
Sistema online de apoyo al programa de Mentorías FDI



TRABAJO FIN DE GRADO GRADO EN INGENIERÍA INFORMÁTICA

Sergio Gil Jiménez
Johanna Aguilar Méndez

Director:
Juan Antonio Recio García

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Junio 2015

Sistema online de apoyo al
programa
de Mentorías FDI

*Memoria que presentan para obtener el título de
grado en Ingeniería Informática*

**Sergio Gil Jiménez
Johanna Aguilar Méndez**

Dirigida por el Doctor

Juan Antonio Recio García

**Departamento de Ingeniería del Software e Inteligencia
Artificial**

**Facultad de Informática
Universidad Complutense de Madrid**

Junio 2015

Autorización

Sergio Gil Jiménez y Johanna Aguilar Méndez, alumnos matriculados en el grado Ingeniería Informática especialidad Tecnología de la Información, autorizan a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, la memoria, el código y la documentación de la aplicación online, todo ello realizado durante el curso académico 2014-2015 bajo la dirección de Juan Antonio Recio García, profesor del Departamento de Ingeniería del Software e Inteligencia Artificial.

Sergio Gil Jiménez

Johanna Aguilar Méndez

*A nuestras familias
y
amigos*

Agradecimientos

*Con mis maestros he aprendido mucho;
con mis colegas, más; con mis alumnos
todavía más.*

Proverbio hindú.

Una vez concluido el presente trabajo de fin de grado, queremos expresar nuestros sinceros agradecimientos a todas aquellas personas que han vivido este proyecto con nosotros, directa o indirectamente han contribuido con sus opiniones, conocimientos y han estado presentes en todo momento, brindándonos una palabra de aliento para seguir adelante.

En primer lugar, a nuestro director de proyecto, Dr. Juan Antonio Recio García, nuestro imperecedero agradecimiento por confiarnos la realización de este importante proyecto en beneficio de nuestra universidad, por su paciencia y dedicación, por compartir sus vastos conocimientos durante toda nuestra formación académica.

A nuestros compañeros por su apoyo incondicional, con ellos hemos compartido aulas, laboratorios, despachos e incontables horas de trabajo desde el primer día que nos propusimos esta gran meta, hemos acumulado experiencias, recuerdos en cada lugar de la universidad y sobre todo la mejor compañía os podemos asegurar. Por las conversaciones científicas a la hora de la comida, las reuniones en grupo para estudiar exámenes y divertirnos después de los mismos, más que compañeros nos demostraron el verdadero valor de la amistad. Gracias por todos los momentos compartidos que se quedarán grabados en lo más profundo de nuestro corazón.

Finalmente a nuestras familias, el motor de nuestras vidas, aquellos que nos brindan una sonrisa al llegar a casa después de un largo día de trabajo, quienes nos dan ánimos después de un difícil examen, quienes han creído en nosotros y están incondicionalmente a pesar de las dificultades, este esfuerzo es dedicado a ustedes. Gracias por todo el apoyo, sin vosotros no habiéramos logrado concluir el proyecto.

Resumen

Me lo contaron y lo olvidé; lo vi y lo entendí; lo hice y lo aprendí.

Confucio

La universidad es una experiencia maravillosa y aunque se llega con entusiasmo y ganas de enfrentarse a todo, recibir ayuda de personas que ya conocen el proceso siempre viene bien, en España la mayoría de centros universitarios ofrecen programas de apoyo para que los alumnos que asisten por primera vez a la universidad tengan una mejor integración académica y social en la vida universitaria. En el caso de la Universidad Complutense de Madrid el programa de mentorías online se creó con el objetivo fundamental de brindar asesoramiento a los estudiantes que ingresan por primera vez a la universidad, el programa se realiza entre compañeros de la misma facultad, siendo los mentores preparados adecuadamente para desempeñar su rol.

El presente proyecto fue desarrollado con la idea de mejorar la web del programa de mentorías actual creando una sistema online interactivo de fácil uso donde los usuarios puedan acceder desde sus dispositivos móviles o cualquier otro dispositivo. Como alumnos que fuimos en algún momento contar con una aplicación que brinde este tipo de información y que pueda ser accedida desde el móvil, facilitará la vida académica de cualquier estudiante universitario.

De igual manera, el presente proyecto describe a detalles los roles, beneficios y oportunidades que brinda la aplicación al programa de mentorías, identificando cuatro públicos objetivos que harán uso del programa: el coordinador central o administrador total del programa de mentorías, es el encargado de gestionar todos los permisos para realizar cambios dentro de la aplicación; el coordinador de la facultad, existiendo uno por cada facultad que participa y es el encargado de administrar las reuniones celebradas y llevar la gestión de sus mentores y telemácos; los mentores son alumnos ya experimentados en la vida universitaria, especialmente de los últimos cursos, quienes son los responsables de guiar a los telémacos, los alumnos nuevos

que participan en el programa, quienes su función dentro de la aplicación es evaluar a sus mentores.

Todos los usuarios que deseen hacer uso del programa de mentorías deberán loguearse con el correo de la universidad, pero con clave independiente y tendrán diferente interfaz en la aplicación según el rol asignado.

Palabras clave: programa de mentorías, coordinador central, coordinador facultad, mentores, telémacos, usuario.

Summary

The university is a great experience where getting help is always welcome though you did want to overcome everything. In Spain, most of the colleges offer support programs for new students to get them integrated into university life. In Complutense University of Madrid, mentoring program was designed for providing advice to students entering first college. The program is between college partners, previously identifying their guidance and education skills, being the ideal people to develop the ideal profile for the role of mentor as his specialty.

At first the project was developed to improve the current mentoring program of the Faculty of Computer Engineering, however the've planned to apply it to other university faculties due to the need to broaden it to the entire university community, from then on It was displayed at the Faculty of Informatics and to the entire university maybe this year too. The application is an online platform where you can easily interact with users, the same who can access from their mobile device or any other electronic device through an account created from the program, where they can find support for their adaptation to the system university.

Rely on this kind of application brings you this information and it can be accessed from your mobile will make your educational life easier. The program is not only thinking of the students but also those who manage your progress, that's why the application has administrative users who are responsible for the application performance.

Similarly, this project details the roles, benefits and opportunities from mentoring program implementation, identifying four target groups that will use the program: the central coordinator or full mentoring program manager, who is responsible for manage all permissions to make changes within the application; the coordinator of the faculty, one for each involved college and responsible for managing the meetings and lead the management of

their mentors and telematic students; mentors are experienced students in university life, especially in their final years, who are responsible for guiding telematic students, the new students that take part in the program, whose function within the application is to evaluate their mentors.

All users who wish to use the application for the mentoring program must log on with the university mail, but with a separate key and have different interface depending on their role.

Keywords: *mentoring program, central coordinator, coordinator of the faculty, mentors, telematic students, user.*

Índice

Autorización	V
Agradecimientos	IX
Resumen	XI
1. Introducción	1
1.1. Motivación	2
1.2. Objetivo General	3
1.3. Estructura de la memoria	3
2. Descripción del programa de mentorías	5
2.1. Inicio del programa de mentorías	5
2.2. Actualidad del programa de mentorías	6
2.3. Investigación web	6
3. Diseño de la aplicación	9
3.1. Usuarios	9
3.2. Requisitos de la interfaz de usuario.	11
3.2.1. Restricciones técnicas	11
3.2.2. Perfiles de usuario	11
3.3. Casos de usos	12
3.3.1. Casos de uso comunes para los actores	12
3.3.2. Casos de uso autor: Coordinador central	13
3.3.3. Casos de uso autor: Coordinador de la facultad	19
3.3.4. Casos de uso autor: Mentor	30
3.3.5. Casos de uso autor: Telémaco	32
4. Implementación	33
4.1. Tecnologías utilizadas	33
4.1.1. Java	33
4.1.2. Apache Struts 2	34

4.1.3.	JSP	34
4.1.4.	XML	34
4.1.5.	MySql	34
4.1.6.	Bootstrap	35
4.1.7.	HTML	35
4.1.8.	JavaScript	35
4.2.	Estructura de la base de datos	36
4.2.1.	Modelo entidad-relación	36
4.2.2.	Explicación teórica del modelo entidad-relación	36
4.3.	Front-end	40
4.3.1.	Mockups de la aplicación	41
4.3.2.	Prototipo inicial	42
4.3.3.	Prototipo final	44
4.3.4.	Validación de formularios	47
4.4.	Back-end	47
4.4.1.	Estructura interna: Clases Principales y paquetes	48
4.4.2.	Seguridad en la aplicación	57
5.	Conclusiones y trabajo futuro	59
5.1.	Recomendaciones	60
5.2.	Descripción del trabajo realizado por los alumnos	61
5.3.	Trabajo futuro	64
	Bibliografía	65

Índice de figuras

2.1. Página actual mentorías UCM	7
2.2. Aplicación de gestión del programa de mentorías.	7
3.1. Casos de uso de coordinador central	19
3.2. Casos de uso de coordinador de la facultad	29
3.3. Casos de uso de mentor	31
3.4. Casos de uso de telémaco	32
4.1. Diagrama del modelo entidad-relación	36
4.2. Mockups de pantalla de ordenador - Portada	41
4.3. Mockups de pantalla de tablet - Encuesta	41
4.4. Mockups de pantalla de móvil - Portada	42
4.5. Prototipo inicial - Portada	43
4.6. Prototipo inicial - Inicio Sesión	43
4.7. Prototipo inicial - Información	43
4.8. Prototipo inicial - Contacto	44
4.9. Prototipo inicial - Coordinador facultad	44
4.10. Prototipo final - Portada	45
4.11. Prototipo final - Información	45
4.12. Prototipo final - Información detallada	45
4.13. Prototipo final - Coordinador central	46
4.14. Prototipo final - Coordinador facultad	46
4.15. Prototipo final - Mentor	46
4.16. Prototipo final - Telémaco	47
4.17. Prototipo final - Validación de formulario	47
4.18. Vista estructura de las clases en paquetes	48
4.19. Vista de las clases que inician la aplicación	49
4.20. Vista del código de la clase ContextListener	49
4.21. Vista del código de la clase MainController	50
4.22. Vista del código de la clase SessionInterceptor	50
4.23. Vista de las clases de seguridad	51

4.24. Vista de las clases modelo	51
4.25. Vista de las clases Facultad	52
4.26. Vista de las clases útiles de la aplicación	52
4.27. Vista de las clases de las acciones de los usuarios	53
4.28. Vista del código del método ConfigurationManager	53
4.29. Vista del código de la clase DBHandler	54
4.30. Vista del método de la configuración inicial de Email	54
4.31. Vista del método que envía un correo al usuario	55
4.32. Vista del método que envía correo a multiples usuarios	56
4.33. Vista del código del método getDumpAllTablesSql	57
5.1. Algunas instituciones participantes en la Red Iberoamericana de mentoría	64

Capítulo 1

Introducción

*La disciplina es la parte más importante
del éxito*

Truman Capote

El sistema online de apoyo al programa de mentorías para la Facultad de Informática de la Universidad Complutense de Madrid, es una aplicación web que cubre las necesidades de los usuarios que participan en este programa. El programa de mentorías en nuestra universidad fue desarrollado por los vicerrectorados para que los estudiantes nuevos tengan una mejor integración en la vida universitaria.

Todos en algún momento, hemos vivido la experiencia de ser la nueva o el nuevo en la universidad, al comienzo no conoces a nadie, no sabes donde queda tu aula, ni las asignaturas que vas a recibir, en resumidas palabras desconocen el mundo al cual te vas a enfrentar. Las mentorías ayudan a que los nuevos alumnos aprovechen los recursos formativos que la universidad les ofrece para desarrollar al máximo sus aptitudes personales.

La característica principal de la aplicación es que no solo la pueden utilizar los alumnos nuevos, sino también el coordinador de la facultad, con su espacio personal y una vez revisados los respectivos informes puede gestionar los datos de los usuarios a su cargo y dar seguimiento a cada uno de ellos para brindarles un mejor asesoramiento por parte de su mentor asignado.

La aplicación creada cubre las necesidades principales para una gestión efectiva del programa, aunque inicialmente fue pensada para uso de la Facultad de Informática, ésta podrá ser utilizada para gestionar todo el programa de mentorías de la universidad, teniendo un administrador total y un solo administrador por facultad.

Introduction

The online support system mentoring program for the Faculty of Informatics at the Complutense University of Madrid, is a web app that covers the needs of users who are part of the program. The mentoring program at our university was developed by the vice rectorates for new students have a better integration in university life.

At some point, every one of us have lived the experience of being new at the university. At first, you don't know anyone, you don't know where your class is or the subjects you are going to study. To sum up, you don't know the world you will have to deal with. Mentoring helps to these new students to take advantage of the educational resources the university offers them. This project give them the chance to develop their personal skills.

The app main feature not only can be used by new students, but also the college's coordinator. The coordinator could review and manage the users' data and follow every one of them to give them a better advice from his mentor.

The app covers the main needs for an efficient management of the program, even though it was initially thought for its use in the faculty of the computer engineering, this app could be used to manage all the mentoring program of the university.

1.1. Motivación

¿Por qué eligieron este trabajo fin de grado?, ¿Qué os motivó a elegirlo?. Estas fueron algunas de las interrogantes que nos plantearon durante la realización del proyecto, he aquí nuestra contestación:

La principal razón por la que elegimos el presente proyecto fue para contribuir al desarrollo de un programa que aporte beneficios a la universidad, de la cual durante 5 años hemos sido parte y nos sentimos orgullosos. Cada año son más los alumnos que utilizan el programa de y esa fue nuestra principal motivación, saber que con nuestro proyecto podemos ayudar a un gran número de estudiantes que por primera vez forman parte de esta grandiosa aventura, llamada vida universitaria, donde tanto mentores como coordinadores ayudarán a cumplir esta labor.

Por otro lado, se consideró la funcionalidad como principal característica de la aplicación, pues es necesario ofrecer una aplicación a los usuarios que no presente complicaciones al usarla, ya sea para realizar búsquedas de información o para la gestión de los datos informativos, este tipo de nece-

sidades a satisfacer, motivaron la seguridad para desarrollar este proyecto. Otro aspecto que nos motivó fue también la aprobación de la universidad para la puesta en práctica de la aplicación, misma que tendrá perennidad al ser utilizada en los años venideros.

