

Programación

EXAMEN CONVOCATORIA EXTRAORDINARIA (TEMAS 1-7)

(17/06/2025)



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 junto a un archivo comprimido .zip con las clases .java** que hayas generado durante el examen.

(2p) PROBLEMA 1: Partido de Squash

Nos piden traducir el marcador final de un partido de squash conociendo los ganadores de todos los puntos. Cada partido se introducirá como una sucesión de **A's** y **B's** que indicarán que el jugador A o el jugador B ganó ese tanto. Por cada set que termina, se indica con una **"S"**.

Todos los partidos terminarán con una **"F"** que indica que, o bien el partido terminó con un ganador, o bien dejaron de jugar debido a que finalizó el tiempo de alquiler de la pista. Se debe validar que el formato de la cadena de entrada acabe siempre en **"F"**, y a la vez, que no contenga la **"F"** en ninguna otra posición para dar el marcador por válido.

Ejemplos de entrada y salida:

```
Introduce los tantos para cada jugador:
AAAAAAAF
9-0
```

```
Introduce los tantos para cada jugador:
BBBBBBBBBF
0-9
```

```
Introduce los tantos para cada jugador:
AAAAAAAABBBBBBBBBBSAABBBF
9-11 2-3
```

```
Introduce los tantos para cada jugador:
AABBBAAAABBBAAABBBBBSAABBBF
9-11 2-3
```

```
Introduce los tantos para cada jugador:
AAAFAAAA
El formato no es correcto.
```

```
Introduce los tantos para cada jugador:
F
0-0
```

```
Introduce los tantos para cada jugador:
AAAAAAAABBBBBBBBBBSBBBBBBBBBAAAAAAAF
9-11 9-11
```

```
Introduce los tantos para cada jugador:
AAAFAAAAF
El formato no es correcto.
```

(2p) PROBLEMA 2: Contenido más visto

Para las distintas plataformas de *streaming* de vídeo es muy importante la inversión en marketing necesaria para conseguir un nuevo cliente y el tiempo medio de permanencia de ese cliente. Una de las claves para que los clientes no anulen sus suscripciones es que el contenido que les ofrecen se adecúe a sus gustos, y para eso es de vital importancia que funcione bien el algoritmo de recomendación que decide qué películas y series se aconsejan al usuario cuando accede.

Para la primera versión del “*recomendador*”, se quiere anotar el tiempo (en minutos) que cada usuario ve una serie sin interrupción. Una vez introducida, procesaremos esa información y mostraremos un listado con las series que cada usuario ha visto durante al menos 30 minutos. Esa información se utilizará después para recomendar series cuando el usuario vuelva a conectarse.

Ejemplo de entrada:

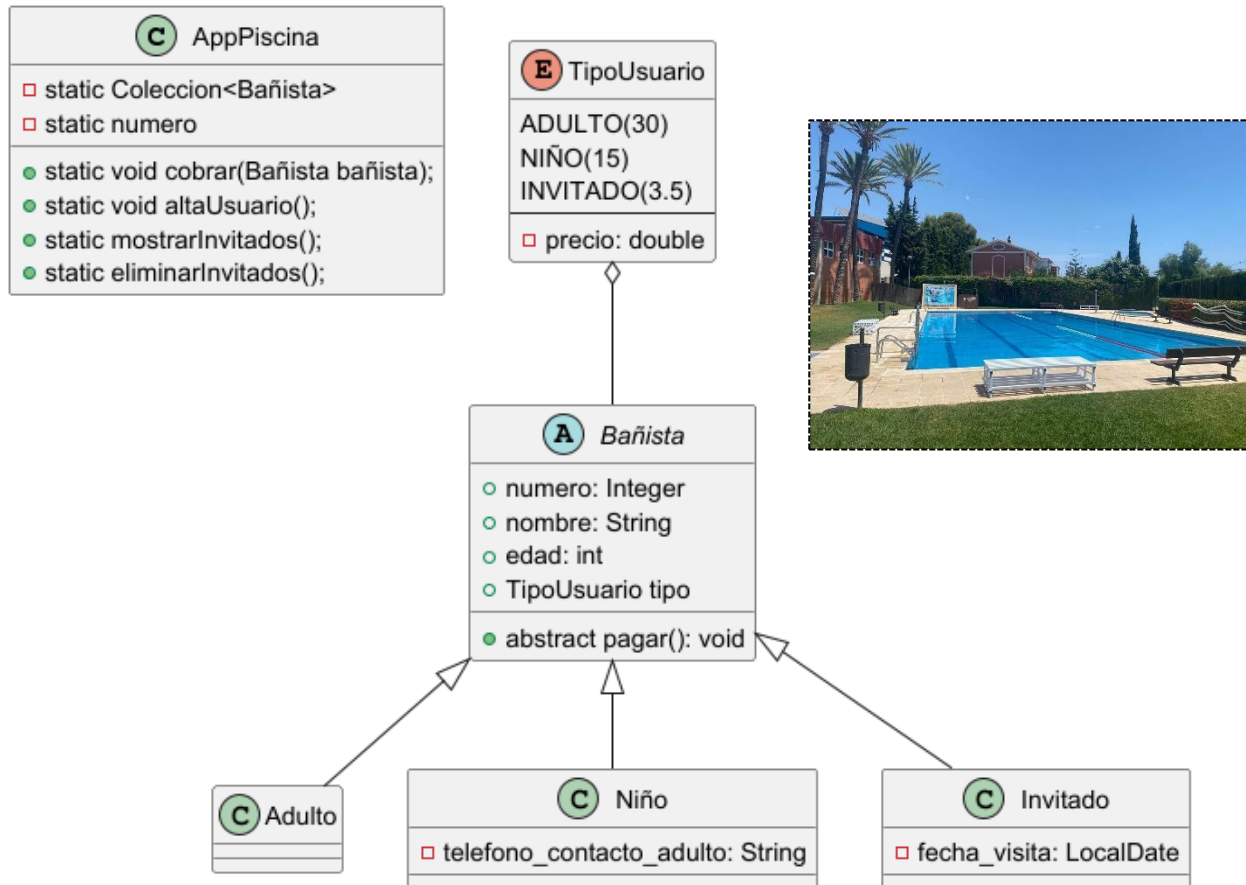
```
"C:\Program Files\Java\jdk-23\bin\java.exe" ...
Introduce el nombre de tu serie... (o 'salir' para terminar)
aquí no hay quien viva
Minutos:
23123
Introduce el nombre de tu serie... (o 'salir' para terminar)
breaking bad
Minutos:
2222
Introduce el nombre de tu serie... (o 'salir' para terminar)
los simpsons
Minutos:
9484
Introduce el nombre de tu serie... (o 'salir' para terminar)
juego de tronos
Minutos:
20
Introduce el nombre de tu serie... (o 'salir' para terminar)
salir
```

Salida:

```
Las series que has visto durante más de 30 minutos son:
breaking bad con 2222 minutos.
los simpsons con 9484 minutos.
aquí no hay quien viva con 23123 minutos.
```

(6p) PROBLEMA 3: App piscina municipal de Mutxamel

El Ayuntamiento de Mutxamel quiere controlar el acceso a las piscinas municipales durante el verano. Para que los conserjes no se líen con la caja y los recibos, a la concejala de Nuevas Tecnologías se le ha ocurrido crear una app para registrar a los usuarios que acceden a la piscina. La app será de uso exclusivo del personal de la piscina, y tendrá las siguiente estructura y funciones:



- a) (1,25p) Crea la clase principal **AppPiscina** que controlará todo el sistema y el esqueleto de la estructura de clases que se muestra. **El atributo *número* del bañista no se puede repetir.**
- b) (1,5p) Implementa el método **altaUsuario()** para que se comporte de la siguiente manera:
- Se muestra un menú para elegir el tipo de bañista a crear.
 - El número de bañista se actualiza automáticamente, sumando 1 número al último creado.
 - Se pide el *nombre* y la *edad* del nuevo bañista.
 - En caso de que el nuevo bañista sea de tipo *NIÑO*, pediremos además un *teléfono* de contacto de alguna persona adulta responsable del menor.
 - En caso de que sea de tipo *INVITADO*, el atributo **fecha_visita** será la fecha actual del sistema.
 - Actualizamos la lista de bañistas.

RECOMENDACIÓN: crear varios bañistas en la lista e imprimirlos antes y después de lanzar este método para comprobar que funciona.

```
*** APP DE MANTENIMIENTO DE LA PISCINA DE MUTXAMEL ***
Alta de usuarios...
  1. Adulto
  2. Niños (<=16 años)
  3. Invitados (pase diario)
-----
Elige un tipo:
1
Introduce el nombre del nuevo bañista:
ismael
Introduce la edad:
21
Nuevo bañista creado.
```

