Anexo III – Especificación de Diseño

**Lista de cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Descripción** | **Autor** |
| 1.1 | 10/10/2011 | Esqueleto inicial | Romina Liuzzi |
| 1.2 | 30/05/2012 | Completar documento | Romina Liuzzi |

Tabla de Contenidos

[*Introducción* 6](#_Toc328439596)

[*Ámbito del software* 7](#_Toc328439597)

[*Diseño de datos* 8](#_Toc328439598)

[Diseño de la base de datos 10](#_Toc328439599)

[*Diseño procedimental* 13](#_Toc328439600)

[Diagramas de secuencia 13](#_Toc328439601)

[Registrar Usuario 14](#_Toc328439602)

[Login 15](#_Toc328439603)

[Recuperar Contraseña 16](#_Toc328439604)

[Modificar Cuenta 17](#_Toc328439605)

[Crear Ejercicio (Iniciar, actualizar, acabar) 18](#_Toc328439606)

[Borrar Ejercicio 21](#_Toc328439607)

[Ver listado de estadísticas disponibles 22](#_Toc328439608)

[Consultar estadística 23](#_Toc328439609)

[Ver histórico de ejercicios 24](#_Toc328439610)

[Ver recorrido asociado a un ejercicio y mejores marcas 25](#_Toc328439611)

[Logout 26](#_Toc328439612)

[Cancelar cuenta 26](#_Toc328439613)

[Diseño arquitectónico 27](#_Toc328439614)

[Paquete: com.caloriecalc.beans 29](#_Toc328439615)

[Paquete: com.caloriecalc.content 29](#_Toc328439616)

[Paquete: com.caloriecalc.dao 30](#_Toc328439617)

[Paquete: com.caloriecalc.lao 31](#_Toc328439618)

[Paquete: com.caloriecalc.presentation 31](#_Toc328439619)

[Paquete: com.caloriecalc.presentation.graphs 32](#_Toc328439620)

[Paquete: com.caloriecalc.security 32](#_Toc328439621)

[Paquete: com.caloriecalc.services 33](#_Toc328439622)

[*Diseño de la Interfaz* 35](#_Toc328439623)

[*Entorno tecnológico del sistema* 40](#_Toc328439624)

# *Introducción*

En este documento nos centraremos en todos aquellos aspectos relacionados con la especificación del diseño, entrando en el dominio de la solución del problema: la fase de desarrollo.

Se dará a conocer el ámbito sobre la que actúa la aplicación; los datos con los que interactúa el sistema, la estructura modular, la estructura del programa en relación con la estructura de los datos, el diseño final de la interfaz y los algoritmos más relevantes.

También se mostrará las pruebas de los diferentes módulos que componen el sistema y el soporte tecnológico utilizado. Por último se incluirá un manual para el programador recogiendo todas las directrices necesarias para desarrollar y depurar la aplicación.

# *Ámbito del software*

Esta aplicación está destinada a fines lúdicos, se trata de una aplicación para móvil categorizada dentro del ámbito de la Salud y Bienestar. La aplicación realiza el cálculo del número de calorías quemadas durante la realización de distintos tipos de ejercicios fomentando los hábitos saludables de los usuarios finales. No existe una edad mínima para los usuarios, pero en general la aplicación está destinada para ser usada por adultos jóvenes y deportistas ocasionales que quieran medir su progreso durante la actividad.

# *Diseño de datos*

Tal como se explicaba en el anexo II del presente proyecto, existen unos requisitos de la información que se deben persistir en el sistema. Durante esta fase se estudian las opciones y se justifican las decisiones tomadas a la hora de definir la especificación de diseño que describirá el sistema.

El siguiente diagrama entidad relación, nos presenta la idea básica a considerar a la hora de abordar estas decisiones. Del ejercicio interesa mantener:

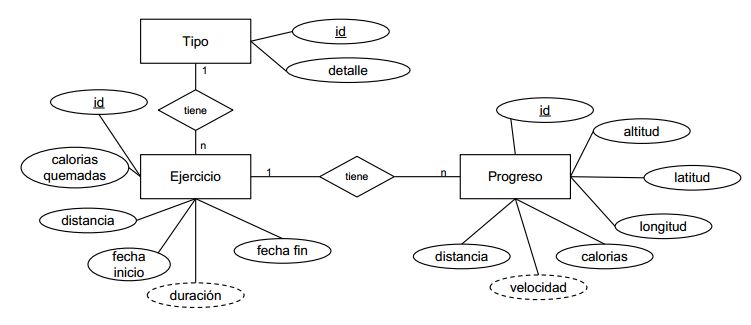


Figura 1. Diagrama entidad relación. Diagrama general

Respecto a estos datos, cabe destacar que la tabla “tipo” se convertirá a un campo simple numérico que identifique un tipo de ejercicio con id en lugar de una tabla separada tal como ilustra la imagen. Los valores aceptados serán validados a nivel código ya que definimos el tipo de ejercicio como un tipo enumerado.

