

# Programación Orientada a Objetos

Universidad Politécnica de Madrid  
E.T.S. de Ingeniería en Sistemas Informáticos  
Departamento de Sistemas Informáticos

## Prácticas curso 2023/24

### Aplicación de Gestión de Planes Sociales (GPS)

#### *Objetivo de la práctica*

El objetivo de esta práctica es desarrollar una aplicación para la organización de planes sociales entre un grupo de usuarios.

La aplicación permitirá a los usuarios crear, unirse, puntuar y gestionar planes sociales, que consisten en una serie de actividades consecutivas.

Una actividad es una acción concreta, como, por ejemplo, "Ir al teatro", "Ir al cine", "Dar un paseo por Madrid", "Reunión en un bar para discutir un tema", "Tomar unas cervezas", "Una partida de Pádel" ...

#### *Requisitos Funcionales*

##### Interfaz de Usuario

Se utilizará un CLI. Un CLI, o **Interfaz de Línea de Comando** (por sus siglas en inglés, Command Line Interface), es un tipo de interfaz de usuario que permite a los usuarios interactuar con un sistema informático a través de comandos de texto escritos en una línea de comandos. En lugar de utilizar una interfaz gráfica de usuario (GUI) con ventanas y elementos visuales, un CLI se basa en texto. Los usuarios ingresan comandos y parámetros utilizando el teclado y reciben respuestas en forma de texto en la pantalla. **Un CLI no está basado en menús, sino en comandos que los usuarios deben conocer con su sintaxis.**

##### Registro de Usuarios

Los **usuarios** deben poder registrarse en la plataforma proporcionando **un nombre de usuario, una edad, un móvil y una contraseña**. **El nombre de usuario y móvil deben ser únicos**. La **edad** debe estar **entre los 14 hasta los 100 años**. La **contraseña** debe tener **un mínimo de 3 caracteres**. Los usuarios podrán realizar **un login y un logout**.

##### Crear Actividades

Los usuarios logeados pueden **crear actividades**. Existen **tres tipos de actividades: genérica, cine o teatro**. Cada actividad debe tener **un nombre, una descripción** detallada de la actividad, una **duración en minutos, un coste y un aforo**. **El aforo es opcional**, y en el caso de que **no se indique** se considerará **sin límite**. La actividad **de teatro** tienen un **descuento del 50%** para los jóvenes (**hasta 25 años**) y un **descuento del 70%** para los pensionistas (**desde los 65 años**). La actividad de **cine** tiene un **descuento del 50%** para jóvenes (**hasta 21 años**).

##### Crear Planes o eliminarlos

Los usuarios logeados pueden **crear planes sociales**. Cada plan debe tener **un nombre una fecha y hora, un lugar de encuentro, una capacidad máxima de participantes**, y estará formada por **un conjunto de actividades secuenciales y un conjunto de participantes**. La **capacidad** es un dato

opcional, y si no se pone indica que no hay límite. Cuando un usuario crea un plan, se vuelve propietario del mismo y es el único que lo puede alterar o borrar. Cuando a un plan se le añade una actividad, el aforo siempre debe ser igual o menor de la actividad añadida, así que es posible el aforo valla decreciendo según se le añade actividades.

Al crear un plan, se puede calcular el tiempo total, sumando los tiempos de cada actividad y estableciendo 20 minutos de desplazamiento entre actividades. También se podría calcular el coste total para un determinado participante.

### Listar Planes

Los usuarios logeados pueden ver la lista de planes disponibles, incluyendo detalles como nombre, descripción, capacidad y número actual de vacantes. Teniendo en cuenta que siempre sean planes de futuro.

En esta consulta se podría ordenar por la puntuación de los planes antiguos obtenidos (del mismo propietario) o por la fecha.

### Unirse a Planes o abandonarlos

Los usuarios logeados pueden buscar y unirse a planes existentes. No se permite unirse a un plan si ha alcanzado su capacidad máxima de participantes, o si se solapan en el tiempo con otro plan suscrito por el propio usuario. Ni tampoco se puede suscribir dos veces al mismo plan.

### Saber la lista de planes suscritos

Los usuarios logeados pueden consultar a que planes están suscritos.

### Consultar el coste

Los usuarios logeados pueden consultar el coste asociado a un plan suscrito.

### Tres consultas de búsqueda

Los usuarios logeados podrán realizar 3 consultas a discreción del alumno, de cualquier índole y con los filtros que el alumno considere.

**Debe haber un control alto de errores, no permitiendo la entrada de comandos con falta de datos, o valores incoherentes o incompatibles. En concreto:**

- Cada vez que crea, se debe controlar que los diferentes datos estén dentro de los límites establecidos y que estén todos los obligatorios.
- Que el aforo de un plan siempre este por debajo de aforo de las actividades que lo componen.
- Que no se puede estar suscritos a dos planes que se solapen en el tiempo. Tampoco se puede suscribirse a un plan más de una vez.
- No se puede suscribir, ni abandonar a un plan de pasado.
- No se puede puntuar un plan que no se ha participado, es decir, planes de futuro o planes de pasado que no se ha estado suscrito. No se puede puntuar dos veces.

