



**Universidad  
Europea** MADRID

# PROYECTO INTEGRADOR

## 1ºDAW

### Jamón asiático



María Qingxuan Garrido

Abel Gonzalez Plaza

Rosa Xiu Qiong Hu

Alejandro Raboso

## Índice

Resumen.....	2
1. Introducción.....	3
2. Objetivos.....	4
3. Tecnologías utilizadas.....	5
4. Desarrollo e implementación.....	7
5. Metodología.....	13
6. Resultados y conclusiones.....	16
7. Trabajos futuros.....	17
Anexos.....	18
Anexo I – Listado de requisitos de la aplicación.....	18
Anexo II – Guía de uso de la aplicación.....	19
Anexo III.....	19
Anexo IV Historia del proyecto.....	19

## Resumen

**Palabras clave:** Java, GitHub, Eclipse, programación, entornos de desarrollo, bases de datos, vistas, aplicación, polideportivo, actividades, diagramas, clases, proyecto integrador, documento.

## 1. Introducción

Nuestro proyecto integrador, titulado como Jamón Asiático, ha sido desarrollado por los estudiantes de 1º de DAW como parte del módulo de programación, bases de datos y entornos de desarrollo.

Se trata de un proyecto que combina diversas áreas de desarrollo, como la programación orientada a objetos, la gestión de bases de datos, el diseño de interfaces y la organización del trabajo del equipo mediante metodologías ágiles.

La idea principal del proyecto es desarrollar una aplicación funcional que permita gestionar de forma más sencilla y eficiente las actividades del polideportivo. Esta aplicación está dirigida tanto a los alumnos como a los monitores quienes podrán utilizarla para inscribirse, organizar o consultar actividades según el espacio y el horario.

Durante el desarrollo se ha buscado un entorno de trabajo real, aplicando herramientas y tecnologías como Eclipse, github, mysql, trello, draw.io, discord etc.

Uno de los ejes fundamentales del proyecto ha sido el modelo-vista-controlador, en la cual ha permitido una mejor organización del código, facilitando su mantenimiento y escalabilidad.

Además, a lo largo del desarrollo se ha seguido una metodología scrum, dividiendo las tareas en sprints y haciendo revisiones frecuentes del progreso. Esta forma de organización nos ha permitido adaptarnos rápidamente.

Con este proyecto hemos aprendido mucho poniendo en práctica los conocimientos adquiridos a lo largo del curso.

## 2. Objetivos

El objetivo principal de este proyecto, es crear una aplicación para los alumnos del polideportivo de la UEM. Para ello necesitaremos:

- Crear el proyecto de Github permitiendo que todos tengan el proyecto en su repositorio local.
- Dividir el proyecto en carpetas para mantener una estructura.
- Realizar el análisis y el diseño de cualquier aplicación empleando técnicas UML para poder ver así la estructura del software y su funcionalidad de una manera ágil.
- Documentar las aplicaciones que utilizaremos.
- Gestionar las diferentes versiones de un software y el trabajo colaborativo.
- Realizar pruebas de testeo sobre los programas para comprobar que tenemos bien lo que vamos realizando.
- Seguir metodologías ágiles para el desarrollo y planificación de programas.

## PROGRAMACIÓN

- Planificar la aplicación.
- Realizar el diseño de clases necesarias para seguir el patrón MVC, así podremos ver mejor dónde está el fallo.
- Implementar cada una de las clases para lograr su objetivo: clases de interfaz gráfica pertenecientes a la vista, clases que representen el modelo y clases encargadas de la lógica de la aplicación pertenecientes al control.
- Desarrollar una aplicación que gestione información almacenada en bases de datos relacionales identificando y utilizando mecanismos de conexión.

## **BASES DE DATOS**

- Generar el modelo entidad/relación.
- Crear bases de datos definiendo su estructura y las características de sus elementos según el modelo relacional.
- Diseñar modelos lógicos normalizados interpretando diagramas entidad/relación.
- Realizar el diseño físico de bases de datos utilizando asistentes, herramientas gráficas y el lenguaje de definición de datos.
- Consultar y modificar la información almacenada utilizando asistentes, herramientas gráficas y el lenguaje de manipulación de datos.

### 3. Tecnologías utilizadas

Enlace al repositorio de Github:

[https://github.com/MariaQGarrido/Proyecto\\_Integradoe\\_1DAW.git](https://github.com/MariaQGarrido/Proyecto_Integradoe_1DAW.git)

Enlace al Trello:

<https://trello.com/invite/b/67dd4873089fda2a0bb3dfce/ATTIcb7f7e452c8cf0191e2c5bd836da28361330B718/proyecto-integrador>

#### - Aplicaciones utilizadas

**Java:** lenguaje usado para desarrollar la aplicación y su funcionalidad.

**GitHub:** usada para guardar el código, trabajar en equipo para mejor coordinación y llenar el control de versiones.

**MySQL:** es un sistema de base de datos donde se almacena toda la información de la aplicación.

#### - Herramientas utilizadas

**Wireframe:** herramienta para diseñar la estructura de la app antes de programarla, antes de hacer las vistas.

**Draw.io:** herramienta para hacer diagramas como el modelo entidad-relación y flujos de trabajo.

**Trello:** herramienta para organizar tareas, repartir el trabajo y hacer seguimiento del proyecto.

**Whatsapp:** herramienta para comunicar de forma más rápida y resolver dudas del día a día.

**Discord:** herramientas para reuniones por voz, compartir pantalla y coordinar el trabajo en grupo.

**Documento Word:** herramienta de redacción de la documentación y los informes del proyecto.

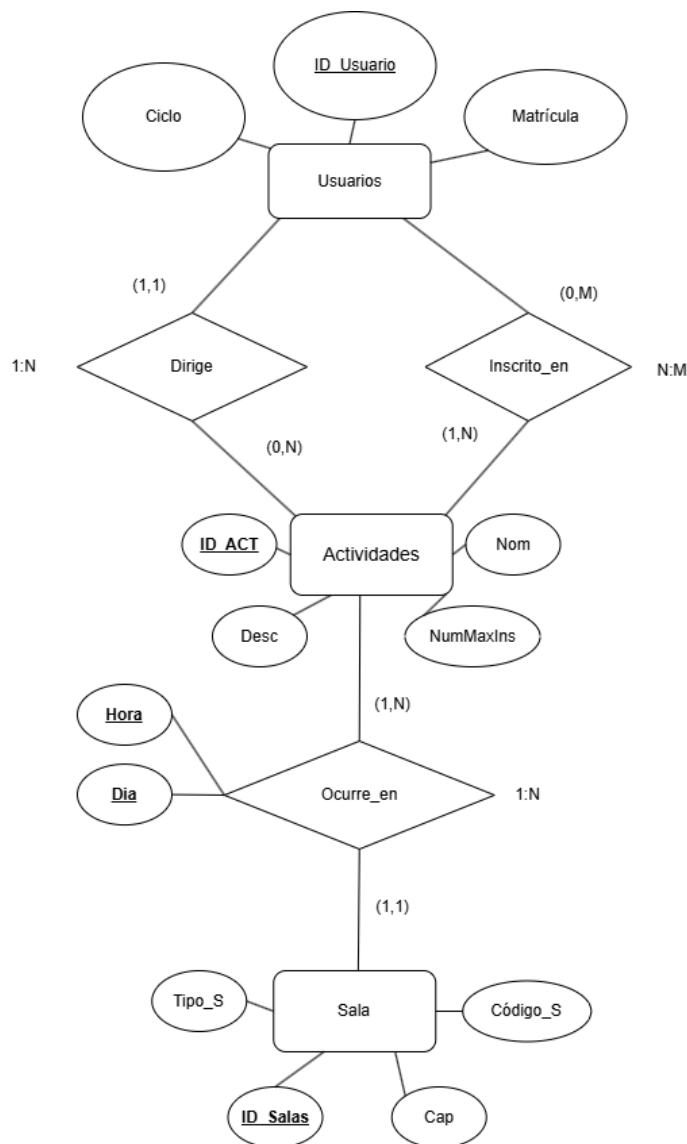
**Google Meet:** reuniones virtuales con el equipo o el profesor para la coordinación.

## 4. Desarrollo e implementación

### BASES DE DATOS

Para poder desarrollar esta aplicación necesitaremos una base de datos. Primero creamos el modelo Entidad/Relación para diseñar y estructurar la información y visualizar y relacionar los datos de nuestra app.

#### Modelo entidad-relación:



En una sala se pueden realizar entre 1 o N actividades, las actividades se pueden realizar en distintas horas y días. Por ende, el día y la hora en que se va a celebrar la actividad es clave primaria.



Una vez aclarado el modelo Entidad/Relación, estableceremos las relaciones entre las entidades mediante claves primarias y foráneas, para facilitar el acceso a la información a través de MySQL.

### **Modelo relacional:**

USUARIOS(ID\_Usuario, Matrícula, Ciclo)

PK: ID

INSCRITO\_EN(ID\_USUARIO\*, ID\_Act\*)

PK: ID\_Usuario, ID\_ACT

FK: ID\_Usuario->(Usuarios), ID\_Act->(Actividades)

ACTIVIDADES(ID\_Act, ID\_Monitor\*, Nom, Desc, NumMaxIns )

PK:ID\_Act

FK: ID\_Monitor->(USUARIOS)

SALAS(ID, Tipo\_S, Cap, Código\_S)

PK:ID

OCURRE\_EN(ID\_Act\*, ID\_Salas\*, Dia, Hora)

PK:Dia, Hora, ID\_Act, ID\_Salas

FK: ID\_Act->(Actividad), ID\_Salas->(Salas)

Se hace la normalización para evitar duplicaciones y actualizar y modificar los datos sin que afecte a otras tablas.

### Normalización:

USUARIOS:

<u>ID_Usuario</u>	Ciclo	Matrícula
111	DAW	2299456
222	DAM	2288433

INSCRITO\_EN:

<u>ID_Usuario</u>	<u>ID_Act</u>
111	456
222	123

ACTIVIDADES:

<u>ID_Act</u>	Descrip	NumMaxIns	Nom	<u>ID_Monitor*</u>
456	xxxxxx	10	Pañuelo	558
123	xxxxxx	11	Fútbol	932

OCURRE\_EN:

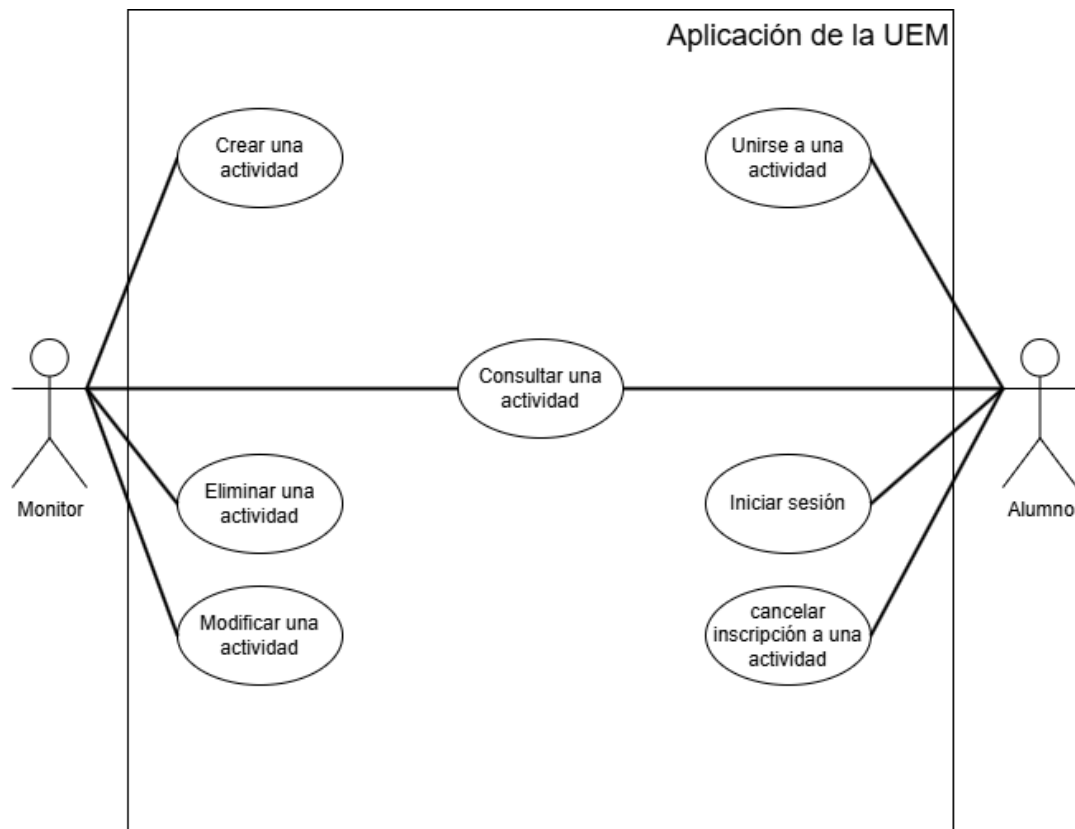
<u>ID_Act*</u>	<u>ID_Salas*</u>	<u>Dia</u>	<u>Hora</u>
456	001	12/05/2025	10:00
123	002	25/04/2025	13:00

SALAS:

ID_Salas	Tipo_S	Capacidad
001	Polideportivo	+10
002	Campo de fútbol	+20

## ENTORNOS DE DESARROLLO

### Diagramas de casos de uso



En este diagrama de caso de uso damos por hecho que el usuario ya tiene una cuenta en la aplicación. El usuario puede ser o bien un instructor o un alumno. El instructor, puede tanto crear las actividades como destruirlas y al crearlas se le asigna una sala a la actividad. El alumno por otra parte puede unirse o salir de una actividad.

## Diseño del logo:



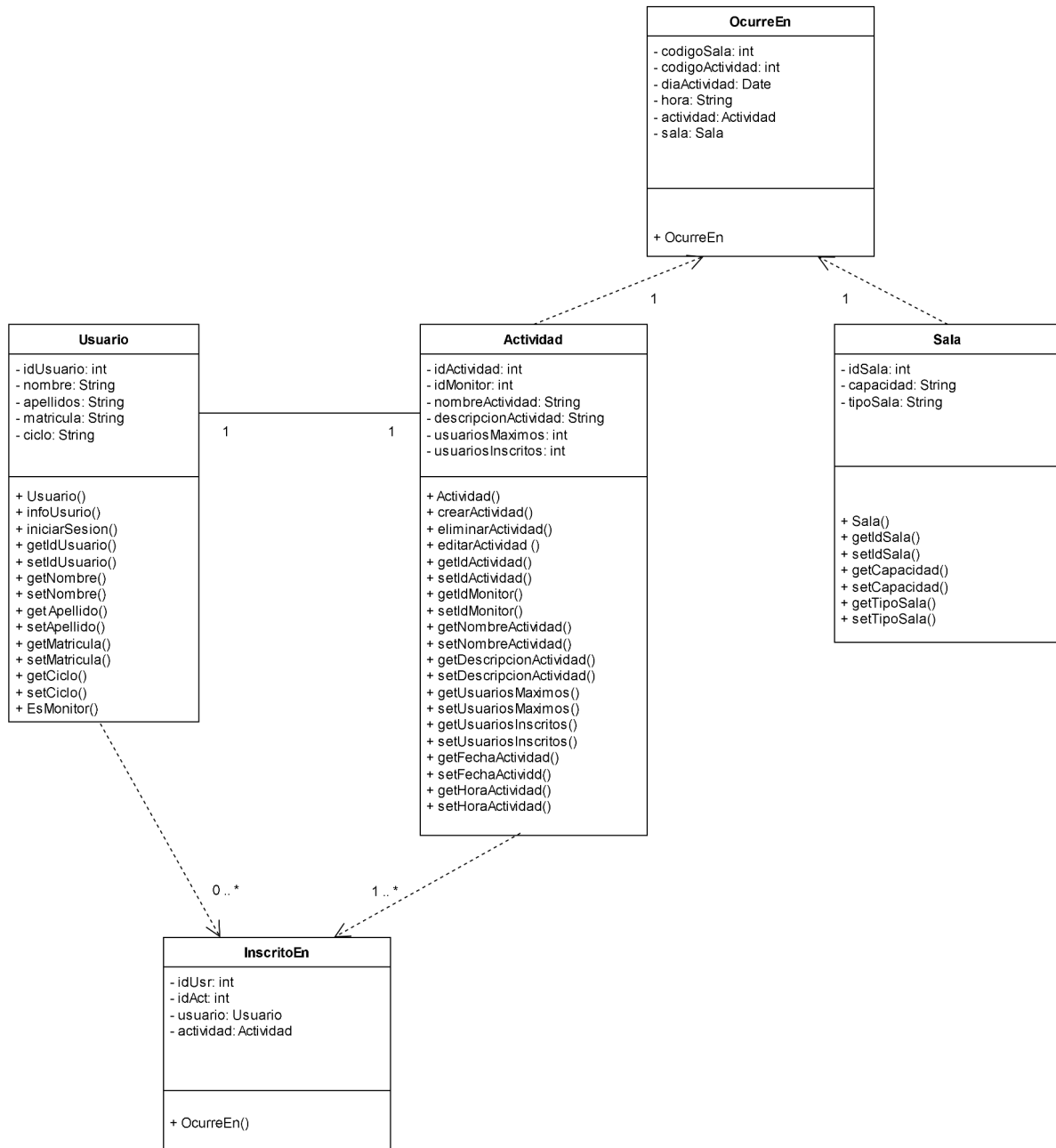
El logotipo fue diseñado para representar el ámbito deportivo del polideportivo, aunque el diseño fue realizado originalmente por un miembro que ya no forma parte del grupo y no se dispone de la justificación completa de las ideas. Hemos elegido los colores naranja y amarillo porque son los que mejor se adaptan a las instalaciones y el ámbito del polideportivo. Y la figura representa la energía del deporte y de las actividades.

## Diagrama de clases:

El siguiente diagrama representa las clases principales de nuestro proyecto: Usuario, Actividad, Sala, InscritoEn y OcurreEn.

Las relaciones entre las siguientes clases funcionan de la siguiente manera:

- Usuario-Actividad: Tienen una relación de asociación (1-1) ya que un usuario le pertenece una actividad, ya sea monitor o persona que hace la actividad.
- Usuario-InscritoEn-Actividad: Estas tres clases tienen una relación de dependencia, pues el usuario con in de usuario necesita inscribirse para poder hacer la actividad.
- Actividad-OcurreEn-Salas: También tiene una relación de dependencia. Pues necesitas tanto el código de la sala como el código de la actividad para poder hacer una actividad en un sitio.



## 5. Metodología

Para el desarrollo de este proyecto se ha utilizado la metodología ágil Scrum, ya que nos permite trabajar de forma más organizada, flexible y adaptativa. Esta metodología se basa en ciclos de trabajo cortos llamados sprints, en los que podemos planificar, desarrollar y revisar el avance del proyecto.

Además se ha utilizado el diagrama de Gantt para visualizar y organizar las tareas a lo largo del tiempo, esto nos ayuda a tener el control más claro sobre la planificación general y seguimiento del progreso de cada parte del proyecto.

ACTIVIDADES	TIEMPO DE DURACIÓN											
	MARZO				ABRIL				MAYO			
	1	2	3	4	1	2	3	4	1	2	3	4
Documento												
Proyecto GitHub - Trello												
Diagrama E/R												
Análisis Sistema												
Creación vista de monitor												
Bases de datos												
Wireframe y vistas												
logo												
Diagrama de clases												
Creación de clases (PR)												
JavaDoc												
Refinamiento programa												
Pruebas Junit												
Integración de BD y PR												
Manual GitHub												
Presentación												

## **6. Resultados y conclusiones**

## **7. Trabajos futuros**



## Anexos

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

#### Análisis y diseño de los requisitos hardware y software.

Eclipse es un entorno de desarrollo integrado (IDE) de código abierto utilizado para programar en diversos lenguajes

- Requisitos de hardware y software requerido:

- Sistema operativo: windows 7 o superior
- Procesador: intel desde 1.2 GHz o equivalente familia AMD
- Memoria de 4 GB de RAM como mínimo, se recomienda 8 GB
- Almacenamiento al menos 2 GB de espacio libre

Sistema operativo compatible

- Windows 10/11 de 64 bits
- macOS de 64 bits
- Linux

Recomendable utilizar una versión de Eclipse del 2024. Versiones más antiguas no procesan caracteres como la 'Ñ'. Para ello se recomienda también:

- Versión de Java del Sistema 5.0 o superior.

SQL se requiere un mínimo de 6GB de espacio disponible en disco, y se recomienda la instalación en equipos con los formatos de NTFS o ReFS. Se admite el sistema de archivos FAT32, pero no se recomienda porque es menos seguro.

- 512 MB de RAM como mínimo mínimo. Se recomienda más.
- Instalador de MySQL.
- Procesador recomendado 2.0 GHz

Sistema operativo compatible

- Windows 10/11 de 64 bits
- macOS
- Linux

Acciones de GitHub:

- Para correr una acción de GitHub en Windows: Un microprocesador de 4 núcleos y 16 GB de RAM.
- Para hasta 740 procesos: 8 núcleos de procesamiento y 64 GB de RAM.

Sistema operativo compatible

- Windows 8.1 o posterior
- macOS
- Linux

## **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 – Historial del proyecto**

Sprint 1: Durante el primer Sprint, no hemos tenido ningún problema significativo.

Sprint 2: Durante el segundo Sprint, hemos tenido una falta de comunicación dentro del equipo, lo que ha generado bloqueos a la hora de completar ciertas tareas. Esto se debió principalmente a la ausencia de un miembro del grupo, ya que habíamos dividido el trabajo en cuatro partes. Aun así, conseguimos finalizar el trabajo a tiempo.

Sprint 3: Durante el tercer sprint, no hemos tenido ningún problema significativo, las tareas planificadas se han desarrollado con normalidad y dentro de los plazos establecidos.