

Desarrollo de una aplicación móvil en sistema Android  
para el control remoto de dispositivos mediante la  
tecnología bluetooth 4.0

**Autor:**

Víctor Hernández Muñoz

**Director:**

Carlos Palau Salvador (Dep. de comunicaciones ETSIT)

**Codirector:**

Javier Soriano Falcó (Director técnico BlauTic)

# INDICE

1. Introducción
  - 1.1 Motivación
  - 1.2 Descripción del proyecto
  - 1.3 Objetivos del proyecto
  - 1.4 Metodología de desarrollo
  
2. Background tecnológico (estado del arte)
  - 2.1 Comunicaciones de corta distancia.
    - 2.1.1 Bluetooth
    - 2.1.2 Tecnología NFC
    - 2.1.3 ANT
    - 2.1.4 ZigBee
    - 2.1.5 Infrarrojos
  - 2.2 Sistemas Operativos en dispositivos móviles
    - 2.2.1 iOS
    - 2.2.2 Android
    - 2.2.3 Symbian
    - 2.2.4 Windows Phone
    - 2.2.5 Blackberry Os
    - 2.2.6 Bada
  - 2.3 Análisis de dispositivos móviles
  - 2.4 Proyectos similares
  
3. Aplicación : Plataforma BlauTiC
  - 3.1 Modelo de casos de usos
  - 3.2 Especificaciones
    - 3.2.1 Funciones principales
    - 3.2.2 Función: configuración
    - 3.2.3 Función: disponibilidad y monitorización de dispositivos
    - 3.2.4 Función: mensajes
    - 3.2.5 Función: sensores
    - 3.2.6 Función: seguridad
    - 3.2.7 Función: sincronización
  - 3.3 Elección de tecnologías
    - 3.3.1 Análisis del sistema Android
      - 3.3.1.1 Historia del nacimiento de la plataforma Android
      - 3.3.1.2 Características principales de la plataforma

- 3.3.1.3 Filosofía de diseño de aplicaciones
- 3.3.2 Entorno de programación
- 3.3.3 Base de datos (Completar)
- 3.3.4 Gráficos y soporte de idiomas
- 3.3.5 Bluetooth 4.0
- 3.3.6 Herramientas utilizadas
  - 3.3.6.1 Android Software Development Kit (SDK)
  - 3.3.6.2 SVN

### 3.4 Planificación

### 3.5 Diseño.

- 3.5.1 Diseño de pantallas y funcionalidades
  - 3.5.1.1 Configuración
  - 3.5.1.2 Dispositivos
  - 3.5.1.3 Mensajes
  - 3.5.1.4 Sensores
- 3.5.2 Esquema de la aplicación

### 3.6 Desarrollo

- 3.6.1 Implementación
  - 3.6.1.1 Sincronización
  - 3.6.1.2 Bluetooth
  - 3.6.1.3 Recursos XML
- 3.6.2 Problemas

## 4. Resultados

- 4.1 Pantallas definitivas
- 4.2 Test de pruebas
- 4.3 Lanzamiento en Google Play

## 5. Costes temporales / económicos

- 5.1 Modelo de negocio

## 6. Conclusiones

- 6.1 Objetivos como proyecto
- 6.2 Posible evolución futura
- 6.3 Productos actuales basados en el proyecto

## 7. Bibliografías y referencias

## 8. Anexos

### 8.1 Instalación del entorno de desarrollo de Android

### 8.2 Utilización de subversión (SVN)

# 1. Introducción

## 1.1. Motivación

Desde el primer momento que decidí iniciar mi proyecto final de carrera (PFC) tuve claro que elegiría uno que me permitiese además de poner en práctica todos los conocimientos teóricos aprendidos durante la carrera aprender algo más. Aprender algo más y enriquecerme de la vida real, de lo práctico, de cosas que se me van a requerir posteriormente en el mundo laboral, y dejar a un lado las interminables fórmulas matemáticas y los días eternos entre las paredes de un laboratorio.

Como amante de las nuevas tecnologías que me considero, me interesé pronto por el mundo de las aplicaciones para dispositivos móviles, smartphones, ya que considero que es un mercado que a día de hoy está en auge y es un activo de futuro para cualquier empresa o negocio que quiera lanzar sus productos y llegar a todos los ciudadanos. Sólo hay que darse cuenta de la rápida proliferación de estos dispositivos en la sociedad, no sólo en los jóvenes, y de la aceptación que tienen las múltiples aplicaciones para móviles que existen.

Además, también me interesé en la posibilidad de realizar mi PFC asociado a una empresa o la realización de alguna práctica de empresa que me dejara ver la situación del mundo laboral. Al terminar los estudios el recién ingeniero tiene que lidiar en un ambiente del cual no ha cursado ninguna asignatura al respecto.

Pues bien, un buen día todo esto se alió y de esa primera reunión con Javier (Director del proyecto BlauTiC) y del interés mutuo por realizar esto, nació este proyecto que me ha aportado toda una experiencia, todo lo que iba buscando, y que me lleva a definirlo sin ninguna duda...la mejor asignatura de la carrera.

## 1.2. Descripción del proyecto

El proyecto BlauTic nace con el objetivo de facilitar y acercar a la sociedad mediante el uso de las nuevas tecnologías el control sobre dispositivos a distancia tanto en un entorno laboral como en situaciones cotidianas. Concretamente se pretende que mediante una aplicación móvil se pueda controlar de forma remota el funcionamiento de cualquier dispositivo electrónico o monitorizar el funcionamiento del mismo. Todo ello con la novedad que supone la utilización de la tecnología inalámbrica bluetooth 4.0. de menor consumo energético, mayor eficiencia y mejores prestaciones que las versiones anteriores.

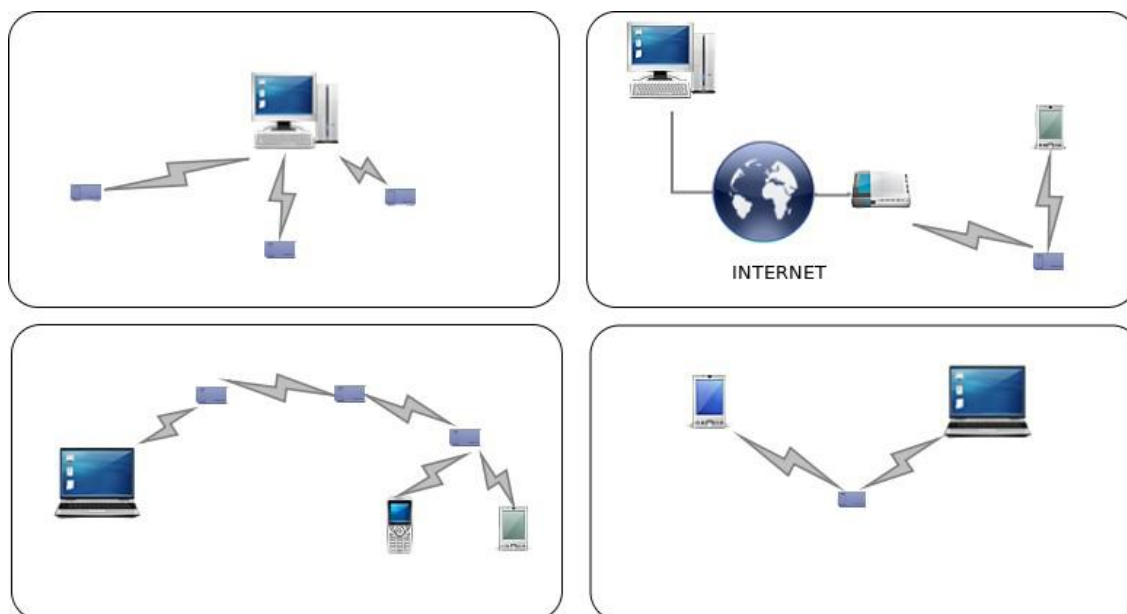
El proyecto integra diferentes tecnologías para proporcionar servicios tecnológicos avanzados basados en los siguientes puntos:

- Comunicaciones inalámbricas de corta distancia y baja tasa de transferencia de datos.
- Múltiples dispositivos interfaz (PC, Smartphone, móvil, PDA, etc..) con entorno gráfico e interactivo.
- Aplicaciones con alta posibilidad de personalización y facilidad de desarrollo de nuevas funcionalidades que cubran necesidades detectadas.

Además, el sistema inicial consta de los siguientes elementos funcionales:

1. Dispositivo electrónico (hardware + software) de diseño propietario, de reducidas dimensiones y bajo consumo, alimentado por una pila botón o una pequeña placa solar con encapsulado. Contiene comunicaciones inalámbricas, memoria, reloj en tiempo real, acelerómetro y puertos de comunicación con sensores/equipos externos. En la siguiente memoria se hará siempre referencia a este elemento como Dispositivo BlauTiC.
2. Aplicación software para teléfono móvil que sirve de interfaz del sistema a los usuarios.
3. Aplicación software para plataforma PC que sirve de interfaz para el administrador del sistema o usuario que debe comunicar con grupos de dispositivos.
4. El sistema se puede completar con el dispositivo de red como plataforma de interfaz a múltiples sistemas distribuidos remotamente: 3G o WiFi + Bluetooth.

En los siguientes gráficos podemos apreciar las diferentes combinaciones de comunicación entre los dispositivos BlauTiC y sus posibles interfaces:



*Figura 1.1 Esquema BlauTiC (Fuente: elaboración propia)*

El sistema aportará la siguiente funcionalidad global que variará exclusivamente según el software interfaz instalado. El hardware es el mismo si no se requiere algún tipo de adaptación para conexión con el equipo externo, como pudiera ser en los PC que necesiten de un adaptador USB para realizar las comunicaciones inalámbricas.

#### GENERALES:

- La principal vía de comunicación inalámbrica será bluetooth 4.0
- Acceso restringido únicamente a los interfaces pertenecientes al sistema
- Sincronización de fecha y hora únicamente por el interfaz configurado como administrador (master).

#### ALMACENAMIENTO/LECTURA DE INFORMACIÓN SOBRE DISPOSITIVOS BLAUTIC:

- Interfaz de usuario gráfica.
- Lectura + escritura de mensajes.
- Mensajes tipo texto, texto con confirmación, calendario o sensores.
- Mensajes guardados con información acerca del usuario y fecha asociada.
- Edición de mensajes.
- Inserción de mensajes predefinidos.
- Envío a grupos de usuario seleccionables.

#### MENSAJES TIPO CALENDARIO Y TEXTO CONFIRMACIÓN. AVISOS:

- Lectura y gestión de mensajes tipo calendario.
- Aviso sonoro/visual de mensajes pendientes de actuación.
- Avisos de mensajes pendientes de lectura + reconocimiento

#### SEGURIDAD/CONTROL

- Almacenamiento e los dispositivos presentes a una determinada fecha/hora
- Monitorización de la presencia de dispositivos dentro de un rango de distancia
- Monitorización del movimiento de dispositivos
- Avisos configurables: pérdida de presencia, objetivo alcanzable, etc..

#### LOCALIZACIÓN INVERSA (Principalmente espacios interiores)

- Identificación de baliza de localización
- Habilidad de funciones según punto de localización:
  - Plano de situación
  - Selección de destino
  - Representación ruta
  - Audios de ruta a seguir
  - Multimedia relacionada con el punto de interés donde se encuentre
  - Cuestionarios/encuestas

#### SENSORIZACION (red de sensores)

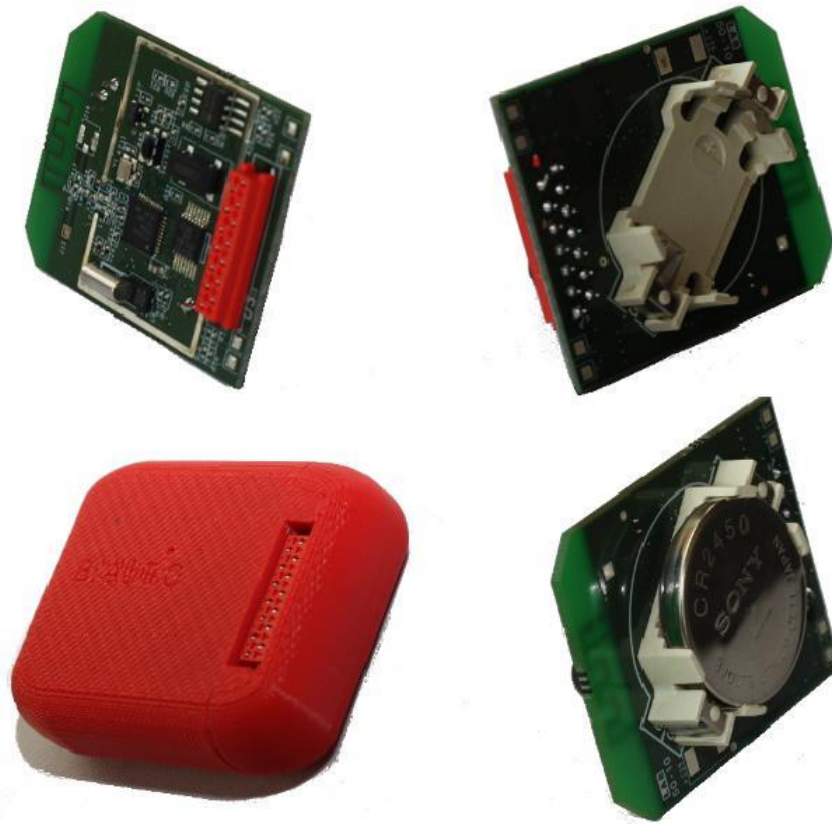
- Configuración de sensores
- Lectura de valores de los sensores con entradas digitales
- Lectura valores entradas analógicas
- Lectura de información externa por puertos de comunicación
- Sensores internos: acelerómetro

#### DISTRIBUCION MASIVA DE INFORMACIÓN ACTUALIZADA EN RED

- Comunicación entre dispositivos formando una red
- Instalación dinámica de la red
- Envío de información bidireccional por la red. Sincronización de datos con servidor remoto.

En el punto cuatro de esta memoria, en las especificaciones, se ampliará con más detalle la funcionalidad que se desea implementar en el software de la aplicación.

A continuación, mediante estas imágenes se muestra el aspecto físico del dispositivo hardware-software inalámbrico BlauTic A1/B1, desarrollado por la empresa BlauTic, y que utilizaremos para la implementación de este proyecto.



*Figura 1.2 Dispositivo BlauTic (BlauTic A1/B1) (Fuente: BlauTic)*

A lo largo de esta memoria citaremos indistintamente como Dispositivo BlauTic o BlauTic A1/B1 para referirnos a él.

### **1.3. Objetivos del proyecto**

Por la parte que a mi me concierne en este proyecto y que por tanto fundamenta mi PFC, deberé diseñar e implementar una aplicación para dispositivos móviles, Smartphone, que se ajuste a las especificaciones del software del proyecto BlauTic.

Para ello realizaré un estudio de mercado y de características de las principales plataformas para dispositivos móviles que existen a día de hoy. El objetivo será adecuar el software a la mejor de ellas desde el punto de vista del desarrollo de la aplicación permitiendo un fácil manejo del código que permita realizar ajustes o modificaciones en el en un futuro así como facilitar la distribución al mercado. También se compararán las diferentes tecnologías de comunicación existentes en la actualidad.



La funcionalidad total de la que debo dotar la aplicación me permitirá profundizar sobre aspectos teóricos aprendidos a lo largo de la carrera como:

- comunicaciones inalámbricas de corta distancia. (Bluetooth)
- comunicaciones de datos a través de red haciendo uso de sistemas 3G y dispositivos WiFi
- manejo de interfaz gráficas en el diseño de programas con distintos lenguajes de programación

## **1.4 Metodología de desarrollo**

A partir de las especificaciones que se den se dividirá el proyecto en distintas fases que contendrán cada una los bloques principales a desarrollar. Se establecerá una metodología de trabajo con las posibilidad de diseñar código a distancia utilizando un software de control de versiones (*Subversion*) y un servidor central donde se subirán los avances y desarrollos del código de la aplicación para que este pueda estar al alcance de todas las personas que trabajen o lo supervisen. Además en este servidor remoto se alojará toda la documentación necesaria referente a las especificaciones del producto.

El equipo de trabajo estará formado en total por tres personas. Un coordinador general y supervisor del proyecto BlauTic en la empresa, en este caso Javier Soriano, un desarrollador senior responsable de la planificación y coordinador de la aplicación móvil, Borja Roig, y yo mismo como desarrollador proyectista fin de carrera.

Se realizarán de manera periódica reuniones presenciales tanto con el supervisor como con el coordinador para controlar, realizar correcciones, y comprobar el estado de dichas fases de desarrollo.

## 2. Background tecnológico (estado del arte)

### 2.1 Comunicaciones de corta distancia

En este apartado se van a describir y analizar algunas de las principales tecnologías existentes para realizar comunicaciones inalámbricas entre dispositivos en corta distancia. La mayoría de estas están implementadas en la banda ISM (Industrial, Scientific and Medical) del espectro radioeléctrico y facilitan las comunicaciones, entre dispositivos móviles y fijos eliminando los cables y conectores.

Destacaremos la coexistencia en la actualidad de todas estas tecnologías ya que cada una tiene su implementación en un campo determinado dependiendo de las necesidades tecnológicas. En el siguiente apartado intentaremos explicar mediante ejemplos actuales los diferentes usos.

#### 2.1.1. Bluetooth

Viene a ser el nombre común por el que se conoce a la especificación IEEE 802.15.1 que define un estándar global de comunicaciones inalámbricas que permite la transmisión de voz y datos entre diferentes dispositivos mediante un radioenlace a 2.4 GHz

Nace a raíz de las diversas investigaciones y estudios realizados por la compañía Ericsson en el año 1994 con el objetivo de encontrar un interfaz de bajo coste y bajo consumo para interconectar distintos dispositivos. Posteriormente, en 1999, se crearía el llamado SIG (*Special Interest Group*) que consistió en la unión de las empresas con mayor peso en el mundo de las telecomunicaciones (Inicialmente Ericsson, Intel, Nokia, Toshiba e IBM a las que luego se unieron Microsoft, 3COM, Lucent y Motorola) que fue quien impulsó definitivamente el desarrollo de esta tecnología.

Los dispositivos que lo implementan pueden comunicarse entre ellos y transmitir voz y datos siempre que se encuentren dentro de un rango de alcance. Una de las ventajas de este sistema es que los dispositivos no requieren estar alineados, incluso pueden estar en habitaciones diferentes. Los dispositivos pueden clasificarse en clases en referencia a su potencia de transmisión y a su rango de cobertura, siendo todos compatibles entre sí.

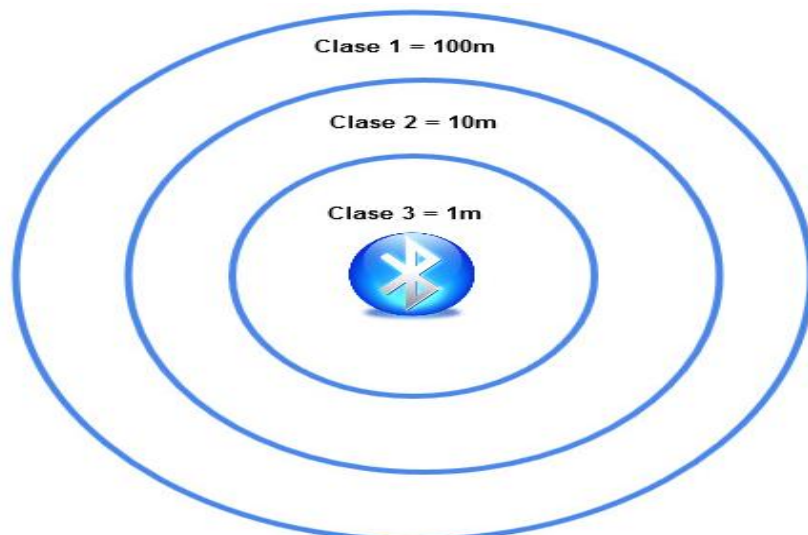


Figura 2.1 Alcance bluetooth (Fuente: elaboración propia)

Además, los dispositivos Bluetooth también pueden clasificarse según su ancho de banda definiéndose así las distintas versiones existentes:

<b>Versión</b>	<b>Ancho de Banda</b>
Versión 1.2	1 <a href="#">Mbit/s</a>
Versión 2.0 + EDR	3 <a href="#">Mbit/s</a>
Versión 3.0 + HS	24 <a href="#">Mbit/s</a>
Versión 4.0	<a href="#">N/D</a>

Como se indican en las especificaciones de Bluetooth, se define un canal un canal de comunicaciones de máximo 720 Kb/s para un rango óptimo de 10 m. La frecuencia de trabajo se encuentra en el rango de 2.4 GHz a 2.48 GHz con amplio espectro que permite una transmisión Full-Duplex, síncrona o asíncrona dependiendo de los dispositivos que estén interconectados. La potencia de salida que se requiere para transmitir a una distancia máxima de 10 m es de 0dBm (1 mW) mientras que la versión de largo alcance transmite entre los 20 y 30 dBm (entre 100 mW y 1 W).

Con los sistemas Bluetooth se puede crear una red de área personal (PAN), o también llamada Piconet, dentro de su rango de alcance de hasta 8 dispositivos donde uno de ellos es designado como máster y los otros 7 esclavos. Existe la posibilidad de aumentar el alcance de la red si se asigna a un dispositivo esclavo otra Piconet ya que un dispositivo Bluetooth puede servir a más de un maestro. La técnica que utiliza Bluetooth para evitar interferencias entre dispositivos es el *Spread-Spectrum frequency hopping*. En esta técnica, un dispositivo utiliza 79 frecuencias individuales escogidas al azar dentro de un rango designado (en este caso la banda ISM) cambiando de una a otra a una cadencia de 1600 veces por segundo, de esta forma es muy improbable que dos transmisores estén a la misma frecuencia en el mismo tiempo. Una vez establecida una Piconet, los miembros de esta saltan de frecuencia al azar al unísono evitando así interferencias con otros sistemas similares vecinos.

El objetivo de bajo consumo y bajo coste se ha conseguido integrando toda la estructura del sistema en un solo chip elaborado con tecnología CMOS. De esta manera se consiguió elaborar una solución de tamaño 9 x 9 mm y que consume aproximadamente un 97% menos de energía que un teléfono celular común. Todo ello ha permitido que todos los dispositivos móviles actuales del mercado incorporen esta tecnología.

#### SISTEMAS DE AUDIO INALAMBRICOS:

Actualmente son muy comunes los sistemas Bluetooth “manos libres” (Parrot CK30XX) que permiten interconectar los móviles al sistema de audio de los coches permitiendo así realizar y recibir llamadas sin necesidad de manejar el dispositivo.

También encontramos en el mercado, infinidad de auriculares Bluetooth que con la ayuda de un pequeño adaptador nos permiten recibir la señal audio de diferentes dispositivos como televisores o equipos.

### 2.1.2. NFC

Son las siglas de *Near Field Communication* y es una de las tecnologías que en la actualidad está teniendo más popularidad. Sus características técnicas basadas en la norma ISO 14443 y el estándar NFCIP-1 lo hacen especialmente apropiado para aplicaciones de seguridad e implementación en el campo del comercio electrónico. Es un sistema de muy corto alcance, menos de 10 cm, que necesita por lo tanto que los terminales estén prácticamente en contacto. A diferencia de las otras tecnologías,

esta trabaja en la banda de los 13,56 MHz pudiendo alcanzar velocidades de 106, 212, 424 o 848 Kbit/s según del entorno en el que se trabaje.

NFC no está pensada para utilizarse en una transmisión masiva de datos, como puede darse con las tecnologías wifi o Bluetooth. Solo sirve para el intercambio rápido de unos pocos bits de información, lo justo para que identifique y valide al usuario.

La implementación de NFC en los dispositivos móviles requiere, como en todas, de la integración de un chip que soporte esta tecnología. En la actualidad, son varios los fabricantes que han apostado e incorporado de serie a sus modelos de móvil esta tecnología, entre ellos: Google (Nexus S), Samsung (Galaxy S II), LG (Optimus NET NFC), Nokia (6131, C7, 700, 701 ), o BlackBerry (Curve 9380). De estas compañías y de las diversas operadoras de telefonía dependerá el futuro y consolidación de esta tecnología.

A continuación citaremos algunos usos de sistemas con NFC:

#### TRANSPORTE PUBLICO:

Desde 2007 existe un proyecto pionero en el metro de Londres que, gracias al acuerdo entre las diversas operadoras de telefonía, permite el pago tanto del billete de transporte como la compra en las diferentes tiendas de las estaciones de metro con el saldo que dispone el usuario en su teléfono móvil.

En España, las empresas de transporte municipal de autobuses de ciudades como Madrid o Málaga incorporan módulos Near Field Communication , utilizando los actuales sistemas de validación sin contacto de sus autobuses, que permiten realizar el pago de los billetes directamente con el teléfono móvil permitiendo así que el usuario pueda olvidarse de llevar dinero en efectivo, tarjetas o similares.

#### COMERCIO:

En Septiembre de 2011 se alcanzó un gran acuerdo entre la empresa Visa y diversas operadoras de telefonía para potenciar el comercio electrónico. El sistema permite que una compra sea cargada a la cuentas del usuario con tan solo acercar el smartphone al terminal instalado en el establecimiento que es quien se encarga de autenticar el usuario. Esta especie de monedero electrónico permite por tanto tener un medio de pago y cobro asociado al número de teléfono móvil.

En la actualidad los prototipos de estos sistemas están instalados en diferentes comercios de diversas ciudades como Sitges, Madrid o Barcelona.

### **2.1.3 ANT / ANT +**

Es la tecnología inalámbrica diseñada y comercializada por la filial de Garmin, Dynastream Innovations Inc. Trabaja también en la banda de los 2.4 GHz y esta principalmente pensada para sistemas de sensores y receptores por sus características. A diferencia del Bluetooth este protocolo de transferencia de datos está pensado para realizar envío de una pequeña cantidad de datos pero de forma continua.

Su principal característica es su bajo consumo, fundamental para la autonomía de equipos inalámbricos, ya que es capaz de aguantar mucho tiempo en modo de espera sin consumir apenas batería. Se estima que con una pila de tipo 2032 se pueden mandar durante 4 años pequeños mensajes cada 2 segundos. Su alta tolerancia a interferencia permite interconectar hasta 65000 sensores del mismo tipo.

Su reducido tamaño ha propiciado la utilización especialmente en equipamiento deportivo (pulsómetro, relojes, dispositivos GPS, medidores de distancia) y en la industria de la automoción.

La firma comercial Nike tiene una amplia gama de productos que incorporan esa tecnología. Nike + Training es un sistema de sensores que se incorpora en algunos modelos de zapatillas deportivas y que permite compartir datos con cualquier iPhone o iPod Touch. El acelerómetro y el sensor de presión que incorpora permite registrar datos como la distancia recorrida, velocidad o número de pasos.

#### **2.1.4 ZigBee**

En una tecnología diseñada por ZigBee Alliance basada en el estándar IEEE 802.15.4 (de redes inalámbricas de área personal, WPAN) y que se desarrolló en 1998 a la par de las tecnologías Bluetooth y Wifi para cubrir el hueco de necesidades que dejan estas tecnologías. La implementación de este protocolo de red, lo hace idóneo para conexiones punto a punto y punto a multipunto con una baja transferencia de datos y que requieran de un bajo consumo energético.

En concreto tiene un consumo de 30 mA transmitiendo y de 3 uA en reposo frente a los 40 mA transmitiendo y 0,2 mA en reposo que tiene el Bluetooth. Además destaca por la sencillez de diseño e implementación.

También transmite en la banda de los 2'4 GHz y su rango de acción oscila entre los 10 y 75 metros dependiendo del entorno donde se utilicen pudiendo transmitir datos a velocidades de 250 Kbps.

Como ya se ha comentado, los protocolos ZigBee están definidos para su uso en aplicaciones con requerimientos muy bajos de transmisión de datos y consumo energético. Su ámbito de uso es variado como en sistemas domóticos, redes de seguridad, control industrial, control de sensores empotrados, etc..

#### **2.1.5 Infrarrojos**

La tecnología infrarroja (IR) fue de las pioneras en la comunicación inalámbrica de dispositivos a corta distancia. Basada en radiocomunicaciones en la zona del infrarrojo, por encima del espectro visible, permite una comunicación bidireccional a velocidades que oscilan entre 9600 bps y 4 Mbps. Esta tecnología fue muy extendida a finales de los años 90, principios del 2000, en distintos dispositivos como teléfonos móviles, controladores remotos de televisores, DVD, teclados de ordenador, etc..

A diferencia de los anteriores sistemas, se requiere una visión directa entre el dispositivo emisor y receptor para poder realizar la comunicación y que esta no supere el metro de distancia.

La utilización de esta tecnología ha caído en desuso a favor también de Bluetooth.

## 2.2. Sistemas operativos en dispositivos móviles

El hecho de que los smartphones han revolucionado el mercado de la telefonía móvil es una realidad. En pocos años se ha pasado de considerarse un juguete caro o un producto reservado a los fanáticos de Apple a considerarlo un instrumento cotidiano de comunicación, o herramienta de trabajo en algunos casos, en todo el mundo.

Analizando los diversos estudios realizados por compañías especializadas del sector como ComScore, VisionMobile o Gartner extraemos que en el tercer cuatrimestre de 2011 la penetración de los smartphones, con respecto a los terminales que carecen de acceso a internet, en el mercado mundial de la telefonía móvil es del 29%. Diferenciando por territorios, este mismo estudio de VisionMobile (Noviembre 2011), estima que la cuota de mercado es del 65% en EEUU, cerca del 50% en Europa y del 17% en Latino América sobre la venta de terminales móviles. En China, país del mundo con mayor población con 1332 millones de habitantes, 900 millones posee un teléfono móvil y casi un 28% son usuarios de un Smartphone.

Sin duda estos grandes datos de penetración están muy ligados a la aceptación que está teniendo en el consumidor estos productos. El gran número de fabricantes que han apostado por este tipo de teléfonos móviles ha permitido una reducción considerable en el precio de comercialización de estos terminales ofreciendo diferentes gamas y modelos al alcance de todo el mundo. Las alianzas entre fabricantes y operadoras es otro de los factores a tener en cuenta en la reducción de precios. En la actualidad podemos encontrar modelos desde 60€ los más económicos hasta llegar a los 600€.

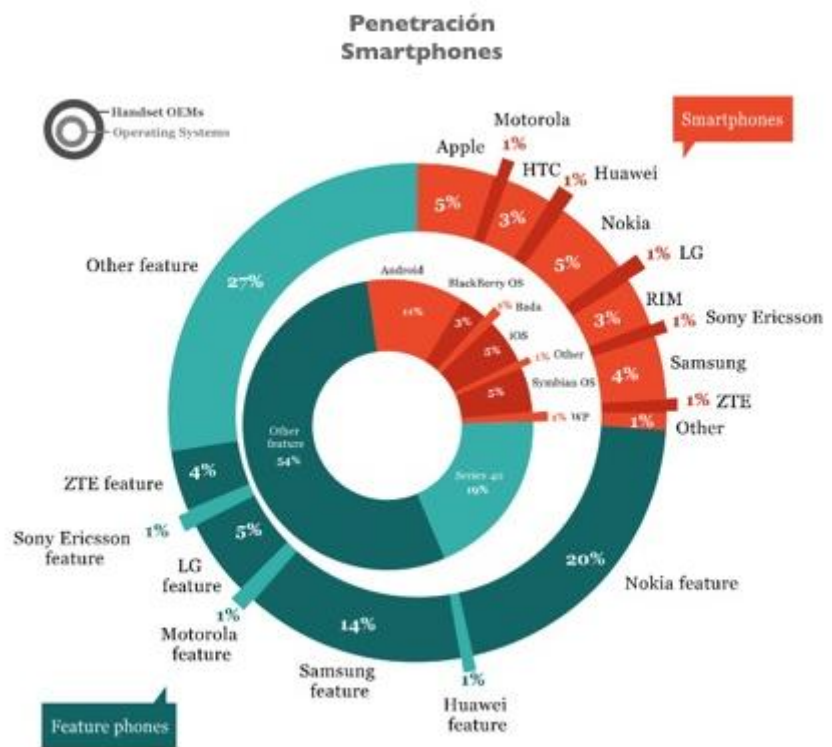


Figura 2.2 Penetración mundial de smartphones (Fuente: comScore)

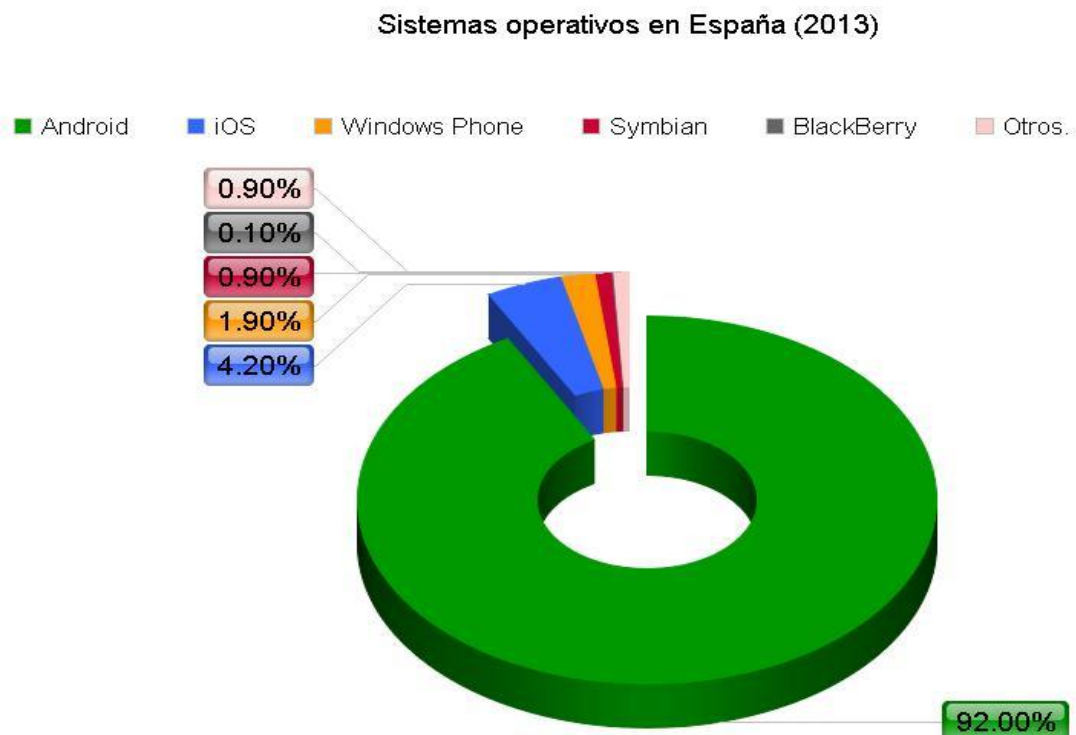
Son numerosas las cifras, números, estadísticas según la región que se analice, es por ello que nos centraremos en la situación y tendencias en los principales mercados europeos como son Alemania, Francia, Italia, España y Reino Unido.

El informe “Spain Digital Future in Focus” publicado en abril de 2013 por comScore indica que España se ha convertido en el país europeo con mayor penetración de smartphones con una tasa del 66% por encima de la media del 57% que marcan el conjunto de países europeos anteriormente mencionados.

En cuanto a como queda la distribución de los distintos sistemas operativos, según el estudio de Kantar Worlpanel ComTech, en 2013 España se consolidó como líder de Europa por lo que hace a la



mayor proporción de smartphone Android en el mercado con una tasa del 92%. Los porcentajes de los demás sistemas operativos se distribuyen con un 4,2% para iOS, un 1,9% WindowsPhone, un 0,9% Symbian y un 0.1% BlackBerry, la gran perjudicada.



*Figura 2.3 Sistemas operativos en España (Fuente: elaboración propia)*

Destacamos el gran éxito del sistema Android en comparación con los demás países europeos donde la tasa de este sistema acapara el 70,4% de la cuota de mercado, iOS un 17,8% y WindowsPhone un 6.8%. En EEUU los porcentajes se dividen entre Android con un 52% y iOS con un 41,9%, mientras que en Japón iOS es líder con un 49,2% frente al 45.8% de tasa de Android.

Recordemos que un Smartphone es un “teléfono inteligente” que nos permite realizar muchas más cosas a parte de llamar, enviar mensajes, hacer fotos o escuchar música. Dependiendo del sistema operativo que elijamos, podremos desde navegar por internet o administrar nuestras cuentas de correos hasta comprar unos billetes de avión o controlar las cámaras de seguridad de nuestro domicilio. Las diversas aplicaciones son ofrecidas por los sistemas operativos en los llamados Markets o Stores (tiendas virtuales) muchas de ellas de manera gratuita. Sin duda las diversas aplicaciones dotan al teléfono de un uso que más allá de la simple comunicación, ahora se convierte en una herramienta más de trabajo y ocio.

Desde este punto de vista de número de aplicaciones, sin ninguna duda los dos sistemas operativos líderes en el mercado tanto por ingresos como por número de aplicaciones son el sistema iOS de Apple y Android. En 2013 se cifró en cerca de 800.000 las aplicaciones disponible en Google Play (plataforma utilizada para comercializar las aplicaciones Android) y en 775.000 las aplicaciones lanzadas distribuidas Apple App Store (exclusivo para iOS en los iPhones), siendo en ambos casos entorno al 85% de ellas gratuitas.

Plataforma	Numero de aplicaciones
Android	800.000
iOS	775.000
Windows Phone	120,00
BlackBerry OS	150.000
Symbian	25.000
Bada	15.000

Tabla 2.1 Número de aplicaciones móviles (Fuente: elaboración propia)

A continuación realizaremos un breve estudio de los principales sistemas operativos que existen en el mercado, analizándolos tanto desde el punto de vista comercial como de desarrollo de aplicaciones. Esto nos servirá para justificar la elección tomada para la implementación de nuestra aplicación del proyecto.



## iOS

Es el sistema operativo de Apple Inc. y que fue introducido en su primer iPhone en enero de 2007. Este sistema no es más que una versión reducida del sistema Mac OSX para PC pero aplicada a su Smartphone y que también puede funcionar en otros dispositivos de Apple como son iPod Touch, iPad o Apple Tv.

Destaca por ser un sistema muy estable, intuitivo y fácil de usar. Su elegancia y su cuidado estilismo, tanto en las aplicaciones como en su diseño, lo hacen ser sin duda unos de los productos más deseados y con más adeptos. Con su AppStore, se convirtió en el primer sistema operativo en disponer de un market a gran escala en el que existen miles de aplicaciones.

El gran inconveniente del iPhone es su elevado coste y la exclusividad de sus productos. No podemos instalar iOS en ningún terminal que no sea Apple (ni Samsung, ni Nokia...). Depende de un ordenador con iTunes instalado para poder realizar acciones como la configuración inicial, pasar contenido multimedia al móvil o actualizaciones.

Las aplicaciones para iPhone se programan en xCode a través de un SDK (Software Development Kit) fácil de utilizar que dispone de un emulador de teléfono integrado. Para ello necesitamos subscribirnos *iPhone Development Program*, licencia de desarrollador con un coste de 99\$ anuales. Además sólo podremos diseñarla a través de un Mac. Tampoco podemos instalar ninguna aplicación que no sea compilada en código nativo.

Actualmente está disponible la versión iOS 5.0.



## Android

Android es el sistema operativo de Google, lo cual permite la gestión en los smartphones que utilizan este sistema de productos muy útiles de la misma firma como Gmail, calendarios, gestión de contactos o Google Maps Navigator de forma sencilla.

Principalmente destaca por tratarse de un sistema abierto lo cual permite que cualquier fabricante pueda amoldarlo a sus terminales y desarrollar en él sus productos, de ahí que las actualizaciones del software dependan de los mismos (Samsung, Lg, HTC, etc...) y no del propio Google.

Visualmente es un sistema que guarda un cierto parecido de estilo con iOS pero mucho más personalizable.

Android, es líder del mercado en cuanto a descarga de aplicaciones. Para gestionar la instalación y descargar de estas aplicaciones podemos utilizar la plataforma Google Play (antiguamente conocida como Android Market) o bien instalarlas directamente en el terminal ya que este sistema operativo permite instalar libremente programas adquiridos de internet .

La facilidad con la que se pueden transferir ficheros entre el ordenador y el terminal móvil es otra de sus bazas. A diferencia de otros sistemas en los que se requiere de un software de sincronización específico, con la simple conexión del terminal al puerto USB del pc es suficiente para poder acceder a todos los recursos y contenidos del dispositivo.

Desde el punto de vista de desarrollo, programar para Android resulta mucho más sencillo. Dispone de una máquina virtual, *Dalvik* (SDK) , fácil de instalar y utilizar en cualquier plataforma : Windows, Linux o Mac. Esta máquina está basada en la actual *JavaRunTimeMachine*, que es la encargada de ejecutar aplicaciones Java, por lo que el lenguaje de programación para las aplicaciones se asemeja bastante a Java. De hecho se dispone nativamente de todas las librerías de Java más las librerías que Google incorpora de su parte.

También se dispone de una web oficial para Android ([developer.android.com](http://developer.android.com)) donde los desarrolladores pueden encontrar abundante documentación, software, tutoriales y ejemplos de diferentes tipos de funcionalidades que permiten la rápida familiarización con este lenguaje, además de infinidad de ejemplos en la red, foros de discusión y páginas web especializadas en desarrollo que sitúan a este sistema operativo como el más utilizado y preferido por los desarrolladores de aplicaciones móviles. Tampoco es necesario invertir en licencias para programar y cargar aplicaciones en el Google Play.

Actualmente en el mercado encontramos terminales que incorporan las versión de Android 2.0 hasta la 4.0, conocida como *Ice Cream Sandwich*, en los dispositivos de gama más alta.

## Symbian

Este es el sistema operativo de Nokia. A día de hoy continua siendo de las plataforma móvil más extendida en el mundo ya que durante años, Nokia ha sido y es el fabricante de dispositivos móviles, no solo smartphones, que más vende en el mundo con un 23,9 % de cuota de mercado.

