

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro para las Investigaciones en Comunicaciones y Redes
Laboratorio de Redes Móviles, Inalámbricas y Distribuidas (ICARO)

Desarrollo de una Herramienta para el Control de Asistencias mediante el uso de Tecnología Bluetooth

Trabajo Especial de Grado presentado ante la Ilustre
Universidad Central de Venezuela por el Bachiller:
Reynaldo Reyes
C.I.: 17074195
E-mail: Reynaldo.Reyes.4@gmail.com

Para optar al título de Licenciado en Computación
Tutor: Profesor Miguel Ángel Astor.

Caracas, Mayo 2016
Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro para las Investigaciones en Comunicaciones y Redes
Laboratorio de Redes Móviles, Inalámbricas y Distribuidas (ICARO)



ACTA DEL VEREDICTO

Quienes suscriben, Miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por el Bachiller Reynaldo Reyes C.I.: 17074195, con el título **“Desarrollo de Herramienta para el Control de Asistencias mediante uso de Tecnología Bluetooth”**, a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 31 de Mayo de 2016, para que su autor lo defendiera en forma pública, mediante una exposición oral de su contenido, y luego respondió satisfactoriamente a las preguntas que le fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, dejándose también constancia de que actuó como Coordinador del Jurado el Profesor Tutor Miguel Astor.

Prof. Miguel Angel Astor
(Tutor)

Prof. Ana Morales
(Jurado Principal)

Prof. Antonio Russoniello
(Jurado Principal)

Resumen

Título: Desarrollo de una Herramienta para el Control de Asistencias mediante el uso de Tecnología Bluetooth.

Autor: Bachiller Reynaldo R. Reyes Rodríguez.

Tutor: Profesor Miguel Ángel Astor Romero.

El manejo de la asistencia de los alumnos al aula de clase de la Facultad de Ciencias de la Universidad Central de Venezuela es una necesidad según su reglamento vigente. Actualmente el cuerpo docente se ve obligado a tomar esta asistencia de manera manual y posteriormente transcribirla en algún tipo de formato para su gestión. El siguiente trabajo se propone aportar una alternativa a este proceso a través de la automatización del chequeo de asistencias utilizando una aplicación para el sistema operativo *Android* que, haciendo uso del escaneo de *Bluetooth*, chequee la asistencia asociando cada estudiante con el *MAC address* de su teléfono personal. La información de estas asistencias se registra a través de *Wi-Fi* en una aplicación Web, la cual se encarga de procesarla para que el profesor pueda gestionar estas asistencias a través de un portal personalizado por el profesor. Estas aplicaciones fueron probadas y evaluadas por los profesores de la Facultad de Ciencias a través de encuestas utilizando las heurísticas de *Nielsen* para garantizar un buen nivel de usabilidad.

Palabras Clave: Asistencia, *Bluetooth*, Escaneo *Bluetooth*, *Android*, Usabilidad, Heurísticas de *Nielsen*.

Índice de Contenido

1. Introducción.....	13
1.1. Planteamiento del Problema.....	14
1.2. Objetivo General.....	15
1.2.1. Objetivos Específicos	15
1.3. Justificación de la propuesta	15
1.4. Distribución del documento	16
2. Marco Teórico.....	18
2.1. Tecnología Bluetooth	18
2.1.1. Historia de Bluetooth	18
2.1.2. Arquitectura de Hardware	19
2.1.3. Pila de Protocolos.....	20
2.1.4. Radio de Bluetooth	21
2.1.5. Protocolo Banda Base	22
2.1.6. Host Controller Interface.....	23
2.2. Escaneo Bluetooth (<i>Bluetooth Scanning</i>)	24
2.2.1. Proceso <i>Inquiry</i> de la Banda base	25
2.2.2. Descubrimientos de Dispositivos con <i>HCI</i>	25
2.2.3. Datos e información generada.....	27
2.3. Teléfonos Inteligentes	28
2.3.1. Historia de los Teléfonos Inteligentes	28
2.3.2. Sistema Operativo – <i>Android</i>	29
2.4 <i>JavaScript</i>	31
2.4.1 <i>MongoDB</i>	32
2.4.2 <i>AngularJS</i>	32
2.4.3 <i>Node.js</i>	33
2.4.4 <i>Express.js</i>	35
2.5. Control de Asistencias	35
2.5.1. Dispositivos de Control de Asistencia.....	35
2.5.2. Sistema de Marcado de Tarjetas	36
2.5.3. Sistemas de Escaneo Biométrico	36

3. Metodología, Herramientas y Diseño	39
3.1 Metodología.....	39
3.2 Herramientas	40
3.3 Diseño e Implementación	41
3.3.1 Requerimientos del Sistema.....	41
3.3.2 Diseño de la Solución	42
3.4 Diagrama de Componentes.....	45
3.5 Diagrama de Casos de Uso.....	45
4. Resultados y Análisis de los Resultados	51
4.1 Escenarios.....	51
4.2 Descripción del Escenario	51
4.3 Implementación de los escenarios	51
5. Conclusiones.....	62
5.1 Contribuciones.....	62
5.2 Limitaciones.....	62
5.3 Trabajos Futuros.....	63
Anexo A: Cuadros de detalle de los casos de uso.	65
Referencias	75

Índice de Figuras

Figura 1: Pila de Protocolos Bluetooth.	21
Figura 2: Interfaz de Controlador del Anfitrión.....	24
Figura 3: Arquitectura de un dispositivo Android.	31
Figura 4: Metodología orientada a prototipos.....	40
Figura 5: Elementos de los escenarios.	44
Figura 6: Diagrama de Componentes.	45
Figura 7: Casos de uso Nivel 1, Módulo de Gestión de Materias.....	46
Figura 8: Casos de uso Nivel 1, Módulo de Gestión de Secciones.....	47
Figura 9: Casos de Uso Nivel 1: Módulo de Gestión de Matriculas	48
Figura 10: Casos de Uso Nivel 1: Modulo de Escaneo de Datos.....	49
Figura 11: Casos de Uso Nivel 1, Modulo de Administración Web.....	49
Figura 12: Visibilidad del estado del sistema.	53
Figura 13: Lenguaje del Usuario	54
Figura 14: Control y Libertad de Uso.	54
Figura 15: Consistencia y Estándares.....	55
Figura 16: Prevención de Errores.	56
Figura 17: Reconocimiento.	57
Figura 18: Flexibilidad y Eficiencia de Uso.....	57
Figura 19: Diseño Minimalista y Estético.	58
Figura 20: Facilidad para el diagnóstico y recuperación de errores.....	59
Figura 21: Ayuda y Documentación.	59

Índice de Tablas

Tabla 1: Versiones de API de Android.	30
Tabla 2: Caso de Uso Agregar Nueva Materia.	65
Tabla 3: Caso de Uso Eliminar Materia Existente.	65
Tabla 4: Caso de Uso Listar Materias	66
Tabla 5: Caso de Uso Gestionar Secciones	66
Tabla 6: Caso de Uso Agregar Sección.....	67
Tabla 7: Caso de Uso Eliminar Sección	68
Tabla 8: Caso de Uso Gestionar Matricula de Sección	68
Tabla 9: Caso de Uso Agregar Estudiante	69
Tabla 10: Caso de Uso Retirar Estudiante	69
Tabla 11: Caso de Uso Actualizar Correo del Estudiante	70
Tabla 12: Caso de Uso Liberar Dirección Bluetooth del Estudiante	70
Tabla 13: Caso de Uso Escanear Dispositivos.....	71
Tabla 14: Caso de Uso Culminar Clase.	71
Tabla 15: Caso de Uso Agregar Nuevo Profesor.....	72
Tabla 16: Caso de Uso Listar Profesores.....	72
Tabla 17: Caso de Uso Modificar Profesor.	73
Tabla 18: Caso de Uso Eliminar Profesor.....	73

Índice de Ecuaciones

Ecuación 1: Relación entre los periodos.	27
---	----

1. Introducción

Bluetooth es un estándar de tecnología inalámbrica utilizada para el intercambio de datos a través de distancias cortas, usando transmisiones de radio de onda corta en la banda ISM (*Industrial Scientific and Medical* - Industrial, Científica y Medica) desde los 2400 a los 2480 MHz, desde dispositivos móviles y fijos, creando redes PAN (*Personal Area Networks* - Redes de Área Personal) con altos niveles de seguridad. Creado por la compañía de telecomunicaciones Ericsson en el año 1994, fue originalmente concebido como una alternativa inalámbrica a los cables de datos RS-232. Puede conectar varios dispositivos, eliminando problemas de sincronización. Entre los protocolos que conforman el estándar Bluetooth está el Banda Base. Este introduce el concepto de *piconets*, que consisten en una estructura que forman varios dispositivos que se encuentran interconectados. A su vez, también se definen los procesos *Inquiry* y *Page*, usados en el escaneo *Bluetooth*. El proceso de *Inquiry* recolecta información sobre otros dispositivos *Bluetooth* cercanos. Mientras que el proceso *Page* invita a otros dispositivos a unirse a una *piconet*.

La información conseguida a través del proceso de *Inquiry* es clave para lo que se denomina escaneo *Bluetooth*, este se encarga de recolección de información específica de dispositivos como lo pueden ser la BD_ADDR (que identifica unívocamente a un dispositivo), el nombre del dispositivo, los distintos servicios que ofrece, la clase del dispositivo que nos puede indicar si este es: un teléfono celular, una computadora portátil, una computadora de escritorio, entre otros. Los datos recolectados podrán ser almacenados y asociados con la hora y el lugar donde fueron registrados. De este modo se pueden generar análisis que nos pueden indicar las características y el comportamiento de los dispositivos escaneados.

Adicional a esto, están los teléfonos inteligentes. Estos son teléfonos móviles que poseen un sistema operativo propio. Los sistemas operativos usados en la actualidad incluyen a *Android* de *Google*, *iOS* de *Apple*, *Symbian* de *Nokia*, *Windows Phone* de *Microsoft*, entre otros. Estos sistemas operativos permiten capacidades similares a las de un computador en términos de cómputo y conectividad, permitiendo conexiones *Wi-Fi* o vía red 3G, procesamiento de información, multitarea, capacidades multimedia (cámara y reproductor de videos/mp3), agendas, administración de contactos, *GPS*, navegadores y por sobre todo la posibilidad de instalar aplicaciones adicionales, bien desarrolladas por la empresa que creó el sistema operativo o por terceros que crean dichas aplicaciones a través de las herramientas de desarrollo puestas a disposición por la empresa.

Ambas tecnologías están en la actualidad al alcance de la mano del usuario común, dada la popularidad de dichos dispositivos, y su capacidad de movilidad abre una gran gama de posibilidades para el desarrollo de herramientas y aplicaciones que pueden ser usadas para optimizar o acelerar tareas que pueden requerir mucho tiempo, resultar tediosas o que estén propensas a errores humanos.

1.1. Planteamiento del Problema

El control de asistencia de los alumnos es una necesidad contemplada en el reglamento de asistencia de la Facultad de Ciencias de la Universidad Central de Venezuela, el cual en su artículo 2 se establece que el profesor debe de manera obligatoria comprobar la asistencia de los alumnos a cada clase. Igualmente, el artículo 11 establece que al menos una vez trimestralmente el profesor debe computar las inasistencias que el alumno posee, y de haber alcanzado el 25% o más de inasistencias deberá tomar las medidas respectivas [24]. Hasta el momento dicho control es realizado de forma manual por cada profesor y en algunos casos no se realiza ningún control.

El control de asistencia manual requiere que los alumnos coloquen su nombre, número de cédula y su firma en una hoja de papel provista por el profesor, con el fin de recopilar información que inequívocamente identifique a los alumnos que han asistido a clase. Sin embargo esto trae muchos problemas consigo, dado que pueden aparecer inconsistencias debido a que no se sabe en realidad si el sistema está siendo usado con honestidad. Los alumnos pueden firmar por otros, así como pueden firmar y posteriormente retirarse del aula de clase dado que su asistencia ya ha sido tomada o llegar en la última hora de la clase y firman su asistencia aun cuando según el reglamento dicho alumno no puede contar como que ha asistido a la clase. Adicional a esto, también existe la posibilidad en el que el profesor no chequee la asistencia de los alumnos ese día en particular debido a un descuido. Posteriormente el profesor debe transcribir la información de cada hoja de asistencia por cada clase a lo largo del semestre, lo cual multiplicado por la cantidad de alumnos por sección y las diferentes materias que el profesor pueda impartir representa un proceso largo, tedioso, engorroso y bastante repetitivo para poder calcular las asistencias de cada uno de los alumnos.

En la actualidad existen variantes para el control de asistencia. Algunas de estas tecnologías van desde el uso de alguna tarjeta la cual contiene el número del estudiante o número de identificación; como credencial con código de barras, credencial con banda magnética o una credencial de proximidad con una antena en su interior; hasta el uso de alguna característica fisiológica como lo pueden ser la huella digital, las venas de los dedos, la palma de la mano o inclusive el rostro de la persona. Sin embargo la inversión en hardware para el funcionamiento de

estas soluciones es considerable, y la realidad actual de la Universidad Central de Venezuela, en el cual la inversión sobre estas tecnologías es extremadamente limitada dados los ajustados presupuestos con los que se cuenta hoy en día obliga a implementar soluciones de bajo costo las cuales puedan ser desplegadas con una cantidad mínima de hardware.

Por lo antes dicho se requiere desarrollar un mecanismo que permita el control de asistencia a las aulas de clase y un mecanismo que permita gestionar la información de dichas asistencias de manera automática y segura.

1.2. Objetivo General

Desarrollar una herramienta que a través de escaneo de *Bluetooth* sobre la plataforma de un teléfono inteligente permita reconocer los dispositivos asociados a los estudiantes y el control de asistencia de los mismos en la Facultad de Ciencias de la UCV.

1.2.1. Objetivos Específicos

- Determinar los requerimientos para el control de asistencia de los estudiantes de la Facultad de Ciencias de la Universidad Central de Venezuela.
- Diseñar la herramienta de software en base a los requerimientos planteados.
- Implementar la herramienta.
- Realizar las pruebas de funcionamiento de la herramienta de software para validar el correcto funcionamiento de la misma.
- Analizar los resultados.

1.3. Justificación de la propuesta

Bluetooth es la tecnología para las comunicaciones inalámbricas de corto alcance con mayor presencia entre los dispositivos móviles de la actualidad. Virtualmente cada teléfono inteligente posee capacidad *Bluetooth* activada. Para el año 2018 se estima que más de 10 mil millones de los dispositivos en el mercado posean capacidad *Bluetooth* [1]. Por tales motivos surgen diversas aplicaciones que se apoyan en la mencionada tecnología para lograr mayor contacto con y entre las personas. Esta tecnología puede otorgar información del comportamiento y características de los dispositivos que estén a su alcance, lo que podría suponer en información relacionada a los dueños de tales dispositivos.

Los teléfonos inteligentes han tenido un auge en los últimos años, convirtiéndose en computadores de bolsillo, con la capacidad de ejecutar aplicaciones hechas a la medida que pueden utilizarse para recolectar, analizar y almacenar información. Además de esto, la masificación del uso de estos dispositivos y su popularidad ha acarreado en que se han puesto a la venta modelos de bajo costo en el mercado, los cuales tienen la capacidad de ejecutar aplicaciones sencillas.

Aprovechando la penetración de estas tecnologías dentro de la población se puede concebir el utilizarlas como un medio para facilitar tareas repetitivas específicas. Como se mencionó anteriormente, una de estas tareas propensas a error humano es el chequeo de asistencia, por lo tanto se puede concebir un sistema que permita el chequeo de asistencia dentro de las aulas de clase de la Facultad de Ciencias de la Universidad Central de Venezuela mediante el uso de la tecnología *Bluetooth* y teléfonos inteligentes.

1.4. Distribución del documento

El resto de este documento está estructurado en los siguientes capítulos:

- **Capítulo 2:** Contiene el marco teórico, donde se abarcan temas la tecnología *Bluetooth*, el escaneo *Bluetooth*, el sistema operativo *Android* y el chequeo de asistencia.
- **Capítulo 3:** Se describe la metodología a seguir, así como también los elementos de software, hardware y el diseño.
- **Capítulo 4:** Se describen los escenarios planteados las pruebas que permitieron medir el desempeño en cada escenario para poder lograr los objetivos y resolver el problema de investigación y se analizan e interpretan los resultados obtenidos de las pruebas realizadas.
- **Capítulo 5:** Presenta las conclusiones, contribuciones, limitaciones y recomendaciones para trabajos futuros.

2. Marco Teórico

2.1. Tecnología Bluetooth

Bluetooth es un estándar de tecnología de comunicaciones inalámbricas, creado para la comunicación entre dispositivos (como teléfonos, dispositivos de manos libres, monitores y equipos médicos, entre otros) sin la necesidad de estar interconectados físicamente por cables.

2.1.1. Historia de Bluetooth

El proyecto fue concebido en 1994 por la compañía sueca de telecomunicaciones *Ericsson* como una alternativa inalámbrica a los cables de datos RS-232 que permitiese la conexión de dispositivos de manera segura, a bajo costo y universalmente disponible a través de una banda de radio frecuencia de corto alcance [2]. Dicho enlace fue llamado MC link.

Luego la compañía apostó por el desarrollo de un estándar abierto para asegurar la interoperabilidad de esta tecnología. Para dicho fin se creó un consorcio de múltiples compañías, las cuales se encargarían de proteger e impulsar la marca de *Bluetooth*. Dicho consorcio nació en Septiembre de 1998 conocido como SIG (*Special Interest Group* – Grupo de Interés Especial) el cual en un primer momento estuvo formado por 5 compañías promotoras: *Ericsson*, *Intel*, *IBM*, *Nokia* y *Toshiba* [4].