Convertirnos en los creadores de un programa de esta naturaleza será el comienzo de nuestra carrera profesional y una gratificante experiencia de la labor cumplida en beneficio de la universidad.

1.2. Objetivo General

Ofrecer a los usuarios del programa de herramientas necesarias para acomodarse a la vida universitaria de una manera fácil a través de la creación de una aplicación web con un diseño *responsive*¹ que pueda ser utilizada desde cualquier dispositivo sea móvil, tablet y/o ordenador y que permita a los usuarios tener la información siempre disponible.

Objetivos Específicos:

- Estudiar la web actual del programa de mentorías para generar un prototipo inicial.
- Identificar las acciones que los usuarios desempeñan asignado un rol.
- Crear una base de datos aceptable a cambios para que a futuro, si existen modificaciones en la misma la aplicación no tenga cambios mayores.
- Crear un diseño web sencillo y de fácil uso para los usuarios, basado en los casos de uso identificados.
- Implementar las funcionalidades del diseño para crear una aplicación adaptable a múltiples dispositivos.
- Encriptar las contraseñas de los usuarios para brindar mayor seguridad en la aplicación.

1.3. Estructura de la memoria

En esta sección se da a conocer la jerarquía de la memoria, aquí os damos una descripción breve de nuestros primeros pasos para lograr lo que ahora es la web actual.

¹Diseño adaptable

- **Capítulo 1: Introducción.**

En este capítulo se describen cuáles fueron las principales motivaciones que llevaron a elegir este proyecto y los objetivos planteados para su desarrollo, además se realiza una breve descripción de la estructura de la memoria.

- **Capítulo 2: Descripción del Programa de Mentorías.**

Aquí se describen los inicios del programa de mentorías y su avances hasta la actualidad, también se presenta la investigación web realizada para estructurar el diseño de la aplicación.

- **Capítulo 3: Diseño de la aplicación.**

Se presenta la ideas claves para el diseño, es decir, el sitio fue pensado para que cada usuario tenga su propio espacio con un menú de acuerdo al rol que desempeña. El objetivo fue crear una aplicación sencilla y que pueda presentar diferentes interfaces, en esta sección se detalla las acciones a las cuales puede acceder cada usuario de acuerdo al rol.

En resumen se presentan los cambios que la aplicación fue tomando hasta llegar a lo que es actualmente.

- **Capítulo 4: Implementación.**

En este capítulo se describe cómo hemos implementado el código de la aplicación y la interfaz del , la estructura que tiene el programa y el por qué fue desarrollado de esa forma, además se indican los lenguajes utilizados durante toda la transformación del sitio y los distintos cambios realizados para llegar al proyecto final.

- **Conclusiones.**

Se presentan las conclusiones personales y la experiencia de los creadores del proyecto, el aprendizaje adquirido para su aproximación al mundo laboral, además se describe el trabajo futuro que se va a realizar para la Red Iberoamericana de Mentorías².

- **Bibliografía.**

Finalmente se presentan las referencias tomadas a lo largo de la redacción de esta memoria.

²<http://mentoriasiberoamerica.org/>

Capítulo 2

Descripción del programa de mentorías

*Solos podemos hacer muy poco;
unidos podemos hacer mucho.*

Hellen Keller

En este capítulo se realiza una descripción completa del programa de mentorías, cuales fueron sus inicios y cómo se llegó a las consideraciones generales puestas en práctica en la aplicación actual. Al ser una aplicación creada recientemente y en etapa de construcción, el sitio ha tenido una acogida asombrosa por parte del alumnado de la facultad de Informática.

En tal razón, se expondrá el objetivo principal por el cual se creó el programa para los estudiantes y desde cuando está en funcionamiento, así como también, los logros alcanzados hasta la actualidad, tomando en consideración que para crear una buena aplicación es importante realizar una investigación previa del funcionamiento del programa actual y así proponer una aplicación sustentada en corregir los errores de la anterior, aspectos que se detallan en la sección 2.3.

2.1. Inicio del programa de mentorías

El programa de mentoría de la Universidad Complutense de Madrid inició en el curso 2012 - 2013 con el apoyo de los Vicerrectorados de Grado, Estudiantes y Evaluación de la Calidad cuyo objetivo es facilitar la integración de los nuevos estudiantes.

Los mentores que reciben una formación específica para desempeñar eficazmente su rol, realizan una media de cinco reuniones con su pequeño grupo

de estudiantes, desde septiembre a marzo, con una duración aproximada de 50 minutos. El nivel de satisfacción con el programa y la utilidad percibida del mismo (de mentores y telémacos).

Con esta iniciativa la universidad hace un esfuerzo para mantener y mejorar las políticas de acogida y acompañamiento a los nuevos estudiantes, involucrándose no sólo el rectorado sino también la mayor parte de las facultades, con la participación de profesores y alumnos. La mentoría se ha convertido en un proceso asentado en la universidad que ayuda a incrementar la calidad de sus servicios y desarrolla el potencial de sus miembros.

Tomado como referencia de la página del programa de mentorías UCM ¹

2.2. Actualidad del programa de mentorías

Actualmente, en el programa de mentorías participan el 80 %² de las facultades de la universidad, donde el coordinador de cada facultad es el encargado de la gestión de las reuniones de sus mentores y telémacos que se celebran durante todo el periodo lectivo.

Este programa desarrollado por los vicerrectorados cada vez se hace más popular entre los alumnos de nuevo ingreso, se espera que la nueva aplicación facilite a los usuarios nuevos su paso por la vida universitaria y a los usuarios coordinadores una mejor administración en los datos.

2.3. Investigación web

El front-end de la aplicación no se basa en otros programas de mentorías que existan en otras universidades, para comenzar el diseño se evalúa la página de mentorías que existe actualmente en la universidad, teniendo como resultado las siguientes observaciones:

- La página no brinda un espacio personalizado para los usuarios.
- Al ser un diseño tradicional y no funcional los usuarios la consideran aburrida a pesar del buen contenido disponible.
- La estructura de la página confunde a los usuarios, teniendo que repetir acciones en algunos casos.

¹<http://www.ucm.es/mentorias>

²Según la página de mentorías UCM - <http://www.ucm.es/mentorias>

Capítulo 3

Diseño de la aplicación

*Existe algo más importante que la lógica:
la imaginación.*

Anónimo

En este capítulo se expondrán los requisitos fundamentales para la construcción de la aplicación, describiendo a detalle las acciones que realiza cada usuario. Además, es en esta etapa donde se cuidan todos los detalles de diagramación de la aplicación para evitar futuros problemas.

Conocer a los usuarios es la base principal para el desarrollo del programa, ya que son ellos quienes van a utilizar la página. Para ello se ha identificado las necesidades de los usuarios, qué roles van a desempeñar y las funciones que desarrollan dentro de la aplicación, identificando qué usuarios van a tener permisos administrativos y cuáles no, con ello se brinda seguridad a la aplicación.

3.1. Usuarios

Existen cuatro usuarios bien identificados, los cuales han sido obtenidos de los roles existentes en la página actual del programa de mentorías, por tanto la página contará con: un administrador, el cual se denomina coordinador central, para cada facultad deberá haber un administrador de sus reuniones y usuarios, a estos se lo conoce como coordinador de la facultad, luego están los estudiantes preparados para ayudar a los nuevos alumnos, llamados mentores y finalmente los nuevos estudiantes que son los telémacos.

A continuación detallamos las funciones así como las limitantes para cada uno de los usuarios:

- **Coordinador central:**

1. **Gestionar todos los usuarios:** Donde puede crear, modificar, listar y eliminar los usuarios de tipo coordinador facultad, mentor o telémaco. Además puede exportar la tabla de usuarios de la base de datos.
2. **Configuración de los cursos:** Puede crear un nuevo curso, cambiar a otro curso que estuviera creado, así como también crear una copia de seguridad de un curso completo.
3. **Gestionar las carreras:** Puede crear, modificar, listar y eliminar una carrera para una facultad.
4. **Informes:** Puede obtener informes de las valoraciones que hacen los telémacos a sus mentores.

- **Coordinador de la facultad:**

1. **Gestionar las reuniones:** Puede crear, modificar, listar y eliminar las reuniones para mentores y telémacos.
2. **Gestionar los mentores:** Puede crear, modificar, listar y eliminar los mentores de su facultad. También puede listar, rechazar o aceptar una petición de partición de un futuro mentor. Además puede enviar correos a todos sus mentores.
3. **Gestionar los telémacos:** Puede crear, modificar, listar y eliminar los telémacos de su facultad. También puede listar, rechazar o aceptar una petición de partición de un futuro telémaco. Además puede enviar correos informativos a todos sus telémacos.
4. **Gestionar las carreras:** Puede crear, listar, modificar y eliminar una carrera de su facultad.
5. **Informes:** Puede obtener los informes de las valoraciones que hacen los telémacos a los mentores después de una reunión de su facultad.
6. **Encuesta:** Poder realizar una valoración final del programa de mentorías, pero esta funcionalidad no esta implementada en la aplicación, por ello se enlaza a las encuestas finales que existen en la pagina actual del programa de mentorías.

- **Mentor:**

1. **Listar telémacos:** Modificar y eliminar los telémacos a su cargo.
2. **Encuesta:** Rellenar las encuestas de seguimiento.
3. **Informes:** Poder acceder a los informes de las valoraciones que le dan sus telémacos.

- **Telémacos:**

1. **Encuesta:** Puede rellenar la encuesta de seguimiento de las reuniones para valorar a su mentor.

3.2. Requisitos de la interfaz de usuario.

3.2.1. Restricciones técnicas

Tomando en cuenta que la aplicación web va a ser utilizada por todo tipo de usuarios utilizando conexiones de velocidad variable, se han optimizado el apartado gráfico y la información de ida y vuelta al servidor para que no exista carga lenta de las imágenes ni de la información.

Además, hoy en día los usuarios acceden a las aplicaciones utilizando dispositivos móviles, por tanto se ha optado por un a las pantallas de estos dispositivos para que no tengan complicaciones de acceso o visualización.

3.2.2. Perfiles de usuario

Es necesario hacer una distinción entre los cuatro tipos de usuarios que usarán la aplicación:

- **Coordinador central:** Es la cuenta del administrador de la aplicación, solo existe un administrador total y su cuenta está creada directamente en la base de datos. Además cuando se crea un curso nuevo, es decir, una nueva base de datos, la cuenta del administrador siempre se mantiene aunque las demás tablas se creen vacías.
- **Coordinador de la facultad:** Es un usuario con permisos administrativos, pero con algunas restricciones, se encarga de la gestión de las mentorías en su facultad y solo existe un coordinador por facultad.
- **Mentor/Telémaco:** Son los usuarios mayoritarios, estudiantes de la universidad. Los mentores tienen permisos para gestionar solo a sus telémacos y los telémacos son usuarios pasivos que su función principal es evaluar a sus mentores a través de encuestas.

3.3. Casos de usos

3.3.1. Casos de uso comunes para los actores

Caso de uso: Login.

- **Descripción:** El usuario quiere loguearse en la aplicación para acceder a todas sus funcionalidades a través de su cuenta de usuario.
- **Flujo básico:**
 1. El usuario selecciona la opción "Iniciar sesión".
 2. La aplicación lo lleva a la sección de inicio de sesión donde se solicitan los parámetros de autenticación (correo/contraseña).
 3. El usuario rellena los parámetros solicitados.
 4. Si los parámetros son rellenos correctamente le permitirá al usuario dar al botón "Iniciar Sesión".
 5. Si la validación de los datos del usuario es aceptada, la aplicación lo lleva a su cuenta.

Caso de uso: Cerrar Sesión.

- **Descripción:** El usuario quiere desconectarse de su sesión.
- **Flujo básico:**
 1. El usuario selecciona la opción del menú de "Salir".
 2. La aplicación lo dirige a la página principal.

Caso de uso: Perfil.

- **Descripción:** El usuario quiere entrar a su perfil donde encuentra su datos personales, y puede modificar su nombre y cambiar la contraseña.
- **Flujo básico:**
 1. El usuario selecciona la opción del menú de "Perfil".
 2. La aplicación le muestra una ventana en el costado izquierdo de la aplicación, con sus datos.

3.3.2. Casos de uso autor: Coordinador central

Caso de uso: Gestión de usuarios.

- **Descripción:** El usuario puede crear distintos tipos de usuario (coordinador facultad, mentor y telémaco) donde tiene que rellenar para cada usuario los campos respecto a su rol, ver listado de usuarios donde puede modificar los datos los usuarios o eliminar su cuenta de usuario, ver el listado de usuarios interesados donde podrá aceptar el usuario o rechazarlo y por último exporta la base de datos de todos los usuarios actuales que usan la aplicación.

Casos de uso especializados de gestión de usuarios:

Alta de usuario

- **Descripción:** El usuario quiere dar de alta a los usuarios que administra, para ello tiene que rellenar los campos necesarios para cada tipo de usuario.
- **Flujo básico:**
 1. El usuario selecciona la opción "Alta usuario".
 2. La aplicación lleva al usuario a la siguiente página donde pide que seleccione el tipo de usuario que quiere dar de alta.
 3. El usuario rellena los campos pedidos.
 4. La aplicación valida los campos rellenos, si falta un campo obligatorio no permite registrar el alta, si esta todo correcto permite dejar al usuario seguir.
 5. Aceptada el alta, la aplicación muestra un mensaje de confirmación del alta o en caso contrario le indica que existe un error.

Listado de usuarios

- **Descripción:** El usuario puede ver la lista completa de todos los usuarios que usan actualmente la aplicación, para cada usuario se puede modificar sus datos o eliminarlo. Se puede hacer una exportación de la lista completa de los datos en formato csv.
- **Flujo básico:**
 1. El usuario selecciona la opción "Listado de usuarios".
 2. La aplicación muestra una tabla de todos los usuarios con la opciones de correo, nombre, rol y la acciones de editar y eliminar, la tabla se puede filtrar por cualquier dato buscado.