El sistema operativo es correcto y fácil de manejar pero muy por debajo del nivel de los anteriormente citados sobretodo en cuanto a número de aplicaciones (OVI, es la tienda de aplicaciones) y administración de correo electrónico. Sin ninguna duda estas carencias han sido detectadas por el equipo de Nokia y han reaccionado dando a lugar a una gran alianza con Microsoft, propietaria de Windows Phone, para incorporar este sistema operativo en sus Smartphone. Todo esto hace pensar que en un futuro no muy lejano el sistema Symbian quedará relegado a los terminales de gama baja de Nokia.

## **Windows Phone**

Es el sistema operativo para móviles de Microsoft que antiguamente se llamaba Windows Mobile. Son muchos los fabricantes que incluyen esta plataforma en sus terminales, HTC, Samsung, LG, etc., y próximamente en Nokia.

Destaca por haber apostado en su última versión, Windows Phone Mango 7.5, por la sincronización de todos sus productos en el llamada Nube o Cloud-Computing como ha hecho también Apple. Esto va a permitir compartir recursos con los ordenadores personales de forma fácil a los usuarios de Windows. Su tienda de aplicaciones llamada Market Place es una de las nuevas pero más logradas, por la facilidad de navegación y detalles de las aplicaciones, pero se queda corta en cuanto a número de programas.

Uno de los principales inconvenientes de esta aplicación es la incompatibilidad con Outlook y la necesidad de instalar un software específico en el ordenador (Zune) para poder transferir fichero en el dispositivo móvil.

Desde le punto de vista de desarrollo de aplicaciones, es necesario utilizar Windows con Visual Studio en sus versiones de pago para poder compilar cualquier aplicación. Además su API es bastante pobre lo que hace que sea muy laborioso conseguir aplicaciones visualmente atractivas.

Con perspectiva de futuro destacaremos nuevamente la alianza con Nokia ya que sin dudad, con una buena línea estratégica, esta alianza podrá competir con Android y iOS.

## **Blackberry OS**

Comentaremos las principales características de este sistema operativo por la influencia que tiene en el mercado actual pero la descartaremos como plataforma para realizar nuestra aplicación ya que las características que tiene los terminales móviles donde reside, carecen de pantalla táctil (ya empiezan a aparecer modelos como el Blackberry Torch 9800 que si la incorporan) para interactuar con el usuario fácilmente como requerimos.

Es la plataforma de la empresa RIM y está pensado especialmente para dar servicio a empresas y profesionales. Como ya hemos comentado, su teclado físico es característico y lo hace muy cómodo para escribir correos electrónicos y utilizar servicios de mensajería instantánea.

Como puntos débiles destacaremos la complejidad en su configuración y la falta de aplicaciones, tanto en número como en calidad, en su tienda App World.

## Bada

Por último hablaremos de Bada que es un sistema operativo que fue creado por Samsung exclusivamente para sus terminales de la serie Wave.

Tiene muy pocas prestaciones que podamos destacar salvo el precio de los terminales smartphones donde reside. Junto a iOS es el único compatible con Outlook permitiendo así la administración de diversas cuentas de correo. Su tienda de aplicaciones es de mala calidad y existen muy pocas aplicaciones. Es un sistema inestable, incompatibles con muchos de los formatos de video y con muchos errores en el manejo de correos electrónico.

Para concluir este análisis sobre los sistemas operativos, vamos a analizar uno de los informes trimestrales publicados en 2013 por Gartner, una de las consultoras de investigación de las tecnologías de la información referentes. El informe sobre ventas de dispositivos y tendencias futuras que publica de manera trimestral nos darán una visión a corto-medio plazo sobre la evolución de los mercados.

Los datos de *Gartner* sobre ventas de smartphones en el mundo durante el segundo trimestre de 2013 muestran que el 79% de los teléfonos vendidos incorporaban el sistema operativo Android en comparación a un 64,2% aparecido en el mismo periodo del año anterior. Esto significa que de las 225,000 unidades de smartphones vendidas, casi 178,000 pertenecieron al sistema operativo de Google. El 20% restante se compone de un 14,2% del sistema iOS, un 3,3% para Microsoft, 2,7 para BlackBerry y el punto porcentual restante entre otros sistemas.

Table 2

Worldwide Smartphone Sales to End Users by Operating System in 2Q13 (Thousands of Units)

Operating System	2Q13 Units	2Q13 Market Share (%)	2Q12 Units	2Q12 Market Share (%)
Android	177,898.2	79.0	98,664.0	64.2
iOS	31,899.7	14.2	28,935.0	18.8
Microsoft	7,407.6	3.3	4,039.1	2.6
BlackBerry	6,180.0	2.7	7,991.2	5.2
Bada	838.2	0.4	4,208.8	2.7
Symbian	630.8	0.3	9,071.5	5.9
Others	471.7	0.2	863.3	0.6
<b>Total</b>	<b>225,326.2</b>	<b>100.0</b>	<b>153,772.9</b>	<b>100.0</b>

Source: Gartner (August 2013)

Tabla 2.2 Informe de ventas mundiales de Smartphone (Fuente: Gartner)

Destacar sobretodo la continua subida de Android años tras año que la consolida como número uno en cuanto a sistemas operativos. También podemos destacar de la tabla como Windows Phone consiguió superar a BlackBerry por primera vez, alcanzando el número 3 del mercado. Sin duda una señal más del porvenir y deterioro de este tipo de terminales.

Con todo ello y viendo que también se cumplen todas las cualidades y facilidades desde el punto de vista del desarrollo de aplicaciones, sin duda justificaremos así la elección de la plataforma Android de Google para realizar la aplicación del proyecto BlauTic.

## 2.3. Análisis de los Smartphones del mercado.

Desde la aparición en 2007 del iPhone de Apple, que sin lugar a dudas marcaría la tendencia a seguir en cuanto a contenidos y especificaciones de los smartphones, la evolución y oferta ha sido muy rápida. La aparición de un numeroso grupo fabricantes como Samsung, HTC, Motorola, LG, Huawei, Dell, Sony-Ericsson, apostando por la plataforma Android de Google, han hecho la competencia a la firma Apple (que a día de hoy continua siendo el principal fabricante de smartphones, con una cuota del 28%) y reduciendo considerablemente sus precios de venta. Hoy en día se pueden adquirir terminales libres a partir de 80€ (Huawei U8100) o si se prefiere, terminales de gama alta como El Samsung Galaxy II (475€) que vendría a competir con el iPhone 4s (600€).

En cuanto a Nokia, que ha sido de siempre el líder mundial de ventas de terminales de telefonía, ha tardado en reaccionar bastante en este mercado y su apuesta de futuro, como ya hemos comentado, viene liderada de la mano de Microsoft con el lanzamiento de la plataforma Windows Phone 7.

En lo que nos concierne a nosotros para la ejecución de nuestro proyecto, analizaremos los terminales que existen a día de hoy o que van a ser lanzados próximamente en el mercado capaces de soportar la tecnología Bluetooth 4.0.

### Motorola RAZR

El Motorola Razr salió al mercado estadounidense en octubre de 2011 y es el primer smartphone que presenta esta marca tras haber sido adquirida por Google. Está compuesto por un procesador ARM Cortex A9 OMAP 440 Dual Core a 1.2GHz con 1GB de memoria RAM y 16GB de memoria interna para almacenamiento. El sistema operativo que integra es Android en su versión 2.3.5 Gingerbread. Su precio es de unos 300\$ y ha sido el primer terminal que hemos podido adquirir con tecnología Bluetooth 4.0.

### HTC Edge

El lanzamiento de este terminal de la firma Taiwanesa HTC está previsto para el primer trimestre de 2012. Será el primer dispositivo con un procesador de cuatro núcleos Nvidia Tegra 3. Además de la tecnología Bluetooth 4.0, incorporará un sistema de sonido Beats Audio (sistema de sonido digital) y conectividad LTE (*Long Term Evolution*). Previsiblemente el sistema operativo que incorpore será el nuevo Android 4.0 *Ice Cream Sandwich*.

### iPhone 4s

Es el último smartphone de Apple a día de hoy. Con su nuevo procesador de doble núcleo Apple 5 y el sistema iOS 5, este terminal se equipara en potencia los últimos modelos de Android. Su precio va desde los 200\$ hasta los 500\$.

De forma semejante con otros productos de la firma como el MacBook Air y el Mac mini, incorpora la tecnología Bluetooth 4.0 en un chip dual que le permite compatibilizarse con el protocolo actual cuando sea necesario.

## NEC Medias PP N-01D y LTE N-04D

Estos terminales cuya comercialización de momento sólo se realiza en Japón también incorporan Bluetooth 4.0. Como curiosidad, estos dispositivos se sincronizan a través de esta tecnología con unos relojes de la firma Casio permitiendo (los G-Shock GB-6900) permitiendo en ellos la monitorización de llamadas entrantes, correos electrónicos incluso la lectura de mensajes de texto.

## LG X3

Ya se están filtrando las características de este novedoso smartphone de la firma LG cuya salida al mercado está prevista para el segundo trimestre de 2012. Destaca por los demás por su potente procesador Tegra 3(mismo que el HTC Edge) y las últimas tecnologías de NVIDIA.

Contará con conectividad Wi-Fi, Bluetooth 4.0, compatibilidad con redes HSPA, LTE y seguramente incorporará un chip NFC. Esta previsto que aparezca equipado con Android 4.0.

## 2.4. Proyectos similares

Debido a la naturaleza de este proyecto, encontramos numerosas soluciones en la actualidad que utilizan en mayor o menor grado algunos de los aspectos técnicos y tecnológicos que vamos a implementar como son: control remoto de dispositivos, monitorización de sensores, bluetooth o comunicaciones inalámbricas.

Desde 2011 encontramos en el mercado multitud de aparatos domésticos y electrodomésticos programables con algún tipo de tecnología inalámbrica que permiten su control remoto. Una muestra clara de la proliferación de estos dispositivos la podemos ver con la firma comercial LG que a principios de 2013 presentó “el nuevo concepto de casa inteligente: Smart Home de LG”.



Figura 2.4 Esquema Smart Home (Fuente: LG)

Mediante la combinación de la tecnología NFC, WIFI o 3G/4G y la utilización de la aplicación Smart Share nos permite controlar desde el dispositivo móvil el funcionamiento de la lavadora, revisar que tenemos en la nevera o compartir fotos, videos o cualquier otro tipo de contenido de forma casi instantánea con el ordenador, el Smartphone o el televisor.

Como ejemplo de aplicaciones o gadgets que combinan la tecnología inalámbrica con sistemas de sensores, en este caso para la monitorización de las constantes vitales básica, quería hacer mención a la gama de relojes **Motorola Motoactv**. Fue el primer modelo de reloj pulsómetro en incorporar GPS y reproducción de música en un mismo dispositivo.

Muy utilizado por amantes al deporte del “running”, gracias al sensor incorporado podemos monitorizar en la pantalla nuestra frecuencia cardiaca así como controlar nuestro ritmo de carrera con la información recogida en el GPS. Además, incorpora una unidad almacenamiento extra para la reproducción de música la cual podemos transferir al dispositivo bien con WIFI o bluetooth.



Figura 2.5 Reloj Motoactv (Fuente: Motorola)

Destacar el éxito y la aceptación popular de este tipo de gadgets “Health and Fitness”, que empezaron a darse a conocer principios de 2012, que ha permitido que hoy en día existan infinidad de modelos y diseños diferentes a precios muy competentes.

Aquí se ha mencionado un par de proyectos como ejemplo de integración de diversas tecnologías. Si concretamos algunos con el uso de la tecnología bluetooth encontramos infinidad de dispositivos que permiten una interconexión rápida y sin cables como: auriculares para la reproducción de música, periféricos informáticos como impresoras o webcams, dispositivos de control remoto de robots, smartwatches, etc...

### 3. APLICACIÓN: Plataforma BlauTic

Como ya hemos indicado en la introducción de esta memoria, el objetivo de este proyecto será realizar la aplicación software para dispositivos móviles, smartphones, del proyecto general BlauTic. Esta aplicación Android permitirá gestionar, mediante comunicaciones basadas en tecnología inalámbrica, información suministradas por los dispositivos hardware. La información se tratará localmente y se enviará mediante interfaz IP basada en 3G o Wifi a un punto central donde podrá ser gestionada y almacenada conveniente.

Esta aplicación soportará las siguientes características comunes a todos los software englobados dentro del proyecto BlauTic:

- Comunicaciones inalámbricas mediante Bluetooth 4.0.
- Gestión de base de datos localizada en el dispositivo móvil. Esta base de datos debe ser descargada con una configuración inicial: interfaces del sistema, dispositivos del sistema, tipos de mensajes disponibles, sensores existentes.
- Características visuales (logos, imágenes, aspecto gráfico...) completamente separadas del código que implementa la funcionalidad y que doten a la aplicación de un manejo al alcance de todos los usuarios de forma sencilla e intuitiva.
- Comunicaciones IP a través de interfaz Wifi o 3G.
- Soporte a múltiples idiomas.
- Descarga remota del paquete software e instalación sobre el dispositivo.
- Detección de versiones actualizables y control de compatibilidad.

En los sucesivos puntos de este apartado se explicará la metodología utilizada para diseñar esta aplicación en base a las especificaciones suministradas. Describiremos mediante bocetos el aspecto visual aproximado que tendrán las diversas pantallas así como la funcionalidad de estas. Justificaremos la elección de las tecnologías empleadas para cumplir con los aspectos técnicos haciendo sobretodo un análisis un poco más profundo sobre el sistema operativo Android.

#### 3.1 Modelo de casos de uso

Para entender en poco mejor la funcionalidad de esta aplicación citaremos a continuación un par de situaciones reales donde podríamos utilizar este proyecto.

##### **Educación:**

Se estudia una hipotética situación dentro del ámbito escolar, donde cada alumno es propietario de un dispositivo BlauTic y donde el profesor/tutor de los alumnos accede a la información de cada uno de los dispositivos desde la aplicación residente en su Smartphone. Además los padres, también desde sus respectivos terminales móviles, tienen acceso a la información que contiene el dispositivo Blautic que llevan consigo sus hijos.

De esta forma el profesor, estando todos los alumnos en la clase, tendría un listado de todos los dispositivos BlauTic asociados en su sistema personalizándolos con sus alumnos. Haciendo uso de la funcionalidad de mensajes, podría notificar a los padres cualquier tipo eventualidad o requerimiento (como citas de tutoría, notificaciones académicas, próximos eventos...) ya que esta información sería recogida en los terminales móviles de los padres automáticamente al llegar los alumnos a casa. Del



mismo modo, un padre puede dejar un mensaje horario al profesor por ejemplo recordándole que debe tomar cualquier tipo de medicamento cada cierto tiempo.

Otro uso, en este caso de la funcionalidad de seguridad de la aplicación, permitiría establecer un radio máximo de cobertura de los dispositivos (entre alumnos y profesor) saltando una alarma en el smartphone del profesor en el caso de que algún alumno se alejara más de la distancia establecida. Esta posible situación podría darse en alguna salida extraescolar.

### **Mantenimiento de equipos:**

La siguiente situación se recrea en un ambiente industrial, donde se ha incorporado a cada máquina un dispositivo BlauTic. El operario de turno, tiene distintos sistemas asignados en su aplicación dependiendo de la zona o hangar que esté controlando en ese momento y de las máquinas dentro de su radio de cobertura que en este caso vendrá marcado por el bluetooth del smartphone.

Con la funcionalidad de sensores podrá controlar a tiempo real la temperatura de las máquinas, así como activar o desactivar con el sensor de salida digital alguna funcionalidad de esta previamente configurada. En el histórico de sensores, podríamos registrar información útil que posteriormente podría ser enviada mediante Wifi a un servidor remoto para un posterior procesamiento.

Con la función de mensajes, se podrían dejar instrucciones concretas en cada una de las máquinas que serían leídas por el operario del siguiente turno de trabajo.

## **3.2. Especificaciones**

En este apartado se definirán las especificaciones concretas de la aplicación “Plataforma BlauTic” a ejecutar sobre los dispositivos móviles, smartphones.

La aplicación estará diseñada para detectar dispositivos BlauTic A1/B1 al alcance y poder recuperar información de mensajes almacenados y lecturas online y/o guardadas en sensores del sistema. La aplicación permitirá enviar los datos a una aplicación remota a través de interfaces IP. El software también permite detectar dispositivos cercanos y monitorizar su presencia dentro de un determinado radio.

### **3.2.1. Funciones principales**

La siguiente lista enumera las funciones básicas a cumplir por la aplicación agrupadas en seis bloques donde indicaremos las características más importantes. Posteriormente serán abordados con más detalle los aspectos técnicos de los mismos:

- Configuración de parámetros generales:
  - Modificación de los parámetros de funcionamiento de la propia aplicación almacenados en base de datos local.
  - Desde el móvil se podrá leer/modificar los datos de los parámetros de configuración del dispositivo BlauTic del perfil General: versión de firmware, Dirección MAC del master,...
- Listar dispositivos en el sistema identificando con símbolos/colores/texto su estado actual: en cobertura, mensajes pendientes, baja batería, sensores habilitados,...



- Mensajes:
  - Avisar si dispositivo tiene mensaje pendiente lectura
  - Leer mensajes dispositivos - Contestar/Confirmar
  - Escribir mensajes dispositivos: texto, horario
  - Avisar si vence un mensaje horario
  - Descargar mensajes a un servidor por Bluetooth3.0
  - Reenviar mensajes por 3G/Wifi a un servidor
  
- Sensores:
  - Configurar medidas de sensores:
    - Habilitar/deshabilitar sensores, frecuencia de medida, frecuencia de envío o envío ante evento: cobertura, movimiento, ..
  - Generación de mensajes si supera umbrales
  - Mostrar valores sensados por dispositivos: digitales y analógicos ...
  - Descargar valores a un servidor por Bluetooth3.0
  - Enviar valores por 3G/wifi a un servidor central.
  - Leer histórico de valores
  
- Seguridad:
  - Generar señal sonora/vibración/sms si hay evento en:
    - Monitorizar dispositivos que abandonan cobertura (tiempo asociado)
    - Monitorizar dispositivos que se mueven (poder concretar movimiento)
  
- Sincronización:
  - Se deberá establecer una comunicación IP a través de interfaz WiFi o 3G con un servidor remoto.

### 3.2.2. Función: Configuración

Esta parte de la funcionalidad tiene como objetivo permitir la configuración de los parámetros de funcionamiento de la aplicación y de los dispositivos.

A continuación se describen las dos acciones posibles a realizar dentro de este bloque de funciones:

- Aplicación: Modificar la configuración base de la propia aplicación
- Dispositivo: Leer/Modificar parámetros generales del dispositivo

#### Opción aplicación:

Los valores de configuración de la aplicación se guardarán en la base de datos interna que se generará en el momento de la descarga e instalación del software. Esta base de datos incorporará los parámetros por defecto entre los que se encuentra los permisos de modificación de parámetros: direcciones MAC bluetooth, de los usuarios master. Además, también se incorporará la información necesaria para identificar todos los interfaces y dispositivos BlauTic que pertenecen a nuestro mismo sistema.

A continuación se describirán los parámetros que debe incluir esta parte de la aplicación:

#### GENERALES

- Número de sistema al que pertenece la aplicación. Valor de lectura. Únicamente modificable como usuario master.
- Versiones de firmware compatibles Valor de lectura

## ENVÍO DE MENSAJES

- Activar descarga de mensajes por bluetooth3.0. La activación de este parámetro indica al interfaz móvil que debe consultar periódicamente sobre la presencia de la MAC destino y en caso de éxito volcarle la información solicitada.
- MAC bluetooth3.0 destino
- Activar envío de mensajes por 3G/Wifi. La activación de este parámetro indica al interfaz móvil que debe enviar a través del interfaz IP la información almacenada al servidor central.
- Dirección del servidor remoto Datos IP del servidor remoto.

## AVISOS

- Activar aviso por dispositivo en cobertura con avisos de sistema: mensaje pendiente y baja batería. Los tipos de aviso son: notificación por icono, sonido, vibración, sms, llamada,... Para sms y llamada hay que configurar el número destino. Los avisos de sistema son comunicados en el paquete SCAN\_RSP y son los siguientes:
  - MemPending
  - PowWarn
- Activar aviso vencimiento de mensajes horarios. Los tipos de aviso son: notificación por icono, sonido, vibración, sms, llamada,... Para sms y llamada hay que configurar el número destino.
- Activar aviso de dispositivo monitorizado fuera de cobertura: Los tipos de aviso son: notificación por icono, sonido, vibración, sms, llamada,... Para sms y llamada hay que configurar el número destino.

### Opción Dispositivo:

Los parámetros del dispositivo pertenecientes al profile General se leen tras establecerse una conexión siempre que el firmware del dispositivo sea compatible con el software y el dispositivo pertenezca al mismo sistema que la aplicación. Los datos que permiten decidir si solicitar una conexión -CONN REQ- se obtienen de la trama de SCAN\_RSP como respuesta a la trama de escaneo SCAN\_REQ:

- GenSystemNumber: Número de sistema
- GenFirmware: Versión de firmware

Los parámetros se guardan en el dispositivo y se pueden leer/modificar una vez realizada la conexión por petición del interfaz móvil.

Este apartado incluirá:

- GenIdNumber Número de serie que identifica al dispositivo. Solo Lectura
- GenName Nombre significativo del dispositivo. Modificable por Master.
- GenDateTime Fecha y hora del dispositivo. Modificable por Master
- GenMacMaster MAC de los interfaces master. Modificable por Master.
- GenUsersMac MACs de los usuarios permitidos en el sistema. Modificable por el Master.
- GenSystemNumber Número del sistema al que pertenece el dispositivo. Modificable por Master

### 3.2.3. Función: Disponibilidad y monitorización de dispositivos

Esta pantalla debe indicar de forma visual el estado de los dispositivos pertenecientes al mismo sistema que el interfaz. Podemos considerarla como la pantalla principal y origen del acceso al resto de funcionalidades.

Los dispositivos deben indicar con código de colores o iconos los estados que son recuperables sin necesidad de establecer una conexión:

- Versión de firmware
- Número de sistema

- Aviso: Baja batería
- Aviso: Mensajes pendientes

Una vez identificados los dispositivos deben ser seleccionables y acceder a las opciones sobre cada dispositivo de forma individual:

- Configuración.
- Mensajes.
- Sensores.
- Seguridad.

### 3.2.4. Función: Mensajes

Este bloque de funcionalidad permite gestionar los mensajes almacenados en el dispositivo. La forma de acceso a la memoria de mensajes es mediante conexión previa con el dispositivo objetivo y a través de atributos del profile Memory.

Los mensajes contienen la siguiente estructura de información. Cada campo de la estructura está asociado a un atributo del dispositivo:

<b>Campo</b>	<b>Descripción</b>	<b>Tamaño</b>
MemM1Number	Número único del mensaje	Entero: 2 bytes
MemM1Source	MAC origen del mensaje	6 bytes
MemM1Destination	MAC destino del mensaje Si no tiene valor el destino son todos los integrantes del sistema	6 bytes
MemM1DateTime	Fecha de creación del mensaje	6 bytes
MemM1Status	Indica el estado en qué se encuentra un mensaje. El significado de sus bits se define en la tabla 2.	1 byte
MemM1Type	Indica el tipo de mensaje almacenado: Tipo = 0 - Texto Tipo = 1 - Calendario/Acción Tipo = 2 - Lectura de mensaje (Pendiente) Tipo = 3 - Medida realizada (Pendiente)	1 byte
MemM1Title	Almacena el título del mensaje	60 bytes
MemM1Content1	Contenido 1 del mensaje	160 bytes
MemM1Content2	Contenido 2 del mensaje	160 bytes
MemM1Action	Acción a ejecutar si se cumple el schedule del mensaje	1 byte

MemM1Schedule	Horario/Calendario de aviso de ejecución de mensajes tipo Calendario	4 bytes
MemM1LinkDown	Mensaje enlazado anterior	Entero: 2 bytes
MemM1LinkUp	Mensaje enlazado posterior	Entero: 2 bytes
	Total	351 bytes

Tabla 3.1 Estructura mensajes (Fuente: BlauTic )

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
N/A	N/A	N/A	N/A	Requiere acción	Confirmado	Requiere confirmación	Pdte lectura

Tabla 3.2 Status mensajes (Fuente: BlauTic)

El firmware debe permitir acceder rápidamente a los mensajes:

- Pendientes de leer.
- Pendientes de confirmación
- De un determinado usuario
- Por tipo de mensaje

Los siguientes atributos también forman parte del profile Memory con el objeto de informar de forma global sobre el estado:

MemPending : Indica si hay mensajes pendientes de leer/confirmar. Se puede leer a través de atributo o recibir en paquete ADV IND.

MemInstalled: Indica la cantidad de memoria instalada

MemUsed :Indica la cantidad de memoria usada

MemWarn :Indica el porcentaje umbral a partir del cual se genera un aviso por memoria casi llena. Sepuede leer o recibir como CONN\_NOTIFICATION.

Las acciones a realizar por esta actividad serán:

- Leer. Obtener mensajes del dispositivo al alcance que no han sido leídos por el interfaz que accede y sobre los que tenga permisos suficientes.
- Filtrar/Ordenar: por Dispositivo, por Interfaz
- Acceder. Acceder a los contenidos de un mensaje en concreto
- Eliminar. Borrar un mensaje
- Mensaje Nuevo:
  - Tipo Texto , Texto con confirmación
  - Tipo Calendario/Acción

Accede a pantalla que permite rellenar tanto título como contenidos.  
En la base de datos inicial debe haber configuración estándar que permita rellenar los campos con mensajes predefinidos. Tb puede existir la posibilidad de incorporar texto libre.  
El diseño gráfico de la aplicación debería ser lo suficientemente flexible como para poder presentar diferentes tipos de formulario.
- Descargar Pasar todos los mensajes del dispositivo a un servidor por bluetooth3.0 o Wifi/3G.

En todo momento desde la lista de mensajes del dispositivos, debemos tener conocimiento del estado del mensaje con una referencia de colores o iconos que nos permita clasificarlos de forma rápida.

### 3.2.5. Función: Sensores

Este bloque de funcionalidad permite gestionar los sensores y las medidas almacenadas en el dispositivo.

Los sensores disponibles para la aplicación son:

- Entrada digital.
- Entrada analógica/Temperatura.
- Salida digital.
- Acelerómetro

En la aplicación se podrá trabajar con tres sensores simultáneamente, eligiendo para el sensor 2 si se utiliza una medida externa o el valor de temperatura del sensor integrado.

Pasamos a numerar las diferentes acciones que debe realizar la aplicación:

#### CONFIGURAR

La aplicación debe proporcionar pantallas para configurar los parámetros descritos seguidamente. La configuración se realiza previa conexión con el dispositivo:

- SnsName. Nombre significativo del sensor.
- SnsReadMode. Valores posibles: Manual, periódico, cambio Valor
- SnsEnableAdv. Habilitar el envío Advertisement.
- SnsEnableConn. Habilitar el envío cuando nos conectamos al sensor.
- SnsEnableLog. Habilitar el registro de logs como mensajes tipo sensor.
- SnsValue. Valor actual del sensor.
- SnsUnit. Unidad de medida del sensor analógico.
- SnsWarmEq. Ecuación de comparación para umbral.
- SnsMesFreq. Frecuencia de medición de valores.
- SnsSendFreq. Frecuencia de envío de valores.

Los siguiente atributos son exclusivos del sensor acelerómetro por sus excepcionales características:

- AceEnable. Habilitar el sensor acelerómetro.
- AceX. Medida del valor en el eje de las x.
- AceY. Medida del valor en el eje de las y.
- AceZ. Medida del valor en el eje de las z.
- AceFreq. Frecuencia de medida del acelerómetro.
- AceFreqEnv. Frecuencia de envío de medidas.
- AceRange. Umbral de rango de aviso.

## ACTIVAR/DESACTIVAR

Acceso rápido a habilitar/deshabilitar la realización de medidas. La forma de activar el sensor es mediante la configuración del modo de realización de medidas que depende del sensor. Los modos de funcionamiento son:

- Modo Manual. Disponible para todos los sensores. El sensor está deshabilitado y sólo realiza medidas cuando se le solicita de forma manual.
- Modo Periódico. Sensor lectura digital y analógica. El sensor está activado y realiza medidas de forma periódica en función del parámetro de frecuencias de medidas.
- Modo interrupción. Para el sensor digital. El sensor está activado y detecta cualquier cambio de valor en su entrada

Además, se puede recibir datos de los sensores de tres modos (pueden configurarse simultáneamente):

- Broadcast. Incluido en el paquete Advertisement del dispositivo.
- Conexión. Mediante pregunta de lectura o recepción de dato asíncrono.
- Log de mensaje. Mensajes con el mismo formato que el resto de mensajes de tipo sensor.

## ACCESO RAPIDO

Se dispondrá de una pantalla de acceso rápido donde se mostrará el valor de la medida obtenido y se gestionará la configuración de los valores más importantes para cada sensor.

- Mostrar valores. Pantalla que muestre permanentemente valores de los sensores habilitados en forma legible con datos del nombre del sensor, valor y unidad de medida.
- Leer valor. Lectura asíncrona del valor existente en el sensor en ese instante
- Escribir valor. Escribir un valor al puerto digital de salida. Valores ON/OFF. La aplicación debe conocer el estado del puerto y permitir ponerlo ON, OFF o hacer un toggle sobre el mismo.
- Activas broadcast. Habilitar el envío de datos sin necesidad de conexión mediante paquete Advertisement.
- Activar Log. Habilitar el registro de mensajes con los valores de las medidas realizadas a la frecuencia configurada para envío.

## REENVIAR ONLINE

Forzar el envío a otro interfaz (servidor) vía Bluetooth o 3G los datos que llegan al móvil en ese instante.

## HISTÓRICO

Mostrar datos históricos de datos de sensores almacenados filtrando por sus diferentes campos: dispositivo, sensor, fecha, valor. Este histórico estará almacenado en mensajes tipo sensor. Además se incorporará la opción de sincronizar estos mensajes.

### 3.2.6. Función: Seguridad

Este bloque de funcionalidad permite gestionar las opciones de activación de Seguridad. Las funciones de seguridad son las siguientes:

- Activar la monitorización de proximidad de determinados dispositivos.
- Activar la pérdida de cobertura de dispositivos monitorizados.
- Utilizar el sensor acelerómetro para enviar información cuando se produce un movimiento determinado y detectado por el sensor.

Desde este apartado podremos configurar los atributos del perfil de seguridad además de habilitar/deshabilitar la monitorización para todos los dispositivos.

Los parámetros que podremos gestionar tras la conexión con el dispositivo son:

SecProxEnable : Habilita la opción de proximidad para que el dispositivo envíe con mayor frecuencia el paquete ADV IND que permite monitorizar la presencia.

Posiblemente se requiera obtener la RSSI (nivel de potencia recibido) y contrastarlo con nivel de potencia transmitido para poder pre-avisar cuando el dispositivo se acerque al umbral de pérdida.

SecAceEnable: Habilitar el uso del acelerómetro para la funcionalidad de proximidad

SecAceThreshold: Fijar el umbral del acelerómetro a partir

### 3.2.7. SINCRONIZACIÓN

A continuación se va a especificar el protocolo de intercambio de información entre los interfaces móviles y el servidor central a través de TCP/IP . (Mediante 3G o Wifi).

El protocolo se basa en las siguientes características generales:

- Protocolo: TCP/Ipv4
- IP: Configurable
- Puerto: Configurable
- Encriptación: Inicialmente abierto.
- Autenticación: Inicialmente sin autenticar.

### COMUNICACIÓN BASE

El interfaz es el responsable de iniciar la conexión TCP con el servidor usando los parámetros de la conexión adecuados. Si un dispositivo interfaz habilita sus comunicaciones TCP/IP con el servidor central, enviará tramas periódicas al servidor con el siguiente formato. De este modo el servidor conoce a todos los interfaces que tiene a su alcance en cada momento.

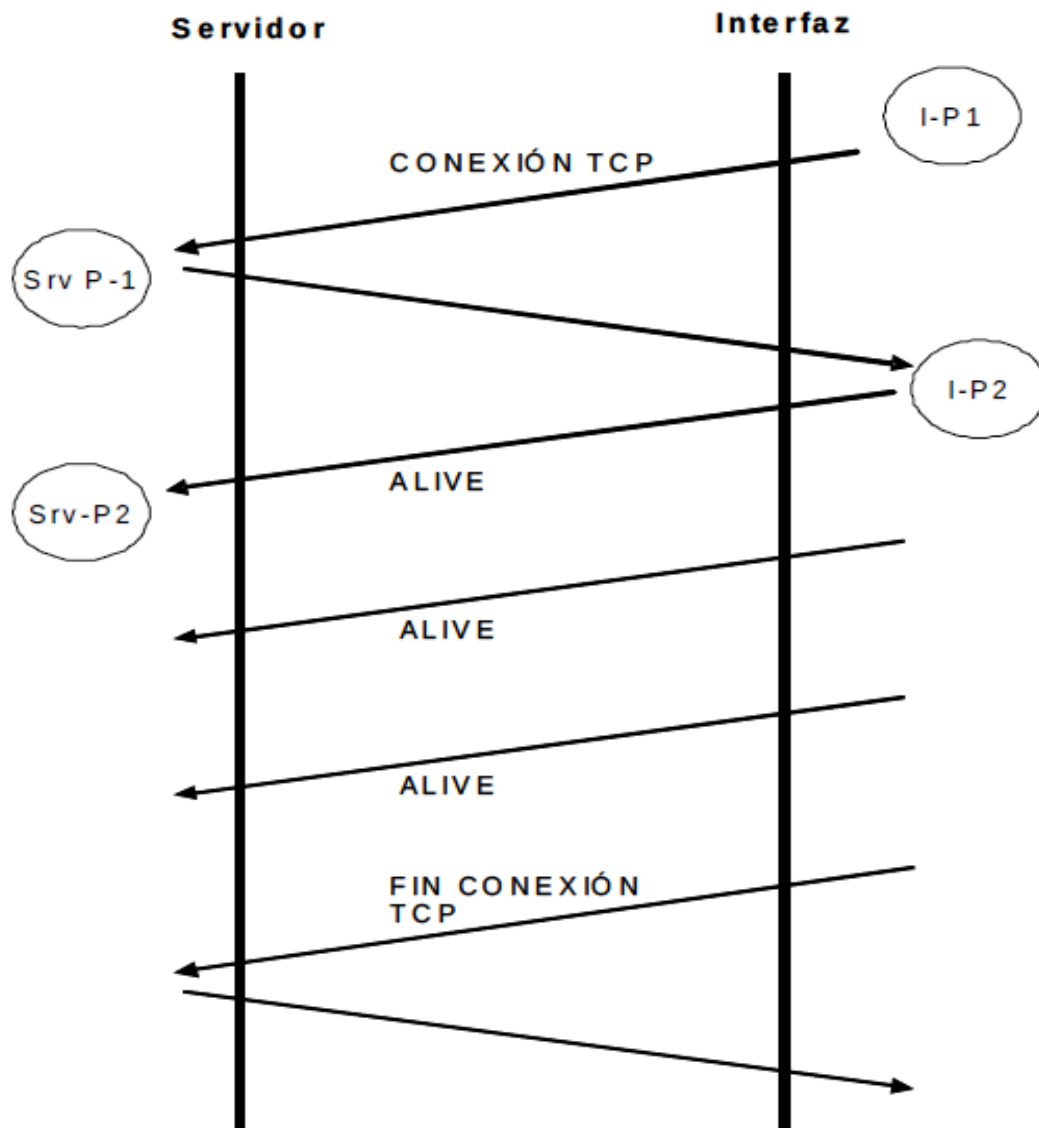


Figura 3.1 Diagrama de conexión TCP con el servidor (Fuente: BlauTic)

En todas las tramas vendrá definidos una serie de campos:

Date. Fecha/hora en la que se ha producido el evento

- Frame. Número que utiliza el cliente para controlar el flujo de tramas. El servidor utiliza en la respuesta el mismo número como referencia a la solicitud.
- Result. Indica el resultado del envío:
  - OK. Si la trama ha sido aceptada como correcta
  - ERROR. Número de error identificando el tipo de error que invalida la transmisión.
    - Err\_1 : sin tipo de trama
    - Err\_2 : sin identificador de interfaz
    - Err\_3 : valor de frame incorrecto
    - Err\_4 : error ene l formato de fecha
    - Err\_5 : diferencia de fecha superior a timeout
    - Err\_10 : sin source
    - Err\_11 : sin destinatario
- Value. Aquellos valores que contengan datos binarios se enviarán mediante la conversión a valores ASCII de su correspondientes valores hexadecimales.



## TRAMAS

Las tramas se enviarán de forma transparente hacia el servidor previa composición de un mensaje XML donde especificares el formato en cada caso. La respuesta por parte del servidor se realizará por tanto bajo la misma codificación.

### 1. Tramas Mensajes

```
<ble>
<type>message</type>
<id>Identificativo del interfaz </id>
<dev>identificativo del dispositivo</dev>
<frame>numero trama</frame>
<date>date</date>
<source>source</source>
<destination> destination </ destination >
<dategen> dategen </ dategen >
<dateread> dateread </ dateread >
<status>status</status>
<title> title </ title >
<content1> content1</ content1>
<schedule> schedule </ schedule >
<action> action</ action>
</ble>
```

### 2. Tramas Sensor

```
<ble>
<type>sensor</type>
<id>Identificativo del interfaz </id>
<dev>identificativo del dispositivo</dev>
<frame>numero trama</frame>
<source>source</source>
<date>date</date>
<sensor_id> sensor_id </ sensor_id >
<value> value </ value>
<unit> unit </ unit >
</ble>
```

## 3.3 Elección de tecnologías

### 3.3.1 Análisis del sistema Android

Después del análisis realizado en el punto tres de esta memoria acerca de los distintos sistemas operativos, decidimos realizar la aplicación sobre la plataforma Android por su versatilidad a la hora de poder diseñar una aplicación y que esta pueda funcionar en diversos smartphones de diversas marcas y por la transparencia a la hora de programarla , recordemos que se trata de código abierto, con infinidad de ejemplos, tutoriales y recursos que podemos encontrar en internet, libros o incluso en la misma página web de desarrolladores oficial de Android (developer.android.com).

#### 3.3.1.1 Historia de la plataforma Android

La historia de Android empieza cuando en agosto de 2005 Google adquiere una empresa en crecimiento llamada Android Inc. Esta empresa se dedicaba al desarrollo de aplicaciones para móviles pero en un principio se justificó la adquisición ante la búsqueda de mentes innovadoras en los ingenieros que trabajaban en ella.

Poco después, en 2007, se anunció la creación del *Open Handset Alliance*, una asociación de 34 grandes empresas dedicadas a desarrollar estándares para dispositivos móviles. Entre estas grupo de empresas se incluye Google, que es la empresa que liderara y dirige la asociación, Nvidia, Intel, Motorola , T-mobile y otras compañías del sector. Con la aparición de esta alianza i del teléfono móvil iPhone de Apple en 2007, creció la posibilidad de que Google se introdujera en este mercado con un producto similar al de Apple. Ese mismo año se lanzó la primera versión Beta de Android, de la que se realizaron múltiples pruebas , pero no fue hasta septiembre de 2008 cuando se comercializó el primer dispositivo móvil, HTC Dream, que incorporaría esta plataforma en su versión Android 1.0.

Desde esta fecha hasta el día de hoy han aparecido 8 versiones, cada una de ella mejorada respecto de la anterior. Eso sí, desde la primera de sus versiones ya se integraron múltiples servicios de Google (Calendar, Gmail... ), opción de navegación web, Android Market, multitarea, mensajería instantánea, Wifi y bluetooth. Por orden cronológico estas son:

- Android 1.0(2008)
- Android 1.5 (Cupcake, Abril 2009)
- Android 1.6 (Donut, Septiembre 2009)
- Android 2.0 (Eclair, Octubre 2009)
- Android 2.2 (Froyo, Mayo 2010)
- Android 2.3 (Gengerbread, Diciembre 2010)
- Android 3.0 (Honeycomb, Mayo 2011)
- Android 4.0 (Ice Cream Sandwich, Octubre 2011)
- Android 4.1 (Jelly Bean, Junio 2012)
- Android 4.2 (Jelly Bean – Gummy Bear, Octubre 2012)
- Android 4.3 (Jelly Bean, Julio 2013)
- Android 4.4 (KitKat, Octubre 2013)

La versión más distribuida de Android es Android 2.2 Froyo, propiciada por la explosión de terminales de gama media-alta y de las primeras tabletas que fueron lanzadas en esa época. El incremento del uso de redes sociales móviles y el cambio de actitud de los usuarios de telefonía a la hora de adquirir un terminal móvil fue otro de los puntos clave.



Como punto a destacar de la última una de las últimas versiones lanzadas, Android Ice Cream Sandwich unifica el sistema para que pueda ser instalado indistintamente en smartphones como en tablets, independientemente de la resolución de pantalla del dispositivo.

### **3.3.1.2 Características principales de la plataforma**

La principal característica de Android es que se trata de una plataforma libre. De esta manera, los desarrolladores de todo el mundo pueden acceder al código y así personalizar y adaptar cada dispositivo móvil a las necesidades de los usuarios. Además, esta libertad se transmite al desarrollo de aplicaciones, ya que no hay diferencias entre los programas que incorpora el sistema operativo de base y las desarrolladas por terceras personas dado a que las herramientas de trabajo son las mismas. La plataforma puede residir en diferentes terminales de distintas marcas (HTC, Huawei, Samsung, ...), de las que dependen las actualizaciones del sistema operativo.

La arquitectura del sistema está diseñada para simplificar la reutilización de componentes. Esto quiere decir que desde cualquier aplicación se puede hacer uso de las funcionalidades de otra incluso integrar los permisos adicionales de esta en la propia aplicación. Además, dependiendo de las características técnicas del dispositivo móvil donde resida este sistema, podremos generar aplicaciones haciendo uso de la cámara de fotos, GPS, bluetooth, acelerómetro, etc... Esto genera un entorno en el cual las aplicaciones generadas por terceros pueden sustituir las aplicaciones por defecto de Android, haciendo uso de toda la infraestructura del sistema.

