

DAW/DAM. UD 1. INTRODUCCIÓN A LAS BASES DE DATOS. PRÁCTICAS NO EVALUABLES

DAW/DAM. Bases de datos (BD)

UD 1. INTRODUCCIÓN A LAS BASES DE DATOS

Prácticas no evaluables

Abelardo Martínez y Pau Miñana

Basado y modificado de Sergio Badal (www.sergiobadal.com) y Raquel Torres.

Curso 2023-2024

Aspectos a tener en cuenta

Importante

Estas actividades son opcionales y no evaluables pero es recomendable hacerlas para un mejor aprendizaje de la asignatura.

Vamos a utilizar la aplicación SQLite debido a su sencillez y facilidad de uso. El objetivo es realizar una toma de contacto con las bases de datos de la forma más sencilla e intuitiva posible.

Si buscas las soluciones por Internet o preguntas al oráculo de ChatGPT, te estarás engañando a ti mismo. Ten en cuenta que **ChatGPT no es infalible ni todopoderoso.**

Es una gran herramienta para agilizar el trabajo una vez se domina una materia, pero usarlo como atajo en el momento de adquirir habilidades y conocimientos básicos perjudica gravemente tu aprendizaje. Si lo utilizas para obtener soluciones o asesoramiento respecto a las tuyas, revisa cuidadosamente las soluciones propuestas igualmente. Intenta resolver las actividades utilizando los recursos que hemos visto y la documentación extendida que encontrarás en el “Aula Virtual”.

1. Información sobre sistemas gestores

Actividad 1 (no evaluable)

¿En qué consiste esta actividad?

Crear un debate en el FORO del AULA VIRTUAL sobre los diferentes SGBD que podemos encontrar en el sector productivo. Tienes un ranking de los más usados en <https://db-engines.com/en/ranking>.

1.1. Instrucciones. Pasos a seguir

Accede al aula virtual, ve al FORO DE LA UNIDAD y sigue estos pasos:

- Consulta los posts en el foro de la unidad por si alguien ha publicado antes lo mismo que tú quieres publicar. **No repitas la misma información en un nuevo post.**
- Escoge un *post* publicado por alguno de tus compañeros o el profesor. Si aún no ha publicado nadie, crea tú mismo un nuevo hilo.
- Responde a ese *post* que hayas escogido (el que quieras), **aportando algo nuevo a ese post** y/o dando tu visión personal (si conoces ese SGBD).
- En esa misma respuesta, presenta un SGBD del que nadie haya hablado antes en el foro o añade nuevos detalles de alguno de los SGBD que ya se hayan presentado en dicho foro.
- Aportaciones de máximo 2 párrafos en total, **evitando el “copy&paste” de Internet.**
- Puedes aportar algún vídeo o foto para hacer más amena la lectura.

2. Crear una BD sencilla (guiado)

Actividad 2 (no evaluable)

¿En qué consiste esta actividad?

Como vimos en la teoría, los datos se pueden almacenar en muchos formatos, siendo los ficheros, las hojas de cálculo y las bases de datos los más comunes.

Existen muchas empresas que empiezan usando hojas de calculo como medio de almacenamiento de la información y posteriormente, cuando el volumen de información o la complejidad de los datos crece, necesitan diseñar una base de datos y exportar los datos.

En esta actividad crearemos, de manera guiada, una serie de datos en una **hoja de cálculo** que “copiaremos” a una **base de datos** sencilla creada con SQLite.

2.1. Instrucciones. Pasos a seguir

Seguiremos estos pasos:

- Utilizar una hoja de cálculo para clasificar una serie de datos en columnas.
- Exportar esa hoja de cálculo a un fichero de texto con separador entre columnas (este fichero se llama fichero CSV).
- Importar los datos desde ese fichero CSV a una base de datos de tipo SQLite.
- Mostrar los datos obtenidos en la base de datos.

Los ficheros **CSV** (del inglés **C**omma-**S**eparated **V**alues), se emplean muy a menudo para hacer copias de seguridad de los datos. Más información en https://es.wikipedia.org/wiki/Valores_separados_por_comas

2.2. Software necesario

El software que vamos a usar es multiplataforma y libre.

1) SQLite. <https://www.sqlite.org/>

- Pasos instalación Linux/Windows:
 - <https://codigofacilito.com/articulos/configurando-sqlite>
 - Recuerda agregar la carpeta al PATH del sistema (solo Windows)
- Pasos instalación Mac: viene preinstalado por defecto.

