Programación

EXAMEN PRÁCTICO – RECUPERACIÓN TEMAS 1-7

(15/04/2025)

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*

1. Crea un nuevo proyecto *Java* (*Maven*) con *IntelliJ* -o el IDE que utilices-. Llámalo "*RECUPERACION-ABRIL*".
2. Crea en el proyecto un paquete nuevo llamado "*ticketmutxa*" para ir añadiendo las clases correspondientes al problema planteado.

PARTE 2: *Resolución de problemas*

Programa en *Java* la solución al siguiente problema. Usa el proyecto que te acabas de crear en el apartado anterior y conforme vayas resolviendo puntos, haz capturas con tus pruebas y pégalas.

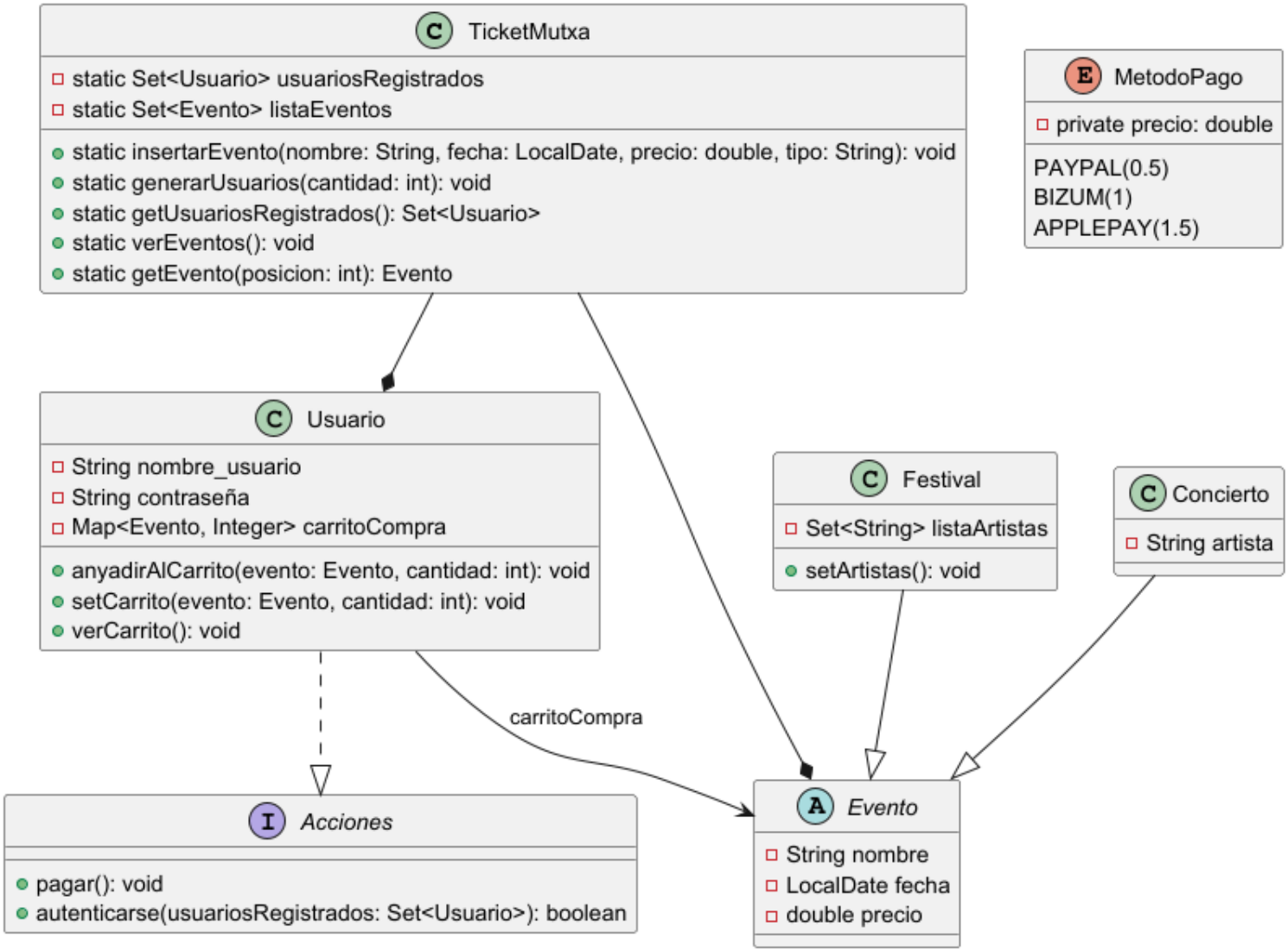
PROBLEMA. *App* de compra de entradas virtual.

