



**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**

	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

## Índice

1.	Documentación BBDD.....	2
1.1.	Apartado de View .....	2
1.2.	Apartado Insert .....	4
1.3.	Sentencias dentro del proyecto, en el Paquete BBDD.....	7
1.3.1.	Clase MetodosAjustesBBDD .....	7
1.3.2.	Clase MetodosAlumnoBBDD .....	7
1.3.3.	Clase MetodosCalsificacionBBDD.....	8
1.3.4.	Clase MetodosJuegoBBDD .....	8
1.3.5.	Clase MetodosRegistro.....	11
1.3.6.	Clase MetodosUpdateBBDD.....	13

	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

## 1. Documentación BBDD

### 1.1. Apartado de View

#### 1.1.1. 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;
```

#### 1.1.2. 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;
```

#### 1.1.3. 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;
```

	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

Contraparte

Drop view view\_calculo\_tablas;

#### 1.1.4. 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;

#### 1.1.5. 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;

#### 1.1.6. 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
```

	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

```
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;
```

### 1.1.7. 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;
```

## 1.2. Apartado Insert

### 1.2.1. 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';
```

### 1.2.2. 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@as
d.es');
```

	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

Contraparte

```
delete from persona where id_usuario = (select id_usuario from usuario where
nombre_usuario='Cbarrios');
```

### 1.2.3. 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');
```

### 1.2.4. 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;
```


### 1.2.5. 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';
```

	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

### 1.2.6. 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';
```

### 1.2.7. 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';
```

	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

## 1.3. Sentencias dentro del proyecto, en el Paquete BBDD

### 1.3.1. Clase MetodosAjustesBBDD

#### 1.3.1.1. 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');
```

#### 1.3.1.2. Metodo cargarAjustes

Sentencia utilizada en SQL:

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

### 1.3.2. Clase MetodosAlumnoBBDD

#### 1.3.2.1. 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 '%%';
```



	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

### 1.3.3. Clase MetodosCalsificacionBBDD

#### 1.3.3.1. 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 '%%';
```

### 1.3.4. 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

#### 1.3.4.1. 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;
```

	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

```
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));
```

#### 1.3.4.2. 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;
```

#### 1.3.4.3. 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;
```

#### 1.3.4.4. Método selectClasificacion:

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

	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

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;
```

#### 1.3.4.5. 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 estadísticas_ " + nombreJuego + " where jugador = ? and
nivel = ?";
```

Sentencia utilizada en SQL:

```
SELECT partidas FROM estadísticas_calculo where jugador = (select id_usuario from usuario
where nombre_usuario='Cbarrios') and nivel = 1;
```

#### 1.3.4.6. 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 estadísticas_ " + nombreJuego + " where jugador = ? and nivel =
?";
```

Sentencia utilizada en SQL:

```
SELECT mediaAciertos FROM estadísticas_calculo where jugador = (select id_usuario from
usuario where nombre_usuario='Cbarrios') and nivel = 4;
```

	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

### 1.3.5. Clase MetodosRegistro

#### 1.3.5.1. 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';
```

#### 1.3.5.2. Método comprobarEmail:

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';
```

#### 1.3.5.3. Método comprobarAlias

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';
```

	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

#### 1.3.5.4. 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';
```

#### 1.3.5.5. 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',1,2
0230503,0);
```

	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

```
insert into ajuste(usuario,tamaño,idioma,sonido) values ((select id_usuario from usuario
where nombre_usuario = 'AlumnoPrueba'),2,'Espanol',0);
```

### 1.3.6. Clase MetodosUpdateBBDD

#### 1.3.6.1. 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;
```

#### 1.3.6.2. 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;
```

#### 1.3.6.3. 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');
```

	Ciclo: Desarrollo de Aplicaciones Multiplataforma	Grupo: DAMT1
	Título: Sentencias SQL del proyecto	

#### 1.3.6.4. 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');
```

#### 1.3.6.5. 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');
```