

# Gestor de visitas para móviles Android.

**Autor:**

Juan de dios Marfil Muñoz  
Ingeniería Técnica de Informática de Gestión

**Consultor:**

Helena Boltà Torrell



Esta obra está sujeta a una licencia de Reconocimiento-No Comercial-Sin Obra Derivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## Índice

<b>1. Introducción .....</b>	<b>- 3 -</b>
1.1 Contexto y justificación del Trabajo .....	1
1.2 Objetivos del Trabajo .....	2
1.3 Enfoque y método seguido .....	2
1.4 Funcionalidades.....	3
1.5 Planificación del trabajo .....	4
1.6 Descripción de otros capítulos de la memoria.....	7
1.7 Sumario de los productos obtenidos .....	7
<b>2. Diseño centrado en el usuario.....</b>	<b>8</b>
2.1 Usuarios y contexto de uso de la aplicación .....	8
2.1.1 Usuarios .....	8
2.1.2 Contexto de uso de la aplicación .....	10
2.2 Diseño conceptual.....	11
2.2.1 Escenarios de uso .....	11
2.2.2 Flujos de interacción.....	13
2.3 Prototipado .....	16
2.4 Evaluación .....	23
<b>3. Diseño Técnico .....</b>	<b>25</b>
3.1 Diagrama Entidad- relación de la base de datos.....	25
3.2 Diagrama UML de clases .....	25
3.3 Diagrama explicativo de la arquitectura del sistema.....	27
3.4 Casos de uso .....	28
<b>4. Implementación .....</b>	<b>35</b>
4.1 Entorno de desarrollo .....	35
4.2 Estructura de la aplicación .....	36
4.3 Estado actual de proyecto.....	38
<b>5. Testing .....</b>	<b>39</b>
5.1 Test Unitarios .....	39
5.2 Pruebas de integración .....	40
<b>6. Documentación técnica .....</b>	<b>43</b>
6.1 Clases.....	43
6.2 Relacion de funcionalidades .....	47
<b>7. Manuales .....</b>	<b>48</b>
7.1 De instalación .....	48
7.2 De usuario .....	49
<b>8. Conclusiones .....</b>	<b>52</b>
<b>9. Bibliografía.....</b>	<b>53</b>

# 1. Introducción

---

## 1.1 Contexto y justificación del Trabajo.

El contexto:

Este proyecto nace de la necesidad de agilizar un registro y seguimiento de visitas, en un entorno comercial para la venta de piezas de recambio del automóvil.

En la actualidad, la realización de un registro de visitas y repartos de un vendedor se hace de forma manual en una plantilla tipo folio A-4, donde el vendedor registra su actividad diaria. Y posteriormente, un administrativo pasa esta actividad a un archivo tipo hoja de cálculo, donde finalmente quedará registrado.

Esta aplicación tendrá la funcionalidad de un registro de visitas comerciales, donde podremos tener una base de datos con: las visitas realizadas, el resultado de las mismas e información para un seguimiento en el tiempo de estas visitas.

En mi trabajo, en el cual desarrollo mi actividad profesional, no existe ningún medio portátil, fácil y rápido para poder realizar un registro de visitas a clientes en la forma descrita, por parte del vendedor de piezas de recambio.

Como puntos fuertes a destacar:

- Ahorro de recursos humanos. No se duplicarán las tareas de registro como hasta entonces, primero en papel y luego en hoja de cálculo.
- Tendremos una herramienta para poder gestionar de manera versátil una base de datos de clientes y visitas.
- Mejoramos el proceso de registro diario de visitas del vendedor, pasando directamente su trabajo a digital.
- Herramienta de trabajo con potencial para ser adaptada y mejorada en función de necesidades futuras.

Como puntos débiles:

- La adaptación/aprendizaje a utilizar la aplicación por parte del personal de ventas.
- La inversión económica inicial en Tablets /móviles.

Se utilizarán conceptos ya usados habitualmente por la fuerza de ventas para hacer la aplicación más fácil de asimilar por parte del usuario, y disminuir así la dificultad en la curva de aprendizaje.

Además tendrá funciones adicionales como “propuesta de visitas”, para facilitar el trabajo al comercial y poder organizar sus visitas dependiendo de la prioridad establecida en el sistema.

El resultado que quiero obtener es el de una aplicación fácil de utilizar, robusta y que se adapte perfectamente a la operativa del trabajo del vendedor de piezas de recambio.

### 1.2 Objetivos del Trabajo.

Este proyecto deberá cubrir varios objetivos fundamentales:

- Aprender a desarrollar un proyecto en la vida real.
- Profundizar en el lenguaje de programación Java/Android con vistas a poder dedicarme profesionalmente a ello.
- Agilizar el trabajo de registro de visitas realizado por el vendedor, ya que realizará la acción “in situ”, en el momento de finalizar la visita.
- Digitalizar el registro de visitas del vendedor, aportando información que antes solo estaba en soporte papel, anulando así la duplicidad de tareas.
- Mejorar la gestión de visitas (información, registro y seguimiento) con funciones como: histórico de visitas por cliente o propuestas de visitas a realizar.

Como entorno de desarrollo IDE, se utilizará Android Studio. Es una plataforma que empieza a consolidarse, con el fuerte respaldo Google.

Y por lo tanto, dirigido a dispositivos Android, concretamente para móviles con pantalla entre 5 y tablets con 7 pulgadas de pantalla.

### 1.3 Enfoque y método seguido.

En Android se utiliza el patrón de arquitectura llamado Modelo Vista Controlador (MVC). Consiste en clasificar los datos de una aplicación en 3 segmentos, con el objetivo de obtener una aplicación bien estructurada, reutilizable, con un código claro y encapsulado. Las 3 partes se definen a continuación:

- Modelo: son las representaciones que construiremos basadas en la información con la que operará nuestra aplicación. En este proyecto se utilizará un modelo basado en base de datos, concretamente, en SQLite.
- Vista: es la interfaz con la que va a interactuar el usuario. El proyecto contará con una vista/pantalla inicial desde donde se accederá a las diferentes funciones propuestas en el proyecto.
- Controlador: son las clases que intermediarán entre el modelo y la vista, para dar vida a nuestra aplicación. Será el centro el eje donde girará toda la funcionalidad de nuestra aplicación.

En cuanto a la estrategia elegida será la del desarrollo de un producto nuevo, orientado al usuario que lo utilizará. Más adelante se planteará la posibilidad de comercializarlo, con las oportunas correcciones y mejoras obtenidas por el testeado del producto inicial.

He decidido que la estrategia elegida sea así, por varios motivos:

- será una forma de testear en real un producto antes de salir al mercado y de forma gratuita.
- Podré realizar mejoras con el objetivo de un producto final sólido y adaptado al mercado al que se dirige.
- Servirá como auto aprendizaje para el proceso de creación de un app.

#### 1.4 Funcionalidades.

Las acciones que debería tener esta aplicación en su primera versión serían:

- *Parte diario de visitas*: registro diario de las visitas realizadas a los clientes, informando del cliente visitado, la gestión comercial realizada y el resultado de esta.
- *Ficha cliente*: donde mostrará los datos del cliente, con el histórico de visitas realizadas.
- *Propuesta de visitas*: está función presentará , en forma de listado, las visitas a realizar en la semana, en función de los parámetros que definamos previamente.

Y para versiones posteriores se incluirá:

- *Sincronización/registro en BD*: está función deberá conectar con una BD “oncloud” ,donde registrará todas las acciones realizadas por el usuario en ese día.
- *Parte de visitas acumulado*: está función mostrará la actividad realizada por el vendedor, en forma de “cuadro resumen”, mostrando datos como: número de visitas, periodicidad por cliente, campañas presentadas, ventas realizadas, etc.
- *Propuesta de ruta optima* :mostrará en forma de listado, las visitas a realizar en el día, pero con la premisa de realizar la ruta más corta posible.

### 1.5 Planificación del Trabajo.

La realización de la planificación está basada en el desarrollo de trabajo propuesto por las pecc. Por lo que agrupo cada una de las 4 entregas en fases de la planificación.

Detalle del cuadro de acciones a realizar y grafico de Gantt:

### **PLANIFICACIÓN POR DIAS**

<i>Id.</i>	<i>Nombre de tarea</i>	<i>Comienzo</i>	<i>Fin</i>	<i>Duración</i>
<b>1</b>	<b>PLAN DE TRABAJO</b>	<b>25/02/2015</b>	<b>11/03/2015</b>	<b>15d</b>
2	Contenido y justificación	25/02/2015	27/02/2015	3d
3	Objetivos del trabajo	28/02/2015	28/02/2015	1d
4	Enfoque y método seguido	01/03/2015	01/03/2015	1d
5	Funcionalidades	02/03/2015	04/03/2015	3d
6	Planificación del trabajo	05/03/2015	07/03/2015	3d
7	Sumario del producto obtenido	08/03/2015	10/03/2015	3d
8	Descripción de la memoria	11/03/2015	11/03/2015	1d
<b>9</b>	<b>ANÁLISIS, DISEÑO Y PROTOTIPO</b>	<b>12/03/2015</b>	<b>08/04/2015</b>	<b>28d</b>
10	Análisis Funcional	12/03/2015	19/03/2015	8d
11	Diseño Técnico	20/03/2015	23/03/2015	4d
12	Prototipo Vertical	24/03/2015	02/04/2015	10d
13	Evaluación del prototipo y corrección	03/04/2015	08/04/2015	6d
<b>14</b>	<b>IMPLEMENTACIÓN</b>	<b>09/04/2015</b>	<b>20/05/2015</b>	<b>42d</b>
15	Estructura de la aplicación	09/04/2015	18/04/2015	10d
16	Codificación y Testing	19/04/2015	20/05/2015	32d
<b>17</b>	<b>ENTREGA FINAL</b>	<b>21/05/2015</b>	<b>21/06/2015</b>	<b>32d</b>
18	Documentación Técnica	21/05/2015	01/06/2015	12d
19	Video presentación de la aplicación	02/06/2015	11/06/2015	10d
20	Memoria Final completa	12/06/2015	16/06/2015	5d
21	Resumen de memoria	17/06/2015	21/06/2015	5d

Id.	Nombre de tarea	Comienzo	Fin	Duración	feb 2015													
					24	25	26	27	28	1	2	3	4	5	6	7	8	9
1	<b>PLAN DE TRABAJO</b>	25/02/2015	11/03/2015	15d														
2	Contenido y justificación	25/02/2015	27/02/2015	3d														
3	Objetivos del trabajo	28/02/2015	28/02/2015	1d														
4	Enfoque y método seguido	01/03/2015	01/03/2015	1d														
5	Funcionalidades	02/03/2015	04/03/2015	3d														
6	Planificación del trabajo	05/03/2015	07/03/2015	3d														
7	Sumario del producto obtenido	08/03/2015	10/03/2015	3d														
8	Descripción de la memoria	11/03/2015	11/03/2015	1d														
9	<b>ANÁLISIS, DISEÑO Y PROTOTIPO</b>	12/03/2015	08/04/2015	28d														
10	Análisis Funcional	12/03/2015	19/03/2015	8d														
11	Diseño Técnico	20/03/2015	23/03/2015	4d														
12	Prototipo Vertical	24/03/2015	02/04/2015	10d														
13	Evaluación del prototipo y corrección	03/04/2015	08/04/2015	6d														
14	<b>IMPLEMENTACIÓN</b>	09/04/2015	20/05/2015	42d														
15	Estructura de la aplicación	09/04/2015	18/04/2015	10d														
16	Codificación y Testing	19/04/2015	20/05/2015	32d														
17	<b>ENTREGA FINAL</b>	21/05/2015	21/06/2015	32d														
18	Documentación Técnica	21/05/2015	01/06/2015	12d														
19	Video presentación de la aplicación	02/06/2015	11/06/2015	10d														
20	Memoria Final completa	12/06/2015	16/06/2015	5d														
21	Resumen de memoria	17/06/2015	21/06/2015	5d														

Id.	Nombre de tarea	Comienzo	Fin	Duración	mar 2015																											
					12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8
1	<b>PLAN DE TRABAJO</b>	25/02/2015	11/03/2015	15d																												
2	Contenido y justificación	25/02/2015	27/02/2015	3d																												
3	Objetivos del trabajo	28/02/2015	28/02/2015	1d																												
4	Enfoque y método seguido	01/03/2015	01/03/2015	1d																												
5	Funcionalidades	02/03/2015	04/03/2015	3d																												
6	Planificación del trabajo	05/03/2015	07/03/2015	3d																												
7	Sumario del producto obtenido	08/03/2015	10/03/2015	3d																												
8	Descripción de la memoria	11/03/2015	11/03/2015	1d																												
9	<b>ANÁLISIS, DISEÑO Y PROTOTIPO</b>	12/03/2015	08/04/2015	28d																												
10	Análisis Funcional	12/03/2015	19/03/2015	8d																												
11	Diseño Técnico	20/03/2015	23/03/2015	4d																												
12	Prototipo Vertical	24/03/2015	02/04/2015	10d																												
13	Evaluación del prototipo y corrección	03/04/2015	08/04/2015	6d																												
14	<b>IMPLEMENTACIÓN</b>	09/04/2015	20/05/2015	42d																												
15	Estructura de la aplicación	09/04/2015	18/04/2015	10d																												
16	Codificación y Testing	19/04/2015	20/05/2015	32d																												
17	<b>ENTREGA FINAL</b>	21/05/2015	21/06/2015	32d																												
18	Documentación Técnica	21/05/2015	01/06/2015	12d																												
19	Video presentación de la aplicación	02/06/2015	11/06/2015	10d																												
20	Memoria Final completa	12/06/2015	16/06/2015	5d																												
21	Resumen de memoria	17/06/2015	21/06/2015	5d																												



### 1.6 Descripción de los otros capítulos de la memoria.

En los siguientes apartados que se realizarán encontraremos:

- Diseño técnico del producto, en el que se incluirán la definición de clases, pantallas/interfaces de usuario y estructura del programa.
- Análisis funcional de cada una de las acciones que realizará el programa.
- Prototipado, en el que se mostrarán un primer boceto estructural del diseño de la interface.
- Implementación y testing, desarrollando el código fuente. Esta fase será testeada continuamente para el correcto funcionamiento del producto final.