Por otra parte contamos con único usuario del sistema, por lo que si bien, sus datos deben persistirse, no se justifica la creación de una tabla de BBDD para su persistencia. Del usuario interesa guardar:

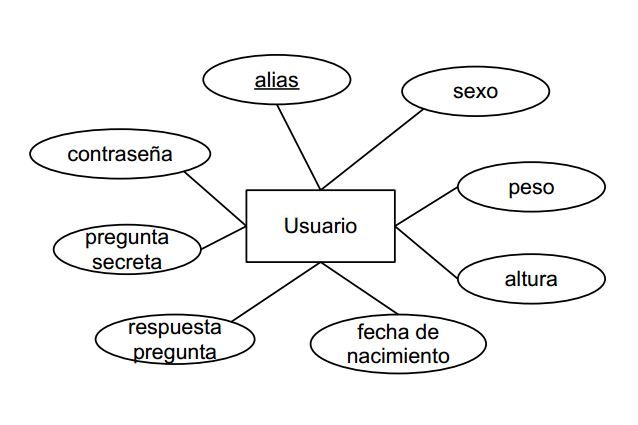


Figura 2. Diagrama entidad relación. Entidad Usuario

Por este motivo se decide guardar los datos de usuario en un fichero de texto fácilmente accesible por la aplicación. El peso del usuario, siendo el único campo sensible a variar a medida que se realicen ejercicios, se registrará en base de datos como un campo más de la tabla ejercicios. De esta manera será posible generar la estadística peso versus fecha.

## Diseño de la base de datos

Con las consideraciones explicadas en el apartado anterior nos queda un esquema de base de datos tal como se ilustra en la siguiente figura.

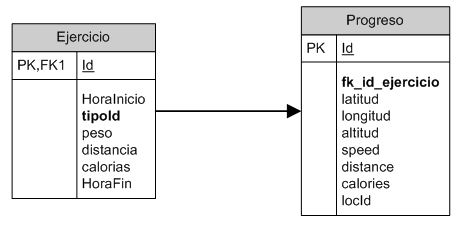


Figura 3. Diagrama de base de datos relacional

La definición de cada una de las tablas se pasa a describir a continuación:

Para la tabla ejercicio:

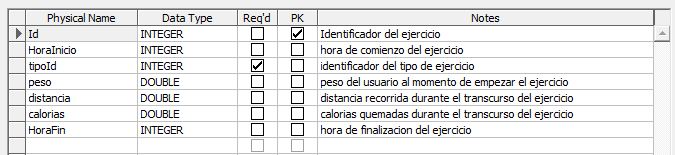


Tabla 1. Definición campos. Tabla ejercicio

Para la tabla del progreso:

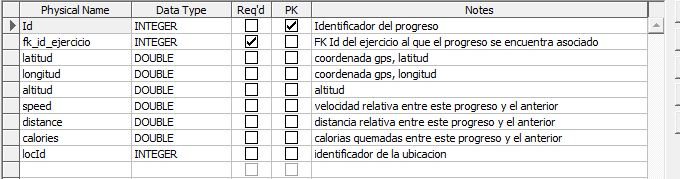


Tabla 2. Definición campos. Tabla progreso

Para el caso del usuario, se persisten los datos en un mapa hash de tipos primitivos, de manera que tenemos una entidad definida por los campos enumerados a continuación.

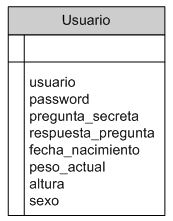


Figura 4. Objeto de tipo Usuario

El objeto se almacenará en formato .xml tal como se muestra en la siguiente ilustración:

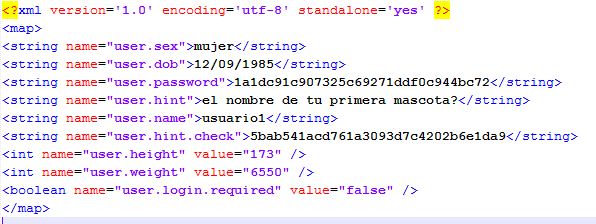


Figura 5. HashMap Usuario

Donde podemos observar que solo se están utilizando tipos primitivos: string, boolean o integer. También cabe destacar que los campos privados se guardan encriptados en MD5.

Para realizar cálculos se realizan una serie de transformaciones sobre estos datos antes de su utilización.

# *Diseño procedimental*

## 

## Diagramas de secuencia

Mediante diagramas de secuencia se mostrarán todos los procedimientos por los que pasa las distintas acciones que se producen en la aplicación. Los diagramas de secuencia se establecen a partir de las plantillas de requisitos funcionales facilitadas en el anexo II del presente documento. En esta sección se recogen los procesos más relevantes de la aplicación.

