UNIVERSIDAD DE OVIEDO

Trabajo Fin de Grado

Help desk simple

Director: MIGUEL RIESCO ALBIZU

Autor: JORGE SIERRA MIRANDA

Resumen

Este proyecto consiste en la creación de un sistema web a medida para la gestión de mensajes de contacto en la escuela de ingeniería informática de Oviedo.

Este sistema de mensajes se alimenta de los mensajes escritos por los visitantes en el formulario de contacto. Cada mensaje crea en el sistema una incidencia.

El sistema permite a los usuarios ver, buscar y contestar a los visitantes desde la plataforma. Las respuestas a los mensajes llegan a los interesados por correo electrónico.

Palabras clave

Mensaje, usuario, correo electrónico, gestión de incidencias, temáticas, aplicación web, java, informática.

Abstract

This project consists of the creation of a web system for managing contact messages in the school of computer science Oviedo.

The system receives messages written by visitors in the contact form. Each message in the system creates an incident.

The system allows users to view, search and reply visitor's messages from the platform. Messages reach those interested via email.

Key Words

Message, user, email, issue management, subjects, web application, java, informatics.

Índice General

1 Memoria del proyecto 12
1.1 Resumen de la Motivación, Objetivos y Alcance del Proyecto 12
1.2 Resumen de todos los aspectos 12
2 Introducción 14
2.1 Justificación del proyecto 14
2.2 Objetivos del proyecto 14
2.3 Evaluación de alternativas 14
2.3.1 Freshdesk 14
2.3.2 osTicket 15
3 Aspectos teóricos 16
3.1 Framework 16
3.2 Framework web 16
3.3 MVC 16
3.3.1 Push-based architecture 17
3.4 CRUD 17
3.5 UML 17
4 Planificación y Presupuesto del proyecto 18
4.1 Planificación 18
4.1.1 Diagrama de Gantt 19
4.2 Resumen del presupuesto 20
5 Análisis 21
5.1 Determinación del alcance 21
5.2 Requisitos del sistema 22
5.2.1 Obtención de requisitos del sistema 22
5.2.2 Requisitos de seguridad 23
5.2.3 Identificación de actores del sistema 23
5.2.4 Especificación de casos de uso 24
5.3 Análisis de interfaces de usuario 30

5.3.1 Descripción de la interface ... 30
5.3.2 Diagrama de navegabilidad ... 31
5.4 Especificación del plan de pruebas ... 32
5.4.1 Casos de prueba ... 32
6 Diseño del sistema ... 38
6.1 Arquitectura del sistema ... 38
6.1.1 Diagrama de paquetes ... 38
6.1.2 Diagrama de despliegue ... 40
6.2 Diseño de clases ... 41
6.3 Diseño de la base de datos ... 50
6.3.1 Descripción del SGBD ... 50
6.3.2 Integración del gestor con el sistema ... 50
6.3.3 Diagrama de la base de datos ... 50
6.4 Diseño de la interface ... 52
6.5 Diseño del plan de pruebas ... 63
6.5.1 Pruebas unitarias ... 63
6.5.2 Pruebas de integración y sistema ... 65
6.5.3 Pruebas de Usabilidad y Accesibilidad ... 65
7 Implementación del sistema ... 67
7.1 Lenguajes de programación, normas y estándares usados ... 67
7.1.1 Java ... 67
7.1.2 HTML ... 67
7.1.3 CSS ... 67
7.1.4 Freemarker ... 67
7.1.5 SQL ... 67
7.1.6 JavaScript ... 68
7.2 Frameworks y librerías ... 69
7.2.1 Spark ... 69
7.2.2 Freemarker ... 69
7.2.3 sql2o ... 69

7.2.4 JUnit ... 69
7.2.5 SLF4J ... 69
7.2.6 jBCrypt ... 69
7.2.7 commons-email ... 69
7.2.8 Gravatar ... 70
7.2.9 W3CSS ... 70
7.2.10 FontAwecome Icons ... 70
7.2.11 jQuerry ... 70
7.2.12 DataTables ... 70
7.3 Herramientas y programas usados ... 71
7.3.1 Microsoft Office ... 71
7.3.2 Dia ... 71
7.3.3 yuml ... 71
7.3.4 GIMP ... 71
7.3.5 Maven ... 71
7.3.6 IntelliJ IDEA ... 71
7.3.7 HSQLDB ... 71
7.3.8 DbVisualizer ... 72
7.3.9 Opera ... 72
7.4 Creación del sistema ... 73
7.4.1 Problemas encontrados ... 73
7.5 Descripción de los aspectos importantes ... 73
7.5.1 Vistas freemarker ... 74
7.5.2 Clase App ... 74
7.5.3 Clases de configuración ... 74
7.5.4 Clase WebConfig ... 74
8 Desarrollo de las pruebas ... 76
8.1 Pruebas unitarias ... 76
8.2 Pruebas de Integración y del sistema ... 77
8.3 Pruebas de usabilidad y accesibilidad ... 84

9 Manuales del sistema 85
9.1 Manual de usuario 85
9.1.1 Manual de instalación 85
9.1.2 Manual de usuario 87
9.2 Manual de propagador 92
9.2.1 Localización del código fuente 92
9.2.2 Importar proyecto 92
9.2.3 Estructura del proyecto 92
9.2.4 Estructura de los paquetes 93
9.2.5 Configuración del proyecto 94
9.2.6 Configurar la base de datos 94
9.2.7 Empaquetado en formato WAR para desplegar en servidores existentes 95
10 Conclusiones y ampliaciones 96
10.1 Conclusiones 96
10.2 Ampliaciones 99
10.2.1 Gestión avanzada de temáticas 99
10.2.2 Mejora de la interface 99
10.2.3 Mejorar el sistema de loggin 100
11 Referencias 101
12 Apéndices 103
12.1 Contenido entregado 103
12.2 Script de creación de tablas para la base de datos 103

Tabla de ilustraciones

Ilustración 3-1Relación de componentes en MVC ... 17
Ilustración 4-1Diagrama de Gantt ... 19
Ilustración 5-1 Diagrama de casos de uso ... 24
Ilustración 6-1Diagrama de paquetes ... 38
Ilustración 6-2 Diagrama de despliegue ... 40
Ilustración 6-3 Diagrama de clases general ... 42
Ilustración 6-4 Diagrama de clases del paquete app ... 43
Ilustración 6-5 Diagrama de clases del paquete config ... 45
Ilustración 6-6 Diagrama de clases del paquete business ... 47
Ilustración 6-7 Diagrama de clases del paquete model ... 48
Ilustración 6-8 Diagrama de clases del paquete persistence ... 49
Ilustración 6-9 Diagrama de la base de datos ... 51
Ilustración 6-10 Plantilla de las vistas ... 52
Ilustración 6-11 Menú de idioma ... 52
Ilustración 6-12 Vista de temas - Tabla ... 53
Ilustración 6-13 Muestra de campo de de entrada ... 53
Ilustración 6-14 Vista de tema ... 54
Ilustración 6-15 Vista de perfil ... 55
Ilustración 6-16 VIsta de edición de perfil ... 56
Ilustración 6-17 Menú de administración ... 56
Ilustración 6-18 Lista de usuarios ... 57
Ilustración 6-19 Lista de temáticas ... 57
Ilustración 6-20 Muestra de mensajes ... 58
Ilustración 6-21 Muestra de colores y formas ... 59
Ilustración 6-22 Muestra de colores y formas con Deuteranopia (ceguera al color verde) ... 60
Ilustración 6-23 Muestra de colores y formas con Protanopia (ceguera al color rojo) ... 61
Ilustración 6-24 Muestra de colores y formas con Tritanopia (ceguera al color azul) ... 62
Ilustración 9-1 Formulario de contacto ... 87
Ilustración 9-2 Mensaje de confirmación de envío ... 87
Ilustración 9-3 Formulario de contacto ... 88
Ilustración 9-4 Menú de idiomas ... 88
Ilustración 9-5 Menú ... 88
Ilustración 9-6 Lista de temas ... 89
Ilustración 9-7 Vista de perfil ... 89
Ilustración 9-8 Edición de perfil ... 90
Ilustración 9-9 Administración de usuarios ... 91

Ilustración 9-10 Formulario de creación de usuarios 91
Ilustración 9-11 Administración de categorías o temáticas 92
Ilustración 10-1 "Tabla" de tareas 96

1 Memoria del proyecto

1.1 Resumen de la Motivación, Objetivos y Alcance del Proyecto

Este proyecto nace del interés de la Escuela de Ingeniería informática de Oviedo por mejorar la comunicación de los visitantes del portal web con el centro y facilitar una respuesta o solución rápida a los mensajes recibidos.

Actualmente los mensajes que se escriben en el portal web son enviados por correo electrónico a una dirección. Es tarea del personal del centro revisar esta cuenta de correo y realizar las acciones convenientes correspondientes a la resolución de dicho mensaje.

El objetivo de este proyecto es mejorar el servicio de contacto asignando de manera inteligente y automática los mensajes a los usuarios que puedan resolver las tareas que resulten de dichos mensajes.

El proyecto se limitada a la gestión de los mensajes recibidos a través del formulario web en el portal de la escuela. Incluidos en el proyecto están además aquellos apartados enfocados al mantenimiento del mismo como puede ser el mantenimiento de usuarios.

1.2 Resumen de todos los aspectos

En esta sección se pretende presentar todos los apartados presentes en este documento.

Introducción: Se describe la razón de plantear el proyecto y los objetivos que planea cumplir.

Aspectos teóricos: Se definen los conceptos y términos básicos usados a lo largo del documento.

Planificación del Proyecto y Resumen de Presupuestos: Se describe la planificación del proyecto a lo largo del tiempo marcando las distintas tareas, fases e hitos. Se encuentra en esta sección un presupuesto general del proyecto sin exceso de detalles.

Análisis: Esta sección contiene la especificación de requisitos y el análisis del sistema, necesario para la fase de diseño.

Diseño del Sistema: Esta sección trata de especificar el diseño base de la aplicación: la arquitectura, clases que debe manejar, diseño de base de datos, plan de pruebas y diseños de la interface de usuario.

Implementación del sistema: Esta sección describe los lenguajes de programación, tecnologías y librerías usadas así como los programas a usar durante el desarrollo.

Desarrollo de las Pruebas: Esta sección amplía en plan de pruebas escribiendo las pruebas realizadas y sus resultados.

Manuales del Sistema: Incluye los manuales necesarios: Manual de instalación, Manual de usuario para todos los distintos roles y manual de programación para mantener y actualizar el sistema.

Conclusiones y Ampliaciones: Sección de análisis retrospectivo sobre la aplicación.

Referencias Bibliográficas: Listado de documentos y sitios web consultados durante el desarrollo.

Apéndices: Apartados adicionales que acompañan la documentación.

2 Introducción

2.1 Justificación del proyecto

Este proyecto planea, como ya se ha mencionado, mejorar el servicio de contacto del portal web de la escuela.

Este sistema permitirá a los empleados de la escuela ahorrar tiempo y esfuerzo en la gestión de los mensajes recibidos por el portal web.

El sistema se plantea hecho a medida a las necesidades y forma de trabajo del centro.

2.2 Objetivos del proyecto

Los principales objetivos del proyecto son los siguientes:

- El sistema estará disponible para su uso en la web y se encontrará en una URI determinada.
- El sistema controlará el acceso de los usuarios a través de un nombre de usuario y una contraseña.
- En el sistema existan perfiles de usuarios y dos grupos de usuario: administradores y usuarios.
- Todos los usuarios podrán responder a aquellos mensajes que sean de su incumbencia.
- Los usuarios administradores podrán crear modificar y desactivar cuentas.

2.3 Evaluación de alternativas

Un HelpdDesk, también llamado Centro de Atención al Usuario es un sistema de recursos humanos y tecnológicos que tiene como objetivo gestionar incidencias, normalmente relacionadas con un sistema informático.

Un HelpDesk común gestiona incidencias obtenidas de distintas fuentes, correo electrónico, formularios web, llamadas telefónicas... y reparte de distintas maneras entre su personal experto.

A continuación se describen sistemas existentes que total o parcialmente cubren las necesidades que se plantean cubrir con el sistema a desarrollar.

2.3.1 Freshdesk

Freshdesk es una plataforma en la nube para la administración de servicio al cliente.

Esta plataforma es capaz de recibir consultas de distintas fuentes y administrarlas automáticamente. Ofrece usuarios, grupos de usuarios, automatización de tareas, foros...

Esta aplicación freemium responde en general a las necesidades que cubre un helpdesk. [10]

2.3.2 osTicket

osTicket es un centro de administración online Open Source que ofrece todos los servicios esperados de un helpdesk. Además ofrece una API para la integración con otros sistemas. [11]

3 Aspectos teóricos

A continuación se describen aquellos aspectos teóricos de capital importancia para el proyecto:

3.1 Framework

En el desarrollo de software es un sistema de ayuda para el desarrollo y/o implementación de una aplicación. Un framework puede ser desde una serie de conceptos, prácticas y criterios hasta un conjunto de librerías dependientes del lenguaje para un objetivo específico. [2] [3]

3.2 Framework web

Un framework web (o para desarrollo web) es un framework enfocado a desarrollar aplicaciones web incluyendo webs dinámicas, servicios web y APIs. Un framework web simplifica o automatiza el tratamiento de actividades comunes en el desarrollo web ofreciendo al desarrollador herramientas para ello.

3.3 MVC

MCV o modelo vista controlador (*model view controller* en inglés) es un patrón de arquitectura de software usado comúnmente en la web. [2] [4]

En este patrón se separan los componentes en tres partes:

Modelo: Representa la información con la que trabaja la aplicación.

Vista: Interface del usuario. En aplicaciones web la vista son las páginas web que ve el usuario.

Controlador: Maneja los eventos producidos, normalmente por interacción de un usuario con la vista, modifica el modelo y ejecuta la lógica de negocio.

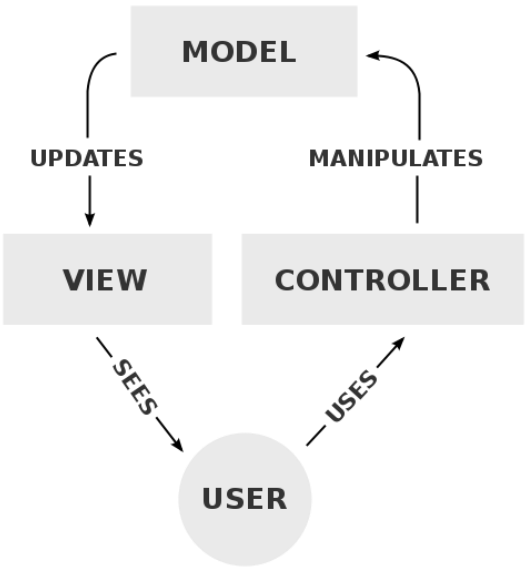

Ilustración 3-1Relación de componentes en MVC

3.3.1 Push-based architecture

Push-based o *action-based* es una arquitectura muy común dentro del MVC en la que el controlador maneja las peticiones http y realiza acciones para cada petición. Tras realizar las acciones pertinentes deben devolver los datos necesarios para la vista. De esta manera se tiene más control sobre lo que se ofrece en la vista. [5]

3.4 CRUD

CRUD es el acrónimo en inglés de c*reate, read, update and delete*. En español: crear, leer, actualizar y borrar. Estas son las cuatro acciones básicas que todo sistema de persistencia tiene.

3.5 UML

Lenguaje unificado de modelado, conocido como UML por sus siglas en inglés, es un lenguaje de modelado más usado en el desarrollo software.

Es un lenguaje gráfico que sirve para representar, especificar, construir y documentar un sistema software. De forma estandarizada describe un “plano” del sistema incluyendo aspectos conceptuales como técnicos. [6]

4 Planificación y Presupuesto del proyecto

4.1 Planificación

El proyecto se ha descompuesto en múltiples tareas para poder gestionarlo.

Las tareas principales son las siguientes y consisten en:

Aprendizaje: investigación y aprendizaje de uso de tecnologías o herramientas desconocidas usadas en el proyecto.

Análisis: múltiples tareas para definir el producto: estudio de alcance, estudio de requisitos, toma de requisitos, realización esquemas...

Diseño: diseñar el sistema a implementar: diseño de clases, diagramas de clase, de interacción, diseño de interfaces, diseño de la base de datos...

Implementación: implementación del código del producto.

Pruebas: tareas de verificación del producto.

Documentación: realizar la documentación necesaria para el proyecto.

El tiempo de desarrollo esperado es de 230 horas. Comenzando el 22 de febrero de 2016, realizándose a lo largo de 20 semanas.

El desarrollo sigue un modelo de prototipos. [4] A lo largo del tiempo se desarrolla una porción funcional del producto. Esta se analiza y prueba. De encontrarse errores estos se corrigen. Después se añade nuevas porciones de funcionalidad al producto volviendo a analizarlo y probarlo.

Es decir, se realizan muchas de las tareas de la planificación a la vez que se hace una parte del prototipo. La tarea de Documentación se realiza a lo largo de todo el proyecto pero el esfuerzo temporal mayor en dicha tarea se realiza al finalizar.

4.1.1 Diagrama de Gantt

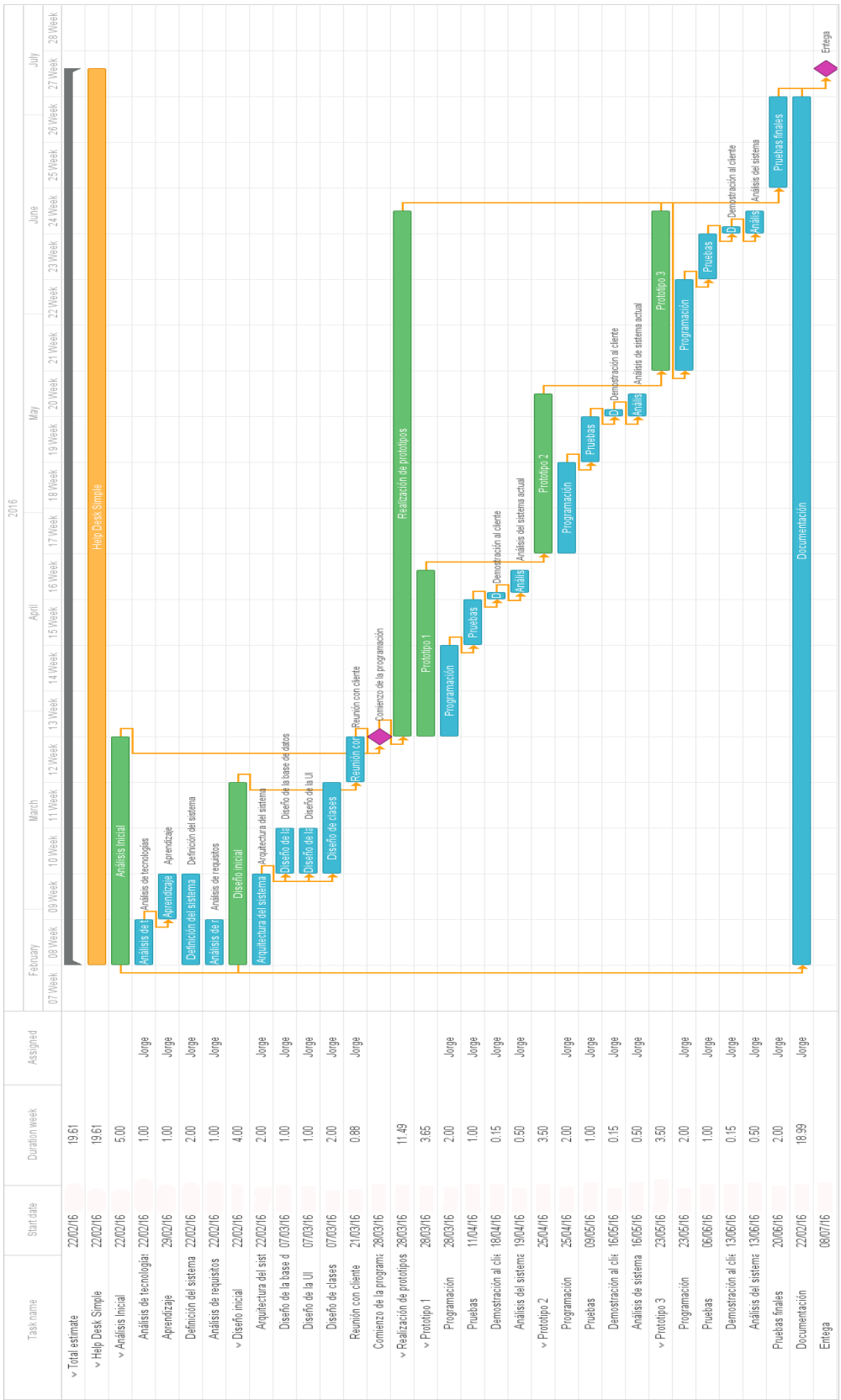

Ilustración 4-1Diagrama de Gantt

4.2 Resumen del presupuesto

A continuación se muestra una tabla de resumen del presupuesto.