Otra característica destacada de este sistema operativo es la unificación de desarrollo para los dispositivos móviles que permite que cualquier aplicación sirva para diferentes dispositivos que utilicen Android, independientemente de la marca, siempre que dispongan de los requisitos necesarios para su funcionamiento.

Para finalizar comentaremos dos aspectos también importantes que incorpora Android, como son el uso de la máquina virtual *Dalvik*, que se analizará posteriormente, y la elección de SQLite como sistema gestor de base de datos por el poco espacio que ocupa.

### 3.3.1.3 Filosofía de diseño de aplicaciones

Actualmente las aplicaciones desarrolladas para smartphones se parecen más a los programas utilizados en un equipo más grande como pueda ser un ordenador, pero con una capacidad de procesador mucho más baja. Es por ello que para hacer de este producto algo preferible y apetecible una aplicación debe ser: rápida, interactiva i fluida.

La interacción de una aplicación con el usuario es uno de los factores clave para que esta tenga éxito. Aplicaciones que no indiquen que proceso se está ejecutando o que se congelen mientras realizan operaciones costosas son ejemplos que pueden provocar la desesperación del usuario. Con tal de evitar incomodas esperas, Android incorpora un mecanismo para ofrecer al usuario , mediante un mensaje llamado ANR (*Application Not Responding*) , la finalización de un proceso o que este se encuentra ejecución. Estos mensaje suelen también aparecer en caso de producirse algún tipo de error y el usuario puede forzar el cierre de dicha aplicación. Android permite el uso de aplicaciones hijo (AsincTask), que se realizan en background, para realizar operaciones costosas en segundo plano mientras el usuario realiza otro tipo de actividades.

Desde el punto de vista de seguridad y gestión de recursos existen una serie de limitaciones en forma de permisos, restricciones y prioridades que hacen que las aplicaciones no puedan hacer uso del sistema a no ser que el usuario lo haya autorizado previamente. Desde nuestra aplicación podemos querer tener acceso a internet, al servicio de mensajería o consultar los mapas de Google. Todos estos permisos por tanto deben ser validados previamente. Una aplicación que gestione por ejemplo la recepción de llamadas telefónicas tendrá siempre prioridad máxima por tratarse de la función esencial de un smartphone. Desde el punto de vista del desarrollador, esta lista de permisos a la cual se dota a la aplicación, viene definida en el documento Manifest.xml que incorpora toda aplicación es su código.

Un aplicación en Android se puede observar como un conjunto de bloques que interaccionan entre ellos. Estos bloque fundamentales son: Activity, Service, ContentProvider y IntentRecivers. Cada una de ellas con una función diferente dentro de las aplicaciones de la plataforma.

Las Actividades (Activity) representan cada una de las pantallas con las que el usuario puede interactuar. Las actividades del sistema Android , las cuales diríamos que son las unidades básicas de la aplicación, tienen un ciclo de vida semejante al de las página de un explorador web en las cuales se puede recuperar las pantallas anteriores a través de un histórico.

Como ya hemos indicado anteriormente, en el código siempre aparecerá el documento AndroidManifest.xml donde se recoge la información básica como, las distintas *activities* que conforman la aplicación y los permisos que esta necesita.

En la siguiente figura encontramos el ciclo de vida completo de una actividad Android.

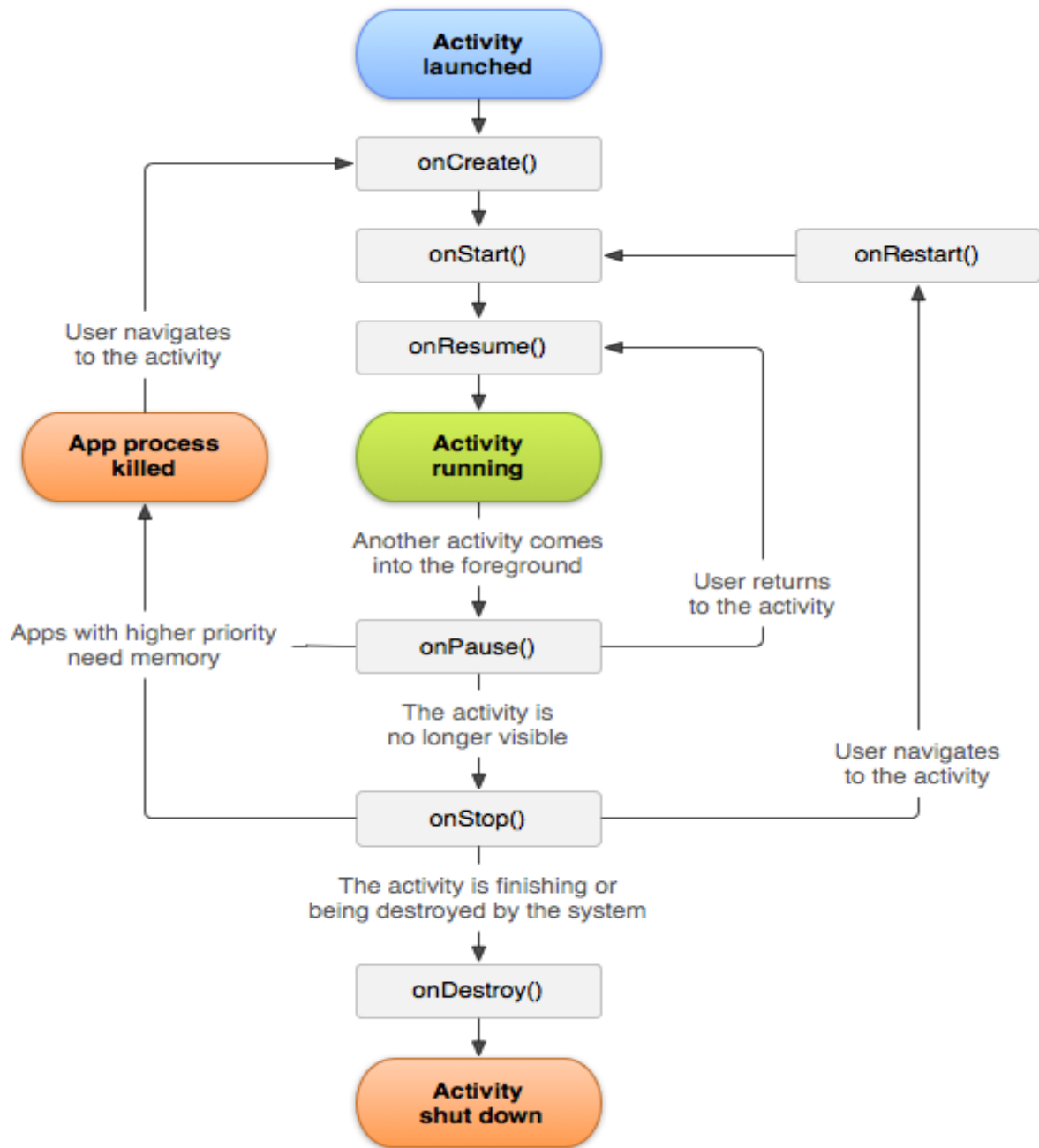


Figura 3.2 Esquema aplicación Android (Fuente: android.developers)

Comentar que todas la información acerca recomendaciones de estilo y buenas prácticas a la hora de realizar cualquier aplicación, y que por tanto hemos seguido en la desarrollo de este proyecto, vienen recogidas y descritas en los documentos disponibles en :

<http://source.android.com/source/code-style.html>

<http://developer.android.com/guide/topics/fundamentals/activities.html>

### 3.3.2. Entorno de programación

El corazón del sistema operativo está basado en un kernel 2.6 de Linux aunque con una amplia cantidad de librerías exclusivas generadas por Android. Java ha estado el lenguaje de programación elegido para el desarrollo de aplicaciones. No obstante esto, la ejecución de estas aplicaciones no se realiza sobre la típica Java Virtual Machine de Sun sino que Android utiliza una máquina virtual especialmente implementada para dispositivos móviles llamada Dalvik Virtual Machine. Una de las principales diferencias de estas dos máquinas es que la Dalvik es una máquina basada en registros mientras que la de Java está basada en pilas. Además, Dalvik ha sido diseñada para el uso con poca memoria lo cual le permite trabajar con un bajo consumo eléctrico.

Con todo ello, hemos elegido Eclipse como el entorno de desarrollo utilizado para desarrollar la aplicación ya que con la simple instalación del plugin gratuito ADT (Android Development Toolkit), podremos integrar y hacer uso del SDK (Software Development Kit) que nos proporciona Google para este cometido en Eclipse. En un capítulo posterior de esta memoria, haremos una referencia más extensa de estas herramientas utilizadas.

### 3.3.3. Base de datos

Por coherencia con el sistema operativo hemos escogido como motor de base de datos de nuestra aplicación SQLite. Este gestor de datos se caracteriza por ocupar muy poco espacio de memoria, disponer de una velocidad suficiente para trabajar en entornos móviles, no necesitar de un servidor, poca configuración y ser de código libre. Además esta base de datos está soportada nativamente por Android que incorpora toda una batería de herramientas necesarias para la creación y gestión de bases de datos SQLite. Como en cualquier otro lenguaje SQL podemos utilizar sentencias SELECT para realizar consultas sobre las tablas y utilizar INSERT, UPDATE o DELETE para introducir o modificar datos. El sistema Android nos va a facilitar el trabajo con un conjunto de clases Java que interactúan directamente con el sistema gestor y nos permite ejecutar estas sentencias sobre la base de datos de SQLite. Ejemplo de ellas son la clase Cursor, que permite el acceso a los resultados de una consulta SELECT de base de datos, o la clase auxiliar SQLiteOpenHelper donde podemos utilizar los métodos onCreate() y onUpgrade() para crear y actualizar respectivamente los datos de nuestra base de datos.

SQLite soporta datos tipo text (string), integer y real (similar a double en java).

Con esta solución manejaremos la información de nuestra aplicación necesaria para su correcto funcionamiento como es: el listado de dispositivos, el almacenamiento de mensajes, opciones de las pantallas, la configuración general de la aplicación, etc...

Como peculiaridad de nuestra aplicación, inicialmente requeriremos de una instalación remota de esta base de datos para poder arrancar por primera vez. En esta residirá información básica referente al sistema y que habrá sido personalizada, como por ejemplo: dispositivos BlauTic asociados, sources que pertenecen a nuestro sistema, información acerca de versiones, etc..

La descarga remota de esta base de datos está pensada que se realice con una pantalla inicial, al iniciar la aplicación, en la que se solicitará al usuario la introducción de una dirección IP proporcionada por el administrador del sistema BlauTic.

### 3.3.4. Gráficos y soporte de idiomas

Siguiendo con las especificaciones, el diseño de la interfaz gráfica se a separar completamente del código fuente de la aplicación y se especificará en ficheros XML que luego serán incorporados. Esto nos permitirá reutilizar un mismo diseño gráfico en diversas pantallas de la aplicación.

Comentar que Android permite tanto código XML como código fuente para realizar el entorno gráfico. Esta última forma será utilizada en los casos que debamos crear pantallas dinámicas, es decir, que interactúen con alguna de las funciones diseñadas en ellas. Puesto que la aplicación se visualizara en pantallas de diverso tamaño, se procurará utilizar diseños simples y intuitivos.

En cuanto al idioma, también se especificarán en ficheros XML. Existirá un fichero por cada idioma soportado en la aplicación. El código fuente incorporará el fichero y mostrará las cadenas de texto al usuario en el idioma seleccionado. Como buena práctica de estilo, se utilizarán resúmenes de las cadenas a mostrar en inglés y en mayúsculas en el código fuente para que este sea fácilmente legible por el programador.

### 3.3.5 Bluetooth 4.0

En este apartado se pretende justificar, a partir del análisis previo de tecnologías de comunicaciones a corta distancia realizado en el punto 2.1 de esta memoria, la utilización de Bluetooth y en concreto la nueva versión 4.0 de este para la trasferencia de datos entre los dispositivos BlauTic propietarios y nuestro dispositivo móvil.

El principal motivo de implementar Bluetooth sin duda es la gran aceptación y consolidación que esta tecnología tiene que por todas los fabricantes de dispositivos móviles lo que ha producido que todos los terminales la integren de serie evitando así la necesidad de utilizar cualquier adaptador externo para compatibilizarla. Esta característica descartaría todas las opciones menos NFC pero nos encontramos con el problema de cobertura de acción. Mientras NFC necesita de una visión casi directa entre los terminales que se comunican (menos de 10 cm), con bluetooth podemos encontrar rangos de acción en el peor de los casos de 1 m entornos con múltiples obstáculos, incluso paredes, entre los dispositivos hasta un máximo de 100 m en entornos libres.

Además bluetooth es de todos las tecnologías la que está diseñada para soportar mayor flujo de datos. Recordemos que nuestra aplicación a parte dar un servicio de monitorización de sensores permitirá el envío de mensajes entre usuarios y la configuración de diversos parámetros de los dispositivos BlauTic, lectura y escritura de datos, con toda la carga de información que todo esto conlleva.

Las siguientes tablas servirán a modo de resumen/comparativo de las diferentes tecnologías de comunicaciones inalámbricas a corta distancia:

	<b>ZigBee</b>	<b>Bluetooth</b>	<b>NFC</b>
<b>Banda de frecuencia</b>	2'4 GHz, 868/915 MHz	2'4 GHz	13'56 MHz
<b>Tasa</b>	250/20/40 Kbps	1-3 Mbps	106-848 Kbps
<b>Tipos de datos</b>	Digital	Digital, audio	Digital
<b>Alcance</b>	1 -100 m	1 -100m	10 cm
<b>Número de disp.</b>	255/65535	8	Punto a punto
<b>Consumo de potencia</b>	30 mA trx. 3 mA reposo	40 mA trx. 0'2 mA reposo	>15 mA trx.
<b>Precio</b>	Bajo	Medio	Media
<b>Complejidad</b>	Simple	Complejo	Simple

*Tabla 3.3 Comparativa comunicaciones inalámbricas (Fuente: elaboración propia)*

Sin duda otro aspecto que se ha tenido en cuenta es el tema de consumos energéticos y costes. A priori esto podría suponer el principal escollo al utilizar esta tecnología si la comparamos con las otras pero este problema queda totalmente solucionado, como veremos a continuación, con la nueva versión de Bluetooth, la 4.0.

### **BLE (Bluetooth Low Energy / Bluetooth 4.0 )**

Es la última versión de este protocolo desarrollado por el Bluetooth SIG a mediados de 2010 y conocido vulgarmente como “Bluetooth de bajo consumo”. Una de las primera compañías en lanzar un dispositivo móvil con esta tecnología fue Apple con su iPhone 4s , como ya había hecho unos meses antes con otros productos como MacBok Air y Mac mini.

Lo revolucionario de esta nueva versión y por tanto su principal característica es la optimización de su consumo energético. El gasto eléctrico en transmisión es inferior a los 20 mA (frente a los 30 mA de la versión 3.0) pero es en su estado de reposo donde el nuevo chip presenta mejores prestaciones. Se estima que en este estado de latencia se puede mantener más de un año con una sola pila de botón. Todas estas características y el hecho de que la circuitería del chip sea aproximadamente la mitad que la de las versiones anteriores reduce notablemente el consumo energético en el terminal móvil del usuario.

Pasando a comentar aspectos más técnicos de esta nueva versión, según las especificaciones de Bluetooth SIG se aumenta el rango de cobertura hasta los 100 m y se consiguen unas tasas de transferencia de datos máximas teóricas entre dispositivos compatibles de hasta 1 Mbps Además se mejora la sincronización entre dispositivos ya que se reduce la latencia de comunicación de 100 ms a solamente 6 ms y los tiempos de encendido que pasan de 6 s a menos de 3 ms. Asimismo se mejora la seguridad al contar con una encriptación 128-bit AES.

Como aspectos negativos encontramos que Bluetooth 4.0 no es compatible con ninguna de las versiones anteriores a pesar de trabajar en la misma frecuencia, por lo que en la actualidad los dispositivos móviles que la integran incorporan un doble chip que conmuta automáticamente según se la versión del dispositivo al que se conecta.



## 3.3.6. Herramientas utilizadas

### 3.3.6.1 SDK (Software Development Kit)

El SDK de Android es una aplicación que se puede descargar de forma gratuita en Google (<http://code.google.com/android/download.html>). En este Kit de desarrollo podemos encontrar todas las herramientas que un programador necesita para elaborar una aplicación: entorno de desarrollo y depuración, librerías, un emulador de terminales móviles, documentación, tutoriales, códigos de ejemplo. El SDK funciona en sistemas Windows XP o Vista, MAC OS X 10.4.8 o superior y en Ubuntu 6.0 o superior y necesita de Python 2.2 y Java 1.5 o superior para funcionar. No obstante, la mayoría de estas opciones quedan en desuso al utilizar el entorno de Eclipse con el plugin ADT.

Este complemento gratuito para Eclipse, llamado Android Development Toolkit, es proporcionado por Google y nos facilita el desarrollo de aplicaciones en este entorno. En el apéndice A de esta memoria tendremos una explicación detallada de la instalación de este entorno de trabajo.

Una vez instaladas y combinadas las piezas (Eclipse + SDK + ADT) comentaremos los aspectos más importantes de este entorno y que hemos utilizado a la hora de desarrollar la aplicación. Las principales características relacionadas con el desarrollo de aplicaciones son:

- Un entorno avanzado de desarrollo Java que incluye compilación de código fuente Java, autocompletar clases e integración de Javadoc.
- Uso de paquetes básicos de librerías Java así como las específicas de Android.
- Depuración de nivel de código fuente.
- Administración e inicio de perfiles de emulador de Android.
- Servicio de Monitorización de Depuración Dalvik (DDMS):
  - Vista de subprocesos y pilas
  - Administración de sistemas de archivos del emulador
  - Control del emulador
  - Registro de sistema y aplicaciones

Al desarrollar aplicaciones Android, hay dos perspectivas de Eclipse que nos han resultado muy útiles: la perspectiva Java donde hemos realizado la escritura de código visualizando en todo momento la estructura del mismo ; y la perspectiva DDMS donde se nos proporciona una especie de panel de control del dispositivo Android en ejecución, en nuestro caso emulador, y se registran todos procesos originados y pudiendo localizar los errores fácilmente.

Para explicar de forma más clara todo el proceso de desarrollo de la aplicación, desde la escritura de código hasta la obtención del paquete final .apk, adjuntamos el siguiente esquema:

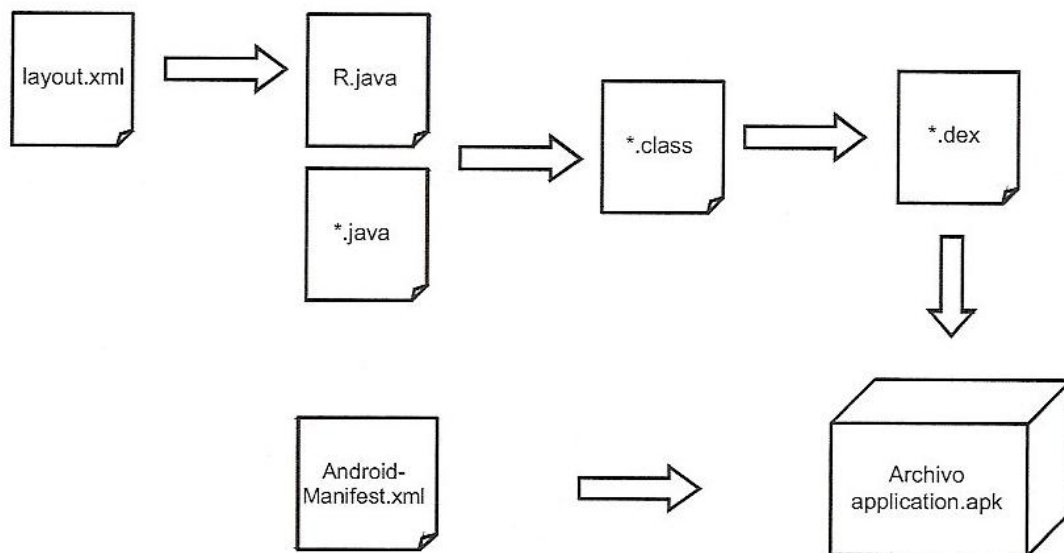


Figura 3.3 Esquema del código de una aplicación (Fuente: android.developers)

Recordemos que a pesar de la dependencia de Java en tiempo de compilación, las aplicaciones de Android no se ejecutan en una máquina virtual de Java, sino en la máquina virtual Dalvik. Por ello, los códigos Java creados por el compilador de Eclipse deben convertirse a formato .dex para utilizarlos en Android. Los archivos XML del proyecto se convierten en una representación binaria, sin embargo, conservan su extensión .xml en el dispositivo.

Como observamos, finalmente se genera el archivo .apk el cual ya podremos utilizar tanto en el emulador como en un dispositivo real.

A continuación haremos una mención especial al emulador, esta útil herramienta que nos ha permitido testear nuestra aplicación durante el proceso de desarrollo y pruebas.

## EMULADOR

Hay muchas razones para utilizar un emulador de calidad en las fases de desarrollo de cualquier aplicación. Una es que disponer de varios dispositivos reales con planes de datos resulta muy costoso. Además si tenemos en cuenta la multitud de dispositivos de diferentes firmas, cada uno con prestaciones diferente, en la que podemos utilizar Android nos hace pensar que no resulta práctico disponer de varios terminales, a menos que se disponga de un enorme presupuesto, como podría suceder con las grandes empresas de software.

Además de ser una herramienta más rápida, desde el punto de vista de ejecución, que el dispositivo hardware real con el emulador nos ha permitido tener en consideración las características físicas del dispositivo, en especial el tamaño de la pantalla, los dispositivos de entrada y la conectividad de red. El emulador incorpora el sistema operativo de la plataforma con una serie de aplicaciones y servicios básicos para poder simular las funciones típicas y el comportamiento de un dispositivo móvil. Estas funcionalidades pueden ser gestionadas y probadas con las aplicaciones desarrolladas mediante la simulación de eventos en el dispositivo, como por ejemplo mensajes SMS, navegador Web o la simulación de llamadas.

### 3.3.6.2 SVN

Otra herramienta de trabajo que se va a utilizar es Subversión. Este software administra el acceso conjunto a los ficheros de la aplicación y mantiene un historial de los cambios realizados en ellos permitiendo un control sobre las versiones realizadas.

Se trabajará con este programa para así poder interactuar en grupo con otros desarrolladores del proyecto que puedan aportar y controlar el código fuente de esta aplicación.

En el apéndice B de la memoria realizaremos una explicación más amplia sobre la instalación y funcionamiento de este software.

## 3.4. Planificación

En cuanto a las fases a seguir en este proyecto se han definido tres módulos de trabajo para llevar a cabo la totalidad de los objetivos. A continuación se detallarán y citarán las distintas tareas que habrá que abordar en cada una de ellas.

### **FASE 1. Configuración y base de datos:**

- Generación de la base de datos
- Codificación de las funciones generales de conexión a la base de datos.
- Desarrollo de las funciones de acceso a los parámetros de configuración, tanto de lectura como de escritura.
- Creación de las pantallas para la visualización y edición de los parámetros configurados.
- Implementación de las funciones básicas del módulo de comunicaciones bluetooth para que permita la conexión con los dispositivos.
- Listado de dispositivos BlauTic. Métodos y creación de la pantalla para el listado

### **FASE 2. Desarrollo del apartado de mensajes:**

- Codificación de la clase mensajes.
- Creación de las funciones de acceso a la base de datos.
- Implementación de los métodos y funciones de lectura, contestación, creación de mensajes nuevos.
- Pantalla de listado de mensajes.
- Generación de filtros que se apliquen a la lista de mensajes.
- Creación de las pantallas necesarias. Visualización, creación de mensajes, buzones...
- Codificación del paquete de comunicaciones 3G / Wifi que permitan la sincronización de datos a través de la red con un servidor remoto.
- Implementación de las comunicaciones mediante bluetooth.
- Descarga de mensaje de un dispositivo.

### **FASE 3. Desarrollo del apartado de sensores:**

- Codificación de la clase sensor.
- Creación de las funciones correspondientes de acceso a la base de datos.
- Diseño de los métodos y funciones para la lectura y escritura de los valores de los sensores.
- Implementación de las comunicaciones bluetooth.
- Métodos de acceso al histórico de sensores.
- Diseño de la pantalla correspondientes: listado de de sensores, visualización de valores, edición de configuración.
- Descarga de valores a un servidor remoto. Paquete de comunicaciones 3G/ Wifi

En cuanto a como se va a estructurar el código de la aplicación se van a definir cuatro capas o niveles con la intención de separar la funcionalidad de este y hacerlo más inteligible en futuras modificaciones.

#### **NIVEL 1. Base de datos.**

Se definirán las funciones básicas que permitirá la instalación de la base de datos (SQLite) que requiere la aplicación tanto de forma manual como automática al iniciar por primera vez la aplicación. En este nivel se alojaran los métodos de acceso a la base de datos que permitirá recuperar o añadir elementos nuevos.

#### **NIVEL 2. Recursos gráficos y pantallas.**

De las dos opciones que permite la plataforma Android para diseñar la interfaz gráfica en las aplicaciones hemos escogido la creación de pantalla mediante códigos XML. Como ya hemos comentado en algún punto de esta memoria, esto nos permite separar lo que es código funcional del aspecto visual además de permitir la reutilización de estas pantallas en diferentes partes.

También se definirá en un archivo XML todos los vocablos y texto utilizadas (strings) en la aplicación permitiendo así de forma fácil la adaptación a distintos idiomas.

Otra parte importante es la formada por los distintos iconos y gráficos (formatos .gif y .jpg) que se utilizan a lo largo de la aplicación. Las dimensiones se ajustarán a los manuales de estilo definidos y que comentaremos con más detalles el punto 3.6 de esta memoria.

#### **NIVEL 3. Bluetooth**

Se separará en otro nivel el código de los distintos métodos que implemente la funcionalidad de Bluetooth. En esta parte se administrará todo lo referente a la conexión, desconexión, escritura y lectura de valores con los dispositivos mediante Bluetooth 4.0.

#### **NIVEL 4. Código fuente.**

Será la parte más densa y estará formada por todo el código que forma el núcleo principal de la aplicación. Los distintos métodos nos permitirán controlar toda la funcionalidad de la aplicación accediendo a los métodos anteriormente citados cuando se requiera.

### 3.5.1 Diseño de pantallas y funcionalidades

En este apartado se empezará a diseñar y explicar las diferentes funcionalidades de las que dispondrá la aplicación así como a esbozar el aspecto gráfico de las pantallas.

Como viene definido en las especificaciones del producto, nuestro interfaz móvil, donde estará instalada esta aplicación, pertenecerá a un sistema el cual estará formado por más interfaces (smartphones, computadoras, equipos portátiles...) y por diferentes dispositivos BlauTic los cuales podremos controlar bajo previa autorización del administrador del sistema.

Este será la pantalla inicial de nuestra aplicación formada por una menú de cinco botones desde los cuales podremos acceder a los apartados que durante este punto van a ser analizados: dispositivos, mensajes, sensores, seguridad y configuración.



*Figura 3.4 Pantalla inicial (Fuente: elaboración propia)*

Al iniciar por primera vez la aplicación la base de datos de la aplicación, donde reside toda la información al respecto de los dispositivos e interfaces de nuestro mismo sistema, será instalada remotamente bajo petición de un código o dirección web que será suministrado por el administrador del sistema. A partir de este momento, se nos asignará un número de sistema y ya podremos acceder al control de los dispositivos BlauTic de los que tengamos autorización

A lo largo de este apartado encontraremos numerosa referencias a la sincronización del dispositivos. Se trata del envío de datos a través de internet, vía WiFi o redes 3G, a un servidor remoto para el posterior procesamiento de estos. En el apartado 3.6.2 de esta memoria abordaremos con mayor detalle como se ha dotado a la aplicación de esta funcionalidad.

### 3.5.1.1 Configuración

Como se indica en las especificaciones del proyecto , desde el apartado de configuración de la aplicación el usuario debe ser capaz de configurar desde aspectos básicos como cambiar el idioma hasta aspectos más técnicos como pudieran ser los referentes a la sincronización de datos con un servidor remoto.

Desde la pantalla inicial de este apartado podremos cambiar el nombre de nuestro dispositivos, acceder de forma rápida a la búsqueda de actualizaciones del software, seleccionar el idioma de la aplicación o acceder a la información web del producto entre otras cosas. Esta pantalla quedará abierta a futuras modificaciones conforme vayan surgiendo aspectos configurables en el desarrollo de la aplicación.

Para organizar más cómodamente todos los puntos configurables, estos quedaran agrupados del siguiente modo:

#### 1.General

Donde se almacenará información técnica como pueda ser la versión del firmware, versión de aplicación, MAC del dispositivo móvil o el número de sistema al que pertenece. Todos estos datos serán valores de lectura.

#### 2.Datos

En este apartado se van configurar los parámetros necesarios en las conexiones. La aplicación tiene dos maneras posibles de realizar conexiones y enviar datos:

BLAUTIC	
<b>Habilitar envío de datos</b> Sincronización de datos con servidor	<input checked="" type="checkbox"/>
<b>Servidor remoto:</b> 192.168.1.1	
<b>Bluetooth</b> Activar Bluetooth	<input checked="" type="checkbox"/>
<b>Dirección MAC destino:</b> 00:00:00:00	
<b>Actualizar dispositivos</b> ¿desea actualizar la lista de dispositivos periódicamente?	<input checked="" type="checkbox"/>
<b>Frecuencia de refresco:</b> Actualiza la lista de dispositivos cada X segundos.	<input type="radio"/>

- Bluetooth 4.0. Conexión con dispositivos Blautic.
- Wifi / 3G. Envío de datos y sincronización con un servidor remoto.

Desde esta pantalla, con una pestaña de selección podremos habilitar el envío de datos a nivel de aplicación es decir, en el caso que una actividad de la aplicación requiera utilizar la red Wifi a la cual estamos conectados o en su defecto el envío mediante 3G, estaremos autorizando su utilización a sabiendas que esto puede ser un servicio con coste adicional. En todo caso, un mensaje de aviso lo informará al usuario. Seleccionada esta opción se nos permitirá editar la dirección web o dirección IP donde se realizarán las conexiones.

Otro botón de selección nos permitirá activar o no el Bluetooth de nuestro terminal móvil.

Además podremos definir el tiempo de refresco en el cual se actualiza la lista de dispositivos es decir, cada cuanto el terminal móvil realizará una búsqueda de los dispositivos BlauTic en cobertura de forma automática.

Figura 3.5 Menú configuración 1  
(Fuente: elaboración propia)

### 3.Avisos

Este apartado nos va a permitir definir el tipo de aviso que preferimos para cada acción de la aplicación.

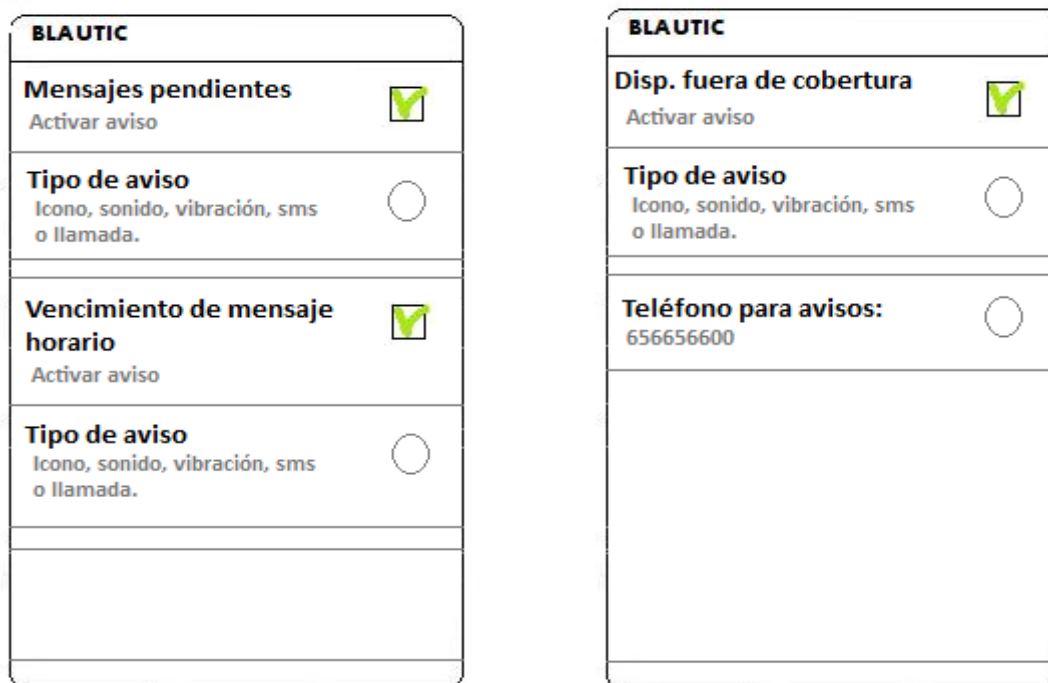


Figura 3.6 Menús de configuración 2,3 (Fuente: elaboración propia)

En concreto se van a poder escoger cinco tipos de aviso distintos:

- Mediante icono.
- Sonido
- Vibración
- Envío de sms
- Llamada.

Para alertar de la siguientes acciones:

- Mensajes pendientes. Si se dispone de algún tipo de mensaje nuevo o pendiente de lectura en el terminal.
- Vencimiento de mensajes horarios.
- Dispositivo fuera de cobertura. Si algún dispositivo perteneciente a nuestro sistema en el cual hayamos activado esta opción de seguridad, abandona nuestra zona de alcance.

Además se va a poder definir un teléfono de contacto para generar en su caso las llamadas de alerta o el envío predefinido de mensajes sms.

### 3.5.1.2 Dispositivos

Al seleccionar esta opción desde el menú inicial de la aplicación, accederemos a la lista de dispositivos BlauTic de nuestro sistema que tenemos actualmente en zona de cobertura. Recordemos que esta búsqueda de dispositivos se realiza mediante la tecnología Bluetooth 4.0 de la cual dispone nuestro terminal móvil, y que el refresco de esta lista es un parámetro configurable como ya hemos visto en el apartado anterior.

Con esta lista se pretende tener una referencia rápida del estado de cada dispositivo, principalmente saber que *profiles* tiene activados, señal de que estamos autorizados al manejo y control de dicha tarea, e indicar si existe algún tipo de alarma en el dispositivo que requiera de nuestra rápida atención, en este caso se ha configurado para que se indique un nivel bajo de la batería del dispositivo BlauTic.




ID	Profile activados	Alert
	   	

Figura 3.7 Layout de dispositivos (Fuente: elaboración propia)

Esta pantalla muestra los valores generales de los dispositivos BlauTic sin llegar a establecerse ningún tipo de conexión con ellos por lo que puede desaparecer de la lista en el momento que se pierda su cobertura. Desde la misma y pulsando el botón de la parte inferior se accederá a un menú de opciones que nos permitirá entre otras opciones filtrar la lista por perfiles indicando en la pantalla información aún más detallada.

Se deja la opción de poder incorporar una fotografía para identificar cada dispositivo de una manera más intuitiva. Para el resto de indicaciones se han escogido los siguientes iconos para mostrar al usuario de forma amigable los perfiles que tienes activados los dispositivos:



Figura 3.8 Dibujos de los perfiles (Fuente: elaboración propia)



El aspecto general de la pantalla de dispositivos será la siguiente:
















LISTA DE DISPOSITIVOS		
ID	Perfiles activados	Alarma
	 	
	 	
		
		
	  	
<b>Menú</b>		

Figura 3.9 Listado de dispositivos (Fuente: elaboración propia)

Seleccionando la opción "Menú" de la pantalla, accederemos a las opciones de filtrado. Estas opciones son:

### 1. General

Con esta opción se volverá al estado inicial de la lista, deshaciendo cualquier filtrado que se haya podido realizar. Los dispositivos BlauTic mostrados serán por tanto los que en ese momento permanezcan en zona de cobertura.

### 2. Mensajes

La pantalla filtrará los dispositivos que tengan el perfil mensajes activado y en ella se visualizará la siguiente información:

- ID. Número de identificación del dispositivo.
- Públicos. Número de mensajes públicos pendientes de leer que alberga el dispositivo BlauTic.
- Privados. Número de mensajes privados, mensaje de los que somos el destinatario, que quedan pendiente de leer/descargar del dispositivo.
- Especiales. Una icono de alarma requerirá la atención ante algún tipo de mensaje especial en el dispositivo que deba ser atendido de forma urgente por nuestra parte.

### 3. Seguridad

Aquí se filtrará de la lista de dispositivos aquellos que tengan dicho perfil activado, y en la pantalla se mostrará:

- ID. Número de identificación del dispositivo.
- Distancia del dispositivo. Distancia aproximada en metros a la cual se encuentra el dispositivo BlauTic.
- Alarma. Donde un icono marcará si se ha superado la distancia máxima permitida y configurada en el apartado seguridad del dispositivo.

#### 4. Localización

Esta opción interactúa con el acelerómetro que incorporan los dispositivos BlauTic dándonos la posición geográfica de los mismos. En la pantalla se mostrará:

- ID. Número de identificación del dispositivo.
- Localización. Punto donde se encuentra el dispositivo.

#### 5. Sensores

Se filtrará la lista por los dispositivos que tengan este perfil activado y se mostrará por pantalla una alarma que indique la urgente atención por parte del usuario:

- ID. Número de identificación del dispositivo.
- Valor sensor.
- Alerta. Un icono alertará de que alguno o algunos de los sensores del dispositivo ha superado algún valor umbral

#### 6. Actualizar

Con esta opción refrescaremos la lista de dispositivos de nuestro sistema de forma manual.

Desde la pantalla de este menú podremos acceder a los cuatro apartados funcionales y configurables de los dispositivos que son:

- Configuración
- Mensajes
- Sensores
- Seguridad

Notar que la aplicación solamente nos dejará acceder a las opciones de las cuales se nos ha concedido la autorización por parte del administrador del sistema. Si no disponemos de ningún tipo de permiso, o el dispositivo no tiene activado ningún perfil, solamente podremos acceder a la parte de configuración donde se muestran parámetros, sólo de lectura, del dispositivo como:

- Nombre del dispositivo.
- Número de serie.
- Fecha y hora.
- MAC de los interfaces Máster.
- Usuarios permitidos.
- Número de sistema.

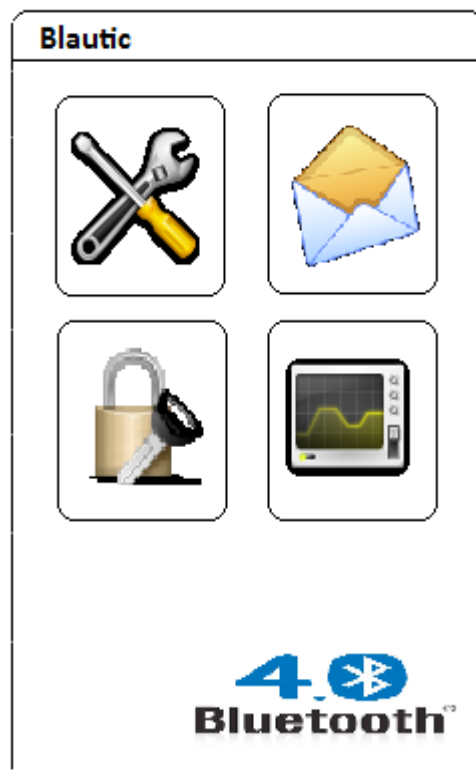


Figura 3.10 Menú para dispositivos  
(Fuente: elaboración propia)

A partir de aquí, la navegación por el resto de opciones de esta pantalla (Mensajes , Sensores y Seguridad) se va a explicar en los sucesivos apartados de este punto de la memoria. Recordar que haremos referencia a ellos cuando se especifique que se ha establecido conexión con el dispositivo BlauTic.

### 3.5.1.3 Mensajes

En este apartado abordaremos el diseño y aspectos gráfico que tendrán las pantallas referentes a los mensajes de la aplicación así como su funcionalidad a tenor de las especificaciones del producto. El siguiente esquema pretende explicar de forma clara el sistema de intercambio de mensajes entre interfaces y dispositivos.

Como interfaces, sources, con capacidad para redactar mensajes en nuestro sistema disponemos de dispositivos móviles (smartphones) o de computadoras que incorporen la aplicación. Realizando un símil con un sistema de telecomunicaciones, estos sources serían Emisor y/o Receptor y los dispositivos BlauTic en este caso realizarían la función de Canal de comunicaciones entre ambos, aunque como veremos más tarde, estos dispositivos también son capaces de emitir mensajes tipo sensor.

No hay que olvidar, como parte fundamental de este proyecto, que en todos estas comunicaciones la tecnología empleada es el Bluetooth 4.0 que incorporan tanto los interfaces como los dispositivos.

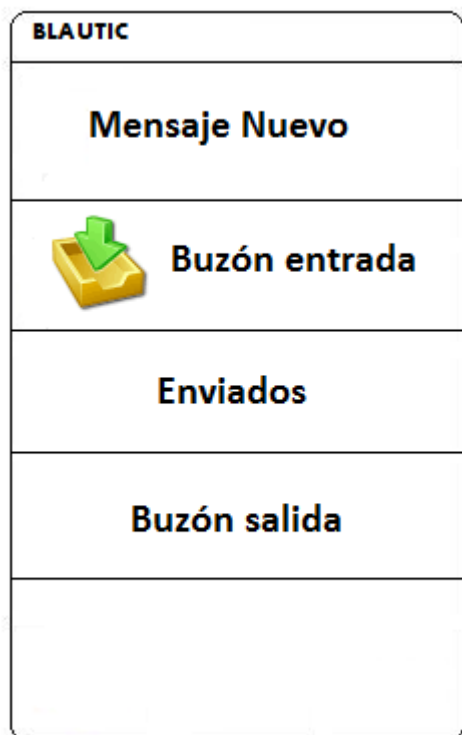


Figura 3.11 Esquema conexión del sistema BlauTic (Fuente: elaboración propia)

Finalmente , cabe diferenciar que existe dos maneras de acceder a la propiedad mensajes: si conectamos con el dispositivo BlauTic, accediendo por lo tanto a sus mensajes almacenados si somos un usuario autorizado (conexión con dispositivo), o accediendo a los buzones de mensajes que están almacenados en la propia memoria del interfaz móvil (sin conexión con el dispositivo).

