

Reglas del juego

- Tal como se le dijo antes, durante este examen usted no puede pedir ABSOLUTAMENTE nada prestado a sus compañeros, ni hablar con ellos.
- Su nombre y su firma a la aceptación del compromiso de no hacer fraude, en la hoja de respuestas a este examen, deben ir en lapicero. Si emplea más de una hoja márkelas TODAS de igual forma.
- Por ningún motivo puede salir del salón, antes de terminar el examen. De manera que si se retira se considerará que terminó su trabajo.
- Puede consultar sus notas, el libro o apuntes; pero se le recomienda que lo haga cuando esté seguro de qué quiere buscar, en caso contrario estas consultas se convierten en una distracción que le hace perder tiempo.
- No se resolverá ninguna duda durante el examen, así que si algo no le queda claro y usted requiere hacer algún tipo de supuesto, ESCRÍBALO y de una breve explicación de la razón por la cual lo supone. Tenga presente que un supuesto no puede contradecir lo que se le está diciendo.
- Debe escribir con letra clara.

La forma de evaluar este parcial es la siguiente:

Ítem	Valor
Parte teórica:	1.5
Ejercicio 1	0.3
Ejercicio 2	1.0
Ejercicio 3	0.2
Parte práctica:	3.5
Completar diagrama de clases	1.2
Análisis y código del método retornarOrdenados():?	0.8
Análisis y el código del método <u>+escribirBytes():void</u>	1.2
Análisis y código del método <u>+guardarObjetos():void</u>	0.3

PARTE TEÓRICA

1. (VALE 0.3) Teniendo en cuenta el diagrama de clases de la Figura 1.

Cuando se ejecuta la aplicación y se intenta adicionar un segundo jugador al equipo, se levanta una excepción. ¿Qué le corregiría al diagrama de clases para que funcione correctamente? Explique.

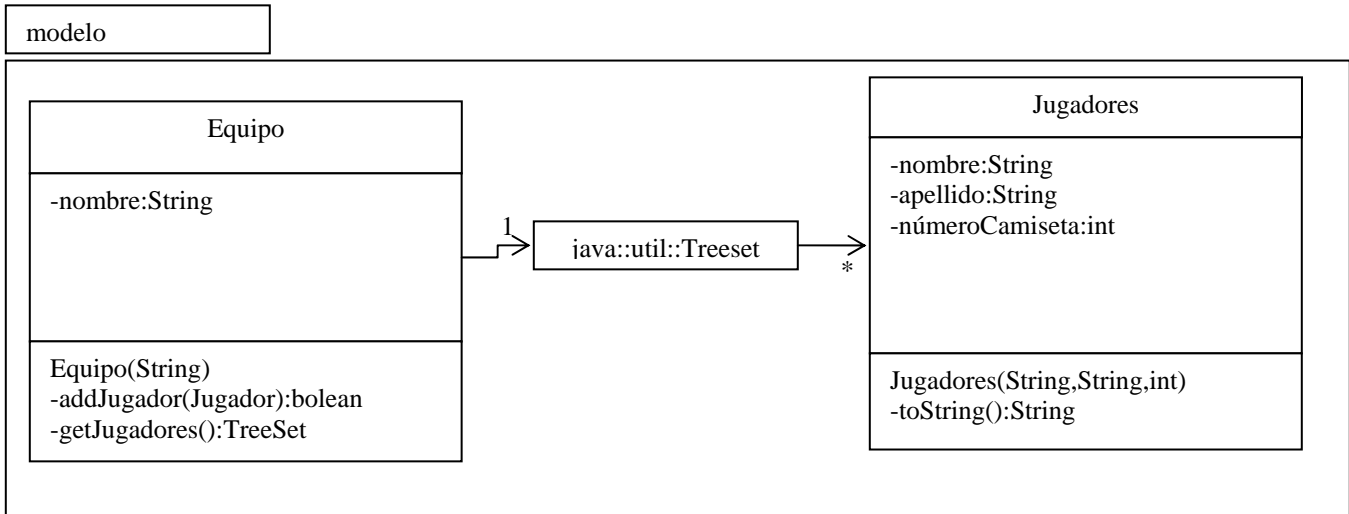
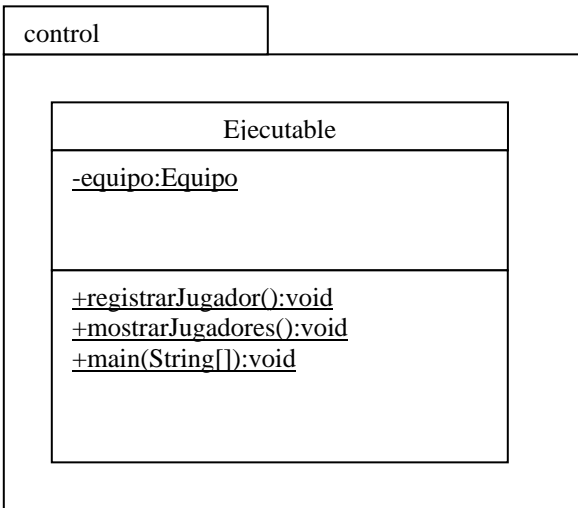


