Recuperatorio Programación Básica 1 - Q1 - C01 - TM - 2023

Gestión de gimnasio

Nos solicitaron desarrollar un producto software que permita a los clientes registrar las actividades que realiza, en base a las disponibles en el gimnasio.

Para esto deberemos incluir las siguientes funcionalidades:

Clase GestionDeGimnasio

Es la clase que tiene la interacción con el usuario del software. Debemos completar:

- **iniciarSesion(Gimnasio gimnasio):** Pide el ingreso de credenciales (DNI y contraseña). Verifica si con esas credenciales se puede iniciar sesión. Si el inicio de sesión es exitoso, se muestra un mensaje de éxito y seguido el menú del cliente. Caso contrario, se muestra el mensaje de error: "Usuario y/o contraseña invalido". Si el inicio de sesión fue exitoso, se deberá mostrar el menú del cliente. Tener presente los parámetros que recibe el método menuCliente().
- **registrarme(Gimnasio gimnasio):** Pide el ingreso de los datos necesarios para crear un cliente. Debe verificar que la contraseña sea válida. En caso de no ser válida, se deberá volver a ingresar hasta que sea válida. Se debe corroborar que el cliente no exista buscándolo por su dni. Si existe, mostrar el mensaje: "El DNI ingresado ya existe." Si no existe, entonces se procede a registrar el cliente en el gimnasio. Se debe mostrar un mensaje en caso de éxito y otro de error si no se pudo registrar.
- **menuCliente(Cliente cliente, Gimnasio gimnasio):**
    - REGISTRAR_ACTIVIDAD: Obtener las actividades disponibles para realizar en el gimnasio (obtenerActividadesOrdenadasPorCaloriasQueQuemaDescendente()) y las muestra. Mientas que no se indique salir con 99, se debe pedir el ingreso del ID de la actividad que se desea realizar, buscarla por su id en el gimnasio y luego hacer que el cliente realice la actividad obtenida.
    - ELIMINAR_ACTIVIDAD_POR_ID: Mientras que no se ingrese 99 para salir, se deberá mostrar la lista de actividades del cliente para luego solicitar el ingreso del ID de la actividad que se desea eliminar de la lista de actividades del cliente. En caso de ser eliminada con éxito, mostrar un mensaje acorde. Hacer lo propio en caso de error.

Nombre y Apellido DNI FIrma

- CALORIAS_QUEMADAS_POR_ACTIVIDAD: Mostrar un menú que incluya los tipos de actividad para que el usuario pueda elegir un tipo de actividad. Obtener la cantidad de calorías quemadas por actividad del tipo de actividad ingresada por el usuario y mostrar por pantalla el mensaje: **Actividad: AEROBICO - Calorias quemadas: 400** donde "AEROBICO" representa la opción ingresada por el usuario y 400 el valor obtenido.
- CLIENTE_MENOS_ACTIVIDADES_REALIZO: Obtiene al cliente que menos actividades realizo y lo muestra por pantalla.

Clase Actividad

Representa una actividad que puede realizar un cliente para quemar calorías. Cada actividad cuenta con un id, nombre, duración (en minutos: 30), cantidad de calorías que quema y un tipo de actividad. Nos interesa mostrar el detalle de una actividad.

Clase Cliente

Representa a un cliente que puede asistir al gimnasio para realizar actividades. De los clientes conocemos su DNI, nombre, contraseña y sabemos que realizan actividades dentro de las que dispone el gimnasio. Daremos un cupo de 10000 actividades por cliente. Deberemos desarrollar los siguientes métodos:

- **realizarActividad(Actividad actividad):** Agrega una actividad a las actividades del cliente.
- **eliminarActividadPorId(int id):** Busca una actividad por su id y en caso de existir, la elimina.
- **obtenerCantidadDeActividadesRealizadas():** Revisa cuantas actividades realizó el cliente y devuelve el valor correspondiente.
- **obtenerCantidadDeCaloriasQuemadasPorActividadDeTipo(TipoActividad tipoActividad):** Obtiene la cantidad de calorías quemadas por el cliente al realizar actividades de un tipo determinado.

Tener presente que se deberá mostrar el detalle de un cliente.

Clase Gimnasio

Representa al gimnasio desde el cual los clientes realizan actividades que luego registran en el software. Un gimnasio posee un nombre, además de tener clientes que se registran y realizan actividades. El gimnasio cuenta con 20 actividades repartidas en

tipos de actividad, las cuales vienen pre cargadas para agilizar las pruebas. Se deberán resolver los siguientes métodos:

- **iniciarSesion(int dni, String contrasenia):** obtiene un cliente por su dni y, si existe, verifica que la contraseña sea la suministrada.
- **registrarCliente(Cliente cliente):** Se debe registrar el cliente, agregandolo a los clientes del gimnasio, solo en caso de que el dni del cliente suministrado no exista entre los clientes actuales.
- **buscarActividadPorId(int id):** Buscar entre las actividades del gimnasio alguna que aplique con el id suministrado y devolverla.
- **buscarClientePorDni(int dni):** Revisa entre los clientes del gimnasio, si algún posee el dni indicado y lo devuelve.
- **validarContrasenia(String contrasenia):** Devuelve verdadero en caso de que la contraseña contenga: 2 o más mayúsculas, un largo mínimo de 8 caracteres y, si tiene numeros y estan juntos, no pueden ser consecutivos. Ejemplo válido: PassWord77 - Ejemplos inválidos: PassWord123 o PassWord234 o PassWord345 -> notar que los números consecutivos son: 1 y 2 o 2 y 3, etc.
- **obtenerElClienteQueMenosActividadesRealizo():** Devuelve el cliente que menos actividades realizó.
- **obtenerActividadesOrdenadasPorCaloriasQueQuemaDescendente():** Ordenar las actividades de forma descendente por la cantidad de calorías que se queman y devolverlas.

Lineamientos

- No es posible modificar la firma de los métodos existentes, pero es posible agregar todos los métodos que se consideren necesarios.
- Se pueden aplicar las mejoras que considere necesarias, siempre cumpliendo el punto anterior.
- Se deberán completar los métodos y código faltante para que el juego funcione correctamente.
- Deberá ajustar el nombre del proyecto suministrado indicando sus apellidos y nombres en donde dice “ApellidosNombres”: PB12023Q1C01TMRecuperatorio-**ApellidosNombres.**
- La versión de java del proyecto deberá ser ajustada a la del laboratorio.

Nombre y Apellido DNI FIrma

Condiciones de aprobación

Se considerará desaprobado todo proyecto que:

a. no compile, o,
b. no se identifiquen los atributos necesarios para resolver lo solicitado, o,
c. los tipos de datos elegidos para los atributos no sean los adecuados, o,
d. no cumpla con las condiciones mínimas descritas a continuación.

Se considerará aprobado aquel examen que cumpla con el 70% de puntaje.

Clase de prueba GestionDeGimnasio	iniciarSesion(Gimnasio gimnasio)	5%
	registrarme(Gimnasio gimnasio)	10%
	menuCliente(Cliente cliente, Gimnasio gimnasio)	10%
Clase Actividad	Constructor, getters y setters necesarios, método para mostrar detalles de la actividad	5%
Clase Cliente	Constructor, getters y setters necesarios	2,5%
	realizarActividad(Actividad actividad	5%
	eliminarActividadPorId(int id)	5%
	obtenerCantidadDeActividadesRealizadas()	5%
	obtenerCantidadDeCaloriasQuemadasPorActividadDeTipo(TipoActividad tipoActividad)	10%
Clase Gimnasio	Constructor, getters y setters necesarios	2,5%
	iniciarSesion(int dni, String contrasenia)	5%
	registrarCliente(Cliente cliente)	10%
	buscarActividadPorId(int id)	2,5%
	buscarClientePorDni(int dni)	2,5%
	validarContrasenia(String contrasenia)	10%
	obtenerElClienteQueMenosActividadesRealizo()	5%

Nombre y Apellido DNI FIrma

	obtenerActividadesOrdenadasPorCaloriasQueQuemaDescendente()	5%
Total		100%

Nombre y Apellido DNI FIrma