

SISTEMA DE INFORMACION SOBRE EL TDAH

El TDAH no es una limitación, es una forma diferente de percibir y experimentar el mundo.

PROYECTO FINAL

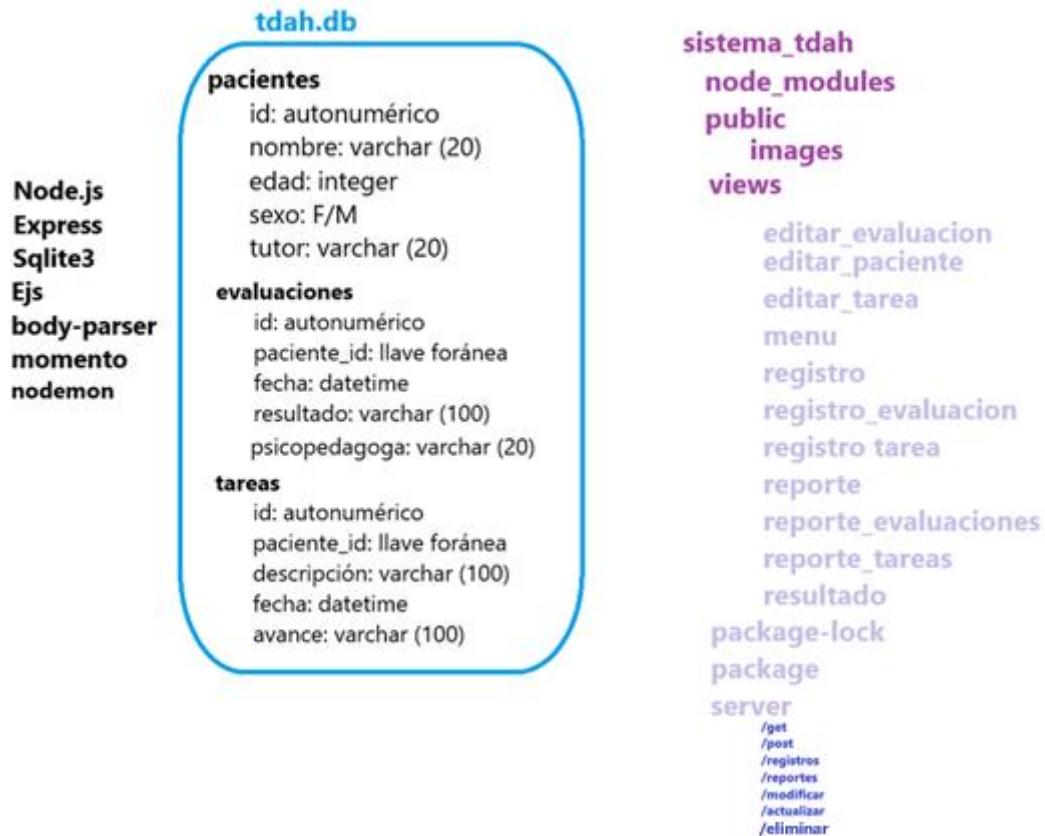


**Autor: Felicidad Quispe Balcazar
Docente: Juan Jose Cervantes**

**CASA DE ESTUDIOS
UNIVERSIDAD AUTONOMA TOMAS FRIAS**

PROYECTO FINAL, SISTEMA DE INFORMACION DE TDAH

1. Estructura de diseño del proyecto.



2. Estructura de la Base de Datos

Se diseñó las siguientes tablas:

- **Tabla Pacientes**

- id: Identificador único del paciente (autogenerado).
- nombre: Nombre del paciente.
- edad: Edad del paciente.
- sexo: Sexo del paciente.
- Tutor: persona encargada del niño ya sea mamá, papá u otro pariente.

- **Tabla Evaluaciones**

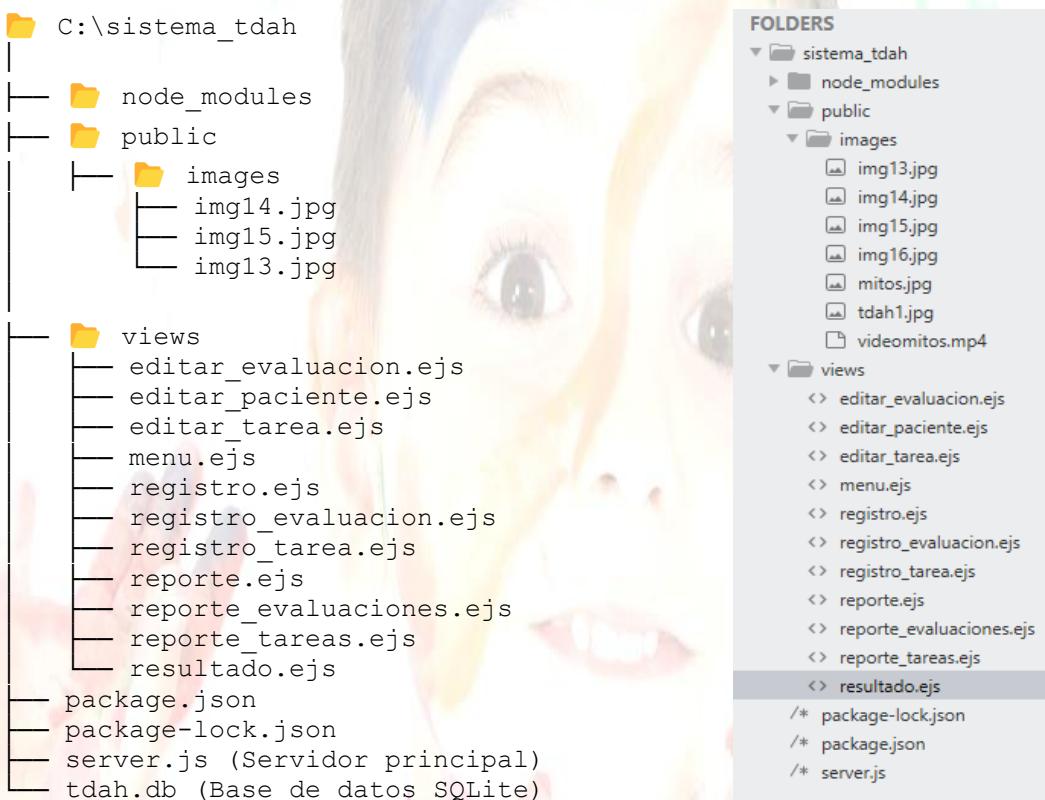
- id: Identificador único de la evaluación (autogenerado).
- paciente_id: Relación con la tabla pacientes (llave foránea).
- fecha: Fecha de la evaluación.

- resultado: Resultado de la evaluación.
- Psicopedagoga: doctora encargada de evaluar al niño

- **Tabla Tareas**

- id: Identificador único de la tarea (autogenerado).
- paciente_id: Relación con la tabla pacientes (llave foránea).
- descripción: Descripción de la tarea.
- fecha: Fecha de asignación de la tarea.
- Avance: tareas que se le da al niño.

