

Proyecto aplicación Android para posicionamiento en interiores

Daniel Hilarion

Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda - Bogotá, Colombia
daniel.hilarion01@correo.usa.edu.co

Alejandro Toloza

Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda - Bogotá, Colombia
alejandro.toloza2@correo.usa.edu.co

Diego Ochoa

Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda - Bogotá, Colombia
diego.ochoa01@correo.usa.edu.co

Primera entrega del proyecto

1. Introducción

Actualmente la forma más difundida para ubicarse en el terreno es mediante uso de algún tipo de un sistema global de navegación por satélite (GNSS). Prácticamente todo móvil inteligente cuenta con receptores, sensores, etc. Pese a su grado de importancia, la señal de los GNSS deja de ser útil en interiores o bajo techo; edificios, centros comerciales, museos y túneles. Es muy difícil y en muchos casos imposible obtener una ubicación precisa, el problema de la localización en interiores todavía está por resolver, no hay ningún sistema ampliamente extendido y tan popular como es el GPS hoy en día. La demanda en la localización ha estado creciendo en los últimos años de manera exponencial gracias a los dispositivos móviles inteligentes. Un problema que presenta el GPS es que solo tiene o da cobertura cuando hay línea de visión directa, por lo que no tendría cobertura en interiores. Frecuentemente en la vida cotidiana es un requisito imprescindible conocer la posición del usuario. Hoy en día en el mercado existe una necesidad de desarrollar sistemas de posicionamiento o localización en interiores. En este proyecto se impuso como principal objetivo utilizar la tecnología Android, ya que cada día las tecnologías Open Source toman mayor fuerza y que cada día expande una mayor cobertura en el mercado convirtiéndose en un referente importante de los O.S (Sistemas operativos).

2. Sensores en un teléfono móvil inteligente

Son una parte vital de un Smartphone, cuando se compra un móvil el consumidor habitualmente fija aspectos tales como: pantalla, cámara, procesador o RAM, sin embargo lo que define a un móvil y a sus funcionalidades son los sensores, ya que estos lo diferencian de un ordenador portátil. ¿Sabías que un móvil inteligente tiene hasta 9 sensores diferentes? A continuación se mostraran Cuáles son, Cómo funcionan y para qué se utilizan.

a) Sensores relevantes de los teléfonos móviles

- **Acelerómetro:** Es un sensor básico para el funcionamiento del móvil. Un acelerómetro permite detectar si se mueve en una determinada dirección, si vibra, o si se agita. Es el sensor que le dice a las Apps si estamos usando el móvil en vertical u horizontal para ajustar la interfaz.
- **Giroscopio:** Mide la orientación en el espacio de un objeto. Este sensor puede saber en qué dirección se mueve el móvil. El Giroscopio es más preciso que el acelerómetro y hace una función en que éste no puede realizar, la cual es medir la rotación del móvil.
- **Sensor Proximidad:** Este sensor muestra la distancia a la que está situada un objeto cercano, ya que no tiene mucho alcance. El sensor de proximidad está compuesto por un emisor de LED de luz infrarroja, y un detector de rayos infrarrojos. El emisor lanza un haz infrarrojo, que rebota en el objeto y vuelve al origen, captando el receptor. Según lo que tarde en volver, se calcula la distancia. .
- **Pantalla Táctil:** Son pantallas convencionales, están compuestas de varias capas: protección, cristal, LCD, etc. Tienen una capa adicional por la que circula una corriente eléctrica de forma constante. Cuando tocamos con el dedo en la pantalla táctil alteramos la electricidad en esa zona, quedando registrada nuestra acción.
- **Sensor Cámara:** Es uno de los más importantes de todos, ya que se usa para hacer fotos. Se trata de un sensor que atrapa la luz magnificada a través de varios lentes. Los móviles modernos tienen varios sensores fotográficos en las cámaras duales e incluso en las nuevas.

- **GPS:**No es un sensor en sí mismo, ya que no mide nada a través de nuestro teléfono. En realidad, se trata de un sistema de comunicación entre un dispositivo y una red de satélites que dan vueltas alrededor de la tierra. El GPS del móvil genera una señal para que un satélite de la red lo detecte. Varios satélites leen la señal desde distintos lugares, y así interpolan nuestra posición exacta. .
- **Reconocimiento facial:**Uno de los sensores más controvertidos que usamos actualmente son las cámaras de reconocimiento facial, que Apple estrenó el año pasado en el Iphone X, y después muchos otros fabricantes. Este tipo de sensores es capaz de reconocer nuestro rostro, incluso aunque llevemos gafas y sombrero, o nos dejemos barba
- **Micrófono:**Es un sensor capaz de medir las ondas acústicas del sonido(principalmente la voz), la misión del micrófono es captar nuestra voz para transmitirla en las llamadas telefónicas o la videoconferencia.

3. Localización en entornos interiores

3.1. Tecnologías para localización en interiores: Wi-Fi, BLE e IMU.

- **WIFI:**Todos están familiarizados con las conexiones inalámbricas que da la posibilidad de conectarse a una red y a internet. Haciendo uso de un proceso de triangulación y con la cantidad de Access point mínima de 3 y el software correcto es posible lograr la ubicación de un dispositivo móvil, la precisión no es muy alta pero tiene la ventaja de usar una infraestructura ya existente con varios propósitos.
- **BLE:** Se basa en el estándar Bluetooth 4.0, funciona en las frecuencias de 2,4 GHz y fue creada por razones de marketing para dispositivos smartphone y tablet, casi en contraste con la tecnología NFC, cuya principal diferencia es la distancia de lectura, que en el caso de la tecnología BLE puede llegar hasta 100 metros.
- **IMU:** Proporcionan la resolución de posicionamiento más precisa y las tasas de actualización más rápidas requeridas por muchas aplicaciones integradas. Además, las IMU proporcionan datos de posición relativa desde un punto de inicio conocido en oposición a la información de posicionamiento absoluta proporcionada por un receptor del GNSS, por lo que los dos tipos de sensores de posición se complementan entre sí. Estas características de las IMU las convierten en muy útiles para argumentar la información de posicionamiento obtenida directamente de los receptores del GNSS.

