Universitat Oberta de Catalunya

Ingeniería Técnica en Informática de Gestión

Trabajo de fin de carrera

Cefalog. Registro de cefaleas

Aplicación nativa de Android

Autor: Cira Rodríguez Brito

Consultor: Victor Carceler Hontoria

9 de junio de 2013

A Felipe,
sin su apoyo no lo hubiera conseguido

TABLA DE CONTENIDO

1.Introducción....5
1.1.Motivación....6
1.2.Alcance y objetivo del proyecto....7
1.2.1.Aplicación nativa para Android....8
1.2.2.Objetivos....8
2.Mercado y competencia....10
2.1.Diario de Dolor de Cabeza (Headache Diary)....10
2.2.Headache Diary....11
2.3.migraineDiary....11
3.Planificación....12
3.1.Plan de contingencia....13
3.1.1.Retrasos e imprevistos....13
3.1.2.Pérdida de información....13
3.2.Recursos....14
3.2.1.Equipamiento....14
3.2.2.Software....14
4.Análisis....17
4.1.Especificación de requisitos y diseño de interfaz....17
4.1.1.Inserción de información....17
4.1.2.Visualización de registros....18
4.1.3.Configuración....18
4.1.4.Compatibilidad....18
4.2.Estructura de base de datos....19
4.3.Estructura de clases....20
4.3.1.Modo lista....21
4.3.2.Modo registro....22
4.3.3.Proveedor de base de datos....23

5.Diario de desarrollo....25

5.1.Primera iteración: preparación, 6 a 8 de mayo....25

5.1.1.Creación de la estructura de la memoria....25

5.1.2.Preparación del entorno....25

5.2.Segunda iteración: desarrollo, 9 a 25 de mayo....25

5.2.1.Creación de la estructura inicial....25

5.2.2.Funcionamiento de los cuadros de diálogo....26

5.2.3.Almacenamiento de registros....27

5.2.4.Visualización de datos....27

5.2.5.Mejora de interacción en la vista para crear registros....28

5.2.6.Edición y eliminación de registros....29

5.2.7.Nota de compatibilidad....29

5.3.Tercera iteración: publicación y pruebas, 26 de mayo....30

5.4.Cuarta iteración: memoria y presentación, 27 de mayo a 9 de junio....31

6.Glosario....32

7.Bibliografía....33

7.1.Referencias para el desarrollo....33

1. Introducción

El dolor es comúnmente definido como una sensación desagradable o molesta. Éste mecanismo ha surgido durante el proceso evolutivo con el fin de comunicar, entre las distintas partes de un ser vivo con sistema nervioso, una alteración en el correcto funcionamiento del organismo, pudiendo ser causado tanto por una agresión externa como por un problema de funcionamiento de entre los subsistemas que lo forman.

En el caso específico de las cefaleas, localizadas en las zonas intra y pericraneal, la causa suelen ser, a menudo y esto parece ser exclusivo de los humanos, extremadamente difícil de diagnosticar. Su naturaleza, es a menudo incapacitante, y existe un gran porcentaje de población afectada: un 90% en mujeres y 80% en varones, durante el último año[1].

Figura 1.1. Cefalea

Las claves para facilitar un diagnóstico se centran en encontrar el foco del dolor, la intermitencia del mismo, los agravantes ambientales, y los síntomas asociados. Teniendo esto en cuenta, puede categorizarse en cefaleas primarias, en las que el dolor es única o principalmente en la cabeza, o cefaleas secundarias, que por el contrario se deben a otra enfermedad y son mucho más fáciles de diagnosticar.

No obstante, según el Dr. Felius Titus y la Dra. Patricia Pozo, el 90% de las consultas se deben a cefaleas primarias que, incidimos, son las más complejas de diagnosticar, por lo que se concluye que la migraña y la cefalea tensional son las que empujan

1 Comprender la migraña. ISBN-9788497354158

mayoritariamente a los pacientes a acudir a un especialista.

Dado el índice de incidencia sobre la población de las cefaleas, y cotejando este porcentaje con el alto índice de penetración de dispositivos móviles “inteligentes” en el mercado, surgió la idea de desarrollar una herramienta auxiliar de registro para dejar constancia de episodios de dolor a fin de facilitar la posterior tarea de diagnóstico por parte de un especialista médico.

Este tipo de “diario de migrañas” es una prescripción común, ya que facilita enormemente un diagnóstico diferencial atendiendo a la frecuencia e intensidad de los episodios, permitiendo encontrar desencadenantes ambientales, fisiológicos o neuronales, dependiendo de la hora de inicio y duración, y factores recurrentes hormonales en el caso de la periodicidad.

El disponer de una herramienta sencilla con una interfaz limpia y accesible supone minimizar la molestia de realizar el apunte durante o con posterioridad al episodio, de modo que se ha ahondado especialmente en los análisis de usabilidad de interfaz y en la minimización de los datos requeridos a fin de encontrar el subconjunto mínimo esencial de máxima relevancia.

1.1. Motivación

Las estadísticas de la incidencia sobre población son muy esclarecedoras (90% en mujeres y 80% en hombres), de modo que mi pertenencia al grupo de afectados por cefaleas es algo excepcional desde dicho punto de vista.

Se me diagnosticó migrañas al principio de mi adolescencia y, aunque no se haya podido demostrar que los episodios tengan algún tipo de causa genética, en el seno de mi familia más cercana son, también, bastante comunes.

De este modo, en el entorno de un estudio médico, un especialista me encomendó registrar los episodios de cefalea y descubrí lo complicado de registrar los episodios más fuertes (que suelen durar desde horas a días) en una libreta manuscrita dado lo altamente incapacitante de los mismos.

A día de hoy, un alto porcentaje de la población en España, un 96% según el Instituto Nacional de Estadística[2], disponen de un teléfono móvil, lo que lo convierte en una gran herramienta dónde mantener el registro, en vez de optar por un elemento separado y exclusivo para tal fin.

2 Penetración telefonía móvil en hogares, ONTSI, 3 de octubre de 2012

1.2. Alcance y objetivo del proyecto

El proyecto está enfocado en la creación de una herramienta auxiliar para el diagnóstico de cefaleas periódicas por parte de un especialista, minimizando la molestia de mantener un registro pormenorizado de las mismas.

La aplicación, para facilitar su difusión, deberá estar disponible para el mayor número de dispositivos móviles posibles, lo que implica enfocarse en dispositivos Android, un sistema operativo desarrollado por Google que, según los últimos informes de Gartner (empresa de análisis especializada en tecnologías de la información) copará el 58% del mercado en 2013, y seguirá liderando durante los siguientes años con más del 50% del mismo.