3. Estructura Final del Proyecto.



Para poder acceder a las imágenes se creo una carpeta con el nombre public dentro de la carpeta incluir las imágenes en otra carpeta como se muestra en la estructura llamada “images”, en el servidor configurar Express para que public sea accesible, de la siguiente manera:

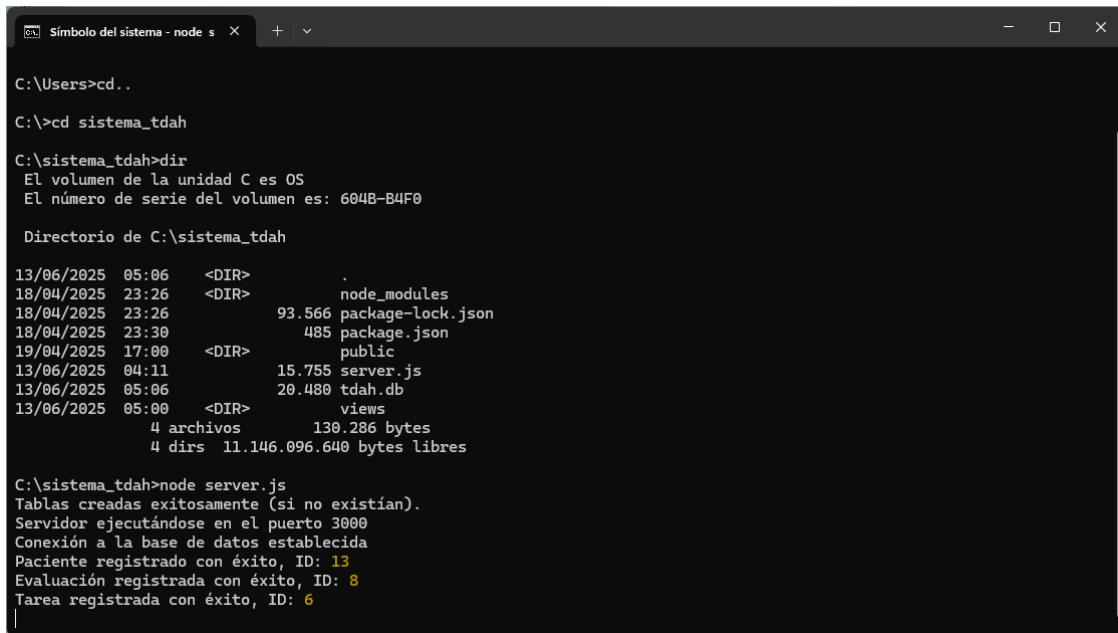
```
app.use(express.static('public'));
```

4. Instalación Necesaria.

```
C:\sistema_tdah>npm list
sistema_tdah@1.0.0 C:\sistema_tdah
```

```
+-- body-parser@2.2.0
+-- ejs@3.1.10
+-- express@5.1.0
+-- moment@2.30.1
+-- nodemon@3.1.9
`-- sqlite3@5.1.7
```

5. Ejecución del Sistema TDAH.



The screenshot shows a terminal window titled "Símbolo del sistema - node s". The command entered is "node server.js". The output shows the directory structure of the project, including files like package-lock.json, package.json, and server.js, and a database file tdaah.db. It also displays the creation of tables in the database and the start of the server on port 3000. Success messages indicate patient, evaluation, and task registrations.

```
C:\Users>cd..
C:\>cd sistema_tdah
C:\sistema_tdah>dir
El volumen de la unidad C es OS
El n m ero de serie del volumen es: 604B-B4F0

Directorio de C:\sistema_tdah

13/06/2025  05:06    <DIR>          .
18/04/2025  23:26    <DIR>          node_modules
18/04/2025  23:26                93.566 package-lock.json
18/04/2025  23:30                485 package.json
19/04/2025  17:00    <DIR>          public
13/06/2025  04:11                15.755 server.js
13/06/2025  05:06                20.480 tdaah.db
13/06/2025  05:00    <DIR>          views
        4 archivos           130.286 bytes
        4 dirs   11.146.096.640 bytes libres

C:\sistema_tdah>node server.js
Tablas creadas exitosamente (si no existian).
Servidor ejecut ndose en el puerto 3000
Conexi n a la base de datos establecida
Paciente registrado con éxito, ID: 13
Evaluaci n registrada con éxito, ID: 8
Tarea registrada con éxito, ID: 6
```

6. Funcionamiento del Sistema TDAH.

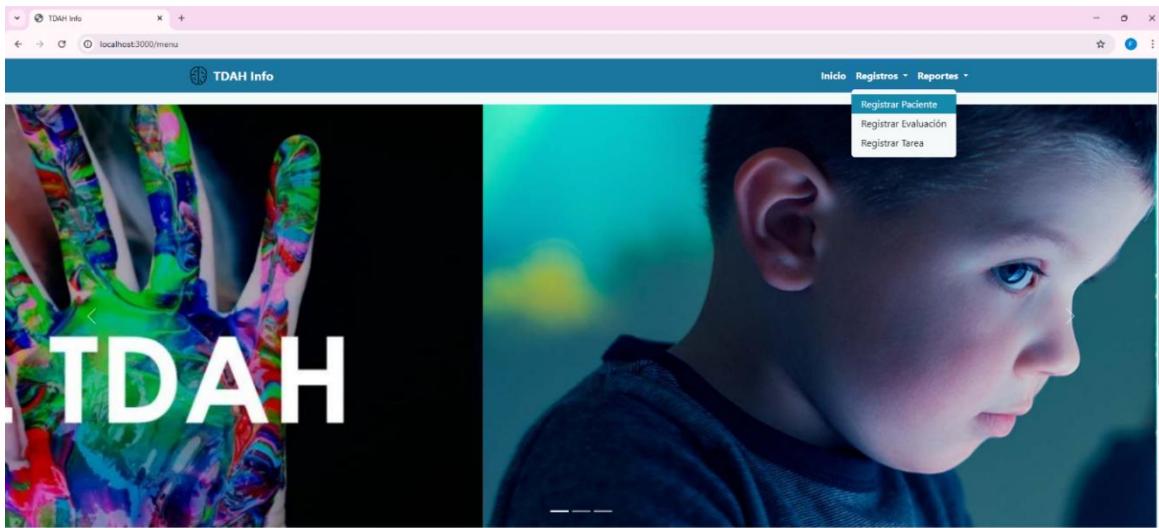
a. P gina Principal del Sistema TDAH es menu.ejs.

Para poder ingresar al sistema TDAH, se debe escribir en el navegador localhost:3000/menu o se debe ingresar al siguiente link:

<http://localhost:3000/menu>

al ingresar mostrara una pagina principal, donde tiene men , un carrusel de 3 im genes, una peque a informaci n sobre el TDAH, estrategias de apoyo sobre el TDAH y el footer.