Registrar Usuario

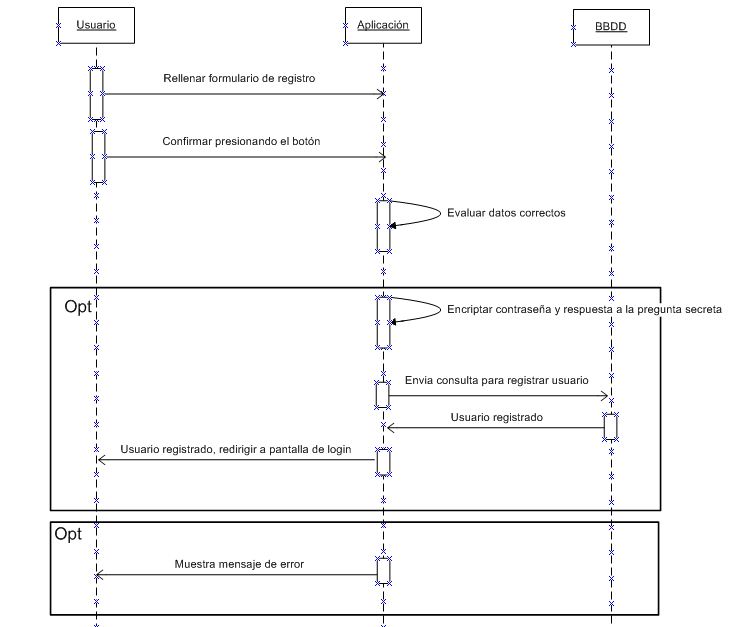


Figura 6. Diagrama de secuencias. Registrar Usuario

Login

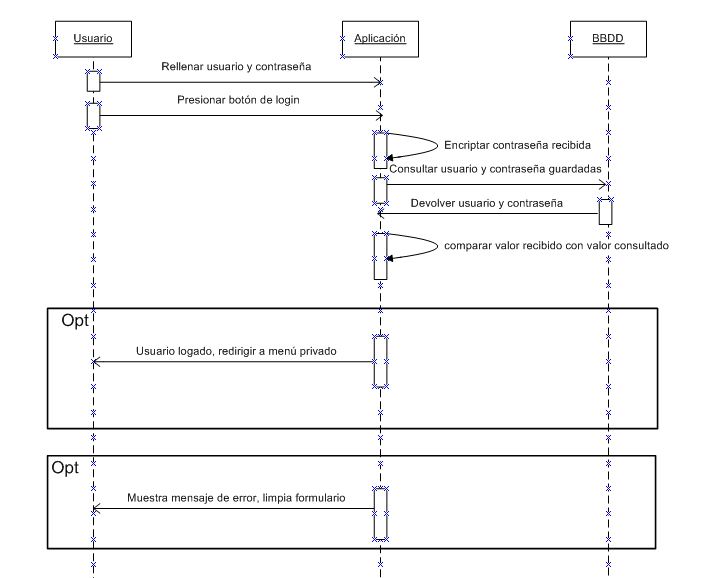


Figura 7. Diagrama de secuencia. Login - login fallido

Recuperar Contraseña

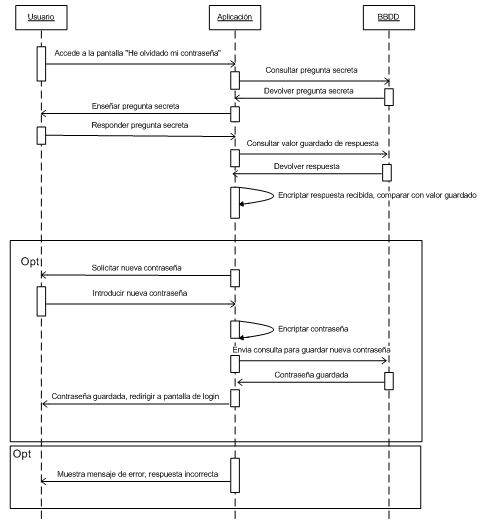


Figura 8. Diagrama de secuencias. Recuperar Contraseña

Modificar Cuenta

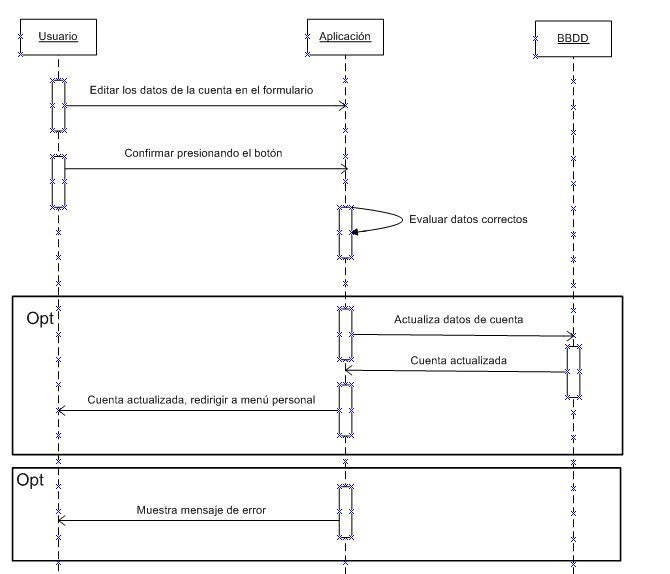


Figura 9. Diagrama de secuencias. Modificar cuenta

Crear Ejercicio (Iniciar, actualizar, acabar)

Este caso se divide en tres procesos:

1. Crear un ejercicio
2. Actualizar ejercicio (cada vez que se recibe una nueva localización)
3. Finalizar ejercicio

El ejerció se crea cuando el usuario selecciona un tipo de actividad y presiona el botón correspondiente al tipo de ejercicio seleccionado. El registro inicialmente se crea completando los campos tipo de ejercicio, fecha de inicio y peso del usuario que lo realiza. El proceso se describe en el diagrama de secuencias “Crear Ejercicio”.

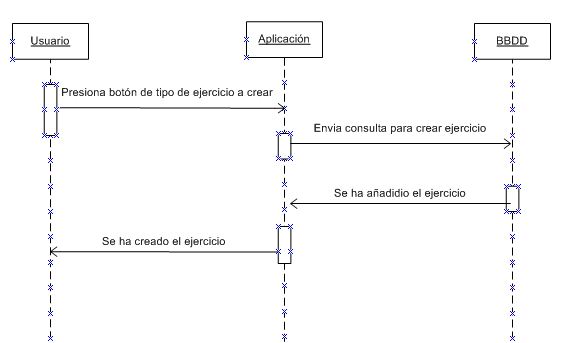


Figura 10. Diagrama de secuencias. Crear Ejercicio

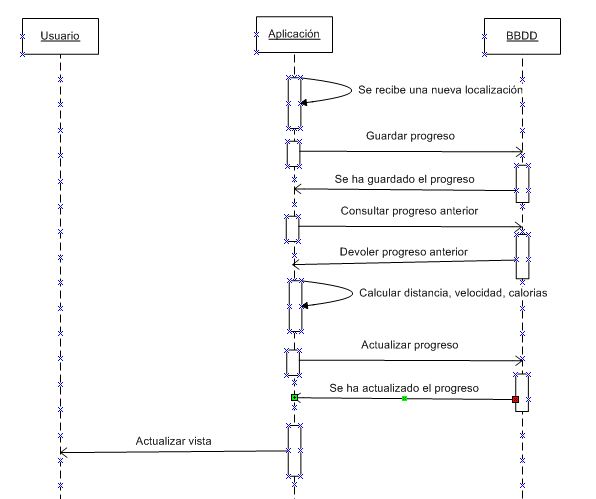
Una vez creado el ejercicio y mientras este dure, la aplicación estará recibiendo actualizaciones con los cambios en la localización del usuario. Cada notificación recibida creara un registro progreso asociado al ejercicio en la base de datos. Este registro contendrás una fecha, unas coordenadas GPS y unos campos distancia, calorías y velocidad inicialmente creados vacíos que dependen del progreso anterior (si existe). Se persistirá el progreso, se recuperará el progreso anterior y se realizarán cálculos sobre éstos datos. Se actualizará el registro con los datos calculados. Finalmente se actualizará la vista del sistema. El proceso se recoge en el diagrama “Actualizar ejercicio” presentado a continuación.

Figura 11. Diagrama de secuencias. Actualizar ejercicio: Crear progreso asociado a un ejercicio.

Finalmente cuando el usuario decida acabar el ejercicio el flujo corresponderá con la siguiente secuencia.