2) LibreOffice Calc. <https://es.libreoffice.org/descubre/calc/>

- Te recomendamos instalar el paquete entero de LibreOffice <https://es.libreoffice.org/descarga/libreoffice/>

Como hemos visto en el temario, **SQLite es una base de datos portable**; es decir, que te la puedes llevar donde quieras simplemente copiando el archivo/fichero físico donde se aloja. Te recomendamos que crees una **carpeta de trabajo** donde almacenarás todos los datos, por ejemplo: BD.UD1

2.3. Crear la hoja de cálculo

Si pensamos en un Centro de Estudios que tiene información de profesores, estudiantes, notas, etc., estaríamos hablando de una “futura” base de datos (la del centro de estudios) que contiene varias tablas (profesores, estudiantes, notas, etc.). En este caso, con esta hoja de cálculo, estamos introduciendo los datos de la “futura tabla” de **estudiantes**.

Crearemos los datos por columnas según se muestran con LibreOffice Calc que serían los datos de los estudiantes. Para introducir los datos usaremos un método sencillo (puesto que es un ejemplo docente): basta con rellenar la primera fila de datos y luego utilizar la función autocompletar de Calc; es decir, nos situamos sobre la esquina inferior derecha de una celda y, cuando aparezca una flecha negra, con el ratón arrastramos hacia abajo. En total crearemos 30 alumnos (30 filas).

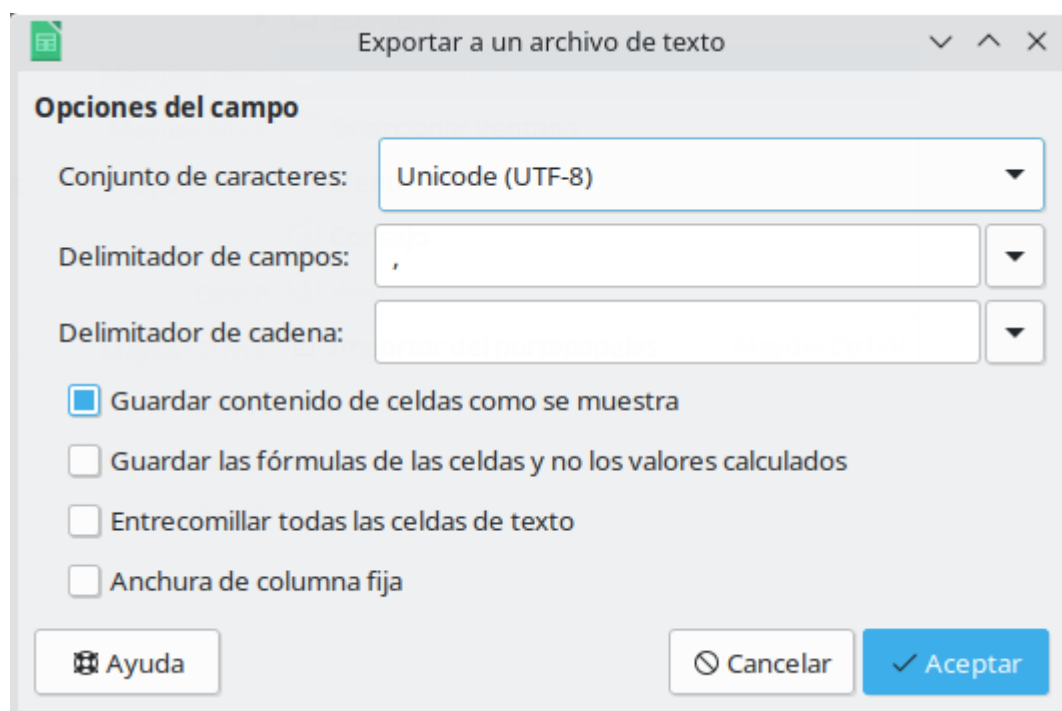
¿Por qué no rellenamos la fila 1 con los títulos de las columnas NIE, nombre, edad y grupo? Porque SQLite importa los datos sin tener en cuenta la fila de títulos y lo crearía como una fila de datos más.

