Anexo I – Plan del Proyecto Software

**Lista de cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Descripción** | **Autor** |
| 1.1 | 11/10/2011 | Esqueleto inicial | Romina Liuzzi |

Tabla de contenidos

[Introducción 5](#_Toc306137160)

[Planificación temporal del proyecto 6](#_Toc306137161)

[Estudio de viabilidad del proyecto 9](#_Toc306137162)

[Resumen de gestión y recomendaciones 11](#_Toc306137163)

[Alternativas 12](#_Toc306137164)

[Descripción del sistema 13](#_Toc306137165)

[Análisis coste/beneficio 14](#_Toc306137166)

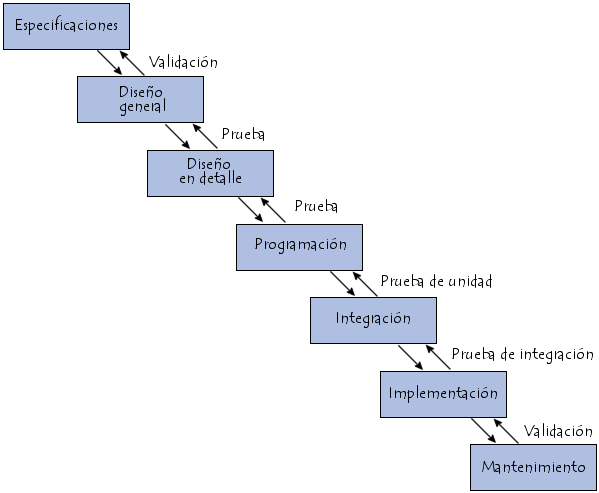
[Evaluación de riesgo técnico 18](#_Toc306137167)

[Consideraciones legales 20](#_Toc306137168)

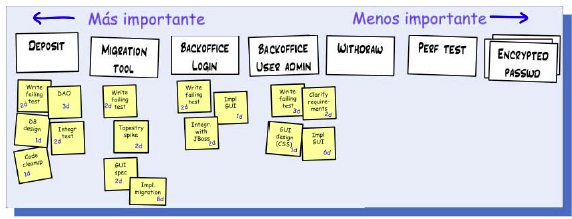
[Otros temas específicos del proyecto 22](#_Toc306137169)

# Introducción

# Planificación temporal del proyecto



Si bien durante el transcurso de la carrera nos hemos familiarizado con metodologías de planificación fuertemente teóricas tales como METRICA v3 y herramientas como diagramas de Gantt o redes de PERT para cuantificar el progreso de un proyecto. En proyectos de fin de carrera, quizás debido a la falta de experiencia, es difícil ceñirse a estas que pautas tan rígidas.

Recientemente he entrado en contacto con SCRUM (o su versión compactada y adaptada al mundo corporativo). En la práctica, empresas punteras están optando por adaptar esta visión reduciendo la documentación técnica, utilizando ayudas visuales más dinámicas (en nuestro caso pizarras) y reaccionando al cambio con agilidad. En síntesis la idea subyacente viene a ser la importancia de trabajar en tareas tan pequeña como sea posible, y debidamente priorizadas. De forma que el proyecto avanza y es posible contar con entregables cada sprint. Un sprint puede durar de 2-3 semanas dependiendo de la naturaleza del proyecto.  


En este proyecto en concreto la mayor parte del tiempo se invirtió en dos tareas muy abstractas.

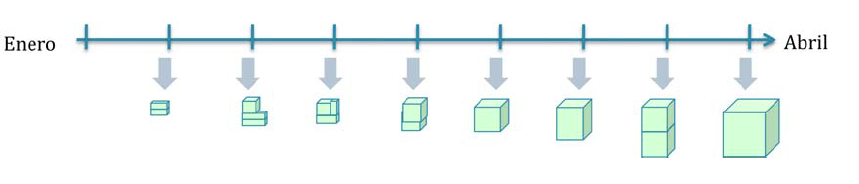
La primera consistió en el estudio previo y determinación de plataforma, lenguaje de programación e IDE. A la vez que se llevo a cabo este estudio se hizo un balance de los costos implicados en cada una de las opciones consideradas.

