



c

CICLO: FP SUPERIOR 2º DAM
MÓDULO DE ACCESO A DATOS

Tarea Colaborativa

Alumno:
Jairo Martínez Garrido

Desarrollo sobre BD Relacionales

Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.

Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.

Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenecen a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.

Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.

© 2018 Fomento Ocupacional FOC SL todos los derechos reservados.

Contenido

1. Documentos que se adjuntan a este informe	2
2. Resto de epígrafes que componen el desarrollo de este informe	2
3. Componentes del grupo (solo en la tarea colaborativa)	2
Descripción de la tarea:	3
Enunciado de la tarea:	5
Resolución:	8
Modelo Relacional:	9
Creación de tablas en MySql:	11
INSERCIONES DE DATOS EN MySql:.....	14
OPERACIONES CRUD EN JDB CON JAVA:	19

1. Documentos que se adjuntan a este informe.

A continuación se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.
2.

2. Resto de epígrafes que componen el desarrollo de este informe.

A partir de aquí se crearán todos los apartados que sean necesarios para la elaboración del informe de la tarea

Se agregarán tantos epígrafes como se necesiten, poniéndoles siempre el estilo correspondiente para que cuando generemos el índice, Word los agregue a este índice. No olvidar que una vez esté todo debidamente realizado debemos de pasar este informe a formato pdf que será el formato en que se realice la entrega.

3. Componentes del grupo (solo en la tarea colaborativa).

[En el último epígrafe SOLO para la tarea colaborativa, debemos de detallar los componentes del grupo y el trabajo realizado por cada componente. Si uno de los componentes no ha realizado su trabajo, o no ha podido contactar con él, el otro alumno lo hará constar en el informe]

El grupo 02 que realiza esta tarea está compuesto por:

- ✦ David Mitjans Muñoz , 47804194M ○
Parte desarrollada por el alumno B
- ✦ Jairo Martinez Garrido, 76652856C ○ Parte
desarrollada por el alumno A

Descripción de la tarea:

Con esta tarea aprenderás a:

- Utilizar el conector idóneo en la aplicación.
- Establecer la conexión.
- Definir la estructura de la base de datos.
- Desarrollar aplicaciones que modifican el contenido de la base de datos.
- Definir los objetos los destinados a almacenar el resultados de las consulas.
- Desarrollar aplicaciones que efectúan consultas.
- Eliminar los objetos una vez finalizada su función.
- Gestionar las transacciones.

Los resultados de aprendizaje que se trabajan en esta tarea son:

- RA2.- Desarrolla aplicaciones que gestionan información almacenada en bases de datos relacionales identificando y utilizando mecanismos de conexión.

Para realizar esta tarea necesitarás:

- NetBeans, Eclipse u otro IDE Java.
- Sistema gestor de base de datos MySQL.
- Conector JDBC para el SGBD MySQL.
- Procesador de textos.

Los requisitos mínimos específicos exigidos para superar esta tarea son:

- Crear de la BD MySQL e insertar los datos exigidos en el enunciado.
- Intentar contactar con el compañero de la tarea por el foro designado para ello.

- Realizar la documentación con capturas de las pruebas de ejecución.

Sin estos requisitos la tarea se dará por NO SUPERADA y el alumno deberá volver a realizarla ya en segunda entrega con la consiguiente penalización.

Los puntos que tiene esta tarea están asignados de la siguiente forma:

- **Fase Inicial:** Reparto del trabajo por parte de los integrantes del grupo y comunicación al profesor, todo ello a través del foro de la tarea.
- **Fase Intermedia:**
 - Creación de la base de datos del instituto e insertar los datos con los requisitos indicados en la descripción de la tarea : **1 punto.**
 - Implementación de las clases que llevan a cabo operaciones de consulta: **1.5 puntos.**
 - Implementación de las clases que llevan a cabo operaciones de inserción: **1.5 puntos.**
 - Implementación de las clases que llevan a cabo operaciones de actualización: **1.5 puntos.**
 - Implementación de las clases que llevan a cabo operaciones de borrado: **1.5 puntos.**
- **Fase Final:**
 - Documentación explicativa de como se ha llevado a cabo el proceso de creación de la base de datos, así como la explotación de la misma: **1.5 punto.**
 - Dentro de la documentación final debe aparecer un apartado final donde se especifique el trabajo realizado por cada miembro: **0.5 puntos.**

Total: 10 puntos máximo.

NOTA IMPORTANTE:

La participación en el foro de esta tarea tendrá una valoración especial dentro de la nota global de participación del módulo. (supone un 30% de la nota de participación)

Para ello el alumno debe de realizar al menos 5 intervenciones referentes a los siguientes temas:

- - Organización del grupo, planificación del trabajo conjunto, etc.
 - Consultas o aportaciones sobre la implementación de la tarea al compañero/a
 - Información sobre el progreso de la tarea.
 - En general todas las aportaciones de información, documentación, etc que ayuden a la realización de la tarea.

CONTEXTO DE LA TAREA

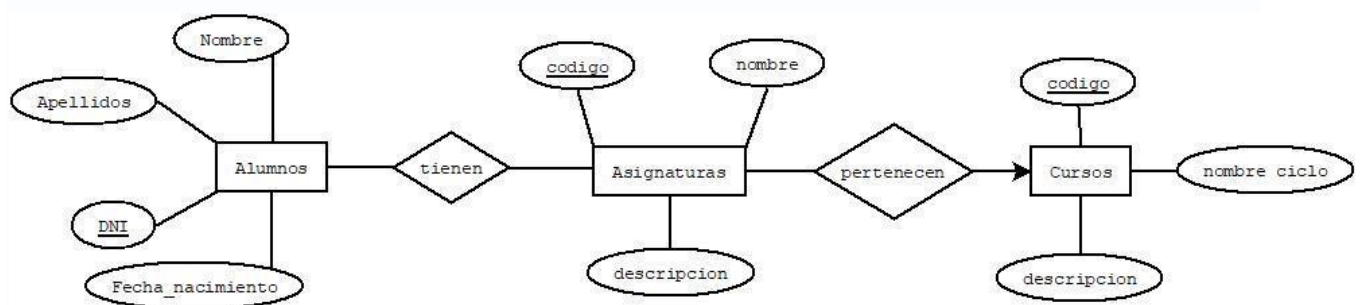
Con esta tarea se va a aprender a trabajar con orígenes de datos relacionales, desde un lenguaje de tercera generación como es Java realizando operaciones de consulta, inserción, actualización y borrado sobre una base de datos relacional.