A continuación describo los aspectos más relevantes que se considerarán para el diseño del producto:

Primero, debe ser un entorno amigable para el usuario final, y por ello, voy a utilizar exclusivamente los conceptos y terminología que ya conocía el usuario final del producto.

En segundo lugar, tengo presente los diferentes escenarios posibles: planificar la ruta de reparto/visitas, propuesta de visitas diarias, registro de las visitas realizadas, gestión de la ficha cliente y sincronización de la base de datos local

Una vez analizado los escenarios y los diferentes flujos de interacción generados se diseñará la principales interfaces de la aplicación. Primero en forma de bocetos, y luego, en un prototipo de alto nivel.

Finalmente, y una vez realizado el diseño técnico del producto, se testeará sobre el emulador, y posteriormente sobre los siguientes dispositivos:

- Tablet Samsung Galaxy Tab, de 10.1”.
- Móvil Xiaomi MI3, de 5”.
- Tablet Vexia de 7”.

### 1.7 Sumario de los productos obtenidos.

La entrega final del proyecto contará con una memoria en detalle, el código fuente de la aplicación y un video demostración.

Obtendremos una app android, enfocado principalmente a tablets con pantalla de 7", pero será compatible para móviles a partir de 5". La aplicación podrá ser instalada de forma manual, inicialmente.

Una vez testada se colgará en el market de Android, para su descarga. Será gratuita en su versión elemental, y de pago, para la versión Premium, que tendrá funciones adicionales y actualización de mejora.

## 2. Diseño centrado en usuario

---

### 2.1 Usuarios y Contexto de la aplicación.

#### 2.1.1 Usuarios.

El conocimiento sobre las personas o usuarios, y el entorno donde van a utilizar la aplicación, son imprescindibles para un diseño centrado en el usuario.

Para esta fase he utilizado como método de indagación, la observación e investigación contextual. Consiste en la observación de los usuarios en su entorno habitual, conociendo así, de manera objetiva qué hacen los usuarios, y en qué condiciones lo hacen.

En el desarrollo de la investigación, he ido anotando las diferentes tareas que realizan en una jornada de trabajo de las personas que finalmente serán los usuarios de la aplicación.

Muestro, en síntesis, la secuencia de acciones realizada por los vendedores en el cuadro anexo:

Nº SECUENCIA	DESCRIPCIÓN
1	Recepción, en almacén, de mercancía pedida a los proveedores.
2	Organización de la mercancía en función del cliente que la ha solicitado.
3	Preparación de ruta de reparto/visita para dejar la pieza solicitada.
4	Salida a reparto/visita.
5	Llegada a punto de destino, dejando la mercancía solicitada si la hubiera.

6	Presentación de campañas comerciales del mes.
7	Desarrollo de la visita
8	Cierre de la visita
9	Anotación manuscrita de la visita y los resultados obtenidos, en papel
10	Se repite la secuencia 5-9, hasta finalización del reparto.
11	Final de reparto y regreso a almacén.
12	Registro en Hoja de Cálculo, de toda la información recopilada en la visitas.

A continuación detallo la información que registran, obtenida en el punto 12:

-Datos de la visita: Número de cliente, Razón Social, Persona de Contacto, Teléfono, Tipo de Visita (Reparto, Cortesía, Prospección, Presentación Campaña) y Resultado de la Visita (Reparto, Venta, Cliente Nuevo, Otros).

-Datos genéricos: N° de la semana, Fecha, Vendedor.

Otros aspectos importantes a destacar son:

-El ritmo de trabajo, de forma general, es bastante alto.

-En entorno donde se desarrolla su actividad es exterior, donde la mayor parte del tiempo se pasa en un vehículo.

Por otro lado, hemos identificado 2 perfiles de usuario, que describimos en las siguientes fichas:

*Perfil 1: Vendedor/Repartidor*

*Personas de edades comprendidas entre 30-45 años, residentes en ciudad, urbanitas, con experiencia en el uso de dispositivos móviles ya que todos tiene móvil de gama media-alta, y nivel de estudios medios. Con intereses diversos pero con facilidad de aprendizaje.*

*Perfil 2: Responsable Ventas.*

*Persona de 52 años de edad, residentes en ciudad, con experiencia en el uso de tecnología, y con nivel de estudios superiores. Reacio a cambios pero con facilidad de aprendizaje.*

### 2.1.2 Contexto de uso de la aplicación.

Para el perfil 1, será un entorno exterior la mayor parte del tiempo, el cual se repartirá equitativamente entre talleres y el vehículo de transporte. Y de uso continuo, porque deberá registrar cada visita una vez finalizada.

Para el perfil 2, será un entorno de oficina, en interior, donde estará conectado a Internet, con un ordenador de sobremesa.

Seguidamente, detallo el conjunto de tareas que los usuarios necesitarán para alcanzar sus objetivos con la aplicación. Para el perfil 1:

1. Crear, editar, mostrar y eliminar visitas /repartos.
2. Búsqueda de visitas.
3. Propuesta automática de visitas.
4. Buscar ficha de cliente.
5. Sincronización de datos con BD.

Y para el perfil 2, además de las ya incluidas en el perfil 1, también debería tener las siguientes tareas:

6. Búsqueda del parte acumulado de visitas por vendedor.
7. Visualización del parte de visitas acumulado por vendedor.
8. Crear, editar, mostrar y eliminar ficha de cliente.

La conclusión de esta fase es que debemos de crear una aplicación:

- Rápida, clara e intuitiva, debido a que el ritmo de trabajo así lo demanda.
- La inserción de datos deberá ser lo más escueta posible, con el fin de ahorrar tiempo en el trabajo para el usuario que está en el exterior.
- Los colores utilizados deberán facilitar la visibilidad porque la aplicación se usará principalmente en el exterior.
- Todo el desarrollo estará enfocado al diseño centrado en el usuario.
- No incluir todas las funciones en la app directamente, con la finalidad de mejorar su funcionalidad. Solo se incluirán las estrictamente necesarias para el funcionamiento diario del vendedor. He observado que incluir las altas de cliente por la tablet, es demasiado lento, incumpliendo el objetivo de usabilidad y rapidez que buscamos en la ejecución de tareas.

- Existirán 2 versiones de la aplicación, normal y profesional. La normal, será gratuita e incluirá las funciones del perfil 1. Y la profesional, que contendrá la totalidad de las funciones y tendrá un coste económico.

## 2.2 Diseño Conceptual.

En este apartado determinamos las necesidades de los usuarios, desde su óptica, elaborando así un diseño de producto. Enfocado a una aplicación móvil.

Y por otro lado, se describirán los flujos de interacción dispositivo-usuario mostrando gráficamente la estructura general de la aplicación.

### 2.2.1 Escenario de uso.

A continuación detallo los posibles escenarios de uso que podremos encontrar en la práctica habitual de la aplicación:

#### **Escenario 1: planificar ruta reparto.**

Perfil Usuario	Vendedor /Repartidor
Contexto	En interior, dentro del almacén
Objetivo	Planificar la ruta de repartos
Tarea	Creación y edición de repartos, para incluirlo en el flujo de visitas diario.
Necesidad Información	Saber los repartos del día.
Funcionalidades	Localizar los albaranes diarios de reparto y organizar su ruta.
Desarrollo de tarea	Abrir la aplicación, pulsar sobre “reparto”, clicamos sobre crear y mostrará plantilla de generación de reparto, introducimos datos necesarios y guardamos.

#### **Escenario 2: planificar visita manual.**

Perfil Usuario	Vendedor/Repartidor
Contexto	En interior, dentro del almacén
Objetivo	Planificar la ruta de visitas comerciales.
Tarea	Creación y edición de visitas, para incluirlo en el flujo de visitas diario.
Necesidad Información	Conocer los clientes que necesita visitar según el criterio del vendedor.
Funcionalidades	Mirar que clientes lleva tiempo sin visitar o conviene visitar para desplegar campaña comercial.
Desarrollo de tarea	Abrir la aplicación, pulsar sobre “visitas”, clicamos sobre crear, mostrará plantilla de generación de visita, introducimos datos necesarios y guardamos.

### Escenario 3: propuesta automática de visita.

Perfil Usuario	Vendedor / Repartidor
Contexto	En interior, dentro del almacén
Objetivo	Generar la ruta de visitas comerciales.
Tarea	Generar la ruta de visitas/reparto, mostrando las visitas según criterio seleccionado por el sistema.
Necesidad Información	Conocer la ruta de visitas y repartos del día de trabajo.
Funcionalidades	Generación automática de la ruta del día de trabajo.
Desarrollo de tarea	Abrir la aplicación, pulsar sobre " <i>Planing diario de visitas</i> ", mostrará listado de propuestas de visitas según criterio elegido, y realizamos el registro del parte diario de visitas. Esta propuesta aparecerá en el planing diario de visitas como "realizada".

### Escenario 4: registro del parte diario de visitas.

Perfil Usuario	Vendedor/Repartidor.
Contexto	En el exterior, dentro del automóvil o en el taller del cliente.
Objetivo	Registrar la visita/reparto realizado.
Tarea	Registro de datos obtenidos como resultado de la visita realizada.
Necesidad Información	Conocer el desarrollo y resultado final de la visita.
Funcionalidades	Introducir los datos obtenidos en la visita.
Desarrollo de tarea	Dentro de " <i>Planing diario de visitas</i> ", seleccionamos la visita realizada, introducimos datos solicitados de la visita y guardamos cambios.

### Escenario 5: gestión de la ficha cliente.

Perfil Usuario	Vendedor/Repartidor o Responsable de ventas.
Contexto	En el interior principalmente, en oficina.
Objetivo	Gestionar ficha de cliente nuevo/antiguo.
Tarea	Crear, modificar o eliminar cliente.
Necesidad Información	Datos del cliente a crear, modificar o eliminar.
Funcionalidades	Introducir los datos obtenidos.
Desarrollo de tarea	Abrir la aplicación, pulsar sobre " <i>cliente</i> ", buscamos el cliente a gestionar o creamos uno nuevo si esa es la finalidad, introducimos los datos necesarios y guardamos cambios.

### Escenario 6: sincronización con la BD.

Perfil Usuario	Vendedor/Repartidor o Responsable de ventas.
Contexto	En el interior, en oficina.
Objetivo	Actualizar las visitas y fichas de clientes, transcurridos en el día de trabajo.
Tarea	Actualizar información de la base de datos.
Necesidad Información	Tener Wifi con conexión a Internet.
Funcionalidades	Actualizar base de datos
Desarrollo de tarea	Abril la aplicación, conectarse a Internet sobre wifi, pulsar sobre "actualizar datos" de la aplicación y esperar respuesta del SO confirmando que la actualización se ha realizado correctamente.

**Escenario 7: Gestión del parte de visitas acumulado.**

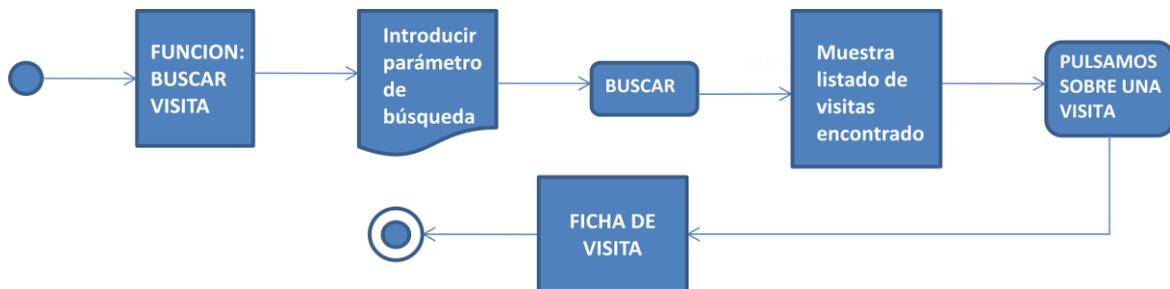
Perfil Usuario	Responsable de ventas.
Contexto	En el interior, en oficina.
Objetivo	Medir el rendimiento del vendedor
Tarea	Analizar los datos acumulados del vendedor, tanto en visitas como en campaña y su efectividad.
Necesidad Información	Parte diario de visitas registradas por el vendedor en el mes.
Funcionalidades	Analizar los datos obtenidos.
Desarrollo de tarea	En el menú inicio del programa pulsaremos sobre búsqueda de la ficha del vendedor, filtrar datos mensuales y analizar resultados.

2.2.2 Flujos de interacción.

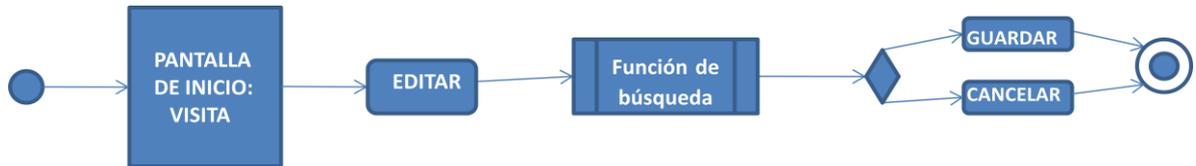
**CREAR UNA VISITA/REPARTO.** En el menú de inicio encontraremos varias opciones, seleccionamos VISITA/REPARTO, aparecerá una pantalla con las opciones crear, editar, mostrar o eliminar, seleccionamos CREAR, nos aparecerá una ficha de visita nueva en la que introduciremos los datos solicitados, una vez realizada la inserción de datos tendremos la opción de guardar la ficha de visita o salir de la función.



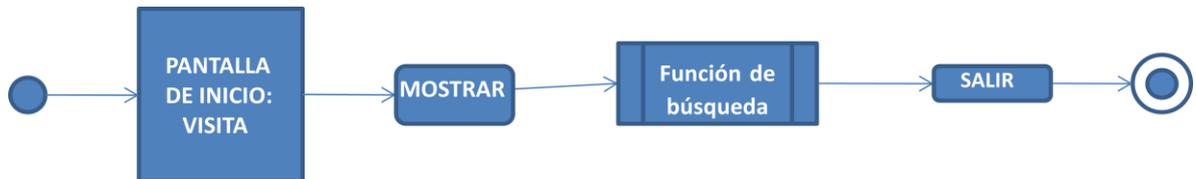
**BUSCAR UNA VISITA/REPARTO:** Esta función tiene acceso indirecto, siempre a través de buscar, editar o eliminar visita. Introduciremos los parámetros de búsqueda por semana o día, nos mostrará un listado con las visitas encontradas, clicaremos sobre una de las visitas y nos mostrará la Ficha de Visita.