Ese mismo año se adopta el nombre de Bluetooth para la tecnología. Este nombre proviene del rey escandinavo *Harald Blatand* que era apodado “*Bluetooth*” debido una enfermedad bucal, que le daba un tono azul a sus dientes [2]. El nombre Bluetooth fue adoptado por *Ericsson* con el propósito de unificar y concentrar esfuerzos de diferentes industrias como la de computación y la de telefonía celular, entre otras, para el desarrollo de la naciente tecnología inalámbrica.

Para el año 1999 es publicada la versión 1.0 del estándar Bluetooth y en diciembre de ese mismo año se unen al SIG cuatro compañías más: *3Com Corporation*, *Lucent Technologies Inc.*, *Microsoft* y *Motorola*. En el año 2000 salen al mercado los primeros dispositivos que hacen uso de la nueva tecnología. A partir de ese momento empiezan a salir gran variedad de aparatos equipados con *Bluetooth*, como: teléfonos celulares, computadoras de escritorio, computadoras portátiles, impresoras, periféricos, dispositivos manos libres.

Para el año 2002 había más de quinientos productos disponibles en el mercado. Ese mismo año la *IEEE* crea el grupo de trabajo 802.15.1 [10] que definiría la capa física (*PHY*) y de control de acceso al medio (*MAC*) de las redes PAN.

En el año 2003 se lanza la nueva versión 1.2 del estándar [3], a partir de ese momento cada nueva versión tendrá retro-compatibilidad. Y en noviembre del año 2004 se adopta la nueva versión 2.0 que traería consigo una mejora en la tasa de datos que se transmiten de un dispositivo a otro. En julio de 2007 se actualiza a la versión 2.1 cuyas características más relevantes son la reducción en el consumo de energía y una mejora en la seguridad, además de que se hace necesaria la interacción del usuario para conectar dos dispositivos.

En 2009 se adopta la versión 3.0 de alta velocidad del estándar *Bluetooth* [3], la cual cuenta con las siguientes características: El uso de AMP (*Alternate MAC/PHY* – *MAC/PHY* Alternas): que permite la utilización alternativa de una capa física y de control de acceso al medio, aunque aún se utiliza el radio de Bluetooth para el descubrimiento, asociación y configuración de perfiles. Cuando es necesaria la transmisión de grandes cantidades de datos se hace uso de una capa física y de control de acceso al medio alternativas como la del estándar 802.11 (*WiFi*).

El 30 de Junio del año 2010 se adopta la versión 4.0 [5], la cual tiene como principal cualidad el bajo consumo de energía. Gracias a este diseño, esta nueva tecnología consume solo una fracción de la energía comparada con sus predecesoras.

La versión 4.0 contiene todas las especificaciones en una (la tecnología clásica de Bluetooth, la tecnología de bajo consumo de energía de Bluetooth y la tecnología de alta velocidad de Bluetooth), las cuales pueden ser usadas en conjunto o separadas de acuerdo a las capacidades del dispositivo [4].

2.1.2. Arquitectura de Hardware

El hardware que compone el dispositivo *Bluetooth* está formado por dos partes. Un dispositivo de radio, encargado de modular y transmitir la señal. Y un controlador

digital que está compuesto por un *CPU*, por un *DSP* (*Digital Signal Processor* – *Procesador de Señales Digitales*) llamado *LC* (*Link Controller* - *Controlador de Enlace*) y de las interfaces con el dispositivo anfitrión [12].

El *LC* está encargado de hacer el procesamiento de la Banda Base y del manejo de los protocolos *ARQ* (*Automatic Repeat reQuest* – *Repetición de Petición Automática*) y *FEC* (*Forward Error Correction* – *Corrección de Errores hacia Adelante*) de la capa física. Además, se encarga de las funciones de transferencia (tanto asíncrona como síncrona), codificación de audio y cifrado de datos. Por otro lado, el *CPU* del dispositivo se encarga de atender las instrucciones relacionadas con Bluetooth del dispositivo anfitrión para así simplificar su operación. Para ello el *CPU* se comunica con otros dispositivos por medio del protocolo *LMP* (*Link Manager Protocol* – *Protocolo de Manejo de Enlaces*).

2.1.3. Pila de Protocolos

El objetivo principal de la especificación de *Bluetooth* es que todas las aplicaciones y dispositivos que implementan esta tecnología sean capaces de operar entre sí. Para conseguir esta interoperabilidad, las aplicaciones en dispositivos remotos deben ejecutarse sobre la misma pila de protocolos.

Diferentes pilas de protocolo son usadas por aplicaciones. Sin embargo los protocolos del “Núcleo de *Bluetooth*” deben ser implementados por la mayoría de los dispositivos. La especificación de *Bluetooth* define una interfaz entre el dispositivo anfitrión y el dispositivo *Bluetooth* denominada *HCI* (*Host Controller Interface* – *Controlador de Interfaz del Anfitrión*). Esta se puede ubicar debajo de la capa *L2CAP* (*Logical Link Control and Adaptation Protocol* – *Protocolo de Adaptación y Control de Enlace Lógico*) o también puede posicionarse encima de esta [18]. En la Figura 1 se puede observar cómo se comunican los niveles de los protocolos usados.

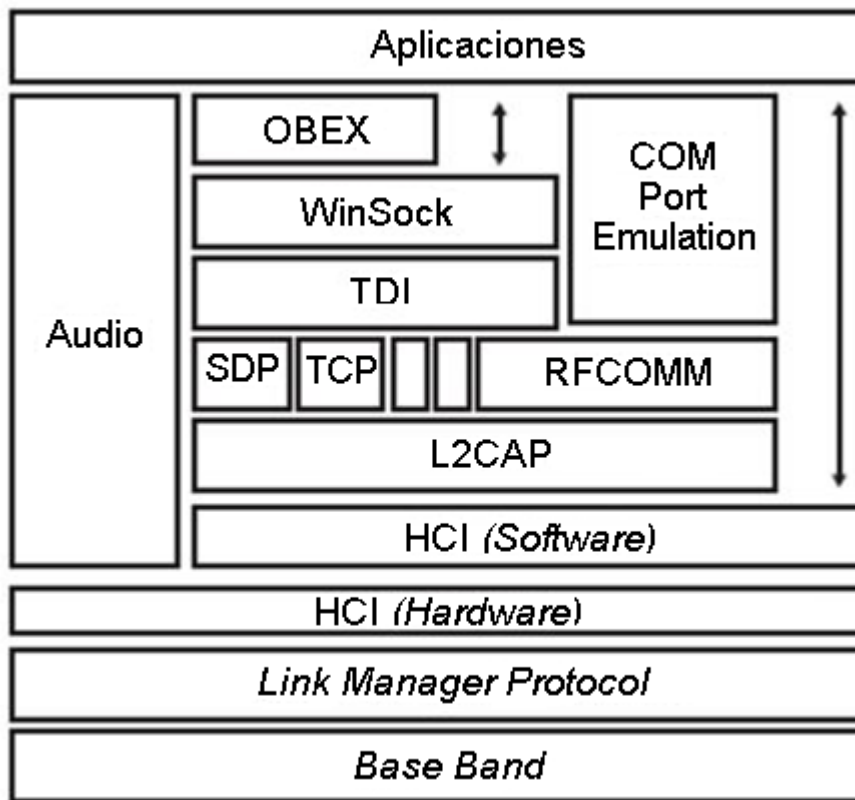


Figura 1: Pila de Protocolos Bluetooth. Tomado de [42]

2.1.4. Radio de Bluetooth

Bluetooth trabaja en la banda ISM de 2.4 GHz a 2.5 GHz, destinada para su uso en el área industrial, científica o médica. Las autoridades reguladoras alrededor del mundo han abierto dichas bandas para su uso en sistemas de bajo poder de emisión, que pueden trabajar sin la necesidad de una licencia, pero bajo una estricta regulación.

Bluetooth trabaja con el método *FHSS* (*Frequency Hopping Spread Spectrum* - transmisión de saltos de frecuencia de espectro ensanchado) que opera sobre 79 canales, de 1 MHz de ancho cada uno, comenzando en la mayoría de los países en la frecuencia de 2.502 GHz y acabando en la frecuencia de 2.5835 GHz. Regulaciones en ciertos países pueden contraer la banda ISM de 2.5 GHz. Por tal motivo el radio de *Bluetooth* y el protocolo de Banda Base (que se describe más adelante) pueden adaptarse a dos alternativas, una donde opera con 79 canales y otra con 23 canales, cada uno de 1 MHz de ancho. Y para tener en cuenta las emisiones fuera de banda se establecen intervalos de guarda al principio y al final de la banda de frecuencias de trabajo.

Para *FHSS* en la banda *ISM* de 2.5 GHz las regulaciones restringen un pico máximo de poder de salida del emisor de no más de 1 watt (30 dBm) [17]. Por otra parte, 75 de los 79 canales de frecuencia deben ser usados de forma pseudo-aleatoria con un tiempo total sobre cada canal que no exceda los 0,4 segundos dentro de un período de 30 segundos. Bluetooth utiliza el máximo número de canales disponibles, con una alta tasa de saltos, 1.600 saltos pseudo-aleatorios por segundo a través de todas estas frecuencias. De este modo se puede lograr una gran resistencia al ruido.

Bluetooth emplea *GFSK* (*Gaussian Frequency Shift Keying* - Modulación por Desplazamiento de Frecuencia Gaussiana), con las siguientes características:

- El índice de modulación debe estar entre 0.28 y 0.35.
- Un uno binario se representa por una desviación positiva de frecuencia y un cero binario como una desviación negativa.
- La desviación mínima no ha de ser menor de 115 KHz.
- El error de los pasos por cero (diferencia de tiempo entre el período de símbolo ideal y el tiempo de cruce medido) debe ser menor de $\pm 1/8$ del período de símbolo.

2.1.5. Protocolo Banda Base

En la sección anterior se describió el protocolo de Radio de Bluetooth que se encarga de definir la forma como se envían y reciben los datos. Sin embargo, no se han mencionado consideraciones como qué datos se transmiten y cuándo, qué datos se esperan recibir y en qué momento, cuál es la frecuencia portadora y qué potencia se va a utilizar para transmitir estos datos. Esta responsabilidad recae sobre el LC, el cual implementa el protocolo de Banda Base y los procesos relacionados con este.

Este protocolo introduce el concepto de *piconets* y como estas son creadas. Las *piconets* son redes en las que un dispositivo está conectado con hasta otros 7 de manera activa y con hasta otros 255 que se encuentran inactivos, los cuales pueden ser activados por dicho dispositivo en cualquier momento. Más adelante se definirá con profundidad este tipo de redes. A su vez, debido a que Bluetooth hace uso de *FHSS* para transmitir los paquetes en espacios de tiempo definidos sobre un conjunto de frecuencias determinadas, este protocolo define los procesos de *Inquiry* (Pregunta) y *Page* (Búsqueda) para sincronizar la secuencia de saltos de frecuencia y el reloj entre dos dispositivos Bluetooth. También se definen los

paquetes de bajo nivel usados.

2.1.6. Host Controller Interface

La HCI proporciona una interfaz de comandos para el controlador de la Banda Base, el gestor de enlaces, y acceso a los parámetros de configuración. Dicha interfaz facilita un método uniforme de acceso a las capacidades de la Banda Base de *Bluetooth*.

La Figura 2 proporciona una visión general de las capas inferiores de software. El *firmware HCI* implementa los comandos *hardware HCI* mediante el acceso a los comandos de gestión a nivel de enlace, a los registros de *hardware* de estado, registros de control y de eventos. Varias capas pueden existir entre el controlador de *HCI* en el sistema anfitrión y el *firmware HCI* del dispositivo *Bluetooth*. Estas capas intermedias, ofrecen la posibilidad de transferencia de datos sin la necesidad del conocimiento íntimo de los mismos.

HCI es usado para aislar la Banda Base de *Bluetooth* y el administrador de enlaces de un protocolo de transporte tal como el *RS-232* o *USB* (*Universal Serial Bus* - Bus Serial Universal). Esto permite una interfaz estándar para el *hardware* de *Bluetooth*. Un manejador de dispositivos *HCI* en el anfitrión es usado para interactuar con una aplicación *Bluetooth* con el protocolo de transporte. Actualmente existen tres mecanismos de transporte soportados: *USB*, *RS-232* y el *UART* (*Universal Asynchronous Receiver/Transmitter* – Transmisor/Receptor Asíncrono Universal). Utilizando *HCI*, una aplicación *Bluetooth* puede acceder al *hardware* de *Bluetooth* sin el conocimiento de la capa de transporte u otros detalles de implementación del hardware.

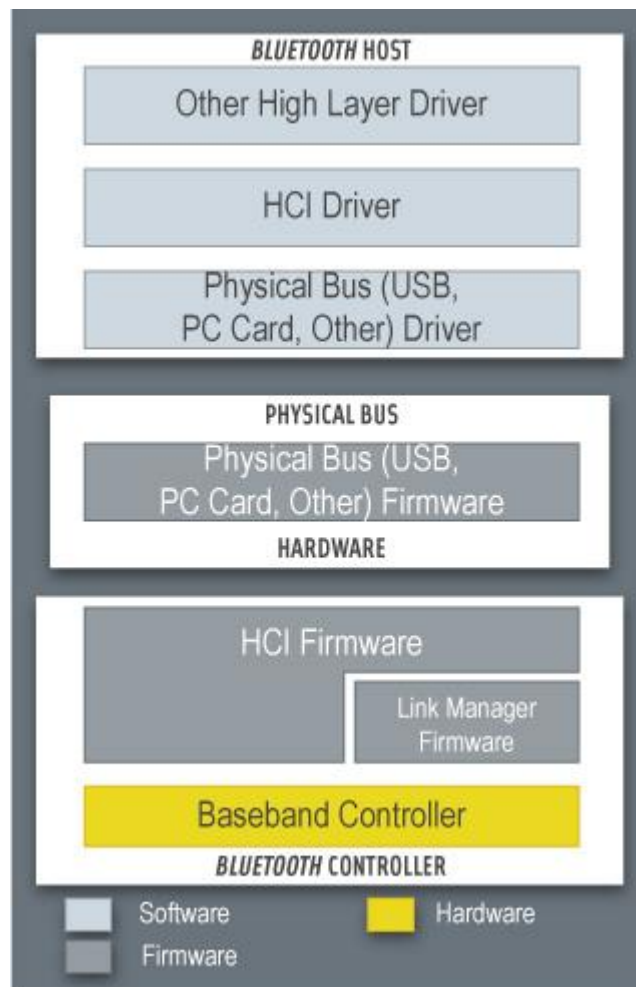


Figura 2: Interfaz de Controlador del Anfitrión. Tomado de [43].

2.2. Escaneo Bluetooth (*Bluetooth Scanning*)

Debido a la masificación que ha tenido la tecnología Bluetooth, hoy en día es común que los dispositivos móviles como teléfonos celulares, teléfonos inteligentes, tabletas, PDAs, computadoras portátiles, entre otros, incluyan dentro de sus funcionalidades básicas la mencionada tecnología.

Dentro de las características de Bluetooth se encuentra la posibilidad que tiene cada dispositivo de recolectar información de otros dispositivos Bluetooth que se encuentren en un radio de alcance cercano. Esta información puede resultar bastante útil para diversos fines.

El escaneo *Bluetooth* involucra una serie de subprocesos que permiten recolectar información de los dispositivos dentro del radio de alcance del dispositivo que actúe como escáner *Bluetooth*, manipular los datos relevantes que se generen de dicho escaneo y almacenarlos para su posterior uso. A continuación se describen estos subprocesos que en conjunto permiten llevar a cabo un escaneo *Bluetooth*.

2.2.1. Proceso *Inquiry* de la Banda base

Durante el proceso de descubrimiento de otros dispositivos *Bluetooth*, se emplea el proceso *Inquiry* que se explicó en detalle en el Capítulo 2, el cual está compuesto por un sub-estado *Inquiry* que es asumido por el potencial maestro de una *piconet* y dos sub estados, *Inquiry Scan* e *Inquiry Response* que son asumidos por los potenciales esclavos de una *piconet*.

Este proceso posee algunas desventajas. En primer lugar, toma mucho tiempo y produce un alto consumo de recursos si se usa regularmente. Y en segundo lugar, es bastante ineficiente para transmitir datos de forma simultánea. Sin embargo puede resultar bastante útil la utilización de este método para recolectar información de los dispositivos vecinos.

2.2.2. Descubrimientos de Dispositivos con *HCI*

La especificación de *Bluetooth* en la sección *HCI* define una interfaz de comandos y eventos asociados al descubrimiento de dispositivos que se encuentren en un área cercana. Estos comandos y eventos asociados al descubrimiento de dispositivos se describen a continuación:

- ***Inquiry Command***: Este comando permite que el dispositivo *Bluetooth* entre en Modo *Inquiry*. Este modo es usado para descubrir otros dispositivos *Bluetooth* cercanos.
- ***Inquiry Result Event***: Este evento indica que uno o múltiples dispositivos *Bluetooth* han respondido al proceso *Inquiry* actual.
- ***Inquiry Result with RSSI Event***: Este evento indica que uno o múltiples dispositivos *Bluetooth* han respondido al proceso *Inquiry* actual incluyendo la información *RSSI*.
- ***Inquiry Cancel Command***: Este comando trae como consecuencia la detención del proceso *Inquiry* actual.
- ***Inquiry Complete Event***: Este evento indica que el proceso *Inquiry* actual ha concluido.
- ***Periodic Inquiry Mode Command***: este comando es usado para configurar al dispositivo *Bluetooth* para que efectúe el proceso *Inquiry* de

forma automática y periódicamente.