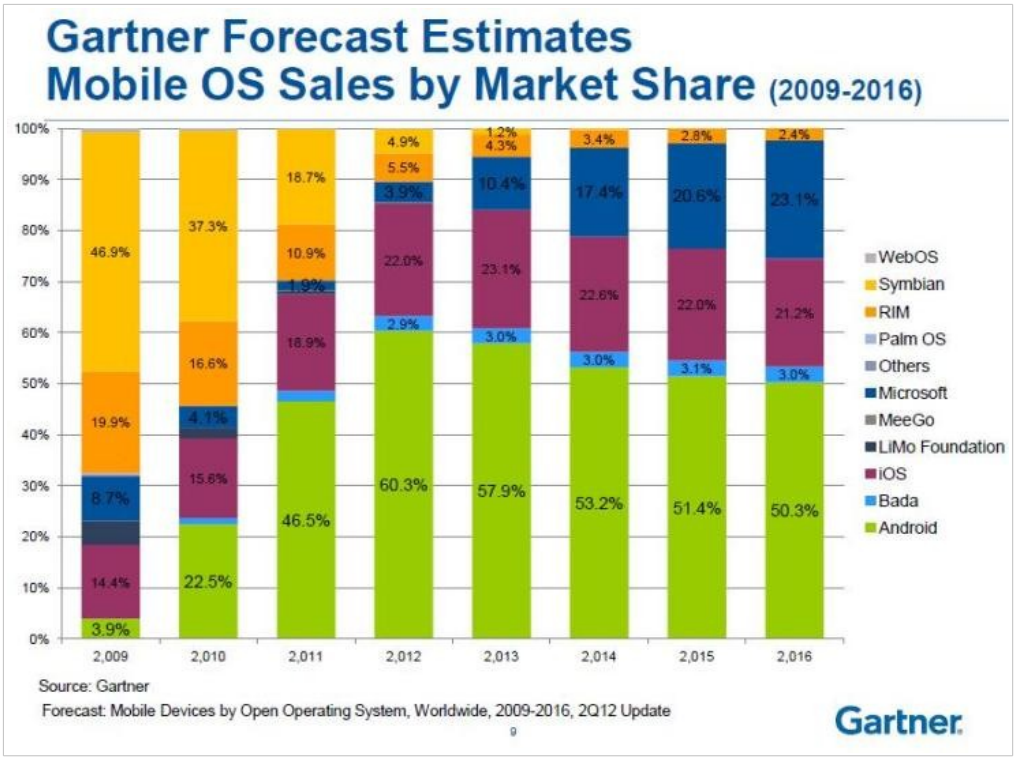

Figura 1.2. Datos y previsión de la distribución de sistemas operativos móviles en los próximos años[3]

Otro requisito en aras de la difusión, es emplear el sistema de distribución de software oficial de la plataforma, en este caso Google Play, lo que requiere cumplir ciertos requisitos y registrarse en la plataforma de desarrolladores de Google.

Además, para permitir una posterior adaptación o mejora por parte de cualquier usuario, profesional médico, o permitir su integración en cualquier plataforma médica, se ha decidido tanto hacer público el código de la aplicación, como licenciar el

3 Gartner: Bring your Own Device (BYOD): Mobility Trends and Securing the Transition. Lawrence Pingree, Research Director

mismo con una licencia de software libre,

1.2.1. Aplicación nativa para Android

A la hora de desarrollar una aplicación móvil, la principal problemática es lograr que sea compatible con el máximo de dispositivos con mayor número de propietarios, es decir, lograr maximizar el mercado potencial.

Android, una plataforma libre (salvo algunos componentes opcionales) desarrollada mayoritariamente por Google, ha logrado una gran difusión y se encuentra disponible en múltiples dispositivos electrónicos de consumo de un gran número de fabricantes alrededor del mundo.

Worldwide Smartphone Sales to End Users by Operating System in 4Q12 (Thousands of Units)

Operating System	4Q12 Units	4Q12 Market Share (%)	4Q11 Units	4Q11 Market Share (%)
Android	144,720.3	69.7	77,054.2	51.3
iOS	43,457.4	20.9	35,456.0	23.6
Research In Motion	7,333.0	3.5	13,184.5	8.8
Microsoft	6,185.5	3.0	2,759.0	1.8
Bada	2,684.0	1.3	3,111.3	2.1
Symbian	2,569.1	1.2	17,458.4	11.6
Others	713.1	0.3	1,166.5	0.8
Total	**207,662.4**	**100.0**	**150,189.9**	**100.0**

Source: Gartner (February 2013)

Figura 1.3. Ventas de teléfonos móviles según sistema operativo del último cuarto de 2012[4]

A esta gran penetración de mercado hay que sumar el bajo coste de entrada al mercado de desarrollo de aplicaciones para el ecosistema Android. Las librerías y herramientas necesarias para hacerlo son gratuitas y, usualmente, libres.

El coste de registrarse en el programa de desarrolladores de Google es bajo, $25 USD. Esto permite el acceso a la plataforma de publicación de aplicaciones, Google Play Developer Console, donde, además, podemos obtener estadísticas de uso y errores, sugerencias de mejora y recibir las opiniones de los usuarios.

1.2.2. Objetivos

Tanto el hecho de desarrollar una herramienta que pudiese resultar útil como la adquisición de conocimientos y competencias durante el proceso son los objetivos principales.

4 Gartner Says Worldwide Mobile Phone Sales Declined 1.7 Percent in 2012

De hecho, la adquisición de experiencia no sólo sería útil para dispositivos Android específicamente, sino también para dispositivos móviles táctiles en general, en cuanto a los patrones de interacción como el diseño específico que ello conlleva son extrapolables.

Dada la finalidad didáctica del proyecto, se ha optado por trabajar directamente con las librerías nativas y la interfaz de programación de aplicaciones (API por sus siglas en inglés) oficiales del sistema, evitando usar herramientas que simplifiquen el desarrollo a costa de abstraer o ofuscar la arquitectura subyacente.

La fase de desarrollo llevará consigo la adquisición de experiencia con diversas herramientas en las que destacan el entorno de desarrollo integrado escogido, el sistema de versionado de código, máquinas virtuales para ejecutar el código en Android, además de otras no previstas.

Otro objetivo relevante es la experimentación del ciclo de vida de la aplicación, desde la fase de análisis, desarrollo y publicación como, finalmente, la recepción de opiniones y comentarios por parte de los usuarios.

2. Mercado y competencia

El mercado de dispositivos Android es muy diverso y variado en cuanto a nacionalidades e idiomas, por lo que me centraré en dos de los idiomas más usados (tanto por hablantes nativos como por quienes lo emplean como lenguaje auxiliar), inglés (en su vertiente americana) y castellano (vertiente de España).

La utilidad de herramientas del tipo de la que es objeto de este proyecto lleva a la coexistencia previa con muchas otras aplicaciones con funcionalidades similares en la plataforma de distribución oficial, Google Play, entre las que destacaré las siguientes:

2.1. Diario de Dolor de Cabeza (Headache Diary)

Una aplicación muy sencilla gráficamente pero con muchas funcionalidades, en inglés y español.

No se trata de una aplicación nativa, es decir, no trabaja directamente con el sistema, sino que utiliza una capa de abstracción, en este caso un navegador web, al tratarse de una aplicación desarrollada usando tecnologías mucho más sencillas que las nativas del sistema.

Este tipo de tecnologías pensadas para ahorrar tiempo de desarrollo, conllevan un gasto innecesario de recursos en el dispositivo, y por tanto de batería (algo crítico en dispositivos móviles) así como un retraso perceptible entre las diferentes pantallas de la aplicación, causada a veces por la red y siempre por la capa extra que añade tener que renderizar el lenguaje de marcado (HTML) y el de estilo en cascada (CSS) en vez de dibujar directamente con las librerías del sistema, mucho más optimizadas.

Permite exportar los datos a una hoja de cálculo, para facilitar el posterior análisis de los datos.

Tiene una puntuación de 4 de 5 puntos máximos y, según los datos oficiales de Google Play, ha sido descargado entre mil y cinco mil veces.

El instalador ocupa 365 KB, un buen tamaño dado los límites a los que a día de hoy las operadoras móviles someten a sus usuarios en el caso de las tarifas planas, y aceptable en el caso de los clientes con facturación por ancho de banda consumido.

Se trata además de una aplicación gratuita y sin publicidad.

play.google.com/store/apps/details?id=de.gerfeldersoftware.headachediary

2.2. Headache Diary

Aplicación de diseño elaborado y con muchas funcionalidades. Es aparentemente poco usable, y sólo está disponible en inglés, lo que es una barrera de entrada para un alto porcentaje de usuarios.

Los usuarios la han puntuado con 4.1 de 5 puntos máximos y ha sido descargada entre cien mil y quinientas mil veces.

Es una aplicación de tamaño a considerar, 2,3 MB, lo que implica problemas con las tarifas de datos más restrictivas a la hora de descargarlo e instalarlo.
Posee una versión gratuita con anuncios y funcionalidad recortada y otra de pago.

play.google.com/store/apps/details?id=com.delayr.diary.lite

2.3. migraineDiary

Aplicación muy sencilla con pocas opciones.

Los usuarios le han dado una puntuación media de 2.8 de 5 puntos máximos, y ha sido descargada entre mil y cinco mil veces.

