

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**SISTEMA DE GESTIÓN DE ACTIVIDADES PARA
CAMPAMENTOS**

**Borja Mendoza García
Tutor: Rosa María Carro Salas**

JUNIO 2016

SISTEMA DE GESTIÓN DE ACTIVIDADES PARA CAMPAMENTOS

AUTOR: Borja Mendoza García
TUTOR: Rosa María Carro Salas

Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2016

Resumen

Actualmente estamos tan habituados a la integración de las nuevas tecnologías en las tareas cotidianas, tanto en el ámbito profesional como en el personal, que no nos percatamos de que, no hace tantos años, la mayoría de estas tareas se realizaban de forma manual. Este trabajo se enmarca en el ámbito de la gestión de actividades de ocio y tiempo libre. Más concretamente, el propósito de este trabajo consiste en la creación de un sistema que facilite las labores de coordinadores y monitores de ocio y tiempo libre durante la preparación y el desarrollo de los campamentos que organizan.

El sistema diseñado y desarrollado se divide en dos partes, relacionadas entre sí, pero diferenciadas tanto en funcionalidad como con respecto al momento en que se utilizan. La primera de ellas consiste en una aplicación web creada para dar soporte a la incorporación y almacenamiento de toda la información relacionada con el campamento, desde los datos de los niños para su inscripción al campamento, hasta las actividades que van a realizar los monitores durante el mismo, pasando por la gestión de tareas y guardias a realizar. Esta aplicación está pensada para su uso previo al inicio del campamento. La segunda parte consiste en una aplicación móvil desarrollada para Android, que permite a los monitores, una vez que está teniendo lugar el campamento, acceder a toda la información desde sus *smartphones*, sin necesidad de tener que cargar y portar dosieres de papel en todo momento, y poder actualizar las actividades en caso de imprevistos o de desear adaptar las actividades inicialmente previstas conforme a las necesidades que surgen en el campamento, de forma que todos tengan la información actualizada rápidamente.

Estas dos aplicaciones están implementadas de una forma modular y escalable que permite la incorporación de nuevas funcionalidades realizando los mínimos cambios en el resto de módulos, e incluyen toda la funcionalidad esperada, con una interfaz sencilla, intuitiva y accesible para cualquier tipo de usuario.

Para la realización de este trabajo se ha contado con la experiencia de un grupo de monitores de ocio y tiempo libre, que han participado como usuarios finales de las aplicaciones desarrolladas, colaborando desde la identificación de los requisitos de las mismas hasta la evaluación del trabajo realizado.

Este documento presenta todo el proceso de desarrollo del proyecto desde que surgió la idea inicial, incluyendo la educación de requisitos de sendas aplicaciones, su diseño, su implementación y las pruebas realizadas con usuarios reales. Finalmente, presenta las conclusiones sobre el trabajo realizado y los planes de trabajo futuro.

Palabras clave

Aplicación Android, Aplicación Web, Diseño centrado en el usuario, Gestión de campamentos, Actividades de ocio y tiempo libre.

Abstract

Nowadays, we are so used to the integration of new technologies in daily routines, in both personal and private contexts, that we do not realize that, not many years ago, most of these tasks were performed manually. This work falls within the scope of leisure and free time activity management. More specifically, the purpose of this project is the creation of a system to facilitate the coordinators and instructors' work during the preparation and development of camps.

The system designed and developed is divided into two related parts. These parts are distinct between them, regarding both functionality and the time at which they are expected to be used. The first part consists of a web application to support getting and storing all the information related to the camp, including children's data for their registration, information about all the activities prepared by the monitors, and data for task and guard management. This application is aimed to be used before the camp starts. The second part of this project is a mobile application developed for Android, that allows the monitors, once the camp has started, to get access to all the data through their smartphones, without needing to carry paper dossiers during all the day. Moreover, it allows them to modify the activities planned when unexpected issues arise or in case that they wish to adapt the activities according to new detected needs. With this approach, everyone has all information updated immediately.

Both of these applications are implemented in a modular and scalable way that allows the incorporation of new features making minimal changes in other modules, and include all the expected functionality with simple, intuitive and accessible interface for the users.

A group of free-time instructors has participated as final users of the applications developed, helping us from the requirement gathering stage until the evaluation of the work done.

This document presents the entire development process since that initial idea, including requirement elicitation of both applications, their design, implementation, and evaluation with real users. Finally, it presents the conclusions about the work done and plans for future work.

Keywords

Android application, Web application, User-centered design, Camp management, recreation and leisure activities.

Agradecimientos

Hace ya muchos años desde que empecé a recorrer el camino del cual ahora se acerca el final. En todos estos años, mucha gente me ha acompañado, me ha ayudado, me ha enseñado, y me ha dado fuerzas para seguir cuando yo no veía salida. Espero no olvidarme de nadie.

En primer lugar, agradecer a mi familia, en especial a mis padres y a mi hermana. Sin su ayuda no hubiera podido empezar esta carrera que ahora llega a su fin. Han sido unos años duros, en los que han estado a mi lado aguantándome y dándome apoyo en épocas de exámenes y prácticas (es decir, semana tras semana sin descanso, durante años). Gracias.

Dar las gracias también a todos mis compañeros de carrera, este recorrido no habría sido el mismo sin vosotros. Son muchos, algunos llegaron desde el principio (Lucía, Jose, Marcos, Diego y Rafa), otros más adelante (Carlos, Miguel, Javi, Sergio, Mario...), pero todos han acabado por construir parte de la persona que soy hoy. Especial mención a Marcos Muñoz, que también me ha estado ayudando durante la realización de este proyecto, prestándome tanto su servidor para poder desplegar la base de datos, como su propio tiempo ayudándome a solucionar problemas con el mismo (esos errores 500 me los recordará siempre).

A mis amigos, gracias por estar ahí pese a viento y marea, aunque hubiera épocas en las que casi os olvidarais de mi cara por no poder salir debido a los estudios.

Un enorme GRACIAS también a este gran equipo de monitores. A la gran mayoría ya los conocía desde hace mucho, pero estos últimos 3 años con ellos han sido estupendos, y han pasado de ser amigos a ser familia. Lucía, Laura, Iván, Rocío, Noelia, Juan, Kike, Victoria, Alicia, Sergio, Leticia, Alberto, Sandra, Elvira, Mario, Nicolás, Patricia...de nuevo, gracias. Este proyecto es por y para vosotros.

Dar las gracias a todos mis profesores, de los que siempre he aprendido algo, y han sido pieza clave en que yo hoy me encuentre aquí.

Gracias especialmente a Rosa, fue la primera persona que se me vino a la cabeza cuando surgió la idea de este proyecto, y no dudó un momento en aceptar la propuesta, pese a que sabíamos que iba a ser un curso duro.

No puedo olvidarme de mis antiguos compañeros de Deloitte: Jorge Soto, Jorge Izquierdo, Loli, Guille, Miguel, Javi... y muchos más, que me han estado acompañando durante mi primera incursión al mundo laboral.

¡GRACIAS!

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos	1
1.3	Organización de la memoria.....	2
2	Análisis	3
2.1	Introducción.....	3
2.2	Catálogo de requisitos	5
2.2.1	Requisitos Funcionales	5
	Requisitos de la aplicación web	5
	Requisitos de la aplicación móvil.....	11
2.2.2	Requisitos No Funcionales	12
3	Diseño.....	13
3.1	Diagramas de casos de uso	13
3.2	Diagrama de clases	16
3.3	Base de Datos	20
3.4	Primer Prototipo	21
4	Desarrollo	23
4.1	Entorno y herramientas de trabajo.....	23
4.1.1	El lenguaje de programación Java	23
4.1.2	El <i>framework</i> de Spring	24
4.1.3	El entorno de desarrollo.....	25
4.1.3.1	Spring Tool Suite.....	25
4.1.3.2	JUnit	25
4.1.4	Base de datos MySQL	25
4.1.5	Amazon Web Services (AWS).....	26
4.1.6	Android.....	26
4.1.7	Android Studio	27
4.2	Desarrollo de la aplicación	27
4.2.1	Desarrollo de la aplicación web.....	27
4.2.2	Desarrollo de la aplicación Android.....	34
5	Pruebas y resultados	37
5.1	Inspecciones de código.....	37
5.2	Pruebas unitarias.....	37
5.3	Pruebas de compatibilidad.....	37
5.4	Pruebas de usuario	37
5.5	Pruebas de regresión.....	40
6	Conclusiones y trabajo futuro.....	41
6.1	Conclusiones	41
6.2	Trabajo futuro.....	42
	Referencias	43
	Glosario	i
	Anexos.....	A
A	Diseño completo de la Base de Datos	A
B	Esquema de tablas completo	A
C	Diagrama de clases completo.....	A
D	Integración con la Base de Datos desde Spring	A
E	Pruebas de compatibilidad	A
F	Encuesta de satisfacción.....	A
G	Distribución de las carpetas	A

INDICE DE FIGURAS

FIGURA 1. DIAGRAMA DE CASOS DE USO. ACCESO A LA PLATAFORMA.....	14
FIGURA 2. DIAGRAMA DE CASOS DE USO. GESTIÓN DE INSCRIPCIONES.....	14
FIGURA 3. DIAGRAMA DE CASOS DE USO. GESTIÓN DE ACTIVIDADES	15
FIGURA 4. DIAGRAMA DE CASOS DE USO. GESTIÓN DE GRUPOS	15
FIGURA 5. DIAGRAMA DE CASOS DE USO. GESTIÓN DE TAREAS Y GUARDIAS	16
FIGURA 6. DIAGRAMA DE CASOS DE USO. APLICACIÓN MÓVIL	16
FIGURA 7. DIAGRAMA DE CLASES. CONTROLLERS	17
FIGURA 8. DIAGRAMA DE CLASES. DAO 1.....	18
FIGURA 9. DIAGRAMA DE CLASES. DAO 2.....	19
FIGURA 10. DIAGRAMA DE CLASES. MODELOS 1	19
FIGURA 11. DIAGRAMA DE CLASES. MODELOS 2.....	20
FIGURA 12. DIAGRAMA E/R SIMPLIFICADO	21
FIGURA 13. BOTONES CON ICONOS ORIENTATIVOS.....	22
FIGURA 14. ESTRUCTURA DE LA PÁGINA WEB	22
FIGURA 15. PRIMER PROTOTIPO	22
FIGURA 16. ÍNDICE TIOBE EN LA COMUNIDAD DE PROGRAMADORES	24
FIGURA 17. MÓDULOS DEL ENTORNO DE EJECUCIÓN DE SPRING	25
FIGURA 18. ESTADÍSTICAS DE USO DE ANDROID	26
FIGURA 19. DISTRIBUCIÓN DE LAS VERSIONES DE ANDROID.....	27
FIGURA 20. DIAGRAMA DE CLASES. PATRÓN DAO	28
FIGURA 21. FORMULARIO DE CREACIÓN DE ACTIVIDADES	29
FIGURA 22. MENSAJE DE CREACIÓN DE ACTIVIDAD CORRECTO	29
FIGURA 23. VISOR DE ACTIVIDADES	30
FIGURA 24. EDICIÓN DE GRUPOS.....	30
FIGURA 25. ADMINISTRACIÓN DE TAREAS. ELIMINACIÓN	31
FIGURA 26. GENERACIÓN DE DOCUMENTOS PDF.....	33
FIGURA 27. FICHERO APPLICATION-CONFIG.XML – CONEXIÓN CORREO GMAIL	33
FIGURA 28. ENVÍO DE CORREO ELECTRÓNICO	33
FIGURA 29. CONEXIÓN APLICACIÓN ANDROID Y MYSQL.....	34
FIGURA 30. PANTALLA DE ACCESO A LA APP ANDROID.....	35
FIGURA 31. CONEXIÓN ANDROID CON BBDD MYSQL.....	35
FIGURA 32. LISTADO Y VISUALIZACIÓN DE ACTIVIDADES	36
FIGURA 33. DIAGRAMA E/R COMPLETO.....	A
FIGURA 34. DIAGRAMA DE CLASES COMPLETO. PARTE 1	B
FIGURA 35. DIAGRAMA DE CLASES COMPLETO. PARTE 2	C
FIGURA 36. DIAGRAMA DE CLASES	A
FIGURA 37. FICHERO POM.XML – DEPENDENCIAS	A

FIGURA 38. FICHERO MVC-CONFIG.XML	A
FIGURA 39. FICHERO APPLICATION-CONFIG.XML – CONEXIÓN BBDD	B
FIGURA 40. FICHERO APPLICATION-CONFIG.XML - SESSIONFACTORY Y BEANS DE CLASES.....	B
FIGURA 41. VISUALIZACIÓN WEB. NAVEGADOR GOOGLE CHROME.....	A
FIGURA 42. VISUALIZACIÓN WEB. NAVEGADOR MOZILLA FIREFOX.....	A
FIGURA 43. VISUALIZACIÓN WEB. NAVEGADOR INTERNET EXPLORER.....	B
FIGURA 44. DISTRIBUCIÓN DE CARPETAS. PROYECTO WEB. JAVA	A
FIGURA 45. DISTRIBUCIÓN DE CARPETAS. PROYECTO WEB. JSP	B
FIGURA 46. DISTRIBUCIÓN DE CARPETAS. PROYECTO ANDROID.....	C

INDICE DE TABLAS

TABLA 1. IMPLEMENTACIÓN DE LA SEGURIDAD DE DATOS	32
TABLA 2. DEPENDENCIA LIBRERÍA ITEXT	32
TABLA 3. PRUEBAS DE USUARIO. FUNCIONALIDAD. BLOQUE 1.....	38
TABLA 4. PRUEBAS DE USUARIO. FUNCIONALIDAD. BLOQUE 2.....	39
TABLA 5. PRUEBAS DE USUARIO. FUNCIONALIDAD. BLOQUE 3.....	39
TABLA 6. PRUEBAS DE USUARIO. INTERFAZ.	39
TABLA 7. TABLA USER.....	A
TABLA 8. TABLA MONITOR.....	A
TABLA 9. TABLA KID	B
TABLA 10. TABLA FATHER.....	C
TABLA 11. TABLA MOTHER	C
TABLA 12. TABLA CONTACT.....	C
TABLA 13. TABLA DIET.....	D
TABLA 14. TABLA ALLERGY	D
TABLA 15. TABLA MEDICALDATA.....	E
TABLA 16. TABLA KIDSGROUP.....	E
TABLA 17. TABLA ACTIVITY.....	F
TABLA 18. TABLA TASK.....	F
TABLA 19. TABLA WATCH.....	F

1 Introducción

1.1 Motivación

La labor de un monitor de ocio y tiempo libre previa a la realización de un campamento, así como sus tareas durante el desarrollo del mismo, es frenética. Es mucho el trabajo y las actividades que hay que preparar, mucho que organizar, y en demasiadas ocasiones la información está muy distribuida y el acceso a la misma es complicado.

Actualmente, las inscripciones a los campamentos de los que el autor de esta memoria es monitor, se realizan de forma física, teniendo que acudir los padres/tutores/familiares de forma presencial a recoger los formularios, y volver otro día a presentarlos. Esto genera retrasos en dichas inscripciones, y la necesidad, por parte del coordinador general del campamento, de organizar toda la información y comprobar que es correcta y completa.

De igual forma, los monitores han de rellenar las fichas correspondientes a las actividades que se realizarán en el campamento de forma individual, y enviarlas al coordinador para su revisión y organización.

El coordinador, una vez recopilada toda la información sobre asistentes y actividades, tiene que organizar el calendario, los grupos de asistentes, las tareas a realizar por cada uno, los menús... teniendo que revisar y actualizar la información en el caso de producirse cambios.

Este Trabajo de Fin de Grado surge para cubrir la necesidad de disponer de algún sistema informático que almacene toda la información en un único sitio, que sea accesible a todos los usuarios acorde a su perfil dentro del campamento, y que dé soporte, de forma sencilla, a la realización de todas las tareas de recogida y gestión de la información previas al campamento y durante el desarrollo mismo. Además, la aplicación móvil debe permitir a los monitores la actualización de las actividades durante la realización del campamento, teniendo todos ellos la información actualizada rápidamente en caso de imprevistos en mitad del campamento.

1.2 Objetivos

Este proyecto tiene como propósito la implementación de un sistema que permita la gestión de la información y de las actividades y tareas necesarias para la realización de un campamento. El proyecto se divide en dos partes:

- Una plataforma web (desde este punto, la llamaremos aplicación web), dirigida tanto a los padres como a los propios monitores del campamento, destinada a la recogida y almacenamiento de la información en una base de datos situada en un servidor externo.
- Una aplicación móvil (desde este punto, aplicación móvil), dirigida únicamente a los monitores, que se conecte a dicha base de datos y les permita acceder sencilla y rápidamente a todos los datos del campamento y gestionar las distintas tareas y actividades durante el mismo.

Aunque el proyecto se ha pensado para esta aplicación específica, se ha realizado de la forma más genérica posible, de modo que, si en un futuro, se desearan añadir nuevos módulos o funcionalidades, puedan integrarse realizando un número de cambios mínimo sobre la parte ya desarrollada.

1.3 Organización de la memoria

La estructura de esta memoria es la presentada a continuación:

En el *capítulo 2* se presenta el análisis del proyecto realizado, que incluye la elaboración del catálogo de requisitos que se satisfarán posteriormente.

El *capítulo 3* se centra en el diseño de la aplicación. En este capítulo se describe el diseño de interacción realizado (mediante diagramas de casos de uso), la organización de la información (mediante diagramas E-R) y el diseño de interfaz de las aplicaciones.

A lo largo del *capítulo 4* se describen las tecnologías empleadas para el desarrollo, incluyendo las plataformas de desarrollo, los lenguajes de programación utilizados y el gestor de bases de datos escogido. También en este capítulo se explica la implementación de algunos de los módulos y funcionalidades más importantes.

En el *capítulo 5* se detalla el proceso de verificación y de validación del proyecto a través de las pruebas realizadas.

Finalmente, en el *capítulo 6* se realiza una valoración final del proyecto desarrollado, incluyendo sus ventajas y posibilidades de mejora, así como algunas ideas de trabajo futuro.

2 Análisis

2.1 Introducción

En el desarrollo de un campamento participa un amplio número de personas, cada una con un perfil distinto. Los perfiles que se manejan a lo largo de este proyecto, cada uno con sus diversas funciones, son los que se presentan a continuación:

Coordinador de campamento: es el encargado de la gestión principal del campamento. En la actualidad, maneja todos los datos de inscripciones, calendarios y cocina en formato físico. En concreto:

- Tras las inscripciones, el coordinador debe organizar y archivarlos datos en función de la edad de los asistentes al campamento, separando en dos grupos (mayores y menores de 13 años).
- El coordinador formará los grupos de niños y se los asignará a cada monitor.
- El coordinador creará el dossier de actividades, revisando y organizando cronológicamente las actividades enviadas por el equipo de monitores.
- El coordinador organizará el cuadrante de tareas para el campamento según los grupos formados. De igual modo, organizará el cuadrante de guardias que realizarán los monitores en el periodo del campamento.
- El coordinador organizará, junto al equipo de cocina, el menú de las comidas para el campamento, así como la lista de la compra necesaria.

Monitor de campamento: cada monitor asistente al campamento se encargará de preparar una serie de actividades que entregará al coordinador para su revisión.

Equipo de cocina: junto con el coordinador, se encarga de tomar las decisiones acerca del menú que se servirá cada día en el campamento, de la preparación de la lista de la compra y del contacto con proveedores de alimentos para su entrega.

Usuario público: serán los padres, madres o responsables de los niños asistentes al campamento. Actualmente realizan el registro en el campamento de forma presencial rellenando los formularios de inscripción.

Para la identificación de requisitos, en primer lugar, se llevaron a cabo diversas entrevistas con los usuarios mencionados anteriormente, que se describen a continuación.

El objetivo de cada reunión es, por una parte, validar la propuesta inicial y los requisitos identificados por el autor de esta memoria y, por otra parte, recoger posibles requisitos o necesidades que pudieran no haber sido identificadas. Cabe mencionar que al ser el autor

de esta memoria uno de los monitores, y poseer, por tanto, conocimiento experto en el tema a tratar, el mapeo del modelo mental al modelo conceptual es más sencillo.

Reunión con coordinador de campamento: se presenta la idea del proyecto, a fin de definir con mayor exactitud el alcance y las funcionalidades deseadas. El coordinador aporta la información y documentación que en la actualidad se maneja de forma física, para elaborar un modelo que se pueda automatizar. En particular, los documentos aportados han sido los formularios de preinscripción e inscripción al campamento del curso pasado, junto con distintos dosieres de actividades de los campamentos realizados en años anteriores.