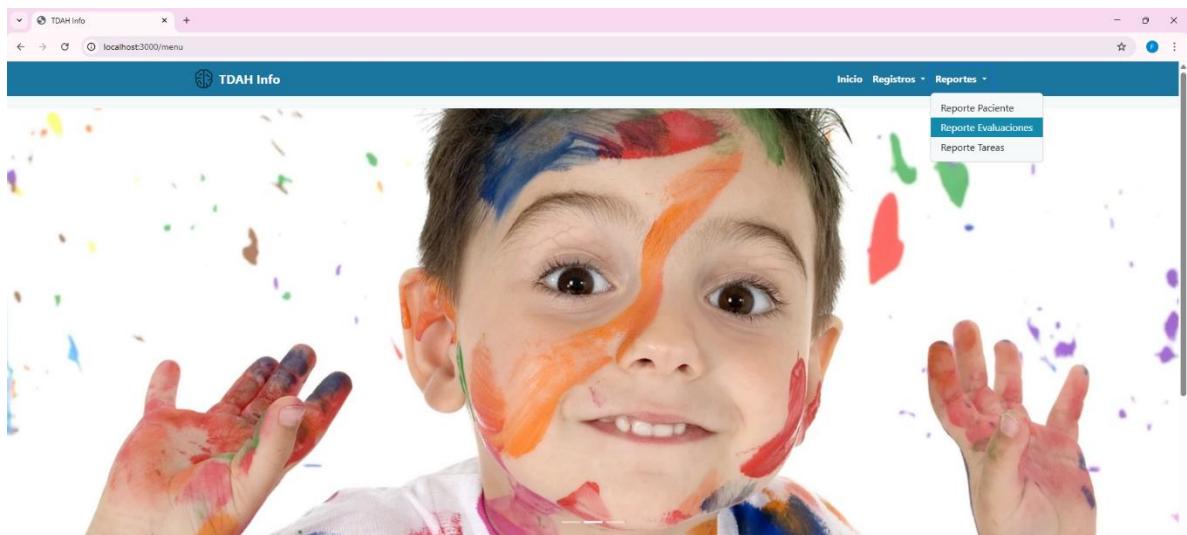
En el men  hay 3 opciones “**Inicio**” donde se redirige a la misma p gina men , en “**Registros**”, hay un sub men  como se muestra en la siguiente imagen:



Una Nueva Perspectiva

El TDAH no es una limitación, es una forma diferente de percibir y experimentar el mundo. Cada persona con TDAH tiene talentos únicos y perspectivas valiosas que enriquecen

La última opción es el menú “**Reportes**”, que también tiene submenú como se muestra en la imagen:



Una Nueva Perspectiva

El TDAH no es una limitación, es una forma diferente de percibir y experimentar el mundo. Cada persona con TDAH tiene talentos únicos y perspectivas valiosas que enriquecen

Se modifico para que sea atractivo al usuario que realizara los cambios en la base de datos, se implementó el botón eliminar, en reportes en los 3 submenús, en todos los botones se pusieron iconos, desde fontawesome, como se muestra en las siguientes imágenes el antes y el después:

Antes.

The screenshot shows a web application window titled "Reporte de Pacientes". The URL is "localhost:3000/reporte". The main content is a table titled "Reporte de Pacientes" with columns: ID, Nombre, Edad, Sexo, Tutor, and Acciones. The data rows are:

| ID | Nombre | Edad | Sexo | Tutor | Acciones |
|----|--------------------|------|-----------|------------------|----------------------------|
| 1 | Mathias Rodrigues | 7 | Masculino | Edwin Rodrigues | <button>Modificar</button> |
| 2 | Brian Vargas | 15 | Masculino | Alberto Vargas | <button>Modificar</button> |
| 3 | Emely Leon | 10 | Femenino | Felicidad Quispe | <button>Modificar</button> |
| 4 | Stephany Chumacero | 12 | Femenino | Felicidad Quispe | <button>Modificar</button> |
| 5 | Laura Treviño | 5 | Femenino | Claudia Lopez | <button>Modificar</button> |
| 6 | Laura Treviño | 5 | Femenino | Claudia Lopez | <button>Modificar</button> |

At the bottom are two buttons: "Registrar otro paciente" and "Inicio".

Después.

The screenshot shows the same web application window after changes. The URL is "localhost:3000/reporte". The table now has more rows, indicating new entries or edits:

| ID | Nombre | Edad | Sexo | Tutor | Acciones |
|----|--------------------|------|-----------|------------------|--|
| 1 | Mathias Rodrigues | 7 | Masculino | Edwin Rodrigues | <button>Modificar</button> <button>Eliminar</button> |
| 2 | Brian Vargas | 15 | Masculino | Alberto Vargas | <button>Modificar</button> <button>Eliminar</button> |
| 3 | Emely Leon | 10 | Femenino | Felicidad Quispe | <button>Modificar</button> <button>Eliminar</button> |
| 4 | Stephany Chumacero | 12 | Femenino | Felicidad Quispe | <button>Modificar</button> <button>Eliminar</button> |
| 5 | Laura rojas | 5 | Femenino | Alex Rojas | <button>Modificar</button> <button>Eliminar</button> |
| 9 | Laura Treviño | 5 | Femenino | Ever Treviño | <button>Modificar</button> <button>Eliminar</button> |
| 10 | Johanna Añez | 12 | Femenino | Emilia Duarte | <button>Modificar</button> <button>Eliminar</button> |

At the bottom are two buttons: "Registrar otro paciente" and "Inicio".

