, según la cantidad de alumnos que va desde los 30 a 50 alumnos según datos del sistema ‘Académico en línea’ de la ESPOL para las materias de formación básica.

Considerando una materia de Formación Básica que tiene 2 sesiones por semana de 1 hora con 30 minutos, el tiempo usado es de 20 minutos a la semana, un 11% del total.

1.2. Justificación

En la actualidad los teléfonos inteligentes se han convertido en herramientas poderosas e imprescindibles en el quehacer diario de la persona, abarca una amplia gama de soluciones y oportunidades debido a su tamaño y versatilidad, ya no solo son utilizadas para mensajería o llamadas, sino que hoy en día es posible realizar muchas más actividades como el trabajo colaborativo, videoconferencias o juegos en línea. Según la Agencia de Control y Regulación de las Telecomunicaciones (ARCOTEL) en Ecuador a julio del 2017, el 46.4% de usuarios del servicio móvil poseen un *smartphone* [2] y para el año 2020 se estima que el porcentaje llegue a 64%. Por su popularidad entre los estudiantes universitarios es una gran alternativa que ayude a cumplir con el objetivo de realizar la tarea de tomar la asistencia de los estudiantes en el aula de clases de forma automatizada con ayuda del teléfono inteligente, porque ahorra tiempo valioso del horario de clases, esfuerzo y papeleo.

1.3. Objetivos

1.3.1. Objetivo principal

Implementar una solución tecnológica que permita automatizar el proceso de toma de asistencia a clases con ayuda de teléfonos inteligentes de los estudiantes.

1.3.2. Objetivos específicos

- Desarrollar un software que permita al profesor generar un código único en base a información propia de cada paralelo de cada materia.
- Desarrollar un dispositivo de hardware para emitir un código único para cada sesión de clase para cada materia dictada en un semestre determinado.
- Desarrollar un software que permita al estudiante capturar el código generado por un dispositivo para validar su asistencia a clases.

1.4. Alcance

La solución por implementar estará dividida en cuatro componentes:

- La primera es la aplicación móvil del profesor (emisor) que será capaz de iniciar el evento de la toma de asistencia, la aplicación envía un código identificador de la clase usando tecnología bluetooth a un dispositivo controlador.
- Dicho dispositivo estará conectado a unas matrices led en las cuales se codifica el código que se recibió por parte de la aplicación móvil del profesor.
- La tercera es la aplicación móvil del estudiante (capturador), que hace uso de la cámara del teléfono y usa técnicas de procesamiento digital de imágenes es capaz de decodificar el código mostrado en las matrices led y realizar una estimación de la posición del estudiante dentro del aula.
- La cuarta es una aplicación web con base de datos en un servidor remoto capaz de verificar la información enviada por el profesor y los estudiantes con el fin de aprobar o reprobar las asistencias recibidas.

1.5. Resultados esperados

Dado el alcance del proyecto se espera que el sistema pueda cotejar la información enviada por los estudiantes contra la información enviada del profesor para que al finalizar la toma de asistencia obtenga en su teléfono un informe con los estudiantes presentes en su clase, además de, poder visualizar la ubicación estimada de ellos dentro del aula.

CAPÍTULO 2

2. DISEÑO

2.1. Arquitectura de la solución

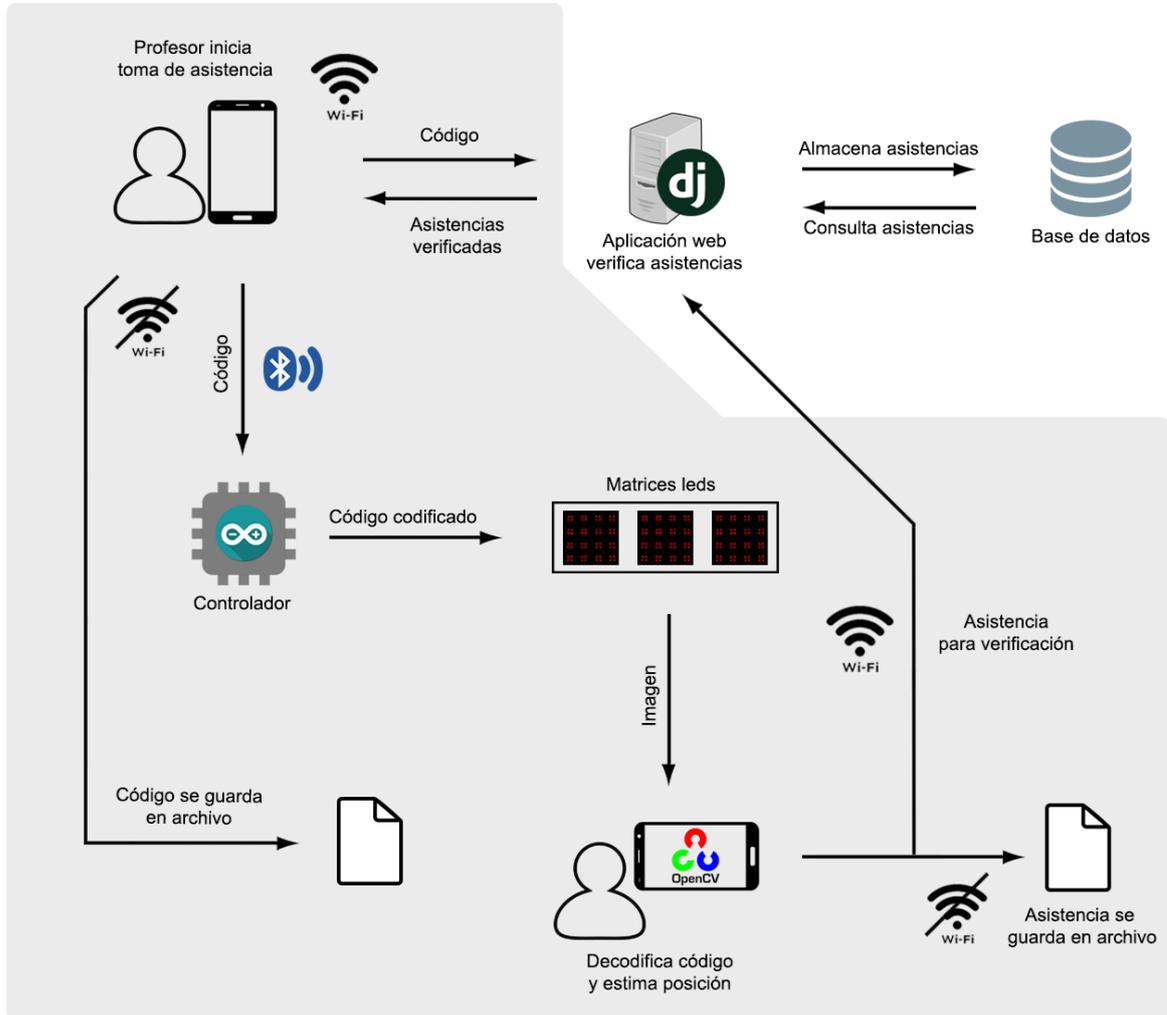


Figura 2.1: Arquitectura de la solución.

Componentes:

- Matrices led que mostrarán el código identificador.
- Aplicación móvil que enviará el código a la matriz de led (emisor).
- Aplicación móvil que decodifica el código identificador mostrado en la matrices led y estima la posición del estudiante dentro del aula (capturador).
- Servidor web remoto que almacenará la información enviada por el emisor y el capturador para verificar la asistencia a clase del estudiante.

2.2. Diagrama de Despliegue

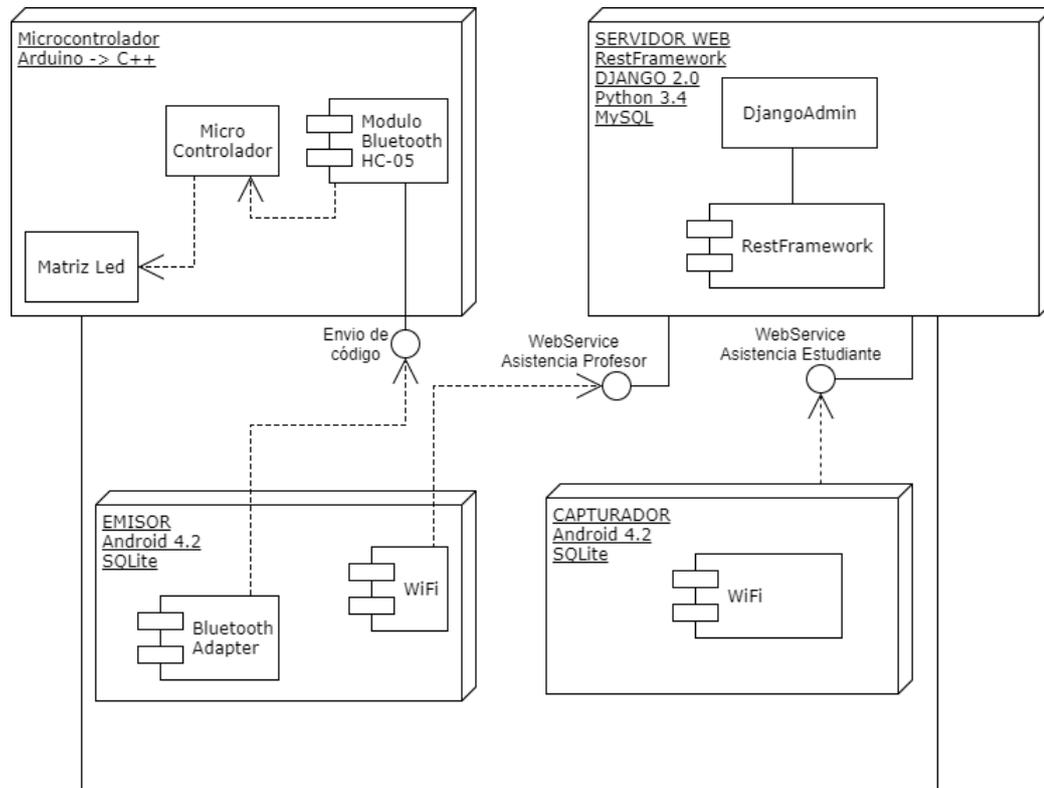


Figura 2.2: Diagrama de despliegue de la solución.

La figura 2.2 muestra los componentes más relevantes a usar de cada elemento del sistema, el microcontrolador que está conectado a la matriz led, será programado bajo su propio lenguaje, que es una adaptación de C++, del cual se usará la función de extender la cantidad de salidas que ofrece el microcontrolador y la librería *bluetooth* con la cual se ofrece una interfaz para la aplicación emisor.

