

Programación

EXAMEN PRÁCTICO TEMA 7 – ESTRUCTURAS DINÁMICAS DE DATOS

(04/04/2025)



LEE ATENTAMENTE LAS SIGUIENTES INSTRUCCIONES ANTES DE EMPEZAR:



- **Recopila en un documento de texto las evidencias de todo el examen.** Guárdalo de vez en cuando para no perder el avance de tu trabajo.
- Cuando termines, **pásalo a PDF y sube el documento creado a la entrega de AULES.**

PARTE 1: Configuración del entorno (0,5p)

1. Crea un nuevo repositorio llamado “EXAMEN_UD7_[nombre]” desde *SourceTree*. El repositorio debe crearse en local y tener su espejo en remoto, por lo tanto, sincronízalo con *GitHub*.

Pega a continuación la URL a tu nuevo repositorio de *GitHub*:

2. Crea un nuevo proyecto *Java (Maven)* con *IntelliJ* -o el IDE que utilices- dentro del repositorio que acabas de crear. Llámalo “EXAMEN UD7”.
3. Crea en el proyecto dos paquetes nuevos llamados “*chatbot*” y “*excursion*” para ir añadiendo las clases correspondientes al problema 1 y el problema 2.

Sincroniza los cambios en tu repositorio remoto.

PARTE 2: Resolución de problemas

Programa en *Java* la solución a los siguientes problemas. Usa el proyecto que te acabas de crear en el apartado anterior.

Si no has conseguido crearlo correctamente, utiliza alguno de los proyectos que ya tenías para los ejercicios de clase y pega la URL de *GitHub* del repositorio al que vas a subir los cambios.

PROBLEMA 1 (3,5p). Simula un *Chatbot* básico que interactúe con el usuario y responda a preguntas comunes.

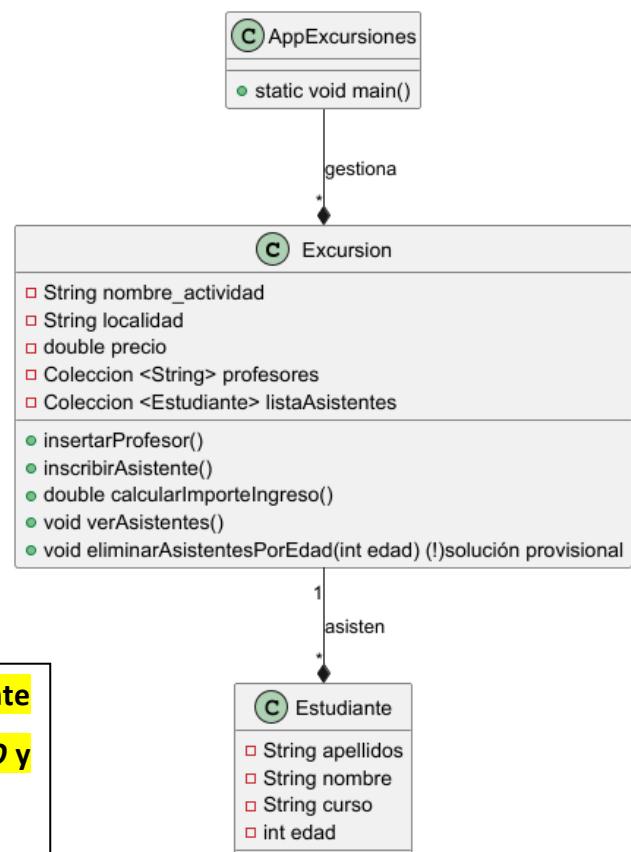
Ejemplo de comportamiento:

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\lib\idea_rt.jar" -jar C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea64.exe
Chatbot: Bienvenido! Escribe <salir> cuando quieras acabar la conversación...
Tú: hola
Chatbot: Hola, ¿En qué te puedo ayudar?
Tú: ¿cómo estás?
Chatbot: ¡Estoy listo para ayudarte!
Tú: de lujo
Chatbot: No te entiendo :(
Tú: HOLA
Chatbot: Hola, ¿En qué te puedo ayudar?
Tú: adiós
Chatbot: ¡Hasta luego!
Tú: gracias
Chatbot: ¡De nada! Que tengas un buen día.
Tú: me voy
Chatbot: No te entiendo :(
Tú: salir
Chatbot: ¡Adiós!