- **Exit Periodic Inquiry Mode Command:** este comando es utilizado para salir del modo *Inquiry* periódico.
- **Read Inquiry Scan Activity Command:** Este comando permite leer los valores de los parámetros de configuración para los intervalos de escaneo *Inquiry*, como también la ventana de escaneo *Inquiry*. El intervalo de escaneo *Inquiry* define la cantidad de tiempo entre dos escaneos consecutivos *Inquiry*, y la ventana de escaneo *Inquiry* define la duración del escaneo *Inquiry*.
- **Write Inquiry Scan Activity Command:** Este comando permite escribir los valores de los parámetros de configuración para los intervalos de escaneo *Inquiry*, como también la ventana de escaneo *Inquiry*. El intervalo de escaneo *Inquiry* define la cantidad de tiempo entre dos escaneos consecutivos *Inquiry*.
- **Read Inquiry Scan Type Command:** Este comando es usado para leer los parámetros de configuración de los tipos de escaneo *Inquiry* del dispositivo. Estos parámetros pueden establecer el escaneo *Inquiry* sea de forma normal o de barrido entrelazado.
- **Write Inquiry Scan Type Command:** Este comando es usado para escribir los parámetros de configuración de los tipos de escaneo *Inquiry* del dispositivo. Estos parámetros pueden establecer el escaneo *Inquiry* sea de forma normal o de barrido entrelazado.
- **Read Inquiry Mode Command:** Este comando es utilizado para leer el parámetro de configuración del modo *Inquiry* del dispositivo *Bluetooth* local.
- **Write Inquiry Mode Command:** Este comando es utilizado para escribir el parámetro de configuración del modo *Inquiry* del dispositivo *Bluetooth* local..

Comando *Inquiry*

Este comando provoca que el dispositivo *Bluetooth* entre en un modo *Inquiry*. Este modo es usado para descubrir otros dispositivos. El parámetro de entrada contiene el *LAP* (*Lower Address Part* – Parte Baja de la Dirección) de donde el código de acceso *Inquiry* va a ser derivado cuando se realice el proceso. El parámetro de *Inquiry_Length* especifica el tiempo de duración de este modo, y el último parámetro *Num_Responses* especifica el número de respuestas que podrán ser recibidas antes de que el modo *Inquiry* concluya. Cuando el proceso termina, el controlador envía un evento de *Inquiry* completado al anfitrión indicando que el proceso ha terminado. Los parámetros de este evento contendrán un resumen de los resultados obtenidos de este proceso, donde se reportan el número de dispositivos *Bluetooth* que respondieron. Es de notar que cuando un dispositivo responde a un mensaje de pregunta, un evento *Inquiry Result* es generado para notificar al anfitrión del descubrimiento.

Comando *Inquiry Periódico*

El comando *Periodic_Inquiry_Mode* usado para configurar al dispositivo *Bluetooth* para que efectúe el proceso *Inquiry* de forma automática y periódicamente. Los parámetros *Max_Period_Length* y *Min_Period_Length* definen el rango de tiempo entre dos preguntas consecutivas, desde el inicio de una pregunta hasta el comienzo de la próxima pregunta. El controlador va a utilizar este rango para determinar un nuevo tiempo aleatorio entre dos preguntas consecutivas para cada pregunta. El parámetro de entrada contiene el *LAP* de donde el código de acceso de pregunta va a ser derivado cuando se realice el proceso de pregunta. El parámetro de *Inquiry_Length* especifica el tiempo de duración del modo de *Inquiry*. Y el último parámetro *Num_Responses* especifica el número de respuestas que podrán ser recibidas antes de que el modo de pregunta concluya. Este comando es completado cuando el proceso de pregunta ha sido iniciado por el dispositivo *Bluetooth*, un evento de comando completado es enviado del controlador al anfitrión. Cuando todo el proceso *Inquiry* periódico es completado, el controlador va a enviar un evento de pregunta completada al anfitrión indicando que el último proceso de pregunta periódica ha terminado. Cuando un dispositivo *Bluetooth* responde a un mensaje de pregunta, un evento de resultado de pregunta es generado para notificar al anfitrión del descubrimiento. Como se nota en la Ecuación 1.

$$Max_Period_Length > Min_Period_Length > Inquiry_Length$$

Ecuación 1: Relación entre los periodos.

2.2.3. Datos e información generada

Todo dispositivo *Bluetooth* es capaz de realizar descubrimiento de dispositivos, que le permiten recolectar información de otros dispositivos en un radio de 5 a 10 metros. Esta información incluye la *BD_ADDR*, el nombre del dispositivo, la clase del dispositivo, los servicios del dispositivo y opcionalmente información *RSSI* [20].

La dirección del dispositivo *Bluetooth* es un identificador único del dispositivo que lo diferencia de cualquier otro. El nombre del dispositivo es un campo que puede ser establecido por el usuario, por ejemplo: "Teléfono de Pedro". La clase y servicios del dispositivo es un conjunto de tres enteros que corresponden a los servicios que ofrece el dispositivo y el tipo de dispositivo que ha sido descubierto, por ejemplo: "Teléfono Celular", "Computadora Portátil", "Computadora de Escritorio". Por último la información *RSSI* nos provee información del nivel de potencia de la señal que existe con el dispositivo descubierto.

Estos datos pueden ser almacenados para producir información útil que puede ser utilizada en distintas aplicaciones en diferentes campos. Estos se pueden clasificar de la siguiente forma: las redes sociales, el rastreo y localización bajo techo, la medición de flujo de personas, la computación ubicua, entre otros. Dependiendo del caso, se almacenarán los datos que sean necesarios y luego serán interpretados de acuerdo a las necesidades.

2.3. Teléfonos Inteligentes

El término teléfono inteligente o *Smartphone* se refiere a un teléfono móvil construido sobre un sistema operativo móvil avanzado que combina características de los sistemas operativos de los computadores personales y características propias de los dispositivos telefónicos. Adicional a esto, otras características generales de un teléfono inteligente son la de poseer una interfaz para el ingreso de datos, bien sean un teclado físico o a través de una pantalla táctil, ofrecer capacidad de conexión a Internet a través de varias vías, soporte de manejo de correo electrónico, posicionamiento *GPS*, administración de contactos, y la que sería la más importante en los últimos años, que es la instalación de aplicaciones sobre el SO del teléfono el cual permite una experiencia más personalizada para el usuario.

2.3.1. Historia de los Teléfonos Inteligentes

Los dispositivos que combinan telefonía y cómputo han sido conceptualizados desde 1973, y su primera aparición en el mercado fue a mediados de 1994 cuando *IBM* lanzó su ***IBM Simon*** [26], el cual además de realizar llamadas permitía la recepción de fax y correos electrónicos. Sin embargo el término teléfono inteligente no fue acuñado sino hasta 1997, cuando *Ericsson* lo usó para describir el concepto detrás de su GS 88 “Penélope”.

En 1996, *Nokia* lanzó al mercado su *Nokia 9000*, el cual se convirtió en el teléfono más vendido en aquella época. Se trataba de un teléfono *palmtop* combinado con un PDA (Personal Digital Assistance – Asistente Digital Personal) de *HP*. En los primeros prototipos ambos dispositivos estaban ensamblados a través de una bisagra. Al abrirse, se tenía una pantalla de 640 x 200 píxeles en el tope con un teclado QWERTY en el fondo. El dispositivo permitía la recepción de correo electrónico y navegación *Web* basada en texto a través del sistema operativo *GEOS*.

A finales de los 90s la mayoría de los teléfonos móviles poseían solo capacidades de telefonía básica, así que el uso de dispositivos *PDA* por separado era común,

los cuales poseían versiones tempranas de sistemas operativos como *Palm OS*, *BlackBerry OS* o *Windows CE* [22].

Al principio de 2001, Palm Inc. Introdujo al mercado el *Kyocera 6035*. El cual combinaba las capacidades de un *PDA* con un teléfono móvil, además de dar soporte para navegación *Web* limitada. Luego entre 2002 y 2004 *HTC* lanzo en Europa sus 3 modelos "*Wallaby*", "*Falcon*" e "*Himalaya*" los cuales utilizaban el sistema operativo *Windows Mobile "Pocket PC"* [28].

2.3.2. Sistema Operativo – *Android*

El sistema operativo ***Android*** es una plataforma de código abierto basado en el ***kernel*** de ***Linux*** fue desarrollado por la empresa *Android Inc.* fundada en Octubre de 2003 por Andy Rubin y posteriormente adquirida por la empresa *Google* en 2005. Luego desarrolladores de software y hardware como lo son *Intel*, *HTC*, *Motorola* y *Samsung* entre otros, forman la **Open Handset Alliance** con el fin de establecer estándares abiertos para dispositivos móviles [21] [25].

El primer teléfono en utilizar *Android* fue el *HTC Dream* en 2008 basado sobre el *kernel* de *Linux* versión 2.6. La suite de software que incluía el teléfono consistía en la integración de aplicaciones propietarias de *Google*, como *Google Maps*, *Google Calendar*, *Gmail* y un navegador *Web*.

Desde 2008, *Android* ha tenido numerosas actualizaciones las cuales han mejorado de manera incremental el desempeño del sistema operativo y arreglando problemas de versiones anteriores. Cada actualización mayor se identifica en orden alfabético con el nombre de un postre o dulce, como se muestra en la Tabla 1 [29] [7].

Versión	Nombre	Fecha	Nivel de API
1.5	Cupcake	Abril 2009	3
1.6	Donut	Septiembre 2009	4

2.0 – 2.1	Eclair	Octubre 2009	7
2.2	Froyo	Mayo 2010	8
2.3 – 2.3.2	Gingerbread	Diciembre 2010	9
2.3.3 – 2.3.7	Gingerbread	Febrero 2011	10
3.1	Honeycomb	Mayo 2011	12
3.2	Honeycomb	Julio 2011	13
4.0.3 – 4.0.4	Ice Cream Sandwich	Diciembre 2011	15
4.1.x	Jelly Bean	Julio 2012	16
4.2.x	Jelly Bean	Noviembre 2012	17
4.3	Jelly Bean	Julio 2013	18
4.4	KitKat	Octubre 2013	19
5.0	Lollipop	Noviembre 2014	21
6.0	Marshmallow	Octubre 2015	23

Tabla 1: Versiones de API (*Application Programming Interface* – *Interfaz de programación de aplicaciones*) de Android.

El sistema operativo *Android* puede ser visto como una pila de diferentes capas, donde cada capa es una agrupación de varios componentes de programa. Todos ellos comprenden sistema operativo, *middleware* y algunas aplicaciones importantes. Los componentes de una capa solo proveen servicios a los componentes de la capa directamente superior a él. En la Figura 3 se puede ver el diagrama de la Arquitectura.

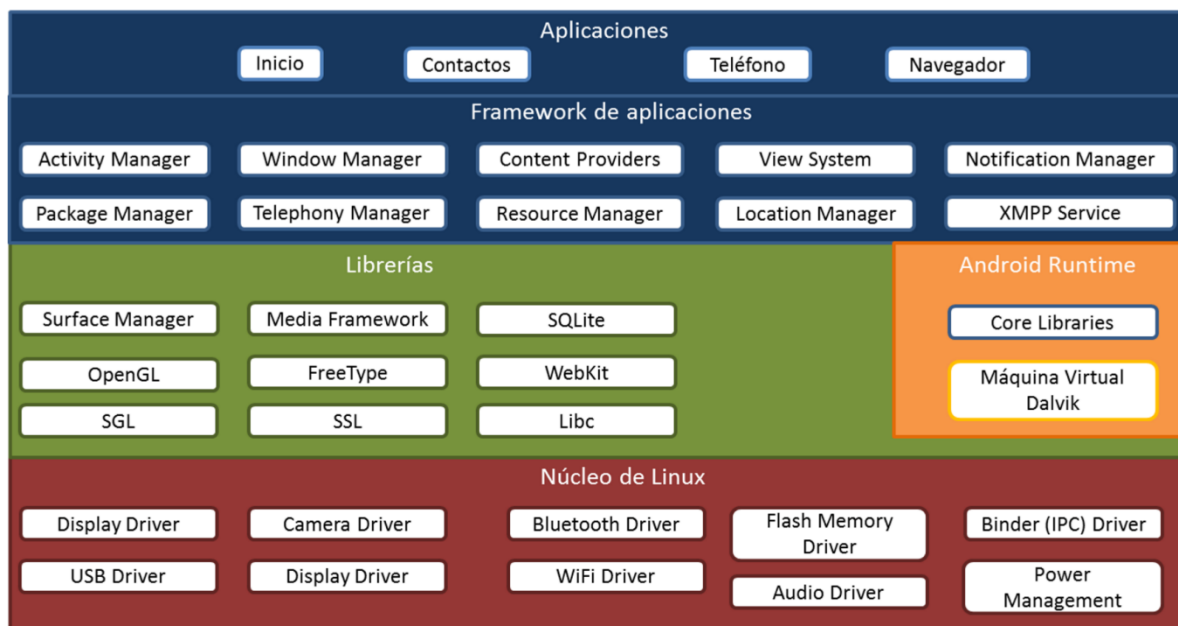


Figura 3: Arquitectura de un dispositivo *Android*. Tomado de [40]

2.4 JavaScript

JavaScript es un lenguaje de programación interpretado, dinámico, orientado a objetos, basado en prototipos, imperativo, débilmente tipado. Estandarizado en la especificación *ECMAScript*.

Es junto con *HTML* y *CSS* una de las tecnologías base para la producción de desarrollo de contenido en *Internet*. La mayoría de las páginas *web* lo emplean y es soportado por todos los navegadores modernos sin la necesidad de *plug-ins*. Generalmente es usado como un lenguaje interpretado de lado del cliente, pero con el uso de compilación en tiempo de ejecución se han desarrollado plataformas para el uso de *JavaScript* como lenguaje del lado del servidor como es el caso de *Node.js*.

Para el desarrollo de este trabajo especial se utilizó *JavaScript* tanto del lado del servidor como del lado del cliente a través de la implementación del “*M.E.A.N. Stack*” (*MongoDB* como manejador de bases de datos no relacional, *Express.js* como *middleware*, *AngularJS* como *framework* de desarrollo del lado del cliente y *Node.js* como sistema de desarrollo del lado del servidor) para crear las funcionalidades completas de la aplicación *web*.

2.4.1 MongoDB

MongoDB (que toma su nombre de la palabra inglesa *humongous* significando muy grande) desarrollado en el año 2007 por la compañía *10gen* es una base de datos orientada a documentos multiplataforma. Es clasificada como una base de datos no relacional, no *SQL*, dado que no utiliza el enfoque de las estructuras de bases de datos común de los manejadores tradicionales, en cambio está orientado a documentos de esquema dinámico basados en *JSON* (llamado *BSON* en *MongoDB*), son de esquema dinámico porque cada entrada puede ser un esquema de datos diferente al resto de los registros almacenados. Este formato permite que la integración de datos en ciertos tipos de aplicación sea rápidas y sencillas.

Entre las funcionalidades que posee *MongoDB* están:

Consultas *Ad hoc*: *MongoDB* soporta consultas de rango, de campo y búsquedas con expresiones regulares. Las búsquedas pueden retornar campos específicos e inclusive pueden retornar funciones de *JavaScript* definidas por el usuario.

Indexado: Cualquier campo de un documento de *MongoDB* puede ser indexado, incluyendo arreglos y documentos embebidos.

Replicación: *MongoDB* provee una gran disponibilidad para sets de replicación. Un set de replicación consiste en dos o más copias de datos. Cada replica puede actuar como la réplica primaria o secundaria en cualquier momento. La réplica primaria realiza todas las acciones de escritura y lectura por defecto. Las réplicas secundarias mantienen una copia de los datos de la primaria usando replicación nativa. Cuando la réplica primaria falla, el set de replicación hace un proceso de elección para determinar cuál replica secundaria debería reemplazarla. Las réplicas secundarias pueden opcionalmente realizar las operaciones de lectura.

Balanceo de Cargas: *MongoDB* escala horizontalmente utilizando un proceso conocido como *sharding*. El usuario escoge una llave de *shard*, la cual determinará como se distribuirán los datos en la colección. Los datos se separan en rangos y se distribuyen a través de múltiples *shards* (un *shard* es un maestro con uno o más esclavos). Esto se hace con la finalidad de poder correr *MongoDB* en múltiples servidores, bien sea para balancear la carga o para duplicar los datos y mantener el sistema funcional en caso de problemas de *hardware*.

Ejecución de *JavaScript* del lado del servidor: *JavaScript* puede ser usado en las búsquedas, funciones de agregación (como *MapReduce*) pueden ser enviadas directamente a la base de datos para ser ejecutadas.

2.4.2 AngularJS

AngularJS es un *framework* basado en *JavaScript* de aplicaciones web de fuente abierta mantenido por *Google* utilizado para la creación de aplicaciones web *single-page*. Funciona leyendo primero la página *HTML*, que tiene embebida

atributos adicionales. Angular interpreta estos atributos como directivas las cuales enlazan partes de la entrada o salida de la página a un modelo que es representado en variables estándar de *JavaScript*. El valor de estas variables puede ser puesto manualmente dentro del código o puede ser recibido desde recursos *JSON* estáticos o dinámicos.

AngularJS está construido sobre la creencia que la programación declarativa debería ser usada para crear interfaces de usuario y conectar componentes de *software*, mientras que la programación imperativa esta mejor capacitada para definir la lógica de negocio de una aplicación. Adaptando y extendiendo el *HTML* tradicional *AngularJS* presenta contenido dinámico a través de un enlazado de doble vía permitiendo sincronización entre las vistas y los modelos. Como resultado de esto, se le quita énfasis a la manipulación explícita del *DOM* con la meta de mejorar el desempeño y la capacidad de hacer pruebas.

