Moventaa

Aplicación móvil para la gestión y control de ventas para vendedores

Marcos Martinez Cedillo
Ingeniería Técnica de Informática de Gestión

Consultor: Helena Boltà Torrell, Jordi Almirall López

Enero del 2015

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	Moventaa
Nombre del autor:	Marcos Martinez Cedillo
Nombre del consultor:	Helena Boltá Torrell, Jordi Almirall López
Fecha de entrega (mm/aaaa):	01/01/15
Área del Trabajo Final:	Desarrollo aplicaciones dispositivos móviles (Android)
Titulación:	*Ingeniería Técnica de Informática de Gestión*

Resumen del Trabajo (máximo 250 palabras):

Aplicación desarrollada bajo tecnología Android destinada a la gestión y control de ventas de productos para vendedores.

El objetivo principal de esta aplicación es la de proporcionar una herramienta de trabajo para los empleados, con el rol de vendedor, de una empresa que se dedica a la venta de productos en centros comerciales y que cuyo proceso de venta y por la disposición de des-localización geográfica de sus empleados en todo el territorio nacional, requieren de una herramienta de control de las ventas, stock de productos y de gestión a tiempo real.

El enfoque está pensado como una aplicación de herramienta para el trabajo, que facilite y ayude al empleado y que a su vez proporcione información directa a los departamentos de ventas de la compañía.

Entre las características principales funcionales de la aplicación, destacamos su facilidad de uso, un sistema de autorizaciones fácil para el usuario, uso del sensor del gps del teléfono para determinar la ubicación del empleado, uso de la cámara del teléfono para escanear códigos de barras de productos y para el envío de bajas de enfermedad.

A nivel tecnológico, la aplicación ha sido desarrollada bajo el marco de trabajo de App Engine de Google que facilita la tarea de desarrollo y despliegue de aplicaciones backend de Android para desarrolladores.

Abstract (in English, 250 words or less):

App developed under android technology for management and control of the sales of products for vendors.

The principal objective of this app is to do a tool for the vendors role employed of a enterprise market products on a shopping center and who his sales process and employed distribution around the country, require a control sales tool, stock and real-time management.

The approach was designed as an application tool for the job, to facilitate and assist the employee and which in turn provide directly to the sales of the company information.

The principal functional features of the aplication are, easy-to-use, user easy authentication, gps sensor location, camera barcode scanner and the use of the phone camera to capture a picture proof of sick leave to send to the company.

A technological level, this app, it was developed under the Google App Engine framework, which facilitates the task of developing and deploying Android backend applications for developers.

Palabras clave (entre 4 y 8):

Ventas, Control, Geolocalización, Agenda, Dispositivo móvil, Gestión, Recordatorios, App Engine

Índice de contenido

Capítulo 1: Introducción....8
1.1 Contexto y justificación del Trabajo....8
1.2 Objetivos del Trabajo....9
1.3 Enfoque y método seguido....10
Diseño centrado en el usuario....10
1.4 Planificación del Trabajo....11
1.4.1 Recursos necesarios....11
1.4.1.1 Software....11
1.4.1.2 Hardware....11
1.4.2 Listado de tareas a realizar:....12
1.4.3 Planificación temporal....12

Capítulo 2: Usuarios y contexto de uso....15
2.1 Métodos de indagación....15
2.2 Perfiles de usuario identificados....19
2.3 Contextos de uso del aplicativo....20
2.4 Análisis de tareas....20
2.5 Conclusiones finales....21

Capítulo 3: Diseño conceptual....22
3.1. Escenarios de uso....22
3.2. Flujos de interacción....25

Capítulo 4: Prototipado....26
4.1 “Sketches” escaneados....26
4.2 Prototipo horizontal....30

Capítulo 5: Evaluación....39
5.1 Diseño del test de usuario....39
5.2 Las tareas que los usuarios deben realizar....39
5.3 Cuestionario relacionado con las tareas del usuario....40
5.4 Resultado de la evaluación....40

Capítulo 6: Diseño técnico de la aplicación....41
6.1 Diseño de la arquitectura....41
6.1.1 Arquitectura a nivel de servidor....42
6.1.2 Arquitectura aplicación cliente....42
6.2 Diagrama UML/Entidad....44
6.3 Diagrama UML diseño del software....47
6.3.1 Diseño en servidor....47
Endpoints publicados para el cliente móvil....47
Endpoints publicados internos para la gestión operativa....49
6.3.2 Diseño en cliente....51

Capítulo 7: Desarrollo....64
7.1 Descripción de las herramientas utilizadas....64
7.1.1. Entorno de desarrollo Eclipse-IDE....64
7.1.2. Bibliotecas de terceros para Android....64
ButterKnife....64
Zxing 1D/2D....65
Caldroid....65
Date4J....65
7.2 Análisis del estado del proyecto....67
7.2.1 Estado del proyecto....67
7.2.2 Plan a futuro....69

Capítulo 8: Pruebas....70
8.1 Definición del plan de pruebas....70
8.2 Definición del plan de pruebas....71
8.2.1 Test funcional de la aplicación – Casos de prueba....71
Test-0001 - Registro del empleados....71
Test-0002 - Selección de centro asociado....72
Test-0003 - Muestra de productos para el centro seleccionado....73
Test-0004 - Alta de una venta....73
Test-0005 - Detalle de un producto....74
Test-0006 - Solicitar stock....75
Test-0007 - Buscador....76
Test-0008 - Solicitud de vacaciones....76
Test-0009 - Nóminas....77
Test-0010 - Descarga de la Nómina....78
Test-0011 - Envío de baja....78
Test-0012 – Prueba GPS....79

Capítulo 9: Conclusiones....80

Capítulo 10: Glosario....82

Capítulo 11: Bibliografía....83

Anexo 1: Instalación....85
1.1 Configuración del proyecto....85
1.2 Compilación....89
1.3 Población de los datos en la base de datos....89
1.4 Ejecución de las aplicaciones....90
1.5 Instalación solo del aplicativo android....91

Índice de ilustraciones

Ilustración 1: Flujos de interacción....26
Ilustración 2: Scretches escaneado 1....27
Ilustración 3: Scretches escaneado 2....28
Ilustración 4: Scretches escaneado 3....29
Ilustración 5: Scretches escaneados 4....30
Ilustración 6: Arquitectura....42
Ilustración 7: Model-View-Presenter....44
Ilustración 8: Diagrama Entidad Relación....45
Ilustración 9: Diagrama UML: MoventaaEndpoint....48
Ilustración 10: Diagrama UML: CentroEndpoint....49
Ilustración 11: Diagrama UML: VentaProductoCentroEndpoint....49
Ilustración 12: Diagrama UML: ProductoEndpoint....50
Ilustración 13: Diagrama UML: ProductoCentroEndpoint....50
Ilustración 14: Diagrama UML: MaintenanceTasksServlet....51
Ilustración 15: Diagrama UML: PlacesHelper....51
Ilustración 16: DiagramaUML: EmpleadoProductoVentas....52
Ilustración 17: Diagrama de paquetes Moventaa....53
Ilustración 18: Diagrama UML Cliente - MainActivity....54
Ilustración 19: Diagrama UML Cliente - EmployedActivity....55
Ilustración 20: Diagrama UML Cliente - ListProductosFragment....56
Ilustración 21: Diagrama UML Cliente - ListNominasActivity....57
Ilustración 22: Diagrama UML Cliente - VacacionesActivity....57
Ilustración 23: Diagrama UML Cliente - ProductoSearchActivity....58
Ilustración 24: Diagrama UML Cliente - GPSActivity....58
Ilustración 25: Diagrama UML Cliente - ListCentrosActivity....59
Ilustración 26: Diagrama UML Cliente - ProductoInfoActivity....60
Ilustración 27: Diagrama UML Cliente - BusquedaCentrosActivity....61
lustración 28: Diagrama UML Cliente - ListCentrosUsuarioActivity....62
Ilustración 29: lustración 24: Diagrama UML Cliente - BajaActivity....62
Ilustración 30: Diagrama UML Cliente - ImageDownloader....63
Ilustración 31: Diagrama UML Cliente - Utils....63
Ilustración 32: Diagrama UML Cliente - UserPrefsData....64
Ilustración 33: Diagrama UML Cliente - CloundEndpointUtils....64
Ilustración 34: Modificaciones diseño: Barra progreso....69
Ilustración 35: Modificaciones diseño: Menu....69
Ilustración 36: Modificaciones diseño: Buscador....69
Ilustración 37: Instalación - Carpetas....86
Ilustración 38: Instalación - Importación....86
Ilustración 39: Instalación - Define Workspace....87
Ilustración 40: Instalación - Ubicación proyectos....87
Ilustración 41: Instalación - Proyecto Android....88
Ilustración 42: Instalación - Workspace final....88
Ilustración 43: Android propiedades proyecto....89
Ilustración 44: Android propiedades butterknife....89
Ilustración 45: Andorid propiedades butterknife factory....90
Ilustración 46: Proceso de carga....91
Ilustración 47: Ejecutar como web-aplication....91
Ilustración 48: Arranque parte servidora....92

Ilustración 49: Aplicación Moventaa en emulador.....93

Capítulo 1: Introducción

1.1 Contexto y justificación del Trabajo

Los dispositivos móviles han tenido durante los últimos años un gran impacto dentro de la sociedad. Su uso ha proliferado en varios ámbitos, tanto en el personal como en el profesional. Podemos observar que cualquier persona dispone de un “smartphone” , ya no importa la edad o condición social o las dificultades que puedan entrañar, utilizar un móvil y acceder a la información desde cualquier lugar se ha vuelto tan cotidiano como cualquier otra actividad.

Una parte de la comunicación de las relaciones humanas se canalizan a través de los dispositivos móviles, ya sea en forma de conversaciones de voz, videoconferencias, “emails”, “chat”, redes sociales, etcétera. Por ello que un teléfono móvil ya no solo sirve para hablar, si no que es una herramienta de contacto, comunicación y de trabajo para el día a día de muchas personas. Es más, podríamos incluso decir que sin móvil el trabajo no sería tal y como hoy lo tenemos.

Por ello que los móviles han generado una revolución en lo que los clientes esperan de las compañías de las que consumen, una mayor comunicación, un mayor compromiso y una respuesta más rápida frente a sus necesidades.

Además, los dispositivos móviles han acercado la tecnología a las personas, en parte por su interacción amigable de su interfaz, por su diseño que agrada al uso y al tacto, y si a todo esto, le incluimos un hardware potente con diferentes tipos de elementos, disponemos en nuestras manos de una herramienta de trabajo profesional.

Estos motivos, nos hacen pensar que aprovechando las ventajas que nos ofrecen, podemos desarrollar una aplicación que cubra profesionalmente la necesidad de una empresa, en particular con su departamento de ventas. Para ponernos en contexto, la empresa dispone de un departamento de ventas cuyo personal se encuentra repartido por todo el ámbito nacional. El personal que forma el departamento es muy heterogéneo, diferentes edades, aptitudes en la venta y forma de trabajar que hacen muy complicada la gestión del departamento, y si a esto le añadimos el desplazamiento geográfico, en algunas ocasiones se hace casi imposible.
Estas necesidades nos ponen en primera instancia en resolver el registro y stock de productos vendidos, así como el calculo a fin de mes de las comisiones por cada vendedor. Actualmente, el proceso se lleva a cabo con una hoja excel que deben de rellenar los empleados, y por los diversos motivos indicados anteriormente, esta hoja no se cumplimenta correctamente ni en forma ni en tiempo.

Lo que plantemos como resultado a obtener, es una aplicación móvil que disponga de un interfaz claro, sencillo y por lo tanto amigable para cualquier usuario. Esta aplicación permitirá introducir al vendedor día a día, todas las

ventas realizadas de cada uno de los productos de los que dispone, así como obtener información de los productos, promociones y consulta de stock disponible. Incluimos como objetivo que se pueda introducir la venta ya sea de manera manual o utilizando mecánicas alternativas como por ejemplo: leyendo el código de barras del producto.

Por otro lado, queremos dar una vía de comunicación desde la empresa hacia sus vendedores a través del móvil, pero también queremos que la aplicación sea una vía de información bidireccional, desde el propio móvil hacia la empresa, casi a "tiempo real" de los productos que se están vendiendo. Esta información es muy útil para una empresa ya que haciendo procesos de minería de datos, pueden tomar estrategias comerciales y de marketing de sus productos en base a los ratio de consumo y ventas.

En definitiva, el objetivo principal de esta aplicación es la de proponer al teléfono móvil como una herramienta de trabajo desde, hacia y para la empresa así como para el vendedor.

1.2 Objetivos del Trabajo

A continuación, enumeramos los objetivos que pretenden alcanzarse con la ejecución de este trabajo:

- Crear una aplicación bajo la plataforma Android que permita la gestión y control de venta de productos para los vendedores.
- Registro sencillo y rápido de empleados con el rol vendedor.
- Controlar el stock de productos vendidos y pendientes.
- Calcular el coste de comisiones vinculadas a los productos vendidos.
- Recibir notificaciones de promociones y ofertas de productos.
- Recibir recordatorios del cumplimiento de la hoja de ventas.
- Enviar las ventas computadas en el mes al sistema centralizado o ERP.
- Gestionar el listado de productos en base al centro ubicado por geolocalización.
- Lectura de los productos con el código de barras para contabilizar las ventas

1.3 Enfoque y método seguido

Para llevar a cabo el proyecto utilizaremos la metodología de desarrollo en cascada. Se ha decidido utilizar esta metodología debido a la relación existente entre las entregas planificadas de la UOC y los procesos relacionados con el desarrollo del proyecto, aunque se ha adaptado ligeramente para cubrir correctamente la metodología.

Las metodologías en cascada se basan en el que las fases del proyecto se encuentran ordenadas en cascada y es estrictamente necesario finalizar la fase en curso para poder comenzar con la fase siguiente.
La metodología en cascada contienen las siguientes fases:

- **Análisis de requisitos.** En esta fase analizaremos los requisitos establecidos del funcionamiento esperado por la aplicación y que cubra las necesidades especificadas por lo usuarios.
- **Diseño**. En esta fase se realiza un diseño de alto nivel de las pantallas del aplicativo así como de la arquitectura empleada en el sistema-
- **Implementación.** Fase en la que se dedicará a tiempo completo a la codificación de la aplicación.
- **Verificación.** Fase dedicado al plan de pruebas y verificación funcional y de integración de los diferentes componentes arquitectónicos.
- **Finalización.** En esta fase se dedicará a la generación de la documentación relacionada con el Trabajo Final de Carrera, Presentaciones y cualquier otra documentación requerida que fuera necesaria para dar por cerrado el proyecto.

Diseño centrado en el usuario

Uno de los aspectos más importantes, que hay que tener en cuenta, es la usabilidad de la aplicación, sobre todo debido a que el objetivo de esta aplicación es la de ofrecer una alternativa utilizable por cualquier usuario de la compañía.

Por lo tanto, el diseño de interfaz de usuario deberá seguir los principios y buenas prácticas fundamentales para un diseño de interfaces para dispositivos móviles. Las consideraciones que hay que tener en cuenta es que sean:

- Simple, “menos es más”. Interfaces claros con pocos componentes visuales y atractivos para el usuario hacen que las aplicaciones sea más usables por parte del usuario.
- Intuitivo, la utilización de iconos visuales acercan intuitivamente al usuario el significado de lo que supuestamente realiza la acción.
- Diseño adaptado, existen múltiples versiones de tamaño de pantalla y cada vez estas son más grandes. El diseño debe adaptarse al tamaño de la pantalla y ofrecer la información clara
- Utilización de gestos, la aplicación de gestos con el móvil y asociadas acciones con el dispositivo ofrecen al usuario alternativas de utilización.

- Interactivo, dar una respuesta a los eventos del usuario, dan la sensación de fluidez de la aplicación.

1.4 Planificación del Trabajo

En este apartado se detallan los recursos necesarios para llevar a cabo el proyecto

1.4.1 Recursos necesarios

Indicamos a continuación los recursos necesarios tanto de hardware como de software para la ejecución del proyecto.

1.4.1.1 Software

- Entorno de desarrollo compuesto por
    - Entorno de desarrollo integrado - Eclipse IDE
    - ADT (Android Developer Tools) Bundle
    - Android SDK Platform
    - Librería zxuing (BarCode Reader)
    - Google Maps SDK
    - Java 1.7
    - Sistema gestor de base de datos SQLite
    - Google Plugin AppEngine for Eclipse
- Cuenta asociada en Google AppEngine

Para la elaboración de la documentación del proyecto se utilizará las herramientas ofimáticas libres de la familia LibreOffice (Writter, Calc, …).
Para la planificación y seguimiento del proyecto se hará uso de la herramienta GanttProyect, con licencia GPL .

1.4.1.2 Hardware

- Equipo de sobremesa Intel core I5 a 2,67 Mhz, 8 gb de ram con S.O. Windows 7 - 64 bits
- Dispositivo móvil de ultima generación con sistema operativo Android, OctaCore Mediatek, 2 Gb de ram

1.4.2 Listado de tareas a realizar:

Enumeramos el listado de tareas necesarias a realizar:

- Identificar, analizar y describir las funcionalidades y casos de uso del aplicativo.
- Identificar y describir los flujos funcionales y de navegabilidad del aplicativo desde la perspectiva del usuario
- Diseñar el interfaz gráfica de la aplicación en modo “wireframe” como prototipo.
- Crear una cuenta en AppEngine para alojar los servicios que se ejecutan en la nube.
- Crear scripts en el servidor de AppEngine para la recepción de ventas y consultas realizadas desde el dispositivo móvil.
- Generar los eventos en la nube para las notificaciones push y control del empleado con el rol de vendedor.
- Codificar el aplicativo en la tecnología Android.
- Planificación y ejecución de un plan de pruebas para validar el correcto funcionamiento del aplicativo.
- Elaboración de la memoria final y presentación del proyecto.

1.4.3 Planificación temporal

La siguiente tabla muestra los hitos y tareas principales de las que se componen el proyecto.

Código Hito	Nombre	Fecha Inicio	Fecha Fin	Duración en días
01	Análisis previo. Plan de trabajo.	17/09/2014	01/10/2014	15
01.01	Decisión del proyecto a realizar	17/09/2014	19/09/2014	3
01.02	Preparación y búsqueda de información.	20/09/2014	23/09/2014	4
01.03	Planificación del proyecto	24/09/2014	25/09/2014	2
01.04	Creación del documento: “Plan de trabajo”	26/09/2014	01/10/2014	6
01.05	Hito: Entrega PEC1	01/10/2014	01/10/2014	Hito
02	Análisis de requisitos	02/10/2014	12/10/2014	10
02.01	Análisis funcional y de arquitectura	02/10/2014	11/10/2014	10
03	Diseño	12/10/2014	29/10/2014	18
03.01	Diseño de las pantallas o WireFrame	12/10/2014	17/10/2014	6
03.02	Diseño de las arquitectura	18/10/2014	21/10/2014	4
03.03	Creación del documento	22/10/2014	29/10/2014	8
03.04	Hito: Entrega PEC2	29/10/2014	29/10/2014	Hito
04	Implementación y pruebas	30/10/2014	10/12/2014	42
04.01	Desarrollo aplicación móvil	30/10/2014	23/11/2014	25

04.02	Desarrollo datos en Cloud	24/11/2014	01/12/2014	7
04.03	Ejecución del plan de pruebas	02/12/2014	05/12/2014	4
04.04	Creación del documento	06/12/2014	09/12/2014	6
04.05	Hito: Entrega PEC3	10/12/2014	10/12/2014	Hito
05	Final del TFC	11/12/2014	09/01/2015	30
05.01	Redacción de la memoria final	11/12/2014	30/12/2014	20
05.02	Presentación visual	31/12/2014	09/12/2014	10
05.03	Hito: Entrega final	09/01/2015	09/01/2015	Hito

Diagrama de Gantt

Nombre	Fecha de inicio	Fecha de fin	Dura...
Análisis previo	17/09/14	1/10/14	15
Decisión del proyecto	17/09/14	19/09/14	3
Preparación y búsqueda ...	20/09/14	23/09/14	4
Planificación del proyecto	24/09/14	25/09/14	2
Creación del documento	26/09/14	1/10/14	6
Entrega PEC1	1/10/14	1/10/14	0
Análisis de requisitos	2/10/14	11/10/14	10
Análisis funcional y de arq...	2/10/14	11/10/14	10
Diseño	12/10/14	29/10/14	18
Diseño de pantallas	12/10/14	17/10/14	6
Diseño arquitectura	18/10/14	21/10/14	4
Creación del documento	22/10/14	29/10/14	8
Entrega PEC2	29/10/14	29/10/14	0
Implantación y pruebas	30/10/14	10/12/14	42
Desarrollo aplicación móvil	30/10/14	23/11/14	25
Desarrollo datos en Cloud	24/11/14	30/11/14	7
Plan de preubas	1/12/14	4/12/14	4
Documentación	5/12/14	10/12/14	6
Entrega PEC3	10/12/14	10/12/14	0
Finalización	11/12/14	9/01/15	30
Redacción de la memoria	11/12/14	30/12/14	20
Presentación visual	31/12/14	9/01/15	10
Entrega Final	9/01/15	9/01/15	0

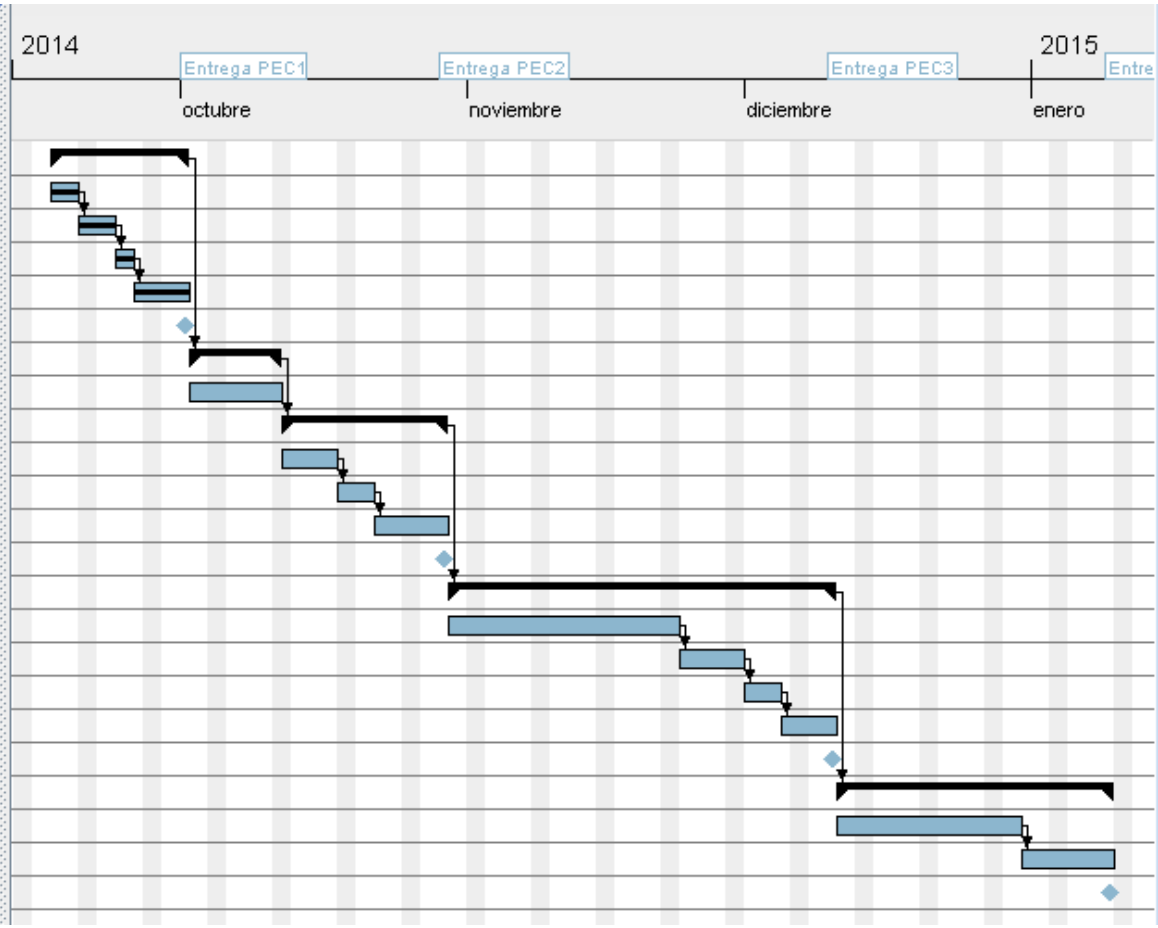


Ilustración 1: Gantt de la planificación y estimación del proyecto

Capítulo 2: Usuarios y contexto de uso

2.1 Métodos de indagación

Para obtener información precisa sobre el contexto de uso y las necesidades de los usuarios, se utilizaron varios métodos de indagación de la información. Estos métodos se han basado en el documentación aportada del Diseño Centrado en el Usuario.

