

INGENIERÍA WEB  
PROYECTO DE DESARROLLO  
*Ingeniería Informática*  
*Universidad de Cádiz*  
Curso 2024-25

Profesorado de Ingeniería Web

10 de octubre de 2024

## Índice

|  |          |
|--|----------|
| <b>1. Introducción</b>                   | <b>2</b> |
| <b>2. Método de desarrollo</b>           | <b>2</b> |
| 2.1. Actividades . . . . .               | 2        |
| 2.1.1. Inicio del proyecto . . . . .     | 2        |
| 2.1.2. Desarrollo del proyecto . . . . . | 3        |
| 2.1.3. Cierre del proyecto . . . . .     | 3        |
| 2.2. Roles . . . . .                     | 3        |
| 2.3. Herramientas . . . . .              | 4        |
| <b>3. Entregables</b>                    | <b>5</b> |
| <b>4. Entrega y evaluación</b>           | <b>5</b> |
| 4.1. Hitos de entrega . . . . .          | 5        |
| 4.2. Evaluación . . . . .                | 6        |

## 1. Introducción

El proyecto consiste en el desarrollo de una aplicación web usando los métodos, técnicas y herramientas de la Ingeniería Web.

En el proyecto se pondrán en práctica las técnicas y métodos introducidos durante las clases teóricas de la asignatura. Asimismo, es fundamental el aprendizaje de los lenguajes, frameworks y herramientas necesarias para construir el proyecto, por lo que tal aprendizaje deberá realizarse como investigación individual de cada estudiante.

El trabajo estará organizado en equipos de tres personas, donde cada una de ellas debe desempeñar varios roles. El trabajo se organizará en iteraciones sucesivas. Excepcionalmente, en el caso de grupos con un número de estudiantes diferente, se ajustará la carga de trabajo exigida por el profesor.

## 2. Método de desarrollo

El desarrollo del proyecto se realizará con el [método SCRUM](#), un método ágil de desarrollo, que estará dividido en varias *iteraciones* semanales. En cada iteración se pretende alcanzar un objetivo mediante la realización de unas *actividades* encaminadas a aprender una serie de *técnicas* y empleando unas *herramientas* de ayuda al desarrollo. En estas actividades participarán una serie de *roles* y se generarán unos *entregables* en forma de software y de documentos.

### 2.1. Actividades

Los trabajos comenzarán con una fase de inicio en donde se planificará todo lo referente al proyecto, así como se prepararán todos los entornos necesarios. Luego comenzará el desarrollo en sí en donde todos los roles establecidos desempeñarán las labores propias de la Ingeniería Web en cada una de las iteraciones del proyecto. De manera simultánea, se realizarán diversas actividades de gestión y evaluación del proyecto conforme el software va evolucionando. Finalmente en la fase de cierre, se acabarán las actividades necesarias y se procederá a la evaluación final.

#### 2.1.1. Inicio del proyecto

Durante la primera fase del proyecto se requiere llevar a cabo una serie de actividades de planificación y configuración:

- Formar los equipos de trabajo
- Preparar el entorno de trabajo en local y en la nube: sistema de control de versiones de código, seguimiento del proyecto, repositorio documental, comunicaciones, etc.
- Desarrollar el plan de proyecto, asignando los roles asignados a cada miembro del equipo, definiendo el roadmap del proyecto, la periodicidad de reuniones, los acuerdos relativos a la gestión del trabajo en grupo y a los

posibles conflictos y seleccionando las herramientas a utilizar durante el proyecto.

### 2.1.2. Desarrollo del proyecto

Durante esta fase del proyecto se realizarán una serie de actividades tanto de desarrollo como de gestión y evaluación, enmarcadas dentro de iteraciones o incrementos. Son las siguientes:

- Planificar las características a incluir en la nueva iteración (sprint backlog), mediante tareas en la herramienta de seguimiento
- Analizar los requisitos necesarios
- Diseñar, implementar y probar la funcionalidad requerida
- Actualizar, en su caso, la documentación
- Opcionalmente, realizar la auto-evaluación de lo realizado en el proyecto durante el sprint anterior, incluyendo las evidencias que argumentan cada una de las evaluaciones otorgadas.