Su tamaño es de 210 KB. Se trata de una aplicación gratuita.

play.google.com/store/apps/details?id=org.zerodogg.migraineDiary

3. Planificación

Debido a motivos laborales y personales, se dispone de poco tiempo de planificación, aproximadamente 1 mes, durante el cual se definirán los requisitos, características y funcionalidad. Se analizarán además los problemas y soluciones de diseño que requiere la alta fragmentación entre versiones de Android en el mercado, cada una con librerías y aspecto diferente, que además provoca que el diseño de las pantallas deba ser adaptado a varios casos.

Se ha decidido emplear metodologías de desarrollo ágil, esto es desarrollo incremental, flexible, y enfocando el patrón de análisis-implementación-pruebas a tareas en vez de al proyecto en su totalidad.

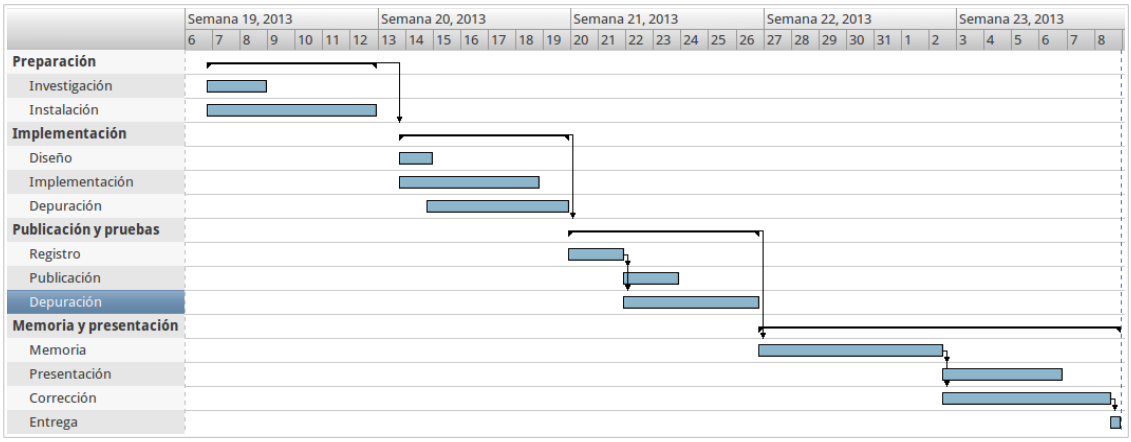

Figura 3.1. Diagrama de Gantt

La fase de desarrollo será dividido en 4 iteraciones principales:

- **Preparación, 6 a 8 de mayo**
  Durante este período tendrá lugar la investigación inicial y la configuración de las herramientas para desarrollar, incluyendo la preparación y configuración de máquinas virtuales, búsqueda de documentación y ejemplos para las librerías empleadas, definición de la estructura básica del proyecto y la organización de módulos inicial (que normalmente es modificada durante el transcurso).

- **Implementación, 9 a 25 de mayo**
  Durante esta fase se implementan las ventanas de la aplicación y se les dota de funcionalidad, se diseña y conecta la aplicación a la base de datos que almacenará la aplicación, y se comprueba en diferentes máquinas virtuales y dispositivos con diferentes versiones de Android que el comportamiento es el esperado.

- **Publicación y pruebas, 26 de mayo**
  Se ejecutará la búsqueda y correcciones de errores final, y me daré de alta como desarrolladora en la plataforma de Google, se generará una firma criptográfica única, se empaquetará la aplicación, firmará el instalador y se subirá a la plataforma de distribución de contenido oficial para Android, Google Play, en la que se creará y completará un perfil para la aplicación y, finalmente, la aplicación pasará a estar disponible fácilmente. Tras esto, se verá la respuesta y comentarios por parte de los usuarios y, en el caso de que algún error sea notificado será corregido.

- **Memoria y presentación, 27 de mayo a 8 de junio**
  Se completará la memoria, se redactarán los apartados que falten y se creará, a partir de ella, una presentación acerca del proyecto. Se empleará un margen para revisión, que será empleado para aplicar los cambios en la memoria y en la aplicación sugeridos por coordinador de proyecto.

3.1. Plan de contingencia

A continuación, se describen la planificación para aquellas situaciones que puedan modificar el plan previsto.

3.1.1. Retrasos e imprevistos

El calendario planificado da poco margen para solucionar imprevistos. Las fechas estipuladas deberían cumplirse en aras de permitir la realización de todas las tareas programadas.

Como consecuencia, la aparición de cualquier imprevisto que pueda provocar un retraso en las fechas, provocará que deba solucionarse lo antes posible, sucediendo esto fuera del horario establecido inicialmente.

3.1.2. Pérdida de información

Para evitar las consecuencias de cualquier eventualidad que conlleve pérdida de datos de prueba, código, o recursos esenciales, se emplearán diferentes soluciones buscando la máxima infalibilidad posible con un coste de recursos mínimo, tanto en lo económico como de tiempo.

El proyecto será respaldado en un servicio en línea de forma automática. Dicho servicio poseerá características de seguridad y será lo más fiable posible, con sus propios sistemas de respaldo.

El código del programa desarrollado será publicado, conforme a su desarrollo en un servidor remoto de control de versiones que proveerá, además de respaldo, la posibilidad de revertir cambios en el código fuente.

3.2. Recursos

A continuación se detallan los recursos utilizados hasta el momento en el desarrollo del proyecto.

3.2.1. Equipamiento

Samsung NP-R530, un portátil con pantalla de 16 pulgadas, será el equipo en el que se desarrollará la aplicación. Su tamaño y teclado completo, permitirá ser usado intensivamente en la fase de desarrollo.

Posee 4 GB de RAM, que serán consumidos mayoritariamente por las máquinas virtuales y el entorno de desarrollo, y en muy menor medida por el sistema operativo.

Su procesador, Intel Core i3-330M, de cuatro núcleos a 2,13 GHz, es suficientemente potente para ejecutar fluidamente el pesado entorno de desarrollo escogido, además de poseer extensiones de virtualización que permiten una ejecución más fluida de las máquinas virtuales con Android sobre las que se ejecutará y depurará la aplicación.

ZTE Blade, un teléfono móvil que permitirá probar la aplicación en un entorno real. Es un dispositivo con tres botones físicos, pantalla pequeña, poca densidad de píxeles y Android 2.3.3.

Motorola Razr XT910, un teléfono móvil que permitirá probar la aplicación en un entorno real. Es un dispositivo de cuatro botones táctiles, de pantalla grande y media densidad de píxeles, con Android 4.1.2.

3.2.2. Software

Ubuntu 12.10 de 64 bits, es el sistema operativo escogido. Su estilo UNIX permite ahorrar tiempo a la hora de configurar el entorno de desarrollo de Java y su núcleo Linux permiten ahorrarse el tiempo en búsqueda de drivers para los dispositivos Android a conectar. Además, las herramientas incluidas tales como grep y awk así como los entornos de scripting como bash y python permitirían, si fuese necesario, automatizar pequeñas tareas repetitivas.

- Posee un sistema de paquetes y repositorios, como otras distribuciones de Linux, que permite instalar una gran cantidad de software de forma muy rápida y sencilla, y que será usado para instalar el resto de software.
- Los entornos de escritorio escogidos son Gnome Shell y Xfce, ambos muy completos y con utilidades como herramientas de renombrado masivo, calculadora, emulador de terminal, escritorios múltiples, etcétera.

LibreOffice 3.6.2.2 como solución ofimática, usado para generar documentos de texto y diapositivas que serán exportadas a PDF para entregar. Se trata de un software libre y gratuito, respaldado y desarrollado por una gran comunidad. Está a su vez basado en OpenOffice, desarrollado por Sun Microsystems y posteriormente cedido por Oracle a la fundación sin ánimo de lucro Apache.

Google Chrome y Opera serán los navegadores para leer documentación, investigar, y emplear las herramientas web auxiliares durante la duración del proyecto.