Desde el primer momento se han identificado los usuarios principales de la aplicación, ya que es requisito de la empresa el de disponer de una herramienta y procedimiento más productivo y eficiente que el actual para los vendedores de la empresa, por lo que podemos centrar nuestros objetivos de obtención de información sobre estos perfiles sin necesidad de realizar un estudio previo del conjunto de posibles usuarios. Por lo tanto y por la información dispuesta por la empresa, la futura aplicación estará destinada a los usuarios con perfil o rol de vendedor.

Los métodos de indagación escogidos pretender obtener las necesidades y requisitos que se deberán satisfacer en la aplicativo móvil y que cumplan con las necesidades expuestas por los propios usuarios.

Se han seleccionado dos tipos de métodos para la obtención de la información, uno de ellos de tipo cualitativo basado en la observación y otro cuantitativo utilizando encuestas a los usuarios.

A continuación exponemos el planteamiento, desarrollo y resultados obtenidos en cada método.

Observación e Investigación Contextual

Durante la primeras fase de investigación se ha utilizado la técnica de investigación contextual en la que se observa a los usuarios mientras interaccionan con el actual sistema. Al ser un sistema utilizado por un grupo bastante amplio, podemos obtener información diversa de como es utilizado y que impedimentos o mejoras pueden ser aplicadas.

También nos ofrece información de como las tecnologías pueden suponer una barrera para algunas personas y como debemos solventar estos problemas en la nueva aplicación y evitar así la barrera tecnológica

Del resultado de la investigación obtenemos que:

- Utilizan una hoja de calculo para incluir las ventas realizadas de cada producto.
- En la hoja de calculo tienen que identificar el producto, el número de ventas realizadas de ese producto y el número de stock que queda en tienda. Esto lo tienen que realizar por cada línea de producto que hayan vendido y que corresponde a una fila de la hoja de calculo.
- La hoja de calculo devuelve como resultado de la introducción de los datos anteriores, la comisión que se llevará el vendedor.

- Cada mes cambian la hoja de calculo con las nuevas comisiones establecidas para el mes siguiente, en ocasiones estas hojas se reciben incluso fuera del mes, por lo que el empleado tiene que escribir las notas en un papel aparte que luego tiene que transcribir a la hoja de calculo.
- También cambian el formato de la hoja de calculo por lo que algunas veces no saben como deben rellenarlas o incluso contienen varias hojas lo que produce dudas de cual deben rellenar y su procedimiento.
- Hay usuarios que no saben como utilizar un hoja de calculo, ni saben para que sirve, aunque hacen el esfuerzo para tenerla cumplimentada a tiempo.
- Rellenar la hoja de calculo se hace un proceso largo y tedioso para el usuario, lo que conlleva a que se realice al final del mes, y en ocasiones fuera del plazo estipulado, por lo que la dirección y el departamento de contabilidad nunca dispone de toda la información a tiempo y actualizada correctamente.
- Suele existir discrepancias entre los elementos vendidos y los imputados en la hoja de calculo, algunas veces por despiste y otras por aumentar el número de ventas y comisiones.
- Las vacaciones las solicitan con otra hoja de calculo diferente, que envían por correo electrónico, en la mayoría de los casos no se acuerdan si han gastado o no todos los días disponibles, lo que les obliga a tener que llamar por teléfono para consultar el actual computo de sus vacaciones.
- No tienen un canal común para enviar las bajas, en algunas ocasiones lo envían por fax, o le sirve con llamar por teléfono y luego enviarlas con un mensajero que paga la propia empresa.
- Por otro lado, las nominas las reciben por correo electrónico y no tienen forma de poder volver a descargarlas.

Métodos de encuesta

En una segunda fase, hemos utilizado la técnica de investigación basada en encuestas. Los motivos de utilizar las encuestas se deben a que cierta parte de los empleados se encuentran desplazados geográficamente y no es viable físicamente, económicamente y por tiempo, observarlos en su propio entorno de trabajo, por lo que se decidió utilizar un método basado en las encuestas, que en parte solventa estos problemas.

Este método se planificó enviando un correo electrónico en el que se indicaba una dirección web donde se había publicado un formulario de encuestas basado en la plataforma gratuita de Google de formularios llamada Google Forms, y en la que te permite diseñar formularios de encuestas y obtener directamente el computo de las mismas dentro de una hoja de calculo para su posterior análisis.

Utilizar este sistema ha permitido el anonimato de la encuesta, por lo que los entrevistados se han visto libres de contestar con total sinceridad y sin sentirse incómodos al verse observados en su puesto de trabajo, tal y como ocurre con otros métodos de observación.

Las preguntas se diseñaron en base a los resultados obtenidos en el método de investigación contextual, en base para verificar que todos los usuarios tenían los mismos problemas o requerimientos independientemente de su lugar o condición

Incluimos en el siguiente punto las preguntas que se realizaron en la encuesta con Google Forms

Preguntas de la encuesta:

- ¿Qué edad tiene?: [edad]
- ¿Es hombre o mujer? [hombre o mujer]
- ¿Cuáles son sus estudios mínimos?: [Obligatorio, Bachillerato, Ciclo Formativo, Estudios superiores]
- ¿Tiene conocimiento de ofimática? [si, no]
- ¿Cuantos productos diferentes puede vender a lo largo del día? [1 a 5, 5 a 10, más de 10]
- ¿Dispone información del catalogo completo de productos.? [si, no]
- ¿Dispone de un detalle del producto? [si, no]
- ¿Sabe cuantos y cuales son los productos ha vendido a lo largo del mes? [si, no]
- ¿Puede obtener información de sus ventas en el último año? [si,no]
- ¿Qué mejoraría del actual sistema? [texto libre]
- ¿Cree que la forma de rellenar las ventas es rápido y eficaz? [si, no]
- Indique los motivos en caso negativo: [texto libre]
- ¿Dispone de teléfono móvil inteligente o smartphone? [si, no]
- ¿Qué móvil tiene? [Iphone, Samsung, Nokia, Lg, Otros]
- ¿Utiliza las redes sociales en el móvil? [si, no]
- ¿Utiliza el móvil a menudo? [si, no]
- ¿Tendría algún problema por utilizar el teléfono móvil en su puesto de trabajo? [si, no]
- ¿Le gustaría conocer o disponer de una herramienta para solicitar las vacaciones? [Si, No, tal vez]
- ¿Y para las bajas de trabajo? [Si, No, tal vez]
- ¿Y un lugar para descargar las nóminas? [Si, No, tal vez]

Resultados de la encuesta:

De la encuesta obtenemos los siguientes resultados:

Pregunta	Resultados
¿Qué edad tiene?	Entre 18 y 50 años
¿Es hombre o mujer?	Hombre: 30%

	Mujeres: 70%
¿Cuáles son sus estudios mínimos?	Obligatorio: 20% Bachillerato, 70% Ciclo Formativo: 4% Estudios superiores : 6%
¿Tiene conocimiento de ofimática?	SI: 40% NO: 60%
¿Cuantos productos diferentes puede vender a lo largo del día?	1 a 5: 20% 5 a 10: 55% 10 o más:25%
¿Dispone información del catalogo completo de productos.?	SI: 20% NO: 80%
¿Dispone de un detalle del producto?	SI: 20% NO: 80%
¿Sabe cuantos y cuales son los productos ha vendido a lo largo del mes?	SI: 40% NO: 60%
¿Puede obtener información de sus ventas en el último año?	SI: 0% NO: 100%
¿Qué mejoraría del actual sistema?	La facilidad de uso, que no lleve demasiado tiempo. Que se pueda realizar en cualquier lugar.
¿Cree que la forma de rellenar las ventas es rápido y eficaz?	SI: 0% NO: 100%
Indique los motivos en caso negativo:	En la mayoría de los casos aportan desconocimiento en el uso de la herramienta. Farragoso, poco útil y lleno de inconsistencias.
¿Dispone de teléfono móvil inteligente o smartphone?	SI: 12% NO: 88%
¿Qué móvil tiene?	Iphone 10% Samsung 57% Nokia 5% Lg: 10% Otros: 32%
¿Utiliza las redes sociales en el móvil?	SI: 78% NO: 22%
¿Utiliza el móvil a menudo?	SI: 85% NO: 15%

¿Tendría algún problema por utilizar el teléfono móvil en su puesto de trabajo?	SI: 10% NO: 90%
¿Le gustaría conocer o disponer de una herramienta para solicitar las vacaciones?	SI:65% NO: 23% Tal vez: 12%
¿Y para las bajas de trabajo?	SI: 72% NO: 20% Tal vez: 8%
¿Y un lugar para descargar las nóminas?	SI: 50% NO: 30% Tal vez: 20%

Recibimos un total de 30 encuestas realizadas. Una vez analizados los datos, obtenemos las siguientes conclusiones de la encuesta:

- Los empleados no saben utilizar el actual sistema de computo de ventas.
- Le parece engorroso y poco útil.
- Pierden mucho tiempo y en ocasiones no tienen información de a quién, cómo y cuando tienen que enviarlo.
- Saben utilizar un smarphone y la gran mayoría disponen de un teléfono con estas características.
- Además de disponer de un sistema de ventas, les gustaría poder acceder a la información del producto.
- Les gustaría poder disponer de un sistema de vacaciones y bajas del empleado, así como poder tener un lugar donde poder consultar las nominas del empleado.

2.2 Perfiles de usuario identificados

A lo largo de las fase de entrevistas y en base a las observaciones se obtuvieron los siguientes perfiles de la aplicación.

Perfil Vendedor

- Con franjas de edad variadas, entre 18 y 50 años.
- Con cocimientos de las nuevas tecnologías entre bajo y medio.
- Formación académica media/ alta. Con estudios mínimos de graduado escolar o formación profesional.
- Publico diverso y heterogéneo con diferentes tipos de inquietudes tanto profesionales como particulares.
- Con conocimientos ofimáticos entre bajo, medio.
- Con disposición colaborativa y con ganas de cambiar los procedimientos actuales,

2.3 Contextos de uso del aplicativo

Identificamos los siguientes contextos de uso del aplicativo:

- **Contexto espacial:** el usuario utilizará la aplicación en su lugar de trabajo, ubicado en un lugar público en el que se verá sometido a las obligaciones habituales de su propio trabajo tales como atender al público, formalizar una venta, por lo que el aplicativo sirve como apoyo a su propio trabajo.
- **Contexto temporal:** el usuario utilizará durante su jornada laboral en intervalos de tiempo que le permita su puesto de trabajo o durante momentos oportunos ocasionales para consultar información del producto, stock, etc. Ocasionalmente puede ser usada fuera del trabajo para consultar el estado de sus ventas.
- **Contexto usuario - emocional:** la aplicación debe ofrecer una experiencia de usuario agradable para la persona que la utilice. Tiene que ser elegante y atractiva, que inspire la ganas de utilizarla.
- **Contexto usuario - habilidades tecnológicas:** la aplicación deberá tener un UI sencillo, intuitivo, fácil de usar y de utilizar desde el primer momento, evitando pantallas innecesarias o eventos demasiado cargados. Debe contener elementos visuales gráficos que intuyan que la interacción confluye en la ejecución prevista por el evento representado. Ademas debe ser rápida en mostrar la información al usuario, ya que se encuentra en su puesto de trabajo y no debe demorar demasiado en el tiempo. Por estos motivos, el UI debe ser homogéneo y coherente en todas las pantallas, manteniendo la misma estructura de elementos a lo largo de toda la aplicación

2.4 Análisis de tareas

Describimos a continuación las tareas que necesitaran los usuarios en el aplicativo a desarrollar en base a los apartados anteriores.

- Registrar la aplicación con el empleado que accede a ella.
- Introducir día a día y producto a producto las ventas realizadas.
- Utilizar la cámara como lector de códigos de barras para la identificación y venta de un producto.
- Que sea rápido e intuitivo.
- Que obtenga la localización del centro que está trabajando por GPS sin necesidad de meter los datos, por si se trabaja en varios centros.
- Que calcule el total de comisiones mensual por producto vendidos.
- Obtener el stock del almacén y disponibilidad de elementos
- Solicitar stock al almacén central.
- Consultar información adicional del producto.
- Obtener la información de comisión por producto.

- Obtener un listado de las ventas por periodos de tiempo
- Poder consultar las vacaciones y solicitarlas
- Poder enviar bajas de trabajo por un medio o canal.
- Utilizar la cámara como elemento para registrar el comprobante de una baja.
- Poder descargarse las nominas.

2.5 Conclusiones finales

De los métodos aplicados obtenemos las siguientes conclusiones más relevantes:

- La gran parte de usuarios utilizan el teléfono móvil asiduamente.
- Hay una parte de los usuarios que no tienen conocimientos ofimáticos, o que nunca han utilizado antes un ordenador.
- Hay algunos usuarios que no tienen cuenta de correo electrónico o no es lo que más utilizan actualmente.
- La franja de edad es bastante amplia entre los 18 y 50 años.

Capítulo 3: Diseño conceptual

En el siguiente capítulo se describen cuales son los escenarios de uso o funcionalidades que se darán en el aplicativo. También se identifican los roles o actores del sistema que utilizarán el aplicativo.

Dentro de los escenarios de uso identificamos al siguiente perfil:

- **perfil vendedor**

3.1. Escenarios de uso

Perfil:	vendedor
Contexto	Se encuentra en su puesto de trabajo. Completa una venta de un producto, lo que implica contabilizar el producto vendido.
Objetivos	* Incluir venta * Incluir venta por código de barras.
Tareas que lleva a cabo	
Escribir en una hoja de papel el producto vendido, fecha de la venta y unidades.	
Necesidades de información	
Tener un listado de los productos vendidos de manera organizada y sin errores	
Funcionalidades que necesita	
Mantener de forma organizada las ventas realizadas por producto y día	
Como desarrolla estas tareas	
El usuario visualizará el listado de productos asociados a la tienda en la que se encuentra, localizará el producto por nombre o foto y pulsará en añadir venta. La venta quedará registrada en la fecha y hora realizada. Podrá decidir si la venta incluye el código de barras o de manera manual	

Perfil:	vendedor
Contexto	Se encuentra en su puesto de trabajo. En un momento determinado, el cliente solicita información adicional del producto que le está vendiendo.
Objetivos	Obtener información del producto
Tareas que lleva a cabo	
Busca dentro de los catálogos de los productos información que le pueda ser útil para la venta	
Necesidades de información	
Tener información detallada de los productos que vende	
Funcionalidades que necesita	
Obtener de manera eficaz una fuente de información de los productos que vende.	

Como desarrolla estas tareas
El usuario visualizará el listado de productos asociados a la tienda en la que se encuentra, localizará el producto por nombre o foto y pulsará en el botón ver detalle, en pantallas sucesivas se va mostrando los datos relativos al producto.

Perfil:	vendedor
Contexto	Se encuentra en su puesto de trabajo. Observa que no lo queda stock del producto en tienda por lo que necesita que le sea enviado unidades del producto.
Objetivos	Solicitar stock del producto.
Tareas que lleva a cabo	
Comprueba las ventas realizadas y el número total de unidades apuntados. La resta de ambas unidades le indica el stock restante	
Necesidades de información	
Tener una manera de saber el stock restante y el que queda en la sede central para realizar peticiones de unidades	
Funcionalidades que necesita	
Llevar un control de las unidades, stock y cuando se recibirá el stock.	
Como desarrolla estas tareas	
El usuario visualizará el listado de productos asociados a la tienda en la que se encuentra. Solicitará que le envíen unidades del stock de un producto en concreto. Al cabo de un tiempo, recibirá una confirmación al móvil indicando la fecha de entrega del stock.	

Perfil:	vendedor
Contexto	Se encuentra en su puesto de trabajo o en su casa y desea conocer las comisiones que lleva ganadas
Objetivos	Consulta de comisiones
Tareas que lleva a cabo	
Muestra un listado de las comisiones y ventas realizadas por semana o mes	
Necesidades de información	
Tener una forma de conocer las ventas y comisiones que se llevará el vendedor en el mes en curso, o las que ha recibido de meses anteriores.	
Funcionalidades que necesita	
Tener una forma de conocer las ventas y comisiones	
Como desarrolla estas tareas	
El usuario accede a un listado de informes por pantalla, seleccionará la opción de listado por mes o semana. La información se presentará en pantalla en el orden y formato seleccionado anteriormente.	

Perfil:	vendedor
Contexto	Se encuentra en su puesto de trabajo o en su casa y desea solicitar las vacaciones
Objetivos	Solicitar vacaciones
Tareas que lleva a cabo	
Se muestra un calendario y se van marcando día a día las vacaciones	
Necesidades de información	
Tener un medio por el que disponer de las vacaciones	
Funcionalidades que necesita	
Poder solicitar y visualizar las vacaciones solicitadas	
Como desarrolla estas tareas	
El usuario accede a un calendario y va seleccionando aquellos día que solicita como vacaciones.	

Perfil:	vendedor
Contexto	Se encuentra fuera de su puesto de trabajo y necesita comunicar una baja
Objetivos	Comunicar baja
Tareas que lleva a cabo	
Se muestra un formulario para enviar la baja, si es requerido puede fotografiar la baja y enviarla adjunto a la petición	
Necesidades de información	
Poder confirmar que se ha enviado un baja.	
Funcionalidades que necesita	
Poder tener la forma de comunicar a la empresa una baja y que ese pueda adjuntar una foto de la misma.	
Como desarrolla estas tareas	
El usuario accede a la aplicación y selecciona su perfil, en este opción tendrá un botón de enviar baja. Escribirá un texto y adicionalmente podrá realizar una fotografía con el justificante de baja.	