Como parece que ya se acaba el curso, el alumnado de los primeros cursos de ciclos formativos está tan contento que no paran de organizar eventos: paellas, conciertos de grupos locales,…

ticketmutxa

Han decidido que, dado que ya son capaces de diseñar y programar "cosas", van a desarrollar una aplicación que les ayude en el proceso de compra de entradas a dichos eventos para no tener que estar pendientes de Bizums sueltos y registrar a mano a la gente que se apunta.

Los grupos de 1º DAM/DAW se han organizado para programarla juntos, y la solución que han presentado es la siguiente:



Realiza un proyecto en *Java* para implementar la aplicación dada. Tienes libertad para modificar cualquier parte de la estructura, siempre que respetes la funcionalidad que se pide a continuación. Implementa también todos los *getters* y *setters* que necesites. Usa *Lombok* si te resulta útil. No se permite el uso de funciones *lambdas* y *streams*.

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

a) El programa principal estará en una clase llamada ***AppCompraEntradas***, que contendrá toda la lógica de la aplicación. Lo primero que hará es cargar la información desde *TicketMutxa*, pero como de momento solamente es un prototipo, deberá generar info de prueba. Usa el siguiente *main* de ejemplo, que crea 2 eventos y genera 4 usuarios:

```
public static void main(String[] args){

    //crear eventos
    TicketMutxa.insertarEvento("Paellas", LocalDate.of(2025, 4, 16),20,"festival");
```

```
    TicketMutxa.insertarEvento("Concierto Estopa", LocalDate.of(2025, 4,
21),40,"concierto");

    // crear usuarios
    TicketMutxa.generarUsuarios(4);

}
```

- (1p) En la clase ***TicketMutxa***, el método ***insertarEvento()*** se encargará de crear nuevos eventos en función del ***tipo*** que se reciba por parámetro (sólo hay dos opciones, "*Concierto*" ò "*Festival*"), y los guardará en su ***listaEventos***. Si el tipo de evento recibido no es alguno de los permitidos, se debe devolver *"No existe el tipo de evento."* y no hacer nada. Dos eventos se consideran iguales si tienen el mismo ***nombre*** y ***fecha***.
- (1p) Por su parte, el método ***generarUsuarios()*** creará 4 *Usuarios* con ***nombre_usuario*** y ***contraseña*** aleatorios, y los añadirá a la lista ***usuariosRegistrados***. **El formato debe ser de 8 caracteres, y no se pueden repetir**. Usa esta cadena como referencia de caracteres válidos:

String caracteres = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";