Process finished with exit code 0
```

PROBLEMA 2 (6p). App de gestión de excursiones para el IES MUTXAMEL.

El centro ha decidido implementar un sistema para controlar al alumnado que va a una actividad extraescolar. La app la ha realizado un grupo de 1º DAM como parte del módulo *Proyecto Intermodular*, y la solución de diseño que han presentado es la siguiente:



Realiza un proyecto en Java que implemente la lógica de la aplicación dada, usando POO y estructuras dinámicas de datos.

Condiciones para la construcción de las clases y otras cosas a tener en cuenta

- (1p) Se debe controlar que no se inserten estudiantes duplicados en la lista de asistentes de la clase **Excursion**. Se considera que un estudiante es igual a otro si tiene los mismos **apellidos, nombre y curso**.
- (1p) El método **verAsistentes()** imprime la lista de asistentes, ordenada por *curso* y *apellidos*.
- (0,5p) Para ayudar al profesorado a calcular el importe que le deben ingresar a la empresa que organiza la excursión, nos han pedido implementar una funcionalidad **calcularImporteIngreso()** que devuelva la cantidad automáticamente en función del número de asistentes inscritos.
- (1p) Mientras se desarrolla la app, se recibe un aviso de un problema para una determinada actividad donde ya había inscritos bastantes estudiantes, ya que **han de eliminarse de la lista todos aquellos asistentes que todavía no hayan cumplido los 16 años** por motivos legales. A los desarrolladores se les ocurre, de forma provisional, añadir una funcionalidad **eliminarAsistentesPorEdad()** que recorra la lista de inscritos y vaya eliminando todos aquellos cuya edad sea menor a 16 años.
- (0,5p) De momento, nuestros compañeros de 1º DAM no tienen una app funcional con menús, pero se “apañan” para ir probando las funcionalidades implementadas con el siguiente código:

```
public class AppExcursiones {  
  
    public static void main(String[] args) {  
  
        System.out.println("***** APP EXCURSIONES *****");  
  
        Excursion terra_mitica = new Excursion("Terra Mítica", "Benidorm", 30.0);  
  
        terra_mitica.insertarProfesor();  
  
        terra_mitica.inscribirAsistente();  
        terra_mitica.inscribirAsistente();  
        terra_mitica.inscribirAsistente();  
        terra_mitica.inscribirAsistente();  
  
        terra_mitica.verAsistentes();  
  
        terra_mitica.eliminarAsistentesPorEdad(16);  
        System.out.println("\nDespués de eliminar a los menores de 16 años: ");  
        terra_mitica.verAsistentes();  
  
        System.out.println("\nImporte a ingresar para la actividad " +  
        terra_mitica.getNombre_actividad() + " en " + terra_mitica.getLocalidad() + ":  
" + terra_mitica.calcularImporteIngreso() + " €.");  
  
    }  
}
```

1. (1p) ¿Qué solución propondrías sobre el diseño de clases de la app para evitar tener que lanzar el método *eliminarAsistentesPorEdad()* cada vez que tengamos una actividad limitada por edad?

2. (1p) Pruebas.

Lanza tu programa y explica las pruebas que vas haciendo conforme vas implementando código. Toma estas como referencia:

- a) Probar camino feliz (crear una excursión junto a sus respectivos profesores e insertar asistentes). Comprueba que se ha creado todo bien.
- b) Probar que no se insertan 2 estudiantes que se consideran iguales en la lista de asistentes.
- c) Probar que se eliminan correctamente de la lista de asistentes todos aquellos estudiantes que no alcancen la edad mínima para realizar una actividad.
- d) Probar que se ordena correctamente la lista de asistentes por *apellido y nombre*.
- e) Probar que se ordena correctamente la lista de asistentes por *curso*.

Ejecución de ejemplo

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBr
***** APP EXCURSIONES *****

Introduce el nombre del profesor/a para la excursión en Benidorm
Patricia
Profesor añadido correctamente a la excursión Terra Mítica

Creando estudiante...
Inserta los apellidos:
Boix Bernabeu
Inserta el nombre:
Luis
Inserta el curso:
1ºESO
Inserta su edad:
13
Añadido correctamente el estudiante Luis Boix Bernabeu del curso 1ºESO
```

```
Creando estudiante...
Inserta los apellidos:
Hernández Pastor
Inserta el nombre:
Andrea
Inserta el curso:
2ºESO
Inserta su edad:
14
Añadido correctamente el estudiante Andrea Hernández Pastor del curso 2ºESO

Creando estudiante...
Inserta los apellidos:
Hernández Pastor
Inserta el nombre:
Andrea
Inserta el curso:
2ºESO
Inserta su edad:
14
El estudiante ya existe en la lista de asistentes a la actividad Terra Mítica
```

```
Creando estudiante...
Inserta los apellidos:
Hernández Cuenca
Inserta el nombre:
Guillermo
Inserta el curso:
2ºESO
Inserta su edad:
16
Añadido correctamente el estudiante Guillermo Hernández Cuenca del curso 2ºESO

Lista de inscritos para la actividad Terra Mítica:
[1]. 1ºESO - Boix Bernabeu, Luis (13 años).
[2]. 2ºESO - Hernández Cuenca, Guillermo (16 años).
[3]. 2ºESO - Hernández Pastor, Andrea (14 años).

Después de eliminar a los menores de 16 años:

Lista de inscritos para la actividad Terra Mítica:
[1]. 2ºESO - Hernández Cuenca, Guillermo (16 años).

Importe a ingresar para la actividad Terra Mítica en Benidorm: 30.0 €.
```