- Google Chrome es un navegador desarrollado por Google basado en el motor webkit, muy fiel a los estándares web definidos por el WorldWideWeb Consortium, y con el motor de Javascript mas rápido, lo que permitirá ejecutar ágilmente aplicaciones web complejas.
- Opera es un navegador desarrollado por Opera Inc., que incluye Opera Turbo, una tecnología que permite mejorar enormemente la velocidad de navegación en redes lentas, además de ahorrar ancho de banda, útil cuando se accede a través de tarifas de datos con límite.

Dropbox, un servicio web de almacenamiento remoto y permanente accesible, provee además un cliente que se integra con el escritorio lo que permite mantener copias de seguridad periódicas de forma sencilla.

Facilethings, un servicio web para planificar tareas personales basado en la metodología GTD.

Git, un sistema de versionado de código basado en sincronización entre repositorios, libre y desarrollado por una comunidad, que permite operar en un repositorio local y sincronizarlo con otro remoto.

Git-Hub, servicio de hospedaje web para código y servidor para Git, para proyectos libres que permite que los usuarios puedan seguir el desarrollo de un proyecto, marcar incidencias, crear proyectos derivados y proponer cambios en el código

fuente activamente.

Eclipse 3.6, entorno de desarrollo integrado compatible con módulos específicos para el desarrollo en Android, accesos directos a las máquinas virtuales, depuración remota, utilidades para Git, coloreado y auto-completado de sintaxis y ayuda integrada. Se trata de un software desarrollado en su momento por Sun Microsystems y propiedad actualmente de Oracle, los mismos propietarios de Java, lo que asegura que se mantiene actualizado y posee un soporte excepcional para Java.

- **Android Development Tools,** plugin para Eclipse que proporciona un entorno integrado para el desarrollo de aplicaciones para Android.
- **EGit,** es un proveedor de Eclipse Team para el sistema de control de versiones Git.
- **GreenUML,** plugin para Eclipse que permite crear diagramas UML a partir del código fuente.

Android SDK, provee las librerías y herramientas necesarias para el desarrollo de aplicaciones para Android.

Android SDK Manager, permite gestionar la descarga de herramientas, plataformas, etc para el desarrollo con el Android SDK.

Android Virtual Device Manager, proporciona la gestión de dispositivos virtuales para probar las aplicaciones de Android desarrolladas.

Geany, entorno de desarrollo integrado, minimalista, sencillo y ligero, con soporte y coloreado de sintaxis para muchos lenguajes, y un terminal 'bash' integrado, ideal para crear 'scripts' y depurarlos, y para usarlo como simple editor de texto plano.

Planner, programa de gestión de proyectos útil para generar el diagrama de Gant.

Inkscape, programa de dibujo vectorial, libre, que permite crear gráficos escalables y exportarlos como imágenes de mapa de bits comprimidas.

Gimp, programa de edición de imagen, libre.

Dia, programa de edición de diagramas de propósito general.

4. Análisis

4.1. Especificación de requisitos y diseño de interfaz

La aplicación se divide en tres partes principales: Inserción de información, una vista para visualizar los registros y un apartado de configuración.

Además de esto se ha de tener en cuenta las diferencias de interfaz entre distintas versiones de Android.

4.1.1. Inserción de información

Dado que la finalidad de la aplicación radica en el registro de los episodios de cefalea, se hará especial hincapié en esta sección, que permitirá indicar la duración, intensidad, e información relevante y observaciones de la forma más sintetizada posible, para que pueda ser fácilmente usado durante un episodio de cefalea minimizando la frustración del usuario.

Recapitulando:

- Las entradas del diario exigirán al usuario tan sólo un mínimo de información, dando la posibilidad de añadir o completar el registro posteriormente.
- La interfaz será clara y concisa, minimizando la cantidad de texto y agrupando los controles de entrada en secciones bien definidas con un título descriptivo.
- Se emplearán los controles mas intuitivos para cada tipo de dato, esto es, selectores de día, mes, año, hora y minuto para ubicar el episodio en un marco temporal; un selector desplegable para categorizar el episodio, un campo numérico para elegir la intensidad y, finalmente, un cuadro de texto par añadir, opcionalmente, información extra.

Como se ha mencionado anteriormente, completar un registro durante un episodio se trata de una tarea compleja, por lo que se debe permitir completar la información posteriormente, esto es, editar el registro, en cualquier momento, mientras lo visualiza.

Por otro lado, también se tendrá como requisito poder eliminar los registros cuando el usuario lo estime oportuno.

4.1.2. Visualización de registros

La vista principal contendrá una lista cronológica de los registros según la fecha y hora de la cefalea introducidas por el usuario. Esta lista representará información relevante tal como la fecha, el tipo y la intensidad, de modo que cada registro resulte fácilmente identificable, de forma clara y concisa.

Para la mostrar la intensidad en el registro, se ha decidido representar el mismo con un código de color (amarillo, naranja y rojo), mediante círculos, aprovechando la metáfora visual ideada a la hora de crear el icono.

Para que el flujo en la aplicación resulte intuitivo, en la misma vista principal de ubicará un botón para añadir registro, con un icono autodescriptivo, además de permitir visualizar y editar cada registro de la lista pulsando sobre él, siendo este el paradigma más usado en vistas de datos de este tipo en toda clase de dispositivos táctiles.

4.1.3. Configuración

Puesto que es imposible tener en cuenta todas las categorías, en este caso para los tipos de dolor, que cualquier usuario pueda llegar a desear, se ha decidido hacer configurable este aspecto, permitiendo al usuario crear tantos tipos como desee, y no restringiendo de ninguna manera la forma de categorizar los episodios.

Para facilitar la entrada de texto, se permite crear nuevas categorías desde la misma vista de creación de registro, minimizando los pasos necesarios y agilizando el uso de la aplicación.

Es por esto último por lo que se ha decidido ocultar la vista 'settings', ya que carece de utilidad real, aunque podrá activarse posteriormente en cuanto se considere que pueda albergar alguna utilidad.

4.1.4. Compatibilidad

Dado las diferentes versiones del conjunto de librerías para las diferentes versiones de Android, y dado que se busca la compatibilidad con el mayor número de dispositivos posible, deberá tenerse en cuenta las diferencias entre los controles nativos de interfaz de usuario existente entre dichas versiones.

Es por ello por lo que se deberá implementar vistas parciales específicas, fragmentos según la nomenclatura de Android, para incluir controles alternativos para diferentes versiones de la interfaz de programación de aplicaciones (API por

sus siglas en inglés), permitiendo así un aspecto nativo y la mejor experiencia de usuario posible en cada caso.

4.2. Estructura de base de datos

La aplicación, por su sencillo diseño, no requiere de una base de datos compleja para albergar tanto los registro como los tipos de cefalea.

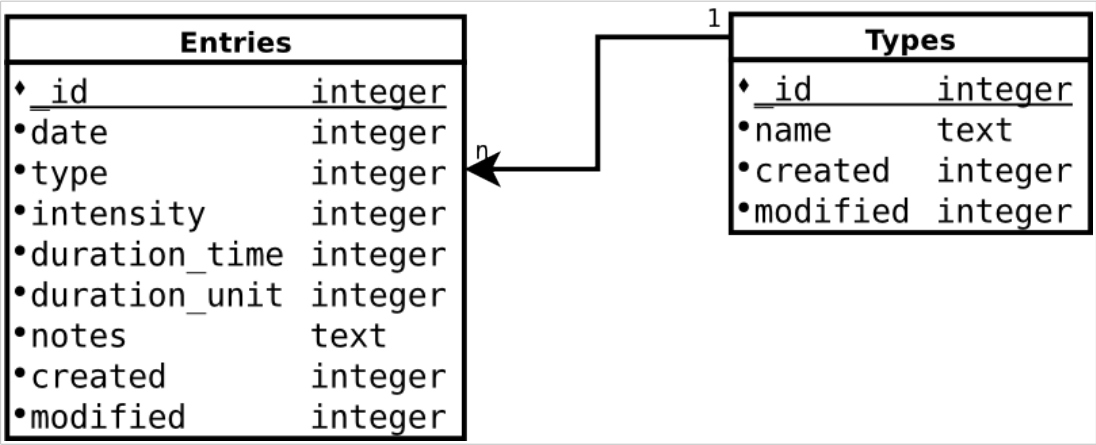

Figura 4.1. Esquema de base de datos

La tabla de registros, denominada 'Entries', constará de:

- **_id**, un identificador interno único
- **date**, la fecha y hora del episodio en formato UNIX
- **type** hará referencia a un '_id' de la tabla 'Types' y definirá el tipo de registro. Se trata, pues, de una clave foránea.
- **intensity**, un valor acotado del 0 al 9 que hará referencia a la intensidad del dolor.
- **duration_time**, un valor numérico usado en conjunción con 'duration_unit' para especificar la duración del episodio.
- **duration_unit**, un identificador interno que hará referencia a la unidad del valor especificado en 'duration_time'.
- **notes**, texto de ancho variable que incluirá los apuntes que el usario desee.
- **created**, fecha y hora en formato UNIX de cuando fue creado el registro.
- **modified**, fecha y hora en formato UNIX de la última modificación del registro.

