Universidad Rey Juan Carlos

**INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN**

**Curso Académico 2010/2011**

**Proyecto de Fin de Carrera**

# GENERACIÓN DE UNA APLICACIÓN PARA BLACKBERRY ORIENTADA A LA SOLICITUD DE INFORMACIÓN EN INSTALACIONES FOTOVOLTAICAS

Autor: Francisco José Guerrero Peña

Tutor: Emilio José San Martín Fuentes

# Agradecimientos

*A mis padres y hermana, por dar prioridad a mi formación y caprichos estudiantiles ante cualquier cosa.*

*A Tamara, por tranquilizarme en los momentos tensos, y animarme en los difíciles.*

*A mi tutor, por su esfuerzo y ayuda en la realización del proyecto.*

*A Jesús, Miguel y Javier, por su comprensión, ayuda y ánimos.*

*A mis compañeros, por hacerme más llevadero el paso por la Ingeniería.*

# Índices

Índice de contenidos.……..….- 4 -

Índice de figuras.....……...- 8 -

Resumen...……..…….....- 11 -

# Índice de contenidos

**1. Introducción ..... - 12 -**

1.1 BlackBerry. ..... - 12 -

1.1.1 RIM ..... - 12 -

1.1.2 Software propietario. ..... - 12 -

1.1.2.1 Ventajas del Software propietario. ..... - 13 -

1.1.2.2 Desventajas del Software propietario. ..... - 13 -

1.1.3 Sistema operativo BlackBerry ..... - 14 -

1.1.4 Introducción para el desarrollo de aplicaciones. ..... - 14 -

1.1.4.1 J2ME. ..... - 15 -

1.1.5 Definición de API (Interfaz de Programación de Aplicaciones). ..... - 15 -

1.2 Tecnología empleada. ..... - 16 -

1.2.1 Eclipse. ..... - 16 -

1.2.1.1 Definición de Plug-in. ..... - 16 -

1.2.2 BlackBerry Java Plug-in. ..... - 16 -

1.2.2.1 Introducción. ..... - 16 -

1.2.2.2 Principales beneficios. ..... - 16 -

1.2.3 BlackBerry App World. ..... - 18 -

1.2.3.1 ¿Qué es? ..... - 18 -

1.2.3.2 Características principales. ..... - 18 -

1.2.3.3 Tiendas de aplicaciones similares. ..... - 19 -

1.2.3.4 Principales aplicaciones. ..... - 19 -

1.2.3.5 Preguntas más frecuentes en el entorno BlackBerry App World. ..... - 20 -

**2. Empresa ..... - 22 -**

2.1 Dedicación de DeltaTec Systems. ..... - 22 -

2.1.1 Introducción. ..... - 22 -

2.1.2 Sector solar. ....- 22 -

2.2 Sistema de monitorización. ....- 22 -

2.2.1 Dispositivo instalado en los parques solares. ....- 23 -

2.2.2 Datos e informes ofrecidos. ....- 23 -

2.3 Clientes. ....- 24 -

2.4 Objetivos del proyecto ....- 24 -

**3. Desarrollo ....- 25 -**

3.1 Arquitectura del Sistema. ....- 25 -

3.1.1 Servidor. ....- 26 -

3.1.2 Componentes del Software. ....- 26 -

3.2 Protocolo HTTP. ....- 26 -

3.2.1 Solicitudes HTTP. ....- 27 -

3.2.2 Respuesta HTTP. ....- 27 -

3.3 Especificación del Sistema. ....- 28 -

3.4 Requisitos. ....- 29 -

3.5 Diseño de Casos de uso. ....- 30 -

3.5.1 Actor. ....- 31 -

3.5.2 Interfaz de usuario. ....- 31 -

3.6 Análisis. ....- 32 -

3.7 Proceso. ....- 38 -

3.8 Diagrama de clases ....- 42 -

3.8.1 InmodoClients. ....- 42 -

3.8.2 SaludationsScreen. ....- 43 -

3.8.3 Communications. ....- 44 -

3.8.4 Installations. ....- 44 -

3.8.5 TextBoxField. ....- 45 -

3.8.6 ProgressBar. ....- 45 -

3.8.7 ScrollableImageField. ..... - 46 -

3.8.8 Description. ..... - 47 -

**4. Implementación ..... - 48 -**

4.1 Clase InmodoClient.java ..... - 48 -

4.2 Clase SaludationsScreen.java ..... - 48 -

4.3 Clase Communications.java ..... - 51 -

4.4 Clase Instalacion.java ..... - 52 -

4.5 Clase Data.java ..... - 53 -

4.6 Clase Installations.java ..... - 53 -

4.7 Clase TextBoxField.java ..... - 55 -

4.8 Clase ProgressBar.java ..... - 56 -

4.9 Clase ScrollableImageField.java ..... - 57 -

4.10 Clase Description.java ..... - 59 -

4.10.1 Constructor ..... - 60 -

4.10.2 Menu1_clear, menu2_clear, menu3_clear y menú4_clear ..... - 60 -

4.10.3 Menu1 ..... - 61 -

4.10.4 Menu2 ..... - 61 -

4.10.5 Menu3 ..... - 62 -

4.10.6 Menu4 ..... - 63 -

4.10.7 Cerrar el programa ..... - 63 -

**5. Manual de usuario ..... - 64 -**

5.1 Instalación ..... - 64 -

5.2 Inicio ..... - 65 -

5.3 Elección de instalación. ..... - 66 -

5.4 Menú principal de la aplicación. ..... - 67 -

5.4.1 Sumario de producción. ..... - 67 -

5.4.2 Condiciones atmosféricas. ..... - 68 -

5.4.3 Estadística mensual. ..... - 70 -

5.4.4 Envío de sugerencias. ..... - 70 -

5.4.5 Volver. ..... - 71 -

5.5 Salir de la aplicación. ..... - 71 -

**6. Trabajos futuros y conclusiones ..... - 73 -**

6.1 Conclusiones. ..... - 73 -

6.2 Trabajos futuros. ..... - 74 -

**7. Referencias ..... - 75 -**

# Índice de figuras


Figura 1.1 Logotipos de RIM y de BlackBerry…….- 12 -

Figura 1.2 BlackBerry App World…….- 17 -

Figura 1.3 Diferentes tiendas de aplicaciones para dispositivos móviles……..- 19 -

Figura 2.1 Logotipo de la empresa DeltaTec Systems……- 22 -

Figura 2.2 Dispositivo utilizado en los parques solares……- 23 -

Figura 3.1 Imagen del Sistema……- 25 -

Figura 3.2 Ejemplo de una solicitud HTTP……- 27 -

Figura 3.3 Ejemplo de una respuesta HTTP……- 28 -

Figura 3.4 Funcionamiento del Sistema……- 29 -

Figura 3.5 Diagrama de casos de uso……- 31 -

Figura 3.6 Diagrama de secuencia de "Autenticación de usuario"……- 33 -

Figura 3.7 Diagrama de secuencia del caso de uso "Consultar datos de un parque solar"……- 34 -

Figura 3.8 Diagrama de secuencia del caso de uso "Visualizar usuario"……- 35 -

Figura 3.9 Diagrama de secuencia del caso de uso "Enviar e-mail sugerencia"……- 36 -

Figura 3.10 Diagrama de secuencia del caso de uso "Consultar datos meteorológicos"……- 37 -

Figura 3.11 Diagrama de secuencia del caso de uso "Visualizar gráfica resumen"……- 38 -

Figura 3.12 Esquema para introducir el API en el plug-in……- 39 -

Figura 3.13 Selección de los elementos a configurar……- 40 -

Figura 3.14 Esquema para seleccionar internet en el simulador……- 40 -

Figura 3.15 Visualización de la carpeta "res"……- 41 -

Figura 3.16 Diagrama de clases de la clase InmodoClients……- 42 -

Figura 3.17 Diagrama de clases de la clase SaludationsScreen……- 43 -

Figura 3.18 Diagrama de clases de la clase Communications……- 44 -

Figura 3.19 Diagrama de clases de la clase Installations……- 44 -

Figura 3.20 Diagrama de clases de la clase TextBoxField……..…..- 45 -
Figura 3.21 Diagrama de clases de la clase ProgressBar……..- 45 -
Figura 3.22 Diagrama de clases de la clase ScrollableImageField……..…..- 46 -
Figura 3.23 Diagrama de clases de la clase Description……..- 47 -
Figura 4.1 Esquema de la clase InmodoClients……..- 48 -
Figura 4.2 Esquema de la clase SaludationsScreen……..- 48 -
Figura 4.3 Diferencia entre BasicEditField y PasswordEditField……..- 49 -
Figura 4.4 Ejemplo de LabelField……..- 49 -
Figura 4.5 Ejemplo de StandardTitleBar……..- 50 -
Figura 4.6 Ejemplo de ButtonField……..- 50 -
Figura 4.7 Esquema de la clase Communications……..- 51 -
Figura 4.8 Esquema de la clase Instalacion……..- 52 -
Figura 4.9 Esquema de la clase Data……..- 53 -
Figura 4.10 Esquema de la clase Installations……..- 53 -
Figura 4.11 Ejemplo de ObjtectChoiceField……..- 54 -
Figura 4.12 Esquema de la clase TextBoxField……..- 55 -
Figura 4.13 Esquema de la clase ProgressBar……..- 56 -
Figura 4.14 Ejemplo de ProgressBar……..- 57 -
Figura 4.15 Esquema de la clase ScrollableImageField……..- 57 -
Figura 4.16 Esquema de la clase Description……..- 59 -
Figura 4.17 Ejemplo de ToolbarManager……..- 60 -
Figura 4.18 Iconos representativos de todos los menús……..- 62 -
Figura 5.1 Imagen en la pantalla del dispositivo a la hora
de descargar la aplicación……..- 64 -
Figura 5.2 Representación de la pantalla de inicio……..- 65 -
Figura 5.3 Usuario inválido o contraseña inválida……..- 66 -
Figura 5.4 Pantalla de selección de instalación……..- 66 -
Figura 5.5 Barra de herramientas……..- 67 -
Figura 5.6 Representación del sumario de producción……..- 68 -
Figura 5.7 Pantalla dedicada a las condiciones atmosféricas……..- 69 -

Figura 5.8 Visualización de error…….……..……..……..……- 69 -

Figura 5.9 Gráfica de producción diaria…….…….……..……...- 70 -

Figura 5.10 Aspecto de la pantalla dedicada al envío de sugerencias…….....- 71 -

Figura 5.11 La aplicación se despide del usuario…….…….….- 72 -

# Resumen

El presente escrito trabaja algunos de los aspectos necesarios para elaborar una aplicación de consulta orientada a la línea de teléfonos inteligentes BlackBerry.

Es conocida la complejidad de este tipo de sistemas atendiendo a su arquitectura, de gran tamaño, y al manejo de diferente información. En consecuencia, el documento tiene como objetivo principal explicar a los lectores la estructura, funcionamiento y conocimientos necesarios para la creación de una interfaz de usuario que facilite la solicitud de información en instalaciones fotovoltaicas en este tipo de smartphones.

Inicialmente el documento, informa de la diversidad tecnológica relacionada con la aplicación generada. Son de gran variedad los programas que se describen y aplican en la puesta en marcha del programa siendo de especial relevancia el entorno Eclipse.

DeltaTec Systems es una empresa encargada del software de diversas instalaciones solares, motivo por el que se ha creído conveniente la creación de una aplicación móvil para los dispositivos de la compañía canadiense RIM.

Una vez definida la idea principal, se procede a la reunión de documentación para el desarrollo de dicha aplicación mediante informes técnicos. El primer reporte, y el más importante, es el que especifica la funcionalidad del sistema, quedando expresado en un esquema funcional.

Para la confección del software se han seguido las pautas estándar. En primer lugar, se ha fijado el diagrama de casos de uso, donde quedan plasmadas las ideas principales que deberá poseer la aplicación en las distintas situaciones a las que ha de ser sometida. Se continúa con la parte que concierne al análisis del sistema, en el cual se define el sistema en términos de interacción entre objetos del mismo.

Se prosigue con la confección de los diagramas de clase en notación UML, en los cuales se representan los modelos usados dentro del sistema y las relaciones existentes entre las distintas clases, facilitando así al lector una comprensión y proveyendo una idea más visual del proceso de creación y de los pasos que se han seguido.

Asimismo se ha creado un manual de usuario, en el que se describen las posibles pantallas que pueden ser generadas por la aplicación. El proceso comprende desde la descarga de la aplicación, hasta el uso y la visualización de la información deseada.

En definitiva, el proyecto detalla los pasos a seguir para la elaboración y posterior carga de una aplicación orientada a la solicitud de información en instalaciones fotovoltaicas para dispositivos móviles BlackBerry.

# 1. Introducción

## 1.1 BlackBerry

### 1.1.1 RIM

RIM fue fundada por Mike Lazaridis, Presidente y Co-CEO[1], en 1984 cuando era estudiante en Ontario; y Jim Balsillie asumió el cargo de Co-CEO en 1992. Hace diez años, RIM lanzó la solución BlackBerry como una solución inalámbrica de e-mail para clientes corporativos que cambió el panorama de las comunicaciones. A lo largo de los años, esto se expandió hacia una plataforma inalámbrica robusta y multifacética que soporta una gran variedad de aplicaciones de comunicación, información y entretenimiento para usuarios corporativos y finales. En los últimos años, RIM ha lanzado nuevos dispositivos innovadores y populares para el mercado, incluido el BlackBerry Pearl, el BlackBerry Curve, el BlackBerry Bold y el BlackBerry Storm, el cual ganó el premio Mobile World Congress en 2009.[1]

Figura 1.1: Logotipos de RIM y de BlackBerry.

### 1.1.2 Software propietario

El software propietario o privativo se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo, o cuyo código fuente no está disponible, o está restringido.

[1] Co-CEO: Director ejecutivo.

Según la Fundación de Software Libre (FSF) este concepto se aplica a cualquier software que no es libre, o que solo lo es parcialmente, ya sea por la prohibición de uso, redistribución, o modificación, o porque requiere un permiso expreso del titular del software.

#### 1.1.2.1 Ventajas del Software propietario

- **Propiedad y decisión de uso del software por parte de la empresa:** El productor invierte dinero en la creación de aplicaciones, o de hardware, concretamente en el estudio y desarrollo de la misma, y de no ser por este estilo no sacaría beneficio alguno, lo que conlleva no compensar la inversión, y desvanecer el continuo desarrollo.

- **Mejor acabado en la mayoría de aplicaciones:** El desarrollador de software propietario, generalmente, da un mejor acabado a las aplicaciones en cuestiones, tanto de estética, como de usabilidad de la aplicación. Aunque muchas aplicaciones de software libre tienen un excelente acabado, aquí se nota de forma especial el cuidado y el esfuerzo del desarrollador de software propietario.

- **Las aplicaciones número uno son propietarias.**

- **Menor necesidad de técnicos especializados:** El mejor acabado de la mayoría de sistemas de software propietario y los estándares de facto actuales permiten una simplificación del tratamiento de dichos sistemas, dando acceso a su uso y administración, a técnicos que requieren una menor formación, reduciendo costes de mantenimiento.

- **Mayor mercado laboral actual.**

- **Protección de las obras con copyright.** [2]

#### 1.1.2.2 Desventajas del Software propietario

- **Imposibilidad de copia, modificación o redistribución.**

- **Venta de la compañía fabricante:** Si una empresa de software es comprada por otra más poderosa, es probable que esa línea de software quede descontinuada y nunca más en la vida vuelva a tener una modificación.

- **El soporte de la aplicación es exclusivo del propietario.**

- **El coste de las aplicaciones es mayor.** [2]

### 1.1.3 Sistema operativo BlackBerry

El BlackBerry OS se remonta a la aparición de los primeros handheld[2] en 1999. Estos dispositivos permiten el acceso a correo electrónico, navegación web y sincronización con programas como Microsoft Exchange o Lotus Notes, aparte de poder hacer las funciones usuales de un teléfono móvil.

Este SO es desarrollado actualmente por RIM que también ofrece servicios de correo electrónico a otros dispositivos a través de BlackBerry Connect. [3]

#### Características

El SO BlackBerry está claramente orientado a su uso profesional como gestor de correo electrónico y agenda. Desde la versión actual, se puede sincronizar el dispositivo con el correo electrónico, el calendario, tareas, notas y contactos de Microsoft Exchange Server.

BlackBerry Entrepise Server (BES) proporciona el acceso y organización del e-mail a grandes compañías identificando a cada usuario con un único BlackBerry PIN. Los usuarios más pequeños cuentan con el software BlackBerry Internet Service, programa más sencillo que proporciona acceso a Internet y a correo POP3 / IMAP / Outlook Web Access sin tener que usar BES. [3]

#### Ventajas e inconvenientes

- Proporciona acceso en tiempo real al correo electrónico.
- Permite la sincronización con Exchange, Notes y GroupWise (diversos correos electrónicos, calendarios, contactos, etc.)
- El SO es software propietario.
- Orientado principalmente a clientes empresariales.

### 1.1.4 Introducción para el desarrollo de aplicaciones

Los dispositivos BlackBerry están destinados a negocios, mejor dicho, están pensados para clientes de una red corporativa, que incluye soluciones integrales con correo electrónico, servicio web, etc. La plataforma de desarrollo de estos es J2ME, aunque existen una serie de peculiaridades, un entorno de desarrollo de aplicaciones llamado BlackBerry MDS Studio.

Java no es la verdadera plataforma para el desarrollo de estas aplicaciones, BlackBerry ha creado su propia plataforma a partir de una base que sí que es J2ME, el sistema se llama BlackBerry Handheld Software, este sistema ofrece por un lado las

[2] Handheld: Describe una computadora portátil que se puede llevar en una mano a cualquier parte mientras se utiliza.

APIs de J2ME, tanto CLDC 1.1 y de MIDP 2.0, y por otro lado las propias de BlackBerry.

Estas APIs ofrecen servicios básicos como interfaz de usuario, localización, acceso a redes y servicios, mensajería, etc., también sirven para darle a la aplicación un toque personal, y característico de BlackBerry.

#### 1.1.4.1 J2ME

J2ME es la versión de Java orientada a los dispositivos móviles. Debido a que los dispositivos móviles tienen una potencia de cálculo baja e interfaces de usuario pobres, es necesaria una versión específica de Java destinada a estos dispositivos, ya que el resto de versiones de Java, J2SE o J2EE, no encajan dentro de este esquema. J2ME es por tanto, una versión reducida de J2SE, plataforma estándar de Java usada en sistemas de escritorio y servidor.