3.2. Definición de RSSI (Received Signal Strength Indicator)

RSSI significa Indicador de intensidad de señal recibida. Es una medida estimada del nivel de potencia que recibe un dispositivo cliente RF desde un punto de acceso o enrutador. A distancias más grandes, la señal se debilita y las velocidades de datos inalámbricas se vuelven más lentas, lo que conduce a un rendimiento de datos general más bajo. La señal se mide mediante el indicador de intensidad de señal de recepción (RSSI), que en la mayoría de los casos indica qué tan buena es la frecuencia radial en particular conectada a cierta distancia.

3.3. Técnica de posicionamiento fingerprinting (análisis de escena)

Trabaja por medio de un lector de huella digital integrado al celular que lleva a cabo dos tareas, el primer método es el de Obtener una imagen de su huella digital y el otro método es el de Comparar el patrón de valles y crestas de dicha imagen con los patrones de las huellas que tiene almacenadas, el método por el cual obtiene una imagen de una huella digital es por medio de un software; El software utiliza algoritmos altamente complejos para reconocer y analizar las minutiae (rasgos específicos de la huella dactilar).

3.4. Podómetro y su funcionamiento.

Es un accesorio empleado para medir el número de pasos y que indirectamente puede servir para medir la distancia, velocidad y cadencia del caminar de una persona durante el día. ¿Cómo funciona? tiene un sensor interno que es capaz de detectar el balanceo producido por cada paso y registrarlo. Es Mediante la personalización de la distancia media del paso, que el podómetro puede deducir aproximadamente distancias, velocidades y cadencias.

4. Beacon Bluetooth

Un beacon es un transmisor hardware que transmite un paquete de información, permitiendo a smartphones, tablets y otros dispositivos electrónicos, realizar acciones específicas como emitir señales de aviso directamente a un dispositivo móvil, sin necesidad de una sincronización de los aparatos.

4.1. Beacon Bluetooth del fabricante Estimote (especificaciones y características técnicas)

Es un dispositivo que emite una señal de radio que puede ser interpretada por un teléfono celular, este aparato es alimentado por una batería y tiene un procesador ARM, memoria, bluetooth, sensores de temperatura y movimiento.

Para que el teléfono móvil pueda recibir señales de este dispositivo es necesario que cuente con:

Bluetooth Low Energy (BLE) activado, Wi-fi, iOS 7 o Android 4.3 o superior.

4.2. Trama de una señal BLE con protocolos iBeacon y Edystone

Los dispositivos beacon, emiten por broadcast señales de corto alcance (unos 50 metros en condiciones ideales). Son pequeños dispositivos que permiten fijarlos a la pared o sobre un cuadro eléctrico, máquina, etc. Utilizan tecnología bluetooth (bluetooth 4.0) de bajo consumo (bluetooth low energy BLE)

5. Trama de una señal WiFi (broadcast)

El término inglés broadcast significa “transmisión o radiodifusión”. Al principio, este concepto se refería a la transmisión de señales de radio y televisión a través de la red inalámbrica. De ahí se derivó su segundo significado: una forma de comunicación dentro de una red informática, mediante la que se envían mensajes sin un receptor específico. Pero ¿qué es concretamente el broadcast en la tecnología de redes, cómo funciona y qué papel desempeñan las direcciones de broadcast? En las redes informáticas, el broadcast es un mensaje que se transmite a todos los miembros de una red y que no necesita ninguna acción de retroalimentación. Un equipo conectado a la red envía un paquete de datos al resto de participantes de la red al mismo tiempo. En este proceso, el emisor no especifica ninguna dirección de destino, lo que distingue el broadcast del llamado unicast, en que el paquete solo se envía a un único destino conocido. La principal ventaja del broadcast es que la información puede distribuirse de forma masiva sin tener que enviarla en más de una ocasión.

Para llevar a cabo el proceso, se requiere una dirección especial que reemplaza las correspondientes direcciones de destino. Esta broadcast IP se utiliza mayormente cuando las direcciones individuales de los miembros de la red son desconocidas. Mediante la dirección de broadcast, los datos y la información se envían a todos los dispositivos de la red, siendo sus componentes responsables de recibir y procesar los datos. De este modo, la función de la broadcast IP es conectar todos los dispositivos de la red.

6. Desarrollo móvil

El desarrollo de aplicaciones móviles es el conjunto de procesos y procedimientos involucrados en la escritura de software para pequeños dispositivos inalámbricos de cómputo, como teléfonos inteligentes o tabletas.

El desarrollo de aplicaciones móviles es similar al desarrollo de aplicaciones web, y tiene sus raíces en el desarrollo de software más tradicional. Una diferencia fundamental, sin embargo, es que las aplicaciones (apps) móviles a menudo se escriben específicamente para aprovechar las características únicas que ofrece un dispositivo móvil en particular. Por ejemplo, una aplicación para juegos podría escribirse para aprovechar el acelerómetro de un iPhone.