AngularJS implementa el modelo de *MVC* para separar la presentación, los datos y los componentes lógicos. Usando inyección de dependencias, *AngularJS* permite que los servicios de lado del servidor tradicionales (como son los controladores dependientes de la vista) pasen al lado del cliente de las aplicaciones web. Esto reduce la carga que el servidor.

AngularJS usa un elemento llamado “*scope*” como el entorno de aplicación conocido fundamentalmente en las ciencias de la computación. Es el contexto de ejecución para las expresiones. El *scope* está distribuido de forma jerárquica haciendo mímica de la estructura *DOM* de la aplicación. Definido en el *ECMA-262* como: “un entorno léxico en el cual un objeto función es ejecutado en los scripts web del lado del cliente. [35]

Es usado como un tipo de objeto que puede estar dentro o fuera de otro entorno o “*scope*” en cualquier parte del programa, siguiendo las reglas de las variables de entorno de *JavaScript* como si fuese cualquier otro objeto.

2.4.3 Node.js

Node.js es un entorno de aplicación de código abierto para el desarrollo de aplicaciones web del lado del servidor. Aun cuando *Node.js* no es un *framework* de *JavaScript*, la mayoría de sus módulos básicos y la capacidad de crear nuevos módulos están basados sobre *JavaScript*, El entorno de ejecución interpreta *JavaScript* usando el motor *V8* de *Google*.

Node.js tiene una arquitectura orientada a eventos con la capacidad de I/O asíncrono. Estas elecciones de diseño están orientadas a la optimización del *throughput* y la escalabilidad en aplicaciones web con muchas operaciones de entrada/salida. [36]

Node.js permite la creación de servidores web y herramientas de red usando una

colección de módulos que manejan funcionalidades básicas y secundarias. Estos módulos usan un *API* diseñado para reducir la complejidad de la escritura de aplicaciones de servidor. [37]

Node.js puede ser considerado un “proto-servidor” que procesa peticiones entrantes en un ciclo de eventos. Entre sus funcionalidades básicas se encuentran:

Manejo de hilos: *Node.js* opera usando un solo hilo, realizando llamadas no bloqueantes de entrada/salida permitiendo el soporte de decenas de miles de conexiones concurrentes sin incurrir en el costo de cambio de contexto de hilos. El diseño de compartir un solo hilo entre todas las peticiones que usen el patrón de diseño *observer* está hecho con la intención de construir aplicaciones altamente concurrentes, donde cualquier función que realiza una llamada de entrada/salida debe usar un *callback*. Para poder acomodar el ciclo de eventos de un solo hilo, *Node.js* utiliza la biblioteca *libuv* que a su vez usa un pool de hilos de tamaño fijo que es responsable por todas las operaciones de entrada/salida no bloqueantes. [36]

La desventaja de este enfoque es que *Node.js* no permite la escalabilidad vertical incrementando el número de núcleos de *CPU* de la máquina que lo corre sin el uso de módulos adicionales (como *cluster*, *StrongLoop Process Manager* o *pm2*). Sin embargo, los desarrolladores puede incrementar el número por defecto de hilos en el pool de hilos de *libuv*, los cuales serían propensos a distribuirse a través de los múltiples núcleos del sistema operativo del servidor. [38]

La ejecución de tareas paralelas en *Node.js* es manejada por un pool de hilos. Las llamadas a función del hilo principal ponen tareas en la cola compartida de tareas, a su vez esta hilvana los hilos de tareas en el pool de hilos y los ejecuta. Las llamadas que inherentemente no bloquean las funciones de sistema (como funciones de red) se traducen a *sockets* no bloqueantes del lado del *kernel*, mientras que las llamadas a funciones que bloquean inherentemente el sistema (como I/O en archivos) corren de manera bloqueante en su propio hilo. Cuando un hilo en el pool de hilos completa una tarea, le informa al hilo principal de esto, lo cual despierta y ejecuta el *callback* registrado. Como los *callbacks* son manejados en serie en el hilo principal, las computaciones de mucha vida y otros procesos asociados al *CPU* congelaran el ciclo de eventos hasta que se completen.

V8: V8 es el motor de ejecución de *JavaScript* construido para *Google Chrome*. Hecho sobre C++, V8 compila de *JavaScript* a código maquina nativo en lugar de interpretarlo en tiempo real. [36]

Las funcionalidades base de *Node.js* residen en una biblioteca *JavaScript*. Los enlazados, escritos en C++, conectan estas tecnologías la una a la otra y al sistema operativo.

Manejo de paquetes: *Npm* (*Node Package Manager*) es el manejador de

paquetes preinstalado de la plataforma de Node.js. Es usado para instalar dependencias y programas de *Node.js* desde un registro de *npm*, esto facilita la organización y manejo de programas externos. Los programas en el registro pueden ir desde simples bibliotecas de soporte (como *Underscore.js*) hasta manejadores de tareas como *Grunt* o *Gulp*.

Ciclo de Eventos: Node.js se registra a si mismo con el sistema operativo para que pueda ser notificado cuando una conexión es hecha, así el sistema operativo puede enviar un *callback*. En tiempo de ejecución de *Node.js*, cada conexión es un pequeño *heap allocation*. Tradicionalmente, los hilos o los procesos relativamente pesados del sistema operativo manejaban cada conexión. *Node.js* usa un ciclo de eventos para mejorar la escalabilidad, en vez de los procesos o los hilos. En contraste con los otros servidores orientados a eventos, el ciclo de vida de *Node.js* no necesita ser definido explícitamente. En cambio, se definen los *callbacks*, y el servidor automáticamente entra al ciclo de evento al final de la definición de un *callback*. *Node.js* sale del ciclo de evento cuando no hay más *callbacks* que realizar. [39]

2.4.4 Express.js

Express.js es el *framework* de servidor de aplicaciones web estándar *de facto* de Node.js, está diseñado para construir aplicaciones web híbridas, de single-page o multi-page. Sirve como una aplicación ligera para la organización de la arquitectura del lado del servidor en un modelo MVC.

2.5. Control de Asistencias

El control de asistencia se refiere a la gestión de la asistencia de usuarios a un área determinada. Los controles de asistencia tienen mucha aplicación en entornos laborales, centros de formación, entre otros. Estos sistemas tienen un registro de usuarios, los cuales están asociados a un horario y un lugar específico.

2.5.1. Dispositivos de Control de Asistencia

En el pasado, era posible manejar un sistema de horarios y asistencia, usando únicamente unas cuantas hojas de papel. Sin embargo, esta práctica puede dar resultados catastróficos debido a la fragilidad del medio y los problemas de seguridad inherentes. Hoy en día, existen sistemas sofisticados, que incorporan estrategias y dispositivos para la captura de datos de forma rápida, organizada y segura.

Un sistema de asistencia manual que utiliza papel requiere que las personas coloquen en la hoja datos personales que las identifiquen unívocamente como ellas mismas, tales como su firma, su huella dactilar, su número de cedula entre otros. Estas hojas requieren de muchas personas para poder realizar reportes basándose en la información recopilada en las mismas, lo cual es un proceso lento y repetitivo que conlleva a un enorme gasto de tiempo al día, a la semana, al mes. Además, dichos reportes pueden presentar inconsistencias debido a varias razones: por un lado, el sistema en sí se presta para eso, mientras que por otro lado, no se sabe en realidad si el sistema es usado con honestidad. Con mucha frecuencia, las personas tienden a colocarse más horas laboradas, tapar retrasos y hasta a firmar por otros. Aunque las personas que realizan estas prácticas no las ven como algo grave, lo cierto es que éstas ocasionan pérdidas alarmantes en una empresa o institución.

2.5.2. Sistema de Marcado de Tarjetas

El sistema de control de asistencia de marcado de tarjetas, funciona asignando una tarjeta única a cada uno de los usuarios del sistema, dicha tarjeta debe ser marcada con un dispositivo, el cual marca el momento en el que la tarjeta fue introducida. Este sistema presenta problemas similares al control de asistencia usando papel, dado que la tarjeta puede ser pasada a otro usuario y este hacer uso de la misma en el dispositivo, así chequeando a otra persona. Esto podría ser evitado utilizando una clave adicional que debe ser ingresada en el sistema a la hora de marcar la entrada y/o la salida, pero dicha clave también puede ser compartida. Adicional a esto, el sistema solo controla las entradas y las salidas del individuo, sin embargo no asegura que el dueño de la tarjeta permanezca en su lugar de trabajo por la duración de su jornada, lo que no soluciona el problema del absentismo laboral.

2.5.3. Sistemas de Escaneo Biométrico

Los sistemas modernos, no sólo son mucho más precisos que sus antecesores, sino también reflejan las horas que han sido trabajadas en realidad y permiten acceder a esta información de manera más rápida y organizada. De esta forma, el arduo papeleo y la lectura de datos manuales pasan a la historia, porque los sistemas modernos cuentan con poderosos programas, para el procesamiento de datos. Estos sistemas hacen uso de la biometría, la cual es una práctica que identifica a las personas, al analizar sus rasgos particulares. Ejemplo de ello son los relojes biométricos de huellas digitales, que solicitan la huella de cada empleado cuando éste ingresa o sale de su trabajo.

También existen los terminales de escaneo de retinas y escaneo de iris, el escáner de retina mide el patrón de venas en el fondo del ojo que se obtiene proyectando una luz infrarroja a través de la pupila. El escáner de iris se realiza utilizando una cámara de video y examina patrones de color únicos en los surcos de la parte coloreada de los ojos. Estos sistemas aun cuando son considerados muy seguros, no funcionan todo el tiempo. Por ejemplo los escáneres de rutina no funcionan en personas ciegas o con cataratas, mientras que la precisión del escáner de iris varía en función de la luz ambiente y del ángulo en que se coloque la cabeza.

Sin embargo, al basarse en estas particularidades es imposible que un trabajador reporte la asistencia de otro, porque no hay manera de prestarse las huellas digitales o los ojos. Además, con estos sistemas biométricos se reducen pérdidas y se mejoran los márgenes de ganancia de una empresa. En un entorno laboral sirven para contabilizar el saldo de horas trabajadas por cada empleado, para llevar un control de la puntualidad y asistencia de los trabajadores, para gestionar de manera automática las incidencias tanto de entradas como salidas especiales del puesto de trabajo o para contabilizar el tiempo de utilización de recursos en entornos compartidos. En centros de formación como escuelas primarias pueden ser utilizados para detectar faltas injustificadas, absentismo o retardos y poder actuar en consecuencia de manera rápida. En escuelas de negocio o universidades de asistencia obligatoria permiten registrar de manera fácil si el alumno ha asistido o no a clase.

Habitualmente estos sistemas tienen un funcionamiento off-line y guardan un registro en memoria del histórico de accesos (con información del usuario, hora y día del acceso) y otros eventos como alarmas u otras incidencias. Este registro puede ser descargado a un PC para su posterior tratamiento.

La autenticación de cada usuario puede realizarse mediante tecnología biométrica, proximidad, tarjetas inteligentes, bandas magnéticas. Estos incluyen uno o más lectores que permiten la autenticación de la persona a partir de una tarjeta de proximidad, chip o banda magnética. En un control de presencia biométrico las personas se identifican mediante la huella digital u otro rasgo biológico.

3. Metodología, Herramientas y Diseño

En este capítulo se detalla la metodología utilizada para planificar y controlar el proceso de elaboración de la aplicación. Además, Se mostrarán y explicarán las diferentes herramientas vinculadas al desarrollo e implementación (tanto en el cliente como en el servidor) de los distintos componentes del sistema realizado.

3.1 Metodología

Para estructurar, planear y controlar el proceso de desarrollo de la aplicación fue utilizada una metodología de orientada a prototipos [32]. Esta se caracteriza por presentar un método de desarrollo iterativo, en el cual se trabaja de cerca con el usuario, diseñando un sistema funcional primario que se aproxime al deseado. Dicho sistema es mostrado al usuario y seguidamente se continúa la elaboración del prototipo basado en las opiniones y recomendaciones dadas por el usuario. La iteración continúa hasta que el sistema se encuentra suficientemente completo tanto para el desarrollador como para el cliente. En esta etapa se afinan los

pequeños detalles que hayan quedado pendientes y se entrega el sistema como un producto final [33].

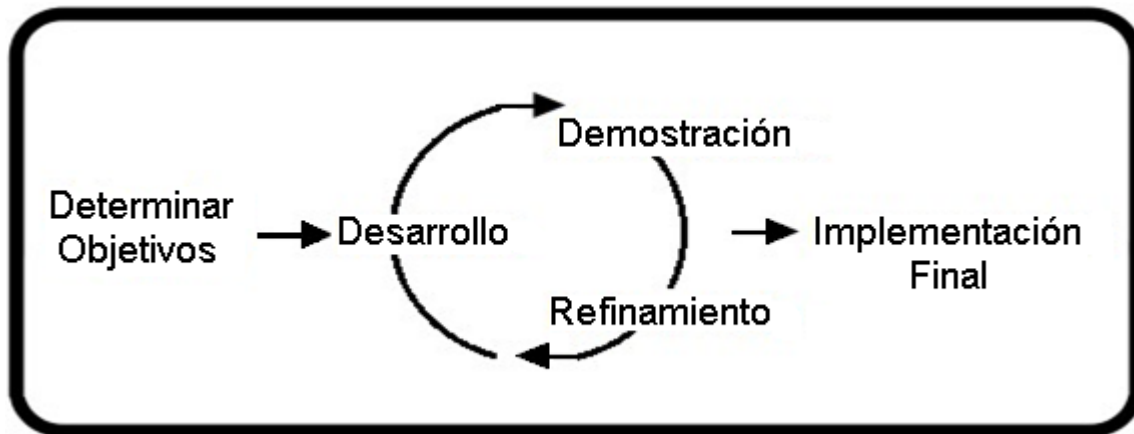


Figura 4: Metodología orientada a prototipos.

La metodología basada en prototipos permite capturar cada uno de los objetivos que plantea el proyecto de forma concreta, diseñando, desarrollando y corrigiendo cada una de sus fases de forma puntual, obteniendo además una retroalimentación continua con el cliente o usuario de la aplicación.

Dicha metodología, fue integrada con un paradigma de programación Orientado a Objetos, con el cual, haciendo uso de conceptos como clases, herencia, objetos, métodos y atributos, entre otros, se alcanza una mayor organización, estructuración y reutilización del código.

3.2 Herramientas

Un conjunto de programas, bibliotecas y entornos de desarrollo fueron integrados y utilizados tanto para la programación del proyecto como para sus pruebas. A continuación se muestran las principales herramientas utilizadas:

- *Android Studio* Versión 1.5. Utilizado para el desarrollo de la aplicación móvil.
- *Java Development Kit* Versión 7. Kit de desarrollo usado por Android Studio.
- Editor de Texto *Sublime* Versión 3.0. Utilizado para la codificación del código de JavaScript.
- *Node.js* Versión 5.3. Entorno de desarrollo utilizado para levantar el servidor

de la aplicación.

- Express.js Versión 4.13.4. Middleware utilizado para la conexión entre el servidor y la vista de la aplicación.
- AngularJS Versión 1.4. Framework de JavaScript utilizado para la creación de las interfaces de usuario.
- *MongoDB* Versión 3.2. Manejador de la base de datos no relacional.
- Computadora personal, procesador núcleo i7, 8GB de memoria RAM, Disco Duro de 1TB y Sistema Operativo *Windows 7* de 64bits
- Teléfono móvil V-Telca Telepatria V865M, *Chipset*: MT6577 (*Dual core* 1GHz), Memoria RAM: 512MB, Memoria interna: 512MB y Sistema Operativo *Android 4.1.2 Jelly Bean*.
- *GIT* como controlador de versiones.
- *Bitbucket* para el almacenamiento y manejo de repositorios remotos.

3.3 Diseño e Implementación

En esta sección se explica el proceso de desarrollo y diseño de la aplicación y de cada uno de los componentes que la conforman, tanto del lado del cliente como del servidor. Se expondrán los requerimientos, se analizarán los diagramas más importantes, los detalles relevantes y además se explica la implementación de la solución, haciendo énfasis en las clases y métodos más significativos.

3.3.1 Requerimientos del Sistema

El requerimiento general del sistema plantea la creación de una aplicación para teléfonos inteligentes, enfocada a escanear dispositivos móviles con capacidades *Bluetooth*, además de la construcción de una aplicación *Web*, la cual permita administrar las materias, secciones y alumnos junto a un módulo de reportes que permita ver los porcentajes de asistencia en diferentes niveles. A partir de esto se obtienen los requerimientos funcionales y no funcionales del sistema.

Requerimientos funcionales

- El proyecto constará de una aplicación móvil y una aplicación web.
- La aplicación web contará con módulos para el manejo de materias, secciones y alumnos que estén ligadas al profesor que utiliza la aplicación.
- El servidor recibirá los datos de la clase enviados por la aplicación móvil y los almacenará en la base de datos.
- La aplicación web contará con un módulo de reportes el cual permitirá al usuario ver estadísticas de asistencia por materias, secciones o alumnos.

- La aplicación web contara con un módulo de administración que permitirá la creación de credenciales de acceso para los profesores.
- La aplicación móvil contará con la capacidad de poder escanear los estudiantes presentes de un aula de clase de una sección específica y enviarlos al servidor una vez finalizada la clase.
- La aplicación web contara con autenticación lo que permitirá el filtrado de las funcionalidades por roles y credenciales.
- La aplicación móvil contara con autenticación lo que permitirá a un profesor ver y escanear únicamente estudiantes que se encuentren en secciones que le pertenezcan.
- La aplicación web contara con el uso del protocolo HTTPS para garantizar un nivel mínimo de seguridad y confidencialidad en la autenticación.