Por otro lado, la tabla 'Types' hará referencia a los tipos de episodio definidos por el usuario, definiendo los siguientes campos:

- **_id**, identificador único del tipo de dolor, que será referenciado por el campo 'type' de la tabla 'Entries'.
- **name**, texto escogido por el usuario para esta categoría.
- **created**, fecha y hora en formato UNIX de cuando fue creado el registro.
- **modified**, fecha y hora en formato UNIX de la última modificación del registro.

4.3. Estructura de clases

La estructura de clases está enfocada en la existencia de dos vistas principales, una para poder visualizar una lista de registros y la otra para crearlos, visualizarlos y editarlos.

Además de las citadas vistas principales, se ha de tener en cuenta que la interacción con la base de datos local donde se guardarán los registros requerirá a su vez una serie de clases específicas, entre ellos el definido como 'proveedor de base de datos', además de otras clases auxiliares que abstraerán ciertos tipos de datos para simplificar su manipulación.

4.3.1. Modo lista

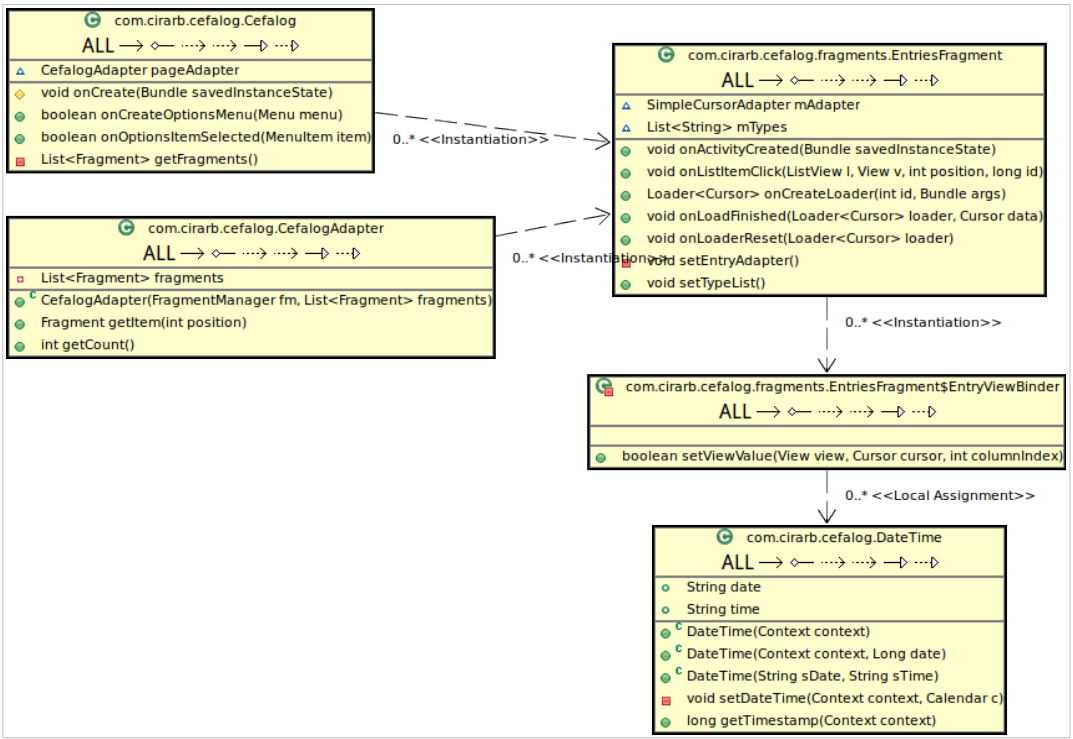

Figura 4.2. Clase Cefalog y clases relacionadas

Cefalog es la actividad principal del proyecto, y será mostrada al abrir la aplicación. En ella se muestra un menú y una lista de pantallas empleando 'fragmentos', es decir, pantallas secundarias.

La lista de pantallas incluye un único elemento, **EntriesFragment**, que muestra la lista de entradas registradas de forma cronológica.

DateTime es una clase auxiliar, empleada para mostrar los valores de la base de datos y recogerlo de los controles de entrada de usuario, tendiendo en cuenta el formato de fecha y hora del sistema, así como la zona horaria.

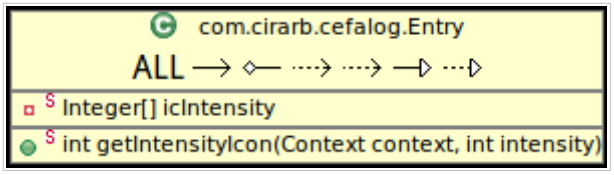

Figura 4.3. Clase Entry

Entry es una clase auxiliar útil para trabajar con los datos conflictivos de los registros, usada para convertir el valor numérico que representa la intensidad del dolor en un icono de color acorde,

4.3.2. Modo registro

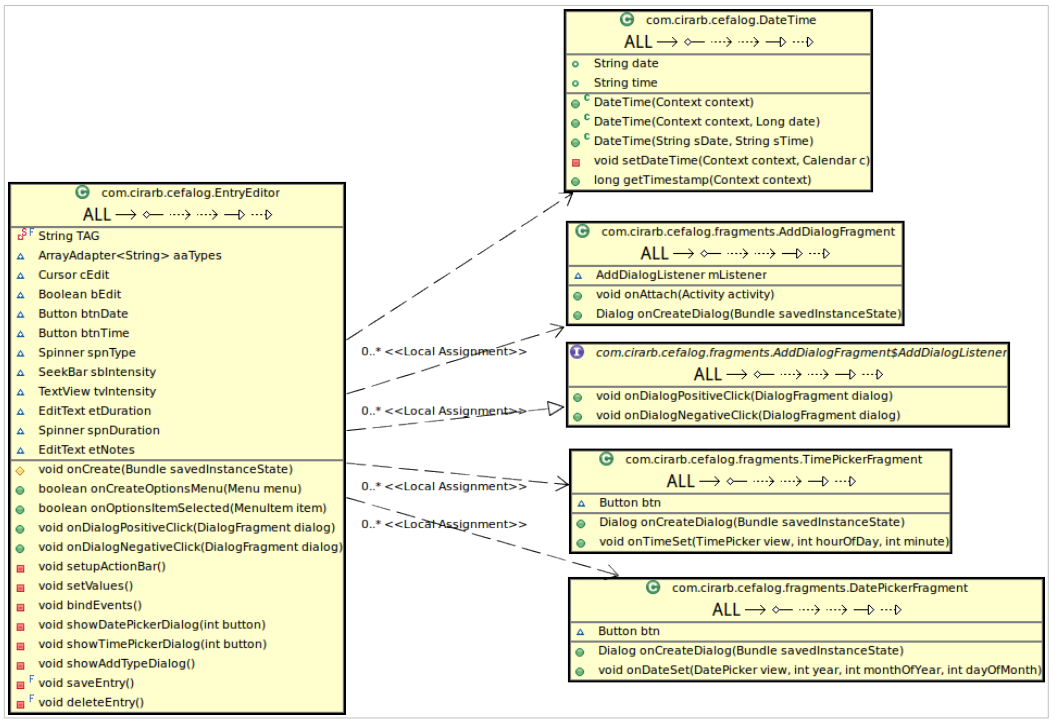

Figura 4.4. Clase EntryEditor y clases relacionadas

EntryEditor representa la vista de un registro, en ella se permite la creación, visualización y modificación de los mismos, dependiendo de cómo sea invocado.