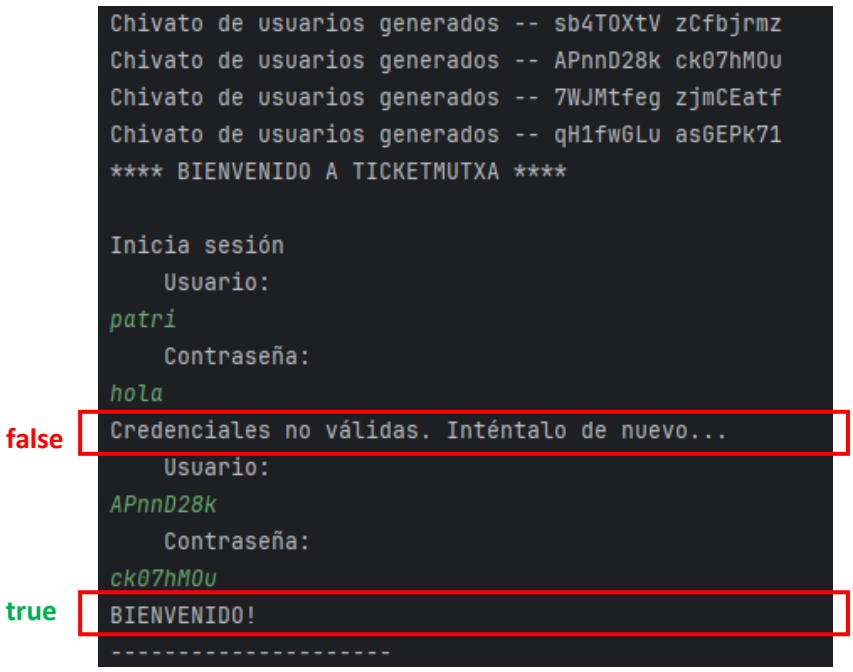
```
Creando festival...
Introduce los artistas que actuarán en el festival (o 'salir' para terminar):
aitana
melendi
salir
Creando concierto...
Artista:
estopa
Chivato de usuarios generados -- Sr9jTtW4 dktmFVfr
Chivato de usuarios generados -- 4EtKRxIs BnUS0ntM
Chivato de usuarios generados -- EecRQXI4 t3DgrypY
Chivato de usuarios generados -- 8ttWM8fc 0wKuPPo0
**** BIENVENIDO A TICKETMUTXA ****
```

TRUCO: imprime un chivato de los usuarios que se van creando para tener datos de autenticación después.

b) (1p) Una vez generada la info necesaria, empieza a ejecutarse la app. Lo primero que hará es pedir al *Usuario* que se identifique haciendo uso del método ***autenticarse()***, al cual le pasaremos la **lista de usuarios** que nos devolverá la clase *TicketMutxa* desde el método ***getUsuariosRegistrados()*.** Esta lista **no se debe poder modificar desde la App**, solamente leerla.

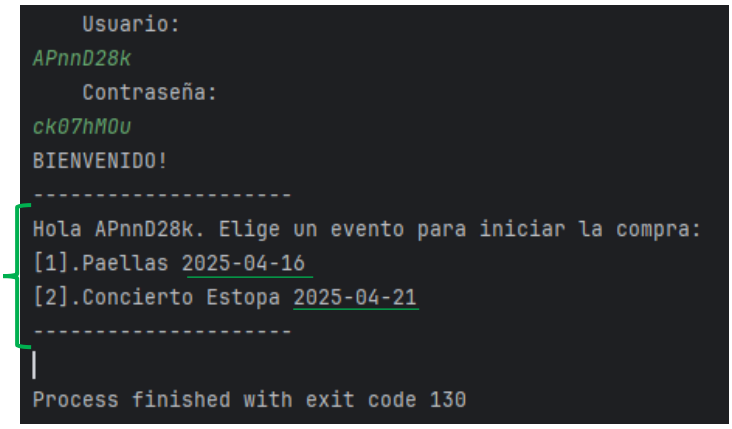
El método *autenticarse()* devolverá ***true*** en caso de que las credenciales introducidas por pantalla coincidan con el ***nombre_usuario*** y la ***contraseña*** de alguno de los usuarios creados

anteriormente, o ***false*** en caso contrario. En función de lo que devuelva, mostraremos al usuario mensajes informativos:



Mientras no se devuelva ***true***, el programa no dejará avanzar. **Guarda el *Usuario* autenticado en una variable *comprador* para usarlo durante el resto del programa**.

c) (1p) Una vez identificados, el programa saludará al usuario y usará el método ***verEventos()*** **para mostrar todos los eventos disponibles ordenados por fecha:**



d) (1p) Nos pedirá que elijamos una opción para proceder a guardar el *Evento* de la compra con el método ***getEvento()***.

