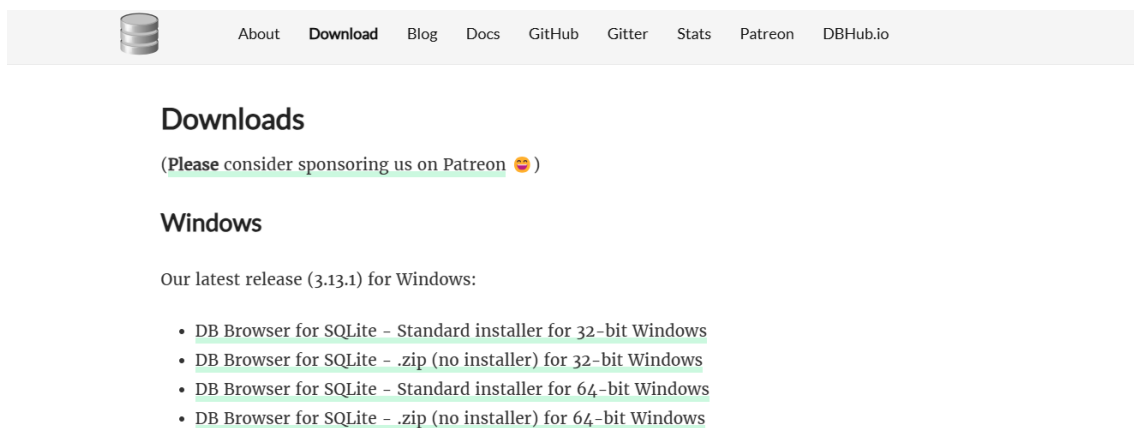


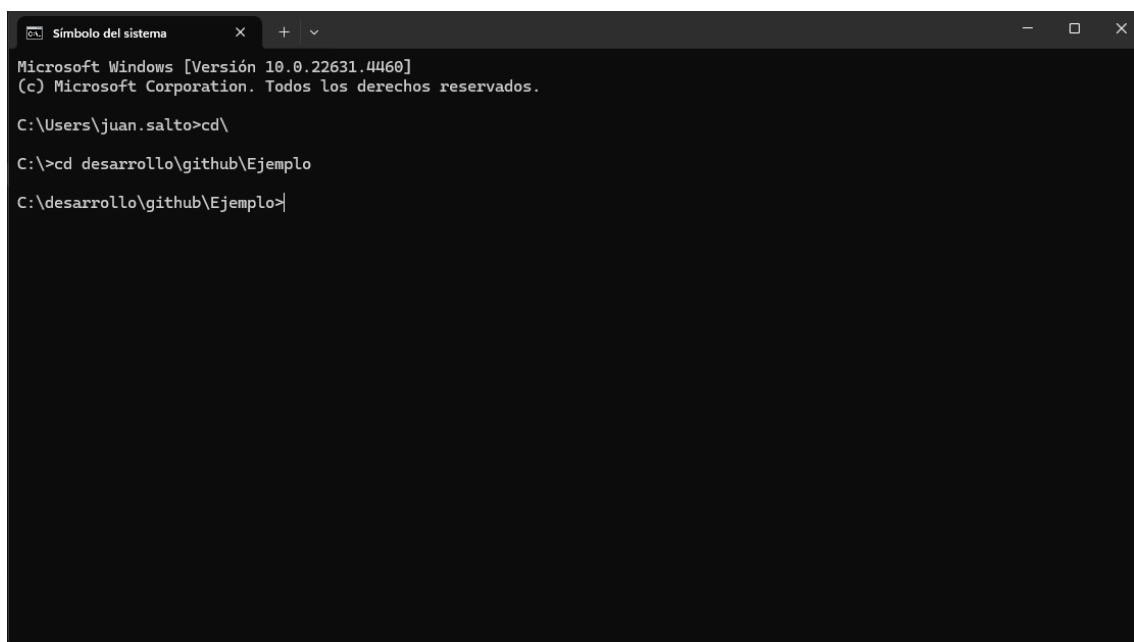
# INSTALACION DE BD (base de datos)

