



UNIVERSIDAD
DEL QUINDÍO®

Res.MEN 014915 - 02 AGO 2022
RENOVACIÓN ACREDITACIÓN

Problema:

Torneo deportivo - Requerimiento 01

Programación Orientada a Objetos

Área de programación

Programa de Ingeniería de Sistemas y Computación

Facultad de Ingeniería

UNIQUEINDÍO
en conexión territorial

www.uniquindio.edu.co





Problema:

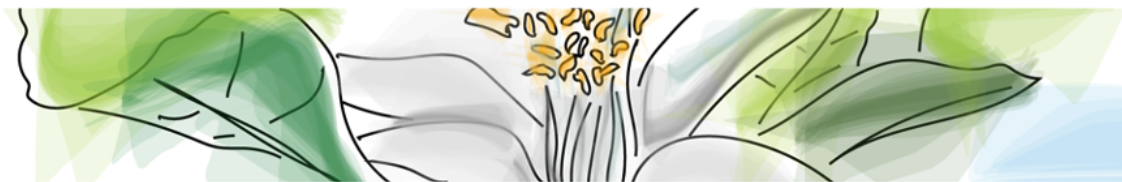
Un importante promotor deportivo le ha solicitado que diseñe un sistema para la gestión de las inscripciones a torneos deportivos. El necesita que la aplicación permita:



Problema:

RQ 01

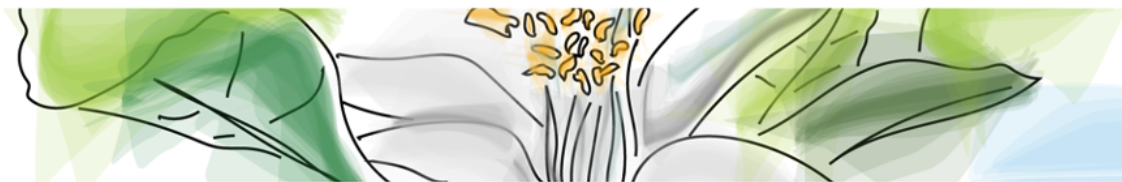
Almacenar la información de un torneo. La información que se requiere es: número máximo de equipos participantes, el nombre del torneo, el límite de edad de los jugadores (en caso de tenerlo), la fecha de inicio de las inscripciones, la fecha de cierre de las inscripciones, el valor de la inscripción y la fecha de inicio del torneo.



Problema:

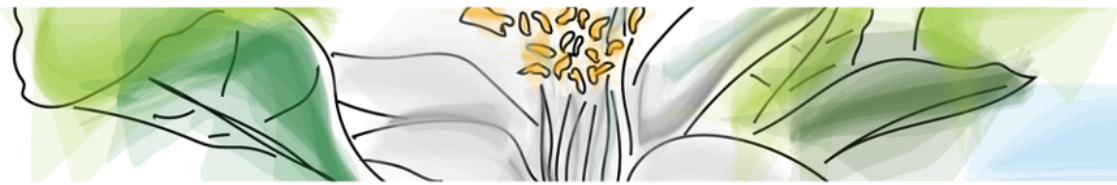
RQ 02

Debido a la planeación de los torneos en ocasiones es necesario modificar las fechas de inscripción e inicio del mismo. Por lo anterior, el promotor solicita que exista la posibilidad de poder modificar dichos datos despues de su creación.



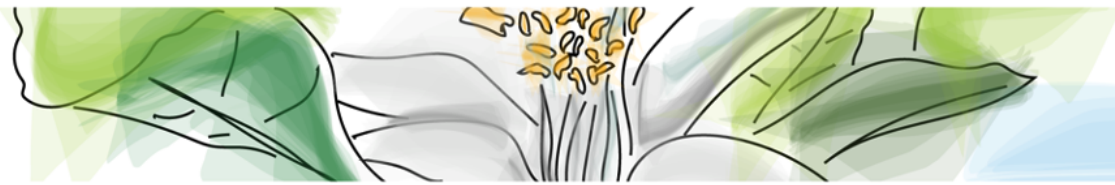
Abstracción: ¿Qué se solicita finalmente? (problema)

- Almacenar la información de un torneo
- Modificar fechas de inscripción e inicio del torneo



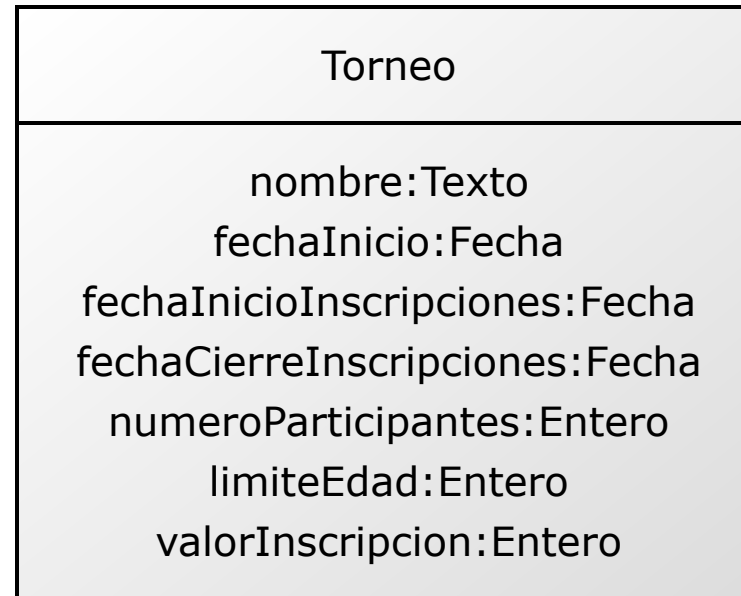
Abstracción: ¿Qué información es relevante dado el problema anterior?

- nombre del torneo : Texto
- fecha de inicio del torneo : Fecha
- fecha de inicio de las inscripciones : Fecha
- fecha de cierre de las inscripciones : Fecha
- número máximo de equipos participantes : Entero
- límite de edad de los jugadores (en caso de tenerlo) : Entero
- valor de la inscripción : Entero

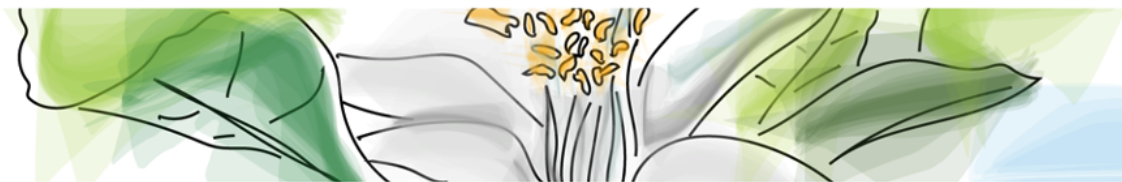


Abstracción: ¿Cómo se agrupa la información relevante?