Si se elige una opción que no existe en la ***listaEventos***, el método *getEvento()* devolverá ***null*** y tras mostrar un mensaje de error al usuario volverá a pedir la opción:

```
BIENVENIDO!
--------------------
Hola mTc5xz4Z. Elige un evento para iniciar la compra:
[1].Paellas 2025-04-16
[2].Concierto Estopa 2025-04-21
--------------------
3
ERROR. Elige una opción correcta.
2

Estás comprando entradas para el evento Concierto Estopa
```

En caso de existir, **el método *getEvento()* devolverá el *Evento* que corresponda y lo guardaremos en una variable para usarlo durante el resto del programa**.

e) (1,5p) Una vez sabemos el evento de la compra, el programa generará una **cola virtual falsa** para que parezca que nuestro sitio web está vendiendo entradas a tope. Esta cola debemos crearla volviendo a hacer uso del método ***getUsuariosRegistrados()*,** que devolverá de nuevo la **lista de usuarios** registrados.

Se deben añadir a la cola todos los usuarios registrados en el orden que se lean de la lista, excepto el usuario que pertenece al comprador actual, que se añadirá el último.

```
Inicia sesión
    Usuario:
mmQbzloJ
    Contraseña:
bV1OP2Ys
BIENVENIDO!
--------------------
Hola mmQbzloJ. Elige un evento para iniciar la compra:
[1].Paellas 2025-04-16
[2].Concierto Estopa 2025-04-21
--------------------
1

Estás comprando entradas para el evento Paellas
Espera tu turno...
 -- Estás en la posición 4 de la cola virtual.
 -- Estás en la posición 3 de la cola virtual.
 -- Estás en la posición 2 de la cola virtual.
 -- Estás en la posición 1 de la cola virtual.
¡Es tu turno!
```

Añade una instrucción:

Thread.sleep(3000);

para dar realismo a la cola y que vaya actualizando la posición cada 3 segundos.

f) (1p) Cuando sea nuestro turno, la app nos pedirá el número de entradas a comprar y las añadiremos al carrito haciendo uso del método ***anyadirAlCarrito()*** del *Usuario*.

El carrito tendrá una estructura de mapa o diccionario. Se debe controlar que no se puedan añadir más de 7 y mostrar un error en caso de que se intenten colar más:

```
Estás comprando entradas para el evento Paellas
Espera tu turno...
 -- Estás en la posición 4 de la cola virtual.
 -- Estás en la posición 3 de la cola virtual.
 -- Estás en la posición 2 de la cola virtual.
 -- Estás en la posición 1 de la cola virtual.
¡Es tu turno!
¿Cuántas entradas quieres? (máximo 7):
----------------------------------------
8
No puedes añadir más de 7 entradas...
5
Carrito: 5 entradas para Paellas. Importe total: 100.0€. Gastos de gestión: por calcular.
```

En caso de que la cantidad de entradas sea correcta, se mostrará el contenido del carrito haciendo uso del método ***verCarrito()***.

g) (0,5p) Además, mostraremos un menú usando el método ***imprimirMenu()*** con las siguientes opciones (para editar el carrito o terminar con nuestra compra realizando el pago):

```
Carrito: 5 entradas para Paellas. Importe total: 100.0€. Gastos de gestión: por calcular.
Elige una opción...
[1]. Añadir más entradas.
[2]. Eliminar entradas.
[3]. Pagar y finalizar.
----------------------------
```

Si se elige una opción que no está contemplada, el programa reportará un error:

```
Carrito: 5 entradas para Paellas. Importe total: 100.0€. Gastos de gestión: por calcular.
Elige una opción...
[1]. Añadir más entradas.
[2]. Eliminar entradas.
[3]. Pagar y finalizar.
----------------------------
4
Opción no válida.
Carrito: 5 entradas para Paellas. Importe total: 100.0€. Gastos de gestión: por calcular.
Elige una opción...
[1]. Añadir más entradas.
[2]. Eliminar entradas.
[3]. Pagar y finalizar.
----------------------------
```