Posee un menú que mostrará las acciones adecuadas dependiendo del estado, que puede ser de creación o edición.

Para facilitar la interacción con ciertos controles, EntryEditor se vale de una serie de diálogos: AddDialog, TimerPickerFragment y DatePickerFragment.

AddDialog se trata de un sencillo diálogo de introducción de texto que será empleado para añadir nuevos tipos de dolor directamente desde el selector, simplificando el proceso de creación de los elementos de la tabla 'Types'.

TimePickerFragment es un diálogo que permite seleccionar una hora y minuto, siendo su diseño e implementación dependiente de la versión de Android.

DatePickerFragment es un diálogo que permite seleccionar una fecha, siendo su diseño e implementación dependiente de la versión de Android.

4.3.3. Proveedor de base de datos

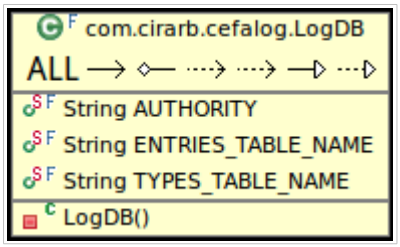

Figura 4.5. Clase LogDB

LogDB proporciona la definición de las dos tablas con sus atributos comunes.

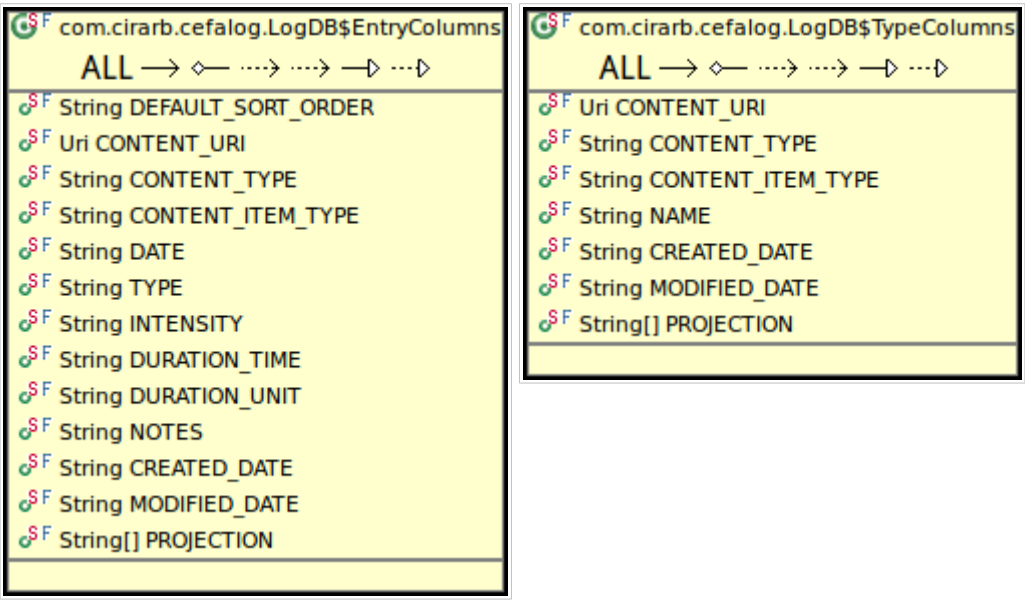

Figura 4.4. Clases EntryColumns y TypeColumns

EntryColumns y **TypeColumns** se encuentran en LogDB implementando los atributos específicos para cada una de las tablas a las que hacen referencia.

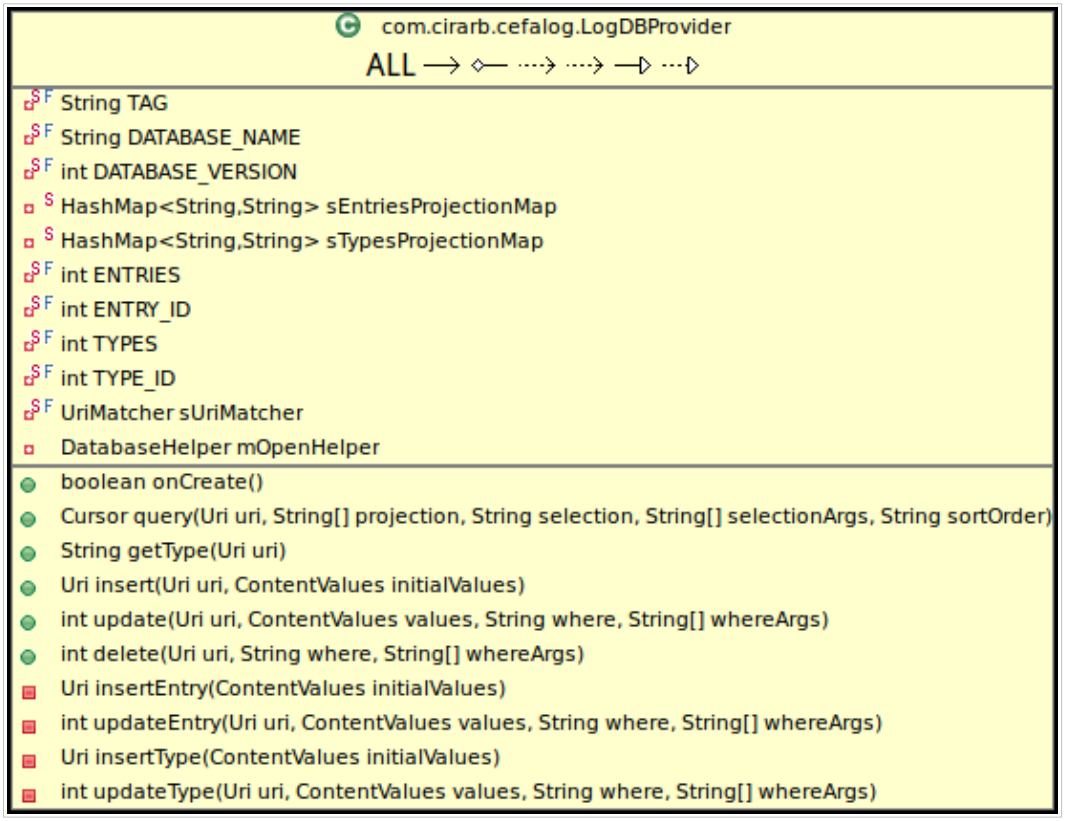

Figura 4.5. Clase LogDBProvider

LogDBProvider, provee a la aplicación de una capa que provee de comunicación con la base de datos y permite realizar en ella las operaciones requeridas de inserción, actualización y borrado en las diferentes tablas.

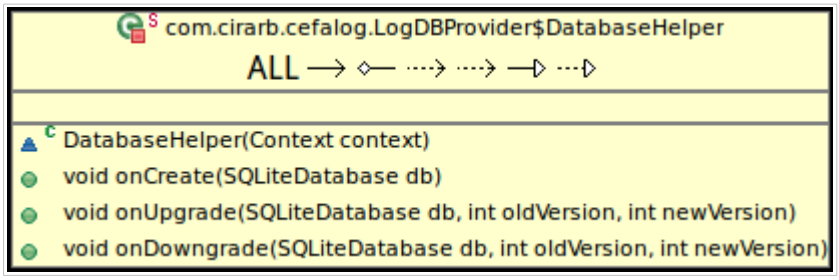

Figura 4.6. Clase DatabaseHelper

DataBaseHelper, implementada dentro de LogDBProvider, se encarga de crear la base de datos o actualizarla si hace falta.