3. Listado los datos el usuario tiene la opción de editar o eliminar los datos de cada usuario mostrado y además se puede exportar en formato csv todo el contenido de la tabla o de los datos filtrados.

Según la petición que haga el usuario se realizan distintas acciones que explicamos a continuación:

Editar: El usuario al seleccionar la opción de editar un usuario de la lista, la aplicación le muestra una ventana emergente donde puede editar los campos mostrados, una vez editados los datos y darle a la opción actualizar, la aplicación cierra la ventana emergente y muestra un mensaje de confirmación si todo esta correctamente o un mensaje de error si fallo y en la tabla se muestran los campos actualizados.

Eliminar: El usuario selecciona la opción de eliminar un usuario de la lista, la aplicación le muestra una ventana emergente donde le pide la confirmación de la eliminación, si el usuario confirma la eliminación, la aplicación cierra la ventana emergente y le muestra un mensaje de confirmación que ha sido eliminado o un mensaje de error si fallo y en la tabla el usuario eliminado ya no aparece.

Exportar en csv: El usuario selecciona la opción de exportar en csv los usuarios mostrados en la tabla y la aplicación genera la descarga automáticamente.

Listado de usuarios interesados

- **Descripción:** El usuario puede ver la lista completa de todos los posibles usuarios para la aplicación, para cada usuario se puede aceptar la petición de participación o rechazarla.
- **Flujo básico:**
 1. El usuario selecciona la opción "Listado de usuarios interesados".
 2. La aplicación muestra una tabla de todos los usuarios que han enviado una petición para participar en el programa, las opciones mostradas en la tabla son correo, nombre, rol, comentario, facultad y las acciones de confirmación y rechazo. La tabla se puede filtrar por cualquier dato buscado.
 3. Listado los datos el usuario tiene la opción de rechazar o confirmar la petición del usuario.

Según la petición que haga el usuario se realizan distintas acciones que explicamos a continuación:

Confirmación: El usuario al seleccionar la opción de confirmación de petición de usuario de la lista, la aplicación le muestra una ventana emergente donde pide una confirmación de alta del usuario y al darle a la opción confirmación, la aplicación cierra la ventana emergente y muestra un mensaje de confirmación si todo esta correctamente o un mensaje de error si fallo y en tabla mostrada la petición de usuario se elimina.

Rechazar: El usuario selecciona la opción de rechazar la petición de usuario de la lista, la aplicación le muestra una ventana emergente donde le pide la confirmación de la eliminación, si el usuario confirma la eliminación, la aplicación cierra la ventana emergente y le muestra un mensaje de confirmación que ha sido rechazado o un mensaje de error si fallo y en la tabla la petición de usuario se elimina.

Exportar: El usuario selecciona la opción de exportar en csv las peticiones de los posibles usuarios mostrados en la tabla y la aplicación genera la descarga automáticamente.

Exportar usuarios

- **Descripción:**El usuario puede exportar la tabla completa de usuarios de la base de datos de la aplicación con su contenido y se genera un archivo formato zip con las consultas sql.
- **Flujo básico:**
 1. El usuario selecciona la opción "Exportar usuarios".
 2. La aplicación lleva al usuario a la siguiente página donde pide que confirme si realmente quiere hacer una copia de seguridad de la tabla usuarios de la base de datos.
 3. El usuario al confirmar la copia de seguridad y seleccionar exportar.
 4. La aplicación genera la descarga del archivo.

Caso de uso: Configuración de cursos

- **Descripción:** El usuario puede crear un nuevo curso, para ello se puede borrar los datos actuales de la base de datos o mantenerlos y puede

seleccionar de un listado la base de datos a crear.

Casos de uso especializados de configuración de curso:

Curso nuevo

- **Descripción** El usuario puede crear un curso nuevo para un año lectivo, donde se le informa que los datos de las tablas de la base de datos se reinician, pero también puede mantener su información si lo desea.
- **Flujo básico:**
 1. El usuario selecciona la opción "Curso nuevo".
 2. La aplicación lleva al usuario a la siguiente página donde informa con un mensaje de alerta que los datos de la base de datos pueden reiniciarse y pide al usuario el nombre de la base de datos y que seleccione si desea conservar los datos actuales en la nueva base de datos.
 3. El usuario rellena los campos pedidos.
 4. Si la aplicación confirma que todos los campos están rellenos de forma correcta le permite al usuario crear el nuevo curso y se muestra un mensaje de confirmación si se realizó todo correctamente.

Listado de cursos

- **Descripción** El usuario puede seleccionar la base de datos que desee para utilizar la aplicación
- **Flujo básico:**
 1. El usuario selecciona la opción "Listado de cursos".
 2. La aplicación lleva al usuario a la siguiente página donde informa con un mensaje de alerta que si selecciona otra base de datos se cambiarán los datos y pide al usuario que elija la base de de datos quiere utilizar.
 3. El usuario selecciona el campo pedido.
 4. Si la aplicación confirma que el usuario seleccionó el campo pedido le permite al usuario cambiar la base de datos y le muestra un mensaje de confirmación si todo salió correctamente.

Casos de uso: Gestión de Carreras

- **Descripción** El usuario crear una nueva carrera para las facultades de la base de datos, puede ver la lista completa de todas las carreras de las facultades para cada carrera se puede modificar sus datos o eliminarla. Se puede hacer una exportación de la lista completa de los datos en formato csv.

Casos de uso especializados de gestión de carreras:

Nueva carrera

- **Descripción:** El usuario puede crear una nueva carrera para una facultad registrada en la base de datos.
- **Flujo básico:**
 1. El usuario selecciona la opción "Nueva carrera".
 2. La aplicación lleva al usuario a la siguiente página donde pide que seleccione una facultad y el nombre de la nueva carrera.
 3. El usuario rellena los campos pedidos.
 4. Si los campos son rellenados correctamente, permite al usuario crear la carrera y si se realiza la acción aparece un mensaje de confirmación de la nueva carrera creada.

Listado Carreras

- **Descripción** El usuario puede ver la lista completa de todas las carreras de las facultades que usan actualmente la aplicación, para cada carrera se puede modificar sus datos o eliminarla. Se puede hacer una exportación de la lista completa de los datos en formato csv.
- **Flujo básico:**
 1. El usuario selecciona la opción "Listado de carreras".
 2. La aplicación muestra una tabla de todas las carreras con la opciones de nombre, facultad y la acciones de editar y eliminar. La tabla se puede filtrar por cualquier dato buscado.
 3. Listado los datos el usuario tiene la opción de editar o eliminar los datos de cada carrera y además se puede exportar en formato csv todo el contenido de la tabla o de los datos filtrados.

Según la petición que haga el usuario se realizan distintas acciones que explicamos a continuación:

Editar: El usuario al seleccionar la opción de editar un carrera de la lista, la aplicación le muestra una ventana emergente donde puede editar los campos mostrados, una vez editados los datos y darle a la opción actualizar, la aplicación cierra la ventana emergente y muestra un mensaje de confirmación si todo esta correctamente o un mensaje de error si fallo y en la tabla se muestran los campos actualizados.

Eliminar: El usuario selecciona la opción de eliminar una carrera de la lista, la aplicación le muestra una ventana emergente donde le pide la confirmación de la eliminación, si el usuario confirma la eliminación, la aplicación cierra la ventana emergente y le muestra un mensaje de confirmación que ha sido eliminado o un mensaje de error si fallo y en la tabla la carrera eliminada ya no aparece.

Exportar en csv: El usuario selecciona la opción de exportar en csv las carreras y la aplicación genera la descarga automáticamente.

Caso de uso: Informes.

- **Descripción:** El usuario puede ver los informes de seguimiento de las encuestas realizadas por facultad. Los informes son obtenidos de las encuestas realizadas por reunión.
- **Flujo básico:**
 1. El usuario selecciona la opción "Informes encuesta".
 2. La aplicación pide la selección de la facultad que desea saber el informe.
 3. El usuario selecciona la facultad.
 4. La aplicación le muestra una tabla de todas los mentores evaluados, los campos mostrados en la tabla son: correo, reunión, número de evaluaciones y la nota media de las valoraciones de la encuesta.
 5. Listado los datos el usuario tiene la opción de exportar en formato csv la tabla mostrada.

Exportar en csv: El usuario selecciona la opción de exportar en csv los informes de las valoraciones de la encuesta y la aplicación genera la descarga automáticamente.

Caso de uso: Encuesta

- **Descripción:** El usuario puede evaluar el programa de mentorías con una encuesta final.
- **Flujo básico:**
 1. El usuario selecciona la opción "Encuesta".
 2. La aplicación lo redirecciona a la página de mentorías de la universidad.

La figura (3.1) muestra el diagrama de casos de uso de coordinador central.

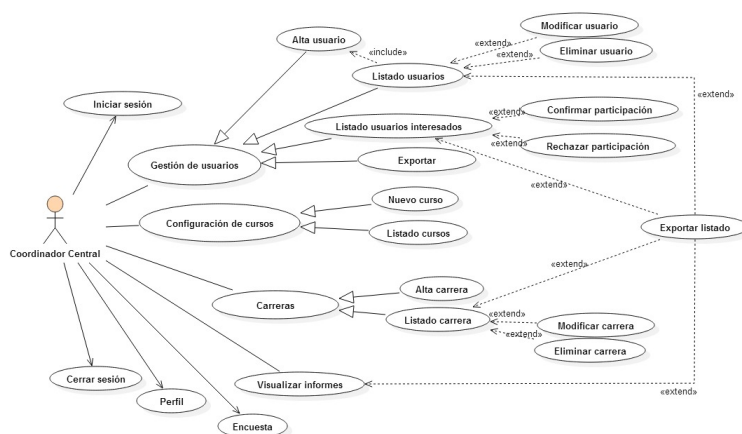


Figura 3.1: Casos de uso de coordinador central

3.3.3. Casos de uso autor: Coordinador de la facultad

Caso de uso: Gestión de Reuniones

- **Descripción:** El usuario puede crear una nueva reunión para sus mentores y telémacos de su facultad, puede ver listado de reuniones creadas donde puede modificar los datos de cada reunión o eliminar la reunión. También puede realizar la exportación de la tabla de reuniones de la base de datos.

Casos de uso especializados de gestión de reuniones:

Nueva reunión

- **Descripción:** El usuario puede crear una nueva reunión donde tiene que indicar el tema de la reunión, la descripción de la misma y la fecha que se celebra.

- **Flujo básico:**

1. El usuario selecciona la opción "Nueva reunión".
2. La aplicación lleva al usuario a la siguiente página donde pide que rellene todos los campos.
3. El usuario rellena los campos pedidos.
4. Si los campos son rellenos correctamente, permite al usuario crear una reunión y aparece un mensaje de confirmación de la nueva reunión creada.

Listado reuniones

- **Descripción:** El usuario puede ver la lista completa de todas las reuniones convocadas, para cada reunión se puede modificar sus datos o eliminarla. Se puede hacer una exportación de la lista completa de los datos en formato csv.

- **Flujo básico:**

1. El usuario selecciona la opción "Listado de reuniones".
2. La aplicación muestra una tabla de todas las reuniones con la opciones de tema, descripción, fecha y la acciones de editar y eliminar. La tabla se puede filtrar por cualquier dato buscado.
3. Listado los datos el usuario tiene la opción de editar o eliminar los datos de cada reunión y además se puede exportar en formato csv todo el contenido de la tabla o de los datos filtrados.

Según la petición que haga el usuario se realizan distintas acciones que explicamos a continuación:

Editar: El usuario al seleccionar la opción de editar una reunión de la lista, la aplicación le muestra una ventana emergente donde puede editar los campos mostrados, una vez editados los datos y darle a la opción actualizar, la aplicación cierra la ventana emergente y muestra un mensaje de confirmación si todo esta correctamente o un mensaje de error si fallo y en la tabla se muestran los campos actualizados.

Eliminar: El usuario selecciona la opción de eliminar una reunión de la lista, la aplicación le muestra una ventana emergente donde le pide la confirmación de la eliminación, si el usuario confirma la eliminación, la aplicación cierra la ventana emergente y le muestra un mensaje de confirmación que ha sido eliminado o un mensaje de error si fallo y en la tabla

la reunión eliminada ya no aparece.

Exportar en csv: El usuario selecciona la opción de exportar en csv las reuniones y la aplicación genera la descarga automáticamente.

Exportar reuniones

- **Descripción:** El usuario puede exportar la tabla completa de reunión de la base de datos con su contenido y se genera un archivo zip con las consultas sql.
- **Flujo básico:**
 1. El usuario selecciona la opción "Exportar".
 2. La aplicación lleva al usuario a la siguiente página donde pide que confirme si realmente quiere hacer una copia de seguridad de la tabla reunión de la base de datos.
 3. El usuario al confirmar la copia de seguridad y seleccionar exportar.
 4. La aplicación genera la descarga del archivo.

Caso de uso: Gestión de mentores.

- **Descripción:** El usuario puede dar de alta un nuevo mentor en su facultad, ver listado de mentores donde puede modificar los datos de cada mentor o eliminar al mentor y también puede realizar una exportación de la tabla de mentores de la base de datos.

Casos de uso especializados de gestión de mentores:

Alta mentor

- **Descripción:** El usuario puede crear un nuevo mentor para su facultad donde tiene que indicar los datos de nombre, correo, dni.
- **Flujo básico:**
 1. El usuario selecciona la opción "Alta mentor".
 2. La aplicación lleva al usuario a la siguiente página donde pide que rellene todos los campos.
 3. El usuario rellena los todos campos.
 4. Si los campos son rellenos correctamente, permite al usuario dar de alta a su nuevo mentor y aparece un mensaje de confirmación.

Listado mentores

- **Descripción:** El usuario puede ver la lista completa de todos mentores de su facultad, para cada mentor se puede modificar sus datos o eliminarlo. Se puede hacer una exportación de la lista completa de los datos en formato csv.
- **Flujo básico:**
 1. El usuario selecciona la opción "Listado y modificación".
 2. La aplicación muestra una tabla con los datos de todos los mentores con la opciones de correo, nombre, dni y la acciones de editar y eliminar. La tabla se puede filtrar por cualquier dato buscado.
 3. Listado los datos, el usuario tiene la opción de editar o eliminar los datos de cada mentor y además se puede exportar en formato csv todo el contenido de la tabla o de los datos filtrados.