Requerimientos no funcionales

- El dispositivo donde se desplegara la aplicación móvil debe contar con el sistema operativo Android.
- La versión mínima de Android necesaria para el despliegue de la aplicación es la 2.2 (Froyo, *API* nivel 8). Se utiliza el *API* nivel 8 para asegurar que la solución sea compatible con la mayor cantidad de dispositivos.
- El dispositivo donde se desplegara la aplicación móvil de contar con capacidad Bluetooth con la versión 2.1 como mínimo. Se desconoce cuál es el funcionamiento con versiones anteriores debido a que las pruebas se hicieron en dispositivos usando esta versión.

3.3.2 Diseño de la Solución

En base a los requerimientos mencionados en la sección 3.3.1 Requerimientos del Sistema, se considera como punto de inicio de la aplicación el escaneo y captura de datos.

El segundo punto a desarrollar es usabilidad de la aplicación, la cual debe ser sencilla e intuitiva debido a que podría ser usada por personas poco o nada familiarizadas con la manipulación de teléfonos inteligentes. Para esto se plantea un alto nivel de automatización del sistema, ofreciendo al usuario solo la posibilidad de iniciar y detener el programa, dejando el almacenamiento, envío y vaciado de los datos en manos de la aplicación, disminuyendo al máximo la posibilidad de errores por parte del factor humano.

Se tomó en cuenta el concepto de diseñar la aplicación como una solución de bajo costo, por lo tanto las funcionalidades de la aplicación deben ser capaces de ejecutarse en teléfonos inteligentes de gama baja. Por ello se seleccionó el *API* de aplicación más bajo posible (*API* 8, *Android* versión 2.2) de manera de tener la

mayor cantidad posible de dispositivos compatibles con la aplicación (100% de los dispositivos activos en el *Google Store*). [34]

Adicional a la aplicación móvil, el sistema debe ofrecer un servicio que permita recibir y almacenar todos los datos obtenidos. Por tal motivo se plantea el uso de un programa del lado del servidor.

Se decidió utilizar una aplicación web para la visualización y acceso a los datos almacenados en el servidor, como la asistencia por materia, por sección y por alumno.

La integración de todos los componentes mencionados anteriormente (la aplicación móvil y la aplicación web), unida a los diferentes métodos para la obtención, visualización y descarga de los datos, son los que conforman el sistema en el cual se centra este Trabajo Especial de Grado.

En la Figura 11 se pueden apreciar los elementos involucrados en todos los escenarios y la manera en la que interactúan.

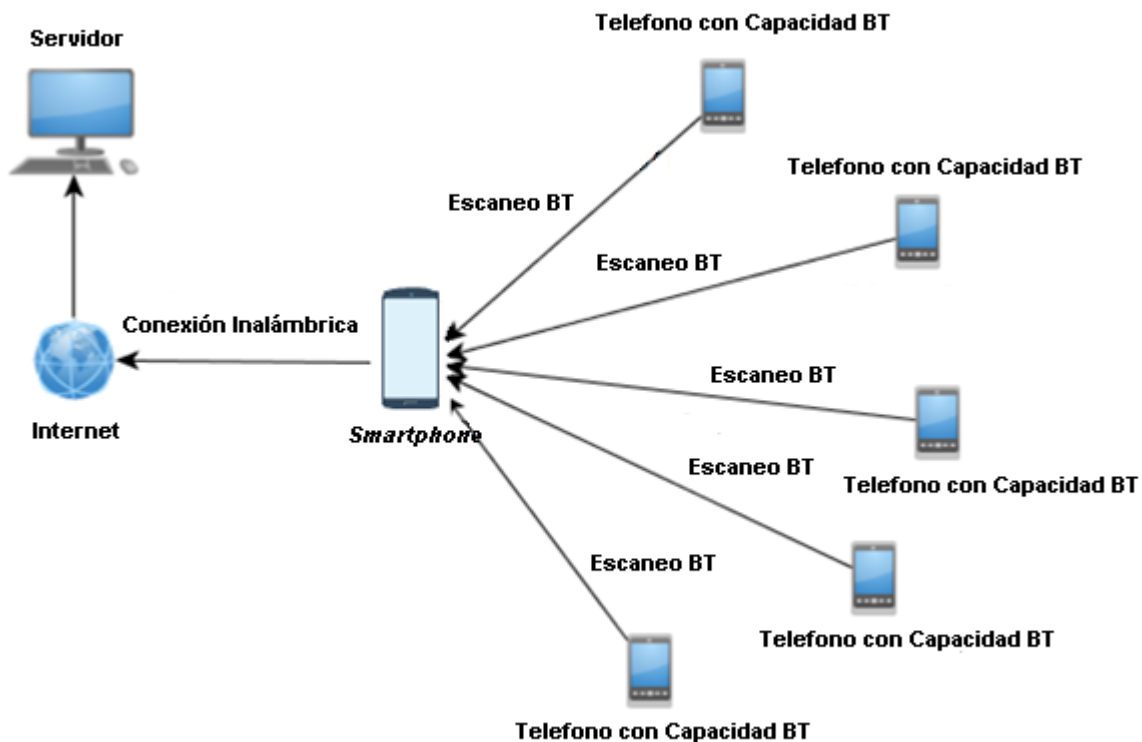


Figura 5: Elementos de los escenarios.

1. **Servidor:** Está ubicado en cualquier parte de la Internet. Recibe la información del dispositivo inteligente del profesor.
2. **Smartphone:** Este es el dispositivo con el cual el profesor escaneara los teléfonos de los alumnos y enviará la información recopilada a través de Internet.
3. **Teléfonos con Capacidad BT:** Estos teléfonos no deben ser necesariamente dispositivos inteligentes, cualquier dispositivo con capacidad de conectividad Bluetooth puede ser usado. Se refiere como teléfonos considerando que los alumnos poseen teléfonos móviles personales que cumplen estas condiciones. Estos son de cantidad variable dependiendo de los alumnos presentes.

3.4 Diagrama de Componentes

El sistema tiene dos roles definidos, el rol de usuario administrador, el cual se encarga de toda la gestión de profesores y sus credenciales. Y el rol de usuario profesor, el cual se encarga de gestionar sus materias y la información del chequeo de asistencia de sus secciones.

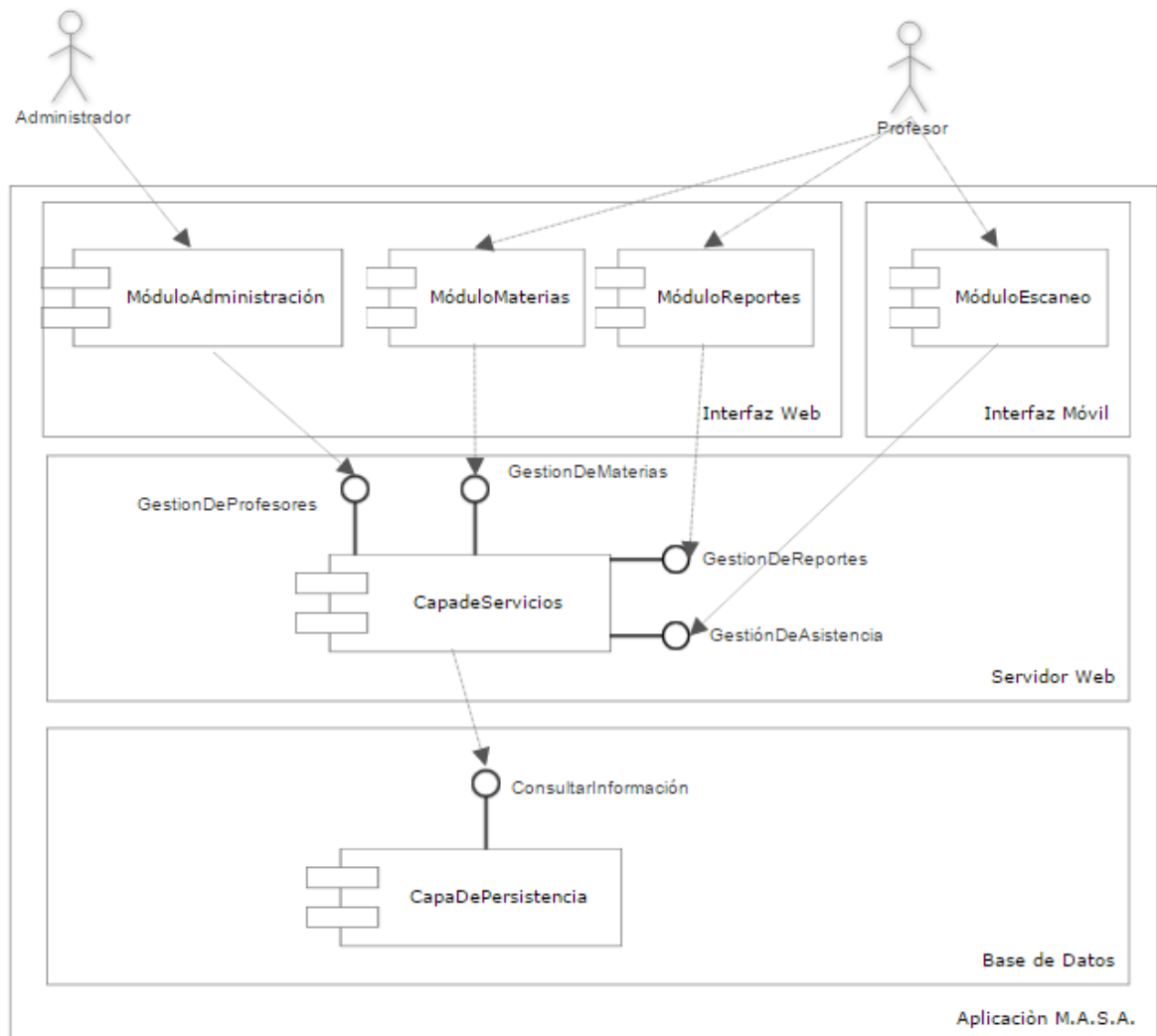


Figura 6: Diagrama de Componentes.

3.5 Diagrama de Casos de Uso

Una vez descrito el diseño de la solución, se procede al desarrollo del diagrama de casos de uso, el cual sirve para mostrar las funciones de un sistema desde el punto de vista de sus interacciones con el exterior sin entrar en la descripción

detallada o en la implementación de sus componentes [11].

3.5.1 Módulo de Gestión de Materias

Este módulo permite las acciones de creación, listado, modificación y borrado de las materias por cada profesor que se autentica con sus credenciales en el sistema.

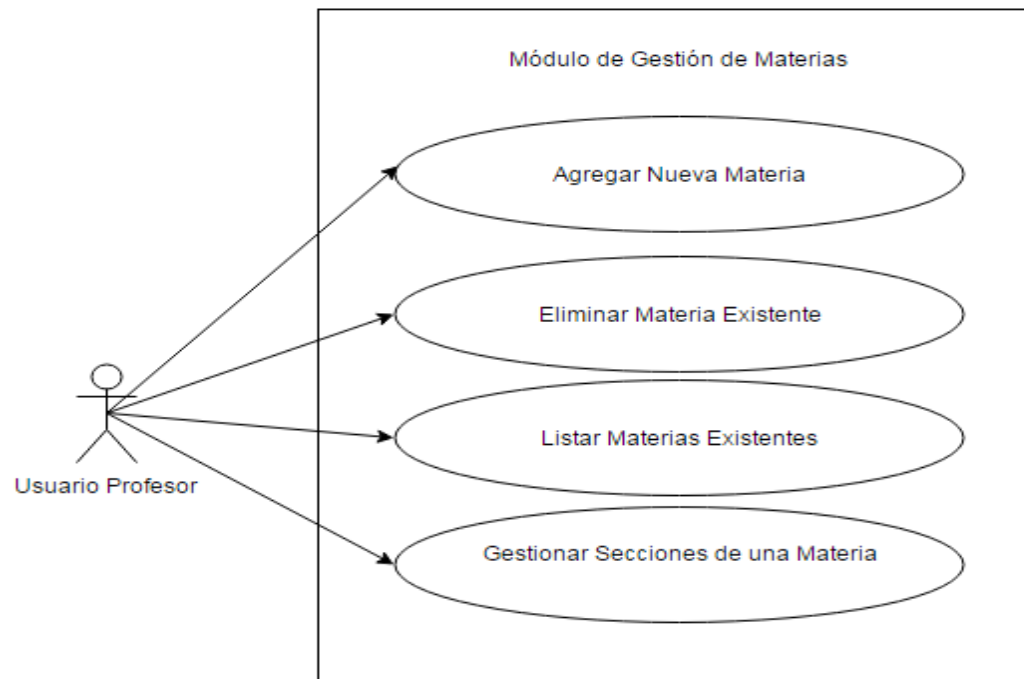


Figura 7: Casos de uso Nivel 1, Módulo de Gestión de Materias.

3.5.2 Módulo de Gestión de Secciones

Este módulo permite las acciones de creación, listado, modificación y borrado de las secciones por cada materia que el profesor autenticado con sus credenciales en el sistema tiene guardadas. Al momento de crear una nueva sección es necesario contar con el archivo .XLS del listado de estudiantes provisto por el Departamento de Control de Estudios. Dicho archivo es cargado al sistema para determinar el nombre de la sección, su semestre y sus estudiantes de manera automática. El nombre y código de la materia son definidos por el profesor al momento de crear la materia en el módulo de gestión de Materias.

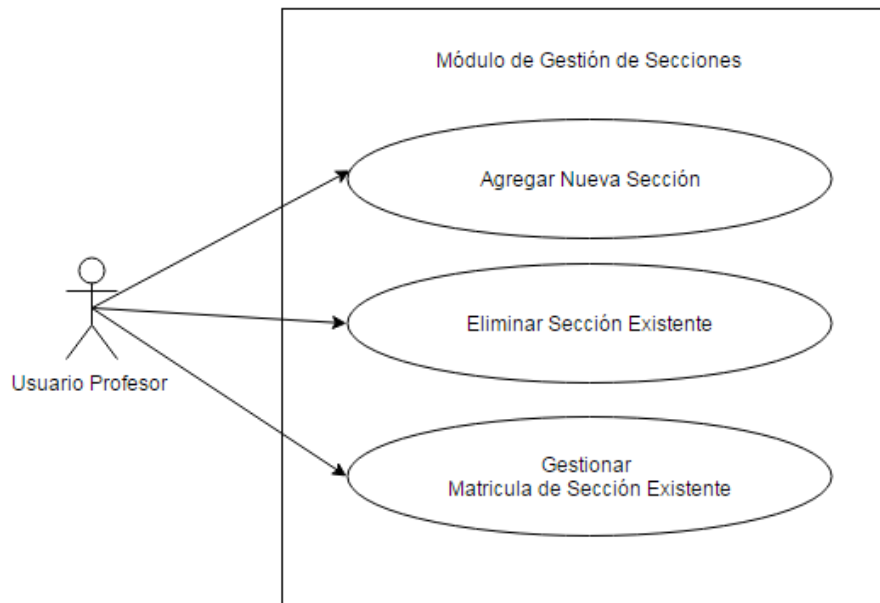


Figura 8: Casos de uso Nivel 1, Módulo de Gestión de Secciones.

3.5.3 Módulo de Gestión de Matriculas

Este módulo permite las acciones de agregar y retirar estudiantes de la matrícula de una sección. Adicional a esto, este módulo permite actualizar la dirección de correo de un estudiante y la capacidad de borrar la dirección MAC de un dispositivo de un estudiante para poder realizar el pareo entre estudiante y dispositivo nuevamente. Esto es especialmente útil en caso de que el estudiante cambie de dispositivo.

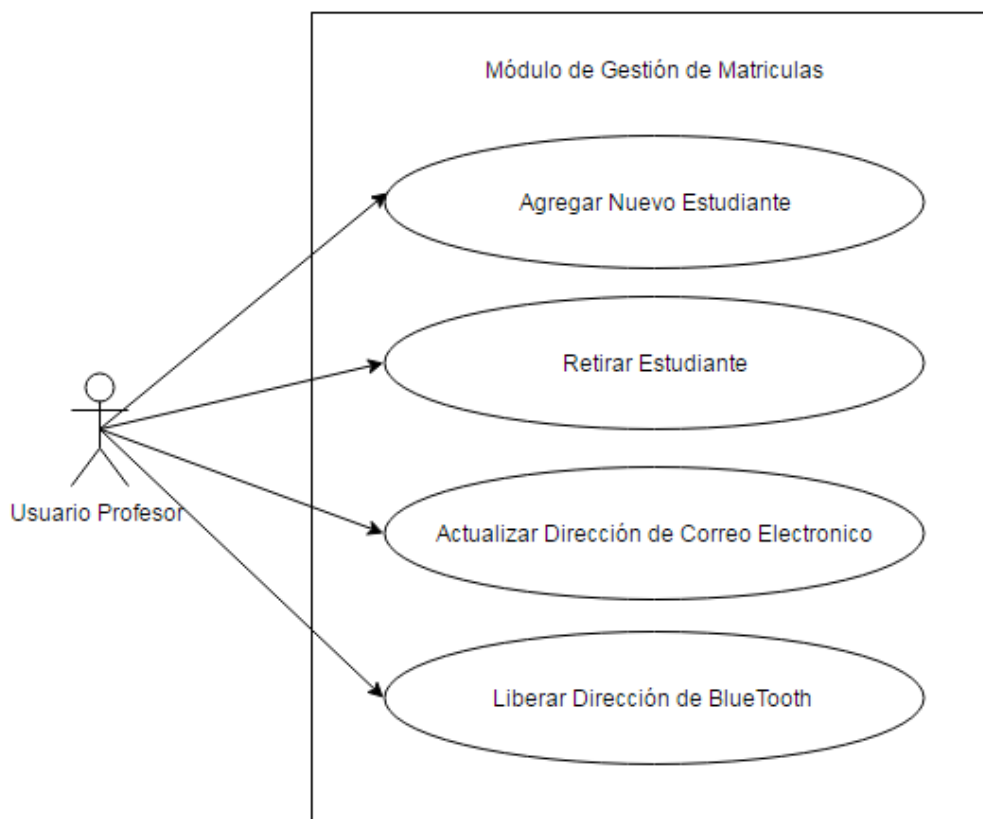


Figura 9: Casos de Uso Nivel 1: Módulo de Gestión de Matriculas

3.5.4 Módulo de Escaneo de Datos

Este módulo permite las acciones de escanear dispositivos y culminar la clase, estas acciones se realizan concretamente sobre la aplicación móvil, y es de notar que es una pre-condición para poder culminar la clase que al menos se haya realizado el escaneo de dispositivos al menos una vez.