Reunión con monitores de campamento: al igual que se realizó con el coordinador, se realiza una presentación del proyecto y se recogen ideas y propuestas desde su punto de vista como monitores y su experiencia de pasados años.

Equipo de cocina: como parte del equipo de campamento, también se presenta el proyecto y se recogen sus propuestas para ser evaluadas.

De esta reunión surgió, entre otras cosas, la propuesta de automatizar la creación del menú del campamento, registrando para cada comida los ingredientes necesarios, a fin de posteriormente identificar las alergias alimenticias de forma más rápida.

Esta propuesta, tras analizarla detenidamente en una segunda reunión con el equipo de cocina, ha sido rechazada debido a la complejidad que conlleva el registro de todos los alimentos, ya que algunas alergias no dependen de un alimento en general, sino de sus componentes, y éstos, a su vez, pueden variar de una marca a otra. Por lo tanto, se decide que el menú de cocina se siga gestionando de forma manual.

Cabe mencionar que, para llevar a cabo estas reuniones, no ha sido necesario elaborar unos primeros bocetos en papel para comunicar la idea inicial del producto al usuario final y contrastarla, porque la idea inicial estaba clara. De hecho, la idea surge durante el desarrollo del campamento del año pasado en el que se comentó la necesidad de tener una aplicación disponible para facilitar la gestión del mismo. Sí se han utilizado documentos de campamentos anteriores (aportados por el coordinador) y durante las reuniones realizadas se tomaron notas con las conclusiones extraídas de cada una de ellas. En la etapa de diseño sí se elaborarán prototipos a validar por los usuarios.

A partir de las entrevistas mencionadas, se elabora el catálogo de requisitos detallado que se expone a continuación.

2.2 Catálogo de requisitos

A continuación, se describirán los diversos requisitos de los sistemas que se desarrollarán, divididos en dos bloques principales según su tipo: requisitos **funcionales** y requisitos **no funcionales**. Los requisitos **funcionales** son aquellos que recogen los aspectos relativos al software que implementado en las aplicaciones. Los requisitos **no funcionales** recogen el resto de aspectos del sistema, como la seguridad, interfaz o tecnologías empleadas.

La nomenclatura escogida para la enumeración de los requisitos es la siguiente:

- **RF / RNF:** Requisito Funcional / Requisito No Funcional
- **AW / AM:** Aplicación Web / Aplicación Móvil
- **A / M / P:** Administrador / Monitor / Público
- **GI / GG / GA / GT:** Gestión de Inscripciones / Grupos / Actividades / Tareas

2.2.1 Requisitos Funcionales

Dividiremos los requisitos funcionales en dos bloques: requisitos de la aplicación web, y requisitos de la aplicación móvil. Éstos, a su vez, estarán divididos según sean requisitos de la aplicación o requisitos relacionados con los roles de usuario.

Requisitos de la aplicación web

RF-AW (1). – La aplicación permitirá el acceso a distintos tipos de usuarios.

RF-AW (1.1). – Habrá 3 tipos distintos de usuarios:

- Administrador del sistema (coordinador)
- Monitor
- Público

RF-AW (1.2). – La aplicación prestará distintas funcionalidades según el usuario que interactúe con la misma.

RF-AW (1.3). – El acceso se realizará con la introducción de los siguientes campos:

- Nombre de usuario o correo electrónico
- Contraseña

RF-AW (2). – La aplicación permitirá el registro de nuevos usuarios. El formulario de inscripción solicitará los siguientes campos ¹:

- Nombre de usuario *
- Correo electrónico *
- Contraseña *

¹ Los campos indicados con (*) son obligatorios

RF-AW (3). – La aplicación permitirá a los usuarios la modificación de sus datos.

RF-AW (4). – La aplicación permitirá a los usuarios la recuperación de su contraseña vía correo electrónico.

RF-AW (5). – La aplicación permitirá a los usuarios la eliminación de su cuenta.

RF-AW (6). – La aplicación permitirá a los usuarios el cierre de sesión.

RF-AW (7). – La aplicación mostrará una página de información y contacto a los usuarios.

A continuación, se detallarán las funcionalidades que cada tipo de usuario tendrá dentro de la aplicación:

Tipo de usuario "Administrador"

RF-AW-A-GI (8). – El administrador podrá dar de alta a los niños asistentes al campamento mediante la incorporación de sus datos. Los datos a rellenar en el formulario serán los siguientes, ordenados por grupos ²:

- Datos personales:
 - Nombre *
 - Primer apellido *
 - Segundo apellido
 - Fecha de nacimiento *
 - Calle *
 - Número *
 - Piso
 - Puerta
 - Código Postal
 - Población
- Datos de los progenitores (en dos grupos):
 - Nombre del padre *
 - Primer apellido del padre *
 - Segundo apellido del padre
 - DNI del padre *
 - Correo electrónico del padre
 - Teléfonos:
 - Teléfono del domicilio *
 - Teléfono móvil del padre
 - Teléfono del trabajo del padre

² Los campos indicados con (*) son obligatorios

- Nombre de la madre*
- Primer apellido de la madre *
- Segundo apellido de la madre
- DNI de la madre *
- Correo electrónico de la madre
- Teléfonos:
 - Teléfono del domicilio *
 - Teléfono móvil de la madre
 - Teléfono del trabajo de la madre
- Otros contactos (será obligatorio al menos un contacto adicional):
 - Nombre *
 - Primer apellido *
 - Segundo apellido
 - Parentesco *
 - Teléfono de contacto *
- Datos acerca de las alergias:
 - Alergia a medicamentos
 - Alergia a alimentos
 - Alergia al polen
 - Alergia a animales
 - Otros tipos de alergias
 - Datos (campo en el que se especificarán las alergias señaladas)
- Datos relativos a dietas
 - Dietas (será un booleano que indicará si el niño/a tendrá alguna dieta específica a lo largo del campamento)
 - Datos
- Enfermedades
 - Enfermedad crónica
 - Asma
 - Corazón
 - Celiaquía
 - Diabetes
 - Infecciones de oído
 - Jaquecas / migrañas

- Intolerancias
- Dermatitis
- Hemorragias
- Sonambulismo
- Otras
- Tratamiento (campo en el que se especificará el tratamiento para las enfermedades señaladas)
- Otros datos médicos
 - Grupo sanguíneo *
 - Medicación
 - Qué medicación
 - Dosis de la medicación
 - Motivo de la medicación
 - Cartilla de vacunación completa
 - Antitetánica
 - Datos extra (cualquier dato de interés que se quiera añadir)

RF-AW-A-GI (9). – El administrador del sistema podrá modificar los datos de los niños/as inscritos.

RF-AW-A-GI (10). – El administrador podrá eliminar del sistema niños/as inscritos.

RF-AW-A-GI (11). – El administrador podrá consultar los datos de los niños/as inscritos.

RF-AW-A-GI (12). – El administrador podrá enviar correos electrónicos informativos a los usuarios registrados.

RF-AW-A-GG (13). – El administrador podrá crear los grupos de niños/as que estarán asignados a cada monitor durante el campamento.

RF-AW-A-GG (13.1). – El administrador podrá crear grupos de pequeños (menores de 13 años).

RF-AW-A-GG (13.2). – El administrador podrá crear grupos de mayores (mayores de 13 años).

RF-AW-A-GG (14). – El administrador podrá modificar los grupos de niños/as que tendrá cada monitor asignado durante el campamento.

RF-AW-A-GG (14.1). – El administrador podrá añadir niños/as nuevos a un grupo ya creado.

RF-AW-A-GG (14.2). – El administrador podrá eliminar niños/as del grupo en el que están asignados.

RF-AW-A-GG (14.3). – El administrador podrá añadir un monitor a un grupo ya creado.

RF-AW-A-GG (14.4). – El administrador podrá eliminar un monitor de un grupo ya creado.

RF-AW-A-GG (15). – El administrador podrá eliminar un grupo de niños-monitor creado.

RF-AW-A-GG (16). – El administrador podrá visualizar los grupos de niños-monitor creados.

RF-AW-A-GA (17). – El administrador podrá crear una actividad de campamento.

Los datos a rellenar serán los siguientes³:

- Nombre de la actividad *
- Tipo de actividad *
- Responsable de la actividad *
- Día y hora de la actividad *
- Lugar de realización de la actividad
- Objetivos de la actividad *
- Descripción de la actividad *
- Organización y equipos *
- Material necesario *
- Otros datos de interés

RF-AW-A-GA (18). – El administrador podrá modificar cualquier actividad creada previamente.

RF-AW-A-GA (19). – El administrador podrá eliminar cualquier actividad creada previamente.

RF-AW-A-GA (20). – El administrador podrá ver cualquier actividad creada previamente.

RF-AW-A-GA (21). – El administrador podrá crear una nueva tarea.

Los datos a rellenar en el formulario son los siguientes⁴:

- Número de grupo
- Tarea asignada
- Día

RF-AW-A-GT (22). – El administrador podrá eliminar una tarea.

RF-AW-A-GT (23). – El administrador podrá visualizar el cuadro de tareas.

RF-AW-A-GT (24). – El administrador podrá crear una nueva guardia.

³ Los campos indicados con (*) son obligatorios

⁴ Los campos indicados con (*) son obligatorios

Los datos a rellenar en el formulario son los siguientes⁵:

- Monitor asignado
- Tipo de guardia asignada
- Día

RF-AW-A-GT (25). – El administrador podrá eliminar una guardia.

RF-AW-A-GT (26). – El administrador podrá visualizar el cuadro de guardias.

Tipo de usuario "Monitor"

RF-AW-M-GI (27). – El monitor podrá consultar los datos de los niños/as inscritos.

RF-AW-M-GG (28). – El monitor podrá visualizar su grupo de niños/as - asignado.

RF-AW-M-GA (29). – El monitor podrá crear una actividad de campamento.

Los datos a rellenar serán los siguientes⁶:

- Nombre de la actividad *
- Tipo de actividad *
- Responsable de la actividad *
- Día y hora de la actividad *
- Lugar de realización de la actividad
- Objetivos de la actividad *
- Descripción de la actividad *
- Organización y equipos *
- Material necesario *
- Otros datos de interés

RF-AW-M-GA (30). – El monitor podrá modificar cualquiera de sus actividades creadas previamente.

RF-AW-M-GA (31). – El monitor podrá eliminar cualquiera de sus actividades creadas previamente.

RF-AW-M-GA (32). – El monitor podrá ver cualquier actividad creada previamente.

RF-AW-M-GT (33). – El monitor podrá visualizar el cuadro de guardias.

RF-AW-M-GT (34). – El monitor podrá visualizar el cuadro de tareas.

Tipo de usuario "Público"

⁵ Los campos indicados con (*) son obligatorios

⁶ Los campos indicados con (*) son obligatorios

Los requisitos **RF-AW-P-GI (35) – RF-AW-P-GI (38)**, referentes al registro de asistentes en el campamento, son equivalentes a los descritos en los requisitos **RF-AW-C-GI (8) – RF-AW-C-GI (11)**, con la diferencia de que el perfil de éstos es Público.

RF-AW-P-GI (39). – El usuario público podrá enviar correos electrónicos de consulta a los administradores.

RF-AW-P-GI (40). – El usuario público podrá generar un documento PDF con un resumen de la inscripción.

RF-AW-P-GI (40.1). – El usuario podrá guardar en su máquina local el PDF generado.

RF-AW-P-GI (40.2). – El usuario podrá imprimir el PDF generado.

Requisitos de la aplicación móvil

RF-AM (41). – La aplicación permitirá el acceso a distintos tipos de usuarios.

RF-AW (41.1). – Habrá 2 tipos distintos de usuarios:

- Administrador del sistema (coordinador)
- Monitor

RF-AW (41.2). – El acceso se realizará con la introducción de los siguientes campos:

- Nombre de usuario
- Contraseña

RF-AM (42). – La aplicación permitirá a los usuarios el cierre de sesión.

Tipo de usuario “Administrador” y "Monitor"

RF-AM-AM-GI (43). – La aplicación permitirá a los usuarios la visualización del listado de niños asistentes al campamento.

RF-AM-AM-GI (43.1). – La aplicación permitirá la visualización de los datos de cada uno de los niños de forma individual.

RF-AM-AM-GA (44). – La aplicación permitirá a los usuarios la visualización del listado de actividades del campamento.

RF-AM-AM-GA (44.1). – La aplicación permitirá la visualización de los datos de cada uno de las actividades de forma individual.

RF-AM-AM-GA (44.2). – La aplicación permitirá la actualización de los datos de las actividades de las que es responsable.

RF-AM-AM-GG (45). – La aplicación permitirá a los usuarios la visualización del grupo de niños que tienen asignados.

RF-AM-AM-GT (46). – La aplicación permitirá a los usuarios la visualización del cuadrante de tareas del campamento.

RF-AM-AM-GT (47). – La aplicación permitirá a los usuarios la visualización del cuadrante de guardias del campamento.

2.2.2 Requisitos No Funcionales

1.- Requisitos de interacción e interfaz

RNF-AW (1). – La interfaz gráfica de la aplicación web deberá ser sencilla, de forma de que cualquier usuario, independientemente del grado de conocimiento que tenga sobre la materia, sea capaz de explotar de forma completa la funcionalidad de la aplicación.

2.- Requisitos de documentación

RNF-AW (2). – El sistema web se presentará en castellano, pues de momento el ámbito de uso está acotado a campamentos en el ámbito del territorio español.

3.- Requisitos de mantenibilidad y portabilidad

RNF-AW (3). – La aplicación web se desarrollará siguiendo los estándares correspondientes en cada caso, de modo que sea compatible en distintos navegadores y/o Sistemas Operativos.

RNF-AW (3.1). – Los navegadores para los que se desarrollara la aplicación web serán Internet Explorer, Mozilla Firefox, y Google Chrome, por su mayor extensión en el mercado.

4.- Requisitos de verificación y portabilidad

RNF-AW (4). – La aplicación web contará con un sistema de cifrado de datos para proteger las contraseñas de los distintos usuarios en la base de datos.

5.- Requisitos de rendimiento

RNF-AW (5). – El sistema web deberá tener un tiempo de respuesta inferior a 2 minutos.

RNF-AM (6). – La aplicación móvil permitirá el acceso a 1 usuario por dispositivo de forma concurrente.

RNF (7). – La base de datos distribuida deberá permitir el acceso concurrente a 50 usuarios.

6.- Requisitos tecnológicos

RNF-AM (8). – La aplicación móvil será compatible con la plataforma Android para móviles y tabletas con conexión a Internet.

3 Diseño

3.1 Diagramas de casos de uso

Para definir de forma clara la interacción de los usuarios dentro de la aplicación web, se han desarrollado los siguientes diagramas de casos de uso, organizados según la funcionalidad de la aplicación descrita en el catálogo de requisitos en el punto 2.2.1

En primer lugar, se procede a definir los distintos objetos que conforman cada uno de los diagramas:



Agente: será el usuario que interviene en el diagrama, según su perfil dentro de la aplicación.



Globo de acción: representa las distintas funcionalidades que se realizan en el diagrama.



Línea o flecha: relaciones entre los agentes y las acciones

En primer lugar, como se refleja en la Figura 1, se define la interacción para el acceso a la plataforma web, en la que intervienen todos los distintos perfiles de la aplicación. Cualquiera de los actores podrá iniciar sesión en la aplicación. Esto puede generar dos respuestas: si el login es correcto se mostrará la pantalla principal de la aplicación, y si el login es incorrecto se volverá a mostrar la página de acceso con la opción de recuperar la contraseña. Cabe destacar aquí que los perfiles de administrador y monitores serán gestionados manualmente y dados de alta de forma directa en la BBDD, por lo que el único usuario que precisará de registro será el **público**. Una vez se ha realizado el inicio de sesión, se acceden a distintas funcionalidades en función del perfil que el usuario tenga dentro de la plataforma.

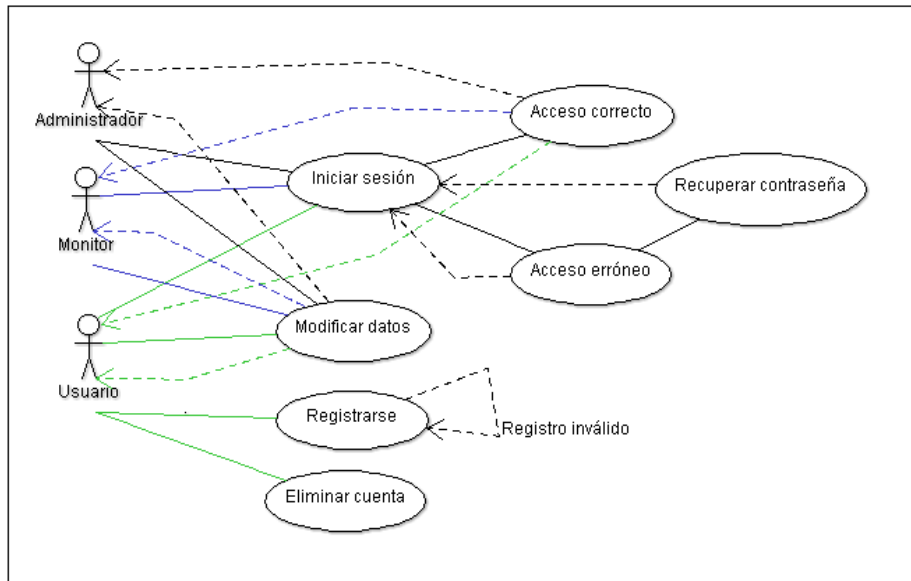


Figura 1. Diagrama de casos de uso. Acceso a la plataforma

La Figura 2 refleja la interacción con la plataforma para el registro de participantes en las bases de datos del campamento. En la realización de las acciones referentes al registro de niños en el campamento intervienen únicamente los usuarios públicos y el administrador del sistema, que podrá incorporar los datos de los niños que aún realicen la inscripción según el método antiguo para darles de alta. El monitor, dentro de este esquema, únicamente podrá visualizar los datos de los niños una vez que estos han sido dados de alta. Al diagrama recogido en la Figura 2 también se ha añadido la acción de envío de correos electrónicos. Esta acción estará disponible para todos los usuarios.

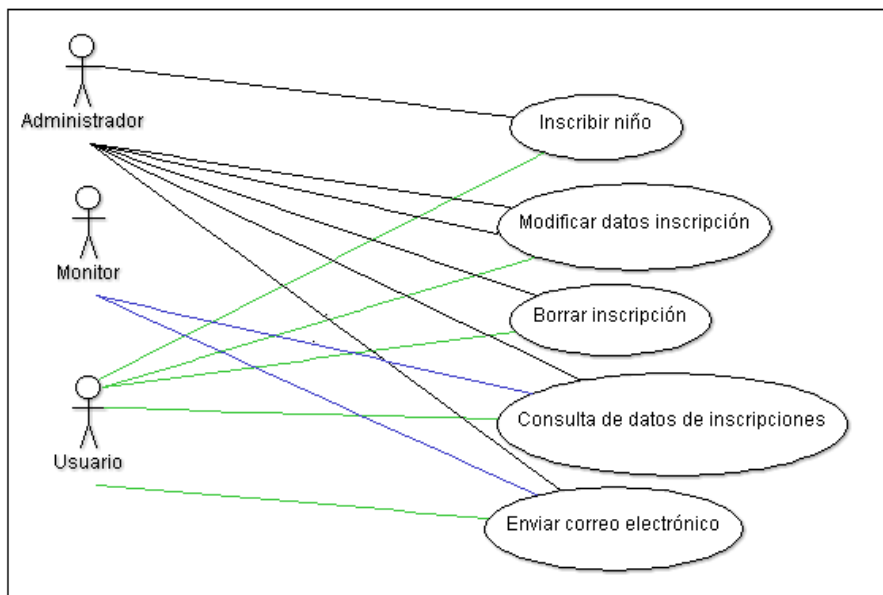


Figura 2. Diagrama de casos de uso. Gestión de inscripciones

La Figura 3 muestra el diagrama de casos de uso correspondiente a las acciones disponibles para la gestión de actividades del campamento. Estas acciones podrán ser llevadas a cabo únicamente por monitores y administradores/coordinadores.

Dentro de las acciones llevadas a cabo por el usuario con perfil de **monitor**, únicamente podrá realizar la modificación y/o eliminación de sus propias actividades.