Escuela de Ingeniería Informática de Oviedo							
	Unidades	Unidad	Detalles	Precio unitario	Depreciación	Importe	
						Parcial	Total
1	Recursos Humanos						6900
1.1	230	horas	Aprendizaje, análisis, diseño. Implementación y pruebas	30	NO	6900	
2	Hardware						71.81
2.1	1	ud.	Packard Bell Tsx66-hr-260sp	782.4	61.36471	61.36	
2.2	1	ud.	Impresora EPSON Stylus 550	80	10.45752	10.45	
3	Software						19.13
3.1	1	ud.	Windows 7 Professional	125	9.803922	9.80	
3.2	1	ud.	Microsoft office 2010	119	9.333333	9.33	
4	Consumibles y servicios						40
4.1	1	ud.	Material de oficina	15	NO	15	
4.2	1	ud.	Paquete de cartuchos de tinta	20	NO	20	
4.3	1	ud.	Paquete folios DIN A4	5	NO	5	
					Subtotal:		7030.94 €
					IVA:		1476.50 €
					Presupuesto total final:		8507.44 €

La columna marcada como *Depreciación* indica que el objeto presupuestado no es un gasto específico del proyecto. El importe parcial de dicho producto es el equivalente a la amortización de dicho producto con respecto a su coste inicial, vida útil y tiempo de uso para el proyecto (20 semanas).

Depreciación = (Precio total / semanas de vida útil) *semanas de uso en el proyecto

5 Análisis

5.1 Determinación del alcance

Se pretende construir un sistema que sirva para automatizar las tareas relacionadas con la gestión de mensajes en la Escuela de Ingeniería Informática de Oviedo. Los mensajes recibidos se asignarán de manera automática e inteligentemente a los usuarios del sistema.

El acceso al sistema se realiza vía WEB, siendo necesario el registro de los usuarios por parte de administradores para permitir el acceso y uso de dicho sistema.

El sistema maneja incidencias agrupándolas por temática. Los administradores del sistema asignan a los usuarios ciertas temáticas para que respondan a los temas entrantes en el sistema. Los usuarios responden y realizan acciones sobre los temas pertenecientes a sus temáticas asignadas.

Las incidencias entran al sistema por un formulario web disponible para cualquier visitante.

El sistema además permite la administración de usuarios a los administradores permitiéndoles, además de asignar las temáticas, crear, modificar y cancelar las cuentas de usuario. Pueden también crear y borrar temáticas en función de las necesidades.

El sistema tiene una interface web que comunica la aplicación con los usuarios a partir de correos electrónicos.

El sistema controla los mensajes, las respuestas y los usuarios que acceden a él.

El sistema no integra ninguna funcionalidad específica al manejo de las tareas creadas a partir de los temas, es decir, no cuenta con ninguna funcionalidad de: agenda de tareas, calendario, ni control. Únicamente maneja la comunicación.

5.2 Requisitos del sistema

5.2.1 Obtención de requisitos del sistema

A continuación se muestran agrupados los requisitos del sistema:

5.2.1.1 Requisitos de Usuario

5.2.1.1.1 Gestión de incidencias

Código	Nombre	Descripción
R1.1	Registrar temas	Se debe añadir un tema de ayuda al sistema una vez rellenado el formulario de contacto.
R1.2	Avisar del registro de temas	Al registrar un tema debe avisarse por correo a los usuarios asignados a su temática.
R1.3	Cambiar temática	Un usuario asignado o administrador puede cambiar la temática de un tema abierto.
R1.4	Avisar del cambio de temática	Al cambiar la temática de un tema debe avisarse por correo a los usuarios asignados a la nueva temática.
R1.5	Contestar a un tema	Los usuarios designados contestarán a través de un formulario a los temas asignados.
R1.6	Avisar al interesado cuando se contesta a un tema	Cuando se contesta a un tema se envía la respuesta por correo al visitante que creó la incidencia original.
R1.7	Cerrar tema	Cuando se considere resuelto un usuario asignado puede cerrar un tema dándolo como terminado e impidiendo nuevas respuestas.

5.2.1.1.2 Gestión de usuarios

Código	Nombre	Descripción
R2.1	Registrar usuario	Un usuario administrador puede crear un nuevo usuario en el sistema.
R2.2	Modificar datos de un usuario	Un usuario administrador o el propio usuario pueden modificar sus datos.
R2.3	Cancelar un usuario	Un administrador o el propio usuario pueden eliminar el usuario. Esto impide al usuario acceder al sistema sin borrarlo.
R2.4	Reactivar un usuario	Un administrador puede reactivar un usuario.
R2.5	Modificar permisos de administración	Un usuario administrador dar y eliminar poderes de administración a otros usuarios.

5.2.1.1.3 Gestión de temáticas

Código	Nombre	Descripción
R3.1	Registrar una temática	Un usuario administrador puede crear una nueva temática.
R3.2	Eliminar una temática	Un usuario administrador puede eliminar una temática.
R3.3	Asignar una temática	Un usuario administrador puede asignar una temática a un usuario del sistema
R3.4	Desasignar una temática	Un usuario administrador puede eliminar una asignación.

5.2.1.1.4 Gestión de acceso/sesión

Código	Nombre	Descripción
R4.1	Iniciar sesión	Un usuario activo puede iniciar sesión en el sistema con su usuario y contraseña
R4.2	Cerrar sesión	Un usuario puede salir de sesión desconectándose.

5.2.2 Requisitos de seguridad

Los requisitos de seguridad están relacionados con la seguridad web.

Código	Nombre	Descripción
R5.1	Impedir acceso sin sesión	El sistema impedirá el acceso a las vistas a visitantes que no tengan una sesión iniciada. Todos se redirigirán a la vista de acceso para que inicien sesión.
R5.2	Impedir acciones sin sesión	El sistema impedirá realizar acciones a visitantes que no tengan una sesión iniciada. Todos se redirigirán a la vista de acceso para que inicien sesión.

5.2.3 Identificación de actores del sistema

El sistema diferenciará tres tipos de actores.

Visitantes

Los visitantes son aquellos usuarios que usarán el sistema con el objetivo de enviar un comunicado a través de la web a la escuela.

Usuarios

Los usuarios son aquellos usuarios designados a responder a los visitantes. Tienen cierto control sobre los temas pero sólo si pertenecen a temas que tengas asignados.

Administradores

Los administradores tienen la tarea de mantener el sistema, asignando a los usuarios las temáticas que les correspondan. Tienen control sobre los usuarios y acceso total a todas las temáticas.

5.2.4 Especificación de casos de uso

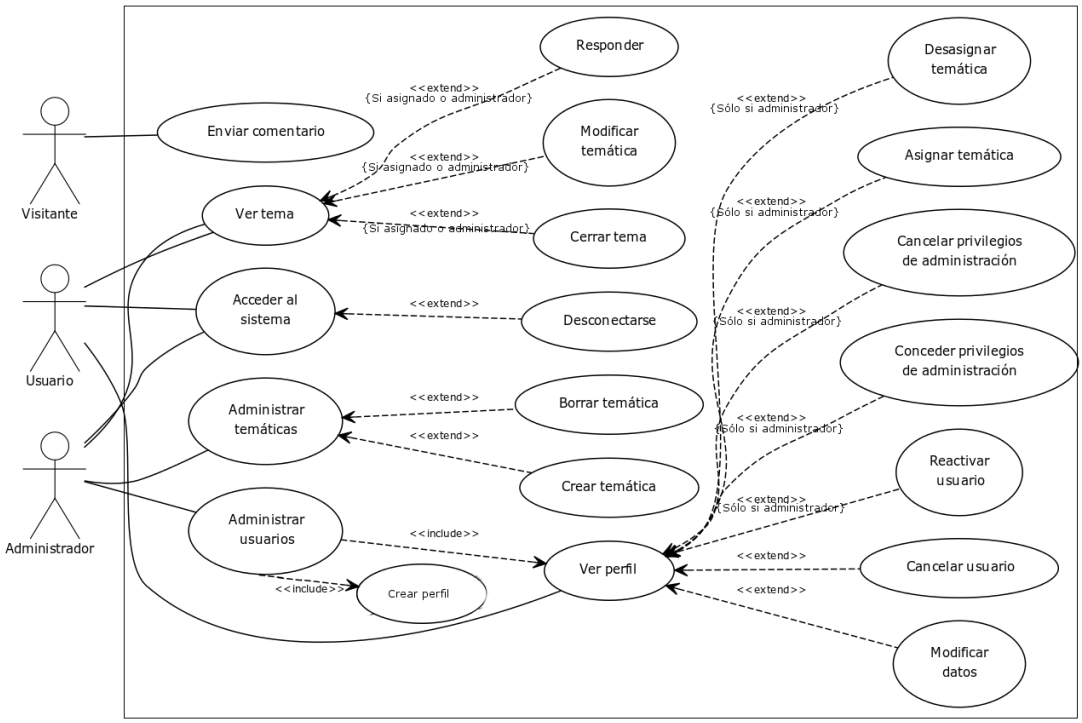

Ilustración 5-1 Diagrama de casos de uso

Nombre del Caso de Uso
Enviar comentario
Rol
Visitante
Descripción
El visitante accede al formulario de contacto y rellena todos los campos: • Nombre • Correo • Tema • Temática • Comentario Posterior mente pulsa el botón de enviar.
Alternativa
Faltan datos en el formulario, no se envía nada.

Nombre del Caso de Uso
Acceder al sistema
Rol
Usuario, Administrador
Descripción
El usuario accede al formulario de inicio de sesión e introduce su nombre de usuario y contraseña. Pulsa el botón para enviar el formulario y accede al sistema.
Alternativa
La combinación de usuario y contraseña es incorrecta y permanece en la página mostrándole un mensaje de error.

Nombre del Caso de Uso
Desconectarse
Rol
Usuario, Administrador
Descripción
El usuario pulsa el botón de desconectarse y sale de la aplicación
Alternativa
-

Nombre del Caso de Uso
Ver tema
Rol
Usuario, Administrador

Descripción
El usuario accede a la vista de un tema y se muestra el comentario y las respuestas.
Alternativa
El tema no existe, se muestra la lista de temas.

Nombre del Caso de Uso
Responder
Rol
Usuario, Administrador
Descripción
El usuario rellena el formulario de respuesta y pulsa Responder. Aparece en la vista su respuesta.
Alternativa
El usuario no es administrador ni tiene asignado el tema. Aparece un mensaje de error.

Nombre del Caso de Uso
Cerrar tema
Rol
Usuario, Administrador
Descripción
El usuario usa la opción de cerrar tema. El tema aparece como cerrado.
Alternativa
El usuario no es administrador ni tiene asignado el tema. Aparece un mensaje de error.

Nombre del Caso de Uso
Modificar temática
Rol
Usuario, Administrador
Descripción
El usuario selecciona una nueva temática y la asigna. El tema se actualiza.
Alternativa
El usuario no es administrador ni tiene asignado el tema. Aparece un mensaje de error.

Nombre del Caso de Uso
Ver perfil
Rol
Usuario, Administrador

Descripción
El usuario accede al perfil. Se muestran los datos.
Alternativa
El usuario no existe. Aparece un mensaje de error.

Nombre del Caso de Uso
Modificar datos.
Rol
Usuario, Administrador
Descripción
El usuario modifica en un formulario los datos que desea cambiar y lo envía. Aparece un mensaje de confirmación.
Alternativa
Algún dato es erróneo o el usuario no es administrador e intenta modificar otro usuario. Aparece un mensaje de error.

Nombre del Caso de Uso
Cancelar usuario
Rol
Usuario, Administrador
Descripción
El usuario cancela el usuario. Aparece un mensaje de confirmación.
Alternativa
El usuario no es administrador e intenta modificar otro usuario. Aparece un mensaje de error.

Nombre del Caso de Uso
Dar privilegios de administrador
Rol
Administrador
Descripción
El usuario da privilegios de administración. Aparece un mensaje de confirmación.
Alternativa
-

Nombre del Caso de Uso
Quitar privilegios de administrador
Rol
Administrador
Descripción

El usuario quita privilegios de administración. Aparece un mensaje de confirmación.
Alternativa
-

Nombre del Caso de Uso
Asignar temáticas
Rol
Administrador
Descripción
El usuario asigna una temática que ha seleccionado para el usuario. Aparece un mensaje de confirmación.
Alternativa
Otro administrador ya ha asignado esa temática al mismo usuario y aparece un mensaje de error con dicha información.

Nombre del Caso de Uso
Desasignar temáticas
Rol
Administrador
Descripción
El usuario desasigna una temática que ha seleccionado para el usuario. Aparece un mensaje de confirmación.
Alternativa
Otro administrador ya ha desasignado esa temática al mismo usuario y aparece un mensaje de error con dicha información.

Nombre del Caso de Uso
Activar usuario
Rol
Administrador
Descripción
El usuario activa un usuario desactivado.
Alternativa
-

Nombre del Caso de Uso
Crear temática
Rol
Administrador
Descripción

El administrador accede a la administración de temáticas y registra una nueva temática con un nombre.
Alternativa
-

Nombre del Caso de Uso
Eliminar temática
Rol
Administrador
Descripción
El administrador accede a la administración de temáticas y selecciona una nueva temática para eliminar. Se elimina dicha temática.
Alternativa
Existen temas o usuarios asignados a esa temática. Se muestra un mensaje de error y no se elimina.

Nombre del Caso de Uso
Crear perfil
Rol
Administrador
Descripción
El administrador crea desde la administración un perfil nuevo rellenando los datos.
Alternativa
Algún dato es erróneo, se muestra un mensaje de error.

5.3 Análisis de interfaces de usuario

5.3.1 Descripción de la interface

5.3.1.1 La plantilla principal

Cabecera
Menú
Contenido
Pie de página

Cada una de las siguientes partes forma parte del contenido de la plantilla principal para la vista.

5.3.1.2 Lista de temas

Filtro

Dato 1	Dato 2	...	Acciones
Tema 1	...	...	[VER]
...	...	...	[VER]
Tema n	...	...	[VER]

5.3.1.3 Vista del tema

Incidencia
Respuesta 1
...
Respuesta n
Formulario de respuesta / Acciones

5.3.1.4 Vista de perfil

Dato 1
Dato 2
...
Dato n
[EDITAR]

Editar despliega un diálogo con un formulario con todos los datos y los botones de cancelar/aceptar.

5.3.1.5 Vista de usuarios

Una tabla con todos los usuarios con botones de [VER] y [MODIFICAR] que despliegan diálogos idénticos a la vista de perfil

5.3.1.6 Vista de temáticas

Una lista de temas con todos los temas. Botones de [MODIFICAR] para modificarlas en un diálogo.

Un botón de [AÑADIR] que despliega un diálogo para añadir una nueva temática.

5.3.1.7 Vista de creación de incidencias

Utiliza la plantilla base pero sin menú de navegación.

Es un formulario con todos los campos.

5.3.2 Diagrama de navegabilidad

Las vistas se relacionan de la siguiente manera:

```
\---Home
    +---Acceder
    |   +---Administración
    |   |   +---Vista de temáticas
    |   |   \---Vista de usuarios
    |   +---Desconectarse
    |   +---Ver perfil
    |   \---Ver temas
    |       +---Tema 1
    |       +---Tema 2
    |       \---Tema n
    \---Creación de incidencias
```

Las páginas del tercer nivel (Administración, desconectarse, ver perfil, ver temas están disponibles en un enlace en el menú si el usuario en sesión tiene permisos para acceder a dicha vista.

Desde todas las páginas se puede acceder a Home.

5.4 Especificación del plan de pruebas

Utilizando la metodología de desarrollo de prototipos las pruebas se realizan al finalizar un prototipo.

Las pruebas unitarias se realizan a través de Maven al compilar la aplicación. Las pruebas unitarias revisan el funcionamiento concreto del modelo y acceso a la base de datos. Las pruebas se realizan en un entorno estéril (base de datos vacía) verificando tanto los resultados esperados corrutos como los posibles resultados erróneos.

Las pruebas unitarias son pocas ya que la aplicación no realiza operaciones lógicas complicadas.

El resto de funcionalidades es probado por un grupo de voluntarios prueban toda la funcionalidad del prototipo.

Se prueban todos los casos de uso de forma independiente probando también los resultados alternativos. Prueba todos los roles de usuario para cada combinación de caso de uso. Los resultados se analizan para poder mejorar la funcionalidad y arreglar todos los bugs que aparezcan.

También se revisa el correcto funcionamiento de la integración del sistema con el servidor de correos usado en la prueba y se analiza la usabilidad de la aplicación.

Los voluntarios prueban además la usabilidad y accesibilidad de la aplicación.