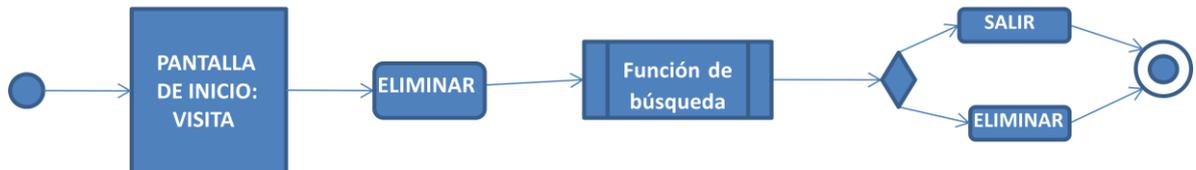
**EDITAR UNA VISITA/REPARTO:** En el menú de inicio encontraremos varias opciones, seleccionamos VISITA, aparecerá una pantalla con las opciones crear, editar, mostrar o eliminar, seleccionamos EDITAR y ejecutará la secuencia de búsqueda mostrando finalmente la ficha de visita a editar, modificaremos los datos oportunos y guardaremos los cambios realizados.



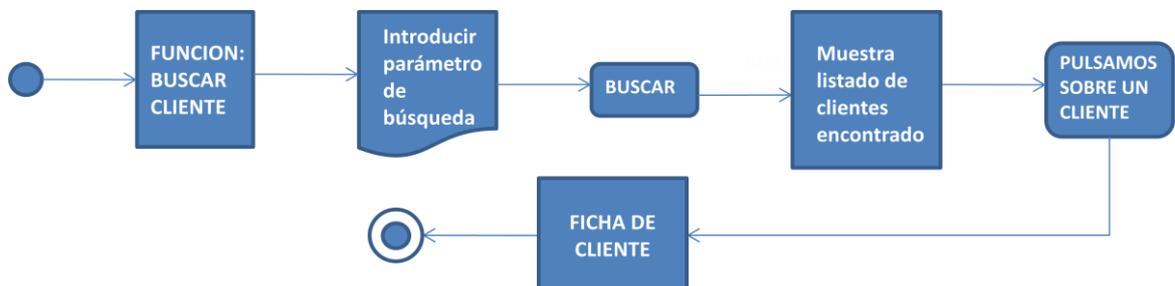
**MOSTRAR UNA VISITA/REPARTO:** En el menú de inicio encontraremos varias opciones, seleccionamos VISITA, aparecerá una pantalla con las opciones crear, editar, mostrar o eliminar, seleccionamos MOSTRAR y ejecutará la secuencia de búsqueda mostrando finalmente la ficha de visita, una vez visualizada saldremos de la ficha.

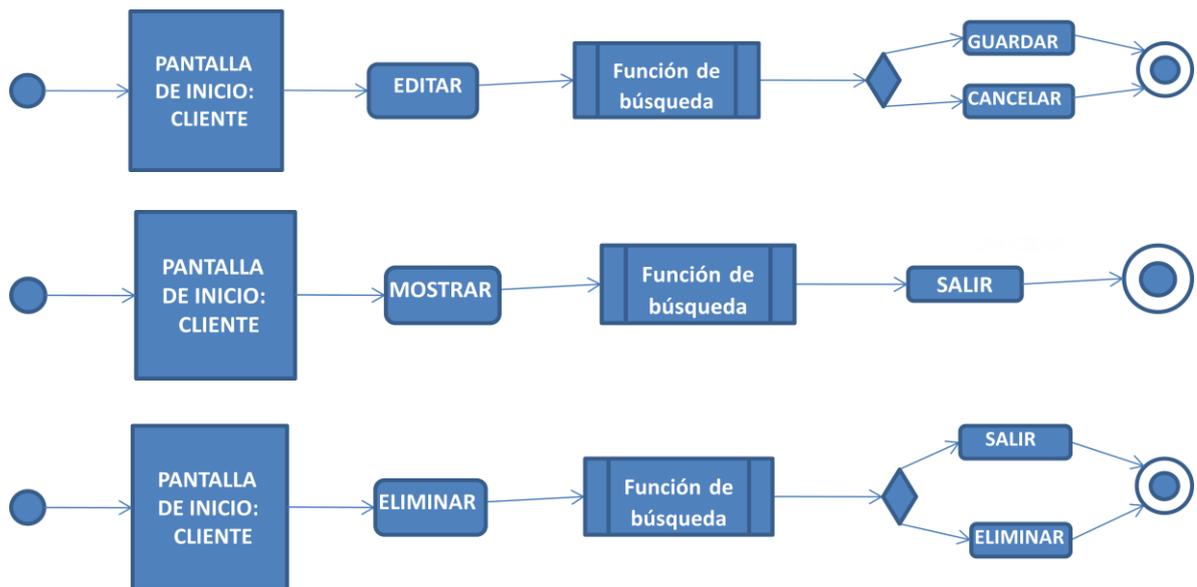


**ELIMINAR UNA VISITA/REPARTO:** En el menú de inicio encontraremos varias opciones, seleccionamos VISITA, aparecerá una pantalla con las opciones crear, editar, mostrar o eliminar, seleccionamos ELIMINAR y ejecutará la secuencia de búsqueda mostrando finalmente la ficha de visita, pulsaremos la opción de “eliminar visita”, nos aparecerá un texto flotante indicando que la visita ha sido eliminada y saldrá al menú de inicio.

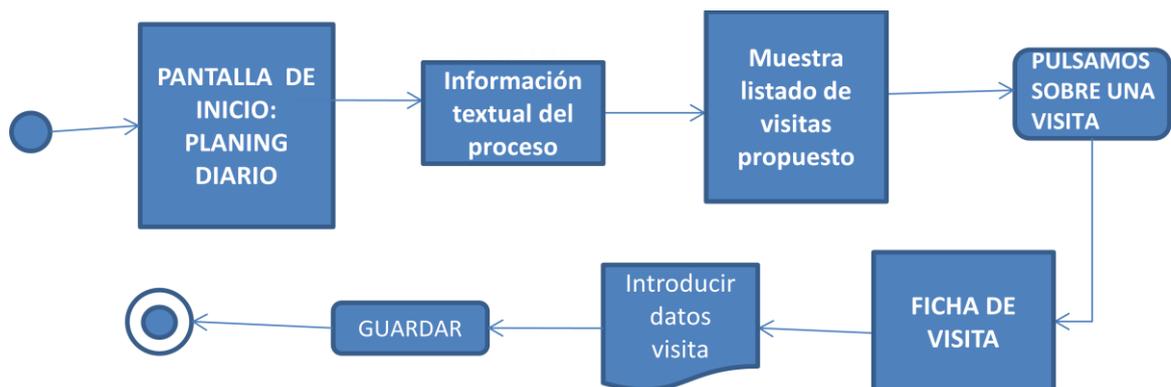


**CREAR, BUSCAR, EDITAR, MOSTRAR Y ELIMINAR CLIENTE:** la secuencia de acciones para estas funciones serán las mismas que las detalladas para las acciones de Visitas, por lo que obviaré la descripción de las acciones a desarrollar con el objetivo de hacer más amena la lectura. Estas funciones estarán disponibles para la versión profesional, excluyendo la función de búsqueda que si estará en la versión normal.





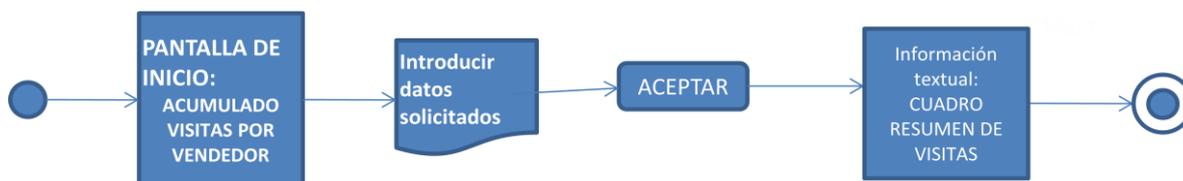
**PLANING DIARIO:** En el menú de inicio encontraremos varias opciones, seleccionamos PLANING DIARIO, nos aparecerá un texto flotante sobre la pantalla comunicando “generando planificación de visitas....Espere”. Una vez finalizado, aparecerá un listado de visitas a realizar, presionaremos sobre una de ella y mostrará la ficha de visitas a editar, introducimos los datos oportunos y guardamos cambios. Luego volverá al listado de visitas a realizar. La visita realizada ya no aparecerá en el listado.



**SINCRONIZACIÓN BASE DATOS:** En el menú de inicio encontraremos varias opciones, seleccionamos SINCRONIZACIÓN BD. Nos aparecerá una pantalla informativa comunicando que vamos a proceder a sincronizar datos con la BD central, pulsamos “sincronizar”. Aparecerá ventana flotante indicando que el proceso está en curso. Finalmente mostrará por pantalla que la sincronización es correcta o ha existido algún error en el proceso.

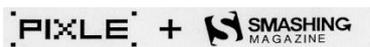


**BUSQUEDA DEL PARTE DE VISITAS ACUMULADO POR VENDEDOR:** En el menú de inicio encontraremos varias opciones, seleccionamos ACUMULADO VISITAS VENDEDOR, nos solicitará el vendedor y el periodo por meses a mostrar, pulsaremos “aceptar” y visualizará por pantalla los indicadores más relevantes del trabajo realizado por el vendedor. Esta función solo estará disponible para la versión profesional.



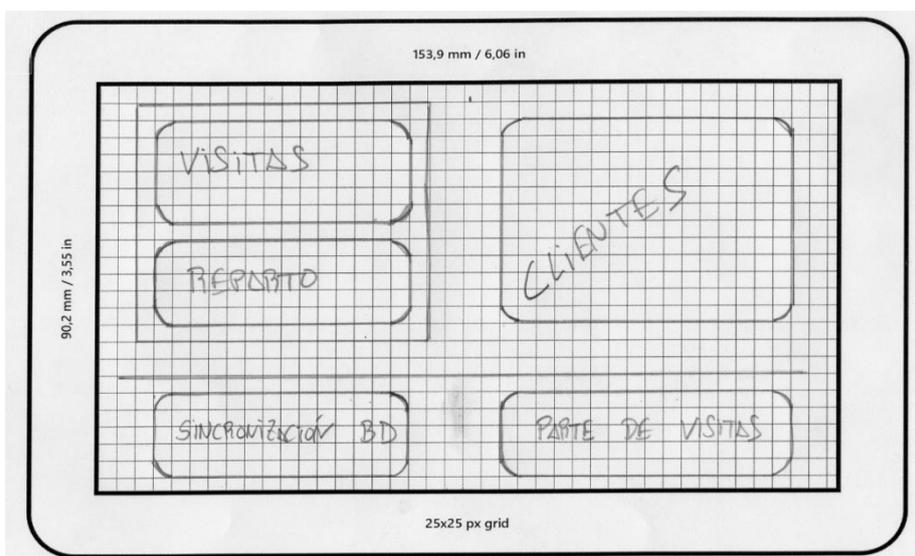
### 2.3 Prototipado.

Inicialmente expongo los primeros bocetos / sketches de la aplicación. Muestro las pantallas principales de cada una de las acciones detalladas en el apartado anterior. Se ha tomado como patrón de dibujo una de las plantillas recomendadas en la pec, concretamente:



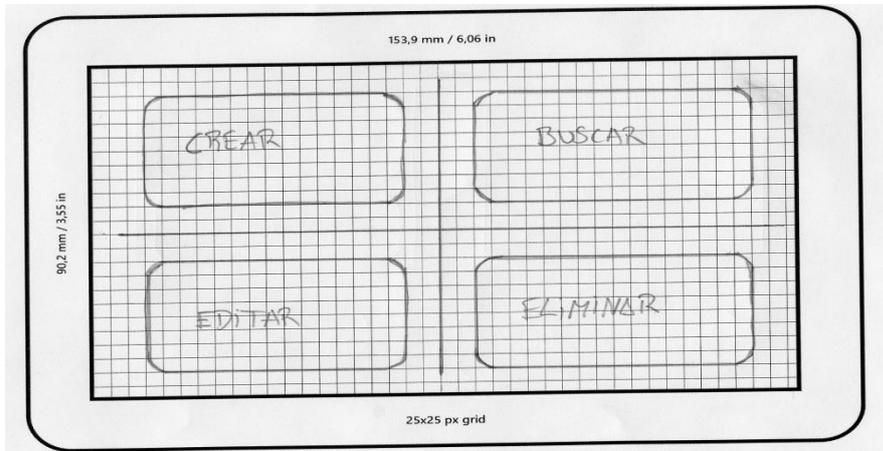
#### Pantalla inicial:

Esta sería la pantalla de inicio una vez ejecutamos la aplicación. Muestra las diferentes opciones que podemos ejecutar.