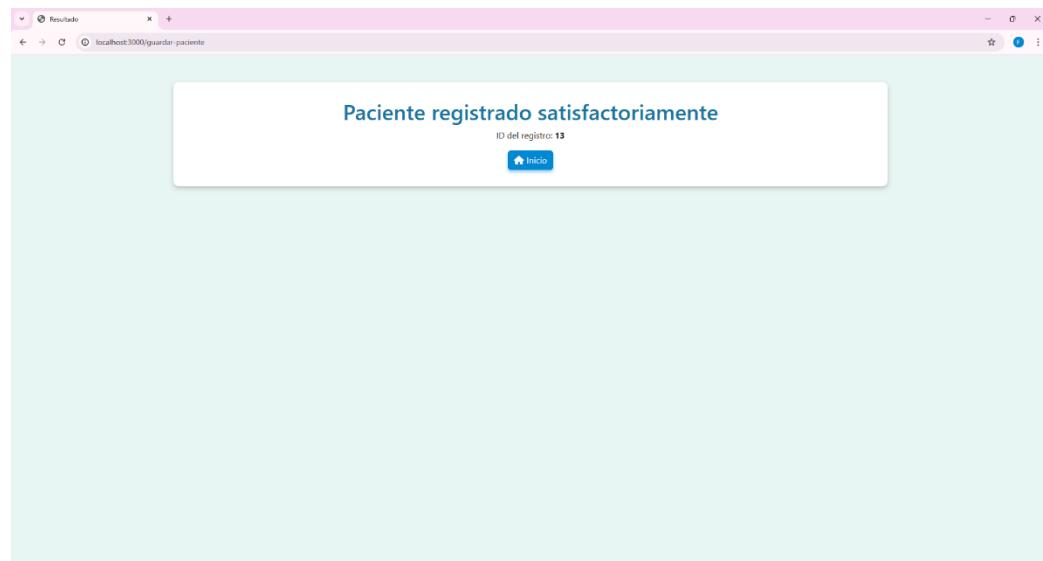
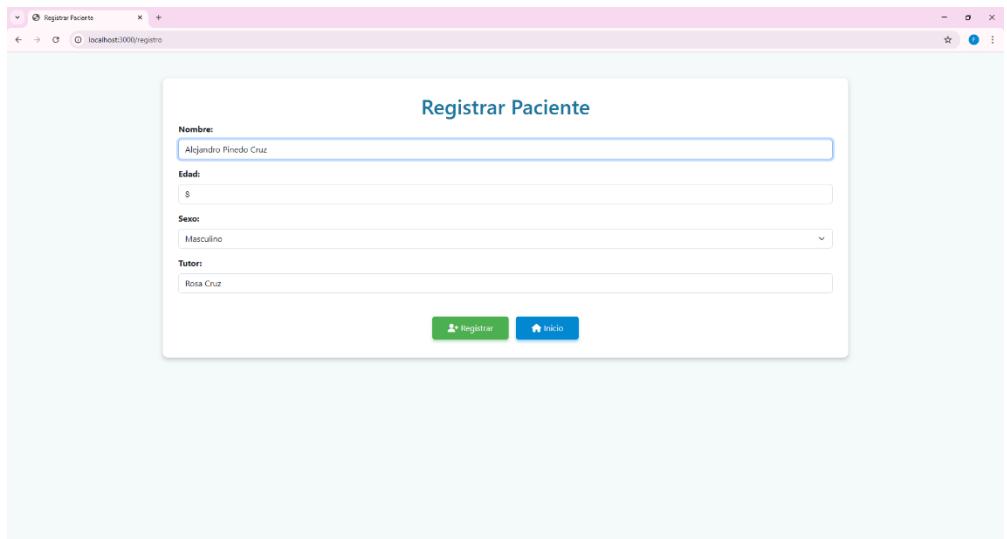
b. Menú Registros y sub menús.

En este menú hay 3 submenús, donde se debe llenar los datos del paciente con TDAH, que se detallara más adelante.

i. Registrar Paciente.

Al elegir esta opción le mostrara el siguiente formulario, donde se debe llenar todos los campos para que se pueda registrar correctamente, hay 2 botones, registrar e inicio, al presionar inicio le manda el formulario menú, una vez llenando los datos se presiona el botón registrar, y le aparece otro formulario (llama al formulario `resultado.ejs`) donde le sale un mensaje “**Paciente registrado**”

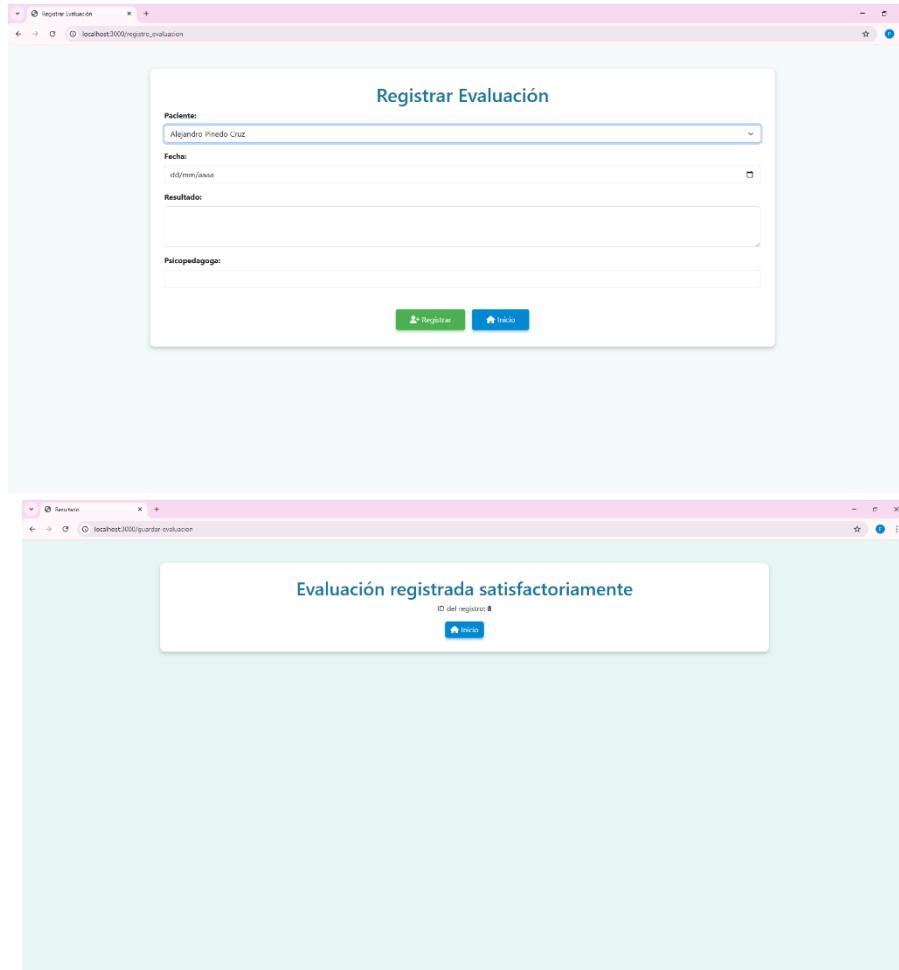
satisfactoriamente”, también le muestra el numero de id con que se registro el paciente, hay un solo botón de inicio, que le mandara al menú ósea a la página principal. Las imágenes muestran lo que se detalló:



ii. Registrar Evaluación.