5.4.1 Casos de prueba

Caso de Uso 1: Enviar comentario	
Prueba	Resultado Esperado
Rellenar todos los datos del formulario y enviar	El sistema registra un tema más
Dar a enviar sin terminar de rellenar el formulario	Se remarcan las entradas sin rellenar y se muestra un mensaje

Caso de Uso 2: Acceder al sistema	
Prueba	Resultado Esperado
Rellenar el formulario de acceso con los datos de un usuario administrador	Se entra en el sistema
Rellenar el formulario de acceso con los datos de un usuario normal	Se entra en el sistema

Rellenar el formulario de acceso con los datos de un usuario escribiendo la contraseña incorrectamente	No se entra en el sistema
Pulsar enviar sin escribir un dato	No se envía el formulario y se remarca el dato faltante
Rellenar con datos de un usuario existente desactivado	No se entra al sistema

Caso de Uso 3: Desconectarse	
Prueba	Resultado Esperado
Tras entrar en sesión pulsar el botón del menú Desconectarse	Se sale del sistema y se va al formulario de acceso

Caso de Uso 4: Ver tema	
Prueba	Resultado Esperado
Se accede con un usuario administrador a un tema a partir de la lista de temas	Se muestra el tema
Se accede con un usuario normal a un tema a partir de la lista de temas	Se muestra el tema
Se accede con un usuario cualquiera al tema a parir del ID en la dirección	Se muestra el tema
Se acceder a un tema por su ID a partir de la dirección sin iniciar sesión	Se envía al usuario a la vista de inicio de sesión
Se accede con un usuario cualquiera al tema a parir del ID en la dirección siendo una ID inexistente	Se muestra un mensaje de error

Caso de Uso 5: Responder	
Prueba	Resultado Esperado
Administrador envía respuesta	Aparece la respuesta en la vista

Usuario asignado envía respuesta	Aparece la respuesta en la vista
Usuario no asignado envía respuesta	Se muestra un error

Caso de Uso 6: Cerrar tema	
Prueba	Resultado Esperado
Administrador cierra tema	Aparece el tema como resuelto
Usuario asignado cierra	Aparece el tema como resuelto
Usuario no asignado cierra	Se muestra un error

Caso de Uso 7: Cambiar temática a un tema	
Prueba	Resultado Esperado
Administrador cambia la temática al tema	Se actualiza la temática
Usuario asignado cambia la temática al tema	Se actualiza la temática
Usuario no asignado cambia la temática al tema	Se muestra un error

Caso de Uso 8: Ver perfil	
Prueba	Resultado Esperado
Administrador ve su propio perfil	Se muestra el perfil con todos los datos y el botón de editar
Administrador ve el perfil de otro	Se muestra el perfil con todos los datos y el botón de editar
Usuario ve su propio perfil	Se muestra el perfil con todos los datos y el botón de editar
Usuario ve el perfil de otro	Se muestra el perfil con datos simplificados
Cualquier usuario accede a la vista de un perfil que no existe	Se muestra un mensaje de error

Caso de Uso 9: Modificar datos	
Prueba	Resultado Esperado
Administrador modifica sus datos correctamente	Se modifica y muestra un mensaje mostrando un aviso de éxito
Administrador modifica su contraseña correctamente	Se modifica y muestra un mensaje mostrando un aviso de éxito

Administrador modifica sus datos incorrectamente	No se modifica y muestra un mensaje de error
Administrador modifica su contraseña incorrectamente	No se modifica y muestra un mensaje de error
Administrador modifica los datos de otro usuario correctamente	Se modifica y muestra un mensaje mostrando un aviso de éxito
Administrador modifica la contraseña de otro usuario correctamente	Se modifica y muestra un mensaje mostrando un aviso de éxito
Administrador modifica los datos de otro usuario incorrectamente	No se modifica y muestra un mensaje de error
Administrador modifica la contraseña de otro usuario incorrectamente	No se modifica y muestra un mensaje de error
Usuario modifica sus datos correctamente	Se modifica y muestra un mensaje mostrando un aviso de éxito
Usuario modifica su contraseña correctamente	Se modifica y muestra un mensaje mostrando un aviso de éxito
Usuario modifica sus datos incorrectamente	No se modifica y muestra un mensaje de error
Usuario modifica su contraseña incorrectamente	No se modifica y muestra un mensaje de error

Caso de Uso 10: Cancelar usuario	
Prueba	Resultado Esperado
Administrador cancela su usuario	Se cancela el usuario y actualiza la vista
Usuario cancela su usuario	Se cancela el usuario y actualiza la vista
Administrador cancela otro usuario	Se cancela el usuario y actualiza la vista

Caso de Uso 11: Dar privilegios de administrador	
Prueba	Resultado Esperado
Administrador da privilegios a otro usuario	Se otorgan privilegios al usuario y se actualiza la vista

Caso de Uso 12: Quitar privilegios de administrador

Prueba	Resultado Esperado
Administrador quita privilegios a otro usuario	Se eliminan los privilegios al usuario y se actualiza la vista

Caso de Uso 13: Quitar privilegios de administrador	
Prueba	Resultado Esperado
Administrador quita privilegios a otro usuario	Se eliminan los privilegios al usuario y se actualiza la vista

Caso de Uso 14: Asignar temática	
Prueba	Resultado Esperado
Administrador asigna una temática a otro usuario	Se asigna la temática al usuario y se actualiza la vista
Administrador asigna una temática a otro que ya haya sido asignado a esa temática entre que el administrador empezase a editar el perfil y el momento de que realizase la acción	Se muestra un error

Caso de Uso 15: Desasignar temática	
Prueba	Resultado Esperado
Administrador desasigna una temática a otro usuario	Se desasigna y se actualiza la vista
Administrador desasigna una temática a otro que ya haya sido entre que el administrador empezase a editar el perfil y el momento de que realizase la acción	Se muestra un error

Caso de Uso 16: Activar usuario	
Prueba	Resultado Esperado
Administrador cancelar una cuenta de usuario	Se actualiza la vista y se cancela el usuario en la base de datos

Caso de Uso 17: Crear temática	
Prueba	Resultado Esperado
Administrador crea una temática desde la administración	Se crea la temática en la base de datos y actualiza la vista

Caso de Uso 18: Eliminar temática	
Prueba	Resultado Esperado
Administrador elimina una temática en desuso desde la administración	Se elimina la temática en la base de datos y actualiza la vista
Administrador elimina una temática en uso desde la administración	Se muestra un mensaje de error

Caso de Uso 19: Crear usuario	
Prueba	Resultado Esperado
Administrador crea un usuario desde la administración	Se crea un usuario en la base de datos y redirige a la administración
Administrador crea un usuario desde la administración con datos erróneos	Se muestra un mensaje de error

6 Diseño del sistema

6.1 Arquitectura del sistema

6.1.1 Diagrama de paquetes

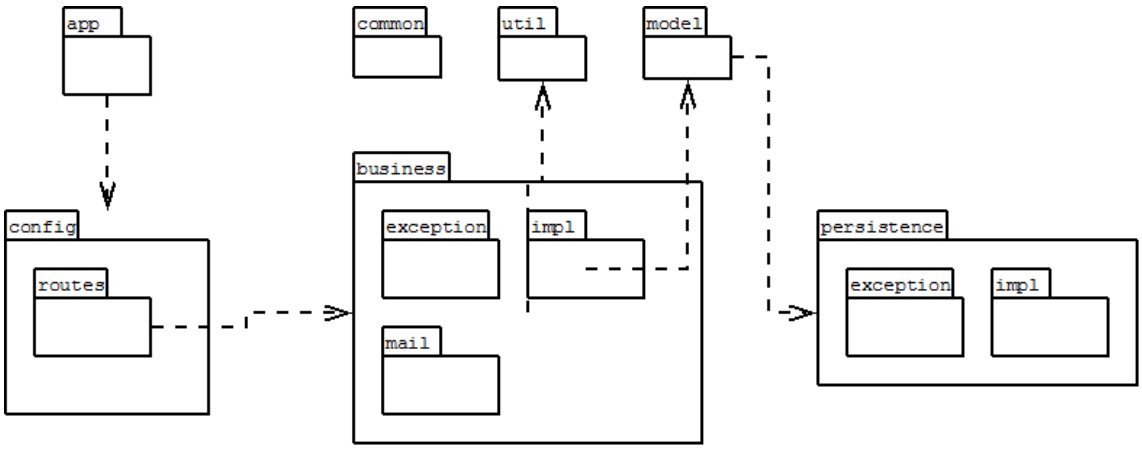

Ilustración 6-1Diagrama de paquetes

La aplicación diseñada según el patrón de n capas separa las clases en los anteriores paquetes.

Aparecen remarcadas las relaciones de los paquetes por funcionalidad. Además de estas relaciones existen las relaciones necesarias para mantener el modelo de n capas, es decir, el paquete common está relacionada con todas las capas, ya que contiene las factorías, y el paquete model contiene clases que transportan los valores por toda la aplicación desde la persistencia a la vista creada en las rutas.

Adicionalmente las clases del paquete model actúan como fachada adicional a la capa de persistencia imitando un patrón ActiveRecord. Esto permite utilizar un código más fluido y legible en la capa de negocio para tratar con la capa de persistencia.

El paquete útil puede ser accedido desde cualquier capa ya que contiene clases relacionadas con tareas que pueden resultar comunes a dos o más capas.

El paquete mail contiene las funcionalidades relacionadas con el correo electrónico. Se encuentra dentro del paquete de negocio porque es independiente del resto de capas.

A continuación se describe brevemente todos los paquetes:

6.1.1.1 app

Contiene la clase principal de la aplicación, el main,.y la administración de los parámetros de entrada.

6.1.1.2 business

La capa de negocio de la aplicación. Contiene las interfaces de los servicios.

6.1.1.3 business.exception

Contiene las excepciones específicas de la lógica de negocio.

6.1.1.4 business.impl

Contiene la implementación de la lógica de negocio.

6.1.1.5 business.mail

Contiene las clases específicas del sistema de envío de correo.

6.1.1.6 common

Contiene la factoría principal para independizar cada capa.

6.1.1.7 config

Contiene la configuración de la aplicación: carga todos los ficheros de propiedades y crea el mapeo de rutas de la aplicación.

6.1.1.8 config.routes

El mapeo de rutas de la aplicación. Estas clases asignan programáticamente las URIs a las acciones a realizar. Se encargan también de la gestión de los parámetros de las URIs y dirigir a las vistas.

6.1.1.9 model

Las clases del modelo de la aplicación. Son transversales a toda la aplicación ya que están presentes en todas las capas.

6.1.1.10 persistence

Capa de persistencia de la aplicación. Contiene las interfaces de la persistencia.

6.1.1.11 persistence.exception

Contiene las excepciones específicas de la capa de persistencia.

6.1.1.12 persistence.impl

Contiene la implementación de la capa de persistencia.

6.1.1.13 presentation

Contiene clases relacionadas con la vista.

6.1.1.14 util

Contiene clases de utilidad.

6.1.2 Diagrama de despliegue

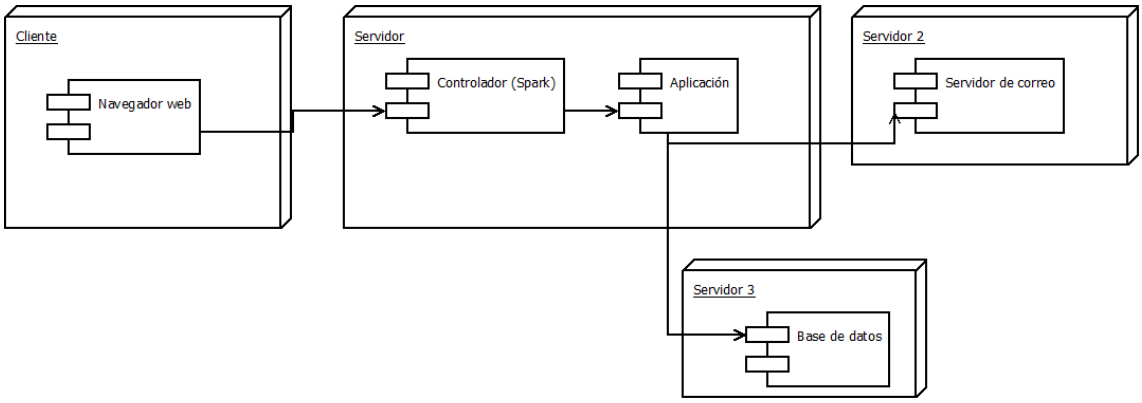

Ilustración 6-2 Diagrama de despliegue

Al ser la base de datos y el servidor de correos totalmente modulares y configurables se pueden encontrar en mismo servidor que la aplicación, en dos distintos, o en el mismo servidor externo a la aplicación.

Desde el navegador web el cliente o visitante se accederá a la aplicación realizando llamadas controladas por el framework Spark.

6.2 Diseño de clases

EL diseño general sigue un patrón de N capas donde cada capa se comunica con las contiguas.

Las capas principales son:

- Vista
- Configuración de acciones
- Negocio
- Persistencia

No presentes como capas están:

- Clases de utilidad
- Clases de infraestructura (factorías)
- Modelo (Es transversal a la aplicación)
- Clases para la ejecución de la aplicación (aquello que se ejecuta para que todo comience a funcionar)

Para sencillez el diagrama se ha dividido en partes. Cada una de las partes es explicada a fondo en un posterior diagrama en detalle.

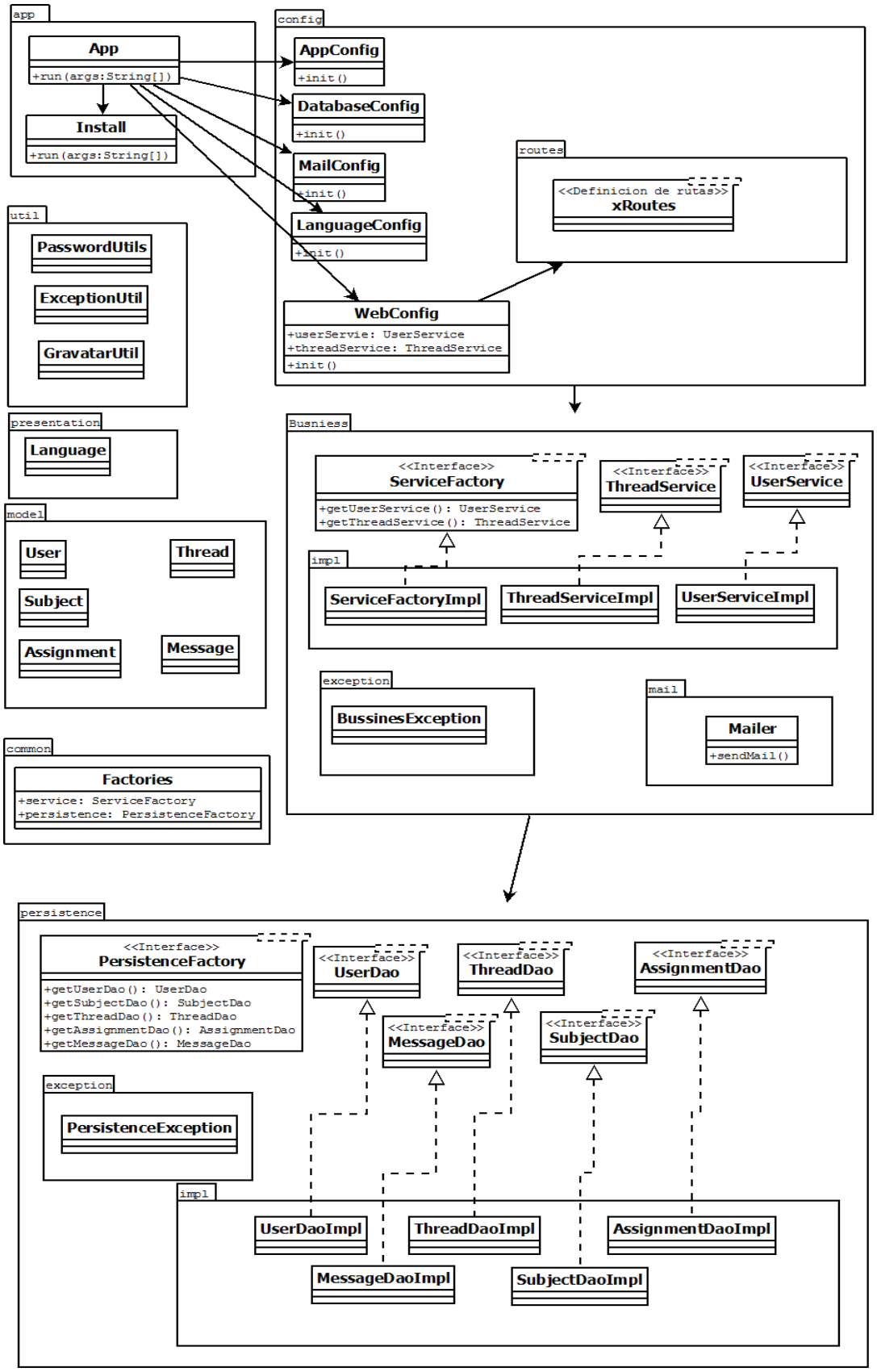

Ilustración 6-3 Diagrama de clases general

La clase principal de la aplicación se encuentra en el paquete app, con el nombre de App.

Es una clase que lee los argumentos de entrada y ejecuta las clases adecuadas para la configuración de entrada.

Desde la clase App se inicializan las distintas clases del paquete config en el orden adecuado con la configuración especificada en los parámetros de entrada.

La clase Install, se ejecuta desde App cuando se ejecuta la aplicación en modo instalación (según parámetros de entrada de App).

Desde la clase install se crean objetos y se persisten en la base de datos pasando por el ActiveRecord que es una fachada de la persistencia real de la aplicación.

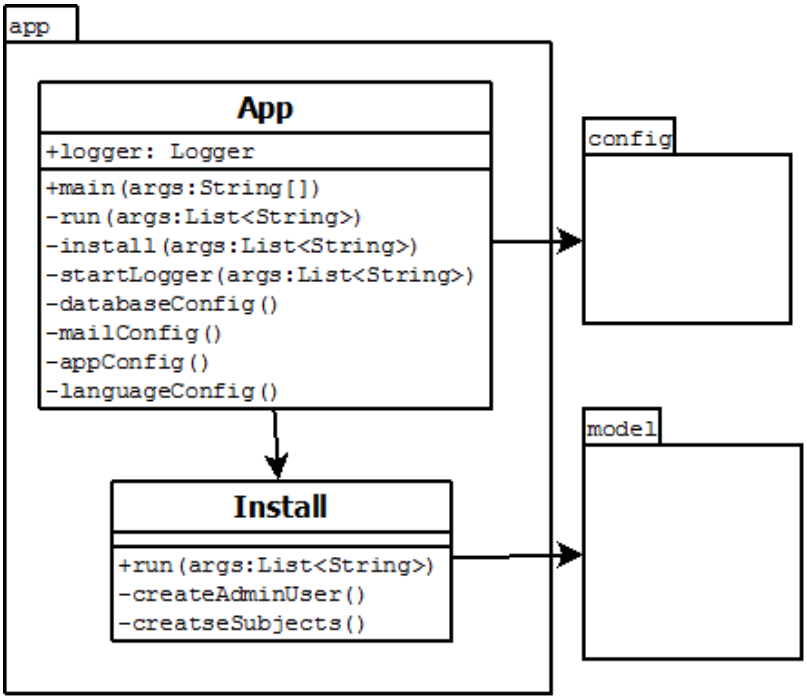

Ilustración 6-4 Diagrama de clases del paquete app

Las clases del paquete config se corresponden con las clases de configuración de la base de datos, del correo y de la aplicación web.

Es el corazón de la aplicación ya que la configuración web es la encargada de configurar y lanzar el controlador.

Las clases AppConfig, MailConfig, LanguageConfig y DatabaseConfig cargan los ficheros de propiedades cuando se inician (init) dando acceso a ellas a través del método estático get en cada una de ellas.

La clase LanguageConfig da acceso de manera ordenada a una serie de ficheros. Cada uno de estos ficheros de idiomas distintos crea un Language nuevo, un elemento usado para la presentación de la página.

La clase DatabaseConfig además se encarga de ofrecer de forma estática una conexión a la base de datos a través de sql2o.

La clase WebConfig tiene la misma misión, configurar la parte web de la aplicación; pero realiza su función de manera distinta. De forma declarativa utiliza las funciones del framework Spark para definir las acciones asociadas a cada ruta. Para modularizar la configuración de rutas se mantiene en WebConfig la funcionalidad común de todos los apartados web.

La declaración de rutas se realiza en clases distintas que agrupan declaraciones con funcionalidad parecida. Por ejemplo en HelpRoutes se definen las rutas para genrar la vista del formulario de contacto (get "/help") y de envío del formulario (post "/help").

Todas las clases definen las rutas en su constructor y reciben como parámetro la referencia al objeto de tipo WebConfig. Así tienen acceso a los métodos comunes y a la siguiente capa, la capa de servicios.

Para acceder a la capa de servicios el objeto de clase WebConfig guarda referencias a los servicios obtenidas con la factoría.