La aplicación emisor será desarrollada para Android con Java y XML, usa una base de datos local que almacena información obtenida del servidor remoto, la aplicación emisor se conectará al servidor remoto a través de una interfaz de servicio web para obtener datos necesarios para realizar la codificación y para registrar las asistencias del profesor.

La aplicación capturador, usa la API de la cámara del teléfono para decodificar lo mostrado por la matriz led, esta información más otras recopilados del dispositivo se comparan, con las asistencias enviadas por

los profesores, en caso de coincidir, la asistencia se da por válida, este proceso se hace a través de un servicio web ofrecido por el servidor remoto.

El servidor remoto brinda una interfaz tipo formulario para que toda la información requerida para las aplicaciones sea ingresada al sistema antes de que este entre a operar, también almacena la información de las asistencias.

2.3. Hardware

El sistema tiene 3 matrices led construidas con 64 diodos emisores de luz (led) cada una, agrupados de 4 en 4 para tener redundancia en caso de que un led se queme, la matriz pueda seguir trabajando con los 3 restantes. Debido a los cálculos que se harán, la matriz tendrá la siguiente forma:

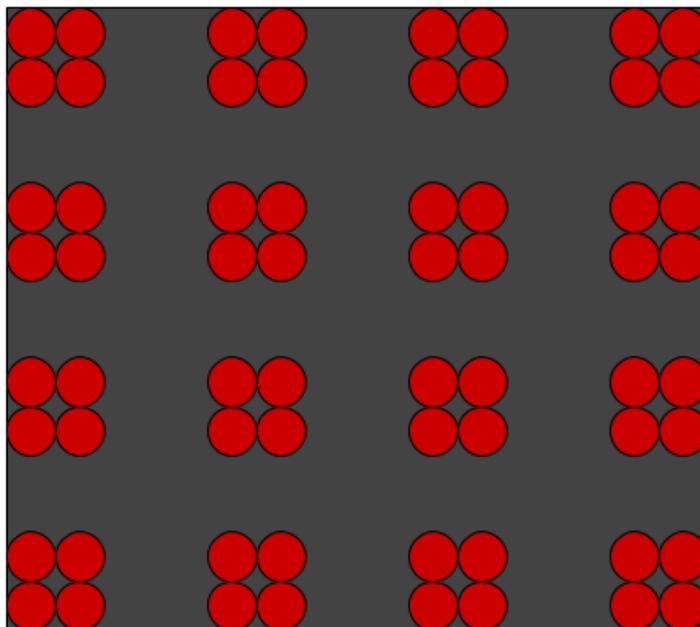


Figura 2.3: Diseño una matriz de led

Cada grupo de 4 led representará 1 bit que podrá ser 0 en caso de estar apagado o 1 en caso de estar encendido.

Ya que se representarán códigos correspondientes a identificadores de registros en el servidor remoto solo usa números, un dígito es representado por 4 bits.

$$2^n \geq 10; n = 4; 2^4 = 16$$

Así en cada matriz estarán representados 16 bits, 1 dígito por cada fila, un conjunto de leds representa un identificador de registro en el servidor.

Los leds están rodeados por un material color negro para incrementar el contraste del objeto con la cámara en el momento de realizar la detección de borde de la aplicación CAPTURADOR.

La elección de los diodos led rojos, se da por el patrón de radiación de los diodos led de 940 nm (infrarrojos) que es de 20 grados.

Debido a que nuestro sistema incluye el uso la cámara del teléfono desde distintos ángulos, los diodos infrarrojos de 940 nm fueron cambiados por diodos rojos de 625nm, que tienen un patrón de radiación mayor de 45 grados.

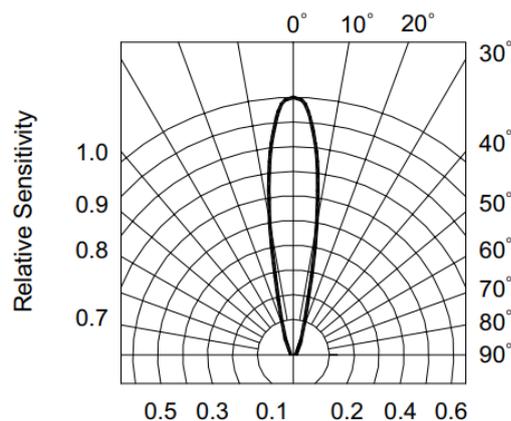
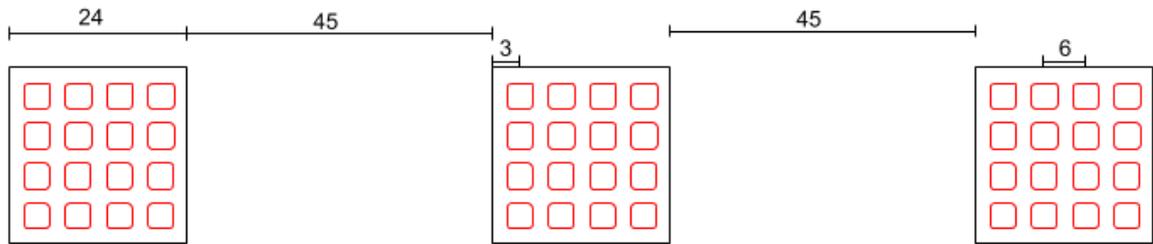


Figura 2.4: Patrón de radiación ofrecido por los diodos led de 940nm

La distancia entre cada punto dentro de matriz led es de 6,5 cm, esta separación se estimó para que a 7 metros de distancia, la intensidad de luz percibida desde la cámara no haga que los puntos se superpongan entre sí. con dicha separación entre punto se alcanza un tamaño de matriz de 24 cm con una separación entre cada matriz de 45 centímetros.



*Todas las medidas están en centímetros
 *Recuadros rojos son de 1x1 centímetros

Figura 2.5: Diseño final de la matriz led con sus dimensiones

Los datos requeridos para que la sesión sea única son:

- código de la materia - 1 número de 4 dígitos
- identificador del profesor - 1 número de 4 dígitos
- paralelo - 1 número de 2 dígitos
- identificador del aula - 1 número de 2 dígitos

Las matrices solo representarán números que son los identificadores de dichos registros en la base de datos del servidor remoto.

Cada dígito se representará en binario con 4 bits. Esto permite que se presenten hasta 4 números por matriz, con la cantidad de datos a mostrar solo se necesita un solo encendido de la matriz led.

La representación de la información es:

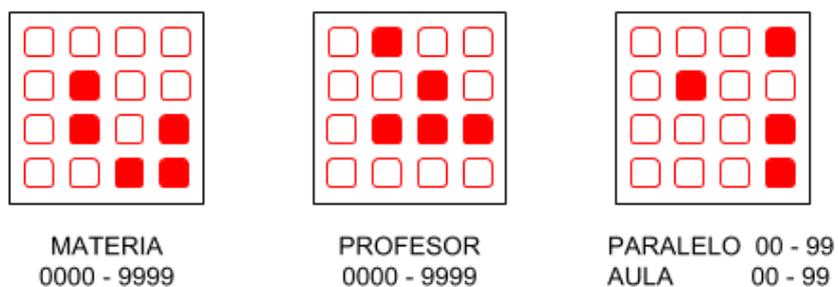


Figura 2.6: Representación de datos en la matriz led (primer encendido)

Para incrementar la disponibilidad del sistema a pesar de que se den fallos en los leds, los datos mostrados se repiten así:

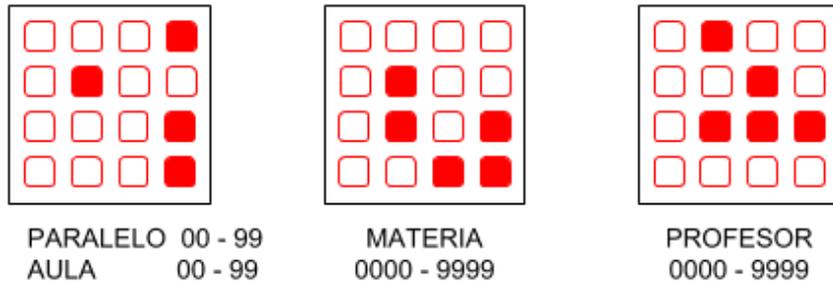


Figura 2.7: Representación de datos en la matriz led (segundo encendido)

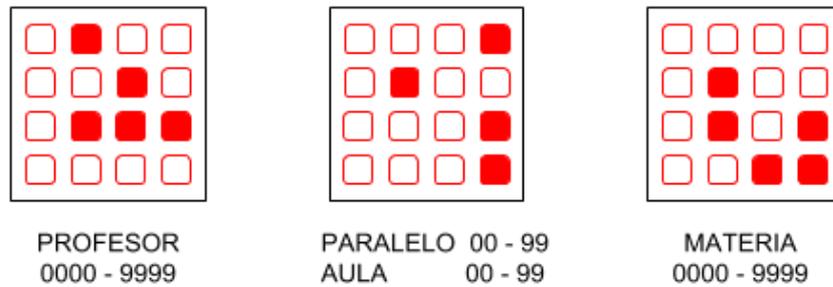


Figura 2.8: Representación de datos en la matriz led (tercer encendido)

Estas secuencias se dan para evitar errores en el proceso de decodificación, durante el cual se comparan los patrones mostrados en cada parpadeo.

CAPITULO 3

3. IMPLEMENTACIÓN

3.1. Matrices Led

3.1.1. Elementos de la matriz

- Microcontrolador: Se requiere un microcontrolador, que tenga varios puertos de entrada y salida para comunicación con otros elementos ya que se requieren 48 puertos de entrada para la matriz led y 2 puertos 2 entrada para la conexión de un módulo bluetooth, se realizó un análisis entre raspberry pi 3 y Arduino uno. Estas son las placas para desarrollo de proyectos de electrónica más conocidas, nuestros requisitos de desarrollo ágil, bajo costo, cantidad de puertos de entradas disponibles acorde las necesidades de conexión, la elección es por el Arduino uno[3].
- Shift Register SN74HC595N [4]: Las placas Arduino tienen un número máximo de 16 salidas digitales y la matriz propuesta requiere 48 salidas digitales. La placa Arduino no es capaz de solventar nuestra necesidad, por lo que se requiere este integrado que su función principal es recibir una señal de datos en serie y la salida es una señal datos en paralelo. El integrado 74HC595 permite con una misma entrada obtener 8 salidas, para su funcionamiento necesita una señal con los datos, una señal de clock que sincroniza la señal enviada a todos los antes de ser mostrada y una señal de latch que activa la señal de datos para ser mostrada o no.