Según la petición que haga el usuario se realizan distintas acciones que explicamos a continuación:

Editar: El usuario al seleccionar la opción de editar un mentor de la lista, la aplicación le muestra una ventana emergente donde puede editar los campos mostrados, una vez editados los datos y darle a la opción actualizar, la aplicación cierra la ventana emergente y muestra un mensaje de confirmación si todo esta correctamente o un mensaje de error si fallo y en la tabla se muestran los campos actualizados.

Eliminar: El usuario selecciona la opción de eliminar una reunión de la lista, la aplicación le muestra una ventana emergente donde le pide la confirmación de la eliminación, si el usuario confirma la eliminación, la aplicación cierra la ventana emergente y le muestra un mensaje de confirmación que ha sido eliminado o un mensaje de error si fallo y en la tabla el mentor eliminado ya no aparece.

Exportar en csv: El usuario selecciona la opción de exportar en csv la lista de mentores y la aplicación genera la descarga automáticamente.

Listado mentores interesados

- **Descripción:** El usuario puede ver la lista completa de personas que quieren se mentor y han pedido participar en el programa de

mentorías, para cada mentor se puede aceptar la participación o rechazarla. Se puede hacer una exportación de la lista completa de los datos en formato csv.

- **Flujo básico:**

1. El usuario selecciona la opción "Listado de mentores interesados".
2. La aplicación muestra una tabla con los datos de todas las peticiones de partición con la opciones de correo, nombre, comentario y la acciones de editar y eliminar. La tabla se puede filtrar por cualquier dato buscado.
3. Listado los datos, el usuario tiene la opción de aceptar o rechazar las peticiones de los futuros mentores y además se puede exportar en formato csv todo el contenido de la tabla o de los datos filtrados.

Según la petición que haga el usuario se realizan distintas acciones que explicamos a continuación:

Confirmar: El usuario al seleccionar la opción de aceptar la petición de un mentor de la lista, la aplicación le muestra una ventana emergente donde tiene que confirmar la aceptación de la petición, al darle al botón "Si" la aplicación cierra la ventana emergente y muestra un mensaje de confirmación de la nueva petición.

Rechazar: El usuario selecciona la opción de rechazar la petición de usuario de la lista, la aplicación le muestra una ventana emergente donde le pide la confirmación de la eliminación, si el usuario confirma la eliminación, la aplicación cierra la ventana emergente y le muestra un mensaje de confirmación que ha sido rechazado o un mensaje de error si fallo y en la tabla la petición de mentor se elimina.

Exportar en csv: El usuario selecciona la opción de exportar en csv la tabla de mentores y la aplicación genera la descarga automáticamente.

Exportar

- **Descripción:** El usuario puede exportar la tabla completa de mentores de la base de datos con su contenido y se genera un archivo zip con las consultas sql.
- **Flujo básico:**

1. El usuario selecciona la opción "Exportar".
2. La aplicación lleva al usuario a la siguiente página donde pide que confirme si realmente quiere hacer una copia de seguridad de la tabla mentor de la base de datos.
3. El usuario al confirmar la copia de seguridad y seleccionar exportar.
4. La aplicación genera la descarga del archivo.

Enviar

- **Descripción:** El usuario puede enviar correo a todos sus mentores
- **Flujo básico:**
 1. El usuario selecciona la opción enviar.
 2. La aplicación lleva al usuario a la siguiente página donde pide que indique el tema del mensaje y su texto.
 3. El usuario al darle enviar se envía el correo a todos los mentores de la facultad.

Caso de uso: Gestión de telémacos.

- **Descripción:** El usuario puede dar de alta un nuevo telémaco en su facultad, ver listado de telémacos donde puede modificar los datos de cada telémaco o eliminar al telémaco y también puede realizar una exportación de la tabla de telémacos de la base de datos.

Casos de uso especializados de gestión de reuniones:

Alta telémaco

- **Descripción:** El usuario puede crear un nuevo telémaco para su facultad, donde tiene que indicar los datos de nombre, correo, carrera, turno y mentor asignado.
- **Flujo básico:**
 1. El usuario selecciona la opción "Alta telémaco".
 2. La aplicación lleva al usuario a la siguiente página donde pide que rellene todos los campos.
 3. El usuario rellena los todos campos.
 4. Si los campos son rellenos correctamente, permite al usuario dar de alta a su nuevo telémaco y aparece un mensaje de confirmación.

Listado telémacos

- **Descripción:** El usuario puede ver la lista completa de todos telémacos de su facultad, para cada telémaco se puede modificar sus datos o eliminarlo. Puede hacer una exportación de la lista completa de los datos en formato csv.
- **Flujo básico:**
 1. El usuario selecciona la opción "Listado y modificación".
 2. La aplicación muestra una tabla con los datos de todos los telémacos con la opciones de correo, nombre, correo mentor, turno y la acciones de editar y eliminar. La tabla se puede filtrar por cualquier dato buscado.
 3. Listado los datos, el usuario tiene la opción de editar o eliminar los datos de cada telémaco y además se puede exportar en formato csv todo el contenido de la tabla o de los datos filtrados.

Según la petición que haga el usuario se realizan distintas acciones que explicamos a continuación:

Editar: El usuario al seleccionar la opción de editar un telémaco de la lista, la aplicación le muestra una ventana emergente donde puede editar los campos mostrados, una vez editados los datos y darle a la opción actualizar, la aplicación cierra la ventana emergente y muestra un mensaje de confirmación si todo esta correctamente o un mensaje de error si fallo y en la tabla se muestran los campos actualizados.

Eliminar: El usuario selecciona la opción de eliminar un telémaco de la lista, la aplicación le muestra una ventana emergente donde le pide la confirmación de la eliminación, si el usuario confirma la eliminación, la aplicación cierra la ventana emergente y le muestra un mensaje de confirmación que ha sido eliminado o un mensaje de error si fallo y en la lista el telémaco eliminado ya no aparece.

Exportar en csv: El usuario selecciona la opción de exportar en csv la lista de mentores y la aplicación genera la descarga automáticamente.

Listado telémacos interesados

- **Descripción:** El usuario puede ver la lista completa de personas que quieren ser telémaco y han pedido participar en el programa

de mentorías, para cada telémaco se puede aceptar la participación o rechazarla. Se puede hacer una exportación de la lista completa de los datos en formato csv.

- **Flujo básico:**

1. El usuario selecciona la opción "Listado de telémacos interesados".
2. La aplicación muestra una tabla con los datos de todas las peticiones de partición con la opciones de correo, nombre, comentario y la acciones de confirmar y rechazar. La tabla se puede filtrar por cualquier dato buscado.
3. Listado los datos, el usuario tiene la opción de aceptar o rechazar las peticiones de los futuros mentores y además se puede exportar en formato csv todo el contenido de la tabla o de los datos filtrados.

Según la petición que haga el usuario se realizan distintas acciones que explicamos a continuación:

Confirmar: El usuario al seleccionar la opción de aceptar la petición de un telémaco de la lista, la aplicación le muestra una ventana emergente donde tiene que confirmar la aceptación de la petición, al darle al botón "si" la aplicación cierra la ventana emergente y muestra un mensaje de confirmación de la nueva petición.

Rechazar: El usuario selecciona la opción de rechazar la petición de usuario de la lista, la aplicación le muestra una ventana emergente donde le pide la confirmación de la eliminación, si el usuario confirma la eliminación, la aplicación cierra la ventana emergente y le muestra un mensaje de confirmación que ha sido rechazado o un mensaje de error si fallo y en la tabla la petición del telémaco se elimina.

Exportar en csv: El usuario selecciona la opción de exportar en csv la tabla de telémacos y la aplicación genera la descarga automáticamente.

Exportar

- **Descripción:** El usuario puede exportar la tabla de telémacos de la base de datos con su contenido y se genera un archivo zip con las consultas sql.
- **Flujo básico:**

1. El usuario selecciona la opción "Exportar".
2. La aplicación lleva al usuario a la siguiente página donde pide que confirme si realmente quiere hacer una copia de seguridad de la tabla telémaco de la base de datos.
3. El usuario confirma la copia de seguridad y selecciona "Exportar".
4. La aplicación genera la descarga del archivo.

Enviar

- **Descripción:** El usuario puede enviar correo a todos sus telémacos.
- **Flujo básico:**
 1. El usuario selecciona la opción "Enviar".
 2. La aplicación lleva al usuario a la siguiente página donde le pide que indique el tema del mensaje y su texto.
 3. El usuario confirma el envío.
 4. La aplicación le muestra un mensaje de confirmación o fallo según el caso.

Caso de uso: Gestión de carreras

- **Descripción:** El usuario crear una nueva carrera para su facultad, puede ver la lista completa de todas las carreras en su facultad y para cada carrera puede modificar sus datos o eliminarla. Puede hacer una exportación de la lista completa de los datos en formato csv.

Casos de uso especializados de gestión de carreras:

Nueva carrera

- **Descripción:** El usuario puede crear una nueva carrera para su facultad.
- **Flujo básico:**
 1. El usuario selecciona la opción "Nueva carrera".
 2. La aplicación lleva al usuario a la siguiente página donde pide nombre de la nueva carrera.
 3. El usuario rellena el campo pedido.
 4. Si los campos son rellenos correctamente la aplicación, permite al usuario crear la carrera y si se realiza la acción aparece un mensaje de confirmación de la nueva carrera creada.

Listado Carreras

- **Descripción:** El usuario puede ver la lista completa de todas las carreras de su facultad, para cada carrera se puede modificar sus datos o eliminarla. Se puede hacer una exportación de la lista completa de los datos en formato csv.
- **Flujo básico:**
 1. El usuario selecciona la opción "Listado de carreras".
 2. La aplicación muestra una tabla de todas las carreras con las opciones de nombre y las acciones de editar y eliminar. La tabla se puede filtrar por cualquier dato buscado.
 3. Listado los datos de la carrera tiene la opción de editar o eliminar los datos de cada carrera y además se puede exportar en formato csv todo el contenido de la tabla o de los datos filtrados.

Según la petición que haga el usuario se realizan distintas acciones que explicamos a continuación:

Editar: El usuario al seleccionar la opción de editar un carrera de la lista, la aplicación le muestra una ventana emergente donde puede editar los campos mostrados, una vez editados los datos y darle a la opción actualizar, la aplicación cierra la ventana emergente y muestra un mensaje de confirmación si todo esta correctamente o un mensaje de error si fallo y en la tabla se muestran los campos actualizados.

Eliminar: El usuario selecciona la opción de eliminar una carrera de la lista, la aplicación le muestra una ventana emergente donde le pide la confirmación de la eliminación, si el usuario confirma la eliminación, la aplicación cierra la ventana emergente y le muestra un mensaje de confirmación que ha sido eliminado o un mensaje de error si fallo y en la tabla la carrera eliminada ya no aparece.

Exportar en csv: El usuario selecciona la opción de exportar en csv las carreras y la aplicación genera la descarga automáticamente.

Caso de uso: Informes

- **Descripción:** El usuario puede ver los informes de seguimiento de las encuestas realizadas en su facultad a todos los telémacos donde evalúan a los mentores, mostrando la nota media obtenida por reunión.

- **Flujo básico:**

1. El usuario selecciona la opción "Informes encuesta".
2. La aplicación le muestra una tabla de todas los mentores evaluados, los campos mostrados en la tabla son: correo, reunión, número de evaluaciones y la nota media de las valoraciones de la encuesta.
3. Listado los datos el usuario tiene la opción "Exportar en csv" la tabla mostrada.

Exportar en csv: El usuario selecciona la opción "Exportar en csv" la aplicación genera la descarga automáticamente.

Caso de uso: Encuesta

- **Descripción:** El usuario puede evaluar el programa de mentorías con una encuesta final.

- **Flujo básico:**

1. El usuario selecciona la opción "Encuesta".
2. La aplicación lo redirecciona a la página de mentorías de la universidad.

La figura (3.2) muestra el diagrama de casos de uso de coordinador de la facultad.

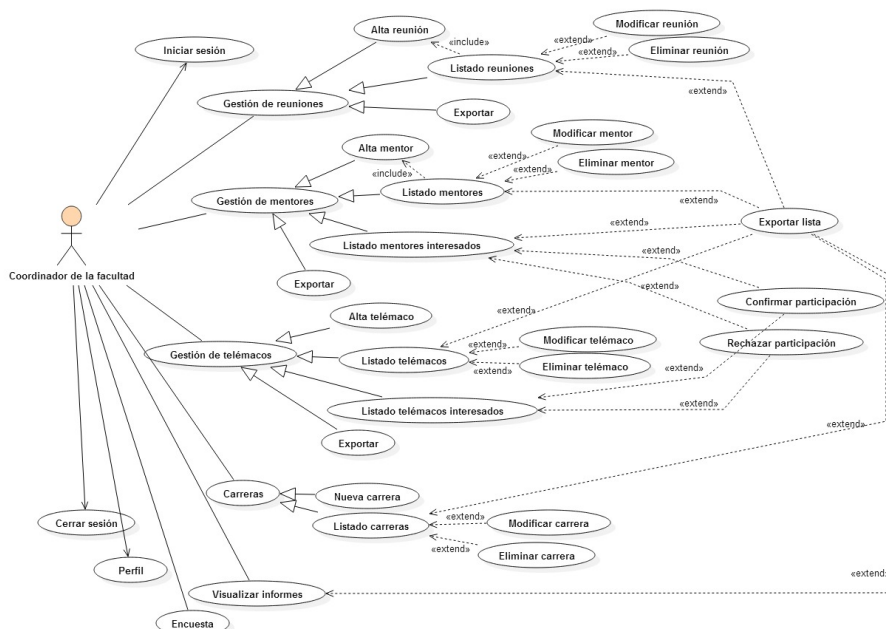


Figura 3.2: Casos de uso de coordinador de la facultad

3.3.4. Casos de uso autor: Mentor

Listado telémacos

- **Descripción:** El usuario puede ver la lista completa de todos sus telémacos, para cada usuario se puede modificar sus datos o eliminarlo. Se puede hacer una exportación de la lista completa de los datos en formato csv.
- **Flujo básico:**
 1. El usuario selecciona la opción "Listado de telémacos".
 2. La aplicación muestra una tabla de todos los telémacos con la opciones de correo, nombre, dni, carrera, turno y la acciones de editar y eliminar. La tabla se puede filtrar por cualquier dato buscado.
 3. Listado los datos el usuario tiene la opción de editar o eliminar los datos de cada telémaco mostrado y además se puede exportar en formato csv todo el contenido de la tabla o de los datos filtrados.

