

# JUEGOS DE PRUEBA DRIVERS

## DriverTablero

### Prueba 1

Descripción: Comprobar si una jugada es legal.

Objetivos: Ver si la comprobación se hace correctamente.

Otros elementos integrados en la prueba:

```
import othello.data.Casilla;
```

```
import othello.data.Pair;
```

Entrada:

```
2 3
```

```
1 (Negra)
```

Salida:

```
correct
```

Resultado de la prueba: ok

### Prueba 2

Descripción: Comprobar si una jugada es legal.

Objetivos: Ver si la comprobación se hace correctamente.

Otros elementos integrados en la prueba:

```
import othello.data.Casilla;
```

```
import othello.data.Pair;
```

Entrada:

```
1 1
```

```
1 (Negra)
```

Salida:

```
Incorrect
```

Resultado de la prueba: ok

### Prueba 3

Descripción: Comprobar si una jugada es legal.

Objetivos: Ver si la comprobación se hace correctamente.

Otros elementos integrados en la prueba:

```
import othello.data.Casilla;
```

```
import othello.data.Pair;
```

Entrada:

```
-111111 11111111
```

```
1 (Negra)
```

Salida:

```
Incorrect
```

Resultado de la prueba: ok

#### Prueba 4

Descripción: Conseguir jugadas legales.

Objetivos: Conseguir todas las jugadas legales para un color en un determinado estado de tablero.

Otros elementos integrados en la prueba:

```
import othello.data.Casilla;
```

```
import othello.data.Pair;
```

Entrada:

```
1 (Negra)
```

```
new Tablero();
```

Salida:

```
[2/3, 3/2, 5/4, 4/5]
```

```
correct
```

Resultado de la prueba: ok

#### DriverDomain

##### Prueba 1

Descripción: Crear un perfil.

Objetivos: Crear un perfil exitosamente.

Otros elementos integrados en la prueba:

```
import othello.domain.CtrlDomain;
```

Entrada:

```
alias
```

Salida:

```
alias.json
```

```
correct
```

Resultado de la prueba: ok

##### Prueba 2

Descripción: Crear un perfil.

Objetivos: Crear un perfil exitosamente.

Otros elementos integrados en la prueba:

```
import othello.domain.CtrlDomain;
```

Entrada:

```
alias
```

Salida:

```
alias ya existe, prueba de nuevo
```

```
Incorrect
```

Resultado de la prueba: ok

### Prueba 3

Descripción: Borrar un perfil.

Objetivos: Borrar un perfil exitosamente.

Otros elementos integrados en la prueba:

Estadística

Entrada:

alias

Salida:

alias borrado correctamente

correct

Resultado de la prueba: ok

### Prueba 4

Descripción: Crear un perfil.

Objetivos: Borrar un perfil exitosamente.

Otros elementos integrados en la prueba:

Estadística

Entrada:

alias

Salida:

alias no existe.

Incorrect

Resultado de la prueba: ok

### Prueba 5

Descripción: Imprimir el ranking.

Objetivos: Imprimir el ranking ordenado en orden descendiente por score.

Otros elementos integrados en la prueba:

Estadística

Entrada:

alias1.json(5 score), alias2.json(3 score), alias3.json(0 score)

Salida:

alias1:

Puntos: 5

Victorias: 2

Derrotas: 1

Empates: 0

alias2:

Puntos: 3

Victorias: 1

Derrotas: 0

Empates: 0

alias 3:

Puntos: 0

Victorias: 0

Derrotas: 2

Empates: 0

correct

Resultado de la prueba: ok

## DriverEscenario

### Prueba 1

Descripción: Undo.

Objetivos: deshacer una acción hecha previamente.

Otros elementos integrados en la prueba:

```
import othello.data.Casilla;
```

```
import othello.data.Pair;
```

```
import java.util.ArrayList;
```

Entrada:

Undo

Salida:

Tablero previo

Correct

Resultado de la prueba: ok

### Prueba 2

Descripción: Undo.

Objetivos: deshacer una acción hecha previamente.

Otros elementos integrados en la prueba:

```
import othello.data.Casilla;
```

```
import othello.data.Pair;
```

```
import java.util.ArrayList;
```

Entrada:

Undo en tablero base

Salida:

Tablero base

Incorrect

Resultado de la prueba: ok

### Prueba 3

Descripción: Redo.

Objetivos: Rehacer una acción deshecha previamente.

Otros elementos integrados en la prueba:

```
import othello.data.Casilla;
```

```
import othello.data.Pair;
```

```
import java.util.ArrayList;
```

Entrada:

Redo

Salida:

Tablero posterior

Correct

Resultado de la prueba: ok

#### Prueba 4

Descripción: Redo.

Objetivos: Rehacer una acción deshecha previamente.

Otros elementos integrados en la prueba:

```
import othello.data.Casilla;  
import othello.data.Pair;  
import java.util.ArrayList;
```

Entrada:

Redo con Stack vacío

Salida:

Tablero sin cambios

Incorrect

Resultado de la prueba: ok

#### **DriverPartida**

#### Prueba 1

Descripción: PrintTree.

Objetivos: Mostrar el árbol de predicción en función del estado del tablero.

Otros elementos integrados en la prueba:

```
import othello.data.Casilla;  
import othello.data.Pair;  
import othello.domain.IA;  
import othello.domain.tablero.Tablero;  
import othello.domain.CtrlDomain;
```

Entrada:

new Tablero()

Salida:

```

Node{cord=-1/-1, player=VACIA, score=0}
Node{cord=2/4, player=BLANCA, score=25}
Node{cord=2/3, player=NEGRA, score=25}
Node{cord=4/2, player=BLANCA, score=35}
Node{cord=3/2, player=BLANCA, score=25}
Node{cord=5/2, player=BLANCA, score=25}
Node{cord=2/2, player=BLANCA, score=40}
Node{cord=1/2, player=BLANCA, score=1}
Node{cord=4/5, player=NEGRA, score=25}
Node{cord=5/3, player=BLANCA, score=25}
Node{cord=5/2, player=BLANCA, score=25}
Node{cord=5/4, player=BLANCA, score=25}
Node{cord=5/5, player=BLANCA, score=25}
Node{cord=5/6, player=BLANCA, score=1}
Node{cord=2/5, player=NEGRA, score=25}
Node{cord=5/3, player=BLANCA, score=25}
Node{cord=4/2, player=BLANCA, score=25}
Node{cord=3/5, player=BLANCA, score=25}
Node{cord=2/6, player=BLANCA, score=1}
Node{cord=3/5, player=BLANCA, score=25}
Node{cord=2/3, player=NEGRA, score=25}
Node{cord=4/2, player=BLANCA, score=25}
Node{cord=3/2, player=BLANCA, score=25}
Node{cord=5/2, player=BLANCA, score=25}
Node{cord=2/2, player=BLANCA, score=25}
Node{cord=1/2, player=BLANCA, score=1}
Node{cord=4/5, player=NEGRA, score=25}
Node{cord=5/3, player=BLANCA, score=35}
Node{cord=5/2, player=BLANCA, score=25}
Node{cord=5/4, player=BLANCA, score=25}
Node{cord=5/5, player=BLANCA, score=40}
Node{cord=5/6, player=BLANCA, score=1}
Node{cord=2/5, player=NEGRA, score=25}
Node{cord=5/3, player=BLANCA, score=25}
Node{cord=4/2, player=BLANCA, score=25}
Node{cord=1/5, player=BLANCA, score=1}
Node{cord=2/4, player=BLANCA, score=25}
Node{cord=5/3, player=BLANCA, score=25}
Node{cord=3/2, player=NEGRA, score=25}
Node{cord=2/4, player=BLANCA, score=25}
Node{cord=2/3, player=BLANCA, score=25}
Node{cord=2/5, player=BLANCA, score=25}
Node{cord=2/2, player=BLANCA, score=25}
Node{cord=2/1, player=BLANCA, score=1}
Node{cord=5/4, player=NEGRA, score=25}
Node{cord=3/5, player=BLANCA, score=35}
Node{cord=2/5, player=BLANCA, score=25}
Node{cord=4/5, player=BLANCA, score=25}
Node{cord=5/5, player=BLANCA, score=40}
Node{cord=6/5, player=BLANCA, score=1}
Node{cord=5/2, player=NEGRA, score=25}
Node{cord=2/4, player=BLANCA, score=25}
Node{cord=3/5, player=BLANCA, score=25}
Node{cord=4/2, player=BLANCA, score=25}
Node{cord=5/1, player=BLANCA, score=1}
Node{cord=4/2, player=BLANCA, score=25}
Node{cord=3/2, player=NEGRA, score=25}
Node{cord=2/4, player=BLANCA, score=35}
Node{cord=2/3, player=BLANCA, score=25}
Node{cord=2/5, player=BLANCA, score=25}
Node{cord=2/2, player=BLANCA, score=40}
Node{cord=2/1, player=BLANCA, score=1}
Node{cord=5/4, player=NEGRA, score=25}
Node{cord=3/5, player=BLANCA, score=25}
Node{cord=2/5, player=BLANCA, score=25}
Node{cord=4/5, player=BLANCA, score=25}
Node{cord=5/5, player=BLANCA, score=25}
Node{cord=6/5, player=BLANCA, score=1}
Node{cord=5/2, player=NEGRA, score=25}
Node{cord=2/4, player=BLANCA, score=25}
Node{cord=3/5, player=BLANCA, score=25}
Node{cord=6/2, player=BLANCA, score=1}
Node{cord=5/3, player=BLANCA, score=25}

```

Correct

Resultado de la prueba: ok

## Prueba 2

Descripción: PartidaAvslA.

Objetivos: Comprobar que una partida se desarrolla correctamente.

Otros elementos integrados en la prueba:

```
import othello.data.Casilla;  
import othello.data.Pair;  
import othello.domain.IA;  
import othello.domain.tablero.Tablero;  
import othello.domain.CtrlDomain;
```

Entrada:

```
new IA(Casilla.BLANCA)  
new IA(Casilla.NEGRA)  
new Tablero()
```

Salida:

la partida se desarrolla correctamente.

Resultado de la prueba: ok

## DriverIA

### Prueba 1

Descripción: testScore.

Objetivos: Comprobar que al hacer una jugada, los swaps de color se reflejan correctamente con la heurística..

Otros elementos integrados en la prueba:

```
import java.util.ArrayList;  
import othello.data.Casilla;  
import othello.data.Node;  
import othello.data.Pair;  
import othello.data.Tree;
```

Entrada:

```
2 3  
1 3  
0 3
```

Salida:

```
9
```

Resultado de la prueba: ok

## Prueba 2

Descripción: testAlphaBeta.

Objetivos: Comprobar que la IA escoge correctamente la coordenada a partir del arbol

Otros elementos integrados en la prueba:

```
import java.util.ArrayList;  
import othello.data.Casilla;  
import othello.data.Node;  
import othello.data.Pair;  
import othello.data.Tree;
```

Entrada:

2 3 5

4 2 9

4 3 -1

Salida:

4/2

Resultado de la prueba: ok