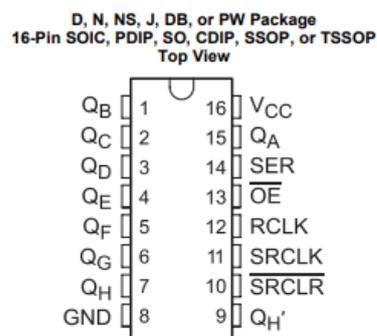


Figura 3.1: Diagrama del SN74HC595N

Dentro de cada bloque de la matriz existen 2 módulos 74HC595 para tener el total de 16 salidas requeridas, en el sistema existen 6 en total.

Estos módulos comparten las señales de clock y latch mientras que la señal de datos es enviada al primer módulo 74HC595 y de allí se conectan las demás entre el pin Qh' a las entradas del siguiente módulo obteniendo una conexión en cascada.

- Diodos Emisores de Luz [5]: Las matrices contendrán 64 led, se prevé el uso de 4 led por punto en la matriz para evitar que se emitan datos erróneos en la matriz las características del led son: 5mm de diámetro, 200 mili candela de intensidad de iluminación, rango de visión del haz de luz, 45 grados.



Figura 3.2: Diodo led rojo 625nm.

- Comunicaciones: El módulo bluetooth realizará la conexión entre el Arduino y la aplicación móvil del profesor, mediante esta la aplicación del profesor enviará el código único de sesión de clases. El módulo bluetooth tendrá configurado el nombre del aula de clase por defecto, las características del módulo bluetooth son, 4 pines de conexión (tierra y voltaje) y 2 de transmisión de datos (emisión y recepción), funciona en el rango de 3.3v y 5 v tiene un alcance de 5 a 10 metros.

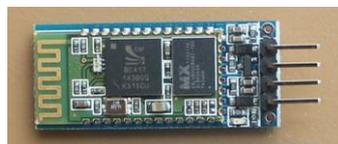


Figura 3.3: Módulo Bluetooth HC-06[6]

3.1.2. Programación del Microcontrolador

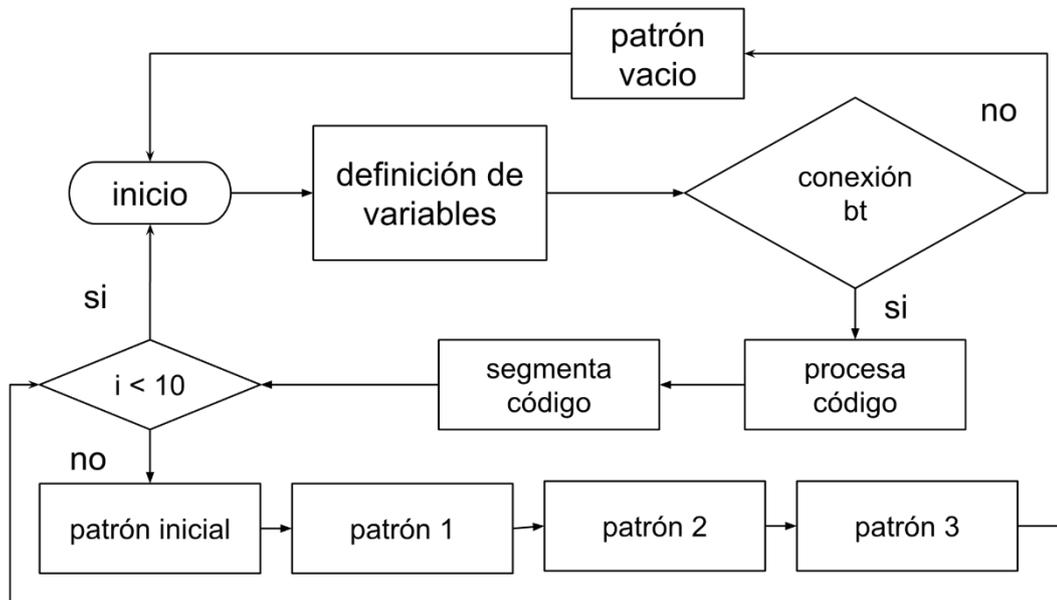


Figura 3.4: Flujograma de proceso presentación de datos en la matriz led.

La conexión bluetooth estará disponible para que la aplicación emisor se conecte, cuando esta conexión sea establecida, la aplicación enviará un código a la matriz led la conexión bluetooth termina y se procesa el código enviado, y si es válido se este código se segmenta para ser enviado a cada matriz y luego muestra en los leds, primero con un patrón inicial, para que la aplicación captador, sepa donde iniciar la decodificación, luego los tres parpadeos antes mencionados.

Esta secuencia desde que la aplicación emisor envía el código a la matriz led, la matriz muestra una única secuencia de código en 10 segundos, lo que haría difícil al estudiante el decodificar el código, por esto la secuencia de parpadeos se repite 10 veces, para que el estudiante tenga varias oportunidades de decodificar.

3.1.3. Diagrama de esquemático de la matriz led

Se muestra la conexión realizada entre componentes, para fines prácticos solo se muestra la conexión a los 8 primeros leds y el shift register, además se muestra la conexión de los pines de clock, latch y

data que son conexiones compartidas entre los 6 shift register que estarán en las 3 matrices.

En el diagrama, las líneas rojas son las conexiones de alimentación que será proporcionado por el Arduino, las líneas color morado son conexiones a tierra y las líneas verdes son las conexiones de datos, se ha considerado que el pin 2 y 4 son las conexiones al controlador bluetooth, los pines 8,11 y 12 son conexiones al shift register: latch, data y clock respectivamente.

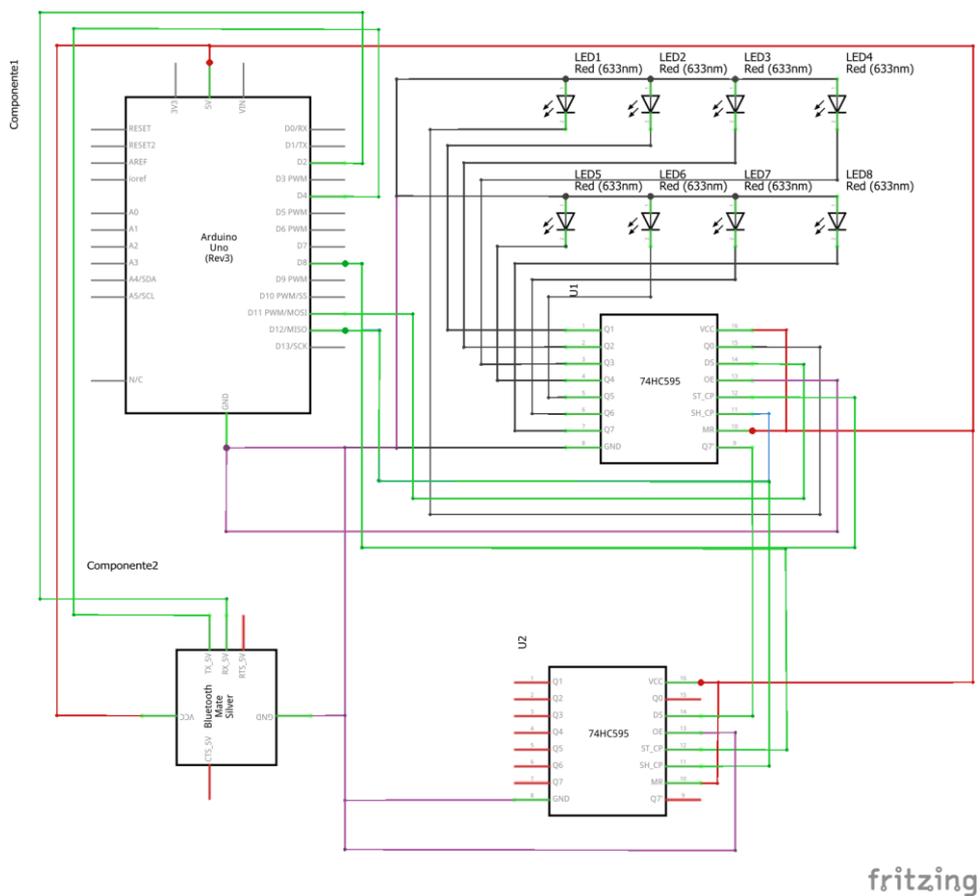


Figura 3.5: Diagrama esquemático del circuito para un bloque de leds

Cada bloque de leds tiene 2 módulos 74H595, solo a el primer bloque de la matriz led llega el dataPin, del primer módulo 74HC595 el pin Q7' se conecta al pin dataIn del siguiente módulo 74HC595 del pin Q7' del segundo módulo sale la señal hacia el segundo bloque de leds, se da la misma situación dentro del bloque dos y el paso al bloque tres.

Para el módulo bluetooth HC-06 se necesitan 4 conexiones una a tierra, otra al 5v, además de 1 pin para transmisión de datos (TX) y otro para recepción de datos (RX).

3.2. Software de inicio de asistencia (emisor)

La aplicación móvil de uso del profesor, denominada emisor, deberá iniciar sesión para validar la identidad del usuario, el inicio de sesión es requerido porque debe obtener datos de las materias, paralelos, aulas y un listado de las direcciones MAC de los módulos bluetooth HC-06 pertenecientes a las aulas donde el profesor dará clases durante el periodo académico, toda esta información es obtenida desde el servidor remoto y debe estar previamente establecida.



Figura 3.6: Pantalla de Inicio de Sesión de la aplicación EMISOR

La aplicación hará uso de una base de datos local para el almacenamiento de la información de las materias, aulas, y paralelos con sus respectivos y horarios, la necesidad de dicha información es para minimizar la cantidad de tareas que el profesor debe de hacer para tomar asistencia, y que por defecto solo necesite seleccionar el aula donde dará clases para generar la asistencia a obtener por el CAPTURADOR.

La aplicación estará desarrollada para el SDK 17 o JellyBean MR1, por el uso de las API BluetoothDevice y BluetoothAdapter.

Para que la aplicación tenga un mayor número de usuarios potenciales, según cifras de Android, desarrollar aplicaciones para SDK 16 o android 4.2

Jelly Bean asegura una compatibilidad con el 80% de sus dispositivos, pero debido a funciones usadas en la aplicación como obtener la dirección MAC del dispositivo, se requiere el uso del SDK 17.