Los principales componentes de la plataforma J2ME son Connected Limited Device Configurations (CLDC) y Mobile Information Device Profiles (MIDP). [4]

**CLDC**

Esta configuración está diseñada para dispositivos con conexiones de red intermitentes, procesadores lentos y memoria limitada como son teléfonos móviles, asistentes personales (PDAs), etc.

La especificación CLDC se ha desarrollado dentro de Java, junto con 500 partners[3] que representan a las industrias fabricantes de dispositivos wireless, proveedores de servicios y terminales de venta.

**MIDP**

Es una versión de Java integrada en el hardware de dispositivos móviles relativamente modernos que permite el uso de programas denominados MIDlets, tales como juegos, aplicaciones o todo tipo de software.

### 1.1.5 Definición de API (Interfaz de Programación de Aplicaciones).

Conjunto de convenciones internacionales que definen cómo deben invocarse unas determinadas funciones de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos APIs comunes a los que deben ajustarse todos los desarrolladores de aplicaciones.

[3] Parterns: Usuarios que codifican gran cantidad de contenido, gozando de buena popularidad, a los que se les da la opción de monetizar sus creaciones con anuncios publicitarios.

En definitiva, herramientas de programación para rutinas, protocolos y software.

## 1.2 Tecnología empleada

### 1.2.1 Eclipse

Eclipse es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java.

Se trata de un entorno de desarrollo integrado (IDE) en el que se puede encontrar todas las herramientas y funciones necesarias para el trabajo, recogidas además en una atractiva interfaz que lo hace fácil de usar.

Cuenta con un editor de texto donde se puede ver el contenido del fichero en el que se está trabajando, una lista de tareas, y otros módulos similares. Si bien las funciones de Eclipse son más bien de carácter general, las características del programa se pueden ampliar y mejorar mediante el uso de plug-ins, uno de ellos es el seleccionado para este trabajo BlackBerry Java Plug-in Eclipse. [5]

#### 1.2.1.1 Definición de Plug-in.

Un plug-in es un programa que incrementa o aumenta las funcionalidades de un programa principal. Por lo general es producido por una compañía diferente a la que produjo el primer programa. Aplicando la definición a este trabajo, el programa principal es el ya anunciado Eclipse, y la compañía diferente es BlackBerry, creador del plug-in con el que se hace frente al código y se ofrece una simulación de lo que se está creando.

### 1.2.2 BlackBerry Java Plug-in

#### 1.2.2.1 Introducción

Se trata de una herramienta que permite a los desarrolladores de Eclipse crear y probar perfectamente robustas aplicaciones móviles para la plataforma BlackBerry desde el entorno de desarrollo integrado de Eclipse. Hace posible que los usuarios de Eclipse desarrollen aplicaciones para el creciente volumen de usuarios de smartphones BlackBerry en todo el mundo, manteniendo a la vez la familiaridad del entorno de desarrollo. También permite a los desarrolladores de Eclipse aprovechar las capacidades y beneficios inherentes a la arquitectura del sistema y las herramientas de desarrollo de BlackBerry.

#### 1.2.2.2 Principales beneficios

- **Extiende una experiencia de desarrollo familiar al desarrollo de aplicaciones móviles:** El plug-in es una descarga gratuita que se integra perfectamente

en el desarrollador, permitiendo desarrollar aplicaciones en Java para la plataforma BlackBerry sin necesidad de aprender un nuevo entorno de desarrollo.

- **Simplifica el desarrollo de aplicaciones móviles:** Soporta estándares de desarrollo Java evitando la complejidad del desarrollo de aplicaciones móviles. El plug-in permite a los desarrolladores aprovechar las herramientas de desarrollo, prueba, reparación y optimización disponibles con BlackBerry, así como la seguridad inherente, la eficiencia de la red, las capacidades de integración de sistemas, la fiabilidad y la arquitectura basada en tecnologías push[4] de la plataforma BlackBerry.

- **Acelera la prueba de las aplicaciones móviles:** Los simuladores integrados de escritorio que apoyan la gama de smartphones BlackBerry permiten a los desarrolladores comprobar fácilmente sus aplicaciones móviles en aspectos como las interacciones de la red móvil y la experiencia del usuario de extremo a extremo.

- **Optimiza el diseño de aplicaciones móviles para los smartphones BlackBerry:** Ofrece la capacidad de crear una sólida aplicación con una interfaz de usuario optimizada para los usuarios de smartphones BlackBerry. [6]

Figura 1.2: BlackBerry App World.

[4] Tecnologías push: describe un estilo de comunicaciones sobre Internet donde la petición de una transacción se origina en el servidor.

### 1.2.3 BlackBerry App World

#### 1.2.3.1 ¿Qué es?

Aplicación íntegra en los dispositivos móviles de BlackBerry en la que los usuarios podrán descubrir y descargar una amplia gama de aplicaciones gratuitas, tanto personales como empresariales.

BlackBerry App World está diseñada para presentar automáticamente un catálogo de aplicaciones disponibles para usuarios de modelos específicos de estos dispositivos, estas aplicaciones pueden ser tanto gratuitas como de previo pago.

Dispone de más de 10.000 aplicaciones diferentes, número muy inferior a sus competidores como veremos más adelante.

#### 1.2.3.2 Características principales

- **Página principal actualizada:** BlackBerry App World presenta varias aplicaciones en su página principal, lo que facilita a los usuarios navegar a través de un número selecto de aplicaciones recomendadas, que varían de vez en cuando, ayudando a los usuarios a descubrir nuevas aplicaciones para todos los aspectos de la vida.

- **Categorías:** Cuenta con una variedad de sectores de aplicaciones que incluyen juegos, entretenimiento, mensajería instantánea, redes sociales, noticias, clima, productividad, entre otras.

- **Almacenamiento de aplicaciones:** Incluye una carpeta llamada "My World", que puede realizar un seguimiento de las aplicaciones que el usuario ha descargado. Es un espacio de almacenamiento personal que permite a los usuarios desinstalar y reinstalar aplicaciones y transferir fácilmente las aplicaciones compradas con anterioridad.

- **Evaluaciones:** Las aplicaciones en BlackBerry App World pueden ser evaluadas por los usuarios y sus comentarios se reflejan en puntuaciones con estrellas. Los usuarios pueden fácilmente acceder y leer las revisiones de otros usuarios, ver imágenes y leer una descripción de cada aplicación antes de descargarla.

- **Otras**: Mayores descargas (área con el "top ten" de las aplicaciones más descargadas), filtro (zona de búsqueda de aplicaciones), recomendaciones (espacio reservado para la recomendación de aplicaciones mediante correo electrónico, SMS, etc.), tiempo limitado (requiere la identificación del usuario cada 20 minutos), etc. [7]

Figura 1.3: Diferentes tiendas de aplicaciones para dispositivos móviles.

#### 1.2.3.3 Tiendas de aplicaciones similares

Actualmente disponemos de cinco sistemas operativos diferentes para un teléfono móvil, cada uno de ellos con su propia tienda de aplicaciones. Cabe destacar App Store, disponible para teléfonos Apple, cuenta con más de 280.000 aplicaciones, en segundo lugar encontramos al sistema operativo Android, con un número elevado de aplicaciones, alrededor de 150.000, y muy por debajo están Ovi Store particular de los dispositivos Nokia, y Market Place Microsoft propia del sistema operativo Windows Phone.

#### 1.2.3.4 Principales aplicaciones

Podemos diferenciar dos tipos de descargas de aplicaciones, las gratuitas y las de pago.

**Gratuitas**

- **Twitter:** Página web en la que se puede enviar mensajes cortos, denominado tweets y con valor máximo de 140 caracteres, a los amigos que se tienen agregados en la lista de contactos.

- **WhatsApp Messenger**: Aplicación de mensajería móvil multiplataforma que sustituye a los SMS y funciona a través del plan de datos de Internet, del cual deberían disponer los usuarios de dichos dispositivos.

- **Pandora:** Simulación de radio, en el que se introduce el autor al que se quiere escuchar y el sistema crea una estación de radio personalizada a partir de él, con otros artistas y canciones que suenan como el autor introducido, en cuanto a ritmos, tiempos, etc.

**De pago**

- **Chat de Facebook:** Aplicación para chatear con los contactos de la red social facebook, debido a que directamente desde la página, BlackBerry no da opción a dicho chat.

- **Diversos juegos.**

#### 1.2.3.5 Preguntas más frecuentes en el entorno BlackBerry App World

**¿Por qué no es posible para todos los usuarios de smartphone BlackBerry obtener acceso a todas las aplicaciones?**

Los proveedores de BlackBerry App World tienen la posibilidad de restringir la disponibilidad y visibilidad de sus aplicaciones según los siguientes criterios:

- Modelo del smartphone BlackBerry, versión del Software, proveedor de servicios inalámbricos, país e idioma.

Si una aplicación no aparece en el smartphone BlackBerry, se habrá incumplido uno o más de estos criterios.

**¿Qué métodos de pago se pueden utilizar para adquirir aplicaciones a través de BlackBerry App World?**

PayPal es el socio oficial para las compras realizadas a través de BlackBerry App World. Esta es una empresa que permite el pago y transferencias de dinero por internet. Actualmente se ha puesto en marcha el pago mediante factura telefónica mensual, aunque por ahora solo con la compañía Movistar.

**¿Cómo determina RIM si una aplicación es apropiada para BlackBerry App World?**

RIM revisa las aplicaciones enviadas para determinar la idoneidad[5] del contenido y realizar diversas pruebas técnicas que garanticen que la aplicación satisface las directrices para proveedores de BlackBerry App World.

[5] Idoneidad: Adecuación que existe entre las características de una persona o cosa y la función, actividad o trabajo que debe desempeñar.

Los proveedores deben probar la compatibilidad de sus aplicaciones con los dispositivos seleccionados durante el proceso de envío para evitar fallos técnicos y rechazos de la aplicación. [8]

# 2. Empresa

## 2.1 Dedicación de DeltaTec Systems

### 2.1.1 Introducción

DeltaTec Systems es una empresa dedicada a proveer los recursos y servicios tecnológicos necesarios para dar eficacia a la tarea diaria de una empresa ajena.

Los servicios que ofrecen van referidos a procesos informáticos, definidos por la empresa a la cual se le realiza el trabajo.

La oferta de servicios de la compañía abarca todos los sectores, proporcionando a cada sector la solución más fiable, económica, y adecuada a su actividad laboral. [9]

Figura 2.1. Logotipo de la empresa Deltatec Systems.

### 2.1.2 Sector solar

Uno de los sectores que ofrece la empresa es la monitorización[6] de parques solares. Ofreciendo datos de producción, presentando estos de forma personalizada a los clientes-inversores de las instalaciones.

Gracias a esta manipulación de los datos, la producción de las placas solares, mejora su rendimiento y permite a los clientes realizar un mantenimiento completo y funcional de sus instalaciones.

## 2.2 Sistema de monitorización

El sistema de monitorización se compone de dos partes, una de ellas instalada en la propia planta fotovoltaica, y la otra ubicada en un servidor remoto para el almacenamiento de los datos y la gestión de los mismos.

[6] Monitorización: Control de las constantes vitales a través de monitores.

### 2.2.1 Dispositivo instalado en los parques solares

Se controla mediante un dispositivo todo lo relacionado con contadores, estaciones meteorológicas, etc. teniendo toda la información más compacta.

Sin dicho dispositivo también se puede obtener toda la información requerida por los clientes, siendo varios los elementos a utilizar Pc´s, PLC´s[7], Web Box[8] desembocando en un control descentralizado, un incremento de los gastos en la instalación del sistema y otros factores negativos derivados.

Figura 2.2. Dispositivo utilizado en los parques solares.

### 2.2.2 Datos e informes ofrecidos

- **Generación automática de informes:** Diariamente el sistema genera y manda de forma automática a los clientes un informe en formato PDF con todo detalle de lo ocurrido el día anterior.

- **Cálculo del rendimiento de las instalaciones:** De forma diaria se podrá calcular el rendimiento de cada instalación en todas aquellas que dispongan de radiómetro[9].

- **Comparativas de rendimiento entre instalaciones.**

- **Disponibilidad de histórico de datos.**

- **Comparativas de instalaciones en fechas determinadas y lugares determinados:** Permite evaluar el funcionamiento de una instalación dependiendo la época del año, con el fin de saber donde es más rentable y cuando un parque solar.

---

[7] PLC´s: Se trata de un equipo electrónico que se ha diseñado para programar y controlar procesos secuenciales en tiempo real.

[8] Web Box: Soluciones integrales de desarrollo de proyectos en internet, elaboración de páginas web, E-marketing, etc.

[9] Radiómetro: Instrumento para detectar y medir la intensidad de energía térmica radiante, en especial rayos infrarrojos.

- **Generación de informes de errores.** Cada 15 minutos se genera un informe de rastreo de errores por las instalaciones. En caso de encontrar algo anómalo se reportan de inmediato. No obstante, es configurable según las exigencias del cliente.

- **Generación de informes de prevención de averías:** El sistema valora su propio comportamiento a lo largo de un histórico de datos, y es capaz de adelantarse a averías futuras, según una base de ingeniería del conocimiento. [10]

## 2.3 Clientes

- Inmodo Solar. [11]
- Rentasol. [12]
- Siliken. Energías renovables. [13]
- Ecoo [14]

## 2.4 Objetivos del proyecto

El proyecto se ha surgido y se ha desarrollado basándose en dos cuestiones fundamentales.

La primera se reside en las circunstancias sociales e informáticas actuales. Se desea tener acceso a más información, desde diferentes puntos, más frecuentemente y con mayor exactitud.

El estudio del presente trabajo se ha focalizado en los parques solares. El propietario de estos no queda satisfecho si una vez al mes recibe en su correo un informe indicando la producción, los errores acontecidos, los desperfectos, etc. Necesita conocer en todo momento lo que sucede en su negocio.

Esta es la razón principal por la cual DeltaTec Systems creó una página web, en la que los dueños de las placas, pueden observar y examinar lo que sucede sus instalaciones fotovoltaicas.

El origen del segundo motivo va asociado al primero, ya que existía el anhelo de dar respuesta a una demanda de gran parte de los clientes. La página fue creada para que, desde el ordenador, se pudiesen atender los progresos de forma habitual, pero no obstante, los propietarios demandaban una interfaz móvil, eficiente y usable.

Así pues, se consideró precisa la creación de una aplicación en la que se expusiera la información más significativa en los dispositivos móviles de la mayoría de los clientes.

# 3. Desarrollo

## 3.1 Arquitectura del Sistema

Figura 3.1 Imagen del Sistema.

Consiste en la interacción del dispositivo con el servidor a través de internet.

Esta arquitectura es del tipo cliente-servidor, más concretamente cliente-servidor de tres capas.

A continuación se puede observar una definición de arquitectura cliente-servidor.

Arquitectura formada por un sistema donde el cliente es una máquina que solicita un determinado servicio y se denomina servidor a la máquina que lo proporciona. Los servicios pueden ser, de ejecución de un determinado programa, de acceso a un determinado banco de información y de acceso a un dispositivo de hardware.

Se les denomina cliente-servidor de tres capas porque una capa es de presentación, otra capa de la lógica de la aplicación y otra capa de la base de datos.

Normalmente este tipo de arquitectura se utiliza en las siguientes situaciones:

- Cuando se requiera mucho procesamiento de datos en la aplicación.
- En aplicaciones donde la funcionalidad este en constante cambio.
- Cuando los procesos no están relativamente muy relacionados con los datos.

- Cuando se requiera aislar la tecnología de la base de datos para que sea fácil de cambiar.
- Cuando se requiera separar el código del cliente para que se facilite el mantenimiento.
- Adecuada para utilizarla con tecnología orientada a objetos. [15]

### 3.1.1 Servidor

Dentro de una arquitectura cliente-servidor, hay diferentes tipos de servidores, en este trabajo se utiliza un servidor web.

Básicamente, un servidor web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante peticiones y solicitudes HTTP.

### 3.1.2 Componentes del Software

#### Presentación

Presentación al usuario de un conjunto de objetos visuales y sirve para llevar a cabo el procesamiento de los datos producidos por el mismo y los devueltos por el servidor.

En el sistema, la presentación es el dispositivo móvil.

#### Lógica de aplicación

Capa responsable del procesamiento de la información que tiene lugar en la aplicación.

Lógica de la aplicación es el servidor con el que interactuamos.

#### Base de datos

Compuesta de los archivos que contienen los datos de la aplicación.

## 3.2 Protocolo HTTP

El propósito del protocolo HTTP es permitir la transferencia de archivos entre un navegador (el cliente) y un servidor web localizado mediante una cadena de caracteres denominada dirección URL.

La comunicación entre cliente-servidor es mediante solicitud y respuesta de HTTP. [16]

### 3.2.1 Solicitudes HTTP

Conjunto de líneas que el navegador envía al servidor. Consta de:

- **Línea de solicitud:** Especifica el tipo de documento solicitado, el método que se aplicará y la versión del protocolo utilizada.

- **Campos del encabezado de solicitud:** Conjunto de líneas opcionales que permiten aportar información adicional sobre la solicitud (navegador, sistema operativo, etc.). Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos y el valor del encabezado. [16]

```
MÉTODO VERSIÓN URL<crlf>
ENCABEZADO: Valor<crlf>
. . . ENCABEZADO: Valor<crlf>
Línea en blanco <crlf>
CUERPO DE LA SOLICITUD
```

```
GET http://es.kioskea.net HTTP/1.0 Accept : Text/html If-Modified-Since :
Saturday, 15-January-2000 14:37:11 GMT User-Agent : Mozilla/4.0 (compatible;
MSIE 5.0; Windows 95)
```

Figura 3.2 Ejemplo de una solicitud HTTP.

- **Cuerpo de la solicitud:** Conjunto de líneas opcionales que deben estar separadas de las líneas precedentes por una línea en blanco y, por ejemplo, permiten que se envíen datos por un comando POST durante la transmisión de datos al servidor utilizando un formulario.

### 3.2.2 Respuesta HTTP

Conjunto de líneas que el servidor envía al navegador. Está constituida por:

- **Línea de estado**: Especifica la versión del protocolo utilizada y el estado de la solicitud en proceso mediante un texto explicativo y un código.

- **Campos del encabezado de respuesta**: Conjunto de líneas opcionales que permiten aportar información adicional sobre la respuesta. Cada una de estas líneas está compuesta por un nombre que califica el tipo de encabezado, seguido por dos puntos y por el valor del encabezado.

- **Cuerpo de la respuesta**: Contiene el documento solicitado. [16]

A continuación se puede observar la sintaxis de una respuesta HTTP, y un ejemplo.

```
VERSIÓN-HTTP CÓDIGO EXPLICACIÓN <crlf>
ENCABEZADO: Valor<crlf>
. . . ENCABEZADO: Valor<crlf>
Línea en blanco <crlf>
CUERPO DE LA RESPUESTA
```

```
HTTP/1.0 200 OK Date: Sat, 15 Jan 2000 14:37:12 GMT Server : Microsoft-
IIS/2.0 Content-Type : text/HTML Content-Length : 1245 Last-Modified : Fri,
14 Jan 2000 08:25:13 GMT
```

Figura 3.3 Ejemplo de una respuesta HTTP.

## 3.3 Especificación del Sistema

Como ha sido comentado con anterioridad, el sistema presente en este proyecto ofrece información acerca de la producción de los parques solares y del clima de estos. Así como enviar notificaciones al correo del administrador, para verificar algún dato, o consultar alguna duda.

A nivel de infraestructura, el sistema está formando por diferentes dispositivos móviles, los cuales, una vez que el usuario inicia la aplicación, esta está constantemente informando de la producción diaria, mensual y de la temperatura. Esta última se toma desde Google, aunque la producción diaria y mensual se encuentra desplegada sobre un servidor web en internet, almacenando esa información en una base de datos.

Para mostrar una gráfica con los meses y los diferentes niveles de producción también existe relación con Google, el proceso es el siguiente: Se mandan los datos que quieren visualizar en formato gráfica y Google devuelve dicha gráfica elaborada.

En la Figura 3.1 se muestra un esquema de la organización del sistema así como la manera en la que se relacionan los dispositivos físicos que lo forman.

A nivel de aplicación, el sistema está formado, básicamente por dos módulos: El dispositivo móvil y el servidor web junto con la base de datos.

El dispositivo instala la aplicación, capaz de visualizar los datos, a través de la cual el usuario interactúa con el sistema. El usuario del dispositivo debe autenticarse al iniciar la aplicación para acceder al sistema, una vez autenticado, ya podrá seleccionar lo que desee visualizar.

Desde el celular el usuario puede realizar peticiones al servidor, producción diaria o mensual, clima en el momento actual en la planta seleccionada. Con la información solicitada el dispositivo pasa a mostrarla.

El servidor atiende las peticiones que envían los dispositivos mediante http, interacciona con la base de datos y devuelve una respuesta http al dispositivo.

El servidor contiene también una base de datos en la que almacena la siguiente información:

- Datos del administrador con su nombre y password.
- Información sobre los usuarios y dispositivos registrados en el sistema.
- Selección de las diferentes plantas solares de casa usuario.
- Información acerca de cada parque solar, como puede ser: Situación geográfica, producción diaria, producción mensual, histórico de datos.

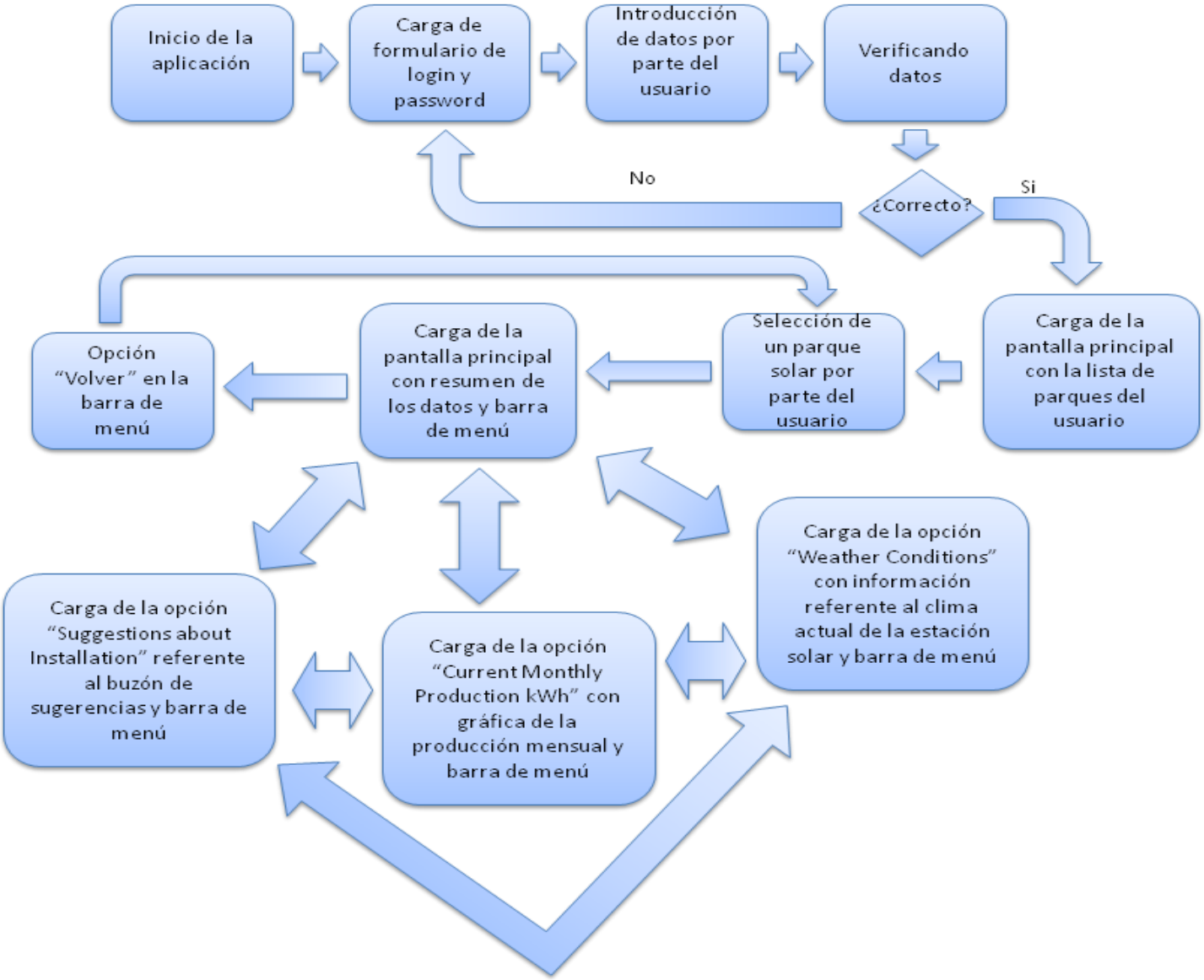


Figura 3.4 Funcionamiento del Sistema.

## 3.4 Requisitos

Para que la aplicación pueda ejecutarse en un dispositivo BlackBerry, este debe cumplir la siguiente tabla:

| REQUISITOS | DESCRIPCIÓN |
|---|---|
| ***Conexión a Internet*** | Acceso a internet para poder interactuar con el servidor. |
| ***CLDC 1.1*** | Configuración que ofrece el API básico para programar dispositivos móviles. |
| ***MIDP 2.0*** | Perfil que ofrece el conjunto de APIs de Java especializadas para programas dispositivos móviles. Ofrece, por ejemplo, paquetes que permiten creación de interfaces gráficas o soporte de red. |
| ***SO 6.0 de BlackBerry*** | Ultima versión actual del sistema operativo para dispositivos BlackBerry. |

Tabla 3.1 Requisitos para ejecutar la aplicación en un teléfono móvil.

## 3.5 Diseño de Casos de uso

A continuación se incluye un diagrama de casos de uso donde se observa, gráficamente, las funciones que debe cumplir el sistema.

Es importante fijar desde un principio la funcionalidad de la que se quiere dotar al sistema, un cambio a este nivel no proporciona mucho cambio en el sistema, sin embargo un cambio a un nivel más adelantado supondría reestructurar de nuevo el sistema. Debido a esto es importante no equivocarse, y estructurar bien el sistema.

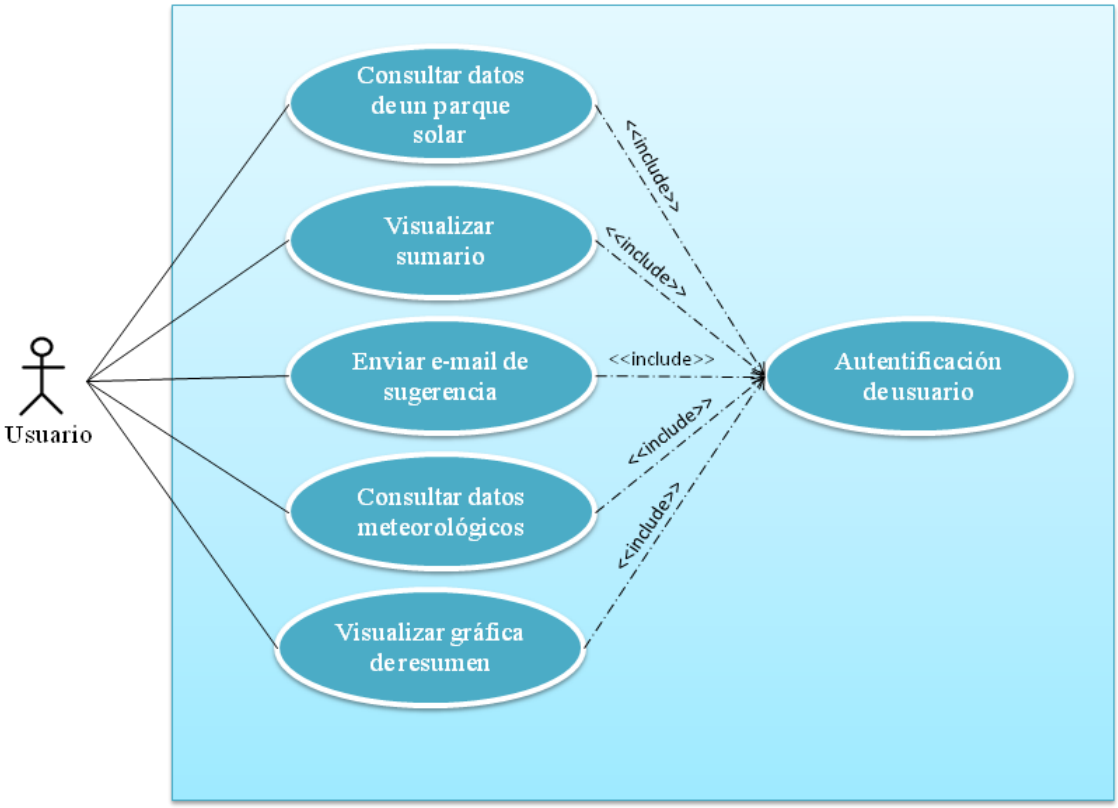


Figura 3.5 Diagrama de casos de uso del sistema.

### 3.5.1 Actor

El actor que interactúa con el sistema es el usuario que tiene instalada la aplicación en su dispositivo móvil, esta le permite consultar los datos de producción que corresponden a su conjunto de placas solares de las diferentes instalaciones que tenga.

Del mismo modo podrá visualizar la situación atmosférica en el parque seleccionado, también cuenta con la opción de enviar un e-mail al buzón de sugerencias retransmitiendo una queja, un error, etc.

Estas opciones se ofrecen al usuario una vez se haya autenticado en el sistema.

### 3.5.2 Interfaz de usuario

Al iniciar la aplicación, el dispositivo le presenta al usuario la siguiente interfaz:

- Pantalla de login: A través de esta el usuario se autentica en el sistema, especificando login y password, asignado por la empresa.
- Pantalla principal: El usuario selecciona la instalación, la cual quiere obtener información acerca de la producción.
    - Sumario: El usuario obtiene una lista con la siguiente información.
        - Producción de ayer.
        - Producción del mes actual.

        - Cantidad de euros obtenidos ayer.
        - Cantidad de euros recaudados este mes.
        - Cantidad de $CO_2$.
    - Condiciones atmosféricas: Mediante esta función el usuario obtiene información acerca del clima actual en la instalación seleccionada. El usuario visualiza:
        - Temperatura.
        - Porcentaje de humedad.
        - Información eólica.
        - Estado climático.
    - Producción mensual actual: Diagrama de barras en el cual el usuario obtiene información acerca de la producción mensual actual, y acerca de la diferencia con la producción mensual de anteriores meses.
    - Sugerencia acerca de la instalación: El usuario puede informar directamente de un error, o preguntar alguna duda que le haya podido surgir enviando un e-mail.

## 3.6 Análisis

En este apartado se describe el sistema en términos de interacción entre objetos del mismo.

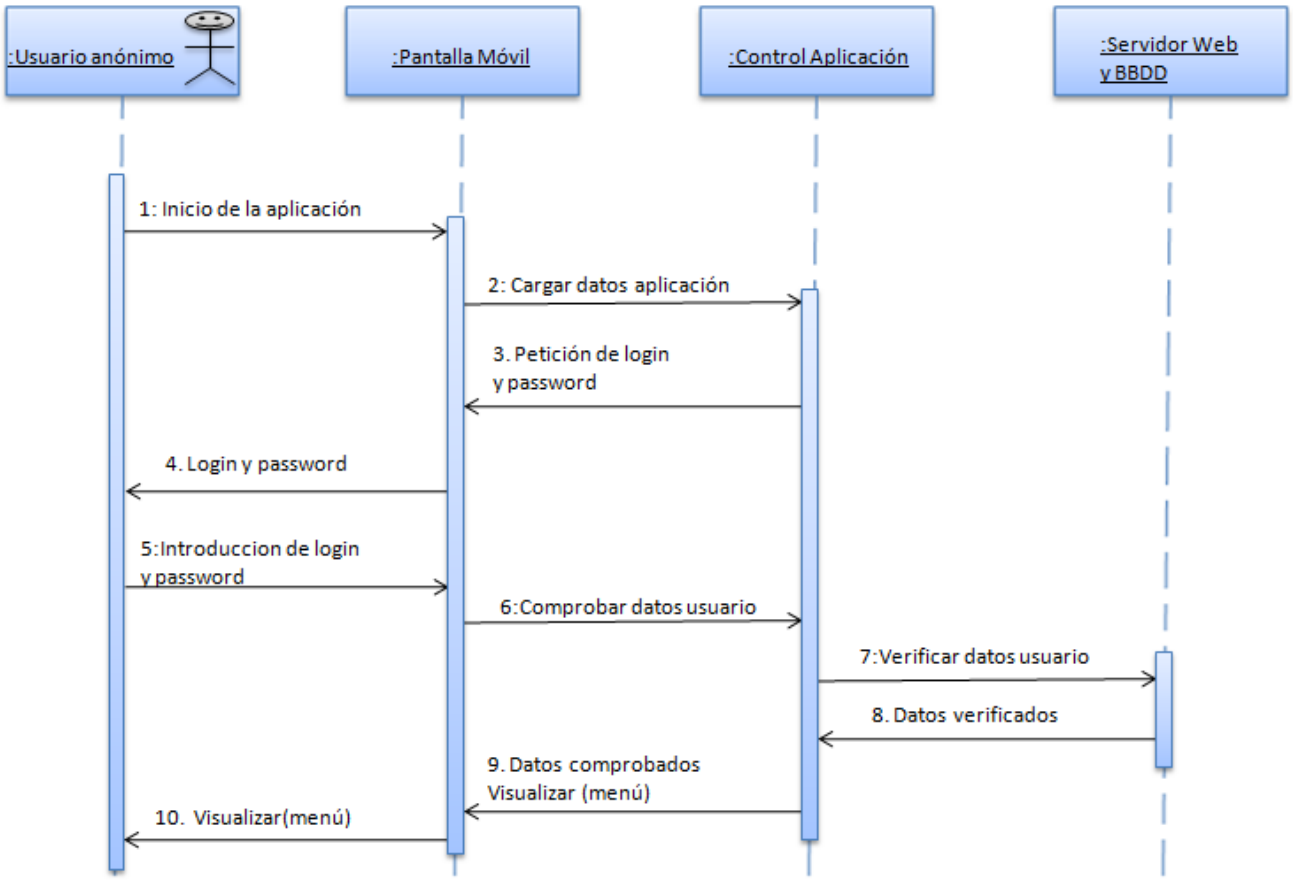


Figura 3.6 Diagrama de secuencia de "Autenticación Usuario".

Solo se describen los caminos básicos, aunque aquí quedan dichos y se dan por entendidos algunos caminos alternativos. Uno de estos puede ser el siguiente.

Pasar desde una función a otra, como por ejemplo, estar visualizando una gráfica y pasar a ver la producción diaria, simplemente pulsando un enlace, otro camino alternativo podría ser la visualización de mensajes de error, porque se produzca un error al interactuar con Google, etc.

### Diagrama de secuencia del caso de uso "Autentificación de usuario"

En la Figura 3.6 se muestra un diagrama de secuencia del caso de uso "Autentificación de usuario".

Este diagrama muestra como el sistema verifica al usuario en la base de datos, y da lugar al siguiente paso en la aplicación, ver el menú principal que consta de una lista con los nombres de los parques solares en los que el propietario es el usuario verificado.

El diagrama podría tener varios caminos alternativos, el más destacable es cuando el servidor no verifica los datos del usuario, en ese caso el usuario vería por pantalla el siguiente mensaje, "Invalid user and/or pass".

En los siguientes casos de uso, se supondrá que el usuario está verificado por el servidor y la base de datos.

### Diagrama de secuencia del caso de uso "Consultar datos de un parque solar"

El siguiente caso de uso, tiene como inicio, una variante de la Figura 3.6, en donde realmente no es del todo correcta.

Cuando un usuario es verificado por el servidor, la gestión de la aplicación vuelve a preguntar al servidor el nombre de los parques del usuario en cuestión, devolviéndole este el nombre de los parques para que los muestre por pantalla al usuario.

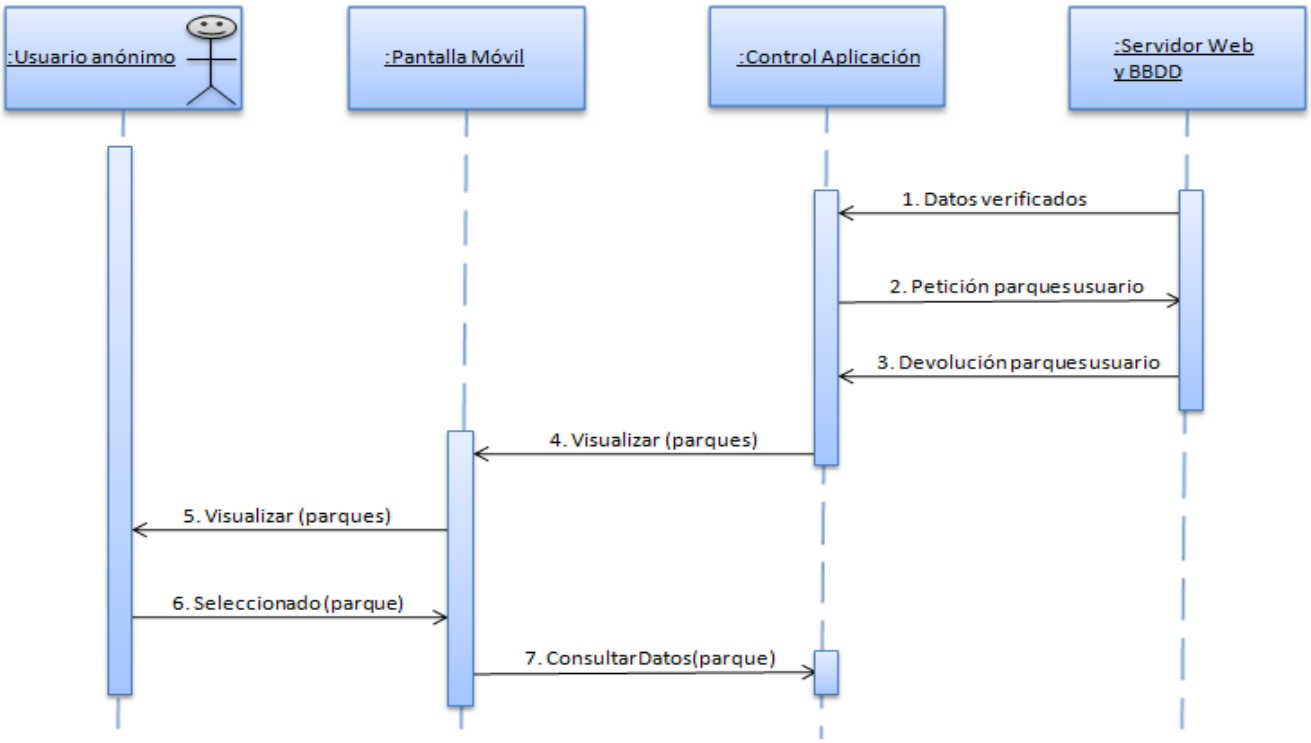


Figura 3.7 Diagrama de secuencia del caso de uso "Consultar datos de un parque solar".

En la anterior figura, se toma como punto de partida, el instante en el que el servidor proporciona a control la verificación de los datos.

### Diagrama de secuencia del caso de uso "Visualizar sumario"

El caso de uso "Visualizar sumario" comienza en el momento que finaliza el caso de uso "Autentificación de usuario", pertenecería al paso número 8 en la Figura 3.7, aunque queda reflejado de manera diferente en la próxima figura, debido a que es el primer paso en este caso de uso.

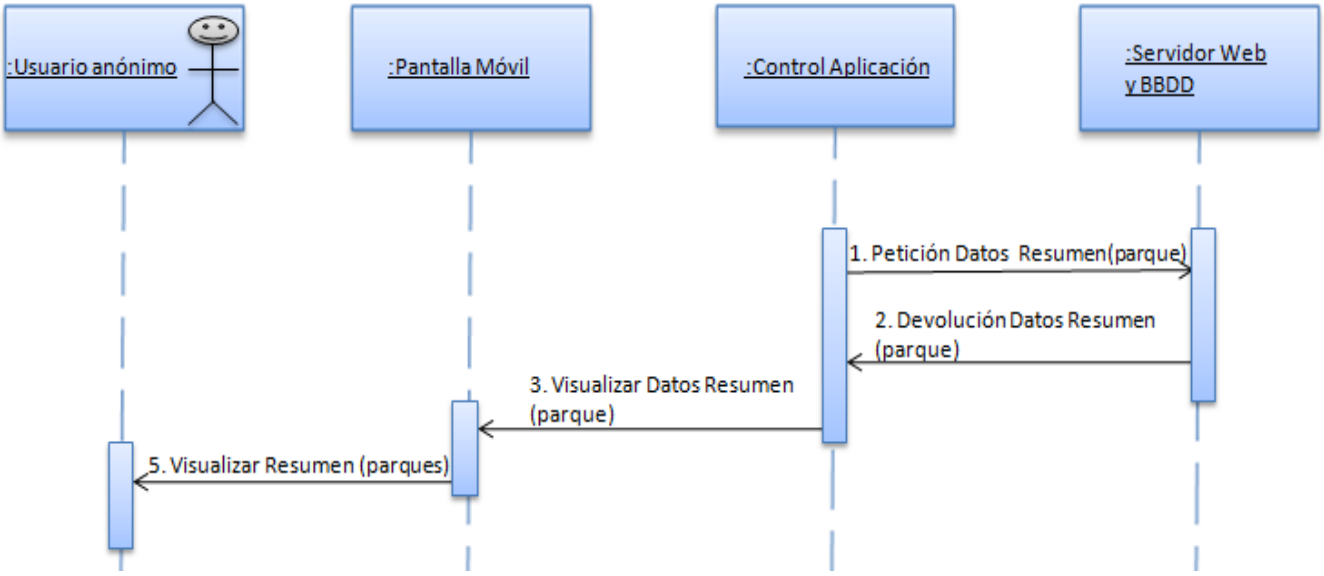


Figura 3.8 Diagrama de secuencia del caso de uso "Visualizar sumario".

En la anterior figura se aprecia como el control del sistema pide al servidor web y a la base de datos los datos de resumen de un parque solar en concreto, y una vez devueltos estos datos, hace que se visualicen por pantalla de una forma predeterminada por la aplicación.

Se ha mostrado una ejecución principal de este caso de uso, en el que se toma como inicio el final del anterior caso de uso "Consultar datos de un parque solar". También se podría iniciar este caso de uso cuando el usuario presiona el botón de activación del resumen de la producción. En donde el inicio sería similar al de la siguiente figura, un ejemplo de este comienzo podría ser: "1. Botón Usuario Activado".

### Diagrama de secuencia del caso de uso "Enviar e-mail de sugerencia"

A continuación se observa representado el caso de uso "Enviar e-mail de sugerencia", en el cual el sistema una vez que han pulsado el botón para enviar e-mail, activa la interfaz necesaria para que el usuario pueda escribir dicho e-mail, y acto seguido pueda enviarlo.

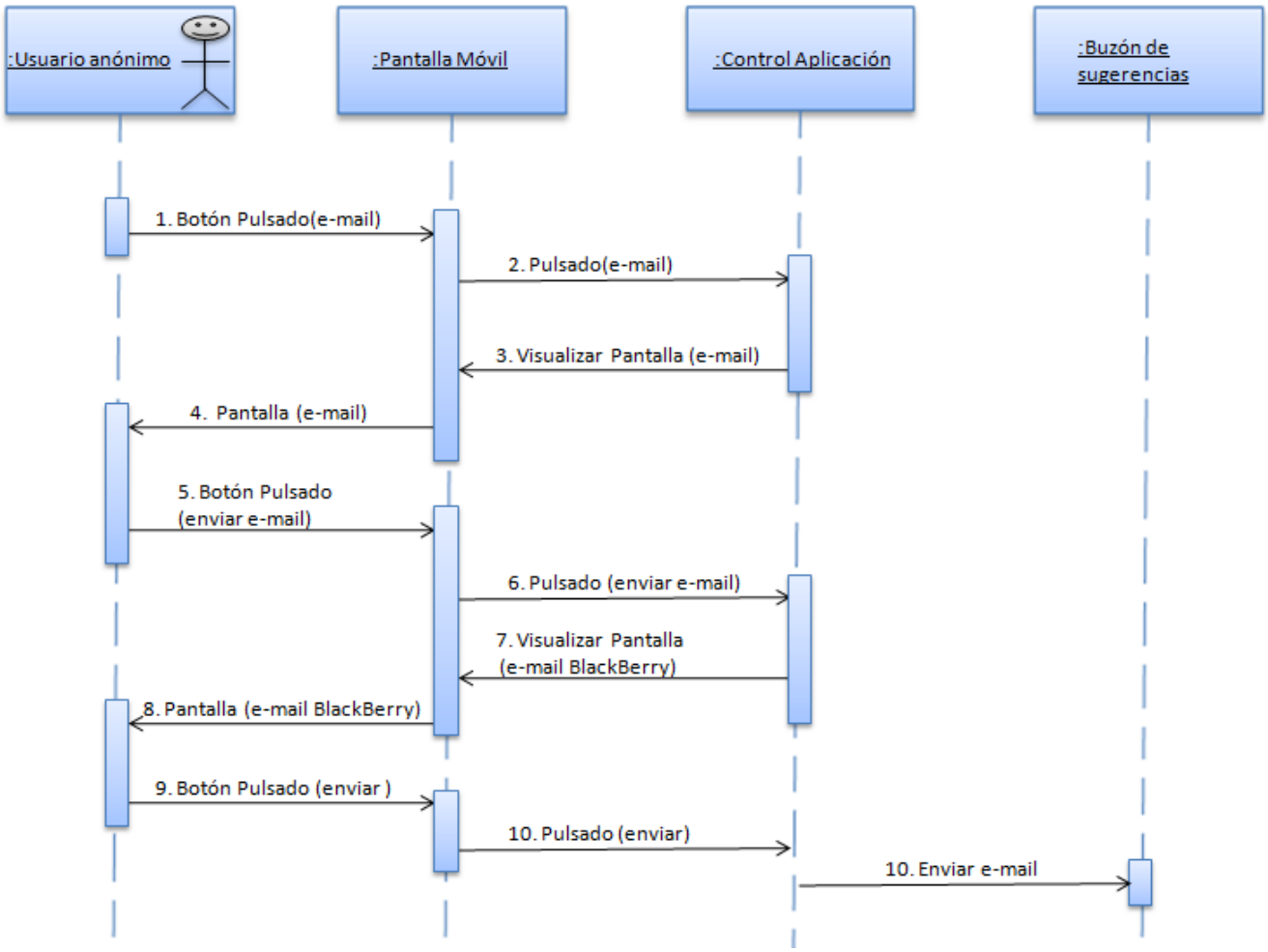


Figura 3.9 Diagrama de secuencia del caso de uso "Enviar e-mail sugerencia".

Cabe destacar una funcionalidad, que queda reflejada en la figura pero que si no se explica puede que no se entienda. En el punto 6 el sistema podría mandar directamente el e-mail, pero lo que hace es activar la aplicación de los operadores BlackBerry para mandar e-mails, y así facilita al usuario el hecho de cambiar el asunto al correo, introducir datos adjuntos, introducir más direcciones a las cuales quiere hacer llegar el correo, etc.

Una vez finalizado este proceso, cuando puse "Enviar", este se enviará definitivamente.

### Diagrama de secuencia del caso de uso "Consultar datos meteorológicos"

La siguiente figura tiene una diferencia con respecto a los diagramas de secuencia ya vistos en este proyecto. Selecciona la zona en la cual se sitúa el parque solar, y pregunta a Google el clima actual que se tiene en dicha zona. Hasta ahora todo se hacía dentro de mi sistema, se preguntaba al servidor y a la base de datos, pero no se pedía nada por internet, por lo tanto se sobreentiende que nuestro dispositivo móvil tiene acceso a internet móvil, y se puede realizar dicha petición a Google.

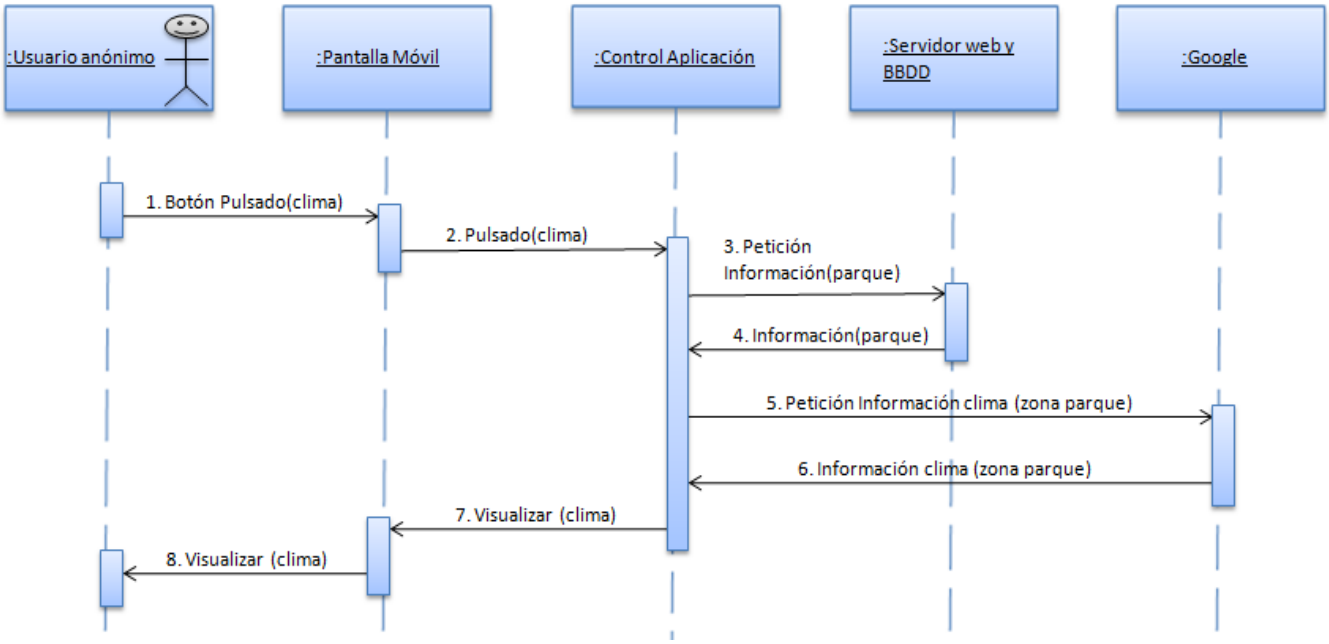


Figura 3.10 Diagrama de secuencia del caso de uso "Consultar datos meteorológicos".

Un camino alternativo en el anterior diagrama de secuencia puede ser cuando no tenga éxito la conexión con Google, en el cual la aplicación hará visualizar un mensaje por pantalla, en el que se haga saber que hay un error. En este error se visualizará el siguiente texto: "Error en Menu3".

### Diagrama de secuencia del caso de uso "Visualizar gráfica de resumen"

El próximo diagrama de secuencia es el último que se describe, como ya se ha dicho antes solo se va a ver los caminos base, aunque se mencionen algunos caminos alternativos.

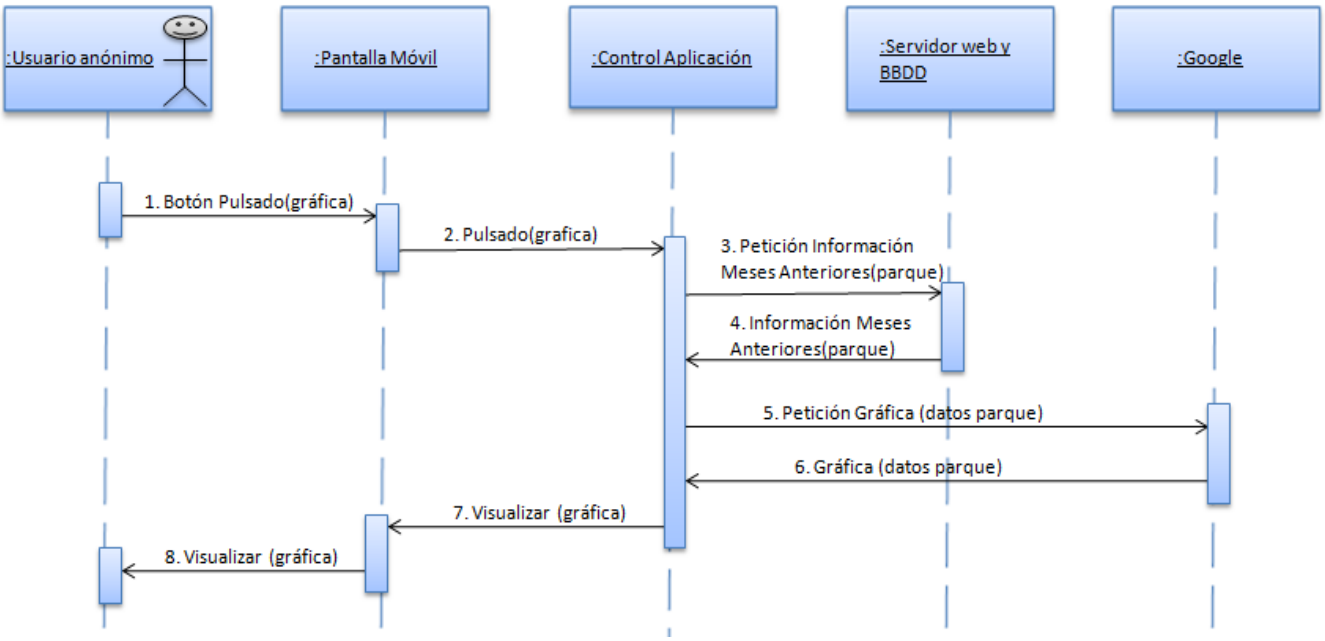


Figura 3.11 Diagrama de secuencia del caso de uso "Visualizar gráfica de resumen".

Se puede observar que la Figura 3.11 y la Figura 3.10 se basan en lo mismo por lo que el camino alternativo descrito anteriormente es idéntico en ambos casos de uso.

## 3.7 Proceso

A continuación se describen los pasos a seguir para empezar a implementar, y poder ejecutar el programa correctamente.

1. **Descarga de eclipse.**

La descarga de eclipse se hace a través de la página oficial de eclipse dedicada a estas, http://www.eclipse.org/downloads/, se señaliza la versión que se desea utilizar y seguidamente se indica el país desde el cual se descarga.

2. **Descarga del plug-in.**

La descarga de dicho plug-in implica darse de alta en BlackBerry Developer Zone, dentro de la página oficial de BlackBerry, cuando se activa la cuenta se recibe un correo en el que está incluida una guía de información, recursos importantes, y un correo electrónico que se puede utilizar para contactar con expertos a la hora de resolver dudas y ofrecer facilidades. Aparte de la guía, en la web de BlackBerry hay tutoriales con diversidad de contenidos como por ejemplo, personalizar la aplicación (interfaz usuario, almacenar datos, etc.) y como distribuir la aplicación.

3. **Búsqueda e instalación de API.**

Existe un API que se encarga de elaborar los diagramas de barras que se necesitan en una de las opciones de la aplicación.

Esta debe estar insertada en el plug-in de BlackBerry, se denomina Google Chart. El inconveniente de esta API es el siguiente: no sirve para BlackBerry, por lo que hay que buscar una que esté vinculada a estos dispositivos, llamada BlackBerry Google Chart API.

Una vez descargada se siguen los siguientes pasos para instalarla:

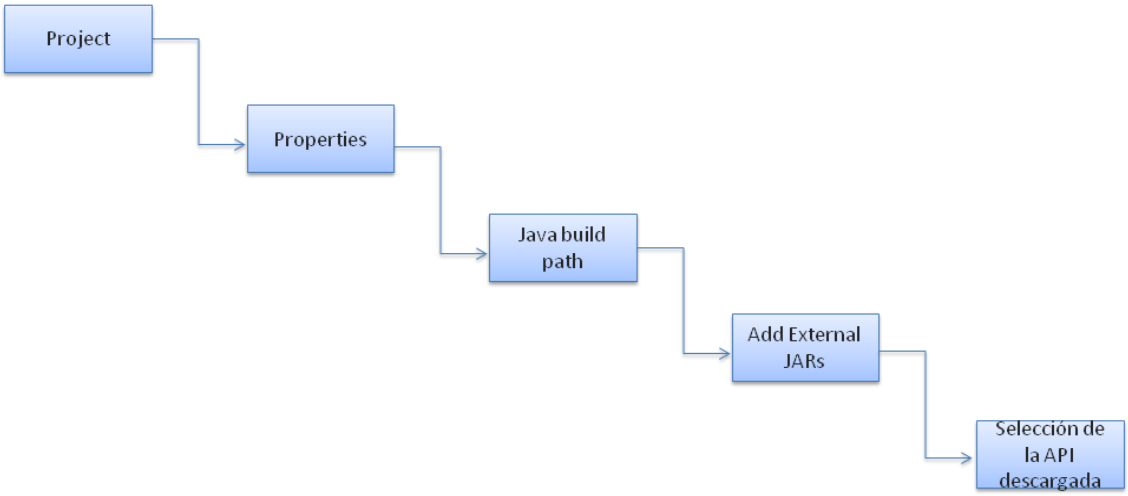


Figura 3.12 Esquema para introducir el API en el plug-in.

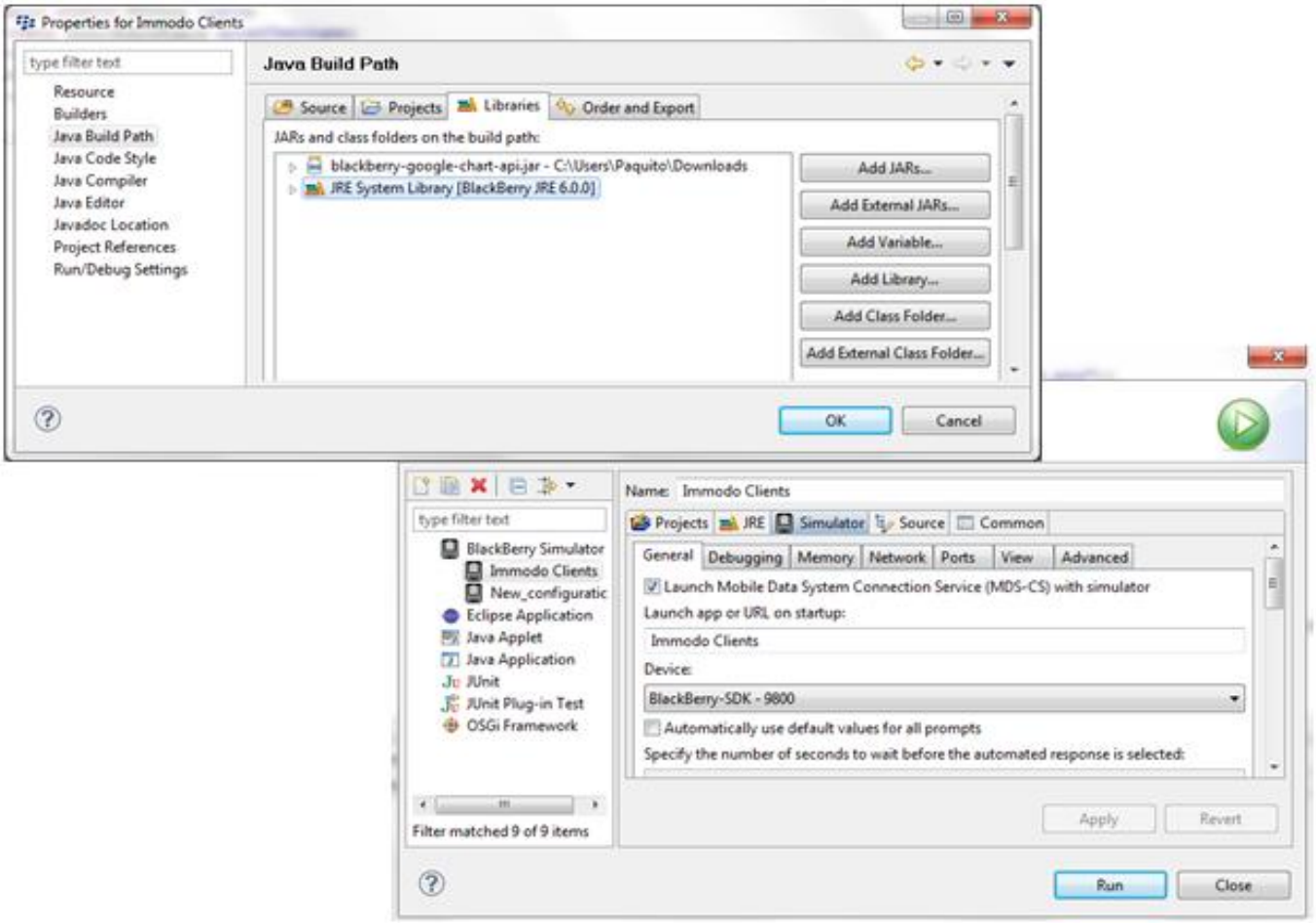


Figura 3.13 Selección de los elementos a configurar.

### 4. Internet en el simulador.

El simulador por defecto no tiene conexión a internet. Dentro del plug-in hay un vínculo desde el cual se da acceso a internet para dicho simulador, para ello se siguen los siguientes pasos.

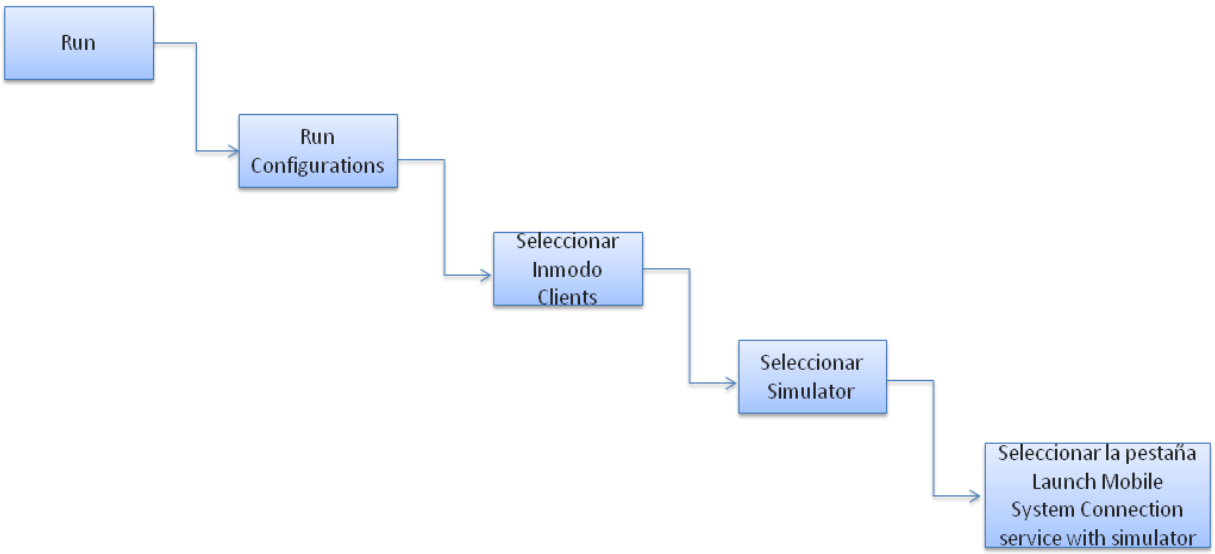


Figura 3.14 Esquema para seleccionar internet en el simulador.

### 5. Introducción de imágenes y descarga de código ya escrito.

El poder ver enlaces con imágenes, símbolos o iconos como por ejemplo un e-mail, un sol, etc. implica el siguiente hecho: se ha creado una carpeta a la hora de implementar con dichas imágenes. Esta se ha llamado res, en la que se han introducido todas las imágenes que luego se utilizará llamando a addIcon, o a getBitmapResource, junto con unos paréntesis y comillas con el nombre de la imagen al completo.

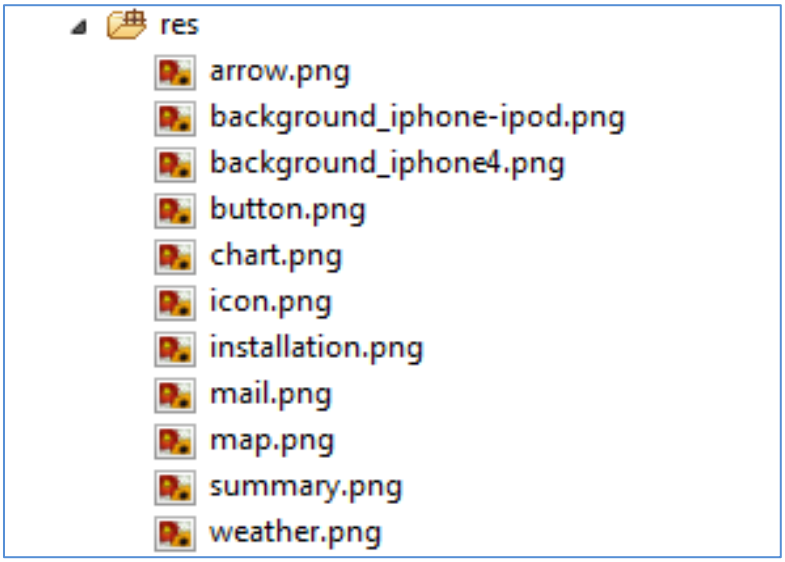


Figura 3.15 Visualización de la carpeta “res”.

A continuación se puede observar un ejemplo: addIcon(“icon.png”).

También destacar la descarga de código ya escrito que circula por internet, como pueden ser las clases ProgressBar.java, ScrollableImageField.java y TextBoxField.java, estas se explicarán más adelante.

Todas estas clases se pueden encontrar en el siguiente foro de Blackberry, http://www.supportforums.blackberry.com.

### 6. Subir la aplicación al AppWorld.

El último paso se realiza es subir la aplicación al AppWorld, la tienda de aplicaciones de BlackBerry, aunque antes de este paso se debe escribir el código que será explicado más adelante.

Para subir la aplicación lo primero que hay que se hace es darse de alta en RIM, como desarrollador. Una vez se ha dado de alta, desde RIM piden que se ingrese una cantidad de dinero estipulado en su cuenta corriente, para gastos de gestión.

El siguiente paso se dispone como el más complejo, porque se tiene que enviar un fax a RIM acreditando que la persona que está detrás del ordenador es la misma persona que está intentando subir la aplicación, esta acreditación deberá llevar DNI, y deberá estar firmada ante notario, esto se hace para evitar datos falsos.

Por último una vez que se han finalizado y se han corroborado los pasos anteriores, desde RIM, mandan un correo electrónico con tres firmas electrónicas que se tienen que agregar al plug-in, la finalidad de las firmas es conseguir que el código pueda ejecutarse de manera correcta en los terminales BlackBerry. Una vez introducidas estas firmas se manda la aplicación entera y en un pequeño plazo la aplicación estará lista para poder ser descargada desde el AppWorld.

## 3.8 Diagrama de clases

A continuación se puede observar los diagramas de clases en notación UML, de lo que es todo el código escrito y de todas las clases que interactúan con el sistema. Estos diagramas se realizan por separado para facilitar la comprensión.

### 3.8.1 InmodoClients

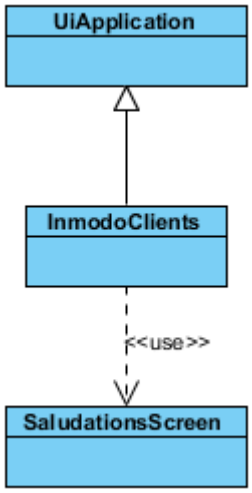


Figura 3.16 Diagrama de clases de la clase InmodoClients.

InmodoClients viene a ser el menú principal de la aplicación, como el constructor de esta, motivo por el cual hereda de UiApplication, toda interfaz de BlackBerry tiene que heredar de esta clase. También tiene una relación de uso con SaludationsScreen ya que esta es la clase que da lugar a la generación de la pantalla principal.

### 3.8.2 SaludationsScreen

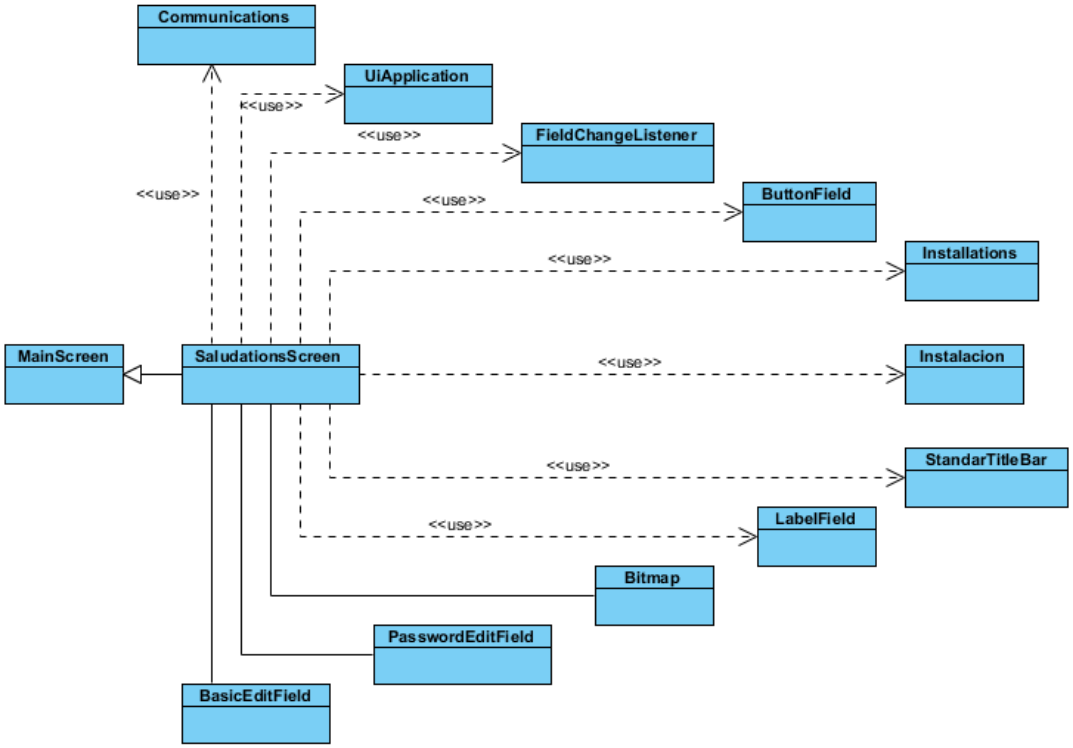


Figura 3.17 Diagrama de clases de la clase SaludationsScreen.

Clase encargada de representar una pantalla en el dispositivo móvil desde el cual se inicia la aplicación, para ello la clase hereda de MainScreen.

La relación entre SaludationScreen y las clases Communications, BasicEditField y PasswordEditField es la petición y corroboración del nombre y la contraseña de usuario.

Installations representa el nombre de la pantalla la cual una vez el usuario ha sido verificado, se invoca. En clases como StandarTitleBar, LabelField, etc. la relación se encuentra en la estética de la pantalla o aplicación.

### 3.8.3 Communications

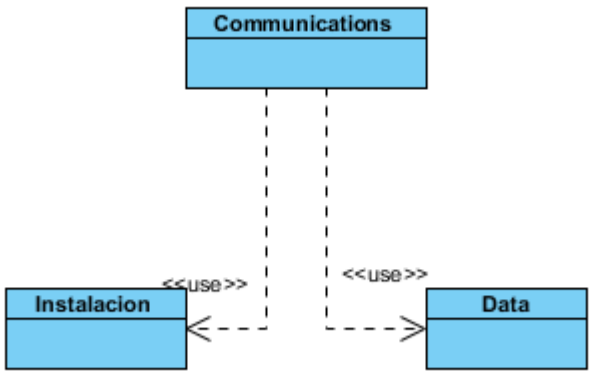


Figura 3.18 Diagrama de clases de la clase Communications.

La clase Communications ejerce una relación de uso con las dos clases creadas para transmitir información, ya que esta clase se utiliza para comunicarse con el servidor web, a la hora de obtener datos de los parques e instalaciones.

### 3.8.4 Installations

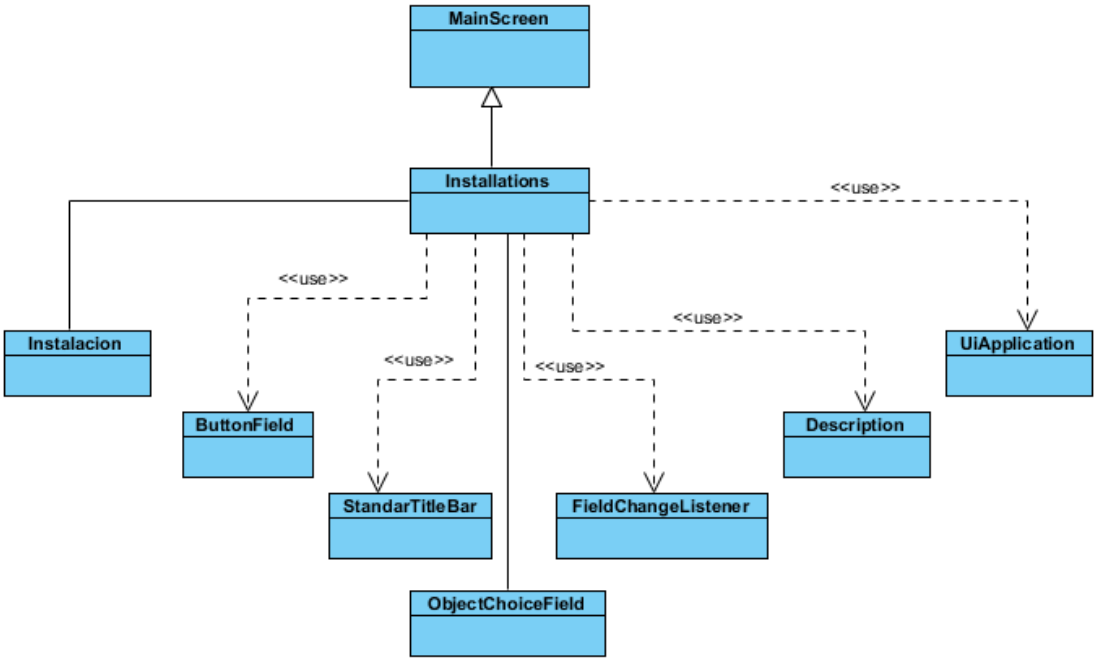


Figura 3.19 Diagrama de clases de la clase Installations.

Como ya se ha citado en clases anteriores, la clase StandarTitleBar se relaciona con Installations por motivos estéticos.

La pantalla representada por Installations, consta de una lista de instalaciones y un botón para seleccionar un objeto de la lista, de estas herramientas se crean las relaciones con las clases ObjectChoiceField, ButtonField y FieldChangeListener. La

siguiente pantalla a mostrar por parte de la aplicación una vez que se ha seleccionado la instalación está representada en la clase Description.

### 3.8.5 TextBoxField

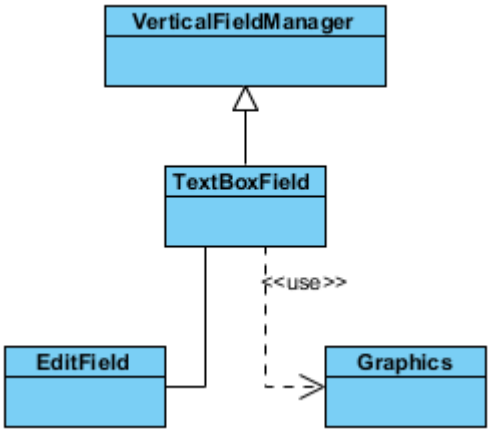


Figura 3.20 Diagrama de clases de la clase TextBoxField.

Editar un campo de texto es el motivo por el que se usa esta clase. La explicación de TextBoxField, se obtiene más adelante.

### 3.8.6 ProgressBar

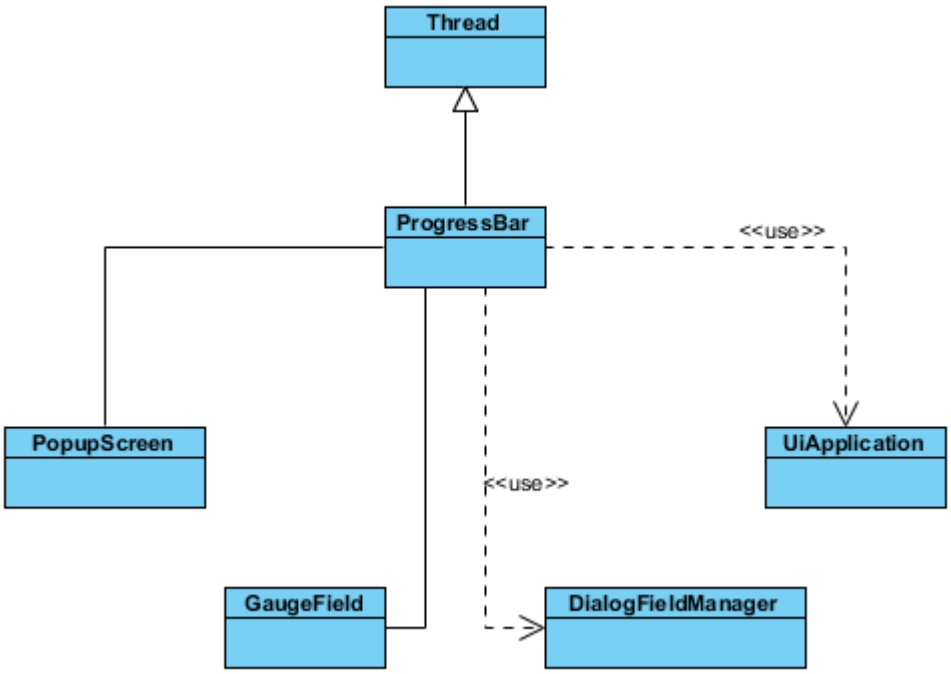


Figura 3.21 Diagrama de clases de la clase ProgressBar.

Clase encargada de proporcionar una pantalla de espera mientras la aplicación piensa lo que se mostrará a continuación.

### 3.8.7 ScrollableImageField

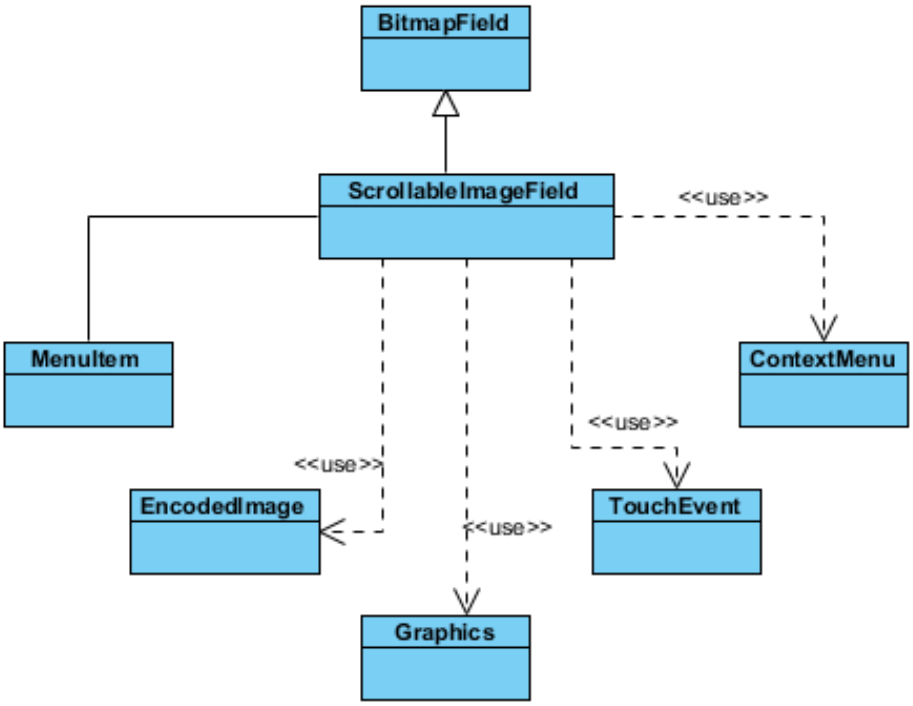


Figura 3.22 Diagrama de clases de la clase ScrollableImageField.

ScrollableImageField se usa en el proyecto para modificar y ajustar el tamaño de una imagen a cualquier dispositivo móvil.

### 3.8.8 Description

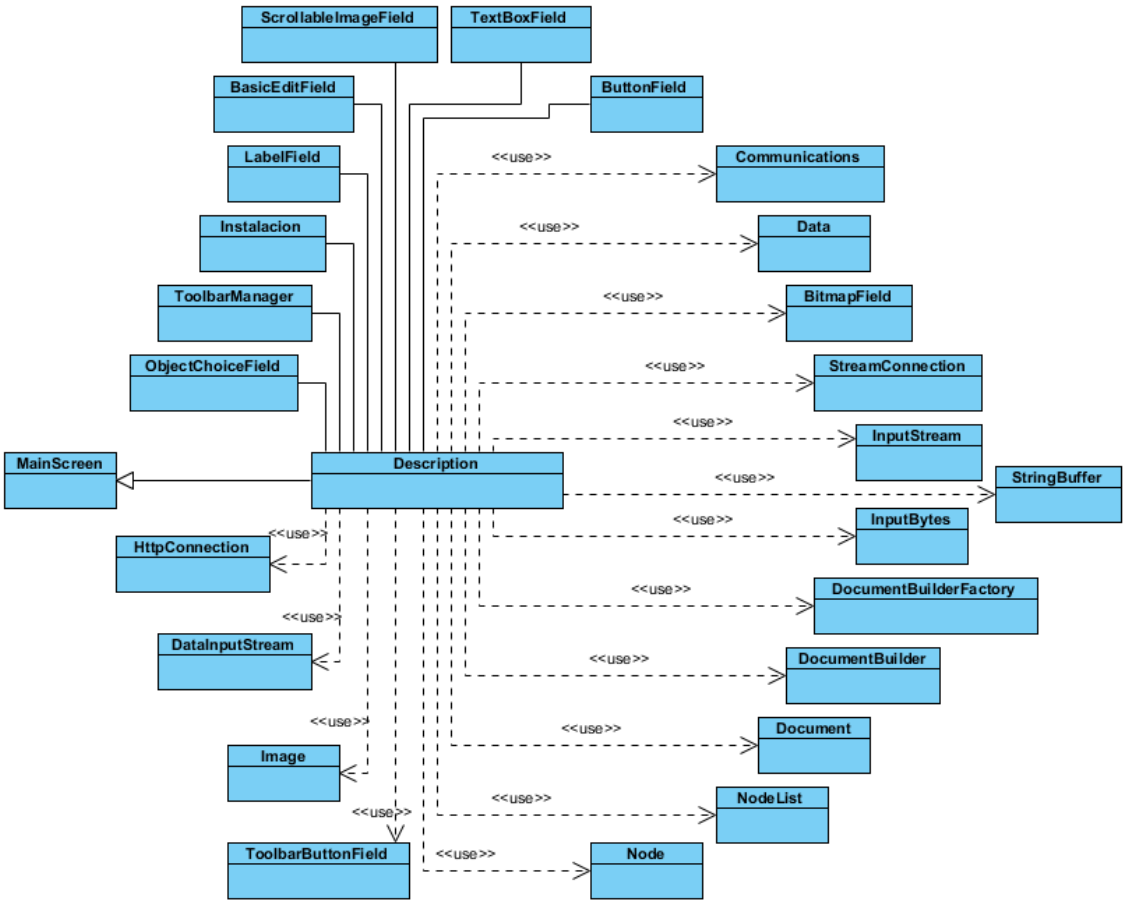


Figura 3.23 Diagrama de clases de la clase Description.

Destacar las clases utilizadas para la conexión y extracción de información atmosférica y de la gráfica existente en un menú que más adelante será explicado, estas clases son: StreamConnection, InputBytes, StringBuffer, InputStream, DocumentBuilderFactory, DocumentBuilder, Document, NodeList, Node, HttpConnection, DataInputStream, BitmapField e Image.

A continuación se explica de manera más explícita para que sirven las clases y porque se ha decidido usar ciertas clases en el trabajo.

# 4. Implementación

En esta sección, se describen brevemente las clases que se han implementado para poder hacer funcionar la aplicación en los dispositivos móviles adecuados.

## 4.1 Clase InmodoClient.java

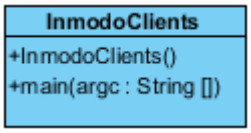


Figura 4.1 Esquema de la clase InmodoClient.

Clase principal de la aplicación, por eso hereda de UiApplication, clase base para todas las aplicaciones de dispositivos que proporcionan una interfaz de usuario.

Una aplicación que proporciona una interfaz de usuario conlleva una pila de pantallas, en los que el objetivo de estas es mostrarse por pantalla, de las que solamente la pantalla que esté en ejecución puede recibir eventos de entrada. Cabe destacar que cada pantalla solo puede aparecer una vez en la pila, si no se generará una excepción.

En la aplicación se inicializa esta pila de pantallas, en el método InmodoClients, se crea una pantalla del tipo SaludationsScreen y se apila.

## 4.2 Clase SaludationsScreen.java

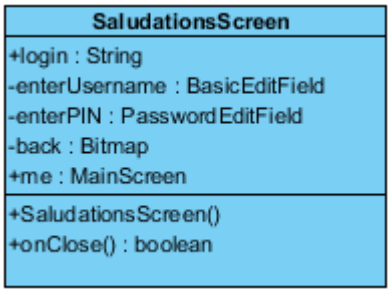


Figura 4.2 Esquema de la clase SaludationScreen.

Para explicar esta clase, se comienza dando una breve explicación de las clases que interactúan en ella.

- BasicEditField: Un campo de texto editable. [17]
- PasswordEditField: Hereda del anterior, y es un campo de texto editable diseñado específicamente para manejar entrada de contraseñas. [17]
- Bitmap: Sirve para mostrar imágenes, que el usuario posteriormente podrá visualizar. [17]

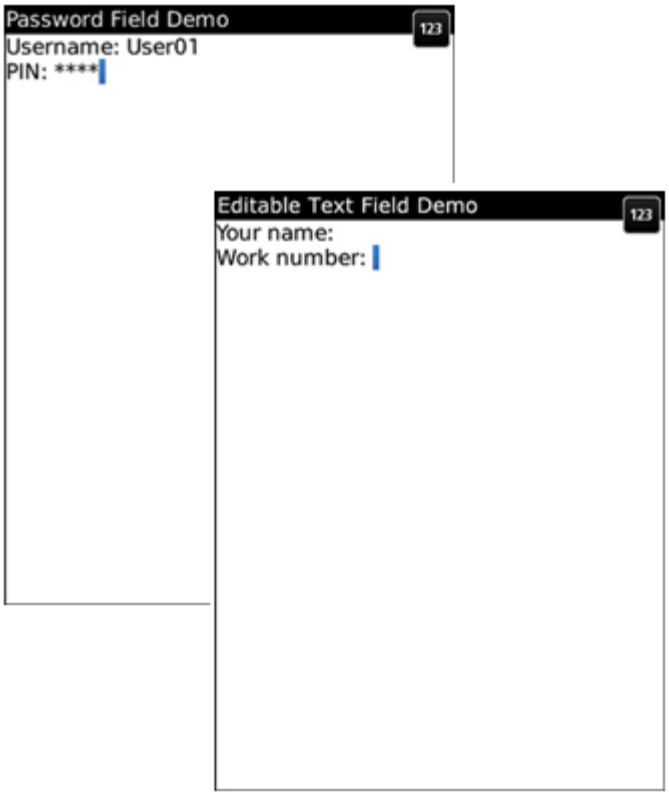


Figura 4.3 Diferencia entre BasicEditField y PasswordEditField. [18]

- MainScreen: Es la pantalla donde se visualiza todo. Contiene un título, un separador y una sección a desarrollar.
- LabelField: Etiqueta utilizada para identificar un componente de la interfaz de usuario. [19]

Figura 4.4 Ejemplo de LabelField. [18]

- StandardTitleBar: Se utiliza para mostrar información a los usuarios en la parte superior de la pantalla, en esta se puede introducir un icono, un título, la hora, la cobertura, etc. [20]

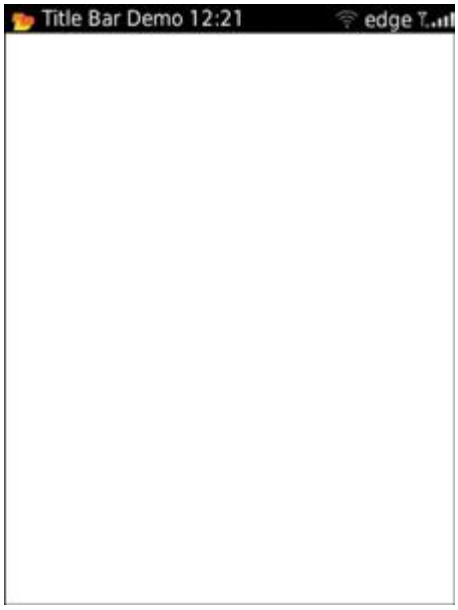


Figura 4.5 Ejemplo de StandardTitleBar. [18]

- ButtonField: Es la manera de interaccionar con el usuario, cuando el usuario pulsa este botón, la aplicación pasará a realizar alguna acción. [17]

Figura 4.6 Ejemplo de ButtonField. [18]

- FieldChangeListener: Se implementa cuando se desea realizar un cambio de pantalla. Para controlar los cambios se implementa el método setChangeListener. [21]

La clase SaludationScreen se utiliza para crear la pantalla de bienvenida de la aplicación, en la que se pide al usuario que se identifique, y se verifica la identificación.

Primeramente se ha creado la pantalla MainScreen, con el título "Inmodo clients", y con un campo de texto destinado al nombre de usuario y un campo de texto de contraseña dedicada a esta. También se ha insertado un botón, para cuando este sea pulsado verificar los datos, y comprobar que instalaciones tiene este usuario, si llega a ser verificado.

La comprobación de datos y la devolución de las instalaciones de las que el usuario es propietario, se evalúa mediante una conexión con la clase Communications, si todo funciona según lo correcto, la siguiente clase en ejecutarse será Installations, a la cual se le dice cuales son los parques solares que puede mostrar.

Para la confección de esta pantalla se ha creado una barra de título, con una imagen que representa a la aplicación, un título en el que pone "Access Control", la hora actual, un espacio para notificaciones y otro destinado a la cobertura.

Todo lo anterior constituye al constructor de la clase SaludationsScreen, aunque esta también consta de otro método que se activa en el momento en el cual el botón de retroceso de la interfaz del dispositivo móvil es pulsado y el usuario quiere salir de la aplicación, este método lanza un mensaje de despedida al usuario.

## 4.3 Clase Communications.java

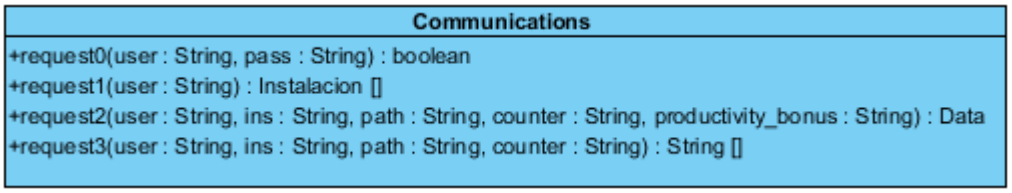


Figura 4.7 Esquema de la clase Conmmunications.

Esta clase no ha sido implementada para este trabajo, viene dada por la empresa para la cual se realiza la aplicación.

Los métodos que se utilizan en la clase SaludationScreen son:

- Request0: Este método recibe dos parámetros, los cuales son utilizados para el nombre de usuario y su contraseña, y con ellos comprueba los datos en la base de datos de la empresa, y devuelve cierto si el usuario y contraseña son legales y falso en caso contrario.

- Request1: Es el método que se utiliza para saber cuáles son los parques solares que corresponden a cierto usuario, pasado por parámetro, si este usuario no tiene ningún parque, el método no devolverá nada (null), por el contrario, devolverá información organizada de estas, esta información es definida por la clase Instalacion.

A continuación se observan los dos métodos que se utilizan en la clase Description, la cual veremos más adelante:

- Request2: Método encargado de devolver los datos generales de una instalación a partir de los datos globales de esta, recibe un parámetro de tipo Instalacion, y devuelve un tipo Data, más generalizado y con información más propia de dicha instalación.
- Request3: Se devuelven datos de producción de todos los días que hayan transcurrido del mes en el que se esté actualmente, por lo que necesita recibir casi todos los datos que conlleva la instalación para poder devolver la información deseada.

## 4.4 Clase Instalacion.java

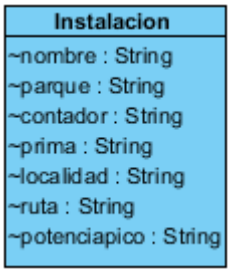


Figura 4.8 Esquema de la clase Instalacion.

Clase dedicada a la información de cada planta solar, de ella se puede obtener:

- Nombre de la instalación.
- Parque al cual pertenece la instalación.
- Valor actual del contador.
- Valor de la productividad.
- Código postal al cual pertenece el parque solar.
- Ruta donde se encuentran los datos de la instalación.
- Vatios de potencia.

## 4.5 Clase Data.java

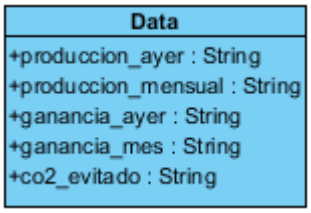


Figura 4.9 Esquema de la clase Data.

Clase dedicada para la información de las instalaciones de manera más concreta que la anterior clase, que es información de una manera más global. Esta consta de:

- Producción que generó la instalación ayer.
- Producción que lleva generada la instalación en el mes actual.
- Ganancia generada en el día anterior por la instalación.
- Ganancia acumulada por los días que han transcurrido del actual mes.
- Co2 evitado.

## 4.6 Clase Installations.java

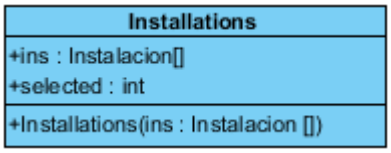


Figura 4.10 Esquema de la clase Installations.

A parte de las ya explicadas ButtonField, StandardTitleBar, FieldChangeListener, que han sido utilizadas para crear la misma interfaz que en la anterior pantalla, y para realizar un cambio de pantalla. También interactúa otra clase que a continuación será nombrada junto con una breve descripción de su uso.

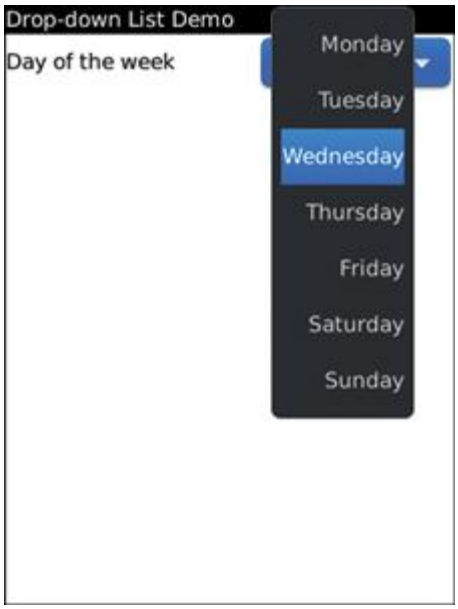


Figura 4.11 Ejemplo de ObjectChoiceField. [18]

- ObjectChoiceField: Un objeto de esta clase es una lista desplegable, se utiliza para obtener varias opciones de texto que los usuarios de la aplicación puedan seleccionar. Solo es posible seleccionar una opción a la vez. [17]

Al llegar a esta clase se supone que el usuario ya ha sido autentificado, y se presenta esta interfaz para que el pueda seleccionar la instalación de la cual quiere obtener información.

Esta pantalla consta de una interfaz similar a la descrita anteriormente en la clase SaludationsScreen, en el MainScreen, como título se cambia "Inmodo clients", por "Installations". El contenido del MainScreen es una lista desplegable donde se podrá elegir la instalación de la cual queremos obtener cierta información. Para poder cambiar de pantalla la interfaz consta de un botón, el cual una vez que es pulsado activa un evento en el que se escoge la instalación que se ha elegido por parte del usuario y activa la siguiente pantalla de la clase Description.

Este constructor también cuenta con la misma barra de título que la clase explicada anteriormente, SaludationsScreen. Esta barra está formada por un pequeño icono, el cual representa la aplicación, un título, "Access Control", la hora actual, y un espacio dedicado para notificaciones e iconos dedicados a la cobertura.

También es necesario volver a crear un método por si el usuario quiere salir de la aplicación directamente, pulsando el botón de retroceso de la interfaz del dispositivo móvil, para que la aplicación se cierre adecuadamente y poder despedirse del usuario.

## 4.7 Clase TextBoxField.java

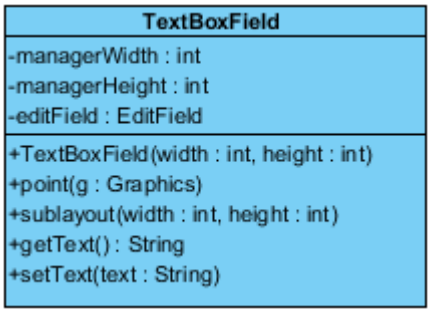


Figura 4.12 Esquema de la clase TextBoxField.

Antes de explicar esta clase cabe destacar que es código ya escrito por otro programador, ha sido buscada por internet, y añadida a nuestro trabajo. Este código fue adquirido desde la siguiente página: http://supportforums.blackberry.com/t5/Java-Development/Custom-TextBoxfield-cursor-problem-in-9800/m-p/597069.

Para entender bien esta clase, se va a explicar para lo que sirven las siguientes clases usadas en el código de esta:

- Graphics: Proporciona una superficie sobre la cual se puede dibujar, escribir, etc.
- EditField: Un campo de texto pero sin formato. [17]

Esta clase se utiliza para reemplazar un campo de texto predefinido, suele ser por motivos estéticos, ya que el campo de texto por defecto siempre tiene el mismo tamaño y esta clase tiene la particularidad de modificar el campo de texto y darle la altura y anchura que se quiera para adaptarlo a la interfaz gráfica.

En un campo de texto, BasicEditField, una vez que se tiene lleno el campo de palabras, se podrá seguir escribiendo pero no se ve lo que se escribe, dando lugar a errores y sin saber que se han cometido, en cambio el campo de texto creado por TextBoxField, una vez que se ha llenado, se sigue viendo lo que se está escribiendo, debido a que van pasando las palabras, sin dejar de ver lo que se escribe.

Este tipo de campo de texto se usa en la clase Descripcion, que se verá más adelante.

## 4.8 Clase ProgressBar.java

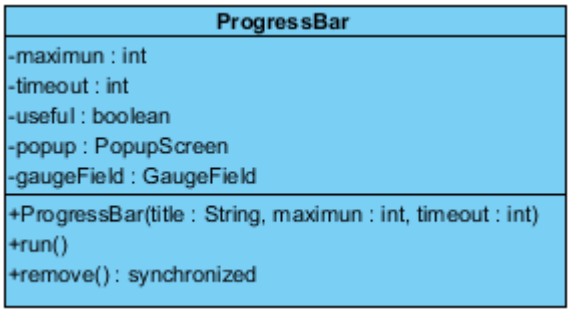


Figura 4.13 Esquema de la clase ProgressBar.

El código de esta clase se ha seleccionado desde internet, la utilidad de esta viene dada en referencia a las pantallas de espera, http://www.blackberryforums.com/developer-forum/61758-progress-bar.html, página donde se recoge el código de la clase.

Destacan tres clases dentro de ProgressBar, que aún no se han descrito y que a continuación se realiza su descripción:

- PopupScreen: Pantalla que proporciona rasgos para construir diálogos y pantallas de estado. En este tipo de pantallas es el administrador el que controla el diseño. [22]
- GaugeField: Es un campo de evaluación, que lleva un su interior una barra de herramientas en la que se muestra un indicador de progreso o un valor numérico.
- DialogFieldManager: Reserva espacio que luego será utilizado para un objeto de la clase PopupScreen. [17]

Estas tres clases van juntas, primeramente se reserva espacio con DialogFieldManager, seguidamente se crea la pantalla PopupScreen, donde se reservó espacio, y dentro de la pantalla, se inserta el campo de evaluación.

Figura 4.14 Ejemplo de ProgressBar. [18]

Después de explicar las clases que interactúan es fácil saber que ProgressBar se diseñó para representar un indicador de progreso que permita mostrar una pista visual donde se indique el progreso de una tarea, siempre y cuando esta se pueda calcular, como por ejemplo lo que queda para la finalización de una descarga, o para cargar una página web, etc.

## 4.9 Clase ScrollableImageField.java

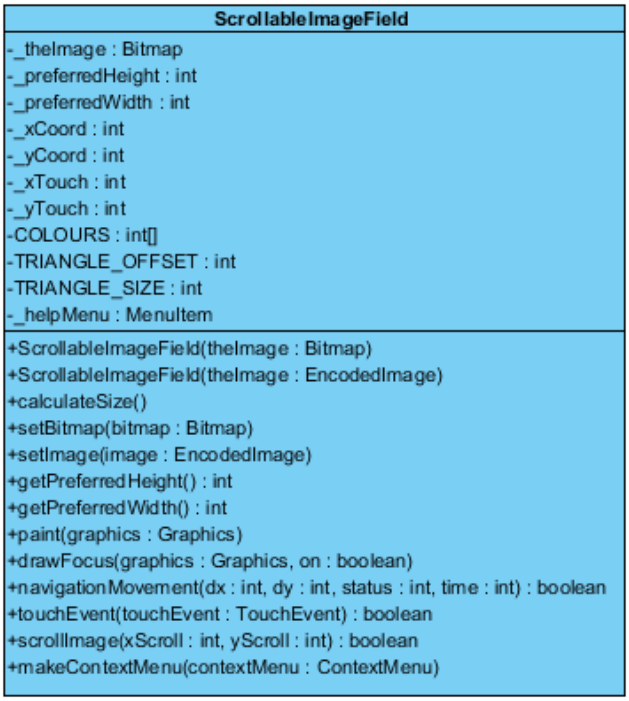


Figura 4.15 Esquema de la clase ScrollableImageField.

Para explicar esta clase lo primero que se va a realizar, como se aprecia a lo largo de la memoria, es explicar las clases que se usan dentro del código de esta.

A parte de las ya explicadas, Bitmap y Graphics, también tienen contacto las siguientes clases:

- EncodedImage: Encapsula imágenes codificadas de varios formatos.
- TouchEvent: Esta clase se elabora para la personalización del comportamiento de la interacción táctil del dispositivo.
- ContextMenu: Esta clase da la opción de crear un menú contextual, e insertarlo dentro de esta llamando al método "makeContextMenu", y se le pasa el menú como parámetro.

La clase ScrollableImageField, es la última clase donde el código se ha seleccionado de internet, este fue adquirido desde la siguiente página web, http://supportforums.blackberry.com/t5/Java-Development/Create-a-scrollable-image-field/ta-p/444955.

Esta clase fue insertada en el trabajo para que todos los usuarios puedan ver las imágenes que están insertadas en la aplicación, ya que ScrollableImageField, se encarga de compatibilizar las imágenes con la pantalla del dispositivo desde el cual se esté ejecutando la aplicación, y de esta forma se puede agrandar, se puede ver completamente entera la imagen, aunque por defecto esta fuese más grande que el tamaño de la propia pantalla.

## 4.10 Clase Description.java

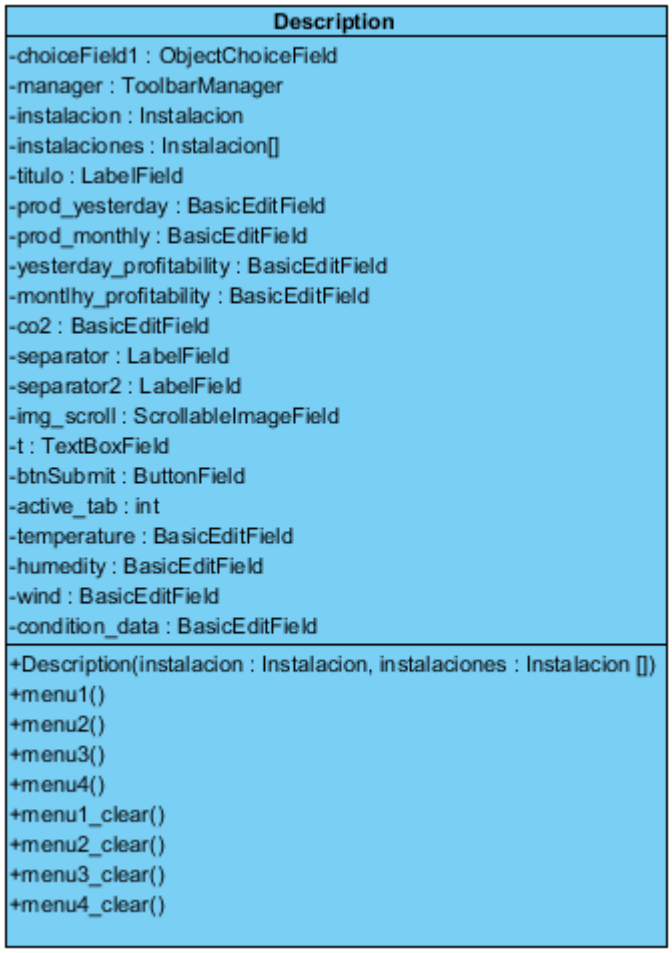


Figura 4.16 Esquema de la clase Description.

Por último, se describe la última y más importante clase dentro de la aplicación. Esta interacciona con la mayoría de las clases explicadas anteriormente, una breve lista de las clases con las que interactúa podría ser la siguiente: ObjectChoiceField, Instalacion, LabelField, BasicEditField, ScrollableImageField, etc.

Las últimas clases que quedan por explicar son las siguientes:

- ToolbarManager: Representa una barra de herramientas, horizontal, que ofrece al usuario una manera rápida y fácil de acceder a las acciones frecuentes de una aplicación o pantalla. Cada barra de herramientas consiste en un conjunto de iconos que aparecen situados en la parte inferior de la pantalla. [23]

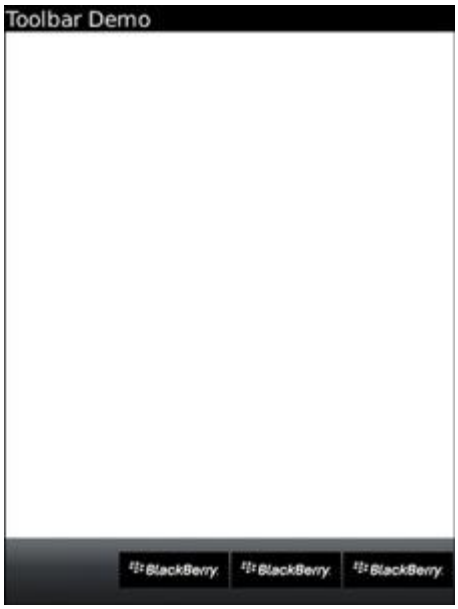


Figura 4.17 Ejemplo de ToolbarManager. [18]

- ToolbarButtonField: Representa los botones dentro de la barra de herramientas. [23]

La explicación de esta clase se va a realizar por partes.

### 4.10.1 Constructor

**public Description(Instalacion instalacion,Instalacion[] instalaciones)**

En el constructor de esta clase, lo primero que se hace es ponerle título a la pantalla, en este caso "Information". Seguidamente se crea la barra de herramientas y los botones que después de ser creados se insertarán en la barra de herramientas, estos botones conllevan unos eventos, que sirven para decir a la aplicación lo que tiene que hacer en el momento en que cierto botón es activado.

Para no perder estética, se crea la barra de título que se ha ido creando en todas las pantallas anteriormente descritas, esta consta de un icono representativo, un título, reloj, notificaciones y espacio dedicado a los iconos de la cobertura.

Por último, se activa el menú1.

### 4.10.2 Menu1_clear, menu2_clear, menu3_clear y menú4_clear

Son los métodos encargados de limpiar la pantalla de los objetos que están añadidos actualmente, no se ha podido hacer un método global porque cada menú tiene unos objetos diferentes activos.

De esta forma nos aseguramos que no eliminamos un objeto no existente en la página actual, ya que se generaría una excepción.

Para saber que menú hay que limpiar, se ha creado un atributo dentro de esta clase que llevará el número del menú que hasta este momento estaba activo.

### 4.10.3 Menu1

**public void menu1()**

En un primer paso se mira si esta activo algún menú anterior, y si esto ocurre, la aplicación se dispone a dejar la pantalla limpia. Seguidamente se dice que la pantalla activa actualmente es la número uno, después de esto, se pasa a la representación del menú en pantalla.

Este menú es el encargado de mostrar un sumario de datos, por lo que usa la clase Communications, concretamente llama al método request2, para así obtener información acerca de la instalación de la cual se quiere mostrar un sumario. A continuación y a modo de resumen se muestran los datos devueltos por el método nombrado anteriormente.

Los datos que se muestran por pantalla son los datos definidos anteriormente en la clase Data, producción y ganancias generadas en el día de ayer y en el mes actual, y el $CO_2$ evitado. Estos se muestran por pantalla dentro de un campo de texto, BasicEditField.

### 4.10.4 Menu2

**public void menu2()**

En este método, al igual que en todos los menús, lo primero que se realiza es averiguar si se tiene que limpiar la pantalla, y qué menú es el que está actualmente en ella.

La función principal del Menu2 es devolver la situación atmosférica actual en la instalación elegida, para ello se realizan peticiones HTTP a Google, añadiendo a la URL el código postal del lugar del cual se requiere información atmosférica.

Si Google no devuelve dicha información se repite el intento de conexión y la petición de información hasta un máximo de 10 veces, en caso de que se agoten las posibilidades, se genera un aviso informativo diciendo que se ha producido un error, y la aplicación retorna al menú 1.

Cuando Google devuelve dicha información, esta se encuentra introducida en una lista de nodos de la cual se selecciona de entre todos los nodos los que se encuentran relevantes para mostrar por pantalla, la información seleccionada es:

- Temperatura.
- Porcentaje de humedad.
- Condición del viendo.

- Condición atmosférica.
- Icono que represente la condición atmosférica.

Toda las conexiones con Google se han podido realizar gracias al API, que existe entre BlackBerry y Google, anteriormente explicado, del cual facilitamos ahora una página web por si se quiere indagar con el, http://code.google.com/intl/es-ES/apis/chart/image/docs/chart_playground.html.

La selección de información se representa en la pantalla de un dispositivo móvil mediante campos de texto. A excepción del icono, que se representa mediante la clase explicada anteriormente, ScrollableImageField.

Figura 4.18 Iconos representativos de todos los menús.

### 4.10.5 Menu3

**public void menu3()**

Este menú se encarga de presentar una gráfica con los datos diarios del mes actual, gracias a la clase Communications, y al método perteneciente a esta, request3, se pueden averiguar los datos que se quieren representar gráficamente, ahora solo falta realizar la gráfica.

Para realizar la gráfica se debe tener agregada al código un API de Google, para este tipo de peticiones, en el que se hace una conexión con su servidor indicándo los datos que se quieren representar, y Google devuelve la gráfica que se tiene que añadir a la pantalla de dicho menú, para que los usuarios puedan visualizarla. Si la conexión falla, se mostrará un evento de error para que el usuario sepa que no ha sido posible la conexión, y pueda volver a intentarlo.

Esto siempre es realizado después de haber limpiado la pantalla del menú que estuviera inicializado anteriormente, evitando así colisiones de menús.

### 4.10.6 Menu4

**public void menu4()**

Como ya se indica en los anteriores menús, lo primero que se debe hacer al inicializar un menú es limpiar lo que hubiese anteriormente, y dejar claro que el nuevo menú activo es menu4.

La interfaz de este menú es bastante simple, una etiqueta en la que se describe brevemente para que se utiliza este menú, un campo de texto, TextBoxField, del tamaño que quiera el administrador, y un botón que cuando es activado recoge el texto que se haya escrito en el espacio reservado para ello, y abre la interfaz que los dispositivos BlackBerry disponen para enviar correos a otros usuarios, pero con la información ya añadida, texto escrito y dirección del buzón de sugerencia de la empresa. Esto se hace por si se quiere añadir algo mas en el texto, y por si se quiere añadir alguna dirección, y que no solo sea escrito al departamento de sugerencias.

### 4.10.7 Cerrar el programa

**public boolean onClose()**

Para cerrar la explicación de esta clase, lo último que se ha creado es un método para cerrar bien la aplicación, y que no se quede abierta en un segundo plano.

# 5. Manual de usuario

En este apartado se explica la instalación, y manejo de la aplicación. En un principio se explicará cómo se instala el software en un dispositivo BlackBerry, y por último el uso de esta, y se verá representada gráficamente.

## 5.1 Instalación

Para la instalación es requisito obligatorio, disponer de un dispositivo móvil BlackBerry.

Segundo requisito obligatorio, disponer de internet en dicho dispositivo, ya que si no sería imposible realizar la descarga del software.

Para dicha descarga se inicia la tienda de aplicaciones de la cual disponen los teléfonos BlackBerry. Una vez inicializado, se busca el icono que representa la palabra buscar, el cual es el siguiente: . Y se busca la aplicación llamada "Inmodo Clients".

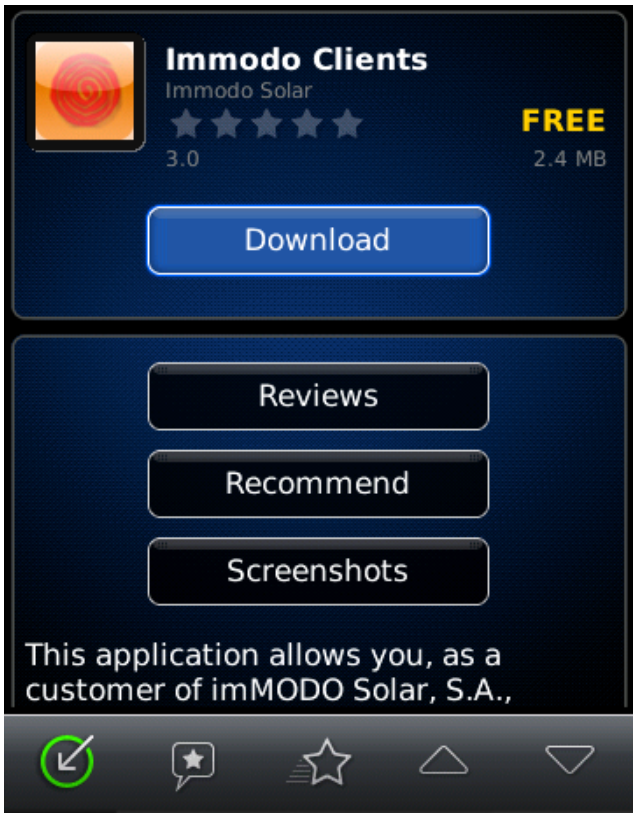


Figura 5.1 Imagen de la pantalla del dispositivo a la hora de descargar la aplicación.

Se descarga y posteriormente se instala automáticamente.

## 5.2 Inicio

La pantalla de inicialización de la aplicación, es una pantalla simple, en el que se dispone de un campo para poner el nombre de usuario y otro campo para poner la contraseña que autentifica al usuario.

Una vez introducidos estos datos se pulsa el botón “Entrar”, que sirve para dar paso a la autentificación del usuario, y una vez autentificado, dar paso a la siguiente pantalla.

Figura 5.2 Representación de la pantalla de inicio.

Se pueden obtener dos respuestas, la respuesta principal será pasar a la siguiente pantalla, y la respuesta secundaria sería una excepción diciendo que no ha sido verificado, en este caso se obtiene un cuadro como el de la siguiente figura.

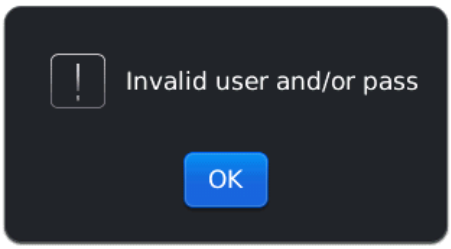


Figura 5.3 Usuario inválido o contraseña inválida.

## 5.3 Elección de instalación

Elegir la instalación la cual se quiere ver información es la finalidad de esta pantalla, para ello dicha pantalla dispone de un menú desplegable con el nombre de las instalaciones que dispone el usuario anteriormente autentificado.

Una vez seleccionada la instalación en el menú, se pulsa el botón "Consultar Datos", y se podrá visualizar la información solicitada.

Figura 5.4 Pantalla de selección de instalación.

## 5.4 Menú principal de la aplicación

Este es el centro de operaciones de la aplicación, desde aquí se podrá visualizar toda la información, y mandar correos electrónicos de sugerencia.

Para poderse mover desde una información a otra, o desde un sub-menú a otro, se dispone de una barra de herramientas que tiene la siguiente estructura.

Figura 5.5 Barra de herramientas.

A continuación se puede observar para que se utiliza cada símbolo.

### 5.4.1 Sumario de producción

En esta parte del menú principal se puede visualizar un resumen de toda la información, en esta pantalla se visualiza la producción de ayer, del mes actual, el dinero que han producido dichos datos de producción, y los kilogramos de $CO_2$ evitado.

Según se accede al menú principal, este subordinado es el menú por defecto, es decir, es el primer menú que visualizará el usuario después de seleccionar la consulta de datos de una instalación.

Figura 5.6 Representación del sumario de producción.

### 5.4.2 Condiciones atmosféricas

Pantalla dedicada a las condiciones atmosféricas por las que pasa actualmente la instalación de la cual se está consultando información.

En esta pantalla se diferencian cuatro datos: Temperatura en grados centígrados, porcentaje de humedad, hacia donde sopla el viento y a cuanta velocidad, y un adjetivo del tiempo que hace, entre ellos, nublado, soleado, lluvioso, despejado, etc.

Figura 5.7 Pantalla dedicada a las condiciones atmosféricas.

Esta pantalla puede causar un error al abrirla, y esta trata de algún tipo de problema con la conexión a internet, o con la conexión del servidor el cual proporciona los datos, este error puede aparecer en dos pantallas, tanto en las condiciones atmosféricas, como en la estadística mensual, y tiene el siguiente aspecto.

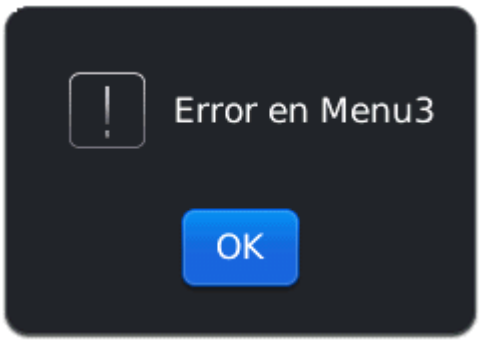


Figura 5.8 Visualización de error.

### 5.4.3 Estadística mensual

Gráfica en la que se representa la producción del mes, diariamente. Es decir, una barra de la gráfica por cada día que lleva el mes, con la finalidad de representar la producción de ese día.

De esta forma se puede ver cuáles son los días que se ha producido más, y cuáles son los días que se ha producido menos.

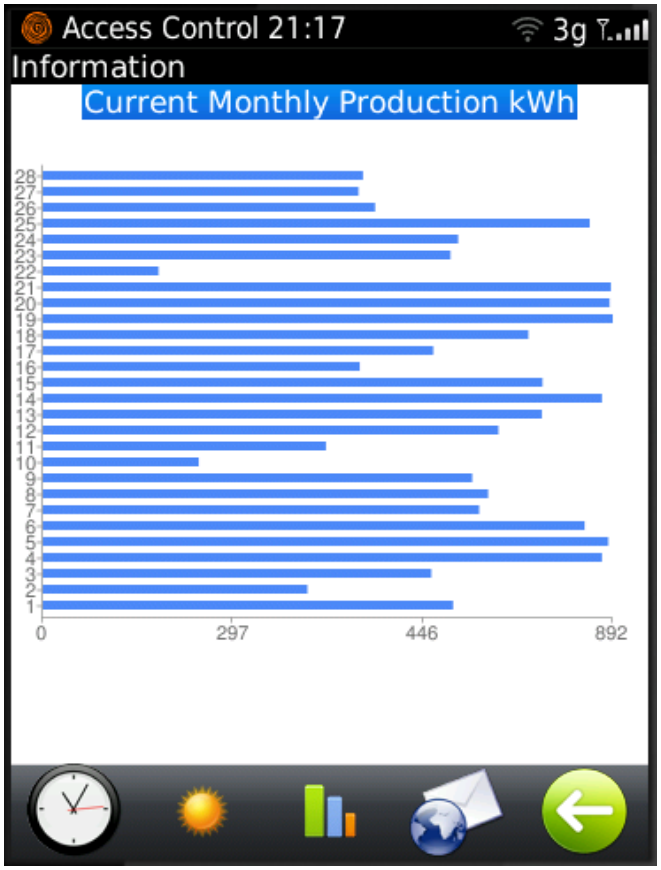


Figura 5.9 Gráfica de producción diaria.

### 5.4.4 Envío de sugerencias

Por si se quiere comunicar algo a la empresa encargada de todo el software referente a las instalaciones, esta aplicación dispone de un espacio reservado para el envío rápido de correos electrónicos.

Esta pantalla tiene el siguiente aspecto.

Figura 5.10 Aspecto de la pantalla dedicada al envío de sugerencias.

### 5.4.5 Volver

Botón utilizado para volver a seleccionar otra instalación por si se quiere consultar otros datos que no sean los de la instalación en la que se está actualmente.

## 5.5 Salir de la aplicación

Pulsando el botón de retroceder, perteneciente a la interfaz del dispositivo móvil se saldrá de la aplicación, quedando está bien cerrada y no en un segundo plano.

Para saber si se ha salido bien, antes de cerrarse por completo se debe obtener un cartel informativo como en el de la Figura 5.11.

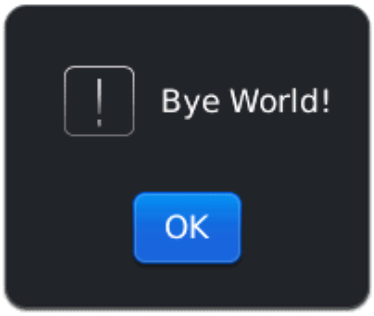


Figura 5.11 La aplicación de despide del usuario.

# 6. Conclusiones y trabajos futuros

## 6.1 Conclusiones

Varias son las conclusiones extraídas a lo largo de la elaboración de la aplicación.

El desarrollo de cualquier sistema requiere un exhaustivo estudio de él para comprender perfectamente sus funcionalidades y así poder detectar derivados.

Trabajar con sistemas grandes y actualmente en funcionamiento representa una gran variedad a la hora de poder implementar una aplicación en Java. Siendo factible elaborarla de muchas maneras diferentes y todas con la misma finalidad, por lo que, gracias a la gran variedad de argumentos con los que se pueden contar, no existe una excesiva dificultad a la hora de diseñar una interfaz.

La complejidad en este tipo de software suele darse en la corrección de los errores y/o captura de excepciones, con el hecho de ir metiendo objetos dispares. La solución a estos es la parte más destacada, y en la que radica la mayor parte del trabajo de un ingeniero en pruebas.

De igual modo, ha sido gratificante el descubrimiento de la variedad de complementos que se pueden usar en la creación de interfaces gráficas

Como conclusión final, en este proyecto y principalmente en el periodo de prácticas en una empresa del sector informático, se han puesto de manifiesto muchos de los conceptos adquiridos en la Universidad, entre los que cabe destacar la importancia en las distintas fases del ciclo de vida del software.

.

## 6.2 Trabajos futuros

La aplicación desarrollada supone un punto de partida fundamental a la hora de poder estudiar y realizar futuros trabajos.

### Sistemas operativos

Para satisfacer la demanda de otros clientes se ha considerado la idea de extender la aplicación a otros dispositivos móviles. Atendiendo al número de usuarios se debe considerar un futuro desarrollo del software para iOS y Android. Hecho que permitiría la adaptación del programa a otros computadores portátiles como los Tablets.

### Base de datos

Otra mejora notable puede ser la modificación de la base de datos agregada al servidor web para reducir el número de conexiones con el mismo, agilizando la aplicación y dotando al entorno una mayor velocidad y rendimiento.

### Prestaciones de la aplicación

Se podrían establecer más funcionalidades que permitan a los clientes un mayor control sobre sus instalaciones solares. La visualización de las placas en cualquier momento sería una de las características que podría resultar interesante desarrollar.

# 7. Referencias

[1] Jay Pastranna. "Un poco sobre la historia de **RIM**"

http://www.clublackberry.com/2011/04/19/un-poco-sobre-la-historia-de-rim/

[2] Anónimo "Diez ventajas del software libre y propietario"

http://www.abadiadigital.com/articulo/diez-ventajas-del-software-libre-y-propietario/

[3] W-Bot "BlackBerry OS"

http://www.comunidad.wilkinsonpc.com.co/rim-blackberry-266/blackberry-os-15297.html

[4] Vfarias "Java en dispositivos móviles"

http://www.blog.espol.edu.ec/vfarias/

[5] Julián Gómez "Eclipse SDK"

http://www.eclipse-sdk.softonic.com/

[6] Josep María "RIM anuncia el BlackBerry JDE Plug-in para Eclipse"

http://www.miblackberry.com/2008/03/31/rim-anuncia-el-blackberry-jde-plug-in-para-eclipse.html

[7] Zona BlackBerry "RIM lanza BlackBerry App World"

http://zonabinaria.net/web/?p=1038

[8] BlackBerry "Escaparate de BlackBerry App World"

http://es.blackberry.com/services/appworld/FAQes.pdf

[9] DeltaTec Systems "Bienvenido a DeltaTec Systems"

http://www.deltatec-systems.com/

[10] DeltaTec Solar "Monitorizacion y control de instalaciones fotovoltaicas"

http://www.deltatec-solar.com/files/documents/PresentacionSistema.pdf

[11] Inmodo Solar "Perfil"

http://www.immodosolar.com/es/empresa/perfil.php

[12] Rentasol "Quienes somos"

http://www.rentasol.es/Quienes-somos_es_13.html

[13] Siliken "Siliken Energía Renovable"

http://www.siliken.com/

[14] Ecooo "Quienes somos"

http://www.ecooo.es/2/Empresa/

[15] Carmelo Cortes "Alternativa de sistema de control y seguimiento personal a través de gps"

[16] Jeff "El protocolo HTTP"

http://es.kioskea.net/contents/internet/http.php3

[17] BlackBerry "Package net.rim.device.api.ui.component"

http://www.blackberry.com/developers/docs/3.6api/net/rim/device/api/ui/component/package-summary.html

[18] BlackBerry Java SDK "Guía de referencia rápida"

http://docs.blackberry.com/en/developers/deliverables/25237/

[19] BlackBerry "Etiqueta"

http://docs.blackberry.com/en/developers/deliverables/25237/Label_fields_1211031_11.jsp

[20] BlackBerry "Class StandardTitleBar"

http://www.blackberry.com/developers/docs/7.0.0api/net/rim/device/api/ui/component/StandardTitleBar.html

[21] BlackBerry "Interface FieldChangeListener"

http://www.blackberry.com/developers/docs/3.6api/net/rim/device/api/ui/FieldChangeListener.html

[22] BlackBerry "Class PopupScreen"

http://www.blackberry.com/developers/docs/4.0.2api/net/rim/device/api/ui/container/PopupScreen.html

[23] BlackBerry "Package net.rim.device.api.ui.toolbar"

http://www.blackberry.com/developers/docs/7.0.0api/net/rim/device/api/ui/toolbar/package-summary.html