Una forma de asegurar que las aplicaciones muestren un rendimiento óptimo en un dispositivo determinado es desarrollar la aplicación (app) de forma nativa en ese dispositivo. Esto significa que, a un nivel muy bajo, el código se escribe específicamente para el procesador de un dispositivo particular. Cuando una app necesita ejecutarse en varios sistemas operativos, sin embargo, hay poco –si es que hay alguno– código que puede ser reutilizado desde el desarrollo inicial. La aplicación debe ser esencialmente reescrita para cada dispositivo específico.

6.1. Beacon Bluetooth del fabricante Estimote (especificaciones y características técnicas)

Los dispositivos beacon, emiten por broadcast señales de corto alcance (unos 50 metros en condiciones ideales). Son pequeños dispositivos que permiten fijarlos a la pared o sobre un cuadro eléctrico, máquina, etc. Utilizan tecnología bluetooth (bluetooth 4.0) de bajo consumo (bluetooth low energy BLE)

Para emitir señales o informar en tiempo de avisos directamente a un dispositivo móvil, sin necesidad de una sincronización de los aparatos, la señal es captada por estos dispositivos y se transmite a un servidor en la nube a través de Internet. El servidor de la nube procesa la información y lleva a cabo análisis o avisos, según la función requerida.

Los Beacon definen las zonas y áreas de trabajo por geolocalización. Se crea un mapa virtual para la definición de áreas productivas, gestión de permisos de acceso y alertas preventivas, guía virtual de escape en caso de situación de emergencia. Estos dispositivos están vinculados con Smartglasses y Smartwatches que gracias a la hiperconectividad, se detectan unos a otros y realizan las funciones para las que se les ha configurado.

Esto nos permite ofrecer múltiples soluciones y pueden proporcionar todo tipo de información. Tienen un gran potencial, siendo las aplicaciones más destacadas las de mejoras en Prevención y en Producción. El hardware se compone de un

controlador con un radio emisor bluetooth. Alimentados por una batería (pila de botón). o directamente conectados a una toma de corriente. Originalmente funcionaban con sistemas de bluetooth tradicionales, pero los dispositivos más nuevos utilizan tecnología bluetooth BLE (de los fabricantes Nordic Semiconductor y Texas Instruments). Tiene un consumo muy reducido y por lo tanto, permiten al beacon funcionar durante mucho más tiempo.

6.2. Describa la trama de una señal BLE que utiliza los protocolos iBeacon y Edystone (diferencias)

BLE utiliza 40 canales, con 3 canales específicos (37, 38 y 39) para paquetes de publicidad y 37 canales (del 1 al 36) para paquetes de datos. La estructura de los paquetes de capas de enlace consta de 4 campos obligatorios: preámbulo, dirección de acceso, carga útil de PDU, comprobación de redundancia cíclica (CRC).

Durante la fase de diseño de un transmisor BLE —y también después de ella—, es importante realizar pruebas para asegurarse de que la información transmitida sea correcta. Es necesario verificar, por ejemplo, la estructura de los paquetes de prueba, la potencia de salida, las características de modulación y el error de frecuencia de portadora. El Bluetooth SIG (Bluetooth Interest Group) es el grupo encargado de desarrollar el estándar bluetooth, y, por lo tanto, el que en el año 2010 presentó el Bluetooth Smart o Bluetooth Low Energy (la versión 4 del estándar) tomando como partida el Wibree de Nokia (2006). BLE es una evolución del estándar bluetooth diseñado para el IoT, y por lo tanto con un enfoque en el bajo consumo.

6.3. Desarrollo nativo y sus ventajas

Las aplicaciones nativas son las aplicaciones desarrolladas con el lenguaje de programación oficial de un sistema operativo en concreto. Si una app está desarrollada para iOS esta app solo se podrá utilizar en todos los dispositivos de iOS. A continuación se mostrarán ejemplos de lenguajes nativos para cada plataforma móvil.

Swift y Objective-C: iOS Java y Kotlin: Android .Net: Windows Phone Obviamente, a diferencia de el desarrollo de aplicaciones en otros lenguajes de programación ajenos a lo nativo de cada plataforma, aquí tendremos ciertas ventajas dependiendo de la plataforma que se elija, vamos a ver cuáles son algunas de ellas.

Máximo rendimiento en dispositivos Al aprovechar las ventajas que nos brinda un lenguaje de programación nativo, estaremos aprovechando también el hecho de sacar el máximo rendimiento a los dispositivos que cuentan con ese sistema operativo.

Se incrementa la experiencia del Usuario Como se indica anteriormente, al obtener un mejor rendimiento, estas a su vez obteniendo una mejor experiencia de usuario en cuanto a tiempos de carga, fluidez y diseño se refiere.

Aplicaciones disponibles desde la App Store Como es de suponer, las aplicaciones totalmente nativas cuentan con una mayor visibilidad en la tienda de aplicaciones, por encima de aplicaciones híbridas o WebApps.

6.4. Aplicación web para dispositivos móviles y sus ventajas

Estas aplicaciones son ‘web’ porque trabajan en la nube y necesitan de internet para funcionar, aunque no necesitan de ser instaladas en el ordenador ya que se ejecutan desde el navegador tradicional, ya sea Google Chrome, Firefox o cualquier otro. Guardan toda la información necesaria en los servidores y cuando la necesitamos se recuperan de allí para mostrársela al usuario.

La mayor ventaja es que en la mayoría de ocasiones el usuario puede recuperar la información propia desde cualquier dispositivo y desde cualquier lugar con su nombre de usuario y contraseña. Seguramente hayas utilizado en más de una ocasión aplicaciones web sin siquiera saberlo. Los servicios de correo electrónico como Gmail o los productos de gestión de documentos en la nube como Google Docs. O simplemente los juegos que ejecuta Facebook desde el navegador como el Candy Crush o FarmVille. Cada vez más estos servicios están más integrados en el navegador.

