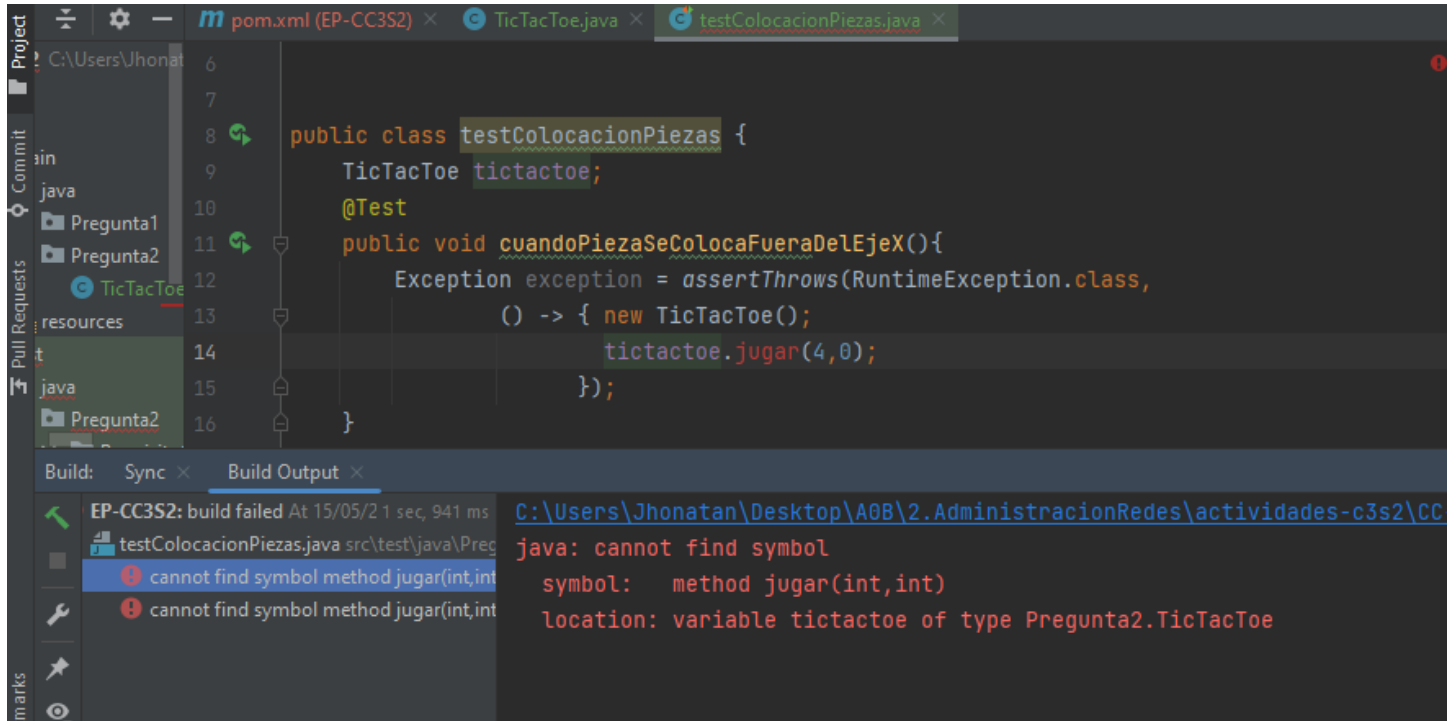


## REQUISITO 1:

RED: falla porque aun no esta implementado el método jugar.



```
public class testColocacionPiezas {
    TicTacToe tictactoe;

    @Test
    public void cuandoPiezaSeColocaFueraDelEjeX(){
        Exception exception = assertThrows(RuntimeException.class,
            () -> { new TicTacToe();
                    tictactoe.jugar(4,0);
                });
    }
}
```

