



**Universidad  
Europea** MADRID

# PROYECTO INTEGRADOR

1ºDAW

MANCHESTER PITIS



AUTORES

ADRIÁN CORRALIZA CHUECA

MARCOS GUTIÉRREZ

ÁLVARO LABRADOR ESCUDERO

HUGO TURRILLO BARRERA



## Índice

Resumen.....	4
1. Introducción .....	5
2. Objetivos .....	6
3. Tecnologías utilizadas.....	8
4. Desarrollo e implementación .....	10
5. Metodología .....	13
6. Resultados y conclusiones.....	15
7. Trabajos futuros .....	16
Anexos .....	17
Anexo I – Listado de requisitos de la aplicación .....	17
Anexo II – Guía de uso de la aplicación .....	19
Anexo III.....	19



## Resumen

Extensión del resumen de 200 a 300 palabras. Debe escribirse al finalizar el Proyecto Integrador y se redacta en un único párrafo.

**Palabras clave:** palabra 1; palabra 2; palabra 3; palabra 4; palabra 5.

## 1. Introducción

Este proyecto tiene como finalidad la creación de una aplicación de gestión para un Polideportivo, permitiendo la administración eficiente de salas, actividades deportivas y usuarios, en esta ocasión acerca de la Universidad Europea. La aplicación ha sido desarrollada utilizando el lenguaje de programación Java, en el entorno de desarrollo Eclipse, con una base de datos MySQL para el almacenamiento y gestión de la información.

El sistema está orientado principalmente a facilitar la inscripción de alumnos en actividades deportivas, asegurar la correcta asignación de salas y llevar el control de ocupación y el uso de sus espacios. Se ha buscado mantener una estructura clara, aplicando buenas prácticas de programación orientada a objetos y favoreciendo la escalabilidad del proyecto. Como parte del diseño visual, se ha creado un logotipo personalizado utilizando inteligencia artificial (OpenArt), con un prompt específico que hacía referencia al nombre del equipo "Manchester Pitis", tomando inspiración del escudo original del Manchester United.

El desarrollo de este se ha centrado en lograr una interfaz amigable y funcional que facilite el acceso de los usuarios a las funcionalidades principales del sistema. Las clases correspondientes a la vista se han diseñado para reflejar fielmente las operaciones disponibles, promoviendo la interacción intuitiva y una experiencia de usuario positiva.

En cuanto a la estructura técnica, se ha implementado un modelo relacional en MySQL, en el que se han definido las tablas necesarias para almacenar los datos de usuarios, actividades, salas y sus respectivas inscripciones. Además, este proyecto ha implicado el uso de herramientas modernas de gestión y colaboración, como Trello y GitHub, permitiendo una adecuada coordinación del equipo. Se ha aplicado una metodología ágil, en particular Scrum, para organizar y distribuir las tareas en sprints, lo que ha contribuido a un desarrollo progresivo y estructurado. Esta forma de trabajo ha facilitado tanto el cumplimiento de objetivos como la adaptación ante posibles incidencias o cambios en los requisitos.

## 2. Objetivos

A continuación, hablaremos acerca de los objetivos que buscamos para poder realizar y completar este trabajo.

Habilidades como, una buena planificación del tiempo, donde una gestión y realización efectiva del trabajo propuesto dentro de los tiempos límites dados para el comienzo y finalización para cada uno de los Sprints en los diferentes módulos como serían Programación, Bases de datos y Entornos de Desarrollo.

El trabajo se realiza de manera conjunta con los integrantes del grupo donde debe haber un respeto y una adecuada autonomía y organización dentro de este para llevar a cabo el trabajo.

En caso de que ocurran o aparezcan conflictos entre integrantes del grupo, se deberán de resolver mediando y contribuyendo a la solución de estos para así fomentar un ambiente de trabajo agradable y siempre actuando de manera respetuosa y tolerante con los compañeros.

Cada uno de los integrantes de forma individual, debe ser capaz de tomar decisiones y resolver los problemas que se le propongan, siguiendo las normas y procedimientos establecidos anteriormente, estos definidos por supuesto en el ámbito de la competencia.

Como objetivos más específicos y centrados en cada uno de los Módulos que constituyen este proyecto, de los cuales ya hemos mencionado que serían Programación, Entornos y Bases de Datos. Nos centraremos en que tratamos en cada uno de estos.