6.5. ¿Qué es un diagrama de casos del sistema?

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso consta de los siguientes elementos:

- Actor: Un actor representa quien o que inicia una acción dentro del sistema, en otras palabras, es simplemente un rol que es llevado a cabo por una persona o cosa. Un Actor en un diagrama Uso-Caso es representado por una figura en forma de persona.
- Uso-Caso: El uso-caso en sí es representado por un ovalo que describe la funcionalidad a grosso modo que se requiere por el sistema.

- Comunicación: Este elemento representa la relación que existe entre un Uso-Caso y un Actor, dicho elemento es representado simplemente por una línea recta que se extiende de la figura del actor hacia el ovalo del uso-caso.
- Limite de Sistema (System Boundry): Empleado para delimitar los limites del sistema, y representado por un rectángulo con color de fondo distintivo.
- Generalización : Una generalización indica que un uso-caso (ovalo) es un caso especial de otro caso, en otros términos, representa una relación padre-hijo, donde el hijo puede ser suplido directamente por el padre en cualquier momento. Este elemento es representado por una línea con flecha que se extiende del uso-caso hijo hacia el uso caso padre (general).
- Inclusión : Una inclusión es utilizada para indicar que un uso-caso (ovalo) depende de otro caso, dicho de otra manera, significa que la funcionalidad de determinado caso se requiere para realizar las tareas de otro. Este elemento es representado por una línea punteada con flecha y comentario `«include»` que se extiende del uso-caso base hacia el uso caso de inclusión.
- Extensión : Una extensión representa una variación de un uso-caso a otro, aunque similar a una generalización, una extensión representa una dependencia específica, mientras una generalización no implica que los usos-casos dependen uno del otro. Este elemento es representado por una línea punteada con flecha y comentario `«extend»` que origina del uso-caso base hacia el uso caso de extensión.

6.6. ¿Qué es un diagrama de actividades?

Un diagrama de actividades muestra el flujo de actividades, siendo un actividad una ejecución general entre los objetos que se está ejecutando en un momento dado dentro de una máquina de estados, el resultado de un actividad es una acción que producen un cambio en el estado del sistema o la devolución de un valor.

Desde un punto de vista conceptual, el diagrama de actividades muestra cómo fluye el control de unas clases a otras con la finalidad de culminar con un flujo de control total que se corresponde con la consecución de un proceso más complejo. Por este motivo, en un diagrama de actividades aparecerán acciones y actividades correspondientes a distintas clases.

Los elementos que contiene un diagrama de actividades son:

- Inicio: El inicio de un diagrama de actividad es representado por un círculo de color negro sólido.
- Actividad : Una actividad representa la acción que será realizada por el sistema la cual es representada dentro de un ovalo.
- Transición: Una transición ocurre cuando se lleva a cabo el cambio de una actividad a otra, la transición es representada simplemente por una línea con una flecha en su terminación para indicar dirección.
- Ramificación (Branch) : Una ramificación ocurre cuando existe la posibilidad que ocurra más de una transición (resultado) al terminar determinada actividad. Este elemento es representado a través de un rombo.
- Unión (Merge) : Una unión ocurre al fusionar dos o más transiciones en una sola transición o actividad. Este elemento también es representado a través de un rombo.
- Expresiones Resguardadas (Guard Expressions) : Una expresión resguardada es utilizada para indicar una descripción explícita acerca de una transición. Este tipo de expresión es representada mediante corchetes [...] y es colocada sobre la línea de transición.
- Fork : Un fork representa una necesidad de ramificar una transición en más de una posibilidad. Aunque similar a una ramificación (Branch) la diferencia radica en que un fork representa más de una ramificación obligada, esto es, la actividad debe proceder por ambos o más caminos, mientras que una ramificación (Branch) representa una transición u otra para la actividad (como una condicional). Un fork es representado por una línea negra sólida, perpendicular a las líneas de transición .
- Join : Una join ocurre al fusionar dos o más transiciones provenientes de un fork, y es empleado para dichas transiciones en una sola, tal y como ocurría antes de un fork . Un fork es representado por una línea negra sólida, perpendicular a las líneas de transición .
- Fin : El fin de un diagrama de actividad es representado por un círculo, con otro círculo concéntrico de color negro sólido.
- Canales (Swimlanes) : En determinadas ocasiones ocurre que un diagrama de actividad se expanda a lo largo de más de una entidad o actor, cuando esto ocurre el diagrama de actividad es particionada en canales (swimlines), donde cada canal representa la entidad o actor que está llevando a cabo la actividad.

6.7. ¿Qué son los requisitos de software?

El Glosario de Terminología Estándar de Ingeniería de Software (IEEE: Standard Glossary of Software Engineering Terminology) define al requisito como:

1. Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.

2. Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
3. Una representación en forma de documento de una condición o capacidad como las expresadas en 1 o en 2.

Los requisitos se han convertido en un punto clave en el desarrollo de las aplicaciones informáticas. Un gran número de proyectos de software naufragan debido a una mala definición, especificación o administración de requisitos. Factores tales como requisitos incompletos o mal manejo de los cambios de los requisitos llevan a proyectos completos al fracaso total.

7. Google App Engine

Este servicio es del tipo Plataforma como Servicio o Platform as a Service (PaaS), nos permite publicar aplicaciones web en línea sin necesidad de preocuparnos por la parte de la infraestructura y enfocándose totalmente en la construcción de nuestra aplicación y en la posibilidad de correrla directamente sobre la infraestructura de Google, es decir, la que Google usa para sus propios productos.