Figura 1 – Diagrama de clases Primer Punto



Código: _____ Nombre: _____

2. **(VALE 1.0)** Los organizadores de un concurso de caricaturas requieren una aplicación que permita consultar la caricatura presentada a través de la cédula del autor. Un ingeniero planteó una solución con el siguiente diagrama de clases:

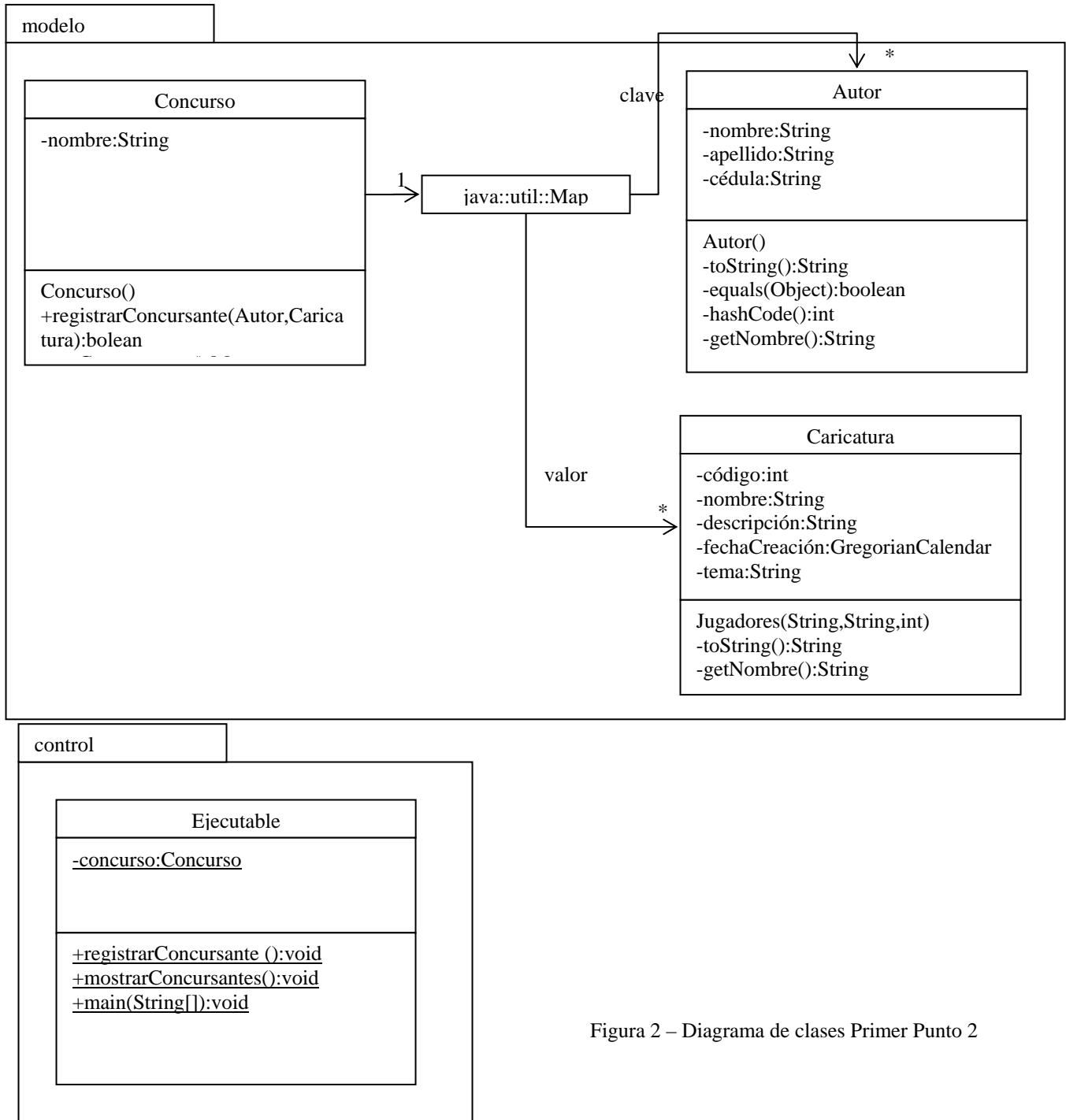


Figura 2 – Diagrama de clases Primer Punto 2

A continuación se presenta el código del método mostrarConcursantes, este método permite consultar los autores y sus caricaturas:

```

public static void mostrarConcursantes(){
    //pinta el título
    System.out.println("Concursantes");
    //se obtienen los concursantes
    Map mapConcursantes=concurso.getConcursantes();
    //se obtiene el iterador para recorrer el Map
    Iterator iterador = mapConcursantes.iterator();
    Autor autor;
    Caricatura caricatura;
    while (iterador.hasNext()) {
        //se obtiene la información para desplegarla
        autor=iterador.next();
        caricatura= iterador.next();
        System.out.println(autor.getNombre()+" "+ caricatura.getNombre());
    }
}
    
```

Sin embargo, este código no compila, por favor indique cuál es el problema y corríjalo.

Código: _____ Nombre: _____

3. **(VALE 0.2)** Cuál es el nombre de la clase que permite crear listas donde los elementos se almacenan de forma encadenada, es decir, cada elemento tiene una referencia a su antecesor y a su sucesor:
- a) AbstractList
 - b) *AbstractSequentialList*,
 - c) LinkedList
 - d) List
 - e) ArrayList

PARTE PRÁCTICA

La Universidad “El Tesoro del Saber”, está necesitando el desarrollo de una aplicación permita registrar la información de los aspirantes a las becas que ofrece para el pago de la matrícula de cada semestre.

