**Antes de entregar el ejercicio revisar si cumple con la siguiente guía de correcciones:**

1) El ejercicio hace lo que se solicita. No incluye clases ni funcionalidades no solicitadas.

2) Si hay un diagrama UML de clase verificar que incluyen todas las clases indicadas con los tipos y alcance de los atributos, métodos indicados, relaciones con el nombre indicado en el diagrama, etc.

3) Si hay valores de variables que no cambian en la clase declararlas correctamente. Utilizar mayúscula. Usarla consistentemente en todo el código

3) Incluir en las clases que corresponden al modelo de la aplicación los gets para acceder a los atributos y los sets para modificarlos. Tanto en el constructor como en los sets se pueden validar los datos que se ingresan. No incluir gets ni sets para atributos que son de uso interno de la clase. No incluir sets que puedan dejar al objeto en un estado inconsistente.

4) Incluir el método toString para ver el estado de un objeto. No llamar a métodos que no corresponden a su estado. No mostrar atributos que son de uso interno de la clase. No llamar a toString, se llama implícitamente con el nombre del objeto.

5) Cuando se utilizan TADs declararlos en la clase e inicializarlos en el constructor. Lanzar la excepción que corresponda si al utilizarla puede fallar. (Por ejemplo si el índice de un arreglo supera la cantidad de elementos lanzar la excepción IndexOutOfBoundsException)

6) Respetar las convenciones utilizadas por los programadores de Java. Utilizar nombres que faciliten entender el código (sustantivos para clases, verbos para métodos, etc.) Los paquetes deben facilitar la organización de un proyecto.

7) Verificar el alcance de los atributos (por ahora los declaramos siempre private), los métodos y las clases. Mientras más restrictivos el código generado es más seguro y fácil de mantener. Las variables declararla lo más cercanas a donde se van a utilizar. (En la misma estructura, for (int i=0; …), dentro del método, etc.)

8) No utilizar entradas/salidas por pantallas (Por ejemplo: System.out.println(…) ) en las clases que corresponden al modelo o lógica de la aplicación. Solo las clases que interactúan con el usuario las pueden utilizar.

9) Respetar el siguiente orden al escribir una clase: después del nombre de la clase escribir los atributos, constructores, métodos y clases internas en ese orden. Identar el código.