Como cualquier otra Plataforma como Servicio, App Engine nos facilita construir, mantener y escalar nuestra aplicación en la medida que sea necesario.

Actualmente se divide en dos módulos:

- Google App Standard Environment: Solo tiene posibilidad de programación con C, Java, Python, PHP y GO. No permite accesos a disco usando estos lenguajes, a cambio, los costes son más económicos.
- Google App Flexible Environment : Puede implementar cualquier lenguaje de programación mediante contenedores Docker, permitiendo mayor control sobre la aplicación. Los costes son proporcionales al consumo de CPU.

8. Bases de datos

Es una colección de información/datos almacenada de cierta forma y con una estructura específica, (eso depende del motor usado, por ejemplo MongoDB o PostgreSQL), que tienen rutinas optimizadas para guardar, eliminar, consultar y gestionar información.

Entonces, una base de datos está conformada por modelos o tablas; cada tabla tiene N atributos, de diferentes tipos de datos, ellos se guardan como registros, por ejemplo: un registro de la tabla de usuarios en una base de datos, tiene la información del nombre y apellido de una persona (más campos dependiendo de nuestra lógica de negocio), y así podemos tener millones de registros, relacionarlos, indexarlos y demás.

8.1. Almacenamiento de datos en base de datos SQL y noSQL

Una base de datos SQL es aquella base de datos relacional que esta escrita en lenguaje SQL (Structured Query Language o lenguaje de consulta estructurado), también pronunciado «sequel». Este lenguaje se considera el lenguaje estándar para las bases de datos según el ANSI (American National Standards Institute o Instituto Nacional Americano de Estándares en español), aunque no es el único que hay.

Una vez que se tiene una base de datos, hace falta un sistema que sea capaz de crear y proporcionar acceso (acceso de distintos niveles) a los datos de la base de datos. Este es el sistema de gestión de base de datos (DBMS, Data Base Management System). Algunos ejemplos de sistemas de bases de datos comunes son: SAP HANA, Microsoft SQL Server y Amazon RDS.

Hablar de bases de datos NoSQL es hablar de estructuras que nos permiten almacenar información en aquellas situaciones en las que las bases de datos relacionales generan ciertos problemas debido principalmente a problemas de escalabilidad y rendimiento de las bases de datos relacionales donde se dan cita miles de usuarios concurrentes y con millones de consultas diarias.

Además de lo comentado anteriormente, las bases de datos NoSQL son sistemas de almacenamiento de información que no cumplen con el esquema entidad-relación. Tampoco utilizan una estructura de datos en forma de tabla donde se van almacenando los datos sino que para el almacenamiento hacen uso de otros formatos como clave-valor, mapeo de columnas o grafos (ver epígrafe 'Tipos de bases de datos NoSQL').

8.2. SQLite

SQLite es una biblioteca escrita en lenguaje C que implementa un Sistema de gestión de bases de datos transaccionales SQL auto-contenido, sin servidor y sin configuración. El código de SQLite es de dominio público y libre para cualquier uso, ya sea comercial o privado. Actualmente es utilizado en gran cantidad de aplicaciones incluyendo algunas desarrolladas como proyectos de alto nivel.

SQLite es un sistema completo de bases de datos que soporta múltiples tablas, índices, triggers y vistas. No necesita un proceso separado funcionando como servidor ya que lee y escribe directamente sobre archivos que se encuentran en el disco duro. El formato de la base de datos es multiplataforma e indistintamente se puede utilizar el mismo archivo en sistemas de 32 y 64 bits.

La base de datos se almacena en un único fichero a diferencia de otros DBMS que hacen uso de varios archivos. SQLite emplea registros de tamaño variable de forma tal que se utiliza el espacio en disco que es realmente necesario en cada momento.

El código fuente está pensado para que sea entendido y accesible por programadores promedio. Todas las funciones y estructuras están bien documentadas.

Ventajas de SQLite:

- **Tamaño:** SQLite tiene una pequeña memoria y una única biblioteca es necesaria para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
- **Rendimiento de base de datos:** SQLite realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.
- **Portabilidad:** se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.
- **Estabilidad:** SQLite es compatible con ACID, reunión de los cuatro criterios de Atomicidad, Consistencia, Aislamiento y Durabilidad.
- **SQL:** implementa un gran subconjunto de la ANSI – 92 SQL estándar, incluyendo subconsultas, generación de usuarios, vistas y triggers.
- **Interfaces:** cuenta con diferentes interfaces del API, las cuales permiten trabajar con C++, PHP, Perl, Python, Ruby, Tcl, Groovy, Qt ofrece el plugin sqlite, etc.
- **Costo:** SQLite es de dominio público, y por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente.

9. Conclusiones

Después de realizar la investigación de cada uno de los temas descritos en este artículo, nos permitimos afirmar que es posible diseñar una aplicación para posicionamiento de interiores haciendo uso de software especializado para esta tarea que es de dominio abierto y no tiene gran dificultad para obtenerse.

Además de esto, gracias a la adaptación de todos estos conceptos ya podemos tener una idea de la manera en que debemos proceder para la construcción del proyecto, los programas y equipo requerido, etc.

Cabe resaltar que aunque esta teoría será importante, todavía hace falta mucha más investigación y manejo de conceptos que se apropiaran más que todo con la práctica y las orientaciones del docente conforme vaya avanzando la asignatura.