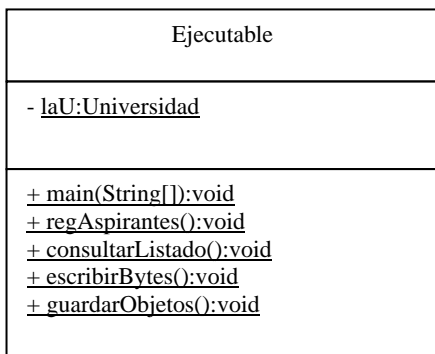
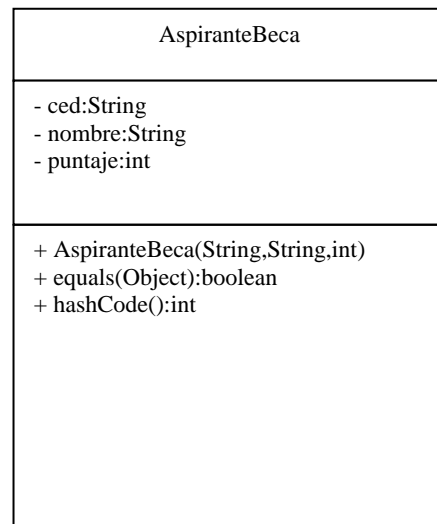
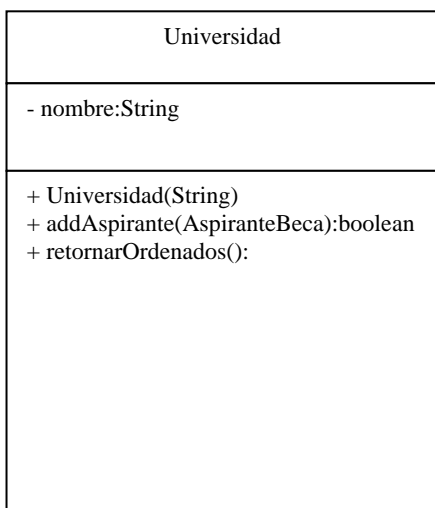
Se debe tener en cuenta que:

- Debe ser una aplicación orientada a objetos
- El diseño debe implementar el MVC (Modelo Vista Control)
- La interfaz de usuario será a través de la consola

El sistema debe permitir:

- **Registrar aspirantes.** De cada aspirante se registra su cédula (este es el identificador), el nombre completo y el puntaje que obtuvo en la prueba de selección (este es un dato de tipo int). No debe permitirse registrar más de un aspirante con el mismo número de cédula y deben almacenarse en el sistema ordenados por su número de identificación (cédula).
- **Consultar aspirantes ordenados por puntaje.** La aplicación debe desplegar un listado con el puntaje, la cédula y el nombre de cada aspirante.
- **Generar archivo de bytes con el listado de aspirantes.** Dado que hay otro sistema que requiere la información de los aspirantes que se registraron, debe ser posible la creación de un archivo de bytes que contenga el puntaje, la cédula y el nombre de cada aspirante.
- **Recuperar la información la próxima vez.** Cada vez que la aplicación a desarrollar se ejecute, debe poder recuperar la información de los objetos que hayan sido registrados anteriormente.

Un estudiante de Algoritmos y Lenguajes I de la Universidad “El Tesoro del Saber”, diseño el siguiente diagrama de clases en UML.



Código: _____ Nombre: _____

Usted debe hacer lo siguiente:

1. **(VALE 1.2)** Aplicar los conocimientos adquiridos en el curso de Algoritmos y Lenguajes II para completar el diagrama de clases de tal forma que permita desarrollar una aplicación que cumpla con los requerimientos.
2. **(VALE 0.8)** El estudiante de Algoritmos y Lenguajes I, pensó que en la clase Universidad debe existir un método al que llamó `+retornarOrdenados():?` Pero no sabe el tipo de dato que debe retornar, ni sabe como implementar el código. El estudiante sabe que el método debe retornar una colección con los **datos de los aspirantes ordenados de acuerdo al puntaje**.
 - a. Escriba el análisis **de este método**. (
 - b. Escriba el código Java del método **COMPLETO**.
3. **(VALE 1.2)** Realizar el **análisis** del método `+escribirBytes():void` de la clase Ejecutable y el **código Java**. Este es el método que guarda la información de los aspirantes, en un archivo de bytes que será procesado por otra aplicación.
4. **(VALE 0.3)** Realizar el **análisis** del método `+guardarObjetos():void` de la clase Ejecutable y el **código Java**. Este es el método que debe permitir guardar los objetos en un archivo para recuperarlos la próxima vez que inicie la aplicación.

Código: _____ Nombre: _____