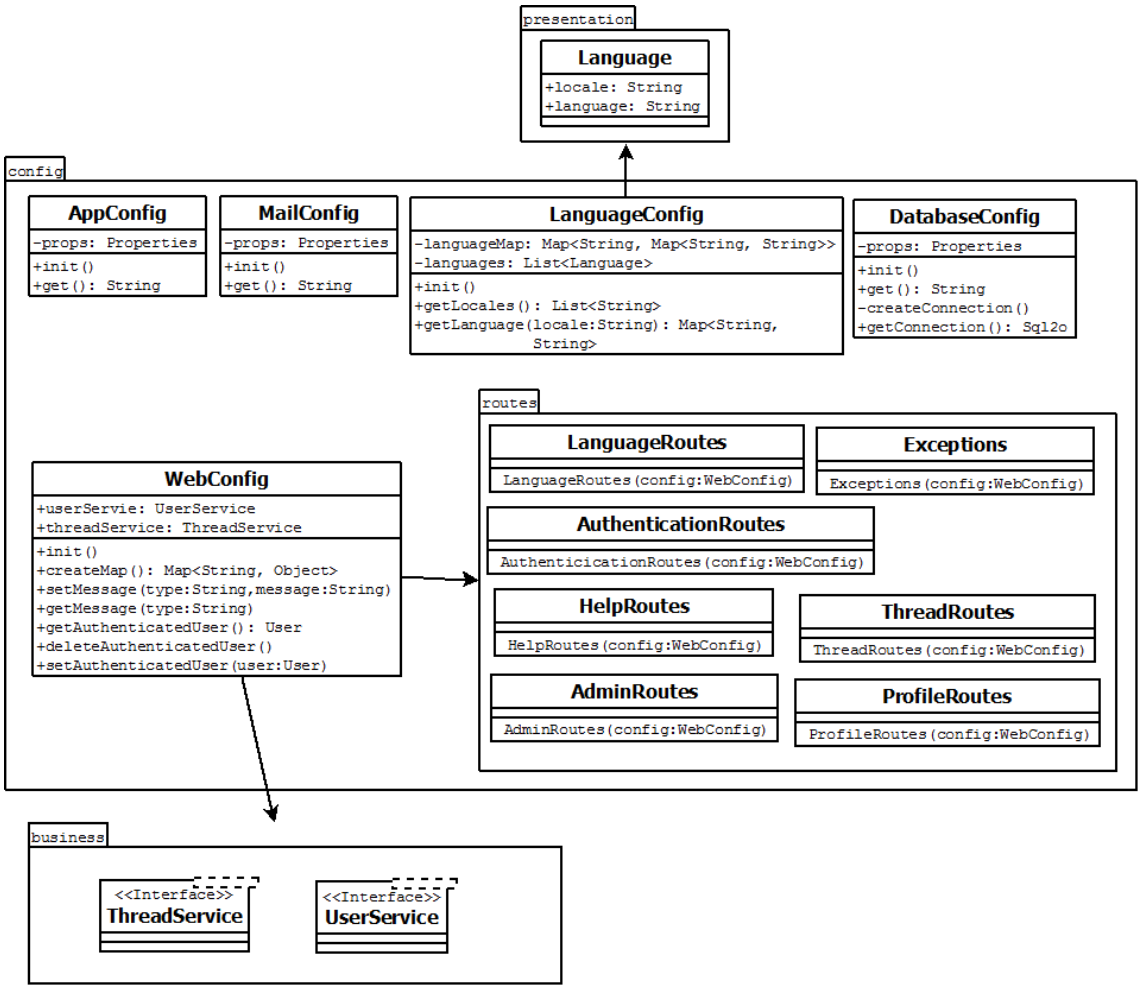

Ilustración 6-5 Diagrama de clases del paquete config

El paquete business se encarga de la lógica de negocio. Recibe los datos que han sido recogidos por las acciones de las rutas para realizar sus tareas. Para independizar la implementación todos los servicios se ofrecen como Interface y se inicializan a través de la factoría.

Dentro de la capa se encuentra la parte mail para el envío de correos. Para mejorar el tiempo de respuesta dentro del paquete mail se encuentra la clase Mailer que se encarga de administrar el envío de correos a través de nuevos hilos de ejecución. Las clases ThreadMail y MessageMail extienden java.lang.Runneable. Mailer les pone el dato de entrada que deben enviar y en sus propios hilos crean y envían el correo.

Las clases del modelo se emplean para transferir los datos por los métodos son objetos del negocio. Envuelven los datos pero no siguen el patrón DTO. Los objetos de negocio siguen un patrón ActiveRecord Este hecho junto con su diseño de interface fluida permiten realizar líneas de código sencillas y legibles para realizar acciones con los objetos.

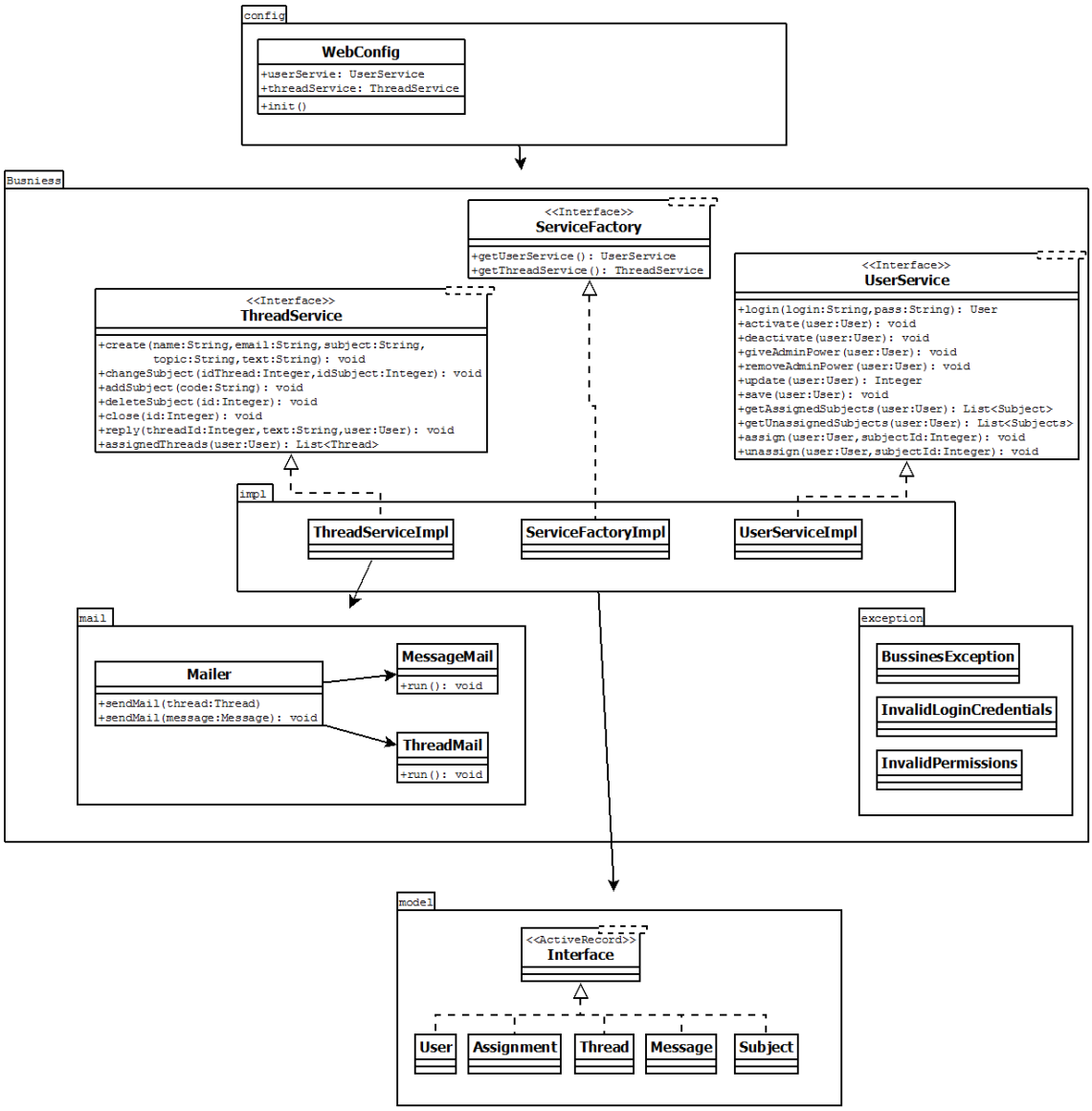

Ilustración 6-6 Diagrama de clases del paquete business

Utilizar el patrón ActiveRecord puede resultar interesante pero tiene una serie de problemas de diseño evidente. No cumple el principio de responsabilidad única, dificultan combinarlo con otros patrones y puede dificultar la testeabilidad de la aplicación. Aun así la comodidad de uso y su simpleza son extremadamente atractivos. Para poder aprovechar sus ventajas sin sufrir sus inconvenientes desde la capa de negocio se actúa con la persistencia como si se usase un auténtico ActiveRecord. Sin embargo este ActiveRecord que hay es una fachada a la siguiente capa de persistencia.

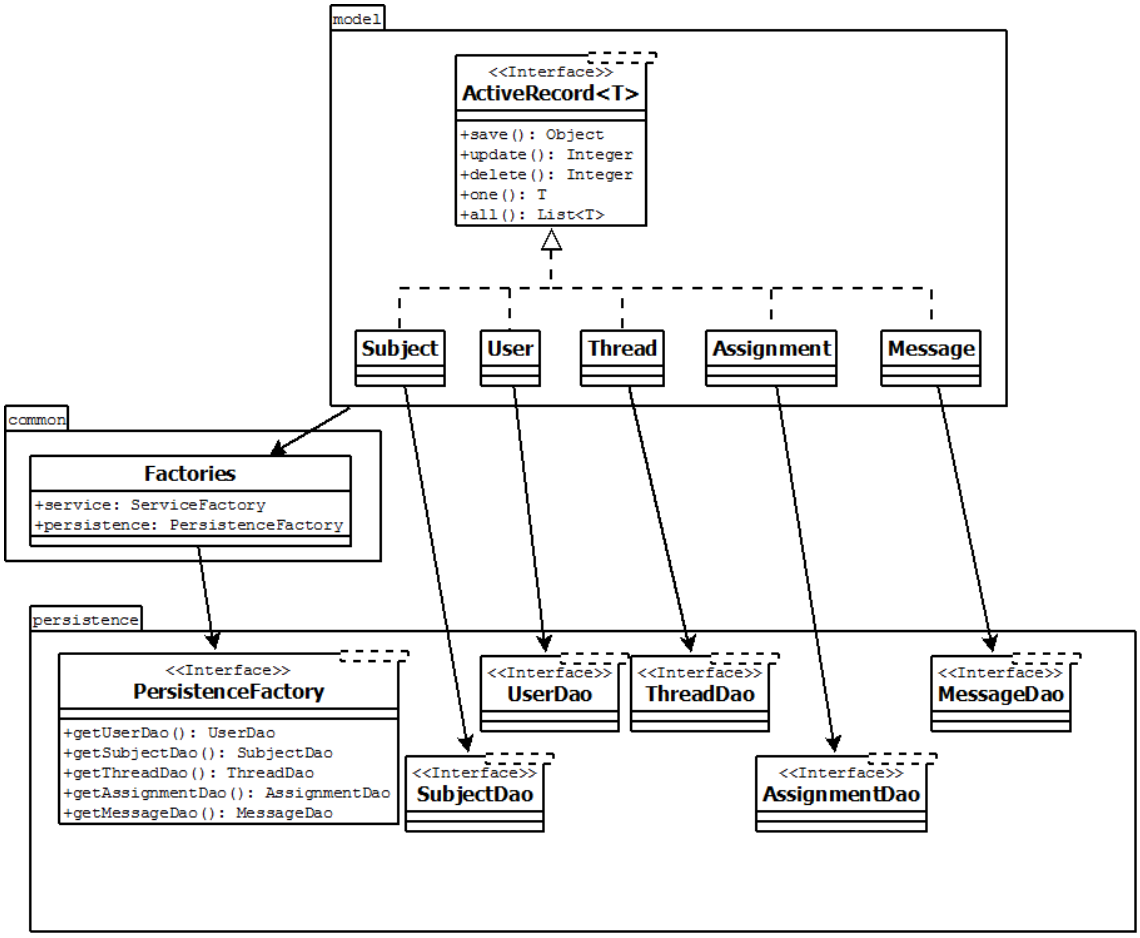

Ilustración 6-7 Diagrama de clases del paquete model

La persistencia es la última capa de todas.

Para independizar totalmente la capa del resto de nuevo se ofrecen interfaces y se esconden las implementaciones en un paquete impl. La instanciación de las clases recae en la factoría.

La persistencia se lleva a cabo con un patrón DAO. En las clases se utilizan sentencia SQL y conexiones creadas en la clase DataBaseCondig. Esto externaliza a la persistencia la conexión (independiente del gestor) y las sentencias (externalizadas).

La clase BussinesException del paquete exception es la excepción específica de esta capa.

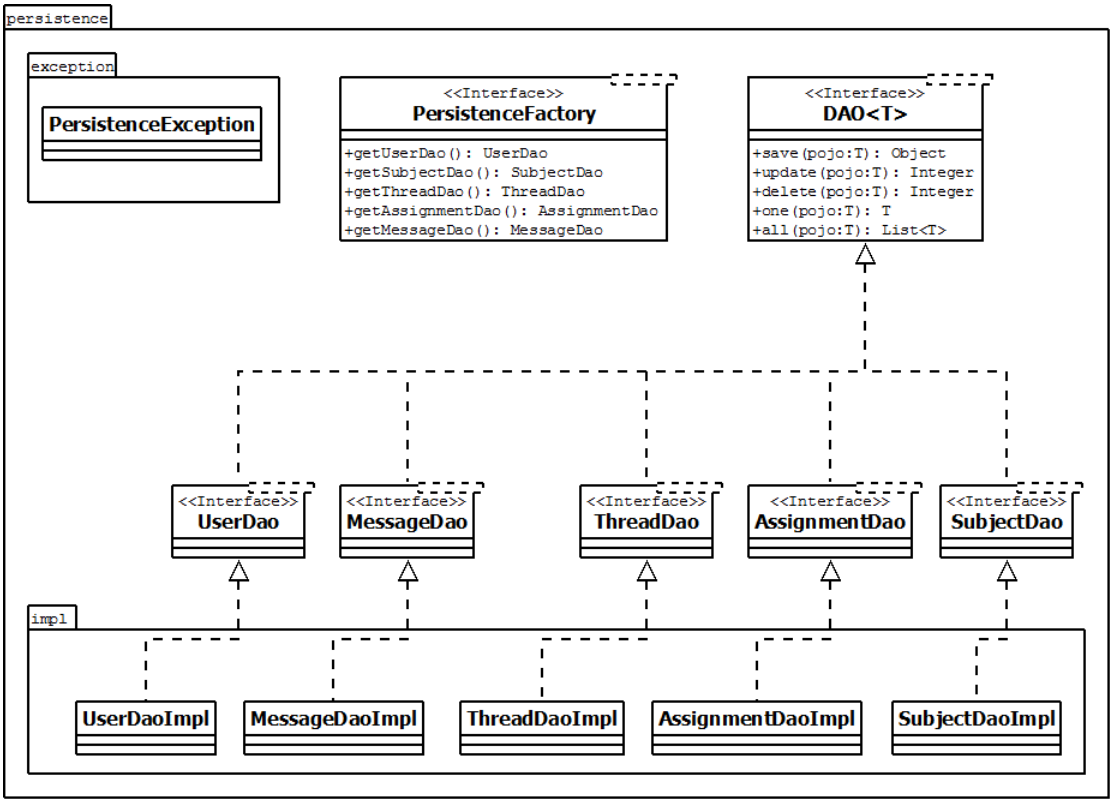

Ilustración 6-8 Diagrama de clases del paquete persistence

6.3 Diseño de la base de datos

6.3.1 Descripción del SGBD

El gestor de la base de datos usado durante el desarrollo es HyperSQL- [1].

Es un gestor de bases de datos relacional escrito en Java con sintaxis SQL estándar, soporta transacciones, procedimientos y triggers SQL. Otras cualidades atractivas son:

- Escaso tamaño en disco.
- Elevada velocidad en acciones SQL básicas.
- No es de pago.

6.3.2 Integración del gestor con el sistema

La aplicación se integra con el gestor de la base de datos por la API jdbc. Además sobre la API jdbc existe una capa de abstracción adicional: sql2o. [7]

sql2o es un framework que simplifica el uso de la API jdbc en gran medida aportando gran parte de la programación relacionada con jdbc dejando al programador únicamente la tarea de declarar las llamadas SQL necesarias. sql2o soporta transacciones y mapeo de objetos en parámetros y resultados de las sentencias SQL.

sql2o realiza la conexión a partir de un DataSource o con una url de la base de datos al estilo jdbc.

Con la externalización de los parámetros de conexión de sql2o se puede integrar el sistema con cualquier base de datos sin realizar cambios. La integración con cualquier base de datos con el sistema requiere de la modificación de los datos de conexión externalizados en ficheros de propiedades y la inclusión en el proyecto de la librería jdbc adecuada a la base de datos.

6.3.3 Diagrama de la base de datos

A continuación se muestra un diagrama de la base de datos con todas las tablas y relaciones.

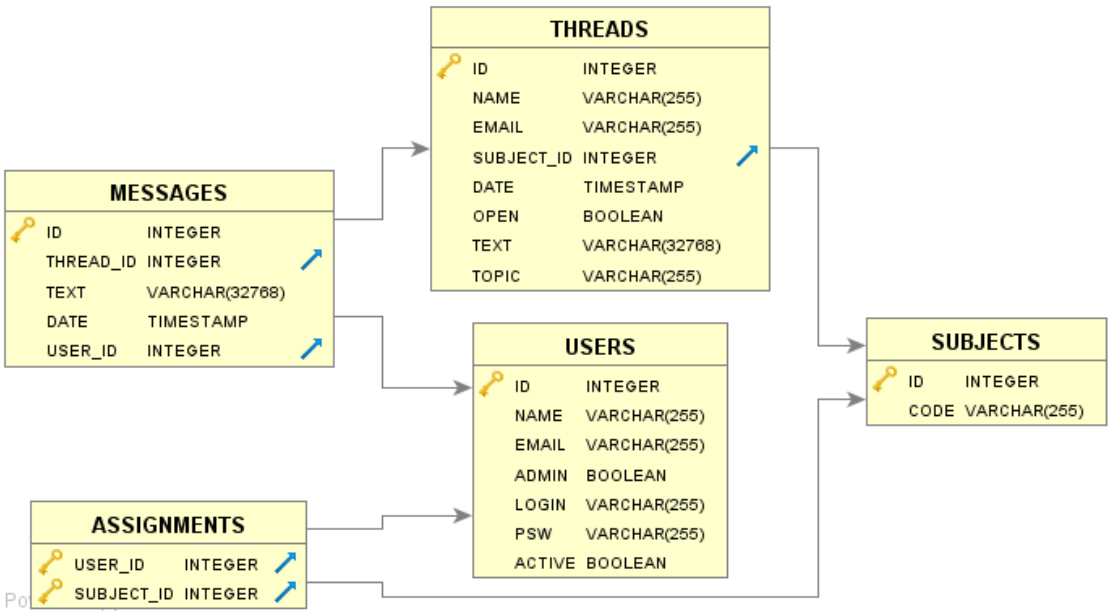

Ilustración 6-9 Diagrama de la base de datos

Existe una copia del script de creación en los apéndices de la documentación.

6.4 Diseño de la interface

Como ya se describió en el análisis la plantilla principal se divide en 4 partes:

- Cabecera
- Menú
- Contenido
- Pie de página

La cabecera muestra únicamente el título de la vista actual.

El pie de página es idéntico para todas las vistas mostrando el nombre de la aplicación y ofreciendo dos enlaces, uno al área privada y otro al formulario de contacto.

Ilustración 6-10 Plantilla de las vistas

El menú ofrecerá siempre la opción del cambio de idioma. Se selecciona el idioma con un desplegable que muestra todos los idiomas disponibles diferenciando el seleccionado actualmente.

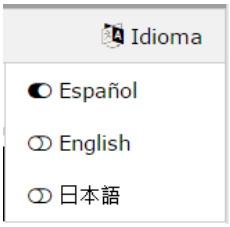

Ilustración 6-11 Menú de idioma

El menú ofrecerá para los usuarios registrados las siguientes opciones:

- Cambio de idioma
- Ir al listado de temas
- Ir al perfil de usuario
- Desconectarse

Adicionalmente el menú ofrecerá una opción para acceder a la administración del sitio a los usuarios administradores.

El contenido será diferente dependiendo de cada vista.

En la vista de temas se ve una tabla con los temas asignados al usuario. Además se ofrecen opciones de ordenación, filtrado y paginación sobre la tabla.