En este formulario se registra la evaluación del paciente al momento de ingresar se le hace una evaluación, se llana todos los campas para poder presionar el botón “Registrar”, donde le llevara a otro formulario (llama a “resultado.ejs”), donde le sale un mensaje de “Evaluacion registrada satisfactoriamente”, le muestra el id, el botón “Inicio”, le llevara a la página

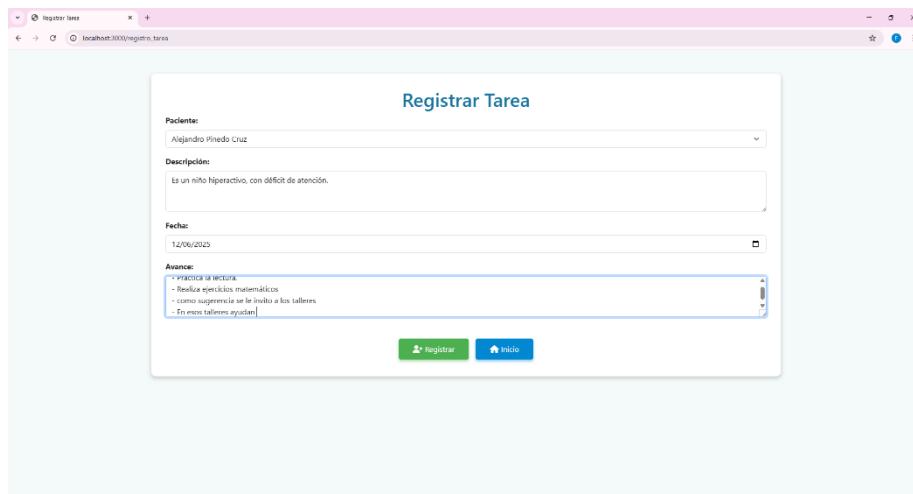
principal que es el menú. Las imágenes muestran lo detallado:



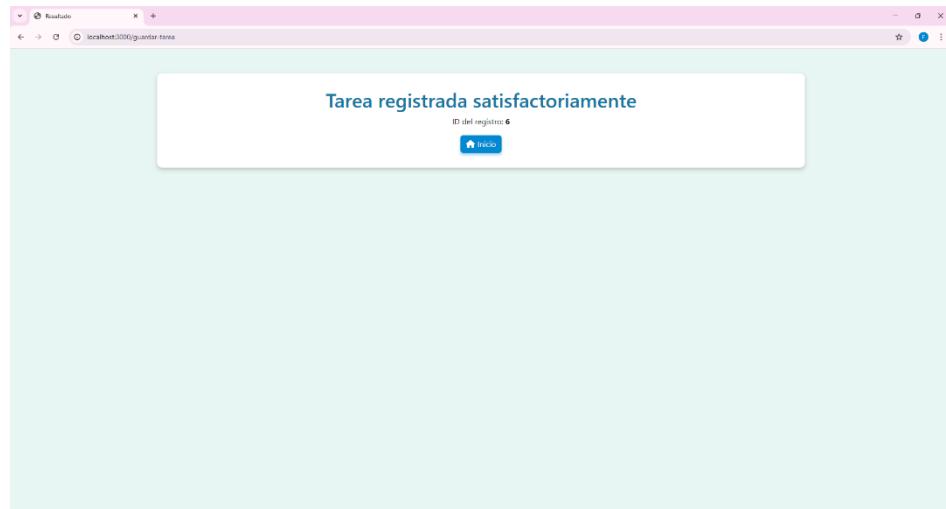
The image contains two screenshots of a web application. The top screenshot shows the 'Registrar Evaluación' (Register Evaluation) page. It has fields for 'Paciente' (Alejandro Pinedo Cruz), 'Fecha' (dd/mm/aaaa), and 'Resultado'. The bottom screenshot shows a success message: 'Evaluación registrada satisfactoriamente' (Evaluation registered successfully) with an ID and a 'Iniciar' button.

iii. Registrar Tarea.

Al igual que las anteriores el llenado es lo mismo como muestra las siguientes imágenes:



The image shows the 'Registrar Tarea' (Register Task) page. It has fields for 'Paciente' (Alejandro Pinedo Cruz), 'Descripción' (Es un niño hiperactivo, con déficit de atención.), 'Fecha' (12/06/2025), and 'Avance' (a list of tasks: practica la lectura, realiza ejercicios matemáticos, como sugerencia se le indica a los talleres, en estos talleres ayudan). The page includes 'Registrar' and 'Inicio' buttons.



c. Menú Reportes y submenú

Aquí se muestra todos los reportes del paciente, “Reporte Paciente”, “Reporte Evaluaciones”, “Reporte Tareas”, en cada reporte tiene 2 acciones (Botones) la de modificar y eliminar datos de la base de datos, también tiene otros dos botones “Registrar otro paciente”, se llama al formulario registrar, e “Inicio”, este va a la página principal, las imágenes de cada caso se mostrarán en los siguientes incisos.

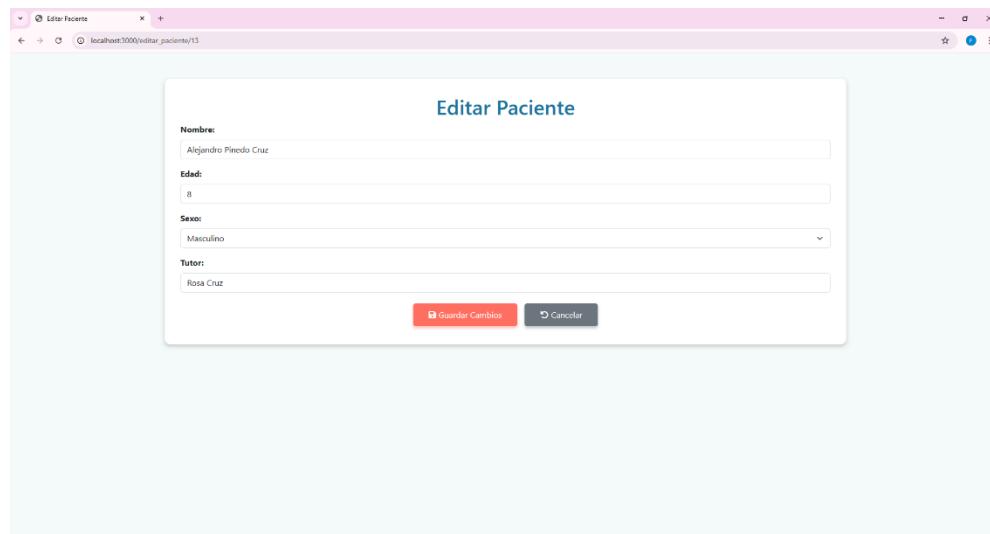
i. Reporte Paciente.

| Reporte de Pacientes | | | | | |
|----------------------|-----------------------|------|-----------|------------------|--|
| ID | Nombre | Edad | Sexo | Tutor | Acciones |
| 1 | Mathias Rodrigues | 7 | Masculino | Edwin Rodrigues | <button>Modificar</button> <button>Eliminar</button> |
| 2 | Brian Vargas | 15 | Masculino | Alberto Vargas | <button>Modificar</button> <button>Eliminar</button> |
| 3 | Emily Leon | 10 | Femenino | Felicidad Quispe | <button>Modificar</button> <button>Eliminar</button> |
| 4 | Stephany Chumacero | 12 | Femenino | Felicidad Quispe | <button>Modificar</button> <button>Eliminar</button> |
| 5 | Laura rojas | 5 | Femenino | Alex Rojas | <button>Modificar</button> <button>Eliminar</button> |
| 9 | Laura Trevíño | 5 | Femenino | Ever Trevíño | <button>Modificar</button> <button>Eliminar</button> |
| 10 | Johanna Añez | 12 | Femenino | Emilia Duarte | <button>Modificar</button> <button>Eliminar</button> |
| 13 | Alejandro Pinedo Cruz | 8 | Masculino | Rosa Cruz | <button>Modificar</button> <button>Eliminar</button> |

[Registrar otro paciente](#) [Inicio](#)

Al presionar “Modificar” se llama al formulario `editar_paciente.ejs`, esta con todos los datos y elige el campo que desee modificar hay 2 botones “Guardar Cambios”, se guardan los cambios y se va al formulario

“Reporte de Pacientes” para verificar si se cambió, “Cancelar”, de igual manera va al formulario “Reporte de Pacientes”.



