

13 de enero de 2025

Examen parcial - 4,5 puntos (Duración: 2,5h)

Antes de comenzar a resolver el examen debes seguir los pasos indicados en la hoja de instrucciones

Continuamos con el dominio relacionado con la gestión del gimnasio. Esta nueva versión, se centra en las competencias CE3 - JDBC y CE4 - Recursividad. Además, incluye tareas correspondientes a las competencias CE1 - Swing y CE2 - Threads, a modo de recuperación, para las personas que no obtuvieron al menos un 45% en cada una de ellas en el examen parcial de noviembre.

Para cada tarea, se indica su puntuación (sobre 4,5 puntos), la clase en la que debes añadir el código y una estimación del número de líneas de código necesarias. El código base está organizado en varios paquetes, cada uno de ellos relacionado con una competencia. En cada paquete encontrarás un método **main()** para ejecutar tu código y ver el resultado: **MainJDBC** (JDBC), **GeneradorRutinas** (Recursividad), **MainJTable** (Swing) y **GymBanner** (Threads). Por último, dentro de la carpeta **resources/doc** encontrarás imágenes y vídeos que te ayudarán a ver con mayor detalle el objetivo de las diferentes tareas.

Tarea 1: Actualizar datos de una BBDD con JDBC [1 punto] [GestorBD] [40 líneas]

El objetivo de esta tarea es la modificación de información de las personas en una BBDD. La interacción con la base de datos se realiza desde la ventana **VentanaGestorAbonos**. Usaremos una BBDD que está en **resources/db/gym.db**. La BBDD ya tiene datos precargados de Actividades, Sesiones, Personas, y las actividades a las que está apuntada cada persona. La base de datos tiene la siguiente estructura de tablas:

ACTIVIDAD: <ul style="list-style-type: none">• ID: Integer autoincremental (PK)• TIPO: Text• PLAZAS_MAXIMAS: Integer	PERSONA: <ul style="list-style-type: none">• ID: Integer autoincremental (PK)• NOMBRE: Text• EMAIL: Text• TIPO_ABONO: Text
SESION: <ul style="list-style-type: none">• ID_ACTIVIDAD: Integer (FK)• DIA: Text• HORA_INICIO: Integer	PERSONA_ACTIVIDAD <ul style="list-style-type: none">• ID_PERSONA: Integer (FK)• ID_ACTIVIDAD: Integer (FK) <p>PK: {ID_PERSONA, ID_ACTIVIDAD}</p>

Desde la **VentanaGestorAbonos**, al seleccionar una persona en la lista de la parte izquierda, en la parte derecha se muestra la información de la persona y las actividades en las que está apuntada. Si se modifica el abono, se actualizan las actividades y se habilita el botón “Actualizar abono y actividades”. Cuando se pulsa ese botón, se invoca el método **updatePersona()** de la clase **GestorBD** que inicialmente está vacío.

Para completar la tarea, tienes que implementar el método **updatePersona()**. Este método tiene que actualizar el tipo de abono en la tabla **PERSONA** y además, hacer cambios en la tabla **PERSONA_ACTIVIDAD**. En este último caso, debes tener en cuenta que la información de la tabla **PERSONA_ACTIVIDAD** es previa a la modificación del abono. Por lo tanto, no es correcta. La lista correcta de actividades en las que está apuntada una persona, está en el objeto **Persona** que recibe el método **updatePersona()**. Esto implica que debes realizar dos tipos de cambios sobre la tabla **PERSONA_ACTIVIDAD** (borrar e insertar).

Puedes validar el correcto funcionamiento de la implementación interactuando con la ventana: (1) cambia el abono de una persona, (2) cierra la ventana, y (3) abre de nuevo la ventana para confirmar que los cambios se han realizado correctamente.

Tarea 2: Recursividad [1 punto] [GeneradorRutinas] [30 líneas]

El objetivo de esta tarea es la generación de rutinas de entrenamiento. Una rutina de entrenamiento tiene una duración específica y se compone de un conjunto de ejercicios (sin repetición) correspondientes a uno o varios grupos musculares y de un nivel de dominio. A partir de una lista de ejercicios, debes generar rutinas de entrenamiento mediante un proceso recursivo. Para lograrlo, tienes que implementar el método **generarRutinas()**. Este método recibe una lista de ejercicios, una duración en minutos, un nivel y una lista de grupos musculares. A partir de esa información, debes generar todas las rutinas posibles (combinaciones de ejercicios) que cumplan lo siguiente:

- La duración total de la rutina debe ser igual a la duración indicada.
- La rutina debe incluir sólo ejercicios del nivel indicado.
- La rutina debe incluir únicamente ejercicios de los grupos musculares indicados (pudiendo ser los ejercicios sólo de uno, más de uno, o todos los grupos musculares indicados).
- La rutina no puede incluir ejercicios repetidos.
- No puede haber rutinas equivalentes (es decir, los mismos ejercicios en distinto orden).

Para comprobar que tu implementación es correcta, tienes el resultado de la ejecución del método **main()** en el fichero **resources/doc/T2 – Recursividad.txt**

Recuperación de las competencias CE1 y CE2

A continuación, se describen 2 nuevas tareas que permiten evaluar de nuevo las 2 competencias correspondientes al examen parcial. A pesar de que hayas liberado alguna de las dos competencias puedes intentar realizar las tareas para mejorar el resultado. La calificación definitiva para cada una de las dos competencias será la más alta entre la obtenida en el examen parcial de noviembre y este examen.

Tarea 3: Renderizado de JTable [1,5 punto] [HorarioGimnasioJTable] [50 líneas]

El objetivo de esta tarea es modificar la funcionalidad de visualización del calendario semanal de sesiones en el JTable que fue parte del examen parcial noviembre. En este caso, queremos que al pasar el ratón sobre las sesiones de una actividad, la celda correspondiente muestre el nombre del tipo de la Actividad y debajo, el número de plazas disponibles y las plazas máximas de la actividad. Además, en caso de que el número de plazas disponibles sea igual a 0, todo el texto aparece en color rojo. La fuente utilizada para el nuevo contenido de las celdas debe ser "Arial" de 18 puntos y negrita.

Si no hay plazas disponibles, también queremos que el color de fondo cuando se muestre la imagen de la actividad sea rojo.

En la carpeta **resources/doc** tienes un vídeo () que muestra el comportamiento esperado.

Tarea 4: Hilos [1 punto] [GymBanner] [30 líneas]

El objetivo de esta tarea es implementar el funcionamiento de un banner que se utilizará para mostrar publicidad de las actividades del gimnasio. El banner se visualiza en una ventana y consiste en un carrusel de imágenes que va cambiando cada 1,5 segundos en una secuencia continua. La ventana proporcionada para implementar esta tarea incluye un botón con el texto "**Ocultar**". Cuando se pulse el botón "**Ocultar**", el carrusel de imágenes se reemplaza por una cuenta atrás de 5 a 0. Al finalizar la cuenta atrás, se vuelve a mostrar la secuencia de imágenes. El botón "**Ocultar**" debe deshabilitarse mientras se muestra la cuenta atrás y estar habilitado cuando se muestre la secuencia de imágenes.

En la carpeta **resources/doc** tienes un vídeo que muestra el comportamiento esperado.