Según la petición que haga el usuario se realizan distintas acciones que explicamos a continuación:

Editar: El usuario al seleccionar la opción de editar un telémaco de la lista, la aplicación le muestra una ventana emergente donde puede editar los campos mostrados, una vez editados los datos y darle a la opción actualizar, la aplicación cierra la ventana emergente y muestra un mensaje de confirmación si todo esta correctamente o un mensaje de error si fallo y en la tabla se muestran los campos actualizados.

Eliminar: El usuario selecciona la opción de eliminar un telémaco de la lista, la aplicación le muestra una ventana emergente donde le pide la confirmación de la eliminación, si el usuario confirma la eliminación, la aplicación cierra la ventana emergente y le muestra un mensaje de confirmación que ha sido eliminado o un mensaje de error si fallo y en la tabla el usuario eliminado ya no aparece.

Exportar en csv: El usuario selecciona la opción de exportar en csv el listado de telémacos mostrados en la tabla y la aplicación genera la descarga automáticamente.

Caso de uso: Informes

- **Descripción:** El usuario puede ver los informes de seguimiento de las encuestas por sus telémacos.

- **Flujo básico:**

1. El usuario selecciona la opción "Informes encuesta".
2. La aplicación le muestra una tabla de todos los mentores evaluados, los campos mostrados en la tabla son: correo, reunión, número de evaluaciones y la nota media de las valoraciones de la encuesta.
3. Listado los datos el usuario tiene la opción "Exportar en csv" la tabla mostrada.

Exportar en csv: El usuario selecciona la opción "Exportar en csv" la aplicación genera la descarga automáticamente.

Caso de uso: Rellenar encuesta

- **Descripción:** El usuario tiene acceso para rellenar una encuesta realizada después de cada reunión.

- **Flujo básico:**

1. El usuario selecciona la opción "Rellenar encuesta".
2. La aplicación lleva al usuario a la siguiente página donde que rellene los campos solicitados.
3. Si los campos son rellenados la aplicación permite al usuario enviar la encuesta.

La figura (3.3) muestra el diagrama de casos de uso de mentor

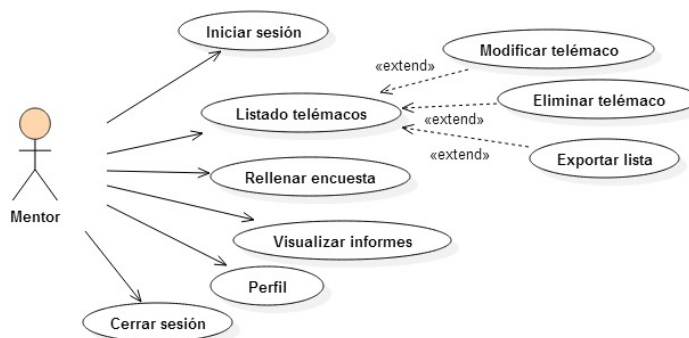


Figura 3.3: Casos de uso de mentor

3.3.5. Casos de uso autor: Telémaco

Rellenar Encuesta

- **Descripción:** El usuario tiene acceso para rellenar una encuesta realizada después de cada reunión para evaluar a su mentor.
- **Flujo básico:**
 1. El usuario selecciona la opción "Rellenar encuesta".
 2. La aplicación lleva al usuario a la siguiente página donde que rellene los campos solicitados.
 3. Si los campos son rellenados la aplicación permite al usuario enviar la encuesta con la evaluación de su mentor.

La figura (3.4) muestra la vista de opción encuesta.

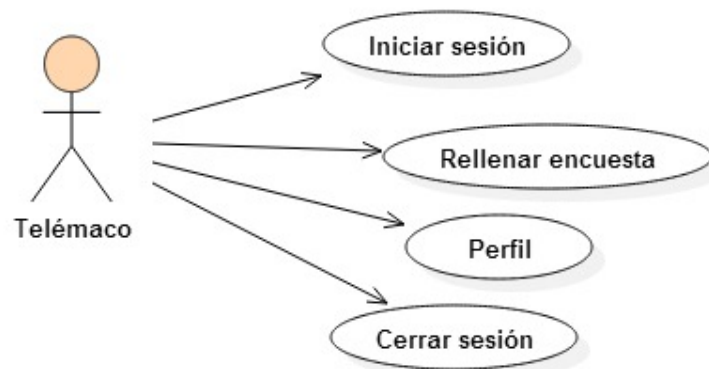


Figura 3.4: Casos de uso de telémaco

Capítulo 4

Implementación

*Si quieres ir rápido, ve solo. Si quieres
llegar lejos, ve acompañado.*

Proverbio africano

En este capítulo se describirán las tecnologías utilizadas en el desarrollo de la aplicación, el prototipo logrado y el resultado final del *front-end*. Además se explicará detalladamente la estructura de la base de datos obtenida después de analizar los requisitos de la aplicación.

Por último se explicará el desarrollo del *back-end*, es decir, la implementación del código, que permite el funcionamiento de la aplicación y quizá uno de los aspectos más interesantes utilizados en el desarrollo de la aplicación es el acceso a la base de datos y la seguridad aplicada para mantener la privacidad de los datos personales.

4.1. Tecnologías utilizadas

Para el desarrollo de la aplicación hemos utilizado tecnologías consideradas actualmente de innovación en el mundo de aplicaciones web, utilizar Java para la implementación del programa junto con el framework de Struts 2, nos permitió ampliar los conocimientos y diseñar una aplicación de fácil y rápida navegación.

4.1.1. Java

Es una tecnología que se usa para el desarrollo de aplicaciones que convierten a la Web en un elemento más interesante y útil. Java no es lo mismo que javascript, que se trata de una tecnología sencilla que se usa para crear páginas web y solamente se ejecuta en el explorador. (Referencia: Oracle (2015a))

4.1.2. Apache Struts 2

Apache Struts 2 es un framework de código abierto de aplicación web para el desarrollo de aplicaciones web Java EE . Utiliza y amplía la API Java Servlet para animar a los desarrolladores a adoptar una arquitectura modelo-vista-controlador (MVC). El framework WebWork escindió de Apache Struts con el objetivo de ofrecer mejoras y refinamientos, manteniendo la misma arquitectura general del marco original de Struts. (Referencia: Wikipedia (2015b))

4.1.3. JSP

La tecnología JavaServer Pages (JSP) permite a los desarrolladores web y diseñadores para desarrollar rápidamente y mantener fácilmente, ricas en información, páginas Web dinámicas que aprovechan los sistemas de negocio existentes. Como parte de la familia de la tecnología Java, la tecnología JSP permite un rápido desarrollo de aplicaciones basadas en Web que son independientes de la plataforma. La tecnología JSP separa la interfaz de usuario de la generación de contenidos, permitiendo a los diseñadores para cambiar el diseño total de la página sin alterar el contenido dinámico subyacente.(Referencia: Oracle (2015b))

4.1.4. XML

Son las siglas en inglés de eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium(W3C) utilizado para almacenar datos en forma legible. Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.(Referencia: Wikipedia (2015c))

4.1.5. MySql

Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más aceptación en todo el mundo. Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples

tablas para almacenar y organizar la información. MySQL destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos. (Referencia: Wikipedia (2015f))

Nosotros la utilizamos por ser de código abierto pero que además es reconocida por su alto rendimiento ya que puede cumplir con las expectativas de rendimiento de cualquier sistema.

4.1.6. Bootstrap

Es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales. (Referencia: Otto (2015))

Nuestro *front-end* ha sido diseñado con este framework y al no ser diseñadores expertos nos ha servido de mucha ayuda, y al tener un diseño responsivo ha mejorado las visualizaciones de nuestra página en múltiples dispositivos.

4.1.7. HTML

Siglas de HyperText Markup Language («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, entre otros. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. (Referencia: Wikipedia (2015d))

4.1.8. JavaScript

(Abreviado comúnmente "JS") es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor. (Referencia: Wikipedia (2015e))

4.2. Estructura de la base de datos

Definidos los casos de uso y puesto en práctica el esquema inicial del proyecto, la creación de la base de datos es fundamental para que el proyecto sea sostenible, debiendo contener una estructura sólida, ya que toda la aplicación funciona en relación a ella.

La base de datos de la aplicación ha sido implementada con MySQL por tener un alto rendimiento y ser de código abierto. El diseño de la base de datos se ha realizado con el modelo entidad-relación, para tener una visión rápida de su estructura. A continuación se muestra cuál el diagrama obtenido y se realiza una explicación teórica de las tablas creadas con relación al diagrama.

4.2.1. Modelo entidad-relación

Para definir una estructura sólida en la base de datos y tener una idea más aproximada a la realidad, seguimos el esquema entidad-relación, que se muestra en la figura (4.1) a continuación:

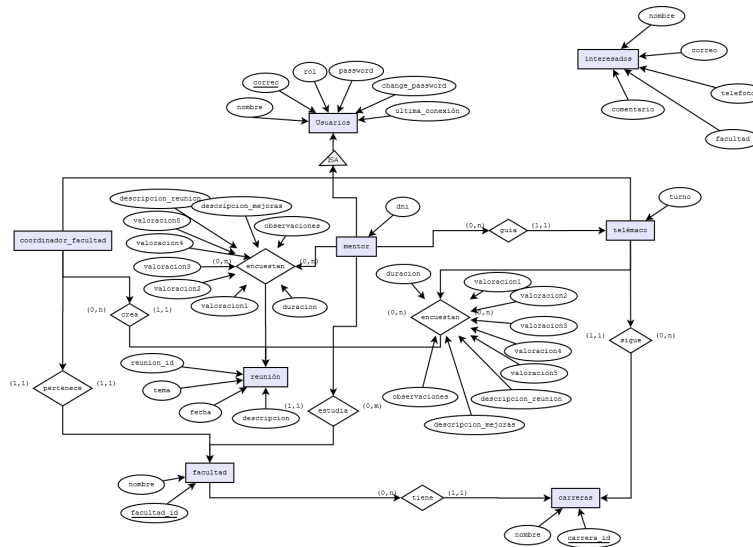


Figura 4.1: Diagrama del modelo entidad-relación

4.2.2. Explicación teórica del modelo entidad-relación

El modelo entidad-relación fue pensado por los siguientes casos:

1. Al tener varios usuarios que desempeñan un rol y cada uno con datos personales distinguibles, la tabla usuarios la hemos definido co-

mo una generalización de las tablas `coordinador_facultad`, `mentor` y `telemaco`.

Tabla `usuarios`: Contiene los atributos comunes de los usuarios como:

- **nombre**: Nombre completo del usuario.
- **correo**: Correo institucional.
- **password**: Contraseña establecida por el usuario.
- **rol**: Es el tipo de usuario, el cual puede ser, coordinador central, coordinador facultad, mentor o telémaco.
- **change_password**: Es un número, que cuando es 1 el usuario no ha iniciado sesión y lo obliga a cambiar la contraseña y cuando está a 0 es porque ya ha cambiado la contraseña.
- **ultima_conexion**: Indica la ultima conexión de la sesión.

Tabla `coordinador_facultad`: Contiene los datos distinguible para el coordinador de la facultad y el identificador de la tabla facultad, que se define como clave única porque solo hay un coordinador por facultad y el correo como clave primaria, porque un coordinador solo puede pertenecer a una facultad. Para especificar que la tabla tiene una generación con la tabla usuarios se genera la relación con el correo de ambas tablas.

- **correo**: Este campo es una referencia al correo de la tabla usuarios.
- **facultad_id**: Este campo es una referencia al identificador de la tabla facultad.

Tabla `mentor`: Contiene los datos distinguibles para el mentor, se define el identificador de la facultad con correo como claves primarias porque un mentor solo puede pertenecer a una sola facultad, pero la misma facultad puede tener muchos mentores. La tabla especifica la generalización de la misma forma que la anterior.

- **correo**: Este campo es una referencia al correo de tabla usuarios.
- **facultad_id**: Este campo es una referencia al identificador de la tabla facultad.
- **dni**: Número nacional de identidad del usuario.

Tabla **telémaco**: La tabla contiene los datos distinguible para el telémaco, aquí existe una relación primordial, que cada telémaco tiene el correo de su mentor y la clave de su carrera por lo que son claves primarias para identificar a un telémaco, ya que no va a existir dos telémacos con datos iguales, la generación que existe con la tabla usuarios se especifica como las tablas anteriores.

- **correo**: Este campo es una referencia al correo de tabla usuarios.
 - **correo_mentor**: Es el correo del mentor que tiene el telémaco, este campo tiene referencia al correo de la tabla mentor.
 - **carrera_id**: Es el identificador de la carrera que estudia el telémaco dentro de su facultad, tiene referencia la con clave primaria de la tabla carrera.
 - **turno**: El campo indica si el alumno tiene un horario de mañana o tarde.
2. Para las facultades y carreras creamos sus propias tablas. La tabla carrera tiene relación con el identificador de la tabla facultad para saber a qué facultad pertenece cada carrera.

Tabla **facultad**: Contiene el identificador único de la facultad y su nombre.

- **facultad_id**: Es el identificador único y clave primaria de la facultad.
- **nombre**: Nombre de la facultad.

Tabla **carrera**: La tabla contiene el identificador de la carrera y su nombre. La tabla está relacionada con el identificador de la tabla facultad para saber su facultad.

- **carrera_id**: Identificador único y clave primaria de la carrera.
 - **facultad_id**: Clave relacionada con la clave de la tabla facultad.
 - **nombre**: Nombre completo de la carrera.
3. Como se celebran reuniones durante el todo el año lectivo y es el coordinador de la facultad quien las convoca, la tabla reuniones tiene relación con la tabla del coordinador de la facultad.

Tabla **reunión**: La tabla contiene los campos necesarios para las reuniones y el identificador del coordinador de la facultad que junto con el identificador de la reunión pasan a ser claves primarias.

- **reunion_id**: Es el identificar de cada reunión.
 - **cood_facultad**: Es el identificador de coordinador que ha creado la reunión.
 - **tema**: Indica el tema que se va hablar de la reunión.
 - **descripción**: Detalla los temas hablamos en la reunión.
 - **fecha**: Especifica la fecha acordada para la reunión.
4. Para cada reunión tienen que ser encuestados los telémacos y mentores, se generó una relación encuesta, esta relación al ser de tipo N:M, es decir, para cada reunión los mentores o telémacos van a ser encuestados y los telémacos o mentores tienen que encuestar todas las reuniones, se crea una tabla encuesta para mentores y otra tabla encuesta para los telémacos, y así tener por separado las encuestas realizadas.

Tabla **encuesta_telemaco** y **encuesta_mentor**: Las tablas contienen atributos idénticos con la excepción que para **encuesta_telemaco** la clave primaria es el correo del telémaco y para **encuesta_mentor** es el correo del mentor que hacen referencia a la tablas telémaco y mentor respectivamente.

Atributos comunes:

- **reunion_id**: Es el identificador de la tabla reunión con la que está relacionado.
- **duración**: La duración de la reunión en formato hh:mm.
- **valoracion1**: Es la valoración de los contenidos abordados en la reunión.
- **valoración2**: Es la valoración del tiempo dedicado a la reunión.
- **valoración3**: Es la valoración del mentor y su comportamiento en la reunión.
- **valoración4**: Es la valoración que indique la utilidad de la reunión.
- **valoración5**: Es la valoración del grado de aprendizaje conseguido.
- **descripcion_reunion**: Es la descripción del funcionamiento correcto de la reunión.
- **descripcion_mejoras**: Descripción de lo peor de la reunión y las mejoras.

- **observaciones:** Las observaciones que dan para mejorar las reuniones.

Atributo `tabla encuesta_telemaco:`

- **correo_telemaco:** Es una clave primaria, que hace referencia a la relación con el correo de la tabla `telémaco`.

Atributo `tabla encuesta_mentor:`

- **correo_mentor:** Es la clave primaria, que hace referencia a la relación con el correo de la tabla `mentor`.

5. Para que un alumno participe en el programa de mentorías, tanto como mentor o `telémaco` tienen que rellenar una solicitud de participación para ser aceptados.

Tabla `interesados:` La tabla contiene los datos que pedimos en el formulario de la inscripción para ser un nuevo `telémaco` o `mentor`.

- **correo:** Es el correo institucional.
- **nombre:** Nombre completo del participante.
- **rol:** El rol que quiere desempeñar.
- **telefono:** El número teléfono móvil del participante para contactar más rápidamente con él.
- **facultad_id:** La facultad a la que pertenece.
- **comentario:** Este atributo es opcional, es por si el participante quiere dejar un mensaje de su participación.

4.3. Front-end

La presentación de todo proyecto es el *front-end*. El *front-end* es la parte visible de la aplicación que interactúa con los usuarios, por ello utilizamos un framework con diseño web adaptable como Bootstrap para generar la aplicación. Bootstrap es perfecto para programadores con conocimientos básicos en diseño, existiendo millones de sitios interesantes que han sido construidos con este framework.

La aplicación fue pensada para la comodidad del usuario, teniendo un acceso rápido y sencillo a sus acciones. Se inicia con una plantilla base para realizar la página principal y una vez reconocidas las acciones de cada usuario se realiza el diseño de las páginas para los roles de los usuarios.

4.3.1. Mockups de la aplicación

Los figuras que presentamos a continuación son los mockups para los distintos tipos de dispositivo que se adapta la aplicación.

Ordenador



Figura 4.2: Mockups de pantalla de ordenador - Portada

Tablet

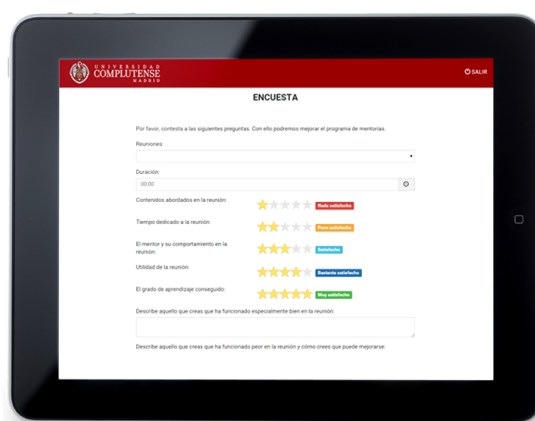


Figura 4.3: Mockups de pantalla de tablet - Encuesta

Móvil



Figura 4.4: Mockups de pantalla de móvil - Portada

4.3.2. Prototipo inicial

El primer prototipo fue el que nos dio una idea base de cómo sería nuestra aplicación, además de darnos a conocer los errores que teníamos para luego encontrar un solución óptima.

Página principal

La página principal fue pensada para que a primera vista, el usuario tenga acceso rápido a la información, esta página es la que comúnmente llamamos “index” fue la primera pagina creada para desarrollar el estilo que contendría el resto de la aplicación.

- **Portada:** Es la imagen principal de la aplicación, donde el usuario pueda tener acceso a la demás información, queríamos algo que sea sencillo pero que la mirarlo se entienda, la imagen no era la más adecuada para lo que queríamos reflejar, por lo que desde un principio se optó por el cambio.



Figura 4.5: Prototipo inicial - Portada

- **Iniciar sesión:** Esta sección es donde el usuario tiene que loguearse para acceder a su espacio personalizado.

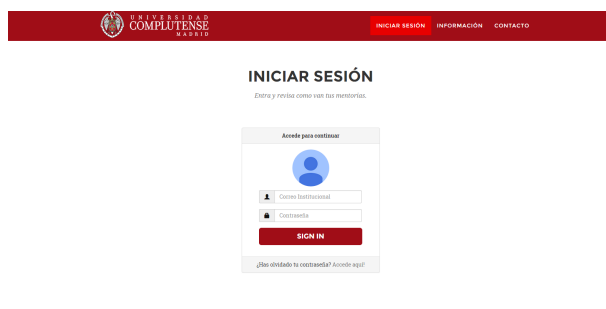


Figura 4.6: Prototipo inicial - Inicio Sesión

- **Información:** Esta sección muestra la información necesaria que un usuario necesita para saber que es el programa de mentorías, como inicio pusimos una información básica sobre lo que es el programa de mentorías.



Figura 4.7: Prototipo inicial - Información

- **Contacto:** Esta sección al principio fue definida como contacto y sería para realizar consultas, que pudieran tener las personas interesadas.

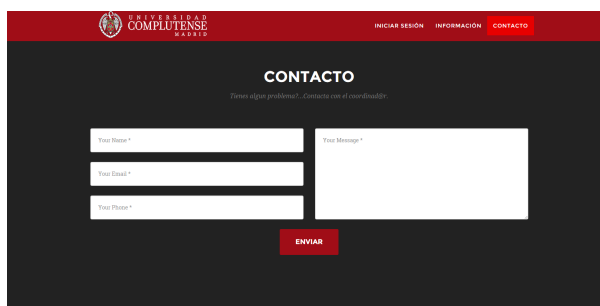


Figura 4.8: Prototipo inicial - Contacto

Las páginas para los usuarios se diseñaron igual a la figura(4.9) pero con las respectivas funcionalidades de sus usuarios.

La figura (4.9) muestra un ejemplo de la vista que tenía el coordinador de la facultad al iniciar sesión.



Figura 4.9: Prototipo inicial - Coordinador facultad

4.3.3. Prototipo final

Después de varios cambios al diseño, el prototipo final obtenido no se desvía mucho al inicial. Se cambiaron los estilos de los botones, enlaces y la forma de visualizar los formularios que anteriormente lo hacíamos con ventanas modales.

Página principal

- **Portada:** El cambio de la imagen en la portada fue primordial, para reflejar lo que el programa de mentorías quiere para sus usuarios.



Figura 4.10: Prototipo final - Portada

- **Iniciar Sesión:** Esta sección no realizó ningún cambio.
- **Información:** Cambiamos la información por la que tiene actualmente la pagina de mentorías de la universidad.



Figura 4.11: Prototipo final - Información



Figura 4.12: Prototipo final - Información detallada

- **Participar:** Anteriormente esta sección era contacto, pero se cambio para que los alumnos puedan participar como mentores o telémacos.

La vista final que tienen cada usuario al iniciar sesión en la aplicación son las siguientes:

Coordinador central



Figura 4.13: Prototipo final - Coordinador central

Coordinador facultad



Figura 4.14: Prototipo final - Coordinador facultad

Mentor



Figura 4.15: Prototipo final - Mentor

Telémaco

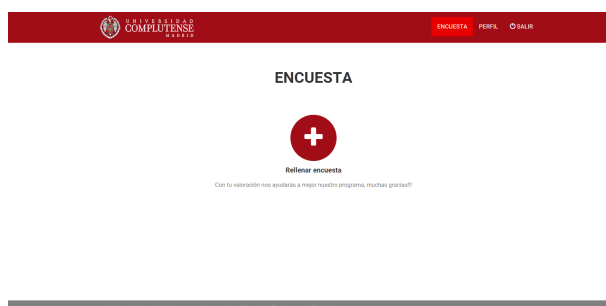


Figura 4.16: Prototipo final - Telémaco

4.3.4. Validación de formularios

La validación de formularios se hizo con un plugin `BootstrapValidation`, que nos permite comprobar que los campos rellenados por el usuario sean correctos.

Si los campos no tiene el formato correcto o un campo obligatorio no esta rellenado no permite pulsar el botón para enviar los datos.

La figura (4.17), muestra el formulario que tiene que rellenar el coordinador central para dar de alta un nuevo usuario.

Figura 4.17: Prototipo final - Validación de formulario

4.4. Back-end

El *back-end* es el lado del servidor que procesa la entrada de la información desde el *front-end*, lo que el *back-end* realmente espera son los datos recolectados del usuario desde el *front-end*, para procesarlos y luego devolver una respuesta al *front-end* para generar una salida entendible para el usuario.

EL *back-end* de nuestra aplicación lo hemos implementado con tecnología Java Server Pages(JSP) que nos permite poner etiquetas HTML normales con segmentos de código *servlet* para la generación del contenido dinámico y Struts 2 que es un framework orientado a acciones, que se basa en declarar su arquitectura en forma de un archivo XML.

4.4.1. Estructura interna: Clases Principales y paquetes

La estructura de nuestra aplicación es muy extensa, por eso vamos a explicar las clases más importantes con sus paquetes contenedores que dan funcionamiento a la aplicación.

El proyecto está realizado en lenguaje Java, bajo el entorno de desarrollo de Java EE para Eclipse.

La aplicación está organizada en paquetes con el prefijo de mentorías, bajo el cual dependen subpaquetes y el archivo de configuración de todas las acciones *struts.xml* como lo muestra en la figura(4.18).

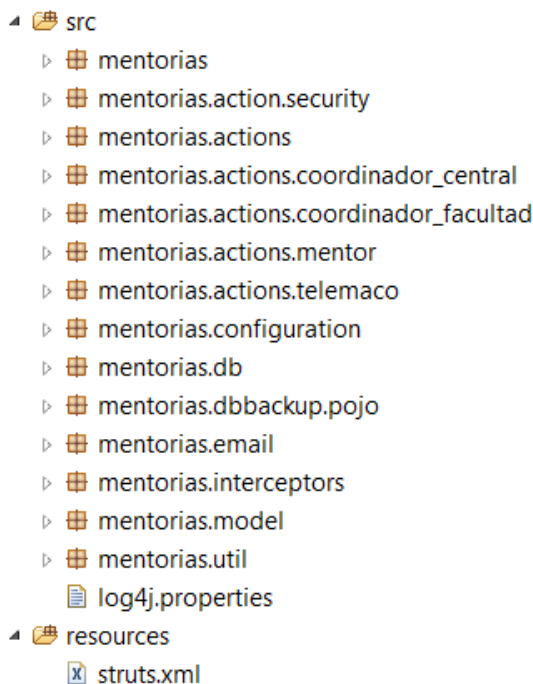


Figura 4.18: Vista estructura de las clases en paquetes

Clases encargadas del inicio de la aplicación

La figura(4.19) muestra el contenido del paquete con las clases que inician la aplicación y cierran la sesión de un usuario.

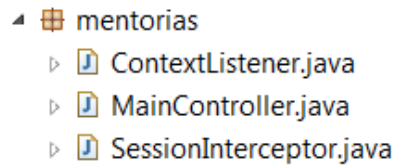


Figura 4.19: Vista de las clases que inician la aplicación

- **ContextListener**: Clase que se encarga de inicializar la aplicación, en la figura (4.20) se muestra su código donde llamar a la clase **MainController** que explicamos a continuación.

```

public class ContextListener implements ServletContextListener {

    static Logger log = Logger.getLogger(ContextListener.class);

    /**
     * On context created/published
     */
    public void contextInitialized(ServletContextEvent ev) {
        try {

            // Preloading all data
            log.info("starting execution");
            ServletContext context = ev.getServletContext();
            String appRoot = context.getRealPath("");
            MainController.setWORKING_PATH(appRoot);
            MainController.getInstance().init().run();

        } catch (Exception e) {
            log.error("Error initializing the application. It was closed immediately");
            System.exit(0);
        }
    }

    /**
     * On context destroy
     */
    public void contextDestroyed(ServletContextEvent ev) {}
}

```

Figura 4.20: Vista del código de la clase ContextListener

- **MainController**: La clase se encarga de comprobar la conectividad a la base de datos, cargar el archivo de configuración y de cargar el módulo de email (sus configuraciones). Si en algunas de las clases se produce un error, la aplicación no se inicia.

La clase está diseñada con el patrón Singleton¹ para asegurarse que solo existe una única instancia de esta clase. Como se ha comentado anteriormente se comprueba si existe una instancia de la clase y sino existe se crea una. Creada la instancia se encarga de llamar a las clases pertinentes.