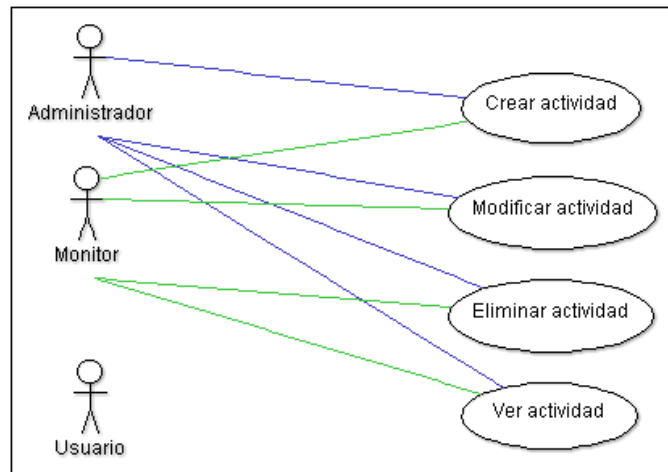


Figura 3. Diagrama de casos de uso. Gestión de actividades

Las Figura 4 refleja las acciones de gestión de grupos, mientras que la Figura 5 muestra la gestión de tareas y guardias. Estas acciones serán llevadas a cabo, en su mayoría, únicamente por el **coordinador**, que tendrá los permisos para llevar a cabo todas ellas. El **monitor**, en estos casos, sólo podrá realizar las acciones relativas a la visualización de contenidos.

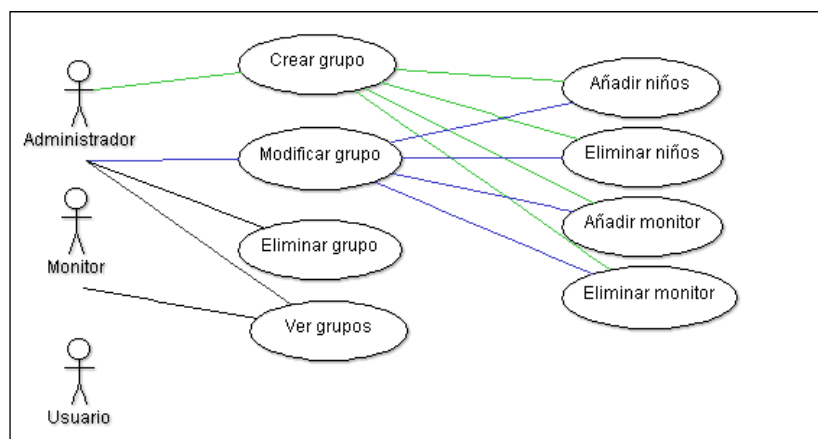


Figura 4. Diagrama de casos de uso. Gestión de grupos

La Figura 6 muestra la interacción del usuario dentro de la aplicación móvil, con las funcionalidades definidas en el punto 2.2.1.

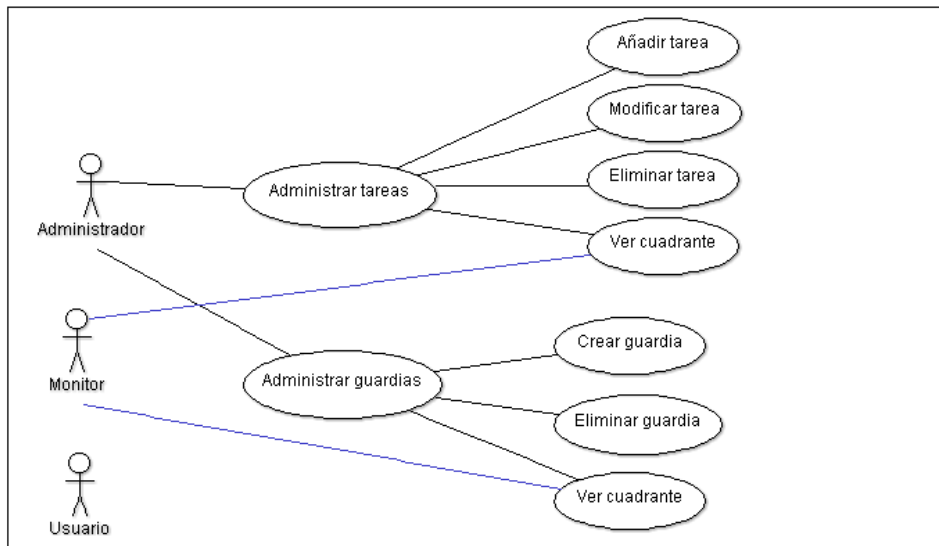


Figura 5. Diagrama de casos de uso. Gestión de tareas y guardias

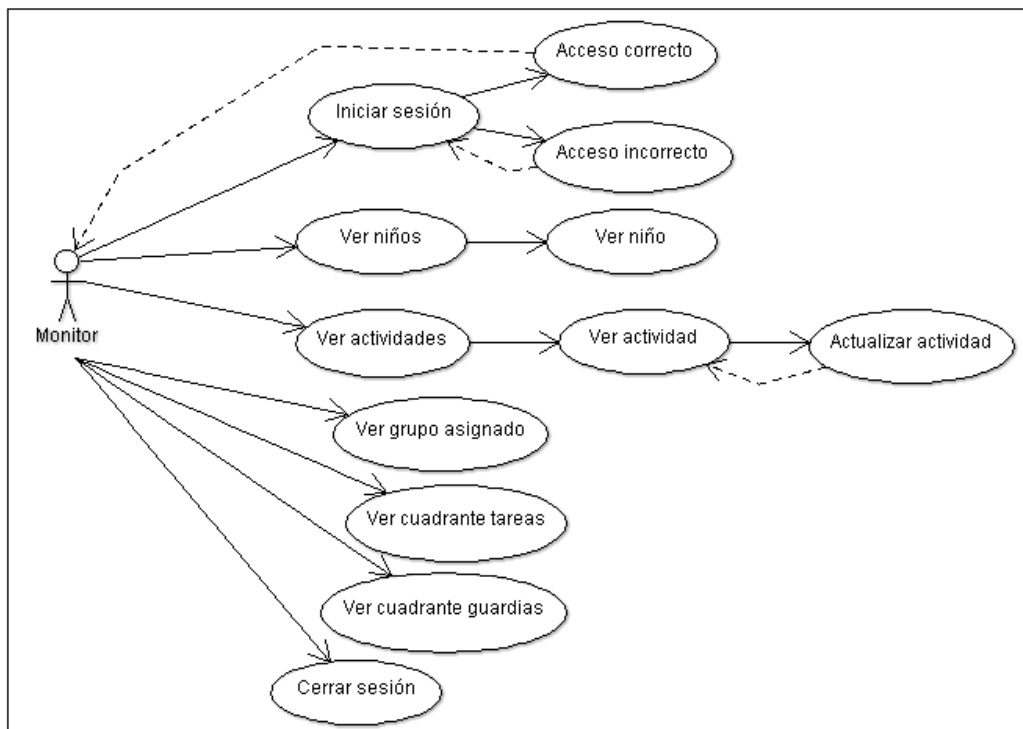


Figura 6. Diagrama de casos de uso. Aplicación móvil

3.2 Diagrama de clases

Para el desarrollo del proyecto, se elabora un diagrama de clases, dividiendo tres grupos principales: controladores, DAO y modelos.

Controladores: estas clases Java, mostradas en la Figura 7, se encargarán de recoger las acciones introducidas y por los usuarios, y actualizar los modelos y las vistas según dichos eventos. Habrá un controlador según el tipo de acciones que realicen (registro de actividades, registro de niños, encriptado de datos...).

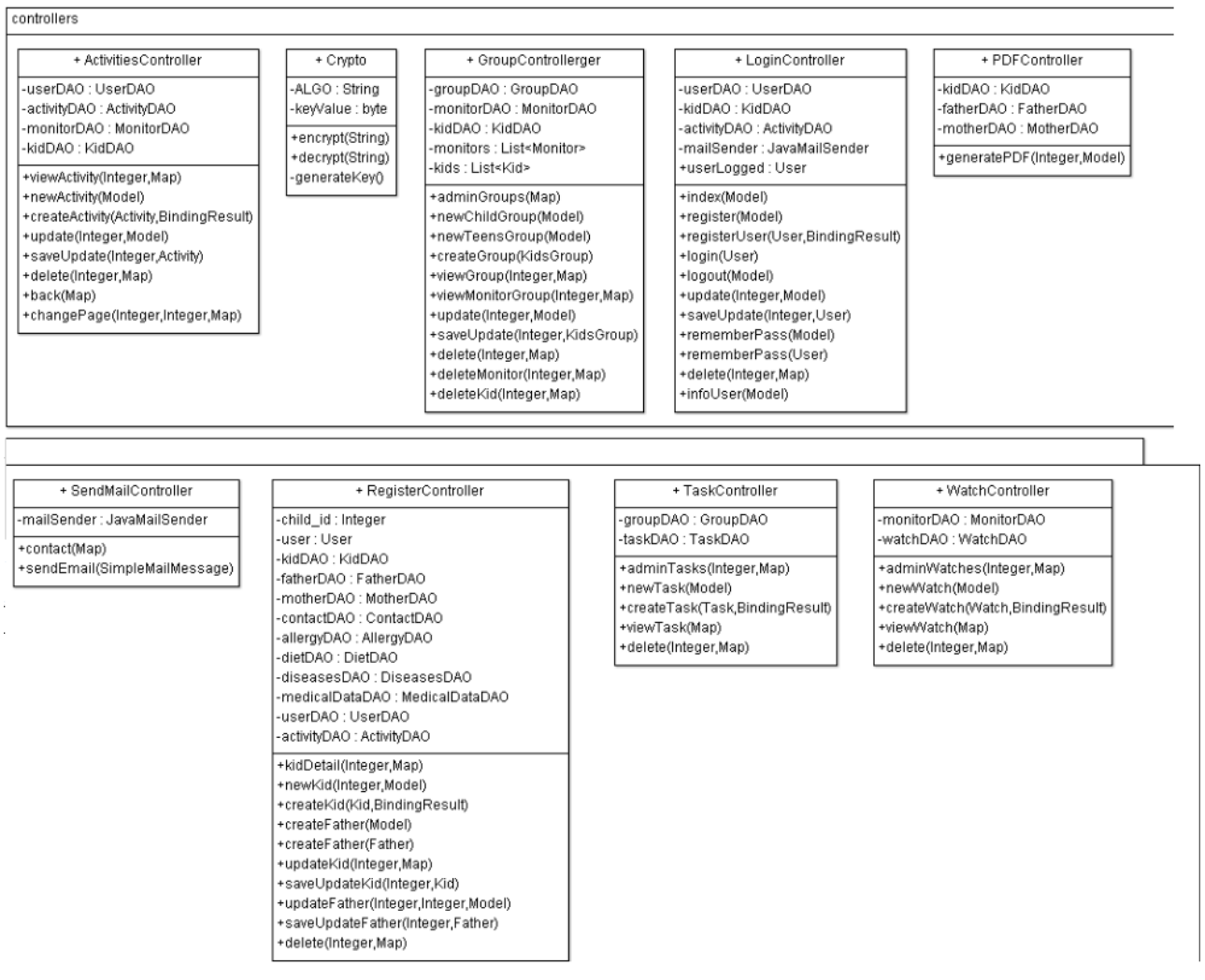


Figura 7. Diagrama de clases. Controllers

DAO: En la Figura 8 y la Figura 9, se distinguen dos subgrupos. El primero engloba las interfaces DAO que contendrán los métodos que deberán implementarse para la interacción del sistema con los datos, y un segundo grupo, HibernateDAO, cuyas clases implementarán las interfaces definidas previamente. Existirá una interfaz DAO, con su correspondiente implementación HibernateDAO para cada uno de los modelos de datos que se va a manejar dentro de la aplicación y que se verán a continuación.

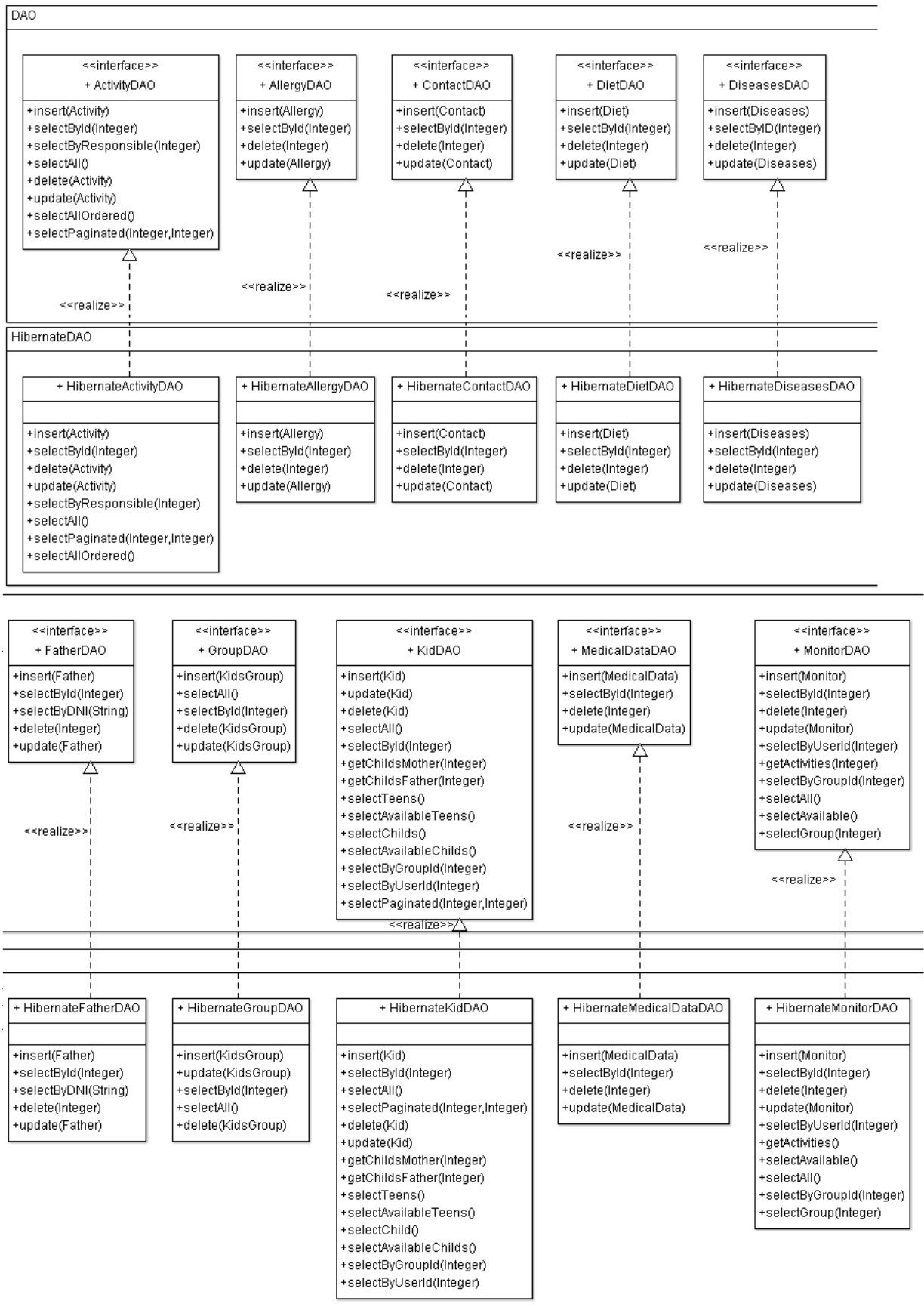


Figura 8. Diagrama de clases. DAO 1

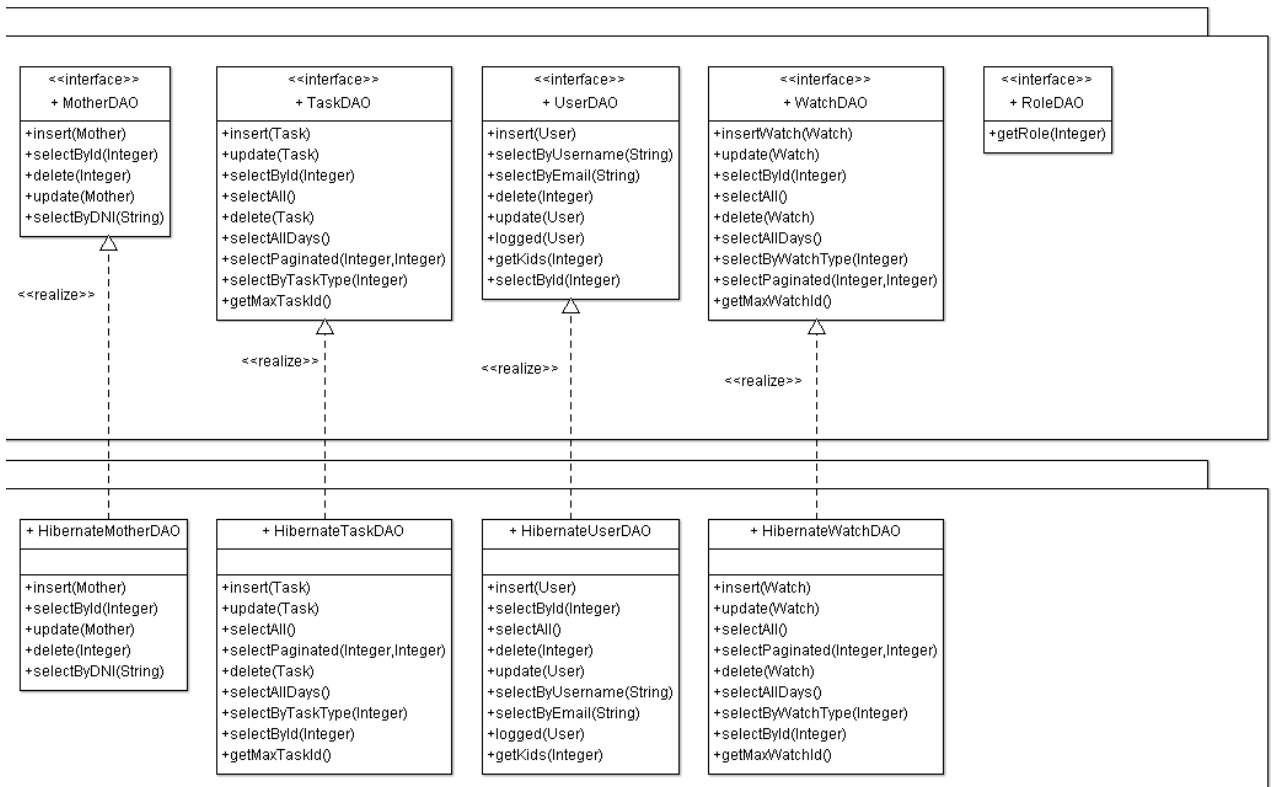


Figura 9. Diagrama de clases. DAO 2

Modelos: el grupo de clases de la Figura 10 junto con la Figura 11 representará el modelo de datos que se manejará dentro de la aplicación, y que se almacenará en la base de datos para su persistencia. Se creará una clase por cada modelo, que tendrá como atributos los datos que se precisan para cada uno según queda especificado en el punto 2.2.1. En el diagrama de clases, por cuestiones de espacio, se han omitido los métodos get y set de cada atributo en cada modelo.