10. Apéndice

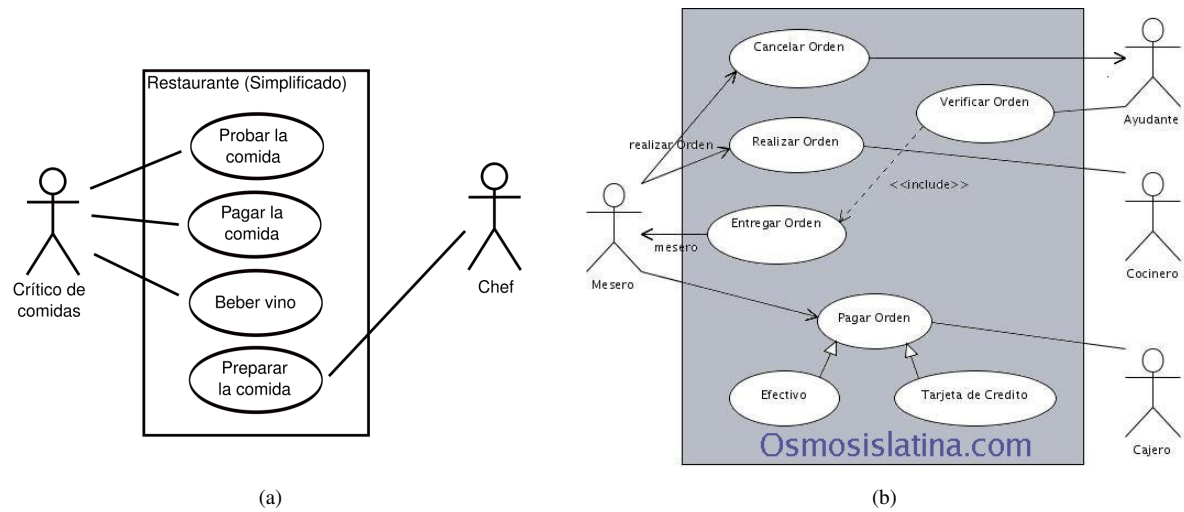


Figura 1: Ejemplos de un diagrama de casos de uso

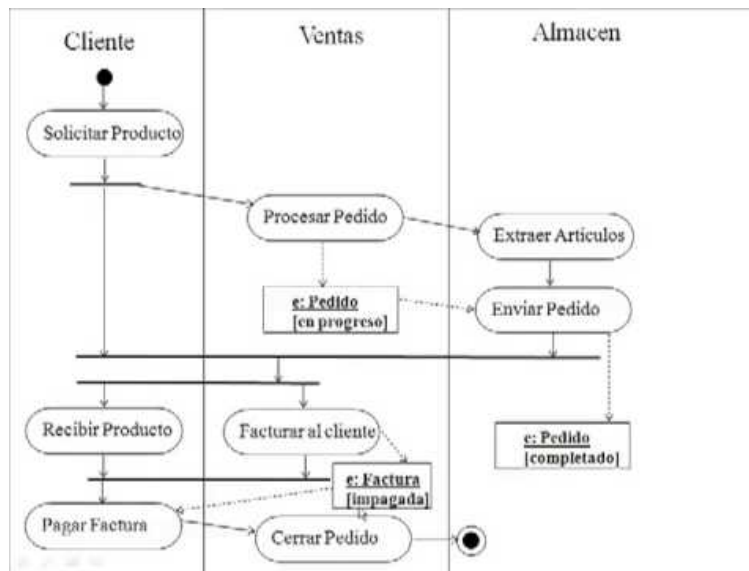


Figura 2: Ejemplo de un diagrama de actividad

Diferencias entre bases de datos SQL y NoSQL		
	SQL	NoSQL
Estructuración	Sí	No
Escalabilidad	Baja	Alta
Flexibilidad	Baja	Alta
Estandarización	Sí	No
Aplicaciones	Muchas	Pocas
Compatibilidad con SQL	Sí	Incompatible
Rapidez	Depende	Depende
Soporte	Alto	Bajo
Integridad datos	Mayor	Menor

tic.PORTAL

Figura 3: Diferencias entre bases de datos SQL y NoSQL

Segunda entrega del proyecto
(El enlace al vídeo de la explicación se encuentra en la carpeta .zip al igual que la carpeta con la aplicación)

