

## **Project reversihgm**

*Author: Herrero-Morilla-Gomez*

*Date: 2013-11-07*

---

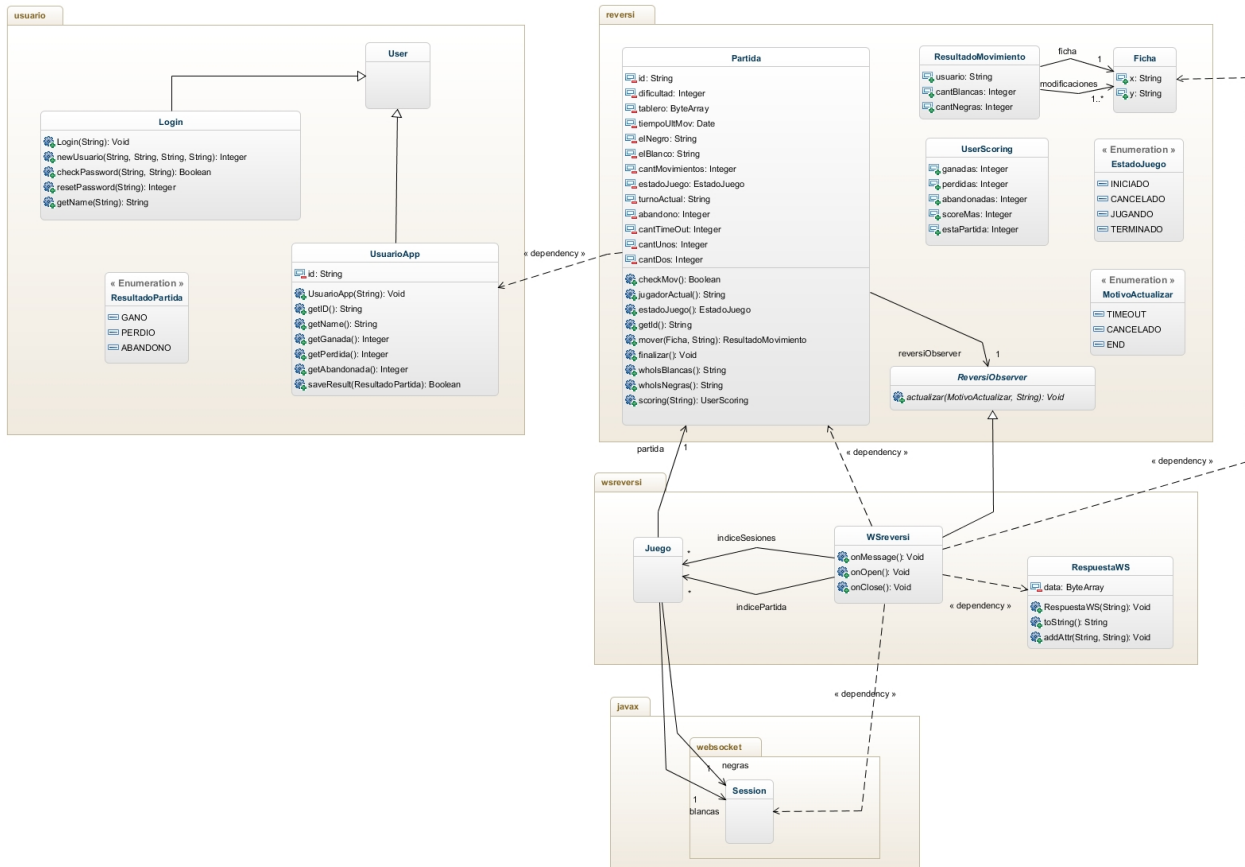
# Table of Contents

Overview	
Model	.....
Diagrams	.....
Packages	
usuario	.....
User	.....
Login	.....
UsuarioApp	.....
Enumerations	.....
reversi	.....
Partida	.....
ReversiObserver	.....
ResultadoMovimiento	.....
Ficha	.....
UserScoring	.....
Enumerations	.....
Enumerations	.....
javax	.....
websocket	.....
Session	.....
wsreversi	.....
WSreversi	.....
Juego	.....
RespuestaWS	.....