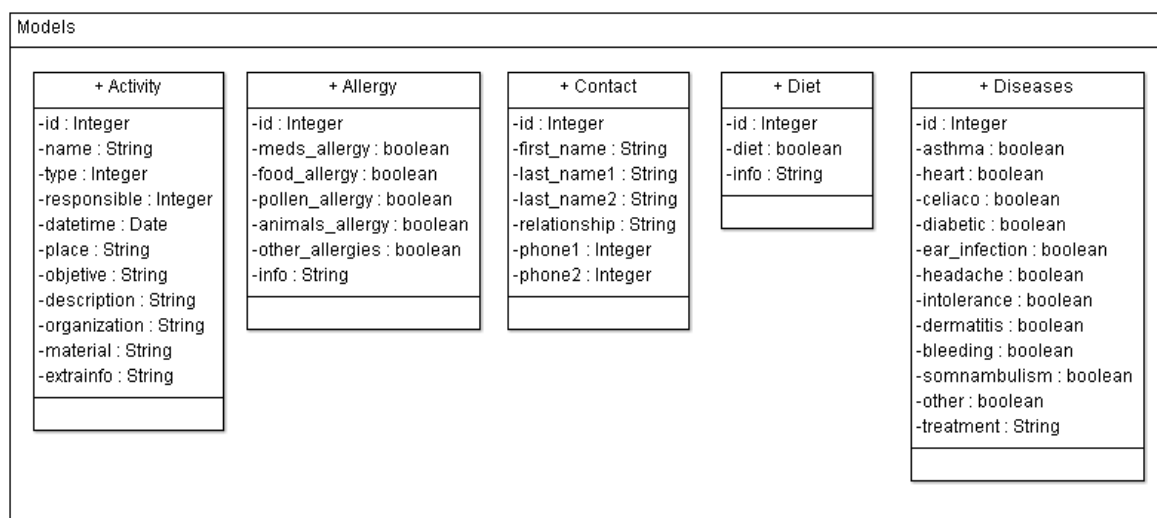


Figura 10. Diagrama de clases. Modelos 1

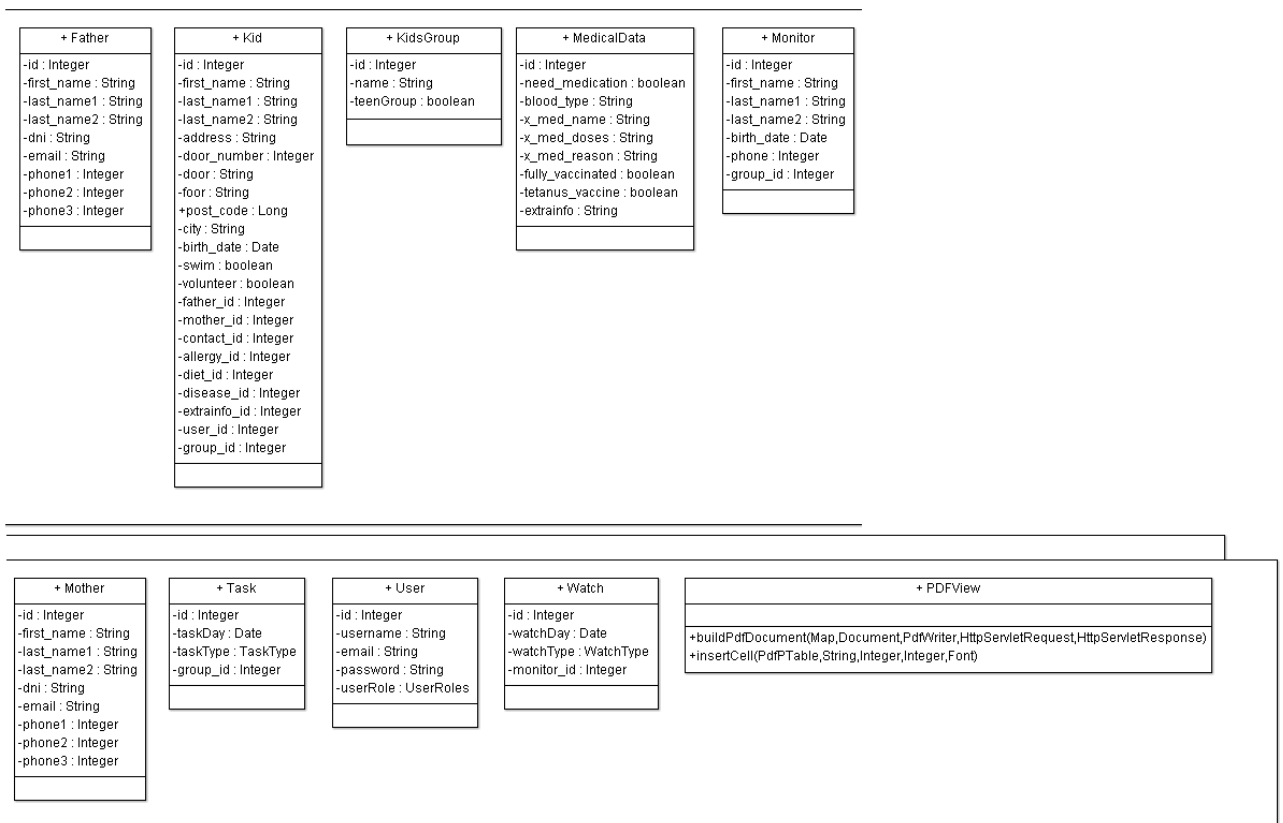


Figura 11. Diagrama de clases. Modelos 2

Debido a que el esquema completo es muy extenso, queda recogido completo en el Anexo C.

3.3 Base de Datos

El diseño de la base de datos se ha realizado considerando toda la información que actualmente se maneja de forma física a través de los formularios de inscripción (en el caso del registro de participantes en el campamento), o de las memorias de campamento que el conjunto de monitores y coordinadores elaboran. El diagrama Entidad-Relación creado para este propósito queda reflejado en la Figura 12. Debido al número tan grande de atributos, el diagrama completo se hace demasiado extenso. Queda reflejado en el Anexo A. En el Anexo B se da el detalle completo de todas las tablas con sus atributos.

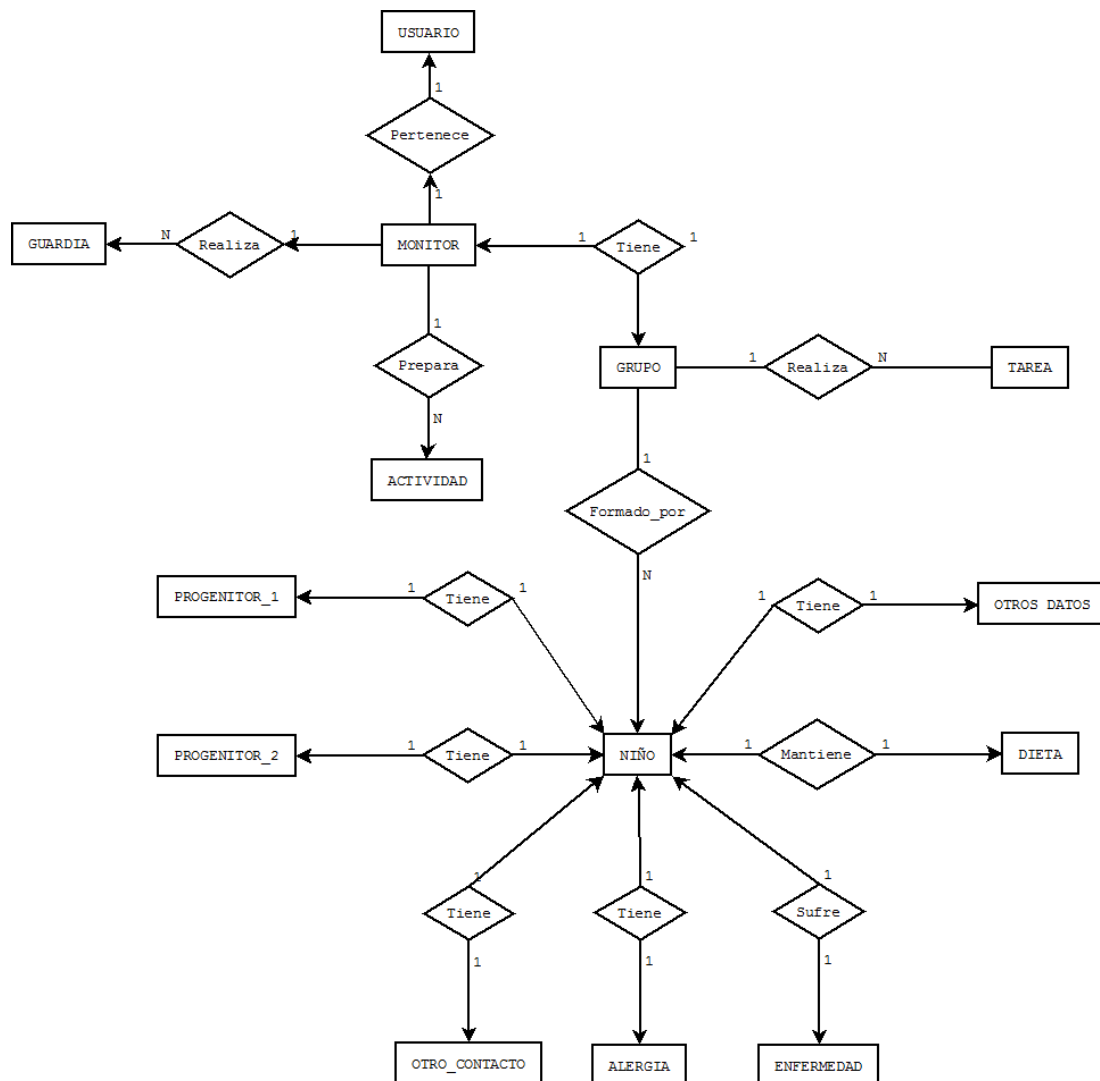


Figura 12. Diagrama E/R simplificado

3.4 Primer Prototipo

Se ha desarrollado un primer prototipo de forma incremental. Inicialmente el prototipo no contenía funcionalidad y se utilizó para validar las decisiones tomadas con respecto al diseño de la interacción e interfaz. Para el desarrollo de este prototipo se tuvieron en cuenta los principios y guías para el diseño de la interacción[11]. En particular:

- **Consistencia y simplicidad:** los nombres de las acciones son específicos a la funcionalidad deseada.
- **Memoria humana:** que el usuario no precise recordar grandes cantidades de datos.
- **Orientación cognitiva:** con el fin de minimizar el esfuerzo necesario para la realización de tareas, los botones o iconos seleccionables para llevar a cabo las mismas tienen un color o un icono que al usuario le es fácilmente reconocible, por ejemplo, una papelera para eliminar un objeto, un disco para guardar, un ojo para visualizar...La Figura 13 muestra ejemplos de algunos de los iconos utilizados.



Figura 13. Botones con iconos orientativos

- **Feedback informativo:** tras la realización de alguna acción, se muestra al usuario un mensaje que le informa si se ha realizado correctamente o ha surgido un error.
- Empleo **moderado** de fuentes y colores: a lo largo de la aplicación, no se varían las fuentes del texto, ni se satura la pantalla con diversos colores.
- **Despliegue de las pantallas:** las pantallas de la aplicación mantienen el formato, variando lo menos posible, con el fin de simplificar las acciones del usuario.

La Figura 14 muestra la estructura básica de la aplicación web y la Figura 15 presenta el primer prototipo desarrollado, el cual solo permitía el cambio de una ventana a otra.



Figura 14. Estructura de la página web



Figura 15. Primer prototipo

Tras la realización de este primer prototipo, se fueron añadiendo una por una las funcionalidades definidas en el apartado 2.2. Durante todo este proceso se han contado con una de las monitoras que ha ido validando el prototipo en distintos momentos de su desarrollo, y se han ido considerando sus sugerencias para la elaboración del prototipo final. Este prototipo ha sido refinado hasta obtener la aplicación final. En el siguiente apartado se describe cómo se ha llevado a cabo este proceso y en el apartado 1 se presentan las pruebas realizadas con todos los usuarios y los resultados obtenidos.

4 Desarrollo

4.1 Entorno y herramientas de trabajo

Como la aplicación está pensada para ser ejecutada en dispositivos móviles, es necesario utilizar tecnología que así lo permita. El sistema operativo elegido para dar soporte a la ejecución de la aplicación móvil ha sido Android, al ser el sistema más común entre los usuarios de teléfonos inteligentes, y entre el grupo de monitores con el que se ha trabajado durante la realización del proyecto.

Se ha escogido desarrollar el proyecto en Java mediante el *framework* de Spring por las facilidades que este nos da para el desarrollo de aplicaciones web mediando páginas JSP, la integración de librerías y dependencias de forma automática, o la posibilidad de usar Spring Security como forma de gestionar los accesos y permisos a distintas páginas según los usuarios. En los siguientes subapartados se explican las características de todas las tecnologías y herramientas y se justifica su elección para la realización de este proyecto.

4.1.1 El lenguaje de programación Java



Java es un lenguaje de programación de **propósito general, concurrente, orientado a objetos** que fue diseñado para tener tan pocas dependencias de implementación como fuera posible.

Las aplicaciones Java son generalmente compiladas a **bytecode** (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente, lo cual es necesario para este proyecto, al poder ejecutarse en distintos tipos de máquinas dependiendo de los usuarios.

Algunas de las principales características de este lenguaje de programación, que han motivado su selección para la implementación de este proyecto, son:

- **Orientado a objetos:** permite el polimorfismo, herencia (no herencia múltiple), encapsulamiento, abstracción, interfaces...
- **Distribuido:** se ha diseñado para trabajar en ambientes de redes y contienen una gran biblioteca de clases para la utilización del protocolo TCP/IP, incluyendo HTTP y FTP.
- **Sólido:** El código Java no se quiebra fácilmente. Es un lenguaje con muchas restricciones que, en la mayoría de los casos, no permite una conversión forzada (*cast*).

Además, este lenguaje es el más empleado en la actualidad en el desarrollo de aplicaciones, según rankings como TIOBE o PYPL, como se muestra en la Figura 16 [6] [7].

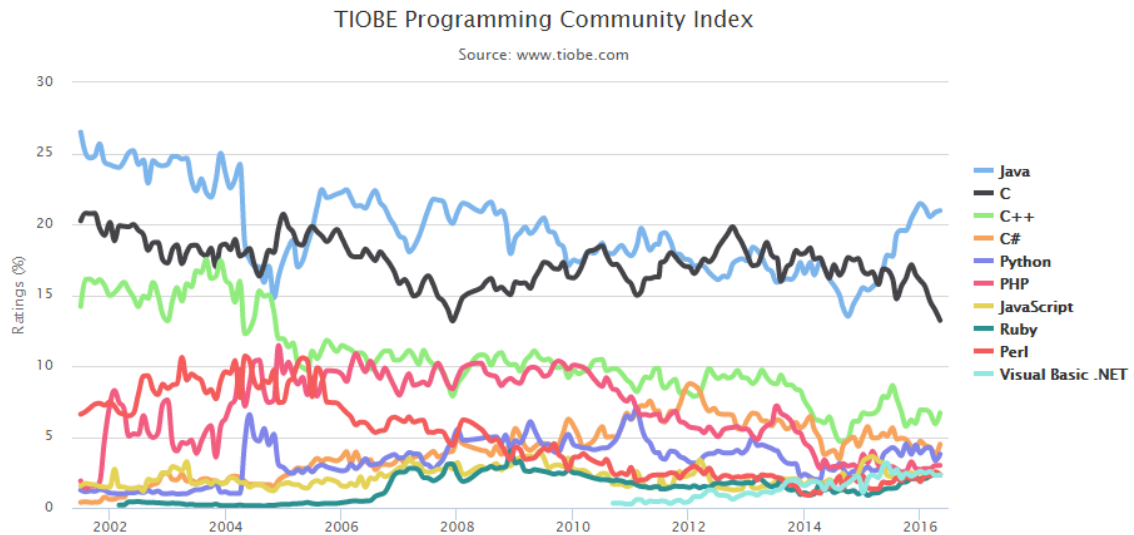


Figura 16. Índice TIOBE en la comunidad de programadores

4.1.2 El *framework* de Spring

Spring es un *framework* de desarrollo y contenedor de inversión de control *open source* para crear aplicaciones Java. La inversión de control es un método de programación en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, siendo las bibliotecas (que generalmente implementan las estructuras de alto nivel) las que invocan al código de usuario (que implementa las tareas a bajo nivel). En junio de 2003 se publicó la primera versión de este *framework* bajo la licencia Apache 2.0. A continuación, se presenta una lista de los principales servicios que proporciona el *framework*.

Acceso a base de datos: permite interactuar gestores de bases de datos que pueden ser relacionales o NoSQL, usando tanto conectores con bajo nivel de abstracción como JDBC o con mayor nivel de abstracción como ORMs.

Gestión de transacciones: unifica distintas APIs de gestión y coordina las transacciones para objetos Java.

Modelo vista-controlador (MVC, Model View Controller): es un *framework* basado en controladores, vistas y el modelo.

Acceso remoto: da soporte al acceso remoto mediante varias tecnologías como pueden ser RMI, CORBA o protocolos basados en HTTP, incluyendo servicios web (SOAP).

Autenticación y autorización: dispone de herramientas para procesos de seguridad que abarcan distintos estándares, protocolos y herramientas. Es denominado *Spring Security*.

Testing: da soporte de clases para el desarrollo de unidades de prueba e integración.

La Figura 17 muestra los módulos de los que está compuesto el entorno de ejecución de Spring.

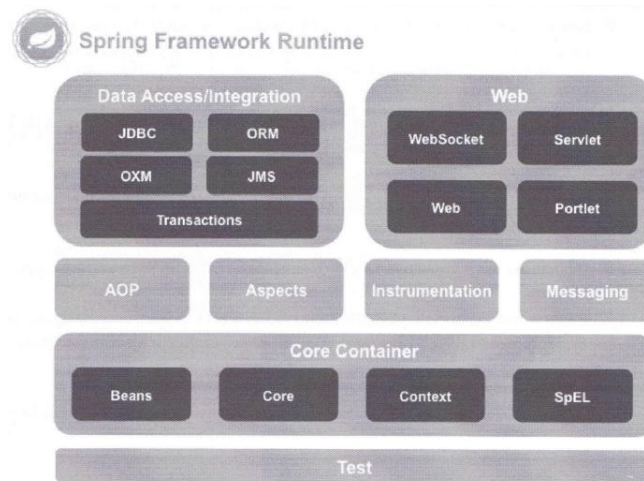


Figura 17. Módulos del entorno de ejecución de Spring

4.1.3 El entorno de desarrollo



Actualmente, una de las formas más cómodas para desarrollar proyectos Spring e incluir las dependencias de librerías externas de forma automática, es empleando Eclipse.

Este entorno ofrece soporte para la creación de proyectos Maven, y se le añadirá el *plugin* Spring Tool Suite (STS) para la creación de nuestro proyecto Spring.

4.1.3.1 Spring Tool Suite



Los conjuntos de herramientas de STS proporcionan combinaciones listas para el uso del soporte de idiomas, soporte de la infraestructura y soporte de ejecución, y los combinan con el existente de Java, Web y herramientas Java EE desde Eclipse [8].

4.1.3.2 JUnit



JUnit es un *framework* de desarrollo de **pruebas unitarias** para Java. Es fácilmente integrable dentro del entorno de desarrollo Eclipse escogido, y permite desarrollar un conjunto de pruebas que verifiquen el funcionamiento del código de forma automática. También es útil para realizar **pruebas de regresión** para comprobar que el funcionamiento del sistema no ha variado tras realizar cambios en el código.

4.1.4 Base de datos MySQL



Se ha escogido MySQL como gestor de Base de Datos relacionales, en lugar de una base de datos NoSQL debido a que el volumen de datos no va a ser excesivamente grande, y se prioriza la integridad de los mismos.

Permite crear, acceder y administrar bases de datos. Es un gestor rápido y robusto que, para una aplicación web que no tiene grandes requerimientos, nos resulta muy adecuado [9].

4.1.5 Amazon Web Services (AWS)



Para el alojamiento de la base de datos del proyecto, se ha empleado AWS. AWS es una colección de servicios web (entre los que se encuentra un servidor de BBDD) que unidos forman una plataforma de computación en la nube [17]. Se ha tomado prestada una máquina en la plataforma de AWS para alojar en él los ficheros necesarios para que la aplicación móvil pueda hacer las consultas necesarias a la base de datos. Dicha máquina, tras ser configurada, cuenta con Apache como servidor web y PHP5 como intérprete de código, para que el servidor pueda atender las peticiones de la aplicación.

4.1.6 Android



Android es un Sistema Operativo basado en Linux, diseñado principalmente para teléfonos móviles y tablets. También se ha abierto camino en el mercado de los relojes inteligentes, televisores e incluso automóviles. Desde su nacimiento, la venta de dispositivos móviles con el sistema Android ha ido aumentando exponencialmente, vendiendo más unidades que Windows Phone e iOS juntos, como se muestra en la Figura 18, con una tasa de mercado superior al 80% [13].

Region	2015* Shipment Volumes	2015* Market Share	2015* YoY Growth
Android	1,161.1	81.2%	9.5%
iOS	226.0	15.8%	17.3%
Windows Phone	31.3	2.2%	-10.2%
Others	11.3	0.8%	-16.8%
TOTAL	1,429.8	100.0%	9.8%

Figura 18. Estadísticas de uso de Android

Dentro de este sistema, existen distintas versiones, desde la 1.0, nacida a finales de 2008 hasta Android Marshmallow (6.0), en octubre de 2015. Tras el lanzamiento de cada versión, éstas son distribuidas según operadoras y móviles de terminal. A fecha del 6 de junio de 2016, la distribución de versiones es la mostrada en la Figura 19 [14].