11. Diagrama de casos del sistema

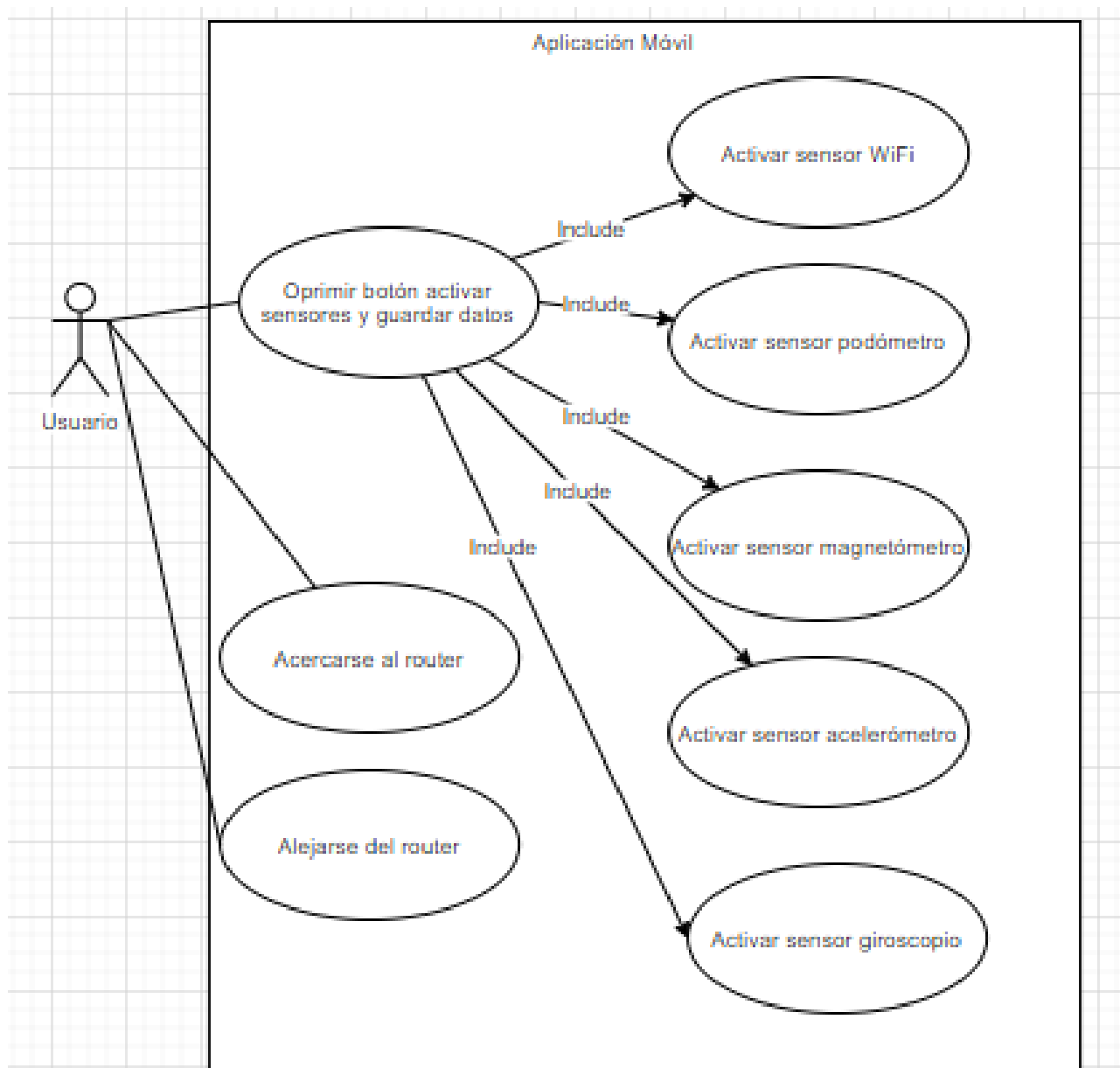


Figura 4: Diagrama de casos de uso de la aplicación

12. Diagrama de casos

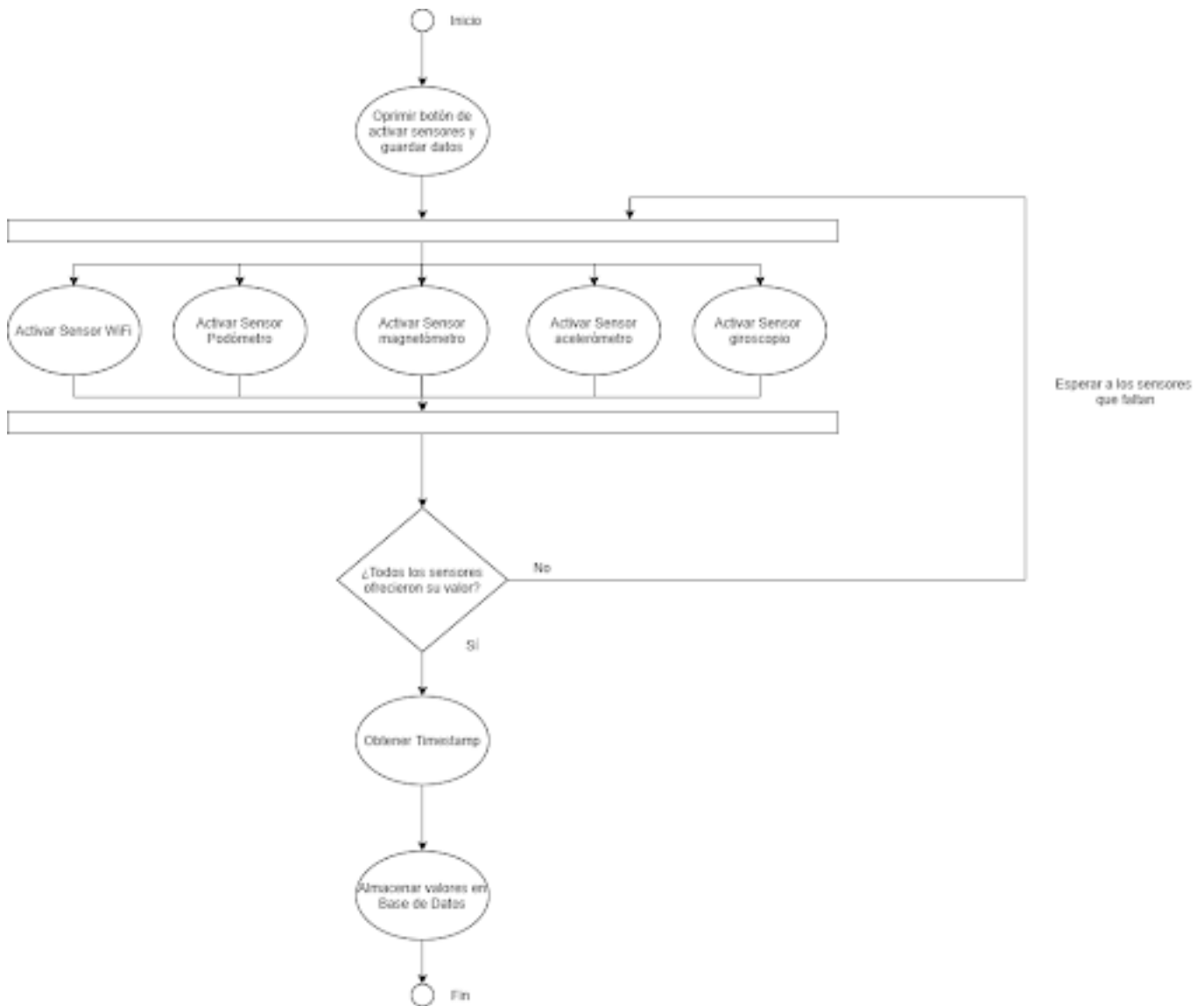


Figura 5: Diagrama de actividades de la aplicación

13. Conclusiones

Luego de realizar la primera practica con el proyecto se pudo evidenciar las ventajas que tienen las bases de datos SQLite por su tamaño y su rendimiento más óptimos a la hora de realizar la aplicación. También se pudo demostrar que la precisión a la hora de la toma de datos por los sensores puede variar dependiendo de muchos factores, la efectividad que tengan estos dentro del dispositivo de cada persona, la calidad de la señal Wifi a la cual este conectado el dispositivo, la distancia a la cual se este del módem, entre otros.

A pesar de estos factores, se puede decir que la aplicación cumple su objetivo principal ya que guarda los datos de los sensores requeridos del dispositivo móvil en la respectiva base de datos y los muestra al usuario. A partir de este punto, se enfocara el avance tenido hasta el momento hacia más opciones que permitan un mejor manejo de los datos guardados, así como el uso de estos para lograr el objetivo de posicionarse con la mayor exactitud posible dentro de interiores.

Tercera entrega del proyecto

NOTA: En la entrega del proyecto de este corte se hizo el trabajo correspondiente al proyecto del COVID-19 en Python, esto debido a una confusión con la guía donde se encontraban los proyectos ya que en la sección donde se encontraba el título del proyecto de Android, el trabajo a realizar que se encontraba en esa sección era el del COVID-19.

14. Conclusiones

Se pudo realizar la aplicación en Python requerida satisfactoriamente utilizando Jupyter Notebook. Allí se pudieron representar los datos guardados en la base de Datos por medio de las diferentes gráficas requeridas, como diagramas de tortas, diagramas de barras, entre otras más. Se modelaron datos tanto del país en general como para la ciudad de Bogotá, donde en este último se modelaron por localidades. Por último, cabe decir que en el desarrollo del proyecto se adquirió gran conocimiento acerca del lenguaje Python y del uso de Jupyter Notebook para realizar aplicaciones y documentos con gráficas correctamente ordenadas.