De Bases de datos, aprenderemos a crear una base de datos que se verá definida tanto su estructura y características de los elementos que contenga según el modelo relacional. Tendremos que diseñar modelos lógicos normalizados interpretando el Modelo Entidad/Relación, que deberemos definir con anterioridad. Con el uso de asistentes, herramientas Gráficas y el lenguaje de definición de datos, realizaremos el diseño físico de la base de datos. Por último, con los asistentes, herramientas... que acabamos de mencionar deberemos consultar y modificar la información almacenada.

Acerca de Programación, los objetivo serían, la realización de clases necesarias para así ser capaces de seguir el patrón MVC, posteriormente, la implementación de clases de interfaz gráfica, clases que representen el modelo, o clases encargadas de la lógica de la aplicación

pertenecientes al control. Desarrollaremos una aplicación que gestione información almacenada en la base de datos, que ya hemos comentado que deberemos de realizar, estos datos relacionales de la base de datos se verán gestionados identificando y utilizando mecanismos de conexión. Una vez finalicemos esta tarea, habremos completado la lección y a su vez alcanzado la competencia completa para desarrollar componentes software en lenguajes de programación orientada a objetos.

Por último, Entornos de Desarrollo, pondremos en práctica nuestros conocimientos y lecciones vistas a lo largo del curso, de manera que se deberá realizar un análisis y diseño de cualquier aplicación empleando técnicas UML, documentar las aplicaciones empleadas para el proyecto, gestionaremos las diferentes versiones de un software y el trabajo colaborativo, se deberá realizar también pruebas de testeo sobre cada uno de los programas que realicemos y por supuesto documentarlos. Por último, seguir las metodologías ágiles para el desarrollo y planificación de los programas.



### 3. Tecnologías utilizadas

Herramientas de Desarrollo:

- Eclipse: En el desarrollo de la aplicación se ha utilizado el lenguaje de programación Java, dentro del entorno de desarrollo integrado Eclipse, por ser una combinación ampliamente utilizada en proyectos educativos y profesionales. Java ha sido fundamental para construir la lógica de la aplicación, aprovechando sus características de programación orientada a objetos y su capacidad para trabajar con interfaces gráficas, conexiones a bases de datos y estructuras complejas.

Gracias a Eclipse, se ha podido trabajar de forma eficiente en la creación de las distintas capas de la aplicación siguiendo el patrón MVC (Modelo-Vista-Controlador). Este entorno ha ofrecido herramientas para la gestión del código, integración con GitHub y la posibilidad de realizar pruebas y depuración de manera ordenada y visual. Java se ha utilizado específicamente para programar la interfaz gráfica (usando Swing), gestionar la conexión con la base de datos mediante JDBC, y controlar toda la lógica necesaria para realizar las operaciones de gestión de usuarios, salas y actividades. Eclipse ha facilitado un entorno centralizado para organizar, editar y probar todo el desarrollo.

- Créate – Open Art: Durante el desarrollo del proyecto, también se hizo uso de la inteligencia artificial generativa, en concreto de la herramienta **OpenArt**, para diseñar el logotipo representativo del equipo. OpenArt permite crear imágenes a partir de descripciones detalladas, conocidas como *prompts*, que el usuario proporciona. En este caso, se utilizó un *prompt* creativo en el que se especificaba que el diseño debía estar inspirado en el escudo del **Manchester United**, ya que el equipo había adoptado el nombre ficticio de **Manchester Pitis**. Se indicó que el logotipo debía mantener una estética similar, con un estilo heráldico, un fondo circular en tonos rojos, y un emblema central potente que transmitiera fuerza y unión.

Gracias a esta herramienta, se obtuvo un diseño visual atractivo y totalmente personalizado, que ayudó a reforzar la identidad del equipo dentro del proyecto.

### Herramientas Organizativas:

- GitHub: Aplicación/Instrumento que nos ayuda de una manera más organizada, por ejemplo, su seguimiento activo sobre cada uno de los cambios que se realizan en el repositorio en el que trabajamos, y su control de versiones. Además de su gran compatibilidad a la hora del trabajo grupal a distancia, como es su capacidad de almacenar o compartir los archivos subidos en un equipo de los integrantes del grupo y la facilidad del resto de estos, de poder actualizar su versión de datos, con la actualización de la versión nueva, ya sea de cambios o datos añadidos o borrados.