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.9%
4.1.x	Jelly Bean	16	6.8%
4.2.x		17	9.4%
4.3		18	2.7%
4.4	KitKat	19	31.6%
5.0	Lollipop	21	15.4%
5.1		22	20.0%
6.0	Marshmallow	23	10.1%

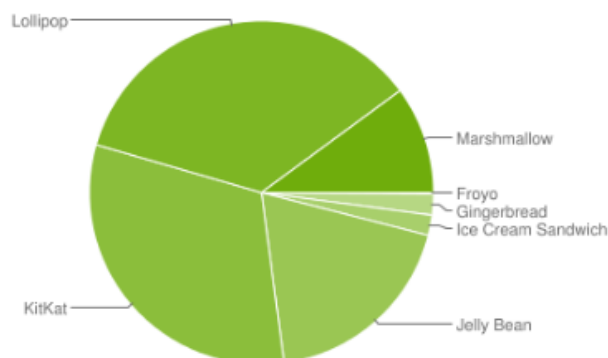


Figura 19. Distribución de las versiones de Android

4.1.7 Android Studio



Android Studio es un entorno de desarrollo para la plataforma Android.

Da soporte para desarrollar aplicaciones para dispositivos móviles, tablets, Android Wear (relojes inteligentes), Android TV y Android Auto (Android para vehículos). Este entorno fue creado para sustituir al entorno de desarrollo que previamente se empleaba para la codificación de aplicaciones Android, que era Eclipse.

4.2 Desarrollo de la aplicación

4.2.1 Desarrollo de la aplicación web

La parte web del proyecto ha sido desarrollada siguiendo el modelo de arquitectura Modelo-Vista-Controlador (MVC). Este tipo de aplicaciones se separan en tres componentes: el modelo, la vista y el controlador, siendo así más fácil su actualización y mantenimiento futuro, ya que se pueden implementar nuevos módulos sin modificar el código desarrollado previamente. En el punto 3.2, en el que se ha definido el diagrama de clases de la aplicación, se ve reflejado esto. A continuación, se definen los componentes de este modelo, y su implementación en el proyecto:

Modelo: El modelo es la representación de la información que el sistema maneja, y gestiona el acceso a dicha información. El controlador realiza las peticiones de acceso a los datos y dichos datos son presentados a través de las vistas al usuario.

Vista: La vista es la presentación de los datos del modelo de modo que el usuario pueda interactuar con ellos.

Controlador: El controlador encapsula los datos y las funcionalidades. Es el intermediario entre el modelo y la vista, modificando esta última si se presenta algún cambio en el modelo que presenta.

Para el acceso a los datos, se emplea el patrón DAO (Data Access Object), que es uno de los Diseños Core J2EE. La ventaja del uso de este patrón es que no cambia independientemente de la implementación subyacente para el acceso a los datos (ficheros, BBDD...) [10].

La Figura 20 representa el diagrama de clases en el que se basa este patrón. A continuación, se definen los componentes que lo conforman.

- **Client:** es el objeto que quiere acceder a la fuente de datos para almacenar o consultar datos.
- **DataAccessObject:** abstrae al objeto del modo de acceso a la fuente de datos.
- **DataSource:** representa la fuente de datos.
- **Data:** es un objeto intermedio entre los datos y el DAO empleado para transportar los datos entre una capa y otra.
- **ResultSet:** representa el resultado final del acceso a los datos.

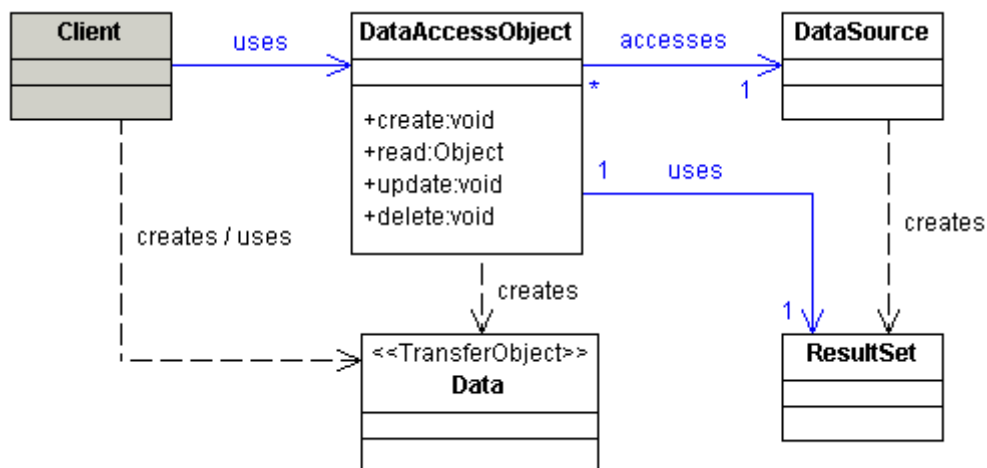


Figura 20. Diagrama de clases. Patrón DAO

En la aplicación, las funcionalidades principales son CRUD (Create-Read-Update-Delete), es decir, la creación, visualización, actualización y eliminación de los datos.

A continuación, se presentan algunas capturas de pantallas de la aplicación a través de las cuales se facilita el acceso a estas funcionalidades.

Creación: las funciones de creación (inscripción de niños, alta de actividad, creación de grupo...), presentan al usuario un formulario adecuado al tipo de objeto que se vaya a

registrar. La Figura 21 presenta una primera parte del formulario de creación actividades, y la Figura 22 muestra un mensaje del sistema en el que se indica que la actividad se ha creado correctamente, junto a las opciones de crear una nueva, o finalizar.

Figura 21. Formulario de creación de actividades

Figura 22. Mensaje de creación de actividad correcto

Visualización: la siguiente función es la de visualización de los datos. En la Figura 23 se muestra la página de visualización de los datos de la actividad seleccionada, “Linternazo”.

Actualización: al ejecutar las funciones de actualización, la aplicación accede a la base de datos para obtener el objeto seleccionado, y muestra una vista similar a la de creación, con la diferencia de que los campos están completos con los datos previos. La Figura 24 muestra los datos asociados a un grupo, el de “Pequeños 1”, con los monitores y niños asignados al grupo, así como los que están disponibles para ser añadidos a dicho grupo.

Visor de actividades

Actividad: Linternazo	
Tipo: Vélada	Responsable: Lucia Mendoza
Día y hora: 6/07/16 22:00	Lugar: Exteriores
Objetivos: Fomentar el compañerismo	
Descripción: Todos deberán coger sacos, esterillas, ropa...utensilios para poder esconderse y ocultar su identidad. Los monitores son los que la ligan, y es como un escondite inglés. Los acampados deben llegar hasta donde están los monitores sin ser reconocidos. Los monitores contarán con la ayuda de la luz de una linterna.	
Organización y equipos: Se juega individualmente. Los monitores deberán averiguar la identidad de los acampados.	
Material: Linternas. Sacos. Ropa. Esterillas...	
Otras cosillas de interes (variantes, precauciones...): Indicar a los niños que se pueden intercambiar la ropa y demás utensilios, pero que todo debe ser devuelto sin excepción.	

[Volver](#)

© Sistema de gestion de campamentos. 2016. Mendoza Garcia, Borja

Figura 23. Visor de actividades

Página de gestion de grupos

Desde esta página puede añadir/eliminar niños y monitores al grupo seleccionado

Nombre del grupo:	<input type="text" value="Pequeños 1"/>		
Monitores asignados:	Monitores sin grupo	Niños asignados:	Niños sin grupo:
Laura Garcia Eliminar	<input type="checkbox"/> Borja Mendoza	Silvia Jimenez Eliminar	<input type="checkbox"/> Irene Jimenez
Alicia Serrano Eliminar	<input type="checkbox"/> Lucia Mendoza		
	<input type="checkbox"/> Rocio Garcia		
	<input type="checkbox"/> Alberto de la Vega		
	<input type="checkbox"/> Victoria Manzano		
	<input type="checkbox"/> Enrique Candelas		
	<input type="checkbox"/> Juan Serrano		
	<input type="checkbox"/> Leticia del Valle		
	<input type="checkbox"/> Sergio Rodriguez		
	<input type="checkbox"/> Elvira Gonzalez		
	<input type="checkbox"/> Nicolas Montoro		
	<input type="checkbox"/> Mario W		

[Volver](#)

[Guardar y continuar](#)

© Sistema de gestion de campamentos. 2016. Mendoza Garcia, Borja

Figura 24. Edición de grupos

Eliminación: la última de las acciones disponibles dentro del CRUD, se refiere a la eliminación del registro de un objeto en la base de datos. La Figura 25 muestra la página de administración de tareas, en la que, seleccionando el botón de *Eliminar*, se borrará el registro de base de datos asociado a la tarea escogida.

Administración de tareas

En este panel se muestran las distintas acciones disponibles para la administración de tareas

Listado de tareas

Nueva tarea

Ver cuadrante de tareas






Tarea			
BAÑOS_EXTERIORES	04-07-2016	1	
BAÑOS_INTERIORES	05-07-2016	1	
SALON_GRANDE	06-07-2016	1	
LIMPIEZA_EXTERIORES	03-07-2016	1	
PORCHE_ESCALERAS	07-07-2016	1	
PERIODICO	09-07-2016	1	

Figura 25. Administración de tareas. Eliminación

Además de las funciones CRUD, dentro de la aplicación existen también otras funcionalidades que se describen a continuación.

Encriptación de datos: tras el registro de un usuario en la plataforma, sus datos quedan almacenados en la base de datos, quedando la contraseña visible en caso de conseguir acceso externo. Para evitar esto, se ha implementado un sistema de cifrado basado en el algoritmo AES. AES (Advanced Encryption Standard) es un algoritmo de cifrado por bloques. Tiene un tamaño fijo de bloque de 128 bits, y tamaños de clave de 128, 196 o 256 bits. Una de las ventajas de este algoritmo sobre otros algoritmos de cifrado, como puede ser MD5, es que permite el almacenamiento del texto cifrado para posteriormente descifrarlo de nuevo de ser necesario. Esto resulta provechoso para el proyecto, para permitir que el usuario pueda recuperar su contraseña en caso de olvidarla [4].

Para las funciones de cifrado/descifrado, se declara la llave de 16, 32 o 64 bytes con la que se vayan a cifrar las cadenas, y en cada una de las funciones, se obtiene la instancia del algoritmo a emplear, iniciándola en modo encriptación o desencriptación según corresponda. La Tabla 1 muestra esta implementación.

Generación de documentos: para la generación de documentos en PDF, Java cuenta con la librería iText, que facilita esta labor. *iText* es una librería *open source* para la creación de documentos. Para su uso, se añade el código reflejado en la Tabla 2 en el fichero *pom.xml* del proyecto.

Definición de datos	<pre>private static final String ALGO = "AES"; private static final byte[] keyValue = new byte[] { 'T', 'F', 'G', 'S', 'i', ' ', 's', 't', 'G', 'e', 's', 't', 'C', 'a', 'm', 'p', 's'};</pre>
Cifrado	<pre>public static String encrypt(String data) throws Exception { Key key = generateKey(); Cipher c = Cipher.getInstance(ALGO); c.init(Cipher.ENCRYPT_MODE, key); byte[] encVal = c.doFinal(data.getBytes()); String encryptedValue = new BASE64Encoder().encode(encVal); return encryptedValue; }</pre>
Descifrado	<pre>public static String decrypt(String encryptedData) throws Exception { Key key = generateKey(); Cipher c = Cipher.getInstance(ALGO); c.init(Cipher.DECRYPT_MODE, key); byte[] decodedValue = new BASE64Decoder().decodeBuffer(encryptedData); byte[] decValue = c.doFinal(decodedValue); String decryptedValue = new String(decValue); return decryptedValue; }</pre>

Tabla 1. Implementación de la seguridad de datos

```
<dependency>
  <groupId>com.lowagie</groupId>
  <artifactId>itext</artifactId>
  <version>2.1.7</version>
</dependency>
```

Tabla 2. Dependencia librería iText

En la aplicación, se ha creado una función que, dado un conjunto de datos recibidos a través del controlador, genera un documento con el texto con el formato que haya definido el programador en el código del modelo. En el caso de este proyecto, genera un PDF de autorización para la inscripción del niño que previamente ha sido dado de alta en la plataforma, como se refleja en la Figura 26.

Envío de correos electrónicos: los proyectos Spring permiten integrar de forma sencilla el envío de correos electrónicos. Basta con crear un *bean* en el fichero de configuración (application-config.xml) del proyecto, y crear un controlador para que, una vez el usuario ejecute la acción de envío desde la vista, recoja los datos introducidos, los procese, y envíe el mensaje.

Ficha de inscripción

1								
Apellidos: Jimenez Garcia Nombre: Silvia Fecha de nacimiento: 22-12-2004 Nombres de los padres: Ricardo y M ^a Antonia Telefonos de los padres: 917642386 - 912345678 (padre) 912451267 (madre) Email de los padres: elrichilin@hotmail.com (padre) tonign@gmail.com (madre) Direccion: Calle/Avd: Ejidillo N ^o : 11 Piso: Letra: Poblacion: Colmenar Viejo C.P: 28770								
Autorizacion de asistencia al campamento de verano								
<table border="1"> <tr> <td>Nombre y apellidos del padre/madre/tutor</td> <td>Ricardo Jimenez M^a Antonia Garcia</td> </tr> <tr> <td>DNI</td> <td>70082012T (padre) 70082013T (madre)</td> </tr> <tr> <td>Domicilio de localizacion</td> <td>Ejidillo N^o: 11 Piso: Letra:</td> </tr> <tr> <td>Telefonos de localizacion</td> <td>917642386 - 912345678 (padre) 912451267 (madre)</td> </tr> </table>	Nombre y apellidos del padre/madre/tutor	Ricardo Jimenez M ^a Antonia Garcia	DNI	70082012T (padre) 70082013T (madre)	Domicilio de localizacion	Ejidillo N ^o : 11 Piso: Letra:	Telefonos de localizacion	917642386 - 912345678 (padre) 912451267 (madre)
Nombre y apellidos del padre/madre/tutor	Ricardo Jimenez M ^a Antonia Garcia							
DNI	70082012T (padre) 70082013T (madre)							
Domicilio de localizacion	Ejidillo N ^o : 11 Piso: Letra:							
Telefonos de localizacion	917642386 - 912345678 (padre) 912451267 (madre)							

Figura 26. Generación de documentos PDF

El fichero de configuración será similar al mostrado en la Figura 27, introduciendo la cuenta de correo y contraseña desde la que se quiera enviar los mensajes. En la Figura 28 se muestra la recepción correcta de un correo electrónico enviado a múltiples destinatarios desde la web.

```

<bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">
  <property name="host" value="smtp.gmail.com" />
  <property name="port" value="587" />
  <property name="protocol" value="smtp"></property>
  <property name="username" value="usertfggestioncampamentos@gmail.com" />
  <property name="password" value="██████████" />
  <property name="javaMailProperties">
    <props>
      <prop key="mail.smtp.auth">true</prop>
      <prop key="mail.smtp.starttls.enable">true</prop>
      <prop key="mail.debug">true</prop>
    </props>
  </property>
</bean>

```

Figura 27. Fichero application-config.xml – Conexión correo Gmail

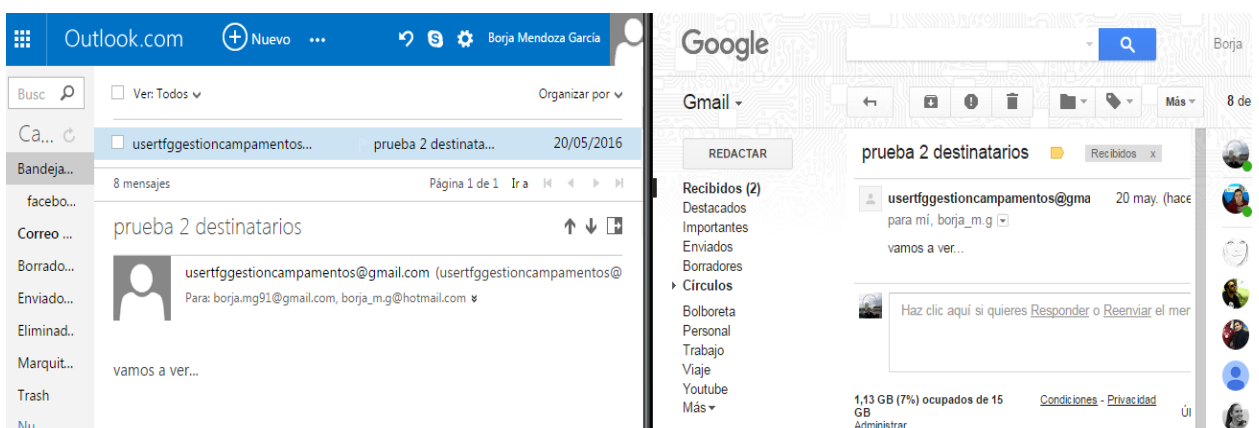


Figura 28. Envío de correo electrónico

4.2.2 Desarrollo de la aplicación Android

Para el desarrollo de una aplicación Android, existen una serie de elementos clave. Estos elementos son [16]:

- **Vista (View):** son los elementos que componen la interfaz de la aplicación. Son, por ejemplo, un botón, una entrada de texto, etc. Lo habitual es definir estas vistas utilizando un fichero XML y dejar que el sistema cree los objetos a partir de dicho fichero, pero también pueden ser creados manualmente a partir de código Java, ya que todos estos objetos extienden de la clase *View*.
- **Layout:** es un conjunto de vistas agrupadas de una determinada forma.
- **Actividad (Activity):** cada una de las pantallas de una aplicación Android es conocido como *actividad*. Su función es la creación de la interfaz de usuario.
- **Intención (Intent):** representa la voluntad de realizar alguna acción. Se utiliza cada vez que se quiere lanzar una actividad.

La funcionalidad principal de esta aplicación Android es facilitar el acceso a los datos que previamente se han almacenado en el servidor mediante la aplicación web desarrollada en este mismo proyecto, de forma que los monitores tengan acceso a toda la información del campamento de una forma rápida y ordenada.

Para permitir que la aplicación tenga acceso a Internet y pueda acceder a los datos, en el fichero *AndroidManifest.xml* se tiene que indicar:

```
<uses-permission android:name="android.permission.INTERNET" />
```

El acceso a los datos se realizará a través de ficheros PHP, que se encuentran subidos al servidor, y que devuelven los datos solicitados desde la aplicación, siguiendo el sistema mostrado en la Figura 29.

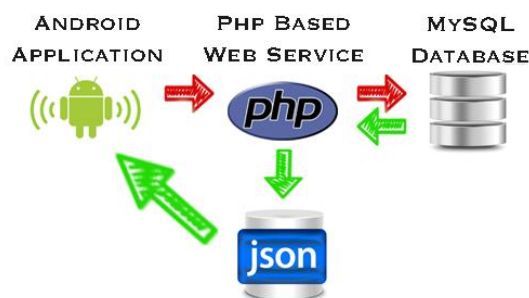


Figura 29. Conexión aplicación Android y MySQL

Por ejemplo, para realizar el login en la aplicación, la actividad encargada de mostrar esta página, contará con las vistas necesarias para el ingreso de los datos por parte del usuario, y un botón que ejecutará la intención de acceso, como se muestra en la Figura 30.



Figura 30. Pantalla de acceso a la app Android

Al ejecutarse la acción de login, se ejecuta una tarea asíncrona, que conecta con el servidor, y ejecuta un fichero PHP que se encuentra allí almacenado. Esta acción queda reflejada en la Figura 31.

```
private static final String LOGIN_URL = "http://www.SchoolPipe.com/TFG/login.php";
List params = new ArrayList();
params.add(new BasicNameValuePair("username", username));
params.add(new BasicNameValuePair("password", password));
JSONObject json = jsonParser.makeHttpRequest(LOGIN_URL, "POST", params);
```

Figura 31. Conexión Android con BBDD MySQL

El método *makeHttpRequest* de la clase *JSONParser*, recibe la url del fichero dentro del servidor, el método de acceso (GET o POST), y una lista de parámetros. Este método, ejecutará el fichero dentro del servidor, y se obtendrá la respuesta en la variable *json* de tipo *JSONObject*. De esta variable, obtendremos los datos deseados, accediendo a ellos a partir de unas etiquetas, que serán los nombres con los que se ha almacenado la respuesta en el response del fichero PHP.

