



Facultad de  
**Ciencias Sociales**

Talavera de la Reina. UCLM

CAMPUS DE EXCELENCIA INTERNACIONAL

**INGENIERÍA DE SOFTWARE-II**

**MEMORIA DEL PROYECTO**

***THE PIQUE BLINDERS***

**CURSO 2022-2023**

## Índice

Miembros del equipo .....	2
Características del proyecto .....	2
Producto solicitado .....	2
Objetivos del proyecto .....	2
Estructura de éste .....	3
Cronograma de actividades.....	3
Entregables.....	3
Pasos del proyecto .....	4
Planificación .....	4
Análisis.....	4
Diseño.....	4
Implementación .....	4
Pruebas.....	5
Mantenimiento .....	5
Conclusiones .....	5

## Miembros del equipo

El equipo de trabajo de la empresa está formado por los siguientes miembros:

- ❖ Bermejo, Estrella
- ❖ Jiménez Batuecas, Alejandro Ramiro
- ❖ Mena Castellanos, Pedro
- ❖ Saiz García, Alejandro

## Características del proyecto

A continuación, se detallan las características del proyecto llevado a cabo por el conjunto de los miembros del equipo.

### Producto solicitado

Realización de un proyecto de desarrollo software siguiendo los métodos y técnicas de ingeniería del software vistas en teoría, así como aplicando las herramientas presentadas en las diferentes sesiones de prácticas. El proyecto se basa en la construcción de un sistema informático para la gestión de enseñanzas propias de la Universidad de Castilla-La Mancha.

### Objetivos del proyecto

Antes de abordar este proyecto, se habían establecido los cinco objetivos que se describen en los puntos siguientes:

1. Desarrollar un entorno de trabajo adecuado, independiente de la metodología de desarrollo.
  - ❖ Se han utilizado herramientas colaborativas como Office 365 para dar soporte a la comunicación, el control y la colaboración entre los miembros del equipo.
  - ❖ Para el desarrollo del código, se ha utilizado el lenguaje Java, por lo que se han revisado los principios básicos de la Programación Orientada a Objetos y aspectos más avanzados, como patrones de diseño.
2. Poner en valor la utilización de una metodología de desarrollo software iterativa e incremental, usando el concepto de versión y de gestión de configuración para planificar el desarrollo del software y monitorizar su avance.
  - ❖ Se ha usado GitHub como repositorio del proyecto. Esta herramienta *Source Code Management (SCM)* nos facilita la administración de las distintas versiones y modificaciones del código del programa y sus ficheros de configuración, registrando todos los cambios del repositorio para poder consultarlos posteriormente e incluso revertirlos, sin tener que hacer copias de seguridad.
  - ❖ También se ha empleado la herramienta de construcción de software Maven.
3. Comprender el concepto de calidad del producto de software.
  - ❖ Dentro de calidad, se han identificado los requisitos, tanto funcionales como no funcionales, derivados del *quality by design* y se han incorporado al diseño.
  - ❖ Además, se han seguido y resuelto problemas relacionados con la calidad del software.
4. Probar (verificar y validar) el sistema desarrollado para asegurar que su funcionamiento es correcto y conforme a lo que el cliente solicitó.
  - ❖ Se han implementado los casos de prueba usando Junit.

- ❖ Se ha integrado la automatización de la verificación de las pruebas en el proceso de construcción del software y la generación automática de informes, mediante la herramienta Maven.
- ❖ Se ha tenido en cuenta el plan de gestión de configuración.
- 5. Gestionar los planes de mantenimiento del sistema desarrollado.
  - ❖ Se ha utilizado la herramienta SonarCloud para abordar este objetivo.

### Estructura de éste

Para desarrollar este proyecto de software se ha utilizado SCRUM, una metodología de desarrollo ágil basada en un proceso iterativo e incremental. Se trata de un marco ágil adaptable, rápido, flexible y eficaz que está diseñado para ofrecer valor al cliente durante todo el desarrollo del proyecto. El objetivo principal de SCRUM es satisfacer la necesidad del cliente a través de la transparencia en la comunicación, responsabilidad colectiva y progreso continuo.

### Cronograma de actividades

En una metodología de SCRUM contamos con los siguientes eventos:

- ❖ *Sprint*: Es la unidad básica de trabajo de un equipo. Esta es la principal característica que marca la diferencia entre SCRUM y otros modelos de desarrollo ágil.
- ❖ Planificación del Sprint: El objetivo es definir qué realizar en el siguiente Sprint y cómo efectuarlo. Esta reunión se realiza al inicio de cada Sprint y se define cómo enfocar el proyecto a partir de las etapas y plazos del *Product Backlog*.
- ❖ Scrum semanal: El objetivo es evaluar el progreso y la tendencia hasta el final del Sprint, sincronizando las actividades y creando un plan para los próximos 7 días.
- ❖ Revisión del sprint: El objetivo es mostrar qué trabajo se ha completado respecto al backlog del producto para futuras entregas. Se revisa el sprint terminado.

Para desarrollar la planificación y gestión de los diferentes *sprints*, se ha utilizado la plataforma ClickUp. Se han subido al repositorio las capturas de cada uno de los *sprints* desde la entrega intermedia. Dentro de la carpeta "*sprints*", encontraremos dos carpetas, en "*sprintDocuments*" tenemos los documentos con la información de cada Sprint, mientras que en "*sprintImages*", tenemos las imágenes correspondientes a cada Sprint realizado.