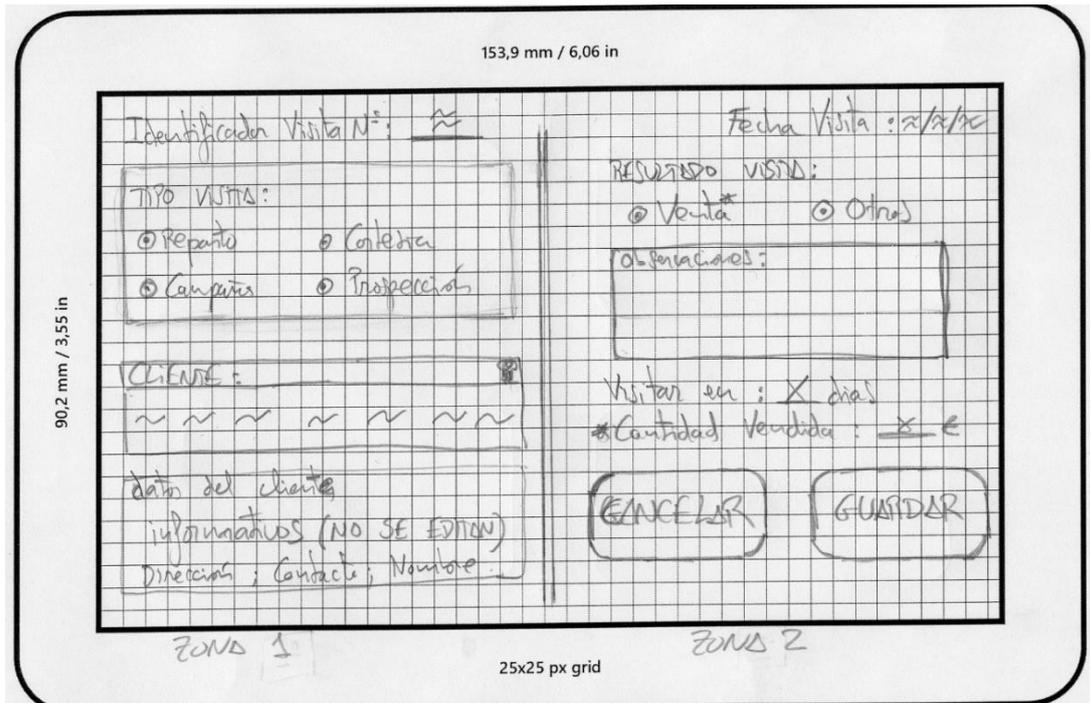
Pantalla intermedia de opciones en ficha de visitas, reparto y cliente:

Una vez hemos clicado en cualquiera de las opciones de "Cliente", "Vistas" o "Reparto", de la pantalla inicial, nos mostrará que deseamos hacer dentro de la opción elegida de la pantalla inicial.



Pantalla de la ficha visita / reparto:

Esta plantilla muestra la ficha de Visitas que tendrá que cumplimentar el vendedor en su labor diaria de visitas y/o repartos, una vez que ha pulsado la opción Visita → Crear o Editar:



### Pantalla de la ficha cliente:

Muestra la ficha de cliente, con información a cumplimentar. La secuencia de acciones sería Cliente → Crear o Editar.

90,2 mm / 3,55 in

**FICHA CLIENTE**

Id. Cliente : \_\_\_\_\_ Fecha Alta : \_\_\_\_\_

Partida Social : \_\_\_\_\_

CIF/NIF : \_\_\_\_\_

Domicilio : \_\_\_\_\_

Ciudad : \_\_\_\_\_

Provincia : \_\_\_\_\_

Persona Contacto : \_\_\_\_\_

Teléfono : \_\_\_\_\_

Móvil : \_\_\_\_\_

Puerto : \_\_\_\_\_

CANCELAR      GUARDAR

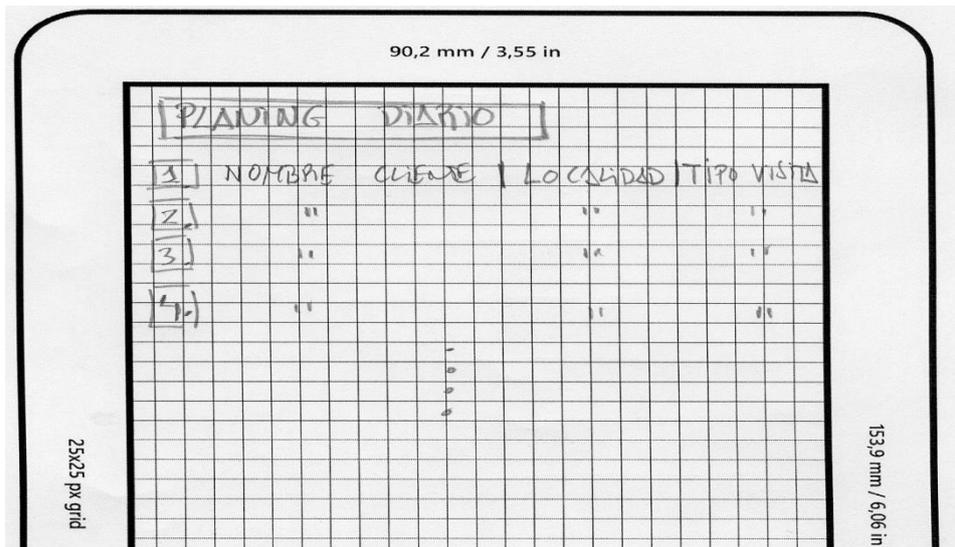
25x25 px grid      153,9 mm / 6,06 in

### Pantalla del planing diario de visitas:

Esta acción se realizará cuando se hayan completado las acciones de reparto y visitas. Una vez ejecutada el sistema mostrará un listado de visitas-repartos a realizar teniendo en cuenta las siguientes prioridades:

- En primer lugar mostrará las visitas de tipo reparto.
- En segundo lugar las creadas por nosotros ese día o días anteriores que no han sido realizadas.
- En último lugar las propuestas por el sistema, que se parametrizarán a través del menú de configuración, indicado por el valor de frecuencia de visita.

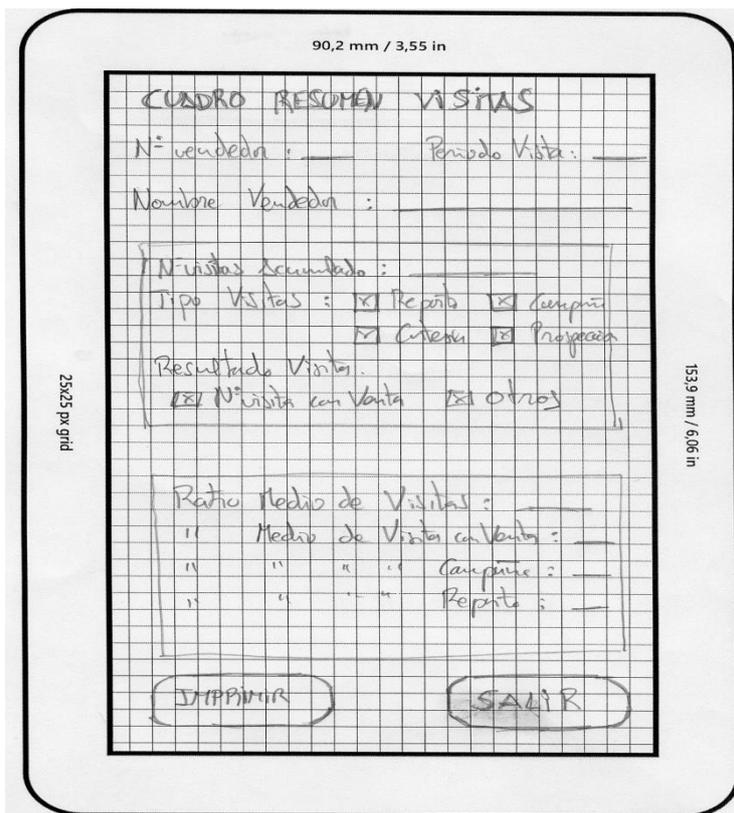
Una vez en el layout, clicaremos sobre un ítem del listado y nos mostrará la ficha de visita a completar.



Pantalla del cuadro resumen de “visitas por vendedor” :

Se accede a través del menú inicio del programa, donde aparecerá la configuración y el acumulado de visitas por vendedor. Pulsaremos sobre ella y nos aparecerá una pantalla indicándonos el vendedor a mostrar, y a continuación, mostrará sus resultados para poder analizarlos.

Esta pantalla solo estará disponible en la versión profesional.



## **PROTOTIPADO DE ALTO NIVEL**

A continuación muestro las pantallas descritas anteriormente en un prototipo de alto nivel, realizado con la plataforma de desarrollo Android Studio:

### **Pantalla Inicial**



### **Pantalla Intermedia**



### Planing diario de visitas



### Ficha de visitas



**Pantalla ficha de clientes**

**GestorVisitas**

Numero de Cliente: 48  
Fecha Alta: 20/10/2014

Razon Social: Taller El tuercas SL  
Cif/Nif : E2895967  
Domicilio: Av Fuerzas Armadas s/n  
Localidad: Utrera CP: 48558  
Provincia: Sevilla

Persona Contacto: Name  
Telefono: 959558877  
Mail: \_\_\_\_\_  
Puesto: \_\_\_\_\_

CANCELAR GUARDAR

**Pantalla acum. visitas vendedor**

**GestorVisitas**

**RESUMEN DE VISITAS POR VENDEDOR**

Numero Vendedor: 4  
Nombre : AutoCompleteTextView

**Nº visitas total : 38**  
**Desglose por tipo:**  
\* Reparto = 11 \* Campañas = 25  
\* Cortesia = 12 \* Prospeccion = 17

**Resultado Visitas:**  
\* N° visitas con Venta = 21  
\* N° visitas Otros = 17

**RATIOS MEDIOS DEL VENDEDOR :**  
\* Reparto = 1.1  
\* Cortesia = 1.6  
\* Campañas = 0.8  
\* Prospeccion = 1.5  
\* N° visitas Otros = 1.0  
\* N° visitas con Venta = 0.6

ACEPTAR

## 2.4 Evaluación.

Para realizar la evaluación se han expuesto a los integrantes del grupo de ventas las diferentes pantallas del prototipo explicándole brevemente su finalidad, e indicándoles que explicasen como realizarían las diferentes tareas descritas en el punto de “Tareas a realizar”.

Queda pendiente de analizar el menú de inicio con la configuración del sistema y el acumulado de visitas por vendedor, que se incluirá en la implementación, debido a que en el prototipado no se ha podido valorar.

A continuación describo las diferentes cuestiones analizadas.

### **1. Preguntas del test de evaluación del usuario:**

- ¿Cuál es tu nombre y apellidos?
- ¿Cuántos años tienes?
- ¿Cómo valorarías tu nivel en tecnología?
- ¿Tienes teléfono de última generación / Tablet? ¿Con que frecuencia la usas?
- ¿Crees que te facilitaría el trabajo un sistema informatizado de tu actividad?  
¿Por qué?

### **2. Tareas a realizar.**

- Arrancar la aplicación.
- Registrar una nueva visita.
- Modificar o anular una visita.
- Registrar un cliente nuevo.
- Modificar o anular una ficha de cliente.
- Realizar una búsqueda de visita.
- Realizar una búsqueda de cliente.
- Actualizar información con la base de datos.
- Visualizar un cuadro resumen de visitas por vendedor.
- Configurar los parámetros del sistema

### **3. Preguntas referentes de las tareas a realizar.**

- ¿Cómo calificarías el nivel de dificultad para realizar la tarea A? Valoración del 1 al 10.
- ¿Cómo crees que debería ser según tu criterio? ¿Te sería más fácil así?

- ¿Hechas en falta alguna función más con el objetivo de agilizar tu trabajo diario? ¿Cuál / es?
- ¿Cómo valoras la aplicación de forma global?
- ¿Utilizarías esta aplicación en tu trabajo diario?

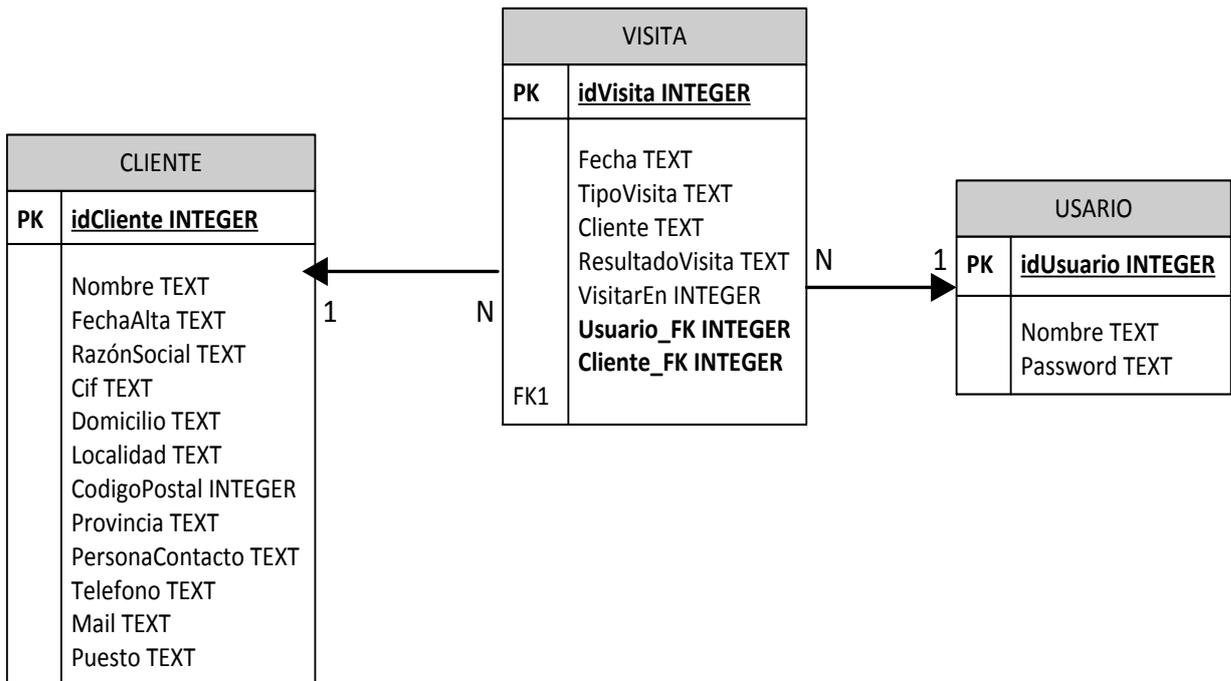
#### **4. Conclusiones y mejoras a realizar.**

- a) Se ha optado por visualizar todas las pantallas verticalmente, porque así facilitamos la comprensión al usuario.
- b) Cambio de orden de la pantalla inicial, poniendo la secuencia natural de las funciones realizadas en el trabajo diario.
- c) Cambio de CheckBox en vez de RadioButton, porque los tipos de visitas no son excluyentes. Por ejemplo, se puede hacer un reparto y presentar una campaña.
- d) Y como punto de mejora pendiente de realizar, establecer en los índices del cuadro resumen un sistema de colores indicando si el índice es bueno (verde), normal (amarillo), malo (rojo).
- e) La terminología usada se entiende bien, por ser la utilizada en el ámbito de su trabajo.
- f) Mejora del entorno sensorial de la aplicación con el objetivo de hacerla más atractiva, incluyendo icono de la app, dibujos explicativos en la pantalla inicial y color en el resto de pantallas siguiendo una mismo patrón visual.
- g) Como sugerencia de mejoras en posibles versiones futuras:
  - Enlace con navegador para encontrar la dirección del cliente estableciendo la ruta a seguir.
  - Mostrar ruta de visitas/reparto optima en función del tiempo y los kilómetros a realizar.