3.2.1. Recepción de información desde el servidor remoto

Una vez que el usuario (profesor) ha iniciado sesión en el sistema; mediante el uso de la librería Volley la aplicación hace solicitudes de información al servidor remoto mediante el servicio web.

Solicita la información del profesor, materias, edificios, aulas, paralelos donde el identificador del profesor y materia coincida.

```
[
  {
    "id_materia":1,
    "id_profesor":1,
    "id_aula":20,
    "anio":"2017",
    "termino":"2",
    "numero_paralelo":"10",
    "dia1":"LUN",
    "dia2":"MIE",
    "dia3":"JUE",
    "hora1":"11:30",
    "hora2":"11:30",
    "hora3":"12:30",
  }
]
```

Figura 3.7: Listado de paralelo donde el profesor tiene identificador 1

El primer listado a consultar es el paralelo, se envía el identificador del usuario (profesor) y se obtienen los identificadores de los paralelos y de las aulas, del mismo modo se consulta por estos valores.

Los datos obtenidos se almacenan en una base de datos local con el mínimo de tablas manteniendo semejanza con la estructura de la base de datos del servidor remoto.

Esta base de datos local se crea en el dispositivo cuando se inicia sesión.

```
private static final String paralelo="CREATE TABLE core_paralelo (id integer NOT NULL " +
"PRIMARY KEY AUTOINCREMENT, identificador varchar(10) NOT NULL UNIQUE, anio varchar(4) " +
" NOT NULL, termino varchar(1) NOT NULL, numero_paralelo varchar(2) NOT NULL, " +
"dial varchar(3) NOT NULL, dia2 varchar(3) NULL, id_materia_id integer NOT NULL, " +
"id_profesor_id integer NOT NULL, hora1 time NOT NULL, hora2 time NOT NULL, hora3 time " +
"NOT NULL, dia3 varchar(3) NULL)";
```

Figura 3.8: Creación de tabla paralelo en la base de datos local

Las tablas creadas en la base de datos local son: profesor, materia, paralelo, edificio, aula, arduino.

La tabla edificio es usada en caso de que la sesión de clase no se realice en el aula habitual, el sistema pueda sugerir conectarse a aulas cercanas al aula original.

La tabla arduino es necesaria porque contiene las direcciones MAC de los dispositivos *bluetooth*, esta dirección, se usa para establecer la conexión entre la aplicación y la matriz led para enviar el código.

La tabla paralelo contiene información sobre los días de las sesiones de clase y la hora de inicio de las mismas, esta información se usa para que la aplicación filtre los dispositivos *bluetooth* sincronizados con el dispositivo y muestre como primera opción el dispositivo al cual debe conectarse según la hora del dispositivo.

3.2.2. Envío de códigos a matriz led

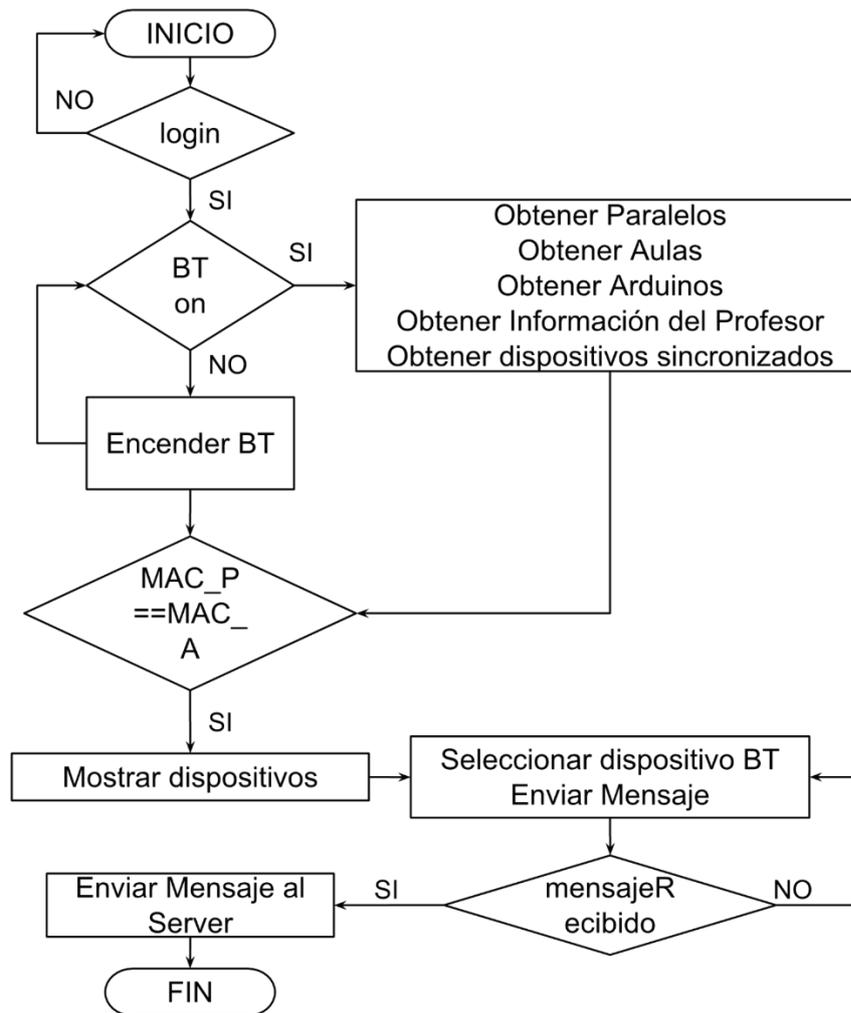


Figura 3.9: Diagrama de Flujo principal de la aplicación EMISOR

Luego de iniciar sesión, esta se mantendrá iniciada hasta que el usuario decida cerrar la sesión manualmente, para evitar la petición de datos al servidor y que el resto de funcionalidades, puedan ser usadas sin estar conectados a una red de datos.

La aplicación por defecto necesita usar el Bluetooth para conectarse a la matriz led, en caso de no estar activado, hace la petición al usuario, en caso de ser negativa, la aplicación se cierra.

Luego de activar el *bluetooth*, la aplicación consulta en la base de datos local según la hora del dispositivo, el paralelo y su respectiva materia, el

aula donde se da clases a dicho paralelo y la MAC del módulo bluetooth HC-06 asociado a dicha aula, en caso de encontrar el paralelo, muestra una ventana, donde aparece el aula y el edificio al que se conectará para tomar la asistencia.

Además de mostrar el aula a conectar, la aplicación crea el código a transmitir a la matriz led:

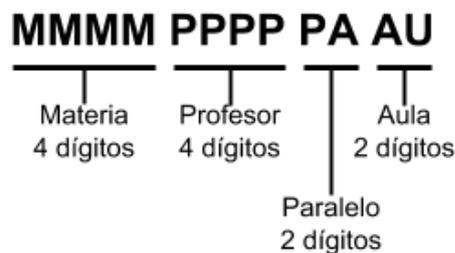


Figura 3.10: Mensaje a enviar a la matriz led.

Los datos necesarios para crear el código se obtienen de la base de datos local, la creación del código no es visible al usuario.

El usuario selecciona el aula y con la MAC del módulo *bluetooth* asociado al aula, se establece la conexión entre el EMISOR y la matriz led.

La aplicación envía el código, y si la matriz led devuelve un mensaje exitoso, se prepara el mensaje a enviar al servidor remoto.

El código a enviar al servidor remoto consta de:

- El código decodificado de la matriz led.
- Fecha actual.
- Potencias de las señales WIFI capturadas en el aula de clases.

En caso de no existir una conexión a una red de datos, el código queda en espera hasta que detecte una red de datos activa.

Una vez transmitido el código a la matriz led, la conexión establecida entre ellas se cierra, y la matriz led no estará disponible mientras se muestra el código.

3.2.3. Visualización de estudiantes

Todas las asistencias son visibles para el profesor y personal responsable en el servidor remoto según las políticas de cada institución y son modificables desde la consola de administración.

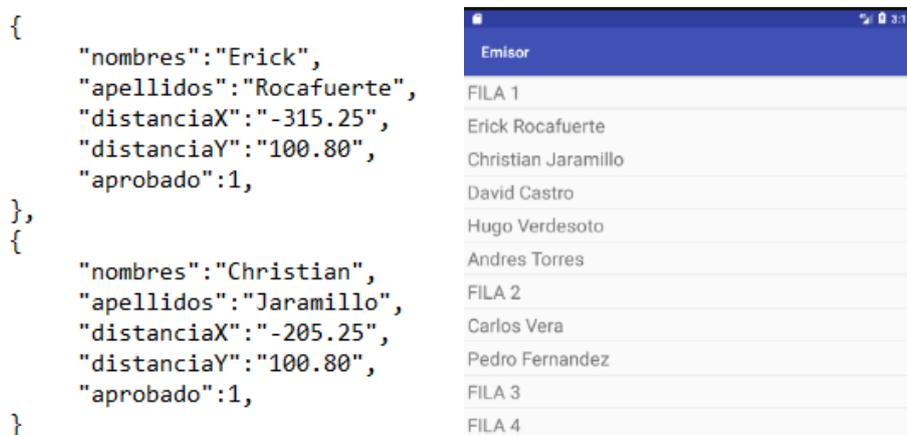


Figura 3.11: Lista de estudiantes JSON y organizado por filas en la aplicación Emisor.

En caso de que el profesor esté conectado a una red de datos podrá visualizar los estudiantes que hayan registrado su asistencia agrupados por fila.

Las asistencias de los estudiantes son tomadas desde el servidor remoto una vez que son validadas, la aplicación emisor toma estas asistencias en formato JSON y son cargadas en un ArrayAdapter y mostradas en un ListView en orden.

La descarga de datos se organiza por posición del estudiante (distanciaX, distanciaY) estos valores se encuentran en la tabla asistencia estudiante del servidor remoto.

3.3. Software decodificador de asistencia (capturador)

3.3.1. Información del estudiante

Previo a la decodificación el estudiante debe ingresar su información personal como nombres, apellidos y matrícula necesarios para ser visualizados en el momento que llegan las asistencias verificadas al teléfono del profesor.

Desde la pantalla principal dar clic en el ícono para ir a la pantalla e ingresar la información requerida.

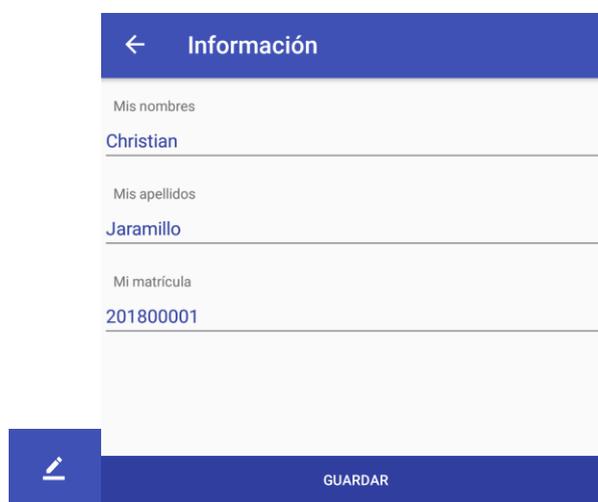


Figura 3.12: Ícono y pantalla donde ingresa su información personal

3.3.2. Recepción de la imagen

Una vez que el profesor da comienzo a la toma de asistencia el estudiante debe dar clic en el ícono de la cámara para abrir el decodificador, se debe apuntar hacia las matrices para dar inicio al proceso.

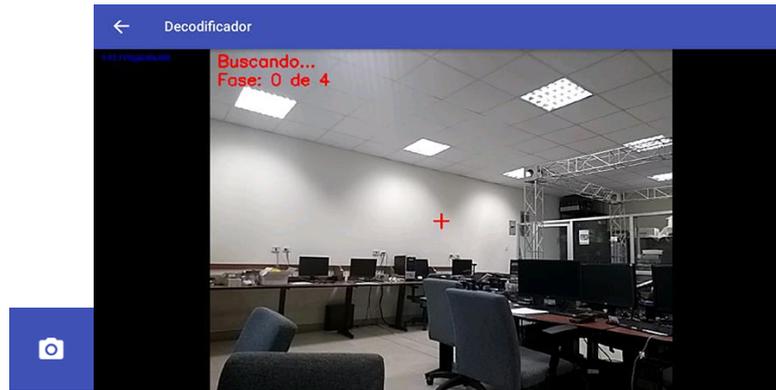


Figura. 3.13: Ícono y pantalla del decodificador

3.3.3. Procesamiento de la imagen

Antes de lograr la decodificación del código generado en las matrices led es necesario realizar procesamiento de las imágenes capturadas por la cámara en tiempo real.

Para lograr este objetivo se hace uso de la librería de visión artificial OpenCV (Open Source Computer Vision Library) ya que está optimizado para ser eficiente en cuanto a recursos computacionales con enfoque a aplicaciones sofisticadas en tiempo real [7], compatible con Android y de distribución gratuita.

El primer paso es encontrar la ubicación las matrices dentro de la imagen para ello se realiza varios procesos previos de acuerdo al siguiente diagrama de flujo:

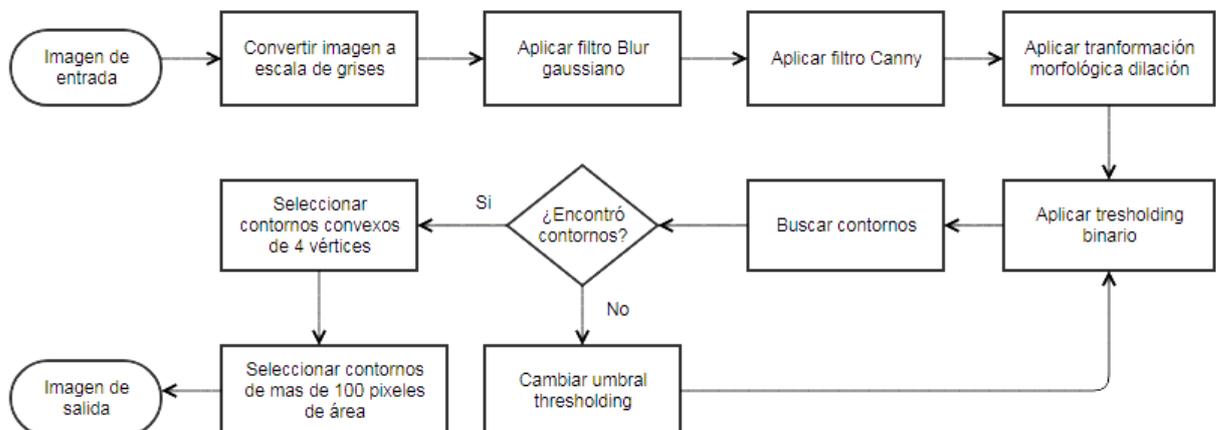


Figura 3.14: Diagrama de flujo del procesamiento de la imagen

La imagen de entrada es convertida a escala de grises para ser procesada, luego se aplica un filtro de suavizado con un kernel 3x3 gaussiano para reducir el ruido no deseado, a continuación, se aplica el filtro canny para hallar los bordes de las matrices y luego aplicar una vez la transformación morfológica dilación para unir aquellas líneas que contengan segmentos vacíos.

Finalmente se aplica segmentación binaria (thresholding) para hallar las zonas de interés, si no se llegara a encontrar alguna se incrementa el umbral de segmentación y se repite el proceso.

Con las formas encontradas se procede a discriminarlas, buscando primero los contornos convexos que posean 4 vértices y con un área de al menos 150 pixeles, dichas formas serán consideradas como los bordes de las matrices.

3.3.4. Decodificación

Una restricción importante al momento decodificar es que el estudiante apunte su cámara tal que la cruz ubicada en el centro de la cámara apunte a la matriz central para iniciar el proceso, sobre todo para mantener estabilidad y que la decodificación se realice lo más preciso posible.

3.3.5. Particionamiento

Un paso previo a la decodificación de la información es la partición lógica de cada matriz, como cada matriz posee 16 led uniformemente separados, es necesario realizar 16 particiones de igual área a cada matriz, cada partición represente la zona de interés de cada led necesario para conocer si está encendido o apagado.

Se itera sobre cada partición para estimar la cantidad de pixeles blancos (luz de los leds) con respecto al total de pixeles de la partición para obtener el porcentaje de led encendidos, para evitar interferencia de los pixeles blanco por producto del ruido o de situaciones no deseadas se toma el valor de límite 6% para determinar que la partición contiene un led encendido (valor 1) caso contrario se considera apagado (valor 0).

Cada fila de las matrices representa un número de 0 a 9 en binario, en total se obtiene un número identificador de 12 dígitos, para el siguiente ejemplo: '12345678912'.

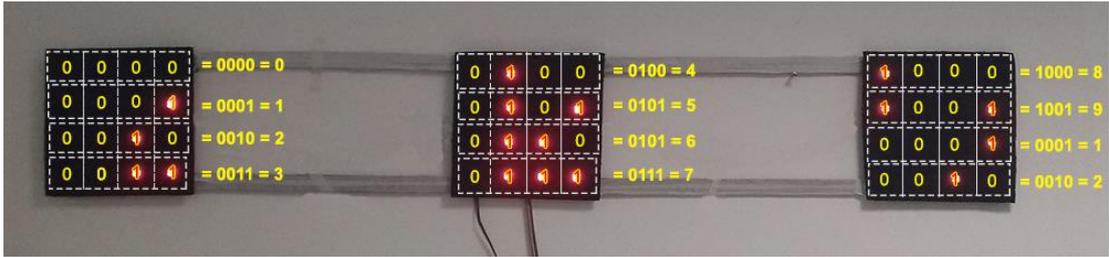


Figura 3.15: Particionamiento de las matrices

3.3.6. Fases de decodificación

El proceso de decodificación está compuesto de 4 fases, la primera fase para dar inicio al proceso y las restantes para la obtención del identificador, redundando en el envío de la información para aumentar las probabilidades de éxito en la decodificación.

Fase 1: Comprende la detección del patrón de inicio (todos los leds encendidos) para dar comienzo a la decodificación.

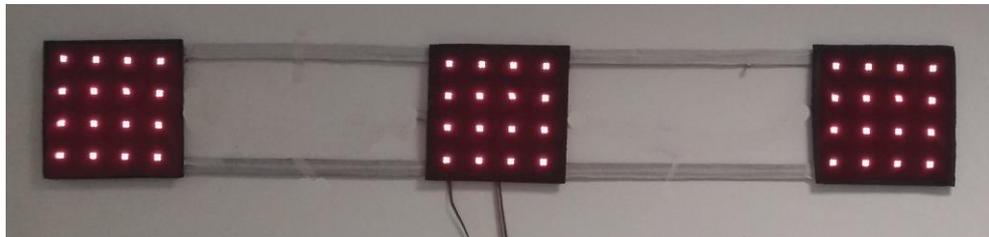


Figura 3.16: Fase 1 de la decodificación.

Fase 2.- Comprende la obtención del identificador, con la información replicada en las matrices en orden de izquierda a derecha.



Figura 3.17: Fase 2 de la decodificación.

Con el código identificador '012345678912' los dígitos '0123' se generan en la matriz 1, los dígitos '4567' en la matriz 2 y los dígitos '8912' en la matriz 3 para su decodificación.

Fase 3.- Comprende la obtención del identificador, con la información replicada en las matrices en el siguiente orden:

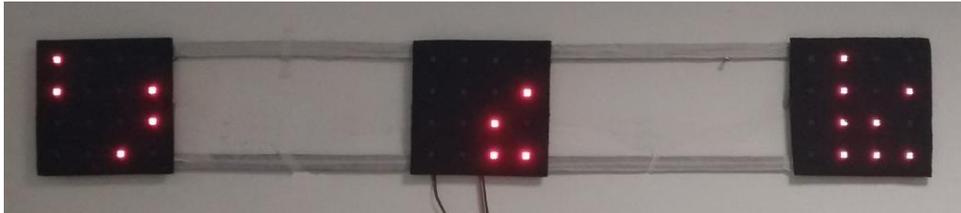


Figura 3.18: Fase 3 de la decodificación.

Los dígitos '4567' se imprimen en la matriz 3, los dígitos '8912' en la matriz 1 y los dígitos '0123' en la matriz 2 para su decodificación.

Fase 4.- Comprende la obtención del identificador, con la información replicada en las matrices en el siguiente orden:



Figura 3.19: Fase 4 de la decodificación

Los dígitos '8912' se imprimen en la matriz 2, los dígitos '0123' en la matriz 3 y los dígitos '4567' en la matriz 1 para su decodificación.

Y como resultado se realiza una verificación de todos los identificadores decodificados obtenidos para devolver el más común como identificador final, este ejemplo es '12345678912'.

