# 

# 

# 

# 

# 

DOCUMENTACIÓN BD PANDEMIC

DAME

2023

DAW1

# 

# 

# 

ÍNDICE:

[**Elementos BD 2**](#_fg8g73igw9bb)

[TABLAS 2](#_88q1zk718bcf)

[PARTIDAS: 2](#_h2ixir9lye2u)

[USUARIOS 3](#_2d12lqinzru)

[OBJETOS 4](#_bh2c7x92t595)

[CIUDADES 4](#_48vdspakbx82)

[VACUNAS 4](#_9no0esmqrg6q)

[VARRAYS 5](#_v9rohfhylzj0)

[CIUDADESPARTIDAS 5](#_iebuzp8wx87d)

[VACUNASPARTIDAS 5](#_2yft18j6waw)

[PROCEDURES: 5](#_ggqe7r3rlbnp)

[CREARUSUARIO 5](#_l5uoxgfvmjas)

[SECUENCIAS: 6](#_kjzo5h6scaqw)

[IDP\_SEQU 6](#_2q5c0uldnxiu)

[IDU\_SEQU 6](#_zeasxz3ydug6)

[**Modelo entidad/relación 7**](#_6k0ah71kthno)

[DIAGRAMA DE CLASSES 7](#_nvqa5b2qp2u)

[MODELO RELACIONAL 7](#_php11mb3hkze)

[**RANKING 8**](#_d692dt5lovsi)

[Insertar datos ranking: 8](#_9hz376rmg9yq)

[EN QUE MOMENTO: 8](#_sad3qomf6j72)

[SENTENCIAS: 8](#_1zkab2n76vao)

[EXTRAER DATOS RANKING: 9](#_wuoickqtub1q)

[EN QUÉ MOMENTO: 9](#_1r4fmk54b4uw)

[SENTENCIAS: 10](#_n9thwwyddhqq)

[**Partidas 10**](#_hlouusecgrj3)

[Insertar datos partida: 10](#_kwr9y1b5h1fv)

[EN QUÉ MOMENTO: 10](#_x4ylssz97036)

[SENTENCIAS: 11](#_39zuq4l8a9xj)

[Update datos partida: 11](#_3ewq416udvav)

[EN QUÉ MOMENTO: 11](#_es6cvv5uwrfk)

[SENTENCIAS: 12](#_34hr18fbf9l9)

[EXTRAER DATOS PARTIDAS: 13](#_n8lf9gurlgmh)

[Cargar partidas guardadas: 13](#_rs6cukdin4nu)

[EN QUÉ MOMENTO: 13](#_wdpmbw4qihaz)

[SENTENCIAS: 13](#_hoall9f0wlmg)

[Iniciar una partida guardadas: 14](#_mvz7cd1wt3iq)

[EN QUÉ MOMENTO: 14](#_k6ou7mycttvm)

[SENTENCIAS: 15](#_26wxk65unr5t)

[**USUARIOS 17**](#_xwe0wo88mstv)

[CREAR USUARIOS: 17](#_r790drlabssj)

[EN QUÉ MOMENTO: 17](#_bwhoppexbo6a)

[SENTENCIAS: 18](#_pvsv0xl4gc6z)

[CONSULTAR USUARIOS: 19](#_yxk9rptq7201)

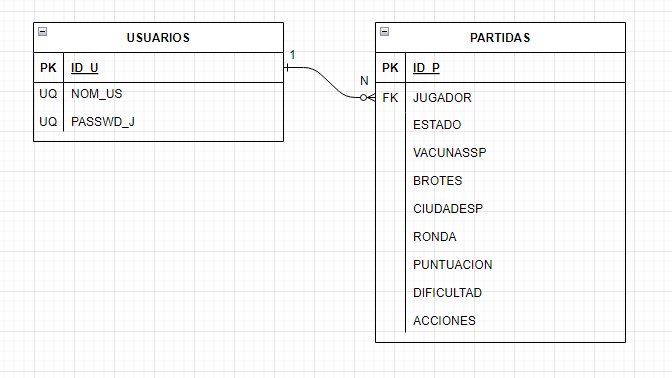
[EN QUÉ MOMENTO: 19](#_gh9gb7b1w1e6)

[SENTENCIAS: 19](#_r49q039fyafq)