Para el caso de la visualización de los listados de niños y actividades, primero se obtendrá un listado de todos los elementos de dicho tipo, y se mostrarán a través de una vista de tipo *ListView*, que mostrará, en forma de botones, todos los datos obtenidos de la BBDD. Este listado estará representado en forma de botones, que, una vez pulsados, volverán a hacer una consulta a la BBDD para obtener los datos del niño o actividad seleccionados. La visualización de niños y actividades es un *layout* de tipo *scroll* de modo que pueda visualizarse toda la información completa desplazando la pantalla de forma vertical.

En la Figura 32 se muestra un ejemplo de las pantallas correspondientes a la visualización de actividades (en concreto, la actividad “Juegos en la oscuridad”).



Figura 32. Listado y visualización de actividades

5 Pruebas y resultados

En este apartado se especificarán las pruebas de validación realizadas.

Al ser una aplicación orientada a usuarios finales, una parte muy importante de esta fase ha sido la realización de pruebas con usuarios reales, a fin de validar tanto el funcionamiento correcto del sistema, como la satisfacción del usuario con respecto a la funcionalidad ofrecida por la aplicación y el diseño de la interfaz e interacción llevado a cabo.

5.1 Inspecciones de código

Durante todo el desarrollo del proyecto, y durante la fase de pruebas, se han realizado inspecciones generales de código, con el fin de encontrar bloques de códigos o poco optimizados, pudiendo así ser eliminados o modificados para obtener un código más limpio y eficiente. Con este tipo de pruebas también se han detectado errores en el código.

5.2 Pruebas unitarias

Antes de iniciar la codificación, se planificó un conjunto de pruebas unitarias que probaran todas y cada una de las funcionalidades descritas en el catálogo de requisitos, una por una, y a través de las cuales se pudiera verificar el correcto funcionamiento del sistema antes, durante y tras el desarrollo. No se enumeran aquí por motivos de espacio. Son, básicamente, una prueba por cada método implementado en cada clase, teniendo en cuenta todas las posibles casuísticas, realizando pruebas con casos normales con valores correctos conocidos, pruebas con datos erróneos, pruebas ilegales... Estas pruebas se realizaron a través del *framework* *jUnit*, ya descrito previamente.

5.3 Pruebas de compatibilidad

Tras el desarrollo de la aplicación, se han realizado pruebas en los distintos navegadores mencionados en el punto 2.2.2, para verificar que ni la funcionalidad ni la interfaz dan lugar a errores independientemente del explorador utilizado. La Figura 41 del Anexo E muestra la interfaz vista a través de Google Chrome. La Figura 42 la muestra en Mozilla Firefox y la Figura 43 en Internet Explorer.

5.4 Pruebas de usuario

Dado que la aplicación está orientada a usuarios finales, se han realizado pruebas con un conjunto de coordinadores, monitores y padres involucrados en anteriores campamentos.

Al ser usuarios con conocimientos acerca del contenido de la aplicación, no ha sido necesaria una explicación extensa acerca de los campamentos y de cómo se organizan, ni

tampoco de las pruebas a realizar, cuyo contenido entendieron con rapidez. Previamente a la realización de las mismas, se les proporcionó un formulario de satisfacción para recoger sus opiniones y así poder obtener información y elaborar estadísticas de satisfacción al finalizar. A continuación se describen algunos aspectos del cuestionario, que está disponible completo en el Anexo F. El cuestionario está basado en una escala de Likert con cinco niveles de respuesta:

1: Totalmente en desacuerdo	2: Más bien en desacuerdo	3: Ni de acuerdo ni en desacuerdo	4: De acuerdo	5: Totalmente de acuerdo
------------------------------------	----------------------------------	--	----------------------	---------------------------------

Se han realizado pruebas con un total de 10 usuarios finales, de los cuales 1 tiene el perfil de coordinador, 6 son monitores, y 3 son usuarios públicos. Los resultados obtenidos son los siguientes, divididos por bloques de preguntas, según el tipo de usuario con el que accede a la aplicación:

En el primer bloque de preguntas, recogido en la Tabla 3, los usuarios principales son usuarios públicos y coordinadores del sistema, encargados del registro de niños en la plataforma. Como se puede observar dicha tabla, la funcionalidad peor valorada la relativa al alta de usuarios es el envío de correos electrónicos, que dos usuarios han puntuado con un 3; el mayor problema fue encontrar la opción de envío, no realizar el envío en sí.

Pregunta	NS/NC	1	2	3	4	5
El registro en la plataforma es sencillo					33.3%	66.6%
Se accede fácilmente a la plataforma					10%	90%
La comunicación con otros usuarios es sencilla (envío de correos)				20%	70%	10%
Puedo modificar mis datos de forma sencilla					30%	70%
Puedo registrar un nuevo niño en el campamento fácilmente					25%	75%
Puedo visualizar los datos de un niño fácilmente					10%	90%
Puedo editar los datos de un niño fácilmente					25%	75%
Puedo eliminar los datos de un niño fácilmente					25%	75%
Genero el documento PDF de forma sencilla						100%

Tabla 3. Pruebas de usuario. Funcionalidad. Bloque 1

El bloque 2 está centrado en las acciones propias del grupo de usuarios con perfil de monitor y coordinador. En este caso el grado de satisfacción de los usuarios es elevado. La Tabla 4 muestra los porcentajes de satisfacción para cada elemento valorado.

Pregunta	NS/NC	1	2	3	4	5
Puedo registrar una nueva actividad en el campamento fácilmente					14.3%	85.7%
Puedo visualizar los datos de una actividad fácilmente						100%
Puedo editar los datos de una actividad fácilmente						100%
Puedo eliminar los datos de una actividad fácilmente					14.3%	85.7%
La visualización del grupo que tengo asignado es sencilla	14.3%				14.3%	71.4%
La visualización del cuadrante de tareas es sencilla					14.3%	85.7%
La visualización del cuadrante de guardias es sencilla					14.3%	85.7%

Tabla 4. Pruebas de usuario. Funcionalidad. Bloque 2

Por último, en el bloque 3 de preguntas acerca de la funcionalidad del sistema sólo participa el coordinador. Como se puede observar a partir de los datos recogidos en la Tabla 5, el usuario tiene un alto grado de satisfacción. El cuestionario también contiene varias preguntas sobre la interfaz e interacción de la aplicación, recogidas en la Tabla 6.

Pregunta	NS/NC	1	2	3	4	5
La administración de grupos se realiza de forma sencilla						100%
La administración de tareas se realiza de forma sencilla						100%
La administración de guardias se realiza de forma sencilla						100%

Tabla 5. Pruebas de usuario. Funcionalidad. Bloque 3

Pregunta	NS/NC	1	2	3	4	5
La apariencia de la aplicación es agradable				40%	30%	30%
La interfaz de la aplicación (letras, colores, nº de elementos en la pantalla) cansa la vista		80%	20%			
Se puede contactar con el administrador de forma sencilla					50%	50%
Puedo visualizar la información de mi cuenta de forma sencilla				10%	20%	70%
Los mensajes de ayuda/error son precisos				10%	10%	80%
El modo de realizar las acciones que deseo en la aplicación es intuitivo					20%	80%
Encuentro con rapidez cómo realizar las acciones que deseo				10%	40%	50%
Orientarse dentro de la aplicación es sencillo					50%	50%

Tabla 6. Pruebas de usuario. Interfaz.

Cabe mencionar que la respuesta para la pregunta “La interfaz de la aplicación (letras, colores, cantidad de elementos en la pantalla) cansa la vista”, se valorará internamente con un 5 cuando la respuesta sea “Totalmente en desacuerdo” y con 1 cuando sea “Totalmente de acuerdo”. Esto es debido a que se trata de una sentencia negativa. Como se puede observar en la Tabla 6, la mayoría de los usuarios valoran los aspectos relativos a la

interfaz e interacción muy bien o bien. Finalmente, los usuarios pudieron realizar valoraciones adicionales, respondiendo a dos preguntas sobre los aspectos más positivos y los negativos que destacarían. El resumen de las valoraciones emitidas por los 10 usuarios que realizaron las pruebas es:

¿Destacarías como positivo algún otro aspecto de esta aplicación o de tu interacción con ella? ¿Cuál/es?

- Fácil e intuitiva aún sin tener grandes conocimientos del uso de Internet.
- La interfaz. Que se distingan los botones por colores según las acciones que realicen.
- Es muy intuitivo en cuanto a tener control sobre los datos.
- La puesta en común de todos los datos (niños, actividades, grupos...etc.) antes del inicio del campamento facilita la organización del mismo para los monitores.
- Al mantenerse el formato actual de los formularios de inscripción y de alta de actividades, el cambio resulta más sencillo.
- Es muy cómodo poder editar las actividades en el momento sin tener que volver a hacer un nuevo documento y enviarlo al coordinador de nuevo.

¿Encuentras mejorable algún aspecto de esta aplicación o de tu interacción con ella? ¿Cuál/es?

- En la parte de información y contacto añadir también los datos de las personas responsables del campamento.
- La barra de navegación parece ajena a la web.
- Se sobreentiende que los niños tienen padre y madre y eso no corresponde a la realidad social. Sería recomendable poner “tutor/a” y que sea obligatorio sólo uno de los dos.
- Al eliminar un niño/actividad, pedir confirmación.
- La apariencia de la aplicación queda “sosa”.

5.5 Pruebas de regresión

Tras la realización de todas las pruebas planificadas, con su consiguiente identificación de errores, éstos han sido subsanados dentro de la aplicación. Una vez se han realizado estas correcciones, se han vuelto a ejecutar todas las pruebas previstas en el plan de pruebas para comprobar tanto que los errores localizados en la iteración anterior habían sido correctamente corregidos, como para verificar que no se han generado nuevos. Para la realización de estas pruebas, vuelve a ser de gran ayuda el banco de pruebas unitarias que se había creado previamente.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

El objetivo de este proyecto era el desarrollo de un sistema que facilitara las tareas de gestión de un campamento. El resultado del trabajo realizado ha ido un paso más allá de la idea inicial que se originó en el campamento del verano pasado, cubriendo la mayor parte del trabajo necesario para aliviar la carga que supone la realización de las tareas manualmente. Aun así, existen aún tareas automatizables que pueden ser implementadas en el sistema y se comentarán con algo más de detalle en el contexto del trabajo futuro.

Se ha desarrollado un sistema capaz de gestionar todos los datos necesarios para el inicio de un campamento, y una aplicación móvil que permite a los monitores, en primer lugar, el acceso a toda la información relacionada con el campamento de una forma rápida y sencilla desde un dispositivo móvil, evitando el tener que portar el dossier de documentación completo de un lugar a otro durante el desarrollo del mismo. Además, permite la rápida actualización de las actividades a realizar directamente desde el mismo dispositivo, actualizando así de forma inmediata dicha información para el resto de monitores y permitiendo de este modo adaptar las actividades a la dinámica de los grupos y al ritmo del campamento en general.

Las aplicaciones desarrolladas incluyen toda la funcionalidad definida en el catálogo de requisitos, con una interfaz sencilla, intuitiva y accesible a todo tipo de usuarios, independientemente de sus conocimientos técnicos. El desarrollo del proyecto ha sido incremental, asegurando el correcto funcionamiento de un módulo funcional completo antes del inicio del siguiente, siendo así más fácil la realización de planes de pruebas unitarias y de regresión completos. Se ha diseñado de forma modular, para facilitar la incorporación de nuevas funcionalidades con la realización de los mínimos cambios en el resto del sistema. De igual modo, se ha intentado que sea lo más reutilizable posible para otros tipos de sistemas, como pudiera ser la gestión de una residencia de ancianos, en el que los auxiliares pudieran tener los datos de los residentes en el teléfono y no tener que cargar con carpetas de informes.

El poder tener acceso a usuarios finales durante todo el proyecto ha supuesto un beneficio añadido tanto a la hora de identificar y validar los requisitos de las aplicaciones, como en el momento de realizar pruebas reales con las personas a las que va dirigido el proyecto.

Tanto el autor de esta memoria, como diseñador y desarrollador de las aplicaciones creadas, como los demás monitores del campamento, estamos contentos con el resultado final obtenido, conscientes de los beneficios que este tipo de aplicaciones aportan dentro de un entorno en el que una buena organización reduce mucho el número de errores humanos y minimiza el estrés, contribuyendo a generar un ambiente mucho más agradable, que finalmente repercute en un campamento en el que todos, tanto niños como monitores y coordinadores, disfruten más.

6.2 Trabajo futuro

Tras la finalización de este proyecto, se abren varias líneas de trabajo futuro. Éstas son:

- Generación de alertas configurables por el usuario en la aplicación móvil para el recordatorio de actividades, guardias, tareas, u horas en las que los niños asignados en el grupo deben tomar medicinas, en su caso.
- Revisión de la propuesta por parte del equipo de cocina para tratar de automatizar la generación de menús, y creación de un nuevo perfil en la aplicación móvil que les genere las respectivas alertas en caso de alergias.
- Realización de pruebas con más monitores y posible implementación de nuevas funcionalidades o mejoras, en su caso, conforme a los resultados de dichas pruebas.
- Negociación con monitores y coordinadores para poner en marcha la utilización de la aplicación en el campamento de este verano de 2016.
- Ampliación de la implementación de la aplicación móvil al sistema operativo iOS, de modo que los usuarios de estos terminales tengan acceso a las mismas herramientas que los usuarios Android.

Referencias

- [1] Eugenia Pérez, “Desarrollo de aplicaciones mediante el Framework de Spring”, ed. Rama, 2015
- [2] Mark Otto, Jacob Thornton, “Bootstrap 3, el manual oficial”, LibrosWeb.es, 2016
Disponible en: http://librosweb.es/libro/bootstrap_3/ [Último acceso: 29/06/16]
- [3] Francisco Grimaldo, “Desarrollando una aplicación Spring Framework MVC v3 + JPA paso a paso”. Disponible en:
<http://www.uv.es/grimo/teaching/SpringMVCv3PasoAPaso/part1.html> [Último acceso: 29/06/16]
- [4] Farham Khwaja, “Java: Encryption and Decryption of Data using AES algorithm”, 2011. Disponible en: <http://www.code2learn.com/2011/06/encryption-and-decryption-of-data-using.html> [Último acceso: 29/06/16]
- [5] “Spring Web MVC with PDF View Example (using iText 5.x)”, Codejava.net, 2016.
Disponible en: <http://www.codejava.net/frameworks/spring/spring-web-mvc-with-pdf-view-example-using-itext-5x> [Último acceso: 29/06/16]
- [6] “TIOBE index”. TIOBE software BV, 2016. Disponible en:
http://www.tiobe.com/tiobe_index [Último acceso: 29/06/16]
- [7] Pierre Carbone, “PYPL PopularitY of Programming Language”. Creative Commons Attribution 3.0 Unported License. 2016. Disponible en:
<http://pypl.github.io/PYPL.html> [Último acceso: 29/06/16]
- [8] Spring Tool Suite, Pivotal Software, 2016. Disponible en: <https://spring.io/tools>
[Último acceso: 29/06/16]
- [9] MySQL. Disponible en: <https://www.mysql.com/> [Último acceso: 29/06/16]
- [10] Deepak Alur, John Crupi and Dan Malks, “Core J2EE Patterns. Best Practices and Design Strategies”, ed. Prentice Hall, 2nd Edition, 2003.
<http://corej2eepatterns.com/DataAccessObject.htm> [También con contenidos en línea. Último acceso: 29/06/16]
- [11] B. Shneiderman, C. Plaisant, “Diseño de Interfaces de Usuario. Estrategias para una Interacción Persona-Computadora Efectiva”. Pearson Education. Addison-Wesley. 2006
- [12] “Android Studio. The Official IDE for Android”. Disponible en:
<https://developer.android.com/studio/index.html> [Último acceso: 29/06/16]

- [13] Iosu Lerga “Android aplasta a iOS en cuota de mercado este año”. 2016.
<https://www.tutecnomundo.com/android-aplasta-ios-cuota-mercado-este-ano/> [En línea]
- [14] “Estadísticas de Android”. Disponible en:
<https://developer.android.com/about/dashboards/index.html#Platform> [Último acceso: 29/06/16]
- [15] JUnit. Disponible en: <http://junit.org/junit4/> [Último acceso: 29/06/16]
- [16] Jesús Tomás Gironés, “El gran libro de Android”, ed. Marcombo, 4ª edición, 2014
- [17] Amazon Web Services (AWS) – Servicios de informática en la nube, Amazon Web Services, Inc., 2006. Disponible en: <http://aws.amazon.com/es/about-aws/> [Último acceso: 29/06/16]

Glosario

ANDROID	Sistema Operativo basado en Linux.
API	(A pplication P rogramming I nterface) Interfaz de programación de aplicaciones.
BOOTSTRAP	Es un framework para crear interfaces web con JavaScript y CSS.
ECLIPSE	Entorno de programación
FRAMEWORK	Estructura conceptual y tecnológica de soporte definido que sirve de base para la organización y desarrollo de software.
JAVA	Lenguaje de programación orientado a objetos empleado en el proyecto.
JAVABEANS	Modelo de componentes para el lenguaje java que permite utilizar los objetos como parte de la base de datos.
JSON	(J ava S cript O bject N otation) Formato ligero para el intercambio de datos.
JSP	(J ava S erver P ages) Tecnología orientada a la creación de páginas web dinámicas.
MVC	(M odel- V iew- C ontroller) Patrón de arquitectura software.
MySQL	Sistema de gestión de base de datos de libre distribución.
SPRING	Framework para el desarrollo de aplicaciones en Java.
PHP	(P HP H ypertext P re-processor) Lenguaje de programación originalmente diseñado para el desarrollo web de contenido dinámico.