## 3. Diseño técnico

### 3.1 Diagrama Entidad- relación de la base de datos.

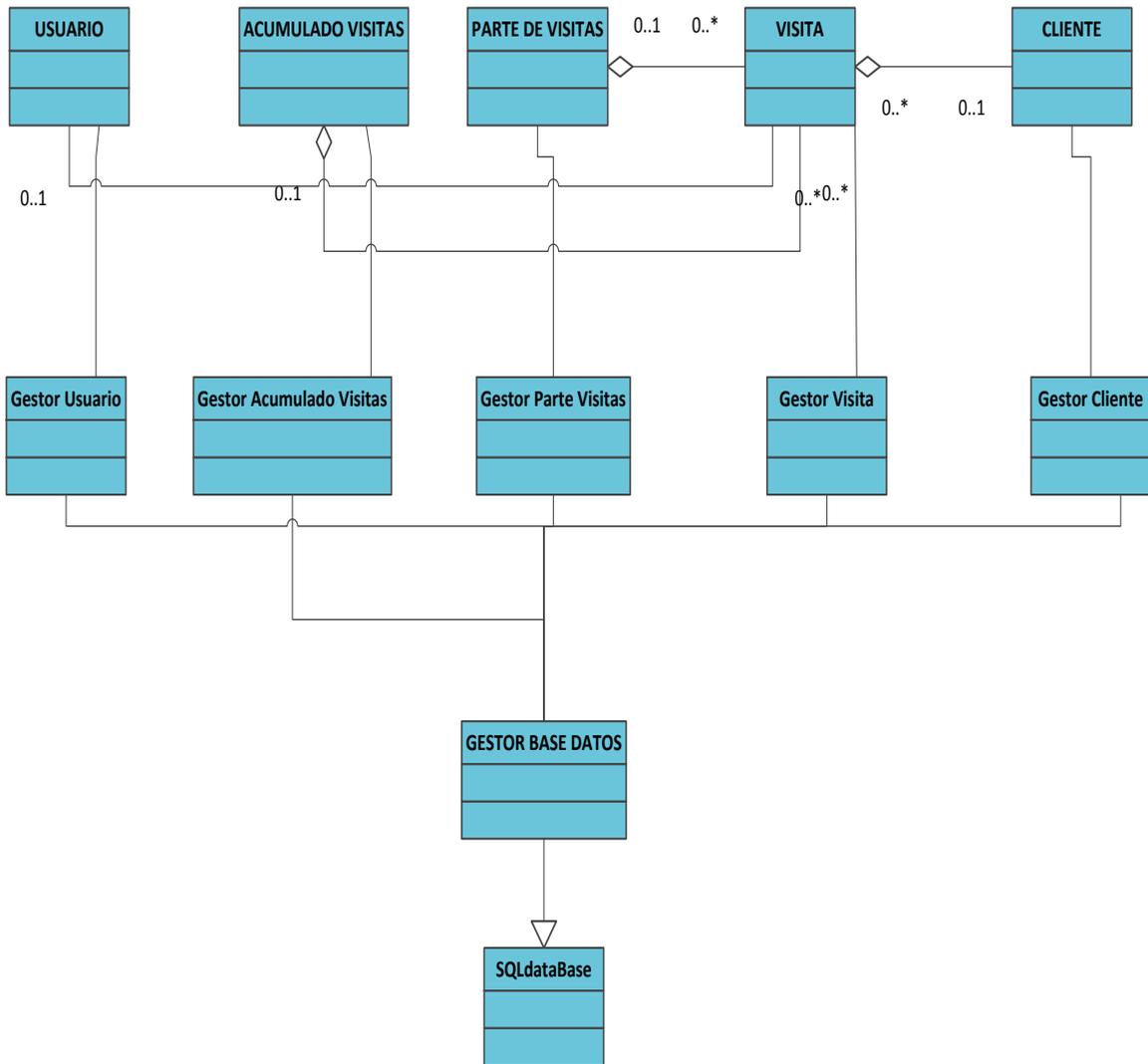
A continuación muestro las tablas a crear en la base de datos que utilizaremos. He considerado 3 tablas, ya que los objetos de “Resumen de Visitas” y “Planing diario”, se generarían a partir de los datos obtenidos de las tablas siguientes:



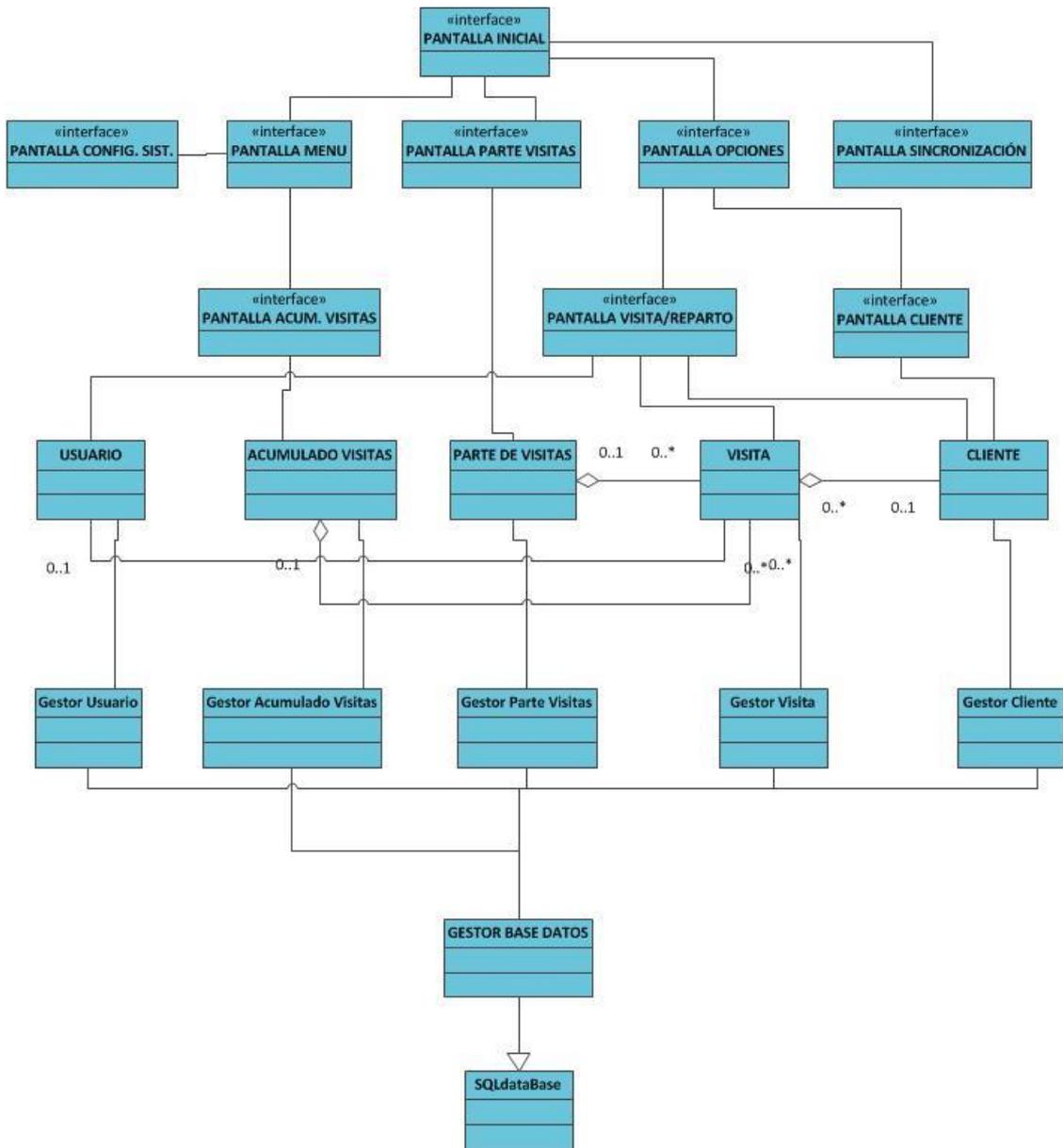
La clave primaria de cada tabla la indico en negrita y subrayada. Y las indicadas como (FK), son claves foráneas.

### 3.2 Diagrama UML de clases.

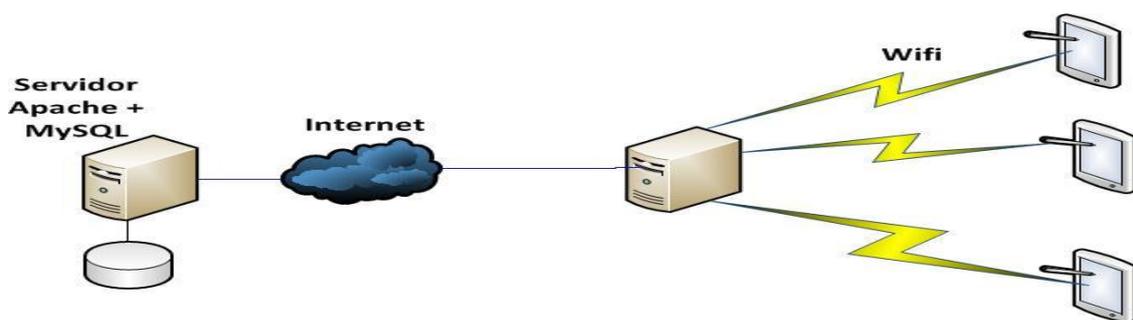
A continuación muestro las clases relacionadas con el modelo expuesto:



Seguidamente expongo el diagrama completo de clases que tendrá la aplicación. En este diagrama se muestra la totalidad de clases, incluyendo las interfaces y las clases gestoras de la aplicación:



3.3 Diagrama explicativo de la arquitectura del sistema.



La base de datos del sistema vendrá soportada a través de la plataforma XAMPP, que estará conectada a internet. Esta base de datos se hará sobre MySQL, con servidor Apache.

Las tablets almacenarán los datos en local, y cuando se conecten a una red wifi con conexión a internet, sincronizarán los datos almacenados en local, si procede.

### 3.4 Casos de uso.

<b>Identificador</b>	<b>CU-001</b>
<b>Nombre</b>	<b>CREAR VISITA.</b>
<b>Prioridad</b>	Alta
<b>Descripción</b>	Hacer el registro de una visita o reparto, cumplimentando la pantalla de visitas.
<b>Actores</b>	Vendedor de piezas de recambio
<b>Precondiciones</b>	Estado de la aplicación en ejecución, con la actividad de la pantalla inicial activa.
<b>Iniciado por</b>	Botón “reparto” o “visitas”, de la pantalla inicial.
<b>Flujo</b>	Usuario: Pulsa sobre botón “reparto” o “visita” → Usuario: pulsa botón “crear” → Sistema: lanza pantalla visita → Usuario: cumplimenta los datos solicitados. (Si el botón pulsado es “reparto”, vendrá preseleccionado en tipo de visita) y pulsa botón “guardar”. → Sistema: informa, a través de una ventana emergente, si a finalizado o no correctamente la función, y saldrá a la pantalla principal.
<b>Post condiciones</b>	Estado de la aplicación en ejecución, con la actividad de la pantalla inicial activa. Y registro de la visita creado.
<b>Notas:</b> sin comentarios	

<b>Identificador</b>	<b>CU-002</b>
<b>Nombre</b>	<b>BUSCAR VISITA.</b>
<b>Prioridad</b>	Alta
<b>Descripción</b>	Localizar y visualizar una visita creada en la base de datos.
<b>Actores</b>	Vendedor o Responsable de ventas.

<b>Precondiciones</b>	Estado de la aplicación en ejecución, con la actividad de la pantalla inicial activa. Y visita creada en la base de datos.
<b>Iniciado por</b>	Botón “reparto” o “visitas”, de la pantalla inicial.
<b>Flujo</b>	Usuario: Pulsa sobre botón “reparto” o “visita”→ Usuario: pulsa botón “buscar” → Sistema: lanza pantalla de búsqueda_visita → Usuario: cumplimenta los datos solicitados y pulsa botón “buscar”. → Sistema: muestra lista de elementos encontrados → Usuario: pulsa sobre la visita elegida → Sistema: lanza pantalla visita, con la visita seleccionada → Usuario: visualiza datos elegidos y pulsa “salir”.
<b>Post condiciones</b>	Estado de la aplicación en ejecución, con la actividad de la pantalla inicial activa.
<b>Notas:</b> la búsqueda de visita, se realizará por fecha / cliente . Es la forma más práctica y fácil de poder recordar una visita para el usuario. Por id no sería factible.	

<b>Identificador</b>	<b>CU-003</b>
<b>Nombre</b>	<b>EDITAR VISITA.</b>
<b>Prioridad</b>	Alta
<b>Descripción</b>	Localizar y modificar una visita creada en la base de datos.
<b>Actores</b>	Vendedor o Responsable de ventas.
<b>Precondiciones</b>	Estado de la aplicación en ejecución, con la actividad de la pantalla inicial activa. Y visita creada en la base de datos.
<b>Iniciado por</b>	Botón “reparto” o “visitas”, de la pantalla inicial.
<b>Flujo</b>	Usuario: Pulsa sobre botón “reparto” o “visita”→ Usuario: pulsa botón “buscar” → Sistema: lanza pantalla de búsqueda_visita → Usuario: cumplimenta los datos solicitados → Usuario: pulsa botón “buscar”. → Sistema: muestra lista de elementos encontrados → Usuario: pulsa sobre la visita elegida → Sistema: lanza pantalla visita, con la visita seleccionada → Usuario: modifica los datos mostrados y pulsa “guardar” →

	Sistema: informa, a través de una ventana emergente, si a finalizado o no correctamente la función, y saldrá a la pantalla principal.
<b>Post condiciones</b>	Estado de la aplicación en ejecución, con la activity de la pantalla inicial activa. Y el registro de la visita modificado.
<b>Notas:</b> la búsqueda de visita, se realizará por fecha / cliente . Es la forma más práctica y fácil de poder recordar una visita para el usuario. Por id no sería factible.	

<b>Identificador</b>	<b>CU-004</b>
<b>Nombre</b>	<b>ELIMINAR VISITA.</b>
<b>Prioridad</b>	Alta
<b>Descripción</b>	Localizar y eliminar una visita creada en la base de datos.
<b>Actores</b>	Vendedor o Responsable de ventas.
<b>Precondiciones</b>	Estado de la aplicación en ejecución, con la activity de la pantalla inicial activa. Y visita creada en la base de datos.
<b>Iniciado por</b>	Botón “reparto” o “visitas”, de la pantalla inicial.
<b>Flujo</b>	Usuario: Pulsa sobre botón “reparto” o “visita”→ Usuario: pulsa botón “buscar” → Sistema: lanza pantalla de búsqueda_visita → Usuario: cumplimenta los datos solicitados y pulsa botón “buscar”. → Sistema: muestra lista de elementos encontrados → Usuario: pulsa sobre la visita elegida → Sistema: lanza pantalla visita, con la visita seleccionada → Usuario: pulsa botón “eliminar” → Sistema: informa, a través de una ventana emergente, si a finalizado o no correctamente la función, y saldrá a la pantalla principal.
<b>Post condiciones</b>	Estado de la aplicación en ejecución, con la activity de la pantalla inicial activa. Y el registro de la visita eliminado.
<b>Notas:</b> la búsqueda de visita, se realizará por fecha / cliente . Es la forma más práctica y fácil de poder recordar una visita para el usuario. Por id no sería factible.	

<b>Identificador</b>	<b>CU-005</b>
<b>Nombre</b>	<b>MOSTRAR PLANING DE TRABAJO.</b>
<b>Prioridad</b>	Alta
<b>Descripción</b>	Visualiza las visitas y repartos pendientes de realizar por el vendedor.
<b>Actores</b>	Vendedor
<b>Precondiciones</b>	Estado de la aplicación en ejecución, con la actividad de la pantalla inicial activa. Y visitas creadas, sin cerrar, en la base de datos.
<b>Iniciado por</b>	Botón “parte de visitas”, de la pantalla inicial.
<b>Flujo</b>	Usuario: Pulsa sobre botón “parte de visitas” → Sistema: lanza pantalla de planing diario mostrando visitas pendientes de cerrar → Usuario: visualiza información.
<b>Post condiciones</b>	Pantalla planing diario activa.
<b>Notas:</b> para salir de esta pantalla es usuario deberá pulsar el botón “salir”. El concepto de “visita cerrada”, se refiere a que los campos de “resultado de visita” estarán sin definir.	

<b>Identificador</b>	<b>CU-006</b>
<b>Nombre</b>	<b>CERRAR VISITA PLANIFICADA.</b>
<b>Prioridad</b>	Alta
<b>Descripción</b>	Con esta acción el usuario registrará el resultado de la visita / reparto realizado al cliente.
<b>Actores</b>	Vendedor
<b>Precondiciones</b>	Estado de la aplicación en ejecución, con la actividad de la pantalla inicial activa. Y visitas creadas, sin cerrar, en la base de datos.
<b>Iniciado por</b>	Botón “parte de visitas”, de la pantalla inicial.
<b>Flujo</b>	Usuario: Pulsa sobre botón “parte de visitas” → Sistema: lanza pantalla de planing diario → Usuario: visualiza información y pulsa sobre la visita a cerrar → Sistema: lanza pantalla visita, con los datos de la visita seleccionada → Usuario:

	registra el resultado de la visita y pulsa el botón “guardar” → Sistema: informa, a través de una ventana emergente, si a finalizado o no correctamente la función, y saldrá a la pantalla principal.
<b>Post condiciones</b>	Estado de la aplicación en ejecución, con la activity de la pantalla inicial activa. Y la visita cerrada.
<b>Notas:</b> El concepto de “visita cerrada”, se refiere a que los campos de “resultado de visita” estarán sin definir.	

<b>Identificador</b>	<b>CU-007</b>
<b>Nombre</b>	<b>BUSCAR CLIENTE.</b>
<b>Prioridad</b>	Media
<b>Descripción</b>	Localizar y visualizar un cliente creado en la base de datos.
<b>Actores</b>	Vendedor o Responsable de ventas.
<b>Precondiciones</b>	Estado de la aplicación en ejecución, con la activity de la pantalla inicial activa. Y cliente creado en la base de datos.
<b>Iniciado por</b>	Botón “cliente”, de la pantalla inicial.
<b>Flujo</b>	Usuario: Pulsa sobre botón “cliente” → Sistema: muestra pantalla “ficha de cliente” en blanco → Usuario: pulsa buscar → Sistema: lanza pantalla buscar_cliente → Usuario: cumplimenta nombre del cliente y pulsa botón “buscar” → Sistema: muestra lista de elementos encontrados → Usuario: pulsa sobre el cliente preferido → Sistema: lanza pantalla “ficha cliente”, con el cliente seleccionado → Usuario: visualiza datos y pulsa “salir”.
<b>Post condiciones</b>	Estado de la aplicación en ejecución, con la activity de la pantalla inicial activa.
<b>Notas:</b> sin comentarios.	

<b>Identificador</b>	<b>CU-008</b>
<b>Nombre</b>	<b>SINCRONIZACIÓN BASE DE DATOS.</b>
<b>Prioridad</b>	Media
<b>Descripción</b>	Actualiza las visitas creadas y cerradas, por

	el vendedor, en la base de datos.
<b>Actores</b>	Vendedor o Responsable de ventas.
<b>Precondiciones</b>	Estado de la aplicación en ejecución, con la actividad de la pantalla inicial activa. Y visitas creadas pendientes de actualizar.
<b>Iniciado por</b>	Botón “sincronización”, de la pantalla inicial.
<b>Flujo</b>	Usuario: Pulsa sobre botón “sincronización” → Sistema: lanza pantalla de sincronización → Usuario: pulsa sobre el botón “sincronizar” → Sistema: muestra información de la ejecución en ventana emergente e informa sobre el resultado de la acción.
<b>Post condiciones</b>	Estado de la aplicación en ejecución, con la actividad de la pantalla inicial activa. Y la actualización de las visitas realizada.
<b>Notas:</b> sin comentarios.	

Las siguientes funciones de casos de uso serán incluidas en la versión Premium de la aplicación. Aunque no es objeto de la app versión inicial, para trabajo final de carrera, se detalla a continuación:

9. Búsqueda y visualización, del parte acumulado de visitas por vendedor.
10. Configuración del sistema.
11. Crear usuarios.

<b>Identificador</b>	<b>CU-009</b>
<b>Nombre</b>	<b>VISUALIZAR PARTE VISITA ACUM. VENDEDOR</b>
<b>Prioridad</b>	Media
<b>Descripción</b>	Muestra parte de visitas acumulado del vendedor que se haya seleccionado.
<b>Actores</b>	Responsable de ventas.
<b>Precondiciones</b>	Estado de la aplicación en ejecución, con la actividad de la pantalla inicial activa. Y visitas creadas y cerradas, en el transcurso del periodo seleccionado.
<b>Iniciado por</b>	Menú / Visitas Acumuladas, de la pantalla principal.
<b>Flujo</b>	Usuario: pulsa sobre el menú de la pantalla principal, en la opción “visitas acumuladas” → Sistema: muestra pantalla

	de visitas acumuladas por vendedor -> Usuario: selecciona nombre del vendedor y mes. Pulsa botón “mostrar” -> Sistema: busca la información y la muestra. -> Usuario: visualiza información, y pulsa el botón “salir”.
<b>Post condiciones</b>	Estado de la aplicación en ejecución, con la activity de la pantalla inicial activa. Y la consulta realizada.
<b>Notas:</b> sin comentarios.	

<b>Identificador</b>	<b>CU-010</b>
<b>Nombre</b>	<b>CONFIGURACIÓN SISTEMA.</b>
<b>Prioridad</b>	Media
<b>Descripción</b>	Parámetros a configurar por el administrador, para la conexión a la base de datos, de la función de sincronización.
<b>Actores</b>	Responsable de ventas/Administrador.
<b>Precondiciones</b>	Tener conexión a Internet activa.
<b>Iniciado por</b>	Menú / Configuración, de la pantalla principal.
<b>Flujo</b>	Usuario: pulsa sobre Menú/configuración de la pantalla principal -> Sistema muestra pantalla de “configuración “ con los datos a cumplimentar -> Usuario: escribe los parámetros de conexión, y pulsa botón “guardar” -> Sistema: actualiza los datos obtenidos, informa de la realización o no de la acción, y sale a la pantalla de inicio.
<b>Post condiciones</b>	Estado de la aplicación en ejecución, con la activity de la pantalla inicial activa. Y la configuración del sistema actualizada.
<b>Notas:</b> sin comentarios.	

<b>Identificador</b>	<b>CU-011</b>
<b>Nombre</b>	<b>CREAR USUARIO.</b>
<b>Prioridad</b>	Media
<b>Descripción</b>	Función que realiza el alta de un usuario en el sistema.
<b>Actores</b>	Responsable de ventas/Administrador.

<b>Precondiciones</b>	Estado de la aplicación en ejecución, con la activity de la pantalla inicial activa.
<b>Iniciado por</b>	Menú / Usuarios, de la pantalla principal.
<b>Flujo</b>	Usuario: pulsa sobre Menú/Usuarios de la pantalla principal -> Sistema muestra pantalla de “usuarios “ con los datos a cumplimentar -> Usuario: escribe los parámetros de conexión, y pulsa botón “guardar” -> Sistema: actualiza los datos obtenidos, informa de la realización o no de la acción, y sale a la pantalla de inicio.
<b>Post condiciones</b>	Estado de la aplicación en ejecución, con la activity de la pantalla inicial activa. Y el usuario creado en el sistema.
<b>Notas:</b> sin comentarios.	

## 4. Implementación

---

### 4.1 Entorno de desarrollo.

Para este trabajo hemos utilizado el nuevo IDE de desarrollo en Android, concretamente, Android Studio. Está basado en un IDE para Java llamado IntelliJ IDEA. Este nuevo entorno nos proporciona herramientas para el desarrollo en android, sin tener que instalar adicionalmente plugin como ocurría con Eclipse.

Proporciona herramientas para el desarrollo en android al igual que en Eclipse con el plugin ADT. Además también nos ofrece numerosas opciones adicionales, destacando:

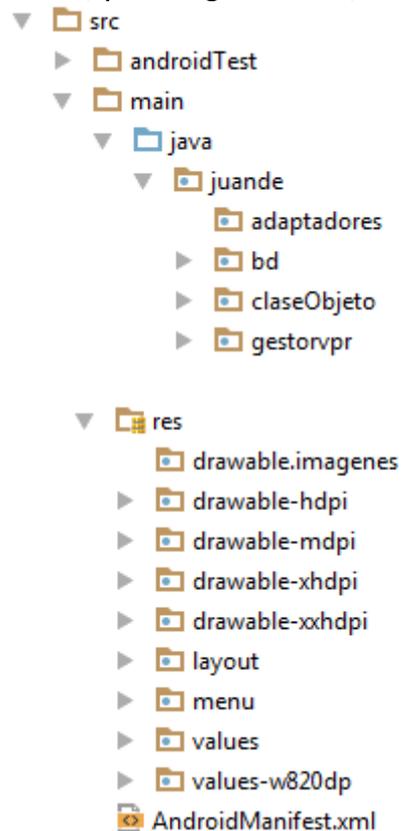
- Soporte para Gradle
- Refactorización específica para Android y soluciones rápidas.
- Herramientas para medir el rendimiento, usabilidad, compatibilidad de versiones y otros problemas.
- ProGuard y capacidad para firmar apps
- Asistentes basados en plantillas para crear diseños y componentes en Android.
- Y un editor de diseño “drag and drop” para los componentes de la UI, con vista previa de los layouts en múltiples configuraciones de pantalla .

Está herramienta la impulsa Google, con lo cual, será la plataforma de desarrollo principal en un breve espacio de tiempo, sino lo es ya.

Como apoyo al proyecto, se ha utilizado la plataforma independiente de bases de datos XAMPP, que consiste principalmente en una base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.

## 4.2 Estructura de la aplicación.

A continuación muestro la estructura completa de la aplicación, describiendo en detalle las carpetas utilizadas, y más significativas, de la aplicación.



- **carpeta src:** contiene el código fuente de la aplicación, dentro de ella está el código de la aplicación.
- **carpeta gen:** contiene el código generado de forma automática por el SDK.
- **carpeta bin:** en esta ubicación se compila el código y se genera el .apk, fichero comprimido que contiene la aplicación final lista para instalar.
- **carpeta res :** contiene los recursos usados por la aplicación. En ella se encuentran la subcarpetas

**Layout,** contiene ficheros XML con vistas de aplicación permitiéndonos configurar las diferentes pantallas que compondrán la interfaz de usuario de la aplicación.

**Menu ,** fichero XML con los menús de la actividad.

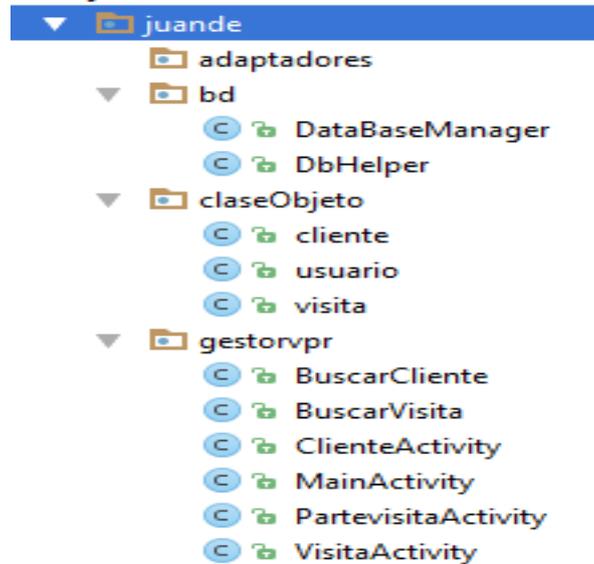
**Values,** fichero XML para indicar los valores de tipo string, color, estilo o dimensión.

**Drawable,** carpeta que contiene los ficheros de imágenes (JPG o PNG), y descriptores de imágenes en XML.

-**Fichero AndroidManifest.xml,** contiene la estructura de la aplicación Android, donde se indican las actividades, los servicios, los proveedores de contenido y los permisos

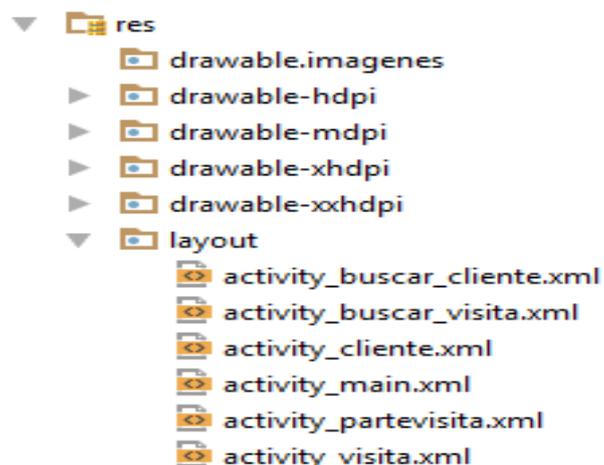
que requerirá la app. También incluye la versión mínima de Android para poder ejecutarla.

Por otro lado, el código fuente ubicado en la ruta *src/main/java/juande*, contiene las siguientes subcarpetas:



- **Carpeta “adaptadores”**: donde se encontrarán los adaptadores personalizados para la versión Premium. La versión presentada ha tomado los adaptadores genéricos ofrecidos por Android.
- **Carpeta “bd”** : subsistema que contiene las clases y métodos necesarios, para la creación de la base de datos, y, creación, actualización y borrado de datos.
- **Carpeta “claseObjeto”**: contiene la definición de los atributos y funciones, de las clases objeto utilizadas en esta aplicación, con sus getters y setters.
- **Carpeta “gestorvpr”**: contiene las clases tipo entidad de la aplicación. Donde se definen la funcionalidad asociada a cada layout de la aplicación.

Además, en la carpeta “res”, podemos encontrar la siguiente estructura:



En ella destacamos los diferentes layouts / interfaces usados en la aplicación. Y las carpetas “drawable xxx”, donde residen las imágenes utilizadas, con sus diferentes versiones de resolución.

### 4.3 Estado actual del proyecto.

Podemos afirmar que las funciones planteadas para la versión inicial, están conseguidas. Recordemos que eran:

- *Parte diario de visitas:* registro diario de las visitas realizadas a los clientes, informando del cliente visitado, la gestión comercial realizada y el resultado de esta. Que incluye
  - alta, baja y edición de visitas realizadas.
  - búsqueda de clientes por nombre.
  - búsqueda de visitas por fecha.
- *Ficha cliente:* donde mostrará los datos del cliente, con persona de contacto y tipo cliente, incluyendo la función de “búsqueda de clientes” por razón social.
- *Propuesta de visitas:* esta función presentará en forma de listado, las visitas que están pendientes de cerrar, en función de los parámetros que se hayan definido, incluyendo la función “OnClick”, capturando los datos seleccionados y mostrando la visita para su edición.

Queda pendiente para la entrega final, dar robustez a la aplicación, mejorando la detección de errores y control de acciones correctoras para evitar el mal uso de la aplicación al usuario. Las detectadas para su corrección son:

- *cargar base de datos del cliente, como paso principal*
- *en el proceso de búsqueda de cliente o visita, de la pantalla “Visitas”, debe registrar lo marcado en el Checkbox, cuando retorne de la búsqueda.*
- *cuando seleccionemos una visita, en la pantalla “Parte de Visitas”, debe registrar lo marcado en el Checkbox, cuando retorne de la búsqueda.*
- *actualizar ficha visita si se modifica, cuando esté creada, y no generar una nueva.*
- *anular la tecla física de “atrás” del dispositivo portátil, para evitar inconsistencias en la captura del identificador de visitas.*

Por otro lado, debo mencionar que se ha eliminado la “pantalla intermedia”, haciendo caso a las recomendaciones del consultor, y dando un enfoque sobre el objeto en cuestión y no sobre la acción.

También, aunque no esté planificado me gustaría incluir, dos acciones que tenía pensado para la versión Premium, que son:

- Alta y registro de usuarios.
- Sincronización con la base de datos “oncloud”.

Y para versiones posteriores se incluirá:

- *Sincronización/registro en BD*: esta función deberá conectar con una BD “oncloud”, donde registrará todas las acciones realizadas por el usuario en ese día.
- *Parte de visitas acumulado*: esta función mostrará la actividad realizada por el vendedor, en forma de “cuadro resumen”, mostrando datos como: número de visitas, periodicidad por cliente, campañas presentadas, ventas realizadas, etc.
- *Propuesta de ruta optima*: mostrará en forma de listado, las visitas a realizar en el día, pero con la premisa de realizar la ruta más corta posible.

## 5. Testing

---

### 5.1 Test Unitarios.

Se ha comprobado el correcto funcionamiento de los diferentes componentes de cada módulo, asegurando que cada uno de los módulos funcione correctamente por separado. Concretamente se han comprobado las siguientes clases:

- BuscarCliente : función búsqueda y función listview.
- BuscarVisita : función búsqueda y función listview.
- ClienteActivity : inserción de datos búsqueda visita, función salir y función buscar.
- MainActivity : que cada botón realice lo esperado.
- PartevisitaActivity: la función “setOnItemClickListener”;
- VisitaActivity : las funciones Onclick y setOnClickListener.

Posteriormente, con las Pruebas de Integración, se ha comprobado el funcionamiento del sistema en cuestión.

### 5.2 Pruebas de integración.

En el siguiente apartado se detalla las acciones que deberá realizar la aplicación, ordenando estas, en función de la pantalla en la que nos situemos. Se detalla la acción que hará el usuario y el resultado que debe ejecutar:

<b>PANTALLA INICIAL</b>			
<b>Nº</b>	<b>ACCION</b>	<b>RESULTADO ESPERADO</b>	<b>VERIFICACIÓN</b>
1	Pulsar sobre el botón "reparto"	Muestra la pantalla de Visitas	OK
2	Pulsar sobre el botón "visitas"	Muestra la pantalla de Visitas	OK
3	Pulsar sobre el botón "cliente"	Muestra la pantalla de Ficha Cliente	OK
4	Pulsar sobre el botón "parte de visitas"	Muestra pantalla "parte de visitas", con el listado de visitas a realizar	OK
5	Pulsar sobre el botón "sincronización"	Muestra pantalla de sincronización	Versión Premium
<b>NºTOTAL ACCIONES</b>	<b>CORRECTAS</b>	<b>ERRORES</b>	<b>% VERIFICACIÓN</b>
<b>5</b>	<b>4</b>	<b>0</b>	<b>80,00%</b>

<b>PANTALLA REPARTO/VISITA</b>			
<b>Nº</b>	<b>ACCION</b>	<b>RESULTADO ESPERADO</b>	<b>VERIFICACIÓN</b>
1	Rellenar los campos de edición	Mostrar correctamente la información editada	OK
2	Pulsar sobre la lupa de búsqueda cliente	Muestra pantalla de búsqueda de cliente.	OK
3	Pulsa sobre el botón "buscar"	Muestra pantalla de búsqueda de visita por fecha	OK
4	Pulsar el botón "Salir"	Salir a pantalla principal	OK
5	Pulsar sobre el botón "Guardar"	Guardar el reparto/visita generado, en la BD, mostrando feedback del resultado de la acción	OK
6	Pulsar el botón eliminar	Elimina la visita mostrada en pantalla, de la BD,	OK

		mostrando feedback del resultado de la acción.	
<b>NºTOTAL ACCIONES</b>	<b>CORRECTAS</b>	<b>ERRORES</b>	<b>% VERIFICACIÓN</b>
6	6	0	100%

<b>PANTALLA CLIENTE</b>			
<b>Nº</b>	<b>ACCION</b>	<b>RESULTADO ESPERADO</b>	<b>VERIFICACIÓN</b>
1	Pulsar botón "Buscar"	Muestra pantalla de búsqueda de cliente	OK
2	Pulsar el botón "Salir"	Salir a pantalla principal	OK
<b>NºTOTAL ACCIONES</b>	<b>CORRECTAS</b>	<b>ERRORES</b>	<b>% VERIFICACIÓN</b>
2	2	0	100%

<b>PANTALLA PARTE DE VISITAS</b>			
<b>Nº</b>	<b>ACCION</b>	<b>RESULTADO ESPERADO</b>	<b>VERIFICACIÓN</b>
1	Visualizar las visitas propuestas por el sistema	Mostrar listado de visitas pendiente de cerrar por fecha y cliente.	OK
2	Pulsar sobre una fila de la lista	Mostrar la pantalla de visita, en modo edición, de la visita seleccionada	OK
<b>NºTOTAL ACCIONES</b>	<b>CORRECTAS</b>	<b>ERRORES</b>	<b>% VERIFICACIÓN</b>
2	2	0	100%

<b>PANTALLA BUSCAR VISITA (subrutina)</b>			
<b>Nº</b>	<b>ACCION</b>	<b>RESULTADO ESPERADO</b>	<b>VERIFICACIÓN</b>
1	Rellenar los campos de búsqueda, con la fecha de la visita y pulsar "buscar".	Mostrar visitas existentes	OK
2	Pulsar sobre un ítem de la lista	Captura los datos de esa visita, y enviarlos a la pantalla visita	OK

		para mostrarlos.	
<b>NºTOTAL ACCIONES</b>	<b>CORRECTAS</b>	<b>ERRORES</b>	<b>% VERIFICACIÓN</b>
2	2	0	100%

<b>PANTALLA BUSCAR CLIENTE(subrutina)</b>			
<b>Nº</b>	<b>ACCION</b>	<b>RESULTADO ESPERADO</b>	<b>VERIFICACIÓN</b>
1	Rellenar los campos de búsqueda, con la razón social del cliente.	Mostrar cliente, si existe	OK
2	Pulsar sobre un ítem de la lista	Captura la Razón Social de la búsqueda encontrada y la muestra en la pantalla Visita	OK
3	Pulsar sobre un ítem de la lista	Captura el objeto cliente encontrado y lo muestra en la Ficha cliente.	OK
<b>NºTOTAL ACCIONES</b>	<b>CORRECTAS</b>	<b>ERRORES</b>	<b>% VERIFICACIÓN</b>
3	3	0	100%

La diferentes pruebas realizadas se han hecho sobre simulador de Android Studio, y dispositivo físico. Concretamente:

- Simulador Genérico , pantalla de 5''.
- Simulador Genérico, pantalla de 6''.
- Simulador Genérico, pantalla de 7''.
- Dispositivo móvil físico marca Xiaomi Mi3, pantalla de 5''.
- Tablet Samsung Galaxy, pantalla de 10,1''.

Los test de funcionamiento se han realizado en los dispositivos físicos de dos formas distintas:

- A través de Android Studio, con conexión a USB, usado el dispositivo como emulador.
- Instalando la .apk generada, descargada de internet, a través del servicio de alojamiento de archivos multiplataforma en la nube dropbox.

En síntesis, podemos confirmar que se ha cumplido el objetivo marcado, a nivel estructural y funcional. Pero aún queda por mejorar ciertos aspectos de

funcionamiento, que aunque a groso modo ejecutan, no fluyen como debería. Estas acciones correctoras descritas anteriormente, se solucionarán para la entrega final.