- Torneo

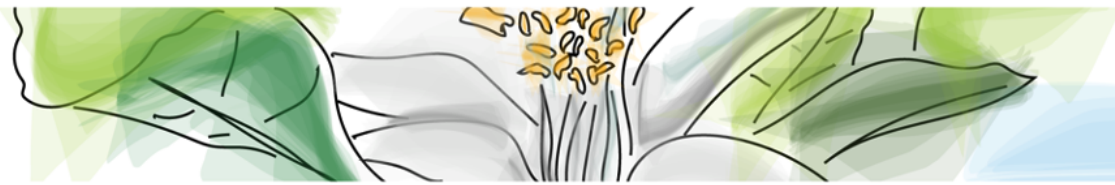


CREATED WITH YUML



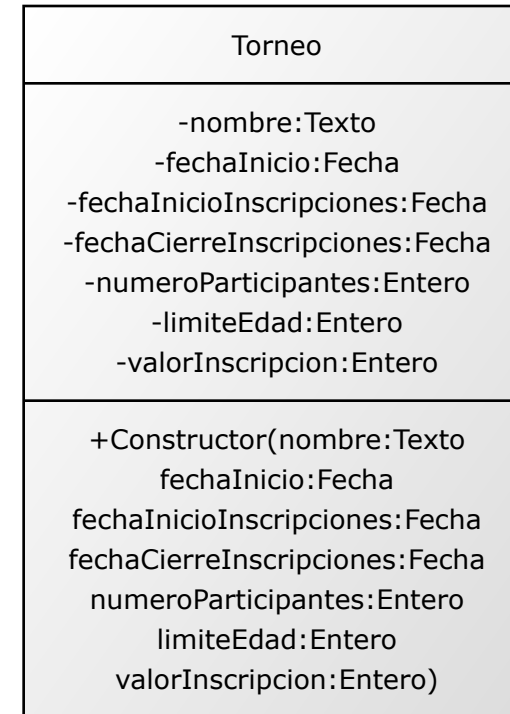
Abstracción: ¿Qué solicitan finalmente?

- Almacenar la información de un torneo
- Recuperar la información de un torneo
- Modificar las fechas de inscripción e inicio del torneo

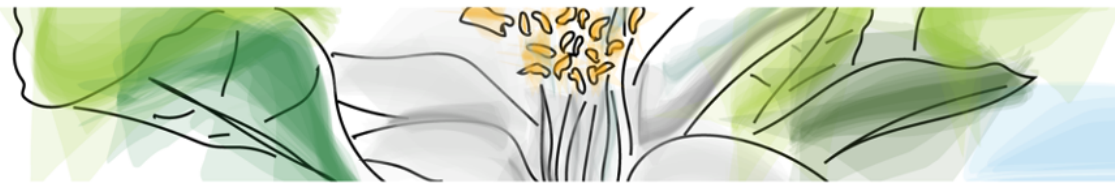


Descomposición: ¿Cómo se distribuyen las funcionalidades?

- Almacenar la información de un torneo



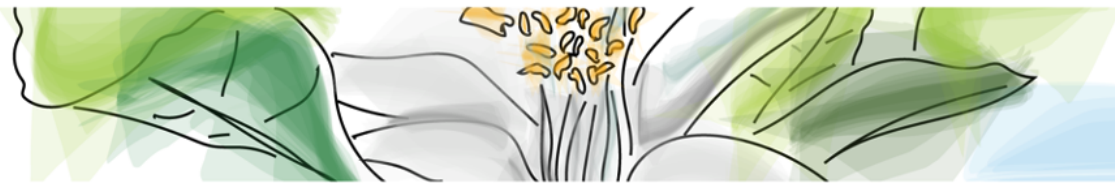
CREATED WITH YUML



Descomposición: ¿Cómo se distribuyen las funcionalidades?

- Recuperar la información de un torneo

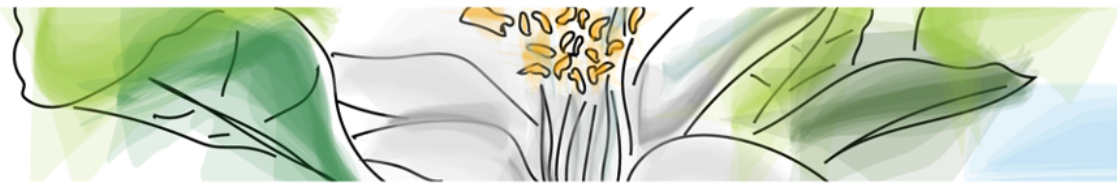
Torneo
<ul style="list-style-type: none">-nombre:Texto-fechaInicio:Fecha-fechaInicioInscripciones:Fecha-fechaCierreInscripciones:Fecha-numeroParticipantes:Entero-limiteEdad:Entero-valorInscripcion:Entero
<ul style="list-style-type: none">+Constructor(nombre:Texto fechaInicio:Fecha fechaInicioInscripciones:Fecha fechaCierreInscripciones:Fecha numeroParticipantes:Entero limiteEdad:Entero valorInscripcion:Entero)+getNombre():Texto+getFechaInicio():Fecha+getFechaInicioInscripciones():Fecha+getFechaCierreInscripciones():Fecha+getNumeroParticipantes():Entero+getLimiteEdad():Entero+getValorInscripcion():Entero



Descomposición: ¿Cómo se distribuyen las funcionalidades?

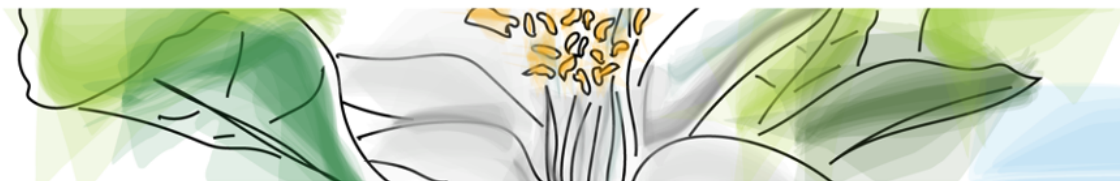
- Modificar las fechas de inscripción e inicio del torneo

Torneo
<ul style="list-style-type: none">-nombre: Texto-fechaInicio: Fecha-fechaInicioInscripciones: Fecha-fechaCierreInscripciones: Fecha-numeroParticipantes: Entero-limiteEdad: Entero-valorInscripcion: Entero
<ul style="list-style-type: none">+Constructor(nombre: Texto fechaInicio: Fecha fechaInicioInscripciones: Fecha fechaCierreInscripciones: Fecha numeroParticipantes: Entero limiteEdad: Entero valorInscripcion: Entero)+getNombre(): Texto+getFechaInicio(): Fecha+getFechaInicioInscripciones(): Fecha+getFechaCierreInscripciones(): Fecha+getNumeroParticipantes(): Entero+getLimiteEdad(): Entero+getValorInscripcion(): Entero+setFechaInicio(fechaInicio: Fecha)+setFechaInicioInscripciones(fechaInicioInscripciones: Fecha)+setFechaCierreInscripciones(fechaCierreInscripciones: Fecha)



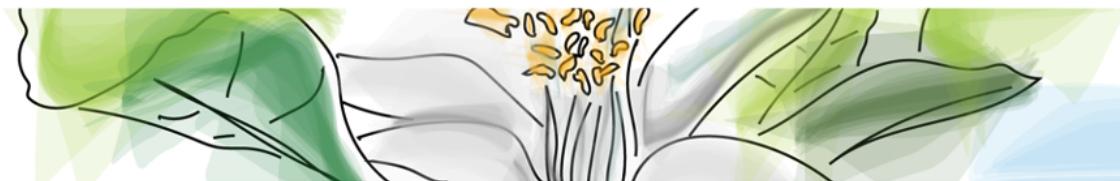
Descomposición: ¿Qué debo hacer para probar las funcionalidades?

- Almacenar datos de prueba
- Recuperar los datos de prueba
- Verificar que los datos almacenados coinciden con los datos recuperados
- Modificar datos de la prueba
- Recuperar datos de la prueba
- Verificar que los datos almacenados y modificados coinciden con los datos recuperados



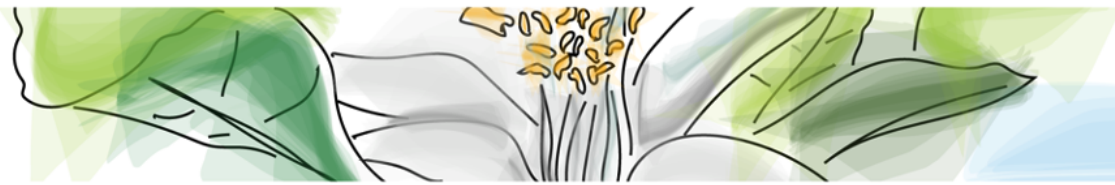
Descomposición: ¿Qué debo hacer para probar las funcionalidades?

Prueba	Entrada de datos	Salida (Resultado)
Datos Completos	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0	Torneo creado con los datos proporcionados Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0
Con datos nulos	null null null null 24 0 0	Error, faltan datos para la creación del torneo
Participantes negativos	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 -24 0 0	Error: porque no se puede crear un torneo con número negativo de participantes
Limite edad negativo	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 -1 0	Error: porque no se puede crear un torneo con número negativo en el límite de edad
Inscripción negativa	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 -1	Error: porque no se puede crear un torneo con número negativo en la inscripción
Fechas Inscripción tardías	Copa Mundo 2023-10-01 2023-11-01 2023-11-15 24 0 0	Error: Las inscripciones no pueden ser posteriores al inicio del torneo
Cierre inscripción anterior al inicio	Copa Mundo 2023-10-01 2023-08-15 2023-08-01 24 0 0	Error: El cierre de las inscripciones no puede ser anterior al inicio de las inscripciones.



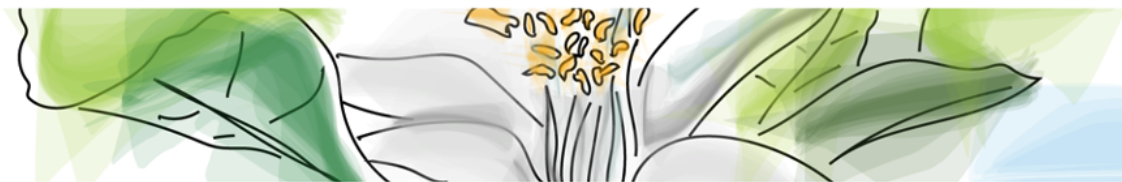
Descomposición: ¿Qué debo hacer para probar las funcionalidades?

Prueba	Entrada de datos	Salida (Resultado)
Modificar fecha inicio, valido	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Nueva fecha de inicio 2023-10-12	Torneo creado con los datos proporcionados Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Torneo con los datos modificados Copa Mundo 2023-10-12 2023-08-01 2023-09-15 24 0 0
Modificar fecha inicio, con null	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Nueva fecha de inicio null	Torneo creado con los datos proporcionados Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Error la fecha de inicio no puede ser null
Modificar fecha inicio, con fecha de inicio anterior a las inscripciones	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Nueva fecha de inicio 2023-07-01	Torneo creado con los datos proporcionados Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Error la fecha de inicio es anterior a las inscripciones
Modificar fecha inicio de inscripciones, valido	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Nueva fecha de inicio de inscripciones 2023-08-10	Torneo creado con los datos proporcionados Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Torneo con los datos modificados Copa Mundo 2023-10-01 2023-08-10 2023-09-15 24 0 0
Modificar fecha inicio de inscripciones con null	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Nueva fecha de inicio de inscripciones null	Torneo creado con los datos proporcionados Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Error la fecha de inicio de inscripciones no puede ser null



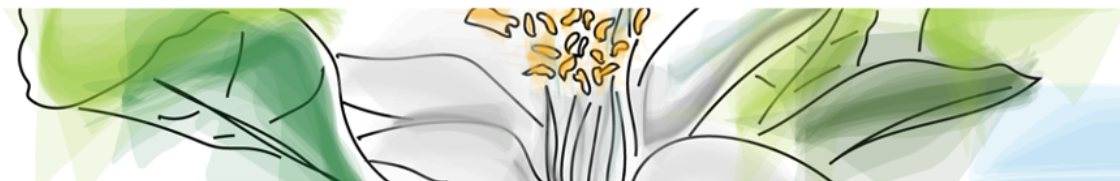
Descomposición: ¿Qué debo hacer para probar las funcionalidades?

Prueba	Entrada de datos	Salida (Resultado)
Modificar fecha inicio de inscripciones con fecha de inicio posterior a la fecha de cierre de inscripciones	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Nueva fecha de inicio de inscripciones 2023-09-16	Torneo creado con los datos proporcionados Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Error la fecha de inicio de inscripciones no puede ser posterior a la fecha de cierre
Modificar fecha de cierre de inscripciones, valido	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Nueva fecha de cierre de inscripciones 2023-09-16	Torneo creado con los datos proporcionados Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Torneo con los datos modificados Copa Mundo 2023-10-01 2023-08-10 2023-09-16 24 0 0
Modificar fecha de cierre de inscripciones con null	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Nueva fecha de cierre de inscripciones null	Torneo creado con los datos proporcionados Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Error la fecha de cierre de inscripciones no puede ser null
Modificar fecha de cierre de inscripciones con fecha posterior a la fecha de inicio del torneo	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Nueva fecha de cierre de inscripciones 2023-10-02	Torneo creado con los datos proporcionados Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Error la fecha de cierre de inscripciones no puede ser posterior a la fecha de inicio del torneo
Modificar fecha de cierre de inscripciones con fecha anterior a la fecha de inicio de inscripciones	Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Nueva fecha de cierre de inscripciones 2023-07-30	Torneo creado con los datos proporcionados Copa Mundo 2023-10-01 2023-08-01 2023-09-15 24 0 0 Error la fecha de cierre de inscripciones no puede ser anterior a la fecha de inicio de inscripciones



Reconocimiento de patrones: ¿Qué puede reutilizar?

- No aplica




Codificación:

Usando una clase

Constructor con todos los atributos + métodos gets y sets

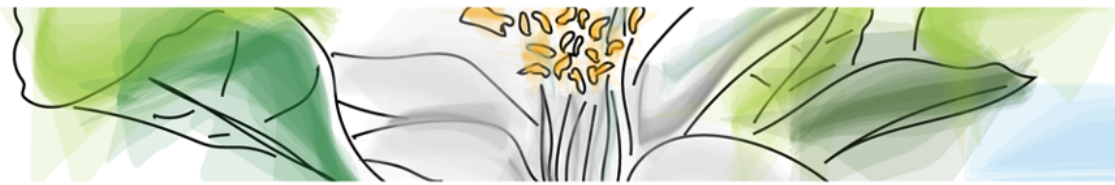
Torneo
<ul style="list-style-type: none">-nombre:Texto-fechaInicio:Fecha-fechaInicioInscripciones:Fecha-fechaCierreInscripciones:Fecha-numeroParticipantes:Entero-limiteEdad:Entero-valorInscripcion:Entero
<ul style="list-style-type: none">+Constructor(nombre:Texto fechaInicio:Fecha fechaInicioInscripciones:Fecha fechaCierreInscripciones:Fecha numeroParticipantes:Entero limiteEdad:Entero valorInscripcion:Entero)+getNombre():Texto+getFechaInicio():Fecha+getFechaInicioInscripciones():Fecha+getFechaCierreInscripciones():Fecha+getNumeroParticipantes():Entero+getLimiteEdad():Entero+getValorInscripcion():Entero+setFechaInicio(fechaInicio:Fecha)+setFechaInicioInscripciones(fechaInicioInscripciones:Fecha)+setFechaCierreInscripciones(fechaCierreInscripciones:Fecha)



Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Clase para probar el funcionamiento del Torneo
 * @author Área de programación UQ
 * @since 2023-08
 *
 * Licencia GNU/GPL V3.0 (https://www.gnu.org/licenses/lgpl-3.0.html#license-text)
 */
package co.edu.uniquindio.poo.torneodeportivo;
import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.junit.jupiter.api.Assertions.assertTrue;
import java.time.LocalDate;
import java.util.logging.Logger;
import org.junit.jupiter.api.Test;

public class TorneoTest {
    /**
     * Instacia para el manejo de logs
     */
    private static final Logger LOG = Logger.getLogger(TorneoTest.class.getName());
}
```




Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo almacene y recupere los datos
 *
 */
@Test
public void datosCompleto() {
    LOG.info("Inicio de prueba datos completos...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-08-01|2023-09-15|24|0|0
    // ERROR: Esta línea falla porque no existe Torneo
    Torneo torneo = new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1), LocalDate.of(2023, 8, 1), LocalDate.of(2023, 9, 15), (byte)24, (byte)0, 0);

    // Recuperación y verificación de datos
    assertEquals("Copa Mundo", torneo.nombre());
    assertEquals(LocalDate.of(2023, 10, 1), torneo.fechaInicio());
    assertEquals(LocalDate.of(2023, 8, 1), torneo.fechaInicioInscripciones());
    assertEquals(LocalDate.of(2023, 9, 15), torneo.fechaCierreInscripciones());
    assertEquals((byte)24, torneo.numeroParticipantes());
    assertEquals((byte)0, torneo.limiteEdad());
    assertEquals(0, torneo.valorInscripcion());

    LOG.info("Fin de prueba datos completos...");
}
```



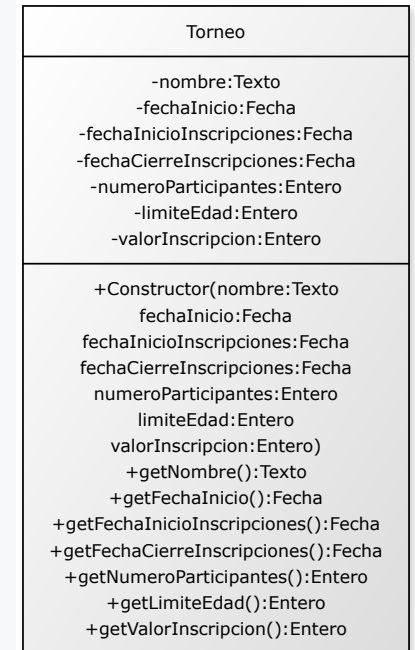
Codificación: ¿Cómo escribo la solución en JAVA?

```
package co.edu.uniquindio.poo.torneodeportivo;

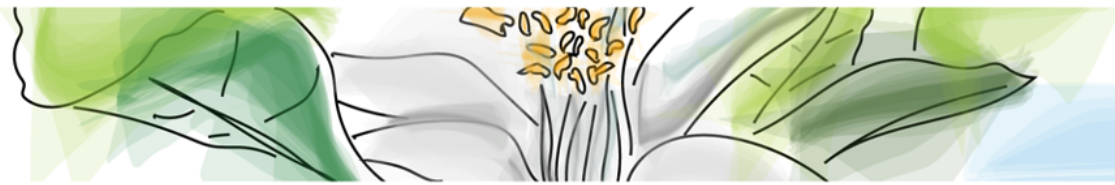
import java.time.LocalDate;

public record Torneo(String nombre,
                    LocalDate fechaInicio,
                    LocalDate fechaInicioInscripciones,
                    LocalDate fechaCierreInscripciones,
                    byte numeroParticipantes,
                    byte limiteEdad,
                    int valorInscripcion) {

}
```



CREATED WITH YUML




Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo almacene y recupere los datos
 *
 */
@Test
public void datosCompletos() {
    LOG.info("Inicio de prueba datos completos...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-08-01|2023-09-15|24|0|0
    // ERROR: Esta línea falla porque no existe Torneo
    Torneo torneo = new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1), LocalDate.of(2023, 8, 1), LocalDate.of(2023, 9, 15), (byte)24, (byte)0, 0);

    // Recuperación y verificación de datos
    assertEquals("Copa Mundo", torneo.nombre());
    assertEquals(LocalDate.of(2023, 10, 1), torneo.fechaInicio());
    assertEquals(LocalDate.of(2023, 8, 1), torneo.fechaInicioInscripciones());
    assertEquals(LocalDate.of(2023, 9, 15), torneo.fechaCierreInscripciones());
    assertEquals((byte)24, torneo.numeroParticipantes());
    assertEquals((byte)0, torneo.limiteEdad());
    assertEquals(0, torneo.valorInscripcion());

    LOG.info("Fin de prueba datos completos...");
}
```

```
✓ [I] torneo-deportivo 28ms
✓ [I] {} co.edu.uniquindio.poo.torneodeportivo 28ms
✓ [I] TorneoTest 28ms
  ✓ [I] datosCompletos() 28ms
```



Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo valide que se ingrese los datos
 *
 */
@Test
public void datosNulos() {
    LOG.info("Inicio de prueba datos nulos...");
    // Almacenar los datos de prueba null|null|null|null|24|0|0
    assertThrows(Throwable.class, ()-> new Torneo(null, null, null, null, (byte)24, (byte)0, 0));

    LOG.info("Fin de prueba datos nulos...");
}
```

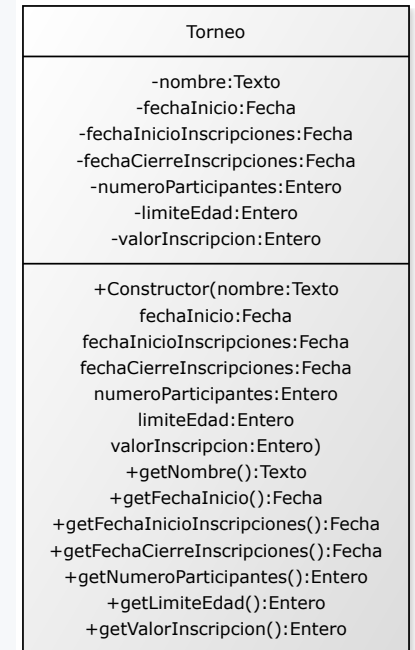
```
✓ [x] torneo-deportivo 34ms
✓ [x] {} co.edu.uniquindio.poo.torneodeportivo 34ms
✓ [x] TorneoTest 34ms
  ✓ [x] datosCompletos() 28ms
  [x] datosNulos() 6.0ms
```


Codificación: ¿Cómo escribo la solución en JAVA?


```
import java.time.LocalDate;

public record Torneo(String nombre,
                    LocalDate fechaInicio,
                    LocalDate fechaInicioInscripciones,
                    LocalDate fechaCierreInscripciones,
                    byte numeroParticipantes,
                    byte limiteEdad,
                    int valorInscripcion) {

    public Torneo{
        assert nombre != null;
        assert fechaInicio != null;
        assert fechaInicioInscripciones != null;
        assert fechaCierreInscripciones != null;
    }
}
```



CREATED WITH YUML



Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo valide que se ingrese los datos
 *
 */
@Test
public void datosNulos() {
    LOG.info("Inicio de prueba datos nulos...");
    // Almacenar los datos de prueba null|null|null|null|24|0|0
    assertThrows(Throwable.class, ()-> new Torneo(null, null, null, null, (byte)24, (byte)0, 0));

    LOG.info("Fin de prueba datos nulos...");
}
```

```
✓ [t] torneo-deportivo 30ms
✓ [c] {} co.edu.uniquindio.poo.torneodeportivo 30
✓ [t] TorneoTest 30ms
  ✓ [p] datosCompletos() 29ms
  ✓ [p] datosNulos() 1.0ms
```

Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo valide que el ingreso de número
 * de participantes negativo
 */
@Test
public void participantesNegativos() {
    LOG.info("Inicio de prueba número de participantes negativo...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-08-01|2023-09-15|-24|0|0
    assertThrows(Throwable.class, ()-> new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1),
        LocalDate.of(2023, 8, 01), LocalDate.of(2023, 10, 15), (byte)-24, (byte)0, 0));

    LOG.info("Fin de prueba número de participantes negativo...");
}
```

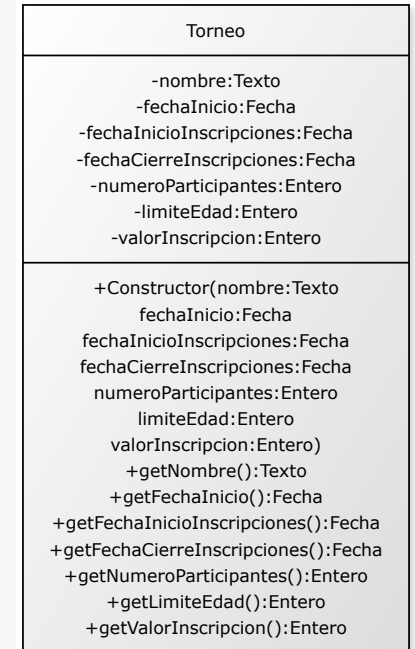
```
✓ [X] torneo-deportivo 34ms
✓ [X] {} co.edu.uniquindio.poo.torneodeportivo 34ms
✓ [X] TorneoTest 34ms
  ✓ [X] datosCompleto() 28ms
  ✓ [X] datosNulos() 0.0ms
  ✓ [X] participantesNegativos() 6.0ms
```

Codificación: ¿Cómo escribo la solución en JAVA?

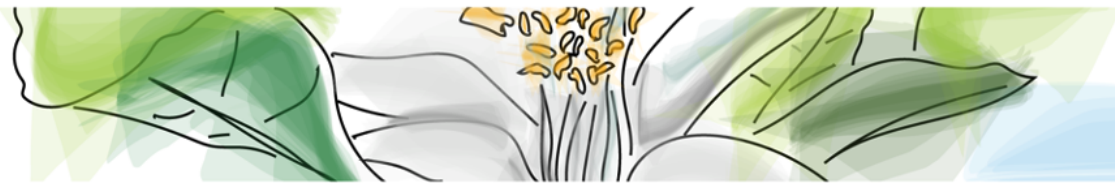
```
import java.time.LocalDate;

public record Torneo(String nombre,
                    LocalDate fechaInicio,
                    LocalDate fechaInicioInscripciones,
                    LocalDate fechaCierreInscripciones,
                    byte numeroParticipantes,
                    byte limiteEdad,
                    int valorInscripcion) {

    public Torneo{
        assert nombre != null;
        assert fechaInicio != null;
        assert fechaInicioInscripciones != null;
        assert fechaCierreInscripciones != null;
        assert numeroParticipantes >= 0;
    }
}
```



CREATED WITH YUML

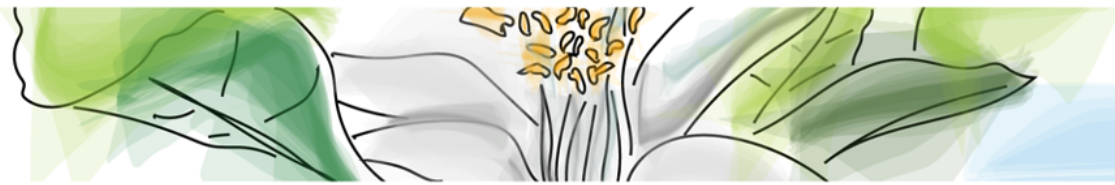


Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo valide que el ingreso de número
 * de participantes negativo
 */
@Test
public void participantesNegativos() {
    LOG.info("Inicio de prueba número de participantes negativo...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-08-01|2023-09-15|-24|0|0
    assertThrows(Throwable.class, ()-> new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1),
        LocalDate.of(2023, 8, 01), LocalDate.of(2023, 10, 15), (byte)-24, (byte)0, 0));

    LOG.info("Fin de prueba número de participantes negativo...");
}
```

```
✓ [I] torneo-deportivo 31ms
✓ {} co.edu.uniquindio.poo.torneodeportivo 31r
✓ [I] TorneoTest 31ms
  ✓ [I] datosCompleto() 29ms
  ✓ [I] datosNulos() 1.0ms
  ✓ [I] participantesNegativos() 1.0ms
```




Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo valide que el
 * ingreso de limites de edades negativo
 *
 */
@Test
public void limiteEdadesNegativo() {
    LOG.info("Inicio de prueba limites de edades negativo...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-08-01|2023-09-15|24|-1|0
    assertThrows(Throwable.class, ()-> new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1),
        LocalDate.of(2023, 8, 01), LocalDate.of(2023, 10, 15), (byte)24, (byte)-1, 0));

    LOG.info("Fin de prueba limites de edades negativo...");
}
```

```
✓ [X] torneo-deportivo 37ms
✓ [X] {} co.edu.uniquindio.poo.torneodeportivo 37ms
✓ [X] TorneoTest 37ms
  ✓ [X] datosCompleto() 29ms
  ✓ [X] datosNulos() 1.0ms
  ✓ [X] participantesNegativos() 1.0ms
  ✓ [X] limiteEdadesNegativo() 6.0ms
```

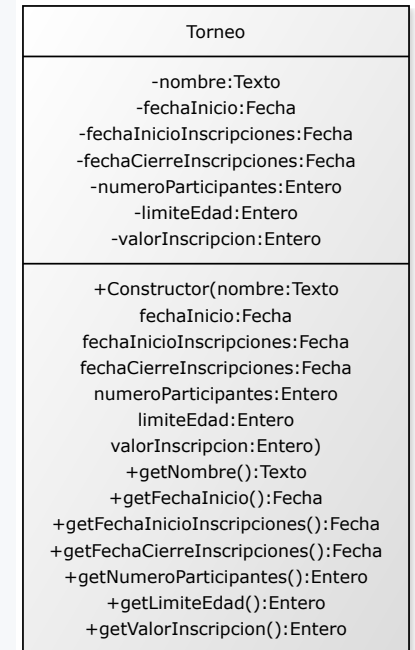



Codificación: ¿Cómo escribo la solución en JAVA?


```
import java.time.LocalDate;

public record Torneo(String nombre,
                    LocalDate fechaInicio,
                    LocalDate fechaInicioInscripciones,
                    LocalDate fechaCierreInscripciones,
                    byte numeroParticipantes,
                    byte limiteEdad,
                    int valorInscripcion) {

    public Torneo{
        assert nombre != null;
        assert fechaInicio != null;
        assert fechaInicioInscripciones != null;
        assert fechaCierreInscripciones != null;
        assert numeroParticipantes >= 0;
        assert limiteEdad >= 0;
    }
}
```



CREATED WITH YUML




Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo valide que el
 * ingreso de limites de edades negativo
 */
@Test
public void limiteEdadesNegativo() {
    LOG.info("Inicio de prueba limites de edades negativo...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-08-01|2023-09-15|24|-1|0
    assertThrows(Throwable.class, ()-> new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1),
        LocalDate.of(2023, 8, 01), LocalDate.of(2023, 10, 15), (byte)24, (byte)-1, 0));

    LOG.info("Fin de prueba limites de edades negativo...");
}
```

```
✓ [I] torneo-deportivo 36ms
✓ [I] {} co.edu.uniquindio.poo.torneodeportivo 36
✓ [I] TorneoTest 36ms
  ✓ [I] datosCompleto() 30ms
  ✓ [I] datosNulos() 1.0ms
  ✓ [I] participantesNegativos() 3.0ms
  ✓ [I] limiteEdadesNegativo() 2.0ms
```



Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo valide que el
 * ingreso de valor de inscripción negativa
 */
@Test
public void inscripcionNegativa() {
    LOG.info("Inicio de prueba inscripción negativa...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-08-01|2023-09-15|24|0|-1
    assertThrows(Throwable.class, ()-> new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1),
        LocalDate.of(2023, 8, 01), LocalDate.of(2023, 10, 15), (byte)24, (byte)0, -1));

    LOG.info("Fin de prueba inscripción negativa...");
}
```

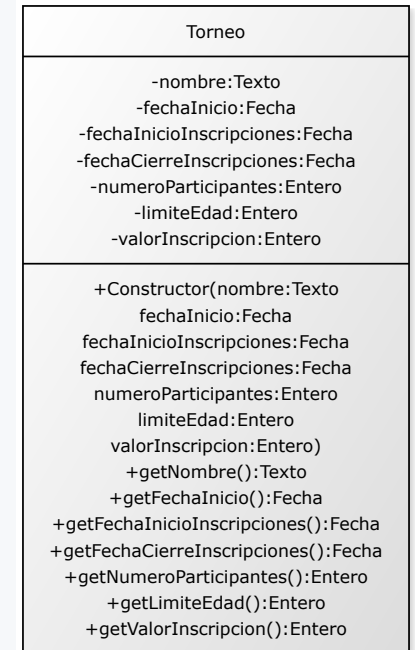
```
✓ [x] torneo-deportivo 48ms
✓ [x] {} co.edu.uniquindio.poo.torneodeportivo 48ms
✓ [x] TorneoTest 48ms
  ✓ [x] datosCompleto() 35ms
  ✓ [x] datosNulos() 0.0ms
  ✓ [x] participantesNegativos() 2.0ms
  ✓ [x] limiteEdadesNegativo() 2.0ms
  ✓ [x] inscripcionNegativa() 9.0ms
```

Codificación: ¿Cómo escribo la solución en JAVA?


```
import java.time.LocalDate;

public record Torneo(String nombre,
                    LocalDate fechaInicio,
                    LocalDate fechaInicioInscripciones,
                    LocalDate fechaCierreInscripciones,
                    byte numeroParticipantes,
                    byte limiteEdad,
                    int valorInscripcion) {

    public Torneo{
        assert nombre != null;
        assert fechaInicio != null;
        assert fechaInicioInscripciones != null;
        assert fechaCierreInscripciones != null;
        assert numeroParticipantes >= 0;
        assert limiteEdad >= 0;
        assert valorInscripcion >= 0;
    }
}
```



CREATED WITH YUML




Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo valide que el
 * ingreso de valor de inscripción negativa
 */
@Test
public void inscripcionNegativa() {
    LOG.info("Inicio de prueba inscripción negativa...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-08-01|2023-09-15|24|0|-1
    assertThrows(Throwable.class, ()-> new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1),
        LocalDate.of(2023, 8, 01), LocalDate.of(2023, 10, 15), (byte)24, (byte)0, -1));

    LOG.info("Fin de prueba inscripción negativa...");
}
```

```
✓ [I] torneo-deportivo 32ms
✓ [I] {} co.edu.uniquindio.poo.torneodeportivo 32ms
✓ [I] TorneoTest 32ms
  ✓ [I] datosCompleto() 28ms
  ✓ [I] datosNulos() 1.0ms
  ✓ [I] participantesNegativos() 1.0ms
  ✓ [I] limiteEdadesNegativo() 1.0ms
  ✓ [I] inscripcionNegativa() 1.0ms
```



Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo valide que el
 * ingreso de inscripciones posteriores a la fecha de inicio del torneo
 *
 */
@Test
public void inscripcionTardia() {
    LOG.info("Inicio de prueba inscripción tardia...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-11-01|2023-11-15|24|0|0
    assertThrows(Throwable.class, ()-> new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1),
        LocalDate.of(2023, 11, 01), LocalDate.of(2023, 11, 15), (byte)24, (byte)0, 0));

    LOG.info("Fin de prueba inscripción tardia...");
}
```



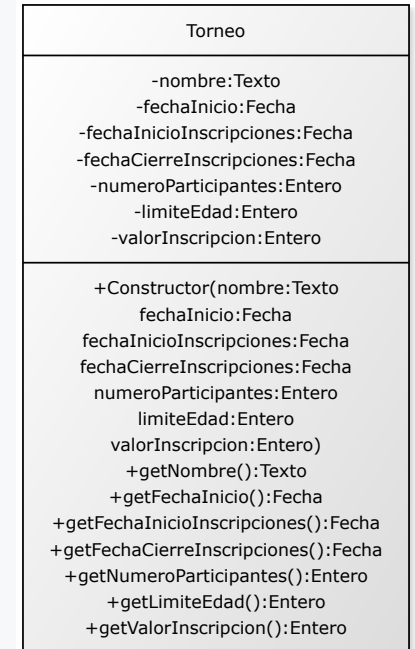
torneo-deportivo	43ms
co.edu.uniquindio.poo.torneodeportivo	43ms
TorneoTest	43ms
datosCompleto()	30ms
datosNulos()	2.0ms
participantesNegativos()	1.0ms
limiteEdadesNegativo()	1.0ms
inscripcionNegativa()	0.0ms
inscripcionTardia()	9.0ms

Codificación: ¿Cómo escribo la solución en JAVA?


```
import java.time.LocalDate;

public record Torneo(String nombre,
                    LocalDate fechaInicio,
                    LocalDate fechaInicioInscripciones,
                    LocalDate fechaCierreInscripciones,
                    byte numeroParticipantes,
                    byte limiteEdad,
                    int valorInscripcion) {

    public Torneo{
        assert nombre != null;
        assert fechaInicio != null;
        assert fechaInicioInscripciones != null;
        assert fechaCierreInscripciones != null;
        assert numeroParticipantes >= 0;
        assert limiteEdad >= 0;
        assert valorInscripcion >= 0;
        assert fechaInicio.isAfter(fechaInicioInscripciones) &&
            fechaInicio.isAfter(fechaCierreInscripciones);
    }
}
```




CREATED WITH YUML




Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo valide que el
 * ingreso de inscripciones posteriores a la fecha de inicio del torneo
 *
 */
@Test
public void inscripcionTardia() {
    LOG.info("Inicio de prueba inscripción tardia...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-11-01|2023-11-15|24|0|0
    assertThrows(Throwable.class, ()-> new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1),
        LocalDate.of(2023, 11, 01), LocalDate.of(2023, 11, 15), (byte)24, (byte)0, 0));

    LOG.info("Fin de prueba inscripción tardia...");
}
```



✓	torneo-deportivo	36ms
✓	{ } co.edu.uniquindio.poo.torneodeportivo	36ms
✓	TorneoTest	36ms
✓	datosCompleto()	29ms
✓	datosNulos()	1.0ms
✓	participantesNegativos()	2.0ms
✓	limiteEdadesNegativo()	2.0ms
✓	inscripcionNegativa()	1.0ms
✓	inscripcionTardia()	1.0ms



Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo valide que el ingreso
 * de inicio inscripciones posteriores a la fecha
 * de cierre de inscripciones
 *
 */
@Test
public void cierreInscripcionAnteriorInicio() {
    LOG.info("Inicio de prueba Cierre inscripción anterior al inicio...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-11-01|2023-11-15|24|0|0
    assertThrows(Throwable.class, ()-> new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1),
        LocalDate.of(2023, 8, 15), LocalDate.of(2023, 8, 1), (byte)24, (byte)0, 0));

    LOG.info("Fin de prueba Cierre inscripción anterior al inicio...");
}
```



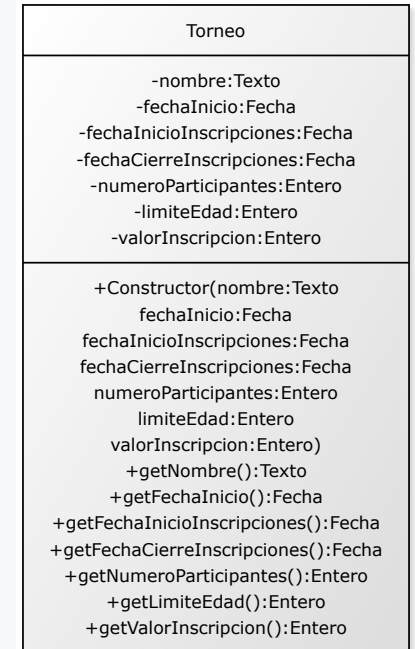
torneo-deportivo	49ms
co.edu.uniquindio.poo.torneodeportivo	49
TorneoTest	49ms
datosCompletos()	32ms
datosNulos()	0.0ms
participantesNegativos()	2.0ms
limiteEdadesNegativo()	2.0ms
inscripcionNegativa()	2.0ms
inscripcionTardia()	3.0ms
cierreInscripcionAnteriorInicio()	8.0ms

Codificación: ¿Cómo escribo la solución en JAVA?


```
import java.time.LocalDate;

public record Torneo(String nombre,
                    LocalDate fechaInicio,
                    LocalDate fechaInicioInscripciones,
                    LocalDate fechaCierreInscripciones,
                    byte numeroParticipantes,
                    byte limiteEdad,
                    int valorInscripcion) {

    public Torneo{
        assert nombre != null;
        assert fechaInicio != null;
        assert fechaInicioInscripciones != null;
        assert fechaCierreInscripciones != null;
        assert numeroParticipantes >= 0;
        assert limiteEdad >= 0;
        assert valorInscripcion >= 0;
        assert fechaInicio.isAfter(fechaInicioInscripciones) &&
            fechaInicio.isAfter(fechaCierreInscripciones);
        assert fechaCierreInscripciones.isAfter(fechaInicioInscripciones);
    }
}
```




CREATED WITH YUML




Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo valide que el ingreso
 * de inicio inscripciones posteriores a la fecha
 * de cierre de inscripciones
 *
 */
@Test
public void cierreInscripcionAnteriorInicio() {
    LOG.info("Inicio de prueba Cierre inscripción anterior al inicio...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-11-01|2023-11-15|24|0|0
    assertThrows(Throwable.class, ()-> new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1),
        LocalDate.of(2023, 8, 15), LocalDate.of(2023, 8, 1), (byte)24, (byte)0, 0));

    LOG.info("Fin de prueba Cierre inscripción anterior al inicio...");
}
```



torneo-deportivo	51ms
co.edu.uniquindio.poo.torneodeportivo	51ms
TorneoTest	51ms
datosCompletos()	38ms
datosNulos()	4.0ms
participantesNegativos()	3.0ms
limiteEdadesNegativo()	1.0ms
inscripcionNegativa()	2.0ms
inscripcionTardia()	1.0ms
cierreInscripcionAnteriorInicio()	2.0ms



Codificación: ¿Cómo pruebo la solución en JAVA?

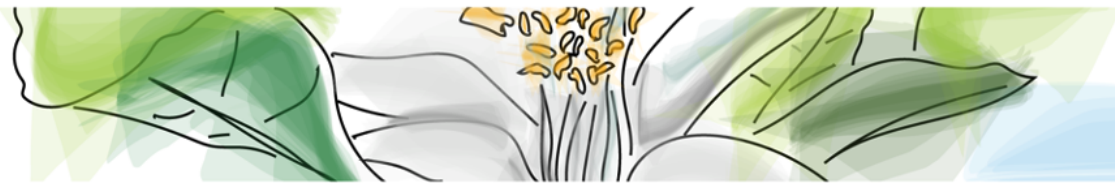
```
* Clase para probar la modificación de fechas en el Torneo
* @author Área de programación UQ
* @since 2023-08
*
* Licencia GNU/GPL V3.0 (https://raw.githubusercontent.com/grid-uq/poo/main/LICENSE)
*/
package co.edu.uniquindio.poo.torneodeportivo;

import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.junit.jupiter.api.Assertions.assertThrows;

import java.time.LocalDate;
import java.util.logging.Logger;

import org.junit.jupiter.api.Test;

public class TorneoModificarFechasTest {
    /**
     * Instancia para el manejo de logs
     */
    private static final Logger LOG = Logger.getLogger(TorneoModificarFechasTest.class.getName());
}
```



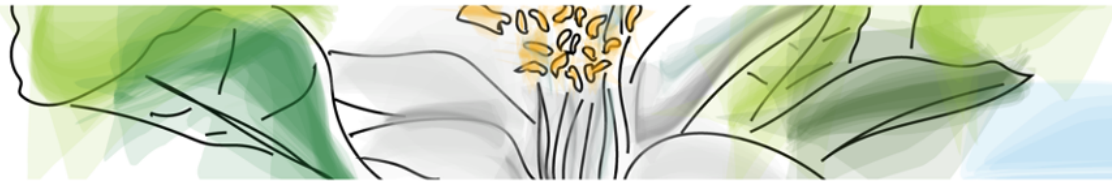
Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo permita la modificación de la fecha de inicio
 *
 */
@Test
public void modificarFechaInicio() {
    LOG.info("Inicio de prueba modificar fecha de inicio valida...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-08-01|2023-09-15|24|0|0
    Torneo torneo = new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1), LocalDate.of(2023, 8, 1), LocalDate.of(2023, 9, 15), (byte)24, (byte)0, 0);

    // Modificación de la fecha
    // Error: el método setFechaInicio no existe
    torneo.setFechaInicio(LocalDate.of(2023, 10, 12));

    assertEquals(LocalDate.of(2023, 10, 12), torneo.fechaInicio());

    LOG.info("Fin de prueba modificar fecha de inicio valida...");
}
```

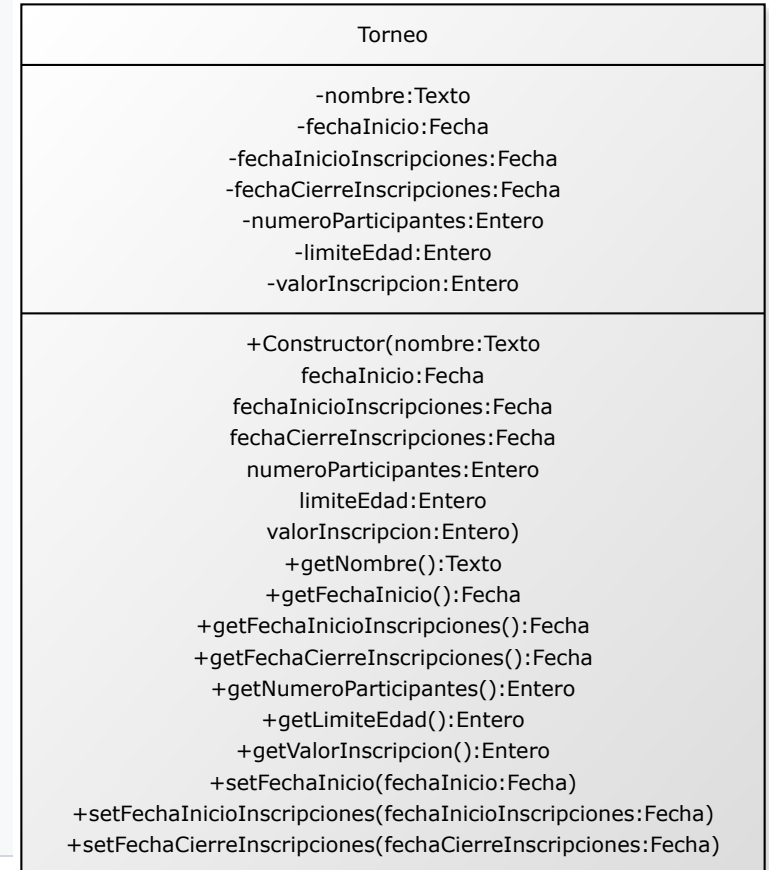


Codificación: ¿Cómo escribo la solución en JAVA?

```
package co.edu.uniquindio.poo.torneodeportivo;

import java.time.LocalDate;

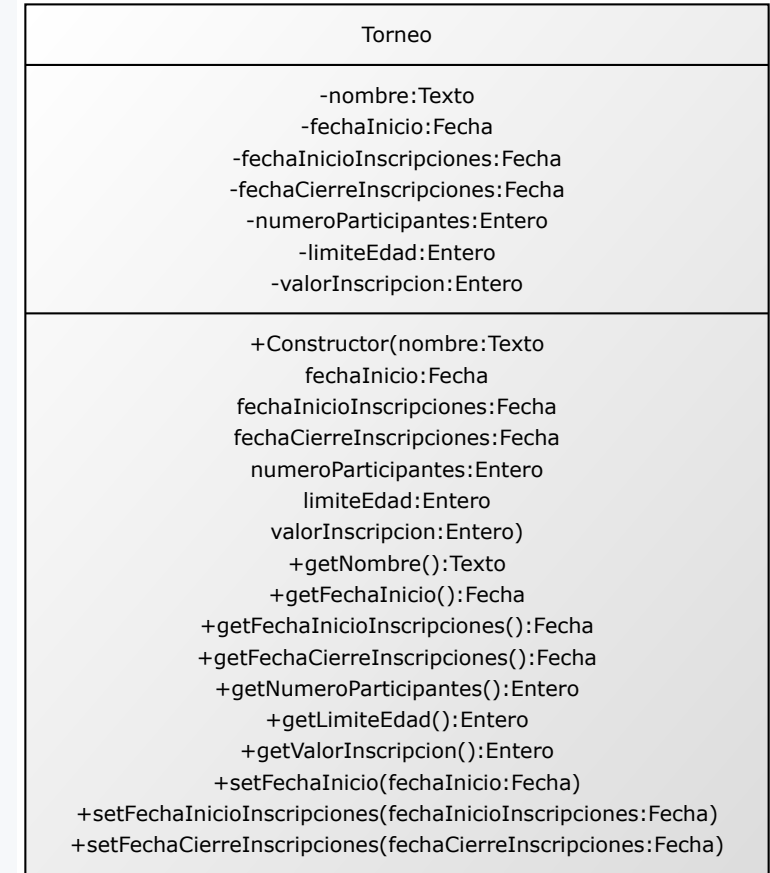
public class Torneo {
    private final String nombre;
    private LocalDate fechaInicio;
    private LocalDate fechaInicioInscripciones;
    private LocalDate fechaCierreInscripciones;
    private final byte numeroParticipantes;
    private final byte limiteEdad;
    private final int valorInscripcion;
}
```



CREATED WITH YUML

Codificación: ¿Cómo escribo la solución en JAVA?

```
public Torneo(String nombre, LocalDate fechaInicio,
               LocalDate fechaInicioInscripciones,
               LocalDate fechaCierreInscripciones,
               byte numeroParticipantes,
               byte limiteEdad, int valorInscripcion) {
    assert nombre != null;
    assert fechaInicio != null;
    assert fechaInicioInscripciones != null;
    assert fechaCierreInscripciones != null;
    assert numeroParticipantes >= 0;
    assert limiteEdad >= 0;
    assert valorInscripcion >= 0;
    assert fechaInicio.isAfter(fechaInicioInscripciones) &&
           fechaInicio.isAfter(fechaCierreInscripciones);
    assert fechaCierreInscripciones
           .isAfter(fechaInicioInscripciones);
    this.nombre = nombre;
    this.fechaInicio = fechaInicio;
    this.fechaInicioInscripciones = fechaInicioInscripciones;
    this.fechaCierreInscripciones = fechaCierreInscripciones;
    this.numeroParticipantes = numeroParticipantes;
    this.limiteEdad = limiteEdad;
    this.valorInscripcion = valorInscripcion;
}
```



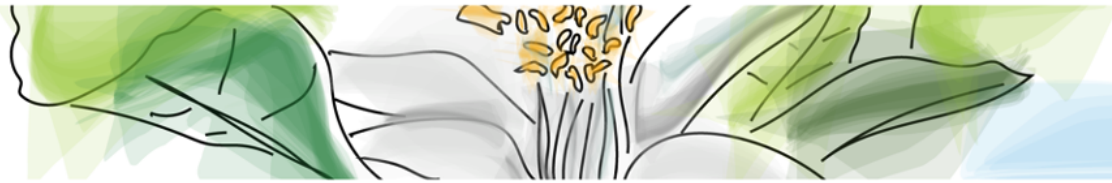
CREATED WITH YUML

Codificación: ¿Cómo escribo la solución en JAVA?

```
public String getNombre() {  
    return nombre;  
}  
  
public LocalDate getFechaInicio() {  
    return fechaInicio;  
}  
  
public LocalDate getFechaInicioInscripciones() {  
    return fechaInicioInscripciones;  
}  
  
public LocalDate getFechaCierreInscripciones() {  
    return fechaCierreInscripciones;  
}  
  
public byte getNumeroParticipantes() {  
    return numeroParticipantes;  
}  
  
public byte getLimiteEdad() {  
    return limiteEdad;  
}  
  
public int getValorInscripcion() {  
    return valorInscripcion;  
}  
  
public void setFechaInicio(LocalDate fechaInicio) {  
    this.fechaInicio = fechaInicio;  
}
```

Torneo
<ul style="list-style-type: none">-nombre:Texto-fechaInicio:Fecha-fechaInicioInscripciones:Fecha-fechaCierreInscripciones:Fecha-numeroParticipantes:Entero-limiteEdad:Entero-valorInscripcion:Entero
<ul style="list-style-type: none">+Constructor(nombre:Texto fechaInicio:Fecha fechaInicioInscripciones:Fecha fechaCierreInscripciones:Fecha numeroParticipantes:Entero limiteEdad:Entero valorInscripcion:Entero)+getNombre():Texto+getFechaInicio():Fecha+getFechaInicioInscripciones():Fecha+getFechaCierreInscripciones():Fecha+getNumeroParticipantes():Entero+getLimiteEdad():Entero+getValorInscripcion():Entero+setFechaInicio(fechaInicio:Fecha)+setFechaInicioInscripciones(fechaInicioInscripciones:Fecha)+setFechaCierreInscripciones(fechaCierreInscripciones:Fecha)

CREATED WITH YUML



Codificación: ¿Cómo pruebo la solución en JAVA?

```
/**
 * Verificar que la clase Torneo permita la modificación de la fecha de inicio
 *
 */
@Test
public void modificarFechaInicio() {
    LOG.info("Inicio de prueba modificar fecha de inicio valida...");
    // Almacenar los datos de prueba Copa Mundo|2023-10-01|2023-08-01|2023-09-15|24|0|0
    Torneo torneo = new Torneo("Copa Mundo", LocalDate.of(2023, 10, 1), LocalDate.of(2023, 8, 1), LocalDate.of(2023, 9, 15), (byte)24, (byte)0, 0);

    // Modificación de la fecha
    // Error: el método setFechaInicio no existe
    torneo.setFechaInicio(LocalDate.of(2023, 10, 12));

    assertEquals(LocalDate.of(2023, 10, 12), torneo.fechaInicio());

    LOG.info("Fin de prueba modificar fecha de inicio valida...");
}
```

```
✓ [I] torneo-deportivo 726ms
✓ {} co.edu.uniquindio.poo.torne...
✓ [I] TorneoModificarFechasTest...
  ✓ [I] modificarFechaInicio() 1.0...
> [I] TorneoTest 725ms
```