GitHub: [https://github.com/alvaritto7/PROYECTO\\_INTEGRADOR/tree/main](https://github.com/alvaritto7/PROYECTO_INTEGRADOR/tree/main)

- D iscord: En base a la posibilidad de cada uno de los integrantes a la hora de trabajar, y teniendo en cuenta cada uno de nuestros horarios se dificultaba la formalidad presencial de los 4 integrantes del grupo, a la hora de continuar progresando con el transcurso de trabajo, optando as , por la herramienta comunicativa que ofrece **D iscord**, disponiendo as  de la posibilidad de llamadas online y la conocida formalidad video presencial a la hora de trabajar.
- Trello: Se ha aprovechado en gran medida su funci n principal, la cuales son la visualizaci n y administraci n del proyecto, sirviendo as , que todos los integrantes del grupo puedan llevar un seguimiento sin problemas del estado, progreso y las propiedades de las tareas del proyecto en tiempo real. Sirviendo as  tambien para la asignaci n de tareas a cada integrante, gracias a la posibilidad de creaci n e implementaci n de etiquetas a los anuncios de las tareas.

Trello: <https://trello.com/b/g1sjZFfA/machester-pitis>

## 4. Desarrollo e implementación

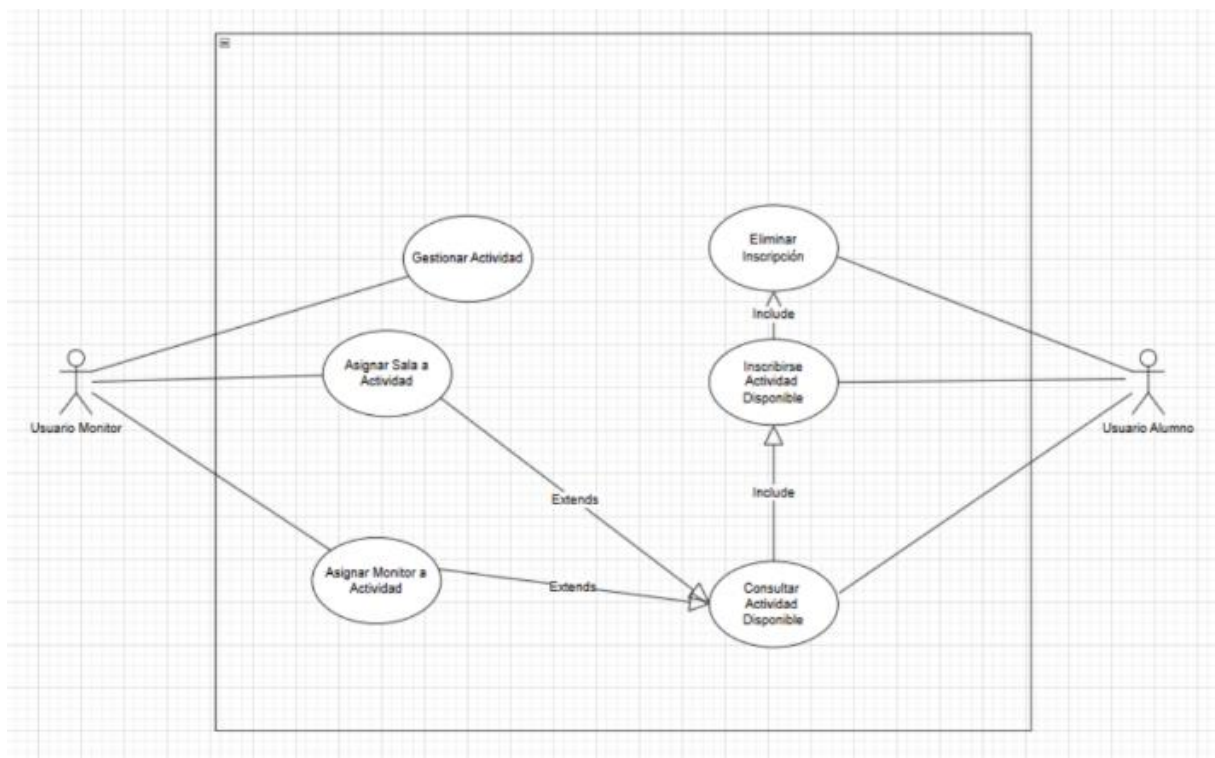
Para este apartado se recomienda una extensión de 7 a 15 páginas. Comenzar con los diagramas (diagrama de caso de uso, diagrama de clases, diagrama relacional y MER). Detalles sobre cómo está construida la aplicación, funcionamiento, relación de la aplicación con la base de datos, diseño del logo y por qué se ha optado por ese diseño, pruebas realizadas a la aplicación (y qué pruebas se han realizado en caso de ser pruebas manuales), y cualquier detalle sobre el funcionamiento del proyecto. Se debe mostrar directamente el resultado, y no como si fuera trabajo en progreso. Se pueden incluir capturas de pantalla de la aplicación en este apartado.