# **Modelo entidad/relación**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **DIAGRAMA DE CLASSES**



### ***MODELO RELACIONAL***

# 

**PRIMARY KEY** / ***FOREIGN KEY***

USUARIOS(**id\_u**, nom\_us, passwd\_j)

PARTIDAS(**id\_p**, ***jugador***, estado, vacunasp, brotes, ciudadesp, ronda, puntuación, dificultad, acciones)

***jugador*** hace referencia a USUARIOS(id\_u)

# **Elementos BD**

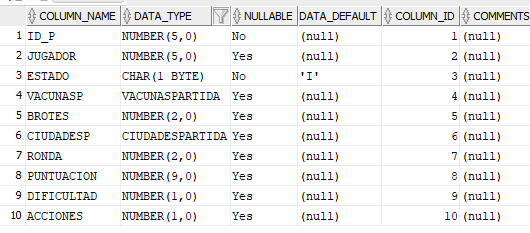
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **TABLAS**

### ***PARTIDAS:***

Aquí guardamos todos los datos relacionados con las partidas, tanto las acabadas (Ranking) como las inacabadas(Cargar Partidas).

| **CAMPO** | **TIPO DE DATO** | **NULLABLE** |  | **DESCRIPCIÓN** |
| --- | --- | --- | --- | --- |
| ID\_P | NUMBER(5,0) | NOT NULL | Primary key | Id de la partida |
| JUGADOR | NUMBER(5,0) |  | Foreing key | Id del jugador al que pertenece la partida |
| ESTADO | CHAR(1) | NOT NULL | DEFAULT = ‘I’ | Estado de la partida:  ‘I’=Partida inacabada  ‘A’=Partida acabada |
| VACUNASP | VARRAY  (vacunaspartida) |  |  | Estado de las vacunas de la partida |
| BROTES | NUMBER(2,0) |  |  | Numero de brotes de la partida |
| CIUDADESP | VARRAY  (ciudadespartida) |  |  | Estado de las ciudades de la partida |
| RONDA | NUMBER(2,0) |  |  | Numero de ronda de la partida |
| PUNTUACIÓN | NUMBER(9,0) |  |  | Puntuacion de la partida |
| DIFICULTAD | NUMBER(1,0) |  |  | Dificultad de la partida:  0=Facil  1=Normal  2=Dificil |
| ACCIONES | NUMBER(1,0) |  |  | Acciones restantes de la ronda |



### ***USUARIOS***

Aquí guardamos todos los usuarios registrados en nuestro programa.

| **CAMPO** | **TIPO DE DATO** | **NULLABLE** |  | **DESCRIPCIÓN** |
| --- | --- | --- | --- | --- |
| ID\_U | NUMBER(5,0) | NOT NULL | Primary KEY | id del Usuario |
| NOM\_US | VARCHAR2(20) |  | UNIQUE | Nombre de usuario |
| PASSWD\_J | VARCHAR2(15) |  | UNIQUE | Contraseña del usuario |

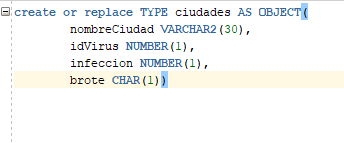
## 

## **OBJETOS**

### ***CIUDADES***

Este objeto nos sirve para guardar los datos de las ciudades de la partida.

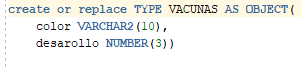
| **NOMBRE DATO** | **TIPO DATO** | **DESCRIPCIÓN** |
| --- | --- | --- |
| NOMBRECIUDAD | VARCHAR2(30) | Nombre de la ciudad |
| IDVIRUS | NUMBER(1) | Virus de la ciudad |
| INFECCION | NUMBER(1) | Numero de infección de la ciudad |
| BROTE | CHAR(1) | Si la ciudad tiene brote o no:  ‘S’=Brote/‘N’=No brote |



### ***VACUNAS***

Este objeto nos sirve para guardar los datos de las vacunas de la partida.

| **NOMBRE DATO** | **TIPO DATO** | **DESCRIPCIÓN** |
| --- | --- | --- |
| COLOR | VARCHAR2(10) | Color de la vacuna |
| DESARROLLO | NUMBER(3) | Desarrollo de la vacuna |



## **VARRAYS**

### ***CIUDADESPARTIDAS***

Array de objetos(ciudades), donde guardamos los datos de las ciudades de la partida.



### ***VACUNASPARTIDAS***

Array de objetos(vacunas), donde guardamos los datos de las vacunas de la partida.



## **PROCEDURES:**

### ***CREARUSUARIO***

Esta función sirve para crear un nuevo usuario cuando se registra un nuevo usuario en nuestro juego.

**Parámetros entrada:**

-nom VARCHAR2: Nombre del usuario.

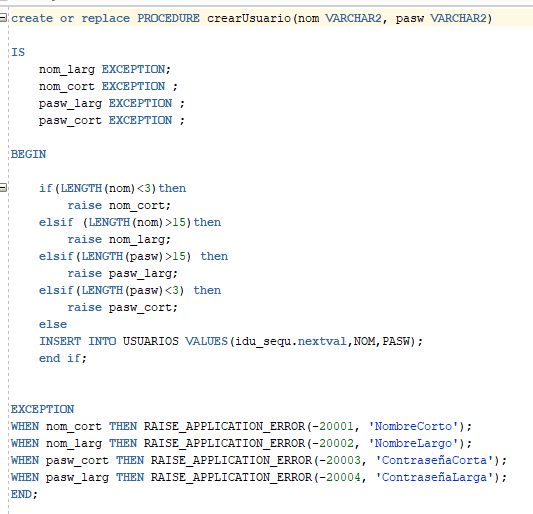
-pasw VARCHAR2: Password del usuario.

**Excepciones:**

-nom\_cort: Cuando el nombre es más corto de 3 caracteres(-20001).

-nom\_larg: Cuando el nombre es más largo de 15 caracteres(-20002).

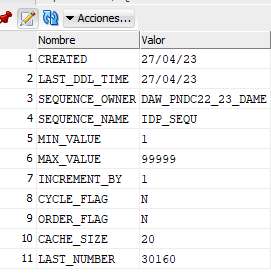
-pasw\_cort: Cuando la contraseña es más corta de 3 caracteres(-20003).

-pasw\_larg: Cuando la contraseña es más larga de 15 caracteres(-20004).

## **SECUENCIAS:**

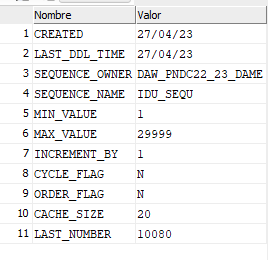
### **IDP\_SEQU**

Secuencia para crear un autonumérico, id\_p(Partidas).



### ***IDU\_SEQU***

Secuencia para crear un autonumérico, id\_u(Usuarios).



# **RANKING**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Insertar datos ranking:**

### **EN QUE MOMENTO:**

Los datos de las partidas acabadas(Partidas para el ranking) se guardarán en la tabla partidas, para diferenciar los registros de partidas guardadas utilizamos el campo ‘estado’(‘A’=Acabada/’I’=Inacabada).

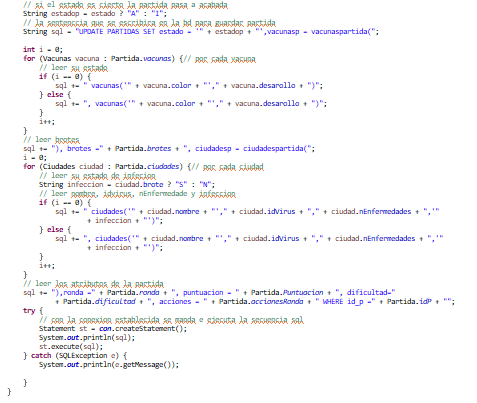
Cuando acabemos una partida,ya sea ganando o perdiendo, se hara un UPDATE a la partida que hemos acabado(cuando se inicia o se carga guardamos el id\_p en java) con los datos correspondientes y poniendo el estado en ‘A’.