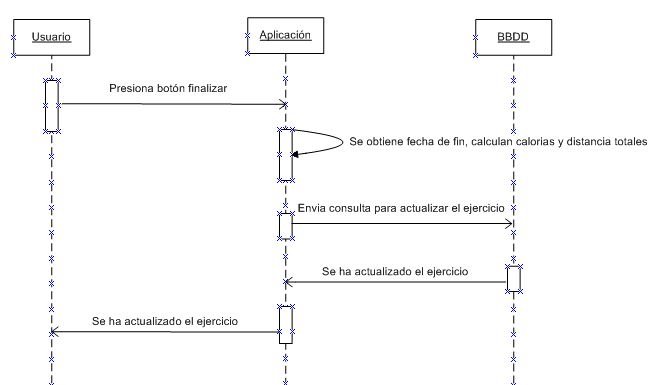


Figura 12. Diagrama de secuencias. Finalizar ejercicio

Borrar Ejercicio

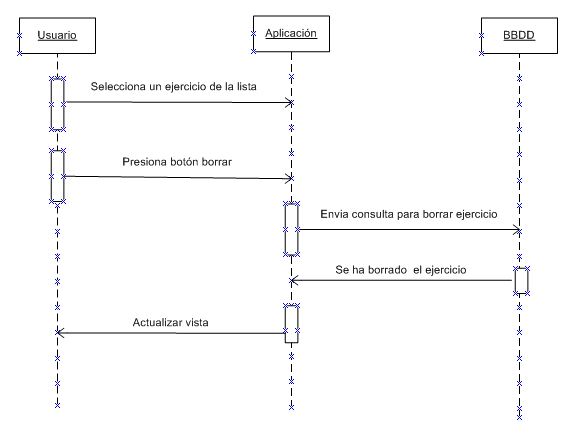


Figura 13. Diagrama de secuencias. Borrar Ejercicio

Ver listado de estadísticas disponibles

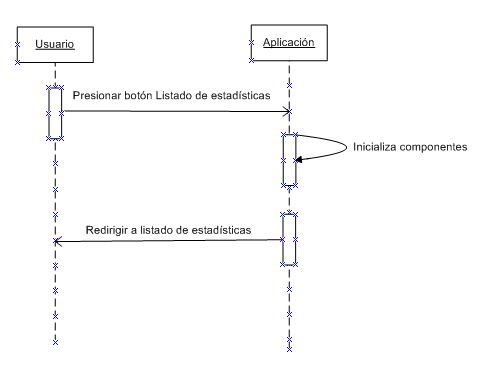


Figura 14. Diagrama de secuencias. Ver listado de estadísticas disponibles

Consultar estadística

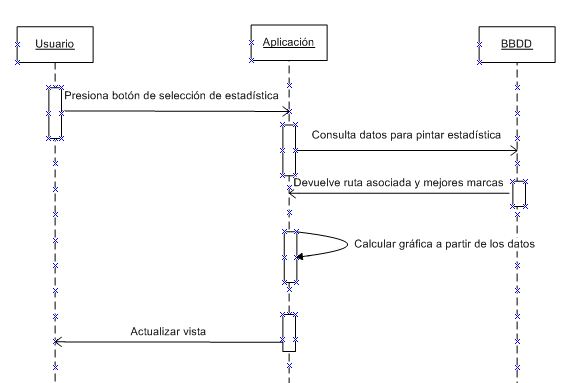


Figura 15. Diagrama de secuencias. Consultar estadística

Ver histórico de ejercicios

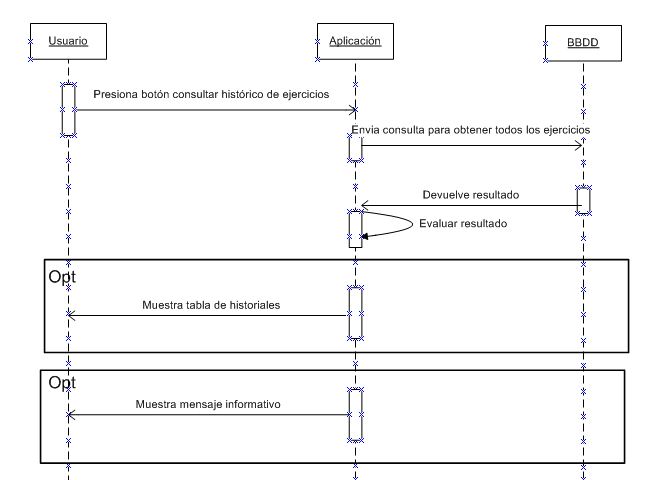


Figura 16. Diagrama de secuencias. Ver histórico ejercicios

Ver recorrido asociado a un ejercicio y mejores marcas

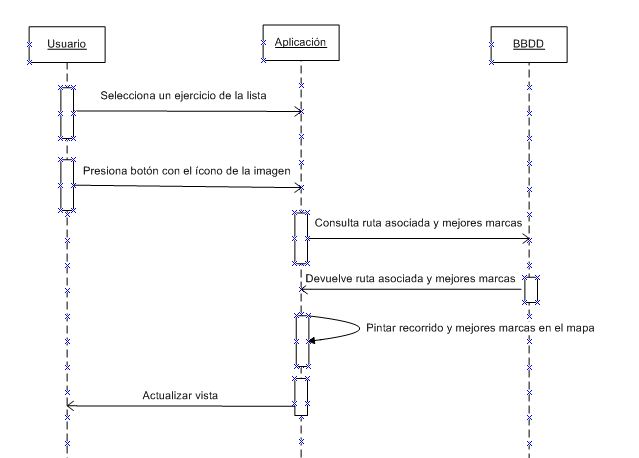


Figura 17. Diagrama de secuencias. Consulta recorrido y mejores marcas asociadas a un ejercicio

Logout

Cualquier flujo de la aplicación que acabe la aplicación desencadenará el logout, siendo preciso volver a cursar el proceso de login siempre que se vuelve a iniciar la actividad. Este proceso no se encuentra recogido con su correspondiente diagrama de secuencias por tratarse de un proceso gestionado a nivel de SO.