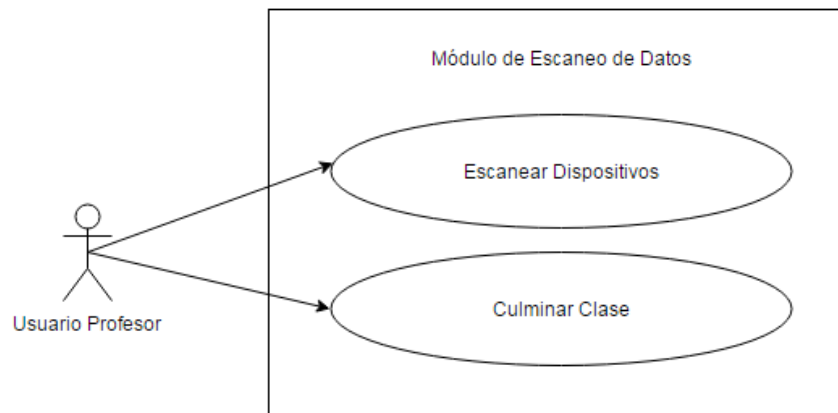


Figura 10: Casos de Uso Nivel 1: Módulo de Escaneo de Datos.

3.5.5 Módulo de Administración de Aplicación Web

Este módulo permite las gestionar a los profesores, estas funciones solo pueden ser realizadas por el administrador del sistema. Es de notar que el eliminado de la información de un profesor también incluye todos los datos que ese profesor tuviese como materias, secciones, alumnos y asistencias.

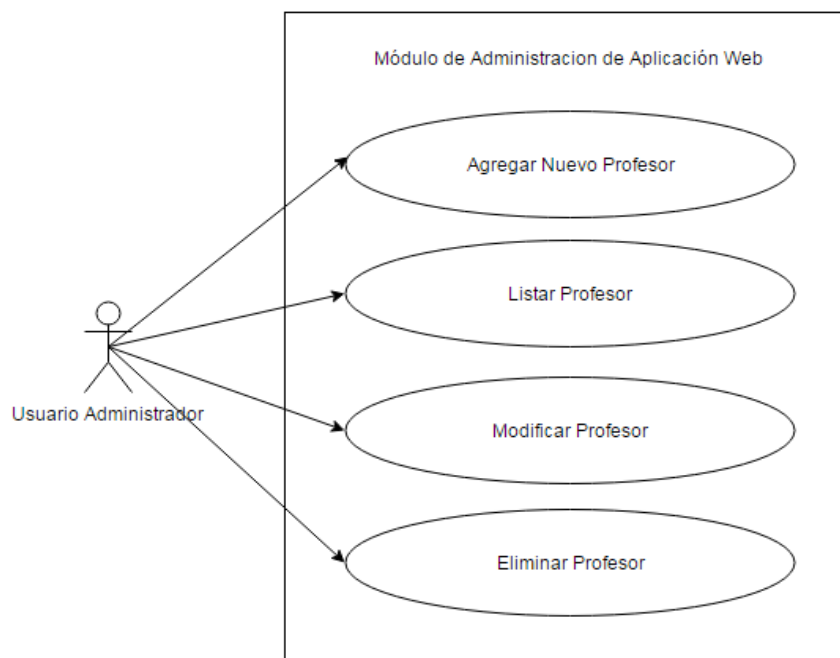


Figura 11: Casos de Uso Nivel 1, Módulo de Administración de la Aplicación Web

4. Resultados y Análisis de los Resultados

4.1 Escenarios

En este capítulo se describe el diseño de las pruebas realizadas, comenzando por escoger los escenarios en los que se realizaron las pruebas de desempeño, además de los pasos a seguir para la implementación de dichos escenarios y la selección y ejecución de las pruebas necesarias para la obtención de resultados y posterior análisis.

4.2 Descripción del Escenario

El escenario escogido para estas pruebas fue creado en un ambiente controlado en el laboratorio ICARO de la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela.

Se utilizaron una variedad de dispositivos diferentes para probar la conectividad del teléfono que realiza el escaneo con los dispositivos escaneados. Entre los dispositivos se encuentran:

- 2 *Tablets* con el Sistema Operativo *Android*
- 2 Teléfonos con capacidad de Conectividad *Bluetooth*
- 3 Teléfonos con el Sistema Operativo *Android*
- 1 Laptop con conectividad *Bluetooth*.

4.3 Implementación de los escenarios

Se realizaron varias rondas de pruebas de escaneos de los dispositivos con diferentes profesores invitados de manera de poder identificar problemas de usabilidad.

4.3.1 Configuración

A continuación se describen todos los pasos para la configuración.

Servidor

Para poder realizar el despliegue de la aplicación en una computadora son necesarios que se encuentren instalados *MongoDB* (usado como manejador de base de datos), *Node.js* (usado como plataforma web), *npm* (usado para el manejo de dependencias de *Node.js*) y *git* (usado para el manejo de versiones). Adicional a esto, es necesario instalar *Bower* (usado para el manejo de dependencias de las vistas).

En primera instancia, se debe ejecutar el manejador de base de datos no relacional MongoDB, para crear las conexiones necesarias. Una vez este corriendo, se ejecutan los comandos “*npm install*” y “*bower install*” en la ventana de comandos del sistema operativo dentro de la carpeta de proyecto para instalar todas las dependencias necesarias para el despliegue de la aplicación. Finalizadas ambas instalaciones el archivo “*server.js*” el cual se encuentra en la carpeta *app* se ejecuta utilizando el comando “*node server.js*”. Una vez ejecutado aparecerá el siguiente mensaje en la ventana de comandos “*listening in port 3000*”. En este punto la aplicación Web es alcanzable desde la computadora que lo ejecuta desde cualquier navegador Web moderno a través de la “<dirección IP de la maquina>:3000”.

Teléfono del profesor

En la aplicación móvil debe de ser configurada en el apartado de “preferencias” la dirección IP de la computadora donde está corriendo el servidor Web.

Teléfonos de los alumnos

Aun cuando puede variar de un modelo a otro, los pasos generales serían los siguientes:

- Dirigirse al menú principal Sistema > Preferencias > Conexiones de Red.
- Elegir la pestaña “Inalámbrica” y entrar “Configuración Bluetooth”.
- Activar la conectividad Bluetooth.
- Permitir el descubrimiento Bluetooth del dispositivo.
- Presionar el botón “Aplicar”.

Es importante notar que para los primeros escaneos realizados sobre los teléfonos, los alumnos deben cambiar el nombre de su dispositivo Bluetooth a su número de cedula, de manera que su cedula pueda ser emparejada con su *Bluetooth Address*. Una vez hecho esto no es necesario que el dispositivo tenga la cedula como nombre ya que se utilizara su *Bluetooth Address* como identificador.

4.3.2 Ejecución de las pruebas

Se hicieron diferentes rondas de pruebas junto con profesores invitados de la escuela de computación. En un principio, cada profesor creó una materia nueva y secciones nuevas usando un listado Excel de estudiantes modificado, el cual contenía estudiantes con números de cedula del 1 al 15, de manera de facilitar el nombrado de los dispositivos de prueba. Una vez creadas las secciones se procedió a activar el modo de descubrimiento de los dispositivos de prueba y a escanear con el teléfono del profesor los dispositivos de la sección creada.

Al finalizar el escaneo, se pudieron apreciar automáticamente los dispositivos escaneados en el módulo de reportes. Adicional a esto, las direcciones MAC de los dispositivos escaneados fueron guardadas en la base de datos, por lo que ya no sería necesario en futuros escaneos la necesidad de nombrar los dispositivos con la cedula de identidad del alumno.

4.3.3 Resultados de las pruebas

Para determinar la usabilidad de la aplicación se realizó una encuesta basada en el principio de las heurísticas de Nielsen. Esta encuesta se separó en 10 preguntas clave para determinar posibles problemas en las interfaces de usuario. A continuación podemos ver las preguntas y los resultados generales a las mismas.

Pregunta 1. Visibilidad del estado del sistema

¿Considera que el sistema le informa de su estado actual de manera clara todo el tiempo a través de los elementos de la aplicación? Por ejemplo, ¿La navegabilidad de la página es fácil debido a que siempre sabe en donde está en todo momento y esto facilita la creación de un mapa mental del sitio?

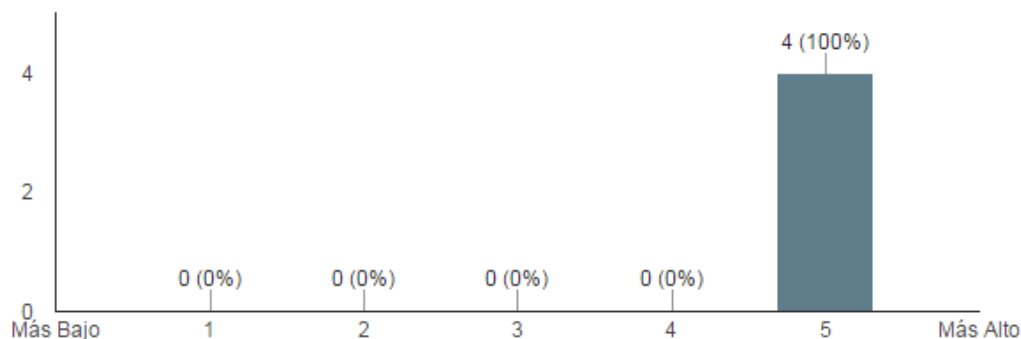


Figura 12: Visibilidad del estado del sistema.

Como se puede apreciar, los encuestados estuvieron conformes con la visibilidad del estado del sistema, por lo que este aspecto no tuvo modificaciones luego de la ronda de pruebas.

Pregunta 2. Lenguaje del Usuario

¿Considera que el sistema utiliza un lenguaje común a través de frases, terminología y conceptos, que haga su utilización acorde con su realidad?

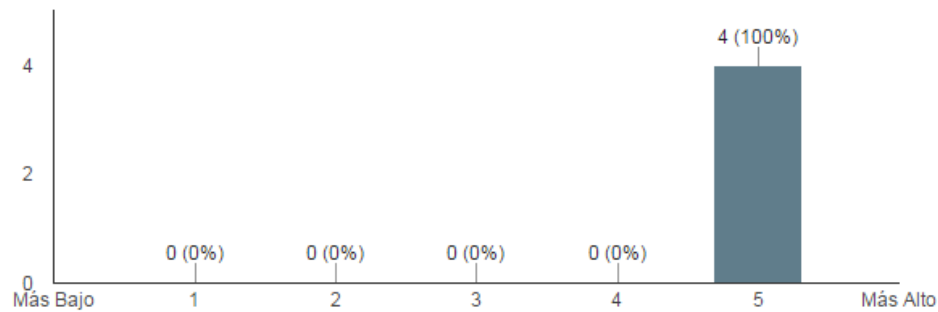


Figura 13: Lenguaje del Usuario

Aquí se puede ver que los encuestados estuvieron de acuerdo el aspecto del lenguaje del usuario usado en el sistema, por lo tanto, esto no genero revisiones en las aplicaciones.

Pregunta 3. Control y Libertad de Uso

¿Considera que el sistema provee salidas claras y demarcadas en las funcionalidades de la aplicación en caso de que el usuario escoja alguna por error?

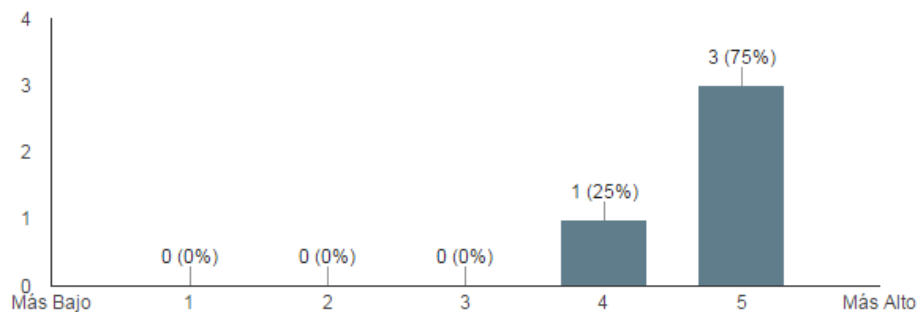


Figura 14: Control y Libertad de Uso.

Se tomaron en consideración las respuestas de los encuestados, y esto se tradujo en una mejora para la facilidad del diagnóstico, reconocimiento y recuperación de errores, a través de mensajes más claros a la hora de presentarse un error, al igual que diálogos de confirmación claros a la hora de hacer cambios sobre el sistema, por ejemplo, a la hora de agregar una sección o al retirar un estudiante.

Pregunta 4. Consistencia y Estándares

¿Considera que el lenguaje, situaciones o acciones son consistentes y siguen un estándar dentro de su función en el sistema? Por ejemplo, ¿Los iconos y términos utilizados para la modificación de items en un menú se mantienen homogéneos a través de la aplicación?

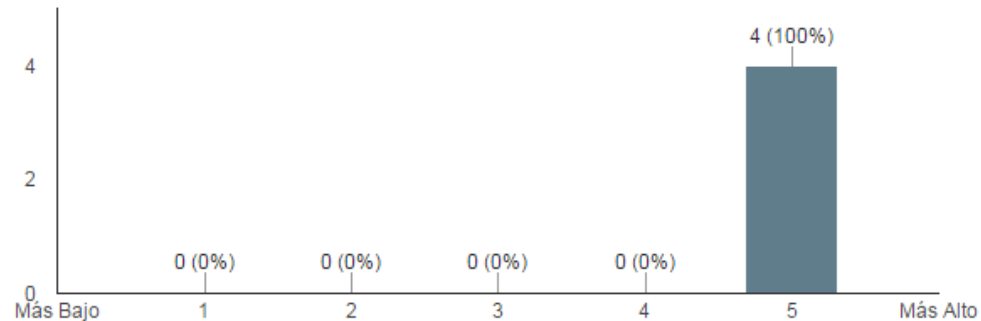
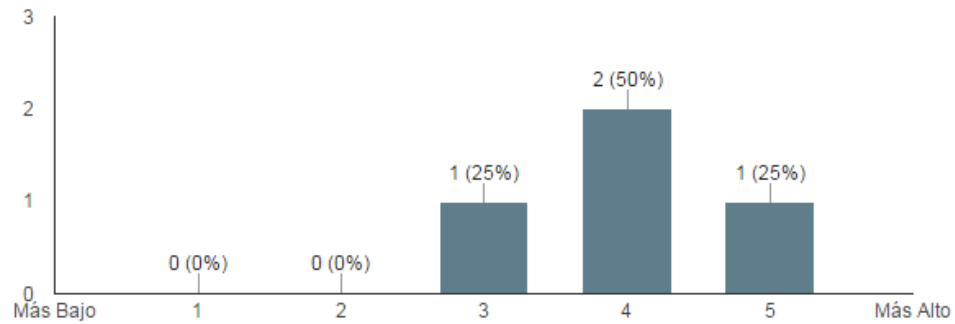


Figura 15: Consistencia y Estándares.

Los encuestados evaluaron la consistencia y estándares del sistema, y no se consiguieron observaciones en la versión actual del sistema.

Pregunta 5. Prevención de Errores

¿Considera que el sistema a través de diseño o diálogos de confirmación previene errores o minimiza la ocurrencia de los mismos?



Explique brevemente el razonamiento de su respuesta (2 responses)

Algunos errores presentes con el usuario en esta versión

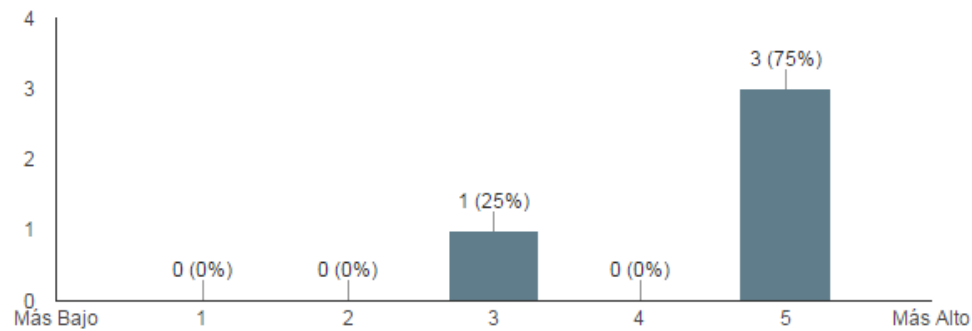
No del todo, sería importante colocar una pequeña leyenda al posicionar el mouse sobre las opciones a elegir, indicándole al usuario que realizara

Figura 16: Prevención de Errores.