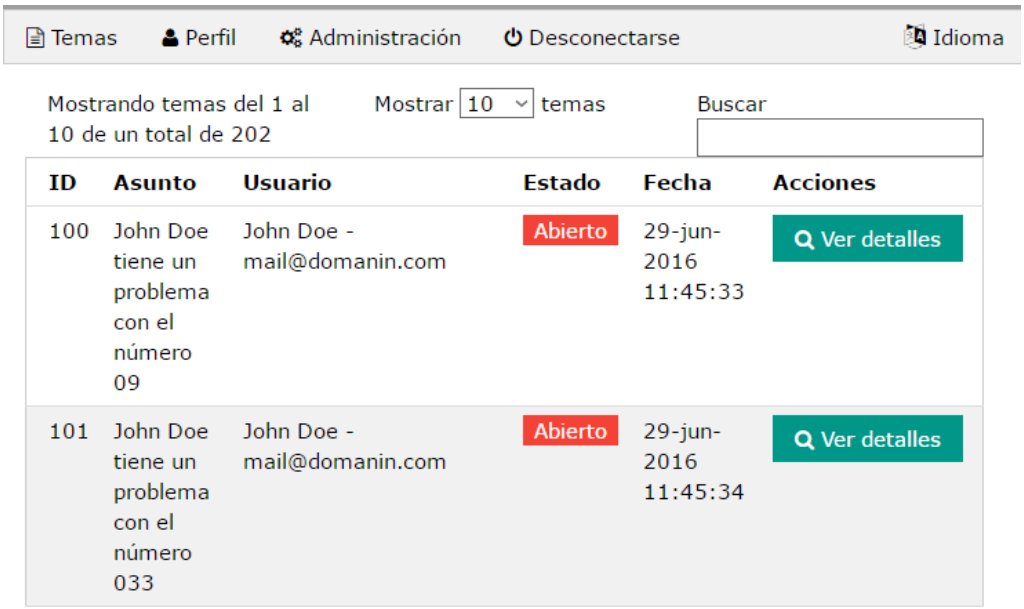

Ilustración 6-12 Vista de temas - Tabla

Los campos de entrada se ven de forma sencilla teniendo todos una etiqueta.

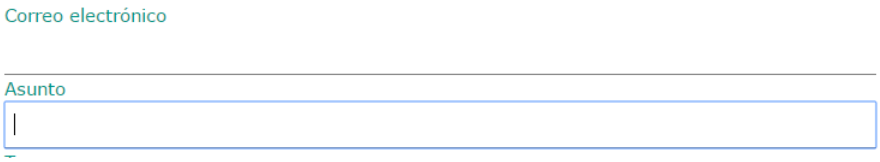

Ilustración 6-13 Muestra de campo de de entrada

Las interface de la vista de los temas mostrará el tema principal, as respuestas y en la parte posterior las opciones.

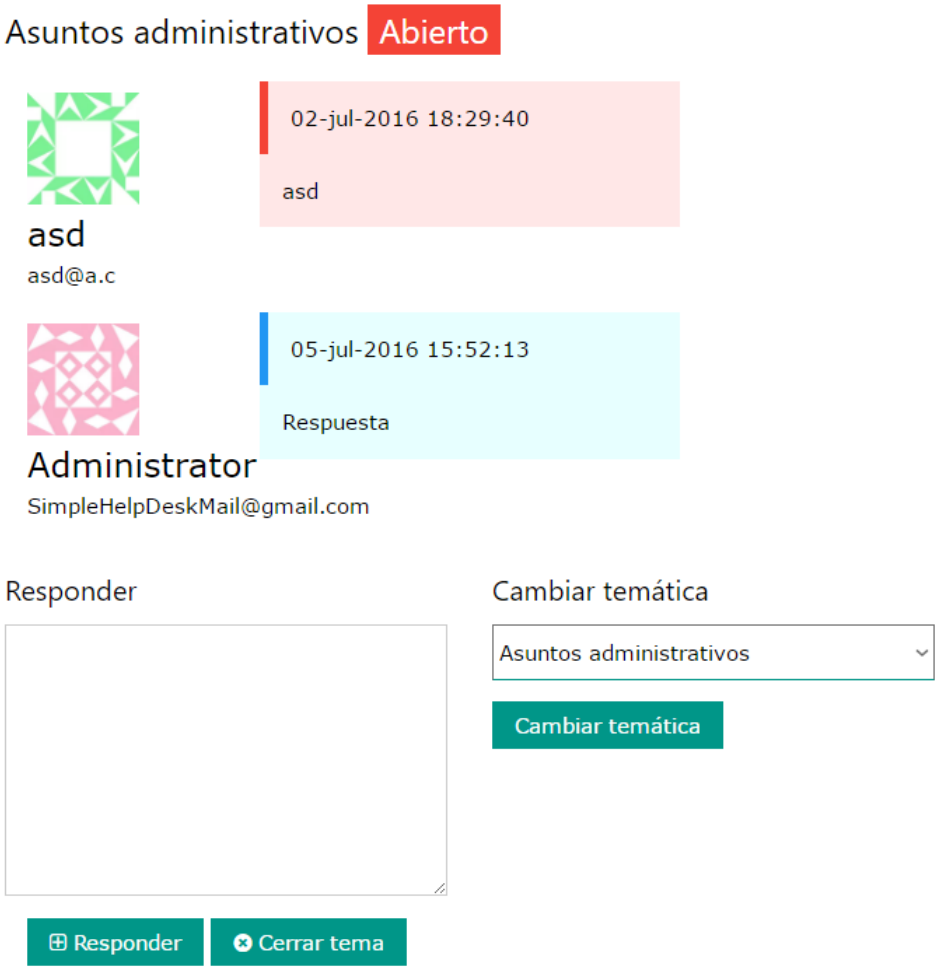

Ilustración 6-14 Vista de tema

La vista del perfil y su edición están separadas en distintas vistas.

La vista de perfil muestra todos los datos del perfil en forma de lista así como su imagen y un botón para acceder a la edición.

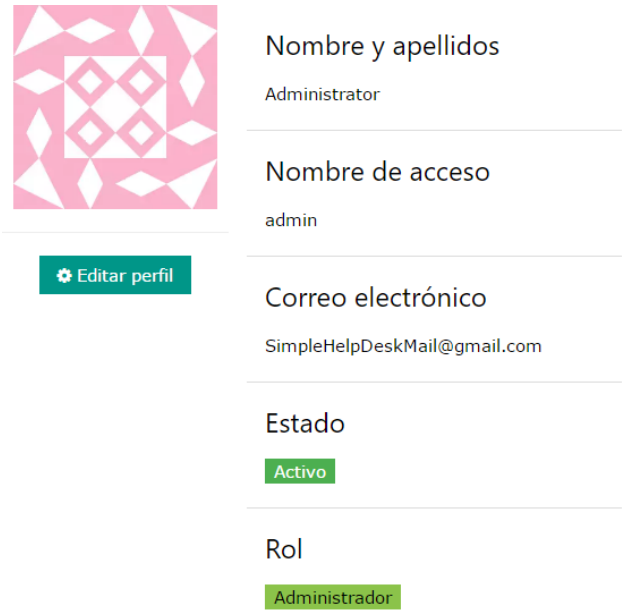

Ilustración 6-15 Vista de perfil

La vista de edición de perfil se divide en cuatro paneles. Los dos primeros, datos y contraseña, disponibles para todos los usuarios. Los siguientes paneles, acciones y asignaciones, se ofrecen únicamente a usuarios administradores. Cada panel es independiente teniendo cada uno una serie de entradas y botones.

Ilustración 6-16 VIsta de edición de perfil

La vista de administrador ofrece un menú que cambia entre dos vistas, los usuarios y las temáticas.

Ilustración 6-17 Menú de administración

La vista de usuarios muestra una lista de usuarios con su correo, imagen y nombre. Para cada uno de ellos hay un enlace para ir a la vista de modificación de dicho perfil. Encima de la lista se encuentra el botón para acceder a la creación de nuevos perfiles.

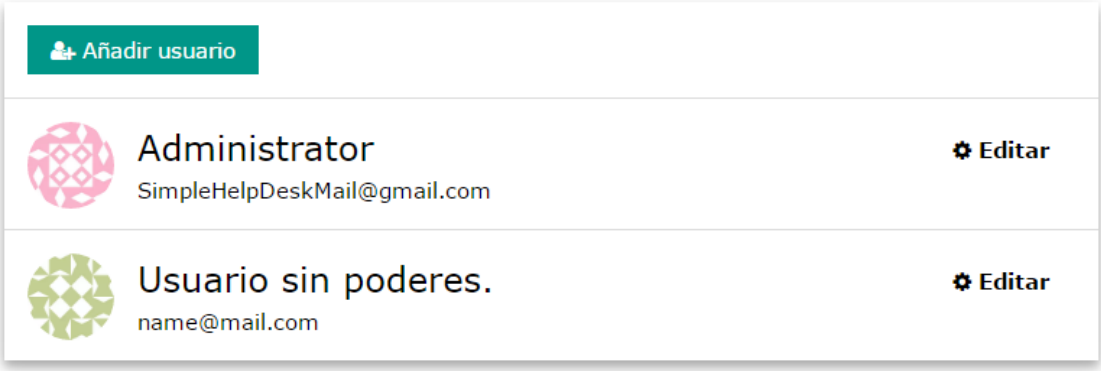

Ilustración 6-18 Lista de usuarios

La vista de categorías es muy similar mostrando menos información y siendo el enlace para eliminar. En vez de mostrar un botón para ir un formulario de creación, la creación de nuevas categorías se realiza en el la misma lista.

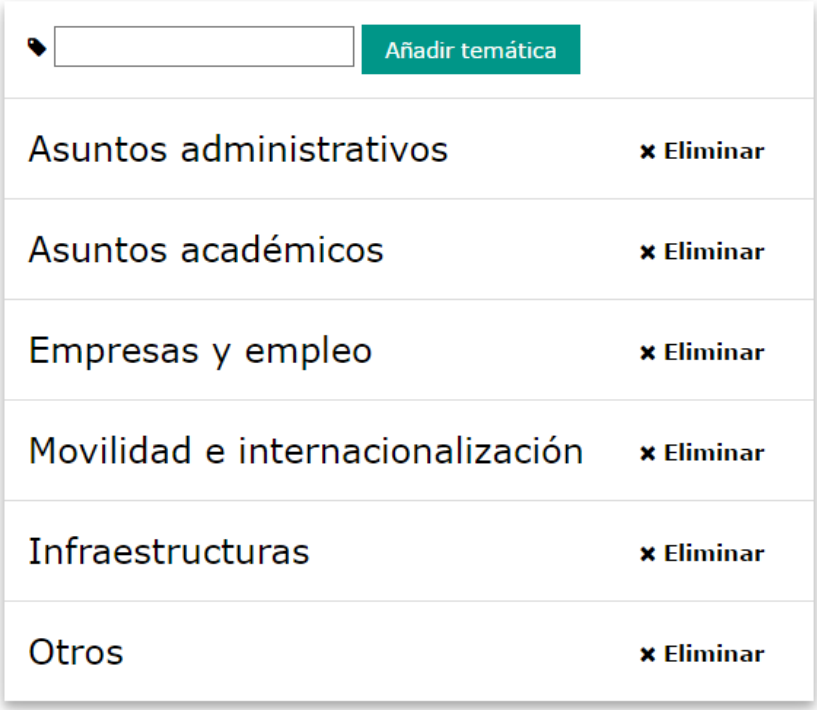

Ilustración 6-19 Lista de temáticas

Los mensajes aparecen siempre al comienzo del contenido dentro de la plantilla.

Son recuadros de color con el mensaje escrito dentro de ellos. Tienen la opción de cerrarse con una cruz en la esquina del recuadro.

Ilustración 6-20 Muestra de mensajes

Para el diseño de la interfaz se han tenido en cuenta factores de usabilidad. La mayoría de las acciones y elementos tienen símbolos que ayudan a distinguirlos y entenderlos.

Además las formas usadas son simples y los colores distinguibles para distintas deficiencias de percepción del color.

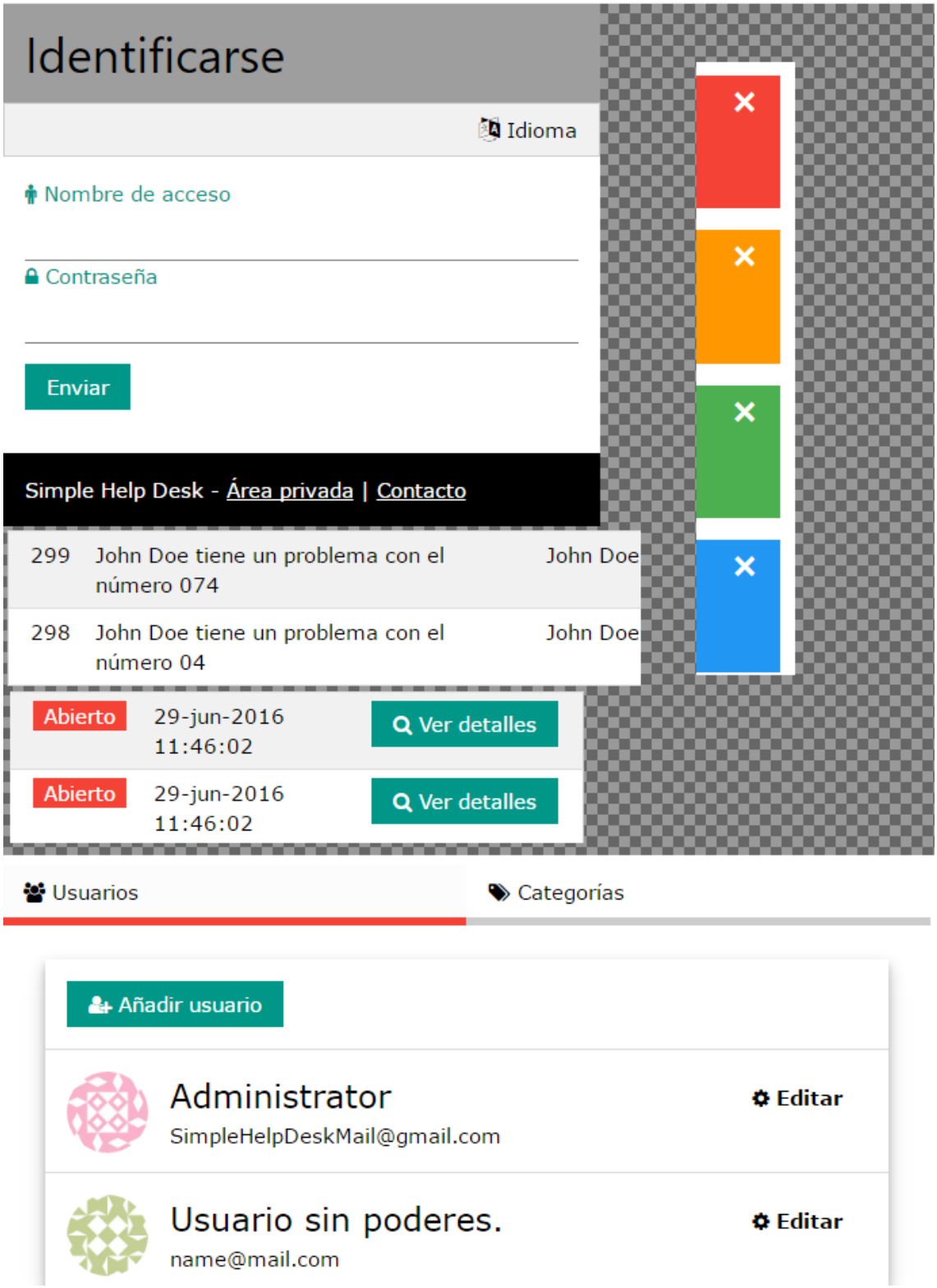

Ilustración 6-21 Muestra de colores y formas

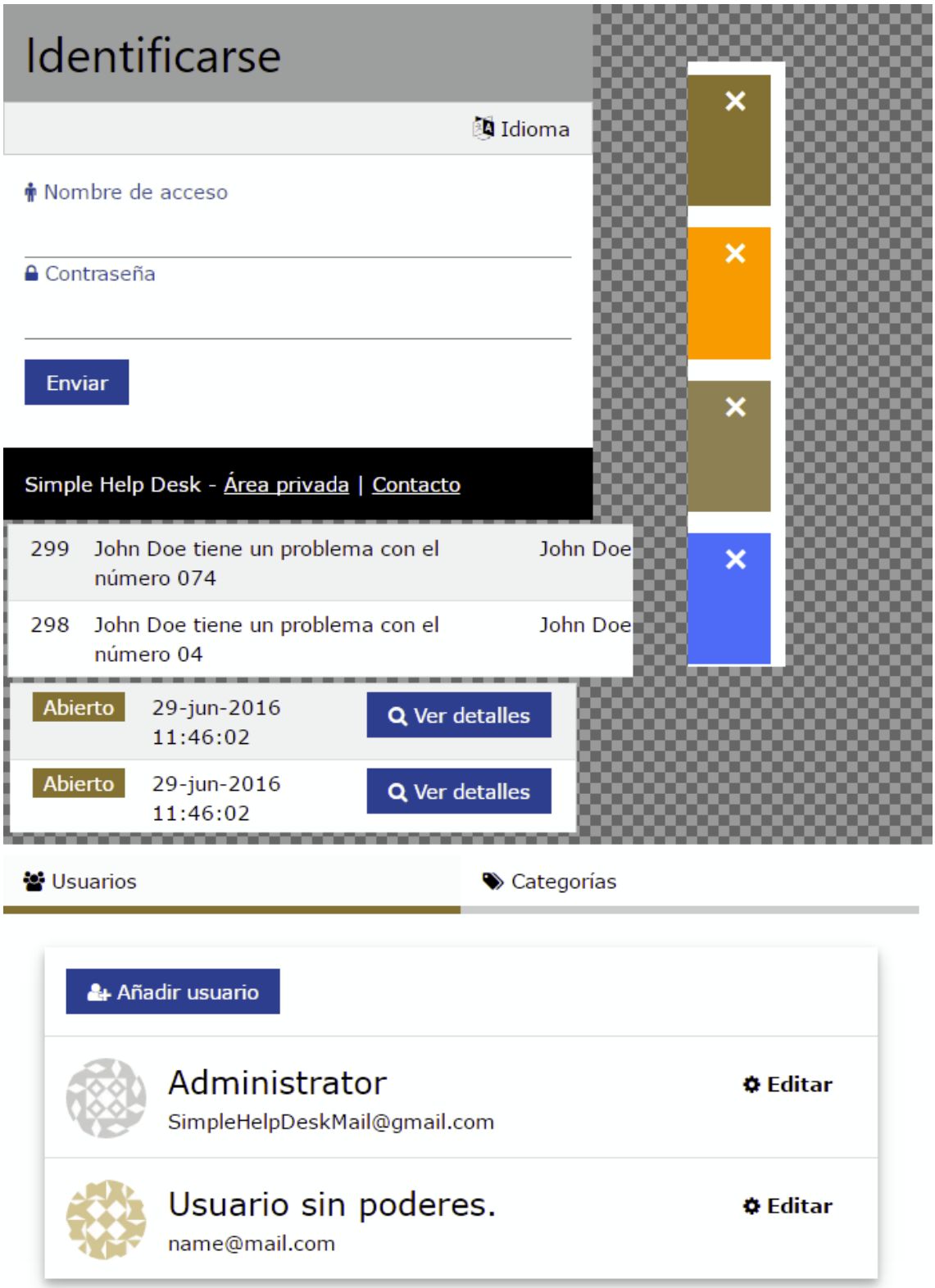

Ilustración 6-22 Muestra de colores y formas con Deuteranopia (ceguera al color verde)

Ilustración 6-23 Muestra de colores y formas con Protanopia (ceguera al color rojo)