Build: Sync x Build Output x

EP-CC3S2: build failed At 15/05/21 sec, 941 ms

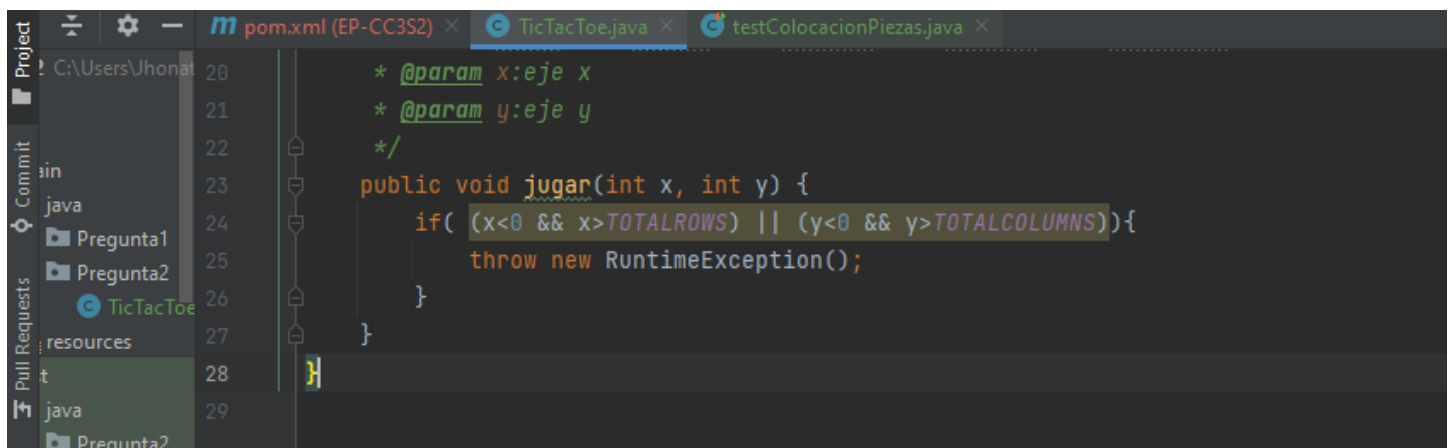
testColocacionPiezas.java src\test\java\Preg

cannot find symbol method jugar(int,int)

cannot find symbol method jugar(int,int)

java: cannot find symbol  
symbol: method jugar(int,int)  
location: variable tictactoe of type Pregunta2.TicTacToe

Implementando el método jugar()



```
* @param x: eje x
* @param y: eje y
*/
public void jugar(int x, int y) {
    if( (x<0 && x>TOTALROWS) || (y<0 && y>TOTALCOLUMNS)){
        throw new RuntimeException();
    }
}
```

GREEN:

```
EP-CC3S2 > src > test > java > Pregunta2 > Requisito1 > testColocacionPiezas > cuandoPiezaSeColocaFueraDelEjeX > Lambda

pom.xml (EP-CC3S2) x TicTacToe.java x testColocacionPiezas.java x

C:\Users\Jhonat...
5      import static org.junit.jupiter.api.Assertions.*;
6
7
8      public class testColocacionPiezas {
9          TicTacToe tictactoe;
10         @Test
11         public void cuandoPiezaSeColocaFueraDelEjeX(){
12             Exception exception = assertThrows(RuntimeException.class,
13                 () -> { new TicTacToe();
14                     tictactoe.jugar(x: 5, y: 2);
15                 });
16         }
17     }

Run: testColocacionPiezas.cuandoPiezaSeColocaFueraDelEjeX x
Tests passed: 1 of 1 test - 29 ms
"C:\Program Files\Amazon Corretto\jdk17.0.6_10\bin\java.exe" ...
Process finished with exit code 0
```

RED, GREEN , de igual modo es para el caso que ponemos fuera del eje Y.

