

# BD RANKING PANDEMIC

## TABLA A CREAR

### TABLA JUGADOR

**JUGADOR**(usuario, nombre, contrasena, numVictorias)

usuario	VARCHAR2(35)	PK
nombre	VARCHAR2(35)	NN
contrasena	VARCHAR2(128)	NN
numVictorias	NUMBER	NN Valor por defecto: 0 En este campo se registra el nº de victorias del jugador.

## SENTENCIA DE CREACIÓN DE LA TABLA

Esta es la sentencia para la creación de la tabla.

```
CREATE TABLE JUGADOR(  
    usuario VARCHAR2(35) PRIMARY KEY,  
    nombre VARCHAR2(35) NOT NULL,  
    contrasena VARCHAR2(128) NOT NULL,  
    numVictorias NUMBER DEFAULT 0 NOT NULL  
);
```

Script Output x

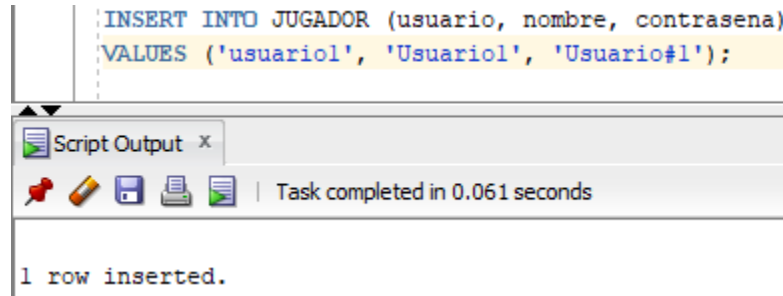
Task completed in 0.105 seconds

Table JUGADOR created.

## INSTRUCCIONES PARA INSERTAR DATOS

Se hará un insert a la hora de crear un nuevo Jugador, donde se indicará el usuario, que es el nombre que introduzca el JUGADOR pero todo en minúsculas, el nombre y la contraseña, el número de victorias no hace falta ya que por defecto se le asignara 0.

```
INSERT INTO JUGADOR (usuario, nombre, contrasena)
VALUES ('usuariol', 'Usuariol', 'Usuario#1');
```

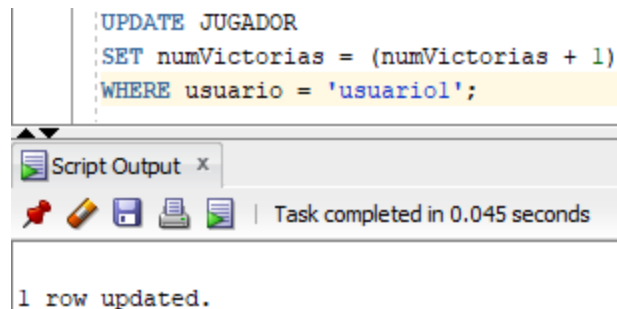


The screenshot shows a SQL script execution window. The script is: `INSERT INTO JUGADOR (usuario, nombre, contrasena) VALUES ('usuariol', 'Usuariol', 'Usuario#1');`. The output pane shows '1 row inserted.' and a status bar indicates 'Task completed in 0.061 seconds'.

## INSTRUCCIONES PARA ACTUALIZAR DATOS

Se actualizarán los datos cuando el usuario gane una partida, en ese caso el número de victorias del usuario incrementara.

```
UPDATE JUGADOR
SET numVictorias = (numVictorias + 1)
WHERE usuario = 'usuariol';
```



The screenshot shows a SQL script execution window. The script is: `UPDATE JUGADOR SET numVictorias = (numVictorias + 1) WHERE usuario = 'usuariol';`. The output pane shows '1 row updated.' and a status bar indicates 'Task completed in 0.045 seconds'.

## INSTRUCCIONES PARA BORRAR DATOS

En caso de querer borrar algún jugador usaremos la siguiente sentencia para eliminarlo.

```
DELETE FROM JUGADOR
WHERE usuario LIKE 'usuario3';
```

Script Output x

Task completed in 0.044 s

1 row deleted.