Aquí se puede apreciar que la opinión de los encuestados vario considerablemente, esto se tradujo en una mejora para la prevención de errores utilizando *tooltips* en cada uno de los botones de las funciones del sistema, de manera que en cualquier momento el usuario pueda conocer cuál es el efecto en el sistema de las acciones posibles.

Pregunta 6. Reconocimiento

¿Considera que el sistema muestra sus opciones, objetos o acciones de una forma explícita de manera que el usuario no debe recordar la información necesaria para interactuar con el sistema sino que se basa en el reconocimiento del mismo?



Explique brevemente el razonamiento de su respuesta (2 responses)

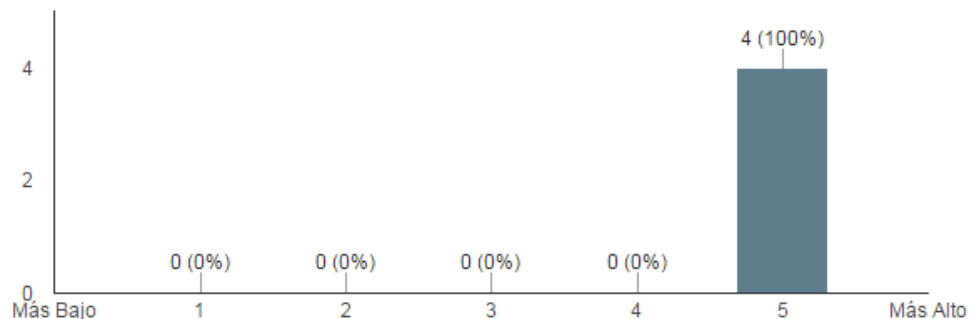
Algunos detalles en la aplicación de celular. Falta alguna ayuda o mayor claridad.

Figura 17: Reconocimiento.

Se tomaron en consideración las observaciones sobre el reconocimiento en el sistema, al utilizar textos de ayuda e iconos en los botones en la aplicación móvil se mejoró la navegación al hacerla más intuitiva.

Pregunta 7. Flexibilidad y Eficiencia de Uso

¿Considera usted que el diseño de la aplicación permite que las funciones que realiza el usuario puedan ser más rápidas o más eficientes a través del uso de accesos directos o funciones que mejoren la experiencia del usuario?



Explique brevemente el razonamiento de su respuesta (1 response)

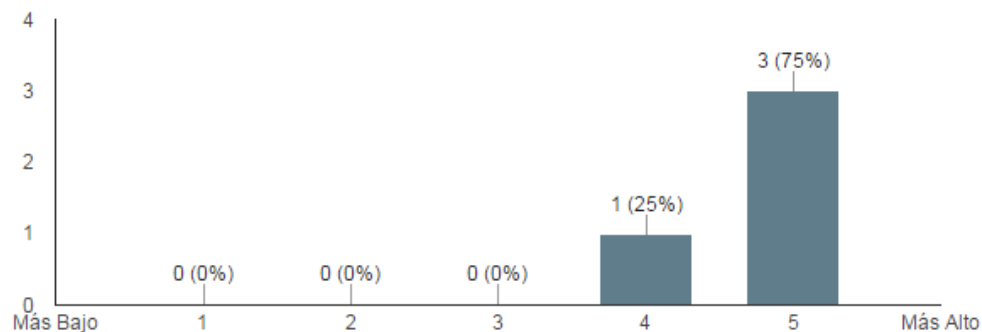
La navegación dentro de la aplicación es bastante rápida y fácil de utilizar

Figura 18: Flexibilidad y Eficiencia de Uso.

Los encuestados no presentaron observaciones en el aspecto de flexibilidad y eficiencia de uso del sistema.

Pregunta 8. Diseño Minimalista y Estético

¿Considera que el diseño del sistema contiene solo información necesaria, es decir que no incurre en redundancias o con datos que compitan con la información vital, para manejar la aplicación exitosamente?



Explique brevemente el razonamiento de su respuesta (2 responses)

En la aplicación de celular faltaría un poco de estética

Bien diseñado

Figura 19: Diseño Minimalista y Estético.

Se tomaron en consideración las respuestas de los encuestados, y esto se tradujo en una mejora para el diseño de la aplicación móvil, a través del uso de estilos e iconos.

Pregunta 9. Facilidad para el Diagnostico, Reconocimiento y Recuperación de Errores

¿Considera usted que los mensajes de error presentados por el sistema son expresados en lenguaje simple, indican la fuente del problema y sugieren de manera constructiva una solución?

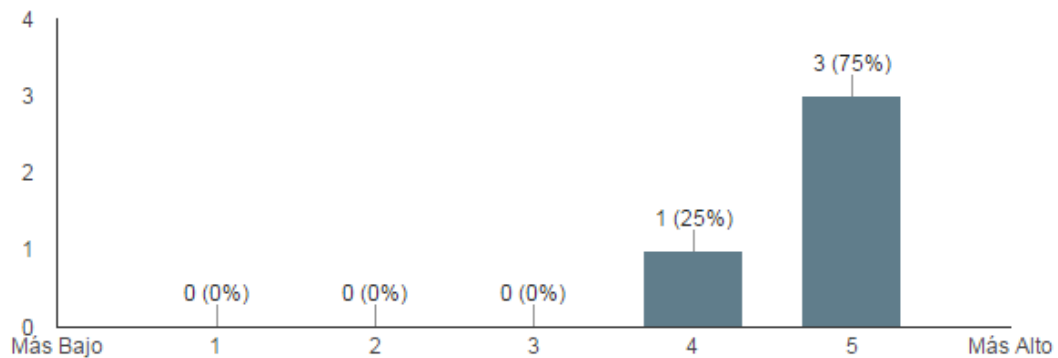
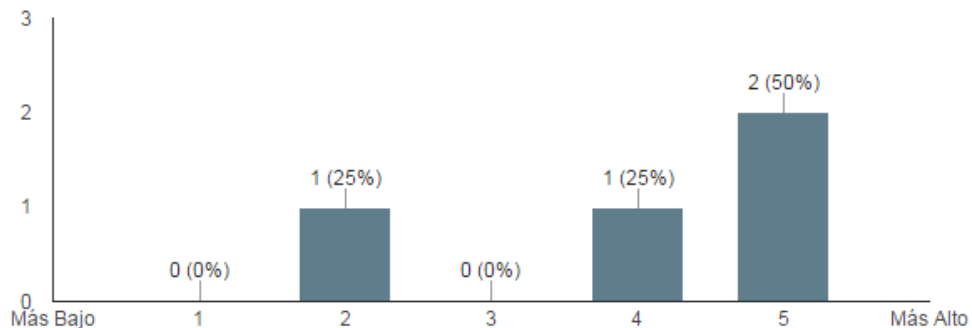


Figura 20: Facilidad para el diagnóstico, reconocimiento y recuperación de errores.

Los encuestados al evaluar la facilidad del diagnóstico, reconocimiento y recuperación de errores tuvieron observaciones, esto se tradujo en mensajes más claros a la hora de presentarse un error, al igual que diálogos de confirmación claros a la hora de hacer cambios sobre el sistema, por ejemplo, a la hora de agregar una sección o al retirar un estudiante.

Pregunta 10. Ayuda y Documentación

¿Considera que el sistema presenta ayuda y documentación necesaria, suficiente, concisa y de fácil entendimiento para la correcta utilización de las herramientas?



Explique brevemente el razonamiento de su respuesta (2 responses)

La verdad no vi ayudas en la aplicación

Figura 21: Ayuda y Documentación.

A través de las respuestas de los encuestados se realizaron mejoras para la ayuda y la documentación utilizando textos de ayuda e iconos en los botones en la aplicación móvil, de manera que sea más intuitiva la navegación en la misma, además de proveer mayor claridad a la hora de utilizarla.

Al culminar la encuesta se les pidió a los profesores listar aspectos positivos y negativos de la interfaz de manera de poder mejorar el diseño de la aplicación.

Aspectos Positivos:

- Fácil uso para pasar listas de asistencia.
- Aplicabilidad en entorno real.
- Interfaz amigable.
- Fácil control de la cantidad de asistencias e inasistencias.
- Fácil Implementación.
- Basado en estándares lo que permitirá implementar mejoras en caso de ser necesario.
- De mucha utilidad en el área académica.

Aspectos Negativos:

- Falta de ayuda y guías.
- Ayuda en interfaz Web.
- Algunos errores de navegabilidad.
- Falta de verificación de algunos campos.
- Falta de estética en la aplicación móvil.

Los aspectos negativos se utilizaron para realizar las siguientes mejoras en el sistema:

- Se crearon validaciones para evitar números de cedula duplicados en los listados de las matriculas de las secciones.
- Se crearon validaciones para evitar nombres de sección duplicados dentro de un mismo semestre.
- Se mejoró la interfaz móvil para poder hacerla más usable.
- Se implementaron más ayudas a través de textos y diálogos de confirmación en la aplicación web.

5. Conclusiones

En este trabajo especial de grado se construyó una plataforma Web y móvil para poder chequear la asistencia de los estudiantes de la Facultad de Ciencias de la Universidad Central de Venezuela al aula de clases.

El objetivo primordial de este trabajo era el de presentar una herramienta de fácil uso y de fácil acceso para el personal docente de la Facultad de Ciencias de manera de poder automatizar el chequeo y la gestión de las asistencias. Durante el desarrollo de este trabajo la usabilidad fue el enfoque principal, dado que está dirigida a todos los profesores, los cuales no necesariamente poseen conocimientos técnicos avanzados en el uso de las herramientas utilizadas.

Finalizadas las pruebas descritas en el Capítulo 4, se analizaron los resultados y se aplicaron las mejoras recomendadas para optimizar la usabilidad de la aplicación, de esta manera una mayor parte del universo docente podría hacer uso de la misma y así ahorrar tiempo en tareas repetitivas como el chequeo y la gestión manual de las asistencias a cada una de las secciones, además de generar un histórico de todas las cátedras que el profesor ha dictado.

5.1 Contribuciones

Las siguientes fueron contribuciones hechas por este trabajo:

- Planteamiento de una solución para proporcionar un sistema automatizado de chequeo de asistencia a los profesores de la Facultad de Ciencias de la Universidad Central de Venezuela.
- Creación de un sistema de gestión de asistencia y reportes históricos por materia, sección y estudiante, permitiendo saber los porcentajes de asistencia de manera automática. Evitando así tener que traspasar listas manualmente a hojas de Excel.

5.2 Limitaciones

Las limitaciones que se presentaron durante el desarrollo de esta investigación fueron las siguientes:

- La versión de Bluetooth utilizada para las pruebas fue de al menos versión 2.2, se desconoce el comportamiento del sistema con versiones anteriores.
- El rango del chequeo de la aplicación móvil está limitada al rango del descubrimiento de dispositivos de Bluetooth (de hasta 10 metros sin

obstáculos), de manera que cualquier factor que influya en su rango de descubrimiento, afectara la efectividad de la aplicación.

- En la fase de pruebas ciertos dispositivos iPhone no pudieron ser detectados por el escaneo de la aplicación, se desconoce la razón de este fenómeno.

5.3 Trabajos Futuros

Entre las recomendaciones que se pueden hacer para futuras investigaciones se encuentran las siguientes:

- Crear una integración con el servidor LDAP de la Facultad de Ciencias de la Universidad Central de Venezuela, de manera que las secciones sean creadas y gestionadas automáticamente.
- Crear un módulo de estudiantes, el cual permita la autenticación y búsqueda de las asistencias del estudiante a las materias que cursa actualmente.
- Establecer una automatización de los escaneos basados en los días y horas específicas de clase de acuerdo al cronograma del semestre, de manera que el chequeo de asistencia sea totalmente transparente al profesor.
- Desplegar el sistema en un entorno de producción para ser utilizado por la comunidad docente de la Facultad de Ciencias de la Universidad Central de Venezuela.

Anexo A: Cuadros de detalle de los casos de uso.

Caso de uso:	Agregar Nueva Materia
Actor:	Usuario Profesor
Descripción:	Permite agregar una materia nueva al listado de materias del profesor.
Flujo Básico:	1. El profesor navega a través de la aplicación web al Módulo de Gestión de Materias. 2. En el Módulo de Gestión de Materias hace click sobre “Agregar Nueva Materia.” 3. El profesor provee al sistema todos los datos requeridos para la creación de la Materia. Como son: Código, Nombre, Cantidad de Créditos y Descripción de la Materia. 4. El profesor presiona el botón “Crear”.
Pre condiciones:	1. El profesor debe estar autenticado en la aplicación. 2. El profesor no debe haber creado una materia igual previamente.
Post condiciones:	Se podrá ver la materia nueva en el listado de materias del profesor y se podrán gestionar las secciones de esta materia.

Tabla 2: Caso de Uso Agregar Nueva Materia.

Caso de uso:	Eliminar Materia Existente
Actor:	Usuario Profesor
Descripción:	Permite modificar una materia en el listado de materias del profesor.
Flujo Básico:	1. El profesor navega a través de la aplicación web al Módulo de Gestión de Materias. 2. En el Módulo de Gestión de Materias hace click sobre “Listar Materias”. 3. El profesor hace click sobre el apartado de “Eliminar” en el listado de las materias en la materia que desea eliminar. 4. El profesor debe hacer click en aceptar cuando aparezca el pop-up para verificar su decisión de eliminar la materia.
Pre condiciones:	1. El profesor debe estar autenticado en la aplicación. 2. La materia debe existir.
Post condiciones:	Se podrá ver el listado actualizado sin la materia, todas las secciones e información asociada a esta materia serán eliminadas.

Tabla 3: Caso de Uso Eliminar Materia Existente.

Caso de uso:	Listar Materias Existente
Actor:	Usuario Profesor
Descripción:	Permite listar todas las materias del profesor.
Flujo Básico:	1. El profesor navega a través de la aplicación web al Módulo de Gestión de Materias. 2. En el Módulo de Gestión de Materias hace click sobre “Listar Ma-terias”.
Pre condiciones:	1. El profesor debe estar autenticado en la aplicación.
Post condiciones:	Se podrá ver el listado de todas las materias.

Tabla 4: Caso de Uso Listar Materias

Caso de uso:	Gestionar Secciones de una Materia
Actor:	Usuario Profesor
Descripción:	Permite gestionar todas las secciones de una materia del profesor.
Flujo Básico:	1. El profesor navega a través de la aplicación web al Módulo de Gestión de Materias. 2. En el Módulo de Gestión de Materias hace click sobre “Listar Ma-terias”. 3. El profesor hace click sobre el apartado de “Secciones” de la ma-teria deseada.
Pre condiciones:	1. El profesor debe estar autenticado en la aplicación. 2. La materia debe existir.
Post condiciones:	Se podrá ver el listado de todas las secciones asociadas a una materia.

Tabla 5: Caso de Uso Gestionar Secciones

Caso de uso:	Agregar Nueva Sección
Actor:	Usuario Profesor
Descripción:	Permite agregar una nueva sección a la materia.
Flujo Básico:	1. El profesor navega a través de la aplicación web al Módulo de Gestión de Materias. 2. En el Módulo de Gestión de Materias hace click sobre “Listar Materias”. 3. El profesor hace click sobre el apartado de “Secciones” de la materia deseada. 4. El profesor hace click sobre el botón “Crear Nueva Sección” 5. El profesor debe proveer todos los datos necesarios para la creación de la materia. Estos datos son extraídos del archivo .xls provisto por la División de Control de Estudios de la Facultad de Ciencias. 6. El profesor hace click en el botón “Crear Sección”
Pre condiciones:	1. El profesor debe estar autenticado en la aplicación. 2. La materia debe existir. 3. El profesor debe poseer el archivo .xls provisto por la División de Control de Estudios de la Facultad de Ciencias de la sección que desea crear.
Post condiciones:	Se podrá ver el listado de las secciones actualizado con la sección nueva.

Tabla 6: Caso de Uso Agregar Sección

Caso de uso:	Eliminar Sección Existente
Actor:	Usuario Profesor
Descripción:	Permite eliminar una sección asociada a una materia.
Flujo Básico:	1. El profesor navega a través de la aplicación web al Módulo de Gestión de Materias. 2. En el Módulo de Gestión de Materias hace click en “Listar Materias”. 3. El profesor hace click sobre el apartado de “Secciones” de la materia deseada. 4. El profesor hace click en el apartado de “Eliminar” en la sección deseada.
Pre condiciones:	3. El profesor debe estar autenticado en la aplicación. 4. La sección debe existir.
Post condiciones:	Se podrá ver el listado actualizado sin la sección, todos los alumnos e información asociada a esta sección serán eliminadas.

Tabla 7: Caso de Uso Eliminar Sección

Caso de uso:	Gestionar Matricula de una Sección
Actor:	Usuario Profesor
Descripción:	Permite gestionar la matrícula de una sección.
Flujo Básico:	1. El profesor navega a través de la aplicación web al Módulo de Gestión de Materias. 2. En el Módulo de Gestión de Materias hace click sobre “Listar Materias”. 3. El profesor hace click sobre el apartado de “Secciones” de la materia deseada. 4. En el listado de secciones el profesor hace click en el apartado de “Matricula” de la sección deseada.
Pre condiciones:	1. El profesor debe estar autenticado en la aplicación. 2. La sección debe existir.
Post condiciones:	Se podrá ver el listado de todos los estudiantes asociados a una sección.