Cancelar cuenta

El proceso para cancelar una cuenta asociada a la aplicación puede realizarse por dos caminos, el primero supone el borrado completo de la aplicación desde el terminal, el siguiente implica borrar los datos persistidos por la misma. Una vez cancelada una cuenta, esta no podrá recuperarse. Se perderán también todos los datos asociados a la misma.

El primer camino requiere presionar el ícono de la aplicación desde el menú principal durante un par de segundos y arrastrarlo a la papelera una vez seleccionado.

En cambio para borrar los datos asociados a la aplicación, es preciso seguir el camino: Ajustes > Aplicaciones > Administrar aplicaciones > CalorieCalc > Borrar datos.

Este proceso no se especifica dentro de este apartado por tratarse de un proceso gestionado a nivel de SO no implementado por el alumno.

# Diseño arquitectónico

La siguiente tabla enseña de forma gráfica y resumida la totalidad de clases java que compone el sistema con su correspondiente distribución en la estructura del sistema.

|  |  |
| --- | --- |
| Paquete | Módulos |
| com.caloriecalc.beans | Ejercicio  LocationItemizaedOverlay  Serie |
| com.caloriecalc.content | Eula |
| com.caloriecalc.dao | DatabaseHelper  DAOEjercicio  DAOProgreso |
| com.caloriecalc.lao | LAOEjercicio  LAOProgreso  Utilities |
| com.caloriecalc.presentation | CalorieCalc  EjercicioActualActivity  MapViewActivity  NetworkSettings  StatsSelectingActivity  UserDataEditActivity  UserLoginActivity  UserRegistrationActivity  ValidateIdentityActivity |
| com.caloriecalc.presentation.graphs | Graph  BarGraph  LineGraph  PieGraph  ScatterGraph  WeightDialGraph |
| com.caloriecalc.security | Encrypt |
| com.caloriecalc.services | LocationService |

Tabla 3. Resumen de paquetes y módulos que componen el sistema

A continuación cada uno de estos paquetes se describe con más detalle.

## Paquete: com.caloriecalc.beans

Dentro de este paquete se localizan todos los beans internos del sistema utilizados para comunicarse entre la capa de datos y la capa de lógica, también serán utilizados para comunicarse entre la capa de lógica y la de presentación.

Estos beans solo incluyen métodos getters y setters además de los atributos que definen los objetos en el esquema de base datos. La clase serie se utiliza para alimentar las gráficas con datos representativos.

Los beans definidos dentro de este paquete son:

* Ejercicio
* Progreso
* Serie

Se cuenta como mínimo con las siguientes clases con sus getters y setters:

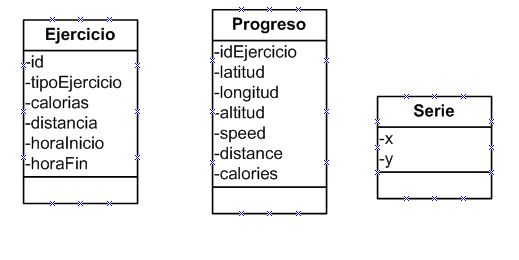


Figura 18. Paquete com.caloriecalc.beans. Diagrama de clases

## Paquete: com.caloriecalc.content

Dentro de este paquete se incluyen las clases relacionadas con el tratamiento de contenido tales como clase Eula utilizada para enseñar el End User Licence Agreement la primera vez que se ejecuta la aplicación.

## Paquete: com.caloriecalc.dao

Contiene las clases que actuarán en la gestión con las bases de datos, esta interfaz corresponde a la capa de datos del sistema. La clase DataBaseHelper se encarga de abrir y cerrar la conexión con la base de datos, su creación y todos los métodos genéricos. Dentro de este paquete también se encuentra una clase por cada tabla presente en el sistema, Ejercicio y Progreso, ambas extienden de DataBaseHelper. Tal como se describe en el siguiente esquema:

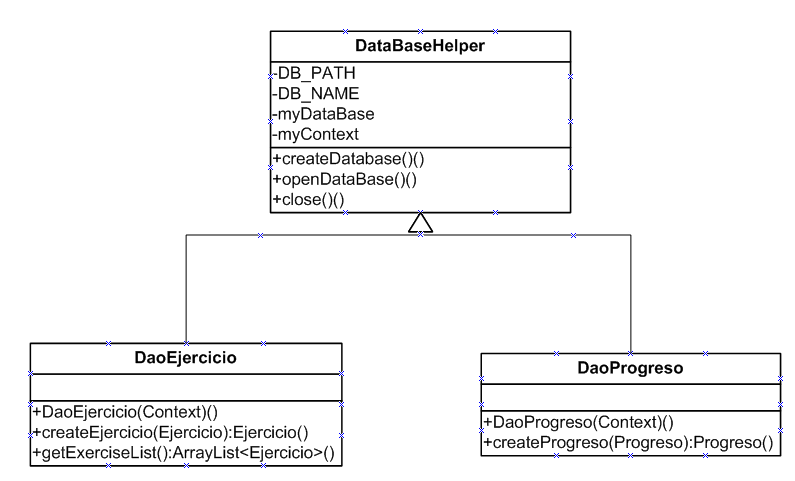


Figura 19. Paquete com.caloriecalc.dao. Diagrama de clases.

## Paquete: com.caloriecalc.lao

Dentro de este paquete se encapsula la lógica de negocio del sistema. Esta capa se corresponde con la capa de lógica de la arquitectura de tres capas.

Como mínimo este paquete deberá contar con una clase que corresponda a cada tipo de dao. Un objeto de lógica puede necesitar de más de un tipo de objeto de datos. Por ejemplo el LaoEjercicio utiliza un objeto de tipo DaoEjercicio, mientras que un objeto de tipo LaoProgreso necesita tanto un DaoProgreso como un DaoEjercicio.

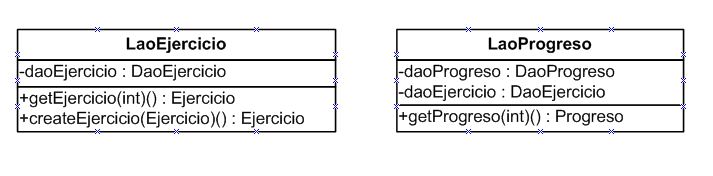


Figura 20. Paquete com.caloriecalc.lao. Diagrama de clases

## Paquete: com.caloriecalc.presentation

Dentro de este paquete se encuentran todos los controladores que interactúan con las vistas. En general existe una clase de presentación por cada vista, aunque una misma vista podría ser perfectamente controlada por más de un controlador.

## Paquete: com.caloriecalc.presentation.graphs

Dentro de este paquete se encapsulan todos los gráficos utilizados por la aplicación. En general tendremos una distribución donde todos los gráficos heredarán de un tipo genérico donde se realicen las configuraciones comunes.

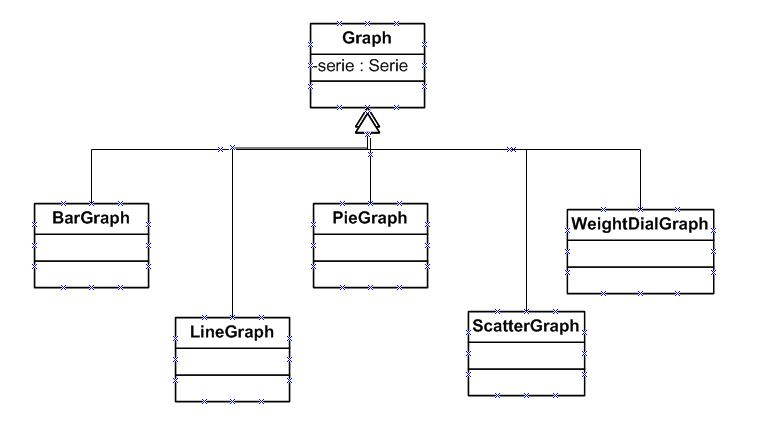


Figura 21. Paquete com.caloriecalc.presentation.graphs. Diagrama de clases

## 

## Paquete: com.caloriecalc.security

Dentro de esta clase se encuentran las clases que tratan el encriptado de claves. El algoritmo de encriptación es MD5. Estas clases participan en los casos de uso de registro de usuario, login y recuperación de contraseña.

## Paquete: com.caloriecalc.services

Dentro de este paquete se cuentan los servicios que no dependen de una vista ni de la interacción con el usuario para funcionar. Se ejecutan en background por el sistema.