- Lanzaremos una **excepción personalizada** si se intenta insertar un bañista de tipo *NIÑO* y se introduce una edad > 16.

```
*** APP DE MANTENIMIENTO DE LA PISCINA DE MUTXAMEL ***
Alta de usuarios...
  1. Adulto
  2. Niños (<=16 años)
  3. Invitados (pase diario)
-----
Elige un tipo:
2
Introduce el nombre del nuevo bañista:
carlos
Introduce la edad:
17
Exception in thread "main" java.lang.RuntimeException Create breakpoint : ERROR al crear un NIÑO. Edad 17 no permitida
at junio.AppPiscina.altaUsuario(AppPiscina.java:55)
at junio.AppPiscina.main(AppPiscina.java:19)
```

- c) (0,75p) Si todo va bien, el método ***cobrar(Bañista bañista)*** se debe lanzar automáticamente antes de finalizar el método anterior (***altaUsuario()***).

Este método debe, a su vez, llamar al método ***pagar()*** según el tipo de bañista que esté siendo atendido.

```
public static void cobrar(Bañista bañista) {
    bañista.pagar();
}
```

```

*** APP DE MANTENIMIENTO DE LA PISCINA DE MUTXAMEL ***
Alta de usuarios...
    1. Adulto
    2. Niños (<=16 años)
    3. Invitados (pase diario)
-----
Elige un tipo:
1
Introduce el nombre del nuevo bañista:
ismael
Introduce la edad:
21
Nuevo bañista creado.
Adulto (abono de tipo ADULTO) pagando 30.0€

```

- d) (0,75p) ¿Ves algún error en el diseño respecto a la visibilidad de los atributos de la clase *Bañista*? Aplica encapsulación y controla que no se puedan dar de alta bañistas con edades inválidas (años<0).

```

*** APP DE MANTENIMIENTO DE LA PISCINA DE MUTXAMEL ***
Alta de usuarios...
    1. Adulto
    2. Niños (<=16 años)
    3. Invitados (pase diario)
-----
Elige un tipo:
1
Introduce el nombre del nuevo bañista:
prueba años
Introduce la edad:
-4
ERROR. No puede insertar bañistas que no han nacido.

```

- e) (0,75p) El método *mostrarInvitados()* imprimirá por pantalla los invitados que contiene la lista de bañistas en el día de hoy. Ayúdate de esto:

```

for (Bañista b : listaBañistas) {
    if (b instanceof Invitado) {
        System.out.println("Es un invitado");
    }
}

```

```

*** APP DE MANTENIMIENTO DE LA PISCINA DE MUTXAMEL ***
Lista de invitados en el día de hoy:
Número: 5 - Nombre: Fran - Edad: 40
Número: 6 - Nombre: Marta - Edad: 12

```

- f) (1p) El método *eliminarInvitados()* se ejecutará diariamente antes del cierre de la piscina para borrar de la lista a los bañistas invitados que hayan acudido durante el día.

```
*** APP DE MANTENIMIENTO DE LA PISCINA DE MUTXAMEL ***
1 patricia 20 ADULTO
2 pablo 30 ADULTO
3 pepe 12 NIÑO
4 carla 8 NIÑO
5 Fran 40 INVITADO
6 Marta 12 INVITADO
Eliminando invitados de hoy...
1 patricia 20 ADULTO
2 pablo 30 ADULTO
3 pepe 12 NIÑO
4 carla 8 NIÑO
```

Datos de ejemplo para introducir en la lista (para pruebas):

```
listaBañistas.add(new Adulto(1,"patricia",20,TipoUsuario.ADULTO));
listaBañistas.add(new Adulto(2,"pablo",30,TipoUsuario.ADULTO));
listaBañistas.add(new Niño(3,"pepe",12,TipoUsuario.NIÑO,"34565432"));
listaBañistas.add(new Niño(4,"carla",8,TipoUsuario.NIÑO,"8569512"));
listaBañistas.add(new Invitado(5,"Fran",40,TipoUsuario.INVITADO));
listaBañistas.add(new Invitado(6,"Marta",12,TipoUsuario.INVITADO));
```

```
*** APP DE MANTENIMIENTO DE LA PISCINA DE MUTXAMEL ***
Alta de usuarios...
  1. Adulto
  2. Niños (<=16 años)
  3. Invitados (pase diario)
-----
Elige un tipo:
3
Introduce el nombre del nuevo bañista:
invitado
Introduce la edad:
23
Nuevo bañista creado.
Invitado (entrada de tipo INVITADO) pagando 3.5€
1 patricia 20 ADULTO
2 pablo 30 ADULTO
3 pepe 12 NIÑO
4 carla 8 NIÑO
5 Fran 40 INVITADO
6 Marta 12 INVITADO
7 invitado 23 INVITADO
```