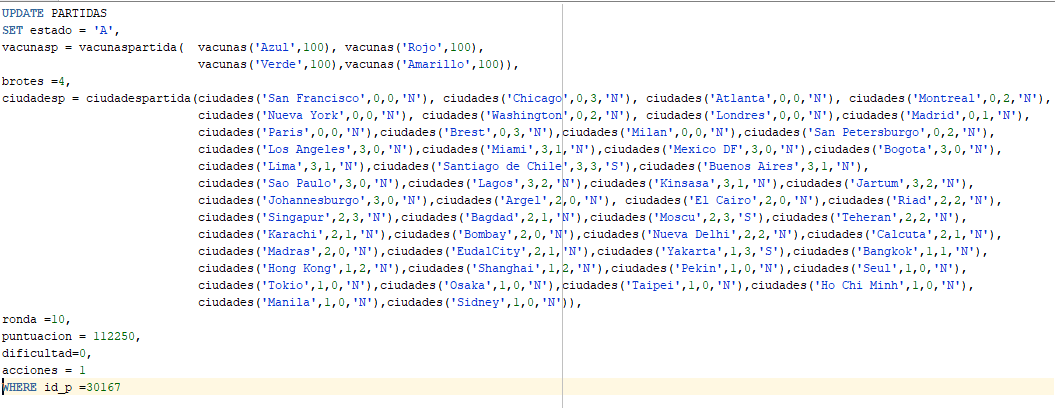
Cuando la partida se pierde se guardan los puntos en negativo.