Por ejemplo:

	A	B	C	D
1	1	Alumno1	1	Grupo1
2	2	Alumno2	2	Grupo2
3	3	Alumno3	3	Grupo3
4	4	Alumno4	4	Grupo4
5	5	Alumno5	5	Grupo5
6	6	Alumno6	6	Grupo6
7	7	Alumno7	7	Grupo7
8	8	Alumno8	8	Grupo8
9	9	Alumno9	9	Grupo9
10	10	Alumno10	10	Grupo10
11	11	Alumno11	11	Grupo11
12	12	Alumno12	12	Grupo12
13	13	Alumno13	13	Grupo13
14	14	Alumno14	14	Grupo14
15	15	Alumno15	15	Grupo15
16	16	Alumno16	16	Grupo16
17	17	Alumno17	17	Grupo17
18	18	Alumno18	18	Grupo18
19	19	Alumno19	19	Grupo19
20	20	Alumno20	20	Grupo20
21	21	Alumno21	21	Grupo21
22	22	Alumno22	22	Grupo22
23	23	Alumno23	23	Grupo23
24	24	Alumno24	24	Grupo24
25	25	Alumno25	25	Grupo25
26	26	Alumno26	26	Grupo26
27	27	Alumno27	27	Grupo27
28	28	Alumno28	28	Grupo28
29	29	Alumno29	29	Grupo29
30	30	Alumno30	30	Grupo30

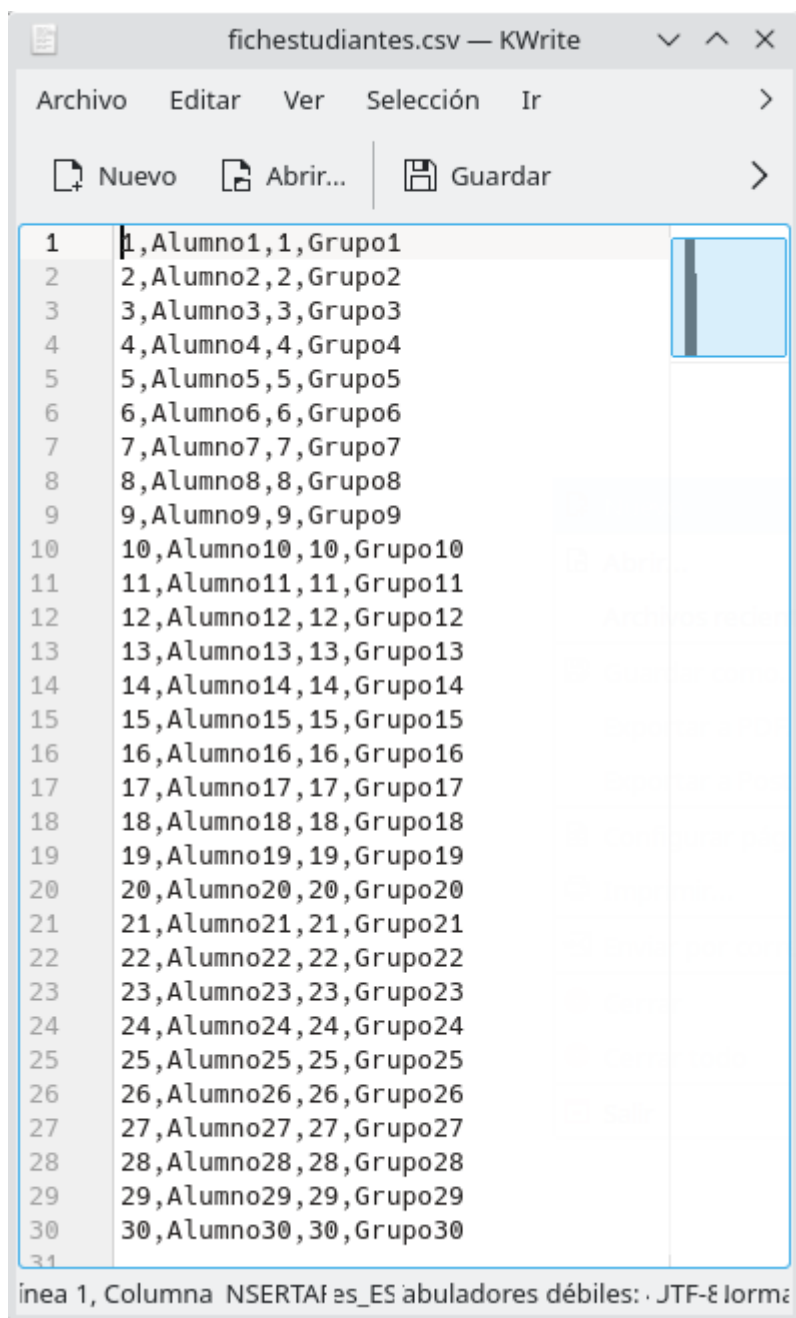
2.4. Exportar datos a CSV

En LibreOffice Calc iremos al menú **Archivo** → **Guardar Como** → **Texto CSV** y le pondremos el nombre de "**fichestudiantes.csv**" indicando que el separador es la coma como indica la captura siguiente:



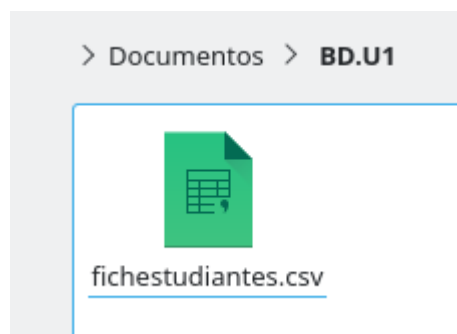
Recuerda grabar ese fichero en tu **CARPETA DE TRABAJO (BD.UD1)**. Si cometes algún error y no puedes seguir, simplemente vacía esa carpeta y empieza de nuevo.

Comprobaremos que el **contenido del fichero CSV** obtenido es el esperado con el editor de textos por defecto (por ejemplo Pluma, Gedit, KWrite o similar en Linux, o Bloc de notas en Windows).



2.5. Crear la base de datos

1) Ve a la carpeta de trabajo (BD.UD1) y comprueba que está dentro de dicha carpeta el fichero CSV generado anteriormente. Por ejemplo, en Linux (entorno gráfico y consola):



```
total 4
-rw-r--r-- 1 user user 653 sep 15 2021 fichestudiantes.csv
```

2) Crea la base de datos donde se alojarán las tablas que vamos a construir. Entramos a SQLite desde la terminal o línea de comandos.

Hay dos tipos de órdenes en **SQLite**:

Comandos: **SÍ** comienzan por punto y **NO** acaban en punto y coma.
Órdenes propias de **SQLite** para su manejo y configuración.

Sentencias: **NO** comienzan por punto y **SÍ** acaban en punto y coma.
Órdenes en **lenguaje SQL** para la **base de datos**.

Para **crear una BD**, el **comando** es muy intuitivo:

```
sqlite3 nombre.db
```

Si solo tecleamos **sqlite3** sin especificar el nombre de la base de datos, luego tendremos **problemas** porque se abre en una bd temporal (*transient in-memory database*) y la información se pierde al cerrar. Para evitarlo, una vez dentro de **SQLite** usa el **COMANDO** **.open nombre.db** (punto open ...), que crea o abre una base de datos (si nombre.db ya existía).

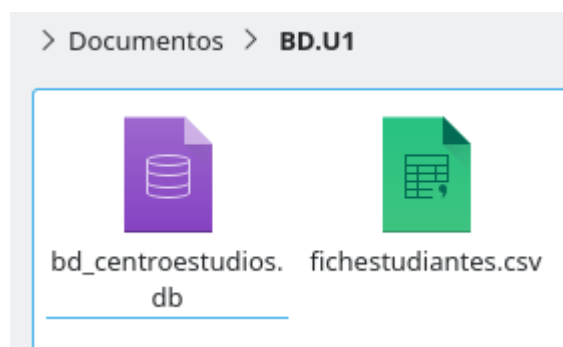
Ejecuta el comando siguiente:

```
sqlite3 bd_centroestudios.db
```

O bien **sqlite3** seguido de **.open bd_centroestudios.db**, para entrar y crear esa base de datos.

Puedes salir en cualquier momento con el COMANDO **.exit** (punto exit) o **.quit** (punto quit) y volver a entrar y borrar el fichero .db para volver a empezar si te equivocas.

Ahora tenemos el fichero CSV y el fichero donde se almacena la nueva BD que acabamos de crear:



2.6. Crear la tabla donde guardar los datos

El **nombre de los campos** puede estar formado por caracteres alfanuméricos (letras y/o números). Aunque hoy en día los SGBD permiten cualquier carácter, ES MUY ACONSEJABLE (por no decir casi obligatorio) seguir las siguientes recomendaciones al colocar nombres a los campos:

- Poner nombres significativos (apellido, nombre, telf, etc.) para que el usuario pueda deducir los datos almacenados en el campo.
- NO incluir espacios en blanco dentro de los nombres de campo. Si se desea que un nombre de campo esté compuesto por más de una palabra, se puede unir con el carácter de subrayado (_). Ejemplo: notas_mensuales.
- NO utilizar caracteres especiales como acentos, \$, &, @, #, %, Ç, etc.