15. Link del repositorio GIT del proyecto:

https://github.com/Diego-260/PROYECTO_SA_TERCERCORTE.git

Referencias

- [1] M. J. Gutiérrez., *Sensores móviles*, 2014. [Online]. Available: <https://elandroidelibre.espanol.com/2014/07/cuales-son-y-para-que-sirven-los-sensores-de-nuestros-android.html>
- [2] J. A. Pascual, *Estos son los sensores que tiene tu móvil*, 2018. [Online]. Available: <https://computerhoy.com/listas/tecnologia/estos-son-todos-sensores-que-tiene-tu-movil-te-contamos-que-sirve-cada-311415?page=1>
- [3] P. I. soluciones, *Tecnologías de localización IOT*, 2017. [Online]. Available: <https://www.pandaid.com/tecnologias-de-localizacion-iot/>
- [4] speedcheck, *¿What is a RSSI?* [Online]. Available: <https://www.speedcheck.org/wiki/rssi/>
- [5] E. S. Almazán, *Qué son los estimote*, 2015. [Online]. Available: <https://www.exentrit.mx/que-son-los-estimote-beacons/>
- [6] I. 4.0, *Industria 4 0*. [Online]. Available: <https://industria40.me/beacons/>
- [7] Ionos, *¿Qué es y cómo funciona una broadcast IP?* [Online]. Available: <https://www.ionos.es/digitalguide/servidores/known-how/broadcast-ip/>
- [8] U. N. A. y a Distancia, *Diagramas de casos de uso*. [Online]. Available: http://stadium.unad.edu.co/ovas/10596_9839/diagramas_de_casos_de_uso.html
- [9] O. Latina, *Diagrama Uso-Caso*. [Online]. Available: <https://www.osmosislatina.com/lenguajes/uml/casos.htm>
- [10] U. N. A. y a Distancia, *Diagrama de actividades*. [Online]. Available: http://stadium.unad.edu.co/ovas/10596_9839/diagramas_de_actividades.html
- [11] O. Latina, *Diagramas de Actividad*. [Online]. Available: <https://www.osmosislatina.com/lenguajes/uml/actividad.htm>
- [12] EcuRed, *Requisitos de Software*. [Online]. Available: https://www.ecured.cu/Requisitos_de_Software#Conceptos
- [13] T. P., *Qué es y cómo funciona Google App Engine*. [Online]. Available: <https://platzi.com/blog/google-app-engine/>
- [14] Wikipedia, *Google App Engine*. [Online]. Available: https://es.wikipedia.org/wiki/Google_App_Engine
- [15] Platzi, *Bases de Datos: Qué es, Tipos, Usos*. [Online]. Available: <https://platzi.com/base-de-datos/>
- [16] tic PORTAL, *Base de datos SQL*. [Online]. Available: <https://www.ticportal.es/glosario-tic/base-datos-sql>
- [17] EcuRed, *SQLite*. [Online]. Available: <https://www.ecured.cu/SQLite>