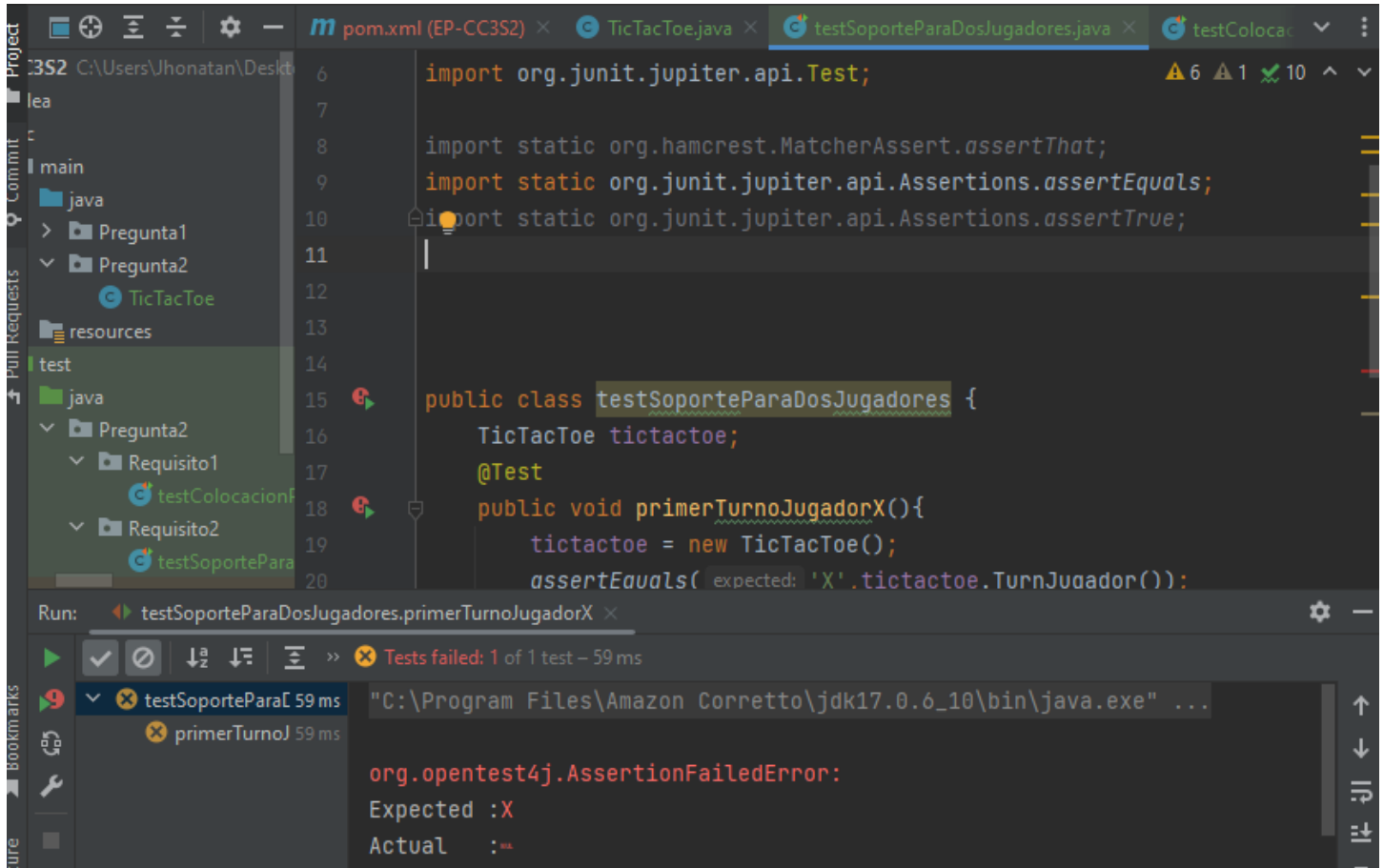
```
11
12         @Test
13         public void cuandoPiezaSeColocaFueraDelEjeX(){
14             Exception exception = assertThrows(RuntimeException.class,
15                 () -> { new TicTacToe();
16                     tictactoe.jugar(x: 5, y: 2);
17                 });
18         }
19         @Test
20         public void cuandoPiezaSeColocaFueraDelEjeY(){
21             Exception exception = assertThrows(RuntimeException.class,
22                 () -> { new TicTacToe();
23                     tictactoe.jugar(x: 1, y: 5);
24                 });
25         }
26     }

Run: testColocacionPiezas.cuandoPiezaSeColocaFueraDelEjeY x
Tests passed: 1 of 1 test - 25 ms
"C:\Program Files\Amazon Corretto\jdk17.0.6_10\bin\java.exe" ...
Process finished with exit code 0
```

En este caso no mucho por refactorizar pero podemos poner un @before para no estar instanciando a cada rato nuestra clase tictactoe.