### **SENTENCIAS:**

**Insertar partida acabada:**

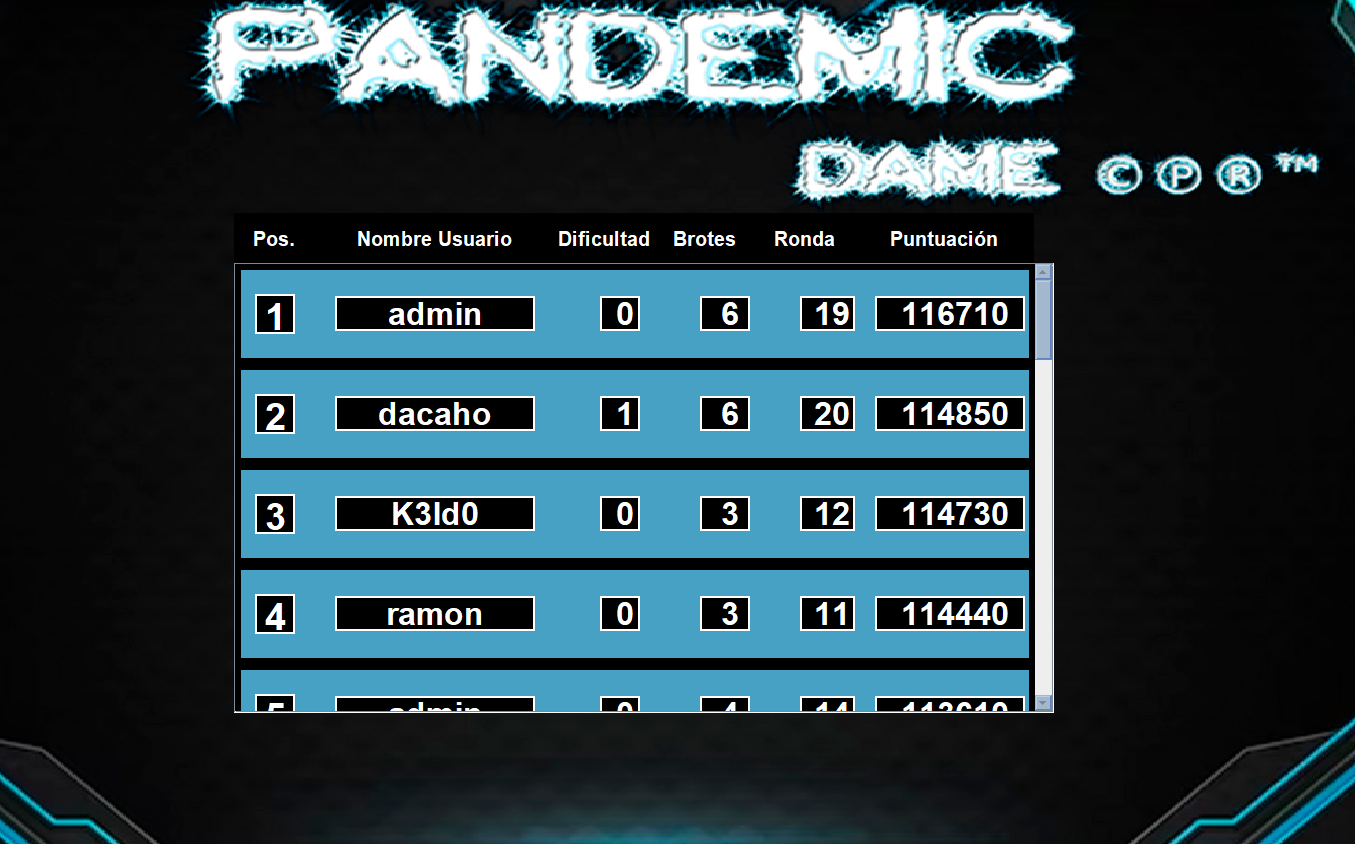
**Java:**



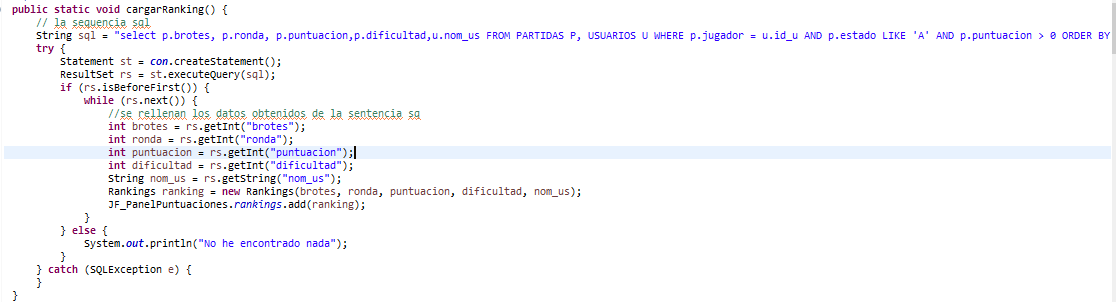
**Oracle:**

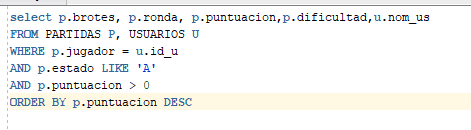
## **EXTRAER DATOS RANKING:**

### **EN QUÉ MOMENTO:**

Se extraerán datos para el ranking cuando el jugador entre en el Panel de puntuaciones. Se hara un Select de los datos necesarios de todas las partidas con estado = ‘A’(Partida acabada ) y la puntuación sea > 0 (Partida ganada).

### **SENTENCIAS:**

Java:

Sql:

# 

# **Partidas**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

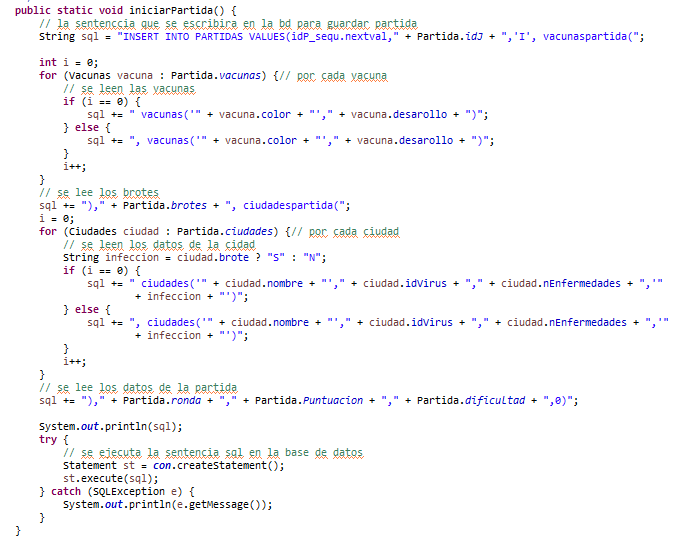
## **Insertar datos partida:**

### **EN QUÉ MOMENTO:**

Cuando se inicia una nueva partida se hace un insert en la tabla Partidas con la nueva partida sin necesidad de que el jugador guarde la partida. El estado de la partida sera ‘I’.