## 1.Descargar software

Lo primero que debemos hacer es ingresar al [sitio oficial](#) donde descargaremos la versión compatible con nuestro sistema operativo. Las opciones de instalación son para poder administrar las bases con una interfaz, funciona bien con las configuraciones de instalación por defecto



## 2. Una vez instalado vamos a ir en un cmd/bash o cli a nuestro proyecto



## 3. Ejecutamos `npm install express sqlite3`

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22631.4460]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\juan.salto>cd\

C:\>cd desarrollo\github\Ejemplo

C:\desarrollo\github\Ejemplo>npm install express sqlite3
npm warn deprecated are-we-there-yet@3.0.1: This package is no longer supported.
npm warn deprecated @npmcli/move-file@1.1.2: This functionality has been moved to @npmcli/fs
npm warn deprecated gauge@4.0.4: This package is no longer supported.
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache i
f you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerfu
l.
npm warn deprecated npmlog@6.0.2: This package is no longer supported.
npm warn deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported

added 126 packages, and audited 192 packages in 5s

24 packages are looking for funding
  run `npm fund` for details

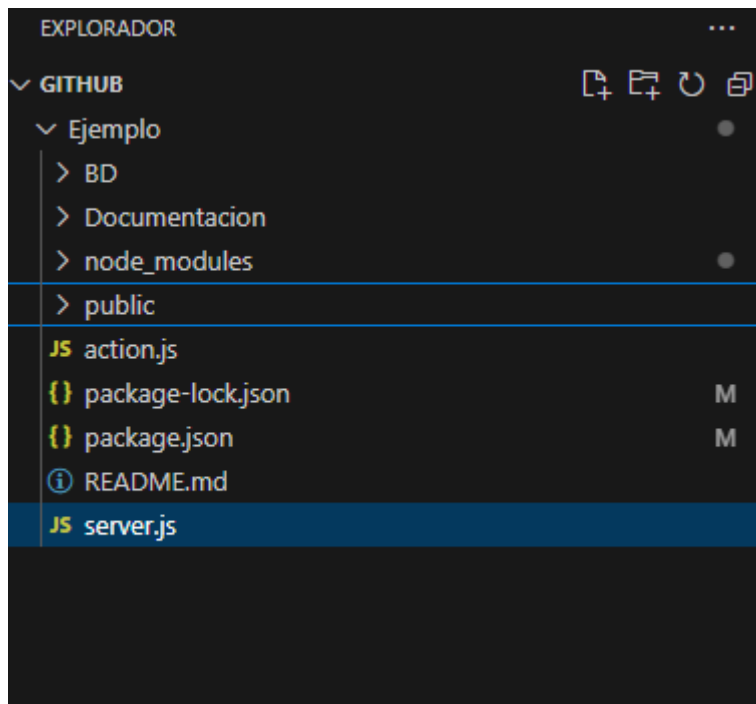
found 0 vulnerabilities

C:\desarrollo\github\Ejemplo>
```

Como podemos ver aparecen estos warning (no es nada grave), es una base ligera cuyas dependencias de express no cuentan con soporte pero funcionan correctamente.

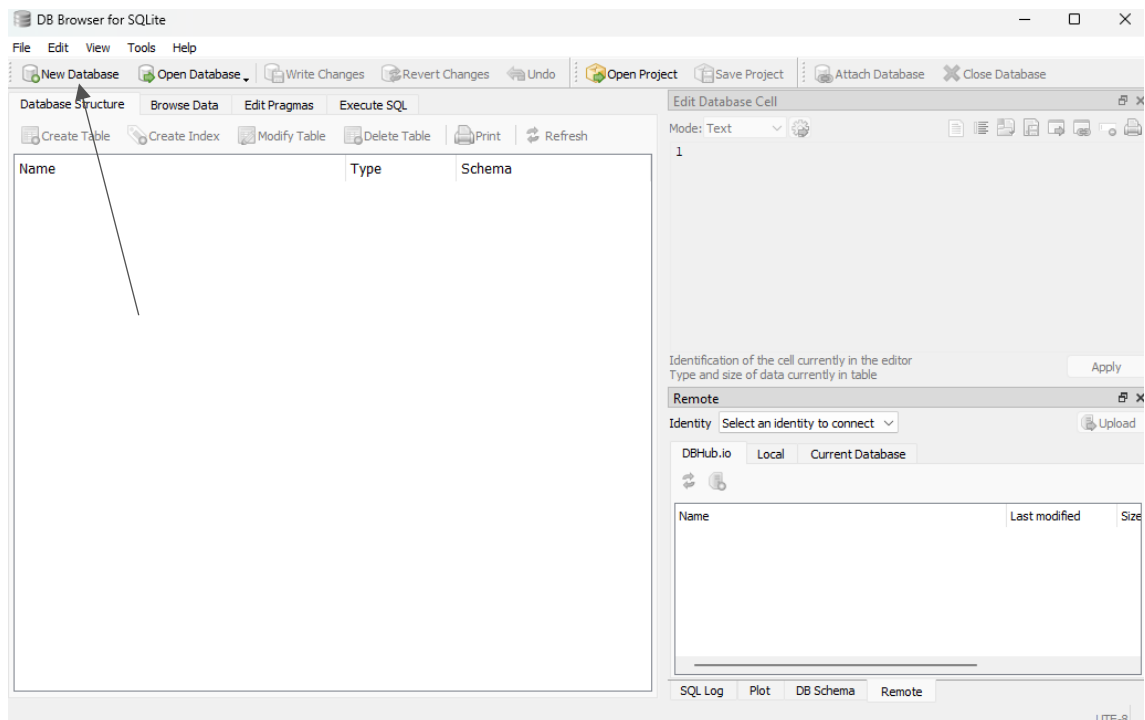
Una vez realizado estos dos pasos debemos configurar la forma en la que nuestro programa interactúa con la base de datos y una base de datos en si misma. Se adjunta pdf para la sintaxis (solo sintaxis, el pdf recomienda usar Access que no se usa mas como BD, también da algunos ejemplos de su sintaxis), luego se crea una tabla que se va a usar en un esquema como referencia. La misma debe contener los datos del formulario que crearon en HTML, como primera instancia.

Para conectar mi pagina con sql, el servidor y la BD se hace de la siguiente manera:

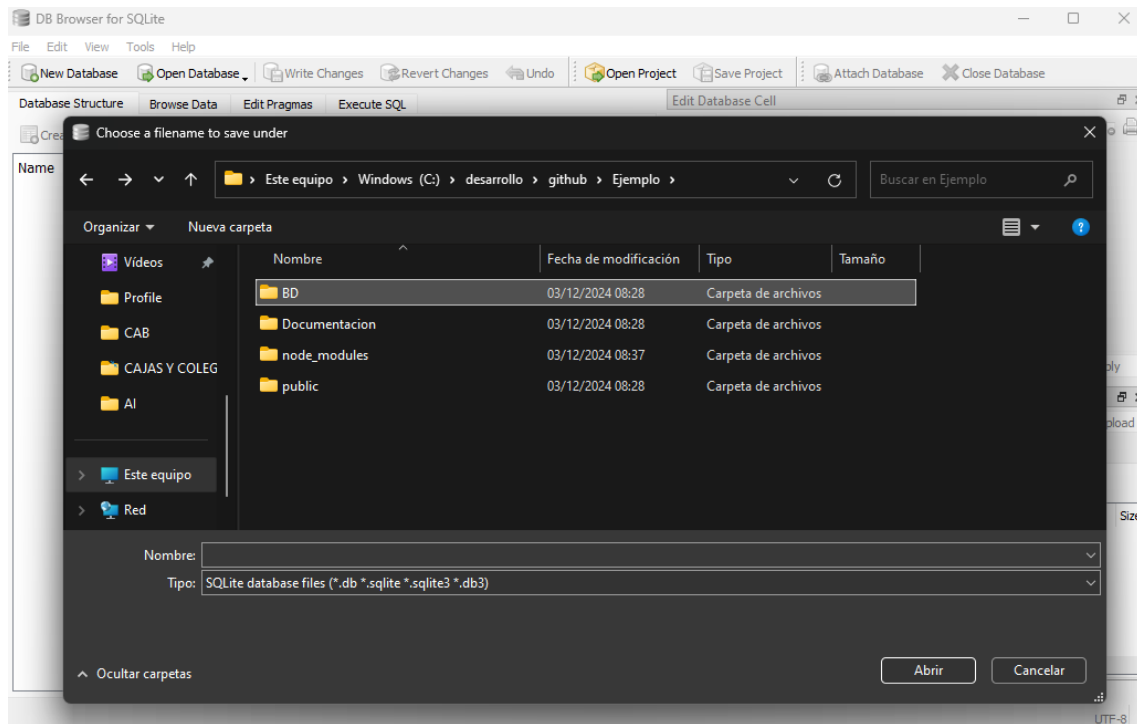


Primero creo un action.js que es una clase que me va a servir para hacer las acciones necesarias de las peticiones.

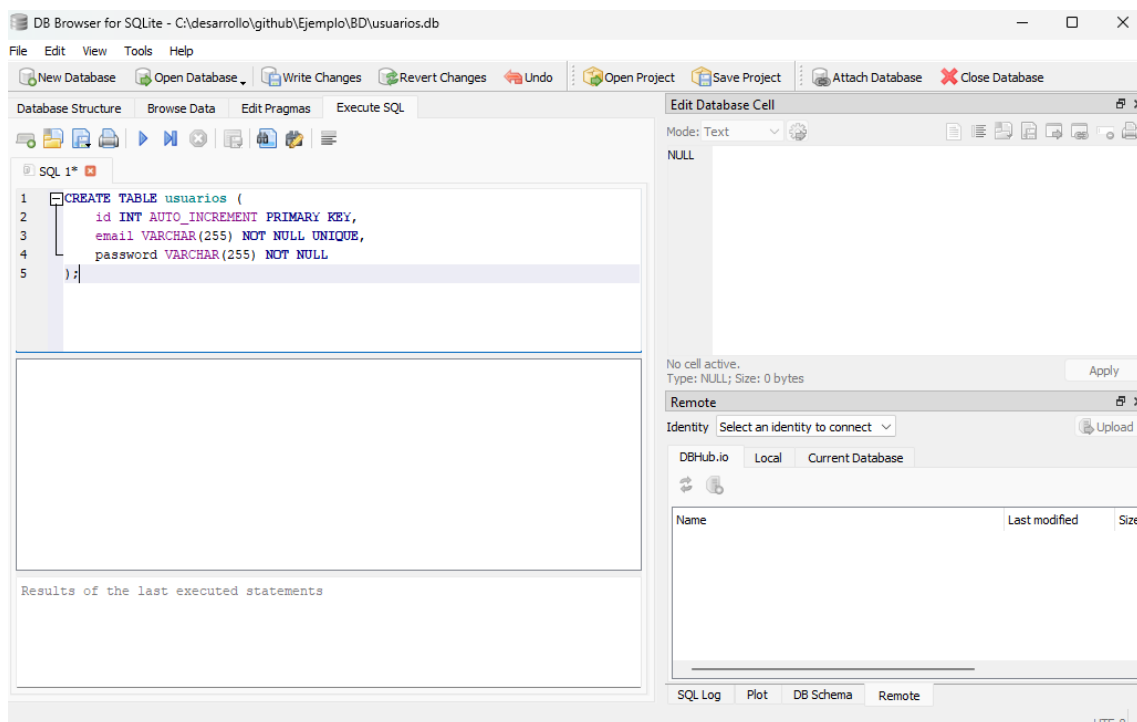
Luego creo mi base de datos, para ellos abro el programa instalado. En New Database



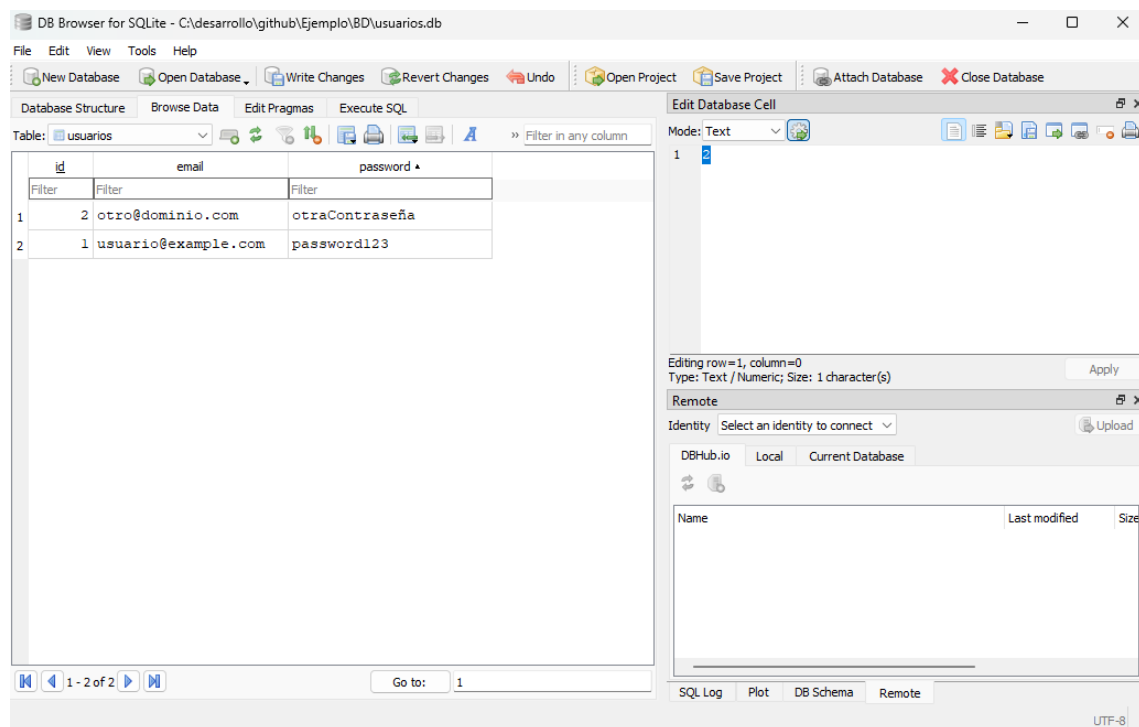
Luego creamos la carpeta BD en nuestro repositorio y dentro guardamos la base de datos con un nombre acorde, un nombre corto con palabras separadas por “\_”, mayúsculas o “-”.



Una vez creada la BD de datos hay que crear la tabla (ver material de sintaxis). En mi caso quiero crear la tabla usuarios. Para ello uso el siguiente comando, donde se manifiestan mis dos valores predeterminados en el formulario.



Una vez hecho esto agrego los valores por comando sql o manualmente. Logrando mi tabla de la siguiente manera



Luego creo el código que me va a permitir verificar que las credenciales email/password de mi aplicación se conecten con la base de datos creada previamente. Notese que paso una sentencia con SELECT, WHERE, AND.

```
1 const sqlite3 = require('sqlite3').verbose();
2 const path = require('path');
3
4 // Ruta completa a la base de datos SQLite
5 const dbPath = 'C:\\Desarrollo\\Ejemplo\\BD\\usuarios.db'; // Ruta absoluta a la base de datos SQLite
6
7 function verificarCredenciales(email, password) {
8   return new Promise((resolve, reject) => {
9     // Conectar a la base de datos SQLite
10    const db = new sqlite3.Database(dbPath, (err) => {
11      if (err) {
12        return reject('Error al conectar a la base de datos: ' + err.message);
13      }
14    });
15
16    db.get('SELECT * FROM usuarios WHERE email = ? AND password = ?', [email, password], (err, row) => {
17      if (err) {
18        db.close();
19        return reject('Error al consultar la base de datos: ' + err.message);
20      }
21
22      if (row) {
23        resolve(true);
24      } else {
25        resolve(false);
26      }
27
28      // Cerrar la base de datos después de la consulta
29      db.close();
30    });
31  });
32 }
33
34 module.exports = { verificarCredenciales };
```