5. Diario de desarrollo

5.1. Primera iteración: preparación, 6 a 8 de mayo

5.1.1. Creación de la estructura de la memoria

Se ha creado un formato preliminar y un guión para poder ir completando la memoria conforme vaya avanzando el proyecto, así como poder dedicar más tiempo a estructurarla y sintetizarla mejor, facilitando así la última iteración y pudiendo dedicar mas tiempo a la segunda.

5.1.2. Preparación del entorno

Usando el gestor de paquetes del sistema, se ha de instalar el kit de desarrollo para la plataforma Java 1.6, Eclipse 3.6

Una vez se ha especificado al sistema la versión de Java a utilizar por defecto, se puede empezar a configurar Eclipse e instalar la extensiones de cliente Git y de utilidades de desarrollo para Android (ADT por sus siglas en inglés).

Usando ADT se puede descarga las librerías para Android, y crear las máquinas virtuales.

Se crea un proyecto de Eclipse, inicialmente para los niveles de API de 11 a 17 (correspondientes a Android 3 a 4.2), y se crea la estructura básica.

Se elije como nombre Cefalog, tras buscar otros nombres alternativos en motores de búsqueda tales como Google, Google Play, GitHub, etcétera.

Se crea un repositorio en Git Hub, y se realiza la primera sincronización de código fuente para comprobar su correcto funcionamiento y estableciendo un punto de partida.

5.2. Segunda iteración: desarrollo, 9 a 25 de mayo

5.2.1. Creación de la estructura inicial

Se diseña el prototipo de las vistas y se implementa en Eclipse:

- Principal, con pestañas deslizantes para intercambiar entre las diferentes vistas y un menú superior con acceso al resto de actividades.
  Esta ventana albergará la lista de registros.

- Settings, una vista de configuración.
- NewEntry, una vista a modo de formulario con un menú superior con un botón de guardar. Posee varios campos, con una etiqueta muy simple y descriptiva, o con un texto explicativo en su interior.
    - Control de selección de fecha y control de selección de hora.
    - Selector de lista para seleccionar el tipo de dolor
    - Un control desplazable para seleccionar la intensidad de dolor.
    - Un campo para escribir la duración y un selector de lista para elegir unidad de tiempo.
    - Un cuadro de texto multilínea para añadir comentarios.

Para facilitar la entrada de información en NewEntry en los controles dedicados a la fecha y hora, se implementan dos cuadros de diálogo: DatePicker y TimePicker. Ambos requieren trabajo extra para asegurar la compatibilidad en distintas versiones de Android.

Una vez llegada a este punto, he aplicado modificaciones para permitir que el proyecto fuese compatible con los niveles de API de 8 a 17 (de Android 2.2 a 4.2), teniendo que reimplementar algunos controles.

También se diseña el icono de la aplicación, un cerebro de alto contraste visto cenitalmente con áreas de dolor marcadas con aros de colores y falso texto.

Figura 5.1. Icono de Cefalog

La metáfora visual resulta efectiva, y el contraste permite que sea fácilmente identificable en los diferentes tamaños requeridos para las diferentes densidades de píxel y resoluciones que deberá soportar la aplicación.

5.2.2. Funcionamiento de los cuadros de diálogo

El diseño de interacción escogido establece que al pulsar en algún control, que muestra el valor previo, se abre el cuadro de diálogo con el valor previo permitiendo modificaciones, un botón para aceptar y otro para cancelar, además

de poder salir haciendo click fuera del citado diálogo.

Dadas las diferencias entre las diferentes versiones de la librería de Android, se han empleado 'fragmentos', esto es, diferentes implementaciones de la misma funcionalidad dependientes de la versión, para asegurar la retrocompatibilidad.

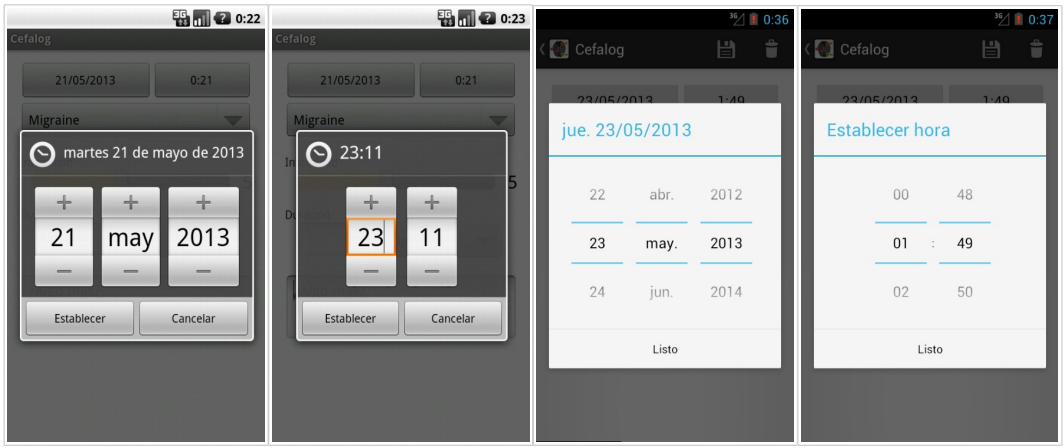

Figura 5.2. Diferencias de los cuadros de introducción de fecha y hora entre Android 2.2 y 4.2

5.2.3. Almacenamiento de registros

Se usará SQLite, una base de datos SQL sin servidor, y con soporte oficial, para almacenar localmente los registros introducidos en la aplicación.

Android dispone de un sistema para abstraer la obtención y guardado de datos, lo que podría permitir cambiar la base de datos posteriormente. Este sistema llamado *ContentProvider* es usado para implementar los siguientes objetos:

- LogDB: encargado de definir la conexión a la base de datos.
- LogDBProvider: el objeto encargado de definir la estructura de la tabla de datos así como de obtener información de los registros, actualizarlos y eliminarlos.

5.2.4. Visualización de datos

Se implementa el fragmento de la lista de registros de la ventana principal recogiendo los datos del LogDBProvider, tras varias pruebas se escoge el diseño de una imagen representando la intensidad a fin de ahorrar espacio, la fecha y el tipo de dolor.

Para mostrar el nivel de dolor mediante un icono, se aprovecha la metáfora en el icono de la aplicación, definiendo aros de tres colores: amarillo, naranja y rojo,

para dolor leve, medio y fuerte respectivamente

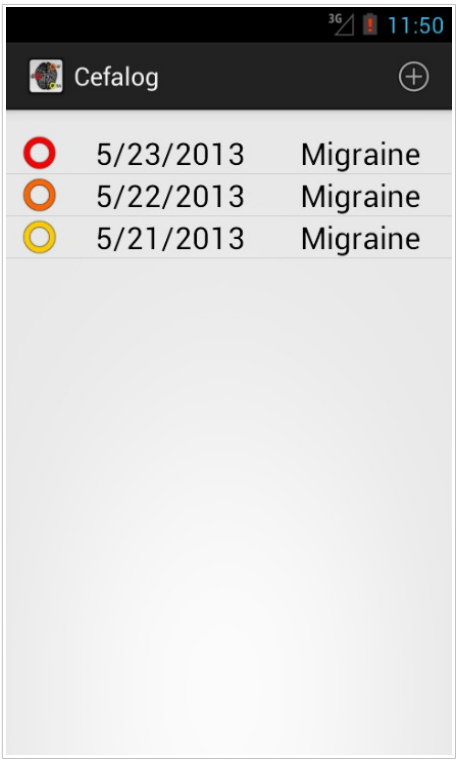

Figura 5.3. Vista principal con los registros guardados

5.2.5. Mejora de interacción en la vista para crear registros

Llegados a este punto, y teniendo la funcionalidad básica, he decidido hacer circular una versión de pruebas a un reducido grupo, proceso tras el cual se cambiaron varios aspectos.

- El control para definir intensidad pasa a ser un control de tipo 'seekbar' (una barra desplazable) con valores discretos del 0 al 9 mostrados con un texto al lado como valores del 1 al 9, facilitando su uso.
- Se eliminan los tipos en el cuadro de selección del tipo de dolor y se añade una opción que, si es seleccionada, permita añadir nuevos elementos al mismo, permitiendo al usuario definir los tipos de dolor que desee.
    - La inserción de nuevos tipos de dolor se realiza mediante un sencillo cuadro de diálogo de introducción de texto, con botones para aceptar y cancelar.

