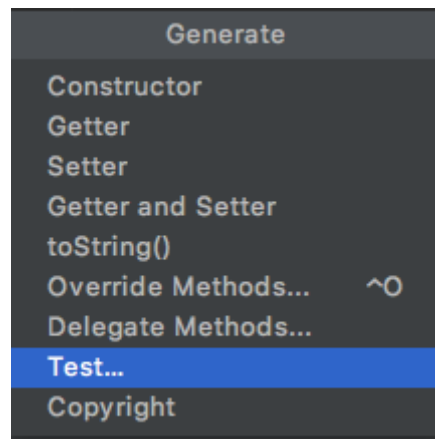


Indice

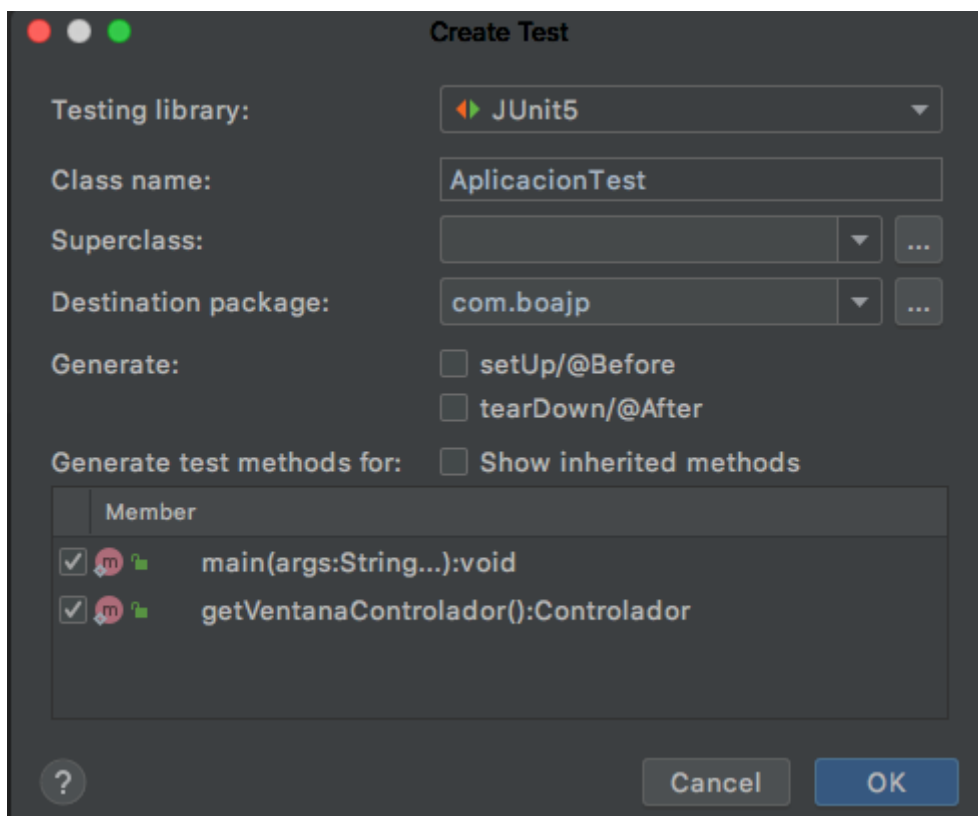
Apartado 1. Como generar un Test.....	2
Apartado 2. Pruebas unitarias.....	3
Prueba 1. Insertar Temporada.....	3
Prueba 2. Modificar Temporada.....	4
Prueba 3. Modificar Temporada (Error planeado).....	5
Prueba 4. Iniciar Sesion.....	6
Prueba 5. Iniciar Sesion (Error planeado).....	7
Prueba 6. Registrarse.....	8

Apartado 1. Como generar un Test

Para generar una clase Test, hacemos click derecho en la clase que queremos probar y cliclamos test



Luego seleccionamos las funciones que queremos probar



Apartado 2. Pruebas unitarias

Prueba 1. Insertar Temporada

Probamos la funcion con la cual nos conectamos a la base de datos e insertamos una temporada nueva.

```
no usages new *
@Test
public void insertarTemporada() throws Exception {
    TemporadasServicio temporadasServicio = new TemporadasServicio();
    temporadasServicio.anadirTemporada((short) 2023, LocalDate.of( year: 2023, month: 06, dayOfMonth: 12),LocalDate.of( year: 2023, month: 06, dayOfMonth: 23));
}
}
```

porada x

Tests passed: 1 of 1 test – 5 sec 365 ms

Como podemos observar el test es exitoso.

```
✓ Tests passed: 1 of 1 test – 4 sec 283 ms
```

Prueba 2. Modificar Temporada

```
no usages new *
@Test
public void modificarTemporada() throws Exception {
    TemporadasServicio temporadasServicio = new TemporadasServicio();
    TemporadaEntidad temporadaEntidad = new TemporadaEntidad((short) 2023, LocalDate.of( year: 2023, month: 86, dayOfMonth: 12), LocalDate.of( year: 2023, month: 86, dayOfMonth: 12));
    temporadasServicio.modificarTemporada(temporadaEntidad);
}

orada <
Tests passed: 1 of 1 test - 4 sec 150 ms
```

Prueba 3. Modificar Temporada (Error planeado)

En este caso repetimos la prueba anterior, pero insertamos la fecha de fin de inscripción antes que la fecha de inicio de inscripción proposito para que el test de un resultado erroneo

```
no usages new *
@Test
public void modificarTemporada2() throws Exception {
    TemporadasServicio temporadasServicio = new TemporadasServicio();
    TemporadaEntidad temporadaEntidad = new TemporadaEntidad((short) 2023, LocalDate.of( year: 2023, month: 86, dayOfMonth: 12), LocalDate.of( year: 2023, month: 86, dayOfMonth: 10));
    temporadasServicio.modificarTemporada(temporadaEntidad);
}
```



Prueba 4. Iniciar Sesión

```
no usages new
@Test
public void pruebaIniciarSesion(){
    try {
        IniciarUsuarioServicio iniciarUsuarioServicio = new IniciarUsuarioServicio();
        iniciarUsuarioServicio.encontrarUsuario( nombreDeUsuario: "bea");
        iniciarUsuarioServicio.iniciarUsuario("bea12345".toCharArray());

    }catch (Exception e){
        JOptionPane.showMessageDialog( parentComponent: null, e.getMessage());
    }
}
```

✓ Tests passed: 1 of 1 test – 4 sec 283 ms

Prueba 5. Iniciar Sesión (Error planeado)

Insertamos un usuario que no existe en la función de iniciar sesión.

```
no usages new *  
@Test  
public void pruebaIniciarSesion() throws Exception {  
  
    IniciarUsuarioServicio iniciarUsuarioServicio = new IniciarUsuarioServicio();  
    iniciarUsuarioServicio.encontrarUsuario( nombreDeUsuario: "noexiste");  
    iniciarUsuarioServicio.iniciarUsuario("kl".toCharArray());  
}  
}
```

❗ Tests failed: 1 of 1 test – 4 sec 573 ms

Prueba 6. Registrarse

```
no usages new *  
@Test  
public void registro() throws Exception {  
    RegistrarUsuarioServicio registrarUsuarioServicio = new RegistrarUsuarioServicio();  
    registrarUsuarioServicio.registrarUsuario( usuario: "pepe", email: "pepe@gmail.com", "pepe12345".toCharArray());  
}  
}
```

✓ Tests passed: 1 of 1 test – 4 sec 158 ms