# 1. Overview

## 1.1 Model Description

## 1.2 Diagrams



## 5 Packages

### 5.1 Package usuario

#### 5.1.3 Class UsuarioApp

##### 5.1.1.1 Attributes

No additional attributes

##### 5.1.1.2 Generalizations

None

##### 5.1.1.3 Operations

#### 5.1.3 Class UsuarioApp

##### 5.1.2.1 Attributes

No additional attributes

##### 5.1.2.2 Generalizations

- **User**

##### 5.1.2.3 Operations

- Void **Login** (parameter : String)

- Integer **newUsuario** (usuario : String, password : String, nombre : String, email : String)

*Description:* 0: OK

1: duplicado

2: error inesperado

- Boolean **checkPassword** (usuario : String, password : String)

- Integer **resetPassword** (usuario : String)

*Description:* 0: ok

1: usuario inexistente

2: error inesperado

- String **getName** (id : String)

## 5.1.3 Class **UsuarioApp**

### 5.1.3.1 Attributes

- **id** : String [1]

### 5.1.3.2 Generalizations

- **User**

### 5.1.3.3 Operations

- Void **UsuarioApp** (id : String)
- String **getID** ()
- String **getName** ()
- Integer **getGanada** ()
- Integer **getPerdida** ()
- Integer **getAbandonada** ()
- Boolean **saveResult** (resultado : ResultadoPartida)

## 5.1.2 **ResultadoPartida**

### 5.1.2.1 Literals

- GANO

- PERDIO

- ABANDONO

## 5.2 Package reversi

### 5.2.5 Class UserScoring

#### 5.2.1.1 Attributes

- **id** : String [1]

*Description:* Se forma concatenando los ID de los usuariosApp (elNegro y elBlanco) mas un Timestamp

- **dificultad** : Integer [1]

*Description:* Los niveles de dificultad son:  
1- Facil (no tienen tiempo para realizar el movimiento)  
2- Medio (60 segundos para realizar el movimiento)  
3- Dificil ( 30 segundos para realizar el movimiento)

- **tablero** : ByteArray [1]

- **tiempoUltMov** : Date [1]

*Description:* Indica el tiempo que tiene el jugador para realizar el movimiento de acuerdo al nivel de dificultad seleccionado para la partida

- **elNegro** : String [1]

- **elBlanco** : String [1]

- **cantMovimientos** : Integer [1]

- **estadoJuego** : EstadoJuego [1]

- **turnoActual** : String [1]

*Description:* Mediante este atributo controlamos quien es el jugador que tiene el turno actualmente.

- **abandono** : Integer [1]

- **cantTimeOut** : Integer [1]

- **cantUnos** : Integer [1]

*Description:* Lleva la cuenta de cuantas fichas negras hay en el tablero.

- **cantDos** : Integer [1]

*Description:* Lleva la cuenta de cuantas fichas blancas hay en el tablero.

### 5.2.1.2 Generalizations

None

### 5.2.1.3 Operations

- Boolean **checkMov** ()

*Description:* Checkea si un movimiento es válido.

- String **jugadorActual** ()

- EstadoJuego **estadoJuego** ()

*Description:* Devuelve un entero que indica el estado actual:

- 1- Iniciado.
- 2- Cancelado
- 3- Jugando
- 4- Terminado.

- String **getId** ()

*Description:* ID es igual al nombre de los Usuarios más el TimeStart

- ResultadoMovimiento **mover** (ficha : Ficha, jugador : String)

*Description:* si no se pudo realizar el movimiento devolver NULL

- Void **finalizar** ()

*Description:* Este método se utilizará cuando si el servidor desea cerrar la conexión y terminar el juego.

- String **wholsBlancas** ()

*Description:* devuelve el ID del jugador Blanco

- String **wholsNegras** ()

*Description:* devuelve el ID del jugador Negro

- UserScoring **scoring** (idUser : String)

*Description:*

*Si la partida esta en progreso, devuelve null*

*Si la partida finalizó, devuelve el UserScoring.*

*Nota:* Chequear que el idUser es válido

#### 5.2.1.4 Associations

- **reversiObserver** : ReversiObserver [1]

### 5.2.5 Class UserScoring

#### 5.2.2.1 Attributes

No additional attributes

#### 5.2.2.2 Generalizations

None

#### 5.2.2.3 Operations

- Void **actualizar** (motivo : MotivoActualizar, partidald : String)

### 5.2.5 Class UserScoring

#### 5.2.3.1 Attributes

- **usuario** : String [1]

- **cantBlancas** : Integer [1]

- **cantNegras** : Integer [1]



### 5.2.3.2 Generalizations

None

### 5.2.3.3 Operations

### 5.2.3.4 Associations

- **modificaciones** : Ficha [1..\*]

- **ficha** : Ficha [1]

## 5.2.5 Class UserScoring

### 5.2.4.1 Attributes

- **x** : String [1]

- **y** : String [1]

### 5.2.4.2 Generalizations

None

### 5.2.4.3 Operations

## 5.2.5 Class UserScoring

### 5.2.5.1 Attributes

- **ganadas** : Integer [1]

- **perdidas** : Integer [1]

- **abandonadas** : Integer [1]

- **scoreMas** : Integer [1]

*Description:* estaPartida: 0: Perdio | 1: Gano | -1: Abandono

- **estaPartida** : Integer [1]

*Description:* indica el resultado de la partida actual para el jugador cuyo ID recibimos como parametro:  
3 )si GANO  
-1 ) si PERDIO  
-3) si ABANDONO

#### **5.2.5.2 Generalizations**

None

#### **5.2.5.3 Operations**

### **5.2.2 EstadoJuego**

#### **5.2.2.1 Literals**

- INICIADO

- CANCELADO

- JUGANDO

- TERMINADO

### **5.2.3 MotivoActualizar**

#### **5.2.3.1 Literals**

- TIMEOUT

- CANCELADO

- END

Description:

significa cuando finaliza la partida.

## **5.3 Package javax**

### **5.3.1 Package websocket**

#### **5.3.1.1 Class Session**

##### **5.3.1.1.1 Attributes**

No additional attributes

#### 5.3.1.1.2 Generalizations

None

#### 5.3.1.1.3 Operations

### 5.4 Package wsreversi

#### 5.4.3 Class RespuestaWS

*Para acceder a los juegos cuando el websocket recibe un mensaje, los busca mediante el ID de la session que origino el evento en el hashmap de juegos*

*ReversiWS: Mensajes I/O*

=====

*## Inputs*

*operacion : INIT*

*id : String -- User ID*

*nivel : String (facil|medio|dificil)*

*operacion : MOVE*

*posx : String (1|2|3|4|5|6|7|8)*

*posy : String (1|2|3|4|5|6|7|8)*

*operacion : QUIT*

*## Outputs*

*operacion : TIMEOUT|INIT*

*turno : Boolean*

*mensaje : String*

*operacion : CANCEL|END|QUIT*

*data : Scoring*

*operacion : ERROR|MSG*

*data: String*

*operacion : MOVER*

*hecho : Boolean*

*hecho(true) -> turno : Boolean*

*hecho(true) -> data : ResultadoMovimiento*

*hecho(false) -> data : String*

*hecho(false) -> ficha : Ficha*

#### **5.4.1.1 Attributes**

No additional attributes

#### **5.4.1.2 Generalizations**

- **ReversiObserver**

#### **5.4.1.3 Operations**

- Void **onMessage** ()

- Void **onOpen** ()

- Void **onClose** ()

#### **5.4.1.4 Associations**

- **indiceSesiones** : Juego [\*]

- **indicePartida** : Juego [\*]

### **5.4.3 Class RespuestaWS**

#### **5.4.2.1 Attributes**

No additional attributes

#### **5.4.2.2 Generalizations**

None

#### **5.4.2.3 Operations**

#### **5.4.2.4 Associations**

- **negras** : Session [1]

- **partida** : Partida [1]

- **blancas** : Session [1]

### 5.4.3 Class RespuestaWS

*Objeto Genérico para uniformar las respuestas que se enviarán a los clientes del websocket. Facilita la creación de las cadenas JSON.*

#### 5.4.3.1 Attributes

- **data** : ByteArray [1]

*Description:* Este atributo es un HashMap<String, Object>

#### 5.4.3.2 Generalizations

None

#### 5.4.3.3 Operations

- Void **RespuestaWS** (type : String)

- String **toString** ()

*Description:* Convierte el HashMap en un String con la representación JSON del objeto

- Void **addAttr** (name : String, data : String)

*Description:* Data es type Object.