#### 1) OPCIONES ACCESIBLES DESDE EL TERMINAL (SIN CONEXIÓN CON DISPOSITIVO BLAUTIC)

Desde el botón mensajes del menú principal de nuestra aplicación accedemos a las siguiente pantalla:



Desde la opción “Mensaje nuevo” tendremos la opción de elegir el tipo del mensaje.

- Texto
- Calendario / Acción

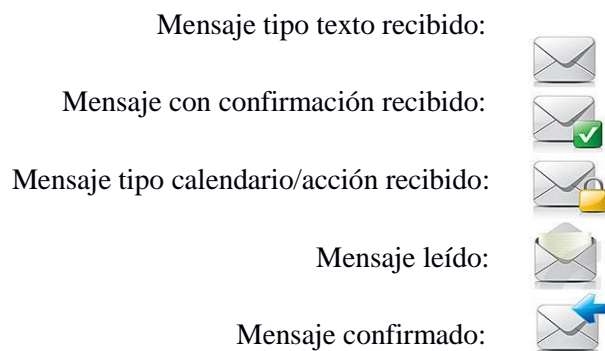
Esta pantalla queda definida con la opción de poder añadir en un futuro más opciones, como la inclusión de plantilla de mensajes predefinidos.

Figura 3.12 Menú mensajes (Fuente: elaboración propia)

#### 1.1. Buzón entrada

Desde la opción Buzón de entrada, accederemos a la pantalla donde estarán listados todos los mensajes almacenados en memoria local, que han sido descargados previa conexión con los dispositivos BlauTic, desde la cual podremos realizar diferentes acciones:

Se definirán diferentes “logos”, para poder identificar visualmente y de forma rápida la información del mensaje, dependiendo del tipo (mensaje texto, mensaje de confirmación, tipo calendario) y del estado de este (leído, sin leer, sin confirmar).



La información mostrada para cada mensaje será la referente al título del mensaje y el *Source* que ha realizado el envío.

Desde el menú visible en la parte inferior de la pantalla, podremos realizad las siguientes acciones: con el botón izquierdo volveremos en el menú contextual hacia atrás; con el botón central podremos reordenar/filtrar los mensajes listados; el botón de la derecha queda reservado para implementar la acción más utilizada en cada pantalla (en este caso sincronizaremos con un servidor remoto).

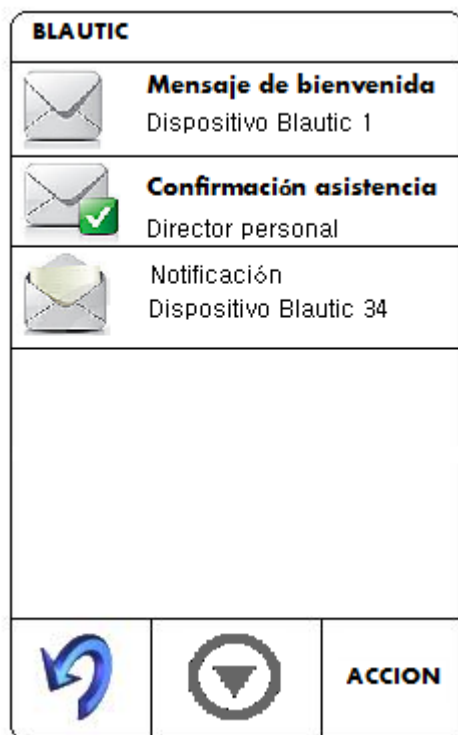
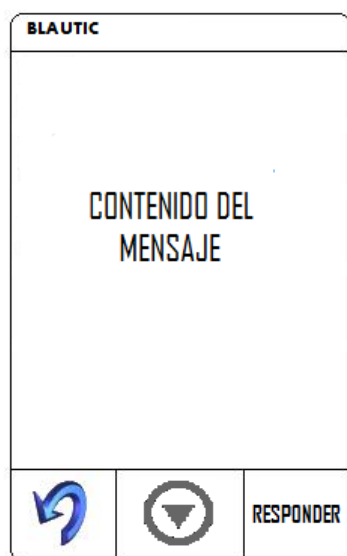


Figura 3.13 Listado de mensajes (Fuente: elaboración propia)

Seleccionando la opción de menú, tendremos la opción de filtrar estos según: dispositivo, source o data.



Una vez filtrados los mensajes según las preferencias del usuario, podremos acceder a la información pulsando sobre cada uno de ellos, obteniendo también diferentes pantallas con diferentes opciones dependiendo del tipo.

Este sería el aspecto de un mensaje de tipo texto. Una vez seleccionado el mensaje, el estado de este pasará a ser de leído.

Con la selección del botón central del menú inferior, podremos realizar las siguientes acciones:

- Marcar como no leído
- Eliminar
- Sincronizar

Figura 3.14 Pantalla escritura de mensajes (Fuente: elaboración propia)

Para los mensaje tipo calendario/acción o que requieran algún tipo de confirmación, el aspecto gráfico de la pantalla sería similar pero en el botón acción rápida tendríamos la opción Confirmar o Validar.

Para los mensaje tipo Calendario/acción (validar) la confirmación deshabilitará las alarmas o avisos que este mensaje lleve asociados.

En los de tipo confirmación (Confirmar), se nos dará la opción de confirmar o no la asistencia a un determinado evento por ejemplo.

Para ambos tipos, las opciones generales desplegadas desde el botón central se mantendrán.

## **1.2. Enviados**

Desde esta opción accederemos a la lista de mensajes que han sido creados y que han sido enviados correctamente.

La lista tendrá un aspecto semejante a las anteriores y también dispondrá de las mismas opciones de menú ya citadas, quedando estas siempre disponibles para posibles ampliaciones. Desde el aspecto visual de los mensajes, podremos realizar un seguimiento rápido y fácil del estado del mensaje, es decir, un pequeño logotipo indicara al usuario si el mensaje enviado ha sido recibido por el destinatario final o por lo contrario aun se encuentra almacenado en el dispositivo Blautic.

## **1.3. Buzón de Salida**

La pantalla del buzón de salida tendrá un aspecto similar a la anteriormente definida pero en ella se almacenarán los mensajes que por algún motivo no se ha podido realizar su envío.

El fallo en el envío puede ser producido porque el dispositivo hacia el cual dirigido el mensaje no se encuentre en alcance en ese momento (pendiente de envío) o se haya producido algún tipo de error en la conexión bluetooth en el momento del envío entre el dispositivo BlauTic y termina móvil.

El reenvío de estos se realizará de forma automática una vez se haya restablecido la conexión en con el dispositivo. Si este envío se realiza correctamente, este pasará por tanto a la lista de mensajes enviados (a efectos de almacenamiento en la base de datos) y a su correspondiente buzón, en caso contrario permanecerá en dicho buzón para futuros intentos de reenvío.

Esto va a permitir al usuario poder trabajar y redactar todos sus mensajes de forma *offline* y que estos se envíen de forma automática al llegar a la zona de cobertura de los dispositivos BlauTic.

Este buzón guarda las mismas opciones de menú definidas anteriormente en el buzón de salida como son la sincronización de mensajes con el servidor o las opciones de filtrado por dispositivo.

## 1.4. Mensaje nuevo.

Al seleccionar la opción de crear un nuevo mensaje, tendremos que elegir el tipo de este: mensaje de Texto/Texto con confirmación o mensaje tipo Calendario/Acción.

Cada uno de ellos guardará algunas peculiaridades especiales que se definirán seguidamente pero otras serán en común en ambos tipos. Esta son y siguiendo el esquema de comunicaciones de mensajes definido al inicio de este apartado:

- Destinatario final. (Para:)  
Donde seleccionaremos de una lista desplegable el *Source* como destinatario final de nuestro mensaje. Permitiremos además la opción de poder generar un mensaje público, es decir, que vaya destinado a todos los Sources de nuestro sistema.
- Dispositivo BlauTic. (Disp:)  
Desde esta lista desplegable, elegiremos el dispositivo BlauTic de nuestro sistema donde queremos depositar el mensaje (canal del envío) pudiendo seleccionar la opción “todos” que conllevará el envío de una copia del mensaje a todos los dispositivos, estemos o no en ese momento en la zona de cobertura Bluetooth.
- Título del mensaje (Titulo:)
- Cuerpo del mensaje.

### 1.4.1. Texto, Texto con confirmación

BLAUTIC		
Dispositivo destinatario		
¿Desea confirmación?		✓
Texto del mensaje		
		ENVIAR

Este es el tipo de mensajes más sencillo donde indicaremos la información a transmitir en el contenido. Además, incorporaremos un botón seleccionable que nos permitirá solicitar al destinatario la confirmación del mismo (pasando entonces a convertirse en un mensaje tipo texto con confirmación).

Este podría ser por tanto un aspecto aproximado de dicha pantalla:

En la parte inferior dispondremos de una serie de botones que nos permitirán: realizar el envío del mensaje, cancelar la creación del mismo y volver a la pantalla anterior o abrir un menú desplegable con diferentes acciones a realizar quedando la definición de estas como un punto a definir.

Se va a dotar a la aplicación de una serie de mensaje de error que surgirán en el momento de enviar el mensaje por si algunos de los campos obligatorio no han sido rellenados o contienen algún tipo de error.

Figura 3.15 Pantalla mensaje tipo Texto (Fuente: elaboración propia)

#### 1.4.2. Mensaje tipo Calendario/acción

The screenshot shows a mobile message interface. At the top, it says 'BLAUTIC'. Below that is a field for 'Dispositivo destinatario'. The main body of the message contains 'Texto del mensaje'. At the bottom, there is a section for 'Alarma sonora' with a clock icon and the text 'Lunes 13:30'. Below this are three buttons: a blue circular arrow (refresh), a grey circle with a downward arrow (options), and a button labeled 'ENVIAR'.

Con este tipo de mensajes seremos capaces de activar una alarma de aviso sobre el terminal móvil al cual va destinado el mensaje. Como vamos a ver, los diferentes parámetros de la alarma podrán ser configurados de forma fácil.

La utilidad de este tipo de mensajes puede verse como una especie de aviso importante ante un determinado evento o simplemente a modo de recordatorio.

En la parte central de la pantalla y debajo del contenido del mensaje dispondremos de un resumen de los aspectos configurados de la alarma:

- Tipo de aviso.
- Duración.
- Fecha: día y hora.

Figura 3.16 Pantalla mensaje tipo Calendario (Fuente: elaboración propia)

Mediante un botón accederemos a la pantalla de configuración de dicha alarma.

Como en el caso anterior, en la parte inferior dispondremos de tres botones desde donde podemos cancelar la creación del mensaje, realizar el envío o acceder al menú de opciones.

La pantalla de configuración de la alarma tendrá aproximadamente este aspecto y en ella podremos editar, siguiendo las especificaciones del diseño, las siguientes características:

- Hora de inicio de la alarma.
- Tipo de aviso: vibración, tono acústico, señal luminosa...
- Días de la semana en la que se activará.
- Intervalo de tiempo entre repeticiones de aviso dentro del mismo día.

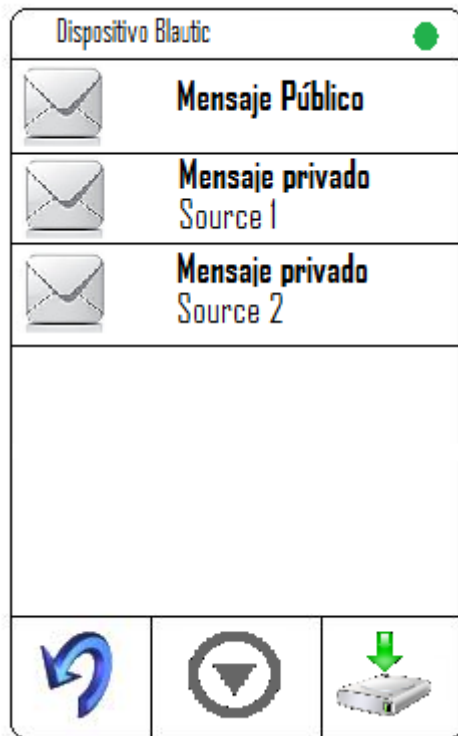
A partir de la información recogida en esta pantalla se rellenará el campo *Schedule* previa codificación de los mismos, siguiendo las especificaciones del diseño.

The screenshot shows the 'Configurar hora de la alarma' screen. It has three input fields for the time: '7', '15', and 'PM'. Below that is a section 'Seleccionar tipo de aviso' with seven checkboxes labeled 'L', 'M', 'X', 'J', 'V', 'S', and 'D'. Underneath is a 'Repetir:' section with a checked checkbox and a 'Cada:' field with an empty input box. At the bottom are two buttons: 'Cancelar' and 'Activar'.

Figura 3.17 Configuración Alarma (Fuente: elaboración propia)



## 2) OPCIONES ACCESIBLES DESDE LA LISTA DE DISPOSITIVOS EN ALCANCE ( CONEXIÓN CON DISPOSITIVO BLAUTIC)



Desde la lista de dispositivos, accesible desde el menú principal de la aplicación, podremos acceder a los mensajes que el dispositivo BlauTic tiene para mostrarnos sean estos de mensajes públicos o privados. El aspecto de la pantalla será el siguiente:

Con el indicador de la esquina superior derecha monitorizaremos en todo momento la conexión al dispositivo.

Con el botón de acceso rápido podremos descargarnos a memoria local todo el bloque de mensajes. Estos serán almacenados en el buzón de entrada para su posterior administración.

*Figura 3.18 Listado mensajes del dispositivo Blautic  
(Fuente: elaboración propia)*

Desde el botón central podremos acceder a diferentes opciones:

- Filtra por *Source*
- Crear nuevo mensaje. Donde accederemos a las pantallas definidas anteriormente.

Como en cualquiera de los buzones, al seleccionar cualquier mensaje accederemos a la pantalla de lectura de los mismos.

### 3.5.1.4 Sensores

Siguiendo las especificaciones , se pasará a diseñar las opciones de visualización y configuración de los sensores que llevan incorporados los dispositivos BlauTic.

La aplicación debe recoger los datos de los diferentes sensores, de todos los dispositivos pertenecientes a nuestro sistema, y monitorizarlos además de permitir la configuración de cada uno de ellos.

Recordaremos que este dispositivo incorpora tres sensores:

- Sensor de temperatura. Salida analógica
- Sensor analógico. Salida analógica
- Sensor digital. Entrada y salida digital.

Recordar también como base esencial de este proyecto , que las comunicaciones entre el terminal móvil y los distintos dispositivos así como la lectura de valores, se realiza mediante la tecnología Bluetooth 4.0.

## 1. OPCIONES ACCESIBLES DESDE EL TERMINAL (SIN CONEXIÓN CON DISPOSITIVO BLAUTIC)

BLAUTIC			
Lista de sensores			
Disp.	Sens1	Sens2	Sens3
453 Sync ON	897	45° Temp.	✓
123 Sync ON	?	-- Temp.	✗
345 Sync OFF	301	55° Temp.	✓
23 Sync OFF	?	37° Temp.	?
<b>Historial</b>			

Figura 3.19 Listado de sensores de los dispositivo Blautic (Fuente:propia)

Desde el menú principal de la aplicación y accediendo al apartado de sensores, visualizaremos una lista con todos los dispositivos asociados a nuestro sistema y el estado de los tres sensores de cada uno de ellos. Para cada sensor se visualizará el valor de este y la fecha de la última medición, si se ha habilitado la opción de monitorización en su configuración en otro caso la visualización del valor solo se realizará con conexión con el dispositivo. Además, una indicación no mostrará si se están sincronizando los valores con el servidor web.

En esta pantalla también dispondremos de un botón de acceso al historial de mensajes tipo sensor de todos los dispositivos, los cuales podremos filtrar como ya hacíamos anteriormente en otros apartados.

Al seleccionar cualquier dispositivo de la lista accederemos automáticamente al menú de configuración de sensores donde ya se realiza la conexión con el dispositivo.

## 2. CONFIGURACION DE SENSORES (CONEXIÓN CON DISPOSITIVO BLAUTIC)

Desde esta pantalla se van a poder monitorizar las mediciones de los sensores y editar de manera rápida algunas de las opciones de configuración más importantes.

Para cada sensor dispondremos de un sencillo *scrollbar* que nos permitirá activar o desactivar de forma rápida la funcionalidad del sensor así como un botón de configuración .

Mostraremos el valor numérico de la medida en el caso de los sensores analógicos y la fecha de la última medición. Para el sensor digital el valor será sustituido por una pequeña imagen que mostrará de forma clara el valor binario en el que se encuentra la salida.

Incorporaremos un botón que nos permitirá leer el valor del sensor manualmente (Leer) en el caso de los sensores analógicos. Con este mismo botón para el caso del sensor







BLAUTIC	
Sensores disp. 463	Sync
Nombre sensor 1	
 227W	Leer
 On 18:45	<input checked="" type="checkbox"/> Monitorizar
Temperatura	
 57°	Leer
 On 19:00	<input type="checkbox"/> Monitorizar
Nombre sensor Digital	
 ✓	<input checked="" type="checkbox"/> Monitorizar
 On	Validar <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
<b>Historial</b>	

Figura 3.20 Listado de sensores de un dispositivo Blautic (Fuente:propia)

digital podremos validar la salida (Validar) según la opción que hayamos seleccionado y realizar el oportuno cambio binario en la salida de este.

Un botón de selección nos permitirá activar la motorización de los valores en el terminal móvil mediante un *broadcast* que realizará el dispositivo. El tiempo de envío de estos valores será un parámetro a configurar.

En la parte superior derecha, incorporaremos un botón desde el cual podremos activar la sincronización de los sensores. Esta acción iniciará el envío de las mediciones de los sensores, a través de la conexión WiFi o 3G, con el servidor remoto que haya sido configurado de la misma manera que se realizaba esta acción en el apartado de mensajes anteriormente descrito. Recordemos que la sincronización será un proceso que se realizará en segundo plano por lo que podremos continuar realizando otras acciones mientras esta se realiza.

Desde esta pantalla también tendremos acceso al historial de mensajes tipo sensor que se hayan podido generar. Pantalla similar a las anteriores donde se manejan listas de mensajes.

Con la selección del botón de configuración accederemos a la siguiente pantalla, donde podremos editar todos los parámetros configurables del sensor indicados en las especificaciones.

Envío de datos	
Broadcast	<input type="checkbox"/>
Conexión	<input type="checkbox"/>
Mensajes	<input type="checkbox"/>
Frecuencia de envío	
Sincronización con servidor	

<b>Nombre</b> Nombre del sensor
<b>Descripción</b> Breve comentario sobre su funcionalidad
<b>Habilitar Sensor</b> <input type="checkbox"/>
<b>Modo manual</b> Se realizarán las medidas cuando se solicite de forma manual
<b>Modo periódico</b> Se realizarán medidas de forma periódica
<b>Frecuencia de envío</b> <input type="radio"/>
<b>Modo interrupción</b> Se registrará el valor ante cualquier cambio en la entrada del sensor

Figura 3.21 Opciones de configuración de los dispositivos Blautic (Fuente: propia)

En base a las especificaciones del productos, podremos editar las siguientes opciones para cada uno de los sensores:

- Nombre del sensor y una pequeña referencia sobre su funcionamiento.
- Botón de que habilite el funcionamiento del sensor.
- Selección del modo de funcionamiento del sensor: manual, periódico o interrupción en el caso del sensor digital. Para el modo periódico una pestaña desplegable nos permitirá seleccionar la frecuencia con la que se tomaran las mediciones.
- Selección del modo de envío de las mediciones al terminal móvil. Recordemos que estas se puedes realizar mediante *broadcast* (opción configurable de forma rápida en la pantalla anterior ), por conexión o quedar registradas en forma de mensajes. También podemos definir la frecuencia con la que se realizará el envío de los datos mediante una pestaña desplegable.
- Otro parámetro configurable nos permitirá editar opciones relacionas con la sincronización de los datos con el servidor remoto.

Recordar que todas estas opciones de configuración, la de los tres sensores de todos los dispositivos que forman nuestro sistema, estarán almacenadas en la base de datos de la aplicación.

### 3.5.2 Esquema de la aplicación

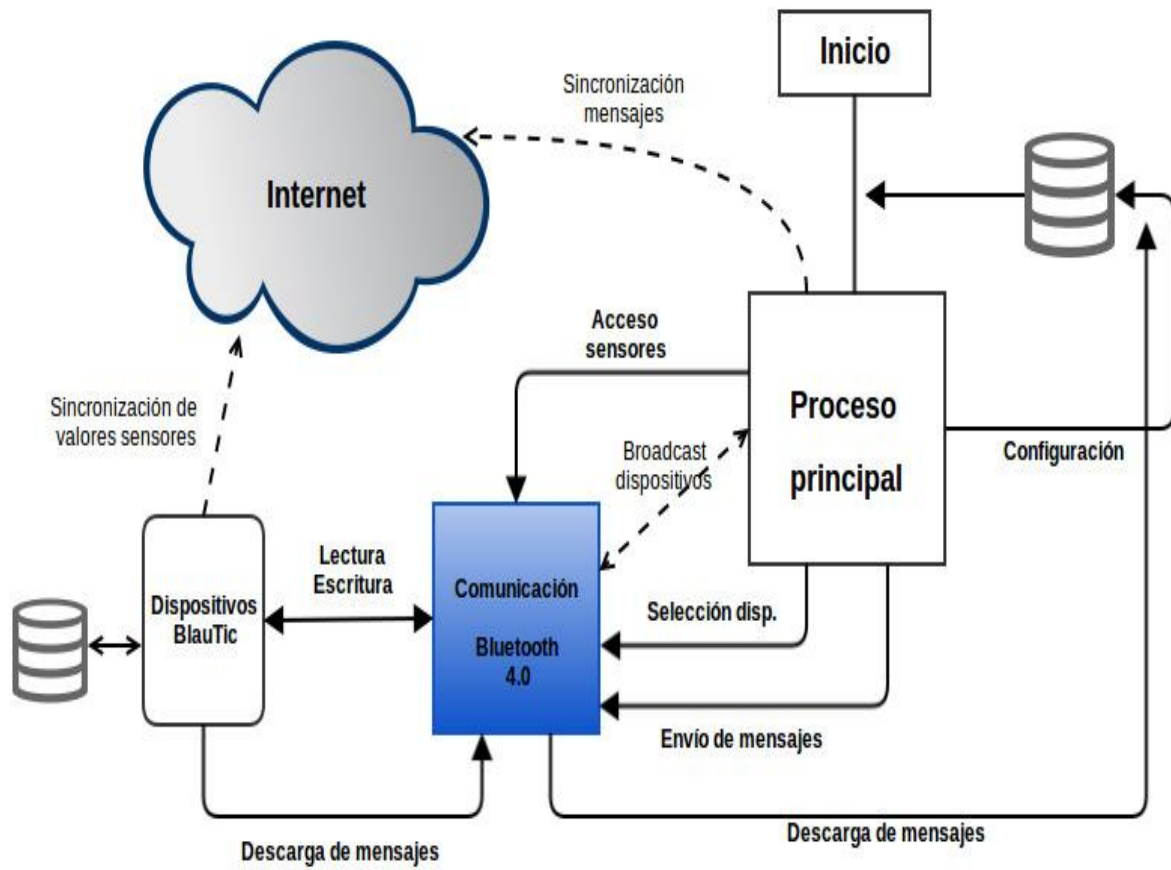


Figura 3.20 Esquema de la aplicación Blauticc (Fuente: elaboración propia)

## 3.6. Desarrollo

El desarrollo de la aplicación se ha completado con la consecución de distintos hitos tecnológicos que hemos ido abordando y en los que hemos ido completando las distintas funcionalidades de la aplicación. En este apartado queremos abordar de manera más detallada los aspectos que a mi entender han resultado más importantes .

### 3.6.1.1. Sincronización

La sincronización de datos entre el terminal móvil y un servidor web es una de las funcionalidades a destacar de esta aplicación. Concretamente esta opción estará disponibles en los apartados de mensajes y sensores para realizar el envío a través de la red de los valores obtenidos en las mediciones de los sensores o de los mismos mensajes para un posterior análisis o procesado.

La conexión a la red se realizará mediante el WiFi del dispositivo móvil o haciendo uso de conexiones 3G. En todo momento se informará al usuario si se dispone de conexión o no y en el caso de que esta sea 3G, de los gastos que esto puede ocasionar si no se dispone de una tarifa acordada con la operadora de telefonía.

Estas notificaciones se realizarán mediante pantallas emergente en el momento de seleccionar el botón “sincronizar” y se dará la opción de cancelar la acción si no se desea continuar. Además, esta acción está relacionada con la opción de configuración “habilitar envío de datos” que el usuario habrá seleccionado o no previamente. El mensaje emergente por tanto advertirá si tenemos el permiso activado a nivel de aplicación además de realizar una comprobación más general de este permiso a nivel de dispositivo móvil . Recordemos que todos los smartphones, en el apartado de configuración de redes , tienen la opción de poder habilitar o no el envío de datos a través de red es decir, si permiten a cualquier aplicación que tenga instalada utilizar este servicio.

A continuación vamos a explicar desde un punto más técnico como se ha desarrollado esta parte de la aplicación y para ello empezaremos definiendo el funcionamiento básico de los *sockets*, que es la herramienta de Java elegida para realizar este tipo de conexiones, haciendo referencia a la definición descrita por el profesor Jean-Merie Rifflet en su libro “*Comunicaciones en UNIX*”.

Los *sockets* son un sistema de comunicación entre procesos de diferentes máquinas de una red. Más exactamente, un *socket* es un punto de comunicación por el cual un proceso puede emitir o recibir información.

Fueron popularizados por *Berkeley Software Distribution*, de la universidad norteamericana de Berkeley. Los *sockets* han de ser capaces de utilizar el protocolo de streams TCP (Transfer Control Protocol) y el de datagramas UDP (User Datagram Protocol).

Utilizan una serie de primitivas para establecer el punto de comunicación, para conectarse a una máquina remota en un determinado puerto que esté disponible, para escuchar en él, para leer o escribir y publicar información en él, y finalmente para desconectarse.

Con todas primitivas se puede crear un sistema de diálogo muy completo.

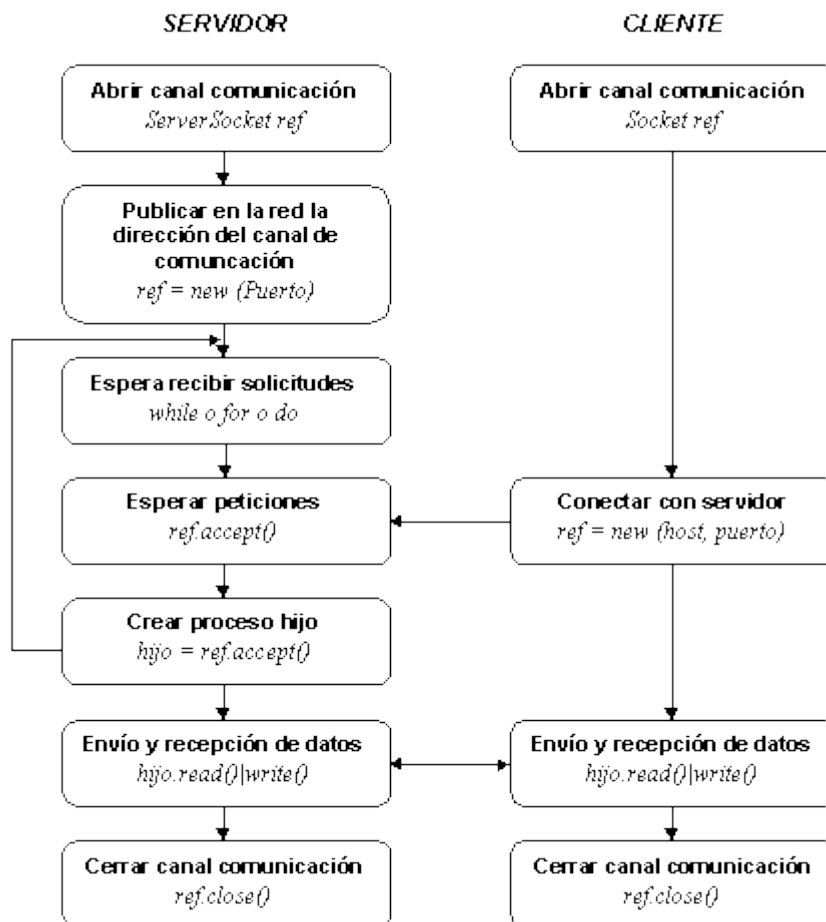


Figura 3.22 Esquema comunicaciones socket (Fuente: "Comunicaciones en UNIX", Jean-Merie Rifflet)

Haciendo referencia a nuestro código y a las primitivas utilizadas, el programa cliente reside evidentemente en nuestro dispositivo móvil. Este programa se conecta a un servidor indicando el nombre de la máquina y el número de puerto en el que el servidor está instalado.

Una vez realizada la comprobación de que la conexión se ha establecido correctamente, se codifican los datos a enviar, se envían y se espera la respuesta por parte del servidor. El programa cliente recoge esta respuesta y después de decodificarla muestra el resultado al usuario indicando si esta ha sido correcta (OK) o en caso contrario una referencia al error producido (ERR\_X).

[...]

```

try {

    notifyIcon(this._context,CONNECTING,null);
    //open the socket
    Socket mySocket = new Socket(myIP,myPort);
    //connection information
    notifyIcon(this._context,CONNECT,"Conetando con Host: " + myIP);

    mySocket.setSoTimeout(2000);
    //evaluate if the connection by socket is available
  
```

```

if(mySocket.isConnected()){
    //send messages of array.
for(int i= 0; i< ListMssg.size(); i++){
if(ListMssg.get(i).getLocalStatus() !=Message.MESS_LOCAL_SYNCSEV){
    notifyIcon(this._context,SENDING,"Enviando datos..." + i + "/" + ListMssg.size());

//Parse message to XML
String message = ListMssg.get(i).toXML();
messageTest=message;
//translate in bytes
byte[] buffer = message.getBytes();
byte[] b = new byte[200];

//Replay three times if the Timeout expires
for(int error= 0; error<3; error++){
    //write the message on the Output buffer
    mySocket.getOutputStream().write(buffer);
try{
    //read the Input Buffer
mySocket.getInputStream().read(b);
    // parse the xml answer to extract the response: OK or Error
String a = new String(b);
    ArrayList<String> response = rx.parseXML(a);
    this.StringResult =rx.result(response);

    break;
    }catch(InterruptedException e){
        notifyIcon(this._context,ERROR_NUM,"Time Out");
StringResult="ERR_0";
    }
    }
[.....]
if(StringResult.equals("OK")){
    Message m = ListMssg.get(i);
    [ . . . ]
}else{
    this.numErr ++ ;
    }

    }
}
}else{
// no connected
notifyIcon(this._context,ERROR_NUM,"No conectado");
}

//result of sincronization
notifyIcon(this._context,FINISH,"" + numErr + "/" + ListMssg.size() + " tipo " + StringResult);

```



```

}catch (UnknownHostException e) {
    notifyIcon(this._context,ERROR_NUM,"Error al conectar con el servidor.");
}catch (IOException e) {
    notifyIcon(this._context,ERROR_NUM,"Conexión con " + myIP +" fallida");
}catch (SecurityException e) {
    notifyIcon(this._context,ERROR_NUM,"Permiso de acceso a red denegados");
mySocket.close();
}

```

Vemos como en primer lugar creamos el socket llamado mySocket al que especificamos un host (myIP) y un número de puerto (myPort) que en este caso son dos parámetros rescatados de la base de datos de la aplicación y que han sido configurados por parte del usuario en las opciones de configuración de datos.

Luego mediante la primitiva ().isConnected() testamos si ha sido posible realizar la conexión para pasar luego a asociar el flujo de datos a nuestro socket mediante mySocket.getOutputStream().write(buffer).

Decir que la información ha sido previamente codificada siguiendo las especificaciones mediante código XML donde las etiquetas y campos de dicho documento variarán dependiendo de si estamos codificando información referente a mensaje o a sensores.

Posteriormente, analizaremos la respuesta obtenida por parte del servidor con la primitiva mySocket.getInputStream().read(b) donde al igual que antes, el flujo de datos será decodificado y se extraerá del documento XML obtenido la etiqueta correspondiente que nos indicará si la transferencia ha sido correcta o no.

Como mecanismo de corrección de error, a este procedimiento de escucha se le permitirán tres acuses de recibo erróneos en cuyo caso se reintentará la transmisión de datos antes de darse esta por errónea y ser notificada al usuario.

Finalmente observamos como una vez finalizadas las operaciones, el socket se cierra mediante el método .close() y además podemos observar la gestión de excepciones realizada para capturar los posibles fallos tanto de los flujos de datos como del socket.

Querría indicar que todo este proceso de sincronización en la aplicación, ha sido realizado con una clase especial de programación en Android llamada AsyncTask. Esta clase permite que los procesos se realicen de forma asíncrona en segundo plano, es decir, sin interferir al usuario en la navegación por la aplicación o impedir la realización de otra tarea diferente. Además indicar que el usuario esta en todo momento informado del estado de esta tarea, ya que en la parte superior del dispositivo móvil, en la barra de notificaciones, se indica toda esta información así como el reporte de errores.

### 3.6.1.2. Bluetooth

En este apartado queremos tratar de una manera un poco más detallada como se ha implementado el código fuente de nuestra aplicación a la hora de establecer las comunicaciones por bluetooth entre el terminal móvil y los dispositivos BlauTic.

Se ha utilizado la clase java nativa en Android `android.bluetooth.BluetoothAdapter` para realizar tareas básicas como habilitar/deshabilitar el adaptador Bluetooth del móvil o el descubrimiento de dispositivos dentro del rango de acción entre otras funcionalidades . Esta última nos permitía obtener un listado de todos los dispositivos en alcance y obtener información básica como la dirección MAC del mismo que a su vez se empleaba para determinar si el dispositivo bluetooth pertenecía a nuestro sistema BlauTic o no.

```
private void scan() {

    //Discover devices in range
    mBluetoothAdapter.startDiscovery();

    // Create a BroadcastReceiver for ACTION_FOUND
    mReceiver = new BroadcastReceiver() {

        public void onReceive(Context context, Intent intent) {

            String action = intent.getAction();
            // When discovery finds a device

            if (BluetoothDevice.ACTION_FOUND.equals(action)) {
                // Get the BluetoothDevice object from the Intent
                BluetoothDevice device =
                intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
                // Add the MAC address to an array adapter to show in a ListView
                blueDev.add(device);
            }

            if(BluetoothAdapter.ACTION_DISCOVERY_STARTED.equals(action)){
                Log.d("BackListStarted", "OK");
                blueDev.clear();
                notifyIcon(context, 0, "Buscando dispositivos");
            }
            if(BluetoothAdapter.ACTION_DISCOVERY_FINISHED.equals(action)){
                notifyIcon(context, 1, "Búsqueda finalizada");
            }
        }
    };
};
```

Al momento de la implementación de este proyecto, no se disponía aun de la API de Google para trabajar con los dispositivos BLE (*Bluetooth Low Energy*). Esta API es la que debe dotar a la aplicación de todas las funcionalidades necesarias para establecer comunicaciones con los dispositivos BLE, como son los dispositivos BlauTic.

Por todo ello, para la implementación de las funciones básicas de comunicación bluetooth se ha hecho uso un paquete de librerías implementadas por Motorola:

```
import com.motorola.bluetoothle.BluetoothGatt;  
import com.motorola.bluetoothle.IBluetoothGattCallback;
```

Con la utilización de ciertas subclases, se han conseguido realizar funciones de lectura y escritura sobre los dispositivos BlauTic como la recogida de los datos de los sensores con los métodos `ReadGattCharacteristicValue(device, uuid, handle)`.

Destacar que al tratarse de una API experimental y en cierto modo “no oficial” en un principio no ha sido posible implementar la funcionalidad de recogida de datos en modo broadcasts. Un sistema con el que se pretendía dejar el Smartphone en modo escucha (sin conexión directa con los dispositivos) y que siguiera recogiendo datos y eventos emitidos por los dispositivos BlauTic. Esto ha implicado que para la lectura de datos de los sensores ha sido necesario siempre establecer una conexión previa con el dispositivo.

### 3.6.1.3. Recursos XML

En el siguiente diagrama se representa la relación entre todos los recursos .xml , .java y las pantallas utilizadas en la aplicación. Esta estructura modular nos permite realizar modificaciones y actualizaciones tanto en el código como en el aspecto gráfico de la aplicación de manera casi independiente.

La implementación de los aspectos gráficos de las pantallas se ha realizado con la herramienta que por defecto integra el SDK de Android. El hecho de que el formato de los recursos gráficos sea .xml nos permite de forma fácil rediseñar las pantallas con cualquier otro software incluso se podría externalizar esta tarea mediante técnicas más avanzadas de diseño gráfico.

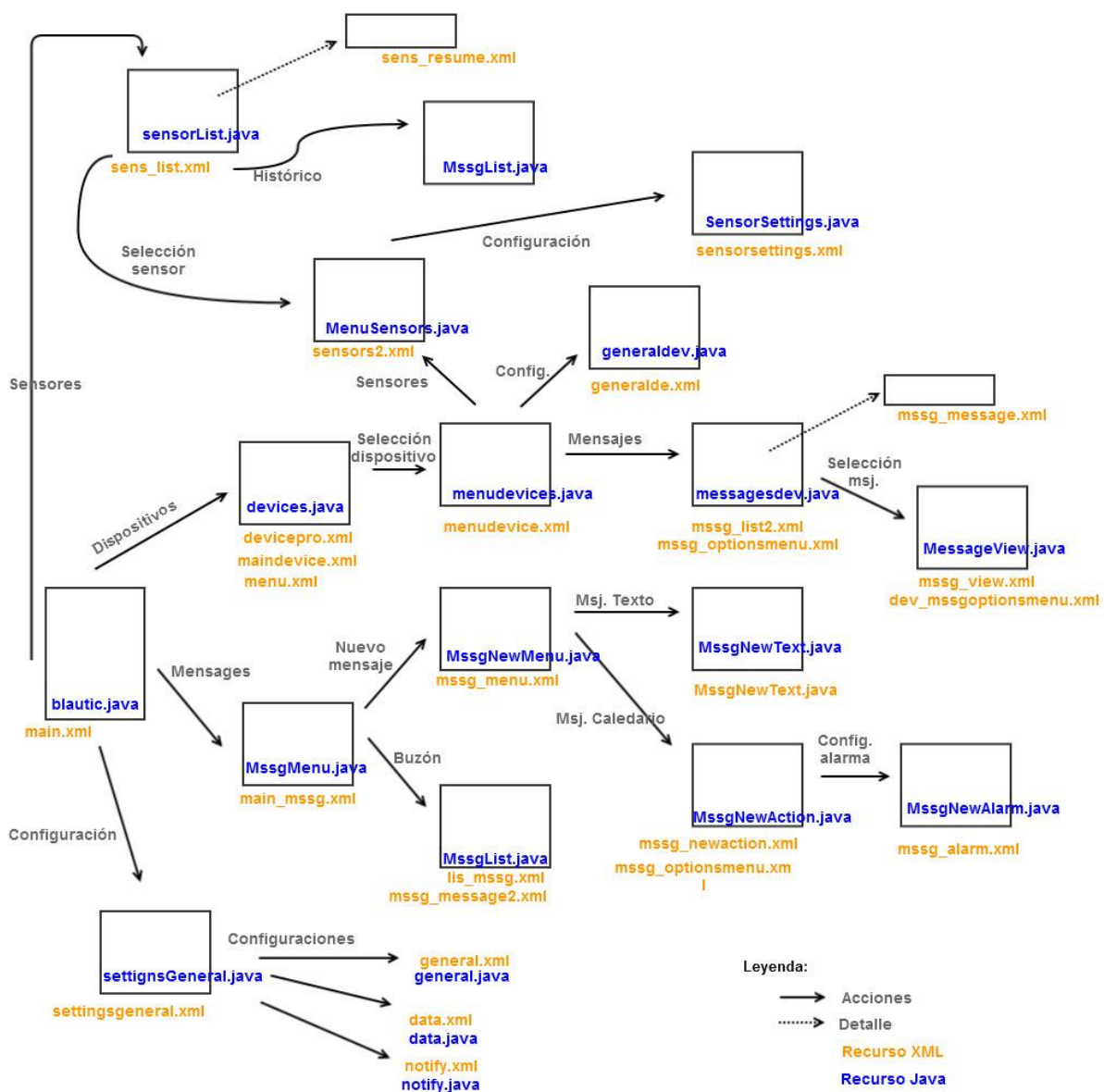


Figura 3.23 Esquema de recursos xml y java de la aplicación (Fuente: elaboración propia)

### 3.6.2. Problemas

El principal problema que nos hemos encontrado en la ejecución de este proyecto, y que ya hemos comentado en apartados anteriores, ha sido el no disponer de la API oficial de Google para dispositivos BLE (*Bluetooth Low Energy*). En su lugar hemos utilizado unas librerías compatibles con Android y producidas por Motorola pero sin llegar a completar al 100% las especificaciones de la aplicación, concretamente en lo referente a la notificación de mensajes en modo broadcast.

También hemos tenido que realizar numerosos ajustes en el diseño de los xml que generaban los recursos gráficos de los menús y pantallas de la aplicación ya que no llegábamos a conseguir un diseño que se ajustará por igual a las distintas dimensiones de pantalla de los smartphones utilizados. En concreto los modelos utilizados para el testeo de la aplicación han sido un HTC Tatum, que dispone de una pantalla de 2,8 pulgadas, y un Motorola Razr equipado con una pantalla SuperAmoled de 4,3 pulgadas.

Aunque el resultado final de la aplicación fue implementada para funcionar en el Motorola Razr, que era el smartphone que disponíamos con la tecnología bluetooth 4.0 y una versión de Android superior, gran parte del desarrollo se ha realizado con el HTC Tatum teniendo que simular en muchas ocasiones las comunicaciones y alguna funcionalidad. Además con este dispositivo era mucho más difícil poder depurar los errores de código ya que teníamos constantes problema a la hora de conectar y sincronizar el teléfono con el SDK que tenía instalado en el ordenador. Esta herramienta, de gran utilidad y cuyo manual de instalación está indicado en el anexo de esta memoria, sirve a modo depurador o compilador del código. El SDK nos permite monitorizar todos los eventos que tiene lugar en el móvil y en caso de error nos da una pista de que funcionalidad es la que ha bloqueado o abortado la aplicación.