Aunque **no hay ninguna obligación** de usar prefijos ni sufijos y **NO se suelen usar en entornos reales**, para hacer más claro este ejercicio usaremos los siguientes prefijos:

- > **bd_** para las bases de datos
- > **t_** para las tablas
- > **c_** para los campos

Una vez creada la BD "bd_centroestudios.db" crearemos la tabla "**t_estudiantes**" con los campos **c_nie**, **c_nombre**, **c_edad** y **c_grupo**, poniendo la siguiente SENTENCIA (haz **copiar y pegar en la terminal**):

```
create table t_estudiantes (  
  c_nie text,  
  c_nombre text,  
  c_edad integer,  
  c_grupo text);
```

Como habrás deducido, junto a cada campo indicamos el tipo de datos que contendrá ese campo. Por ejemplo, el campo **c_edad** es de tipo **entero (integer)**.

Cada uno de los campos se corresponderá con cada una de las columnas separadas por coma del fichero CSV (fichestudiantes.csv) creado anteriormente.

El **COMANDO** **.tables** (punto tables) nos dice que tenemos la tabla "t_estudiantes" creada dentro de otro contenedor llamado "bd_centroestudios".

```
~/Documentos/BD.U1$ sqlite3 bd_centroestudios.db
SQLite version 3.31.1 2020-01-27 19:55:54
Enter ".help" for usage hints.
sqlite> .tables
t_estudiantes
sqlite>
```

Puedes crear otras tablas y ver cómo aparecen listadas con el **COMANDO** **.tables** y borrarlas después con la **SENTENCIA** **drop table nombredetabla**. Prueba a **crear las tablas (vacías)** de profesores y notas con **create table**. Comprueba que existen con **.tables** y luego bórralas con **drop table**.

```
sqlite> create table t_estudiantes (
...>   c_nie text,
...>   c_nombre text,
...>   c_edad integer,
...>   c_grupo text);
sqlite>
sqlite> create table t_profesores (
...>   c_nombre text,
...>   c_nif text);
sqlite>
sqlite> create table t_notas (
...>   c_nifalumno text,
...>   c_nota double);
sqlite> drop table t_profesores;
sqlite> drop table t_notas;
sqlite> .tables
t_estudiantes
sqlite>
```

Piensa que todo son “contenedores de información”, de manera que una **BD** contiene **tablas**, una tabla contiene **campos** y un campo contiene **datos**.

En este caso, hemos creado varios contenedores vacíos (**sin datos**) de información (t_estudiantes, t_profesores y t_notas), que luego hemos borrado, para que entiendas qué función tiene cada contenedor.

Todas las tablas (t_estudiantes, t_profesores, t_notas) están dentro de otro contenedor llamado bd_centroestudios.db.

Vamos ahora a introducir el último nivel de información: **LOS DATOS**. En jerga informática

se denomina POBLAR LAS TABLAS CON DATOS.

2.7. Importar datos

Vamos a “poblar/rellenar” la tabla **"t_estudiantes"** con los datos que teníamos en la hoja de cálculo, en el fichero **"fichestudiantes.csv"**. Seguimos estos pasos:

1. Antes de realizar la importación, le indicaremos a SQLite que el elemento que usaremos para separar los datos dentro del CSV es la “coma” con el **COMANDO** **.separator** (punto separator).
2. Importaremos el fichero **"fichestudiantes.csv"** sobre **"t_estudiantes"** con el **COMANDO** **.import** (punto import), **indicando el fichero y la tabla**. De esta manera, la tabla pasa de ser un mero contenedor vacío a alojar datos, clasificados en campos.

```
sqlite> .separator ','
sqlite> .import fichestudiantes.csv t_estudiantes
```