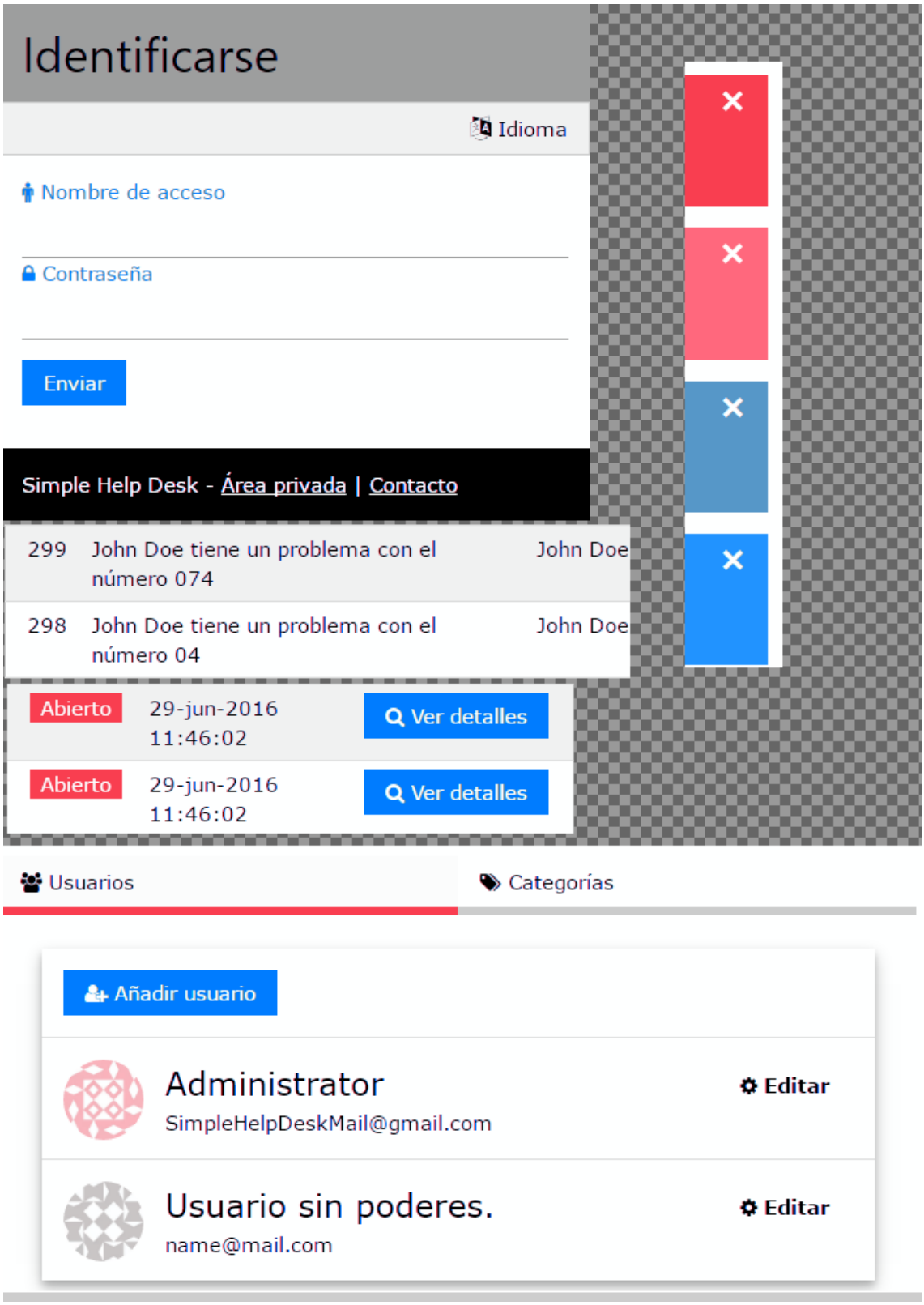

Ilustración 6-24 Muestra de colores y formas con Tritanopia (ceguera al color azul)

6.5 Diseño del plan de pruebas

6.5.1 Pruebas unitarias

Como se especifica en el análisis del sistema, las pruebas unitarias consisten en acciones sencillas del modelo con respecto a la base de datos.

En las pruebas unitarias se utiliza una base de datos en memoria, creando las tablas y su contenido antes de realizar las pruebas.

Se realizan pruebas de todos los métodos CRUD de la base de datos que verifican el correcto funcionamiento de las acciones básicas, las transacciones y las restricciones. Estas pruebas están diseñadas para usar la persistencia que se entrega con el producto. Si se realizan cambios posteriores las pruebas deben ser actualizadas también.

Las pruebas unitarias realizadas son cinco, cada una orientada específicamente a cada elemento o relación del modelo.

6.5.1.1 Prueba de usuarios

En esta prueba se verifica el correcto funcionamiento en los siguientes casos:

Prueba	Resultado esperado
Crear usuario correctamente	Se crea correctamente
Crear usuario con un valor inválido	No se generan cambios
Modificar un usuario con un valor correcto	Se modifica correctamente
Modificar un usuario con un valor incorrecto	No se generan cambios
Buscar un usuario existente	Se obtiene el usuario esperado
Buscar un usuario inexistente	No se obtiene ningún usuario
Eliminar un usuario existente	Se elimina del sistema
Eliminar un usuario inexistente	No se generan cambios
Buscar todos los usuarios	Se obtienen todos los usuarios

6.5.1.2 Prueba de temáticas

En esta prueba se verifica el correcto funcionamiento en los siguientes casos:

Prueba	Resultado esperado
Crear temática	Se crea correctamente
Modificar una temática	Se modifica correctamente
Buscar una temática existente	Se obtiene resultado esperado
Buscar una temática inexistente	No se obtiene ningún resultado
Eliminar una temática existente	Se elimina del sistema
Eliminar una temática inexistente	No se generan cambios

Buscar todos las temáticas	Se obtienen todas las temáticas

6.5.1.3 Prueba de temas

En esta prueba se verifica el correcto funcionamiento en los siguientes casos:

Prueba	Resultado esperado
Crear tema	Se crea correctamente
Modificar un tema	Se modifica correctamente
Buscar un tema existente	Se obtiene resultado esperado
Buscar un tema inexistente	No se obtiene ningún resultado
Eliminar un tema existente	Se elimina del sistema
Eliminar un tema inexistente	No se generan cambios
Buscar todos los temas	Se obtienen todas las temáticas

6.5.1.4 Prueba de mensajes

En esta prueba se verifica el correcto funcionamiento en los siguientes casos:

Prueba	Resultado esperado
Crear mensaje	Se crea correctamente
Modificar un mensaje	Se modifica correctamente
Buscar un mensaje existente	Se obtiene resultado esperado
Buscar un mensaje inexistente	No se obtiene ningún resultado
Eliminar un mensaje existente	Se elimina del sistema
Eliminar un mensaje inexistente	No se generan cambios
Buscar todos los mensajes	Se obtienen todas los mensajes
Eliminar un tema con mensajes	Se obtiene el resultado esperado y un correcto funcionamiento de la transacción

6.5.1.5 Prueba de asignaciones

En esta prueba se verifica el correcto funcionamiento en los siguientes casos:

Prueba	Resultado esperado
Crear asignación	Se crea correctamente
Modificar asignación	Se modifica correctamente
Buscar asignación existente	Se obtiene resultado esperado
Buscar asignación inexistente	No se obtiene ningún resultado
Eliminar asignación existente	Se elimina del sistema
Eliminar asignación inexistente	No se generan cambios
Crear asignación repetida	No se generan cambios

6.5.2 Pruebas de integración y sistema

Con la aplicación compilada se prueba la base de datos que se entrega junto con el prototipo.

Se comprueba que todos los cambios realizados por la aplicación persisten en la base de datos desplegada y que mantiene las restricciones de integridad y respeta las transacciones.

Esto asegura el correcto funcionamiento con la base de datos entregada. Si se realizan cambios posteriores estas pruebas deben ser

Se comprueba el correcto funcionamiento del correo, comprobando, con la cuenta entregada, que los correos llegan a su destino y con correcta forma y contenido.

Se comprueba la correcta generación de la vista en distintos navegadores de internet para asegurar el correcto funcionamiento del motor de generación de vistas HTML.

Se prueba además, como está especificado en el análisis, todos los casos de prueba con todos los usuarios y resultados posibles esperados.

6.5.3 Pruebas de Usabilidad y Accesibilidad

Estas pruebas están enfocadas a determinar el grado de satisfacción de los usuarios con el aspecto, uso y accesibilidad de la aplicación.

Los usuarios que prueban la funcionalidad del sistema deben responder al siguiente cuestionario:

	Totalmente	Casi totalmente	A medias	Poco	Nada
Utilizo el ordenador a menudo					
Utilizo software similar al de la prueba					
Creo que un programa debe ser:					
Intuitivo					
Rápido					
Útil					
Poder navegar la web en un teléfono es importante					
Sufro de alguna deficiencia de percepción					
En el programa probado:					

Los contenidos están bien organizados					
Sé dónde me encuentro dentro de la web					
Me resulta sencillo usar el programa					
El programa se comporta excelentemente					
El tiempo de respuesta es excelente					
El tipo de letra y tamaño es excelente					
Los iconos usados son excelentes					
Los colores usados son excelentes					
La interface es intuitiva y explicativa					
El diseño es claro y atractivo					
El programa está bien estructurado					
Observaciones					

7 Implementación del sistema

7.1 Lenguajes de programación, normas y estándares usados

A continuación se muestran los lenguajes de programación y formatos usados durante el desarrollo de la aplicación:

7.1.1 Java

Java es un lenguaje de programación desarrollado por James Gosling de Sun Microsystem (adquirida por Oracle) y publicado en 1995 formando parte de la plataforma Java Sun Microsystems.

Java es un lenguaje de programación de uso general, concurrente y orientado a objetos.

Los programas escritos en java generalmente se compilan a bitcode ejecutable en cualquier máquina virtual de java (JVM), haciéndolo multiplataforma. Un programa en java se puede ejecutar en una JVM sin importar el sistema operativo ni la arquitectura de la maquina anfitriona.

La versión usada es Java8 por sus prestaciones y ventajas de seguridad sobre las anteriores versiones. [13]

7.1.2 HTML

El HTML, del inglés *HyperText Markup Language*, es un lenguaje de marcado para la descripción de páginas web. Los elementos descritos en el HTML son interpretados por los navegadores para generar la vista de páginas web.

7.1.3 CSS

CSS, del inglés *Cascading Style SheetsI*, es un lenguaje usado para definir la presentación de un documento escrito en un lenguaje de marcado xml. Se usa fundamentalmente para acompañar al HTML y definir su presentación.

7.1.4 Freemarker

Freemarker es un framework para la generación de HTML dinámico a partir de plantillas HTML y marcado específico freemarker. [8]

7.1.5 SQL

SQL, del inglés *Structured Query Language*, es un lenguaje declarativo de acceso a bases de datos relacionales. Utilizando algebra relacional se puede con SQL realizar consultas y modificaciones sobre bases de datos relacionales.

7.1.6 JavaScript

JavaScript es un lenguaje de programación interpretado, dinámico y de tipado débil. Estandarizado en todos los navegadores modernos y definido como el lenguaje de script por defecto en HTML5, JavaScript es un elemento fundamental en el lado cliente para ofrecer mejoras de visibilidad y funcionalidad en las páginas web.

7.2 Frameworks y librerías

7.2.1 Spark

Spark es un micro framework java para la creación de servicios web, incluyendo un gran abanico de posibilidades de uso como páginas web, micro-servicios, APIs REST...

Es un framework MVC action-based en el que se definen las acciones y respuestas de las rutas programáticamente aprovechando las ventajas de la versión 8 de Java. [9]

Es importante no confundirlo con ApacheSpark, un motor de computación en la nube. [12]

7.2.2 Freemarker

Freemarker es un framework para la generación de HTML dinámico a partir de plantillas HTML y marcado específico freemarker. Escogido junto con Spark por su sencillez y alta compatibilidad. [8]

7.2.3 sql2o

sql2o es un framework java para realización de consultas SQL. Simplifica el uso de la API jdbc en gran medida aportando gran parte de la programación específica de jdbc dejando al programador únicamente la tarea de declarar las llamadas SQL necesarias. sql2o soporta transacciones y mapeo de objetos en parámetros y resultados de las sentencias SQL. [7]

7.2.4 JUnit

JUnit es un framework de realización de pruebas repetibles y automáticas. Es la implementación de xUnit para java. Permite realizar pruebas unitarias sobre el código y el el framework de pruebas en java más extendido. [14]

7.2.5 SLF4J

Simple Logging Facade for Java, o SLF4J para abreviar, es una librería java que ofrece una fachada sobre el sistema de loggin estándar de una aplicación. [17]

7.2.6 jBCrypt

jBCrypt es una librería de cifrado de claves para Java. [18]

7.2.7 commons-email

commons-email es una librería que forma parte del conjunto de librerías common de Apache para java. Es una librería específica para el manejo de correos electrónicos que se ejecuta sobre javax.mail de Oracle. [19]

7.2.8 Gravatar

Gravatar es un servicio web que ofrece alojamiento de imágenes y reconocimiento de imágenes de perfil o avatares a nivel global. Esta aplicación no utiliza librerías específicas pero si realiza comunicación y llamadas a la API pública de Gravatar. [20]

7.2.9 W3CSS

W3CSS es un framework CSS gratuito distribuido por w3school como alternativa potente, ligera y más sencilla a Bootstrap, otro framework css muldialmente conocido.

W3CSS ofrece un estilo sencillo y elegante, compatible con la mayoría de los navegadores de internet y preparado para visualizaciones en dispositivos móviles y usando únicamente CSS estándar. [21]

7.2.10 FontAwecome Icons

FontAwesome Icons es un framwork CSS para la inclusión de iconos vectoriales customizables por CSS en páginas web. [22]

7.2.11 jQuerry

jQuerry es un framework JavaScript de uso específico para HTML. Simplifica y especializa JavaScript para un manejo más potente y sencillo del HTML. jQuerry es además la base para muchos Frameworks y librerías JavaScript disponibles para la web. [23]

7.2.12 DataTables

Datatables es un pluggin jQuerry que añade funcionalidad a las tablas HTML como filtrado, paginación... Es potente, ajaxificado y totalmente customizable. [24]

7.3 Herramientas y programas usados

7.3.1 Microsoft Office

Microsoft Office es la suite ofimática de Microsoft para aplicaciones de oficina. Ofrece dos herramientas usadas durante el desarrollo del proyecto: Word, un procesador de texto para la creación de documentos, y Excel, un gestor de hojas de cálculo usado para el cálculo de presupuestos.

7.3.2 Dia

Dia es un programa de propósito general para la creación de diagramas que forma parte del proyecto GNOME. Está planteada como una herramienta modular para cumplir con un distinto rango de necesidades. Se puede usar para crear diferentes tipos de diagrama, incluyendo diagramas UML, diagramas de flujo...

7.3.3 yuml

yuml es una aplicación web para la generación de diagramas de casos de uso online a partir de definiciones textuales. [25]

7.3.4 GIMP

GIMP es un programa de manipulación de imágenes gratuito que forma parte del proyecto GNU. Se ha usado para la composición de algunas imágenes del manual, aplicar filtros de daltonismo y mejorar los diagramas construidos con otras herramientas.

7.3.5 Maven

Maven es un administrador de proyectos software de Apache. Potente y customizable es usado en la aplicación como gestor de dependencias, para compilar el producto final y para ejecutar las pruebas unitarias.

7.3.6 IntelliJ IDEA

IntelliJ IDEA es un IDE para java. Tiene una gran cantidad de pluggins que extienden su ya extenso número de herramientas y está disponible en versión gratuita y comercial.

7.3.7 HSQLDB

HyperSQL es la base de datos usada durante el desarrollo de la aplicación. Es una base de datos relacional rápida y ligera que se puede usar en memoria o como servidor. Es usada en memoria para realizar las pruebas en un entorno estéril y usada como servidor en la aplicación final.

7.3.8 DbVisualizer

Esta es una aplicación que, de manera gráfica, ayuda durante todo el proceso de diseño, creación y mantenimiento de la base de datos. Es compatible con cualquier base de datos compatible con un driver JDBC.

7.3.9 Opera

Opera es el navegador de internet usado para explotar la aplicación en su fase de desarrollo y pruebas. Es gratuito, potente, extensible con pluggins y fácil de usar. Tiene herramientas de desarrollo que han sido útiles durante el proceso. No ha sido el único navegador usado para probar la aplicación pero sí el principal.

7.4 Creación del sistema

7.4.1 Problemas encontrados

Los problemas más graves encontrados durante el desarrollo de la aplicación fueron aquellos relacionados con la codificación y con el despliegue.

7.4.1.1 La crisis de las cookies

Este fue el principal problema en la aplicación en lo referencie a la codificación.

Los mensajes enviados para la vista del cliente eran enviados en cookies. Estos mensajes contenían caracteres Unicode ya que la aplicación está totalmente preparada para la internacionalización. El problema aparece en el proceso de manipulación de las cookies por los navegadores:

- Opera y Chrome: cookies en UTF-8
- Internet Explorer: cookies en la codificación local del ordenador
- Firefox: cookies en UTF-16 pero sin seguir el estándar
- Safari: No recibe ni envía cookies que no contengan exclusivamente ASCII

Esto causó un cambio que complicó el sistema en envío de mensajes a las afectando a todas y cambiando el modo de trabajo de la comunicación con el cliente.

La solución final fue recurrir a administrar atributos en la sesión en vez de enviar cookies volátiles.

7.4.1.2 Externalización total con Maven

Con la total externalización de las propiedades de configuración de la aplicación apareció el siguiente problema. Este problema fue complicado por la falta de experiencia en el uso de pluggins de Maven.

El otro problema residía en la correcta configuración de Maven para utilizar los ficheros de propiedades en una ubicación distinta en estado y producción, empaquetando los ficheros de propiedades internos a la aplicación en una ubicación externa al archivo ejecutable.

La solución se encontró por prueba y error de distintas configuraciones de los pluggins de empaquetado de dependencias y de recursos.

7.5 Descripción de los aspectos importantes

A continuación se muestran los aspectos más importantes desarrollados con la aplicación que necesitan una explicación más avanzada o una mención especial.

7.5.1 Vistas freemarker

Las vistas desarrolladas con freemarker son importantes pues es la parte visible de la aplicación. Las vistas utilizan un marcado específico que permite acceder a objetos, iterar sobre colecciones y calcular condiciones. Permite hacer inserciones y llamadas desde ficheros externos a plantillas y de macros definidas en otros ficheros.

El motor de vistas junta un mapa de objetos con la plantilla de la vista para generar una vista dinámica. Los ficheros de freemarker usan la extensión .ftl.

7.5.2 Clase App

La clase App es la clase principal de la aplicación. Procesa los argumentos de entrada y configura la aplicación para realizar las acciones especificadas con las opciones descritas por los argumentos de entrada.

7.5.3 Clases de configuración

Las clases de configuración son clases estáticas que cumplen la función de leer la configuración de ficheros externos y ofrecerla a toda la aplicación. Estas clases permiten tener un solo punto todos los aspectos de configuración de la aplicación relacionados con cada apartado de la aplicación: configuración general, configuración del correo, configuración de la base de datos, configuración de la internacionalización...

Una vez se inician ofrecen acceso a las entradas de los ficheros allá donde sean necesarias.

Además la configuración se realiza externa a la aplicación que estas clases leen desde ficheros de propiedades externos.

7.5.4 Clase WebConfig

Esta esta clase actúa como el corazón de la aplicación: la clase encargada de configurar el controlador utilizando el framework web Spark.

Ofrece configuración del servidor, el control de la sesión y se encarga de inicializar las clases que definen programáticamente las rutas, respuestas y filtros en Spark.

Para definir las acciones se llaman a métodos de Spark:

Spark.get(ruta, acción)

Esto especifica que cuando se realiza una petición get a una ruta determinada se realiza la acción especificada. En spark se puede definir todos los métodos de peticiones http (get, post update, delete, patch).

Spark permite definir filtros de la misma manera.

Spark.before(ruta, acción)

Esto permite realizar una acción antes de procesar la petición a una ruta determinada. Exciten otros tipos de filtros como after, que se ejecuta tras procesar la petición.

Entre los otros muchos métodos de Spark como por ejemplo Spark.exception(Exception.class, acción) que permite especificar comportamientos cuando se producen excepciones de ese tipo.

8 Desarrollo de las pruebas

8.1 Pruebas unitarias

Las pruebas unitarias prueban el código de todas las clases del modelo de dominio en conjunto realizando una prueba especializada en una clase por cada clase existente.

Como se especifica en plan de pruebas, las pruebas unitarias se realizan cada vez que se compila la aplicación.