3.3.7. Traducción

Cada fila decodificada de 4 bits es transformada a decimal, luego la información traducida es concatenada para formar el código identificador de la asistencia de 12 dígitos.

3.3.8. Estimación de la posición

Luego que el identificador es decodificado el paso siguiente es la estimación de la posición del estudiante dentro del aula de clases.

Con ayuda del modelo de cámara '*thin lens*', triángulos semejantes y proporciones se estima la distancia desde la cámara hasta las matrices led ubicadas sobre la pizarra.

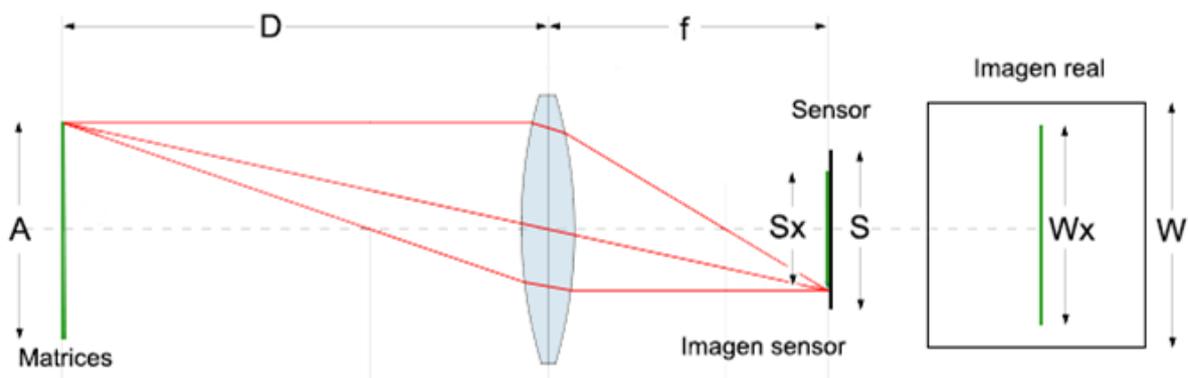


Figura 3.20: Vista superior del modelo '*thin lens*' usado en la solución.

D = Distancia real a estimar (mm)

A = Ancho real del objeto (mm)

f = Distancia focal (mm)

S = Ancho del sensor de la cámara (mm)

S_x = Ancho del objeto en el sensor (mm)

W = Ancho de la imagen (pixel)

W_x = Ancho del objeto en la imagen (pixel)

Se asume que la imagen está enfocada por lo que la distancia focal es aproximadamente igual a la distancia del lente al plano de la imagen entonces del gráfico se deduce que:

$$\frac{D}{A} = \frac{f}{S_x} \quad (3.1)$$

$$\frac{S}{S_x} = \frac{W}{W_x} \quad (3.2)$$

$$D = \frac{f * A}{S_x} \quad (3.3)$$

$$S_x = \frac{S * W_x}{W} \quad (3.4)$$

Reemplazando ecuación (3.4) en (3.3) se obtiene:

$$D = \frac{f * A * W}{S * W_x} \quad (3.5)$$

Donde 'D' es la distancia desde la cámara hasta la matriz led central, 'f' la distancia focal de la cámara, 'S' el ancho del sensor de la cámara, 'W' el ancho de la imagen y 'A' el ancho real del objeto son parámetros conocidos por lo que, el parámetro faltante 'Wx' en la ecuación es el ancho del objeto en la imagen, dicho objeto corresponde al rectángulo que contiene a las tres matrices led y que es calculado en el subcapítulo 3.3.7, y por último la posición del estudiante dentro del aula viene dado por:

$$X = -D * \cos(\alpha) * \sin(\beta) \quad (3.6)$$

$$Y = D * \cos(\alpha) * \cos(\beta) \quad (3.7)$$

Alfa y beta corresponde a los ángulos de rotación '*tilt*' y '*yaw*' de la cámara respectivamente y estimados en el subcapítulo 3.3.6, si 'X' es

negativo indica que el estudiante está ubicado a la izquierda del eje Z con respecto a su perspectiva, caso contrario está ubicado a la derecha.

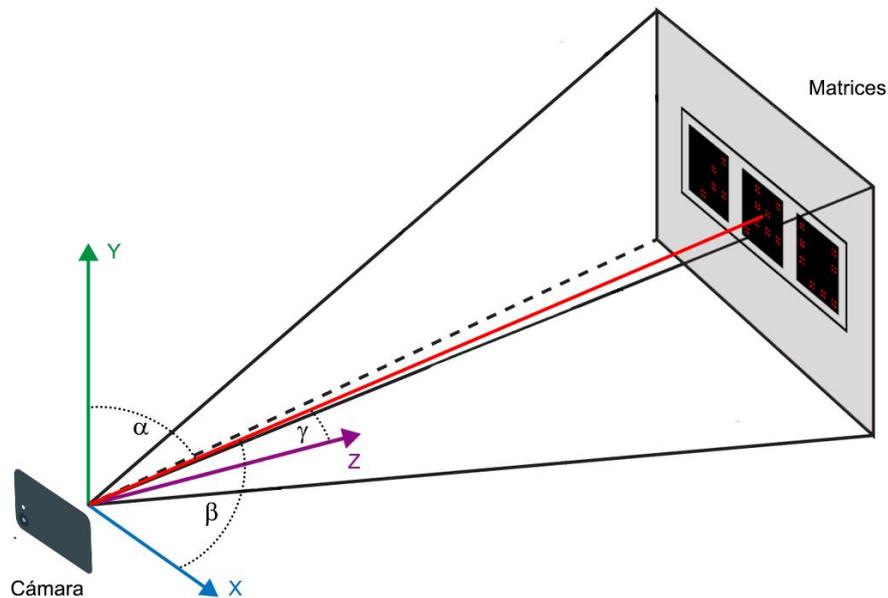


Figura 3.21: Ángulos de rotación de la cámara.

3.3.9. Estimación de ángulos de rotación de la cámara

Cómo la cámara del estudiante no se encuentra alineado ni a la misma altura que las matrices led es necesario estimar los ángulos de rotación de la cámara con respecto a las matrices para luego realizar una corrección de perspectiva de la imagen y finalmente calcular el ancho de las matrices 'Wx' en pixeles.

Para estimar los ángulos se utiliza el método '*Vanishing points*' [9], dado cada imagen obtenida de la cámara y tomando el borde que rodea a las matrices se procede a proyectar líneas desde los bordes horizontales y verticales hasta que se intersecten en el 'infinito', cada punto de intersección es conocido como *vanishing point*.

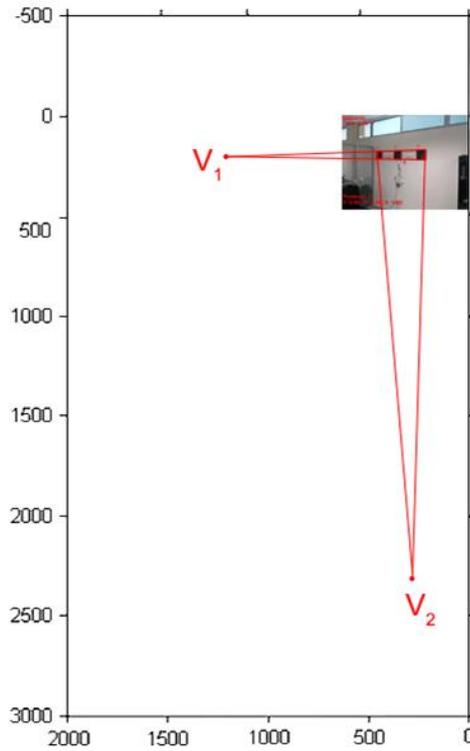


Figura 3.22: Vanishing points estimados en imagen 640 x 480 pixeles.

Para obtener los valores estimados de los ángulos se realiza el siguiente procedimiento:

Hallar las columnas de la matriz de rotación 'R' donde 'R1' y 'R2' son los vectores normalizados de la multiplicación de la inversa de la matriz de los parámetros de la cámara 'K' con cada uno de los vectores de las componentes de los puntos de intersección 'V1' y 'V2' encontrados:

$$R = [R_1 \ R_2 \ R_3] \quad (3.8)$$

$$R_1 = \frac{K^{-1} * V_1}{|K^{-1} * V_1|} \quad (3.9)$$

$$R_2 = \frac{K^{-1} * V_2}{|K^{-1} * V_2|} \quad (3.10)$$

$$R_3 = R_1 \times R_2 \quad (3.11)$$

$$K = \begin{bmatrix} f & 0 & cx \\ 0 & f & cy \\ 0 & 0 & 1 \end{bmatrix} \quad K^{-1} = \begin{bmatrix} \frac{1}{f} & 0 & -\frac{cx}{f} \\ 0 & \frac{1}{f} & -\frac{cy}{f} \\ 0 & 0 & 1 \end{bmatrix}$$

donde 'f' es la distancia focal de la cámara, 'cx' es el punto medio del ancho de vista y 'cy' el punto medio del alto de vista.

Una vez calculado la matriz de rotación R, la estimación del ángulo 'tilt' con respecto al eje y es:

$$\alpha = -\sin(R_3(y)) \quad (3.12)$$

la estimación del ángulo con respecto al eje x 'yaw' es:

$$\beta = -\tan^{-1}\left(\frac{R_3(x)}{R_3(z)}\right) \quad (3.13)$$

para estimar el ángulo con respecto al eje z 'roll' solo es necesario tomar dos de los puntos extremos de las matrices led para hallar el ángulo con respecto al eje horizontal de la imagen, donde 'x' corresponde a la diferencia entre las componentes x_1 y x_2 y 'y' a la diferencia de las componentes y_1 y y_2 .

$$\gamma = \tan^{-1}\left(\frac{y}{x}\right) \quad (3.14)$$

3.3.10. Corrección de perspectiva

Una vez estimados los ángulos de rotación de la cámara se procede a corregir la perspectiva de imagen para ello debemos construir la matriz de corrección 'C' dado los ángulos de la cámara y utilizando la matriz de parámetros intrínsecos (K) de la cámara y su inversa (K^{-1}), además de las matrices de rotación respecto a cada ángulo (R) y a la matriz de traslación (T), obteniendo 'C' la matriz de corrección [8].

$$K = \begin{bmatrix} f & 0 & cx & 0 \\ 0 & f & cy & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad K^{-1} = \begin{bmatrix} \frac{1}{f} & 0 & -\frac{cx}{f} \\ 0 & \frac{1}{f} & -\frac{cy}{f} \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_\beta = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} R_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_\gamma = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C = K * T * (R_\gamma * R_\beta * R_\alpha) * K^{-1} \quad (3.15)$$

Para mantener una imagen corregida centrada se requiere una matriz de traslación 4x4 donde el desplazamiento en 'x' y 'y' sea 0 y 'z' sea 1.

Cómo no hay referencia de las matrices en la nueva imagen corregida hay que volver a realizar el proceso de búsqueda de las matrices de acuerdo al flujo de la figura 3.11, una vez detectados se procede a hallar la distancia 'Wx' (ecuación 3.5) en pixeles entre el punto inferior izquierdo

e inferior derecho del borde que encierra a las matrices para hallar la distancia 'D' (ecuación 3.5) de la cámara a la matriz central, con ayuda de los ángulos de rotación, se encuentra 'X' y 'Y' de las ecuaciones 3.6 y 3.7 respectivamente, que corresponde a las coordenadas de la posición del estudiante dentro del aula de clases.

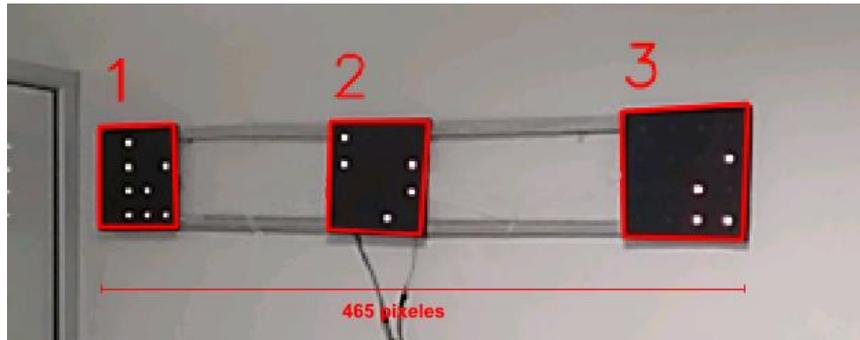


Figura 3.23: Matrices sin corrección 45° con respecto al eje Z

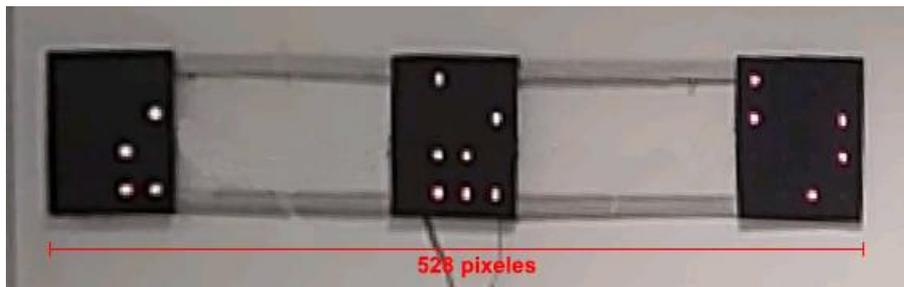


Figura 3.24: Matrices con vista corregida

3.3.11. Recopilación de información de la asistencia

Entre la información recolectada por la aplicación del estudiante para ser enviado al servidor web se encuentra:

- Información personal (nombres, apellidos y matrícula) del estudiante previamente ingresados.
- El código decodificado de las matrices led.
- MAC identificador del teléfono del estudiante.
- Fecha actual.
- Potencias de las señales WIFI capturadas en el aula de clases.

El objetivo de capturar las potencias de las señales WIFI es verificar que el estudiante se encuentre en el mismo lugar donde se toma la asistencia, esto se logra comparando las señales WIFI que son capturadas por la aplicación del profesor, si algún estudiante posea intensidades de potencias muy distintas de las que posee el profesor se concluye que el estudiante no se encuentra en el mismo lugar.

Toda esta información que es considerada como la 'asistencia del estudiante', para ser enviada al servidor web en donde será verificada con la información enviada por el profesor para validar la asistencia, si el estudiante no posee conexión a internet en el momento la información es almacenada en archivo hasta que se conecte y decide enviarla.

3.4. Servidor web remoto

El servidor remoto debe permitir una construcción de un sistema robusto, que tenga un sistema de autenticación incorporado, que permite una migración del sistema de manera fácil.

Por las características planteadas, se escogió el desarrollo del servidor web escrito Python desarrollado con el framework Django que emplea la arquitectura Modelo-Vista Controlador (MVC) y nos permite una construcción automática del modelo en base al controlador.

Además, Django tiene un sistema de autenticación incorporado, que será usado en el presente proyecto 'AUTH-USER'.

También prevé un sistema de administrador, que permite modificar los contenidos guardados en la base de datos.

Para recibir la información enviada por emisor y receptor el servidor remoto tendrá dos servicios de rest framework que es un framework para desarrollo de Servicios Web basado en Python y compatible con Django que nos permite enviar y recibir respuestas en formato JSON.

El servidor remoto permitirá 2 acciones principales:

- Almacenar la información de materias, profesores, paralelos, aulas, edificios y arduinos. Necesarios para la generación de códigos de asistencias.
- Almacenar y validar asistencias, tanto de profesores y alumnos.

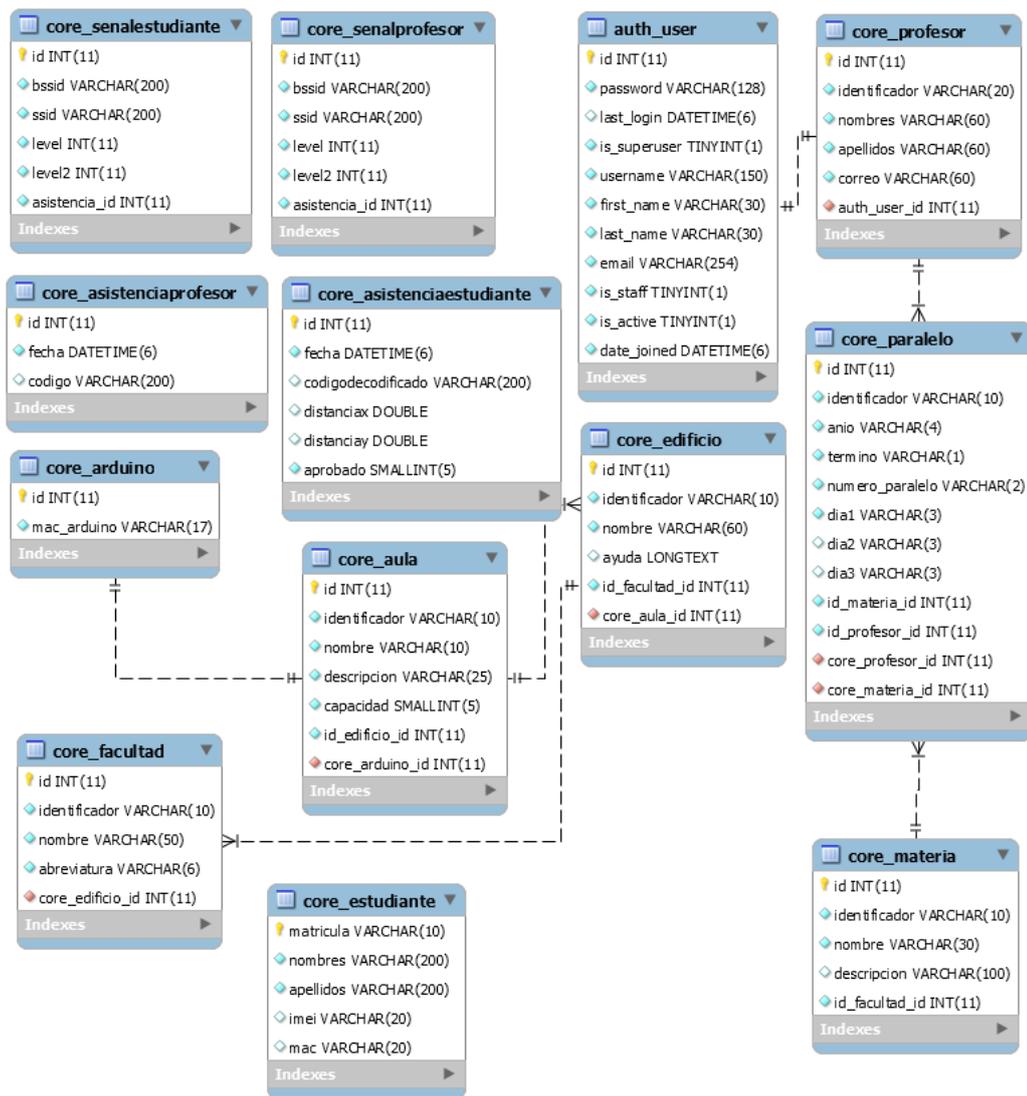


Figura 3.25. Diagrama de entidad relación de la base de datos.

El diagrama entidad relación sólo muestra las clases que interactúan dentro de nuestro sistema, existen más tablas propias de django especialmente las que usa el administrador de django como: auth user, user permissions, permissions, etc.

Las tablas más relevantes en cuanto a la validación de la asistencia son: “asistencia estudiante”, “asistencia profesor”, “senal estudiante” y “senal profesor”.

La aplicación capturadora del estudiante mantiene una lista de las asistencias que haya realizado, la información es enviada en formato json de la siguiente forma:

```
{
  "id": "20101111",
  "nombres": "Erick",
  "apellidos": "Rocafuerte",
  "distanciaX": "-315.25",
  "distanciaY": "100.80",
  "aprobado": 0;
  "senalestudiante":
  {
    "bssid": "045D4BAB96B1";
    "ssid": "8C579BC52CAC";
    "level": "95";
    "level2": "85";
  },
}
```

Figura 3.26: Información enviada en Formato JSON.

El archivo JSON del estudiante que es recibido por el servidor se almacena en dos tablas, y luego se busca el mismo registro para el profesor, en caso de encontrarlo se compara y se acepta o rechaza la asistencia del estudiante.

Al momento que el estudiante ha enviado sus asistencias al servidor puede suceder dos escenarios:

1. Que el profesor haya registrado el código de asistencia y esté a la espera de las asistencias de los estudiantes para verificarlos.
2. Que aún no se haya registrado el código del profesor debido a que el profesor no cuenta con conexión a internet en el momento.

Para el primer caso, con el código registrado solo se procede a verificar las asistencias de los alumnos uno a uno.

Para el segundo caso, no se realiza ninguna verificación y se mantiene la asistencia guardada en la aplicación capturadora hasta que el estudiante la vuelva a enviar.