h) (1p) Si se escogen las opciones 1 ò 2, debemos modificar la cantidad de entradas del carrito para sumar o restar tantas como nos diga el usuario mediante el método ***setCarrito()***. Recuerda que no está permitida la compra de más de 7 entradas por persona, y en caso de que queramos eliminar entradas, tampoco puede haber menos de 0.

```
Carrito: 5 entradas para Paellas. Importe total: 100.0€. Gastos de gestión: por calcular.
Elige una opción...
[1]. Añadir más entradas.
[2]. Eliminar entradas.
[3]. Pagar y finalizar.
---------------------------
1
¿cuántas?
3
No puedes realizar la operación (cantidad entradas=min 0 y máx 7).
Carrito: 5 entradas para Paellas. Importe total: 100.0€. Gastos de gestión: por calcular.
Elige una opción...
[1]. Añadir más entradas.
[2]. Eliminar entradas.
[3]. Pagar y finalizar.
---------------------------
```

i) (1p) Si se elige la opción "*Pagar y finalizar*", invocaremos al método ***pagar()*** para escoger el método de pago y finalizar la compra:

```
Elige una opción...
[1]. Añadir más entradas.
[2]. Eliminar entradas.
[3]. Pagar y finalizar.
---------------------------
3
Elige un método de pago:
PAYPAL (gastos de gestión asociados: 0.5€).
BIZUM (gastos de gestión asociados: 1.0€).
APPLEPAY (gastos de gestión asociados: 1.5€).
Opción:
tarjeta
Método de pago no válido. Vuelve a intentarlo.
Opción:
bizum
Realizando pago con BIZUM (+ 1.0 € de gastos de gestión)
MUCHAS GRACIAS. DISFRUTA DEL EVENTO!
```

En caso de escoger un método de pago no permitido, mostraremos un error al usuario y volveremos a insistir.

Ejecución completa de ejemplo

```
Creando festival...
Introduce los artistas que actuarán en el festival (o 'salir' para terminar):
aitana
melendi
salir
Creando concierto...
Artista:
estopa
Chivato de usuarios generados -- 9FcWA8Eo mYQUvFzI
Chivato de usuarios generados -- bTtTUkGq gBgaYCvm
Chivato de usuarios generados -- sFKD8fk5 OOOmBZs4
Chivato de usuarios generados -- skRóQNpq jzkPB03K
**** BIENVENIDO A TICKETMUTXA ****

Inicia sesión
    Usuario:
bTtTUkGq
    Contraseña:
gBgaYCvm
BIENVENIDO!
---------------------
```

```
---------------------
Hola bTtTUkGq. Elige un evento para iniciar la compra:
[1].Paellas 2025-04-16
[2].Concierto Estopa 2025-04-21
---------------------
1

Estás comprando entradas para el evento Paellas
Espera tu turno...
 -- Estás en la posición 4 de la cola virtual.
 -- Estás en la posición 3 de la cola virtual.
 -- Estás en la posición 2 de la cola virtual.
 -- Estás en la posición 1 de la cola virtual.
¡Es tu turno!
¿Cuántas entradas quieres? (máximo 7):
-----------------------------------------
2
Carrito: 2 entradas para Paellas. Importe total: 40.0€. Gastos de gestión: por calcular.
Elige una opción...
[1]. Añadir más entradas.
[2]. Eliminar entradas.
[3]. Pagar y finalizar.
----------------------------
```

```
----------------------------
1
¿cuántas?
2
Carrito: 4 entradas para Paellas. Importe total: 80.0€. Gastos de gestión: por calcular.
Elige una opción...
[1]. Añadir más entradas.
[2]. Eliminar entradas.
[3]. Pagar y finalizar.
----------------------------
2
¿cuántas?
1
Carrito: 3 entradas para Paellas. Importe total: 60.0€. Gastos de gestión: por calcular.
Elige una opción...
[1]. Añadir más entradas.
[2]. Eliminar entradas.
[3]. Pagar y finalizar.
----------------------------
```

```
----------------------------
3
Elige un método de pago:
PAYPAL (gastos de gestión asociados: 0.5€).
BIZUM (gastos de gestión asociados: 1.0€).
APPLEPAY (gastos de gestión asociados: 1.5€).
Opción:
applepay
Realizando pago con APPLEPAY (+ 1.5 € de gastos de gestión)
MUCHAS GRACIAS. DISFRUTA DEL EVENTO!

Process finished with exit code 0
```

Vídeo demostrativo