Diseño de la Interfaz de Usuario(WiFrame)

- Vista de Inicio: Vista dedicada al inicio de sesión la cual al introducir tus indicios de cuenta, el programa te llevará o a la pantalla dedicada a los alumnos o a la ventana dedicada a los monitores.



- Ventana de Alumno: Esta ventana incluye todas las opciones de inicio que un usuario de tipo Alumno puede elegir entre ellas todo en cuanto a actividades disponibles o sus actividades ya reservadas, también en cuanto al acceso a las salas del centro y sus datos de sesión como usuario de la web.



- Para el Usuario Monitor:

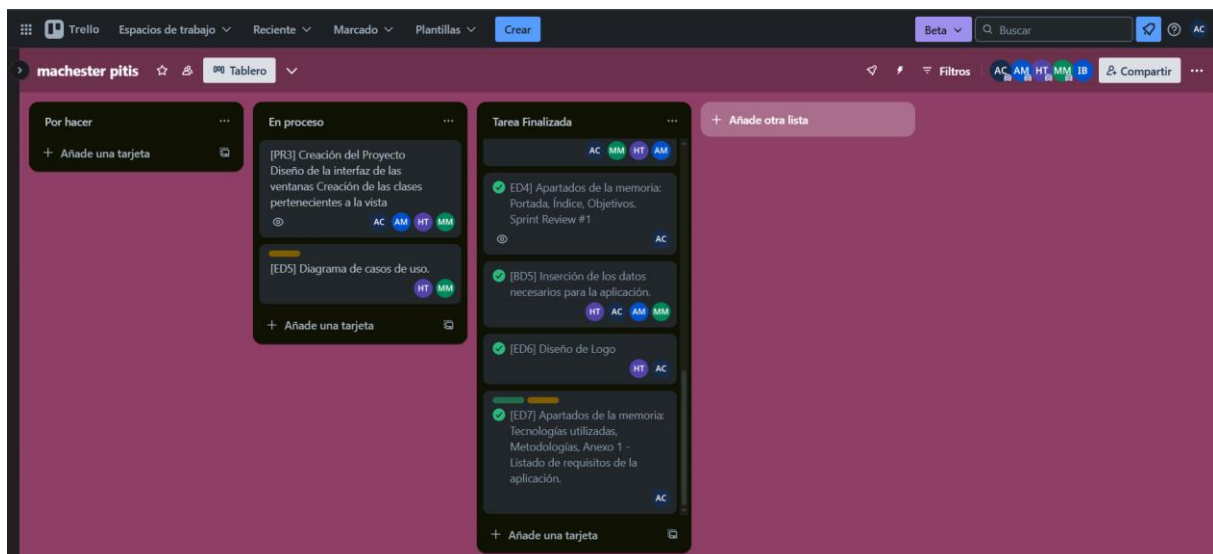
- Modificar Actividad: Cambia detalles de una actividad.
- Asignar Sala a Actividad: Designa un lugar físico para la actividad.
- Asignar Monitor a Actividad: Asigna a un monitor responsable de la actividad.
- Para el Usuario Alumno:
  - Consultar Actividad Disponible: Ver actividades a las que puede inscribirse.
  - Inscribirse a Actividad Disponible: Registrarse en una actividad.
  - Eliminar Inscripción: Cancelar su participación en una actividad.

## 5. Metodología

Para este apartado se recomienda una extensión de 3 a 6 páginas. Explicación de cómo se ha aplicado la metodología Scrum al trabajo, representando el flujo de trabajo con un Diagrama de Gantt, Sprint Backlog y/o Burndown chart, y añadir un enlace a Git y una imagen del tablero Trello al final del proyecto (junto al enlace). Cómo se ha coordinado el equipo para poder finalizar el trabajo. Aportación realizada por cada estudiante.

Para el desarrollo de este proyecto se ha utilizado la metodología Scrum, una metodología ágil que permite estructurar el trabajo en iteraciones cortas llamadas sprints. Esta metodología se ha ajustado muy bien al ritmo de trabajo del equipo, permitiendo una planificación progresiva y la posibilidad de realizar cambios según las necesidades del proyecto. El equipo de trabajo se ha organizado de forma colaborativa, asignando a cada integrante un conjunto de tareas específicas por sprint. Cada miembro ha sido responsable tanto de su parte como de colaborar en las integraciones necesarias. La comunicación entre los miembros se ha mantenido a través de reuniones periódicas y el uso de plataformas colaborativas como Trello para la organización de tareas.

El tablero Trello se organizó en columnas como "Por Hacer", "En proceso" y "Tarea Finalizada". Cada tarjeta incluía una descripción de la tarea, el responsable y la fecha estimada de entrega. Esto permitió una visualización clara del estado del proyecto.



Para asegurar una correcta implementación acerca de la metodología Scrum en el trabajo, se desarrolló de la siguiente forma:

- **Sprint Backlog:** con las tareas que se abordarían en cada sprint, divididas por prioridad y con responsables asignados. (Trello)
- **Daily Stand-ups:** aunque no se realizó a diario, se mantuvieron reuniones frecuentes para resolver dudas, revisar avances y redistribuir tareas si era necesario.
- **Sprint Review y Retrospective:** al final de cada sprint se evaluaron los resultados obtenidos, identificando fortalezas y áreas de mejora, lo cual contribuyó directamente a una optimización continua.

## **6. Resultados y conclusiones**

Para este apartado se recomienda una extensión de 1 a 2 páginas. Resumen de resultados obtenidos en el proyecto y conclusiones del grupo sobre el trabajo realizado.



## **7. Trabajos futuros**

Para este apartado se recomienda una extensión de 1 a 2 páginas. Próximas mejoras que se habrían implementado en caso de haber tenido más tiempo o más conocimiento del lenguaje.

## Anexos

### Anexo I – Listado de requisitos de la aplicación

En la realización del proyecto, sería muy recomendable hacer y un Análisis y diseño de los requisitos Hardware y Software que tendría el Sistema con el que se operara.

#### Requisitos Hardware:

- Procesador:
  - Intel Core i5 de 11ª generación o superior o una AMD Ryzen 5. Ya que las herramientas que utilizaremos como, MySQL, GitHub, GitKraken, Eclipse requieren de capacidad de procesamiento multitarea para manejar la ejecución de aplicaciones de desarrollo, simulación y manejo de bases de datos.
- Memoria RAM:
  - 16GB y un mínimo de 8GB. Este tipo de herramientas pueden consumir mucha memoria cuando se ejecutan múltiples proyectos o bases de datos simultáneamente.
- Almacenamiento:
  - SSD de 512GB mínimos, los discos SSD nos ofrecen un tiempo de arranque y carga muchos mas rápidos.
- Tarjeta Gráfica:
  - Tarjeta Gráfica integrada o una discreta de gama baja sería suficiente, ya que para unas tareas de desarrollo no necesitamos disponer de una GPU pero siempre es útil contar con una para la visualización de interfaces.
- Pantalla:
  - Esta nos proporciona un espacio donde poder trabajar con múltiples ventanas y herramientas.

#### Requisitos Software:

- Sistema Operativo Windows 11:
  - Este es el más actualizado y ofrece una mayor gama de soporte nativo para la mayoría de las herramientas requeridas.

Herramientas de Desarrollo:

- Eclipse IDE
- GitHub Desktop
- GitKraken
- MySQL
- Navegador Web

### **Anexo II – Guía de uso de la aplicación**

Incluirse en este anexo capturas de la aplicación, a modo de manual de usuario, incluyendo tanto el acceso de usuario, como de admin.

### **Anexo III**

...

Incluirse como anexos cuestiones que se consideren fundamentales incluir en el proyecto, pero que por extensión es preferible presentar como anexo, como pueden ser fragmentos de código, o ejecución de las pruebas realizadas.

#### CONSIDERACIONES GENERALES:

Se deben borrar todas las “normas” de lo que hay que seguir en la plantilla.

No se debe modificar el tipo de letra, interlineado, márgenes ni la configuración del documento. Los fragmentos de código deben estar en tipografía “consolas” y en tamaño 11.

En el trabajo se debe utilizar siempre la tercera persona del singular, nunca la primera persona.

En caso de incluir referencias, se debe añadir un apartado “8. Referencias”. Además, se sugiere aplicar las normas APA para nombrar y citar las referencias, ya que serán las que hay que aplicar en el Proyecto de Fin de Ciclo.

Los nuevos anexos se deben consultar con la/s profesora/s antes de añadirse. Además, la parte de “Anexos” debe comenzar en una nueva página.