Al presionar “Eliminar”, llama al formulario “Resultado.ejs”, al implementar el código y poder reutilizar el formulario resultados, se cambió en el código fuente donde hay una variable que se coloca la ruta Eliminar y se declaró una variable para volver al reporte donde se pidió eliminar, hay 2 botones “Confirmar Eliminación” y “Cancelar”, se va a eliminar a Alejandro Pinedo Cruz.

| ID | Nombre | Edad | Sexo | Tutor | Acciones |
|----|-----------------------|------|-----------|------------------|----------|
| 1 | Mathias Rodrigues | 7 | Masculino | Edwin Rodrigues | |
| 2 | Brian Vargas | 15 | Masculino | Alberto Vargas | |
| 3 | Emely Leon | 10 | Femenino | Felicidad Quispe | |
| 4 | Stephany Chumacero | 12 | Femenino | Felicidad Quispe | |
| 5 | Laura rojas | 5 | Femenino | Alex Rojas | |
| 9 | Laura Treviño | 5 | Femenino | Ever Treviño | |
| 10 | Johanna Añez | 12 | Femenino | Emilia Duente | |
| 13 | Alejandro Pinedo Cruz | 8 | Masculino | Ramon Pinedo | |

Le muestra el siguiente mensaje: ¿Estás seguro de que deseas eliminar al paciente Alejandro Pinedo Cruz?, si presiona “Cancelar”, le lleva al formulario “Reporte de Pacientes”, si presiona “Confirmar Eliminación”, nuevamente llama al formulario “Resultado.ejs”, donde le sale un mensaje “Paciente eliminado correctamente”, también le

sale el id, el botón de “[Inicio](#)”, que va a la pagina principal. Los detalles se mostrarán en las siguientes imágenes:

The image consists of three vertically stacked screenshots of a web application interface.

Screenshot 1: A confirmation dialog box titled "¿Estás seguro de que deseas eliminar al paciente Alejandro Pinedo Cruz?". It contains two buttons: "Confirmar Eliminación" (Confirm Deletion) in red and "Cancelar" (Cancel) in grey.

Screenshot 2: A success message box titled "Paciente eliminado correctamente" (Patient deleted correctly). It shows the ID of the deleted patient: "ID del registro: 13". Below it is a blue "Inicio" button.

Screenshot 3: A report page titled "Reporte de Pacientes". It displays a table of patients with columns: ID, Nombre, Edad, Sexo, Tutor, and Acciones (Actions). The table rows are:

| ID | Nombre | Edad | Sexo | Tutor | Acciones |
|----|--------------------|------|-----------|------------------|--|
| 1 | Mathias Rodrigues | 7 | Masculino | Edwin Rodrigues | Modificar Eliminar |
| 2 | Brian Vargas | 15 | Masculino | Alberto Vargas | Modificar Eliminar |
| 3 | Emely Leon | 10 | Femenino | Felicidad Quispe | Modificar Eliminar |
| 4 | Stephany Chumacero | 12 | Femenino | Felicidad Quispe | Modificar Eliminar |
| 5 | Laura rojas | 5 | Femenino | Alex Rojas | Modificar Eliminar |
| 9 | Laura Treviño | 5 | Femenino | Ever Treviño | Modificar Eliminar |
| 10 | Johonna Aríez | 12 | Femenino | Emilio Duarte | Modificar Eliminar |

At the bottom of the report page are two buttons: "Registrar otro paciente" (Register another patient) and "Inicio".

Aquí se verifica que Alejandro Pinedo Cruz se eliminó correctamente.

ii. Reporte Evaluaciones.

El mismo procedimiento que en el “Reporte Pacientes”. Los detalles se mostrarán en las siguientes imágenes:

| ID | Paciente | Fecha | Resultado | Psicopedagoga | Acciones |
|----|-------------------|------------|--|-----------------|--|
| 1 | Mathias Rodriguez | 2023-03-03 | H niño, ingresa con un cuadro de hipersensibilidad incontrolable, se trabaja con los temas de cada sesión, y se entrega ayuda en casa al tutor para que siga con el tratamiento y sepa manejar su hipersensibilidad. | Martha Guerrero | Modificar Eliminar |
| 2 | Stephany Chávez | 2023-04-07 | Se refugio en el área de matemáticas, le gusta las matemáticas, no le gusta la lectura ya que va perdiendo el interés, pero le gusta contar cuentos que se les inventa al momento de contar. | María Guerrero | Modificar Eliminar |
| 3 | Laura rojas | 2023-03-06 | Tiene problemas para seguir instrucciones, pierde la concentración rápidamente y al mismo tiempo es muy leática, habla demasiado e interrumpe. | Gabriela Garcia | Modificar Eliminar |
| 4 | Brian Vargas | 2023-04-16 | dificultad para concentrarse en una tarea, dificultad para manejar su tiempo, parece sentirse apático y desmotivado. | Martha Guerrero | Modificar Eliminar |
| 7 | Emely Leon | 2023-06-04 | Es una niña con déficit de atención, no puede concentrarse. | María Guerrero | Modificar Eliminar |

Registrar otra Evaluación Inicio

Al presionar el botón “Modificar”.

Editar Evaluación

Fecha: 01/04/2023

Resultado: Se refugio en el área de matemáticas, le gusta las matemáticas, no le gusta la lectura ya que va perdiendo el interés, pero le gusta contar cuentos que se les inventa al momento de contar.