### Requisitos Técnicos y de calidad

1. Utilizar el lenguaje Java.
2. Utilizar el repositorio de código GitLab.
3. Utilizar el paradigma de Programación Orientada a Objetos.
4. Aplicar todos los Principios de Diseño y Calidad del Código vistos en clase.
5. Implementar una aplicación mediante un CLI (Interfaz de Líneas de Comandos).
6. Utilizar estructuras de datos adecuadas para almacenar información. Se recomienda utilizar una clase **Poblador**, que le da un contenido inicial que facilite su uso y prueba.
7. Implementar manejo de errores para garantizar la robustez de la aplicación y proporcionar retroalimentación al usuario en caso de errores.

### Ejemplo parcial de secuencia de comandos

Un posible uso del CLI, para situarnos mejor y a modo de ejemplo, podría ser la siguiente:

```
App Gestion de Planes Sociales

gps> create-user:Ana G.;27;645154456;123
  Usuario creado: id:1; nombre:Ana G.; edad:27; móvil:645154456; clave:123
gps> create-user:Jorge;23;666222333;123
  Usuario creado: id:2; nombre:Jorge; edad:23; móvil: 666222333; clave:123
gps> create-user:Ana G.;23;666222333;123
  Error: nombre ya existente
gps> create-user:Eva P.;-5;666000666;123
  Error: la edad no puede ser negativa

gps> login:Ana G.;123
gps-Ana G.> logout

gps> login:Ana G.;123
gps-Ana G.> create-activity:Cine;Cine clásico;Películas históricas premiadas;120;10;30
  Actividad creada: id:1; nombre:Cine clásico; descripción:Películas históricas premiadas; duración: 120 min;
  coste:10€; aforo:30
gps-Ana G.> create-activity:Generic;Cervecita en el Centro; bar El Lacon;Para comentar un tema;60;6
  Actividad creada: id:2; nombre:Cervecita en el Centro; descripción:Para comentar la película;
  duración: 60 min; coste:6€; aforo:sin limite

gps-Ana G.> create-event:Dia de cine;03/10/2023-19.00;Plaza Antón Martín;10
  Plan creado: id:1; propietario:Ana G.; nombre:Dia de cine; fecha: 03/10/2023-19:00; lugar: Plaza Antón Martín
  aforo:10; duración:0; coste:0;
  actividades:
gps-Ana G.> add-activity-plan:1;1
  Actividad añadida: id:1; propietario:Ana G.; nombre:Dia de cine; fecha: 03/10/2023-19:00;
  lugar: Plaza Antón Martín; aforo:10; duracion:120 min; coste:10€;
  actividades:Cine clásico
gps-Ana G.> add-activity-plan:1;2
  Actividad añadida: id:1; propietario:Ana G.; nombre:Dia de cine; fecha: 03/10/2023-19:00;
  lugar: Plaza Antón Martín; aforo:10; duración: 200 min; coste:16€;
  actividades:Cine clásico;Cervecita en el centro

gps-Ana G.> list-events:fecha
  Planes: id:1; nombre:Dia de cine;fecha:03/10/2023-19:00; plazas:10;participantes:-

gps-Ana G.> join-event:1;
  Unido al plan: id:1; propietario:Ana G.; nombre:Dia de cine; fecha: 03/10/2023-19:00;
  lugar: Plaza Antón Martín; plazas:9; participantes:Ana G.
gps-Ana G.> join-event:1;
  Error: No se puede subscribir dos veces al mismo plan
gps-Ana G.> logout
gps>
```

En cualquier caso, hay libertad en el planteamiento del CLI

## Entrega

- El código fuente de la práctica y toda la documentación generada (carpeta docs en el proyecto) deberá de estar en un repositorio de Gitlab (<https://gitlab.etsisi.upm.es>) y debe de ser accesible para el profesor del grupo.
- Video demostrativo del funcionamiento de la práctica.
- Diagrama en UML de todas sus clases y dependencias, incluyendo paquetes.
- Enviar tarea en Moddle, indicando la URL del proyecto en el texto de la entrega y las funcionalidades no implementadas o que no funcionen, o, en caso contrario, indicando que funcionan al 100%.

### AVISO IMPORTANTE

Se recuerda a los estudiantes que, según se establece en el Artículo 13 epígrafe 2 de la normativa de evaluación del aprendizaje vigente, *“Ante la comprobación de fraude académico en una prueba de evaluación, se calificará con la puntuación de **cero** al estudiante o estudiantes implicados en la **calificación final de la convocatoria** correspondiente a la celebración de la prueba (ordinaria o extraordinaria). Además, en función de la gravedad del caso, el Tribunal de la asignatura podrá acordar la realización de un examen especial y equivalente para evaluar los resultados de aprendizaje de la asignatura en la siguiente convocatoria oficial.”*

Lo que implica que **la detección de una copia en ALGUNA DE LAS ENTREGAS** de la práctica **se considerará fraude académico** en una prueba de evaluación y **ACARREARÁ LA PUNTUACIÓN DE CERO EN LA CONVOCATORIA ORDINARIA DE ENERO** y perderá el derecho al resto de pruebas de evaluación. Adicionalmente, los estudiantes involucrados en el fraude académico, además de la correspondiente entrega de prácticas, realizarán un examen especial de toda la materia del curso en la convocatoria de julio de 2024.

Es por ello por lo que se recomienda a los estudiantes que extremen las medidas de seguridad en la creación de proyectos en GitLab de forma que el proyecto creado sea **PRIVADO** y que **sólo se conceda acceso a los miembros del grupo de prácticas** y al profesor o profesores que van a corregir la práctica.