Anexos

A Diseño completo de la Base de Datos

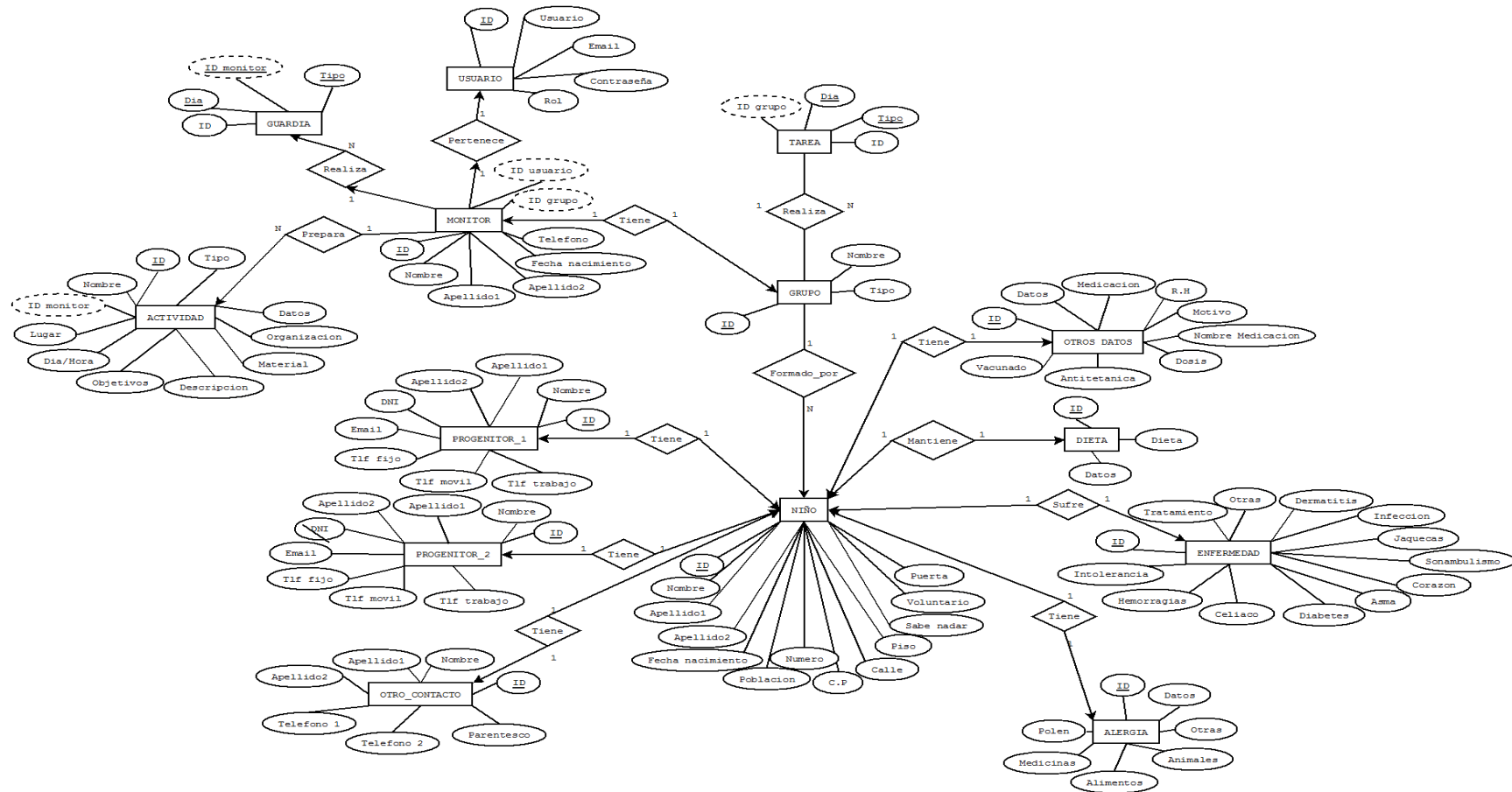


Figura 33. Diagrama E/R completo

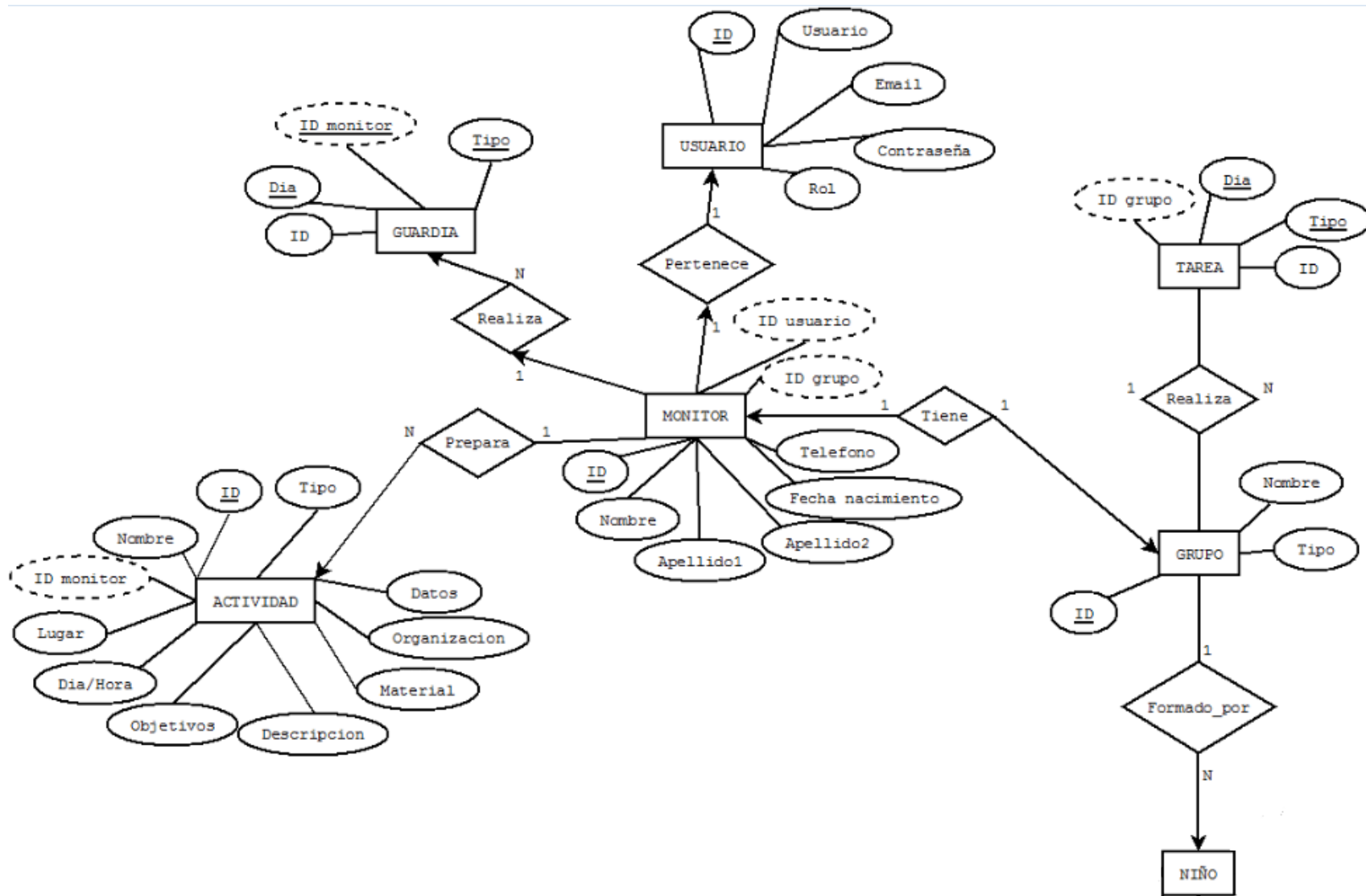


Figura 34. Diagrama de clases completo. Parte 1

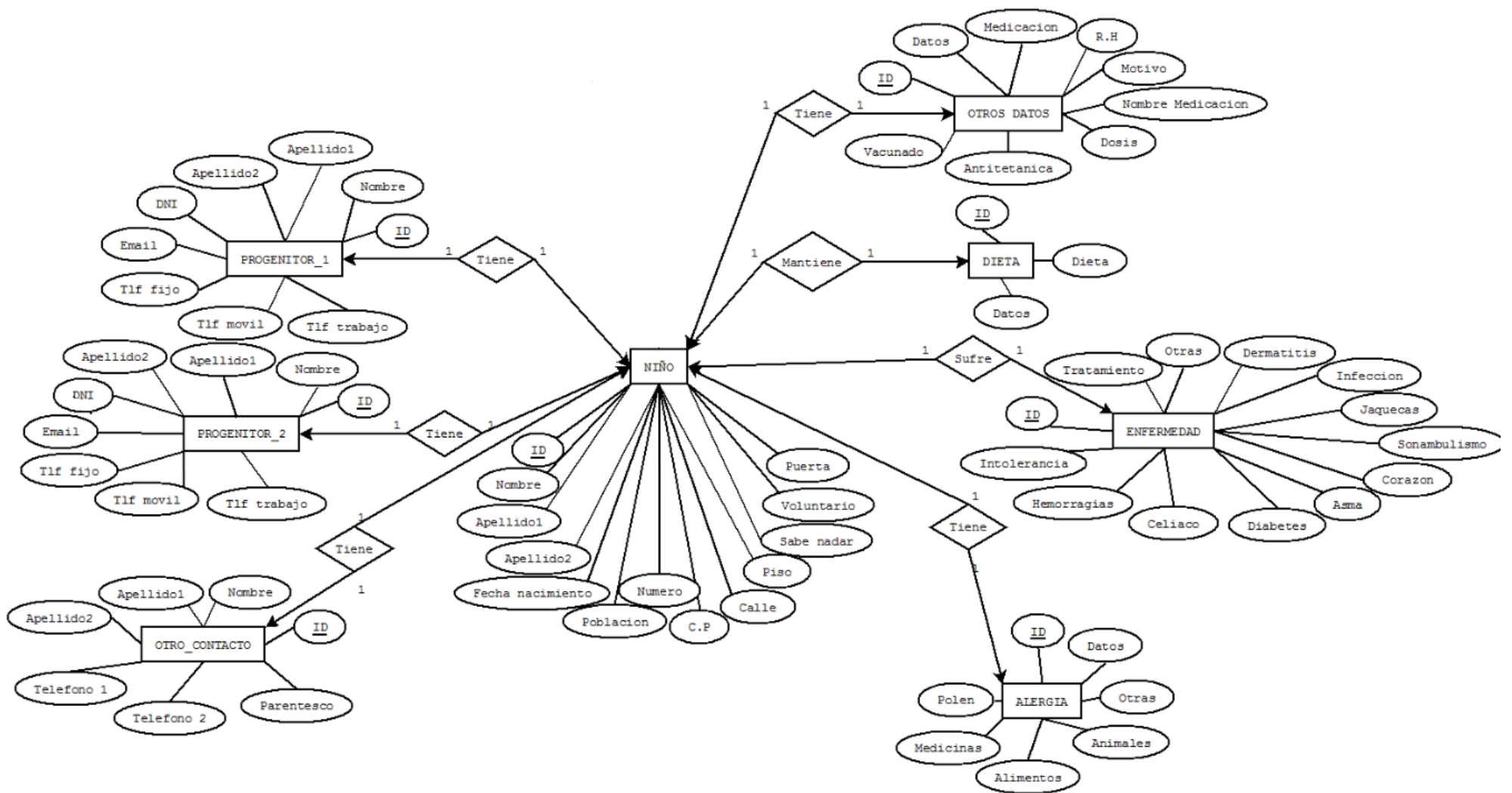


Figura 35. Diagrama de clases completo. Parte 2

B Esquema de tablas completo

Tabla	User
Descripción	Esta tabla almacena los datos referentes a los usuarios que están dados de alta y tienen acceso a la aplicación.
Atributos	<ul style="list-style-type: none">• id: el identificador de usuario• username: el nombre de usuario de acceso• email: correo electrónico del usuario• password: la contraseña de acceso• rol: tipo de usuario dentro de la aplicación

Tabla 7. Tabla User

Tabla	Monitor
Descripción	Esta tabla almacena los datos referentes a los monitores
Atributos	<ul style="list-style-type: none">• id: el identificador de monitor• first_name: nombre del monitor• last_name1: primer apellido del monitor• last_name2: segundo apellido del monitor• birth_date: fecha de nacimiento• phone: número de teléfono• user_id: identificador dentro de la tabla User• group_id: identificador del grupo asignado

Tabla 8. Tabla Monitor

Tabla	Kid
Descripción	Esta tabla almacena los datos referentes a los niños
Atributos	<ul style="list-style-type: none">• id: el identificador de niño• first_name: nombre del niño• last_name1: primer apellido del niño• last_name2: segundo apellido del niño• birth_date: fecha de nacimiento

	<ul style="list-style-type: none"> • address: dirección del domicilio • door_number: número del domicilio • floor: piso • door: letra o número de la puerta • post_code: código postal del domicilio • city: población • swim: booleano que indica si el niño sabe nadar • volunteer: booleano que indica si el niño asiste voluntariamente al campamento • father_id: identificador del padre • mother_id: identificador de la madre • contact_id: identificador del contacto auxiliar • allergy_id: identificador de alergia • diet_id: identificador de dieta • disease_id: identificador de enfermedades • extrainfo_id: identificador de los datos médicos • user_id: identificador del usuario que le dio de alta en la plataforma • group_id: identificador del grupo al que está asignado
--	--

Tabla 9. Tabla Kid

Tabla	Father
Descripción	Esta tabla almacena los datos referentes al padre
Atributos	<ul style="list-style-type: none"> • id: el identificador del padre • first_name: nombre del padre • last_name1: primer apellido del padre • last_name2: segundo apellido del padre • dni: DNI del padre • email: correo electrónico del padre • phone1: teléfono de contacto principal del padre

	<ul style="list-style-type: none"> • phone2: teléfono de contacto auxiliar del padre • phone3: teléfono de contacto auxiliar del padre
--	--

Tabla 10. Tabla Father

Tabla	Mother
Descripción	Esta tabla almacena los datos referentes a la madre
Atributos	<ul style="list-style-type: none"> • id: el identificador de la madre • first_name: nombre de la madre • last_name1: primer apellido de la madre • last_name2: segundo apellido de la madre • dni: DNI de la madre • email: correo electrónico de la madre • phone1: teléfono de contacto principal de la madre • phone2: teléfono de contacto auxiliar de la madre • phone3: teléfono de contacto auxiliar de la madre

Tabla 11. Tabla Mother

Tabla	Contact
Descripción	Esta tabla almacena los datos referentes al contacto auxiliar
Atributos	<ul style="list-style-type: none"> • id: el identificador del contacto auxiliar • first_name: nombre del contacto auxiliar • last_name1: primer apellido del contacto auxiliar • last_name2: segundo apellido del contacto auxiliar • phone1: teléfono de contacto principal de la madre • phone2: teléfono de contacto auxiliar de la madre • relationship: parentesco o relación del contacto con el niño

Tabla 12. Tabla Contact

Tabla	Diet
Descripción	Esta tabla almacena los datos referentes a las dietas
Atributos	<ul style="list-style-type: none"> • id: el identificador de la dieta • diet: un booleano que indica si se trata de una dieta normal o especial • info: los datos de la dieta indicados por el usuario que da de alta al niño

Tabla 13. Tabla Diet

Tabla	Allergy
Descripción	Esta tabla almacena los datos referentes a las alergias
Atributos	<ul style="list-style-type: none"> • id: el identificador de la alergia • animals_allergy: un booleano que indica si se tiene alergia a algún animal • food_allergy: un booleano que indica si se tienen alergias alimenticias • meds_allergy: un booleano que indica si se tiene alergia a algún medicamento • pollen_allergy: un booleano que indica si se tiene alergia al polen • other_allergy: un booleano que indica si se tiene alguna otra alergia distinta a las anteriores • info: los datos indicados por el usuario que da de alta al niño en referencia a las alergias marcadas

Tabla 14. Tabla Allergy

Tabla	MedicalData
Descripción	Esta tabla almacena los datos referentes a los datos médicos
Atributos	<ul style="list-style-type: none"> • id: el identificador • blood_type: el RH del niño • fully_vaccinated: booleano que indica si el niño tiene la

	<p>cartilla de vacunación completa</p> <ul style="list-style-type: none"> • need_medication: booleano que indica si el niño precisa tomar medicación durante el campamento • tetanus_vaccine: booleano que indica si el niño tiene puesta la vacuna antitetánica • x_med_doses*: indica la dosis de medicamento necesaria • x_med_name*: indica el nombre del medicamento • x_med_reason*: indica la razón de la medicación • extrainfo: datos relevantes indicados por el usuario
--	--

Tabla 15. Tabla MedicalData

Respecto a esta última tabla, indicar que los campos marcados con un asterisco (*) se encuentran en la base de datos 4 veces, sustituyendo la “x” por los siguientes códigos:

- b: indica que el medicamento se proporcionará en el desayuno
- l: indica que el medicamento se proporcionará en la comida
- d: indica que el medicamento se proporcionará en la cena
- o: indica que el medicamento se proporcionará a otra hora distinta de las anteriores

Tabla	KidsGroup
Descripción	Esta tabla almacena los datos referentes a los grupos
Atributos	<ul style="list-style-type: none"> • id: el identificador del grupo • name: nombre del grupo • teenGroup: un booleano que indica si el grupo es de mayores o pequeños

Tabla 16. Tabla KidsGroup

Tabla	Activity
Descripción	Esta tabla almacena los datos referentes a las actividades
Atributos	<ul style="list-style-type: none"> • id: el identificador de actividad • name: nombre de la actividad • type: tipo de actividad

	<ul style="list-style-type: none"> • responsable: identificador del monitor responsable • datetime: día y hora de la actividad • place: lugar de la actividad • objective: objetivos de la actividad • description: descripción de la actividad • organization: organización y equipos de la actividad • material: material de la actividad • extrainfo: otras consideraciones indicadas por el monitor
--	---

Tabla 17. Tabla Activity

Tabla	Task
Descripción	Esta tabla almacena los datos referentes a las tareas
Atributos	<ul style="list-style-type: none"> • id: el identificador de tarea • task_type: tipo de tarea • task_day: día de la tarea • group_id: identificador del grupo asignado a la tarea

Tabla 18. Tabla Task

Tabla	Watch
Descripción	Esta tabla almacena los datos referentes a las guardias
Atributos	<ul style="list-style-type: none"> • id: el identificador de guardia • watch_type: tipo de guardia • watch_day: día de la guardia • monitor_id: identificador del monitor asignado a la guardia

Tabla 19. Tabla Watch

C Diagrama de clases completo

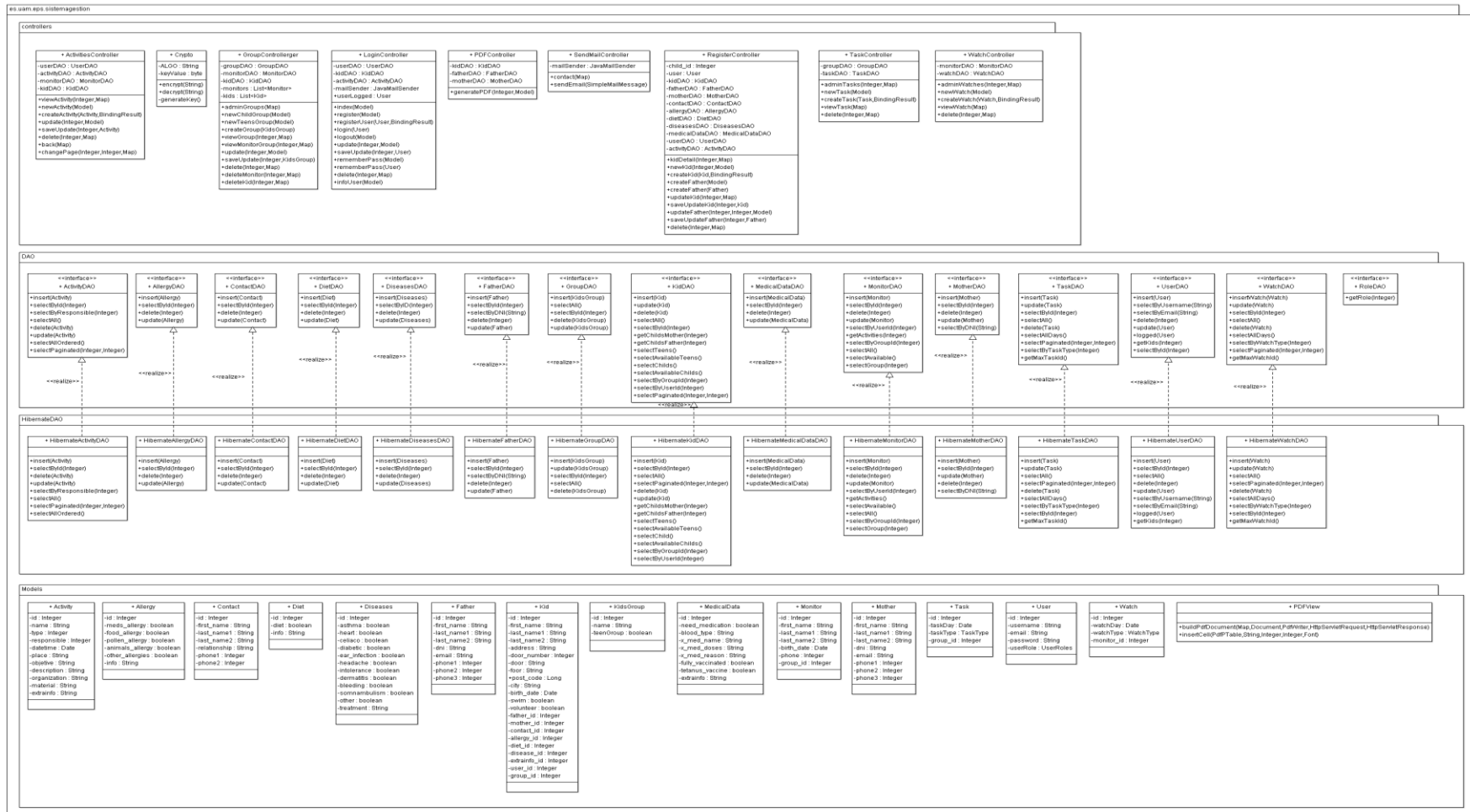


Figura 36. Diagrama de clases

D Integración con la Base de Datos desde Spring

Para integrar Hibernate con nuestro proyecto Spring MVC los ficheros de configuración deben tener lo siguiente:

pom.xml: Se deben incluir las siguientes dependencias tal como muestra la Figura 37.

```
<spring-framework.version>3.2.3.RELEASE</spring-framework.version>
<hibernate.version>4.2.1.Final</hibernate.version>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-tx</artifactId>
  <version>${spring-framework.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>${spring-framework.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-orm</artifactId>
  <version>${spring-framework.version}</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>${hibernate.version}</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-entitymanager</artifactId>
  <version>${hibernate.version}</version>
</dependency>
```

Figura 37. Fichero pom.xml – Dependencias

mvc-config.xml: El fichero de configuración de Spring debe indicar el paquete que reúna todas las clases que van a ser anotadas, como indica la Figura 38.

```
<context:component-scan base-package="es.uam.eps.sistemagestion" />
```

Figura 38. Fichero mvc-config.xml

application-config.xml: Se deben anotar todas las transacciones y establecer el *datasource* como en la Figura 39.

```

<tx:annotation-driven transaction-manager="transactionManager" />

<!-- We define a Bean for datasource (MySQL) -->
<bean id="dataSource"
      class="org.springframework.jdbc.datasource.DriverManagerDataSource">
  <property name="driverClassName" value="com.mysql.jdbc.Driver" />
  <property name="url" value="jdbc:mysql://localhost:3306/campamento" />
  <property name="username" value="██████" />
  <property name="password" value="██████" />
</bean>

```

Figura 39. Fichero application-config.xml – Conexión BBDD

Además, se crean el *sessionFactory* con las propiedades de Hibernate, indicando el *datasource* que hemos creado previamente.

Se debe incluir también el *vean* de *transactionManager*, incluyendo todos los DAO que se utilizarán. Esto queda reflejado en la Figura 40.

```

<bean id="sessionFactory"
      class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">
  <property name="dataSource" ref="dataSource" />
  <property name="annotatedClasses">
    <list>
      <value>es.uam.eps.sistemagestion.models.Kid</value>
      <value>es.uam.eps.sistemagestion.models.Father</value>
    </list>
  </property>
  <property name="hibernateProperties">
    <props>
      <!-- Shows on console SQL results -->
      <prop key="hibernate.show_sql">true</prop>
      <!-- SQL Dialect that Hibernate will use to communicate with DB -->
      <prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>
      <!-- Allows to create tables on DB. Values: validate: validates DB schema
            and makes no updates update: updates DB once created create: drops actual
            schema and creates a new one create-drop: creates a new DB and drops it when
            SessionFactory is destroyed -->
      <prop key="hibernate.hbm2ddl.auto">update</prop>
    </props>
  </property>
</bean>
<!-- Transaction Manager Bean -->
<bean id="transactionManager"
      class="org.springframework.orm.hibernate4.HibernateTransactionManager">
  <property name="sessionFactory" ref="sessionFactory" />
</bean>
<!-- We must include all DAO we'll use and inject sessionFactory -->
<bean id="userDAO" class="es.uam.eps.sistemagestion.dao.impl.HibernateUserDAO">
  <property name="sessionFactory" ref="sessionFactory" />
</bean>
<bean id="kidDAO" class="es.uam.eps.sistemagestion.dao.impl.HibernateKidDAO">
  <property name="sessionFactory" ref="sessionFactory" />
</bean>

```

Figura 40. Fichero application-config.xml - SessionFactoy y beans de clases

E Pruebas de compatibilidad

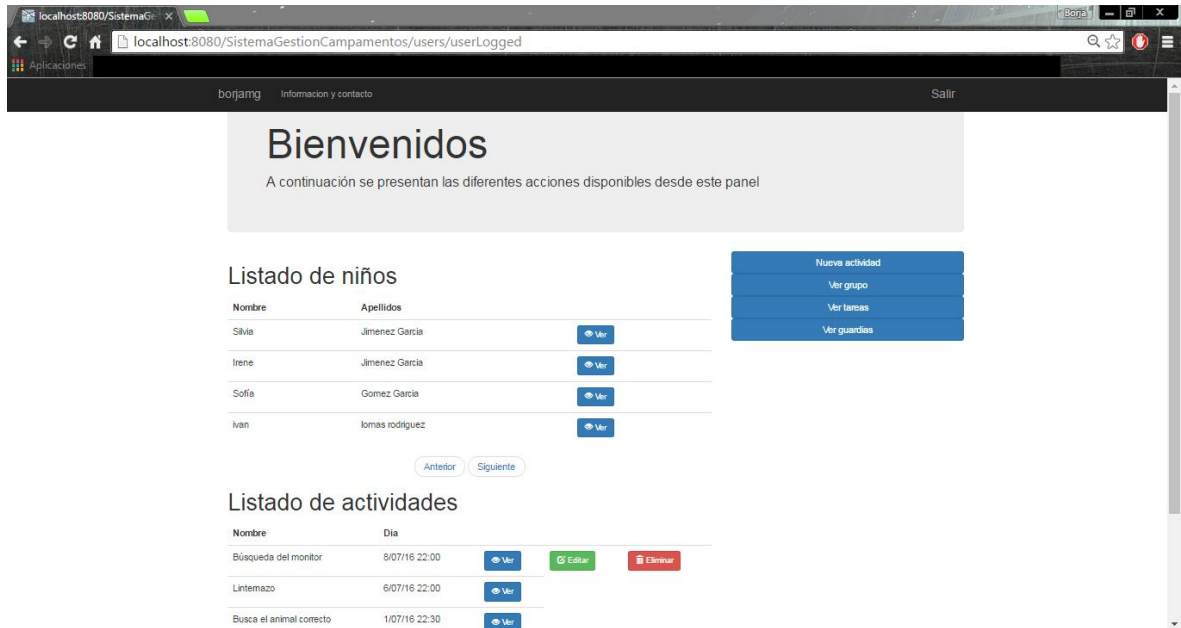


Figura 41. Visualización web. Navegador Google Chrome

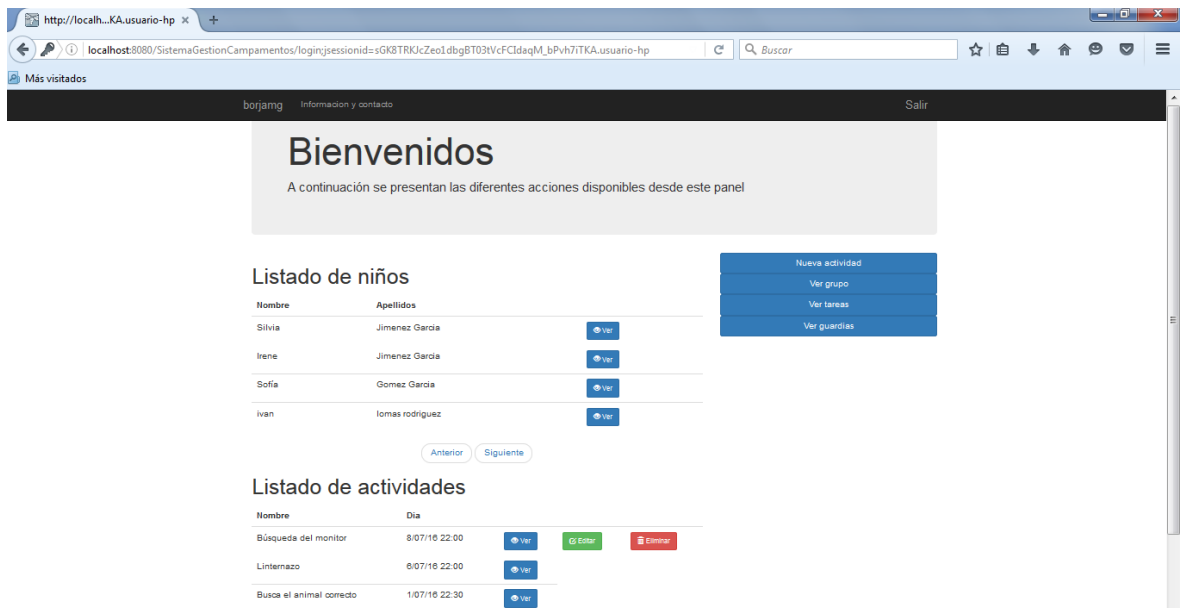


Figura 42. Visualización web. Navegador Mozilla Firefox

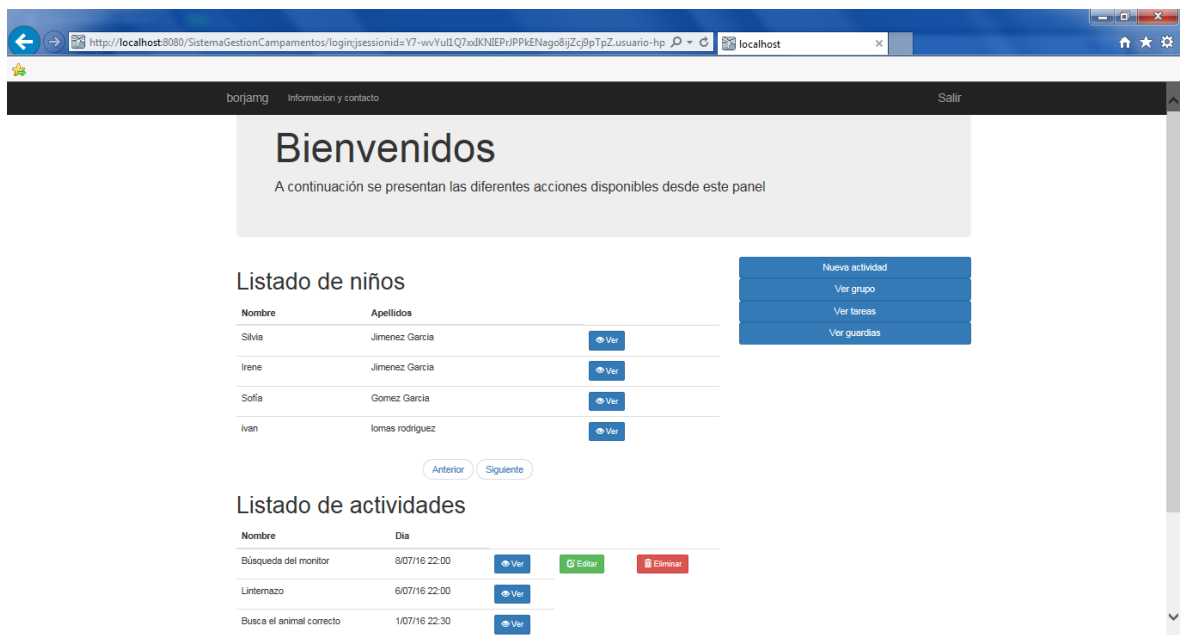


Figura 43. Visualización web. Navegador Internet Explorer

F Encuesta de satisfacción

Nombre y apellidos: _____

¿Qué tipo de usuario final representa?

- Coordinador
- Monitor
- Público

Por favor, responda a las siguientes preguntas marcando la opción deseada según el criterio:

NS/NC: Prefiere no opinar sobre la pregunta realizada	1: Totalmente en desacuerdo	2: Más bien en desacuerdo	3: Ni de acuerdo ni en desacuerdo	4: De acuerdo	5: Totalmente de acuerdo
--	---	---	--	-------------------------	---------------------------------------

Acerca de la funcionalidad de la plataforma:

Pregunta	NS/NC	1	2	3	4	5
1. El registro en la plataforma es sencillo						
2. Se accede fácilmente a la plataforma						
3. La comunicación con otros usuarios es sencilla (envío de correos)						
4. Puedo modificar mis datos de forma sencilla						
5. Puedo registrar un nuevo niño en el campamento fácilmente						
6. Puedo visualizar los datos de un niño fácilmente						
7. Puedo editar los datos de un niño fácilmente						
8. Puedo eliminar los datos de un niño fácilmente						
9. Genero el documento PDF de forma sencilla						
Desde este punto, únicamente para perfil de coordinador/monitor:						
10. Puedo registrar una nueva actividad en el campamento fácilmente						
11. Puedo visualizar los datos de una actividad fácilmente						
12. Puedo editar los datos de una actividad fácilmente						
13. Puedo eliminar los datos de una actividad fácilmente						
14. La visualización del grupo que tengo asignado es sencilla						
15. La visualización del cuadrante de tareas es sencilla						
16. La visualización del cuadrante de guardias es sencilla						
Desde este punto, únicamente para perfil de coordinador:						
17. La administración de grupos se realiza de forma sencilla						

18. La administración de tareas se realiza de forma sencilla						
19. La administración de guardias se realiza de forma sencilla						

Acerca de la interacción e interfaz:

Pregunta	NS/NC	1	2	3	4	5
20. La apariencia de la aplicación es agradable						
21. La interfaz de la aplicación (letras, colores, cantidad de elementos en la pantalla) cansa la vista						
22. Se puede contactar con el administrador de forma sencilla						
23. Puedo visualizar la información de mi cuenta de forma sencilla						
24. Los mensajes de ayuda/error son precisos						
25. El modo de realizar las acciones que deseo en la aplicación es intuitivo						
26. Encuentro con rapidez cómo realizar las acciones que deseo						
27. Orientarse dentro de la aplicación es sencillo						

¿Destacarías como positivo algún otro aspecto de esta aplicación o de tu interacción con ella? ¿Cuál/es?

¿Encuentras mejorable algún aspecto de esta aplicación o de tu interacción con ella? ¿Cuál/es?

¡Muchas gracias por tu colaboración!

G Distribución de las carpetas

A continuación, se muestran imágenes que representan la distribución de las carpetas dentro del proyecto desarrollado en eclipse. Esta distribución se divide en dos partes: una primera parte con las clases Java desarrolladas (Figura 44), y una segunda con las páginas JSP (Figura 45).

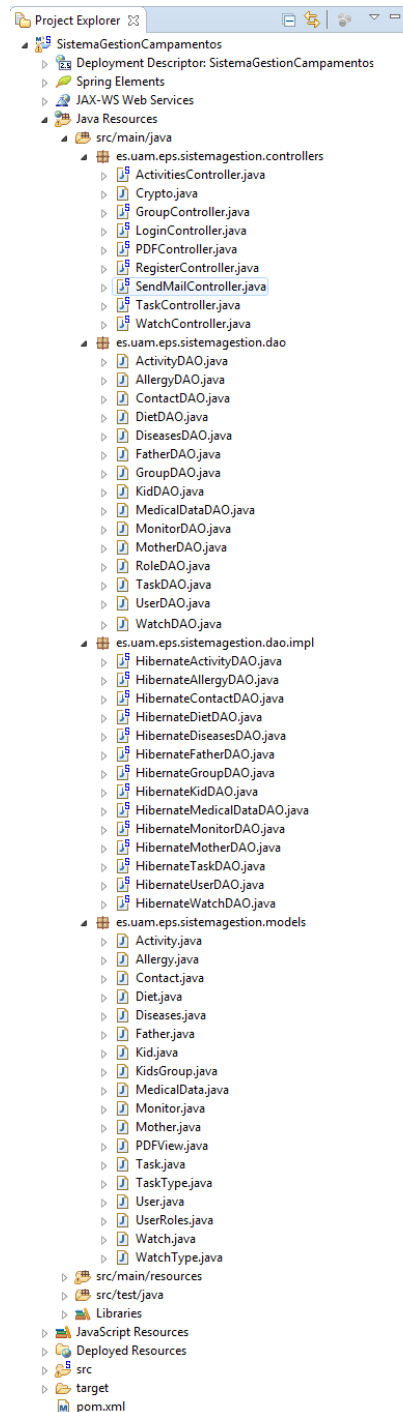


Figura 44. Distribución de carpetas. Proyecto web. Java

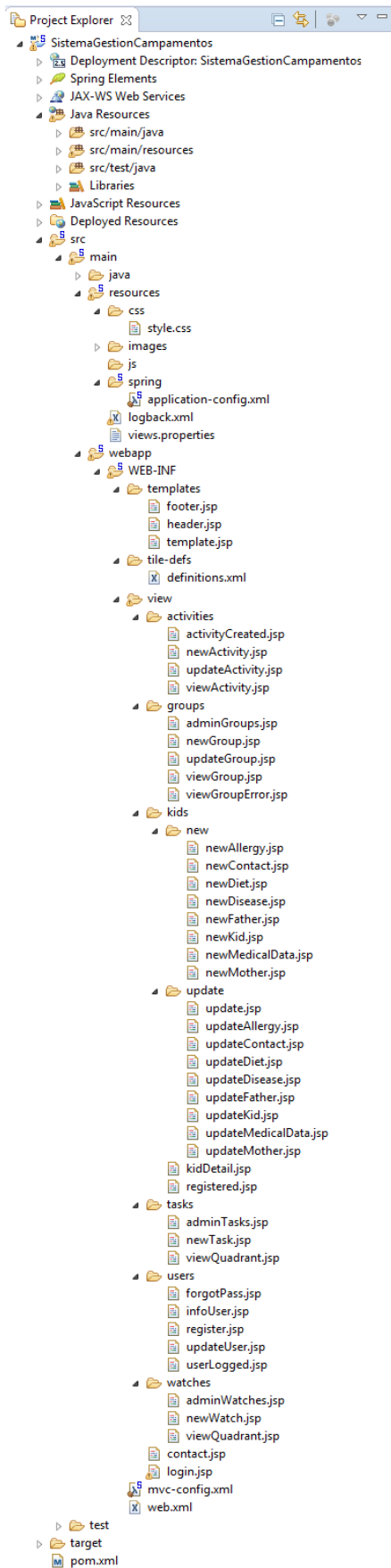


Figura 45. Distribución de carpetas. Proyecto web. JSP

La estructura del proyecto Android tiene una estructura similar a la del proyecto web desarrollado en eclipse, como se muestra en la Figura 46. En este caso, el código se encuentra en la carpeta *src*, teniendo las clases Java en una carpeta con el mismo nombre. Los *XML* que definen la interfaz gráfica de las pantallas se encuentran en las carpetas *layout* y *layout-land*, según sea para orientación vertical u horizontal respectivamente. En la carpeta *drawable* se encontrarán todos los recursos que emplee la aplicación, como pueden ser imágenes o ficheros de configuración.

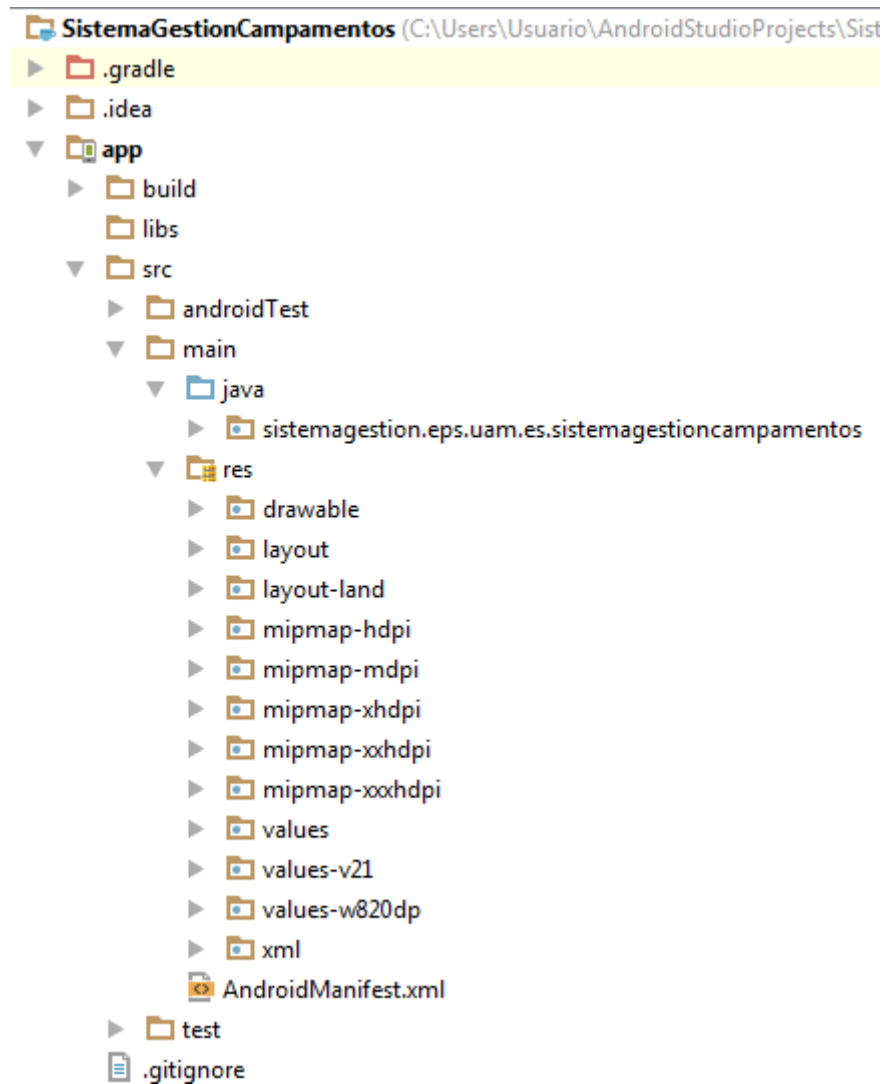


Figura 46. Distribución de carpetas. Proyecto Android