Psicopedagoga: **Gab**

Gabriela Garcia Guardar Cambios Cancelar

Se cambio a la paciente con otra doctora, los cambios se muestran al momento de presionar “Guardar Cambios”, como esta en el recuadro verde el cambio.

| ID | Paciente | Fecha | Resultado | Psicopedagoga | Acciones |
|----|-------------------|------------|---|------------------------|--|
| 1 | Mathias Rodriguez | 2023-03-03 | El niño, ingresa con un cuadro de hipersensibilidad incontrolable, se trabaja con los temas de cada sesión, y se entrega ayuda en casa al tutor para que siga con el tratamiento y sepa manejar su hipersensibilidad. | María Guerrero | Modificar Eliminar |
| 2 | Stephany Chávez | 2023-04-07 | Se refugio en el área de matemáticas, le gusta las matemáticas, no le gusta la lectura ya que va perdiendo el interés, pero le gusta contar cuentos que se les inventa al momento de contar. | Gabriela Garcia | Modificar Eliminar |
| 3 | Laura rojas | 2023-03-06 | Tiene problemas para seguir instrucciones, pierde la concentración rápidamente y al mismo tiempo es muy leática, habla demasiado e interrumpe. | Gabriela Garcia | Modificar Eliminar |
| 4 | Brian Vargas | 2023-04-16 | dificultad para concentrarse en una tarea, dificultad para manejar su tiempo, parece sentirse apático y desmotivado. | María Guerrero | Modificar Eliminar |
| 7 | Emely Leon | 2023-06-04 | Es una niña con déficit de atención, no puede concentrarse. | María Guerrero | Modificar Eliminar |

Registrar otra Evaluación Inicio

Al presionar el botón Eliminar le aparece el formulario donde le pregunta: ¿Deseas eliminar la

evaluación de Mathias Rodrigues? Y preciona cualquiera de los 2 botones que se muestra.



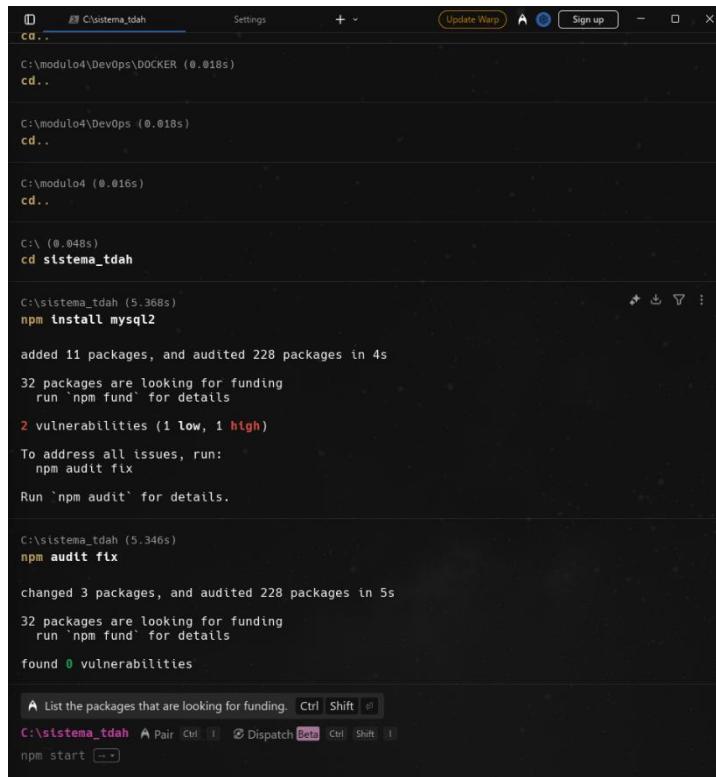
iii. Reporte Tareas.

El mismo procedimiento que en el “[Reporte Pacientes](#)”. El formulario “[Reporte de Tareas](#)”, tiene los botones modificar y eliminar que funcionan de igual manera que las demás.

| Reporte de Tareas | | | | | |
|-------------------|-------------------|--|------------|--|----------|
| ID | Paciente | Descripción | Fecha | Avance | Acciones |
| 1 | Mathias Rodrigues | Se conoció al momento de armar los compuertas, de acuerdo al momento de la lectura se observó varias veces, al igual en la caligrafía algunas veces confunde, ejemplo la letra d con la t. | 2025-03-03 | - Rompecabezas de abecedario - Rompecabezas variadas - Lectura - Caligrafía | |
| 2 | Laura Rojas | Hizo la tarea del colegio, se le ayudó, lo guío a aprender los colores y sonreí las diferentes frases. | 2025-04-04 | - Aprende Colores - aprendiendo los colores y sonrisas | |
| 4 | Brian Vargas | Le gustó tocar la guitarra, eliminó sus miedos de adicción - Inicio clases de guitarra. | 2025-06-02 | - Le gusta la matemática, retorna lo avanzado - Es un bien que lo gusta - Toma clases de guitarra. | |

7. Cambiar la base de Datos a MySQL.

Para cambiar de SQLite a otra Base de Datos MySQL sin que pierda ninguna funcionalidad el sistema, lo primero, que se hace es instalar la base de datos con “[npm install mysql2](#)”, se instaló correctamente la base de datos, pero hay 2 vulnerabilidades ósea fallas de seguridad, para corregirlas se pone “[npm audit fix](#)”, también hay dependencias de apoyo económico, pero funcionara con el sistema de tdah ya que no es complejo, la instalación y corrección se muestra en la siguiente imagen:



```

C:\sistema_tdah
Settings + Update Warp A Sign up X
cd..
C:\modulo4\DevOps\DOCKER (0.018s)
cd..

C:\modulo4\DevOps (0.018s)
cd..

C:\ (0.048s)
cd sistema_tdah

C:\sistema_tdah (5.368s)
npm install mysql2

added 11 packages, and audited 228 packages in 4s
32 packages are looking for funding
  run `npm fund` for details
  2 vulnerabilities (1 low, 1 high)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.

C:\sistema_tdah (5.346s)
npm audit fix

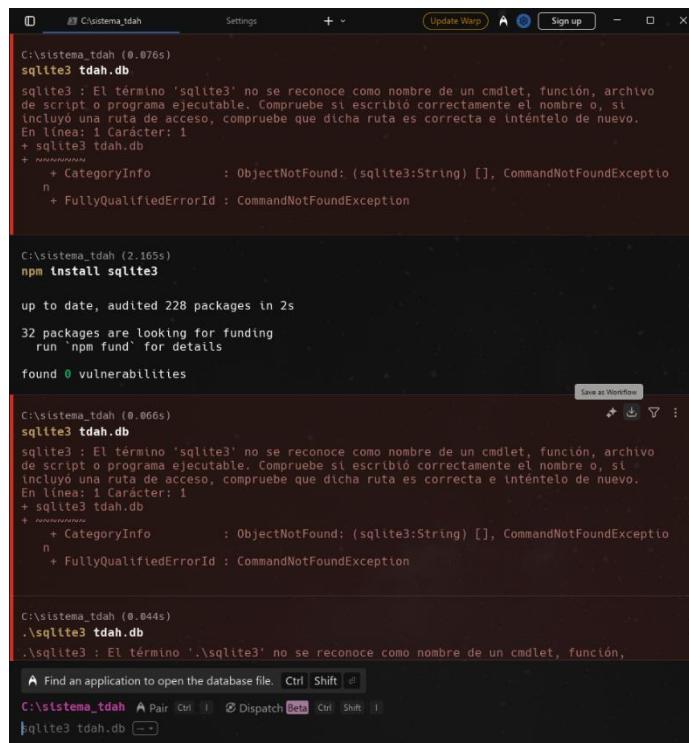
changed 3 packages, and audited 228 packages in 5s
32 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities

A List the packages that are looking for funding. Ctrl Shift ⌘
C:\sistema_tdah A Pair Ctrl ⌘ ⚡ Dispatch Beta Ctrl Shift ⌘
npm start [-]