Tabla 8: Caso de Uso Gestionar Matricula de Sección

Caso de uso:	Agregar Nuevo Estudiante
Actor:	Usuario Profesor
Descripción:	Permite agregar un nuevo estudiante a la matrícula.
Flujo Básico:	1. El profesor navega a través de la aplicación web al Módulo de Gestión de Materias. 2. En el Módulo de Gestión de Materias hace click en “Listar Materias”. 3. El profesor hace click sobre el apartado de “Secciones” de la materia deseada. 4. El profesor hace click sobre el apartado de “Matricula” de la sección deseada. 5. El profesor hace click sobre el botón “Agregar Estudiante”. 6. El profesor debe proveer todos los datos necesarios para la creación del estudiante. Estos datos son: Número de Cedula de Identidad, Nombre y Apellidos y Dirección de Correo Electrónico del Estudiante. 7. El profesor hace click en el botón “Crear Agregar Alumno”
Pre condiciones:	1. El profesor debe estar autenticado en la aplicación. 2. La materia debe existir. 3. La sección debe existir. 4. El alumno no debe existir en la matricula.

Post condiciones:	Se podrá ver el listado de todos los estudiantes actualizado con el nuevo alumno.
--------------------------	---

Tabla 9: Caso de Uso Agregar Estudiante

Caso de uso:	Retirar Estudiante
Actor:	Usuario Profesor
Descripción:	Permite retirar un estudiante de la matrícula.
Flujo Básico:	1. El profesor navega a través de la aplicación web al Módulo de Gestión de Materias. 2. En el Módulo de Gestión de Materias hace click sobre “Listar Materias”. 3. El profesor hace click en el apartado de “Secciones” de la materia. 4. El profesor hace click en el apartado de “Matricula” de la sección. 5. El profesor hace click en el apartado de “Retirar” del estudiante.
Pre condiciones:	1. El profesor debe estar autenticado en la aplicación. 2. La materia debe existir. 3. La sección debe existir. 4. El alumno debe existir en la matricula.
Post condiciones:	Se podrá ver el listado de todos los estudiantes actualizado sin el alumno, toda la información de este alumno será eliminada.

Tabla 10: Caso de Uso Retirar Estudiante

Caso de uso:	Actualizar Correo Electrónico del Alumno
Actor:	Usuario Profesor
Descripción:	Permite actualizar el correo electrónico de un alumno.
Flujo Básico:	1. El profesor navega a través de la aplicación web al Módulo de Gestión de Materias. 2. En el Módulo de Gestión de Materias hace click en “Listar Materias”. 3. El profesor hace click en el apartado de “Secciones” de la materia. 4. El profesor hace click en el apartado de “Matricula” de la sección. 5. El profesor hace click en el apartado de “Actualizar Correo” del estudiante deseado. 6. El profesor debe proporcionar la dirección de correo electrónica nueva del estudiante. 7. El profesor hace click en el botón de “Actualizar Correo Electrónico”
Pre condiciones:	1. El profesor debe estar autenticado en la aplicación. 2. La materia debe existir.

	- La sección debe existir. - El alumno debe existir en la matricula.
Post condiciones:	Se podrá ver el correo actualizado del estudiante en listado de todos los estudiantes.

Tabla 11: Caso de Uso Actualizar Correo del Estudiante

Caso de uso:	Liberar Dirección de Bluetooth
Actor:	Usuario Profesor
Descripción:	Permite liberar la dirección Bluetooth de un estudiante de la matrícula.
Flujo Básico:	- Navega a través de la aplicación web al Módulo de Gestión de Materias. - En el Módulo de Gestión de Materias hace click en “Listar Materias”. - Hace click sobre el apartado de “Secciones” de la materia deseada. - Hace click sobre el apartado de “Matricula” de la sección deseada. - Hace click sobre el apartado de “Liberar Dirección del Estudiante” del estudiante deseado.
Pre condiciones:	- El profesor debe estar autenticado en la aplicación. - La materia debe existir. - La sección debe existir. - El alumno debe existir en la matricula.
Post condiciones:	La información de la base de datos de los estudiantes es actualizada. Se podrá ver el campo de Dirección de Bluetooth vacío del estudiante.

Tabla 12: Caso de Uso Liberar Dirección Bluetooth del Estudiante

Caso de uso:	Escanear Dispositivos.
Actor:	Usuario Profesor.
Descripción:	Permite el escaneo de los dispositivos de los estudiantes de una sección
Flujo Básico:	- El profesor navega a través de la aplicación móvil al Listado de Materias tocando el botón de “Listar Materias”. - En el listado de las materias que tiene el profesor toca la materia deseada para desplegar sus secciones. - En caso de estar desactivado el Bluetooth, la aplicación pedirá permiso para activarlo. - En el listado de estudiantes se toca el botón de “Iniciar Escaneo”.

Pre condiciones:	1. El profesor debe estar autenticado en la aplicación. 2. La materia debe existir. 3. La sección debe existir. 4. El listado de alumnos debe existir.
Post condiciones:	Se actualizara la asistencia de los alumnos en el dispositivo del profesor una vez finalizado el proceso de descubrimiento de Bluetooth.

Tabla 13: Caso de Uso Escanear Dispositivos.

Caso de uso:	Culminar Clase.
Actor:	Usuario Profesor.
Descripción:	Permite actualizar la base de datos con el listado de las asistencias del día correspondiente.
Flujo Básico:	1. El profesor navega a través de la aplicación móvil al Listado de Materias tocando el botón de “Listar Materias”. 2. En el listado de las materias que tiene el profesor toca la materia deseada para desplegar sus secciones. 3. En caso de estar desactivado el Bluetooth, la aplicación pedirá permiso para activarlo. 4. En el listado de estudiantes se toca el botón de “Culminar Clase”.
Pre condiciones:	1. El profesor debe estar autenticado en la aplicación. 2. La materia debe existir. 3. La sección debe existir. 4. El listado de alumnos debe existir en la matricula. 5. El profesor debe haber realizado el “Escanear de Dispositivos” al menos una vez.
Post condiciones:	Se actualiza la asistencia de los estudiantes en la base de datos.

Tabla 14: Caso de Uso Culminar Clase.

Caso de uso:	Agregar Nuevo Profesor
Actor:	Usuario Administrador
Descripción:	Agrega un profesor nuevo a la base de datos de la aplicación.
Flujo Básico:	1. Ingresa en el “Módulo de Administración”. 2. Ingresa al apartado de “Agregar Profesores”. 3. Provee todos los datos necesarios para la creación del perfil, estos datos son: Número de Cedula de Identidad, Nombre, Apellido, Dirección de Correo, Número de Teléfono y la Contraseña que usara el profesor para autenticarse junto con su número de cedu-

	la. 4. Hace click en el botón “Crear Profesor”.
Pre condiciones:	1. El administrador debe estar autenticado. 2. El profesor no debe haberse ingresado previamente a la base de datos.
Post condiciones:	La información de la base de datos de los profesores es actualizada. La información del listado de los profesores es actualizada.

Tabla 15: Caso de Uso Agregar Nuevo Profesor.

Caso de uso:	Listar Profesores
Actor:	Usuario Administrador
Descripción:	Listar todos los profesores existentes en la base de datos de la aplicación.
Flujo Básico:	1. Ingresa en el “Módulo de Administración”. 2. Ingresa al apartado de “Listar Profesores”.
Pre condiciones:	1. El administrador debe estar autenticado. 2. Deben existir profesores en la base de datos.
Post condiciones:	La información de la base de datos de los profesores es actualizada. La información del listado de los profesores es actualizada.

Tabla 16: Caso de Uso Listar Profesores.

Caso de uso:	Modificar Profesor
Actor:	Usuario Administrador
Descripción:	Modifica un profesor en la base de datos de la aplicación.
Flujo Básico:	1. Ingresa en el “Módulo de Administración”. 2. Ingresa al apartado de “Listar Profesores”. 3. Hace click sobre el apartado de “Modificar” en el profesor deseado. 4. Provee los datos necesarios para la modificación del perfil, estos datos son: Dirección de Correo y Número de Teléfono. 5. Hace click en el botón “Modificar Profesor”

Pre condiciones:	El administrador debe estar autenticado. El profesor debe haberse ingresado previamente a la base de datos.
Post condiciones:	La información de la base de datos de los profesores es actualizada. La información del listado de los profesores es actualizada.

Tabla 17: Caso de Uso Modificar Profesor.

Caso de uso:	Eliminar Profesor
Actor:	Usuario Administrador
Descripción:	Elimina un profesor de la base de datos de la aplicación.
Flujo Básico:	1. Ingresa en el “Módulo de Administración”. 2. Ingresa al apartado de “Listar Profesores”. 3. Hace click sobre el apartado de “Eliminar” en el profesor deseado. 4. Hace click en aceptar cuando se valide la eliminación del profesor.
Pre condiciones:	1. El administrador debe estar autenticado. 2. El profesor debe haberse ingresado previamente a la base de datos.
Post condiciones:	La información de la base de datos de los profesores es actualizada. El listado de los profesores ya no mostrará al profesor eliminado.

Tabla 18: Caso de Uso Eliminar Profesor.

Referencias

- [1] ABI Research. *With an Installed Base of 10 Billion Devices Expected in 2018, Bluetooth will be an Essential Tool for Building the Internet of Everything*. London, United Kingdom. August 2013. Disponible: <https://www.abiresearch.com/press/with-an-installed-base-of-10-billion-devices-expec>. [Consultado Mayo 2013].
- [2] Bluetooth SIG, Inc. *Bluetooth Technology Fast Facts*. Disponible: <http://www.bluetooth.com/Pages/Fast-Facts.aspx>. [Consultado Julio 2013].
- [3] Bluetooth SIG, Inc. *History of the Bluetooth Special Interest Group*. Disponible: <http://www.bluetooth.com/Pages/History-of-Bluetooth.aspx> [Consultado Julio 2013].
- [4] Bluetooth SIG, Inc. *Specification of Bluetooth System. Covered Core Packing version 2.1*, July 2007. Disponible: http://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=241363. [Consultado: Junio 2013].
- [5] Bluetooth SIG, Inc. *What is the latest version of the Bluetooth specification? Where can I find GATT based Profile specifications?* Bluetooth.org. September 2011. Disponible: https://www.bluetooth.org/ticketing/view_article.cfm?action=article_comment&aid=45. [Consultado Agosto 2013].
- [6] Bray T. *What Android Is*. Ongoing. November 2010. Disponible: <http://www.tbray.org/ongoing/When/201x/2010/11/14/What-Android-Is>. [Consultado Agosto 2013].
- [7] Cunningham A. *Android 4.3 announced, bringing incremental changes to Jelly Bean*. Wired UK. July 2013. Disponible: <http://www.wired.co.uk/news/archive/2013-07/24/android-43>. [Consultado Septiembre 2013].
- [8] Chavez C. *Android PSA: Stop Using Task Killer Apps*. Phandroid. June 2011. Disponible: <http://phandroid.com/2011/06/16/android-psa-stop-using-task-killer-apps-now/>. [Consultado Noviembre 2013].
- [9] Gehrmann C., Persson J., Smeets B. *Bluetooth Security*. Artech House. July 2004.
- [10] Gilb. J. *Wireless Multimedia: A Guide to the IEEE 802.15.3 Standard*. IEEE Standards Association. New York, New York. March 2004.
- [11] Hopkins B., Antony R. *Bluetooth for Java*. Apress. Berkley, California. March 2003.

- [12] IEEE Standard for Information Technology - *Telecommunications and information exchange between systems - Local and metropolitan area networks*. IEEE Standard 802.15.1. Chapter 6. Architecture. June 2005.
- [13] International Communication Union. *ISDN user-network interface layer 3 specification for basic call control*. May 1998. Disponible: <http://www.itu.int/rec/t-rec-q.931/e>. [Consultado Mayo 2013].
- [14] Jormalainen S., Laine J. *Security in the WTLS*. Computer Science and Engineering. Helsinki University of Technology. November 1999. Disponible: <http://www.tml.tkk.fi/Opinnot/Tik-110.501/1999/papers/wtls/>. [Consultado Junio 2013].
- [15] Kostakos V. *The privacy implications of Bluetooth*. University of Madeira, Carnegie Mellon University. April 2008.
- [16] Meyer D. *Linux developer explains Android kernel code removal*. ZDNet. February 2010. Disponible: <http://www.zdnet.com/news/linux-developer-explains-android-kernel-code-removal/389733>. [Consultado Julio 2013].
- [17] Miller B.A., Bisdikian C. *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications*. Prentice Hall. September 2000.
- [18] Muller N. J. *Bluetooth Demystified*. Mc Graw Hill. April 2001.
- [19] Mulliner C., Herfurt M. *Blueprinting*. Trifinite.org. December 2004. Disponible: http://trifinite.org/trifinite_stuff_blueprinting.html. [Consultado Junio 2013].
- [20] Nathan E., Pertland A. *Reality Mining: Sensing Complex Social Systems*. MIT Media Laboratory, Cambridge. April 2005.
- [21] Open Handset Alliance. *Industry Leaders Announce Open Platform for Mobile Devices*. November 2007. Disponible: http://www.openhandsetalliance.com/press_110507.html. [Consultado Julio 2013].
- [22] Oxford Language Dictionaries Online. *Smartphone*. Oxford: Oxford University Press. Disponible: <http://www.oxforddictionaries.com/definition/english/smartphone>. [Consultado: Julio 2013].
- [23] Paul R. *Dream (sheep++): A developer's introduction to Google Android*. Ars Technica. February 2009. Disponible: <http://arstechnica.com/gadgets/2009/02/an-introduction-to-google-android-for-developers/>. [Consultado Junio 2013].
- [24] Facultad de Ciencias, Universidad Central de Venezuela. *Reglamento de Asistencia a Clase*. Caracas, Venezuela. Mayo 2003. [Consultado Febrero 2014].
- [25] Russakovskii A. *Custom ROMs For Android Explained – Here Is Why You Want Them*. Android Police. August 2012. Disponible: <http://www.androidpolice.com/2010/05/01/custom-roms-for-android-explained-and-why-you-want-them/>. [Consultado Agosto 2013].
- [26] Sager I. *Before iPhone and Android Came Simon, the First Smartphone*. Bloomberg Businessweek. June 2012. Disponible:

<http://www.businessweek.com/articles/2012-06-29/before-iphone-and-android-came-simon-the-first-smartphone>. [Consultado Febrero 2014].

[27] Simpson W. *The Point-to-Point Protocol (PPP)*. RFC 1661. July 1994.

[28] Snyder D. *HP iPAQ h6315 Pocket PC Phone Edition*. Pocketnow.com. July 2004. Disponible: <http://pocketnow.com/review/hp-ipaq-h6315-pocket-pc-phone-edition>. [Consultado Julio 2013].

[29] D'Orazio D. *Android KitKat is the next version of Google's mobile operating system*. The Verge. September 2013. Disponible: <http://www.theverge.com/2013/9/3/4690214/android-kit-kat-is-the-next-version-of-googles-mobile-operating-system>. [Consultado Septiembre 2013].

[30] Whitethorn J. *Android malware gives itself root Access*. Techadar. September 2012. Disponible: <http://www.techradar.com/news/phone-and-communications/mobile-phones/android-malware-gives-itself-root-access-1062294>. [Consultado Enero 2014].

[31] Whitson G. *Real World Test Show that Android Task Killers Are Still Useless*. Lifehacker. November 2011. Disponible: <http://lifehacker.com/5862994/real-world-test-show-that-android-task-killers-are-still-useless>. [Consultado Diciembre 2013].

[32] W. Bischofberger, *Prototyping-Oriented Software Development: Concepts and Tools*. September 2011. Springer-Verlag New York Incorporated.

[33] Suh W. *Web Engineering: Principles and Techniques*. Pensilvania. June 2005. Idea Group Publishing, pp 82 – 83.

[34] Estadísticas reflejadas en Android Studio al momento de la redacción de este documento. [Consultado Abril 2016]

[35] ECMA International. *Annotated ECMAScript 5.1*. August 2015. Disponible: <https://es5.github.io/#x10.2> [Consultado Abril 2016].

[36] Orsini L. *What you need to know about Node.js*. November 2013. Readwrite. Disponible: <http://readwrite.com/2013/11/07/what-you-need-to-know-about-nodejs/> [Consultado Abril 2016].

[37] Teixeira P. *Professional node.js: Building JavaScript Based Scalable Software*. John Wiley & Sons. October 2012. Disponible: <https://books.google.com/books?id=ZH6bpbcrlvYC&printsec=frontcover&dq=nodejs&hl=en&sa=X#v=onepage&q=nodejs&f=false>. [Consultado Marzo 2016].

[38] Kasiuk A. *On the problems with threads in Node.js*. April 2015. Future-processing.pl. Disponible: <http://www.future-processing.pl/blog/on-problems-with-threads-in-node-js/>. [Consultado Abril 2016]

[39] Node.js Foundation. *About Node.js*. Node.js Homepage. Disponible: <https://nodejs.org/en/about/> [Consultado Marzo 2016]

[40] Felmore B. *Android ril layer architecture. February 2014*. WantedDroid. Disponible: <http://wanteddroid.com/2014/02/android-ril-layer-architecture/> [Consultado Marzo 2016].

[41] Bluetooth Inc. *Our history*. Disponible: <https://www.bluetooth.com/media/our-history>. [Consultado Mayo 2016].

[42] *Winsock 2: Other Supported Protocols 4 Part 11*. Disponible: <http://www.winsocketdotnetworkprogramming.com/winsock2programming/winsock2advancedotherprotocol4j.html>. [Consultado Mayo 2016]

[43] Bluetooth SIG Inc. *Host Controller Interface (HCI) Architecture*. Disponible: <https://developer.bluetooth.org/TechnologyOverview/Pages/HCI.aspx>. [Consultado Mayo 2016]