¹Patrón Singleton: Este patrón está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto . (Wikipedia,2015)

Otras clases diseñadas bajo el mismo patrón Singleton que explicaremos más adelante son: `ConfigurationManager`, `DBHandler` y `Email`.

```

public static MainController getInstance() {
    if (instance == null)
        instance = new MainController();

    return instance;
}

public void run() {
    Log.info("starting execution");
}

public static String getWORKING_PATH() {
    return WORKING_PATH;
}

public MainController init() throws Exception {
    ConfigurationManager.getInstance().init();
    DBHandler.getInstance().init();
    Email.getInstance().init();
    return this;
}

public static void main(String[] args) {
    // Workaround for getting current root path
    getInstance().run();
}

```

Figura 4.21: Vista del código de la clase `MainController`

- `SessionInterceptor`: La clase se encarga de cerrar la sesión del usuario en caso de inactividad pasado los 10 min.

```

public class SessionInterceptor extends AbstractInterceptor {

    private static final long serialVersionUID = -1576483558752163432L;
    static Logger log = Logger.getLogger(SessionInterceptor.class);
    @Override
    public String intercept(ActionInvocation invocation) throws Exception {
        Map<String, Object> session = invocation.getInvocationContext()
            .getSession();
        if (session.isEmpty()){
            log.info("Session expired");
            return "expired";
        } // session is empty/expired

        log.info("Session connected");
        return invocation.invoke();
    }
}

```

Figura 4.22: Vista del código de la clase `SessionInterceptor`

Paquetes principales

- Paquete `security`: Contiene las clases encargadas de la seguridad de la aplicación, las clases `CoordinadorCentralAction`, `CoordinadorFacultadAction`, `CoordinadorMentorAction` y `CoordinadorTelemacoAction` son las encargadas de comprobar el rol de un usuario a iniciar sesión.

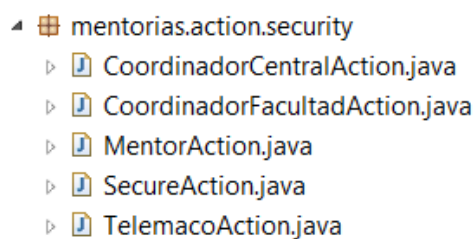


Figura 4.23: Vista de las clases de seguridad

- Paquete `model`: Contiene las clases, que guardan la información obtenida de la base de datos para poder operar esos datos dentro de la aplicación. Son un modelo de clase que se ajusta a los atributos contenidos en una tabla.

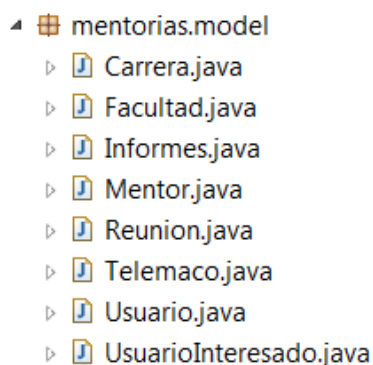


Figura 4.24: Vista de las clases modelo

La figura(4.25) muestra la clase `Facultad` que se ajusta a los atributos de la tabla `facultad` de la base de datos, de esta forma se puede interactuar fácilmente con los datos.

```

public class Facultad {
    private String nombre, facultad_id;

    public Facultad(String nombre, String facultad_id) {
        super();
        this.nombre = nombre;
        this.facultad_id = facultad_id;
    }

    public Facultad() {
        super();
        // TODO Auto-generated constructor stub
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getFacultad_id() {
        return facultad_id;
    }

    public void setFacultad_id(String facultad_id) {
        this.facultad_id = facultad_id;
    }
}

```

Figura 4.25: Vista de las clases Facultad

- Paquete `util`: Este paquete contiene clases necesarias para la aplicación, como encriptación de las contraseñas, generación de contraseñas aleatorias, y comprimir archivos a zip.

```

└─ mentoring.util
   └─ Encryption.java
   └─ Util.java
   └─ ZipFiles.java

```

Figura 4.26: Vista de las clases útiles de la aplicación

- Paquete `actions`: Contiene las clases con las acciones de los usuarios, del cual dependen subpaquetes con las acciones específicas de cada usuario. Las acciones comunes para todos los usuarios son:
 - `ChangePasswordAction`: La clase se encarga del cambio de contraseña de un usuario.
 - `ContactaAction`: Clase que se encarga de las solicitudes de participación que los alumnos envían para ser mentores o telémacos.
 - `LoginAction`: Clase encargada de logear al usuario, si es la primera vez que inicia sesión esta clase llama a `ChangePasswordAction` para que el usuario cambie la contraseña y pueda seguir con el inicio de sesión.

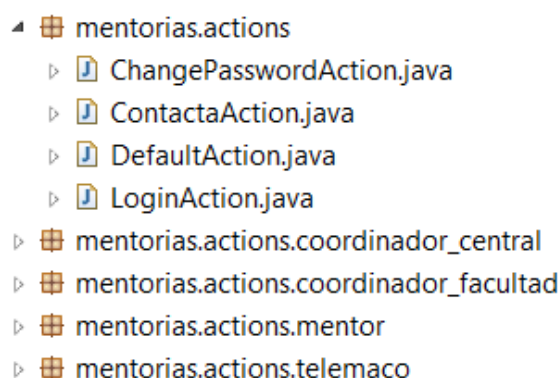


Figura 4.27: Vista de las clases de las acciones de los usuarios

Clases principales

- **ConfigurationManager**: Es la clase encargada de cargar el archivo de configuración de la aplicación. Este fichero de configuración contiene todas las contraseñas y usuarios necesarios para conectarse a la base de datos y a los módulos de correo. La figura (4.28) muestra el código desarrollado para esta clase.

```
public ConfigurationManager init() throws IOException {
    p = new Properties();
    try {
        p.load(new FileInputStream(MainController.getWORKING_PATH() + archivoConfiguracion));
    } catch (IOException ex) {
        Log.error("Not found the config file:" + archivoConfiguracion);
        throw ex;
    }
    return this;
}

public static ConfigurationManager getInstance() {
    if (instance == null)
        instance = new ConfigurationManager();
    return instance;
}
```

Figura 4.28: Vista del código del método ConfigurationManager

- **DBHandler**: Clase encargada de iniciar la conexión a la base de datos. Coge todas las configuraciones del archivo de configuración y comprueba la conectividad con la base de datos, en el caso de que no se tuviera conectividad con la base de datos, se comprueba una segunda vez.

La figura (4.29) muestra el código de la clase.

```

public void init() throws Exception {
    driver_class = ConfigurationManager.getInstance().getAppSetting("db.driverClass");
    if (Util.isNullOrEmptyString(driver_class)){
        Log.error("db.driverClass is not set, please set it in config file.");
        throw new Exception("db.driverClass is not set, please set it in config file.");
    }
    Class.forName(driver_class);
    Log.info("db.driverClass driver set to: " + driver_class);

    connection_url = ConfigurationManager.getInstance().getAppSetting("db.url");
    if (Util.isNullOrEmptyString(connection_url)){
        Log.error("db.url is not set, please set it in config file.");
        throw new Exception("db.url is not set, please set it in config file.");
    }
    Log.info("db.url driver set to: " + connection_url);

    db_name = ConfigurationManager.getInstance().getAppSetting("db.name");
    if (Util.isNullOrEmptyString(db_name))
        throw new Exception("db.name is not set, please set it in config file.");
    Log.info("db.name set to: " + db_name);

    user = ConfigurationManager.getInstance().getAppSetting("db.user");
    if (Util.isNullOrEmptyString(user)) {
        Log.error("db.user is not set, please set it in config file.");
        user = new String();
    } else
        Log.info("db.user set to: " + user);

    password = ConfigurationManager.getInstance().getAppSetting("db.password");
    if (Util.isNullOrEmptyString(user)) {
        Log.error("db.password is not set, please set it in config file.");
        password = new String();
    } else {
        if (Encryption.isPassStyleEncrypted(password))
            password = Encryption.passStyleDecrypt(password);
        else
            Log.info("Security: Please set db.password value to: " + Encryption.passStyleEncrypt(password));
        Log.info("db.password is set.");
    }

    // Trying to perform a connection
    getCon();
    Log.info("Connected successfully to the database");
}

```

Figura 4.29: Vista del código de la clase DBHandler

- **Email:** Esta clase carga la configuración necesaria para poder establecer conexión con el cliente de email. Contiene la funcionalidad para poder enviar mensajes de correo a un único o múltiples usuarios.

La figura (4.30) muestra el método `init` encargado de la configuración inicial de la clase.

```

public Email init() throws IOException {
    p = new Properties();

    p.put("mail.smtp.auth", ConfigurationManager.getInstance()
        .getAppSetting("mail.smtp.auth"));
    p.put("mail.smtp.starttls.enable", "true");
    p.put("mail.smtp.host", ConfigurationManager.getInstance()
        .getAppSetting("mail.smtp.host"));
    p.put("mail.smtp.port", ConfigurationManager.getInstance()
        .getAppSetting("mail.smtp.port"));

    Log.info("Email module has loaded succesfully");

    return this;
}

```

Figura 4.30: Vista del método de la configuración inicial de Email

La figura(4.31) muestra el código del método `sendEmail` encargado de enviar al correo del usuario una contraseña aleatoria.

```
public void sendEmail(String to, String passwordUser) {  
  
    body += "Contraseña: " + passwordUser + "\n";  
    usuario = ConfigurationManager.getInstance().getAppSetting(  
        "mail.correo");  
    password = ConfigurationManager.getInstance().getAppSetting(  
        "mail.password");  
  
    try {  
        if (Encryption.isPassStyleEncrypted(password))  
            password = Encryption.passStyleDecrypt(password);  
        else  
            Log.info("Security: Please set mail.password value to: "  
                + Encryption.passStyleEncrypt(password));  
    } catch (Exception e) {  
        Log.error("Error encrypting the password");  
    }  
  
    Session session = Session.getInstance(p,  
        new javax.mail.Authenticator() {  
            protected PasswordAuthentication getPasswordAuthentication() {  
                return new PasswordAuthentication(usuario, password);  
            }  
        });  
  
    try {  
        Message message = new MimeMessage(session);  
        message.setFrom(new InternetAddress(usuario));  
        message.setRecipients(Message.RecipientType.CC,  
            InternetAddress.parse(to));  
        message.setSubject(subject);  
        message.setText(body);  
  
        Transport.send(message);  
  
        Log.info("Send succesfully: " + to);  
    } catch (MessagingException e) {  
        Log.error(e);  
    }  
}
```

Figura 4.31: Vista del método que envía un correo al usuario

La figura(4.32) muestra el código del método `sendEmailList` encargado del envío de correos múltiple.

```

public void sendEmailList(String asunto, String mensaje,
    List<String> lista) {

    usuario = ConfigurationManager.getInstance().getAppSetting(
        "mail.correo");
    password = ConfigurationManager.getInstance().getAppSetting(
        "mail.password");

    try {
        if (Encryption.isPassStyleEncrypted(password))
            password = Encryption.passStyleDecrypt(password);
        else
            Log.info("Security: Please set mail.password value to: "
                + Encryption.passStyleEncrypt(password));
    } catch (Exception e) {
        Log.error("Error encrypting the password");
    }

    String correos_lista = "";
    for (String correos : lista) {
        if(!correos_lista.isEmpty())
            correos_lista+=",";
        correos_lista += correos;
    }

    Session session = Session.getInstance(p,
        new javax.mail.Authenticator() {
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(usuario, password);
            }
        });

    try {

        Message message = new MimeMessage(session);
        message.setFrom(new InternetAddress(usuario));

        InternetAddress[] iAddressArray = InternetAddress
            .parse(correos_lista);

        message.setRecipients(Message.RecipientType.CC, iAddressArray);
        message.setSubject(asunto);
        message.setText(mensaje);

        Transport.send(message);

        Log.info("Send succesfully to: " + correos_lista);
    }
}

```

Figura 4.32: Vista del método que envía correo a multiples usuarios

- **DBBackupHandler**: La clase hace una copia de seguridad de toda la base de datos, el método `getDumpAllTablesSql` devuelve un `String` con las consultas en sql, para que posteriormente se pueda generar el archivo sql.

```

/**
 * Generates a dump from all tables
 *
 * @return
 * @throws SQLException
 */
public String getDumpAllTablesSql() throws SQLException {
    ResultSet allTablesRs = null;
    StringBuilder sb = new StringBuilder();
    sb.append("-- -----\n");
    sb.append("-- Database dump of database:");
    String database = DBHandler.getInstance().getDBName();
    sb.append(database);
    sb.append("\n");
    sb.append("-- Date: ");
    sb.append(new Date());
    sb.append("\n");
    sb.append("-- -----\n\n");

    sb.append(" -- Disabling foreign checks to avoid errors:\n");
    sb.append("SET FOREIGN_KEY_CHECKS=0;\n\n");

    try {
        Log.info("Retrieving all tables from database..");
        allTablesRs = getAllTablesFromConnection();
        while (allTablesRs.next()) {
            String tableName = allTablesRs.getString(1);
            sb.append(getDumpTablesSql(database, tableName));
        }
        Log.info("Finished!");
    } catch (Exception e) {
        Log.error("Exception trying to get all tables from current database, with msg: "
            + e.getMessage());
    } finally {
        if (allTablesRs != null)
            DBHandler.tryCloseResultSet(allTablesRs);
    }
    return sb.toString();
}

```

Figura 4.33: Vista del código del método `getDumpAllTablesSql`

Archivo de configuración de Struts 2

`struts.xml`: Es el archivo más importante de la aplicación ya que es el archivo de configuración de Struts 2 donde se definen las acciones, es decir, el *mapping* de la aplicación. Con este archivo la aplicación sabe a qué página redirigir en respuesta a una acción y que clase se encarga de realizar la acción.

Para no estar creando por cada *action* una función lo que se hace es que en el *action* ejecuta el método `execute` y se le envía un parámetro "accion" que identifica la acción a ejecutar.

Cuando el *action* devuelve un resultado, ya sea de confirmación o de error se redirige hacia el action principal del que proviene.