## 6. Documentación Técnica

### 6.1 Clases.

A continuación describiremos las principales clases que componen la app, a través de fichas CRC. Concretamente las controladoras, que son las clases donde girará toda la funcionalidad de nuestra aplicación. Y las de la base de datos:

<b>CLASE: Main</b>	
<b>Descripción:</b>	Gestiona las funciones que subyacen de la pantalla principal.
<b>Tipo:</b>	Controlador
<b>Características:</b>	Concreta
<b>Responsabilidades:</b>	Colaboración / Enlace / Vinculo
Gestiona los enlaces a las diferentes pantallas de la aplicación	
<b>CONSTRUCTORES</b>	
+MainActivity();	
<b>Atributos</b>	<b>Métodos</b>
Button btnvisita	onCreate(Bundle savedInstanceState)
Button btncliente	setOnClickListener( View view)
Button btnreparto	onCreateOptionsMenu(Menu menu)
Button btnsincro	onOptionsItemSelected(MenuItem item)
Button btnpartevisita	

<b>CLASE: VisitaActivity</b>	
<b>Descripción:</b>	Gestiona las funciones que subyacen de la pantalla Visita
<b>Tipo:</b>	Controlador
<b>Características:</b>	Concreta
<b>Responsabilidades:</b>	Colaboración
Gestiona las llamadas a las funciones de alta, búsqueda, actualización y eliminación de la visita.	
<b>CONSTRUCTORES</b>	
+VisitaActivity;	
<b>Atributos</b>	<b>Métodos</b>
TextView nvisita	onCreate( Bundle savedInstanceState )
TextView fechavisita	setOnClickListener( View view )
CheckBox checkreparto	setOnItemSelectedListener( AdapterView

	view)
CheckBox checkcortesia	capturaCheck()
CheckBox checkprospeccion	registroFecha()
CheckBox checkcampana	
ImageButton btnbuscacliente	
TextView buscacliente	
Button btnguardarvisita	
Button btneliminarvisita	
Button btnsalirvisita	
Button btnbuscarvisita	
String tp	
DataBaseManager manager	

## CLASE: ClienteActivity

<b>Descripción:</b>	Gestiona las funciones que subyacen de la pantalla Cliente.
<b>Tipo:</b>	Controlador
<b>Características:</b>	Concreta
<b>Responsabilidades:</b>	Colaboración
Gestiona la llamada a la función búsqueda, mostrando los datos obtenidos.	
<b>CONSTRUCTORES</b>	
+ClienteActivity();	
<b>Atributos</b>	<b>Métodos</b>
Button btnsalir	onCreate(Bundle savedInstanceState)
Button btnbuscar	setOnClickListener( View view)
TextView razonsocial	
TextView domicilio	
TextView localidad	
TextView cp	
TextView provincia	
TextView cont1	
TextView cont2	
TextView telefono	
TextView movil	
TextView mail	
TextView tipocliente	

## CLASE: PartevisitaActivity

<b>Descripción:</b>	Gestiona las funciones que subyacen de la pantalla Parte de Visitas
<b>Tipo:</b>	Controlador
<b>Características:</b>	Concreta
<b>Responsabilidades:</b>	Colaboración

Gestiona la llamada a la función de listado de visitas pendientes, y la selección de la visita en el listado mostrado.	
<b>CONSTRUCTORES</b>	
+ PartevisitaActivity ();	
<b>Atributos</b>	<b>Métodos</b>
ListView listavisitas	onCreate(Bundle savedInstanceState)
DataBaseManager manager	setOnClickListener( View view)
Cursor cursor	onItemClick(AdapterView<> parent, View view, int position, long id)
SimpleCursorAdapter adaptador	
visita datospartevisita	

<b>CLASE: BuscarCliente</b>	
<b>Descripción:</b>	Gestiona la búsqueda de un cliente
<b>Tipo:</b>	Controlador
<b>Características:</b>	Concreta
<b>Responsabilidades:</b>	Colaboración
Gestiona la búsqueda de un cliente, por nombre.	
<b>CONSTRUCTORES</b>	
+BuscarCliente();	
<b>Atributos</b>	<b>Métodos</b>
DataBaseManager manager	onCreate(Bundle savedInstanceState)
Cursor cursor	setOnClickListener( View view)
ListView lista	setOnItemSelectedListener( AdapterView view)
SimpleCursorAdapter adaptador	
TextView txbuscacliente	
cliente datoscliente	

<b>CLASE: BuscarVisita</b>	
<b>Descripción:</b>	Gestiona la búsqueda de una visita
<b>Tipo:</b>	Controlador
<b>Características:</b>	Concreta
<b>Responsabilidades:</b>	Gestiona la búsqueda de un cliente
Gestiona la búsqueda de una visita, por fecha de visita	
<b>CONSTRUCTORES</b>	
+MainActivity();	
<b>Atributos</b>	<b>Métodos</b>
DataBaseManager manager	onCreate(Bundle savedInstanceState)
Cursor cursor	setOnClickListener( View view)
ListView lista	onOptionsItemSelected(Menuitem item)
SimpleCursorAdapter adaptador	

TextView txbuscavisita	
ImageButton imgbuscavisita	
visita datosvisita	

## CLASE: DbHelper

<b>Descripción:</b>	Clase motora de la base de datos
<b>Tipo:</b>	Gestor
<b>Características:</b>	Concreta
<b>Responsabilidades:</b>	Generación / Colaboración.
Generar y actualizar la base de datos especificada.	
<b>CONSTRUCTORES</b>	
+ DBHelper(Context context);	
<b>Atributos</b>	<b>Métodos</b>
static final String DATABASE_NAME	onCreate(SQLiteDatabase db)
static final int DB_SCHEME_VERSION	onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)

## CLASE: DataBaseManager

<b>Descripción:</b>	Clase motora de la funcionalidad de la bd
<b>Tipo:</b>	Gestor
<b>Características:</b>	Concreta
<b>Responsabilidades:</b>	Generación / Colaboración.
Crea, busca, actualiza y elimina los datos de la base de datos generada.	
<b>CONSTRUCTORES</b>	
+ DataBaseManager(Context context)	
<b>Atributos</b>	<b>Métodos</b>
static final String TABLE_NAME_CLIENTE*	ContentValues contenedorVisita (String fecha, String tipovisita, String clientevisita, String resultadovisita )
static final String CREATE_TABLE_CLIENTE	ContentValues contenedorCliente (String rsocial,String cif,String domicilio, String localidad,String prov, String cp,String personac1,String personac2, String tel,String mail,String movil, String tipocliente)
static final String TABLE_NAME_VISITA*	insertarVisita(String fecha, String tipovisita, String clientevisita, String resultadovisita)
static final String CREATE_TABLE_VISITA	eliminarVisita(String idVisita)
static final String TABLE_NAME_USUARIO*	modificarVisita(String id,String fecha,

	String tipovisita, String clientevisita,String resultadovisita)
static final String CREATE_TABLE_USUARIO	insertarCliente(String rsocial, String cif,String domicilio, String localidad, String prov, String cp, String personac1,String personac2, String tel, String mail,String movil, String tipocliente)
DbHelper helper	cursorBuscarRazonSocial(String nombre)
SQLiteDatabase db	cursorBuscarClienteid(String id)
	Cursor cursorClientes()
	cursorBuscarVisitaid(String id)
	Cursor cursorBuscarVisitafecha( String fecha)
	cursorVisitas()
	cursorPartevisitas(String rs, String fcha)
	cursorListaPartevisitas()

\*los atributos de las tablas no se han incluido por razones de espacio.

## 6.2 Relación de funcionalidades.

A continuación detallo las funciones definitivas que se van a presentar para el TFC, una vez corregidos los errores anteriores. Están agrupadas en función de la pantalla o interface que mostrará al cliente:

- Pantalla Visita:
  - Alta de una visita nueva, lo que conlleva la cumplimentación del formulario y la búsqueda del cliente a visitar.
  - Búsqueda de una visita creada.
  - Editar/modificar una visita creada.
  - Eliminar una visita creada.
- Pantalla Cliente:
  - Buscar la ficha de un cliente, mostrando la información por pantalla.
- Pantalla Parte de Visitas:
  - Mostrar visitas pendientes de cerrar, para editarlas posteriormente.
- Pantalla Sincronización:
  - Carga la base de datos de clientes en la aplicación.

Y para la versión posteriores, planteo realizar las siguientes funciones adicionales:

- Pantalla Menú/Configuración/Usuarios:
  - Alta de usuarios
- Pantalla Parte de Visitas:

- Visitas acumuladas, que mostrará un enlace a una nueva pantalla, donde se mostrará la actividad de cada usuario, en forma de “cuadro resumen”, mostrando datos como: número de visitas, periodicidad por cliente, campañas presentadas, ventas realizadas, etc.
- Pantalla Sincronización: esta función deberá conectar con una BD “oncloud” ,donde registrará todas las visitas realizadas por el usuario en ese día.

## 7. Manuales

---

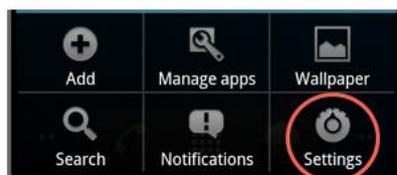
### 7.1 De instalación.

La aplicación está destinada para dispositivos móviles con versiones android de API 15, en adelante. Es decir, a partir del sistema 4.1 Jelly Bean. Y con tamaño de pantalla entre 5'' y 7'', preferentemente.

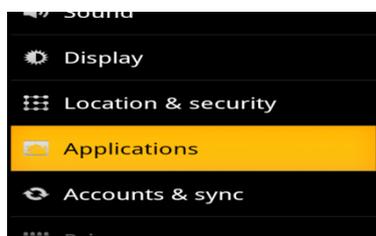
Para instalar el archivo .apk, el dispositivo deberá estar activa la opción que permita la instalación de otras aplicaciones de origen distinto al market de google.

Para ello, procedemos de la siguiente forma:

1. Nos vamos a la configuración del sistema



2. Marcamos en aplicaciones



3. Y seleccionamos “permitir instalación de aplicaciones distintas a Android Market”



Dependiendo del dispositivo móvil que tengamos los pasos a seguir serán diferentes y tendremos que buscar en las instrucciones del terminal. Pero por norma general, debe estar en la configuración del sistema o en la ruta de “Seguridad” → “Administración de dispositivos” → “Orígenes desconocidos”.

Una vez activa esta opción, seleccionamos el archivo .apk, que será enviado por mail o vía dropbox, que es un servicio multiplataforma en la nube, al usuario que lo solicite. Y descargará la aplicación en su móvil.

Una vez descargada, nos iremos a la carpeta de descargas del dispositivo móvil , clicaremos sobre ella. Aceptamos la instalación. Y ya podremos ejecutarla.

## 7.2 De usuario.

### Pantalla de Inicio:

El usuario pulsará sobre el icono de la aplicación, y se mostrará la pantalla inicial. En ella podemos ver diferentes opciones que explicamos a continuación:



**Pantalla de Inicio**

- Reparto: opción para acceder a la pantalla de visitas donde se registrarán los repartos pendientes de hacer. Vendrá la opción de “reparto”, preseleccionada.

- Visitas: opción con la misma finalidad que la anterior pero con la opción del tipo de visita, sin preseleccionar.
- Cliente: mostrará la ficha de clientes en la que podremos buscar los datos del cliente.
- Parte de Visitas: esta pantalla mostrará las visitas pendientes de cerrar, por el vendedor. Una visita pendiente de cerrar, es una visita que no tiene registrada el resultado de la misma.
- Sincronización: ejecuta la conexión de datos, cargando la base de datos de clientes, y visitas actualizadas del servidor.

### Pantalla de Visita:

A través de esta pantalla es usuario registrará toda su actividad de visitas a clientes realizada en el transcurso del día. Se muestra a continuación:



**Pantalla de Visita**

En esta pantalla el usuario puede realizar las siguientes acciones:

- Registrar una visita, cumplimentando los datos que se muestran en pantalla y pulsando el botón “guardar”.
- Buscar una visita, pulsando el botón “buscar” que mostrará la pantalla de búsqueda de visitas. La búsqueda se realiza introduciendo la fecha de visita, y mostrará las visitas realizadas en ese día.
- Eliminar una visita creada, pulsado el botón “eliminar”, después de haber realizado la búsqueda de la visita.
- Salir de la pantalla, pulsando el botón “salir”.

Destacar que la búsqueda de clientes de esta pantalla, lanzará la pantalla de búsqueda de clientes. Y esta consulta debe realizarse por el nombre del cliente.

### Pantalla del parte de Visitas:

Esta pantalla muestra un listado de visitas, que ha generado el usuario con anterioridad, y que están pendientes de cerrar.



**Pantalla Parte de Visitas**

El usuario pulsará sobre una de las visitas pendientes de cerrar que aparecen, y el sistema ejecutará la pantalla de ficha de visitas, para que el usuario cumplimente la visita con el resultado de esta. Cerrando así la visita realizada.

### Pantalla de Cliente:

La siguiente interface muestra una ficha del cliente, con datos útiles para el usuario, si tiene la necesidad de buscar información sobre el cliente que va a visitar.



**Pantalla Cliente**

Al pulsar sobre el botón "buscar", el sistema muestra la pantalla de búsqueda de clientes, en la que deberá introducir el nombre del cliente a buscar y pulsar sobre él. Automáticamente, se mostrará la ficha de cliente con los datos requeridos.

## 8. Conclusiones.

---

Partiendo de los objetivos que pretendía cubrir con este trabajo, debo considerar, a nivel general, que he alcanzado las metas propuestas para este proyecto.

Estimo cumplido los siguientes items:

- Aprender a desarrollar un proyecto en la vida real.
- Profundizar en el lenguaje de programación Java/Android con vistas a poder dedicarme profesionalmente a ello.
- Agilizar el trabajo de registro de visitas realizado por el vendedor, ya que realizará la acción “in situ”, en el momento de finalizar la visita.
- Digitalizar el registro de visitas del vendedor, aportando información que antes solo estaba en soporte papel, anulando así la duplicidad de tareas.
- Mejorar la gestión de visitas (información, registro y seguimiento) con funciones como: histórico de visitas por cliente o propuestas de visitas a realizar.

Por otro lado, este trabajo me ha ayudado a comprender varias cuestiones, que aunque podía suponer, nunca las había experimentado. Son:

- La importancia de visualizar y “desmigra” una idea antes de ponerse a implementar la aplicación.
- Como una función mal implementada, acabará dándote más problemas que soluciones.
- Es FUNDAMENTAL, una idea del conjunto de la aplicación.
- Si para una simple app (de un desarrollador inexperto), he invertido tanto tiempo, ¿!cuanto debe ser una aplicación compleja!?

Además la metodología de la asignatura me ha parecido muy práctica y clara, a la hora de llevarla a cabo. Tan solo, he echado en falta más información técnica para el desarrollo de las aplicaciones en Android.

En un futuro próximo quiero seguir desarrollando esta aplicación con el objetivo de culminar mi idea inicial en su totalidad. Y si es posible, dedicarme profesionalmente a ello.

Gracias a todos por vuestra colaboración

## 9. Bibliografía.

---

**Tomás Girones, Jesús** (2014). El gran libro de Android, 4ªedic. MARCOMBO.

**Tomás Girones, Jesús** (2014). El gran libro de Android Avazado, 2ªedic. MARCOMBO.

**Garrido Barrientos, Sergio** (2014). Diseño de base de datos. EDICIONES KINDLE.

**Insertar, eliminar y actualizar registros en BD:** <http://codigoalonso.blogspot.com.es/2013/11/sqlite-insertar-modificar-eliminar-registros-android.html>

**Recuperar y mostrar registros en un ListView :** <http://codigoalonso.blogspot.com.es/2014/02/Recuperar-Mostrar-Registros-SQLite-Android-CursorAdapter-ListView.html>

**Conexión a una Base de Datos externa con Android:** <http://academiaandroid.com/conexion-base-datos-externa-android-sgbd/>

**Rivas Lequerica, Joan** (2015). Desarrollo De Aplicaciones Para Android. ANAYA.