Para realizar esta tarea es aconsejable revisar los contenidos de las Unidades 1 y 3.

Enunciado de la tarea:

Para abordar esta tarea se va a adoptar una filosofía de trabajo colaborativa, para ello el conjunto de la clase se va a dividir en parejas, que trabajarán conjuntamente en la resolución de la tarea.

Se desea diseñar una base de datos para almacenar y gestionar la información empleada por instituto, teniendo en cuenta el siguiente esquema Entidad-Relación:



Dado el esquema de tablas anterior, se pide:

- Acceder al SGBD MySQL, por línea de comandos haciendo uso del usuario "root" de MySQL.
- Crear una base de datos denominada "instituto".
- Comenzar a usar la base de datos "instituto" que se acaba de crear.
- Una vez creada la base de datos se deben crear las tablas teniendo en cuenta el siguiente reparto de tareas:
 - El miembro A de la pareja se encargará de crear la tabla Alumnos y encargarse de relacionar Alumnos con Asignaturas.
 - El miembro B de la pareja se encargará de crear la tabla Asignaturas y la tabla Cursos.
- Creadas las tablas de la base de datos, se deben insertar registros en este caso el reparto de tareas será el siguiente:
 - Se recomienda ponerse de acuerdo para crear la tabla Asignatura y sus datos ya que deben ser usados por los dos miembros.
 - El miembro A de la pareja deberá insertar al menos 5 alumnos sobre la tabla Alumnos, siendo el primero el miembro A de la pareja y el segundo el miembro B de la pareja. Deberá asignar al menos una asignatura a cada alumno insertado.
 - El miembro B de la pareja deberá insertar 10 asignaturas y 5 cursos, todas las asignaturas deben estar relacionadas con un curso.

- Terminado todo el trabajo, sobre la base de datos se debe crear un proyecto en Netbeans para trabajar con la base de datos del instituto desde Java. En dicho proyecto se deben crear la siguiente clases:
 - Operación de Inserción:
 - ✦ El miembro A de la pareja deberá crear una clase "insertarAlumno.java" cuyo cometido sea el de insertar un nuevo Alumno en la base de datos.
 - ✦ El miembro B de la pareja deberá crear una clase "insertarAsignatura.java" cuyo cometido sea el de insertar una nueva Asignatura en la base de datos.
 - Operación de Actualización:
 - ✦ El miembro A de la pareja deberá crear una clase "modificarFecha.java" que cambie la fecha de nacimiento de un alumno al 01/01/1990.
 - ✦ El miembro B de la pareja deberá crear una clase "modificarCurso.java" que cambie una asignatura de un curso a otro.
 - Operaciones de Consulta:
 - ✦ El miembro A de la pareja deberá crear una clase "consultarAsignaturaAlumno.java" cuyo cometido sea el de consultar el nombre de todos los alumnos y al menos el nombre de la asignatura que tenga asignada cada uno. crear una clase
 - ✦ El miembro B de la pareja deberá crear una clase "consultarAsignaturaCurso.java" cuyo cometido sea el de consultar el nombre de cada asignatura y el nombre de los cursos a las que le pertenecen.
 - Operaciones de Borrado:
 - ✦ El miembro A de la pareja deberá crear una clase "borrarAlumno.java" que deberá borrar un alumno que tenga con asignaturas. crear una clase
 - ✦ El miembro B de la pareja deberá crear una clase "borrarAsignatura.java" que deberá borrar un curso que tenga asignaturas que le pertenezcan a otro curso.

Fase inicial de elaboración conjunta por todos los integrantes del grupo

- El grupo debe decidir un escenario común de trabajo para la resolución de la tarea, estudiando a fondo el paso a tablas del modelo ER y la tabla Asignaturas que va a ser usada por los dos alumnos.
- Marcar un plan de trabajo para resolver la tarea.
- Dividir el enunciado en varias partes para que cada integrante resuelva una. **Lo ideal sería realizar la tarea entre dos alumnos, encargándose uno de la parte de alumnos y otro de la parte de asignaturas y cursos.**

- **Comunicar al profesor dicho escenario y el plan de trabajo a seguir.**

En toda esta fase la comunicación se hará a través del foro de la tarea

Fase intermedia de elaboración individual

- Cada integrante resuelve su parte de la tarea, a partir del escenario común que se ha planteado.
- Mientras se elabora cada parte pueden debatir posibles mejoras a la solución que aporta el compañero.
- Los componentes deberá comunicarse con el compañero a través del foro, avisando de su progreso, consultas, dudas, etc.

Fase final, puesta común y debate de la solución óptima.

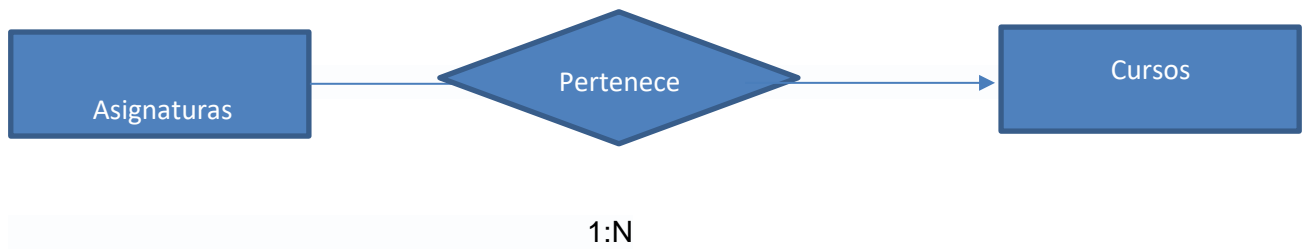
- Toda la puesta en común se hará a través del foro.
- Cada integrante deberá entregar todo el trabajo conjunto haciendo referencia eso sí a la parte que ha desarrollado cada uno.
- Validar entre los dos alumnos los códigos Java generados. Realizar capturas mostrando la ejecución de cada una de las clases.

Los dos integrantes del grupo deben entregar el trabajo completo (el suyo propio y el del compañero). En el informe que se elabore debe aparecer claramente indentificados los dos integrantes del grupo y qué parte del trabajo ha realizado cada uno de los integrantes.

Si alguno no hubiese realizado su parte, el otro integrante deberá realizar la parte común en solitario y lo hará constar en su informe.

Resolución:

Modelo Relacional:



Asignaturas: codigo_asignatura, nombre_asignatura, descripcion, codigo_curso.

Cursos: codigo_curso, nombre_ciclo, descripcion;

Como se trata de una relación (pertenece) 1:N lo que hacemos es propagar la clave primaria de Cursos a Asignaturas. Lo podríamos hacer también con una nueva tabla pertenecen. Donde los campos serían ambas claves primarias.

```
CREATE TABLE Asignaturas(  
  codigo_asignatura INT, nombre_asignatura  
  VARCHAR(50), descripción  
  VARCHAR(300),  
  codigo_curso INT REFERENCES cursos(código_curso),  
  CONSTRAINT codigo_asignatura_pk PRIMARY KEY(codigo_Asignatura)  
);
```

```
CREATE TABLE Cursos(  
  codigo_curso INT, nombre_ciclo  
  VARCHAR(50), descripcion  
  VARCHAR(300),  
  CONSTRAINT codigo_curso_pk PRIMARY KEY(codigo_curso)  
);
```

//Parte Alumno A

Modelo Relacional:**N:M**

Se trata de una relación **N:M** por lo que procedemos a crear una tabla **tienen** intermedia ente las entidades Alumnos y Asignaturas que contendrá las claves primarias y relacionara ambas tablas.

Alumnos: (dni, nombre_alumno,apellidos, fecha_nac)

Asignaturas: (codigo_asignatura, nombre_asignatura, descripcion, codigo_curso)

Tienen: (dni, codigo_asignatura)

```

CREATE TABLE alumnos( dni
VARCHAR(9), nombre_alumno
VARCHAR(50), apellidos
VARCHAR(50), fecha_nac
DATE,
CONSTRAINT dni_pk PRIMARY KEY (dni)
);
  
```

```

CREATE TABLE tienen(
DNI      VARCHAR(9),
codigo_asignatura INT,
CONSTRAINT dni_codigo_asignatura_pk PRIMARY KEY(dni,
codigo_asignatura),
CONSTRAINT dni_fk FOREIGN KEY(dni) REFERENCES alumnos(dni),
CONSTRAINT codigo_asignatura_fk FOREIGN KEY(codigo_asignatura) REFERENCES
asignaturas(codigo_asignatura)
);
  
```

Creación de tablas en MySql:

//Parte Alumno A

```
CREATE TABLE alumnos( dni
VARCHAR(9), nombre_alumno
VARCHAR(50), apellidos
VARCHAR(50), fecha_nac
DATE,
CONSTRAINT dni_pk PRIMARY KEY (dni)
);
```

```
CREATE TABLE tienen(
DNI      VARCHAR(9),
codigo_asignatura INT,
CONSTRAINT dni_codigo_asignatura_pk PRIMARY KEY(dni,
codigo_asignatura),
CONSTRAINT dni_fk FOREIGN KEY(dni) REFERENCES alumnos(dni),
CONSTRAINT codigo_asignatura_fk FOREIGN KEY(codigo_asignatura) REFERENCES
asignaturas(codigo_asignatura)
);
```

//Parte Alumno B

```
CREATE TABLE Asignaturas(  
codigo_asignatura INT, nombre_asignatura  
VARCHAR(50), descripción  
VARCHAR(300),  
codigo_curso INT REFERENCES cursos(codigo_curso),  
CONSTRAINT codigo_asignatura_pk PRIMARY KEY(codigo_Asignatura)  
);
```

//Word me cogió una tilde en descripción y al pasarlo MySql no me di cuenta. Por lo que no se deberían usar tildes. Capaz que debería haber usado otro editor de texto plano al realizar la creación de tablas, que no influyeran autocorrectores de por medio. También se le podría haber llamado -> descripcion_asignaturas para evitar confusión con la descripción de cursos.

```
CREATE TABLE Cursos(  
codigo_curso INT, nombre_ciclo  
VARCHAR(50), descripcion  
VARCHAR(300),  
CONSTRAINT codigo_curso_pk PRIMARY KEY(codigo_curso)  
);
```

Una vez realizado el esbozo de todas las tablas las creamos en MySql accediendo como usuario root:

```
mysql> CREATE DATABASE instituto;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> USE DATABASE instituto;  
ERROR 1049 (42000): Unknown database 'database'  
mysql> USE instituto;  
Database changed
```

```
mysql> CREATE TABLE alumnos(  
-> dni VARCHAR(9),  
-> nombre_alumno VARCHAR(50),  
-> apellidos VARCHAR(50),  
-> fecha_nac DATE,  
-> CONSTRAINT dni_pk PRIMARY KEY (dni)  
-> );  
Query OK, 0 rows affected (0.14 sec)  
  
mysql>  
mysql>  
mysql> CREATE TABLE tienen(  
-> DNI VARCHAR(9),  
-> codigo_asignatura INT,  
-> CONSTRAINT dni_codigo_asignatura_pk PRIMARY KEY(dni, codigo_asignatura),  
-> CONSTRAINT dni_fk FOREIGN KEY(dni) REFERENCES alumnos(dni),  
-> CONSTRAINT codigo_asignatura_fk FOREIGN KEY(codigo_asignatura) REFERENCES asignaturas(codigo_asignatura)  
-> );  
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> CREATE TABLE Asignaturas(  
-> codigo_asignatura INT,  
-> nombre_asignatura VARCHAR(50),  
-> descripción VARCHAR(300),  
-> codigo_curso INT REFERENCES cursos(código_curso),  
-> CONSTRAINT codigo_asignatura_pk PRIMARY KEY(codigo_Asignatura)  
-> );  
Query OK, 0 rows affected (0.07 sec)  
  
mysql>  
mysql> CREATE TABLE Cursos(  
-> codigo_curso INT,  
-> nombre_ciclo VARCHAR(50),  
-> descripcion VARCHAR(300),  
-> CONSTRAINT codigo_curso_pk PRIMARY KEY(codigo_curso)  
-> );  
Query OK, 0 rows affected (0.07 sec)
```

INSERCIONES DE DATOS EN MySql:

Realizamos la inserción de datos a las tablas ya creadas:

//Parte Alumno A

Alumnos:

```
INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
VALUES('76652856C','Jairo','Martinez Garrido','1991-07-19');
INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
VALUES('47804194M','David','Mitjans Muñoz','1992-04-18');

INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
VALUES('75162623A','Yaiza','Rodriguez Heredero','1986-09-17');

INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
VALUES('67802271B','Daviz','Sanchez Sanchez','1989-03-12');

INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
VALUES('75298315C','Laura','Mesonez Ruiz','2019-04-18');

INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
VALUES('09214315K','Fernando','Granados Roas','1993-06-24');

INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
VALUES('50398315L','Abdon','Delgado Garcia','1996-07-12');

INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
VALUES('47298315D','Francisco','Jimenez Ruiz','1998-08-15');

INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
VALUES('64838315C','Alejandra','Torres Hurtado','2000-02-7');

INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
VALUES('73398310C','Maria','Mendez Nuñez','2001-01-10');
```


tienen:

```
INSERT INTO tienen (dni, codigo_asignatura)VALUES ('76652856C',1);
INSERT INTO tienen (dni, codigo_asignatura)VALUES ('47804194M',2);
INSERT INTO tienen (dni, codigo_asignatura)VALUES ('75162623A',3);
INSERT INTO tienen (dni, codigo_asignatura)VALUES ('67802271B',4);
INSERT INTO tienen (dni, codigo_asignatura)VALUES ('75298315C',5);
INSERT INTO tienen (dni, codigo_asignatura)VALUES ('09214315K',6);
INSERT INTO tienen (dni, codigo_asignatura)VALUES ('50398315L',7);
INSERT INTO tienen (dni, codigo_asignatura)VALUES ('47298315D',8);
INSERT INTO tienen (dni, codigo_asignatura)VALUES ('64838315C',9);
INSERT INTO tienen (dni, codigo_asignatura)VALUES ('73398310C',10);
```

```
mysql> INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
-> VALUES('47804194M','David','Mitjans Muñoz','1992-04-18');
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
-> VALUES('75162623A','Yaiza','Rodriguez Heredero','1986-09-17');
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
-> VALUES('67802271B','Daviz','Sanchez Sanchez','1989-03-12');
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
-> VALUES('75298315C','Laura','Mesonez Ruiz','2019-04-18');
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
-> VALUES('09214315K','Fernando','Granados Roas','1993-06-24');
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO alumnos(dni, nombre_alumno, apellidos, fecha_nac)
-> VALUES('50398315L','Abdon','Delgado Garcia','1996-07-12');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO tienen (dni, codigo_asignatura)VALUES ('76652856C',1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO tienen (dni, codigo_asignatura)VALUES ('47804194M',2);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO tienen (dni, codigo_asignatura)VALUES ('75162623A',3);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO tienen (dni, codigo_asignatura)VALUES ('67802271B',4);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO tienen (dni, codigo_asignatura)VALUES ('75298315C',5);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO tienen (dni, codigo_asignatura)VALUES ('09214315K',6);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO tienen (dni, codigo_asignatura)VALUES ('50398315L',7);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO tienen (dni, codigo_asignatura)VALUES ('47298315D',8);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO tienen (dni, codigo_asignatura)VALUES ('64838315C',9);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO tienen (dni, codigo_asignatura)VALUES ('73398310C',10);
Query OK, 1 row affected (0.00 sec)
```

//Parte Alumno B

- **CURSOS:**

INSERT INTO Cursos VALUES (1, 'DAM', 'Desarrollo de aplicaciones multiplataforma');

INSERT INTO Cursos VALUES (2, "DAW", "Desarrollo de aplicaciones Web");

INSERT INTO Cursos VALUES (3, "BD", "Big Data");

INSERT INTO Cursos VALUES (4, "ASIR", "Administracion de sistemas informaticos en red");

INSERT INTO Cursos VALUES (5, "DV", "Desarrollo de videojuegos");

```
mysql> INSERT INTO Cursos VALUES (1, 'DAM', 'Desarrollo de aplicaciones multiplataforma');
Query OK, 1 row affected (0.07 sec)
```

```
mysql> INSERT INTO Cursos VALUES (2, "DAW", "Desarrollo de aplicaciones Web");
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO Cursos VALUES (3, "BD", "Big Data");
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO Cursos VALUES (4, "ASIR", "Administracion de sistemas informaticos en red");
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO Cursos VALUES (5, "DV", "Desarrollo de videojuegos");
Query OK, 1 row affected (0.00 sec)
```

- **ASIGNATURAS:**

```
INSERT INTO Asignaturas VALUES (1, "AD", "Acceso a Datos", 1);
INSERT INTO Asignaturas VALUES (2, "PMDM", "Programacion Dispositivos Moviles", 1);
INSERT INTO Asignaturas VALUES (3, "DW", "Desarrollo Web", 2);
INSERT INTO Asignaturas VALUES (4, "SW", "Servicios Web", 2);
INSERT INTO Asignaturas VALUES (5, "ADD", "Analisis De Datos", 3);
INSERT INTO Asignaturas VALUES (6, "ADI", "Almacenamiento De Informacion", 3);
INSERT INTO Asignaturas VALUES (7, "MEI", "Mantenimiento Equipos Informaticos", 4);
INSERT INTO Asignaturas VALUES (8, "SI", "Sistemas Informaticos", 4);
INSERT INTO Asignaturas VALUES (9, "DA", "Desarrollo En Android", 5); INSERT
INTO Asignaturas VALUES (10, "DU", "Desarrollo en Unity", 5);
```

```
mysql> INSERT INTO Asignaturas VALUES (1, "AD", "Acceso a Datos", 1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Asignaturas VALUES (2, "PMDM", "Programacion Dispositivos Moviles", 1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Asignaturas VALUES (3, "DW", "Desarrollo Web", 2);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Asignaturas VALUES (4, "SW", "Servicios Web", 2);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Asignaturas VALUES (5, "ADD", "Analisis De Datos", 3);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Asignaturas VALUES (6, "ADI", "Almacenamiento De Informacion", 3);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Asignaturas VALUES (7, "MEI", "Mantenimiento Equipos Informaticos", 4);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Asignaturas VALUES (8, "SI", "Sistemas Informaticos", 4);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Asignaturas VALUES (9, "DA", "Desarrollo En Android", 5);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Asignaturas VALUES (10, "DU", "Desarrollo en Unity", 5);
Query OK, 1 row affected (0.00 sec)
```

OPERACIONES CRUD EN JDB CON JAVA:

//Parte Alumno A

- **INSERCIÓN:**

El miembro A de la pareja deberá crear una clase "insertarAlumno.java" cuyo cometido sea el de insertar un nuevo Alumno en la base de datos.

Antes de realizar las clases hemos de importar la librería JDBC.

```

11  /*
12  public class InsertarAlumno {
13
14      public static void main(String[] args) throws ClassNotFoundException, SQLException {
15
16          //Cargamos el driver
17          Class.forName("com.mysql.jdbc.Driver");
18
19          //Conexion con la base de datos
20          Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/instituto","foc","f0m3nt0");
21
22          //Preparar la sentencia
23          Statement sentencia = conexion.createStatement();
24
25          //Lanzar la sentencia
26          String op = "INSERT INTO alumnos VALUES('2235265A','Jorge','Torres Almadra','1982-5-07')";
27          int filas = sentencia.executeUpdate(op); //Filas insertadas
28
29          //Comprobamos cuantas filas se han insertado
30          System.out.println("Mediante la sentencia de insercion "+op+" se han insertado "+filas+" filas");
31
32          //Liberamos los recursos
33          sentencia.close();
34          conexion.close();
35      }
36  }
37  
```

- Lo que hemos de realizar es cargar el driver desde código de jdbc de mysql.
- Realizaremos la conexión a la base de datos.
- Crearemos una sentencia y la ejecutaremos o recorreremos según el caso que nos pidan.
- Luego imprimiremos los resultados por pantalla.

- Cerrar los flujos y liberar recursos.

(Esto se realizará para todas las clases de manera general y común).

- **MODIFICACIÓN:**

El miembro A de la pareja deberá crear una clase “modificarFecha.java” que cambie la fecha de nacimiento de un alumno a 01/01/1990.

```

17 public class ModificarFecha {
18     public static void main(String[] args) throws ClassNotFoundException, SQLException {
19         //Cargamos el driver
20         Class.forName("com.mysql.jdbc.Driver");
21
22         //Conexion con la base de datos
23         Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/instituto","foc","f0m3nt0");
24
25         //Preparar sentencia
26         Statement sentencia = conexion.createStatement();
27
28         //Lanzar sentencia
29         String op = "UPDATE alumnos SET fecha_nac='1990-01-01' WHERE dni='76652856C'";
30         int filas = sentencia.executeUpdate(op); //filas actualizadas
31
32         //Comprobar cuantas filas se han insertado
33         System.out.println("Mediante la sentencia de actualizacion "+op+" se han modificado "+filas+" filas");
34
35         //Liberar recursos
36         sentencia.close();
37         conexion.close();
38     }
39 }
40

```

Report Problems Window	iReport output	Output - DesarrolloBDRelacionales_TC (run) X
<pre> run: Mediante la sentencia de actualizacion UPDATE alumnos SET fecha_nac='1990-01-01' WHERE dni='76652856C' se han modificado 1 filas BUILD SUCCESSFUL (total time: 0 seconds) </pre>		

En la impresión por consola se aprecia que la ejecución del código ha sido correcta, por lo que los cambios deben de haberse realizado.

- **CONSULTA:**

El miembro A de la pareja deberá crear una clase "consultarAsignaturaAlumno.java" cuyo cometido será mostrar el nombre de todos los alumnos y al menos el nombre de una asignatura que tenga asignada cada uno.

Realizamos la consulta con la aplicación de JOIN, para así unir las tres diferentes tablas que intervienen en el proceso de consulta.

```
SELECT column_name(s)
FROM table1
JOIN table2
ON table1.column_name = table2.column_name
JOIN
ON table2.column_name = table3.column_name;
```

SENTENCIA:

```
SELECT A.nombre_alumno, A.dni, A.fecha_nac, T.codigo_asignatura,
S.nombre_asignatura FROM alumnos A JOIN tienen T ON A.dni = T.dni JOIN
asignaturas S ON T.codigo_asignatura=S.codigo_asignatura;
```

La probamos en MySql antes de operar y como vemos funciona correctamente.

```
mysql> SELECT A.nombre_alumno, A.dni, A.fecha_nac, T.codigo_asignatura, S.nombre_asignatura
-> FROM alumnos A JOIN tienen T ON A.dni = T.dni JOIN asignaturas S ON T.codigo_asignatura=S.codigo_asignatura;
```

nombre_alumno	dni	fecha_nac	codigo_asignatura	nombre_asignatura
Fernando	09214315K	1993-06-24	6	ADI
Francisco	47298315D	1998-08-15	8	SI
David	47804194M	1992-04-18	2	PMDM
Abdon	50398315L	1996-07-12	7	MEI
Alejandra	64838315C	2000-02-07	9	DA
Daviz	67802271B	1989-03-12	4	SW
Maria	73398310C	2001-01-10	10	DU
Yaiza	75162623A	1986-09-17	3	DW
Laura	75298315C	2019-04-18	5	ADD
Jairo	76652856C	1990-01-01	1	AD

```
10 rows in set (0.00 sec)

mysql>
```

```
19 public static void main(String[] args) throws ClassNotFoundException, SQLException {
20     //Cargamos el driver
21     Class.forName("com.mysql.jdbc.Driver");
22
23     //Conexion con la base de datos
24     Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/instituto","foc","f0m3nt0");
25
26     //Preparar la sentencia
27     Statement sentencia = conexion.createStatement();
28
29     //Lanzar la consulta
30     ResultSet resultado = sentencia.executeQuery
31     ("SELECT A.nombre_alumno, A.dni, A.fecha_nac, T.codigo_asignatura, S.nombre_asignatura "
32     + "FROM alumnos A JOIN tienen T ON A.dni = T.dni JOIN asignaturas S ON T.codigo_asignatura=S.codigo_asignatura");
33
34     //Recorremos la coleccion para visualizar cada fila
35     //Usamos un bucle mientras haya registros
36
37     while(resultado.next()){
38         System.out.println("Nombre: "+resultado.getString(1)+ "\t\tDNI: "
39         +resultado.getString(2)+"\t\tFecha nacimiento: "
40         +resultado.getDate(3)+"\t\tCodigo asignatura: "
41         +resultado.getInt(4)+"\t\tNombre asignatura: "
42         +resultado.getString(5));
43     }
44
45     //Liberar recursos
46     resultado.close();
47     sentencia.close();
48     conexion.close();

```

Report Problems Window | iReport output | Output - DesarrolloBDRelacionales_TC (run) X

```
run:
Nombre: Fernando          DNI: 09214315K          Fecha nacimiento: 1993-06-24          Codigo asignatura: 6          Nombre asignatura: ADI
Nombre: Francisco         DNI: 47298315D          Fecha nacimiento: 1998-08-15          Codigo asignatura: 8          Nombre asignatura: SI
Nombre: David              DNI: 47804194M          Fecha nacimiento: 1992-04-18          Codigo asignatura: 2          Nombre asignatura: PMDM
Nombre: Abdon              DNI: 50398315L          Fecha nacimiento: 1996-07-12          Codigo asignatura: 7          Nombre asignatura: MEI
Nombre: Alejandra          DNI: 64838315C          Fecha nacimiento: 2000-02-07          Codigo asignatura: 9          Nombre asignatura: DA
Nombre: Daviz              DNI: 67802271B          Fecha nacimiento: 1989-03-12          Codigo asignatura: 4          Nombre asignatura: SW
Nombre: Maria              DNI: 73398310C          Fecha nacimiento: 2001-01-10          Codigo asignatura: 10         Nombre asignatura: DU
Nombre: Yaiza              DNI: 75162623A          Fecha nacimiento: 1986-09-17          Codigo asignatura: 3          Nombre asignatura: DW
Nombre: Laura              DNI: 75298315C          Fecha nacimiento: 2019-04-18          Codigo asignatura: 5          Nombre asignatura: ADD
Nombre: Jairo              DNI: 76652856C          Fecha nacimiento: 1990-01-01          Codigo asignatura: 1          Nombre asignatura: AD
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ejecutamos la sentencia y recorremos **resultado** mientras haya más elementos en la lista para que sean impresos.

- **BORRADO:**





El miembro A de la pareja deberá crear una clase “borrarAlumno.java” y las relaciones que tenga con asignaturas.

```
17 public class BorrarAlumno {
18     public static void main(String[] args) throws ClassNotFoundException, SQLException {
19
20         //Cargamos el driver
21         Class.forName("com.mysql.jdbc.Driver");
22
23         //Conexion con la base de datos
24         Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/instituto","foc","f0m3nt0");
25
26         //Preparar la sentencia
27         Statement sentencia = conexion.createStatement();
28
29         //Lanzar la sentencia
30         String op = "DELETE FROM a, t USING Alumnos AS a INNER JOIN Tienen AS t WHERE a.dni=t.dni AND a.dni='47298315D'";
31         int filas = sentencia.executeUpdate(op); //filas borradas
32
33         //Comprobar cuantas filas se han borrado
34         System.out.println("Mediante la sentencia de eliminacion "+op+" se han borrado "+filas+" filas");
35
36         //Liberar los recursos
37         sentencia.close();
38         conexion.close();
39     }
40 }
```

Report Problems Window | Report output | Output - DesarrolloBDRelacionales_TC (run) X

run:
Mediante la sentencia de eliminacion DELETE FROM a, t USING Alumnos AS a INNER JOIN Tienen AS t WHERE a.dni=t.dni AND a.dni='47298315D' se han borrado 2 filas
BUILD SUCCESSFUL (total time: 0 seconds)

Como vemos la ejecución del código ha sido correcta por lo que el borrado debería haberse realizado, aun así realizamos una consulta para asegurarnos del borrado del alumno con DNI = 47298315D.

Report Problems Window	iReport output	Output - DesarrolloBDRelacionales_TC (run) X
   	run: Nombre: Fernando Nombre: David Nombre: Abdon Nombre: Alejandra Nombre: Daviz Nombre: Maria Nombre: Yaiza Nombre: Laura Nombre: Jairo BUILD SUCCESSFUL (total time: 0 seconds)	DNI: 09214315K DNI: 47804194M DNI: 50398315L DNI: 64838315C DNI: 67802271B DNI: 73398310C DNI: 75162623A DNI: 75298315C DNI: 76652856C Fecha nacimiento: 1993-06-24 Fecha nacimiento: 1992-04-18 Fecha nacimiento: 1996-07-12 Fecha nacimiento: 2000-02-07 Fecha nacimiento: 1989-03-12 Fecha nacimiento: 2001-01-10 Fecha nacimiento: 1986-09-17 Fecha nacimiento: 2019-04-18 Fecha nacimiento: 1990-01-01 Codigo asignatura: 6 Codigo asignatura: 2 Codigo asignatura: 7 Codigo asignatura: 9 Codigo asignatura: 4 Codigo asignatura: 10 Codigo asignatura: 3 Codigo asignatura: 5 Codigo asignatura: 1 Nombre asignatura: ADI Nombre asignatura: PMDM Nombre asignatura: MEI Nombre asignatura: DA Nombre asignatura: SW Nombre asignatura: DU Nombre asignatura: DW Nombre asignatura: ADD Nombre asignatura: AD

//Parte Alumno B

- INSERCIÓN:**

El miembro B de la pareja deberá crear una clase "insertarAsignatura.java" cuyo cometido sea el de insertar una nueva Asignatura en la base de datos.

Antes de realizar las clases hemos de importar la librería JDBC.

```

15 public class insertarAsignatura {
16
17     public static void main(String[] args) throws ClassNotFoundException, SQLException {
18
19         //Paso 2: Cargar el driver
20         Class.forName("com.mysql.jdbc.Driver");
21
22         //Paso 3: Establer conexion con la BD: URL, Usuario, Pass
23         Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/instituto","root","");
24
25         //Paso 4: Preparar la sentencia.
26         Statement sentencia = conexion.createStatement();
27
28         //Paso 5: Lanzar sentencia.
29         String insercion = "INSERT INTO Asignaturas VALUES (11, 'EIE', 'Empresa e iniciativa emprendedora', 1)";
30
31         int filas = sentencia.executeUpdate(insercion); //filas insertada
32
33         System.out.println("Tras la inserción se han insertado " + filas + " filas");
34
35         //Paso 7: Liberar los recursos
36         sentencia.close();
37         conexion.close();
38     }
39 }

```

- Lo que hemos de realizar es cargar el driver desde código de jdbc de mysql.
- Realizaremos la conexión a la base de datos.
- Crearemos una sentencia y la ejecutaremos o recorreremos según el caso que nos pidan.
- Luego imprimiremos los resultados por pantalla.
- Cerrar los flujos y liberar recursos.

(Esto se realizará para todas las clases de manera general y común).

• MODIFICACION

El miembro B de la pareja deberá crear una clase "modificarCurso.java" que cambie una asignatura de un curso a otro.

```

7 public class modificarCurso {
8     public static void main(String[] args) throws ClassNotFoundException, SQLException {
9         //Paso 2: Cargar el driver
10        Class.forName("com.mysql.jdbc.Driver");
11
12        //Paso 3: Establer conexion con la BD: URL, Usuario, Pass
13        Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/instituto","root","");
14
15        //Paso 4: Preparar la sentencia.
16        Statement sentencia = conexion.createStatement();
17
18        //Paso 5: Lanzar sentencia.
19        String modificacion = "UPDATE Asignaturas SET codigo_curso=2 WHERE codigo_asignatura=11";
20
21        int filas = sentencia.executeUpdate(modificacion); //filas insertada
22
23        System.out.println("Tras la modificacion " + modificacion + " se han actualizado " + filas + " filas");
24
25        //Paso 7: Liberar los recursos
26        sentencia.close();
27        conexion.close();
28    }
29 }

```

Output - AD_TC_2019 (run) ×

```

run:
Tras la modificacionUPDATE Asignaturas SET codigo_curso=2 WHERE codigo_asignatura=11 se han actualizado 1 filas

```

Como vemos la modificación se ha realizado correctamente y se ha pasado la asignatura con código 11(EIE) al curso 2 (DAW).

- **CONSULTA**

El miembro B de la pareja deberá crear una clase "consultarAsignaturaCurso.java" cuyo cometido será mostrar el nombre de cada asignatura y el nombre del curso al que están asignadas.

Realizamos la consulta con un Inner Join para que nos devuelva el valor de las dos tablas.

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

SENTENCIA:

```
SELECT nombre_asignatura, nombre_ciclo FROM Asignaturas INNER JOIN Cursos ON
Asignaturas.codigo_curso = Cursos.codigo_curso;
```

La probamos en MySql antes de operar y como vemos funciona correctamente.