## INSTRUCCIONES PARA RECUPERAR INFORMACION DEL RANKING

La sentencia que vamos a usar para mostrar el ranking es la siguiente, un select del nombre y número de victorias del jugador ordenado de forma descendiente, así mostrara primero los jugadores que tengan más victorias.

```
SELECT nombre, numVictorias
FROM JUGADOR
ORDER BY numVictorias DESC;
```

Script Output x

Task completed in 0.096 seconds

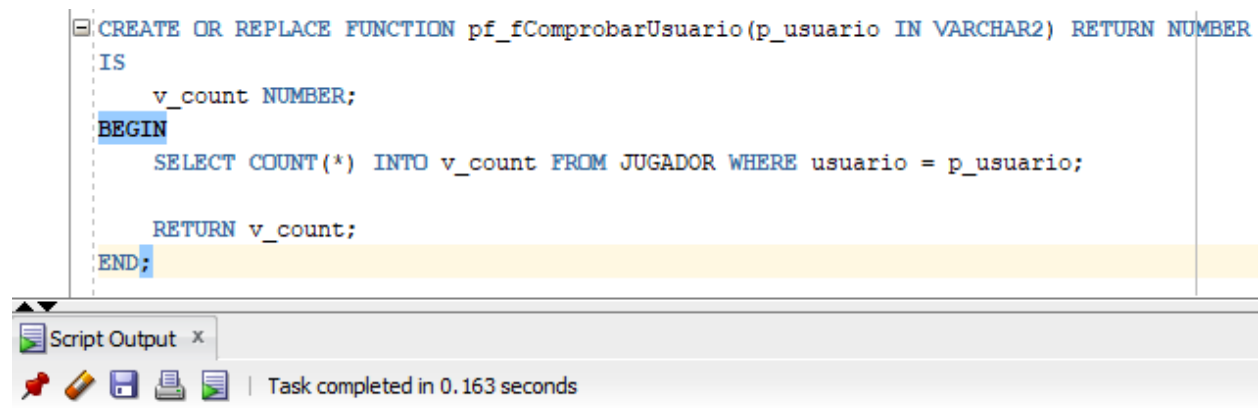
NOMBRE	NUMVICTORIAS
Usuariol	2
Usuario2	0
usuario3	0
Daniel	0

## **FUNCION COMPROBAR USUARIO**

A esta función se le va a pasar como parámetro un nombre de usuario va a hacer un select de la cantidad de JUGADORES con el nombre de usuario que le hayamos pasado. Esta función nos va a servir para saber si un nombre de usuario esta en uso o no. En caso de que el select devuelva 1 significara que el usuario ya esta en uso, si devuelve 0 significara que el usuario está disponible.

```
CREATE OR REPLACE FUNCTION pf_fComprobarUsuario(p_usuario IN VARCHAR2) RETURN NUMBER
IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM JUGADOR WHERE usuario = p_usuario;

    RETURN v_count;
END;
```

The image shows a screenshot of a SQL development environment. The top pane contains a SQL script to create or replace a function named pf\_fComprobarUsuario. The function takes a parameter p\_usuario of type VARCHAR2 and returns a NUMBER. The logic inside the function is to select the count of rows from a table named JUGADOR where the usuario matches p\_usuario, and then return that count. The script is highlighted in yellow. Below the script pane, there is a 'Script Output' pane with a close button. It shows a status bar with icons for a pin, a pencil, a save icon, a print icon, and a refresh icon, followed by the text 'Task completed in 0.163 seconds'.

Function PF\_FCOMPROBARUSUARIO compiled

## **FUNCION INICIAR SESION**

A esta función le vamos a pasar como parámetro un nombre de usuario y una contraseña con la que vamos a hacer un select de la cantidad de JUGADORES que tengan ese nombre de usuario y contraseña. Esta función nos va a servir para saber si existe algún jugador con esas credenciales. En caso de que el select devuelva 1 significara que las credenciales son correctas, si devuelve 0 significara que el usuario no existe o la contraseña está mal escrita.

```
CREATE OR REPLACE FUNCTION pf_fIniciarSesionJugador(p_usuario IN VARCHAR2, p_contraseña IN VARCHAR2) RETURN NUMBER
IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM JUGADOR WHERE usuario = p_usuario AND contraseña = p_contraseña;

    RETURN v_count;
END;
```

Script Output x

Task completed in 0.276 seconds

Function PF\_FINICIARSESIONJUGADOR compiled