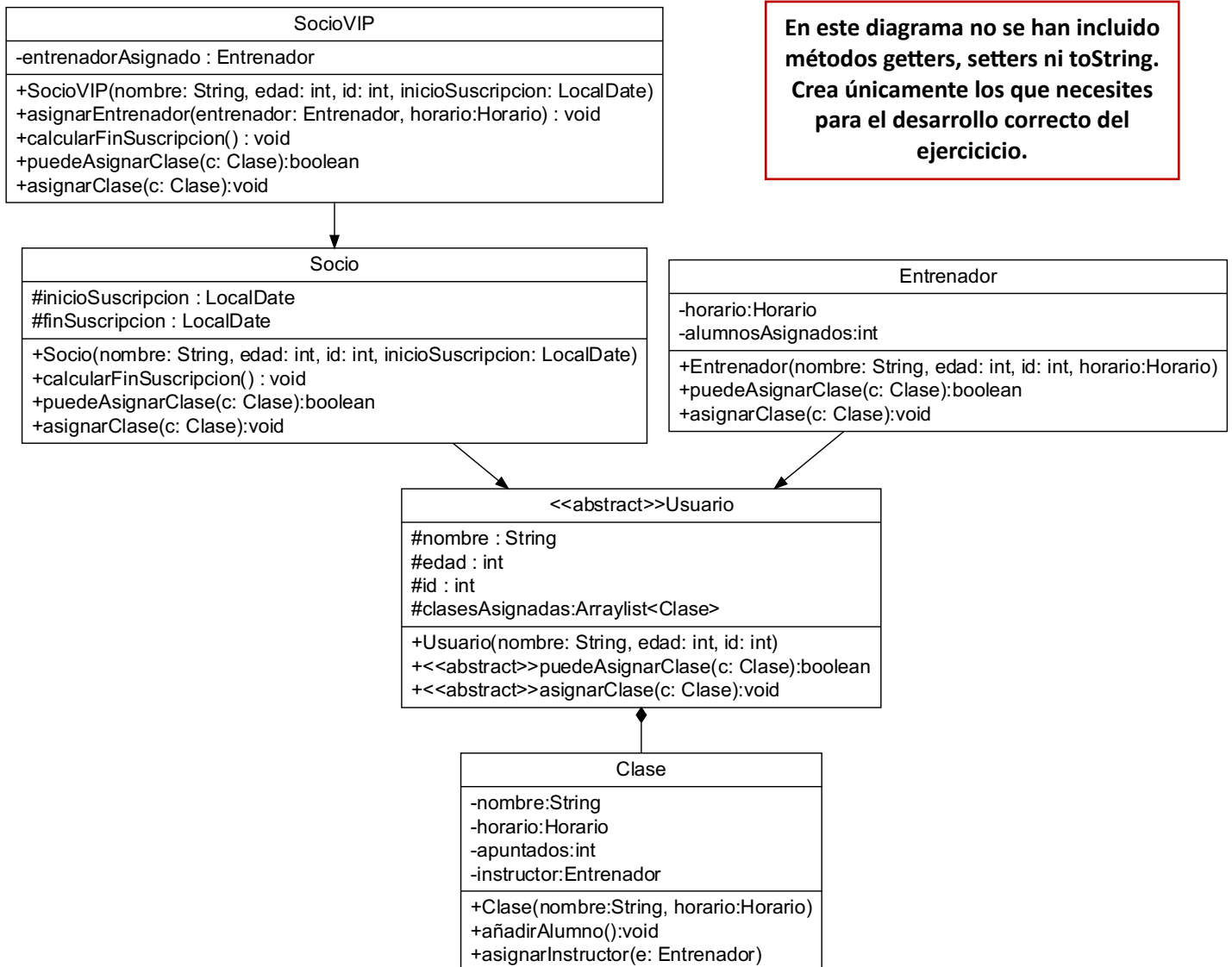


Se pretende desarrollar una aplicación para gestionar un gimnasio. El gimnasio cuenta con diferentes tipos de usuarios: socios regulares, socios VIP y entrenadores. Cada usuario tiene un nombre, una edad, un número de identificación único y un listado de clases asignadas. Los socios (regulares y VIP) tienen también una fecha de inicio de suscripción y una fecha de vencimiento. Los entrenadores tienen un horario (Mañana o Tarde, tipo enumerado) y un número de socios a los que entrenan. La aplicación debe permitir asignar clases a usuarios, mostrar la lista de usuarios registrados, asignar entrenadores a socios VIP.



**En este diagrama no se han incluido métodos getters, setters ni toString. Crea únicamente los que necesites para el desarrollo correcto del ejercicio.**

Además, se deben tener en cuenta las siguientes indicaciones:

- Al crear una Clase será necesario especificar el nombre, y el horario (al principio no tiene alumnos ni instructor). Cuando una clase está llena (máximo 15 alumnos) no se puede apuntar nadie más y solo se le podrá asignar un instructor si no tiene ninguno previamente asignado.
- Al crear cualquier usuario será necesario especificar su nombre, edad e ID (al principio no hay ninguna clase asignada). Si el usuario es un Socio habrá que especificar también la fecha de inicio de la suscripción y, si se trata de un Entrenador, su horario (Mañana o Tarde).
- La suscripción de los socios regulares tiene una duración de 6 meses y la de los socios VIP de 12 meses.
- Cada entrenador puede tener un total de 5 socios para entrenar.
- Al asignar un entrenador a un socio VIP, se debe verificar si el entrenador tiene disponibilidad en su horario.
- Para apuntar a un usuario a una clase, los socios regulares pueden tener un máximo de 5 clases, los socios VIP un máximo de 10 y los entrenadores un máximo de 4.

1. [3.5 puntos] Crea todas las clases necesarias para representar los elementos descritos en el diagrama de clases. Para cada una de las clases del programa, declara los atributos e implementa todos los métodos descritos. Asegúrate de incluir los métodos necesarios para apuntarse a clases y asignar entrenadores. Considera las restricciones y atributos mencionados en el enunciado.
2. [1.5 puntos] Crea e implementa una clase principal en la que se muestre por consola un menú con las siguientes opciones:  
    Bienvenido al Gimnasio. ¿Qué deseas hacer?  
    1. Ver lista de usuarios registrados.  
    2. Apuntar a un usuario a una clase.  
    3. Asignar entrenador a socio VIP.  
    0. Salir del programa.

La clase principal contendrá la lista de usuarios registrados y la lista de clases disponibles. La primera opción del menú se encargará de mostrar la lista de usuarios registrados, incluyendo los detalles específicos de cada tipo de usuario. La segunda opción permitirá al usuario apuntarse a una clase teniendo en cuenta las limitaciones mencionadas. La tercera opción permitirá asignar un entrenador a un socio VIP, verificando la disponibilidad del entrenador y las limitaciones mencionadas de socios y entrenadores. La última opción permitirá salir del programa.

3. [1.5 puntos]
  - a. Leer el contenido del archivo "usuarios.txt" y cargarlo en una lista de usuarios. El archivo "usuarios.txt" tendrá el siguiente formato:  
        Tipo\_usuario;Nombre\_usuario;Edad;ID;Atributo\_usuario  
  
        Donde "Tipo\_usuario" será una cadena con los valores "REGULAR", "VIP" o "ENTRENADOR", y "Atributo\_usuario1" será la fecha de inicio de la suscripción para los socios y el horario para los entrenadores. Asegúrate de manejar correctamente los tipos de datos y crear los objetos correspondientes.
  - b. Leer el contenido del archivo "clases.txt" y cargarlo en la lista de clases. Formato:  
        Nombre;Horario
4. [1 punto] Crea una excepción llamada `EntrenadorNoDisponibleException` que se lance al intentar asignar un entrenador a un socio VIP cuando el entrenador no tenga disponibilidad en su horario. Asegúrate de lanzar y atrapar la excepción correctamente.
5. [1 punto] Documenta de forma completa la clase `Usuario` utilizando los comentarios de documentación.

Además de las puntuaciones descritas anteriormente, se destinarán 1.5 puntos a la evaluación general del resto del programa (orden, formato y legibilidad del código). **Si el programa no compila, la calificación final del examen será directamente de 0.** Se valorará positivamente la corrección del diseño, la eficiencia del código y la ausencia de código repetido. En el escritorio del ordenador se encuentra el archivo `usuarios.txt` y `clases.txt` que se puede emplear para probar el código.

Durante el examen no habrá conexión a Internet y no se permitirá el uso de material de apoyo en ningún formato (electrónico o papel). El uso del teléfono móvil o similar supondrá la expulsión del examen y un suspenso directo.

Para realizar el examen, se podrá usar la aplicación de escritorio IntelliJ IDEA. Para entregar el examen, se guardarán todos los archivos y directorios que contengan el código del proyecto en el escritorio. **El nombre del archivo tendrá el siguiente formato: apellido1\_puesto.zip.**

Se recomienda comprobar con la profesora que el archivo se ha entregado correctamente antes de abandonar el laboratorio.