Al compilar correctamente se han realizado con éxito las pruebas unitarias especificadas en el plan de pruebas:

- Pruebas de usuarios
- Pruebas de temáticas
- Pruebas de Temas
- Pruebas de mensajes
- Pruebas de asignaciones

8.2 Pruebas de Integración y del sistema

El resultado de las pruebas es el siguiente:

Caso de Uso 1: Enviar comentario		
Prueba	Resultado Esperado	Cumple resultado esperado
Rellenar todos los datos del formulario y enviar	El sistema registra un tema más	Sí
Dar a enviar sin terminar de rellenar el formulario	Se remarcan las entradas sin rellenar y se muestra un mensaje	Sí

Caso de Uso 2: Acceder al sistema		
Prueba	Resultado Esperado	Cumple resultado esperado
Rellenar el formulario de acceso con los datos de un usuario administrador	Se entra en el sistema	Sí
Rellenar el formulario de acceso con los datos de un usuario normal	Se entra en el sistema	Sí
Rellenar el formulario de acceso con los datos de un usuario escribiendo la contraseña incorrectamente	No se entra en el sistema	Sí
Pulsar enviar sin escribir un dato	No se envía el formulario y se remarca el dato faltante	Sí
Rellenar con datos de un usuario existente desactivado	No se entra al sistema	Sí

Caso de Uso 3: Desconectarse

Prueba	Resultado Esperado	Cumple resultado esperado
Tras entrar en sesión pulsar el botón del menú Desconectarse	Se sale del sistema y se va al formulario de acceso	Sí

Caso de Uso 4: Ver tema		
Prueba	Resultado Esperado	Cumple resultado esperado
Se accede con un usuario administrador a un tema a partir de la lista de temas	Se muestra el tema	Sí
Se accede con un usuario normal a un tema a partir de la lista de temas	Se muestra el tema	Sí
Se accede con un usuario cualquiera al tema a parir del ID en la dirección	Se muestra el tema	Sí
Se acceder a un tema por su ID a partir de la dirección sin iniciar sesión	Se envía al usuario a la vista de inicio de sesión	Sí
Se accede con un usuario cualquiera al tema a parir del ID en la dirección siendo una ID inexistente	Se muestra un mensaje de error	Sí

Caso de Uso 5: Responder		
Prueba	Resultado Esperado	Cumple resultado esperado
Administrador envía respuesta	Aparece la respuesta en la vista	Sí
Usuario asignado envía respuesta	Aparece la respuesta en la vista	Sí

Usuario no asignado envía respuesta	Se muestra un error	Sí

Caso de Uso 6: Cerrar tema		
Prueba	Resultado Esperado	Cumple resultado esperado
Administrador cierra tema	Aparece el tema como resuelto	Sí
Usuario asignado cierra	Aparece el tema como resuelto	Sí
Usuario no asignado cierra	Se muestra un error	Sí

Caso de Uso 7: Cambiar temática a un tema		
Prueba	Resultado Esperado	Cumple resultado esperado
Administrador cambia la temática al tema	Se actualiza la temática	Sí
Usuario asignado cambia la temática al tema	Se actualiza la temática	Sí
Usuario no asignado cambia la temática al tema	Se muestra un error	Sí

Caso de Uso 8: Ver perfil		
Prueba	Resultado Esperado	Cumple resultado esperado
Administrador ve su propio perfil	Se muestra el perfil con todos los datos y el botón de editar	Sí
Administrador ve el perfil de otro	Se muestra el perfil con todos los datos y el botón de editar	Sí
Usuario ve su propio perfil	Se muestra el perfil con todos los datos y el botón de editar	Sí
Usuario ve el perfil de otro	Se muestra el perfil con datos simplificados	Sí
Cualquier usuario accede a la vista de un perfil que no existe	Se muestra un mensaje de error	Sí

Caso de Uso 9: Modificar datos		
Prueba	Resultado Esperado	Cumple resultado esperado
Administrador modifica sus datos correctamente	Se modifica y muestra un mensaje mostrando un aviso de éxito	Sí
Administrador modifica su contraseña correctamente	Se modifica y muestra un mensaje mostrando un aviso de éxito	Sí
Administrador modifica sus datos incorrectamente	No se modifica y muestra un mensaje de error	Sí
Administrador modifica su contraseña incorrectamente	No se modifica y muestra un mensaje de error	Sí
Administrador modifica los datos de otro usuario correctamente	Se modifica y muestra un mensaje mostrando un aviso de éxito	Sí
Administrador modifica la contraseña de otro usuario correctamente	Se modifica y muestra un mensaje mostrando un aviso de éxito	Sí
Administrador modifica los datos de otro usuario incorrectamente	No se modifica y muestra un mensaje de error	Sí
Administrador modifica la contraseña de otro usuario incorrectamente	No se modifica y muestra un mensaje de error	Sí
Usuario modifica sus datos correctamente	Se modifica y muestra un mensaje mostrando un aviso de éxito	Sí
Usuario modifica su contraseña correctamente	Se modifica y muestra un mensaje mostrando un aviso de éxito	Sí

Usuario modifica sus datos incorrectamente	No se modifica y muestra un mensaje de error	Sí
Usuario modifica su contraseña incorrectamente	No se modifica y muestra un mensaje de error	Sí

Caso de Uso 10: Cancelar usuario		
Prueba	Resultado Esperado	Cumple resultado esperado
Administrador cancela su usuario	Se cancela el usuario y actualiza la vista	Sí
Usuario cancela su usuario	Se cancela el usuario y actualiza la vista	Sí
Administrador cancela otro usuario	Se cancela el usuario y actualiza la vista	Sí

Caso de Uso 11: Dar privilegios de administrador		
Prueba	Resultado Esperado	Cumple resultado esperado
Administrador da privilegios a otro usuario	Se otorgan privilegios al usuario y se actualiza la vista	Sí

Caso de Uso 12: Quitar privilegios de administrador		
Prueba	Resultado Esperado	Cumple resultado esperado
Administrador quita privilegios a otro usuario	Se eliminan los privilegios al usuario y se actualiza la vista	Sí

Caso de Uso 13: Quitar privilegios de administrador		
Prueba	Resultado Esperado	Cumple resultado esperado
Administrador quita privilegios a otro usuario	Se eliminan los privilegios al usuario y se actualiza la vista	Sí

Caso de Uso 14: Asignar temática		
Prueba	Resultado Esperado	Cumple resultado esperado

Administrador asigna una temática a otro usuario	Se asigna la temática al usuario y se actualiza la vista	Sí
Administrador asigna una temática a otro que ya haya sido asignado a esa temática entre que el administrador empezase a editar el perfil y el momento de que realizase la acción	Se muestra un error	Sí

Caso de Uso 15: Desasignar temática		
Prueba	Resultado Esperado	Cumple resultado esperado
Administrador desasigna una temática a otro usuario	Se desasigna y se actualiza la vista	Sí
Administrador desasigna una temática a otro que ya haya sido entre que el administrador empezase a editar el perfil y el momento de que realizase la acción	Se muestra un error	Sí

Caso de Uso 16: Activar usuario		
Prueba	Resultado Esperado	Cumple resultado esperado
Administrador cancelar una cuenta de usuario	Se actualiza la vista y se cancela el usuario en la base de datos	Sí

Caso de Uso 17: Crear temática		
Prueba	Resultado Esperado	Cumple resultado esperado

Administrador crea una temática desde la administración	Se crea la temática en la base de datos y actualiza la vista	Sí

Caso de Uso 18: Eliminar temática		
Prueba	Resultado Esperado	Cumple resultado esperado
Administrador elimina una temática en desuso desde la administración	Se elimina la temática en la base de datos y actualiza la vista	Sí
Administrador elimina una temática en uso desde la administración	Se muestra un mensaje de error	Sí

Caso de Uso 19: Crear usuario		
Prueba	Resultado Esperado	Cumple resultado esperado
Administrador crea un usuario desde la administración	Se crea un usuario en la base de datos y redirige a la administración	Sí
Administrador crea un usuario desde la administración con datos erróneos	Se muestra un mensaje de error	Sí

Los correosos se envían correctamente en forma y contenido: Sí.

Los cambios se ven reflejados de forma instantánea en la base de datos: Sí.

8.3 Pruebas de usabilidad y accesibilidad

El mismo grupo de voluntarios que realizó las pruebas de sistema han rellenado una encuesta relacionada con la usabilidad y accesibilidad de la página.

Los individuos se representan como A y B.

	Totalmente	Casi totalmente	A medias	Poco	Nada
Utilizo el ordenador a menudo	A B				
Utilizo software similar al de la prueba					A B
Creo que un programa debe ser:					
Intuitivo	A B				
Rápido	B	A			
Útil	B	A			
Poder navegar la web en un teléfono es importante		B		A	
Sufro de alguna deficiencia de percepción			B	A	
En el programa probado:					
Los contenidos están bien organizados	A B				
Sé dónde me encuentro dentro de la web	A B				
Me resulta sencillo usar el programa	A B				
El programa se comporta excelentemente	A	B			
El tiempo de respuesta es excelente	B	A			
El tipo de letra y tamaño es excelente	A B				
Los iconos usados son excelentes	A B				
Los colores usados son excelentes	A B				
La interface es intuitiva y explicativa	A	B			
El diseño es claro y atractivo	A B				
El programa está bien estructurado	A	B			
Observaciones:					
A	Ninguna.				
B					

9 Manuales del sistema

9.1 Manual de usuario

9.1.1 Manual de instalación

Para poder usar la aplicación se requiere una máquina virtual de Java compatible con la versión 8 o superior.

La aplicación no requiere un proceso de instalación. Se copiará o descomprimirán los archivos entregados en un directorio. Este directorio se considera el directorio de instalación.

Antes de la primera ejecución se debe configurar el correo, base de datos, primer usuario administrador. Ya se entregan con una configuración por defecto.

Los ficheros de configuración se encuentran en el directorio *conf* dentro del directorio de instalación.

9.1.1.1 Correo

La configuración del correo se encuentra en el fichero de configuración *mail.properties*.

En dicho fichero se configura el correo usado por la aplicación. Por defecto viene configurado con una cuenta de Gmail: `SimpleHelpDeskMail@gmail.com`.

9.1.1.2 Base de datos

La base de datos distribuida con la aplicación es una HSQLDB. Ya viene configurada.

Su configuración se encuentra en el fichero *sql.properties*.

Utilizar otra base de datos es un proceso más complejo que necesita volver a compilar el programa. Para cambiar la base de datos consulte el manual de programador.

9.1.1.3 Servidor

El servidor incluido en la aplicación sólo requiere configurar el puerto. Este se puede configurar en el fichero *app.properties*.

9.1.1.4 Primer uso

La primera vez que se utiliza ejecuta la aplicación es importante utilizar el siguiente comando:

```
java -jar SimpleHelpDesk<VERSION>.jar config
```

Este comando configurará un usuario administrador y unas temáticas predefinidas en el fichero.

9.1.1.5 Despliegue

Para desplegar la aplicación y el servidor se usa una terminar o consola de comandos. También se pueden crear ficheros de scripts para lanzar la aplicación más cómodamente.

La aplicación se lanza con la siguiente línea de comando:

```
java –jar SimpleHelpDesk<VERSION>.jar [OPCIONES]
```

Las opciones son las siguientes:

Sin opciones	Inicia el servidor con logs a nivel INFO
`run`	Inicia el servidor
`config`	Inicia la configuración añadiendo el usuario y temáticas especificadas en app.properties.
`soloAdmin`	Modifica el proceso de configuración para que solo añada el usuario especificado en app.properties y no las temáticas.
`warn` | `info` | `debug` | `trace`	Configura los logs al nivel del comando.

Tabla 9.1 – Comandos de opciones

Se pueden usar varias opciones a la vez.

Ejemplos:

- Iniciar servidor con logs en nivel debug:
  - java -jar `SimpleHelpDesk`<Version>.jar run debug
- Realizar la configuración de un nuevo administrador sin iniciar el servidor.
  - `java -jar SimpleHelpDesk<Version>.jar config soloAdmin`
- Iniciar el servidor con logs en nivel info.
  - `java -jar SimpleHelpDesk<Version>.jar`

9.1.1.6 Idiomas adicionales

Para instalar idiomas adicionales se necesita crear un fichero con el nombre <tag>.propesties. Estos ficheros deben tenerlo las mismas claves que el resto de ficheros de idioma entregados con el producto.

<tag> se corresponde con la etiqueta del idioma a añadir. Debe ser una etiqueta del código de idiomas IETF.

Además debe añadirse dicha etiqueta al fichero de configuración app.properties en la propiedad languages. Los idiomas especificados en dicha propiedad aparecerán en la aplicación en el orden que aparecen en el fichero, siendo el primero el de por defecto.

9.1.2 Manual de usuario

Una vez iniciado el servidor la aplicación se puede usada sin problemas a partir de la dirección del servidor.

9.1.2.1 Mandar incidencia

Se pueden mandas incidencias desde la dirección base de la aplicación web o desde /help.

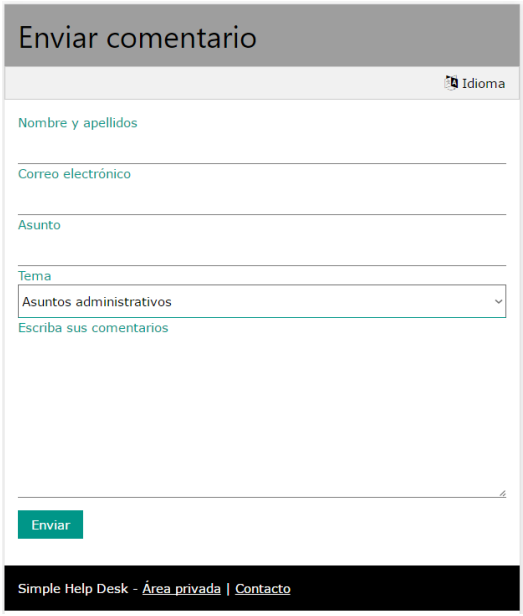

Ilustración 9-1 Formulario de contacto

Tras rellenar todo el formulario al pulsar enviar se crea un nuevo tema.

Si se han rellenado todos los datos correctamente se ve la siguiente pantalla:

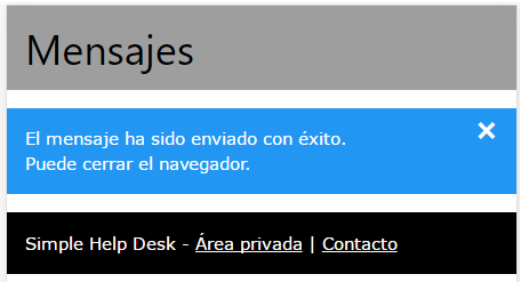

Ilustración 9-2 Mensaje de confirmación de envío

9.1.2.2 Iniciar sesión

Si no se ha iniciado sesión antes el formulario de acceso aparecerá automáticamente cuando intenta acceder a cualquier vista que no sea el formulario de contacto.

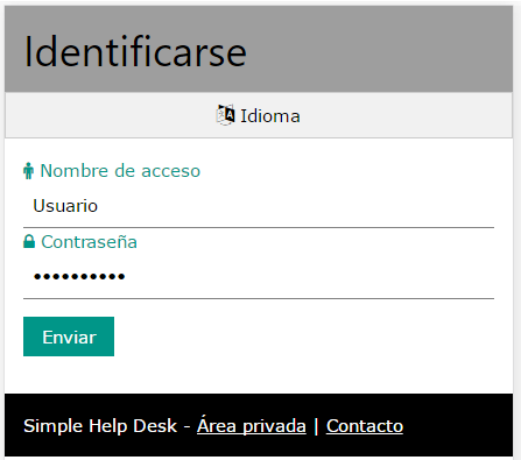

Ilustración 9-3 Formulario de contacto

9.1.2.3 Cambiar el idioma

Se puede cambiar el idioma a uno de los ofrecidos utilizando el desplegable de Idioma del menú.

Ilustración 9-4 Menú de idiomas

9.1.2.4 Usar el menú

El menú, disponible en todas las vistas tiene las siguientes opciones:

Temas: Ir a la lista de los temas

Perfil: Ir al perfil

Administración: Ir al panel de administración; Sólo disponible para administradores.

Desconectarse: Salir de la sesión.

Idioma: Menú de idioma

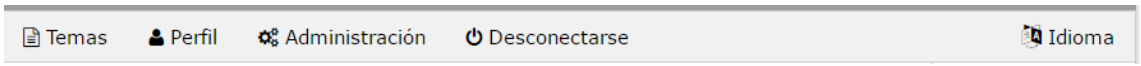

Ilustración 9-5 Menú

9.1.2.5 Listado de temas

La vista de los temas ofrece una lista de todos los temas asignados al usuario actual; Todos para los administradores.

Mostrando temas del 1 al 10 de un total de 201 Mostrar 10 temas Buscar

ID	Asunto	Usuario	Estado	Fecha	Acciones
300	asd	asd - asd@a.c	Abierto	02-jul-2016 18:29:40	Ver detalles
299	John Doe tiene un problema con el número 074	John Doe - mail@domanin.com	Abierto	29-jun-2016 11:46:02	Ver detalles
298	John Doe tiene un problema con el número 04	John Doe - mail@domanin.com	Abierto	29-jun-2016 11:46:02	Ver detalles
297	John Doe tiene un problema con el número 056	John Doe - mail@domanin.com	Abierto	29-jun-2016 11:46:02	Ver detalles
296	John Doe tiene un problema con el número 067	John Doe - mail@domanin.com	Abierto	29-jun-2016 11:46:02	Ver detalles
295	John Doe tiene un problema con el número 045	John Doe - mail@domanin.com	Abierto	29-jun-2016 11:46:02	Ver detalles
294	John Doe tiene un problema con el número 064	John Doe - mail@domanin.com	Abierto	29-jun-2016 11:46:02	Ver detalles
293	John Doe tiene un problema con el número 091	John Doe - mail@domanin.com	Abierto	29-jun-2016 11:46:02	Ver detalles
292	John Doe tiene un problema con el número 044	John Doe - mail@domanin.com	Abierto	29-jun-2016 11:46:02	Ver detalles
291	John Doe tiene un problema con el número 052	John Doe - mail@domanin.com	Abierto	29-jun-2016 11:46:01	Ver detalles

Anterior 1 2 3 4 5 21 Siguiente ...

Ilustración 9-6 Lista de temas

En la vista se puede realizar una búsqueda de los datos o una ordenación. Está disponible la configuración de la paginación, es decir, cuantos registros se muestra a la vez.

9.1.2.6 Vista de perfil

En el perfil se pueden ver los datos personales del usuario así como una imagen y sus temáticas asignadas y el botón de editar en caso de ser administrador o el propio usuario.

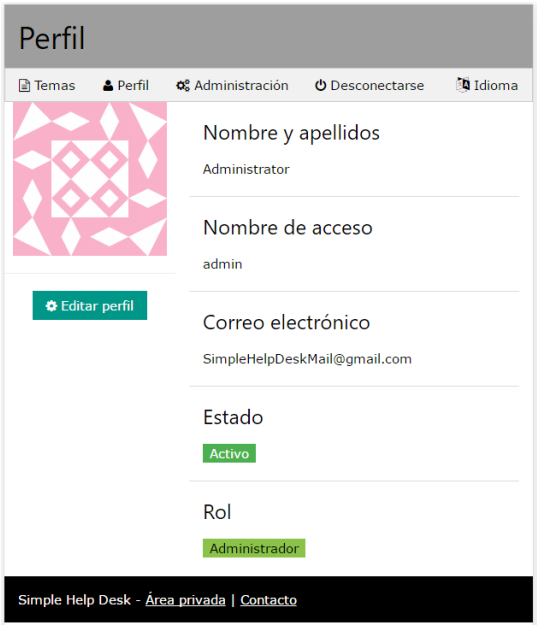

Ilustración 9-7 Vista de perfil

9.1.2.7 Editar perfil

La vista de edición de perfil se divide en distintas partes.

Ilustración 9-8 Edición de perfil

Modificar datos: Modifica los datos del usuario como su nombre, correo y nombre de acceso. El nombre de acceso es único y no se puede repetir entre usuarios.

Cambiar contraseña: Modificar la contraseña actual del usuario.

Temas asignados: Disponible para administradores. Muestra las temáticas asignados actualmente y ofrece la opción de asignar y desasignar temáticas.