Se considera una asistencia aprobada cuando ha pasado exitosamente por las siguientes comprobaciones:

- Que el estudiante no haya cambiado por más de una ocasión su teléfono mediante la verificación de la MAC, si se detecta que la MAC del estudiante ha cambiado más de una vez, se negará la validación de su asistencia.
- Que los códigos identificadores coincidan.
- Que las señales WIFI entre el profesor y el estudiante concuerde.

CAPÍTULO 4

4. RESULTADOS OBTENIDOS

4.1. Resultados de la construcción de las matrices led

Los diodos led de 940nm (infrarrojos) no cubrieron las necesidades de intensidad de luz, ya que esta se disipaba rápidamente conforme se incrementa la distancia, además el ángulo desde el cual se capturaba la fotografía influyó con la cantidad de luz recibida del diodo.

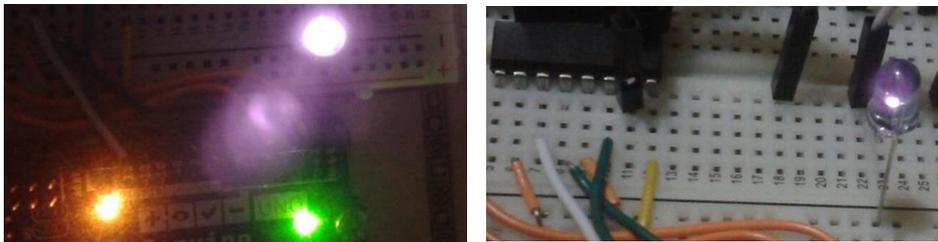


Figura 4.1. Luz emitida por un diodo led de 940nm vista de frente (d), vista sesgada 45°

La matriz led final tuvo un tamaño total de 165 cm, tamaño de cada bloque de leds 25 cm x 25 cm (3 en total), 45 cm de separación entre cada bloque de leds (90 en total), estas dimensiones permiten que la aplicación capturador, decodifique lo mostrado en la matriz led a una distancia de 7 metros y estime la posición del estudiante de manera consistente.



Figura 4.2. Matriz led con atenuación en el primer y tercer bloque de leds

El prototipo de la matriz presentó problemas de atenuación por las conexiones en paralelo realizadas.

4.2. Resultados de decodificación

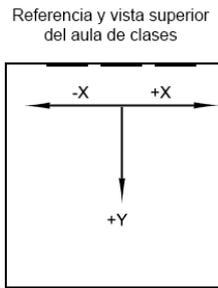


Figura 4.3: Vista superior al aula de clases

	-300	-200	-100	0	100	200	300	(cm)
130		25	15	0	15	25		
200		10	5	0	15	20		
300	25	15	5	0	0	10	25	
400	25	15	15	5	15	18	30	
500	35	18	20	18	25	20	30	
600	35	35	35	25	35	40	45	
700	65	50	40	40	45	60	70	
(cm)								

Figura 4.4: Resultados de decodificación (% error)

Se expresa los resultados de la decodificación en porcentaje de error, mediante 20 intentos por posición, cuando el estudiante se encuentra frente a las matrices led la decodificación es excelente, de acuerdo a los resultados cuando está ubicado en (0, 130) cm el error es de 0%, el error va aumentando gradualmente mientras aumenta 'Y', la decodificación se complica a medida que el ángulo 'yaw' (respecto al eje x) aumenta debido a la estimación no tan precisa mediante el método 'Vanishing points'.

No es posible realizar una decodificación más allá de los 7 metros con esta solución.

4.3. Resultados de estimación de posición

Para estos resultados los valores están expresados en media aritmética y desviación estándar.

Estimación de la componente 'Y' de la posición

	300	200	100	0	100	200	300	(cm)
130		152,32 13,78	140,08 7,5	124,67 3,38	142,20 7,57	147,06 8,73		
200		239,01 16,84	207,48 2,97	200,83 3,55	206,82 1,68	234,10 3,06		
300	379,65 14,71	313,82 4,08	287,86 6,08	306,51 2,22	286,38 6,40	306,48 4,29	383,32 7,92	
400	383,45 7,84	404,40 3,44	385,43 6,98	381,09 4,88	387,89 3,95	396,36 6,68	386,25 3,06	
500	482,11 5,14	486,10 2,33	481,42 3,74	479,42 7,42	480,44 4,29	486,45 9,13	479,81 5,91	
600	579,42 6,74	586,43 8,26	582,80 7,19	584,54 8,20	581,40 7,2	588,52 15,64	573,42 9,87	
700	654,81 4,02	655,46 5,82	663,22 22,33	673,21 13,38	669,76 6,96	655,40 5,82	661,49 8,82	
(cm)								

Figura 4.5: Resultados de la coordenada 'Y' de la posición

Estimación de la componente 'X' de la posición

	300	200	100	0	100	200	300	(cm)
130		184,03 16,34	119,60 6,6	6,44 1,37	88,99 0,87	183,02 7,14		
200		226,62 20,76	99,54 2,50	3,56 4,2	98,66 1,68	226,49 6,7		
300	274,20 9,15	210,20 4,02	120,42 5,57	0,41 3,31	98,91 4,08	182,59 3,97	287,97 3,93	
400	281,40 4,59	207,20 2,48	123,62 4,06	3,15 1,84	92,17 4,9	197,48 11,35	282,19 4,53	
500	285,40 7,00	208,20 3,18	85,32 5,82	10,63 1,68	116,95 10,12	199,50 9,17	284,02 6,23	
600	283,60 4,36	184,90 4,18	112,20 4,45	3,90 2,24	117,60 5,16	188,49 10,2	276,64 16,11	
700	272,86 5,19	171,26 3,82	79,44 5,89	8,18 3,79	85,20 8,38	177,23 8,37	282,17 4,26	
(cm)								

Figura 4.6: Resultados de la coordenada 'X' de la posición

De acuerdo con los resultados se concluye que mientras el estudiante está ubicado en el centro de las matrices la estimación de la posición es casi exacta con porcentaje de error mínimo, pero mientras se desplaza hacia los lados el error va aumentando igual que en la decodificación debido a la estimación no tan precisa del ángulo 'yaw'.

El tiempo del proceso de tomar la asistencia para el profesor toma 2 minutos desde que se conecta a la matriz led, para el estudiante registrar su asistencia toma 5 segundos que es el tiempo que dura la visualización del código en la matriz led.

CONCLUSIONES Y RECOMENDACIONES

Este proyecto contribuye al ahorro de tiempo y recursos al momento de la toma de asistencias, dentro de los aspectos importantes está el diseño de las matrices led que ayuda a transmitir información mediante técnicas de procesamiento de imágenes con ayuda del teléfono del estudiante, sino que a la vez sirve como puntos de referencia para estimar su posición dentro del aula de clases.

Para disminuir el impacto de la luz emitida por los diodos led, se recomienda el uso de diodos led de 940nm conocidos como infrarrojos, cuidando que el patrón de radiación del mismo sea lo mayor posible.

Para escalar el sistema a los dispositivos iOS, se debe de cambiar el módulo bluetooth HC-06 por un módulo *bluetooth* HC-08 o HC-10 debido a que los dispositivos iOS, solo realizan conexiones con Bluetooth v4.

Si se modifica la conexión de bluetooth v2 a v4, en el EMISOR se debe modificar el API usada (BluetoothSocket) a Bluetooth Low Energy o BLE.

Es recomendable el cifrado de la información que se transmite entre la aplicación emisor del profesor y las matrices led.

En la aplicación EMISOR, se recomienda mostrar los resultados de las posiciones de los estudiantes en forma de cuadrícula y no en formato lista.

La decodificación es un proceso altamente sensible, esto quiere decir que si falla un led en ser decodificado correctamente, el código identificador estaría mal por lo que es recomendable realizarla usando teléfonos que permitan una resolución de imágenes entre 640 x 480 y no mayor a 720 x 480, ya que si es menor la calidad de imagen es muy baja y complicaría la decodificación, en cambio, si es muy alta el dispositivo necesitaría grandes recursos para completar el proceso y el número de imágenes procesadas por segundo sería bajo imposibilitando la decodificación.

Es necesario que el ambiente del aula de clases se encuentre iluminado y las matrices led se ubiquen en la pared de tal manera que el color genere un alto contraste con las matrices negras y sean fácilmente detectadas por el decodificador.

Durante la estimación de la posición (fase 4) es necesario que el estudiante se mantenga quieto lo más posible mientras dure el proceso para que el valor estimado sea lo más preciso.

Se concluye una estimación de la posición es casi perfecta cuando el estudiante está ubicado en el centro de las matrices, mientras que el error va aumentando mientras se desplace hacia los lados debido al error de estimación de los ángulos de la cámara.

BIBLIOGRAFÍA

- [1] ESPOL, (2012, julio). Reglamento de los estudios de Pregrado en la ESPOL [online]. pp. 3
Disponible en: <http://www.dspace.espol.edu.ec/xmlui/handle/123456789/21002>
- [2] ARCOTEL, (2017, junio) Resumen Internet Móvil [Online]. Disponible en:
http://www.arcotel.gob.ec/wp-content/uploads/2017/05/Infografia4_2017-Internet-movil_v2.pdf
- [3] Arduino, (2015, septiembre), Arduino Uno Rev.3 Descripción [Online]. Disponible en:
<https://store.arduino.cc/usa/arduino-uno-rev3>
- [4] Texas Instruments, (2015, septiembre), Hoja de datos SN74HC595N [Online]. Disponible en: <http://www.ti.com/lit/ds/symlink/sn74hc595.pdf>
- [5] Sparkfun, (2016, enero), Hoja de datos de Diodo Led [Online], Disponible en:
<https://www.sparkfun.com/datasheets/Components/LED/COM-09590-YSL-R531R3D-D2.pdf>
- [6] Guangzhou HC Information Technology Co., Ltd., (2011, abril), Hoja de datos módulo Bluetooth HC-06 [Online]. Disponible en:
<https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>
- [7] OpenCV Library (2017, diciembre). Disponible en: <https://opencv.org>
- [8] 'Transform a frame to be as if it was taken from above using OpenCV', (2016, mayo).
Disponible en: <https://stackoverflow.com/questions/37117939/transform-a-frame-to-be-as-if-it-was-taken-from-above-using-opencv>
- [9] Vanishing Point, How to compute camera orientation. (2016, octubre) [Online]. Disponible en: <https://www.coursera.org/learn/robotics-perception/lecture/flqF4/vanishing-points-how-to-compute-camera-orientation>