4.4.2. Seguridad en la aplicación

Uno de los campos más importantes en una aplicación web es la seguridad, ya que gracias a esta seguridad se mantiene la privacidad de la información personal como profesional de los usuarios que la utilizan.

Una vulnerabilidad se describe como una debilidad en la aplicación, la cual es aprovechada por un atacante para comprometerla. La labor de los

desarrolladores es evitar que existan vulnerabilidades en la aplicación.

Para garantizar la seguridad en la aplicación se ha optado por una serie de medidas, como la encriptación de contraseñas y la utilización del protocolo "https" que más adelante se explicarán a detalle.

Encriptación de contraseñas

El estándar de cifrado avanzado AES (Advanced Encryption Standard), es uno de los algoritmos más seguros y más utilizados hoy en día, disponible para uso público. Está clasificado por la Agencia de Seguridad Nacional, National Security Agency (NSA), de los Estados Unidos para la seguridad más alta de información secreta "Top Secret". (Wikipedia, 2015a)

AES es un esquema de cifrado por bloques, con claves simétricas que soportan bloques de como mínimo 128 bits. Para darle seguridad a la aplicación hemos implementado una clase basada en este algoritmo, que se encarga de encriptar la contraseña del usuario y no sea visible para terceros, lo que la clase realiza es la encriptación y desencriptación de la contraseña introducida por el usuario con una llave establecida por nosotros.

Protocolo HTTPS

Además gracias a la seguridad "https" durante todo el proceso de *login* la conexión está cifrada en todo momento y se fuerza al usuario a que siempre viaje en conexiones seguras.

Una vez que el usuario inicia sesión se establece dos capas de seguridad:

- La primera por struts que con la ayuda de los interceptores mantenemos la sesión del usuario, para que expire y sea única (`SessionInterceptor`).
- Y una segunda añadimos la funcionalidad de struts aplicando roles de seguridad a cada usuario dependiendo del rol. Si el usuario no pertenece a su rol correspondiente inmediatamente se redirige hacia la página principal. Además se añade un contador de actividad para que expire la sesión si el usuario no está activo (10 minutos). Pasado este tiempo automáticamente se le expulsa hacia el *login* borrando la sesión.

Copia de Seguridad

Se creó un script de linux que hace un `dump2` de la base de datos, es decir, el comando genera una copia de seguridad programada diariamente a las 9.15 de la mañana (`crontab`).

²Comando de Linux que hace copias de seguridad de archivos

Capítulo 5

Conclusiones y trabajo futuro

*Siempre que te pregunten si puedes hacer
un trabajo, contesta que sí y ponte
enseguida a aprender como se hace.*

Franklin Delano Roosevelt

Terminado el proyecto, hemos llegados a las siguientes conclusiones:

- La necesidad de crear un espacio interactivo donde los estudiante de la Universidad Complutense de Madrid reciban el apoyo necesario para acoplarse a la vida universitaria con facilidad nos ha permitido crear un sitio acorde a las necesidades de los mismos, con una nueva y renovada estructura, pero tomando como referencia el sitio anterior, así mismo, los roles de los usuarios han sido modificados brindándoles mayores alternativas dentro de la página.
- La experiencia con la tecnología Struts 2 nos ha permitido desarrollar una aplicación web de complejidad media, así mismo esta tecnología permite desarrollar aplicaciones web dentro de la plataforma Java EE, pudiendo modificar y heredar datos sin problema.
- El nivel de involucramiento por parte del sistema para los nuevos estudiantes sería muy alto, lo que significa que la universidad de cierta manera, transmite desde sus estudiantes experimentados a sus estudiantes juniors metodologías y técnicas de cómo asimilar, mejorar y avanzar de manera homogénea con el resto de estudiantes en lo respectivo al conocimiento
- Finalmente, implementar un programa en toda la comunidad universitaria como prototipo para el resto de facultades, por su funcionalidad y adaptabilidad y que actualmente ha sido puesto en práctica en la facultad de Informática.

Conclusions

Finished the project, we have arrived at the following conclusions:

- *The need to create an interactive place where students at the Complutense University of Madrid receive the support needed to engage the university life easily allowed us to create a site according to their needs, with a renewed structure, but with reference to the previous site, likewise, the user roles have been changed to provide a greater choice within the page.*
- *Experience with Struts 2 technology has allowed us to develop a web application of complex average, furthermore this technology allows to develop web applications within the Java EE platform, inherit and modify data without problem.*
- *The level of involvement for new students by the system would be very high, which means that the university somehow, transmit from their experienced students to the new students some methods and techniques to learn, improve and step up in the same way as the rest of the classmates.*
- *Finally, implement a program in the university community as a prototype for the rest of the faculties, for their functionality and adaptability and which has now been implemented in the Computer Engineering College.*

5.1. Recomendaciones

- Se recomienda implementar la aplicación web del programa de mentorías en el menor tiempo posible en el resto de facultades de la universidad para mantener una plataforma unificada en su totalidad.
- Se recomienda capacitar a los posibles usuarios de la página en la nueva aplicación para que se adapten con facilidad e identifiquen sus nuevos roles.
- Se recomienda estructurar un plan de contingencia, en el hipotético caso que en algún momento el sistema llegue a presentar fallas o colapse, por motivos externos
- Se recomienda que el mentor haga un seguimiento al rendimiento académico del estudiante, para corroborar la factibilidad del programa y así mantener la continuidad del mismo.

5.2. Descripción del trabajo realizado por los alumnos

Contamos personalmente la colaboración de cada integrante del proyecto.

■ Colaboración de Sergio

Durante la parte inicial del proyecto investigué la nueva tecnología del proyecto “Bootstrap” creando prototipos iniciales que no tienen nada que ver con la versión final.

Primeramente definimos todos los requisitos mediante casos de uso, permitiéndonos elaborar un mapa web y definir la base de datos inicial que posteriormente cambiaríamos.

Con mi compañera realizamos la maquetación de una plantilla por defecto del cual iríamos añadiendo aspectos según el rol, cumpliendo los roles de coordinador central y mentor.

Todas las funciones desarrolladas las realizamos con modales, cosa que tuvimos que modificar en toda la aplicación reemplazándolas por páginas.

Gracias a la tecnología de Bootstrap pudimos realizar un diseño adaptable para la mayoría de los dispositivos gracias al uso de la sintaxis de dicha tecnología, con el simple hecho de añadir etiquetas específicas.

Una vez que obtuvimos un prototipo casi final de la aplicación, desarrollé la aplicación que se encargaría de gestionar todos los datos.

Nuestro tutor nos proporcionó un servidor donde se pueda desplegar la aplicación y nos dió total libertad de uso por lo que instalé una serie de servicios que me serían útiles para una rápida coordinación entre profesor y mi compañera.

Instalé un servidor web (Apache HTTP Server) para poder subir el html “estático” de nuestra web, la base de datos MySQL, Tomcat 7 para el despliegue de la aplicación en java y un subversión para la gestión de los cambios de la aplicación.

Gracias a la instalación del servidor pudimos coordinar mejor los cambios realizados en la página y estar en constante comunicación. Todos estos cambios los hacíamos con notepad++ conectarnos por Ftp al servidor con el plugin que este trae por defecto (nppFtp).

Una vez que configuré todo lo necesario para el desarrollo de la aplicación comencé a investigar con qué tecnología se desarrollaría la aplicación. Decidimos utilizar Struts2 porque nos parecía bastante interesante el uso de acciones mediante un archivo de configuración XML y

el uso que hace en sí del patrón Modelo Vista Controlador.

Empecé a crear la aplicación desde cero añadiendo todas las librerías necesarias y la estructura de la aplicación.

Durante el desarrollo inicial le di prioridad a la seguridad de la aplicación creando clases para la encriptación/descriptación de las contraseñas y creando una clase principal del cual van a extender todas las acciones de la aplicación (cada rol tiene una clase encargada de controlar el rol del usuario y comprobar si es correcto) y que controla la sesión y el tiempo de actividad del usuario.

Una vez creada la seguridad, empecé a desarrollar todas las acciones pidiendo a mi compañera las consultas sql para agilizar el desarrollo.

Viendo que continuamente modificábamos la base de datos, decidí crear un script¹ para hacer una copia de seguridad diaria de la estructura de la base de datos, añadiéndolo al crontab.

Como la aplicación estaba puesta en producción y para que el usuario no experimente cambios, decidí crear un contexto para subir la aplicación con los cambios pedidos por el tutor, para que diera el visto bueno antes de pasarlo a producción.

Posteriormente nos ofrecieron la opción de participar en un proyecto de colaboración, por el cual decidimos que la aplicación viajará en modo seguro por "https", por lo que creé un certificado de seguridad y modifiqué algunos archivos de Tomcat² para redirigir todas las conexiones al protocolo "https".

■ Colaboración de Johanna

Primeramente, inicié este gran proyecto informándome sobre el programa de mentorías de la universidad, visitando el sitio actual del programa, conociendo la opinión de los encargados de la página e identificando cuáles eran las principales necesidades de los usuarios.

Definidos los usuarios de la aplicación y los casos de uso iniciales pensamos en el diseño del front-end. El front-end decidimos desarrollarlo

¹Archivo de órdenes

²<http://tomcat.apache.org/>

con Bootstrap, para ello investigué tutoriales en internet que me permitieran conocer su funcionamiento, informándome que este framework es uno de los más utilizados en la actualidad, por lo que existe gran cantidad información disponible.

La investigación se llevó a una plantilla que sirvió como base para el framework, adaptándola para crear una plantilla que sirviera para nuestra aplicación, tomando la decisión de utilizar los colores corporativos de la universidad y mejorando el diseño actual de la página con tendencias modernas que llamasen la atención de los nuevos estudiantes.

Nos dividimos las tareas de creación del front-end con mi compañero, para desarrollarlo por usuarios, primeramente desarrollamos el prototipo inicial. Por mi parte, los usuarios que desarrollé son: coordinador de la facultad y telémaco.

Teniendo la base del diseño desarrollamos las funcionalidades para cada usuario, igualmente nos dividimos el trabajo para los usuarios asignados anteriormente. Al principio de la maquetación guardábamos copias del trabajo realizado en una carpeta compartida por Google Drive pero una vez que mi compañero instaló el servidor web, realizamos una mejor coordinación del trabajo realizado, conectándonos con el plugin de Ftp de Notepad++³.

También me encargué de las modificaciones realizadas en el front-end y la validación de los formularios contenidos en nuestra aplicación. La validación la decidimos hacer por JavaScript, para lo que utilizamos JQuery, un librería de script. Para realizar las validaciones utilicé el plugin BootstrapValidator⁴, que nos permitió hacer validaciones de forma más dinámica en el código HTML.

Para definir la estructura de la base de datos utilicé el modelo entidad-relación transformándolo a tablas con Mysql, obteniendo la base de datos inicial, ya que al ampliar el proyecto a toda la comunidad universitaria, necesitábamos crear nuevas tablas y añadir algunos atributos a otras.

Cuando los cambios de front-end estaban casi terminados, empecé con la redacción de la memoria del proyecto, acogiendo la sugerencia de

³Programa editor de texto - <https://notepad-plus-plus.org/>

⁴<http://formvalidation.io/validators/>

nuestro tutor se realizó en $\text{T}_{\text{E}}\text{X}_{\text{S}}^5$, una plantilla para escribir tesis doctorales, trabajos de grado y trabajos de investigación en $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, por lo que era perfecto para nuestra memoria, teniendo también que investigar el funcionamiento del nuevo programa.

La redacción de la memoria la realicé en un documento compartido en Google Drive para que mi compañero y tutor conocieran los avances y realizarán modificaciones si fuera necesario, lo que nos permitió a todos trabajar en el mismo documento.

5.3. Trabajo futuro

Dado el gran desarrollo de nuestra aplicación y a la utilidad de la misma se pueden ampliar muchas funcionalidades que puede tener un programa de mentorías, es por ello que el trabajo futuro que vamos a realizar, es gracias a una beca concedida por la Red Iberoamericana de Mentorías que desean tener en funcionamiento nuestra aplicación, con las siguientes modificaciones:

- Se creará una versión configurable para los colores y logotipos institucionales de cada universidad.
- Se creará máquinas virtuales para su distribución.
- Se realizará corrección de fallos y mantenimiento por un periodo de 4 meses.

Algunas de las instituciones participantes dentro de la red Iberoamericana de mentorías son:



Figura 5.1: Algunas instituciones participantes en la Red Iberoamericana de mentoría

⁵<https://gaia.fdi.ucm.es/research/texis>

Bibliografía

*Es mucho más recompensante llegar a la
cima de la montaña y compartir tu
experiencia con otros que mostrarte tu
solo exhausto.*

Shandel Slaten

- ORACLE, I. About java. 2015a. Disponible en <https://java.com/es/about/> (último acceso, Junio, 2015).
- ORACLE, I. Javasever pages. 2015b. Disponible en <http://www.oracle.com/technetwork/java/overview-138580.html> (último acceso, Junio, 2015).
- OTTO, M. Bootstrap. 2015. Disponible en <http://getbootstrap.com/> (último acceso, Junio, 2015).
- WIKIPEDIA, F. Advanced encryption standard. 2015a. Disponible en http://es.wikipedia.org/wiki/Advanced_Encryption_Standard (último acceso, Mayo, 2015).
- WIKIPEDIA, F. Apache struts 2. 2015b. Disponible en http://en.wikipedia.org/wiki/Apache_Struts_2 (último acceso, Mayo, 2015).
- WIKIPEDIA, F. Extensible markup language. 2015c. Disponible en http://es.wikipedia.org/wiki/Extensible_Markup_Language (último acceso, Junio, 2015).
- WIKIPEDIA, F. Html. 2015d. Disponible en <http://es.wikipedia.org/wiki/Html> (último acceso, Mayo, 2015).
- WIKIPEDIA, F. Javascript. 2015e. Disponible en <http://es.wikipedia.org/wiki/JavaScript> (último acceso, Mayo, 2015).
- WIKIPEDIA, F. Mysql. 2015f. Disponible en <http://es.wikipedia.org/wiki/MySQL> (último acceso, Junio, 2015).

*Muchos de nosotros somos más capaces que algunos de nosotros,
pero ninguno de nosotros somos tan capaces como todos nosotros.*

Tom Wilson