9.1.2.8 Vista de administrador

La vista de administrador ofrece dos pestañas: Usuario y Categorías.

En la pestaña de usuarios se muestra una lista de todos los usuarios en el sistema con acceso a sus datos y a su modificación. Se puede desde esta vista además acceder a la creación de nuevos usuarios.

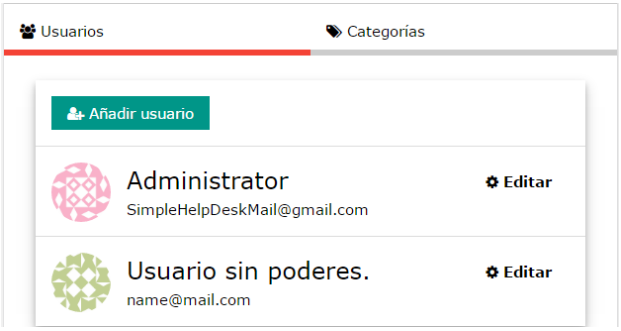

Ilustración 9-9 Administración de usuarios

La creación de nuevos usuarios requiere completar un formulario. El nombre de acceso es único y no se puede repetir.

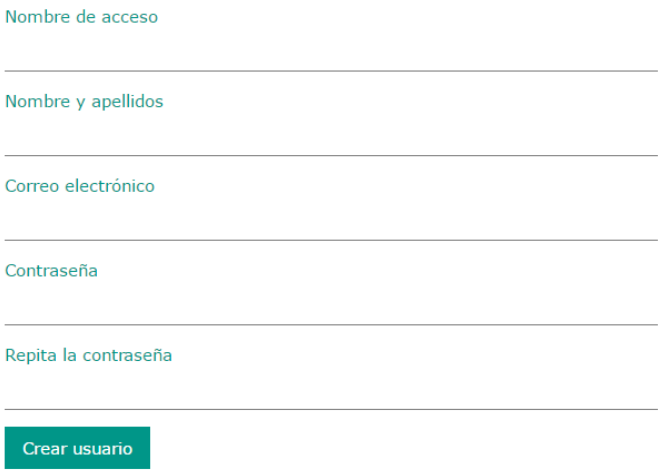

Ilustración 9-10 Formulario de creación de usuarios

La pestaña de categorías ofrece una vista de todas las temáticas en el sistema. Se puede desde esta vista crear nuevas temáticas y eliminar las actuales en caso de que no estén asignadas a ningún usuario o tema.

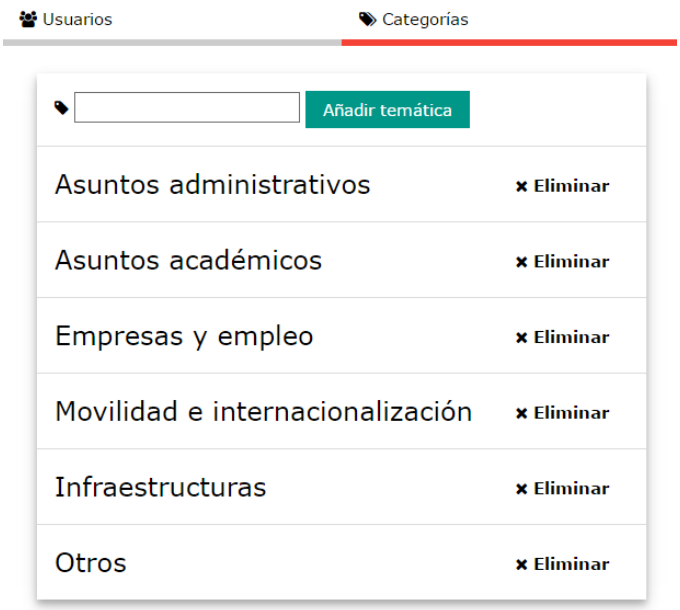

Ilustración 9-11 Administración de categorías o temáticas

9.2 Manual de propagador

9.2.1 Localización del código fuente

El código fuente del proyecto se entrega junto con el producto. Puede importase a cualquier IDE compatible con Maven.

9.2.2 Importar proyecto

Se puede importar el proyecto Maven a cualquier IDE compatible.

Para importar el proyecto desde Eclipse:

> File > Import... > Maven > Existing Maven Project
>
> En el diálogo seleccionar la ubicación del código fuente en Root Directory y en Projects seleccionar el fichero pom.

Para importar en IntelliJ IDEA

> File > Open
>
> En el diálogo seleccionar el fichero pom del proyecto.

9.2.3 Estructura del proyecto

src - Todos los archivos del programa.

> src/main/java - Contiene todos los archivos java.

src/mian/resources - Contiene todos los recursos del programa: ficheros de configuración, plantillas de las vistas html.

target – Los archivos resultantes de la compilación.

WebContent – Carpeta para despliegue web. Ver apartado de empaquetado.

9.2.4 Estructura de los paquetes

app

Contiene la clase principal de la aplicación, el main,.y la administración de los parámetros de entrada.

business

La capa de negocio de la aplicación. Contiene las interfaces de los servicios.

business.exception

Contiene las excepciones específicas de la lógica de negocio.

business.impl

Contiene la implementación de la lógica de negocio.

business.mail

Contiene las clases específicas del sistema de envío de correo.

common

Contiene la factoría principal para independizar cada capa.

config

Contiene la configuración de la aplicación: carga todos los ficheros de propiedades y crea el mapeo de rutas de la aplicación.

config.routes

El mapeo de rutas de la aplicación. Estas clases asignan programáticamente las URIs a las acciones a realizar. Se encargan también de la gestión de los parámetros de las URIs y dirigir a las vistas.

model

Las clases del modelo de la aplicación. Son transversales a toda la aplicación ya que están presentes en todas las capas.

persistence

Capa de persistencia de la aplicación. Contiene las interfaces de la persistencia.

persistence.exception

Contiene las excepciones específicas de la capa de persistencia.

persistence.impl

Contiene la implementación de la capa de persistencia.

presentation

Contiene clases relacionadas con la vista.

util

Contiene clases de utilidad.

9.2.5 Configuración del proyecto

Todas las dependencias, así como la compilación se administran con Maven.

La configuración pro defecto incluye:

Compilación en formato jar

Recursos empaquetados dentro del fichero jar.

Ficheros de propiedades externos al fichero jar en la carpeta conf.

Dependencias almacenadas en carpeta lib externa al fichero jar.

Visite maven.apache.org para aprender más sobre Maven.

9.2.6 Configurar la base de datos

Cambiar la base de datos es uno de los pocos cambios que pueden requerir volver a compilar la aplicación.

Los pasos para cambiar la base de datos son los siguientes:

1. Configurar la dependencia del nuevo controlador de la base de datos en el fichero pom, sustituyendo a la anterior.
2. Cambiar la URL de la conexión a la base de datos en el fichero sql.properties.
3. Comprobar las sentencias SQL del fichero sql.properties asegurando de la compatibilidad del dialecto SQL usado por la base de datos seleccionada.
4. Compilar de nuevo con el comando `maven clean install`.

Consideraciones:

La persistencia se realiza con sql2o. Esta librería ha demostrado compatibilidad con las siguientes bases de datos:

PostgreSQL, MySQL, Oracle, MS Sql Server, IBM DB2, H2 & HyperSQL.

Visite: www.sql2o.org para obtener más información.

Utilizar una base de datos distinta puede requerir modificar todo el sistema de persistencia.

En caso de realizar un cambio total del sistema de persistencia hay que cambiar las siguientes clases/ficheros:

config.DatabaseConfig.Java – Clase que carga la configuración de la base de datos y la conexión.

persistence.* - Paquete con la capa de persistencia.

9.2.7 Empaquetado en formato WAR para desplegar en servidores existentes

Es posible desplegar la a aplicación en otro servidor empaquetándola en formato WAR sin realizar demasiados cambios.

Para empaquetar en formato war:

Configurar Maven para compilar en formato war cambiando a dependencia del pluggin de compilación.

- Hacer que la clase principal (que lanza la aplicación) herede de SparkApplication.y tenga un método init(), que se llamará para iniciarlo.
- Configurar dicha clase como listener en el fichero web.xml.

Este cambio además requiere una revisión completa de ciertos aspectos de la aplicación:

- Localización y/o empaquetado de los ficheros de propiedades.
- Líneas de código especificas del servidor interno como puede ser su configuración.

10 Conclusiones y ampliaciones

10.1 Conclusiones

El objetivo final del proyecto era realizar una aplicación a medida para la gestión de mensajes de contacto para la escuela de ingeniería informática. Esta aplicación debía recibir los mensajes de los visitantes y almacenarlos. Con la llegada de un mensaje uno o varios usuarios en función de su asignación recibirían dicho mensaje por correo. La aplicación permite desde ella contestar al mensaje original. Esta respuesta llega al visitante a partir de la cuenta de correo que había facilitado.

Atendiendo a los objetivos anteriores el proyecto ha alcanzado a cubrir los requisitos principales planteados, y de esta manera, logra facilitar la tarea del manejo de mensajes. Además la aplicación se completa con funcionalidad adicional (de administración).

La comunicación con el tutor del proyecto se realizaba mediante correo electrónico o reuniones en las que se investigaban los requisitos y mostraban prototipos.

El seguimiento del proyecto por parte del equipo de desarrollo (de sólo una persona) se realizaba a partir de una “tabla” de tareas (una plancha de cartón plegable).

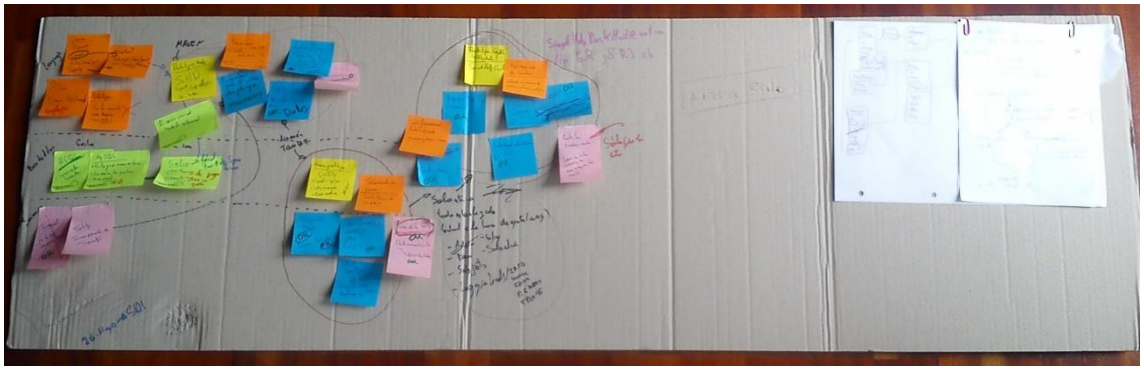

Ilustración 10-1 "Tabla" de tareas

La tabla no era estática pues se modificaba constantemente.

En la parte derecha de la tabla se guardan documentos con requisitos, diagramas, etc.

El resto de la tabla muestra una serie de anotaciones que muestran distinta información:

- Amarillo: muestra el nombre y versión del prototipo. Se rodea esta etiqueta junto con el resto para delimitar los prototipos.
- Naranja: Marcan tecnologías y librerías.
- Verde: Hacen referencia a la base de datos.

- Azul: Requisitos.
- Rosa: Resultado de pruebas sobre el prototipo.

Peronalmente:

He tenido gran libertad para realizar la aplicación y la he aprovechado todo lo que he podido.

He experimentar con las tecnologías, diseños y demás apartados del proyecto. He buscado realizarlo con tecnologías y herramientas modernas e innovadoras. Esto ha hecho que algunas de las elecciones tomadas no hayan sido las óptimas pero tenía como objetivo personal intentar aprender en el proceso.

Pensé en realizarlo en Python para salir de mi zona de confort y forzarme a aprender el uso avanzado del lenguaje. Hubo un intento de realizar la aplicación en Python usando Django como framework. Al final para trabajar en java escogí Spark no sólo porque me resultaba atractivo y novedoso, sino porque era parecido a Django y podía reutilizar todo el diseño que había realizado hasta el momento.

A la hora de trabajar pensé mucho en el diseño de las clases e interfaces. Quería seguir el modelo de capas pero realizando una aproximación más legible y cómoda de usar. Es aquí cuando decido usar elementos como el patrón ActiveRecord como fachada de la persistencia y diseño las interfaces para ser fluidas.

Con interface fluida y ActiveRecord:

```
New Thread()
        .setName(name)
        .setEmail(email)
        .setSubjectId(Integer.parseInt(subject))
        .setTopic(topic)
        .setText(text)
        .setDate(new Date())
        .save();
```

El método save:

```
public Object save() throws PersistenceException {
        return Factory.persistence.getThreadDAO().save(this);
    }
```

Tradicional:

```
Thread thread = new Thread();
thread.setName(name);
thread.setEmail(email);
thread.setSubjectId(Integer.parseInt(subject));
thread.setTopic(topic);
thread.setText(text);
thread.setDate(new Date());

Factory.persistence.getThreadDAO().save(thread);
```

Desde mi punto de vista es más cómodo y legible la primera opción.

Trabajé también en poder externalizar y transformar todos los aspectos posibles en parámetros configurables externos. Es algo que me parecía muy importante porque daba mucha versatilidad a la aplicación. Era una aplicación preparada para trabajar en cooperación con cualquier otra posible.

Aunque se me ocurren mejoras de prácticamente todos los aspectos de la aplicación estoy contento con el resultado de la aplicación y con todo lo que he podido aprender.

10.2 Ampliaciones

A continuación se muestran algunas de las posibles ampliaciones más evidentes que puede recibir la aplicación.

10.2.1 Gestión avanzada de temáticas

Aunque esta ampliación depende del modo de empleo de la aplicación una ampliación sencilla es la ampliación de la gestión de temáticas.

Entre las mejoras existe la posibilidad de internacionalizar el nombre de las categorías. En la entrega no se ha realizado pero ya se diseñó la aplicación pensando en ello. El campo de la base de datos que contiene el texto de la temática se llama "código" y no "nombre" porque guardaría un código a partir del cual se sacase la traducción. Algo así como "tematicaEmpresas" que después fuese "Empresas" en español y...

Para ello hay que guardar en la base de datos una tabla con las traducciones posibles de cada código al idioma.

Al final para mantener la base de datos al margen de su uso no se ha realizado, ya que sería meter internacionalización en el extremo contrario de la aplicación, la persistencia.

Una opción que no requiriese la base de datos sería mantener actualizados los ficheros de propiedades, pero supuestamente ser administrador del sistema (tener cuenta de administrador) no te da acceso a la máquina en la que está la aplicación y los ficheros de propiedades. Además requeriría reiniciar la aplicación para actualizar estos ficheros.

Otra extensión posible es la diferenciación de las temáticas en varios tipos: Internos y externos. Con esta diferenciación se podría tener más temáticas, siendo parte interna.

Así a los visitantes se les ofrecería una serie de temáticas reducida, y una vez aparece una incidencia internamente se tiene un número mayor de temáticas disponibles. Un mensaje que entre con una temática se redirige después a alguna de las temáticas internas relacionadas.

Por ejemplo:

> Temáticas al público: Empresas y empleo,...
>
> Temáticas internas: Ofertas de empleo, Ofertas de colaboración de empresa, Comunicados de empresas,...

10.2.2 Mejora de la interface

Se ha buscado utilizar una interface sencilla, atractiva y fácil de usar. Sin embargo el esfuerzo hecho en la accesibilidad ha sido inferior. Se puede navegar por la página con

el teclado pero la cantidad de tareas de accesibilidad que se pueden realizar sobre una web son muchísimas.

10.2.3 Mejorar el sistema de loggin

Se usa una librería de loggin pero no se aprovecha todo su potencial. Preparar la aplicación para dejar logs por escrito y de mejor calidad es una posible mejora.

11 Referencias

[1] HyperSQL hsqldb.org/

[2] MCC explicado en msdn.microsoft., msdn.microsoft.com/en-us/library/ff649643.aspx

[3] jordisan.net - ¿Qué es un 'framework'? , jordisan.net/blog/2006/que-es-un-framework/

[4] Wikipedia, www.wikipedia.com

[5] stackoverflow – Diferencias sobre tipos de MCV, stackoverflow.com/questions/4801891/difference-between-request-mvc-and-component-mvc

[6] Uml, www.docirs.cl/uml.htm

[7] sql2o, www.sql2o.org

[8] Freemarker, freemarker.org

[9] Spark, sparkjava.com

[10] FreshDesk, freshdesk.com

[11] ssTicket osticket.com

[12] Apache Spark, spark.apache.org

[13] Oracle – API de java, www.oracle.com

[14] jUnit, junit.org/junit4

[15] w3, Etiquetas de idioma www.w3.org/International/articles/language-tags

[16] w3, preguntas sobre la selección de etiquetas de idiomas www.w3.org/International/questions/qa-choosing-language-tags

[17] SLF4J, www.slf4j.org

[18] jBCrypt, www.mindrot.org/projects/jBCrypt

[19] Apache commons email, commons.apache.org/proper/commons-email

[20] Gravatar, gravatar.com

[21] w3css, www.w3schools.com/w3css

[22] FontAwesome, fontawesome.io

[23] jQuerry jquery.com

[24] DataTables, datatables.net

[25] yUML, yuml.me

12 Apéndices

12.1 Contenido entregado

./Leeme.txt	Archivo explicativo
./contenido.pdf	Un mapa de archivos y carpetas
./Base de datos/	Carpeta con la base de datos
./Base de datos/Gestor de bases de datos/	Carpeta con el gestor de base de datos
./Base de datos/Script de creación de tabla.sql	Script de creación de las tablas en SQL
./Código fuente	La carpeta con el código fuente de la aplicación
./Código fuente/Mapa UML global (generado a partir del código).svg	Una imagen con un diagrama UML del proyecto generado con ingeniería inversa
./Código fuente/SimpleHelpDesk/	Proyecto maven. (contiene pom.xml y src)
./Documentación/	Contiene la documentación del trabajo
./Producto/	Contiene el producto final
./Producto/SimpleHelpDesk-1.0.jar	Archivo ejecutable del proyecto compilado
./Producto/lib/	Librerías del proyecto compilado
./Producto/conf/	Archivos de configuración del proyecto terminado

12.2 Script de creación de tablas para la base de datos

```
DROP SCHEMA PUBLIC CASCADE;

--Usuarios
create table users (
      id          INTEGER IDENTITY PRIMARY KEY,
      name  VARCHAR(255) NOT NULL,
      email       VARCHAR(255) NOT NULL,
      admin       BOOLEAN NOT NULL,
      login       VARCHAR(255) NOT NULL UNIQUE,
      psw   VARCHAR(255) NOT NULL,
      active      BOOLEAN DEFAULT true
);

--Categorías
create table subjects(
      id          INTEGER IDENTITY PRIMARY KEY,
      code  VARCHAR(255) UNIQUE NOT NULL
);

--Temas
CREATE TABLE threads(
      id          INTEGER IDENTITY PRIMARY KEY,
      name  VARCHAR(255) NOT NULL,
```

```
      email       VARCHAR(255) NOT NULL,
      subject_id INTEGER NOT NULL,
      date  TIMESTAMP NOT NULL,
      open  BOOLEAN DEFAULT true,
      text  VARCHAR(32768) NOT NULL,
      topic       VARCHAR(255) NOT NULL,
      FOREIGN KEY (subject_id) REFERENCES subjects(id)
);

--Mensajes
CREATE TABLE messages(
      id          INTEGER IDENTITY PRIMARY KEY,
      thread_id   INTEGER NOT NULL,
      text  VARCHAR(32768) NOT NULL,
      date  TIMESTAMP NOT NULL,
      user_id INTEGER NOT NULL,
      FOREIGN KEY (user_id) REFERENCES users(id),
      FOREIGN KEY (thread_id) REFERENCES threads(id)
);

--Asignación categorías
CREATE TABLE assignments(
      user_id INTEGER,
      subject_id INTEGER,
      PRIMARY KEY(user_id, subject_id),
      FOREIGN KEY (user_id) REFERENCES users(id),
      FOREIGN KEY (subject_id) REFERENCES subjects(id)
);
```