

HITO INDIVIDUAL ACCESO A DATOS

Iván González Gordo DAM2



26 de octubre de 2022

campusfp

GETAFE

**TECNOLOGÍAS UTILIZADAS**

Las tecnologías usadas han sido Java 17 con Maven, MySQL y PostgreSQL para las bases de datos y como IDE IntelliJ de JetBrains.

**SCRIPT MYSQL**

DROP TABLE IF EXISTS `facturas`;

CREATE TABLE `facturas` (

`idfactura` int NOT NULL AUTO\_INCREMENT,

`idusUario` int DEFAULT NULL,

`concepto` varchar(50) DEFAULT NULL,

`fecha` varchar(25) DEFAULT NULL,

PRIMARY KEY (`idfactura`),

KEY `idusUario` (`idusUario`),

CONSTRAINT `facturas\_ibfk\_1` FOREIGN KEY (`idusUario`) REFERENCES `usuarios` (`id`)

);

INSERT INTO `facturas` VALUES (1,1,'el mejor coche','17-12-2022'),(2,1,'la mejor moto','17-12-2022'),(3,2,'el mejor libro','17-12-2022'),(4,1,'TECLADOS PARA CLASE DAM 2','01-05-2021'),(5,1,'RATONES PARA CLASE DAM 2','01-05-2021'),(6,2,'PANTALLAS PARA CLASE DAM 2','01-05-2021'),(7,2,'ORDENADORES PARA CLASE DAM 2','01-05-2021');

DROP TABLE IF EXISTS `usuarios`;

CREATE TABLE `usuarios` (

`id` int NOT NULL AUTO\_INCREMENT,

`nombre` varchar(25) DEFAULT NULL,

`apellidos` varchar(100) DEFAULT NULL,

`NIF` char(9) DEFAULT NULL,

PRIMARY KEY (`id`)

);

INSERT INTO `usuarios` VALUES (1,'carlos','gonzalez','49589604V'),(2,'javier','martin','49589605H'),(3,'javier','gutierrez','49589605A');

**SCRIPT POSRGRESQL**

create table usuarios(

id serial PRIMARY KEY,

nombre varchar(25),

apellidos varchar(100),

NIF char(9)

);

create table facturas(

idfactura serial,

idusUario int,

concepto varchar(50),

fecha varchar(25),

PRIMARY KEY (idfactura),

FOREIGN KEY (idusuario) REFERENCES usuarios(id)

);

insert into usuarios(nombre, apellidos, NIF) values('Pepe', 'Perez Perez', '12345678A');

insert into usuarios(nombre, apellidos, NIF) values('Juan', 'Gomez Gomez', '12345678B');

insert into usuarios(nombre, apellidos, NIF) values('Maria', 'Garcia Garcia', '12345678C');

insert into usuarios(nombre, apellidos, NIF) values('Luis', 'Lopez Lopez', '12345678D');

insert into facturas (idusuario, concepto, fecha) values (1, 'TECLADOS PARA CLASE DAM 2', '01-05-2021');

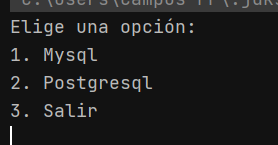
insert into facturas (idusuario, concepto, fecha) values (1, 'RATONES PARA CLASE DAM 2', '01-05-2021');

insert into facturas (idusuario, concepto, fecha) values (2, 'PANTALLAS PARA CLASE DAM 2', '01-05-2021');

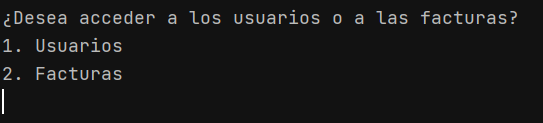
insert into facturas (idusuario, concepto, fecha) values (2, 'ORDENADORES PARA CLASE DAM 2', '01-05-2021');

**Capturas de pantalla**

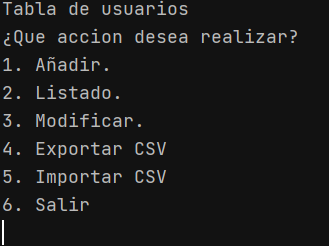
Elección entre acceder un mysql o postgresql:



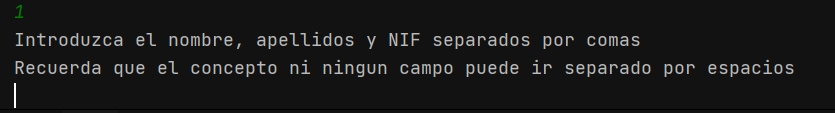
Dentro de cada base de datos puedes acceder a las dos tablas usuarios o facturas:



Si accedes a cualquiera de las dos tablas te indica el nombre de las tablas que vas a usar y un menú con las opciones disponibles:

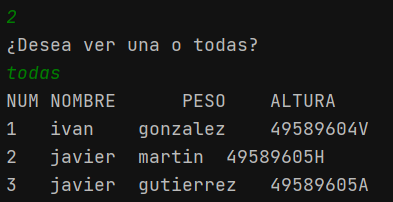


Si eliges añadir:

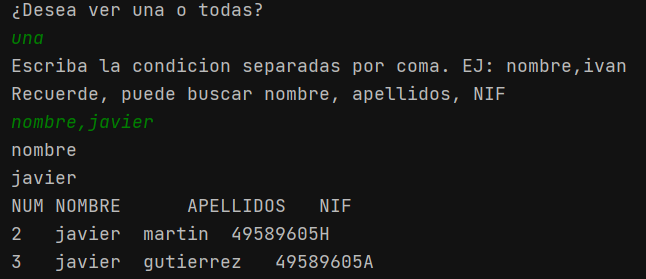


Si eliges listado:

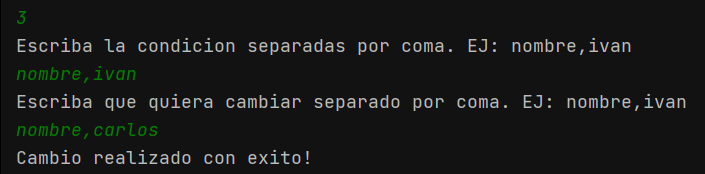
Luego tienes dos opciones: una o todas



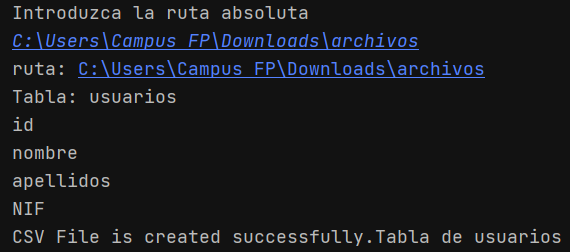
Si deseas ver una tienes que escribir los parámetros de búsqueda:



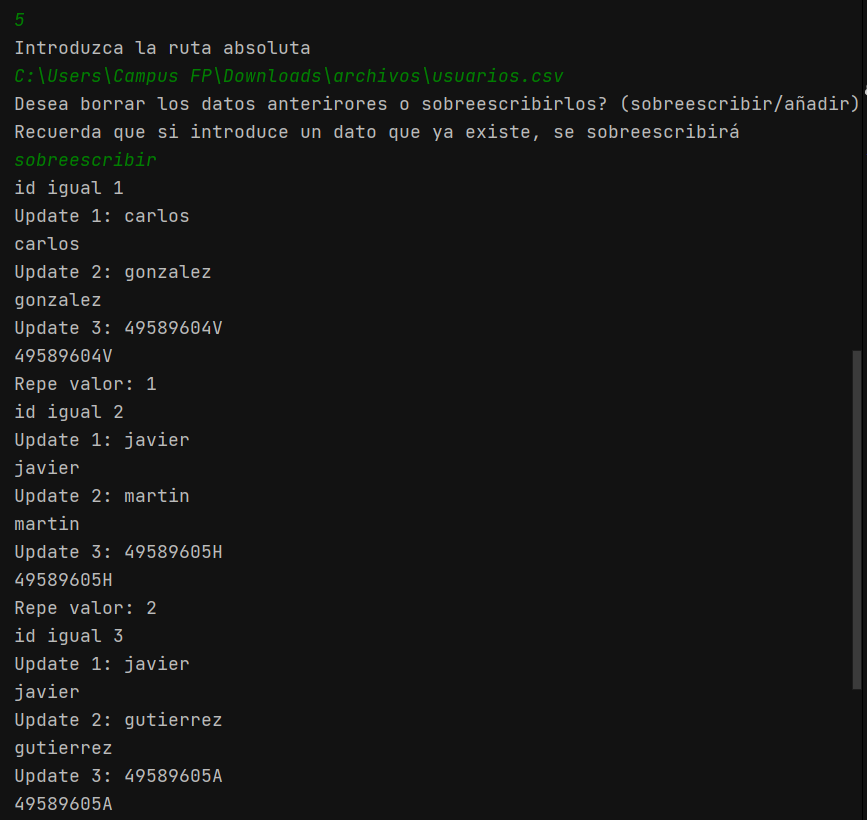
Si eliges la opción de modificar:



Si eliges la opción de exportar CSV:

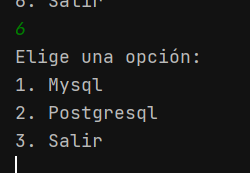


Si eliges la opción de importar CSV:



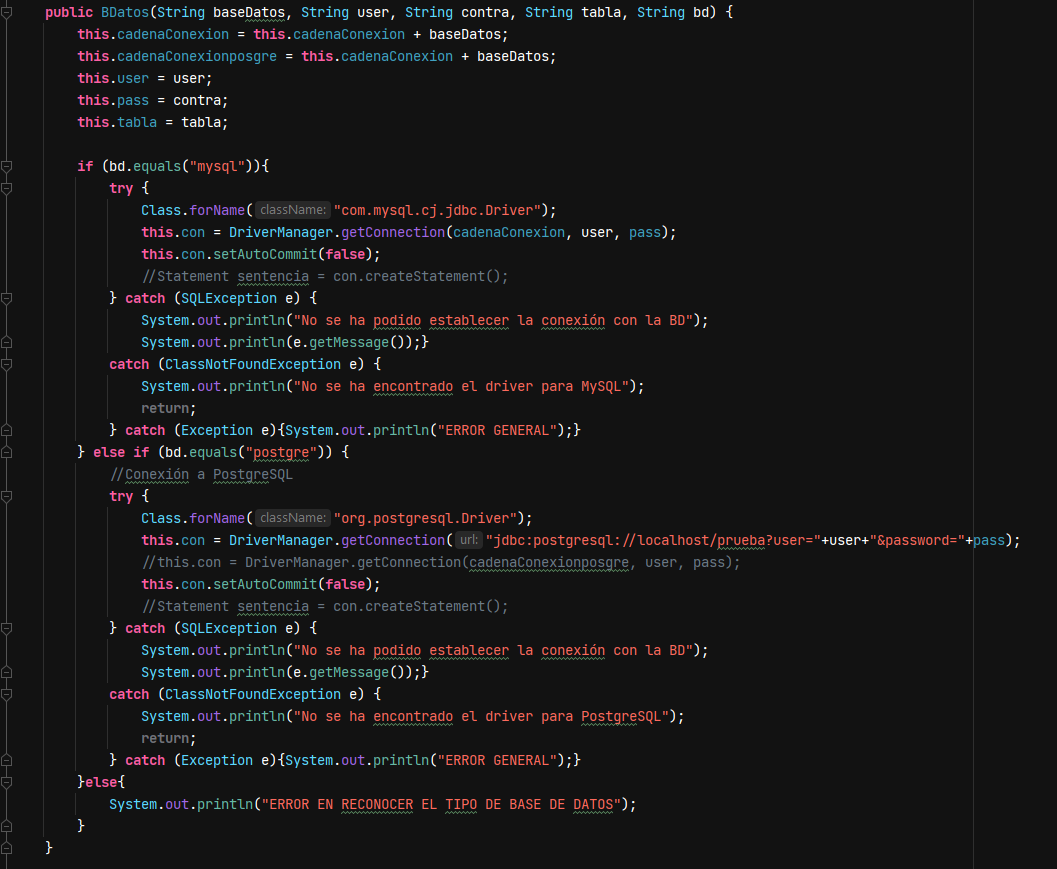


Si eliges la opción de Salir vuelves al menú del principio:



**Explicación de las partes más relevantes del código:**

El constructor de la clase BDatos permite conectar con una base de datos mysql o postgresql y realizar las conexiones para introducirlas como propiedad de una clase. Para ello se utiliza un condicional y en cualquiera de las dos clases utiliza Class.forName para cambiar el drivel de mysql a postgresql y viceversa.





Para sobrescribir lo que ya exista y añadir el nuevo contenido actualizando se realiza este código:



Donde se realizan una serie de comprobaciones y dependiendo de si necesita actualizar o simplemente añadir se añade a la sentencia necesaria y se realiza el commit. A través de varios System.out.println podemos comprobar que los valores repetidos se actualizan.