3.2. Flujos de interacción

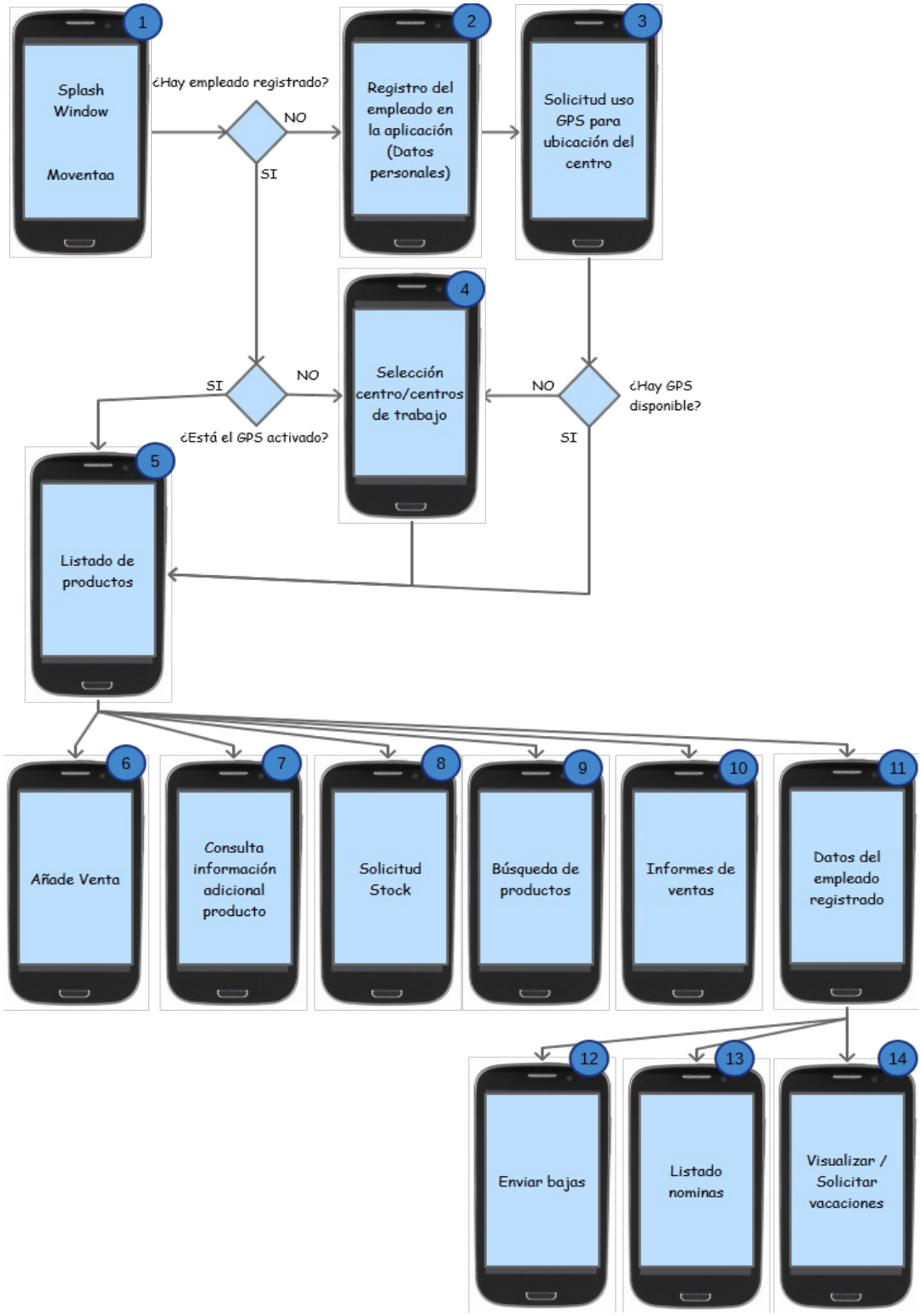


Ilustración 1: Flujos de interacción

Capítulo 4: Prototipado

4.1 “Sketches” escaneados

Ilustración 2: Scretches escaneado 1

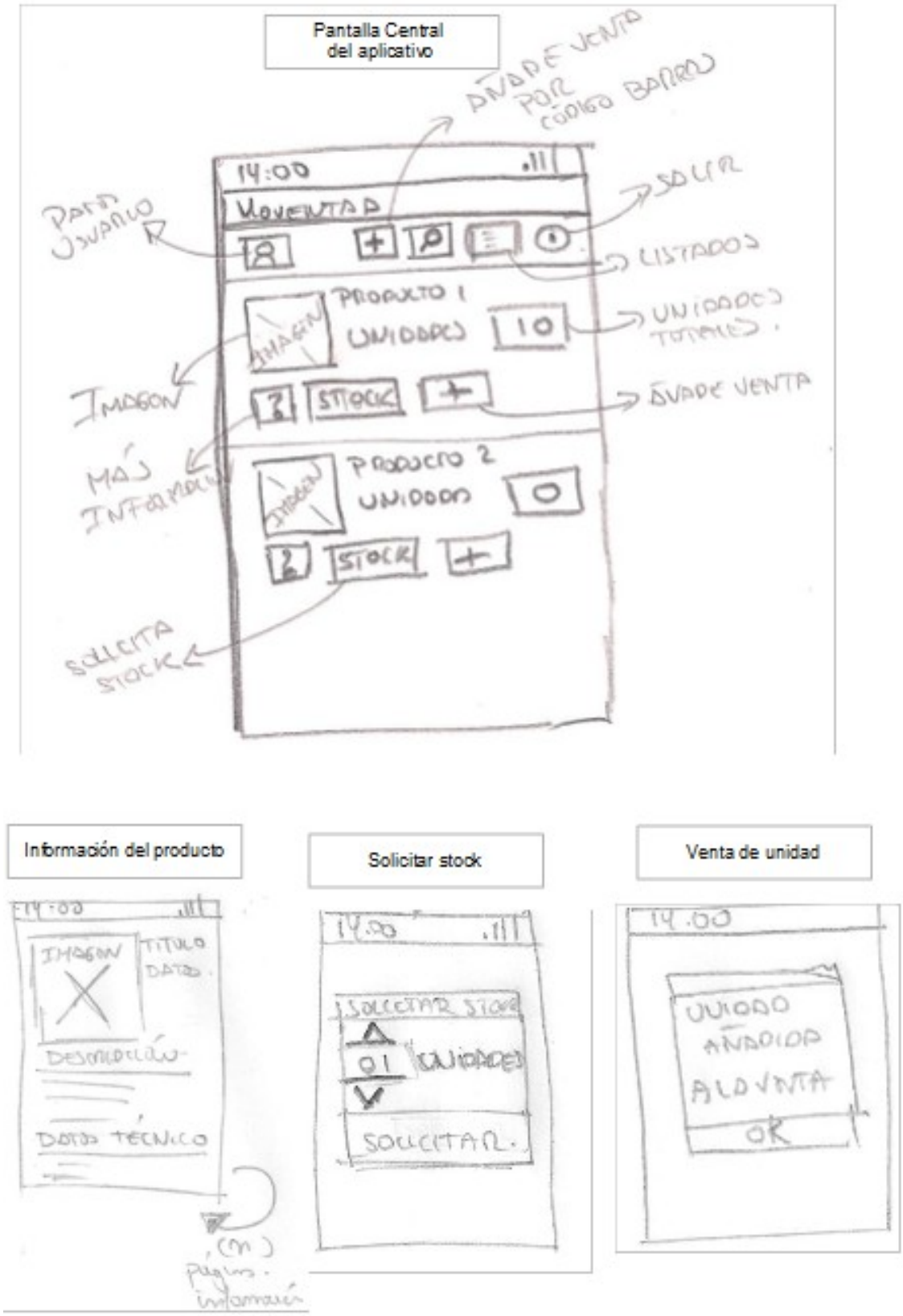


Ilustración 3: Scretches escaneado 2

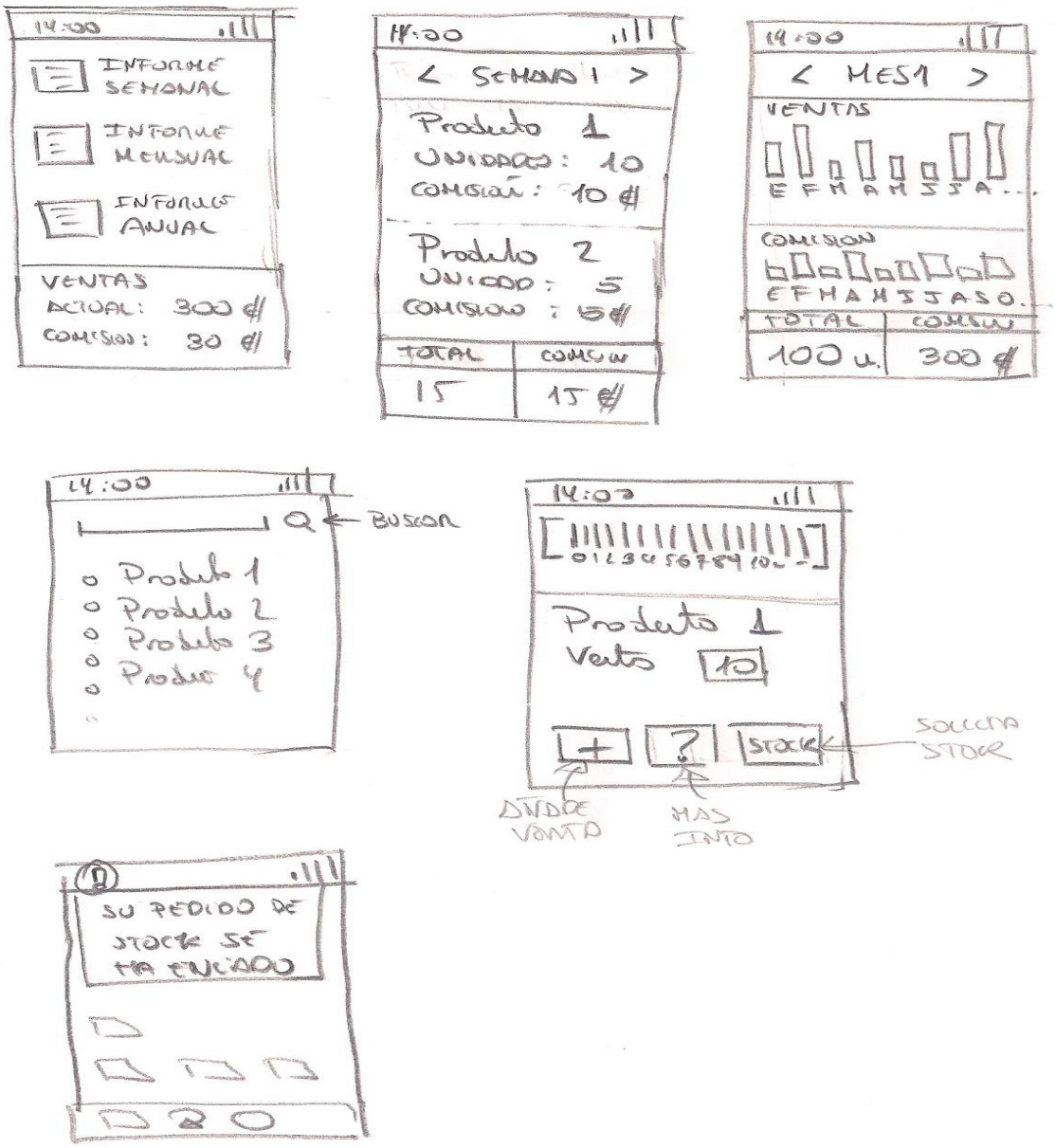


Ilustración 4: Scretches escaneado 3

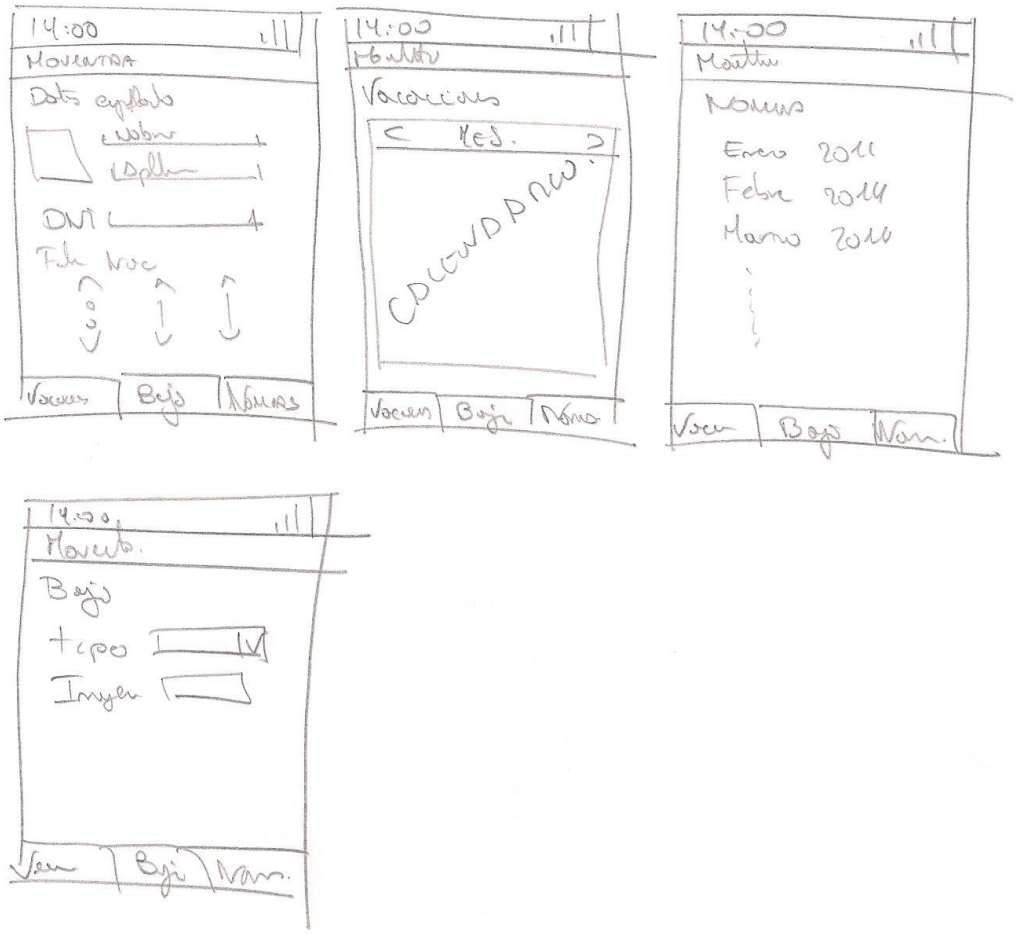
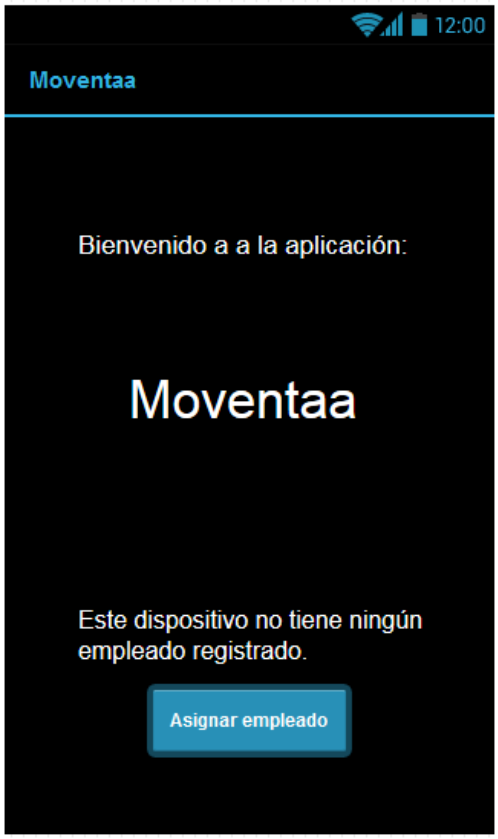


Ilustración 5: Scretches escaneados 4

4.2 Prototipo horizontal

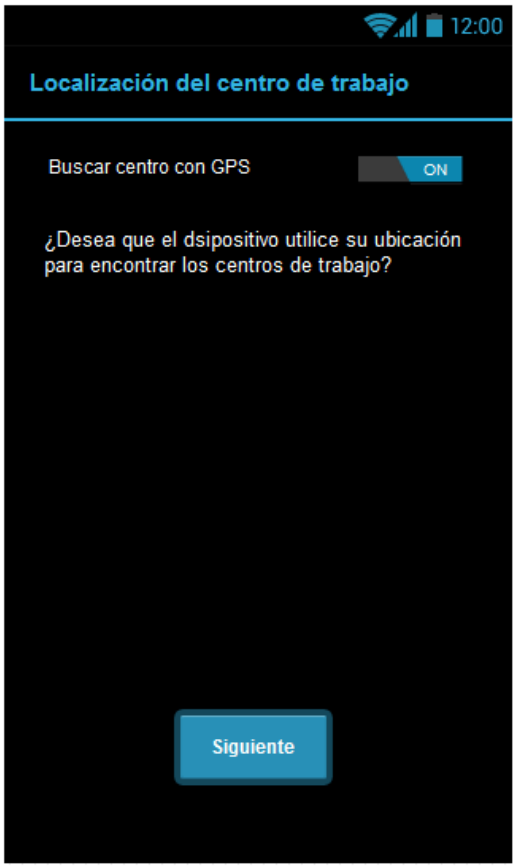


Pantalla de inicio de la aplicación

Esta es la primera pantalla que se muestra cuando se inicia la aplicación, en el caso que no haya un empleado/vendedor registrado se muestra un mensaje y botón en el que se indica que es necesario registrar un empleado

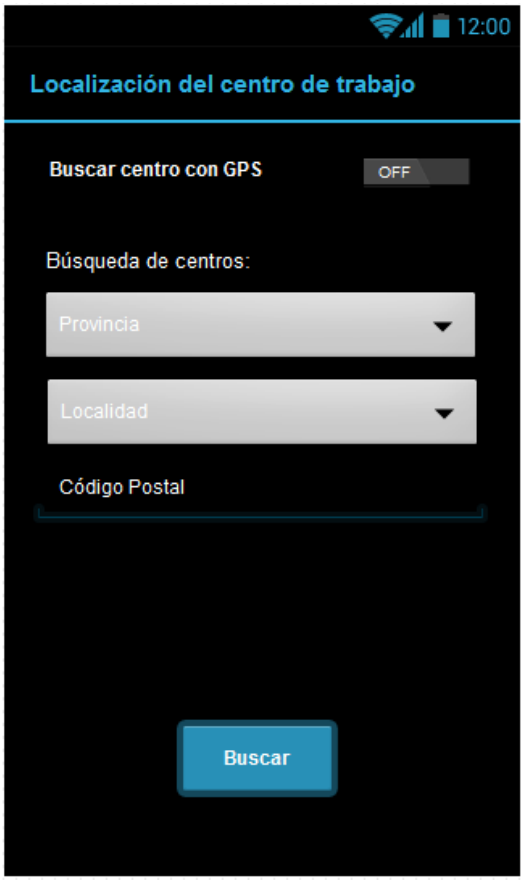
Pantalla de registro

El usuario introduce sus datos personales como comprobante que es un empleado de la empresa y puede utilizar la aplicación



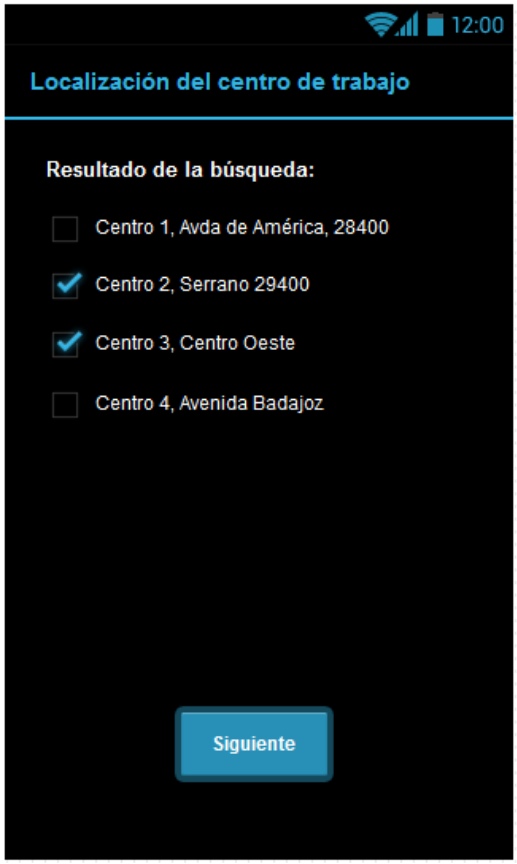
Búsqueda de centro asociado

Desde esta pantalla el usuario puede utilizar su Gps para localizar el centro en el que está trabajando.



Búsqueda de centro sin uso de GPS

Si el usuario no quiere utilizar el GPS o el teléfono no dispone de este hardware podrá seleccionar el centro haciendo una búsqueda por provincia, localidad y código postal

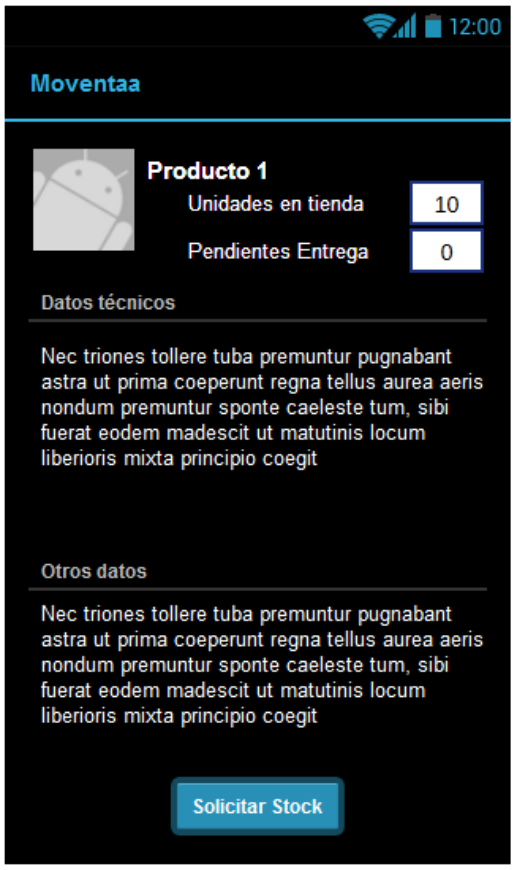


Selección del centro

En esta pantalla el usuario selecciona el centro o centros a los que esta asociado como vendedor.

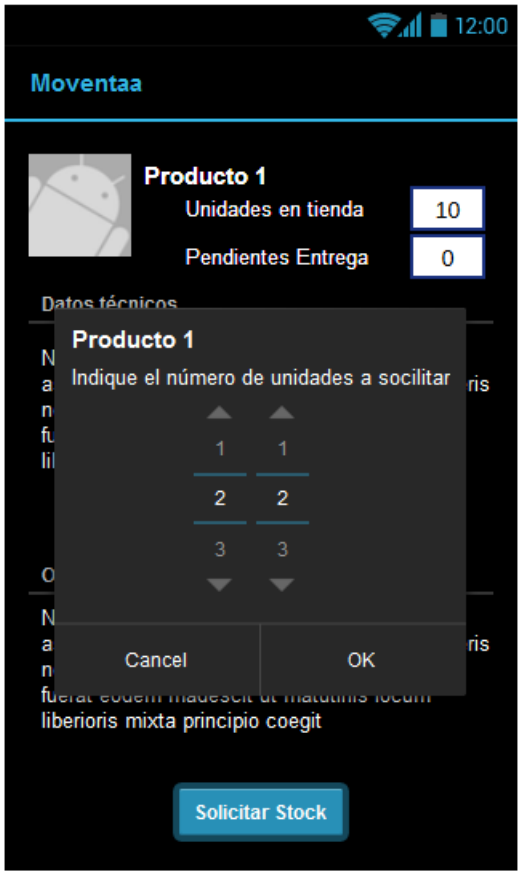
Pantalla principal de la aplicación

Desde esta pantalla se gestionan los operaciones de la aplicación, tales como gestionar la venta de un producto, su información, solicitar stock, búsqueda, informes y venta por código de barras así como acceso a los datos de la cuenta.



Información del producto

Se muestra toda aquella información relativa a un producto.



Solicitud de stock

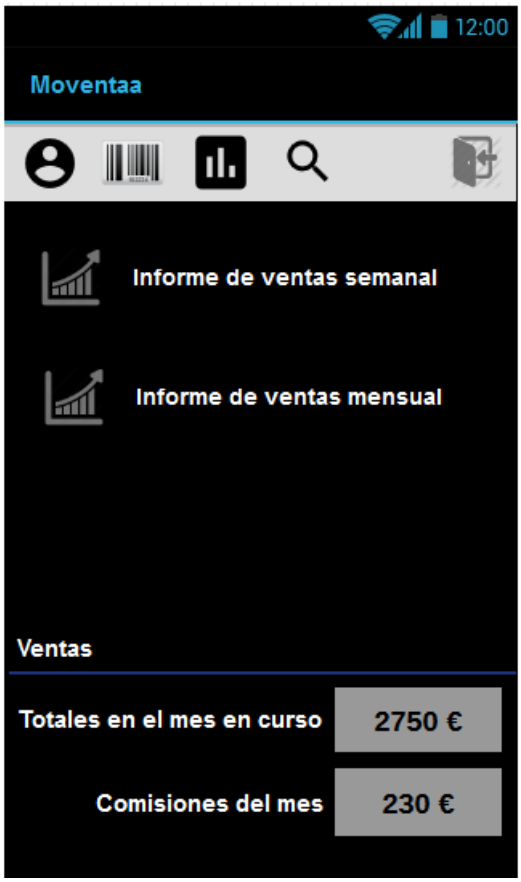
Pantalla para solicitar stock de un producto en particular

Añadir venta por código de barras

Desde esta pantalla se puede utilizar la cámara del móvil como escáner de códigos de barras, una vez detectado el código, la venta será añadida al total de unidades vendidas una vez se pulse el botón venta. También es posible acceder a información o solicitar stock para el producto detectado.

Venta de unidad

Pantalla que se accede al pulsar “Venta” desde el producto en la que hay que indicar el número de unidades vendidas.

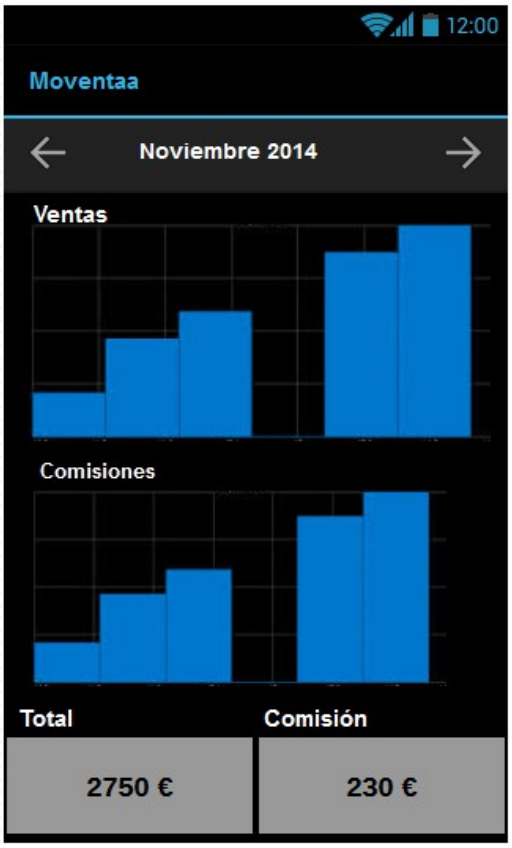


Informes

Pantalla de selección de informes semanal o mensual

Informe semanal

Pantalla con datos del informe semanal por producto con totales de comisiones y ventas



Informe mensual

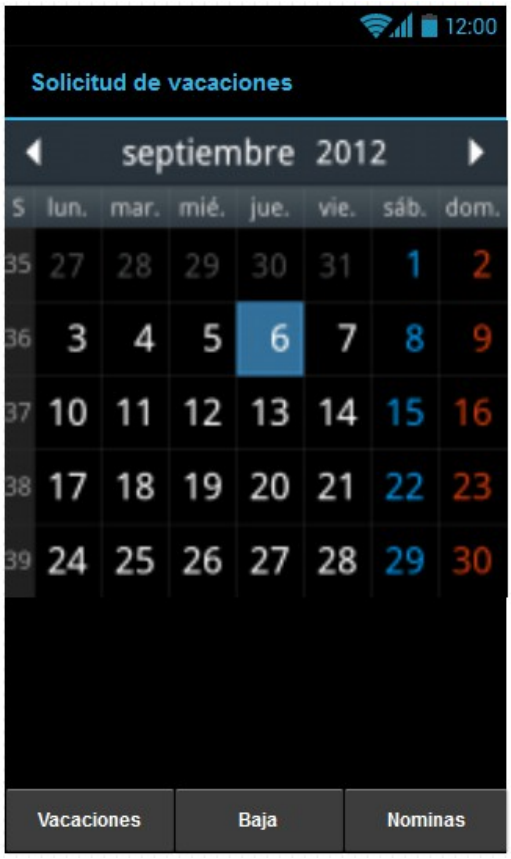
Informe mensual con gráficas de ventas

Búsqueda de producto

Pantalla para la búsqueda de productos por nombre. Pulsando sobre el nombre se accede a la información del producto.

Pantalla datos del usuario

Desde esta pantalla el usuario puede modificar sus datos peresonales y solcicitar las vacaciones, baja o listar sus nominas.

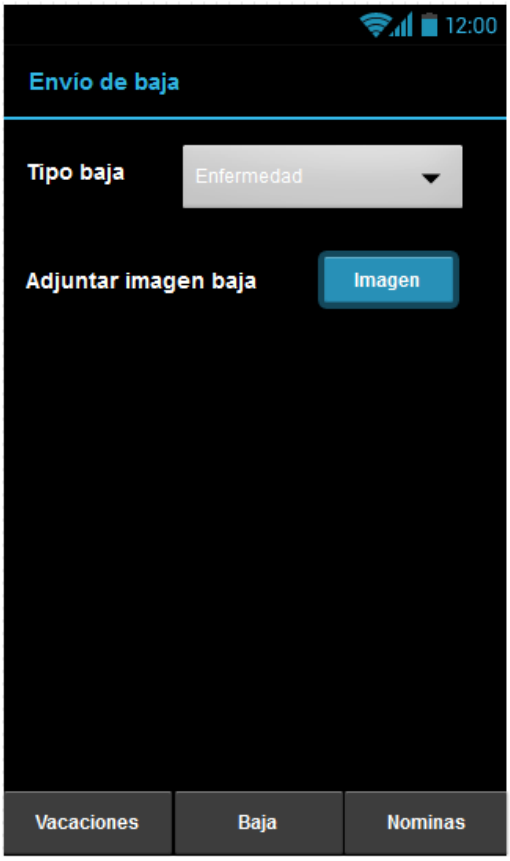


Solicitud de vacaciones

El usuario seleccionará aquellos días que solicita como vacaciones. También puede ver las vacaciones marcadas ya disfrutadas en meses anteriores o modificar las existentes no disfrutadas.

Listado de nóminas

El usuario puede seleccionar la nómina de un mes y verla o descargarla en su móvil.



Envío de baja

El usuario puede enviar una baja, con el motivo e imagen adjunta de una foto o de la galería de su móvil.

Capítulo 5: Evaluación

5.1 Diseño del test de usuario

La evaluación del test de usuario se realizará de manera presencial en un entorno amigable, en el que el usuario no se vea sometido a presiones o interpretaciones de sus opiniones al respecto del prototipo presentado.

La realización del test de usuarios se realizará de un grupo de 8 usuarios sobre un total de 33 posibles usuarios, en el que se intentarán que estén reflejados todos los perfiles detectados.

Se tomará como base de escenario para las pruebas el prototipo de alta fidelidad incluido en el apartado “Prototipo Horizontal” del presente documento, siendo el evaluador el que actúa simulando el comportamiento de la máquina tal y como se indica en el método: “Wizard of Oz Techinque”.

Se aplicará el protocolo de pensamiento manifiesto, según el cual el usuario dirá en voz alta lo que va pensando de la aplicación y que aspectos pueden ser mejorados.

Adicionalmente a la prueba del test realizaremos un cuestionario con el usuario que nos permita indagar y obtener información que durante el test no se haya cubierto.

5.2 Las tareas que los usuarios deben realizar

El objetivo de todo test de usabilidad es conocer el grado de facilidad de uso que presenta la aplicación, si ademas cumple con los estándares dictaminados por los consorcios dedicados al estudio de la usabilidad, así como el nivel de eficacia, productividad y satisfacción que genera al usuario.