La segunda consistió en una inmersión en el mundo del desarrollo de aplicaciones para móviles, donde debemos comprender el hardware sobre el que estamos trabajando, las capas intermedias, la estructura, y una vez tenemos las ideas claras, decidir la solución a nuestro problema.

En mi caso una vez superada la fase de investigación que insumió una parte considerable de mis recursos, las tareas se volvieron más concretas:

1. Conseguir la ejecución de servicios en background (GPS).
2. Almacenamiento de datos en una tabla.
3. Tratamiento de datos a partir de helpers.
4. Diseño y desarrollo de Vistas de usuario.
5. Separación en Actividades.

Estas actividades son bastante más técnicas y todas constituyen partes fundamentales en la resolución del problema que plantea mi aplicación: Calcular las calorías quemadas por un usuario mediante la realización de un ejercicio.



# Estudio de viabilidad del proyecto

# Resumen de gestión y recomendaciones

# Alternativas

# Descripción del sistema

# Análisis coste/beneficio

Las opciones evaluadas durante el transcurso de la investigación derivaron en las siguientes tres finalistas, cada uno presentado con sus costes asociados (el análisis de riesgos se lleva a cabo en la siguiente sección):

**APPLE:**

**Sistema Operativo:** iOS 4.2

**Lenguaje de Programación:** Objective C **(\*)**

**Framework / GUI Library:** Cocoa (MVC - Modelo Vista Controlador)

**IDE:** iOS SDK 4

**SDK:** xCode 3.1

**Distribución de apps:** App Store

**Terminales más populares**

\* iPhone 3Gs

\* iPhone 4

***Notas:*** El SDK se distribuye de forma gratuita desde marzo de 2008 a desarrolladores registrados. Registrarse es gratis. La suite incluye un emulador de iPhone para realizar pruebas. Sin embargo para cargar la aplicación en un iPhone se la debe subir primero al App Store, para lo que hay que pagar un derecho de membresia. Existen tres tipos, en este caso costaría $99 (aproximadamente 75€).

**ANDROID OS**

**Sistema Operativo:** Android

**Lenguaje de Programación:** Java

**IDE:** Eclipse 3.4 o 3.5 (Android Development Tools Plugin)

**SDK:** Android SDK

**Distribución de apps:** Android Market (gratis)

**Terminales más populares**

\* HTC EVO 4G (no disponible en España, no usa tarjeta SIM)

\* Motorola Droid 2 / Milestone 2 (No lo encuentro en España)

\* HTC WildFire

\* Samsung Galaxy S

\* Samsung Galaxy 3 i5800

\* HTC Hero

\* Sony Ericsson Xperia X8

***Notas:*** Esta opción es la más conveniente a nivel costo/beneficio, ya que el SDK se distribuye de forma gratuita así como el IDE sugerido por Google: Eclipse. Google además distribuye sin coste el ADT (Android Development Toolkit) que puede integrarse en Eclipse de forma casi transparente. Los emuladores están incluidos en el plugin, el numero de terminales que podemos emular es casi ilimitado ya que se incluye un AVD manager que permite crearlos al instante. Finalmente toda la documentación con ejemplos y tutorial es distribuida oficialmente a través de android.developer.com. Gracias al Standard OpenSource nos encontramos con infinidad de ejemplos distribuidos extraoficialmente de forma gratuita. El único coste asociado a esta opción surge a la hora de distribuir nuestra aplicación a través de Android Market, el costo es simbólico $25 y a diferencia del caso anterior, no necesitamos pagarlo para instalar el app en nuestro móvil.

**SYMBIAN OS (Nokia)**

**Sistema Operativo:** Symbian OS (Symbian 9.5, Symbian^3)

**Lenguaje de Programación:** C++ **(\*)**

**Framework / GUI Library:** Qt

**IDE:** Qt Creator

**SDK:** Nokia Qt SDK 1.0, Symbian SDK, Carbdie.c++

**Distribución de Apps:** Ovi Store, PC Suite, Nokia Store, Samsung Store

**(\*)** Existe un proyecto open source desde marzo de 2010 ([Qt Jambi](http://www.google.com/url?q=http%3A%2F%2Fqtjambi.sourceforge.net%2F&sa=D&sntz=1&usg=AFQjCNGYW-TjFI1-P3LsBuphzvccgZ3-dA)) mediante el cual se podría programar la aplicación en Java utilizando librerias qt o programar en java y generar el código qt.

