Índice

[1. ANÁLISIS DE PRUEBAS 1](#_Toc523485520)

[1.1. ALCANCE DE LAS PRUEBAS 1](#_Toc523485521)

[1.2. ENTORNO DE PRUEBAS 1](#_Toc523485522)

[2. DISEÑO DE PRUEBAS 2](#_Toc523485523)

[2.1. PRUEBAS UNITARIAS 2](#_Toc523485524)

[2.2. PRUEBAS DE INTEGRACIÓN 3](#_Toc523485525)

[2.3. PRUEBAS DE IMPLANTACIÓN 5](#_Toc523485526)

# ANÁLISIS DE PRUEBAS

### ALCANCE DE LAS PRUEBAS

Serán requeridos los siguientes niveles de pruebas:

* **Pruebas Unitarias**: Pruebas unitarias sobre la capa de servicio, que es la que implementa la lógica de negocio, de manera independiente para cada caso de uso. Se deberá dar énfasis en aquellos que tengan relaciones con sistemas externos. Las clases y los métodos que serán objetos de pruebas se definirán en la fase de diseño. Las pruebas se llevarán a cabo usando en framework de Spring Boot Test y Junit.
* **Pruebas de Integración**: Pruebas de integración sobre el conjunto de clases y paquetes que forman el sistema, desde la capa de vista hasta la de persistencia, para comprobar que funcionan correctamente juntos e identificar errores de entrada y salida de datos, así como asegurar la integridad de que los valores introducidos en la base de datos y escritos en los ficheros. Estas pruebas se llevarán a cabo de forma manual, debido a la complejidad del proceso de negocio y en número reducido de casos de uso.
* **Pruebas de Implantación:** Pruebas que comprueben que el rendimiento y la seguridad de la aplicación una vez desplegada en el entorno de producción.

### ENTORNO DE PRUEBAS

Para el entorno de las pruebas, se empleará el mismo entorno que para el desarrollo.

# DISEÑO DE PRUEBAS

### PRUEBAS UNITARIAS

Las pruebas unitarias se realizarán usando las herramientas proporcionadas por Spring Boot Test y jUnit. El proceso se construirá para que, de manera automática, se lance una vez y se ejecuten todas las pruebas

#### Alcance de las pruebas unitarias

Se testearán los métodos incluidos en la capa de servicio, ya que es la que aloja la lógica de negocio de la aplicación. Para ello, se han definido tres categorías en las que clasificar los métodos de las clases de esta capa, en función de su complejidad, para aplicar el tipo de testeo que mejor se ajuste:

|  |  |  |
| --- | --- | --- |
| **Categoría de complejidad** | **Tipo de testeo** | **Justificación** |
| Alta | Manual | El método realiza varias escrituras en la base de datos y en ficheros. Un testeo manual, usando una herramienta de depuración, permite ejecutar paso a paso el procedimiento y comprobar los valores de las variables que se están escribiendo en tiempo real, lo que permite una granularidad más fina en el control que una prueba de caja negra. |
| Media | Unitario | Métodos que aplican alguna transformación además de acceder a los datos. Con pruebas unitarias se puede comprobar rápidamente que el valor devuelto o escrito es el correcto. |
| Trivial | Ninguno | Métodos que solo escriben o leen la capa de datos. Como estos accesos a datos no se han implementado, sino que se han usado los proporcionados por Spring, no es necesario su testeo. |

#### Definición de las pruebas unitarias

Como la validación de las entradas se realizan en la capa Controladora, la mayoría de los métodos de la capa de servicio no devuelven fallos. La validación de las entradas erróneas se testeará en las pruebas de integración.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CountryService** | | | | |
| **Identificador** | **Método** | **Tipo de testeo** | **Resultado esperado** | **Resultado obtenido** |
| PR-01 | assembleCountry | Unitario | OK | OK |
| PR-02 | mapCountry | Unitario | OK | OK |
| PR-03 | validateCountryFile | Unitario | OK | OK |
| PR-04 | validateCountryFile | Unitario | 2 errores | 2 errores |
| PR-05 | demapCountry | Unitario | OK | OK |
| **GameService** | | | | |
| **Identificador** | **Método** | **Tipo de testeo** | **Resultado esperado** | **Resultado obtenido** |
| PR-06 | createGame | Unitario | OK | OK |
| PR-07 | findOpenGames | Manual | OK | OK |
| PR-08 | findClosedGames | Manual | OK | OK |
| PR-09 | nextTurn | Unitario | OK | OK |
| PR-10 | getChartData | Manual | OK | OK |
| **ScenarioService** | | | | |
| **Identificador** | **Método** | **Tipo de testeo** | **Resultado esperado** | **Resultado obtenido** |
| PR-11 | createScenario | Unitario | OK | OK |
| PR-12 | mapScenario | Unitario | OK | OK |
| PR-13 | validateScenarioFile | Unitario | OK | OK |
| PR-14 | validateScenarioFile | Unitario | 2 errores | 2 errores |
| **TurnService** | | | | |
| **Identificador** | **Método** | **Tipo de testeo** | **Resultado esperado** | **Resultado obtenido** |
| PR-15 | generateTurns | Unitario | OK | OK |
| **UserService** | | | | |
| **Identificador** | **Método** | **Tipo de testeo** | **Resultado esperado** | **Resultado obtenido** |
| PR-16 | register | Unitario | OK | OK |
| PR-17 | register | Unitario | Excepción | Excepción |
| PR-18 | findLogin | Unitario | OK | OK |
| PR-19 | findLogin | Unitario | NULL | NULL |
| PR-20 | findByNickname | Unitario | OK | OK |
| PR-21 | findByNickname | Unitario | NULL | NULL |
| **WarService** | | | | |
| **Identificador** | **Método** | **Tipo de testeo** | **Resultado esperado** | **Resultado obtenido** |
| PR-22 | copyMap | Unitario | OK | OK |
| PR-23 | roll | Manual | OK | OK |

### PRUEBAS DE INTEGRACIÓN

Las pruebas de integración se llevarán a cabo en la aplicación directamente, simulando el comportamiento de un usuario. Se probará cada caso de uso, o conjunto de casos de uso en caso de que varios constituyan un único proceso de negocio, introduciendo entradas erróneas y válidas y comprobando los resultados.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identificador** | **Proceso de negocio** | **Casos de uso** | **Resultado esperado** | **Resultado obtenido** |
| PI-01 | Login | CU01 | OK | OK |
| PI-02 | Login | CU01 | Error por credenciales incorrectas | Error por credenciales incorrectas |
| PI-03 | Registro | CU02 | OK | OK |
| PI-04 | Registro | CU02 | Error por usuario ya existente | Error por usuario ya existente |
| PI-05 | Registro | CU02 | Error por entradas no válidas | Error por entradas no válidas |
| PI-06 | Modificar Perfil | CU04 | OK | OK |
| PI-07 | Modificar Perfil | CU04 | Error por entradas no válidas | Error por entradas no válidas |
| PI-08 | Ver partida | CU05, CU06, CU07 | OK | OK |
| PI-09 | Crear escenario | CU08 | OK | OK |
| PI-10 | Crear escenario | CU08 | Errores en fichero | Errores de fichero |
| PI-11 | Crear partida | CU09, CU10, CU11 | OK | OK |
| PI-12 | Crear partida | CU09, CU10, CU11 | Error por escenario no seleccionado | Error por escenario no seleccionado |
| PI-13 | Crear partida | CU09, CU10, CU11 | Error por jugador inexistente | Error por jugador inexistente |
| PI-14 | Crear partida | CU09, CU10, CU11 | Error por fichero de jugador | Error por fichero de jugador |
| PI-15 | Cerrar partida | CU12 | Error por contraseña incorrecta | Error por contraseña incorrecta |
| PI-16 | Cerrar partida | CU12 | OK | OK |
| PI-17 | Configurar coaliciones | CU13 | OK | OK |
| PI-18 | Resolver tirada | CU14 | Error por coalición vacía | Error por coalición vacía |
| PI-19 | Resolver tirada | CU14 | Error por más del 100% de los recursos | Error por más del 100% de los recursos |
| PI-20 | Resolver tirada | CU14 | Error por país no reconocido | Error por país no reconocido |
| PI-21 | Resolver tirada | CU14 | Error por escenario no reconocido | Error por escenario no reconocido |
| PI-22 | Terminar enfrentamiento | CU15 | OK | OK |
| PI-23 | Avanzar turno | CU16 | OK | OK |
| PI-24 | Actualizar partida | CU17 | OK | OK |

### PRUEBAS DE IMPLANTACIÓN

Para llevar a cabo estas pruebas, se repetirán las pruebas unitarias y de integración en el hosting final. Las unitarias se llevan a cabo de manera automática cuando la aplicación Spring Boot se arranca.