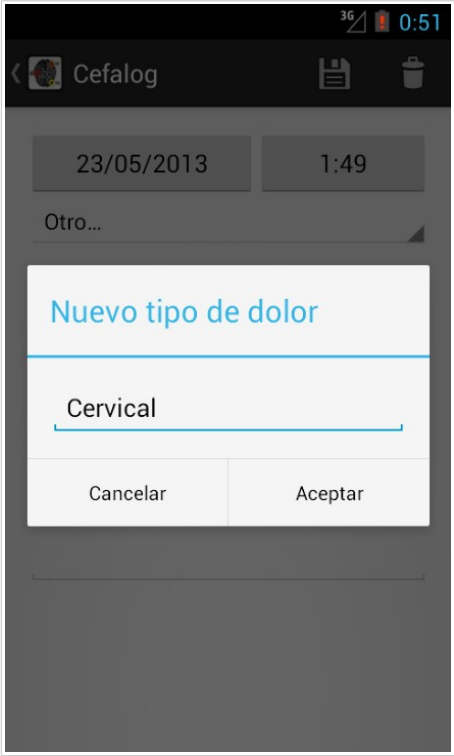

Figura 5.4. Cuadro de diálogo para personalizar el tipo de dolor

5.2.6. Edición y eliminación de registros

Se han terminado de implementar las funcionalidades de gestión de registros.

Para visualizar y editar los datos de un registro, se aprovecha la misma vista de creación, a la que se accede pulsando sobre el elemento de la lista de la vista principal.

Además, se ha añadido en la pantalla de edición un botón que permite eliminar el registro visualizado.

5.2.7. Nota de compatibilidad

Hay que destacar que, debido a carencias en versiones de Android anteriores a la 3 (nombre en clave Honeycomb), he tenido que ubicar en el menú de la aplicación las opciones de guardar, borrar y retroceder, en vez de encontrarse en la barra de título, para asegurar la compatibilidad con versiones antiguas. Ver figura 5.5.

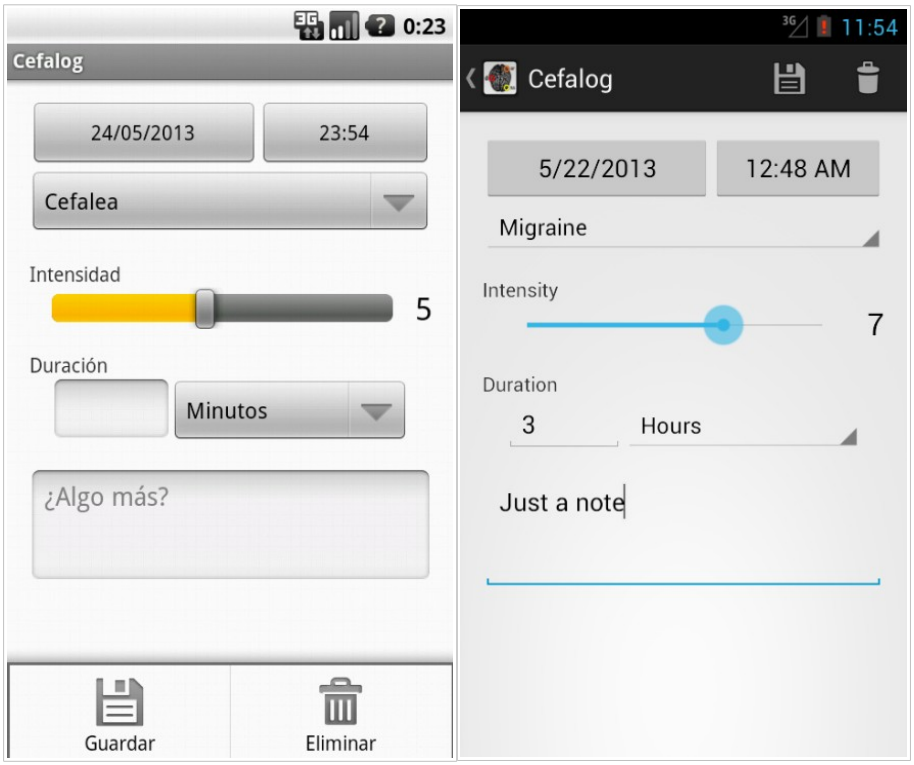

Figura 5.5. Diferencias de la interfaz entre Android 2.3 y 4.2

5.3. Tercera iteración: publicación y pruebas, 26 de mayo

El registro en la plataforma de desarrolladores de Google Play, ha sido mas rápido de lo esperado, se ha creado una clave criptográfica privada y otra pública. Dicha clave criptográfica es utilizada para asegurar al usuario, y a Google, que la aplicación proviene siempre del desarrollador conocido y evitar así problemas de seguridad a la hora de instalar o actualizar.

Se ha hecho el pago de la cuota, y se ha rellenado la página del programa, con capturas de pantalla, el icono, y demás.

Antes de subir el ejecutable, se ha corregido un pequeño fallo a la hora de salir del cuadro de diálogo de añadir tipo de dolor, se ha compilado y firmado el ejecutable con la clave privada, además de subir la clave pública a los servidores de Google.

Figura 5.6. A día de 9 de Junio de 2012, no se ha registrado ningún error en la aplicación.

5.4. Cuarta iteración: memoria y presentación, 27 de mayo a 9 de junio

Se ha reestructurado ligeramente el documento creado durante la primera fase y completado en las siguientes, se ha redactado la introducción y revisado las fechas de las diferentes iteraciones. Se han corregido anomalías en el formato del documento.

6. Glosario

Awk. Herramienta de procesamiento automatizada de texto.

Bash. Shell o interfaz de acceso a los servicios del kernel, para entornos Unix.

Cefalea. Dolor de cabeza.

Drivers. Software que comunica el hardware de los periféricos con el sistema operativo.

Grep. Herramienta para seleccionar líneas de ficheros de texto usando patrones de coincidencia.

GTD. *Get Things Done*, es una metodología de organización personal basada en cinco fases: recopilar, procesar, organizar, revisar, hacer.

Intracraneal. Zona comprendida en el interior de la cavidad craneal.

Pericraneal. Zona comprendida inmediatamente exterior al cráneo.

Repositorio. Directorio centralizado donde se almacena y mantiene paquetes de software.

Script. Pequeña pieza de software pensada para solucionar un problema puntual.

Seekbar. Elemento de interfaz gráfica que permite al usuario seleccionar un valor acotado moviendo un indicador que representa el valor actual sobre una línea que representa el rango de valores.

7. Bibliografía

- Comprender la migraña – Dr. Feliu Titus, Dra. Patricia Pozo. ISBN-9788497354158
- Understanding Open Source and Free Software Licensing – Andrew M. St. Laurent. ISBN-9780596005818
- Gartner: Bring your Own Device (BYOD): Mobility Trends and Securing the Transition. Lawrence Pingree, Research Director – Lawrence Pingree (Gartner, enero de 2013)
- Gartner Says Worldwide Mobile Phone Sales Declined 1.7 Percent in 2012 - Rob van der Meulen, Janessa Rivera (Gartner, 13 de febrero de 2013)
- Penetración telefonía móvil en hogares – Observatorio nacional de las telecomunicaciones y sistemas de la información, Ministerio de industria, energía y turismo, Gobierno de España, sobre la Encuesta sobre equipamiento y uso de Tecnologías de la Información y Comunicación en los hogares, de 3 de octubre de 2012 por el Instituto Nacional de Estadística.

7.1. Referencias para el desarrollo

Toda la información necesaria para la instalación y desarrollo del proyecto he podido encontrarla en el sitio web oficial de Android.

Sin embargo, me gustaría incluir la inestimable ayuda que proporcionan los usuarios de StackOverflow, web de preguntas y respuestas para programadores.