## 4. Resultados

### 4.1. Pantallas definitivas

En el siguiente apartado queremos mostrar una selección de diferentes capturas de pantallas (realizadas con el HTC one) resultado de la primera versión de nuestra aplicación. Podemos ver la distribución y relación de las mismas en la gráfica que se mostró con anterioridad en el apartado 3.6.2.3. de esta memoria.

#### PANTALLA INICIAL Y MENUS DE CONFIGURACION

La captura de la izquierda hace referencia a la pantalla inicial o menú principal de la aplicación. En ella se puede ver una distribución de cuatro botones que dan acceso a las opciones principales de esta aplicación: Dispositivos, Mensajes, Sensores y Seguridad.

En la parte inferior se encuentra el botón de acceso a la configuración general de la aplicación. En esta opción podemos acceder a la información básica del dispositivo como: versiones del software, direcciones MAC, nombre del dispositivo, etc... además de configurar ciertos parámetros necesarios para el funcionamiento de la aplicación como: habilita/deshabilitar el bluetooth, activar la sincronización remota de datos con el servidor, actualizar la base de datos, etc..

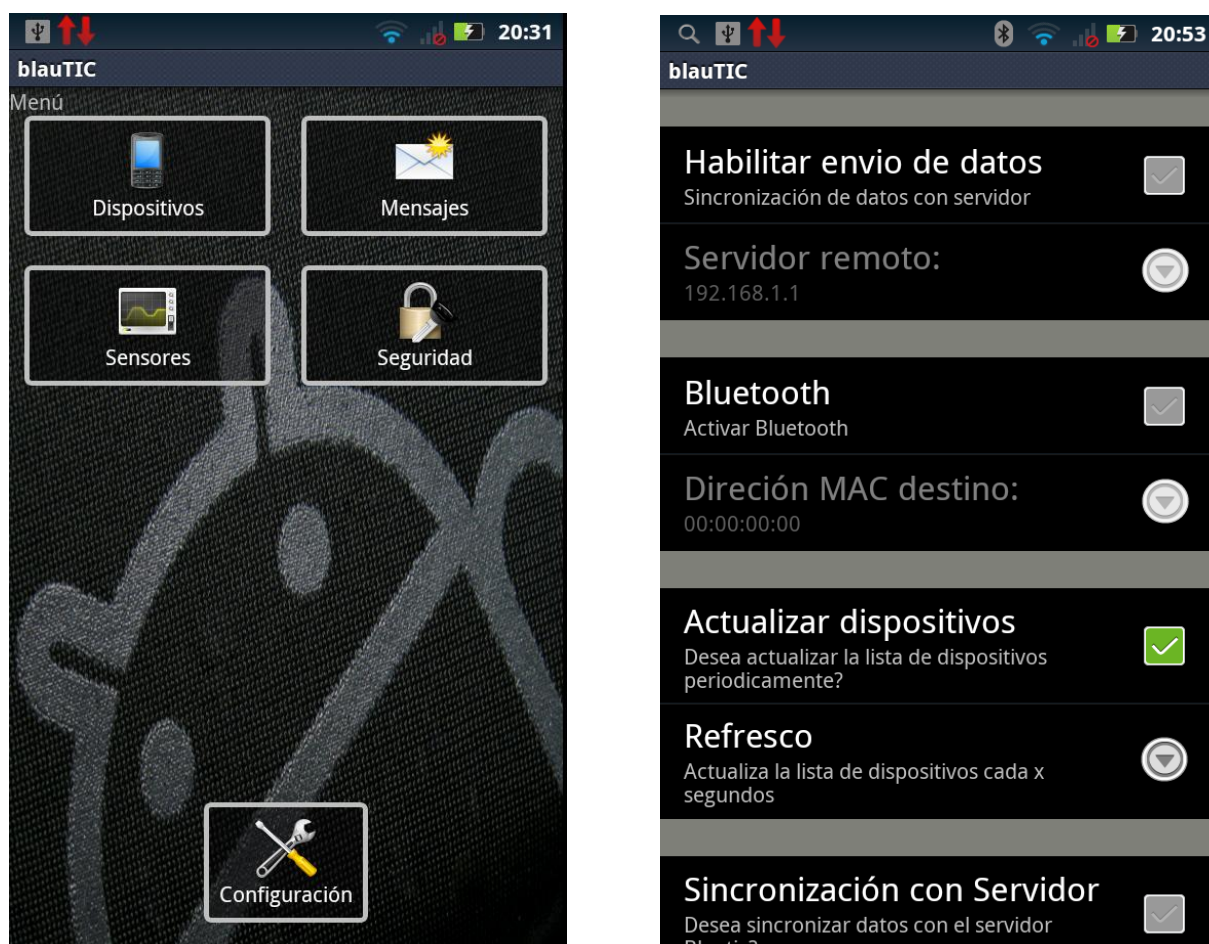


Figura 4.1 Capturas de pantalla 1 (Fuente: elaboración propia)

## LISTADO DE DISPOSITIVOS

La pantalla de “Dispositivos” nos permite obtener el listado de los dispositivos BlauTic A1/B1 que pertenecen a nuestro sistema. Además nos va a mostrar una pequeña información sobre la identidad del dispositivo (dirección MAC y nombre) y las funcionalidades activas en él: sensor, mensajes, alarmas o localización.

Si el dispositivo BlauTic se encuentra dentro de nuestra zona de cobertura bluetooth, aparecerá una señalización luminosa de color verde a la izquierda de la imagen/logotipo del dispositivo.

Las opciones del menú de la parte inferior, que de inicio aparecen ocultas pero que una vez seleccionado el botón de “menú” del dispositivo vuelve a aparecer, nos permiten filtrar los dispositivos según se requiera.

Además, aunque el refresco de la lista de dispositivos es continuo según nos vayamos acercando de la zona de cobertura de estos, se dispone de un botón que fuerza la acción de descubrimiento de los mismos.

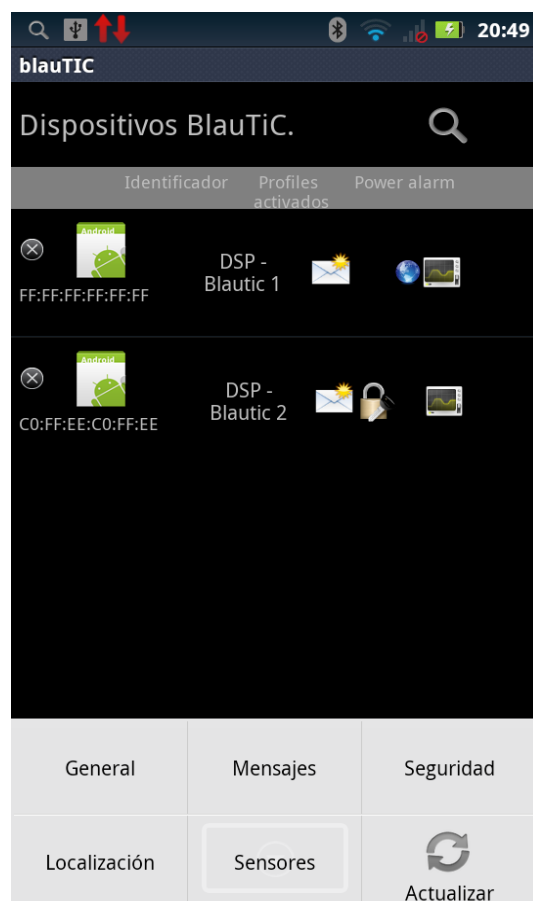


Figura 4.2 Capturas de pantalla 2 (Fuente: elaboración propia)

A continuación vemos el resultado de las pantallas referentes a los mensajes.

## MENSAJES

Desde los distintos menús podremos acceder a listado de mensajes enviados y recibidos así como crear un mensaje tipo texto, tipo calendario o tipo texto con confirmación desde cero.

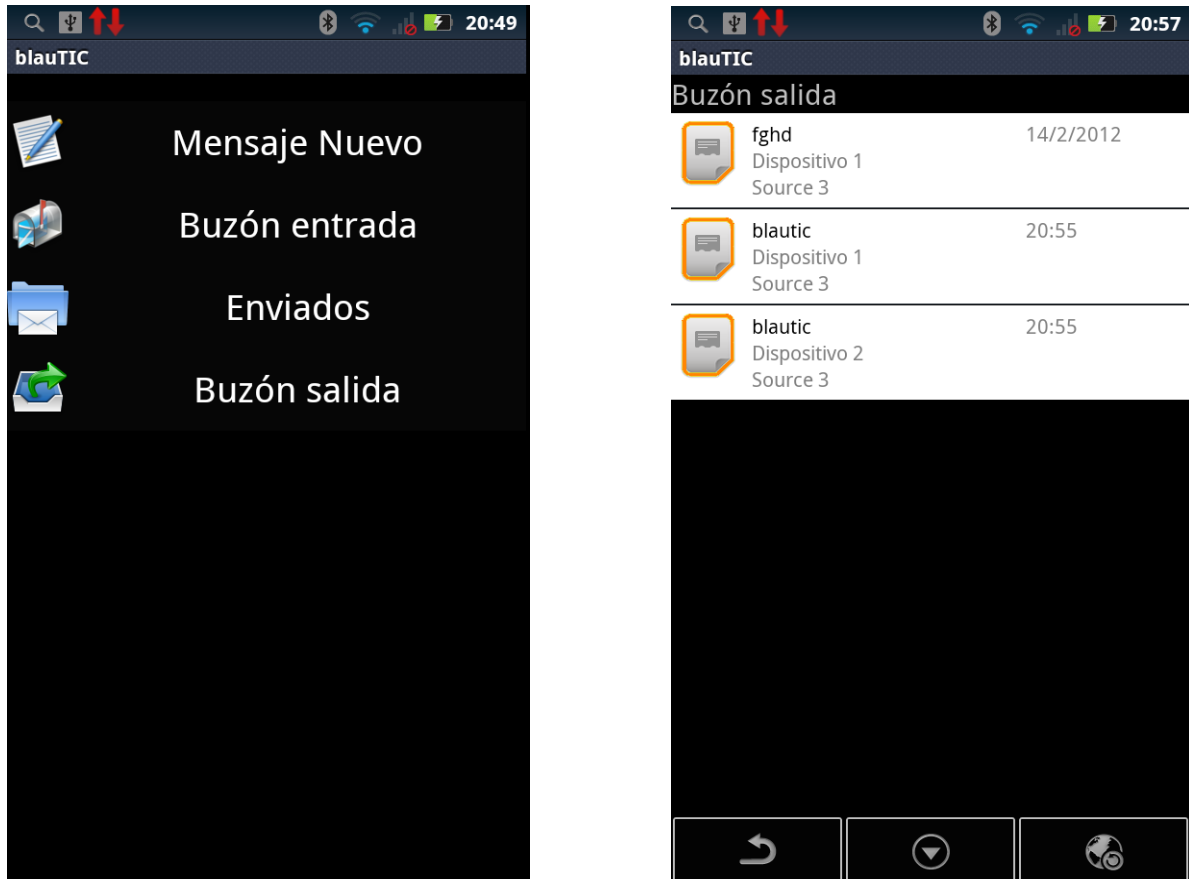


Figura 4.3 Capturas de pantalla 3 (Fuente: elaboración propia)

La pantalla de navegación entre mensajes (captura a la derecha) se realiza a modo de listado pudiendo acceder al contenido de los mismos seleccionándolo el mensaje en cuestión. Además se dispone de un menú desplegable (con la tecla menú) en el que se nos permite filtrar por dispositivo.

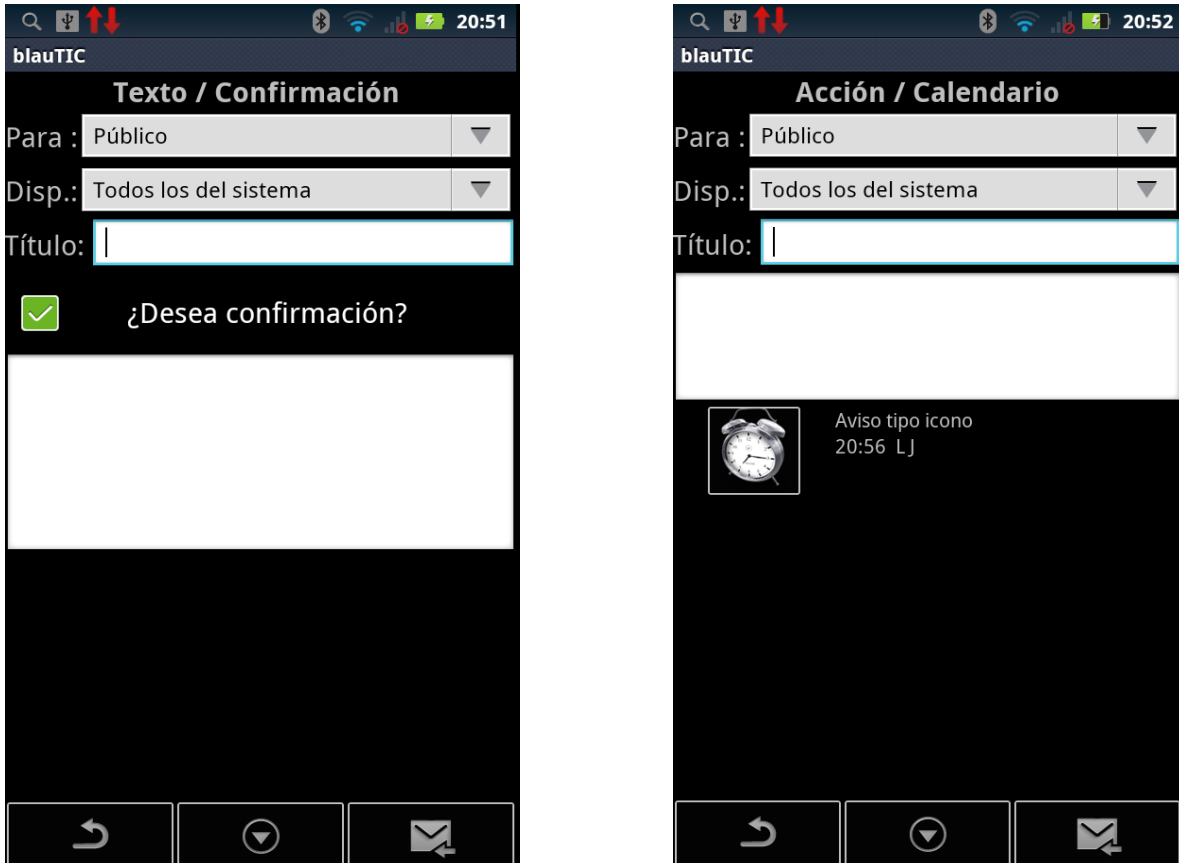
Se ha incorporado también en esta pantalla el botón de “sincronización” (parte inferior derecha). Este permite realizar un volcado masivo de los mensajes a través de la red a un servidor remoto a la dirección se ha configurado previamente en las opciones generales de la aplicación.

En las siguientes capturas se muestran las pantallas de creación de los distintos tipos de mensajes. Siempre se podrá definir si el receptor de los mensajes es uno de los dispositivos de nuestro sistema o si de lo contrario se trata de un mensaje de difusión o broadcast.

Destacar las numerosas opciones de configuración que se han definido en la pantalla de “alarma” y que la dotan de una gran versatilidad a la hora de definir los mensajes de aviso.



## MENSAJES TIPO TEXTO Y CALENDARIO



## COFIGURACION ALARMA



Figura 4.4 Capturas de pantalla 4 (Fuente: elaboración propia)

## SENSORES

Desde la opción “Sensores” del menú principal se accede al listado de sensores de los dispositivos BlauTic A1/B1 de nuestro sistema. Al igual que la opción de “Dispositivos”, en el listado se indicará con una señalización de color verde si el dispositivo BlauTic se encuentra dentro de nuestro rango de cobertura o no. También dispondremos de un pequeño *display* a modo de resumen donde pondremos realizar una motorización a tiempo real de los sensores analógicos y digitales.

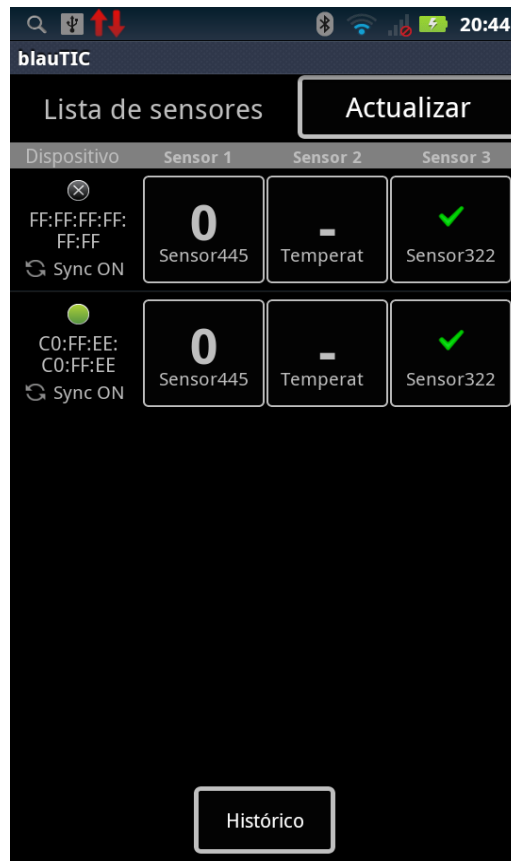


Figura 4.5 Capturas de pantalla 5 (Fuente: elaboración propia)

En la parte inferior de la pantalla se ha dispuesto de un botón donde se almacenan las lecturas de los sensores para su posterior tratamiento.

Seleccionando cualquiera de los cualquiera de ellos, accedemos a las pantallas principales de los dispositivos BlauTic donde podemos ver con mas detalles la lectura de los sensores así como realizar todas las opciones de configuración del dispositivo que se requiera.

Con los distintos botones que se han implementado podemos activar o desactivar cada uno de los sensores del dispositivo así como realizar lecturas manuales (botón leer) o activar la opción “monitorizar” que realizará la lectura automática del valor del sensor cada cierto intervalo de tiempo.

Con el botón de configuración se nos permiten ajustar aspectos generales como el nombre abrevado del dispositivo BlauTic (que aparecerá como información rápida en el listado de sensores o dispositivos) así como una breve descripción de su funcionalidad.

Además la pantalla de configuración nos permite definir los distintos tipos o modos de monitorización de los sensores que se definieron en las especificaciones de este proyecto: modo manual, broadcast, etc...



Figura 4.6 Capturas de pantalla 6 (Fuente: elaboración propia)

Por último, en la parte inferior de la pantalla se han dispuesto un par de botones más. La opción de “sincronizar con el servidor” que ejecuta la acción de envío de los datos de las mediciones a un servidor remoto para su monitorización o análisis de datos y el botón “Histórico” donde encontramos, con mayor detalle, el registro de las mediciones de los sensores del dispositivo.

## 4.2 Test de pruebas

Las pruebas de campo para testear la aplicación se han realizado con el terminal Motorola Razr y varios dispositivos BlauTic. Colocando los dispositivos a diferentes distancia del móvil hemos conseguido que la aplicación rastree y descubra los dispositivos bluetooth en alcance listando únicamente los que pertenecen a nuestro sistema. Previamente se han incorporado las direcciones MAC de los mismos en la base de daos de la aplicación. Una vez establecida la conexión bluetooth con los dispositivos hemos sido capaces de leer valores de los dos sensores analógicos que incorporara así como realizar operaciones de lectura/escritura sobre el otro sensor digital.



*Figura 4.7 Componentes del sistema Blautic (Fuente: elaboración propia)*

Una vez recogidos los datos de los sensores hemos realizado la "sincronización" de estos con un servidor remoto. Hemos podido confirmar como se reciben en el servidor, tanto a través de la red Wifi como la red 3G del móvil, las trazas xml con la información de los sensores y como la aplicación es capaz de interpretar la respuesta del mismo para indicar si la transacción se completó correctamente o no.

Todos los menús y pantallas de configuración de la aplicación como la de los dispositivos BlauTic donde determinamos que sensores quedan habilitados, nombre y descripción, etc... también funciona correctamente.

La escritura y almacenaje de los mensajes tipo texto, texto con confirmación y tipo calendario también se ha dejado operativa en la aplicación.

En el apartado anterior de esta memoria podemos ver el aspecto final de las pantallas y menús de navegación.

Como evidencia de las comunicaciones bluetooth 4.0 establecidas entre los dispositivos Blautic y el terminal móvil hemos utilizado un software tipo "sniffer" para capturar las trazas.

Las capturas que se muestran corresponden a las principales funciones utilizadas en las comunicaciones como son: descubrimiento de dispositivos, lectura de datos, escritura, notificaciones, etc..entre el dispositivo Blautic con dirección MAC C0:FF:EE:C0:FF:EE y el Motorola Razr con dirección MAC del bluetooth 5C:6B:32:B4:07:F5

### EMISIONES BLUETOOTH DEL DISPOSITIVO

P.nbr.	Time [µs]	Channel	Access Address	Adv PDU Type	Adv PDU Header	AdvA	AdvData	CRC	RSSI [dBm]	FCS
1	+0	Ox25	Ox6E99ED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	OxC0FFEEC0FFEE	02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Ox102DDE	-40	OK
2	+104997	Ox25	Ox6E99ED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	OxC0FFEEC0FFEE	02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Ox102DDE	-40	OK
3	+99996	Ox25	Ox6E99ED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	OxC0FFEEC0FFEE	02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Ox102DDE	-40	OK
4	+101873	Ox25	Ox6E99ED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	OxC0FFEEC0FFEE	02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Ox102DDE	-40	OK
5	+103372	Ox25	Ox6E99ED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	OxC0FFEEC0FFEE	02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Ox102DDE	-40	OK
6	+107498	Ox25	Ox6E99ED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	OxC0FFEEC0FFEE	02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Ox102DDE	-40	FCS ERROR
7	+104372	Ox25	Ox6E99ED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	OxC0FFEEC0FFEE	02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Ox102DDE	-40	OK
8	+108120	Ox25	Ox6E99ED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	OxC0FFEEC0FFEE	02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Ox102DDE	-40	OK

Figura 4.8 Captura sniffer 1 (Fuente: elaboración propia)

### CONNECTION REQUEST ( )

23	+99996	Ox25	Ox6E99ED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	OxC0FFEEC0FFEE	02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Ox102DDE	-40	OK
24	+106872	Ox25	Ox6E99ED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	OxC0FFEEC0FFEE	02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Ox102DDE	-40	OK
25	+537	Ox25	Ox6E99ED6	ADV_CONNECT_REQ	Type TxAdd RxAdd PDU-Length 5 0 0 34	Ox5C6B32B407F5	AccessAddr CRCInit WinSize MinOffset Interval Ox55E9993E 46 9A 17 08 0x0000 0x0100	LLData (Part 1)	Latency Timeout DM 0x0000 0x0064 1F FF FF FF F	
26	+278922	Ox0E	Ox55E9993E	L2CAP-C	Data Header LLID NESH SN MD PDU-Length 1 0 0 0 0	OxCDB378				
27	+331	Ox0E	Ox55E9993E	L2CAP-C	Data Header LLID NESH SN MD PDU-Length 1 1 0 0 0	OxCDB5AE				
28	+219770	Ox1C	Ox55E9993E	Control	Data Header LLID NESH SN MD PDU-Length 1 1 1 0 5	Veretoo_End(OK)E	LL_Verison Veretoo_CompID SubVeretoo 0x00 0x0000 0x2834	Ox8796C4		
29	+4379	Ox1C	Ox55E9993E	L2CAP-C	Data Header LLID NESH SN MD PDU-Length 1 0 1 0 0	OxCDBEDE				
30										

Figura 4.9 Captura sniffer 2 (Fuente: elaboración propia)

## NOTIFICATION ( )

P.nbr.	Time [us]	Channel	Access Address	Data Type	Data Header	CRC	RSSI [dBm]	FCS
34	+319772 =5973151	Ox21	Ox95E9938E	L2CAP-C	LLID NESH SH MD PDU-Length 1 0 0 0	OxCDB378	-53	OK
35	+230 =5973381	Ox21	Ox95E9938E	L2CAP-C	LLID NESH SH MD PDU-Length 1 1 0 0	OxCDB5AB	-51	OK
36	+319773 =6293154	Ox0A	Ox95E9938E	L2CAP-C	LLID NESH SH MD PDU-Length 1 1 1 0	OxCDB80D	-29	OK
37	+230 =6293384	Ox0A	Ox95E9938E	L2CAP-S	LLID NESH SH MD PDU-Length 2 0 1 0	L2CAP Header L2CAP-Length ChanId 0x0005 0x0004	ATT_Handle_Value_Notif Opcode AttHandle AttValue 0x1E 0x047 02 00	CRC RSSI FCS 0x83F508 -46 OK
38	+319771 =6413155	Ox18	Ox95E9938E	L2CAP-C	LLID NESH SH MD PDU-Length 1 0 0 0	OxCDB378	-50	OK
39	+231 =6413386	Ox18	Ox95E9938E	L2CAP-C	LLID NESH SH MD PDU-Length 1 1 0 0	OxCDB5AB	-53	OK
40	+319742 =6933128	Ox01	Ox95E9938E	L2CAP-C	LLID NESH SH MD PDU-Length 1 1 1 0	OxCDB80D	-30	OK

Figura 4.10 Captura sniffer 3 (Fuente: elaboración propia)

## READ REQUEST ( )

P.nbr.	Time [us]	Channel	Access Address	Data Type	Data Header	CRC	RSSI [dBm]	FCS
81	+302 =11333404	Ox16	Ox95E9938E	L2CAP-C	LLID NESH SH MD PDU-Length 1 0 1 0	OxCDBEED	-52	OK
82	+319697 =11653101	Ox24	Ox95E9938E	L2CAP-C	LLID NESH SH MD PDU-Length 1 0 0 0	OxCDB378	-39	OK
83	+431 =11653332	Ox24	Ox95E9938E	L2CAP-S	LLID NESH SH MD PDU-Length 2 1 0 0	L2CAP Header L2CAP-Length ChanId 0x0016 0x0004	ATT_Find_Info_Req Opcode Format InfobData 0x05 0x01 48 00 02 29 49 00 03 29 4A 00 04 29 4E 00 01 29 4C 00 03 28	CRC RSSI FCS 0xCBD056 -52 OK
84	+431 =11973102	Ox0D	Ox95E9938E	L2CAP-S	LLID NESH SH MD PDU-Length 2 1 1 0	L2CAP Header L2CAP-Length ChanId 0x0003 0x0004	ATT_Read_Req Opcode AttHandle 0x0A 0x047	CRC RSSI FCS 0x830E4E -36 OK
85	+486 =11973388	Ox0D	Ox95E9938E	L2CAP-C	LLID NESH SH MD PDU-Length 1 0 1 0	OxCDBEED	-49	OK
86	+319746 =12293134	Ox1B	Ox95E9938E	L2CAP-C	LLID NESH SH MD PDU-Length 1 0 0 0	OxCDB378	-39	OK
87	+431 =12293365	Ox1B	Ox95E9938E	L2CAP-S	LLID NESH SH MD PDU-Length 2 1 0 0	L2CAP Header L2CAP-Length ChanId 0x0003 0x0004	ATT_Read_Req Opcode AttValue 0x0B AF 00	CRC RSSI FCS 0xEE80B1 -52 OK

Figura 4.11 Captura sniffer 4 (Fuente: elaboración propia)

## SCAN REQUEST ( )

P.nbr.	Time [us]	Channel	Access Address	Adv PDU Type	Adv PDU Header	AdvA	AdvData	CRC	RSSI [dBm]	FCS
92	+105623 =10186023	Ox25	Ox8E89EED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	0xC0FFEEC0FFEE	02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00	0x102DBB	-52	OK
93	+103747 =10289778	Ox25	Ox8E89EED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	0xC0FFEEC0FFEE	02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00	0x102DBB	-51	OK
94	+527 =10290297	Ox25	Ox8E89EED6	ADV_SCAN_REQ	Type TxAdd RxAdd PDU-Length 3 0 0 12	0x506B32B407F5	0xC0FFEEC0FFEE	0x812D54	-34	OK
95	+327 =10290624	Ox25	Ox8E89EED6	ADV_SCAN_RSP	Type TxAdd RxAdd PDU-Length 4 0 0 37	0xC0FFEEC0FFEE	ScanRespData 05 09 54 65 73 74 05 16 10 FF 00 00 00 00 00 00	0xAFFB3E	-52	OK
96	+99144 =10389768	Ox25	Ox8E89EED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	0xC0FFEEC0FFEE	AdvData 02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00	0x102DBB	-51	OK
97	+108123 =10497891	Ox25	Ox8E89EED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	0xC0FFEEC0FFEE	AdvData 02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00	0x102DBB	-51	OK
98	+100622 =10598513	Ox25	Ox8E89EED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	0xC0FFEEC0FFEE	AdvData 02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00	0x102DBB	-51	OK
99	+108123	Ox25	Ox8E89EED6	ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	0xC0FFEEC0FFEE	AdvData 02 01 05 04 16 08 18 64 03 14 06 18 00 00 00 00	0x102DBB	-51	OK

Figura 4.12 Captura sniffer 5 (Fuente: elaboración propia)

## ERROR ( )

P.nbr.	Time [s]	Channel	Access Address	Data Type	Data Header	L2CAP Header	ATT Handle Value Notify	CRC	RSSI (dBm)	FCS
63	+231 =8430476	Ox15	OxFO4A23D4	L2CAP-S	LLID MESN SN MD PDU-Length 2 0 1 0 9	L2CAP-Length ChanId 0x0005 0x0004	Opcode AttHandle AttValue 0x1E 0x0047 AD 00	0xA35FA3	-52	OK
64	+19740 =8750216	Ox23	OxFO4A23D4	L2CAP-C	LLID MESN SN MD PDU-Length 1 0 0 0 0			0x485000	-34	OK
65	+231 =8750447	Ox23	OxFO4A23D4	L2CAP-C	LLID MESN SN MD PDU-Length 1 0 0 0 0			0x4856D3	-53	OK
66	+19772 =9070219	Ox0C	OxFO4A23D4	Control	LLID MESN SN MD PDU-Length 3 1 1 0 2	LL_Opcode Terminate Ind(0x02)	LL_Terminate_Ind ErrorCode 0x13	0xB4F021	-33	OK
67	+246 =9070465	Ox0C	OxFO4A23D4	L2CAP-C	LLID MESN SN MD PDU-Length 1 0 1 0 0			0x485D4E	-52	OK
68	+426525 =9496990	Ox25	Ox8E89BE06	Adv PDU Type ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	AdvA 0xC0FFEEC0FFEE	AdvData 02 01 05 04 16 08 16 64 03 14 06 18 00 00 00	0x102DDE	-46	OK
69	+103748 =9600738	Ox25	Ox8E89BE06	Adv PDU Type ADV_IND	Type TxAdd RxAdd PDU-Length 0 0 0 37	AdvA 0xC0FFEEC0FFEE	AdvData 02 01 05 04 16 08 18 64 03 14 06 18 00 00 00	0x102DDE	-46	OK

Figura 4.13 Captura sniffer 6 (Fuente: elaboración propia)

### 4.3. Lanzamiento en Google Play

Como objetivo final y método de distribución de la aplicación para los futuros usuarios, vamos a analizar las diferentes maneras que disponemos para difundir nuestra aplicación así como los pasos a seguir para la distribución de la misma.

Una vez creado nuestro archivo .apk que contiene todo el código fuente de la aplicación, la manera más sencilla de la que disponemos para distribuir nuestra aplicación es el envío directo mediante correo electrónico o la descarga directa desde algún sitio web. A primera vista, este modo puede parecer el más rápido y directo para ofrecer nuestra aplicación a los futuros clientes pero perdemos la profundidad de mercado y experiencia que nos da el publicar la aplicación en el mayor escaparate como es Google Play, accesible de serie desde todos los terminales móviles con sistema operativo Android.

Durante el 2013 el número de aplicaciones disponible en Google Play sobrepasaba el millón habiéndose incrementado este número un 2,5 veces con respecto al año anterior. Además se cifra en 50,000 millones el número de descargas registradas. La publicación en este store proporciona la suficiente experiencia al desarrollador para ir adaptando y corrigiendo aspectos de la aplicación ya que se dispone de un canal que nos va a permitir conocer las opiniones de los usuarios, depurar los errores que nos puedan indicar así como obtener una puntuación global de nuestra aplicación.

Otra de las opciones que nos permite este canal de difusión es la comercialización de nuestra aplicación. Existen varias formas de poder obtener beneficios con la comercialización de nuestra aplicación: fijar un precio a la descarga, admisión de anuncios, descarga de contenidos extra o productos integrados, suscripciones, etc...

El precio final de la aplicación queda de la mano del desarrollador de la misma que deberá tener en cuenta la tributación existente en cada momento en el país en el que se vaya a comercializar. La única limitación existente en este punto la marca un listado de países aptos para comercializar/registrarse aplicaciones y el tipo de moneda utilizado en las transacciones.

Tenemos que tener en cuenta que Google cobra un canon para todas las aplicaciones y productos integrados que se comercializan en Google Play equivalente al 30% del precio final que hayamos establecido a nuestra aplicación.



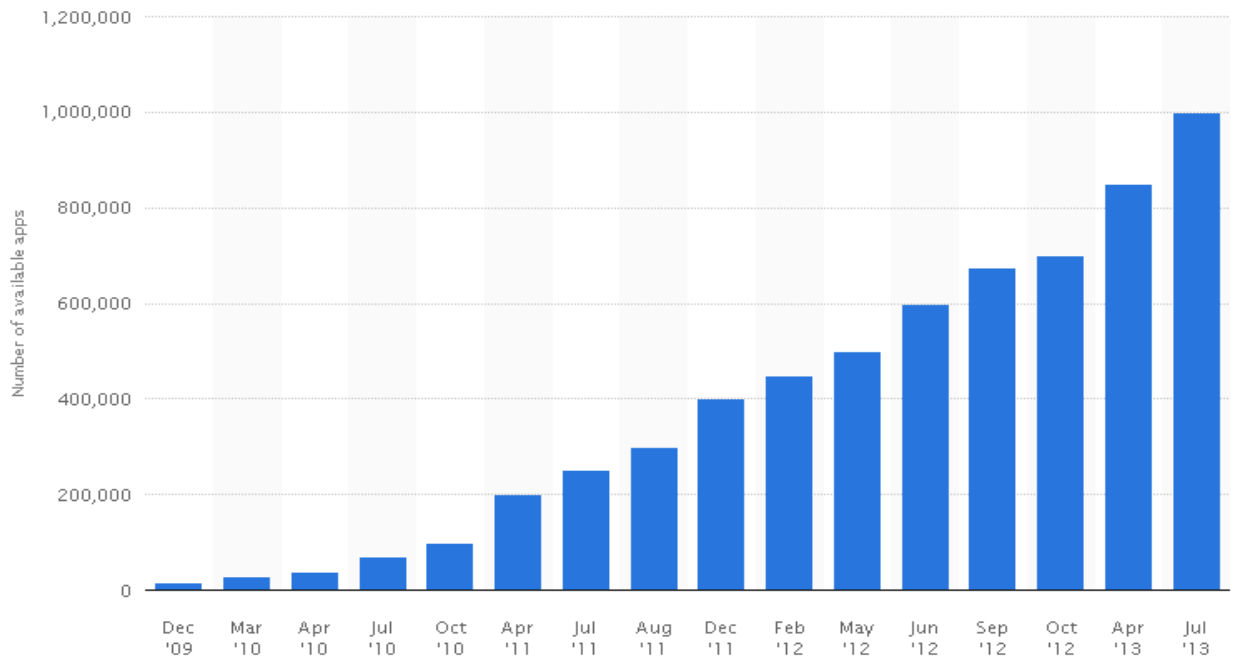


Figura 4.14 Evolución del número de aplicaciones Android (Fuente: android.developers)

## PUBLICACION

Desde un punto de vista más global, podemos ver el hecho de publicar la aplicación en Google Store como un hito final o como objetivo de todo desarrollo de aplicaciones móviles.

Antes de comenzar a subir los archivos de nuestra aplicación y empezar la distribución de la misma nos debemos asegurar que cumple unos mínimos requisitos de diseño y estilo. Para ello existen en la red diversas herramientas gratuitas, así como diversos tutoriales e instrucciones en la web [http://developer.android.com/tools/testing/testing\\_android.html](http://developer.android.com/tools/testing/testing_android.html), para poder testear y depurar nuestra aplicación aunque la mejor sin duda es testearlo directamente sobre un terminal móvil.

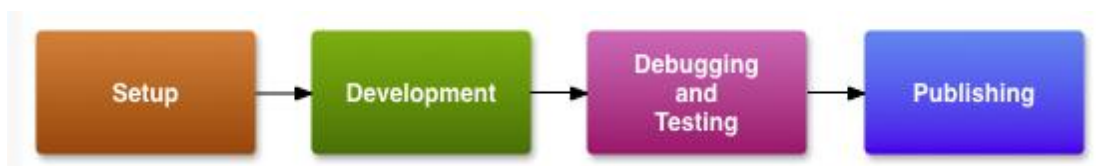


Figura 4.15 Proceso de publicación (Fuente: android.developers)

Para el testeo concreto de esta aplicación se ha hecho uso de tres terminales móviles distintos: HTC One, Motorola Razzr y HTC Tatum. Cada una de ellos con tamaños de pantalla distintos y características de software diferentes. Estas pruebas de funcionamiento y la aplicación del manual de estilo a la hora de elaborar el código fuente, comentado en un apartado anterior de esta memoria, nos ha permitido corregir a lo largo de todo el desarrollo algunos problemas ocasionados sobretodo con la calidad en los gráficos, iconos y las pantallas de navegación de la aplicación.



Para poder completar el proceso de publicación se deben seguir los siguientes pasos:

- Registrarse como desarrollador

El registro en Google Play tiene una cuota de alta de 25\$. En el proceso deberemos introducir varios datos personales, como el nombre del desarrollador que posteriormente se montará en Google Play, y aceptar algunas cláusulas sobretodo si pretendemos distribuir y comercializar nuestra aplicación posteriormente. Para esto debemos completar un paso más y registrarnos en Google Merchant y aceptar la política de exportación que legisla en los EEUU.

Como titulares de la cuenta podemos vincular y habilitar nuestra consola de Google a varios colaboradores del proyecto sin necesidad de que abonen la cuota.

- Subir los recursos de la aplicación

El archivo .apk generado por defecto deberá tener un tamaño máximo de 50 MB aunque si se requiere, se permiten aplicaciones con mayor tamaño utilizando archivos de expansión (parche).

La consola nos permite guardar la aplicación en modo "borrador" por si queremos hacer modificaciones o cambios en la versión.

Se necesitan dos capturas de pantalla obligatoriamente (formato JPEG o PNG , máx: 3840 píxeles y mín: 320 píxeles) así como una imagen para conformar el icono PNG de 32 bits, 512 x 512 píxeles, tamaño máx: 1024 KB) de nuestra aplicación.

Adicionalmente se nos permite subir toda una batería de recursos como imagen promocional, video promocional, así como ocho capturas de pantalla más que sin duda nos servirá para promocionar y llamar la atención a los nuevos usuarios.

Existen parámetros configurables como el nombre de la aplicación, el texto promocional o el texto de la descripción que acompañara a la visualización de la aplicación en Google Play (todos estos parámetros son configurables en el *Store Listing page*).

También tendremos que definir el idioma de nuestra aplicación que por defecto es el inglés. Existe la opción de traducción automática de textos conforme a una lista de idiomas de dispositivos con tecnología Android en diferentes lenguas.

Por último debemos informar de un canal de asistencia para nuestra aplicación como: correo electrónico, teléfono, sitio web.

## **ACTUALIZACIONES**

Una parte esencial para el buen futuro de la aplicación BlauTic serán sus actualizaciones. Nuevas versiones con la que se ajustarán errores, se introducirán nuevas funcionalidades y mejoras o se actualizaran contenidos.

Hay diferenciar estas actualizaciones con las cargas remotas de datos desde el servidor , por ejemplo para dar de alta dispositivos, que ya se ha comentado en capítulos anteriores de esta memoria.

Para realizar una actualización de la aplicación debemos asegurarnos que el nombre del archivo APK es el mismo que el de la versión original. Así mismo nos debemos asegurar que el código de la nueva versión está bien indicado en el *manifest* de la aplicación y que este valor es superior al de la versión actual.

```
<manifest
  android:versionCode="1"
  android:versionName="1.0" >
```

Una vez subida la actualización, transcurrido un mínimo de tiempo de 24 horas los usuarios de la aplicación recibirán la notificación indicándoles de la existencia de una nueva versión. Dependiendo de la configuración del terminal móvil de cada uno la actualización se realizará de manera automática o será el usuario quien la tenga que realizar manualmente.

Además de todo esto anteriormente indicado, la firma del paquete APK actualizado debe coincidir con la firma de la versión actual. El control de firma es un sistema de seguridad que nos exige Google Play para poder publicar y que nos permite garantizar que sólo nosotros podamos modificar y actualizar nuestra aplicación.

## FIRMAR APLICACIÓN ANDROID

Como hemos indicado, como requisito antes de publicar en Google Play, todas las aplicaciones deben estar firmadas y en el caso de las actualizaciones es necesario que la nueva versión se firme con el mismo certificado con el que se creó. Mientras nos encontramos en proceso de desarrollo el SDK utilizará una clave de depuración.

Para obtener nuestra firma no es necesario recurrir a una entidad certificadora así que el proceso lo podemos realizar utilizando algún programa como *Keytool* o *Jarsigner*.