1. Mostraremos los datos importados a la tabla **"t_estudiantes"** con la **SENTENCIA** **select** a la que dedicaremos varias semanas más adelante. Si queremos mostrar el nombre de los campos (o atributos) al mostrar las tablas tecleamos previamente **.headers on** para activar las cabeceras y para mostrar los datos alineados en columnas tecleamos **.mode column**. Después, mostramos los datos con la sentencia **select**.

```
sqlite> .headers on
sqlite> .mode column
sqlite> select * from t_estudiantes;
```

c_nie	c_nombre	c_edad	c_grupo
1	Alumno1	1	Grupo1
2	Alumno2	2	Grupo2
3	Alumno3	3	Grupo3
4	Alumno4	4	Grupo4
5	Alumno5	5	Grupo5
6	Alumno6	6	Grupo6
7	Alumno7	7	Grupo7
8	Alumno8	8	Grupo8
9	Alumno9	9	Grupo9
10	Alumno10	10	Grupo10

3. Crea tu propia base de datos

Actividad 3 (no evaluable)

¿En qué consiste esta actividad?

Te proponemos varios nombres de bases de datos, tablas, campos y tipos de datos y tendrás que crear tu propia base de datos, rellenando una de las tablas mediante una hoja de cálculo como hemos visto en la actividad guiada.

Luego, tendrás que colgar en el foro del módulo una captura de la base de datos que has creado, con los datos de una de las tablas como te mostramos.

Debes crear las tablas y campos (con los tipos adecuados) que guarden relación con la base de datos elegida. Éstas son las propuestas:

Base de datos	Tablas	Campos	Tipos de datos
bd_hospital	t_pacientes	c_nombre	text
bd_biblioteca	t_libros	c_descripcion	integer
bd_restaurante	t_mesas	c_edad	date
bd_centroestudios	t_vuelos	c_origen	
bd_aeropuerto	t_pilotos	c_destino	
	t_prestamo	c_cantidad	
	t_menus	c_fecha	
	t_docentes	c_apellidos	
	t_notas	c_nif	
	t_camras	c_numero	

3.1. ¿Qué tengo que entregar?

Una posible solución, que tendrías que entregar en el foro, sería ésta:

- Base de datos: bd_aeropuerto
- Tabla: t_vuelos. Campos: c_origen, c_destino
- Tabla: t_pilotos. Campos: c_nombre, c_apellidos

```
~/Documentos/BD.U1$ sqlite3 bd_aeropuerto.db
SQLite version 3.31.1 2020-01-27 19:55:54
Enter ".help" for usage hints.
sqlite> create table t_vuelos (
...>   c_origen text,
...>   c_destino text);
sqlite>
sqlite> create table t_pilotos (
...>   c_nombre text,
...>   c_apellidos text);
sqlite> .tables
t_pilotos  t_vuelos
sqlite>
```

Importación de datos a la tabla t_pilotos y listado de los mismos:

```
sqlite> .separator ','
sqlite> .import fichpilotos.csv t_pilotos
sqlite> .headers on
sqlite> .mode column
sqlite> select * from t_pilotos;
c_nombre      c_apellidos
-----
Pedro         García Pérez
Lucía         López Gracia
Sergio        Bielsa Biels
sqlite>
```

Puedes salir en cualquier momento con el **COMANDO** **.exit** (punto exit) o **.quit** (punto quit) y volver a entrar y borrar el fichero .db para volver a empezar si te equivocas.

Hay dos tipos de órdenes en SQLite:

Comandos: SÍ comienzan por punto y NO acaban en punto y coma.

```
.import ...
```

```
.tables
```

Sentencias: NO comienzan por punto y SÍ acaban en punto y coma.

```
select ... ;
```

```
create table ... ;
```

```
drop table ... ;
```

4. Bibliografía

- Comandos principales de SQLite: <https://tutoriales.edu.lat/pub/sqlite/sqlite-commands/sqlite-comandos>
- Importar datos desde un fichero CSV a una tabla de SQLite: <https://donnierock.com/2013/07/17/importar-datos-desde-un-fichero-csv-a-una-tabla-de-sqlite/>
- Primeros pasos con SQLite: <https://www.imaginanet.com/blog/primeros-pasos-con-sqlite3-comandos-basicos.html>
- Export SQLite Database To a CSV File: <https://www.sqlitetutorial.net/sqlite-tutorial/sqlite-export-csv/>
- Getting started with SQLite on Windows: <https://www.youtube.com/watch?v=VZ20Lh4zbRo>
- Datasource.ai. ¿Qué es SQLite Y Cómo Instalarlo? <https://www.datasource.ai/es/data-science-articles/que-es-sqlite-y-como-instalarlo>
- DB Browser for SQLite. <https://sqlitebrowser.org/>



Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](https://creativecommons.org/licenses/by-sa/4.0/)