### 

### **SENTENCIAS:**

Java:

SQL:

## **Update datos partida:**

### **EN QUÉ MOMENTO:**

Cuando el jugador le da al botón de guardar o sale de la partida el estado actual de la partida se guarda en la base de datos para poder retomarla en cualquier momento. Para ello haremos un update de la partida en la que estamos actualizando los datos necesarios. La única diferencia con el guardar partidas acabadas es que el estado = ‘I’.

### **SENTENCIAS:**

Java:

SQL:



## **EXTRAER DATOS PARTIDAS:**

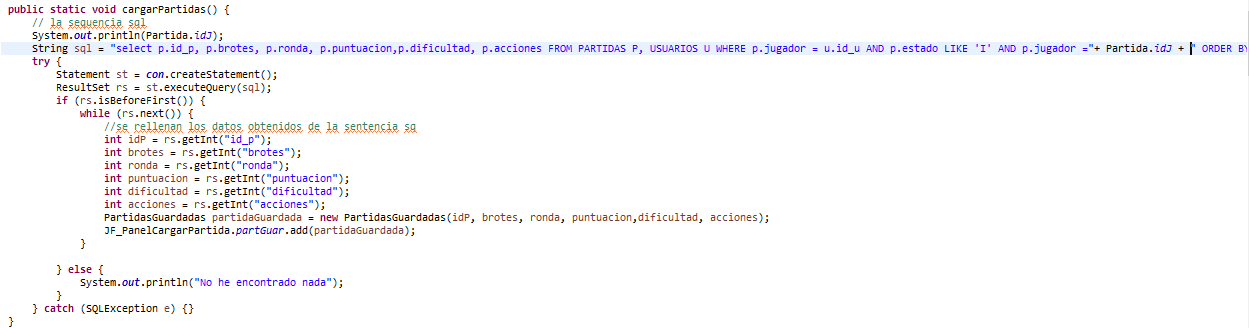
### **Cargar partidas guardadas:**

### **EN QUÉ MOMENTO:**

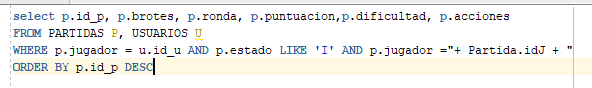
Cuando el jugador entre en el panel cargar partida se le mostrará un scroll con todas las partidas guardadas que tiene el usuario que ha iniciado sesion al principio.

### **SENTENCIAS:**

Java:



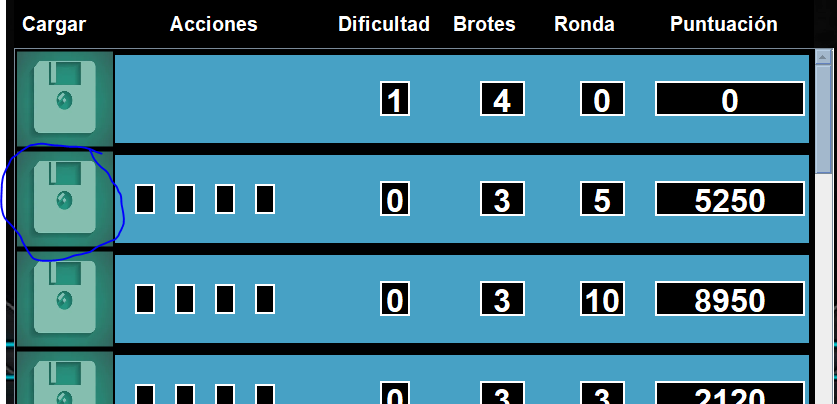
SQL:



### **Iniciar una partida guardadas:**

### **EN QUÉ MOMENTO:**

Cuando estemos en el menu de cargar partidas y le demos al botón de una partida se iniciara la partida guardada sobre la que hemos pulsado.

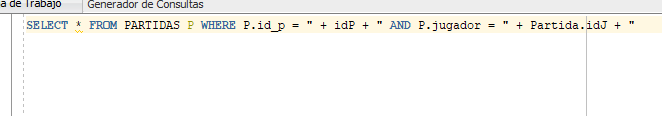
Para iniciar una partida haremos un select de los datos de la partida pasandole el id de partida. Eso es posible ya que el botón que pulsamos tiene como atributo el id de la partida, lo añadimos como atributo cuando iniciamos el botón para mostrar las partidas guardadas.