## REQUISITO 2 :SOPORTE PARA DOS JUGADORES.

RED: falla porque aun no esta implementado el método TurnJugador()



The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes a 'test' directory with a 'java' subdirectory containing 'Requisito1' and 'Requisito2'. The code editor shows the following Java code:

```
import org.junit.jupiter.api.Test;

import static org.hamcrest.MatcherAssert.assertThat;
import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.junit.jupiter.api.Assertions.assertTrue;

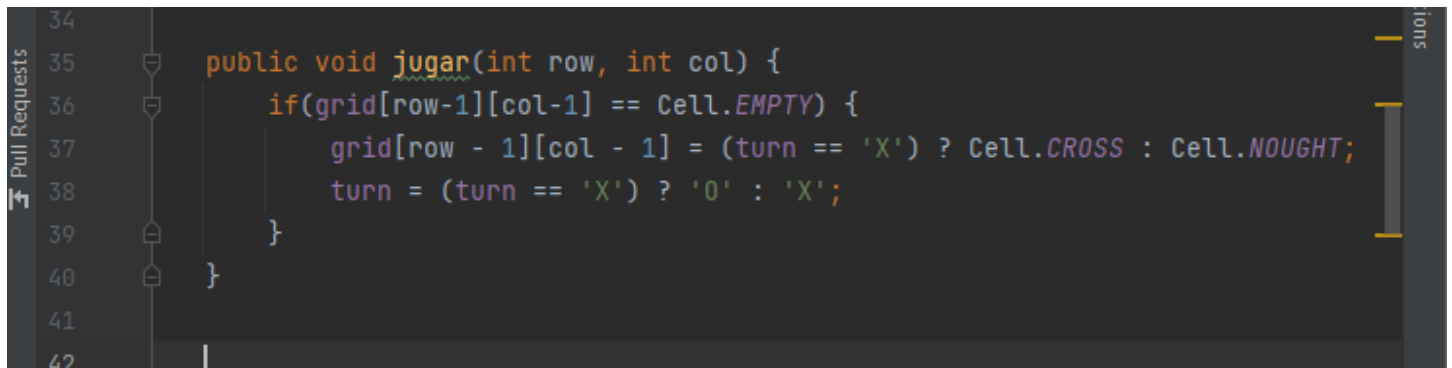
public class testSoporteParaDosJugadores {
    TicTacToe tictactoe;

    @Test
    public void primerTurnoJugadorX(){
        tictactoe = new TicTacToe();
        assertEquals( expected: 'X'.tictactoe.TurnJugador());
    }
}
```

The bottom of the screenshot shows the test execution results. The test 'testSoporteParaDosJugadores.primerTurnoJugadorX' failed. The error message is:

```
org.opentest4j.AssertionFailedError:
Expected :X
Actual   :""
```

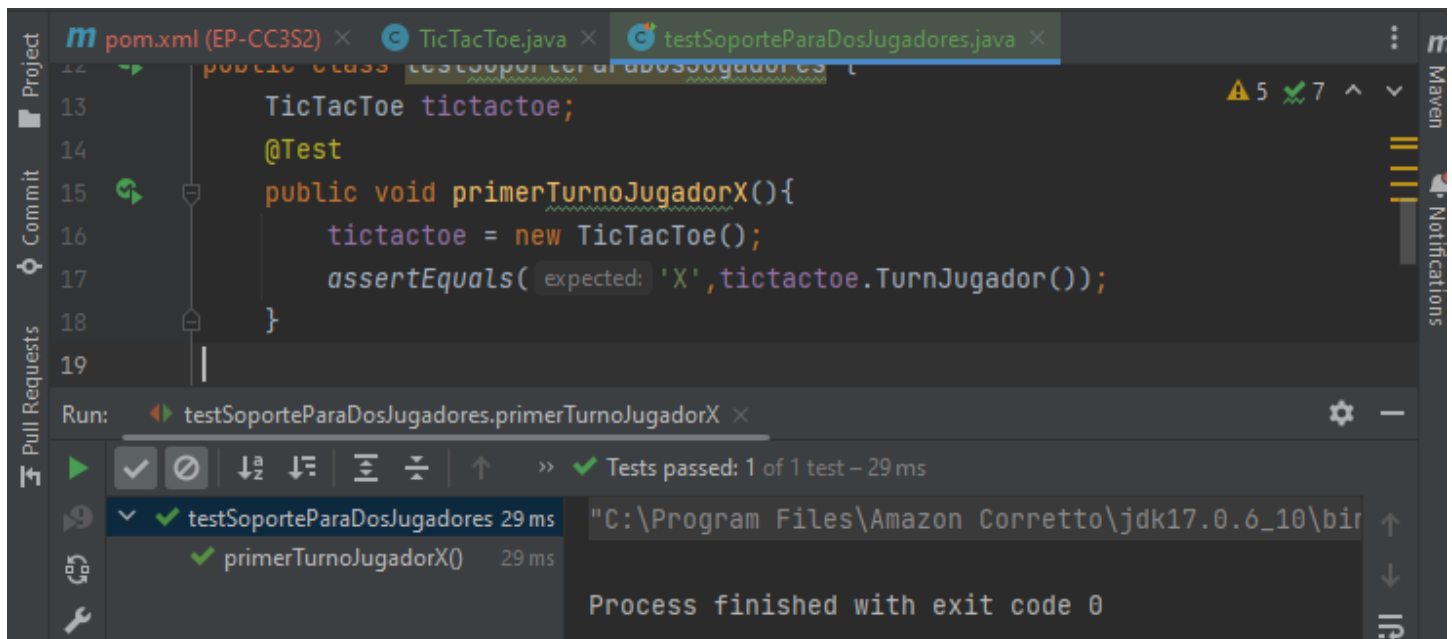
Añadimos un get para que nos muestre el turno de quien es, eso va cambiando en el método jugar.



The screenshot shows the 'jugar' method in a Java class. The code is as follows:

```
public void jugar(int row, int col) {
    if(grid[row-1][col-1] == Cell.EMPTY) {
        grid[row - 1][col - 1] = (turn == 'X') ? Cell.CROSS : Cell.NOUGHT;
        turn = (turn == 'X') ? 'O' : 'X';
    }
}
```

GREEN:



The screenshot shows an IDE with three tabs: `pom.xml (EP-CC3S2)`, `TicTacToe.java`, and `testSoporteParaDosJugadores.java`. The `testSoporteParaDosJugadores.java` tab is active, displaying the following code:

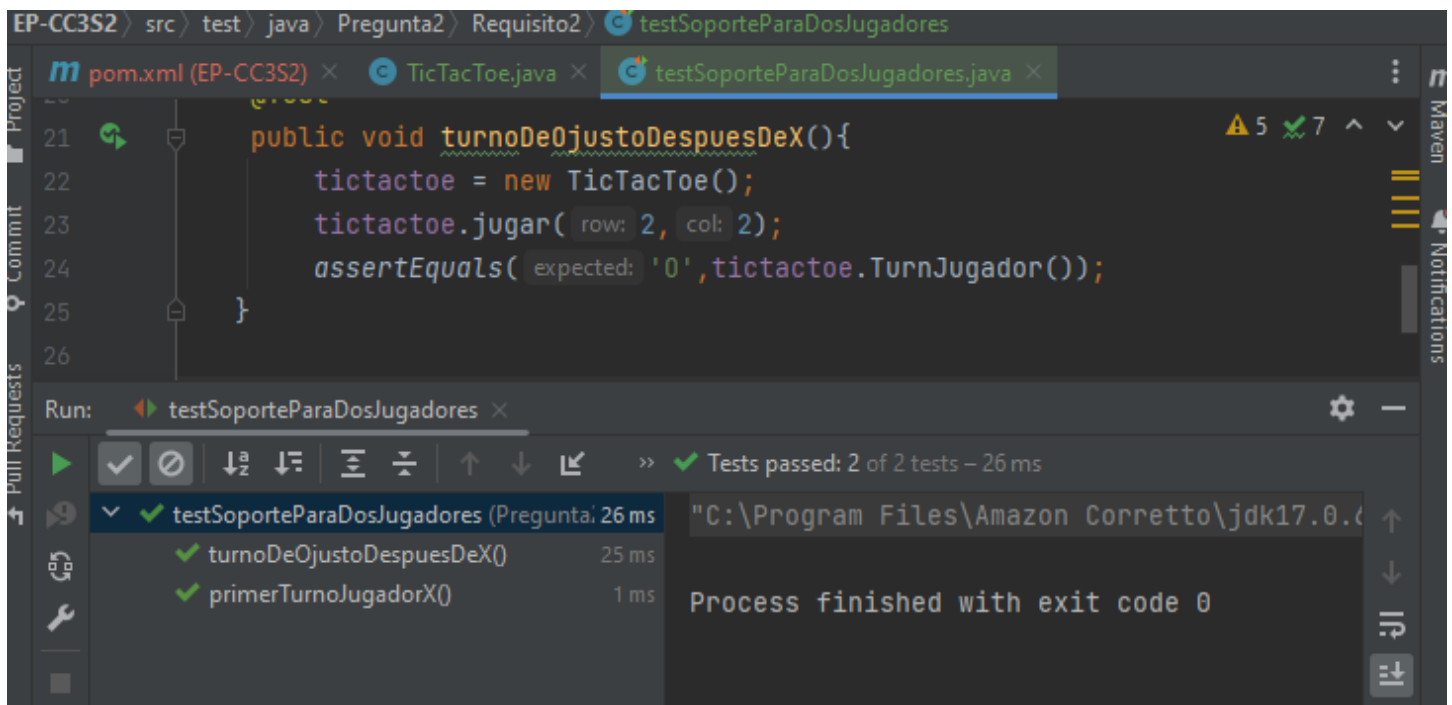
```
12 public class testSoporteParaDosJugadores {
13     TicTacToe tictactoe;
14     @Test
15     public void primerTurnoJugadorX(){
16         tictactoe = new TicTacToe();
17         assertEquals( expected: 'X', tictactoe.TurnJugador());
18     }
19 }
```

Below the code editor, the `Run` tab shows the execution of `testSoporteParaDosJugadores.primerTurnoJugadorX`. The status bar indicates "Tests passed: 1 of 1 test - 29 ms". The test results list shows:

- ✓ testSoporteParaDosJugadores 29 ms
- ✓ primerTurnoJugadorX() 29 ms

The output console displays the path `"C:\Program Files\Amazon Corretto\jdk17.0.6_10\bin"` and the message "Process finished with exit code 0".

Ahora para el caso de “O juega después de X”, hacemos un movimiento y al cambiar de turno corroboraremos que ha cambiado.



The screenshot shows the same IDE with the `testSoporteParaDosJugadores.java` tab active, displaying the following code:

```
21 public void turnoDeOjustoDespuesDeX(){
22     tictactoe = new TicTacToe();
23     tictactoe.jugar( row: 2, col: 2);
24     assertEquals( expected: 'O', tictactoe.TurnJugador());
25 }
26 }
```

Below the code editor, the `Run` tab shows the execution of `testSoporteParaDosJugadores`. The status bar indicates "Tests passed: 2 of 2 tests - 26 ms". The test results list shows:

- ✓ testSoporteParaDosJugadores (Pregunta: 26 ms
- ✓ turnoDeOjustoDespuesDeX() 25 ms
- ✓ primerTurnoJugadorX() 1 ms

The output console displays the path `"C:\Program Files\Amazon Corretto\jdk17.0.6_10\bin"` and the message "Process finished with exit code 0".

Para la 3 prueba seria hacer dos movimientos seguidos.

```
26     @Test
27     public void turnoDeXjustoDespuesDe0(){
28         tictactoe = new TicTacToe();
29         tictactoe.jugar( row: 2, col: 2);
30         tictactoe.jugar( row: 1, col: 1);
31         assertEquals( expected: 'X', tictactoe.TurnJugador());
32     }
```

Run: testSoporteParaDosJugadores.turnoDeXjustoDespuesDe0

Tests passed: 1 of 1 test - 46 ms

testSoporteParaDosJugadores (Pregunta: 46 ms)

turnoDeXjustoDespuesDe0() 46 ms

Process finished with exit code 0

REQUISITO 3: