



ESCOLA TÉCNICA SUPERIOR
DE ENXEÑARÍA

SOLICITUDE DE APROBACIÓN DE ANTEPROXECTO DO TRABALLO DE FIN DE GRAO GRAO EN ENXENARÍA INFORMÁTICA.

Entréguese na Administración da ETSE antes da data indicada no Artigo 15 do Regulamento de traballo fin de grao enxeñaría informática:
O Anteproxecto terá que ser aprobado pola Comisión de Traballos Fin de Carreira de Enxeñaría Informática como mínimo tres meses antes da data de depósito e de solicitude de trámite de defensa do TFG

Datos do/a Alumno/a			
Nome: Pablo González Alonso		DNI: 53185661R	
Enderezo: Subida a Madroa 51 28C		Localidade: Vigo	
Provincia: Pontevedra	C.P.: 36317	Teléfono: 698176253	Correo-e USC: pablo.gonzalez.alonso@rai.usc.es

Requisitos
O alumnado poderá ter pendentes como máximo 75 créditos para completar os estudos, excluídos os correspondentes ao TFG

Datos do Traballo de Fin de Grao	
Título: Desarrollo de una aplicación web para la gestión del mentoring	
Cotitora: Loreto Fernández Fernández Cotitor: José M. Cotos Yáñez	Correo-e: loreto.fernandez@usc.es Correo-e: manel.cotos@usc.es
Áreas de coñecemento: Economía Financiera y Contabilidad. Linguaxes e Sistemas Informáticos.	
Departamento: Dept. Economía Financeira e Contabilidade Dept. Electrónica e Computación	
Empresa (se procede):	

- ☐ O TFG implica intervencións con seres humanos, as súas mostras ou os seus datos.
☐ O TFG realízase vinculado a un proxecto de investigación que contan con informe do Comité de Bioética da USC, do CEI do SERGAS ou doutro órgano equivalente.

(De ser o caso, será preciso adxuntar o informe favorable do Comité de Bioética no correspondente prazo de solicitude de defensa do TFG)

A persoa que asina e cos datos que se indican solicita á Comisión de Traballos Fin de Grao de Enxeñaría Informática a aprobación do anteproxecto que se acompaña.

Santiago de Compostela, 24 de febreiro de 2022

O/A alumno/a

Vº e Pr. O/A Titor(a) e Cotitor(a) do proxecto

Asdo: Pablo González

Asdo: Loreto Fernández

Asdo: José M. Cotos

DESCRIPCIÓN DO ANTEPROXECTO

TÍTULO:

Desarrollo de una aplicación web para la gestión del mentoring

Introdución

El mentoring es uno de los métodos más eficaces para la gestión del talento, ya que permite establecer una relación de ayuda o apoyo entre un profesional cualificado en un determinado campo (mentor) y una persona que requiere una orientación en este ámbito, estudiante o egresado universitario, en particular, a la que transmite sus conocimientos y experiencia (mentorizado). Esta relación puede tener como fin, tanto orientar al mentorizado en la búsqueda de empleo, como apoyarle en el desarrollo de su proyecto profesional.

Cabe destacar que la mentorización no establece ningún tipo de contrato entre las dos partes, si no que se trata de una participación voluntaria de profesionales comprometidos para acompañar, orientar y apoyar a los más jóvenes en el comienzo de su carrera profesional, preparándolos para el tránsito al mercado laboral con las competencias, actitudes, valores y habilidades profesionales y personales necesarias.

Si bien es cierto que existen programas de mentoring en diversas instituciones académicas, como en la USC [1], y entidades profesionales, como el Colegio de Economistas de A Coruña [2], pocos se organizan a través de una plataforma online. Por lo tanto, este TFG pretende desarrollar una aplicación web que permita facilitar y gestionar el proceso de mentoring en un entorno académico, poniendo en contacto a estudiantes universitarios de último año de carrera con profesionales que les puedan aportar experiencia para mejorar su carrera profesional. También serviría para ayudar a los futuros estudiantes a tomar la decisión correcta a la hora de seleccionar el Grado o Master a realizar, o para guiar a los nuevos estudiantes al comienzo de sus estudios.

Esta aplicación permitirá a los mentorizados buscar mentores según el área de conocimiento en el que estén interesados y ponerse en contacto con ellos a través de la propia aplicación. Esta estará compuesta de dos partes claramente diferenciadas: por un lado, un frontend, que sería la parte visible para los usuarios; y, por el otro, un backend, compuesto por la lógica de negocio y la base de datos. Esta segunda parte requerirá de un administrador para realizar el mantenimiento de la aplicación.

Se buscará que la aplicación sea sencilla, intuitiva y usable, y que pueda ser compatible con los diferentes navegadores y dispositivos (tanto ordenadores como dispositivos móviles).

Obxectivos

El objetivo principal del proyecto es el desarrollo de una aplicación web en Java para la gestión del mentoring. Este objetivo está compuesto de los siguientes subobjetivos:

- La aplicación debe permitir a los dos tipos de usuarios registrarse, creando un perfil con sus datos personales, profesionales y de contacto. Dichos perfiles deben poder ser modificados por su propietario y ser accesibles por los demás usuarios.
- Los mentorizados deben tener la posibilidad de establecer un contacto con un mentor, ya sea de manera externa (por ejemplo, mediante correo electrónico) o a través de la propia aplicación, mediante un chat.

- Los usuarios deben poder cargar ficheros en sus perfiles o intercambiarlos entre ellos (mentor-mentorizado) a través de la aplicación. Dichos ficheros deben almacenarse en los repositorios de la aplicación de forma eficiente.
- La aplicación debe ser usable e intuitiva, con una interfaz de usuario agradable y responsive. La aplicación también debe ser modular y segura.
- Los datos almacenados por la aplicación deben servir para generar informes que permitan a los usuarios conocer la actividad y el estado de la propia aplicación.

Descripción técnica

Como ya se comentó, la aplicación estará compuesta de dos partes, por un lado, el backend, que estará formado a su vez por una base de datos y una aplicación Rest. Para la base de datos se hará uso de PostgreSQL, dada su gran cantidad de documentación, facilidad de uso y a mi propia experiencia con esta herramienta.

Para implementar la aplicación Rest, se hará uso de Java, utilizando Spring Boot [3] a través del IDE Spring Tool Suite, en concreto, la versión para Eclipse. El uso de este IDE facilitará el desarrollo de la aplicación puesto que está especialmente diseñado para desarrollar aplicaciones Rest basadas en Spring Boot. También hay que añadir que incluye Maven [4], que se empleará para facilitar la gestión de dependencias; Hibernate [5], es una herramienta de mapeo objeto/relacional que facilitará el mapeo entre los objetos java y las tablas de la base de datos; y JUnit y Mockito, que se utilizará para el testeo de la aplicación. Además de usar estas dos últimas herramientas para realizar pruebas tanto unitarias como de requisitos, también se utilizarán los principios de Nielsen [6] para realizar test de usabilidad a posibles usuarios finales.

Por otro lado, para el frontend, se usará el framework AngularJS [7], dado que permitirá desarrollar una aplicación web con vistas dinámicas, facilitando la manipulación del árbol DOM de HTML y la conexión con la aplicación Rest. Para desarrollar el frontend se utilizará el IDE Visual Studio Code o similar. También se empleará una herramienta de diseño de interfaces web como Dreamweaver para agilizar y facilitar el desarrollo de la interfaz de usuario.

Se utilizará el patrón MVC [8] (Modelo-Vista-Controlador), uno de los modelos de diseño de software más usados y maduros. Este modelo separa los datos de una aplicación y su lógica de negocio (Modelo), de la interfaz de usuario (Vista) y de la lógica de control (Controlador). En nuestro caso, el modelo estará formado por las clases de la aplicación web, tanto las Beans como aquellas dedicadas a la lógica de negocio y al acceso a la base de datos; el controlador será la clase asignada como controladora Rest, y la vista será el frontend.

Para el control de versiones se hará uso de GitHub, almacenando el código en sus repositorios y gestionándolos a través de Sourcetree [9]. Para el diseño de los diagramas se utilizarán herramientas como StarUML. También se utilizará OpenProject o similar como herramienta de planificación temporal del proyecto.

Para diseñar el logo de la aplicación, así como otras posibles ilustraciones o figuras a utilizar, se utilizarán herramientas de diseño como GIMP. Asimismo, para diseñar las diferentes pantallas de la aplicación web se utilizará una herramienta de wireframes y mockups online.

Para generar los informes de la aplicación se hará uso de Microsoft PowerBI. Esta herramienta se conectará con la base de datos para extraer la información necesaria, y se generará un informe a partir de los datos. Finalmente, el informe estará disponible para los usuarios interesados, se renovará mensualmente con los nuevos datos (dado que la estructura del informe ya estaría creada, solo sería necesario indicar a la herramienta que actualice los datos) y se publicará.

El ciclo de vida de desarrollo del proyecto será el incremental. En cada uno de los incrementos se añadirán funcionalidades nuevas a la aplicación, pero también se mejorarán o se corregirán las ya existentes en caso de necesidad. Cabe destacar que no se tratará de un ciclo de vida Incremental

puro, sino que será un híbrido ágil, lo que permitirá realizar estos cambios o modificar lo desarrollado en los incrementos anteriores.

A continuación, se procede a detallar las funcionalidades principales que tendría esta aplicación. En primer lugar, la funcionalidad base será la gestión de usuarios. La aplicación tendrá los dos tipos de usuarios ya comentados, mentores y mentorizados, ambos podrán registrarse/iniciar sesión en la aplicación para poder acceder a todas sus funcionalidades. Al registrarse tendrán que facilitar información personal, profesional y de contacto, en el caso de los mentores, deberán indicar las áreas de conocimiento en las que podrían prestar apoyo. Los mentorizados podrán realizar búsquedas de los mentores disponibles, utilizando para ello una serie de filtros, y acceder al perfil de los mismos, donde podrán visualizar la información pública que estos hayan querido mostrar, incluidos sus datos de contacto.

En segundo lugar, la aplicación dispondrá de una herramienta de chat para que los usuarios puedan comunicarse directamente desde la aplicación. Para iniciar un chat, un mentorizado deberá enviar una solicitud al mentor escogido, presentándose y explicando brevemente el asunto por el cuál desea contactar. Si el mentor lo aceptara, ambos podrían comunicarse a través de un chat. Para implementar esta herramienta se hará uso de un API de mensajería, Converse.js [10] o similar, que nos permitirá gestionar los chats de forma eficiente. Por otro lado, para dar soporte a esta API necesitaremos un servidor XMPP, Ejabberd [11] o similar.

En tercer lugar, la aplicación proporcionará informes de su estado, que incluirán estadísticas de interés como el número de usuarios activos, los mentores disponibles por áreas o el número de relaciones de mentoring abiertas. Estos informes serán accesibles desde la aplicación y resultarán de gran interés tanto para los usuarios como para los administradores.

Y, en último lugar, la aplicación permitirá que los mentores puedan compartir ficheros con los mentorizados y viceversa. Para ello, se creará un repositorio común entre las dos partes donde ambas puedan cargar ficheros.

Medios materiales necesarios.

- **Ordenador personal con acceso a internet:** será el medio donde se desarrolle la aplicación y donde se realice su correspondiente documentación. El software del que deberá disponer es el siguiente:
 - **Java:** lenguaje de programación orientada a objetos que se utilizará para el desarrollo de la aplicación Rest.
 - **Spring Tool Suite:** IDE que usará como entorno de programación para desarrollar la aplicación Rest.
 - **Maven:** framework de Java para la gestión de proyectos que se utilizará para la gestión de las dependencias.
 - **Hibernate:** framework de mapeo objeto/relacional, facilitará la lectura y la escritura de los datos en la base de datos desde la aplicación Rest.
 - **Apache:** servidor web en el que se ejecutará el frontend de la aplicación.
 - **Docker:** tecnología de creación de contenedores que permite la creación y el uso de contenedores tanto en sistemas Unix como en Windows. Permitirá desplegar la aplicación de forma sencilla.
 - **Project Libre:** software de gestión de proyectos de código abierto, se utilizará para generar el plan del proyecto y hacer un seguimiento de este.
 - **Sourcetree:** herramienta GUI que simplifica la interacción con los repositorios de Git.
 - **Visual Studio Code:** IDE de programación que se utilizará para el desarrollo del frontend.
 - **AngularJS:** framework de desarrollo web que se usará para facilitar el desarrollo del frontend.
 - **Bootstrap:** framework de desarrollo web que facilita la creación de interfaces adaptables a diferentes tamaños de pantalla.
 - **PostgreSQL:** gestor de base de datos.

- **Microsoft PowerBI:** herramienta de generación de informes que se usará para generar informes acerca del estado de la aplicación.
 - **StarUML:** herramienta de generación de modelos. Se utilizará para diseñar el backend, tanto para las tablas de la base de datos como para los subsistemas de la aplicación Rest.
 - **Adobe Dreamweaver o similar:** herramienta de diseño web que se utilizará para facilitar la creación de las interfaces de usuario. Dado que esta aplicación es de pago, se usaría la versión de prueba.
 - **Converse.js o similar:** API de mensajería web de código abierto vía XMPP que podría utilizarse para implementar la comunicación entre los usuarios.
 - **Ejabberd o similar:** servidor XMPP de código abierto que daría el servicio de mensajería utilizado por Converse.js
- **Dispositivo móvil:** servirá para probar la adaptación de la interfaz de la aplicación web a los diferentes tamaños de pantalla.
- **Herramientas online que se utilizarán:**
- **Overleaf:** editor de LaTeX online, se utilizará para redactar la documentación del proyecto.
 - **GitHub:** repositorio en la nube de Git, proporcionando un sistema de control de versiones.
 - **Gloomaps:** herramienta online para la creación de EDTs.
 - Herramienta online de diseño de wireframes y mockups.
 - **Spring Initializr:** API web que permite generar de forma rápida un proyecto con Spring Boot y agregar las dependencias básicas necesarias. [12]

Bibliografía.

- [1] Universidad de Santiago de Compostela (2021). Proxecto MentoringUSC, Desenvolvemento profesional - USC. <https://www.usc.gal/gl/proxectos/mentoring> (Visitado 11/02/2022)
- [2] Colexio de A Coruña (s.f.). El colegio pone en marcha un programa de Mentoring para el desarrollo profesional – Economistas Colexio da Coruña. <https://www.economistascoruna.org/es/novedades/actualidad/el-colegio-pone-en-marcha-un-programa-de-mentoring-para-el-desarrollo> (Visitado 11/02/2022)
- [3] Spring (s.f.). Spring Boot - Spring. <https://spring.io/projects/spring-boot> (Visitado 08/02/2022)
- [4] The Apache Maven Foundation (2022-02-08). Wellcome to Apache Maven – Apache Maven Project. <https://maven.apache.org/> (Visitado 08/02/2022)
- [5] Hibernate (s.f.). Hibernate ORM. Your relational data. Objectively - Hibernate. <https://hibernate.org/orm/> (Visitado 08/02/2022)
- [6] Allas Miguelsanz, B. (2017-08-29). Los 10 principios de usabilidad de Jacob Nielsen: be user friendly-Profile. <https://profile.es/blog/los-10-principios-de-usabilidad-web-de-jakob-nielsen/> (Visitado 08/02/2022)
- [7] AngularJS (s.f.). AngularJS. <https://angularjs.org/> (Visitado 08/02/2022)
- [8] Universidad de Alicante (s.f.). Modelo Vista Controlador (MVC) – Universidad de Alicante. <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html> (Visitado 08/02/2022)
- [9] Sourcetree (s.f.). Sourcetree. <https://www.sourcetreeapp.com/> (Visitado 08/02/2022)
- [10] Converse (s.f.). Converse.js messaging freedom – Converse. <https://conversejs.org/> (Visitado 17/02/2022)
- [11] Ejabberd (s.f.). Ejabberd. <https://www.ejabberd.im/> (Visitado 17/02/2022)
- [12] Spring (s.f.). Spring Initializr. <https://start.spring.io/> (Visitado 18/02/2022)

Observacións.

FASES DO TRABALLO E ESTIMACIÓN TEMPORAL

Un traballo de fin de grao suporá 401,25 horas de traballo autónomo do alumnado e 11,25 horas de traballo presencial (titorías e avaliación).

Dedicación semanal prevista (en horas/semana): 30

Fase	Estimación temporal (en semanas)
Gestión Inicial	3.25
Realización de la aplicación base	4
Agregar de la herramienta del chat	2.5
Agregar intercambio de ficheros	1.5
Agregar generación de informes	1.5
Documentación e integración	1
Total	13.75 (412.5 horas)

(Para a xestión do alcance do proxecto deberase incluír en tódolos casos unha EDT segundo se recolle no PMBOK do PMI. Capítulo 5, Xestión do Alcance do Proxecto)

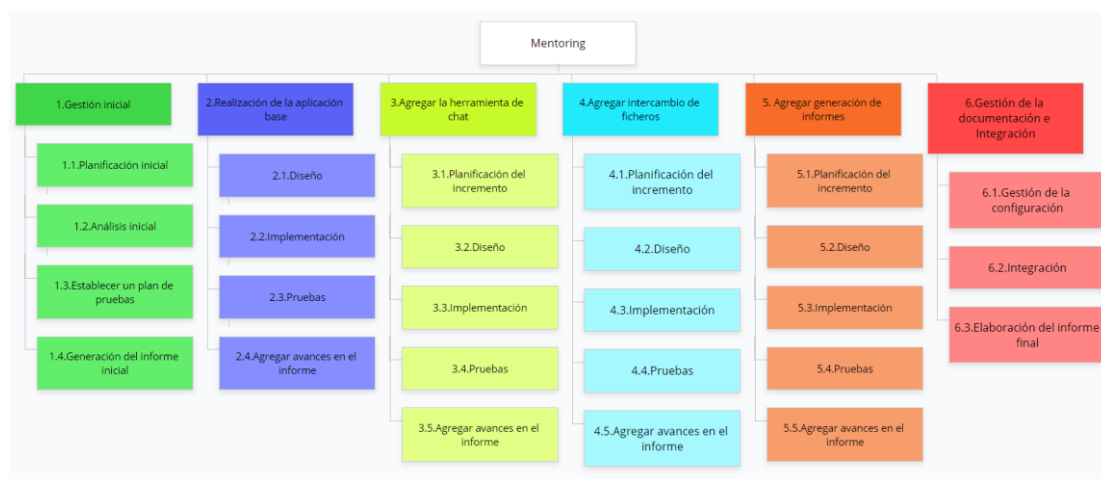


Ilustración 1. EDT del proyecto. Creación propia, utilizando la herramienta online Gloomaps (<https://www.gloomaps.com/>) . Fecha de la última modificación: 25/02/2022.

Diccionario del EDT

Ahora se procede a describir cada una de las tareas resumen (paquetes de primer nivel) contenidas en el EDT, esto ayudará a entender qué se debe realizar con exactitud en cada una de ellas. Estas tareas coinciden con las fases por las que pasará el proyecto.

1. **Gestión inicial.** Es la primera fase del proyecto y la más importante, en la que se realizan las siguientes tareas: en primer lugar, una planificación inicial del proyecto, en la que se define una estimación inicial de tiempos y de recursos necesarios. Esta tarea implicará la realización de un diagrama de Gantt partiendo del EDT de la Ilustración anterior. Una vez establecidos los recursos, se determinará y se adquirirá la formación necesaria, también se preparará el entorno de trabajo con las herramientas a utilizar.

En segundo lugar, el análisis. Para ello se establecerá una visión general de los objetivos a alcanzar. Esta visión nos servirá de base para realizar el análisis de requisitos, tanto de los funcionales como los no funcionales. Se repartirán los requisitos entre los incrementos y se identificará cuál es su prioridad dentro del incremento. Por otro lado, también se identificarán requisitos que, si bien caerían fuera del proyecto, podrían ser abarcables en caso de una sobrestimación del tiempo en la

planificación, o quedar registrados como futuras mejoras. Los requisitos no serán estáticos, podrán ser adaptados o modificados durante los incrementos para ajustarse al alcance del proyecto. Una vez realizado el análisis de requisitos, se llevará a cabo un análisis de los posibles riesgos del proyecto y de los costes del mismo.

En tercer lugar, se establecerá un plan de pruebas. Esta tarea nos permitirá fijar cómo se llevarán a cabo las pruebas. Finalmente, se creará un informe inicial del proyecto con el trabajo realizado hasta el momento. Este documento se irá ampliando a medida que avance el proyecto y permitirá reflejar en el mismo el avance del proyecto de una forma más fiable que se realizase al final del mismo.

2. **Realización de la aplicación base.** En esta fase se desarrollará el primer y mayor incremento. La aplicación base será sobre la que se desarrollen los demás incrementos, y contendrá las funcionalidades necesarias para que los usuarios puedan registrarse, realizar búsquedas y visualizar los perfiles de los mentores disponibles. Esta fase estará compuesta por una serie de tareas, en primer lugar, el diseño. Esta tarea incluye el diseño inicial de la base de datos, así como un diseño inicial de los módulos del backend y de la interfaz del frontend. También incluye el diseño de las pruebas que se realizarán.
En segundo lugar, la implementación, en la que se codificará la primera versión de cada una de las partes que forman la aplicación (base de datos, backend y frontend), ajustándose al diseño realizado. En tercer lugar, se someterá a la aplicación resultante a un proceso de pruebas. Este proceso incluye pruebas unitarias, de cumplimiento de requisitos y de usabilidad. Y, por último, los avances realizados y los resultados obtenidos en este incremento se volcarán en el informe del proyecto.
3. **Agregar la herramienta de chat.** En este incremento se añadirá un chat que permitirá a los mentorizados contactar con los mentores a través de la propia aplicación. Al comienzo se realizaría una planificación del incremento, donde tendría lugar una revisión del avance del mismo. Esta tarea permitirá adaptar el alcance del proyecto al avance de este. También se llevará a cabo un diseño de los cambios a realizar sobre el incremento anterior y de los nuevos elementos a incorporar, así como la implementación de los mismos. Tras esto, se realizarán pruebas sobre las nuevas funcionalidades, también se adaptarán las pruebas del incremento anterior para volver a pasarlas y comprobar que los cambios no hayan afectado a las funcionalidades implementadas en ese incremento. Finalmente, se añadirán los avances y resultados al informe.
4. **Agregar intercambio de ficheros.** En este incremento se añadirá la posibilidad de que los usuarios puedan colgar ficheros en su perfil. También se permitirá a los mentores enviar ficheros a los mentorizados con los que hayan establecido un contacto, y viceversa.
5. **Agregar generación de informes.** En este incremento se añadirá la capacidad de generar informes a partir de los datos de la aplicación contenidos en la base de datos. Dichos informes estarán disponibles para los usuarios en la aplicación, y les permitirán conocer el estado y evolución de la misma.
6. **Gestión de la documentación e integración.** Esta última fase del proyecto consta de una revisión final de los resultados y de una integración, en la que se preparará la aplicación para ser exportada a otros sistemas. También incluye la elaboración del informe final a partir de la documentación existente. Si bien la gestión de la configuración está incluida en esta fase, esta tarea se realizará a lo largo de todo el proyecto, permitiendo mantener la consistencia del mismo y garantizar su calidad.