Dentro de estas tareas especificaremos los objetivos necesarios que queremos que se cumplan en base a los estándares dispuestos de grados de usabilidad, navegabilidad, disposición de la información y diseño.

Test de la disposición de la información

La disposición de la información debe ser clara y concisa que no queda lugar a duda y que pueda verse en cualquier resolución de pantalla, en aquellos caso que por tamaño no se visualice al completo podrá utilizarse pantallas adicionales para su visualización. El usuario indicará si le parece ordenada la forma y contenido de la informaciñon

Test de la navegabilidad

En este test los usuarios navegarán por todas las opciones disponibles de los menús y enlaces, comprobando que las características de navegabilidad corresponden con los criterios establecidos y que no hay caminos sin recorrer o callejones sin salida. Ademas se estudiará si el acceso a todas las opciones se consigue de una manera intuitiva y clara para el usuario.

Test de la presentación y diseño de la aplicación

Puesto que el usuario pasará parte de su tiempo visualizando la aplicación, esta debe ser agradable para la vista, que no produzca cansancio o necesidad de enfocar a diferentes partes de la aplicación. También se analizará si para el usuario tanto el diseño como la iconografÍa utilizada representa el tipo de

acción y si es de manera intuitiva o si era la función que se esperaba al interaccionar con ella.

5.3 Cuestionario relacionado con las tareas del usuario.

Como parte final del proceso de evaluación solicitamos a un grupo de 10 usuarios que nos cumplimenten un cuestionario sobre el diseño y boceto de la aplicación

Incluimos las preguntas que están relacionadas con los puntos de criterio de los test diseñados:

- ¿Cree que la aplicación es fácil de utilizar? [0,1,2,3,4,5] 0 poco fácil, 5 muy fácil.
- ¿Qué opinión tiene de la aplicación? [0,1,2,3,4,5] 0 pésima, 5 excelente.
- ¿Le parece que tenga un diseño agradable? [0,1,2,3,4,5] 0 poco agradable, 5 muy agradable
- ¿Ha encontrado opciones complicadas o no comprensibles? [0,1,2,3,4,5] 0 poco, 5 mucho
- ¿Cree que la aplicación solucionará los problemas actuales? [Si,No]
- ¿Cree que mejorar su productividad en el trabajo? [Si,No]

Resultado del cuestionario:

Pregunta	Resultados
¿Cree que la aplicación es fácil de utilizar?	La media se encuentra en 3,5
¿Qué opinión tiene de la aplicación?	La media se encuentra en 4
¿Le parece que tenga un diseño agradable?	La media se encuentra en 3,2
¿Ha encontrado opciones complicadas o no comprensibles	La media se encuentra en 2,4
¿Cree que la aplicación solucionará los problemas actuales?	SI: 80% NO: 20%
¿Cree que mejorar su productividad en el trabajo?	SI: 78% NO: 22%

5.4 Resultado de la evaluación

Del resultado del cuestionario concluimos que la propuesta ha tenido una buena acogida, el interfaz les parece agradable y todo entienden que es una gran oportunidad de cambiar el actual procedimiento.

En rasgos generales, los encuestados estarían dispuestos a utilizara inmediatamente la propuesta de la aplicación presentada.

Capítulo 6: Diseño técnico de la aplicación

6.1 Diseño de la arquitectura

Para la construcción de la aplicación utilizaremos la arquitectura de servicios en la nube o PaaS de Google App Engine (a partir de ahora lo denominaremos como GAE)

GAE proporciona todo lo necesario para poder desarrollar aplicaciones en la nube, siendo transparente para el desarrollador evitándole los problemas relacionados con el montaje de servicios web, base de datos o otro tipo de arquitecturas relacionadas. Incluye ademas, servicios para el desarrollo de aplicaciones móviles, tales como Enpoints, Google Clound Message (GMC), servicios de autenticacion (Oauth), etc.

Arquitectura de la aplicación diseñada:

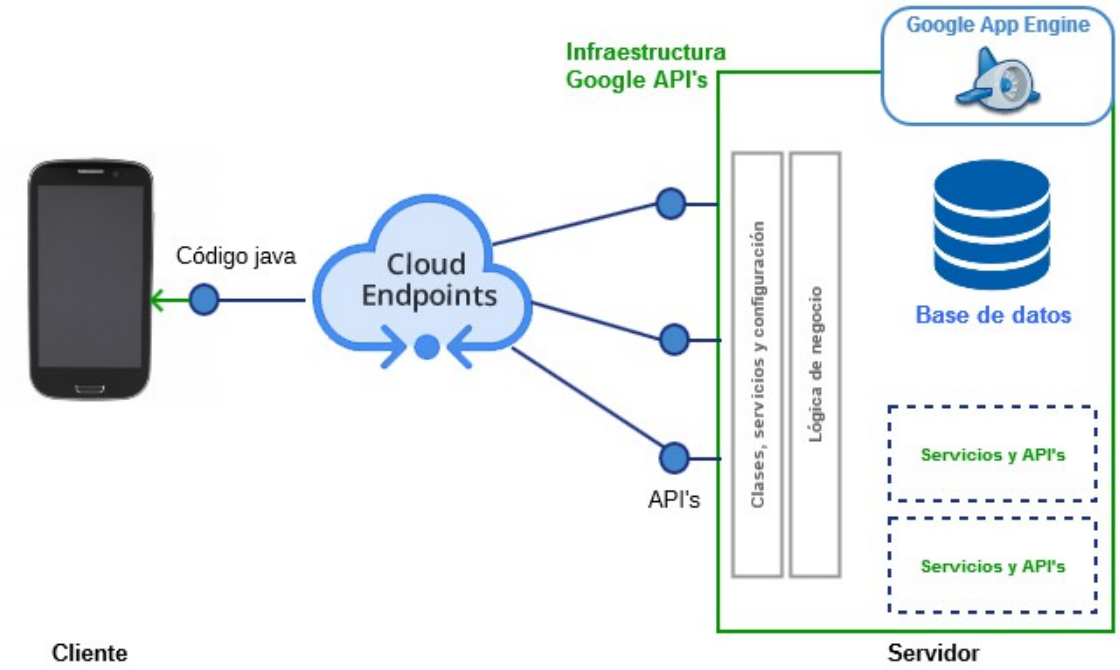


Ilustración 6: Arquitectura

Como podemos ver en la ilustración, en el servidor de GAE se ubican la base de datos, la infraestructura de servicios y las API's de conexión y publicación de servicios hacia el exterior, o denominados Endpoints. Estos Endpoints podrán ser consumidos por cualquier aplicación que tenga los interfaces necesarios para procesar datos en formato JSON, en nuestro caso, será la aplicación móvil quien consumirá estos servicios, ya sea obteniendo información de ellos o solicitando información especifica haciendo llamadas por peticiones RESTFull.

Hay que tener en cuenta que nuestro desarrollo estará compuesto de dos modelos, la parte cliente, que es el software diseñado para el móvil, y la parte

servidora, donde alojaremos la base de datos y la lógica asociada a los objetos del negocio.

Este tipo de diseño va a permitir una alta flexibilidad frente a los cambios, si se diera el caso de modificar la parte servidora, la parte cliente no debe verse afectar ya que el consumo de la información lo realiza por una capa transparente de servicios que debe ser robusta y soportar los cambios del modelo.

6.1.1 Arquitectura a nivel de servidor

Modelo-Vista-Controllador (MVC)

La arquitectura base de GAE está diseñada como un MVC (Modelo-Vista-Controlador).

En GAE, el Modelo (datos) se basa en la capa de persistencia JPA (Java Persistence API) + JDO (Java Data Objects). Esto significa que la persistencia se realiza a nivel de representación del objeto de la vida real, un objeto de tipo "Entity", lo que facilita al desarrollador al abstraer la capa de base de datos del propio diseño de almacenamiento posibilitando por tanto la flexibilidad de utilizar cualquier motor de base de datos.

En la capa Controladora, es donde se definen los Endpoints del GAE, que son servicios de consumo de información, que devuelven los datos en formato JSON.

La capa de presentación o Vista es la delimitada por la visualización del objeto JSON.

Base de datos

Ya que utilizamos la infraestructura de GAE, la base de datos es de tipo BigTable. Estas bases de datos tiene una característica muy importante, utiliza como motor de base de datos NoSQL.

Los motores NoSQL difieren del modelo clásico de las base de datos tradicionales, en que no utilizan relaciones de entidades en su diseño o esquema.

6.1.2 Arquitectura aplicación cliente

A nivel de cliente se ha implementado el modelo de diseño de arquitectura MVP (Model-View-Presenter).

El modelo MVP se basa en separar las capas de Vista y Modelo con una capa intermedia denominada Presenter. Esta ultima actúaa como bus de comunicaciones entre el modelo y la vista siendo la responsable de la solicitud de datos a través de los servicios implementados y de llamar a los metodos de la vista para su visualización de contenido

A parte de las ventajas inherentes de separar el modelo en capas, sobre la capa Presenter podemos diseñar pruebas unitarias que certifiquen el correcto funcionamiento de las firmas de los métodos implementados

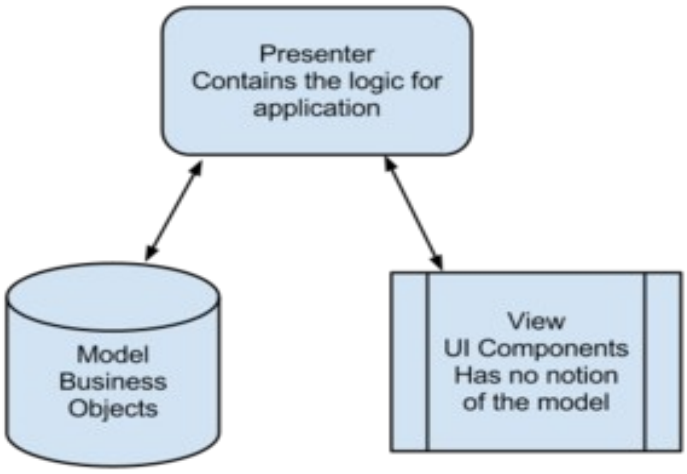


Ilustración 7: Model-View-Presenter

Fuente: Wikipedia DE

6.1.3 Servicios WEB (SOA)

La comunicación existente entre el cliente (móvil) y el servidor (GAE) recibe otro nombre de patrón de diseño denominado como SOA.

Este paradigma nos permite diseñar y desarrollar sistemas distribuidos de aplicaciones. Aplicándolo al desarrollo de esta aplicación, el cliente (móvil) es nuestra aplicación distribuida, y el servidor, los servicios centralizados.

Todas las peticiones del aplicativo se realizan sobre capa de transporte HTTP añadiendo por encima la capa de servicios REST que finalizan en llamadas a los endpoints declarados en el servidor

6.2 Diagrama UML/Entidad

Google App Engine utiliza como motor persistencia la implementación de “JPA API persistence” basando su modelo en POJO's, separando la capa de base de datos con la implantación del código siendo transparente para el desarrollador que motor de base de datos esta utilizando.

Por lo tanto el diseño del diagrama entidad relación se basa en la implementación de JPA como modelo de persistencia.

Diagrama Entidad-Relación

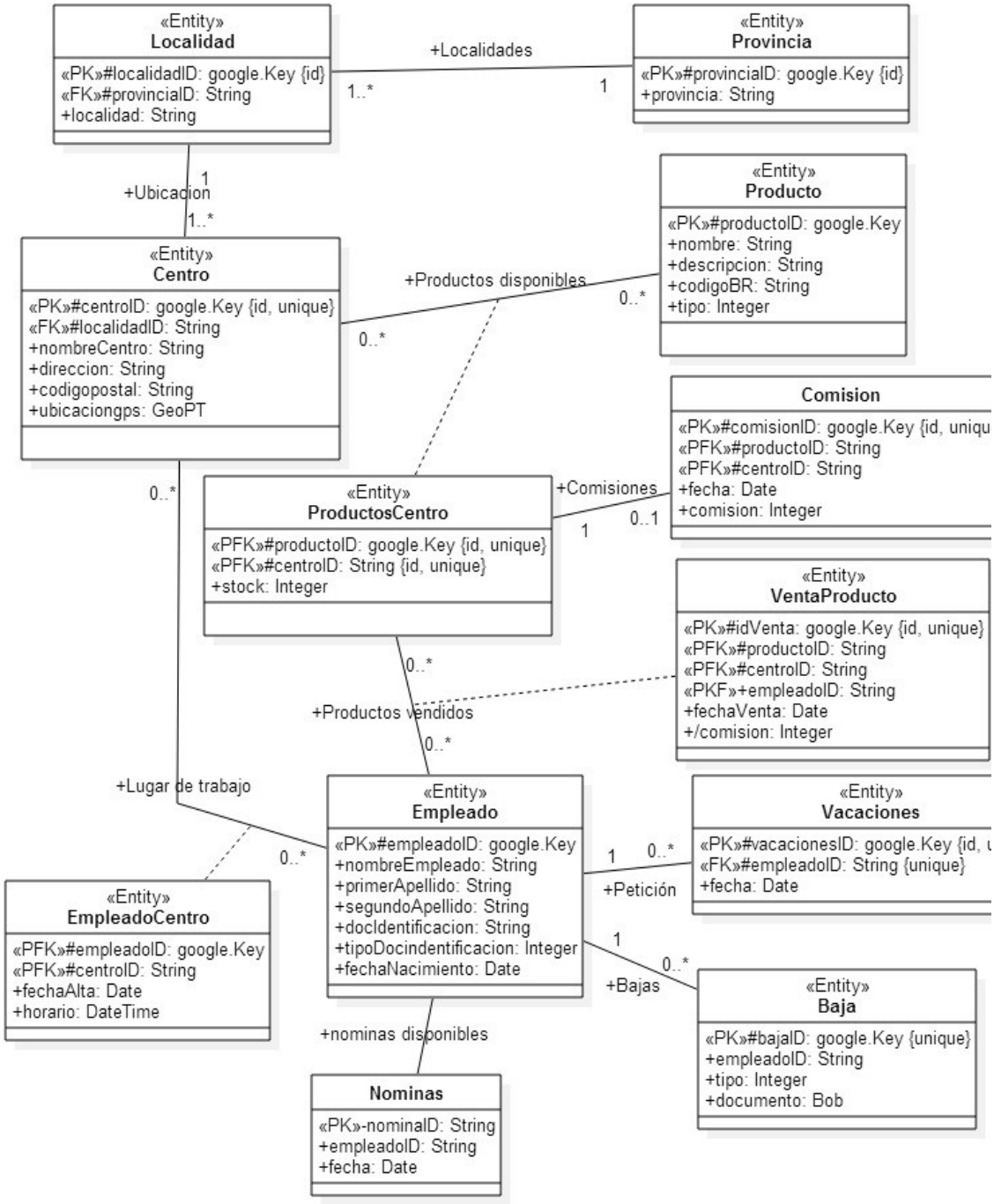


Ilustración 8: Diagrama Entidad Relación

Descripción de las entidades utilizadas en el diseño:

Entidad	Provincia
Descripción	Contiene un listado de las provincias de España
Clave primaria	provinciaID
Clave foránea	No tiene

Entidad	Localidad
Descripción	Contiene un listado de las localidades organizadas por provincias
Clave primaria	localidadID
Clave foránea	ProvinciaID → Provincia.provinciaID

Nota: por motivos de rendimiento y tamaño de la aplicación, solo se han cargado aquellas localidades de la provincia de Madrid

Entidad	Centro
Descripción	Centros organizados por localidad
Clave primaria	centroID
Clave foránea	localidadID → Localidad.localidadID

Entidad	Producto
Descripción	Listado de todos los productos
Clave primaria	productoID
Clave foránea	No tiene

Entidad	ProductosCentro
Descripción	Entidad N<a>N que relaciona los productos que son vendibles por centro.
Clave primaria	productoID,localidaID (clave primaria multiple)
Clave foránea	No tiene

Entidad	Comision
Descripción	Comision que se le atribuye a la venta de un producto en un centro en particular
Clave primaria	comisionID
Clave foránea	productoID → ProductosCentro.productoID centroID → ProductosCentro.centroID

Entidad	VentaProducto
Descripción	Venta realizada de un producto por un empleado
Clave primaria	idVenta

Clave foránea	productoID → ProductosCentro.productoID centroID → ProductosCentro.centroID empleadoID → Empleado.empleadoID

Entidad	Empleado
Descripción	Entidad donde se almacenan los empleados
Clave primaria	empleadoID
Clave foránea	No tiene

Entidad	Vacaciones
Descripción	Entidad donde se almacenan las solicitudes de vacaciones por parte de un empleado
Clave primaria	empleadoID
Clave foránea	empleadoID → Empleado.empleadoID

Entidad	Baja
Descripción	Entidad donde se reciben las bajas enviadas por los empleados con las fotos de la baja
Clave primaria	bajaID
Clave foránea	empleadoID → Empleado.empleadoID

Entidad	EmpleadoCentro
Descripción	Entidad donde se relacionan los empleado con los centros donde pueden trabajar. Es una entidad del tipo N < a > N
Clave primaria	empleadoID → Empleado.empleadoID, centroID → Centro
Clave foránea	No tiene

Entidad	Nominas
Descripción	Entidad donde se publican las nóminas que pueden ser descargadas por el empleado
Clave primaria	idNomina
Clave foránea	empleadoID → Empleado

6.3 Diagrama UML diseño del software

Diferenciamos dos tipos de diseño, el utilizado en la parte servidora y el utilizado en la parte cliente.

6.3.1 Diseño en servidor

GAE proporciona un plugin para Eclipse que permite la generación automática de Endpoints, estos Endpoints incluyen las operaciones básicas CRUD (Create , Read, Update, Delete) sobre una entidad.

Adicionalmente a estas operaciones, se han modelado nuevas firmas de métodos en los Enpoints relacionados con la lógica de negocio implementada para la aplicación cliente. Estos métodos adicionales se han incluido en el Endpoint denominado MoventaaEndpoint.

Endpoints publicados para el cliente móvil

Clase	MoventaaEndpoint
Descripción	Endpoints con métodos de la lógica de negocio común a la aplicación
Paquete	com.example.moventaa.endpoint

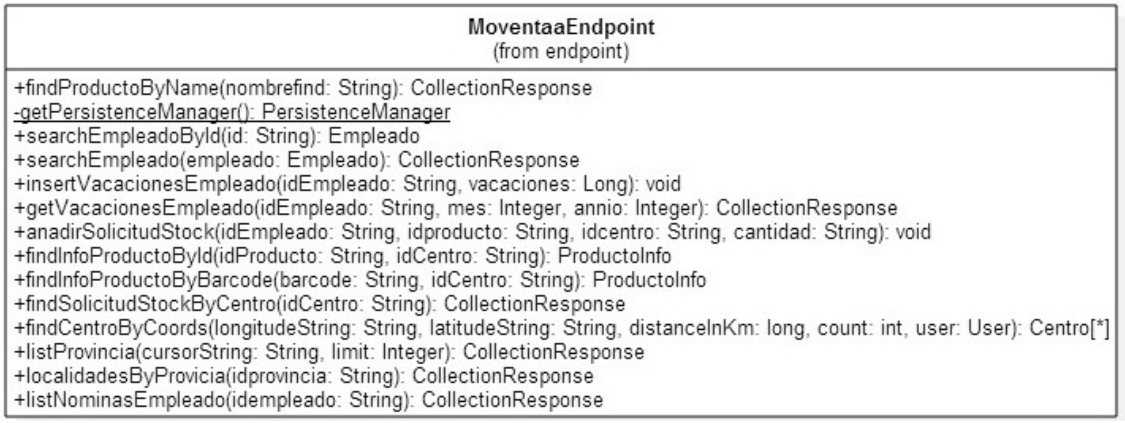


Ilustración 9: Diagrama UML: MoventaaEndpoint

Clase	CentroEndpoint
Descripción	Relacionado con la busqueda de centro por filtro y operaciones CRUD
Paquete	com.example.moventaa.endpoint

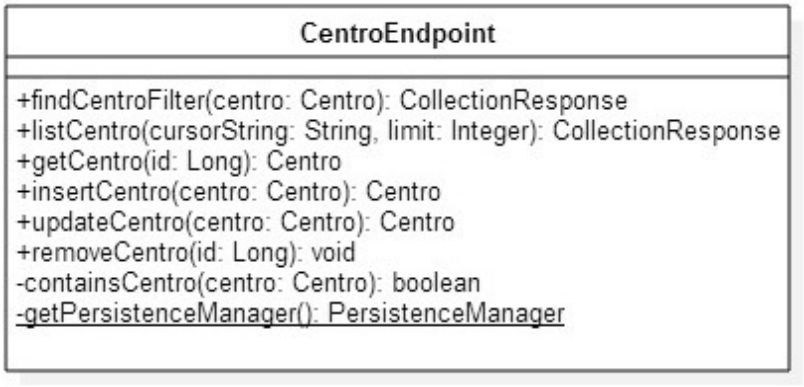


Ilustración 10: Diagrama UML: CentroEndpoint

Clase	VentaProductoCentroEndpoint
Descripción	Enpoint cuya logica de negocio esta relacionada con las ventas relacionadas en un centro de sus productos
Paquete	com.example.moventaa.endpoint

Ilustración 11: Diagrama UML: VentaProductoCentroEndpoint

Clase	ProductoEndpoint
Descripción	Enpoint relacionado con la búsqueda de información de un producto
Paquete	com.example.moventaa.endpoint

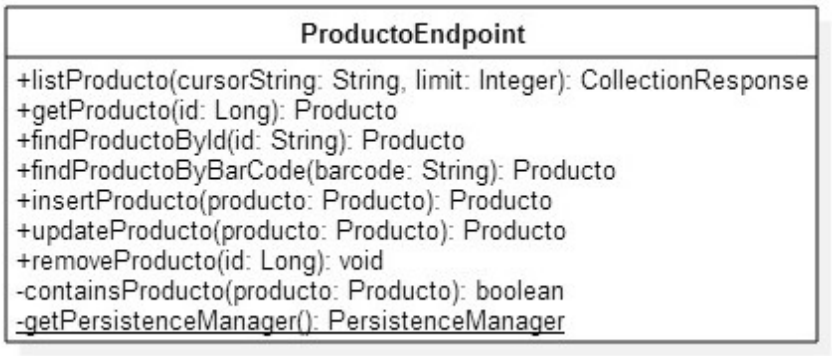


Ilustración 12: Diagrama UML: ProductoEndpoint

Clase	ProductoCentroEndpoint
Descripción	Relacionado con la búsqueda de productos dados en un centro
Paquete	com.example.moventaa.endpoint

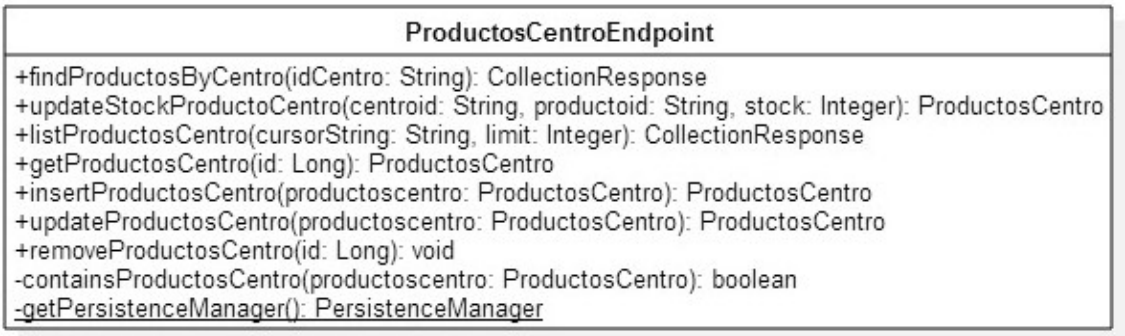


Ilustración 13: Diagrama UML: ProductoCentroEndpoint

Endpoints publicados internos para la gestión operativa

Se han necesitado de enpoints adicionales en la parte servidora para la gestión y mantenimiento de la aplicación.

Estos enpoints tienen la operativa de generar en el servidor un objeto de tipo documento cuyo contenido se basa en las geoposiciones de los centros insertados en la base de datos. Este documento se utiliza para poder realizar busquedas por aproximaciones basadas en posiciones geo de longitud y latitud.

Estos servicios web se basan en la creación del documento como herramienta de mantenimiento y la publicación de un servicio para su búsqueda:

Clase	MaintenanceTasksServlet
Descripción	Servlet que al ser llamado genera un documento en el servidor con los datos de geoposicionamiento de los centros populados en la base de datos
Paquete	com.example.moventaa.endpoint.placeutil

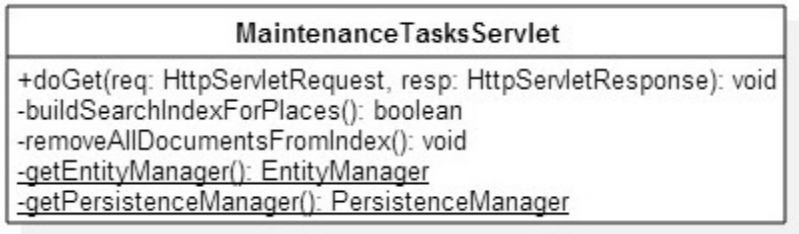


Ilustración 14: Diagrama UML: MaintenanceTasksServlet

Clase	PlacesHelper
Descripción	Endpoint con la utilidad de buscar centros basados en la posición pasada por argumentos. También incluye los métodos para la generación del documento
Paquete	com.example.moventaa.endpoint.placeutil

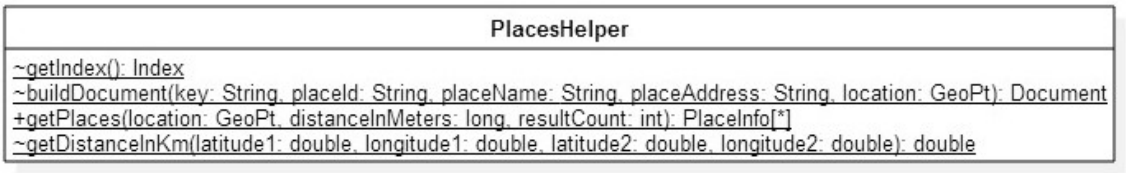


Ilustración 15: Diagrama UML: PlacesHelper

Clase	EmpleadoProductoVentas
Descripción	Obtiene las vetnas realizadas por un empleado en un centro
Paquete	com.example.moventaa.endpoint

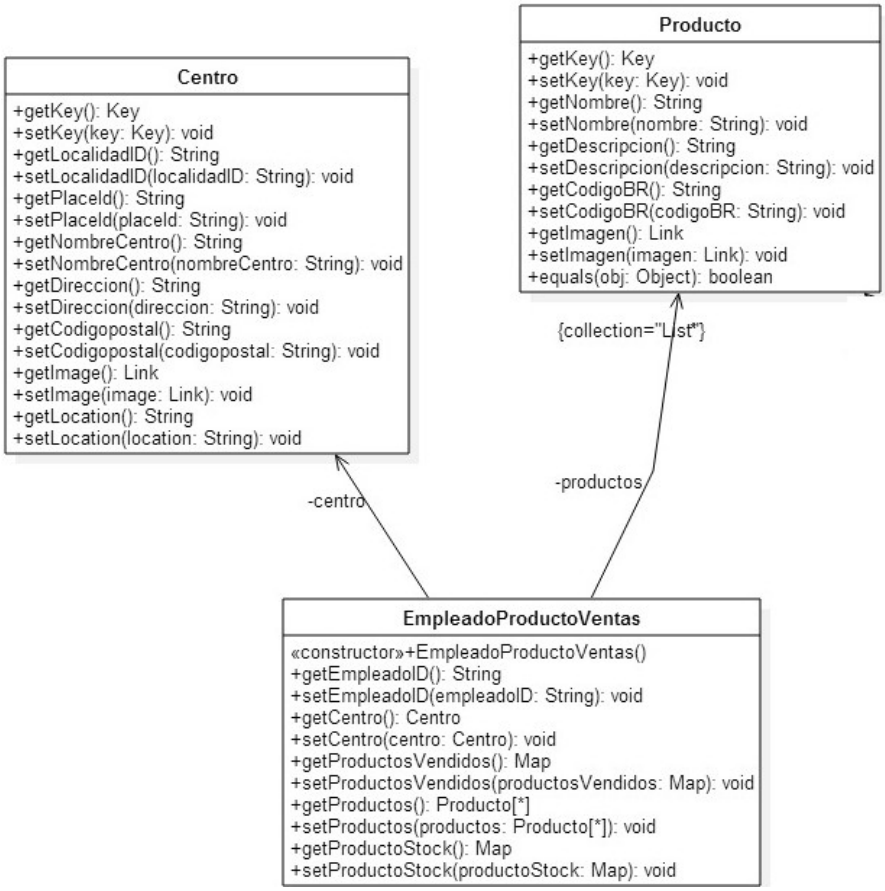


Ilustración 16: DiagramaUML: EmpleadoProductoVentas

6.3.2 Diseño en cliente

En la parte cliente podemos observar diferentes tipos de paquetes. La estructura de la aplicación se ha definido de la siguiente manera:

- com/example/moventa → clases principales de la aplicación
- com/example/moventa/presenter → clases Presenter del modelo MVP
- com/example/moventa/utils → clases de utilidades
- com/example/moventa/bean/*endpoint → endpoints utilizados por la aplicacion
- com/example/moventa/bean/endpoint/model → objetos del modelo de endpoints basados en objetos JSON parser.

También se pueden observar otras paquetes de terceros incluidos dentro la aplicación. Estos paquetes son integraciones de aplicaciones de terceros utilizados dentro de la aplicación

- google/zxing/client/android → biblioteca BarCode 2d/3d
- roomorama.caldroid → biblioteca de implmentación de material Calendar
- antonyt.infiniteviewpager → biblioteca dependiente de caldroid

Diagrama de paquetes de la aplicación Moventaa

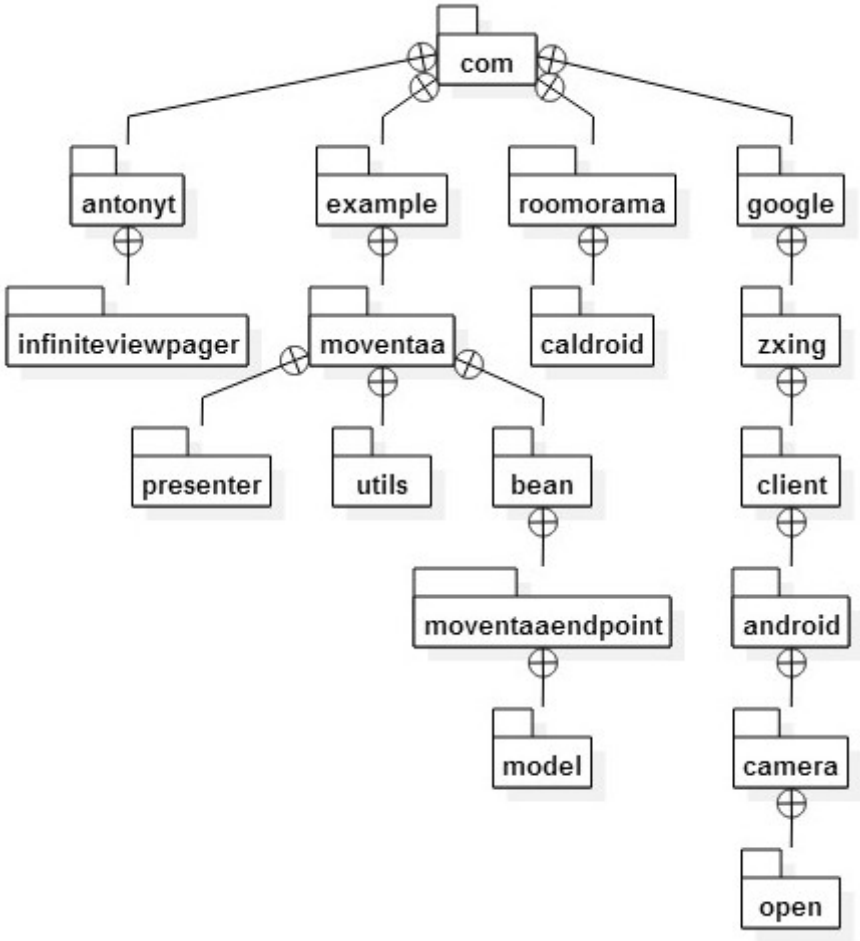


Ilustración 17: Diagrama de paquetes Moventaa

Detallamos a continuación las clases utilizadas:

Clase	MainActivity
Paquete	com.example.moventaa

Clase principal de la aplicación establecida como Launcher y Main dentro del manifiesto de la aplicación. Esta clase se encarga de:

- Comprobar que hay un empleado registrado, en caso contrario lanza la pantalla de registro

- Si hay un empleado registrado, comprueba si tiene activo el GPS en caso afirmativo queda a la espera que reciba una ubicación.

- En caso negativo llama al Intent del listado de Centros seleccionados

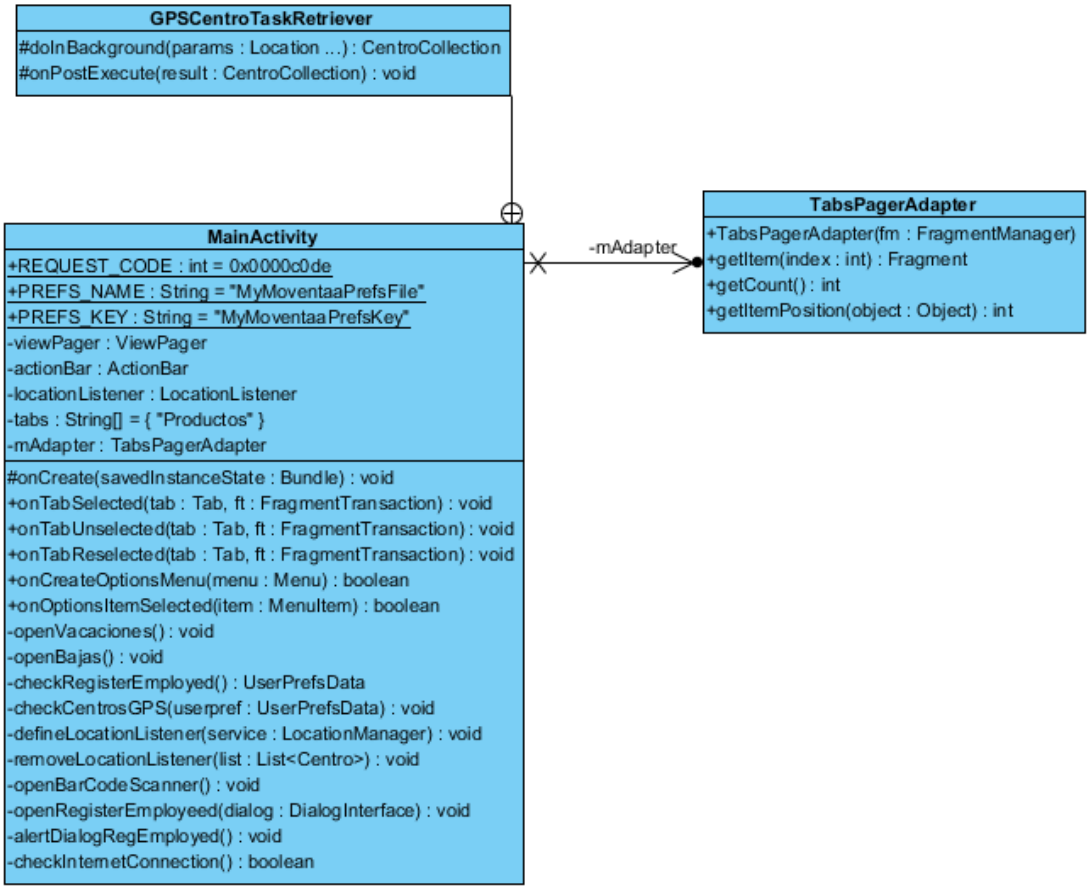


Ilustración 18: Diagrama UML Cliente - MainActivity

Clase	EmployedActivity
Paquete	com.example.moventaa

Actividad encargada de registrar los datos del empleado, los datos son guardados en el servidor y en local con la calse UserPresfData

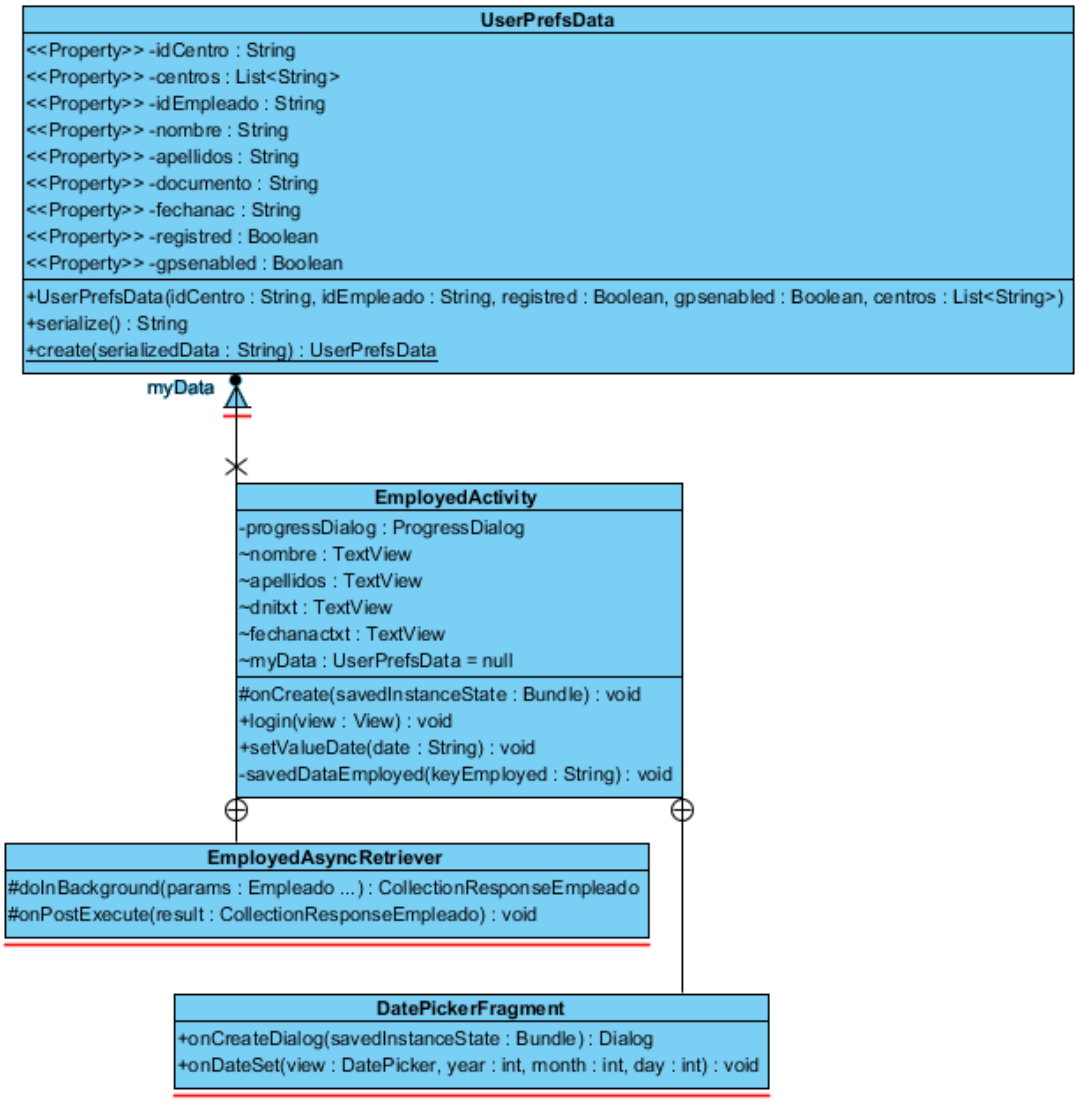


Ilustración 19: Diagrama UML Cliente - EmployedActivity

Clase	ListProductFragment
Paquete	com.example.moventaa

Fragmento que muestra el listado de productos en la pantalla principal

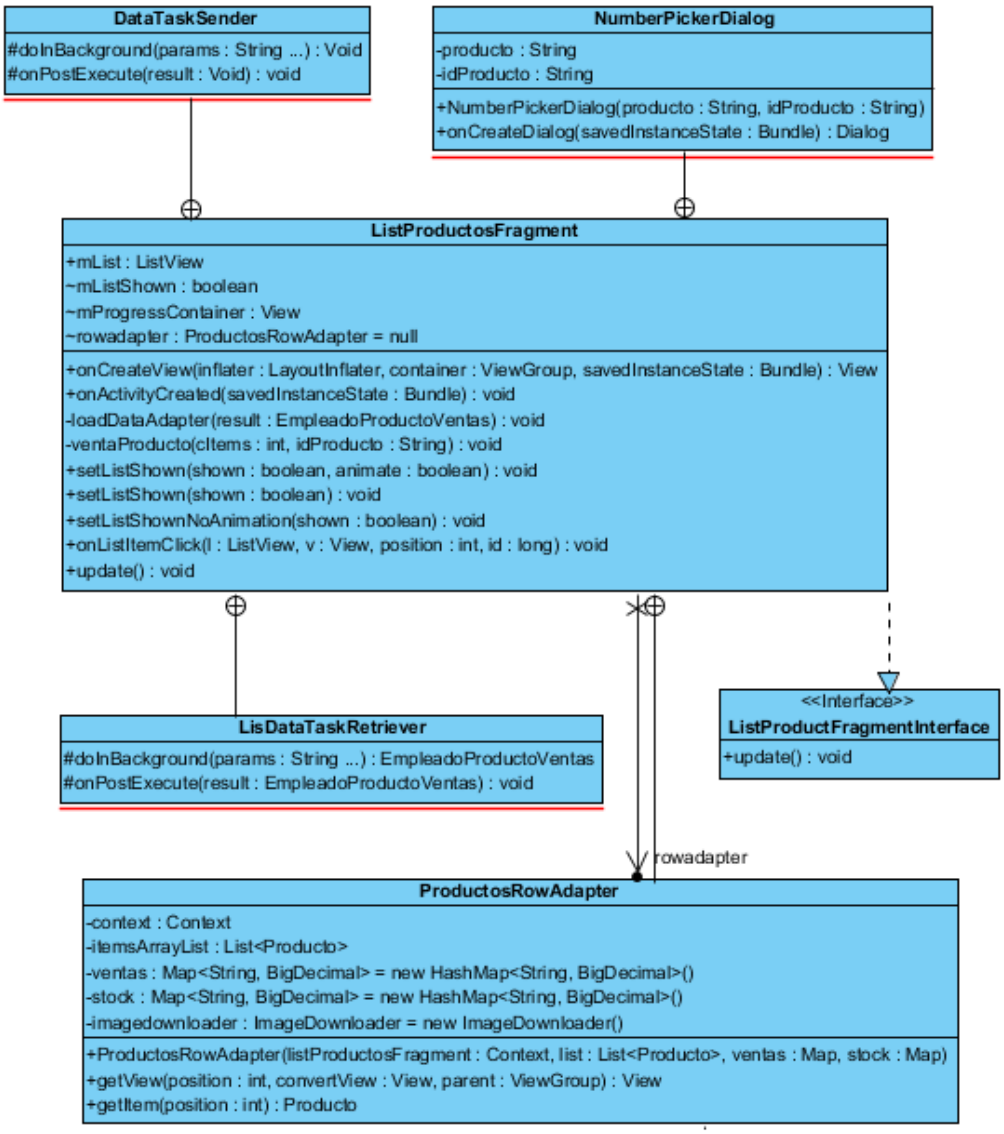


Ilustración 20: Diagrama UML Cliente - ListProductosFragment

Clase	ListNominasActivity
Paquete	com.example.moventaa

Actividad para mostrar el listado de nominas

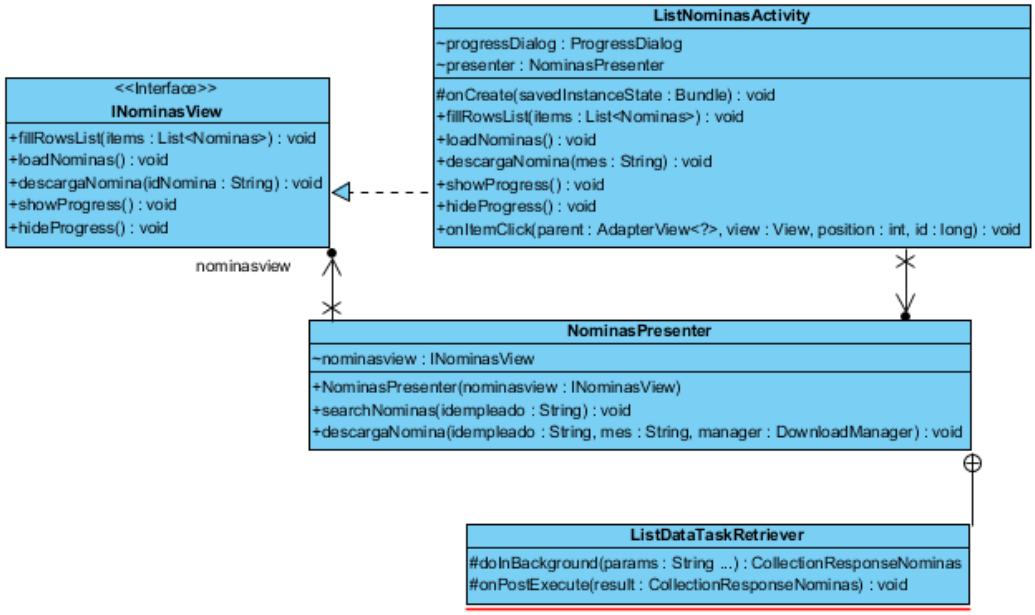


Ilustración 21: Diagrama UML Cliente - ListNominasActivity

Clase	VacacionesActivity
Paquete	com.example.moventaa

Muestra el calendario de Caldroid para que el usuario pueda solicitar las vacaciones

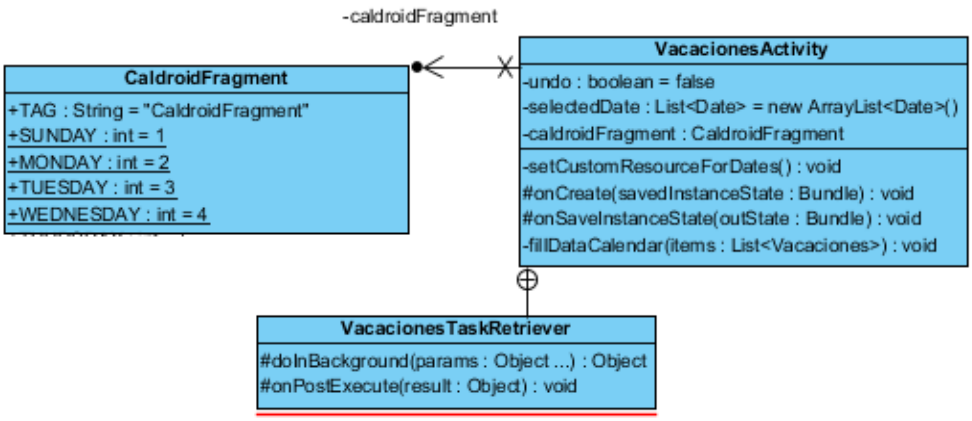


Ilustración 22: Diagrama UML Cliente - VacacionesActivity

Clase	ProductoSearchActivity
Paquete	com.example.moventaa

Implementación de la búsqueda de productos y muestra de los resultados en un ListView

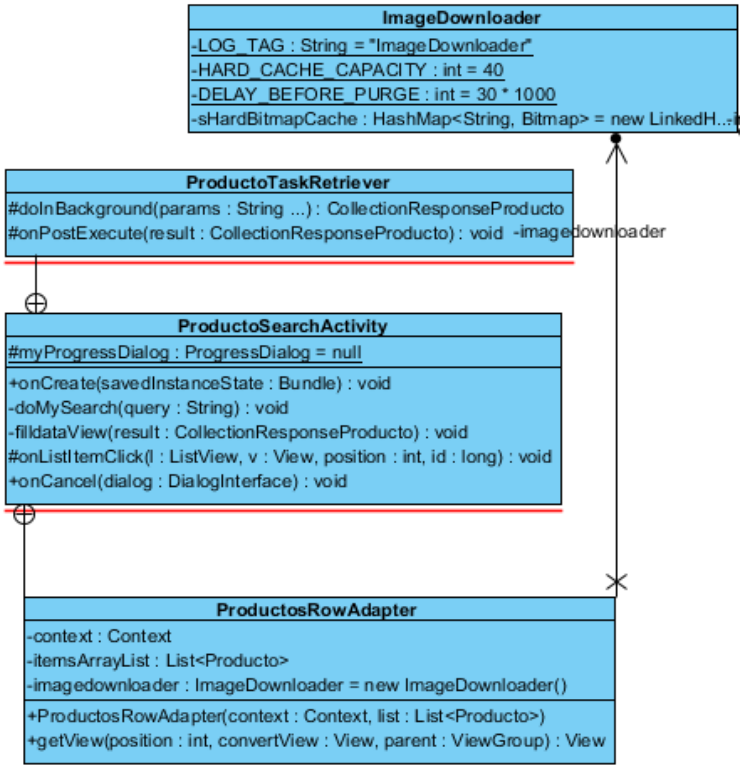


Ilustración 23: Diagrama UML Cliente - ProductoSearchActivity

Clase	GPSActivity
Paquete	com.example.moventaa

Actividad que muestra al usuario la opción de utilizar el GPS como elemento para la búsqueda de centros

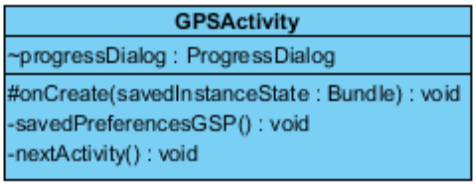


Ilustración 24: Diagrama UML Cliente - GPSActivity

Clase	ListCentrosActivity
Paquete	com.example.moventaa

Actividad que muestra un listado de centros en los que el usuario puede seleccionar para guardar en sus preferencias.

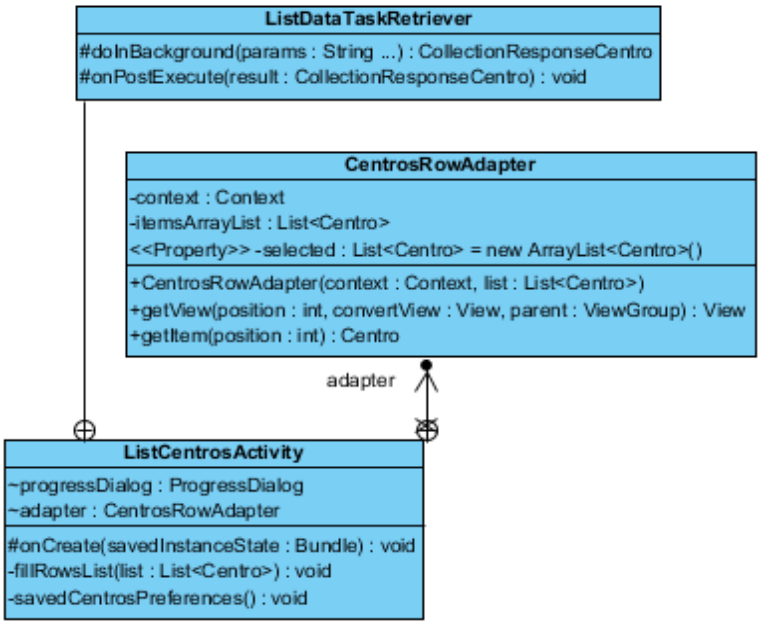


Ilustración 25: Diagrama UML Cliente - ListCentrosActivity

Clase	ProductoInfoActivity
Paquete	com.example.moventaa

Actividad para mostrar los datos de un producto y la posibilidad de solicitar stock

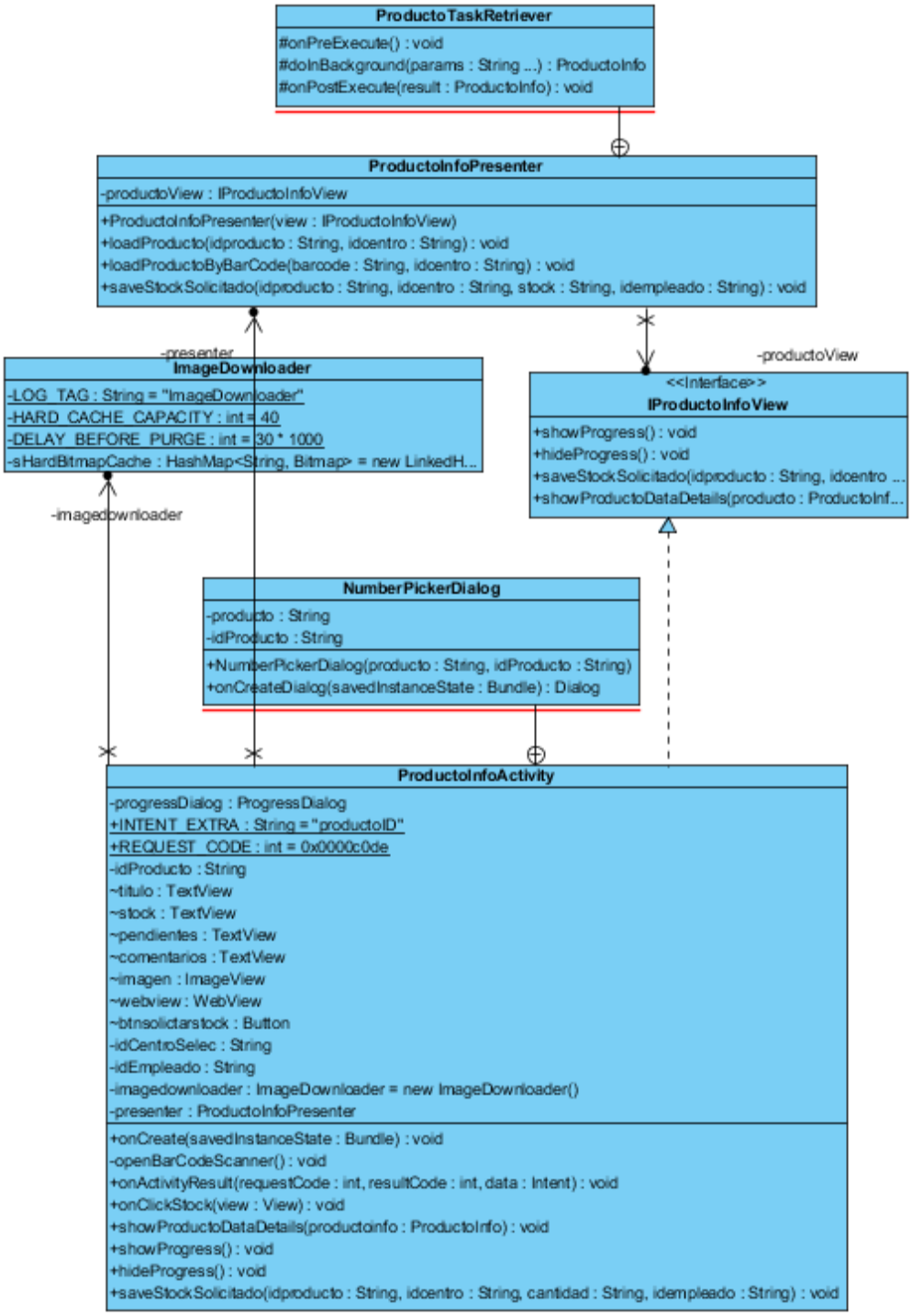


Ilustración 26: Diagrama UML Cliente - ProductoInfoActivity

Clase	BusquedaCentrosActivity
Paquete	com.example.moventaa

Actividad que se dedica a la búsqueda de centros por localidad

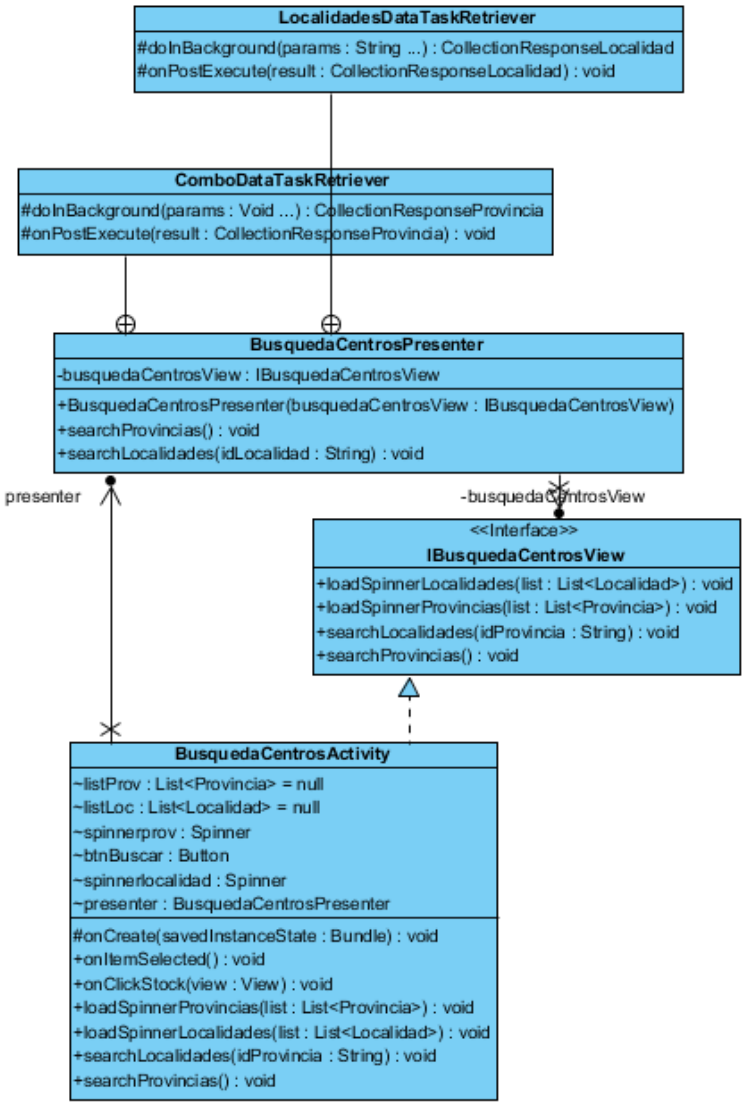
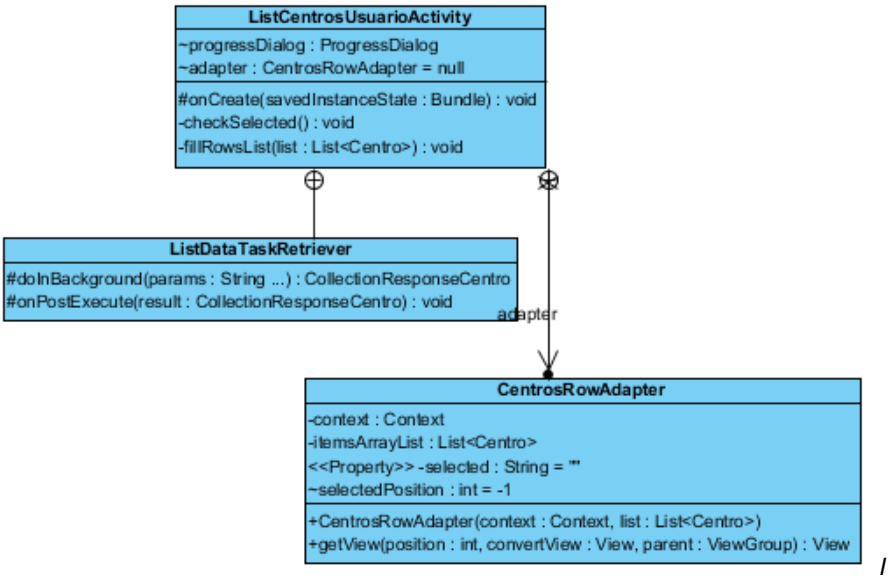


Ilustración 27: Diagrama UML Cliente - BusquedaCentrosActivity

Clase	ListCentrosUsuarioActivity
Paquete	com.example.moventaa

Muestra el listado de centros de las preferencias del usuario para que este elija uno si no tiene activado el GPS



lustración 28: Diagrama UML Cliente - ListCentrosUsuarioActivity

Clase	BajaActivity
Paquete	com.example.moventaa

Actividad para la solicitud de bajas del empleado

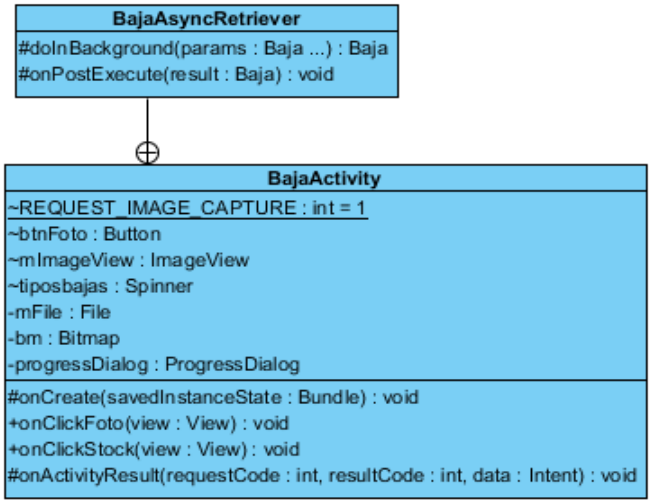


Ilustración 29: lustración 24: Diagrama UML Cliente - BajaActivity

Clase	ImageDownloader
Paquete	com.example.moventaa.util

Utilidad para la gestión de la descarga de imágenes y utilización de cache.

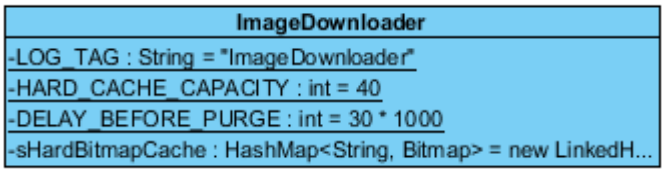


Ilustración 30: Diagrama UML Cliente - ImageDownloader

Clase	Utils
Paquete	com.example.moventaa.util

Clase para la encapsulación de las utilidades usadas en la aplicación

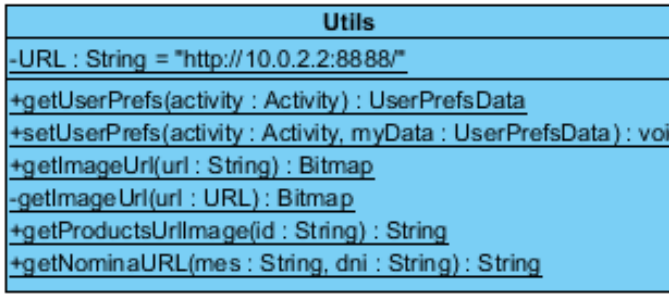


Ilustración 31: Diagrama UML Cliente - Utils

Clase	UserPrefsData
Paquete	com.example.moventaa.util

Implementación del guardado de las preferencias del usuario en formato serializado.

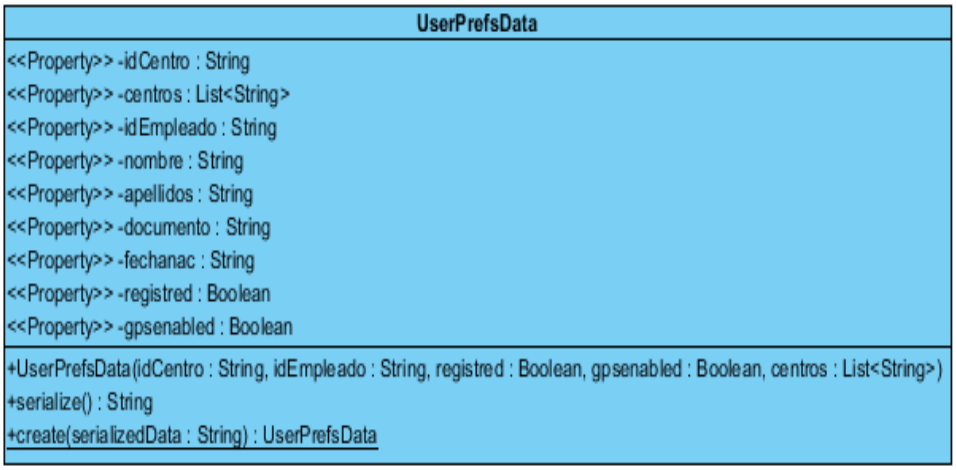


Ilustración 32: Diagrama UML Cliente - UserPrefsData

Clase	CloudEndpointUtils
Paquete	com.example.moventaa.

Utilidad generada por GAE para la ubicación de los métodos comunes a las llamadas de los servicios web.

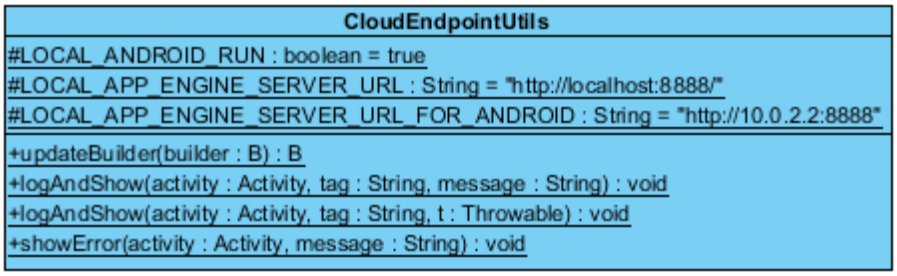


Ilustración 33: Diagrama UML Cliente - CloundEndpointUtils

Capítulo 7: Desarrollo

7.1 Descripción de las herramientas utilizadas

7.1.1. Entorno de desarrollo Eclipse-IDE

En el desarrollo de la aplicación se ha utilizado como IDE Eclipse Luna, al que se le ha añadido los siguientes plugins:

- SDK de Android para la generación de un proyecto de tipo Android.
- https://developer.android.com/sdk/installing/index.html?pkg=tools
- SDK de App Engine de Google para la generación de la parte web.

https://storage.googleapis.com/appengine-sdks/featured/appengine-java-sdk-1.9.15.zip

- Plugin de eclipse de App Engine

https://cloud.google.com/eclipse/docs/install-eclipse-4.4

7.1.2. Bibliotecas de terceros para Android

Para la parte cliente se han utilizado bibliotecas de terceros que detallamos a continuación;

ButterKnife

Biblioteca que permite la inyección de dependencias de objetos de la vista en el View utilizando anotaciones de Java.

Ofrece la ventaja de no tener que estar declarando constantemente el acceso a los objetos de la View declarados en el layout.xml y ademas facilita la implementación de los eventos de aquellos objetos que queremos implemnetar. Por ejemplo, con ButterKnife es tan sencillo implementar un onClick sobre un botón con el siguiente código:

```
OnClick(R.id.submit)
public void submit() {
 // TODO submit data to server...

}
```

Esta biblioteca ha sido desarrollada por Jake Wharton bajo licencia Apache 2.0

```
Copyright 2013 Jake Wharton
Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.
```

Home page: http://jakewharton.github.io/butterknife/

Licencia: Apache 2.0

Logo de la biblioteca:

Zxing 1D/2D

Es un biblioteca open source que habilita un escáner y procesador 1d/2d de imágenes de códigos de barras.

La biblioteca se compone de tres modulos:

- core

- android

- android-client

- android-integration

Los modulos core y android han sido incluidos como bibliotecas dentro del proyecto, el módulo android-client ha sido integrado como código fuente dentro del proyecto utilizando la implementación ya existente del escaneo del código de barras como parte del proyecto.

Home page: https://github.com/zxing/zxing

Licencia: Apache 2.0

Logo de la biblioteca:

Caldroid

Es un interfaz de usuario que visualiza calendarios como si fuera un almanaque en el que se muestra los días de un mes. Ademas de tener un intefaz amigable, permite la selección de días del calendario, eventos y multiples tipos de configuraciones

Home page: https://github.com/roomorama/Caldroid/

Licencia: MIT License

Date4J

Es un implementación que abstrae del uso de la biblioteca Date y Calendar del JDK de Java.

Su uso es opensource y viene incluido con las biblioteca de Caldroid

Home page: http://www.date4j.net/

Licencia:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Hirondelle Systems nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

Google App Engine

Biblioteca para el desarrollo de aplicaciones backend en la que facilita al desarrollador las herramientas necesarias para su creación, publicación y puesta en producción.

Home page: https://cloud.google.com/mobile/app_engine/?hl=es

Licencia: Niveles de acuerdo de Google https://cloud.google.com/appengine/sla

Logo de la biblioteca:

Google Cloud Endpoints

Biblioteca que permite definir la logica de negocio asociada al App Engine y a la que puede ser accedida a traves de peticiones RESTful o RPC Api's siendo multiplataforma.

Home page: https://cloud.google.com/mobile/endpoints/?hl=es

Licencia: Niveles de acuerdo de Google https://cloud.google.com/appengine/sla

Logo de la biblioteca:

Consideraciones adicionales:

Hay que tener en cuenta algunas de las consideraciones que se han tomado a la hora de utilizar las bibliotecas en el proyecto, sobre todo en el caso de Zxing 1D/2D. En este caso, se ha decido incluir las fuentes de la biblioteca dentro del propio proyecto, esta decisión se baja en los siguientes puntos:

- Al incluir los fuentes evitamos que el usuario tenga que descargarse una app adicional de escaneo de códigos de barras.
- Controlamos el flujo funcional de la aplicación.
- Al integrar el escáner dentro del aplicativo, esta tiene un aspecto más profesional.
- Evitamos que el usuario tenga la sensación de acceder a otra aplicación que se encuentra fuera del uso de Moventaa.

Por el contrario, esta decisión implica que:

- La distribución que realicemos de la apk contendrá fijada una versión lanzada de Zxing, por lo que si hay bugs tendremos que volver a realizar una nueva versión.
- No dispondremos compatibilidad para nuevas versiones de android.
- No dispondremos compatibilidad para cualqueir dispositivo hardware.
- Habrá una mayor carga de trabajo a la hora de realizar nuevas versiones de la aplicación.

7.2 Análisis del estado del proyecto

Durante el desarrollo de la aplicación el proyecto ha sufrido varios cambios, tanto a nivel de diseño de los interfaces de usuario como de la arquitectura utilizada.

7.2.1 Estado del proyecto

Funcionalidades no implementadas

De todas la funcionalidades indicadas en fases anteriores, y por motivos de tiempo de desarrollo, no se incluirán en esta entrega las siguientes funcionalidades:

- Informe semanal de ventas realizadas y comisiones conseguidas

- Informe mensual de ventas y comisiones

Despliegue en Google Cloud

La parte del servidor no se ha subido al Cloud Engine de Google, entre otros motivos, debido a que se ha cumplido el plazo del periodo de prueba del Cloud, pasando por tanto al modo de pago, por lo que se ha tomado la decisión de eliminar el despliegue en la nube.

Versión Pre-Alpha

La aplicación se entrega en versión Pre-Alpha, esto significa que la aplicación requiere de un conjunto de pruebas más extenso, revisión del código y pruebas de estrés antes de pasar a una versión lista para poder ser utilizada como Beta dentro de un dispositivo móvil.

Tampoco ha sido posible realizar unas pruebas exhaustivas de la aplicación con un dispositivo hardware, quedando como tareas pendiente antes de su publicación.

Modificaciones del diseño frente al mockup inicial.

A continuación indicamos las modificaciones del diseño del interfaz del usuario que se han realizado frente a la presentación inicial realizada de alto nivel o fidelidad

- Se ha eliminado el “splash window” de inicio de pantalla. Esta decisión se debe a que las buenas prácticas en el diseño de interfaces de android indican que no es aconsejable utilizar “splash windows”, es preferible utilizar una pantalla donde se muestre un barra de progreso

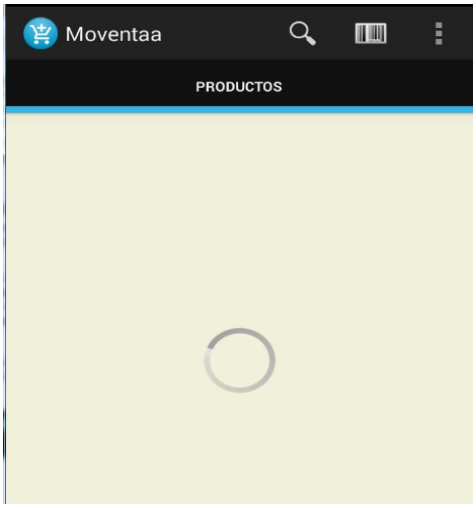


Ilustración 34: Modificaciones diseño: Barra progreso

- Se ha eliminado la toolbar y se han incluido las opciones como menú dentro del nuevo diseño de aplicaciones propuesto por Android.

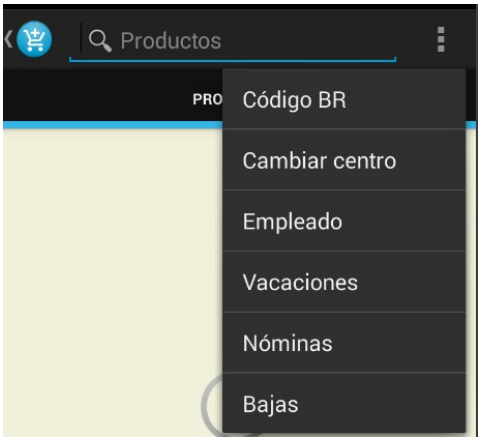


Ilustración 35: Modificaciones diseño: Menu

- El buscador se ha incluido como elemento dentro del cabecera de la aplicación

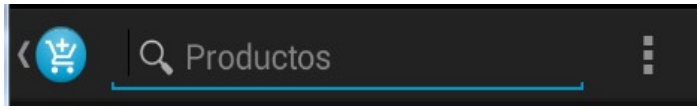


Ilustración 36: Modificaciones diseño: Buscador

7.2.2 Plan a futuro

Las previsión a futuro de las tareas pendientes por realizar son:

- Codificación de los funcionalidades de informes
- Revisión del código fuente en busca de leaks y bugs de rendimiento
- Mejora del feedback hacia el usuario de los eventos que se produzcan a nivel de background de la aplicacación.
- Migración del proyecto a a ADT-Studio y Gradle
- Diseño de pruebas unitarias
- Subir la parte backend o servidor al Cloud Engine de Google.

Capítulo 8: Pruebas

8.1 Definición del plan de pruebas

La aplicación debe probarse en un entorno simulado. La utilización de un entorno simulado se debe a que las pruebas de GPS solo se pueden realizar con este entorno, ya que de otra manera seria muy difícil desplazarse a una ubicación para comprobar su correcto funcionamiento.

Los requisitos necesarios son:

Android	ADT v.19 – Versión 4.4.2
Eclipse IDE	Luna + plugin ADT + plugin AppEngine + plugin Cloud
Python	2.7
SDK Android	Android Manager + Android SDK
Java	1.7 SDK

Descarga del plugin para Eclipse:

Eclipse version	Installation instructions	Direct plugin link
Eclipse 4.4 (Luna)	Plugin for Eclipse 4.4 (Luna)	https://dl.google.com/eclipse/plugin/4.4

Como emulador podemos utilizar:

- El propio emulador de Android que incluye el SDK
- Genymotion, que es un emulador preconfigurado y listo para ser usado en entornos y que presenta la ventaja de ser muy rápido. http://www.genymotion.com/

La configuración del emulador debe tener las siguientes características:

Plataforma	4.4.2
API Level	19
Device	Nexus 4 (4.7”)
Target	Google APIs Level 19
Back Camera	Emulated
SD Card	64 MB

Emulation options	Use Host GPU

8.2 Definición del plan de pruebas

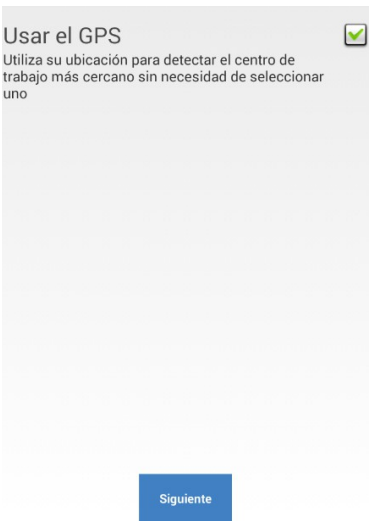
El plan de pruebas se pasa en la relación de un test funcional de la aplicación verificando los resultados obtenidos con el objetivo del plan.

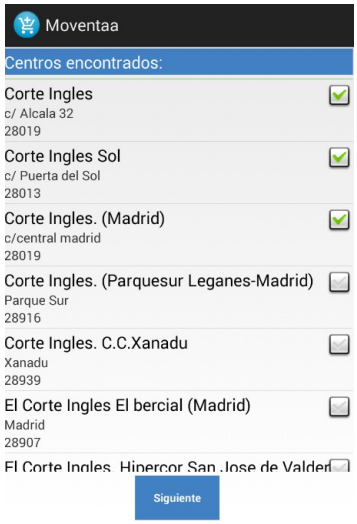
Cada plan de prueba incluye, identificación de la prueba, descripción , acción y resultado esperado, así como el estado final de la prueba siendo los valores OK para una prueba satisfactoria y NO-OK en caso de algú tipo de error,

8.2.1 Test funcional de la aplicación – Casos de prueba

ID	***Test-0001 - Registro del empleados***
Estado	OK
Descripción	Registro del empleados
Acción	Al iniciar la aplicación la primera vez nos debe mostrar un mensaje indicando que no hay empleado registrado, siendo necesario introducir los datos de un empleado, uso de GPS, búsqueda de centros por localidad y provincia, selección de centro.
Datos:	Nombre: Empleado01 Apellidos: Apellido01 Segundo01 DNI: 99999999v Fecha Nac: 20/01/1968 Provincia: Madrid Localidad: Madrid

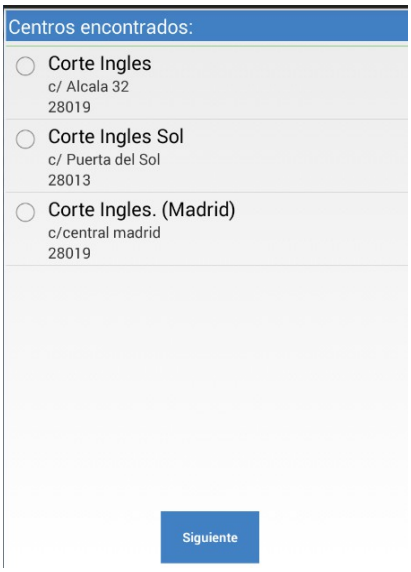
Pantallas del plan de pruebas





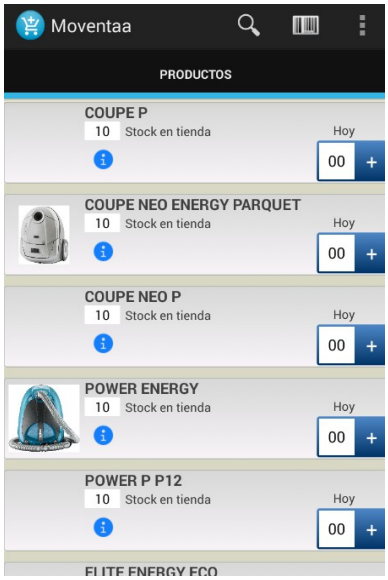
ID	***Test-0002 - Selección de centro asociado***
Estado	OK
Descripción	Selección de centro asociado
Acción	Una vez registrado el empleado, si se han seleccionado más de un centro y no está el GPS activo, debe mostrar una pantalla de selección de centro.
Datos:	Selección “Corte Ingles c/Alcala 32 28019”

Pantallas del plan de pruebas



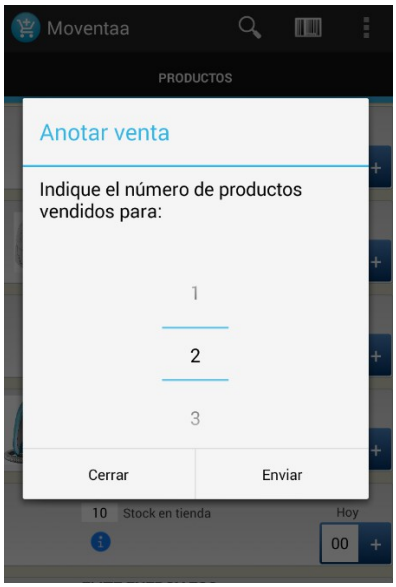
ID	***Test-0003 - Muestra de productos para el centro seleccionado***
Estado	OK
Descripción	Muestra de productos para el centro seleccionado
Acción	Seleccionado el centro, debe mostrar el listado de productos asociados para el centro.
Datos:	Selección “Corte Ingles c/Alcala 32 28019”

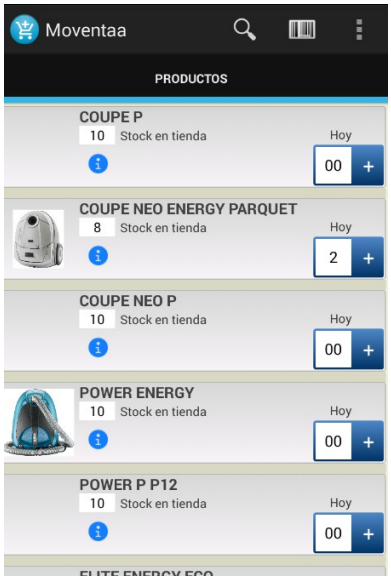
Pantallas del plan de pruebas



ID	***Test-0004 - Alta de una venta***
Estado	OK
Descripción	Alta de una venta
Acción	Seleccionado en la pantalla del listado de productos, en el icono (+) e incluyendo un valor mayor que 0, debe actualizarse el contador con el valor indicado y el stock reducirse en el mismo valor
Datos:	Producto: Coupe Neo Energy Parquet. Unidades vendidas: 2

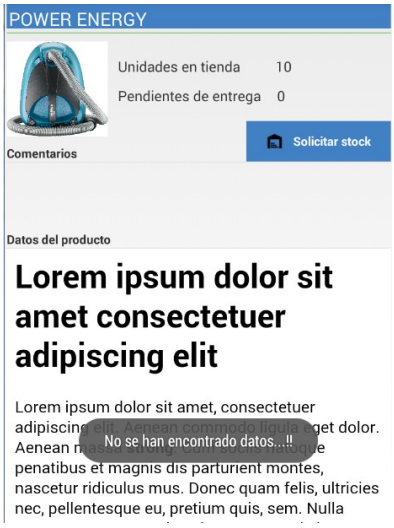
Pantallas del plan de pruebas





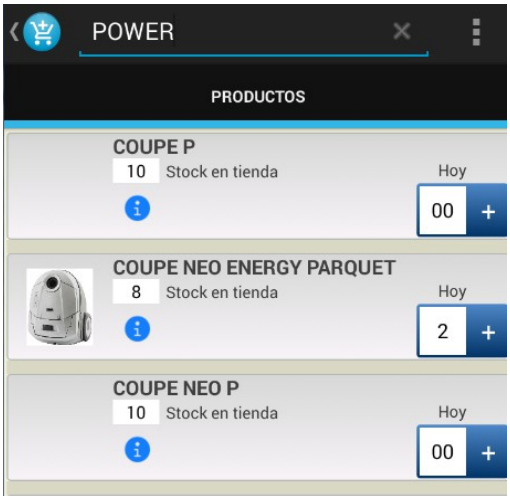
ID	***Test-0005 - Detalle de un producto***
Estado	OK
Descripción	Detalle de un producto
Acción	Seleccionado un elemento del listado de productos, debe poder verse el detalle de este
Datos:	Producto: POWER Energy

ID	***Test-0006 - Solicitar stock***
Estado	OK
Descripción	Solicitar stock
Acción	En el detalle de un producto, al pulsar en el botón solicitar stock debe mostra un dialogo donde introducir el stock y ser enviado al servidor.
Datos:	Producto: POWER Energy

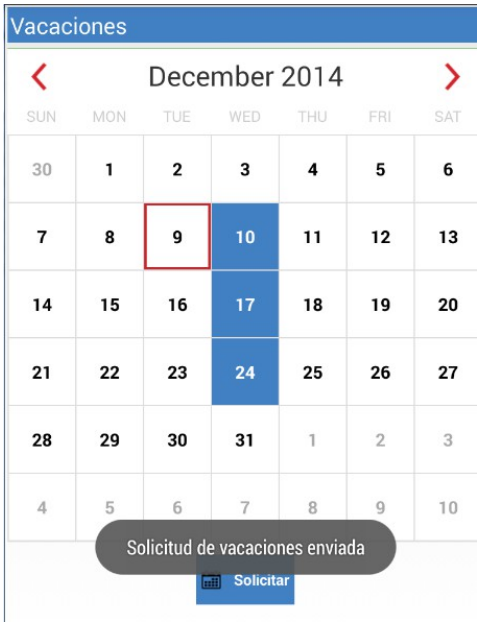


Observaciones: El test no cumple con lo objetivos detallados, aunque si volvemos a acceder al mismo producto podemos comprobar que el contador de pendientes de entrega si se ha actualizado en segunda instancia.

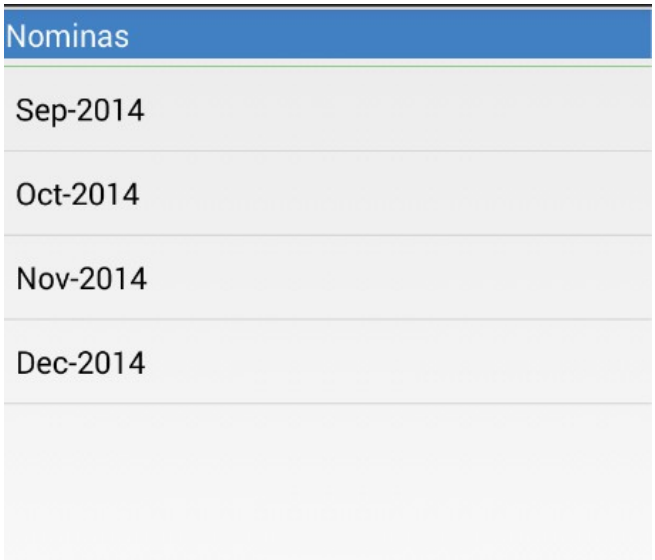
ID	***Test-0007 - Buscador***
Estado	OK
Descripción	Buscador
Acción	Buscamos por texto coincidente del nombre de un producto, mostrando un listado.
Datos:	Búsqueda: POWER



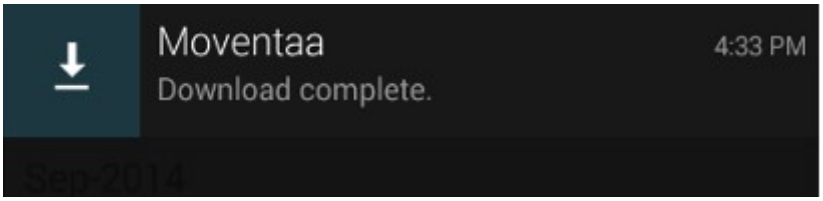
ID	***Test-0008 - Solicitud de vacaciones***
Estado	OK
Descripción	Solicitud de vacaciones
Acción	Accediendo desde el menú a las vacaciones , se pueden seleccionar días y enviarlos al servidor.
Datos:	10,17,24 de diciembre



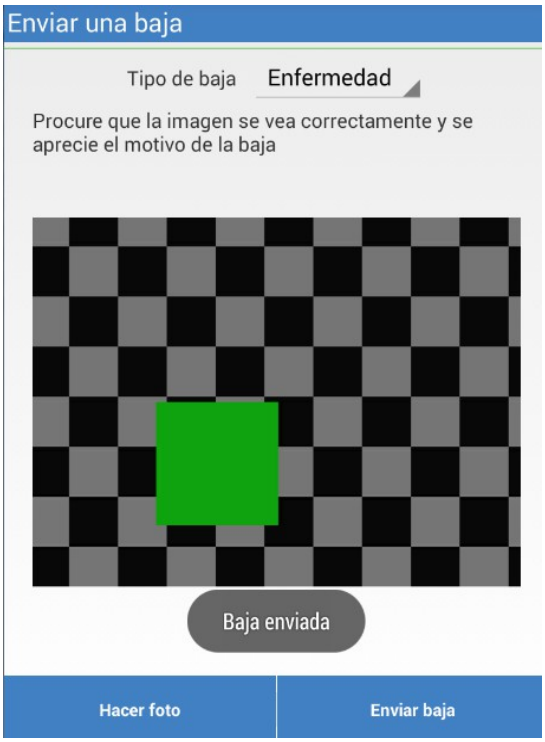
ID	***Test-0009 - Nóminas***
Estado	OK
Descripción	Nóminas
Acción	Accediendo al menú, se puede ver el listado de nominas del empleado
Datos:	



ID	***Test-0010 - Descarga de la Nómina***
Estado	OK
Descripción	Descarga de la Nómina
Acción	Seleccionando sobre un elemento del listado de nominas, se produce la descarga del fichero.
Datos:	Nov-2014

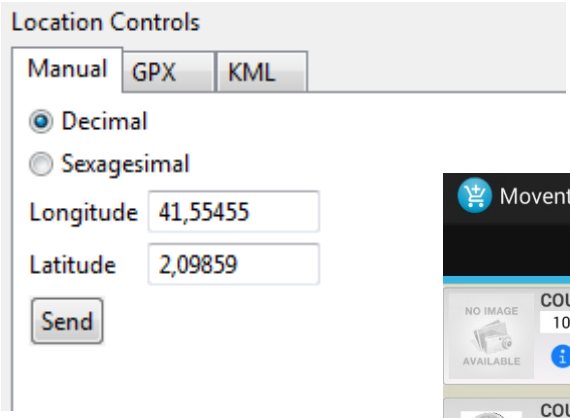


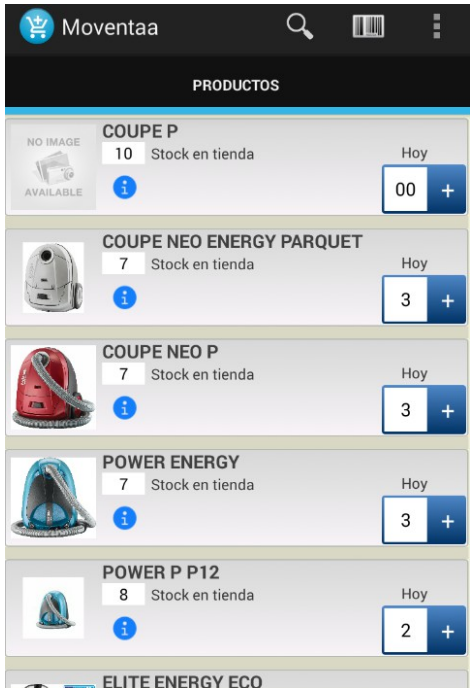
ID	***Test-0011 - Envío de baja***
Estado	OK
Descripción	Envío de baja
Acción	Seleccionando en el menú Baja, podemos enviar una baja, haciendo una foto y seleccionando el tipo de baja
Datos:	Baja: Enfermedad. Foto: Emulador



ID	***Test-0012 – Prueba GPS***
Estado	OK
Descripción	Prueba GPS
Acción	Enviamos al emulador un identificador de GPS y el sistema debe cargar los productos asociados al centro detectado del GPS
Datos:	Longitud 2.09859, Latitud 41.55455

Pre-requisitos, tener configurado el empleado con el valor de GPS seleccionado en los datos.





Capítulo 9: Conclusiones

La realización de este proyecto me ha permitido ampliar mis conocimientos en el desarrollo de aplicaciones para dispositivos móviles, he aprendido a solucionar problemas y dificultades que me han ido surgido durante el desarrollo del mismo y que gracias a los aspectos y conocimientos estudiados durante estos años que me han facilitado el trabajo de aprendizaje.

También he aprendido la importancia de realizar una aplicación acorde a las necesidades, requisitos y aspectos fundamentales de la aplicación y poner los objetivos en un listón que sea capaz de alcanzar.

A nivel de alcance de los objetivos iniciales no ha sido completado, en parte debido a que dispuse el listón de funcionalidades demasiado lejos del alcance sin tener en cuenta el tiempo total disponible para la realización del proyecto. A un nivel profesional, pienso que si que se habría alcanzado todos los objetivos en tiempo y forma.

Por otro lado, a nivel de aplicación de metodología, aunque es la más acertada para un proyecto de la UOC, creo que hay otros tipos que son más aconsejables para este tipo de proyecto, por ejemplo, las metodologías ágiles tales como: scrum.

Aunque la planificación y la metodología estaban bien aplicadas, el papel lo aguanta todo, al final el resultado no ha sido el esperado, en parte porque la planificación era demasiado optimista y poco realista, y aunque se ha tenido que entregar una aplicación con menos funcionalidades esto no ha supuesto un a menor calidad en el producto fina. En cualquier caso ha servido como lección aprendida.

En donde más problemas me he encontrado ha sido en la utilización del App Engine de Google. Aunque lo pintan muy bonito, fácil para el desarrollador, etc, la realidad es bastante distinta, en parte creo que fue una apuesta muy arriesgada el utilizar un marco de desarrollo del que no tenía experiencia y que al irse acumulando con falta conocimientos de otras áreas del desarrollo de aplicaciones móviles me han ocasionado más desventajas que ventajas. En parte se debe a la utilización de JPA y JTA como capas de persistencia del AppEngine, que han limitado bastante el manejo de las consultas en la parte servidora, ya que se encuentra en fases muy tempranas de desarrollo de la API y que la incorporación de DataNucleus como Engine de JPA ocasionó que el proyecto se volviera inestable sin explicación alguna.
Supongo que con más tiempo termina por aprender los pros y contras de las API Engine de Google y determinar para futuros proyectos si es viable su aplicación o no.

En mi opinión creo que es buen método como preparación profesional para enfrentarse a casos reales del mundo laboral.

El tiempo invertido, esfuerzo y trabajo es proporcional y necesario para sentirme satisfecho con el trabajo realizado presentado como proyecto final de carrera.

Como mejoras futuras, La aplicación tiene potencial para desarrollar nuevas líneas. Esta pensada para ser utilizada por las empresas dentro de sus ERP (Enterprise Resource Planning) obteniendo información de tiempo real y de primera mano de sus empleados, facilitando la relación e información bidireccional entre el empleado y la empresa.

Estas líneas pueden estar orientadas en la realización de componentes adicionales de API's de conexión para los ERP tanto profesionales como opensource que existen en el mercado, así como la posibilidad de introducir un sistema de trazabilidad de los productos vendidos. Otra línea de desarrollo podría orientarse a la incorporación de las opiniones y quejas de los clientes que han comprobado el producto y deciden devolverlo por algún motivo y que por norma general, es una información que no suele llegar a los departamentos de calidad de una empresa. La introducción de un sistema de computo de horas trabajas y jornadas laborales en base a la conectividad del dispositivo móvil, tal y como si fuera un sistema de fichaje.

Capítulo 10: Glosario

Términos, abreviaturas y acrónimos utilizados en el documento para la compresión correcta:

- Android, es un sistema operativo que se basa en el núcleo de Linux y diseñado específicamente para dispositivos móviles inteligentes o "smartphones". Aunque si cometido principal fueron los móviles, el auge impulsado ha convertido el sistema operativo en un referente de ser utilizado por cualquier dispositivo que permita una interacción con el usuario.

- Google App Engine (PaaS) es un servicio de alojamiento web que permite ejecutar aplicaciones como servicio (PaaS, del inglés "Plaftform -as-a-Service") sobre la infraestructura de Google.

- Informática en nube ("Cloud Computing"), medios tecnológicos ubicados en la red de internet y que pueden ser utilizados para la computación.

- SDK, del inglés "software development kit" que significa kit de desarrollo de software. Es un conjunto de herramientas de desarrollo de software que permite al programador crear aplicaciones para un sistema en concreto bajo un lenguaje de programación especifico. También se engloba dentro del SDK, notas, recomendaciones y buenas prácticas del desarrollo.

- "Smartphones", dispositivo móvil de última generación.

- Wireframe: técnica de diseño de prototipos de pantallas sencillos basado en líneas que representan en el diseño la ubicación de los componente visuales.

- Stock: cantidad o volumen de productos almacenados y de los se disponen en cierta medida para su venta.
- Usabilidad: criterio por el que se establece si un elemento cumple con las normas de facilidad de uso.
- IDE, entorno de programación
- Eclipse, entorno de desarrollo integrado de código abierto.
- "Framework", marco de trabajo o aplicación que engloba un conjunto de bibliotecas orientadas a la reutilización de los componentes de software.
- Persistencia, capa de programación que maneja el guardado de los datos en la base de datos aplicando estándares ATOM.
- JSON, Javascript Object Notation, notación de objetos utilizada para el intercambio de información entre el cliente y servidor bajo la capa de comunicación y transporte http.

Capítulo 11: Bibliografía

Publicaciones

- **O'Reilly Media** (2012). Head First Android Development.
- **O'Reilly** Neil, Theresa(2012). “Mobile Design Pattern Gallery”
- **O'Reilly** Hoober, Steven; Berkman, Erik(2012). “Designing Mobile interfaces”

Fuentes de internet.

- MVP for Android: how to organize the presentation layer
  [Fecha: 15/Abril/2014] [Autor: Antonio Leiva]
  [Enlace: http://antonioleiva.com/mvp-android/]

- Android Arsenal
  Fuentes de código para Android
  [Enlace: https://android-arsenal.com/]
- Google App Engine
  [Enlace: https://cloud.google.com/appengine/docs]
- AppEngine GitHub
  [Enlace:https://github.com/GoogleCloudPlatform/solutions-mobile-shopping-assistant-backend-java/]
- DataStore Queries
  [Enlace:
  https://cloud.google.com/appengine/docs/java/datastore/queries]

Recomendaciones por parte del consultor:

- Módulo didáctico “Introducción al Trabajo Final”
- Módulo didáctico “Redacción de textos científico-técnicos”
- Módulos didácticos de la asignatura “Tecnología y Desarrollo en Dispositivos Móviles”
- Wiki de la asignatura “Desarrollo de aplicaciones para dispositivos Android”
- Módulos didácticos de la asignatura “Desarrollo de aplicaciones para dispositivos iOS y entornos multiplataforma”

Asistencia a eventos:

CODEMOTION 2014, celebrado en Madrid los días 21 y 22 de Noviembre en el que se trataron algunos temas relacionados con Android.

ANEXO

Anexo 1: Instalación

Definimos a continuación el plan de instalación de la aplicación, el siguiente proceso consta la incorporación de los proyectos dentro de un entorno eclipse IDE, para facilitar el trabajo, hemos definido una alternativa de instalación para el aplicativo Android, pero la parte servidora debe seguir desplegándose con un entorno eclipse.

1.1 Configuración del proyecto

Descomprimir el zip entregado en un directorio.

Dentro del zip encontraremos tres directorios y en la raiz un documento pdf, que es este propio documento.

La aplicación se entrega formando tres módulos o proyectos:

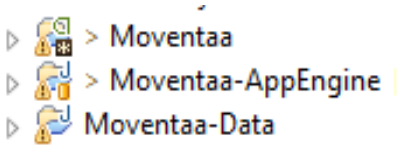


Ilustración 37: Instalación - Carpetas

Moventaa: es el aplicativo cliente movil android

Moventaa-AppEngine: es el backend que incluye los endpoints y war de despliegue sobre un servidor J2EE

Moventaa-Data: aplicación en Python para la población de los datos en el servidor.

Las tres aplicaciones se importarán como proyectos dentro de Eclipse, para realizar esta operación, sobre el Eclipse IDE, pulsaremos:

Botón derecho sobre el workspace > Import

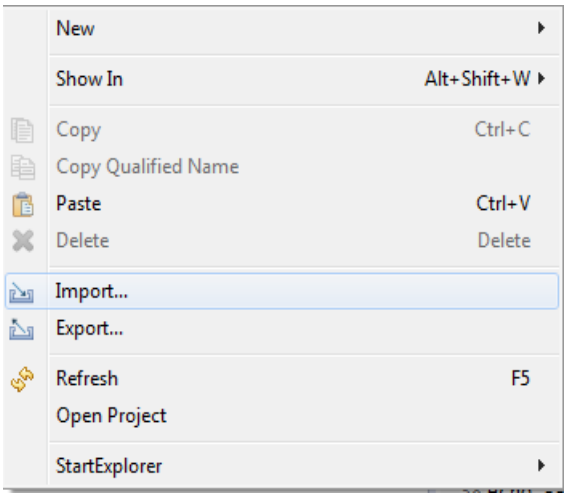


Ilustración 38: Instalación - Importación

Seleccionamos en la siguiente pantalla: Existing Projects into Workspace

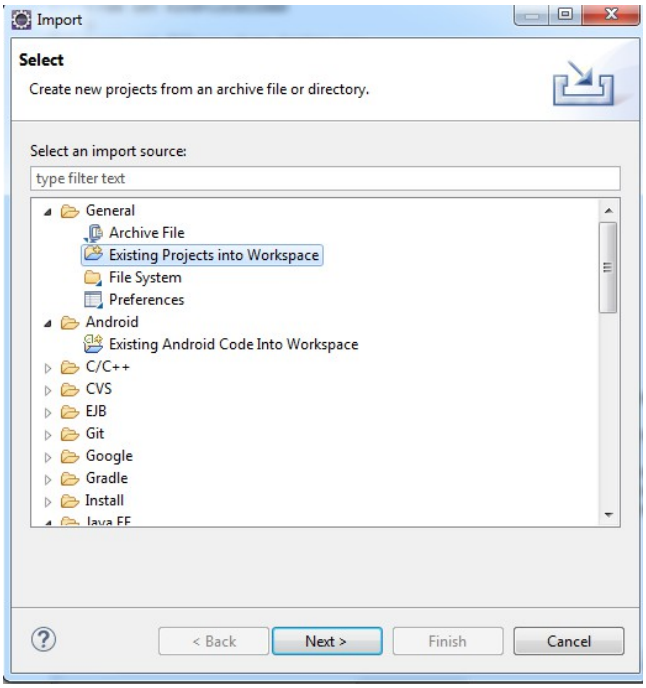


Ilustración 39: Instalación - Define Workspace

Buscamos la ubicación del fichero zip descomprimido y seleccionamos los dos proyectos que no son de android. Debe mostrarse la siguiente pantalla:

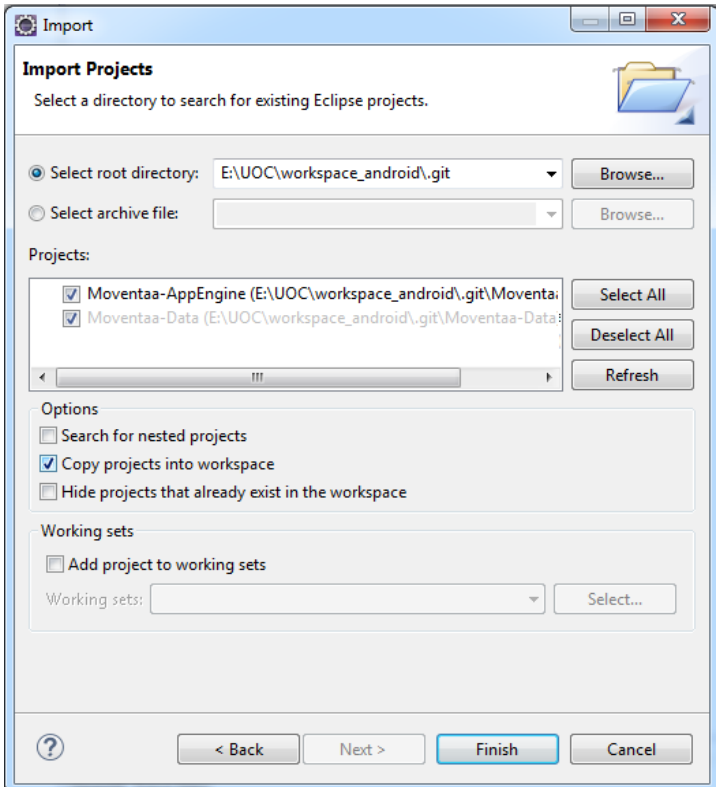


Ilustración 40: Instalación - Ubicación proyectos

Seleccionamos los dos proyectos y pulsamos en Finish.

Ahora debemos importar un proyecto de tipo Android, repetimos los pasos anteriores pero esta vez seleccionando:

Import > Android > Existing Android Code Into Workspace

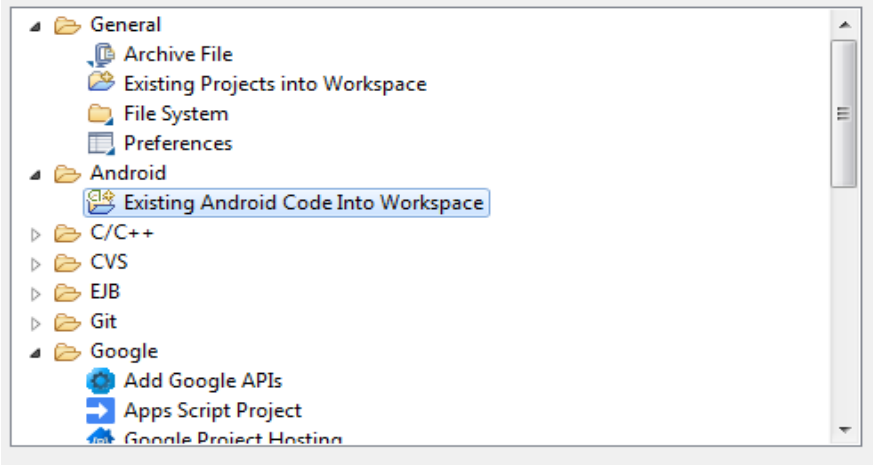


Ilustración 41: Instalación - Proyecto Android

Navegamos hasta la ruta del zip descomprimido, seleccionamos el proyecto Moventaa y lo importamos.

Una vez finalizado el proceso, nuestro workspace debe quedar de la siguiente manera:

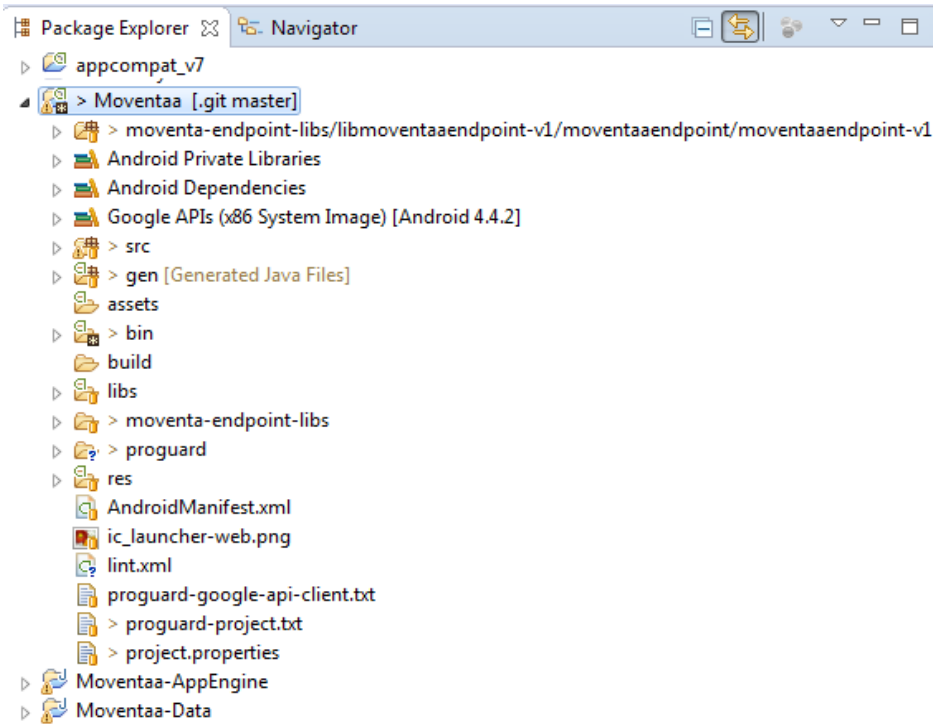


Ilustración 42: Instalación - Workspace final

Pasos adicionales para un proyecto Android:

Nota: Los pasos siguientes solo deben ejecutarse si encontramos problemas a la hora de compilar el proyecto o errores en la importación de los recursos de android del tipo R.

Configuración de la compatibilidad para versiones anteriores de Android, es necesario incluir como “Library” la versión de compatibildiad appcompat_v7, tal y como se muestra en la siguiente imagen:

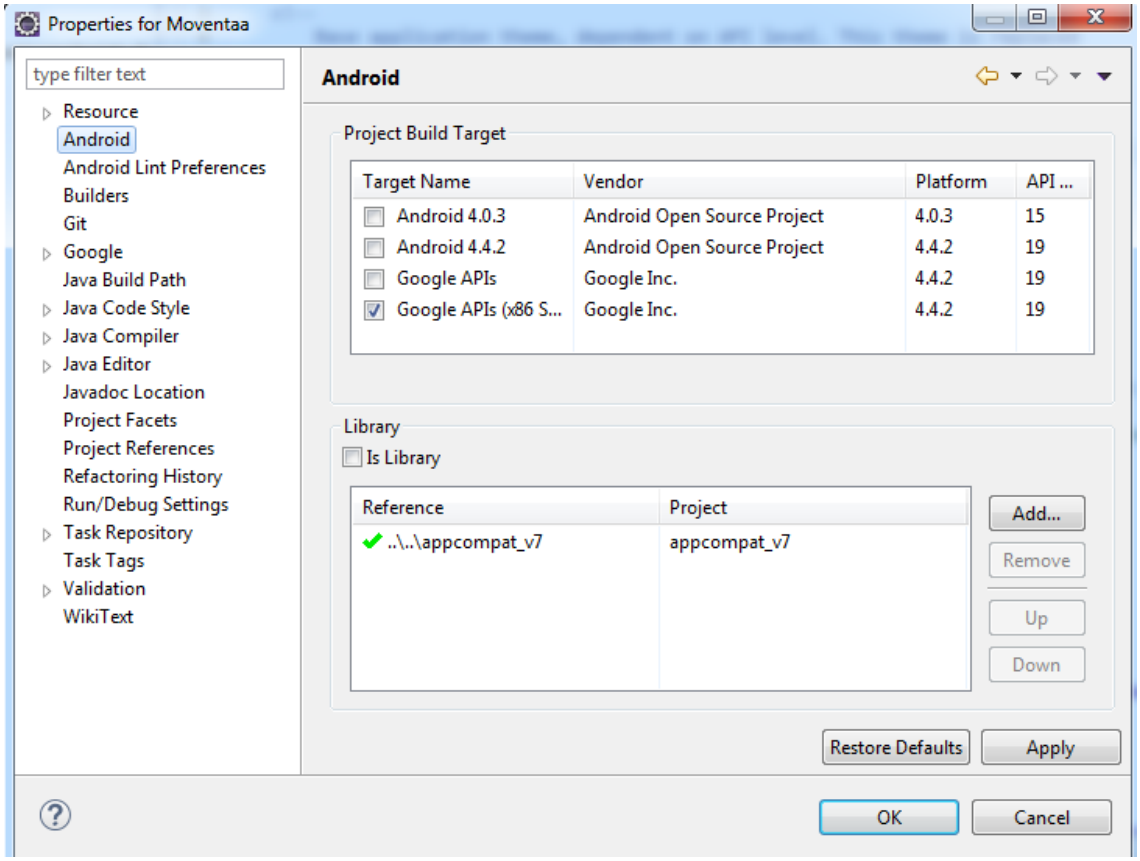


Ilustración 43: Android propiedades proyecto

Compilación para ButterKnife.

Configuramos en “Annotacion Procesing” que genere al .apt_generated

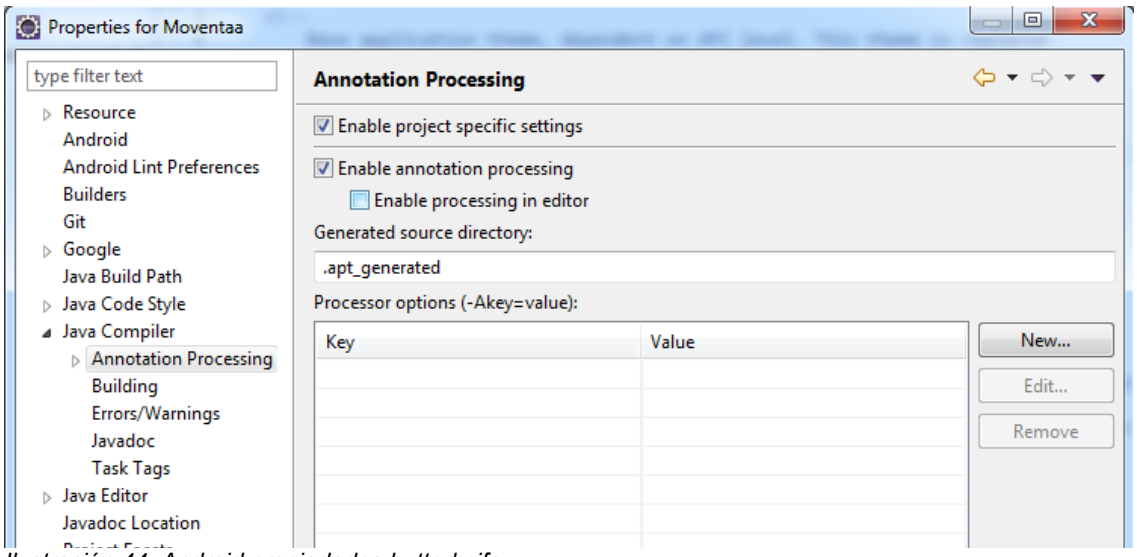


Ilustración 44: Android propiedades butterknife

Especificamos la ubicación de la biblioteca de butterknife

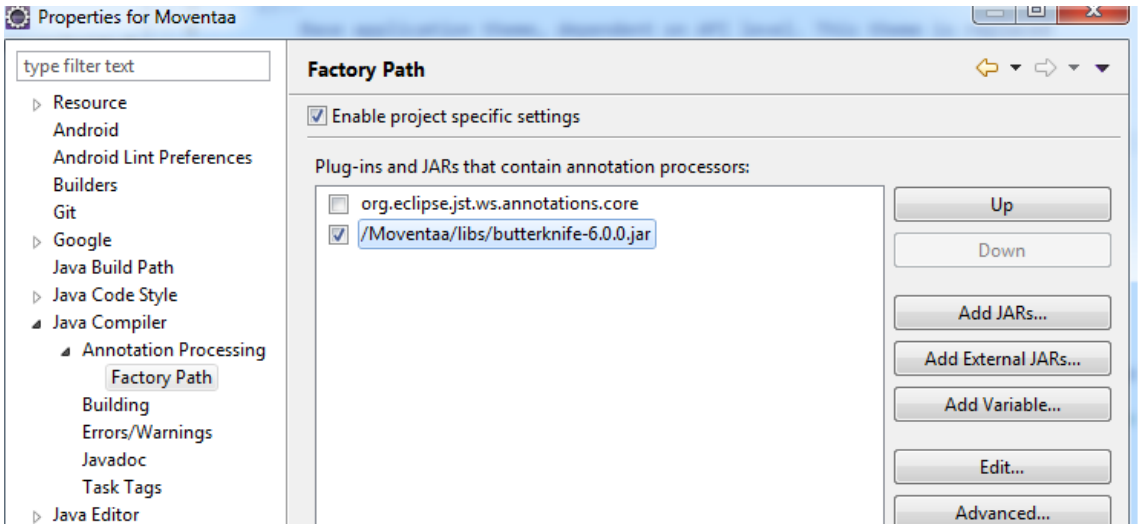


Ilustración 45: Andorid propiedades butterknife factory

1.2 Compilación

Es posible que durante el proceso de importación se generen de nuevo los endpoint de la aplicación cliente, en este caso es necesario eliminar la referencia incluida dentro del proyecto.

1.3 Población de los datos en la base de datos

Nota: este punto puede ser obviado, la aplicación se entrega con una base de datos ya poblada con información, en cualquier caso se explica el procedimiento por si fuera necesario.

Pre-requisitos necesarios:

- Python 2.7, instalado en la ruta

  `c:\Python27\python`

- App Engine instalado en la ruta

```
C:\Program Files (x86)\Google\google_appengine\
```

Nos posicionaremos en la carpeta del proyecto Moventaa-Data, dentro existe un fichero llamado populate.bat

Ejecutamos dicho fichero mostrando por consola el proceso de carga:

```
Dump data....
---------------------------------
Filecsv dump_Localidades.csv
Kind: Localidad
Operation: upload_data
---------------------------------
08:46 PM Uploading data records.
[INFO     ] Logging to bulkloader-log-20141209.204650
[INFO     ] Throttling transfers:
[INFO     ] Bandwidth: 250000 bytes/second
[INFO     ] HTTP connections: 8/second
[INFO     ] Entities inserted/fetched/modified: 20/second
[INFO     ] Batch Size: 10
Password for test@example.com:
[INFO     ] Opening database: bulkloader-progress-20141209.204650.sql3
[INFO     ] Connecting to localhost:8888/remote_api
[INFO     ] Starting import; maximum 10 entities per post
.
[INFO     ] 9 entities total, 0 previously transferred
[INFO     ] 9 entities (2376 bytes) transferred in 6.1 seconds
[INFO     ] All entities successfully transferred
---------------------------------
Filecsv dump_Provincia.csv
Kind: Provincia
Operation: upload_data
---------------------------------
08:47 PM Uploading data records.
[INFO     ] Logging to bulkloader-log-20141209.204700
[INFO     ] Throttling transfers:
[INFO     ] Bandwidth: 250000 bytes/second
[INFO     ] HTTP connections: 8/second
[INFO     ] Entities inserted/fetched/modified: 20/second
[INFO     ] Batch Size: 10
Password for test@example.com: _
```

Ilustración 46: Proceso de carga

El proceso pide un password, pulsar la tecla “enter”.

1.4 Ejecución de las aplicaciones

Primero debemos levantar la parte servidora, para ello seleccionamos el proyecto Moventaa-AppEngine, botón derecho del ratón:

Run As > Web Application

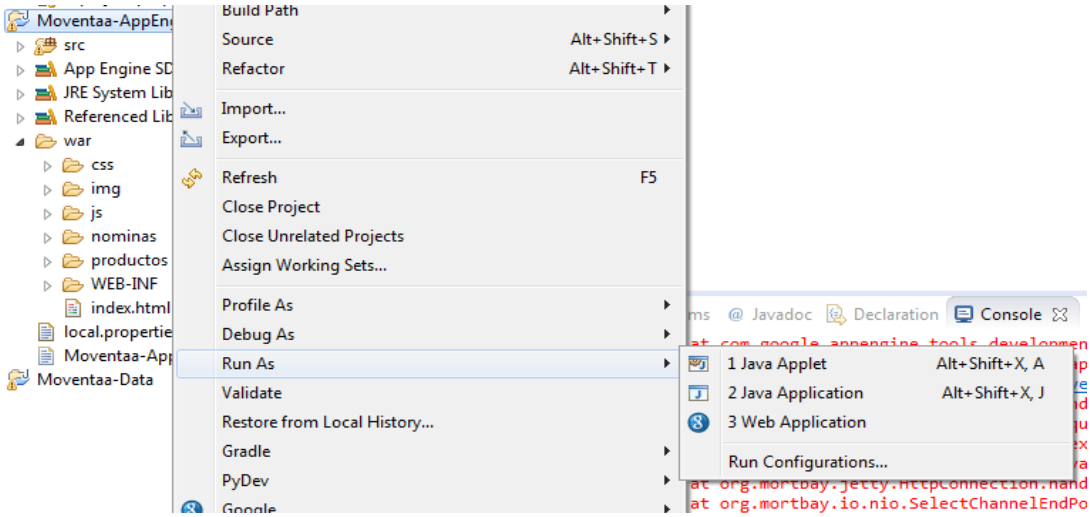


Ilustración 47: Ejecutar como web-aplication

Una vez lanzada la aplicación web si accedemos a la dirección http://localhost:8888/ se nos debe de mostrar una pantalla como la siguiente:

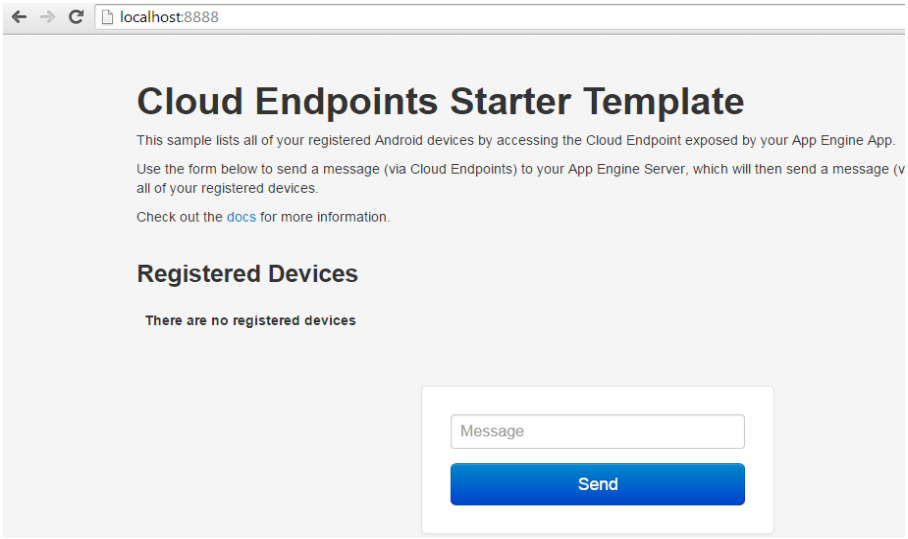
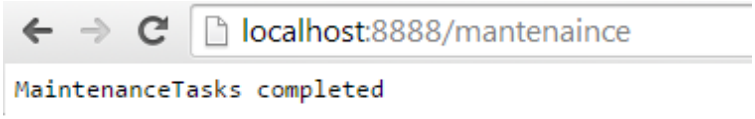


Ilustración 48: Arranque parte servidora

Ahora debemos acceder a la dirección http://localhost:8888/mantenaince para genera el documento de geolocalizaciones de los centros,



Si se muestra el texto: “MaintenanceTasks completed” es el que proceso se ha realizado con éxito.

Una vez comprobado que está disponible el servidor, podemos poner en marcha la aplicación emulada en un entorno de Android.

1.5 Instalación solo del aplicativo android.

Ejecutar un emulador de android:

SDK Manager.exe → Tools → Manage Avds → New Start

Ejecutamos la consola (Windows), Run → cmd y nos movemos a la carpeta plataform_tools del sdk de android.

Pegamos el fichero APK suministrado en 'android-sdk\tools' o 'plataform-tools' y escribimos:

adb install [.apk path]

Ejemplo:

adb install Moventaa.apk

Nota, en entorno Linux el comando es:

```
./adb install fileName.apk (Linux or Mac)
```

Una vez instalado en el emulador ejecutamos la aplicación llamada Moventaa.

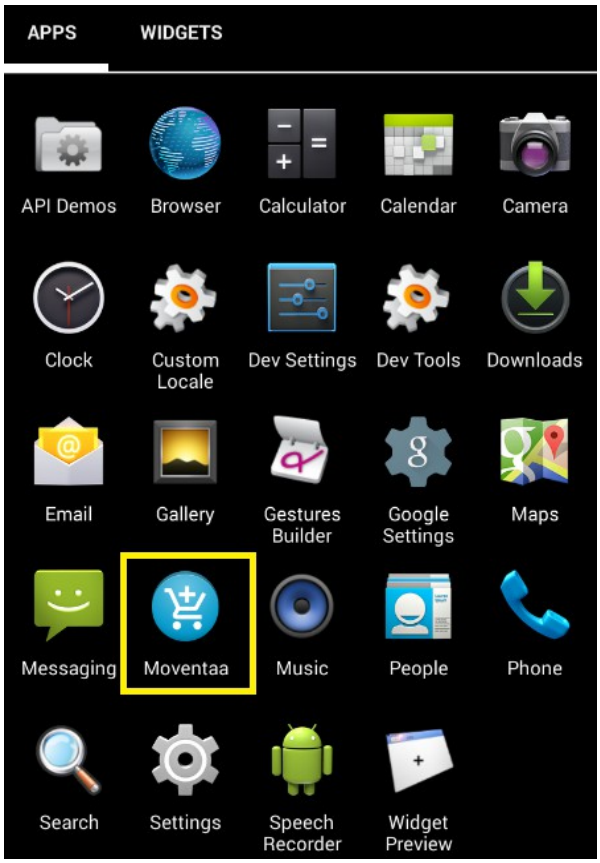


Ilustración 49: Aplicación Moventaa en emulador