***Notas:*** Esta opción cuenta con un coste de iniciación del orden del de Apple, y básicamente los mismo problemas, restricciones técnicas, falta de información libre. Esta opción es la menos popular.

**Terminales más populares**

\* Nokia N97 mini

\* Nokia C7

\* Nokia C6

\* Nokia N95 (es más antiguo, se ha usado en proyectos similares, solo tiene GPS)

# Evaluación de riesgo técnico

El riesgo técnico asociado a la realización de este proyecto dependen de la opción seleccionada en al apartado anterior.

OPCIÓN APPLE:

Apple es muy restrictivo en cuanto su hardware, nuestra capacidad de libertad a la hora de tomar una decisión técnica se ve acotada por las restricciones impuestas por su modelo. El desarrollo de nuestra aplicación dentro de estas condiciones se vio totalmente mermado siendo (más allá del coste técnico) el factor determinante a la hora de descartar esta opción.

1. Apple no admite la multitarea real.
2. Apple no permite la ejecución de servicios en background a aplicaciones desarrolladas por terceros (es decir aplicaciones no nativas).

Nuestra aplicación requiere que un servicio de localización (via GPS o torre wifi) se ejecute en modo background mientras la aplicación está en curso, recogiendo datos que luego se estudian para dar el resultado final. Al no estar clara la manera de saltarse las restricciones anteriormente mencionadas, éstas constituyen un riesgo alto a la hora de ser capaces de desarrollar y distribuir nuestra aplicación dentro de los tiempos previstos.

OPCION ANDROID:

No se observan.

OPCION SYMBIAN:

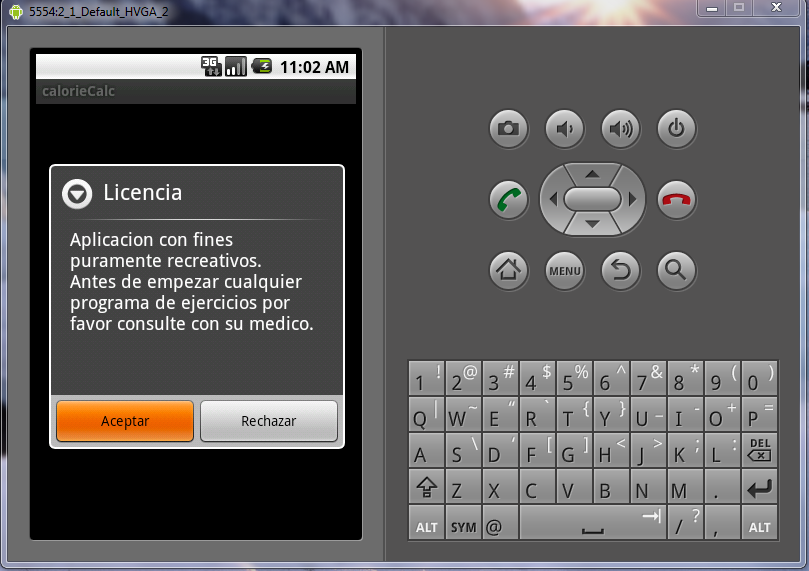
Se descarta debido a la falta de popularidad del sistema.

# Consideraciones legales

Dentro del ámbito de este proyecto surgieron principalmente dos grandes preocupaciones: Copyrights y obligaciones legales en función del tipo de programa desarrollado.

El Copyright y las normas demasiado restrictivas en cuanto a la utilización de herramientas privadas de desarrollo tales como clausulas del Registered Apple Developer Agreement que Apple distribuye junto con IDE y SDKs propietarios. Se adjunta el anexo al final del documento.

Por otro lado al tratarse de una aplicación que trata con temas relacionados con la salud, sensibles a ser mal interpretados, se decide poner un aviso legal deslindando responsabilidades e informando al usuario que la finalidad de la aplicación es puramente recreativa. Este aviso se carga al instalar la aplicación, la instalación se aborta si no se acepta la advertencia.



# Otros temas específicos del proyecto