## Dependencias entre paquetes:

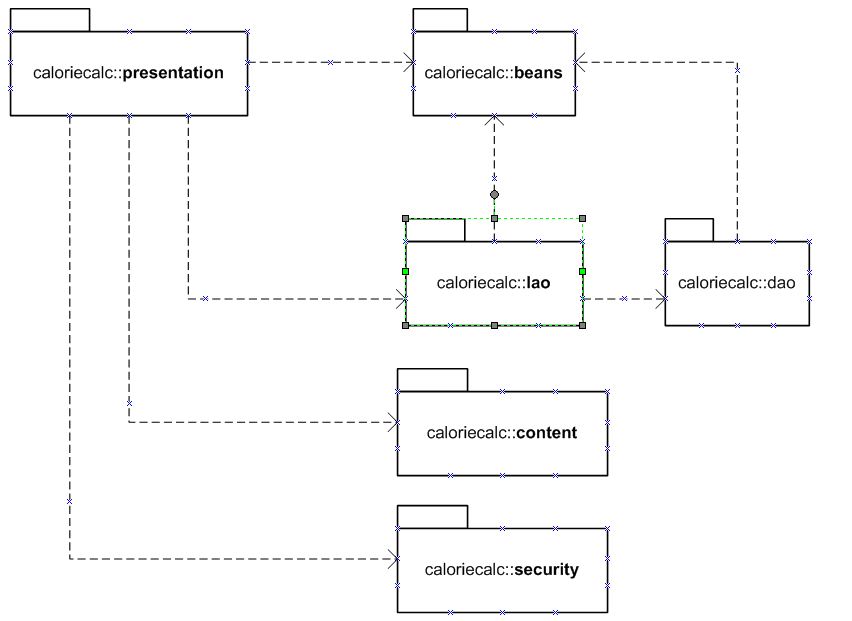


Figura 22. Dependencias entre paquetes

Tal como puede observarse en la imagen la capa de datos, paquete “dao” solo depende del paquete “beans”. La capa de lógica “lao” en cambio depende ambos del paquete “dao” y del paquete “beans”. Como es de esperar, la capa de presentación depende del paquete “lao” y del paquete “beans”. Además esta capa hace uso de los paquetes “content” y “security”. Se puede ver claramente la implementación de la arquitectura de tres capas.

# *Diseño de la Interfaz*

El diseño de la interfaz ha ido tomando forma en base a los distintos requisitos establecidos y por la pruebas de aceptación. Al tratarse de una aplicación para móviles y al tratarse de un sistema mono-usuario, solo se distingue un tipo de vista del sistema.

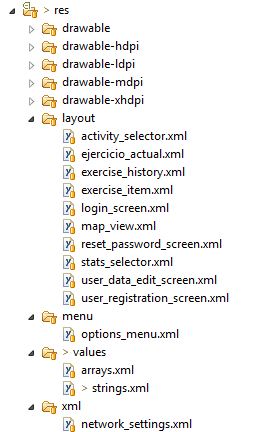
Las interfaces de usuario se diseñan en xml a través del paquete de UI de Android y su localización dentro de la estructura de paquetes es res/layout tal como ilustra la imagen adjunta.

Figura 23. Vistas disponibles en el paquete res

Estas vistas son controladas por controladores que gestionan la interacción del usuario con las vistas a través de EventHandlers. Dentro de este apartado solo nos centraremos en presentar las interfaces de usuario.

Todas las interfaces se presentan a continuación. La funcionalidad completa de cada una de estas pantallas se recoge en el “Anexo V: Manual de Usuario”.

|  |  |
| --- | --- |
| device-2012-04-14-194603.png  Figura 24. eula | device-2012-06-19-012451.png  Figura 25. Formulario de registro |
| device-2012-06-19-014546.png  Figura 26. Pantalla de login | device-2012-06-19-015340.png  Figura 27. Pantalla Recuperar password 1 |
| device-2012-06-19-015431.png  Figura 28. Pantalla Recuperar password 2 | device-2012-04-14-194901.png  Figura 29. Menú principal de la aplicación |
| device-2012-06-19-212518.png  Figura 30. Pantalla de selección de estadísticas | device-2012-06-19-203113.png  Figura 31. Pantalla ejercicio en curso |
| device-2012-06-18-013032.png  Figura 32. Pantalla histórico de ejercicios | device-2012-06-17-115922.png  Figura 33. Pantalla revisión de recorrido |
| Figura 34. Estadística Ejercicio vs. tipo | Figura 35. Estadística Peso vs. Fecha |
| device-2012-06-19-212537.png  Figura 36. Estadística pesos saludables | Figura 37. Estadística Calorías por tipo |

# *Entorno tecnológico del sistema*

Este apartado pretende definir por una parte el equipo físico, el equipo lógico y las comunicaciones que marcan el contexto del sistema software, y por otra todas las restricciones técnicas que existan.

El entorno tecnológico sobre el que residirá la aplicación móvil deberá cumplir con una serie de requisitos mínimos para su correcto funcionamiento.

Por una parte, para proceder a instalar la aplicación es indispensable tener un Smartphone o tablet con por lo menos 157Kb de espacio libre que cuente con un sistema operativo Android versión 2.1 o superior (API 7.0 o superior).

Por otra parte para el correcto funcionamiento de la aplicación es necesario contar con GPS y conexión a internet. A la hora de instalar la aplicación será preciso otorgar permiso para la utilización de estos componentes.

Se estima que prácticamente la totalidad de smartphones Android actuales cumplirán con estos requisitos, ya que las características demandadas por el sistema son bastante standard.