Para utilizar *Keytool* debemos introducir en la consola la siguiente instrucción:

```
$>keytool -genkey -v -keystore nombredelkeystore.keystore -alias nombredelalias -keyalg RSA -
keysize 2048 -validity 10000
```

y deberemos contestar a las siguientes preguntas:

- ¿Cuál son su nombre y apellidos?
- ¿Cuál es el nombre de su unidad de organización?
- ¿Cuál es el nombre de su organización?
- ¿Cuál es el nombre de su ciudad o localidad?
- ¿Cuál es el nombre de estado o provincia?
- ¿Cuál es el código de su país de dos letras de la unidad?

Al ejecutar esto nos pedirá una contraseña tanto para nuestra key como para el alias y se generará finalmente nuestro keystore. Es importante conservar este keystore porque es el que posteriormente utilizaremos para firmar las actualizaciones.

Una vez obtenidas las claves podemos firmar fácilmente nuestra aplicación (o actualización) con Eclipse desde la opción

*Android tools > Export Signed Application Package.*

## SEGUIMIENTO

Una vez registrados con nuestra cuenta de Google Play podemos acceder a la consola de desarrollo (Developer Console). Esta herramienta nos permite acceder y modificar los datos de registro de nuestra cuenta como son el nombre del desarrollador, información de contacto, sitio web... así como gestionar todo lo referente a nuestras aplicaciones.

Con el acceso al *Store Listing page*, que hemos mencionado con anterioridad, podemos modificar la información que se muestra al resto de los usuarios así como ir actualizando los contenidos gráficos. También podremos redefinir o ajustar el precio de nuestra aplicación o acceder al reporte de ingresos generados por la misma.

Esto último resulta ser una herramienta muy útil para monitorizar y controlar el éxito y el impacto que tendrá nuestra aplicación. La aplicación nos permite realizar estadísticas sobre el número de usuarios que tienen la aplicación o sobre el número de dispositivos donde esta está alojada. Nos permite profundizar más y agrupar esta información con valores tipo: ratios, el total de instalaciones, descargas diarias, versiones actualizadas, desinstalaciones de la aplicación.... también podemos obtener información acerca de los usuarios que tienen nuestra aplicación y agruparlos según la versión del sistema operativo, el idioma o el país de procedencia.

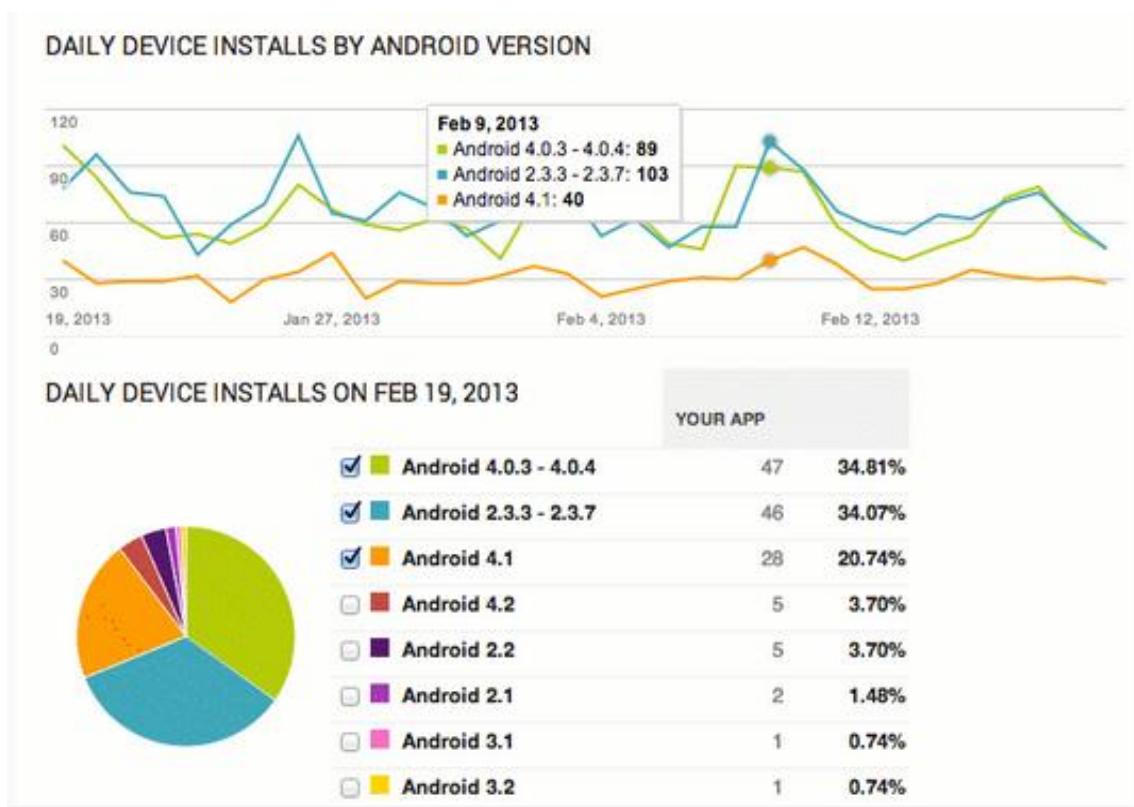


Figura 4.16 Estadísticas aplicaciones (Fuente: android.developers)

Además se nos permite gestionar los privilegios de cada uno de los usuarios de la cuenta para poder controlar (por ejemplo) quien puede tener acceso a la gestión de configuración, o las publicaciones o a los reportes financieros.

## 5. Costes temporales / económicos

Inicialmente se ha organizado el desarrollo completo de la plataforma BlauTic con una planificación a modo de fase/hitos con una estimación temporal de 22 semanas.

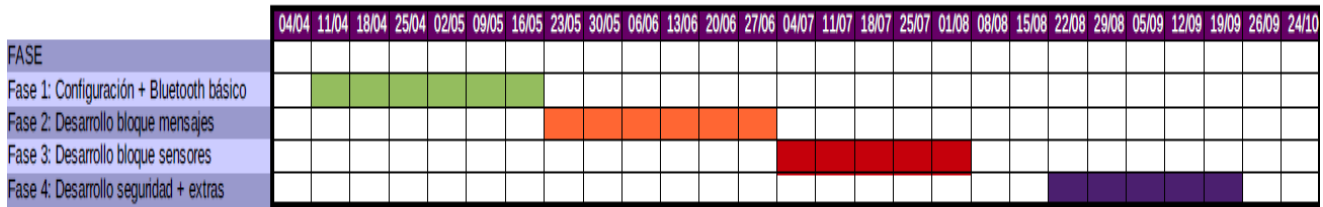


Figura 5.1 Planificación 1 (Fuente: elaboración propia)

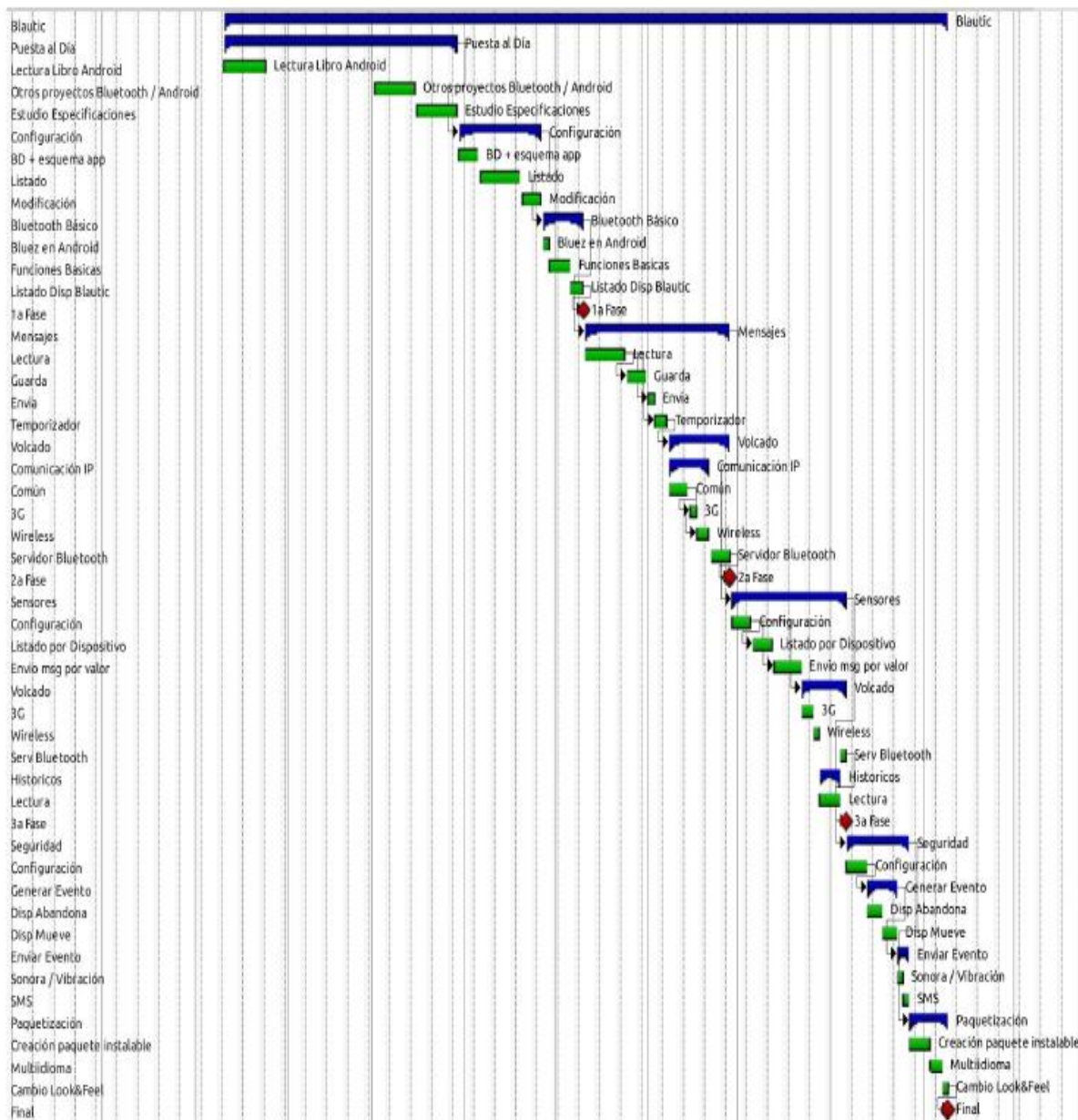


Figura 5.2 Diagrama de hito tecnológicos (Fuente: elaboración propia)

El plan de desarrollo consta de 5 fases a completa aproximadamente en cinco semanas cada una de ellas:

1. Fase 0: Fase inicial de estudio de las especificaciones
2. Fase 1: Inicio desarrollo código de la aplicación
3. Fase 2: Desarrollo del apartado de mensajes
4. Fase 3: Desarrollo del apartado de sensores
5. Fase 4: Desarrollo del apartado de seguridad y últimos ajustes de la aplicación.

En cuanto a recursos empleados, por parte de la empresa se han dispuesto de suficientes dispositivos bluetooth BlauTic para poder realizar los test de la aplicación así como un par de móviles smartphone, modelo HTC tattoo y Motorola MotoRazer y un equipo portátil de última generación. Además se han costado los gastos ocasionados por la adquisición de documentación bibliográfica así como accesos a portales web y revistas digitales con suscripciones.

También se ha dispuesto por parte de la empresa de un estudio de trabajo, cerca de la zona universitaria, con acceso a todos los recursos anteriormente mencionados.

Cabe menciona que este proyecto se ha ejecutado bajo un acuerdo de colaboración entre la empresa BaluTic y la *Universitat Politècnica de València* a modo de prácticas de empresa con una asignación económica suscrita según el convenio firmado entre las dos partes.

## 5.1 Modelo de negocio

Para poder definir un plan de negocio hay que tener en cuenta que la comercialización de la aplicación va a ir ligada siempre a la adquisición de dispositivos hardware BlauTic A1/B1. Además se debe contar con que las aplicaciones que se vayan a distribuir pueden variar tanto en su diseño como en la funcionalidad ya que la idea inicial es que esta se pueda adaptar y personalizar a las distintas exigencias que el cliente vaya requerir por lo que, dependiendo de esta complejidad, el precio final puede variar.

También se tendría que realizar un análisis y valorar el coste de mantenimiento del producto (en el caso de que se requiera) ya que el cliente puede necesitar de actualizaciones y/o ajustes de software de la aplicación que ha contratado.

Estaríamos barajando por tanto estos escenarios:

- Una única aplicación junto a diversos dispositivos hardware:

Esta opción no requeriría de un sistema de distribución masivo como pueda ser Google Play. Se necesitaría una solución software compleja que pudiera administrar los datos y la información obtenida de varios dispositivos BlauTic simultáneamente.

Esta opción comercial es la que se ha implementado en la aplicación BlauTic adquirida por el Departamento de Biomecánica de la *Universitat de València*. En ella se incorpora en un dispositivo móvil tipo tableta la aplicación que permite la monitorización de cuatro sensores BlauTic para analizar y estudiar la biomecánica de la carrera en corredores.

- Un solo dispositivo hardware y varias distribuciones de la aplicación para distintos móviles

Como ejemplo a este sistema, BlauTic ha proporcionado a Lavhek, empresa valenciana del sector del ascensor, una aplicación móvil para el control automatizado de las líneas telefónicas instaladas en las cabinas de los ascensores. Se ha dotado a los distintos técnicos de un sistema que permite la monitorización del estado de las líneas desde sus smartphones.

- Diversos dispositivos hardware y diversas distribuciones de la aplicación.

A priori esta parece ser la opción más costosa y compleja ya que aquí se plantea un escenario en el que habría que definir distintos roles de usuarios y sistemas para poder controlar todos los dispositivos.

Las soluciones comerciales BlauTic mencionadas y que se han puesto a modo de ejemplo, todas ellas están basadas en este proyecto y diseñadas en base a la aplicación “Plataforma BlauTic”. En el apartado 6.3 de esta memoria se analizarán con mayor detalle y se enumerarán algunas otras.



## 6. Conclusiones

Desde el punto de vista técnico podemos concluir que se han alcanzado y completado el 80% de los objetivos del proyecto. Del paquete de especificaciones que se nos solicitaron al inicio no hemos podido acometer la parte referente a la recepción de notificaciones en modo broadcast por parte de la aplicación móvil. Este punto ya ha sido descrito las razones en el apartado 3.6.3 (Problemas) de esta misma memoria.

Referente a la parte por completar, se podría justificar debido al carácter innovador del proyecto. Hemos querido fusionar de algún modo el exitoso y proliferante mundo de las aplicaciones Android para dispositivos móviles con el desarrollo de un software para una tecnología emergente, como es el bluetooth4.0. , donde a fecha de realización del proyecto aun no existían los estándares oficiales en lo referente al uso de funciones desde este sistema operativo.

### 6.1. Objetivos como proyecto

Desde el punto de vista más personal, trabajar con un tema relacionado con las nuevas tecnologías me ha hecho ver lo rápido que se mueve todo a nivel mundial en cuanto avances tecnológicos y tendencias del mercado que impiden a uno hacerse estimaciones a medio-largo plazo ya que todo está evolucionando continuamente.

Desarrollando aplicaciones móviles he tenido la necesidad de entrar en numerosos foros, leer multitud de publicaciones, revistas especializadas y uno se da cuenta de la cantidad de gente que está trabajando o se dedica a este mundo ya sea de manera profesional pero sobre todo de modo amateur. La mayoría de avances y progresos son compartidos por la gente sin ningún tipo de interés económico y en estos foros uno recibe todo tipo de recomendaciones y consejos que en mi caso me han sido de gran ayuda. El sentirme parte de esta “comunidad” de programadores anónima ha sido una de las mayores experiencias desarrolladas en la ejecución de este proyecto.

Por esta parte, concluir que uno de los principales desafíos que deseaba acometer cuando decidí implicarme en este proyecto era el aprender una habilidad nueva como ha sido en este caso el desarrollo de aplicaciones para dispositivos móviles. Estoy satisfecho de haber podido alcanzar este hito personal que estoy seguro será muy beneficioso desde el punto de vista profesional en un futuro.

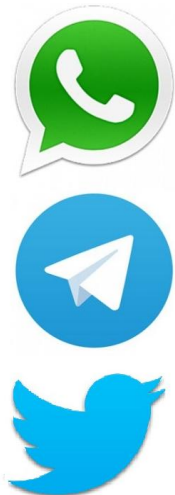
También quiero valorar la gran experiencia que ha resultado ser para mi trabajar con el equipo de Blautic, con Javier y Borja, que me han ayudado siempre en todo lo que les he requerido. La metodología de trabajo, a base de planificación de tareas y de hitos tecnológicos, ha sido todo una novedad para mi y me ha hecho aprender una manera muy eficiente y organizada de trabajar en equipo.

## 6.2. Evolución futura

Los posibles avances y mejoras que se puedan introducir en la aplicación dependerán en un alto grado de la publicación del API de Google para dispositivos BLE. El paquete de funciones que aportará esta nueva API permitirá una mayor eficiencia y estabilidad en lo que respecta a las comunicaciones bluetooth con los dispositivos.

Permitiría completar la funcionalidad de lecturas de notificaciones en modo broadcast que no se ha podido implementar en esta fase del proyecto. Recordar que este modo permite dejar el terminal móvil en modo “escucha” y que este continúe recibiendo datos de los dispositivos de la red BlauTic asociada (por ejemplo de los sensores) sin tener que haber establecido una conexión directa con cada dispositivo previamente. Implementada esta nueva API se debería migrar la parte del código que se ha realizado con las clases java desarrolladas por Motorola al código nativo de Google. Este hecho reduciría notablemente el consumo de energía del terminal móvil aumentando su eficiencia al alargar la duración de la batería.

Otras de las mejoras que se pueden introducir a la aplicación es la conexión directa con redes sociales tipo twitter, whatsapp o telegram. De alguna manera se podrían sincronizar los datos recogidos de los dispositivos BlauTic, cada uno de ellos tendría una cuenta en la red social, para su publicación directa en la cuenta del usuario remoto que requiriera monitorizar los resultados.



## 6.3. Productos actuales basados en el proyecto

Debido a las características y versatilidad que permiten los dispositivos LE de BlauTic existen infinidad de situaciones reales y ambientes de trabajo donde se podría implementar una aplicación móvil.

Hay que resaltar que , en base a este proyecto y a la aplicación “Plataforma BlauTic”, BlauTic ha implementado y ejecutado numerosos proyectos de los que vamos a enumerar algunos de los más destacados.

Algunas de las aplicaciones han sido presentadas y premiadas en concursos TIC y I+D obteniendo excelentes resultados por su carácter innovador, como son:

### PREMIOS Y RECONOCIMIENTOS:

El proyecto “Contenedores inteligentes: dales tu voz” aplica la tecnología inalámbrica BlauTic a la gestión inteligente de recogida de residuos. El sistema establece un canal de comunicación entre la red de contenedores urbanos , los cuales incorporan un dispositivo BlauTic sensor, y un centro gestor de datos donde se recoge todo tipo de información útil para la recogida de los residuos. Como nexo de unión entre estos dos medios entra en juego el ciudadano de a pie que mediante su Smartphone canaliza la información que recoge por bluetooth de los contenedores y la envía a través de la red móvil de datos (3G o Wifi) al centro gestor.





*Figura 6.1 Esquema aplicación “contenedores inteligentes” (Fuente: Blautic)*

Con esta recogida de información se pretende dar solución a diversos aspectos como:

- Zonas con contenedores excesivamente llenos frente a zonas con contenedores siempre vacíos. Suciedad alrededor de contenedores llenos. Falta de datos para el dimensionado correcto del número de contenedores y tipo por zona
- Costes elevados de recogida de residuos en urbanizaciones y zonas rurales.
- Falta de datos reales de la ubicación de los contenedores y de datos fiables para optimización en las rutas de recogida. Costes innecesarios.

La aplicación ha sido galardonada con el primer premio del “concurso TISStartUP by TAI Editorial” como mejor idea innovadora dentro del sector TI.

Otro de los sectores potenciados por BlauTic y que está muy alineado con las tendencias actuales es el sector de la salud y sanidad. El proyecto BlauTic eHealth, galardonado con el premio “LiveStartup Europe” en el año 2012, es muestra de ello.

Este proyecto está compuesto de una amplia gama de productos inalámbricos, que incorporan dispositivos BlauTic, y que forman en conjunto un sistema de recogida de datos vitales del paciente. Este usuario, mediante el uso de una aplicación incorporada en su Smartphone, será capaz de tener todos los datos esenciales para el control de su patología y sentirse más seguro.

Con esta aplicación se pretende unificar en un mismo dispositivo aspectos como el control de movimientos, caídas, ejecución de avisos y recordatorios, etc... además tener monitorizado en todo momento información acerca de termómetro, temperatura ambiente, medidores cardiológicos, niveles de glucosa, etc... en forma de registro interno o en modo de que estos datos puedan ser analizados remotamente por algún equipo médico especializado.



Figura 6.2 Esquema Blautic Health (Fuente: Blautic)

Podemos encontrar con más detalle en la web corporativa [www.blautic.com](http://www.blautic.com).

#### PROYECTOS ACTUALES:

Otro de los trabajos destacados es el adquirido por el Departamento de Educación Física y Deportiva de la Universidad de Valencia donde BlauTic ha desarrollado un equipo nuevo e medición de sensores que permiten realizar medidas de aceleración lineal y angular en los 3 ejes en múltiples puntos del cuerpo humano durante el desarrollo de la actividad física. Su control se realiza desde tablet o Smartphone y con estos datos se pretenden medir el comportamiento del corredor en fatiga con la biomecánica de la carrera.

También se han conseguido comercializar algunas aplicaciones con el instituto tecnológico AITEX (Asociación de Investigación de la Industria Textil) que centra sus productos en el ámbito de la industria textil.

“Habitat” es una aplicación BlauTic para controlar el nivel de monóxido de carbono en una estancia. El sistema está basado en una red de sensores que se encuentran integrados en las cortinas de la habitación y que lanza una alarma, mediante un sistema de iluminación insertado en la misma cortina y mediante avisos al dispositivo móvil, superado cierto umbral.

Otro de los proyectos importantes implementados para este importante instituto tecnológico y que se engloba dentro del desarrollo de nuevos productos relacionados con el sector de los *SmartClothes* (ropa inteligente) son las “Medias inteligentes”.



*Figura 6.3 “Medias inteligentes” (Fuente: Blautic)*

Blautic ha desarrollado para AITEX un sistema inalámbrico de control milimétrico de la elongación muscular en 4 puntos diferentes de medición. El equipo electrónico y el software proporcionados ha permitido a AITEX construir una media inteligente que está siendo probada intensamente en laboratorios de biomecánica para mejorar en el diseño de las medias compresivas.

Estoy seguro que lo referenciado en este apartado es sólo una pequeña muestra de la multitud de aplicaciones y mejoras que con estos dispositivos y la expansión de la tecnología bluetooth 4.0 se pueden acometer en un futuro próximo.

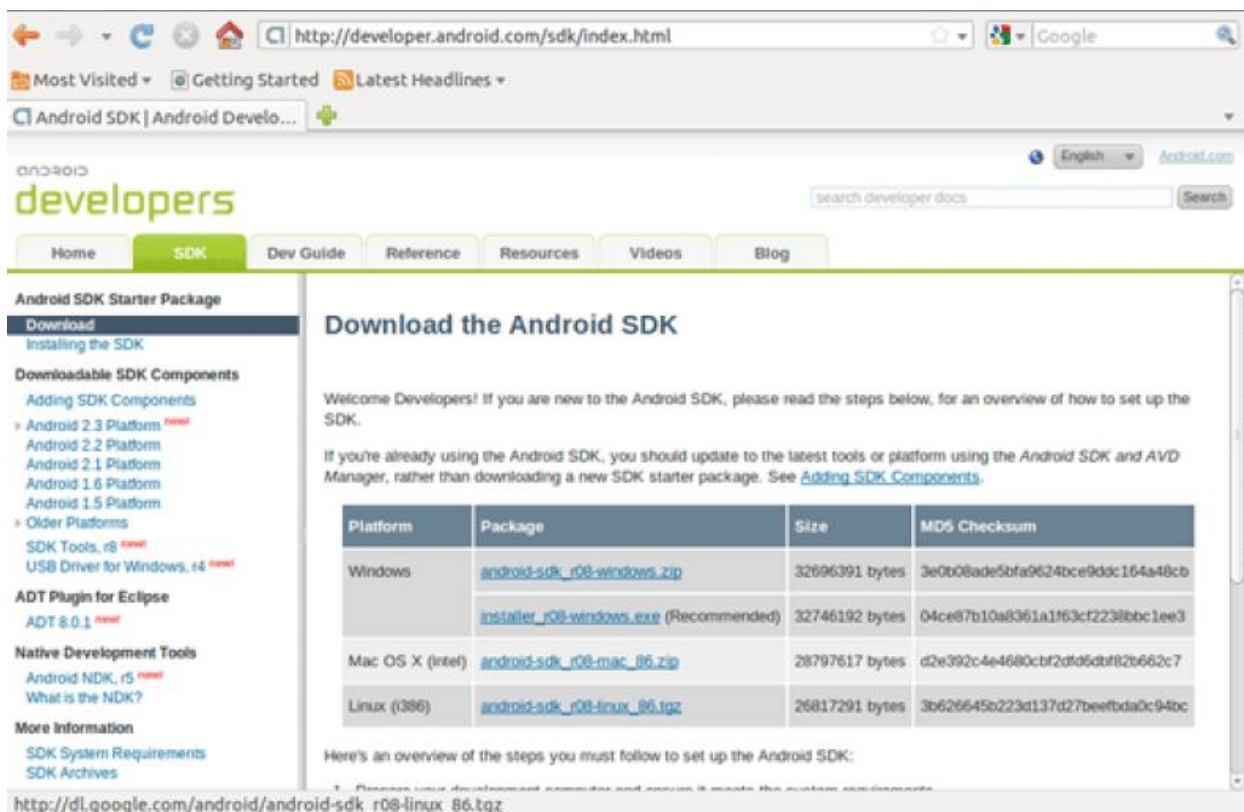
## 7. Bibliografías y referencias

- “*Android, guía para desarrolladores*”. Frank Ableson, Charlie Collins, Robi Sen. Editorial Manning
- “*El servidor de bases de datos relacionales SQL Server*”. Fernando Boronat Seguí, Jesús Tomás Gironés. SPUPV-2000.2278
- “*Comunicaciones en UNIX*”. Jean-Marie Rifflet (McGraw Hill. 1998.)
- “*Wireless Applicatoin Developement*”. Lauren Darrey, Shane Conder. (Developer's library)
- Estudio “Hábitos de compra, posesión, gasto y uso de telefonía móvil en España”. Consultora Kantar Worldpanel ComTech. (2013)
- "Market Share Analysis: Mobile Phones, Worldwide, 2Q13.". Gartner (Agosto 2013)
- “Mobile Platforms: Clash of Ecosystems”. VisionMobile (Noviembre 2011)
- Informe “Spain Digital Future in Focus”. ComScore. (Abril 2013)
- <http://developer.android.com/> (site oficial Android)
- <http://www.xataka.com/> (porta web especializado en gadgets y nuevas tecnologías)

## 8. Anexos

### 8.1. Instalación SDK Android (Software Development Kit) + integración en Eclipse

Abrir un navegador web y acceder a la dirección <http://developer.android.com/sdk/index.html> donde encontraremos diferentes links para la descarga del Android SDK para diferentes plataformas: Windows, Mac OS X y Linux.



Platform	Package	Size	MD5 Checksum
Windows	<a href="#">android-sdk_r08-windows.zip</a>	32696391 bytes	3e0b08ade5bfa9624bce9ddc164a48cb
	<a href="#">installer_r08-windows.exe</a> (Recommended)	32746192 bytes	04ce87b10a8361a1f63cf2238bbc1ee3
Mac OS X (Intel)	<a href="#">android-sdk_r08-mac_86.zip</a>	28797617 bytes	d2e392c4e4680cbf2df05dbf82b662c7
Linux (i386)	<a href="#">android-sdk_r08-linux_86.tgz</a>	26817291 bytes	3b626645b223d137d27beefbda0c94bc

Vamos a explicar la instalación sobre el sistema Linux por lo que seleccionamos y descargamos la opción correspondiente.

```
wget http://dl.google.com/android/android-sdk_r08-linux_86.tgz tar xvfz android-sdk_r08-linux_86.tgz
```

Descomprimos el paquete comprimido.

Deberemos crear un directorio `android-sdk-linux_86` en nuestro directorio `$HOME` así como añadir en la variable de entorno `PATH` los directorios `$HOME/android-sdk-linux_86/tools` y `$HOME/android-sdk-linux_86/platform-tools`.

```
[...]
PATH="$HOME/android-sdk-linux_86/tools:$HOME/android-sdk-
linux_86/platform-tools:$PATH"
```

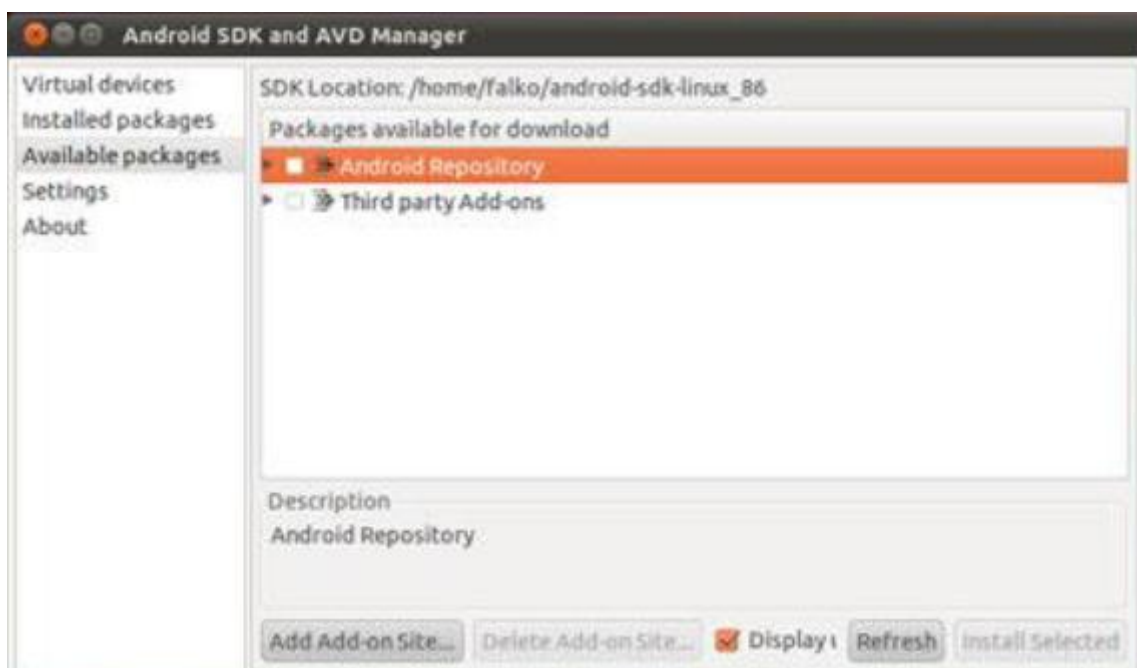
Para hacer efectivos los cambios podemos reiniciar la sesión del ordenador o bien introducir:

```
export PATH="$HOME/android-sdk-linux_86/tools:$HOME/android-sdk-linux_86/platform-tools:$PATH"
```

Iniciamos el Android SDK y aparecerá una pantalla tal que así:



Tendremos que seleccionar del menú de la izquierda la opción *Available packages* para después seleccionar el *Android Repository* y ver en el desplegable todos los paquetes de descarga disponibles.





Seleccionaremos todos los paquetes y iniciaremos la descarga.

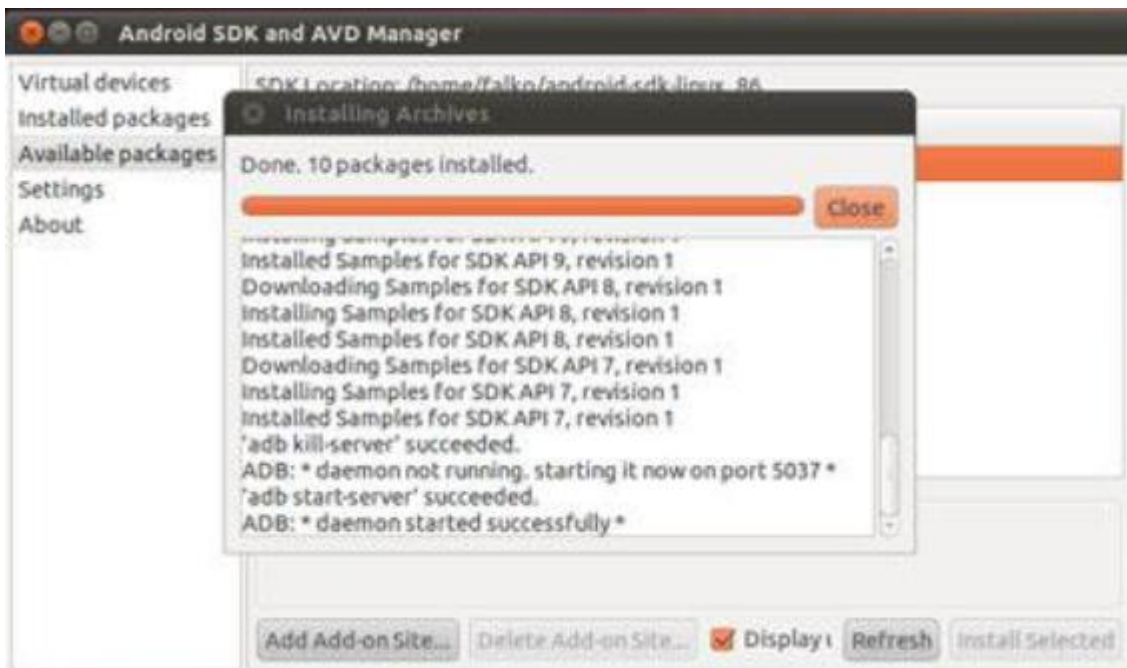


Se descargarán los ficheros



Para completar la descarga el ADB (Android Debug Bridge) puede que nos solicite reiniciarse. Aceptamos.





Para finalizada la descarga, seleccionamos la opción "Close" y ya quedará instalado el software en nuestro equipo.

A continuación mostraremos como integrar este software a la herramienta de programación Eclipse que es la que utilizaremos para desarrollar nuestra aplicación Android.

Iniciar Eclipse y seleccionar: *Help > Install New Software*

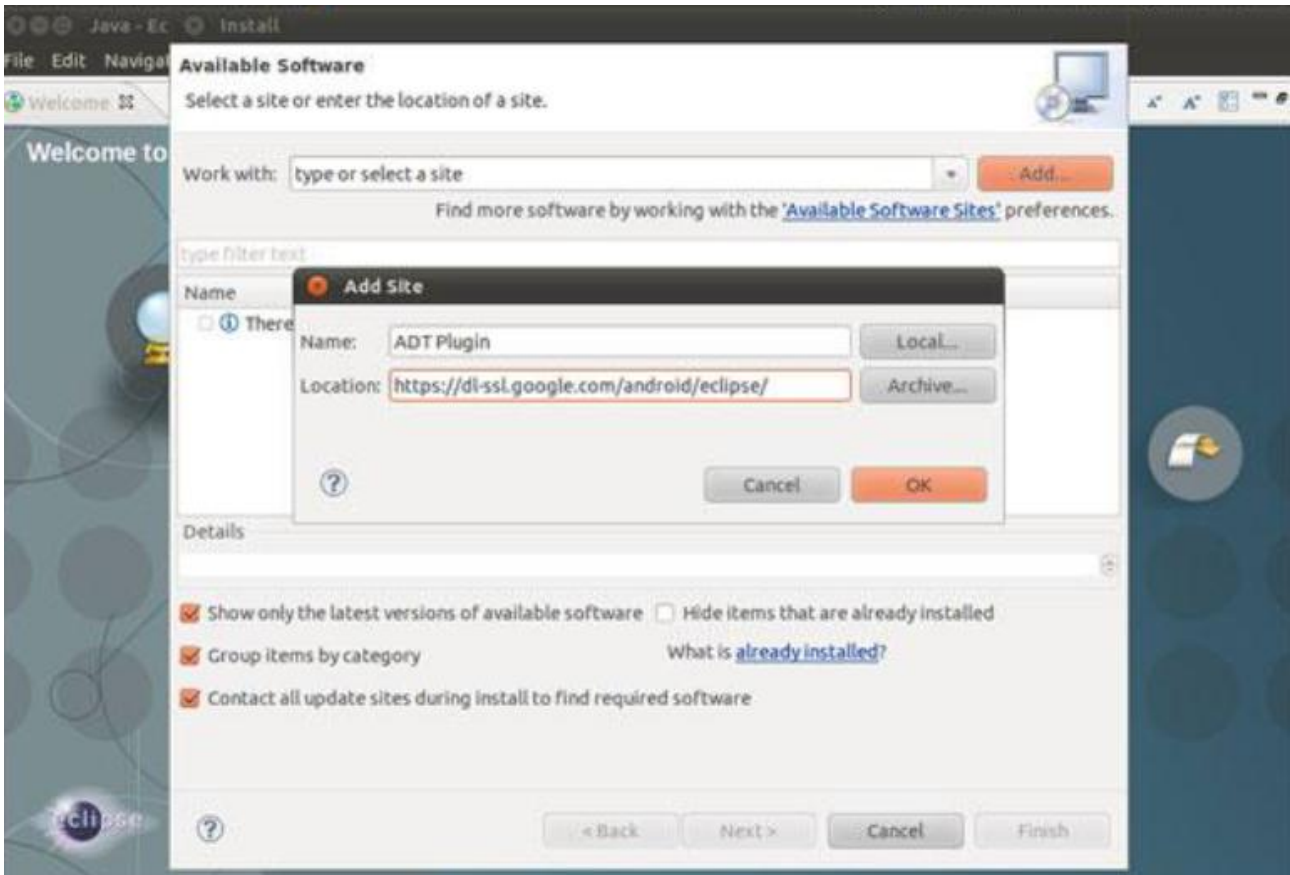




Con la opción "Add.." se nos abrirá un cuadro de dialogo donde deberemos introducir la información del plugin:

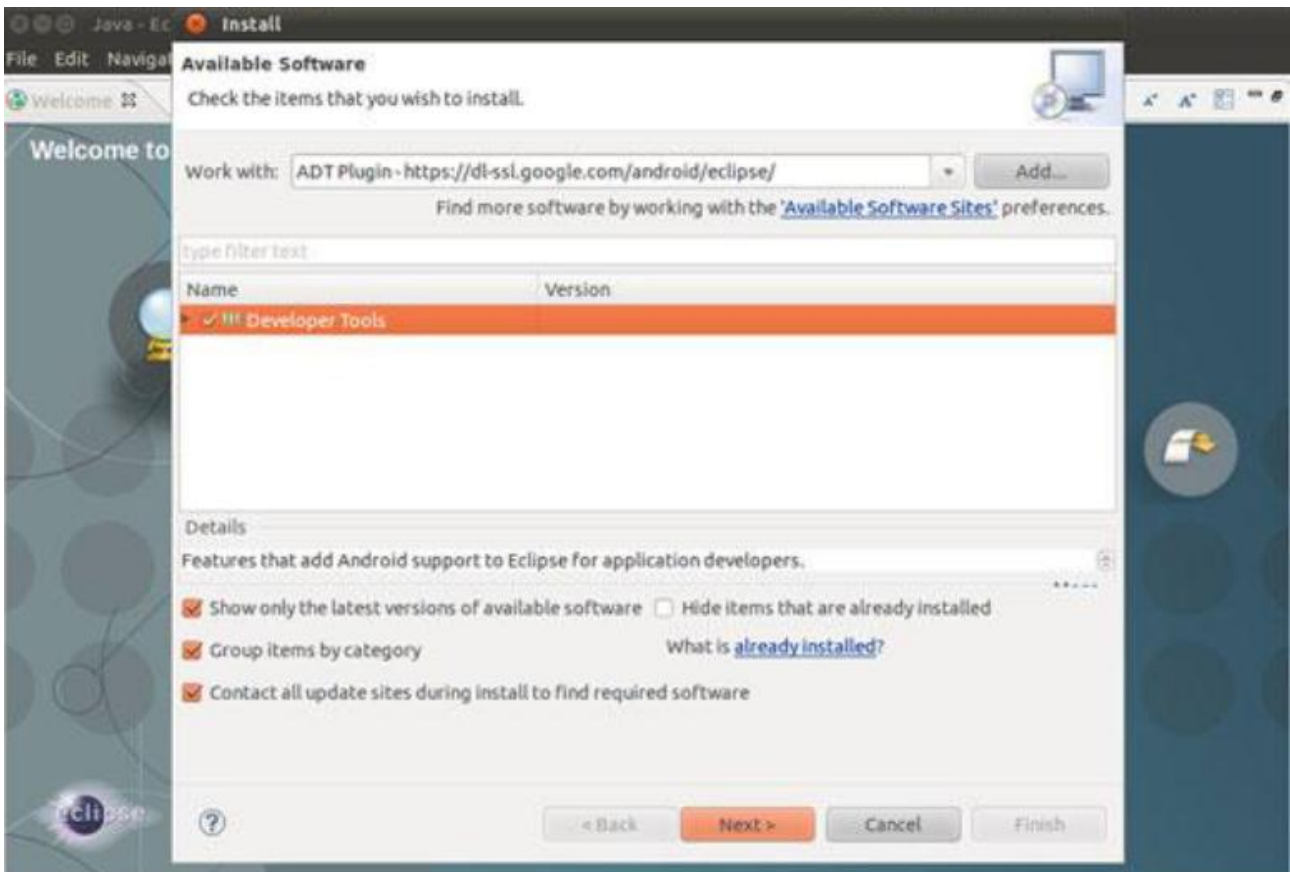
*Name: ADT Plugin*

*Location: <https://dl-ssl.google.com/android/eclipse/>*

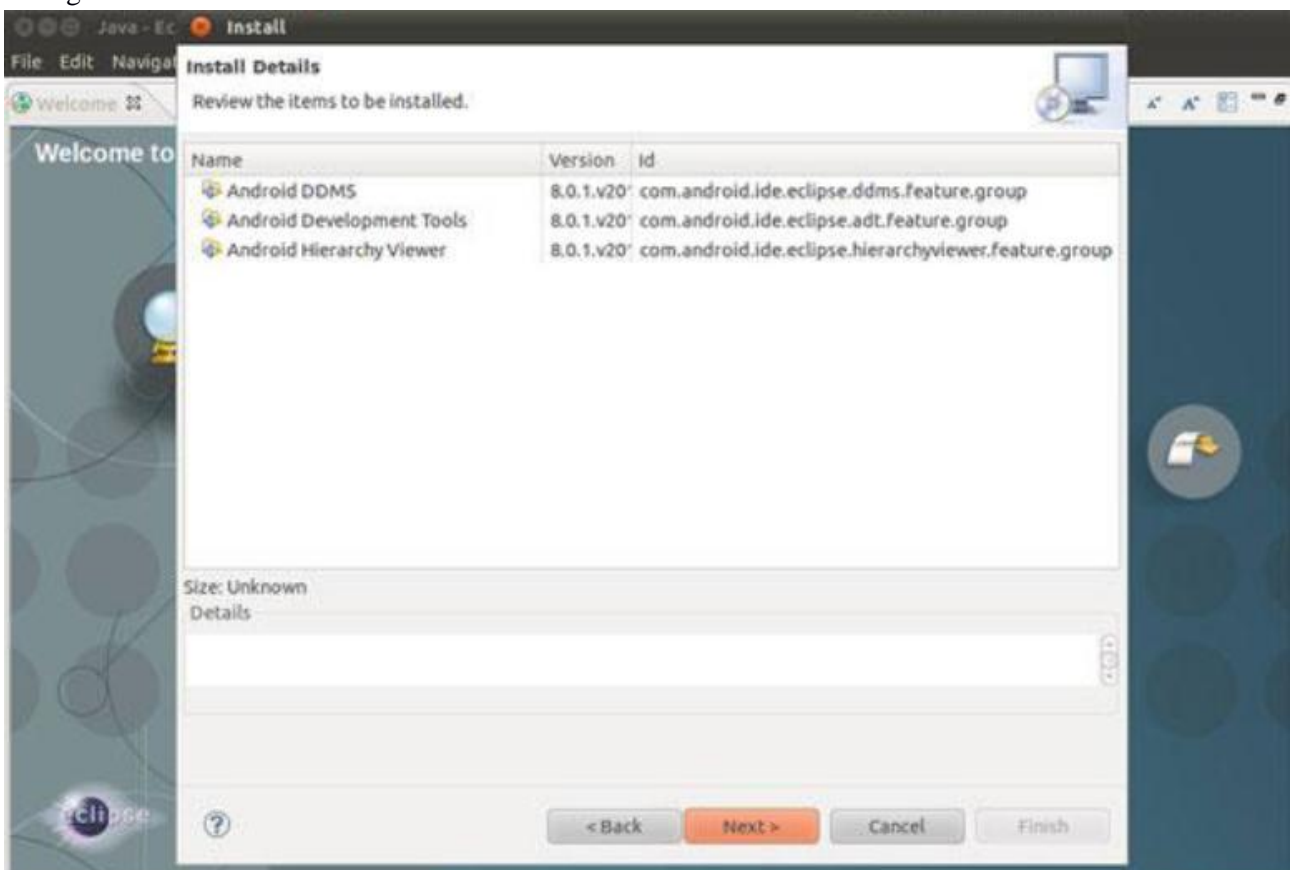


Seleccionando la opción "Avilabe Software Sities" se iniciará la búsqueda de los paquetes de descarga disponibles en este repositorio.

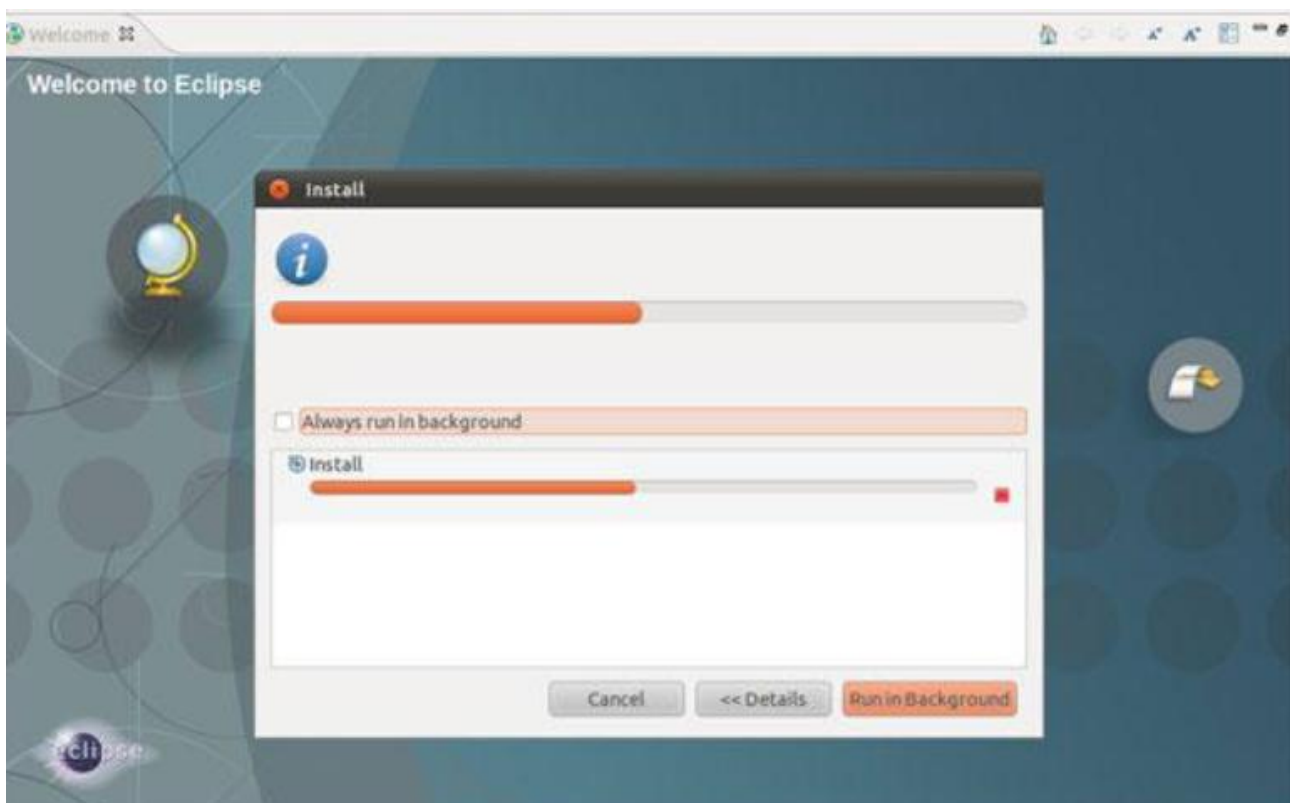
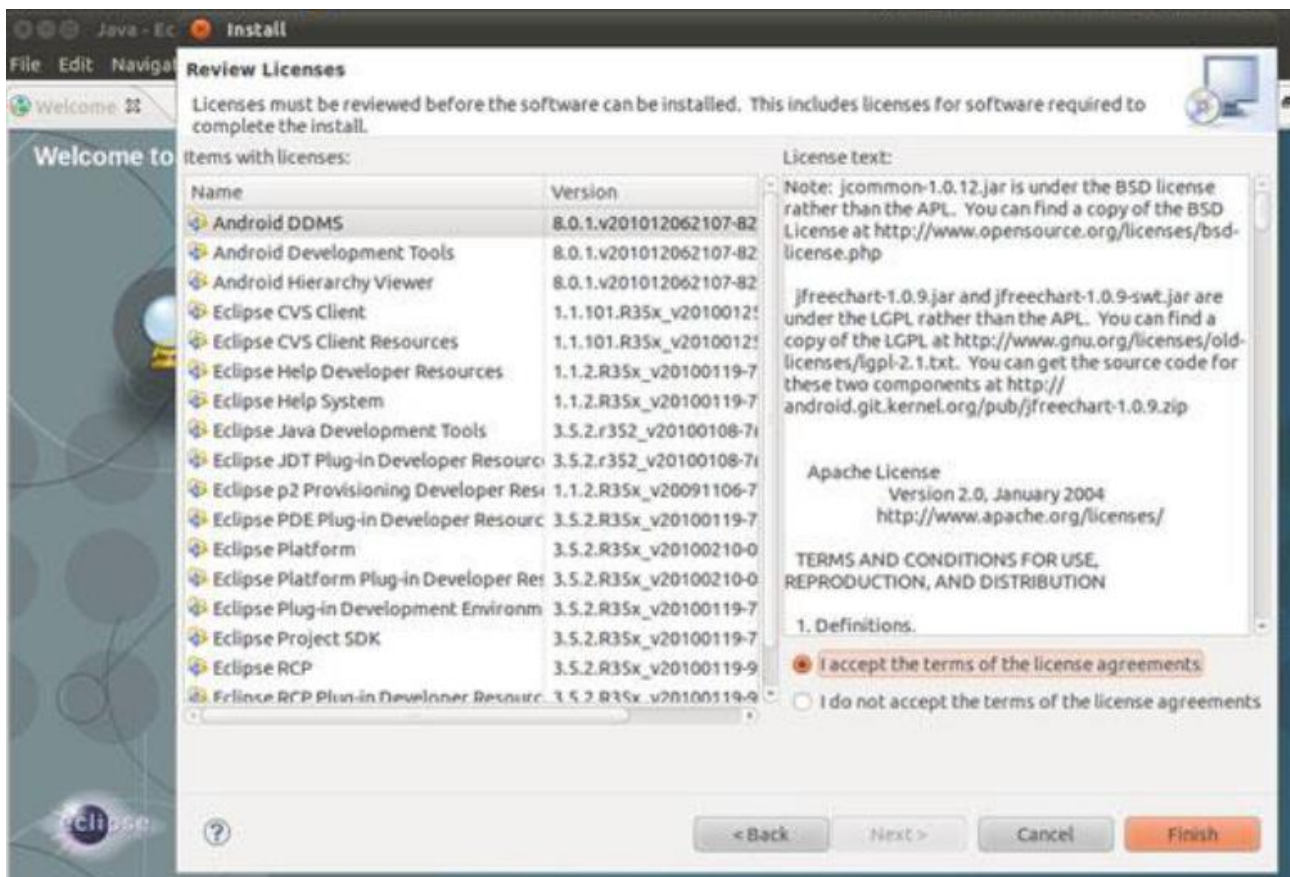
Seleccionaremos el software "Developer Tools" y haremos click en siguiente.



Siguiente



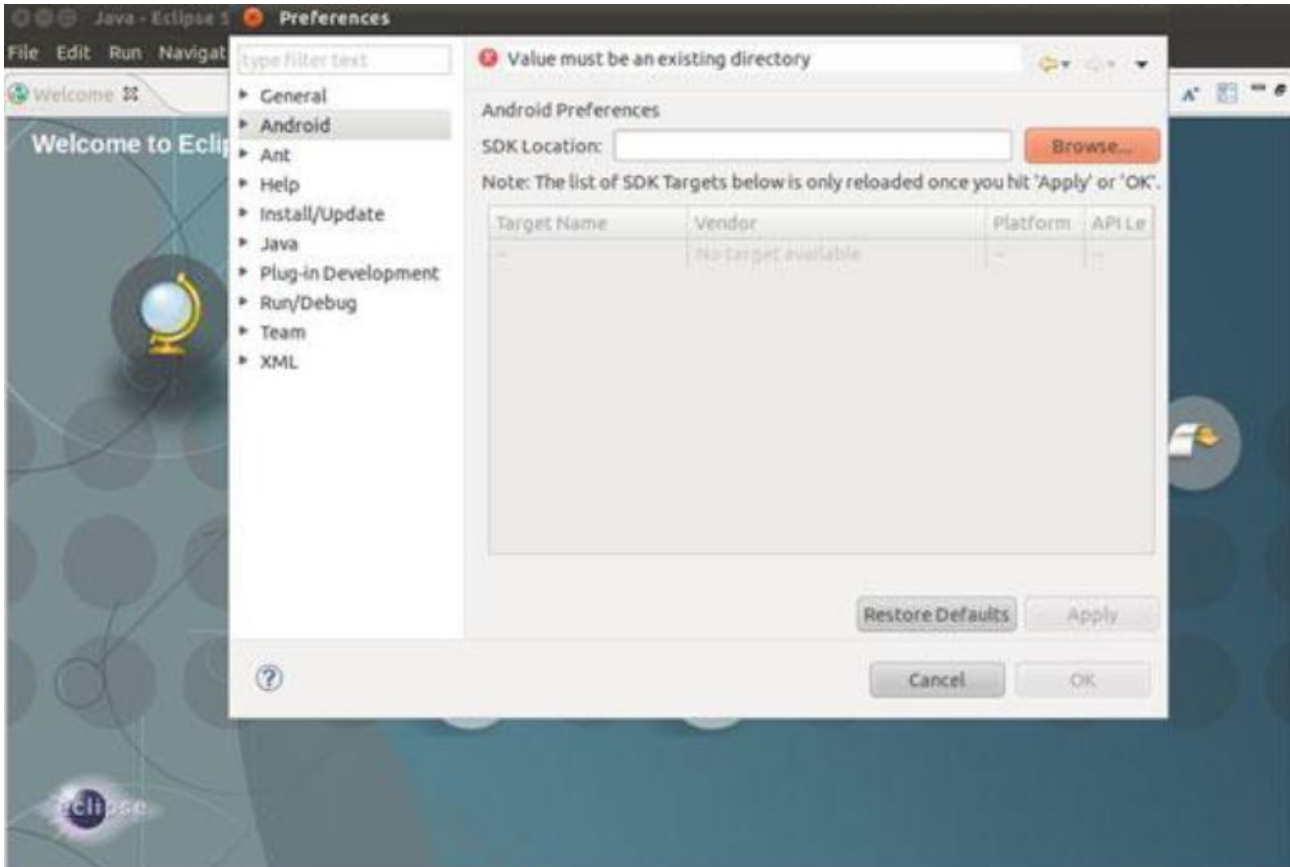
Aceptamos la licencia y automáticamente se iniciará la integración del plugin.



Para completar la instalación, Eclipse necesitará reiniciarse. Aceptamos y volvemos a entrar en la aplicación para terminar de configurar el Android SDK.

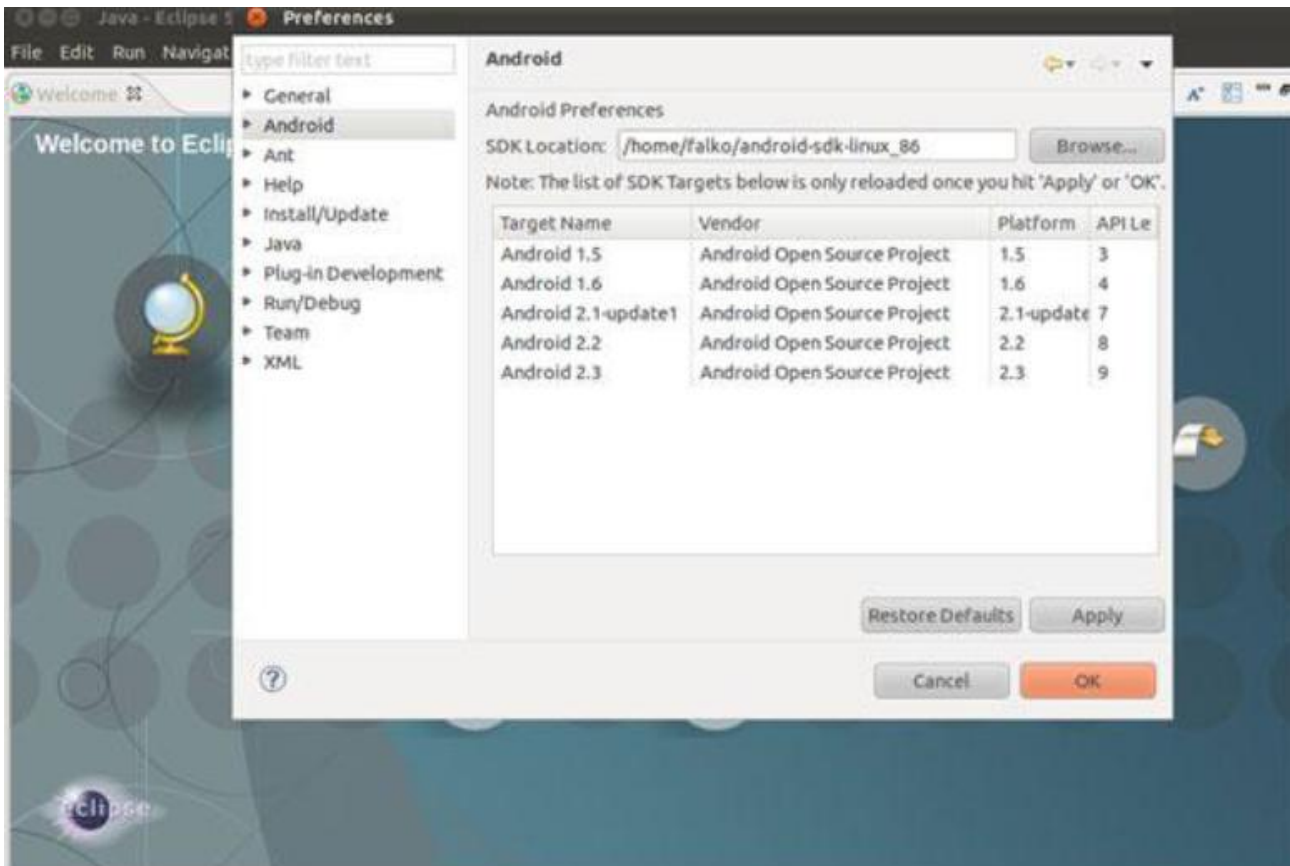
Seleccionamos la opción: *Windows > Preferences*

En la ventana emergente buscaremos en el panel izquierdo la opción de Android y buscaremos la ruta donde se encuentra instalado el SDK.



Una vez seleccionado el directorio donde tenemos la aplicación, en nuestro caso android-sdk-linux\_86, haremos click en la opción "Apply" y "OK".

Para concluir, no aparecerá el listado de versión Android soportadas por la aplicación.



## 8.2 Utilización de Subversive

Adjuntamos en este anexo parte del tutorial "Subversive, cliente de Subversion para Eclipse" publicado en la web de Autentia Real Business Solutions S.L., empresa de soporte y desarrollo informático, como guía de instalación del software utilizado en la ejecución de este proyecto.

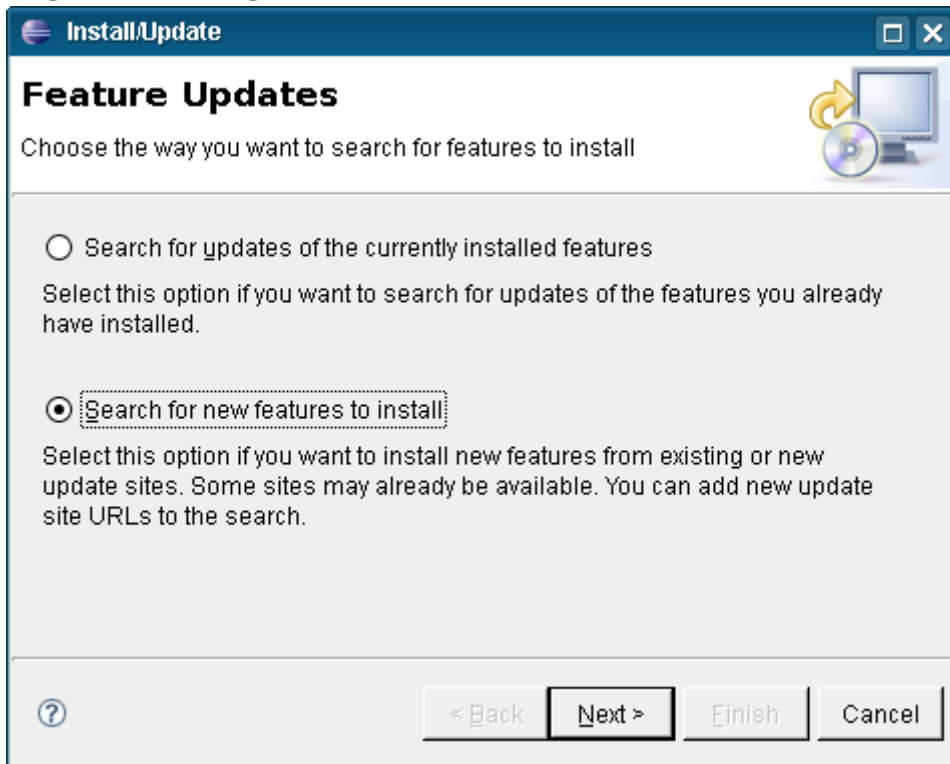
### 1. Introducción

En este tutorial vamos a introducirnos en el manejo de Subversive, un plugin para Eclipse que nos permite trabajar con el ampliamente extendido sistema de control de versiones Subversion (<http://subversion.tigris.org/>).

Subversive es un producto open source de Polarion Community (<http://www.polarion.org/index.php?page=overview&project=subversive>).

### 2. Instalación

Vamos a hacer la instalación usando el "Eclipse Update Manager". Para ello pinchamos sobre el menú: Help --> Software Updates --> Find and Install...

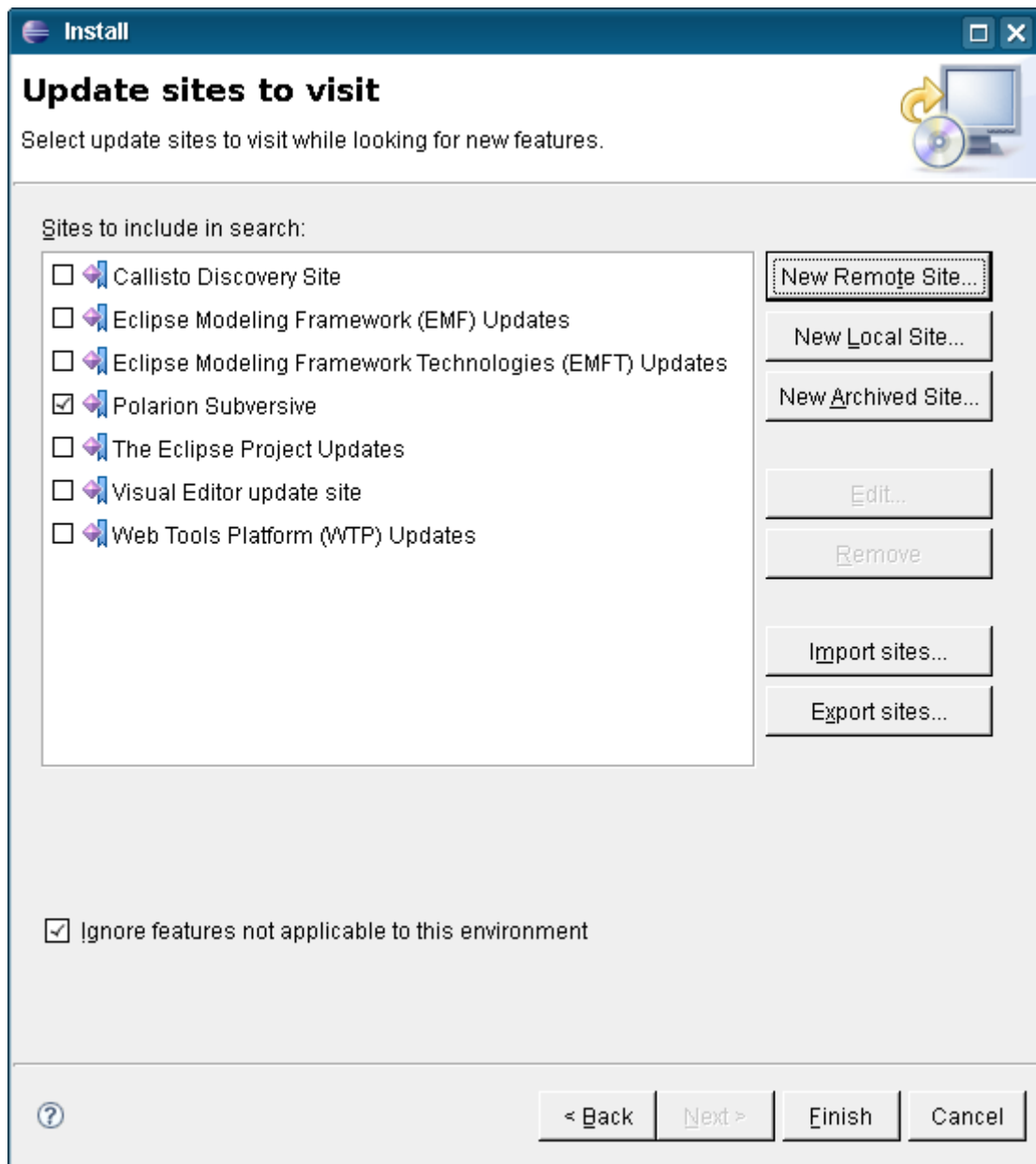


Seleccionamos "Search for new features to install", y pulsamos el botón "Next >".

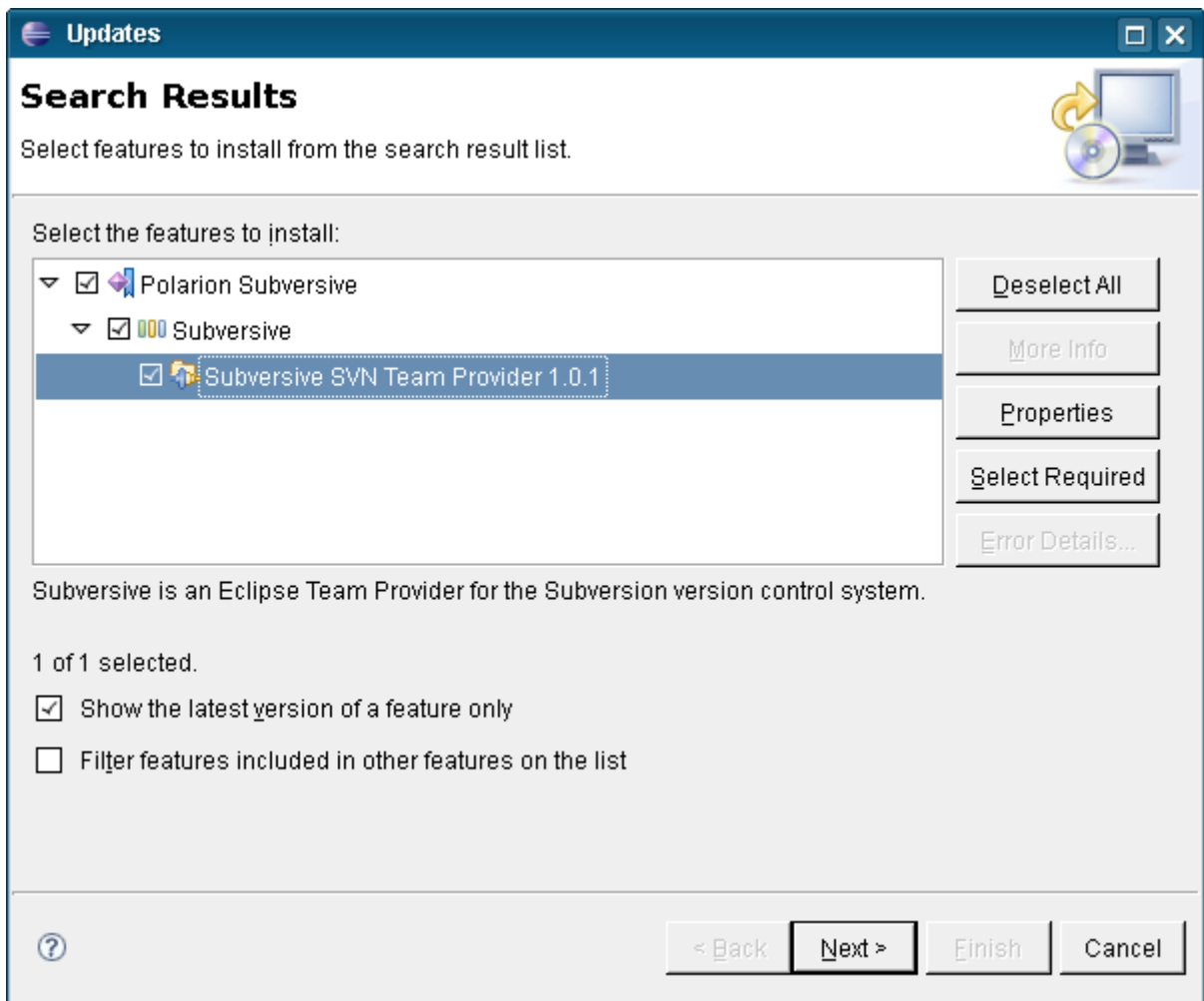
Pinchamos sobre el botón "New Remote Site...".

Nos aparece una ventana donde pondremos el nombre del sitio donde se buscarán actualizaciones de Eclipse (podemos poner lo que queramos, por ejemplo "Polarion Subversive"), y la URL del sitio. Esta debe ser: <http://www.polarion.org/projects/subversive/download/update-site/>

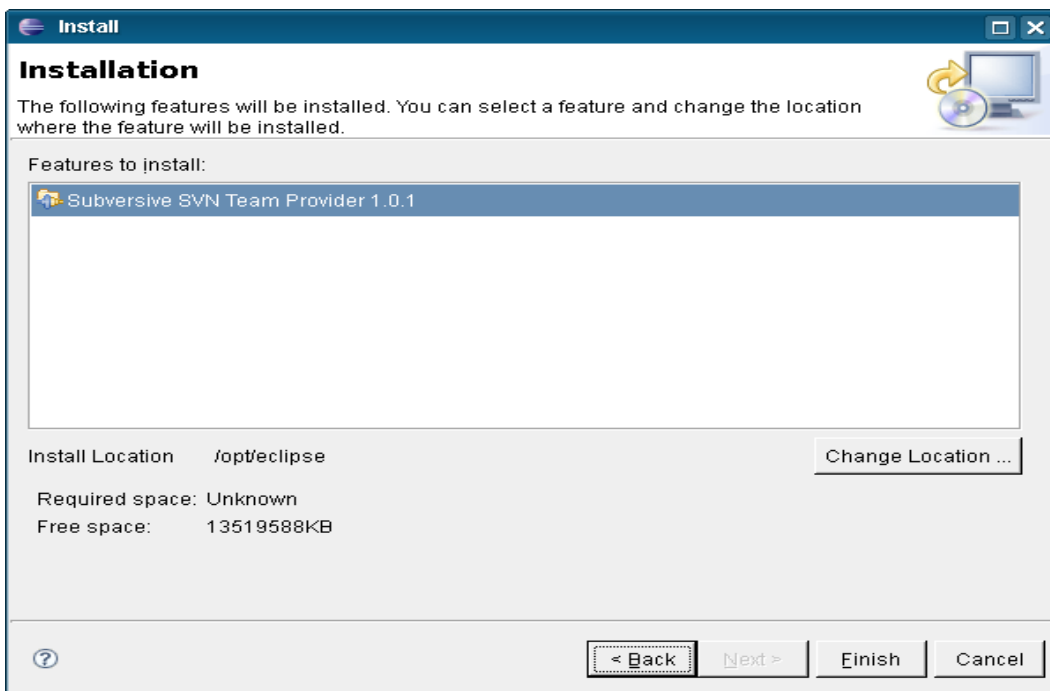




Vemos como nos aparece ya marcado el nuevo sitio que acabamos de añadir. Pulsamos sobre el botón “Finish”.

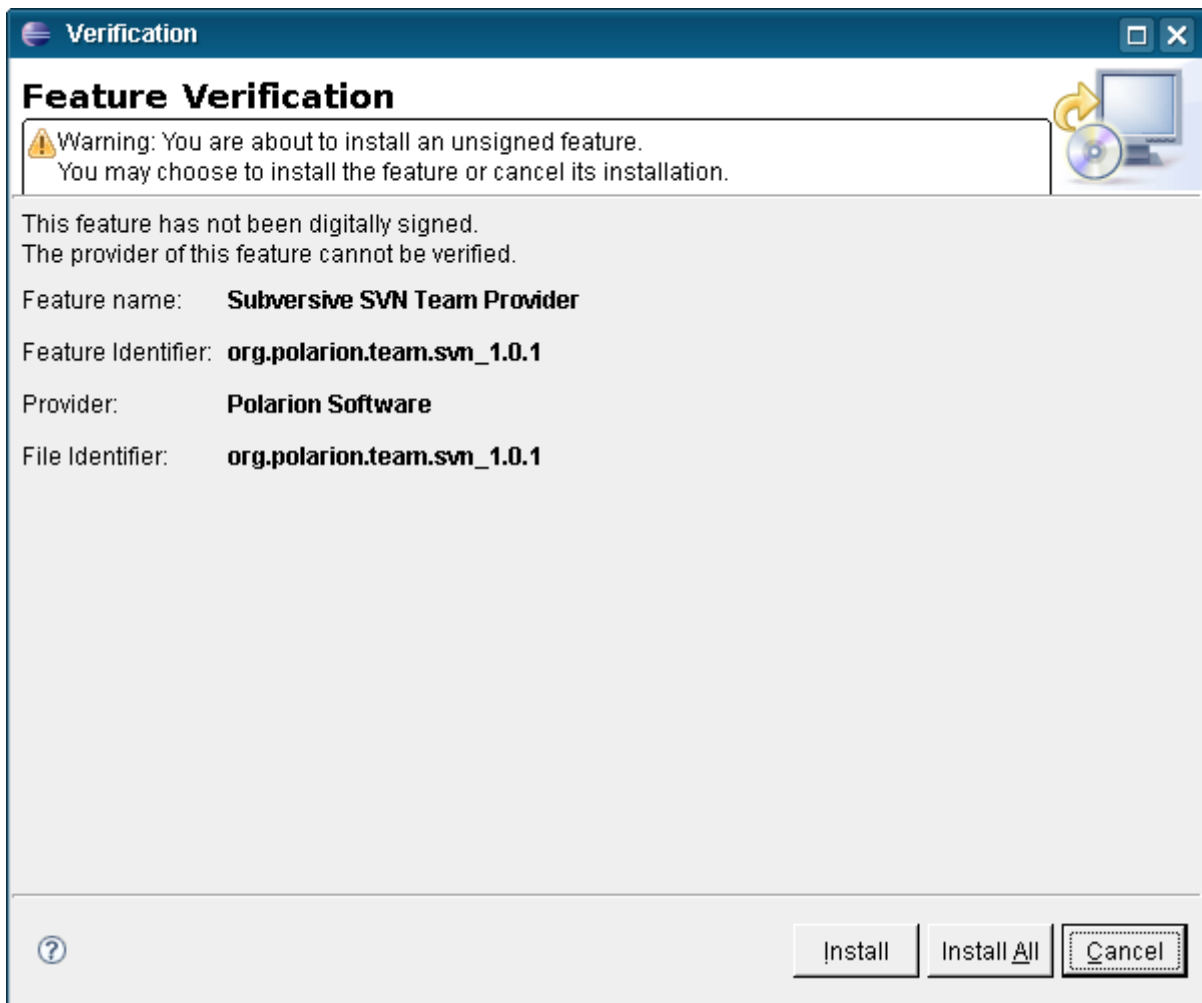


Nos aparece una ventana donde debemos seleccionar la actualización de Eclipse que queremos instalar. En este caso “Subversive SVN Team Provider 10.1”. Y Pulsamos sobre “Next V”. Aceptamos la licencia y pulsamos sobre el botón “Next >”

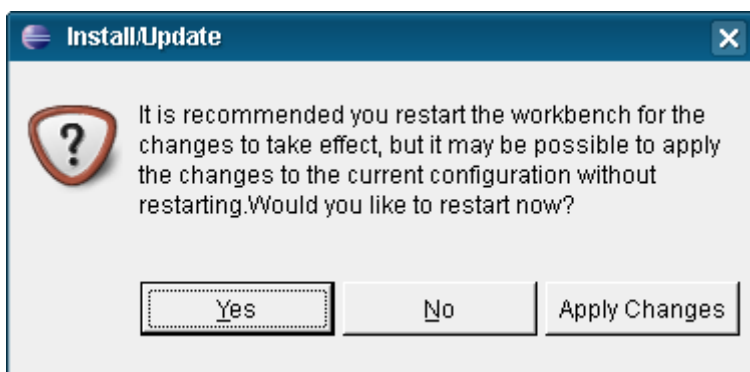




Pulsamos sobre “Finish” para comenzar la descarga e instalación.



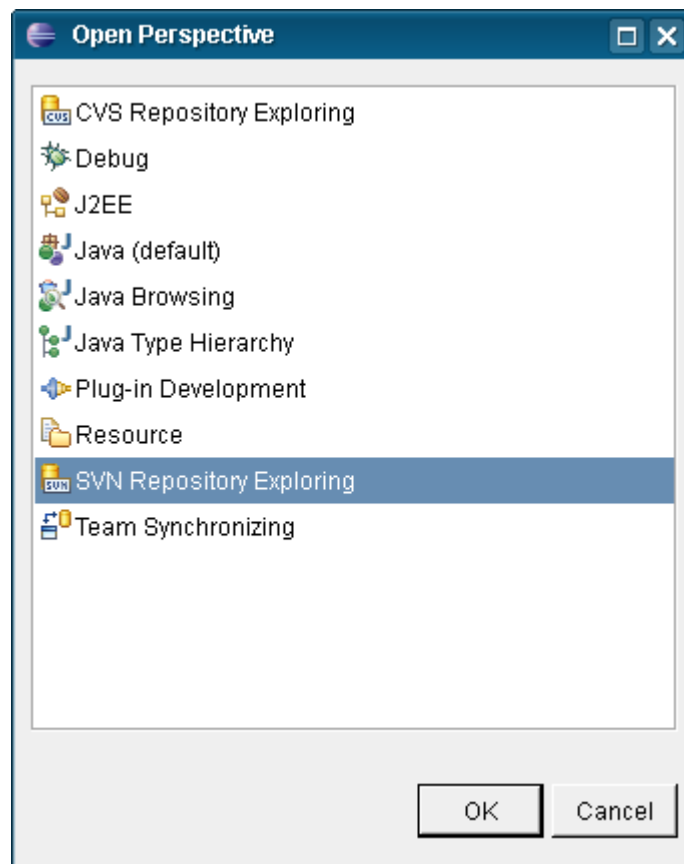
Confirmamos la instalación pulsando sobre “Install All”.



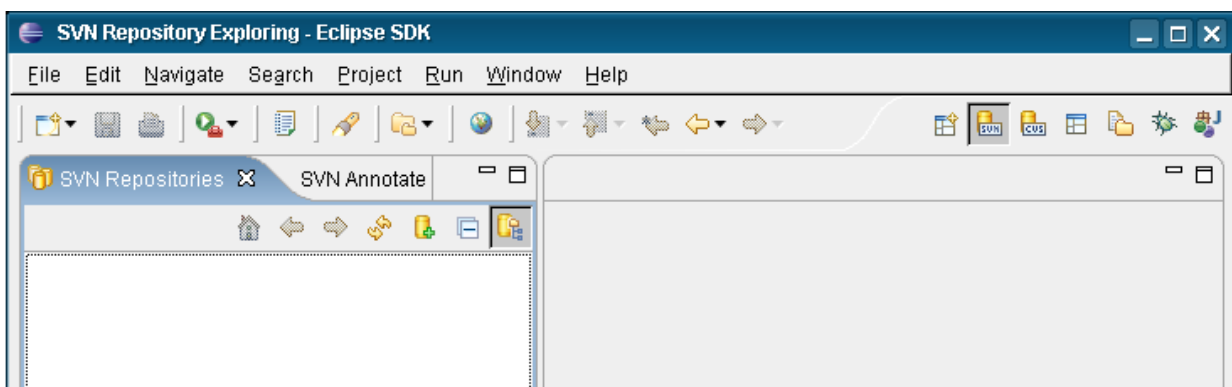
Una vez terminada la instalación es conveniente reiniciar el entorno, así que pulsamos sobre el botón “Yes”.

### 3. Dando de alta un repositorio

Para poder usar un repositorio de Subversion lo daremos de alta en la perspectiva “SVN Repository Exploring”. Para ello podemos pinchar sobre Window --> Open Perspective --> Other...



Seleccionamos “SVN Repository Exploring” y pulsamos el botón “OK”.



Sobre la vista “SVN Repositories” pulsamos con el botón derecho del ratón y seleccionamos New --> Repository Location...

**Edit Repository Location**

### Enter Repository Location Information

Define the SVN repository location information. You can specify additional settings for proxy and svn+ssh, https connections.

General | Advanced | SSH Settings | SSL Settings | Proxy

Root URL:  Browse

Label

Use the repository URL as the label

Use a custom label:

Authentication

User:

Password:

Save password

⚠ Saved passwords are stored on your computer in a file that's difficult, but not impossible, for an intruder to read.

Validate Repository Location on finish

Reset Changes

Finish Cancel

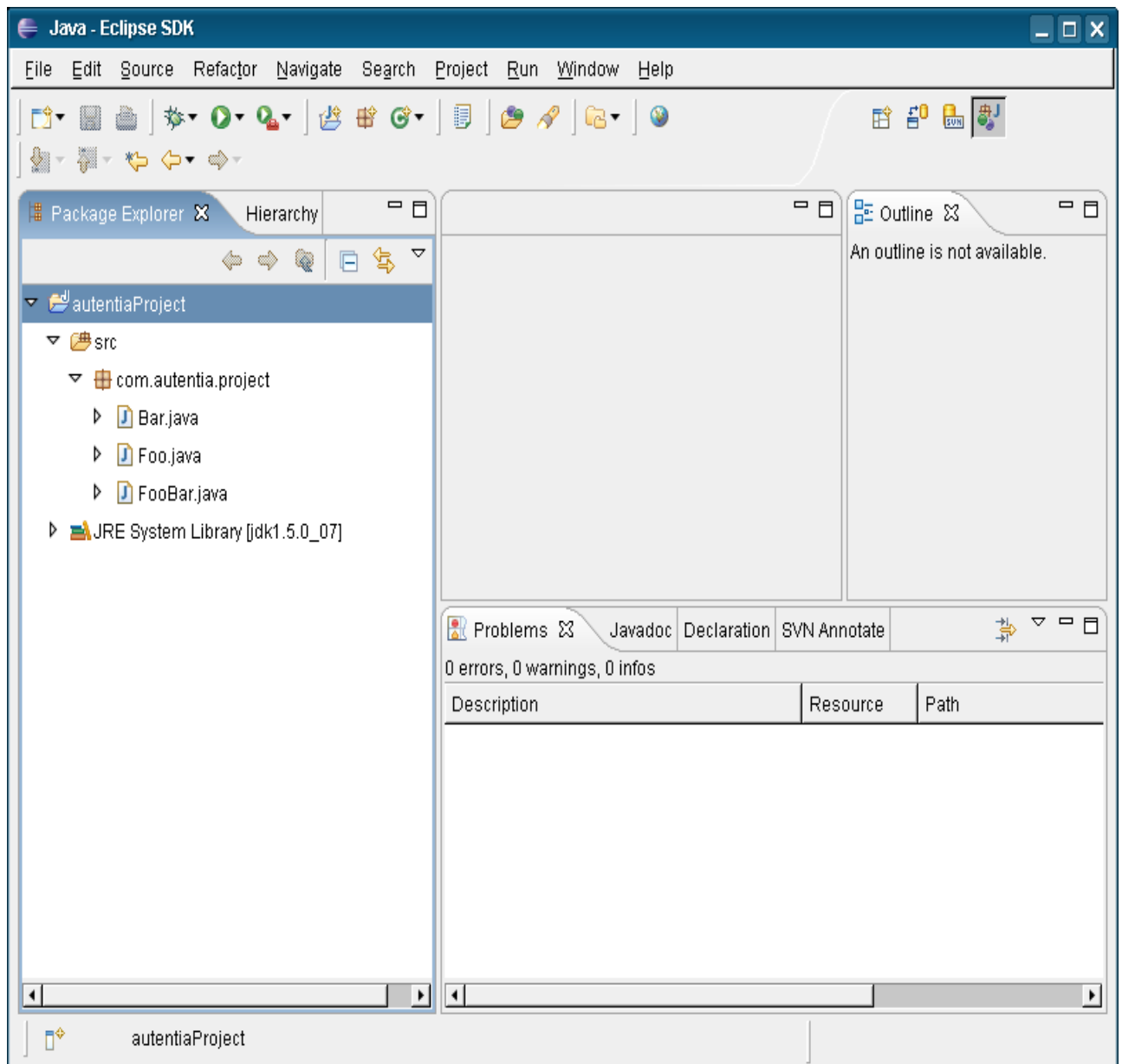
En esta pantalla configuramos la URL por la que accederemos al repositorio. Esta URL empezará por `http://`, `svn://` o `svn+ssh://` dependiendo de como esté configurado Subversión en el servidor (habrá que consultar con nuestro administrador de Subversion).

Ponemos nuestro usuario y nuestra clave y pulsamos sobre el botón “Finish”.

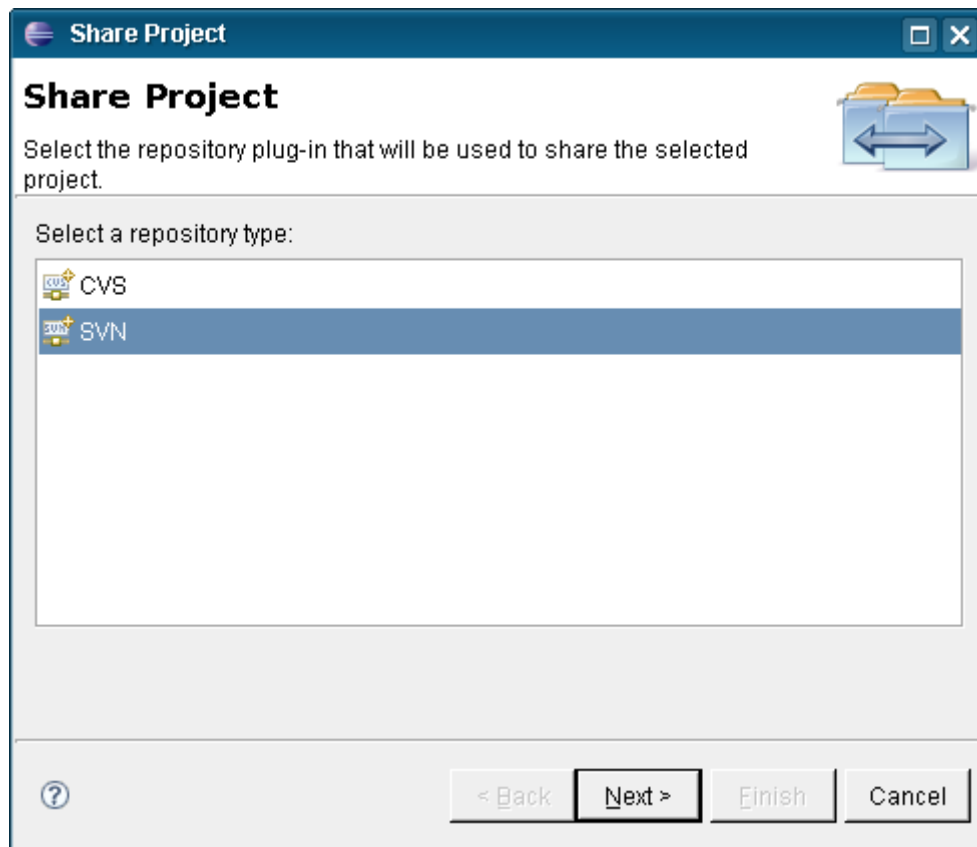
## 4. Trabajando con proyectos

### 4.1. Añadiendo un proyecto al repositorio

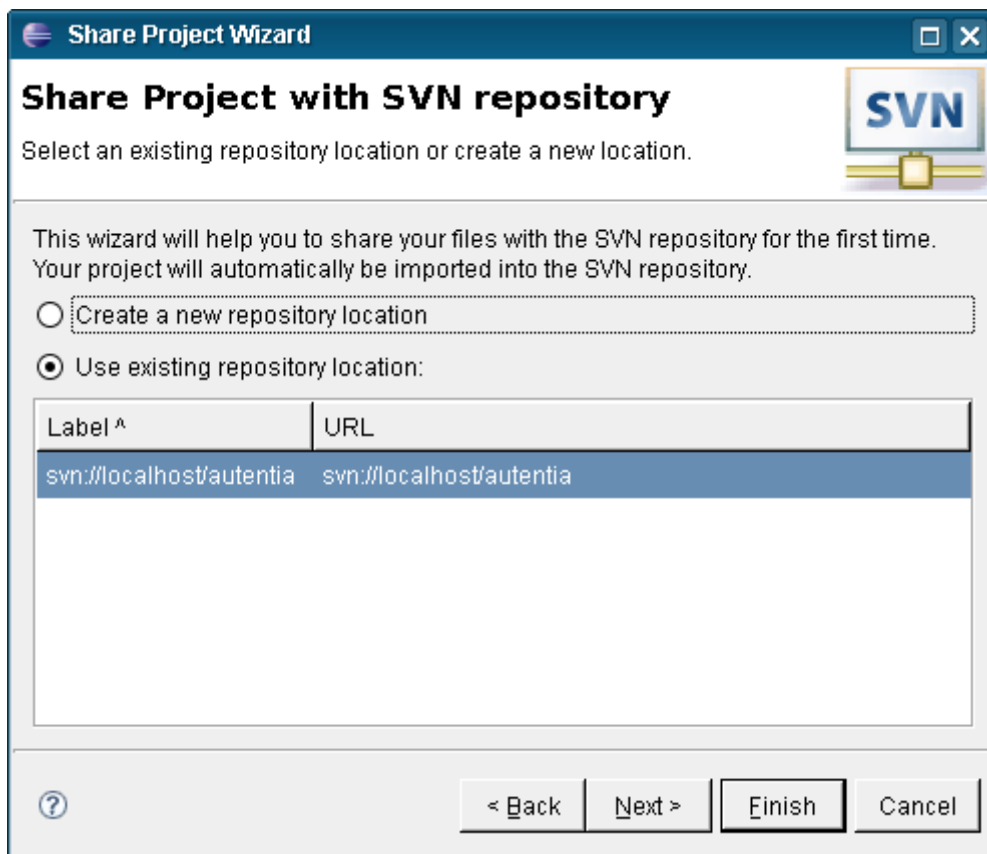
Lo primero que vamos a hacer es dar de alta un proyecto en el repositorio que acabamos de añadir. Para ello vamos a la perspectiva Java donde tenemos creado un pequeño proyecto de prueba.



Pulsamos con el botón derecho del ratón sobre el nombre del proyecto y seleccionamos la opción Team --> Share Project...



Seleccionamos la opción “SVN” para añadir el proyecto a un repositorio de Subversion, y pulsamos el botón “Next >”.



Seleccionamos de la lista el repositorio en el que queremos añadir el proyecto, y pulsamos sobre el botón “Next >”.

**Share Project Wizard**

### Enter a Project name

Select the name of the project in the SVN repository.

Name on Repository

Use project name

Use specified name:

autentiaProject

Project Repository Layout

Use Repository Location layout

Use single project layout

Use multiple projects layout with the specified root name:

autentiaProject

Use Subversion recommended layout ('trunk', 'branches' and 'tags')

Project files location on the repository will be different depending on the selected layout type. You can see future files location below:

svn://localhost/autentia/autentiaProject/trunk

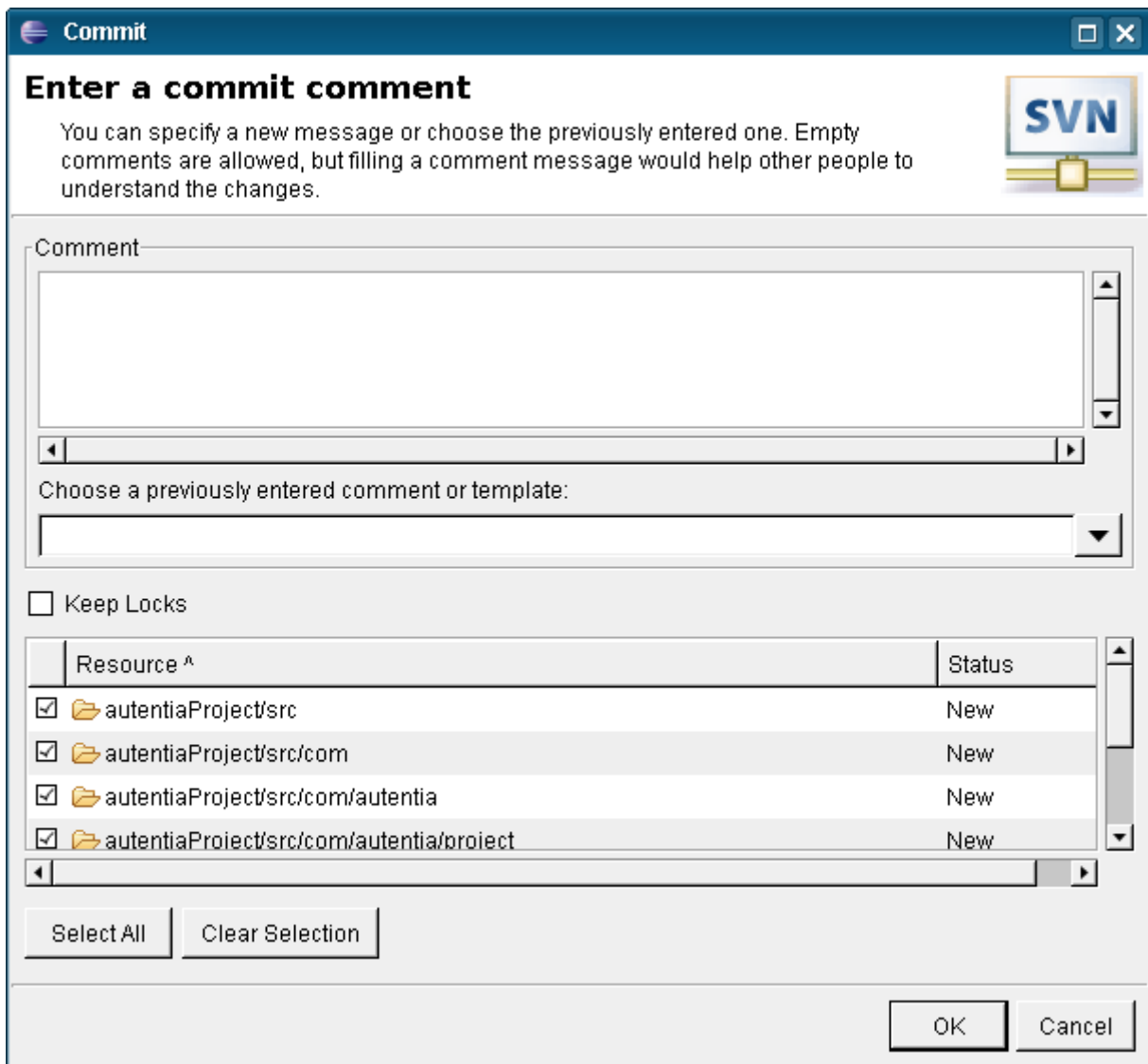
< Back   Next >   Finish   Cancel

Esta ventana es muy importante ya que aquí le diremos la disposición (layout) del repositorio, y esto afecta directamente a la estructura de directorios que se va a crear.

La ventana nos presenta tres opciones, yo escojo la segunda “Use single project layout”. Con esta opción conseguimos que dentro de nuestro repositorio “autentia” se cree un directorio por cada proyecto (en este caso “autentiaProject”), y dentro de este se crearán tres directorios: trunk (donde haremos el desarrollo del día a día), branches (donde guardaremos las ramas), y tags (donde guardaremos la versiones etiquetadas).

Para más información sobre el “layout” del proyecto, es recomendable leer <http://svnbook.red-bean.com/nightly/en/svn.reposadmin.projects.html#svn.reposadmin.projects.chooselayout>.

Una vez elegida la opción que nos interesa, pulsamos sobre el botón “Finish”.



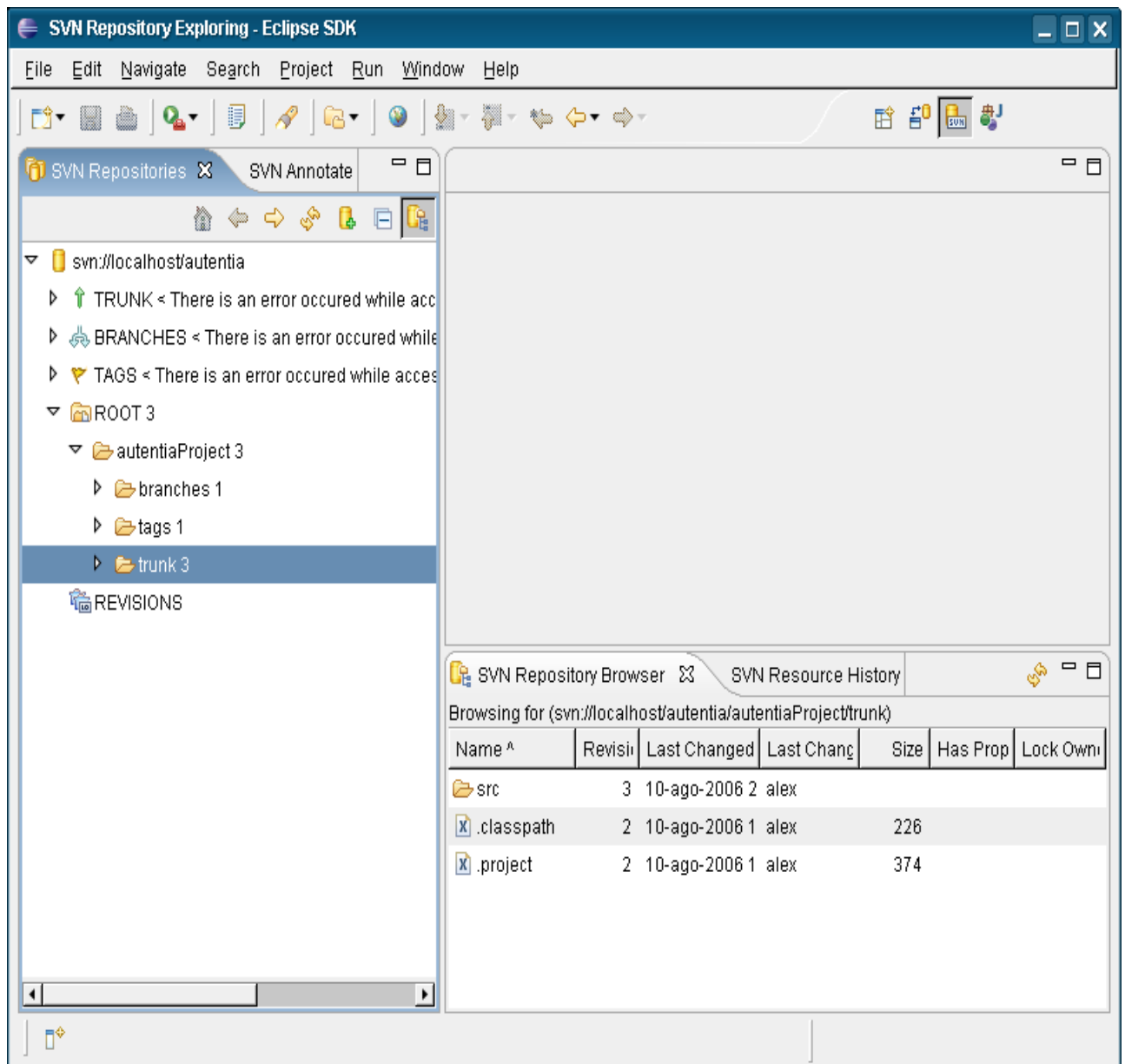
Ahora nos aparece la ventana de Commit (la vamos a ver muchas veces). En esta ventana se nos muestra que cambios se van a subir al repositorio. Debemos (no es estrictamente obligatorio, pero es muy buena práctica) poner un comentario que refleje los cambios que se han hecho en los archivos que se van a subir al repositorio.

Pulsamos sobre el botón "OK" y ya tenemos el proyecto en el repositorio.

#### 4.2. Descargando un proyecto del repositorio

Ahora vamos a hacer la operación inversa. Supongamos que nos incorporamos a un nuevo equipo de trabajo y necesitamos descargar del repositorio los proyectos pertinentes.

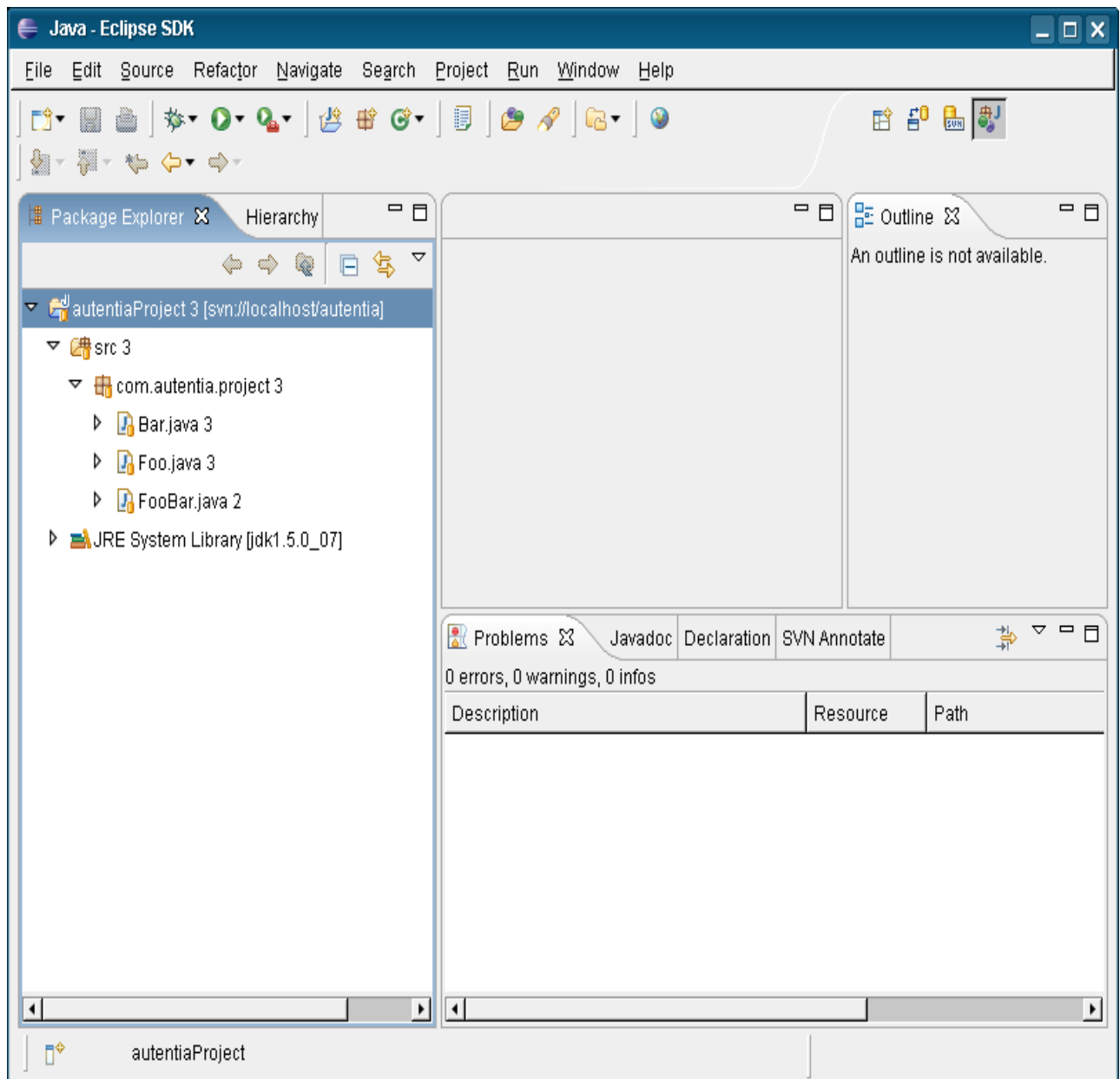
Abrimos la perspectiva "SVN Repository Exploring", y después de dar de alta el repositorio (ver punto 2.) expandimos sus ramas hasta ver algo como:



Pinchamos con el botón derecho del ratón sobre la carpeta “trunk” (donde se hace el desarrollo del día a día), y seleccionamos la opción “Check Out”.

Ya hemos terminado, si vamos a la perspectiva Java veremos como ya tenemos en nuestro entorno el proyecto.





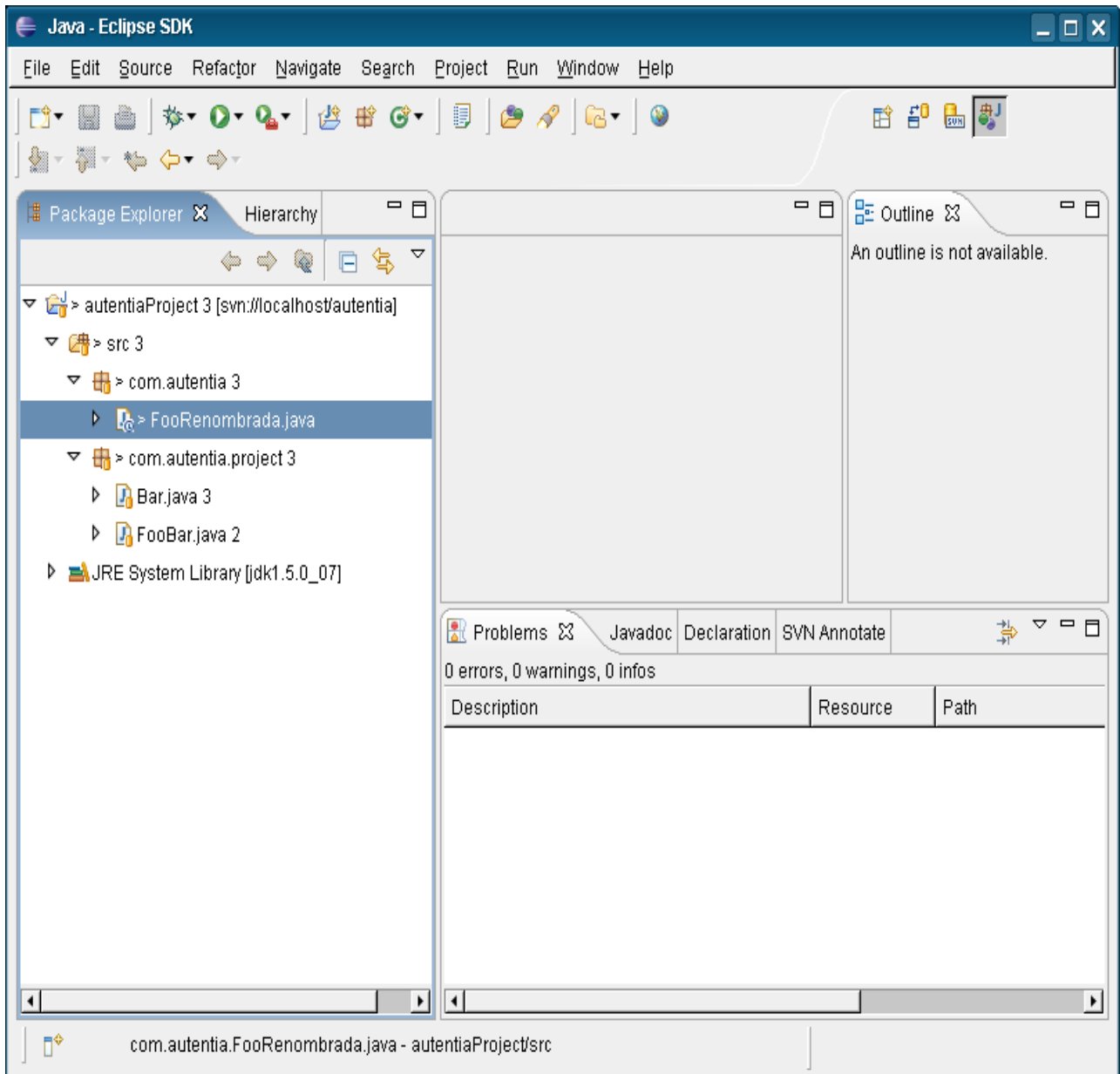
#### 4.3. Haciendo commit en el repositorio de nuestros cambios locales

Como ejemplo de cambios en nuestro proyecto local vamos a ver una modificación que implique cambio de nombre y de localización de un fichero.

Con este ejemplo se pretende ilustrar una de las capacidades más interesantes que tiene Subversion: el versionado de los metadatos.

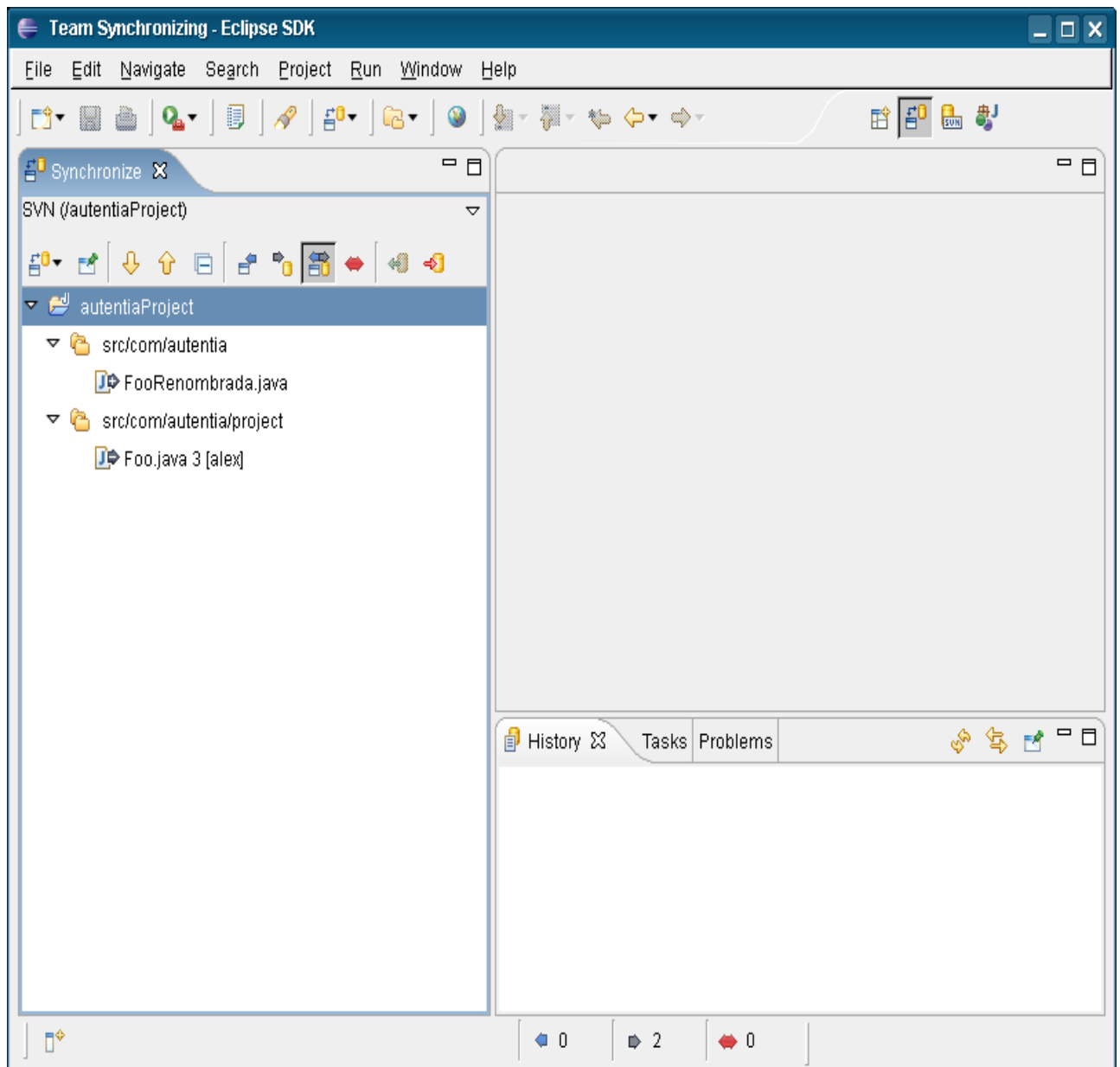
Por el contrario, CVS no es capaz de mantener el histórico de un fichero si este cambia de nombre o de lugar. Para CVS es como si el fichero se borrara y se creara uno nuevo, con lo que se pierde todo el histórico (realmente no se pierde porque el fichero borrado siempre es recuperable, pero entre el antiguo y el nuevo no existe conexión, por lo que resulta difícil localizarlo).

Si vemos la siguiente imagen y la comparamos con la imagen del apartado anterior, se puede apreciar como la clase Foo se ha renombrado y cambiado de paquete.



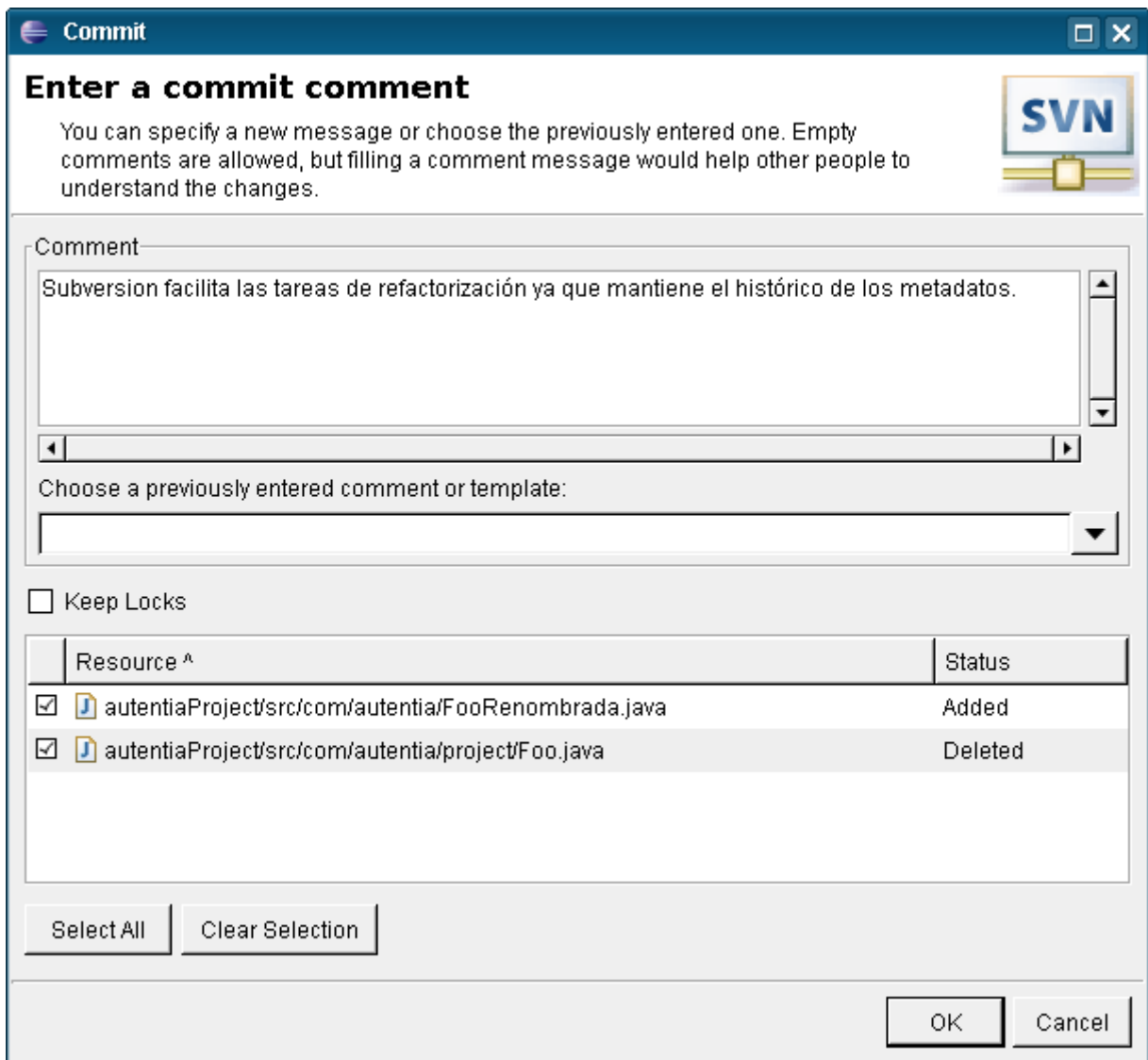
En la imagen también se puede apreciar como Eclipse marca con el carácter “>” los cambios producidos en local, con respecto al repositorio.

Ahora sobre el proyecto pinchamos con el botón derecho del ratón y seleccionamos la opción Team --> Synchronize with Repository. Esto nos lleva a la perspectiva de sincronización, donde podemos hacer commit de nuestros cambios locales, o update de los cambios en el repositorio.



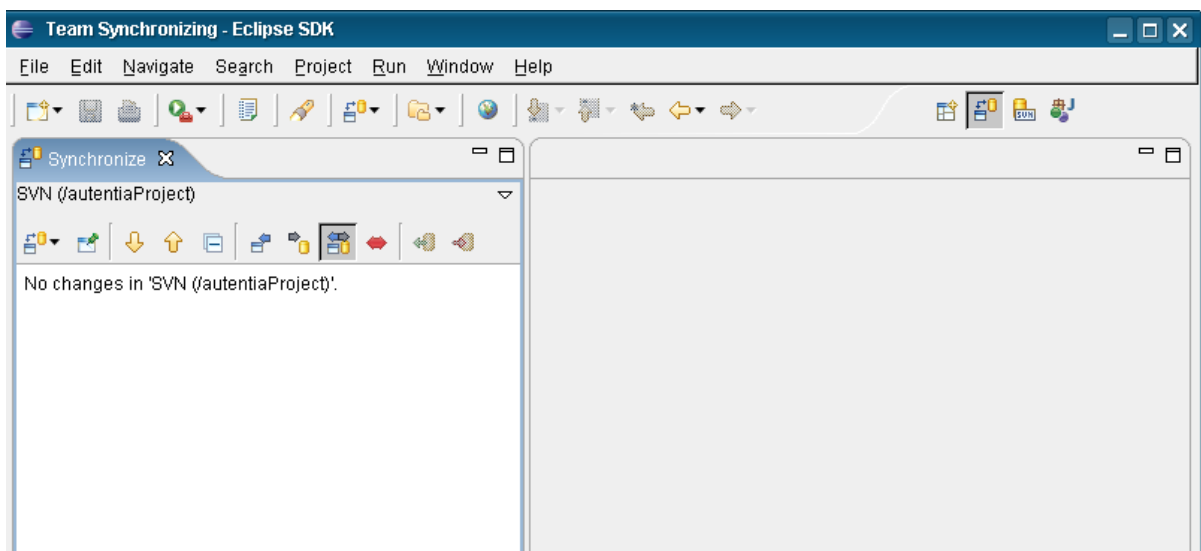
Si desplegamos las ramas vemos que la vista de sincronización nos marca dos cambios, el borrado de un fichero y la aparición de uno nuevo. Por ahora parece el mismo comportamiento que en el CVS, pero tengamos un poco de paciencia.

Pulsamos el icono (Commit All Outgoing Changes) más a la derecha dentro la vista de sincronización (el icono que es una flecha apuntando a un cilindro). Con este icono subimos todos los cambios que tengamos en local al repositorio.

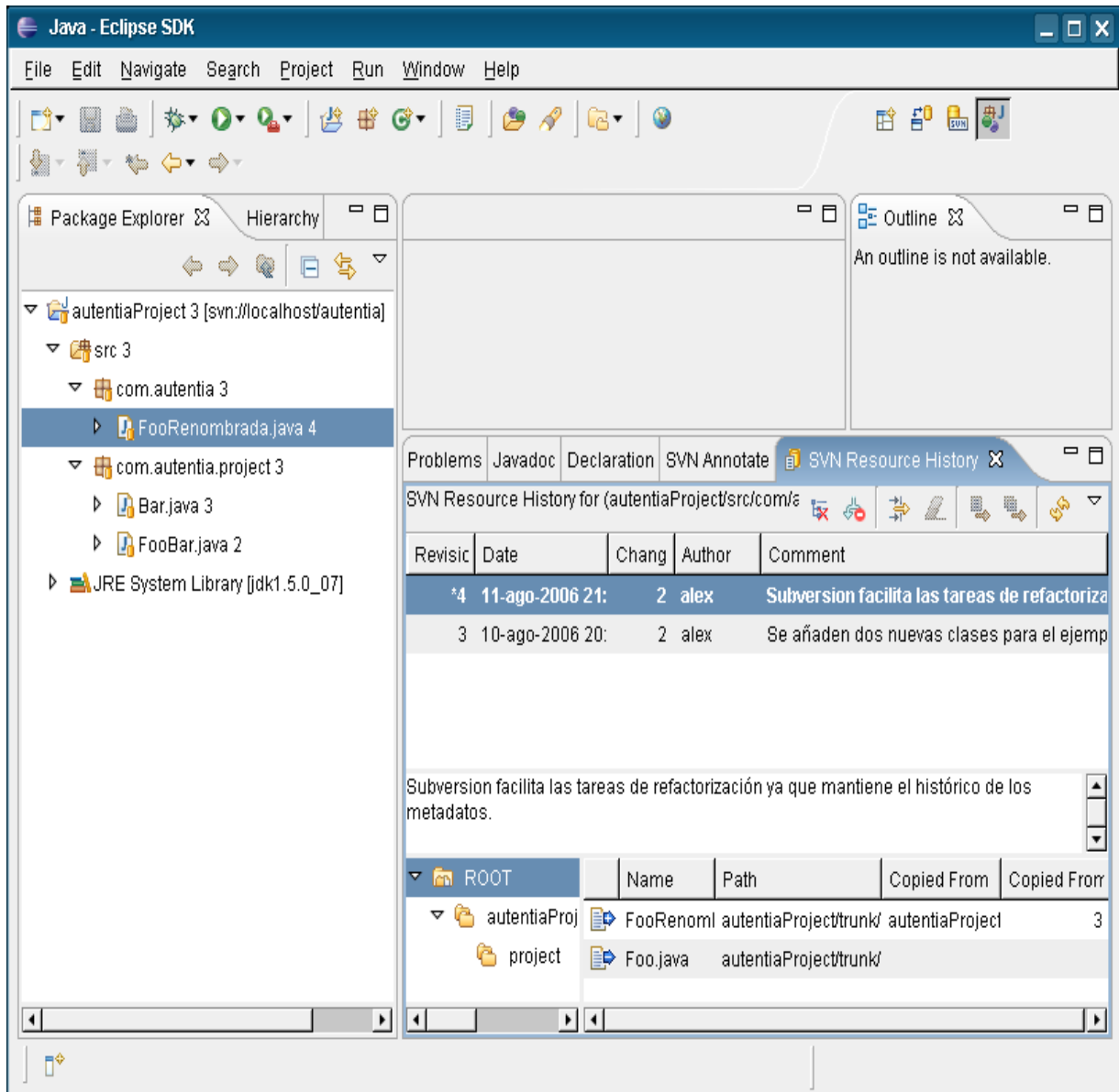


Nos encontramos en la ya conocida ventana de “Commit”. Introducimos un comentario y pulsamos sobre el botón “OK”.

Se ve como ya no quedan cambios que subir al repositorio.



Volvemos a la perspectiva de Java, y pinchamos con el botón derecho del ratón sobre la clase FooRenombrada y seleccionamos la opción Team --> Show Resource History



En la vista “SVN Resource History” podemos ver el histórico de la clase. Se puede apreciar como Subversion mantiene el histórico a pesar del cambio de nombre y de haber cambiado de sitio la clase. De hecho en el histórico se puede ver donde estaba antes y como se llamaba.