Por otro lado, toda la información relativa a las tareas involucradas en el desarrollo deben quedar registradas y convenientemente actualizadas en la forja, para que tanto los estudiantes como los profesores puedan observar el estado real del proyecto.

### 2.1.3. Cierre del proyecto

Durante la fase de cierre del proyecto, los integrantes del mismo deben completar los entregables documentales y el código fuente. El día de la defensa, el profesor accederá a algún servidor con la aplicación desplegada para realizar las pruebas (manuales y automáticas) pertinentes. Los estudiantes realizarán una presentación del proyecto desarrollado. Para la evaluación del proyecto, el profesor hará uso de una rúbrica de evaluación. Los estudiantes deben enviar a través del campus virtual el documento de declaración de autoría relleno y firmado.

## 2.2. Roles

Los participantes asumirán una serie de roles típicos en cualquier proceso de desarrollo de software y en SCRUM, pudiendo una misma persona desempeñar varios roles, y un rol ser desempeñado por varias personas. Los roles posibles a desempeñar por los miembros de un equipo de proyecto son los siguientes:

- *Scrum Master*: hace cumplir las reglas y buenas prácticas, elimina impedimentos
- *Product Owner*: representante del cliente, prioriza historias de usuario

- *Arquitecto software*: diseña la arquitectura de los componentes y de todos los sistemas software que intervengan en el despliegue de la aplicación para ponerla en funcionamiento.
- *Analista*: analiza requisitos; realiza modelos de casos de uso y conceptuales, etc.
- *Diseñador de interfaz de usuario*: realiza mockups de pantallas y diseña y codifica la interfaz de usuario.
- *Programador*: diseña y codifica los componentes de software en Java.
- *Tester*: planifica, codifica y configura la ejecución de casos de prueba unitarias, funcionales y de integración.
- *Administrador de sistemas*: instala y configura los entornos de desarrollo, servidores y sistemas software de los que depende la aplicación (base de datos, servidor web, servidor de aplicaciones, sistema de control de versiones, etc.)

## 2.3. Herramientas

### Herramientas de gestión y documentación

Para la gestión del proyecto se empleará una *forja* alojada en un espacio gratuito en *GitHub*. Los estudiantes podrán registrarse como tales en la plataforma de Github para obtener acceso gratuito a las características de la misma ofrecida en sus planes de pago. El espacio deberá denominarse *iw2024-2025-NOMBRE-DEL-PROYECTO* y se deberá otorgar permisos de acceso (colaborador) al usuario *ruizrube*.

Luego, en la página principal (README.md) del proyecto se debe añadir información de bienvenida con los nombres de los miembros del equipo de trabajo y un enlace a una copia en Google Sheets de la rúbrica de auto-evaluación.

En Github se registrarán las historias de usuario/características e incidencias detectadas como *issues* y los sprints/iteraciones como *milestones*. Se usarán las *releases* para colgar los ficheros empaquetados del software (ficheros jar).

Se utilizará el sistema de wikis de Github como repositorio de documentación de los proyectos, no siendo válido ningún otro formato (pdf, word, etc.)

### Herramientas de comunicación

Además del correo electrónico, se recomienda hacer uso de alguna plataforma de mensajería para hacer más fluida la comunicación entre los miembros del equipo de trabajo.

### Herramientas de diseño y modelado

A lo largo del proyecto será necesario usar herramientas para realizar diagramas de análisis, mockups, diseños, etc. Es potestativo de cada equipo de desarrollo el empleo de las herramientas con las que se encuentre más cómodo, aunque

se requiere que sean de software libre o con licencia de uso no restringida para educación.

### Herramienta de desarrollo

Como entorno gráfico de desarrollo se aconseja usar una versión reciente de *Eclipse* o *IntelliJ*, aunque se puede utilizar cualquier otro entorno de programación. Se requiere el uso de *Apache Maven* como herramienta de gestión del proyecto, construcción, gestión de dependencias, documentación, pruebas y despliegue de la aplicación.

## 3. Entregables

La siguiente es una lista de los entregables a depositar en la plataforma Github.

**E1** Fichero jar con el "binario" de la aplicación

**E2** Código fuente

**E3** Documentación en formato *documents* wiki

- a) *Plan de proyecto*
- b) *Análisis de requisitos*
- c) *Diseño y construcción*
- d) *Pruebas*
- e) *Despliegue y operaciones*

## 4. Entrega y evaluación

### 4.1. Hitos de entrega

Cada una de las iteraciones o *sprint* deben durar entre dos y tres semanas (según se planifique en las etapas iniciales del proyecto) y finalizará con la entrega de una versión del software y los documentos *entregables*, a depositar en la herramienta colaborativa. Cada entrega parcial debe ser marcada con un hito (*milestone*) en la planificación que se haga del proyecto en la forja.

En la tabla 1 se presentan los hitos previstos de entrega. Los hitos 1 a N deberán fijarse por cada equipo en función de la estimación que se haga en la planificación inicial, y adecuarse conforme avance más o menos la ejecución del proyecto.

| Hito       | Descripción                                | Fecha                          |
|------------|--|--------------------------------|
| Hito 0     | Entrega de Plan de proyecto                | 17 de octubre                  |
| Hito 1     | <i>A definir por el equipo</i>             | <i>A definir por el equipo</i> |
| Hito N     | <i>A definir por el equipo</i>             | <i>A definir por el equipo</i> |
| Hito Final | Entrega del proyecto y defensa obligatoria | 9 de enero                     |
| Examen     | <i>Recuperación opcional</i>               | ???                            |

Tabla 1: Hitos del proyecto

## 4.2. Evaluación

La evaluación del proyecto se realizará atendiendo a una serie de criterios que vendrán recogidos en una *rúbrica* o instrumento de evaluación, disponible en el campus virtual. De este modo, se buscarán evidencias del cumplimiento de los criterios descritos anteriormente en la forja del proyecto. Durante el proceso de evaluación, deberán estar presentes todos los miembros del equipo de trabajo, los cuales tendrán que responder a una serie de cuestiones sobre su implicación en el proyecto.

Es importante que haya un equilibrio en la distribución del trabajo, tanto en la asignación de roles, como en el reparto efectivo de tareas, el total de commits en el sistema de control de versiones y el volumen de ediciones sobre los documentos. De no ser así, las calificaciones serán ajustadas por exceso o por defecto a cada integrante del equipo. Se podrán tener en cuenta para la calificación individual, los porcentajes de distribución de la carga de trabajo indicados en el documento de declaración de autoría.

La copia total o parcial de código fuente o de documentación procedente de proyectos realizados por otros compañeros será motivo suficiente para la no superación de la parte práctica de la asignatura. Se harán uso de herramientas de análisis para la detección automática de plagio.

El servidor donde se esté ejecutando el software deberá estar arrancado y preparado para el momento de la defensa. Al entrar en el aula se facilitará al profesor la URL de la aplicación para poder realizar las pruebas pertinentes. Se indicará la IP privada asignada al equipo, en el caso de usar un equipo portátil; o un nombre de dominio, en el caso de un servidor público. En el primer caso, comprobad que no haya ningún cortafuegos que impida acceder a la aplicación web. Asimismo, se deberá indicar las credenciales de acceso de un usuario con privilegios de administración.

La base de datos debe estar suficientemente poblada con los datos necesarios para probar de manera satisfactoria las distintas características implementadas en la aplicación web.