****

**IES AUGUSTO GONZÁLEZ DE LINARES**

**DEPARTAMENTO DE INFORMÁTICA**

**Anexo I**

**Sentencias SQL del proyecto**

PROYECTO FINAL 1º CURSO

GRADO SUPERIOR DE DESARROLLO DE APLICACIONES MULTIPLATAFORMA

2022/2023

Índice

[1. Documentación BBDD 2](#_Toc134039213)

[1.1. Apartado de View 2](#_Toc134039214)

[1.2. Apartado Insert 4](#_Toc134039215)

[1.3. Sentencias dentro del proyecto, en el Paquete BBDD 7](#_Toc134039216)

[1.3.1. Clase MetodosAjustesBBDD 7](#_Toc134039217)

[1.3.2. Clase MetodosAlumnoBBDD 7](#_Toc134039218)

[1.3.3. Clase MetodosCalsificacionBBDD 8](#_Toc134039219)

[1.3.4. Clase MetodosJuegoBBDD 8](#_Toc134039220)

[1.3.5. Clase MetodosRegistro 11](#_Toc134039221)

[1.3.6. Clase MetodosUpdateBBDD 13](#_Toc134039222)

# Documentación BBDD

# Apartado de View

* + 1. **Vista Resto\_Div**:

contiene la información necesaria para actualizar la información presentada a los alumnos en el Juego Resto de una división entera

Sentencia utilizada en SQL:

Create view view\_resto\_div\_tablas as select alias, time\_to\_sec(tiempo\_partida) as tiempo\_partida, aciertos, fecha\_hora, nivel, id\_usuario from resto\_div inner join usuario on id\_usuario = jugador inner join reto on id\_reto=reto;

Contraparte

Drop view view\_resto\_div\_tablas;

* + 1. **Vista Euclides**:

contiene la información necesaria para actualizar la información presentada a los alumnos en el Juego Euclides

Sentencia utilizada en SQL:

Create view view\_euclides\_tablas as select alias, time\_to\_sec(tiempo\_partida) as tiempo\_partida, aciertos, fecha\_hora, nivel, id\_usuario from euclides inner join usuario on id\_usuario = jugador inner join reto on id\_reto=reto;

Contraparte

Drop view view\_euclides\_tablas;

* + 1. **Vista calculo:**

Contiene la información necesaria para actualizar la información presentada a los alumnos en el Juego Calculo

Sentencia utilizada en SQL:

Create view view\_calculo\_tablas as select alias, time\_to\_sec(tiempo\_partida) as tiempo\_partida, aciertos, fecha\_hora,nivel,id\_usuario from calculo inner join usuario on id\_usuario = jugador inner join reto on id\_reto=reto;

Contraparte

Drop view view\_calculo\_tablas;

* + 1. **Vista estadísticas RestoDiv:**

Genera las estadísticas de los jugadores agrupando por nivel y jugador en función del juego

Sentencia utilizada en SQL:

create view estadisticas\_resto\_div as select jugador,reto,nivel,count(\*) as partidas,sum(aciertos) as totalaciertos, truncate(((sum(aciertos)\*5) / (count(\*)\*5)),2) as mediaAciertos , nombre, apellido1,apellido2

from resto\_div inner join reto on id\_reto = reto inner join persona on jugador = id\_usuario group by jugador,reto order by jugador;

Contraparte

Drop view estadisticas\_resto\_div;

* + 1. **Vista Estadísticas Euclides:**

Genera las estadísticas de los jugadores agrupando por nivel y jugador en función del juego

create view estadisticas\_euclides as select jugador,reto,nivel,count(\*) as partidas,sum(aciertos) as totalaciertos, truncate(((sum(aciertos)\*5) / (count(\*)\*5)),2) as mediaAciertos , nombre, apellido1,apellido2

from euclides inner join reto on id\_reto = reto inner join persona on jugador = id\_usuario group by jugador,reto order by jugador;

Contraparte

Drop view estadisticas\_euclides;

* + 1. **Vista Estadísticas Calculo:**

Genera las estadísticas de los jugadores agrupando por nivel y jugador en función del juego

Sentencia utilizada en SQL:

create view estadisticas\_calculo as select jugador,reto,nivel,count(\*) as partidas,sum(aciertos) as totalaciertos, truncate(((sum(aciertos)\*25) / (count(\*)\*25)),2) as mediaAciertos , nombre, apellido1,apellido2

from calculo inner join reto on id\_reto = reto inner join persona on jugador = id\_usuario group by jugador,reto order by jugador;

Contraparte

Drop view estadisticas\_calculo;

* + 1. **Vista Info Alumnos:**

Sentencia utilizada en SQL:

create view view\_info\_alumnos as select nombre\_usuario,alias,nombre,apellido1,apellido2,dni,email,curso,fecha\_naci

from usuario inner join persona using(id\_usuario) where profesor=0;

Contraparte

Drop view view\_info\_alumnos;

# Apartado Insert

* + 1. **Inserta un usuario nuevo a la tabla Usuario:**

Sentencia utilizada en SQL:

insert into usuario (nombre\_usuario,contrasena,alias) values

('Cbarrios',md5(12345),'AliasKaren');

Contraparte

delete from usuario where nombre\_usuario = 'Cbarrios';

* + 1. **Inserta los datos personales del Usuario en la tabla Persona:**

Sentencia utilizada en SQL:

insert into persona(id\_usuario,nombre,apellido1,apellido2,dni,curso,fecha\_naci,profesor,email) values

((select id\_usuario from usuario where nombre\_usuario='Cbarrios'),'Carmen','Barrios','Fernández','12345678T',1,20000419,0,'123@asd.es');

Contraparte

delete from persona where id\_usuario = (select id\_usuario from usuario where nombre\_usuario='Cbarrios');

* + 1. **Guarda para cada usuario los ajustes de la aplicación en la tabla Ajustes:**

Sentencia utilizada en SQL:

insert into ajuste (usuario,sonido) values

((select id\_usuario from usuario where nombre\_usuario='Cbarrios'),0);

Contraparte

delete from ajuste where usuario = (select id\_usuario from usuario where nombre\_usuario='Cbarrios');

* + 1. **Añade los nombres del Reto (juego) y su nivel correspondiente:**

Sentencia utilizada en SQL:

insert into reto (nombre\_reto,nivel) values ('calculo',1);

Contraparte

delete from reto where nombre\_reto = 'calculo' and nivel = 1;

* + 1. **Datos generados aleatoriamente para la tabla calculo:**

Sentencia utilizada en SQL:

insert into calculo(jugador,reto,fecha\_hora,aciertos,tiempo\_partida) values

((select id\_usuario from usuario where nombre\_usuario='Cbarrios'),(select id\_reto from reto inner join persona on nivel = curso inner join usuario using (id\_usuario)

where nombre\_reto = 'calculo' and nombre\_usuario = 'Cbarrios'),'2023-03-23 15:09:30',17,sec\_to\_time(131));

Contraparte

delete from calculo where jugador = (select id\_usuario from usuario where nombre\_usuario='Cbarrios') and fecha\_hora = '2023-03-23 15:09:30';

* + 1. **Datos generados aleatoriamente para la tabla resto\_div:**

Sentencia utilizada en SQL:

insert into resto\_div(jugador,reto,fecha\_hora,aciertos,tiempo\_partida) values

((select id\_usuario from usuario where nombre\_usuario='Cbarrios'),(select id\_reto from reto inner join persona on nivel = curso inner join usuario using (id\_usuario)

where nombre\_reto = 'resto\_div' and nombre\_usuario = 'Cbarrios'),'2023-01-07 18:02:03',4,sec\_to\_time(34));

Contraparte

delete from resto\_div where jugador = (select id\_usuario from usuario where nombre\_usuario='Cbarrios') and fecha\_hora = '2023-01-07 18:02:03';

* + 1. **Datos generados aleatoriamente para la tabla Euclides.**

Sentencia utilizada en SQL:

insert into euclides(jugador,reto,fecha\_hora,aciertos,tiempo\_partida) values

((select id\_usuario from usuario where nombre\_usuario='Cbarrios'),(select id\_reto from reto inner join persona on nivel = curso inner join usuario using (id\_usuario)

where nombre\_reto = 'resto\_div' and nombre\_usuario = 'Cbarrios'),'2023-01-07 18:02:03',4,sec\_to\_time(34));

Contraparte

delete from euclides where jugador = (select id\_usuario from usuario where nombre\_usuario='Cbarrios') and fecha\_hora = '2023-01-07 18:02:03';

# Sentencias dentro del proyecto, en el Paquete BBDD

# Clase MetodosAjustesBBDD

* + - 1. **Metodo guardarAjustes:**

Sentencia utilizada en SQL:

update ajuste set tamaño = 2 ,idioma= 'Espanol',sonido=0 where usuario= (select id\_usuario from usuario where nombre\_usuario='Cbarrios');

* + - 1. **Metodo cargarAjustes**

Sentencia utilizada en SQL:

select tamaño,idioma,sonido from ajuste where usuario = 1;

# Clase MetodosAlumnoBBDD

* + - 1. **Metodo selectAlumnos**:

Devuelve un ArrayList, se utiliza para obtener toda la información de lo alumnos y mostrarlo en el apartado del profesor.

En función de lo que el usuario introduce en la aplicación se modifican los parámetros

Sentencia utilizada en la aplicación:

String select = "select nombre\_usuario,alias,nombre,apellido1,apellido2,dni,email,curso,fecha\_naci from view\_info\_alumnos where nombre\_usuario like ? and alias like ? and nombre like ? and apellido1 like ? and apellido2 like ? and dni like ? and email like ? and curso like ?";

Sentencia utilizada en SQL:

select nombre\_usuario,alias,nombre,apellido1,apellido2,dni,email,curso,fecha\_naci from view\_info\_alumnos

where nombre\_usuario like '%%' and alias like '%%' and nombre like '%%' and apellido1 like '%%' and apellido2 like '%%' and dni like '%%' and email like '%%' and curso like '%%';

# Clase MetodosCalsificacionBBDD

* + - 1. **Metodo selectClasificacion:**

Devuelve un ArrayList, se utiliza para obtener toda la información de los alumnos relativa a las estadísticas y para mostrarlo en el apartado del profesor.

Recibe como parámetro el nombre del juego y en función de lo que el usuario introduce en la aplicación se modifican los parámetros

Sentencia utilizada en la aplicación:

String select = "select nivel,partidas,totalaciertos,mediaAciertos,nombre,apellido1,apellido2 from estadisticas\_"+juego+" where nombre like ? and apellido1 like ? and apellido2 like ? and nivel like ?";

Sentencia utilizada en SQL:

select nivel,partidas,totalaciertos,mediaAciertos,nombre,apellido1,apellido2 from estadisticas\_calculo where nombre like '%%' and apellido1 like '%%' and apellido2 like '%%' and nivel like '%%';

# Clase MetodosJuegoBBDD

Esta clase es utilizada por todos los juegos para buscar e insertar datos en la BBDD

Como el Id del usuario se almacena en un objeto con a información del usuario no es necesario realizar Select complejos para obtener esta información

* + - 1. **Método insertResultado**

Se utilizan dos Selects uno para buscar la id del Reto y otro para realizar el Insert en la tabla

Recibe como parámetro el nombre del Juego y el Nivel y devuelve un ArrayList

Sentencia utilizada en la aplicación:

String select = "select id\_reto from reto where nombre\_reto = ? and nivel = ?";

String insert = "insert into " + nombreJuego + "(jugador,reto,aciertos,tiempo\_partida) values (?,?,?,sec\_to\_time(?))";

Sentencia utilizada en SQL:

select id\_reto from reto where nombre\_reto = 'euclides' and nivel = 3;

insert into euclides(jugador,reto,aciertos,tiempo\_partida) values ((select id\_usuario from usuario where nombre\_usuario='Cbarrios'),(select id\_reto from reto where nombre\_reto = 'euclides' and nivel = 3),4,sec\_to\_time(65));

* + - 1. **Método selectJugadorUltimasPartidas:**

Busca en la BBDD las ultimas 5 partidas de el jugador en el nivel seleccionado

Recibe como parámetro el nombre del Juego y el Nivel

Sentencia utilizada en la aplicación:

String select = "select alias, tiempo\_partida, aciertos, fecha\_hora from view\_" + nombreJuego + "\_tablas where id\_usuario = ? and nivel = ? order by fecha\_hora desc limit 5";

Sentencia utilizada en SQL:

select alias, tiempo\_partida, aciertos, fecha\_hora from view\_calculo\_tablas where id\_usuario = (select id\_usuario from usuario where nombre\_usuario='Cbarrios') and nivel = 1 order by fecha\_hora desc limit 5;

* + - 1. **Método selectJugadorMejoresPartidas:**

Busca en la BBDD las mejores 5 partidas del jugador y las ordena de mejor a peor 1º por puntos luego por tiempo dependiendo del nivel seleccionado.

Recibe como parámetro el nombre del Juego y el Nivel y devuelve un ArrayList

Sentencia utilizada en la aplicación:

String select = "select alias, tiempo\_partida, aciertos, fecha\_hora from view\_" + nombreJuego + "\_tablas where id\_usuario = ? and nivel = ? order by aciertos desc, tiempo\_partida asc limit 5";

Sentencia utilizada en SQL:

select alias, tiempo\_partida, aciertos, fecha\_hora from view\_resto\_div\_tablas where id\_usuario = (select id\_usuario from usuario where nombre\_usuario='Cbarrios') and nivel = 1 order by aciertos desc, tiempo\_partida asc limit 5;

* + - 1. **Método selectClasificacion:**

Busca en la BBDD las 5 mejores puntuaciones de todos los jugadores en el nivel seleccionado por el Usuario

Recibe como parámetro el nombre del Juego y el Nivel y devuelve un ArrayList

Sentencia utilizada en la aplicación:

String select = "select alias, tiempo\_partida, aciertos, fecha\_hora from view\_"+nombreJuego+"\_tablas where nivel = ? order by aciertos desc, tiempo\_partida asc limit 5;

Sentencia utilizada en SQL:

select alias, tiempo\_partida, aciertos, fecha\_hora from view\_resto\_div\_tablas where nivel = 2 order by aciertos desc, tiempo\_partida asc limit 5;

* + - 1. **Método totalPartidas:**

Obtiene de la BBDD el total de partidas del Jugador en ese nivel recibe como parámetro el nombre del Juego y el Nivel

Sentencia utilizada en la aplicación:

String select = "SELECT partidas FROM estadisticas\_" +nombreJuego+ " where jugador = ? and nivel = ?";

Sentencia utilizada en SQL:

SELECT partidas FROM estadisticas\_calculo where jugador = (select id\_usuario from usuario where nombre\_usuario='Cbarrios') and nivel = 1;

* + - 1. **Método mediaAciertos:**

Obtiene de la BBDD la media de aciertos del Jugador en ese nivel recibe como parámetro el nombre del Juego y el Nivel

Sentencia utilizada en la aplicación:

"SELECT mediaAciertos FROM estadisticas\_" + nombreJuego + " where jugador = ? and nivel = ?";

Sentencia utilizada en SQL:

SELECT mediaAciertos FROM estadisticas\_calculo where jugador = (select id\_usuario from usuario where nombre\_usuario='Cbarrios') and nivel = 4;

# Clase MetodosRegistro

* + - 1. **Método comprobarDNI:**

Comprueba si el DNI contiene 8 números y Una letra mayúscula, pasado desde el registro y que no existe en la BBDD

Devuelve un Booleano

Sentencia utilizada en la aplicación:

"String selectDNI = "select dni from persona where dni = ?";

Sentencia utilizada en SQL:

select dni from persona where dni = '12345678T';

* + - 1. **Método comprobrarEmail:**

Comprueba si el Email tiene los parámetros correctos (caracteres + "@" +dominio + "." ), pasado desde el registro y que no existe en la BBDD

Devuelve un Booleano

Sentencia utilizada en la aplicación:

"String selectEmail = "select email from persona where email = ?";

Sentencia utilizada en SQL:

select email from persona where email = '123@asd.es';

* + - 1. **Método comprobrarAlias**

Comprueba si el alias no está vacío, pasado desde el registro y que no existe en la BBDD:

Devuelve un Booleano

Sentencia utilizada en la aplicación:

String selectAlias = "select alias from usuario where alias = ?";

Sentencia utilizada en SQL:

select alias from usuario where alias = 'AliasDaniel';

* + - 1. **Método comprobarNombreUsuario**

Comprueba que el nombre de usuario no está vacío, pasado desde el registro y que no existe en la BBDD

Devuelve un Booleano

Sentencia utilizada en la aplicación:

String selectNombreUsuario = "select nombre\_usuario from usuario where nombre\_usuario = ?";

Sentencia utilizada en SQL:

select nombre\_usuario from usuario where nombre\_usuario = 'Despinosa';

* + - 1. **Método registrarUsuario**

Compuesto de un Select para saber el Id-usuario al registrarlo y 3 Inserts para añadir la información del usuario en todas las tablas necesarias para el registro

String select = "select id\_usuario from usuario where nombre\_usuario = ?";

String insert1 = "insert into usuario(nombre\_usuario,contrasena,alias) values (?,md5(?),?)";

String insert2 = "insert into persona(id\_usuario,nombre,apellido1,apellido2,dni,email,curso,fecha\_naci,profesor) values (?,?,?,?,?,?,?,?,?)";

String insert3 = "insert into ajuste(usuario,tamaño,idioma,sonido) values (?,?,?,?)";;

Sentencia utilizada en SQL:

Para añadir el usuario se realizaria en el siguiente orden:

insert into usuario(nombre\_usuario,contrasena,alias) values ('AlumnoPrueba',md5(12345),'AliasAlumPrueba');

select id\_usuario from usuario where nombre\_usuario = 'AlumnoPrueba';

insert into persona(id\_usuario,nombre,apellido1,apellido2,dni,email,curso,fecha\_naci,profesor) values

((select id\_usuario from usuario where nombre\_usuario = 'AlumnoPrueba'),'nombrePru','apellidoPru','Apellido2Prue','12345678X','email@prueba.es',1,20230503,0);

insert into ajuste(usuario,tamaño,idioma,sonido) values ((select id\_usuario from usuario where nombre\_usuario = 'AlumnoPrueba'),2,'Espanol',0);

# Clase MetodosUpdateBBDD

* + - 1. **Metodo updatePassword:**

Método para actualizar la contraseña del usuario en la BBDD

Sentencia utilizada en la aplicación:

String insert1 = "update usuario set contrasena=md5(?) where id\_usuario=" + Usuario.getIdUsuario();

Sentencia utilizada en SQL:

SET @IDUSUARIO = (select id\_usuario from usuario where nombre\_usuario = 'AlumnoPrueba');

update usuario set contrasena=md5('password') where id\_usuario=@IDUSUARIO;

* + - 1. **Metodo updateAlias:**

Método para actualizar la contraseña del usuario en la BBDD

Sentencia utilizada en la aplicación:

String insert1 = "update usuario set alias=? where id\_usuario=" + Usuario.getIdUsuario();

Sentencia utilizada en SQL:

update usuario set alias='Pepito' where id\_usuario = @IDUSUARIO;

* + - 1. **Método updateApellido1:**

Método para actualizar el primer apellido del usuario

Sentencia utilizada en la aplicación:

String insert1 = "update persona set apellido1=? where id\_usuario=" + Usuario.getIdUsuario();

Sentencia utilizada en SQL:

update persona set apellido1='CambioApellido' where (select id\_usuario from usuario where nombre\_usuario = 'AlumnoPrueba');

* + - 1. **Método updateApellido2:**

Método para actualizar el segundo apellido del usuario

Sentencia utilizada en la aplicación:

String insert1 = "update persona set apellido2=? where id\_usuario=" + Usuario.getIdUsuario();

Sentencia utilizada en SQL:

update persona set apellido2='CambioApellido2' where (select id\_usuario from usuario where nombre\_usuario = 'AlumnoPrueba');

* + - 1. **Método updateEmail:**

Método para actualizar el email del usuario

Sentencia utilizada en la aplicación:

String insert1 = "update persona set email=? where id\_usuario=" + Usuario.getIdUsuario();

Sentencia utilizada en SQL:

update persona set email='pruebacorreo@email.es' where id\_usuario=(select id\_usuario from usuario where nombre\_usuario = 'AlumnoPrueba');