<Hatim Mezouar El Mejdoubi>

< >

BattleShip

*CLASS USER*

**Funcionalitat**: Comprobamos que el método SetNameUser() asigne el nombre que le pasamos de forma correcta y que lo pasa a la función getNameUser(). Utilizamos pruebas de caja negra.

**Localització**:

* **Arxiu:** User.java
* **Classe:** User
* **mètode desenvolupat:** public void setNameUser(String nameUser)

**Test**:

* **Arxiu:** UserTest.java
* **Classe:** UserTest
* **Mètode de test associat a la funcionalitat:** public void testGetSetNameUser()

**Funcionalitat**: Comprobamos que el método SetCoordinates() asigne los valores que le pasamos de forma correcta y que lo pasa a la función getRow(), getCol(), getOrientacion(), getTypeBoat(). Utilizamos pruebas de caja negra con las técnicas de particiones equivalentes y valores límites y frontera. También utilizamos pruebas de caja blanca con las técnicas de Decision Coverage y Condicion Coverage.

**Localització**:

* **Arxiu:** User.java
* **Classe:** User
* **mètode desenvolupat:** public void setCoordinates(int row, int col, int orientacion, int typeBoat)

**Test**:

* **Arxiu:** UserTest.java
* **Classe:** UserTest
* **Mètode de test associat a la funcionalitat:** public void testGetSetcoordinates()

*CLASS BOARD*

**Funcionalitat**: Comprobamos que se pueda colocar los diferentes barcos en las coordenadas introducidas. Utilizamos pruebas de caja negra con las técnicas de particiones equivalentes y valores límites y frontera**.**

**Localització**:

* **Arxiu:** Board.java
* **Classe:** Board
* **mètode desenvolupat:** public int setBoat(int initX, int initY, int orientacion, int boatType)

**Test**:

* **Arxiu:** BoardTest.java
* **Classe:** BoarTest
* **Mètode de test associat a la funcionalitat:** public void testSetBoats()

**Funcionalitat**: Realizamos disparos para comprobar si es agua, fallido o hemos dado un barco. Utilizamos pruebas de caja negra con las técnicas de particiones equivalentes y valores límites y frontera**.**

**Localització**:

* **Arxiu:** Board.java
* **Classe:** Board
* **mètode desenvolupat:** public boolean shotBoat(int x, int y)

**Test**:

* **Arxiu:** BoardTest.java
* **Classe:** BoardTest
* **Mètode de test associat a la funcionalitat:** public void testShootBoat()

**Funcionalitat**: Comprobamos que si se ha ganado la partida. Utilizamos pruebas de caja negra.

**Localització**:

* **Arxiu:** Board.java
* **Classe:** Board
* **mètode desenvolupat:** public boolean checkWinner()

**Test**:

* **Arxiu:** BoardTest.java
* **Classe:** Board
* **Mètode de test associat a la funcionalitat:** public void testCheckWinner()

*CLASS BOAT*

**Funcionalitat**: Comprobamos que los diferentes barcos puedan colocarse en las coordenadas introducidas. Utilizamos pruebas de caja negra con las técnicas de particiones equivalentes y valores límites y frontera**.**

**Localització**:

* **Arxiu:** Boat.java
* **Classe:** Boat
* **mètode desenvolupat:** public boolean checkRule(int x, int y, int orientacion, int[][] board)

**Test**:

* **Arxiu:** BoatTest.java
* **Classe:** BoatTest
* **Mètode de test associat a la funcionalitat:** public void testCheckRule()

**Funcionalitat**: Comprobamos que en las coordenadas el espacio esta disponible. Utilizamos pruebas de caja negra.

**Localització**:

* **Arxiu:** Boat.java
* **Classe:** Boat
* **mètode desenvolupat:** public boolean freeSpace(int x, int y, int orientacion, int[][] board)

**Test**:

* **Arxiu:** BoatTest.java
* **Classe:** BoatTest
* **Mètode de test associat a la funcionalitat:** public void testFreeSpace()