```
mysql> SELECT nombre_asignatura, nombre_ciclo FROM Asignaturas INNER JOIN Cursos
-> ON Asignaturas.codigo_curso = Cursos.codigo_curso;
+-----+-----+
| nombre_asignatura | nombre_ciclo |
+-----+-----+
| AD                | DAM          |
| PMDM             | DAM          |
| DW               | DAW          |
| SW               | DAW          |
| ADD              | BD           |
| ADI              | BD           |
| MEI              | ASIR         |
| SI               | ASIR         |
| DA               | DV           |
| DU               | DV           |
| EIE              | DAW          |
+-----+-----+
11 rows in set (0.07 sec)
```

```

public class consultarAsignaturasCurso {
    public static void main(String[] args) throws ClassNotFoundException, SQLException {

        /**
         * El miembro B de la pareja deberá crear una clase "consultarAsignaturaCurso.java"
         * cuyo cometido será mostrar el nombre de cada asignatura y el nombre del curso al que están asignadas.
         */
        //Paso 2: Cargar el driver
        Class.forName("com.mysql.jdbc.Driver");

        //Paso 3: Establecer conexión con la BD: URL, Usuario, Pass
        Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/instituto","root","");

        //Paso 4: Preparar la sentencia.
        Statement sentencia = conexion.createStatement();

```

Ejecutamos la sentencia y lo recorremos mediante `hasNext()` para imprimir los resultados:

```

//Paso 5: Lanzar sentencia.
ResultSet resultado = sentencia.executeQuery("SELECT nombre_asignatura, descripción, nombre_ciclo FROM Asignaturas "
+ "INNER JOIN Cursos ON Asignaturas.codigo_curso = Cursos.codigo_curso");

//Paso 6: recorreremos la colección para visualizar cada fila
//Hacemos un bucle mientras que haya registros
while(resultado.next()){
    System.out.print("Siglas " + resultado.getString(1) + " -> ");
    System.out.print("Asignatura: " + resultado.getString(2) + " -> ");
    System.out.println("Curso: " + resultado.getString(3) + "\t\t");
    System.out.println("");
}

//Paso 7: Liberar los recursos
resultado.close();
sentencia.close();
conexion.close();

```

```
Debugger Console X AD_TC_2019 (run) X  
run:  
Siglas AD -> Asignatura: Acceso a Datos -> Curso:DAM  
  
Siglas PMDM -> Asignatura: Programacion Dispositivos Moviles -> Curso:DAM  
  
Siglas DW -> Asignatura: Desarrollo Web -> Curso:DAW  
  
Siglas SW -> Asignatura: Servicios Web -> Curso:DAW  
  
Siglas ADD -> Asignatura: Analisis De Datos -> Curso:BD  
  
Siglas ADI -> Asignatura: Almacenamiento De Informacion -> Curso:BD  
  
Siglas MEI -> Asignatura: Mantenimiento Equipos Informaticos -> Curso:ASIR  
  
Siglas SI -> Asignatura: Sistemas Informaticos -> Curso:ASIR  
  
Siglas DA -> Asignatura: Desarrollo En Android -> Curso:DV  
  
Siglas DU -> Asignatura: Desarrollo en Unity -> Curso:DV  
  
Siglas EIE -> Asignatura: Empresa e iniciativa emprendedora -> Curso:DAW  
  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Como apreciamos la consulta de la asignatura, su nombre y al módulo al que pertenece se realiza sin problemas.

- **BORRADO:**

El miembro B de la pareja deberá crear una clase "borrarAsignatura.java" que deberá borrar un curso, asignando las asignaturas que le pertenezcan a otro curso.

```
//Paso 2: Cargar el driver
Class.forName("com.mysql.jdbc.Driver");

//Paso 3: Establecer conexion con la BD: URL, Usuario, Pass
Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/instituto","root","");

//Paso 4: Preparar la sentencia.
Statement sentencia = conexion.createStatement();

//Paso 5a: UPDATEAR que las asignaturas cambien a otro curso.
String update = "UPDATE Asignaturas SET codigo_curso=1 WHERE codigo_curso=5";

//Paso 5b: Lanzar sentencia borrado
String delete = "DELETE FROM Cursos WHERE codigo_curso=5";
//String borrado = "DELETE Cursos ";

int filas = sentencia.executeUpdate(update); //filas insertada
int filasB = sentencia.executeUpdate(delete);

System.out.println("Tras la actualización " + update + " se han modificado " + filas + " asignaturas. \n"
    + "Dejando así el curso vacío de asignaturas");

System.out.println("Tras el borrado " + delete + " \n se han borrado " + filasB + " cursos ");

//Paso 7: Liberar los recursos
sentencia.close();
conexion.close();
}
```

Como vemos la actualización y el posterior borrado se han realizado correctamente.

```
run:
Tras la actualización UPDATE Asignaturas SET codigo_curso=1 WHERE codigo_curso=5 se han modificado 2 asignaturas. Dejando así el curso vacío de asignaturas
BUILD SUCCESSFUL (total time: 0 seconds)
```



```
Debugger Console x AD_TC_2019 (run) x  
run:  
Tras la actualización UPDATE Asignaturas SET codigo_curso=1 WHERE codigo_curso=5 se han modificado 0 asignaturas.  
Dejando así el curso vacío de asignaturas  
Tras el borrado DELETE FROM Cursos WHERE codigo_curso=5  
se han borrado 1 cursos  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Volvemos a consultar los datos:

```
run:  
Siglas AD -> Asignatura: Acceso a Datos -> Curso:DAM  
  
Siglas PMDM -> Asignatura: Programacion Dispositivos Moviles -> Curso:DAM  
  
Siglas DW -> Asignatura: Desarrollo Web -> Curso:DAW  
  
Siglas SW -> Asignatura: Servicios Web -> Curso:DAW  
  
Siglas ADD -> Asignatura: Analisis De Datos -> Curso:BD  
  
Siglas ADI -> Asignatura: Almacenamiento De Informacion -> Curso:BD  
  
Siglas MEI -> Asignatura: Mantenimiento Equipos Informaticos -> Curso:ASIR  
  
Siglas SI -> Asignatura: Sistemas Informaticos -> Curso:ASIR  
  
Siglas DA -> Asignatura: Desarrollo En Android -> Curso:DAM  
  
Siglas DU -> Asignatura: Desarrollo en Unity -> Curso:DAM  
  
Siglas EIE -> Asignatura: Empresa e iniciativa emprendedora -> Curso:DAW  
  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Como vemos el curso DV (Desarrollo de Videojuegos) ha desaparecido y sus correspondientes asignaturas han pasado al curso DAM.