### Entregables

Se han establecido dos fechas de entrega: una a mitad de proyecto y otra al final. El repositorio del proyecto es accesible en la siguiente dirección: [Repositorio del proyecto The Pique Blinders](#)

A mitad de proyecto, se suben el código implementado hasta la fecha, la planificación y gestión que se ha llevado a cabo y el plan de gestión de configuración, además de una pequeña memoria donde se detalla lo entregado al repositorio.

Al finalizar el proyecto, se entrega la presente memoria se detallan las características, los pasos y el resultado del proyecto, junto con la totalidad del código al repositorio.

## Pasos del proyecto

Para realizar este proyecto, hemos considerado las diversas fases del proceso de desarrollo de software, que son planificación, análisis, diseño, implementación, pruebas, despliegue y mantenimiento. A continuación, se detallan las tareas que se han puesto en práctica en cada una de estas fases.

### Planificación

Como anteriormente se comentaba en el cronograma de actividades, al escoger SCRUM, una metodología ágil basada en un proceso iterativo e incremental, la planificación se ha establecido por *sprints* semanales en la plataforma ClickUp.

Para acceder a la carpeta “*sprints*”, haga *click* en el siguiente enlace: [Carpeta sprints](#)

### Análisis

En esta etapa en el ciclo de vida del software, se realizó un análisis de requisitos del sistema a desarrollar y de los requisitos establecidos por el cliente.

### Diseño

En referencia al diseño, se ha utilizado el lenguaje de Java sobre la plataforma de software Eclipse. Se nos proporcionó desde el Campus Virtual de la asignatura parte del análisis y del diseño del sistema, por lo que el equipo se ha ocupado principalmente de los pasos siguientes, que son la codificación, la gestión de configuración, la generación de calidad, la gestión de *testing* y la gestión de mantenimiento.

### Implementación

Como repositorio del proyecto, se ha utilizado la herramienta de SCM -gestión del código fuente-GitHub. La creación y gestión de ramas (*branches*) se ha cumplido siguiendo las siguientes directrices:

1. La rama desarrollo *develop* es creada de la rama principal *master*.
2. La rama lanzamiento *release* es creada a partir de la rama desarrollo *develop*.
3. Desde desarrollo *develop*, también creamos las ramas *feature*, que hacen referencia a las distintas características.
4. Cuando una característica *feature* se ha completado, unimos la rama *feature* a la rama desarrollo *develop*.
5. Cuando la rama de lanzamiento *release* está hecha, se une a la de desarrollo *develop* y a la principal *master*.
6. Si surge un error o problema, lo modificamos en la rama en la que se está trabajando.

Para la implementación del sistema, se ha creado una base de datos mediante el uso de Derby. Los métodos que podemos encontrar en la clase GestorBD son:

- ❖ ConectarBD: Método que crea la base de datos, en caso de que no esté creada todavía, y obtiene la conexión a ésta.
- ❖ DesconectarBD: Método que cierra la conexión con la base de datos.
- ❖ Select: Método que selecciona uno o varios elementos de la base de datos.
- ❖ Insert: Método que introduce determinados valores en la base de datos.
- ❖ Update: Método que actualiza un valor o valores en la base de datos.
- ❖ Delete: Método que borra un valor o valores en la base de datos.

- ❖ **CrearBaseDatosSiNoExiste:** Método que crea una base de datos e introduce en ella ciertos valores en varias columnas de la base de datos.

Para la gestión de configuración, se creó un plan. El documento es accesible a través de la carpeta “planificaciones” del repositorio.

Se ha realizado la construcción del software utilizando Maven, cuyo archivo .jar está ubicado en la carpeta target, que puede encontrarse también en el repositorio del proyecto.

Se ha abordado la generación de calidad junto con la gestión de pruebas y mantenimiento.

### Pruebas

Para la gestión de testing, se ha diseñado un plan de prueba y de los casos de pruebas, cuyo documento se encuentra en la carpeta “planificaciones” dentro del repositorio del proyecto. Estos casos de pruebas se han implementado mediante el uso de Junit. Además, se ha automatizado la generación de informes con el uso de Maven.

Para acceder al Plan de prueba y casos de pruebas, [haz click sobre este link](#).

### Mantenimiento

En cuanto al mantenimiento, se ha elaborado un documento llamado “Plan de Mantenimiento SonarCloud”, que puede encontrarse en la carpeta de planificaciones en GitHub. Este plan se ha seguido para el mantenimiento del sistema a través del programa SonarCloud.

Para acceder al Plan de mantenimiento puedes hacerlo siguiendo [este enlace](#).

### Conclusiones

Una vez finalizado el proyecto, vemos que se han cumplido los objetivos del proyecto: Implementar una metodología ágil de modelo scrum, llevar a cabo un plan de mantenimiento y de prueba para mejorar la calidad del software, hacer un correcto uso de Maven junto a sus dependencias, etc.

Entre los conocimientos adquiridos, destacan:

- ❖ Conocer el alcance global de un producto software.
- ❖ Enfocar un proyecto por etapas y objetivos.
- ❖ Comprender el funcionamiento de GitHub y cómo se aplica al ámbito de la empresa.
- ❖ Distribuir las diversas áreas de trabajo entre los diferentes roles de la organización.
- ❖ Trabajar en equipo.