### **SENTENCIAS:**

Java:



SQL:



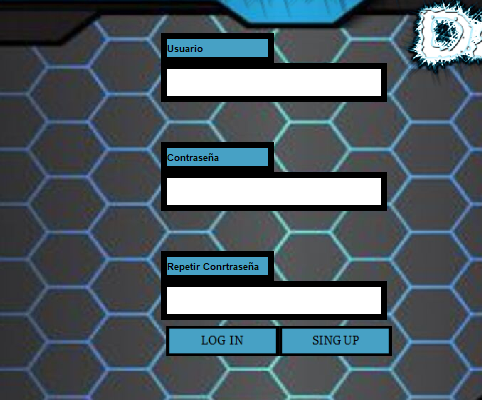
# **USUARIOS**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **CREAR USUARIOS:**

### **EN QUÉ MOMENTO:**

Cuando el usuario esté en la pantalla de Registrarse podrá crear un nuevo usuario, java pedirá nombre usuario, contraseña y repetir contraseña. Desde java comprobaremos que las contraseñas sean iguales y llamaremos a la procedure ‘crearUsuario’ pasándole como parámetros el nombre de usuario y contraseña.

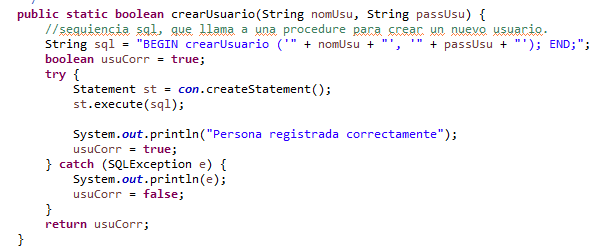


### 

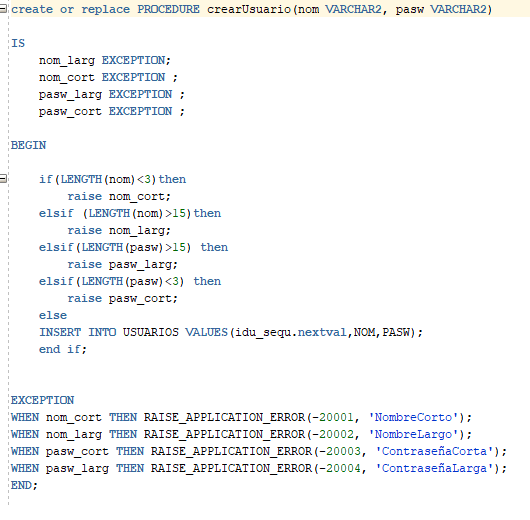
### 

### 

### **SENTENCIAS:**

Java:

SQL:



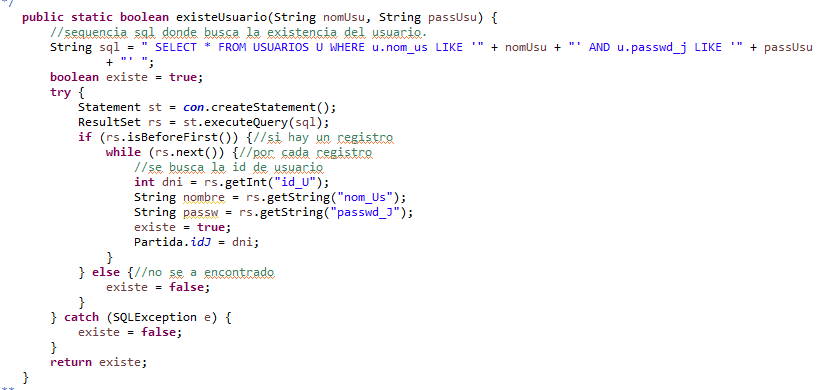
## **CONSULTAR USUARIOS:**

### **EN QUÉ MOMENTO:**

Solo iniciar el juego nos pedirá iniciar sesión. Cuando el usuario introduzca los datos y dé al botón ‘login’ se hará un select donde se comprobará si el usuario existe y la contraseña es correcta. Si los datos de usuario son correctos se iniciará el menú, sino enseñará un mensaje de error conforme los datos del usuario no son correctos.



### **SENTENCIAS:**



Java:

SQL:

