

# Enunciado del Proyecto de Programación

---

## Asignatura: Desarrollo de Aplicaciones Multiplataforma

### Objetivo:

El objetivo de este proyecto es que los alumnos demuestren sus habilidades de programación desarrollando una aplicación de software utilizando Java o Python. Cada alumno deberá proponer su propio proyecto, el cual deberá ser único y diferente al de sus compañeros, y ser aprobado previamente por el profesor.

### Requisitos de Entrega:

- 1.- Cada alumno deberá subir su proyecto a GitHub Classroom, incluyendo todos los commits del desarrollo. La fecha tope de entrega es el 14/06/2024 23:49

<https://classroom.github.com/a/wyRAXziF>

- 2.- Se requiere una memoria explicativa del proyecto (puedes añadir la memoria al repositorio) que contenga al menos los siguientes apartados:
  - Explicación del proyecto y el problema a resolver.
  - Diseño de clases (Diagrama de clases).
  - Diseño de la Base de Datos (Diagrama E/R).
  - Casos de uso (Diagrama de casos de uso).
  - Aspectos a tener en cuenta a la hora de ejecutar el proyecto.
  - Problemas o dificultades encontradas durante el desarrollo.
- 3.- La entrega deberá cumplir con las características mínimas del proyecto descritas a continuación.

### Características Mínimas del Proyecto:

- 1.- Interfaz de Usuario: El proyecto debe tener una interfaz de usuario básica.
- 2.- Uso de Ficheros: La aplicación debe hacer uso de ficheros ya sea para guardar configuraciones, datos de ejecución, exportar/importar datos o llevar un log de uso de la aplicación.
- 3.- Base de Datos: El proyecto debe utilizar una Base de Datos, con al menos 2 entidades y una relación.
- 4.- Tecnología: El proyecto puede ser desarrollado en Java o Python.

### Ejemplos de Proyectos:

- 1.- Listado de Tareas: Una aplicación para gestionar tareas con fecha de resolución, prioridades y opciones de búsqueda. Por ejemplo, una tarea podría ser "Crear lista de la compra" con una prioridad alta y una fecha límite.

- 2.- Conoce la Liga: Una aplicación para gestionar equipos de fútbol, jugadores, asignación de jugadores a equipos y estadísticas. Por ejemplo, la aplicación podría permitir registrar equipos como "Real Madrid" con jugadores como "Cristiano Ronaldo" y "Lionel Messi". Realizar búsquedas, editar datos y modificar asignación de jugador y equipo (traspaso)
- 3.- Controlador de Finanzas: Una aplicación para gestionar ingresos, gastos, categorización de gastos y establecimiento de objetivos financieros. Por ejemplo, la aplicación podría permitir registrar un ingreso como "Sueldo de mayo" y gastos como "Alquiler" y "Comida".
- 4.- Control de Ejercicios de Gimnasio: Una aplicación para registrar ejercicios de gimnasio con pesos, tiempos de descanso y series al fallo. Por ejemplo, la aplicación podría permitir registrar un ejercicio como "Press de banca" con un peso de 50 kg y 3 series al fallo.

## **Rúbrica de Corrección:**

### **Funcionalidad y Cumplimiento de Requisitos:**

Nota máxima de 6 si el proyecto arranca, funciona y cumple con las características mínimas.

#### Diseño y Estructura del Código:

Nota máxima de 7 si el proyecto tiene más de 3 entidades en la Base de Datos y muestra un diseño claro y coherente de las clases y componentes del sistema.

Nota máxima de 8 si el proyecto tiene más de 4 entidades en la Base de Datos y demuestra una estructura modular y bien organizada del código.

Nota máxima de 9 si el proyecto tiene más de 4 entidades en la Base de Datos, implementa un log de uso y muestra una estructura del código que fomenta la reutilización y la extensibilidad.

Nota máxima de 10 si el proyecto tiene más de 4 entidades en la Base de Datos, implementa un log de uso, muestra una estructura del código que fomenta la reutilización y la extensibilidad, y demuestra un alto grado de optimización y eficiencia del código.

### **Buenas Prácticas de Programación:**

Se evaluará la legibilidad del código, asegurándose de que esté bien comentado, con nombres de variables descriptivos y una estructura lógica fácil de seguir.

Se valorará la correcta refactorización del código para eliminar redundancias y mejorar la claridad y eficiencia del mismo.

Se considerará la reutilización de código mediante la creación de funciones o clases que puedan ser utilizadas en diferentes partes del proyecto.

Se premiará la ausencia de "código espagueti", es decir, la evitación de estructuras de código confusas y difíciles de entender.