```

8. Migrar la base de datos de SQLite a la base de datos MySQL.

Para migrar la base de datos se va al Warp y se ejecuta “`sqlite3 tdah.db`”, como no funciona se instala el SQLite “`npm install sqlite3`”, nuevamente probamos y no funciona, porque no se tiene acceso al comando SQLite en la terminal, es una herramienta aparte llamada CLI (Command Line Interface, interfaz de linea de comandos) de SQLite.



```

C:\sistema_tdah
Settings + Update Warp A Sign up X
sqlite3 tdah.db
sqlite3 : El término 'sqlite3' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e intételo de nuevo.
En línea: 1 Carácter: 1
+ sqlite3 tdah.db
+ ~~~~~
  + CategoryInfo          : ObjectNotFound: (sqlite3:String) [], CommandNotFoundException
  n
  + FullyQualifiedErrorId : CommandNotFoundException

C:\sistema_tdah (2.165s)
npm install sqlite3

up to date, audited 228 packages in 2s
32 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities

C:\sistema_tdah (0.066s)
sqlite3 tdah.db
sqlite3 : El término 'sqlite3' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e intételo de nuevo.
En línea: 1 Carácter: 1
+ sqlite3 tdah.db
+ ~~~~~
  + CategoryInfo          : ObjectNotFound: (sqlite3:String) [], CommandNotFoundException
  n
  + FullyQualifiedErrorId : CommandNotFoundException

C:\sistema_tdah (0.044s)
.\sqlite3 tdah.db
.\sqlite3 : El término '.\sqlite3' no se reconoce como nombre de un cmdlet, función,
A Find an application to open the database file. Ctrl Shift ⌘
C:\sistema_tdah A Pair Ctrl ⌘ ⚡ Dispatch Beta Ctrl Shift ⌘
sqlite3 tdah.db [-]

```

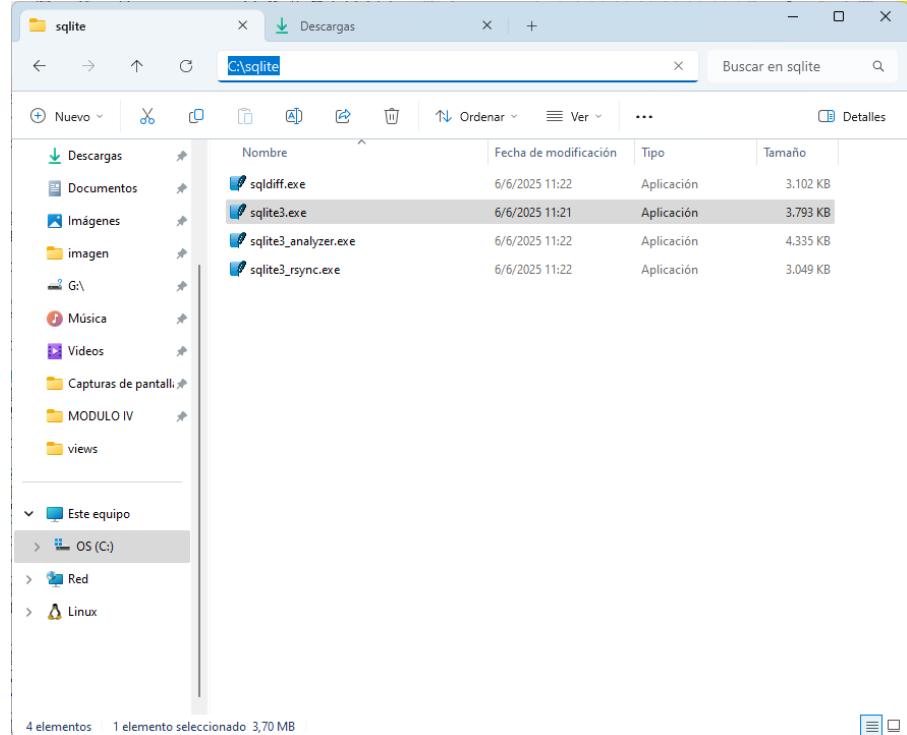
Como en la imagen muestra el error, se va a la pagina oficial de SQLite "<https://www.sqlite.org/download.html>", se busca “**Binarios precompilados para Windows**” o en ingles “**Precompiled Binaries for Windows**”, se descarga “[sqlite-tools-win-x64-3500100.zip](https://www.sqlite.org/2023/sqlite-tools-win-x64-3500100.zip)”, segun a las especificaciones de tu computadora, se extrae el zip como se muestra en la imagen a una carpeta “c:\sqlite”.

The screenshot shows the SQLite download page with the 'Precompiled Binaries for Windows' section highlighted. It lists several files for download, including:

- sqlite-dll-win-x86-3500100.zip**: DLL de 32 bits (x86) para SQLite versión 3.50.1. (1,02 MB)
- sqlite-dll-win-x64-3500100.zip**: DLL de 64 bits (x64) para SQLite versión 3.50.1. (1,28 MB)
- sqlite-tools-win-x64-3500100.zip**: Un conjunto de herramientas de línea de comandos para administrar archivos de bases de datos SQLite, que incluye (1) ejecutables: **sqlite3.exe**. (6,13 MB)

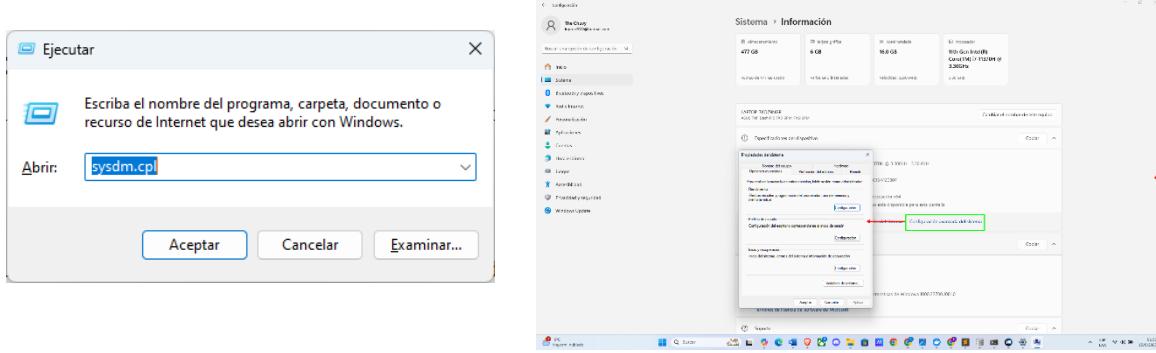
Binarios precompilados para .NET

Sistema.Datos.SQLite Visite el sitio web System.Data.SQLite.org y especialmente la página de descarga del código fuente y los binarios de SQL.

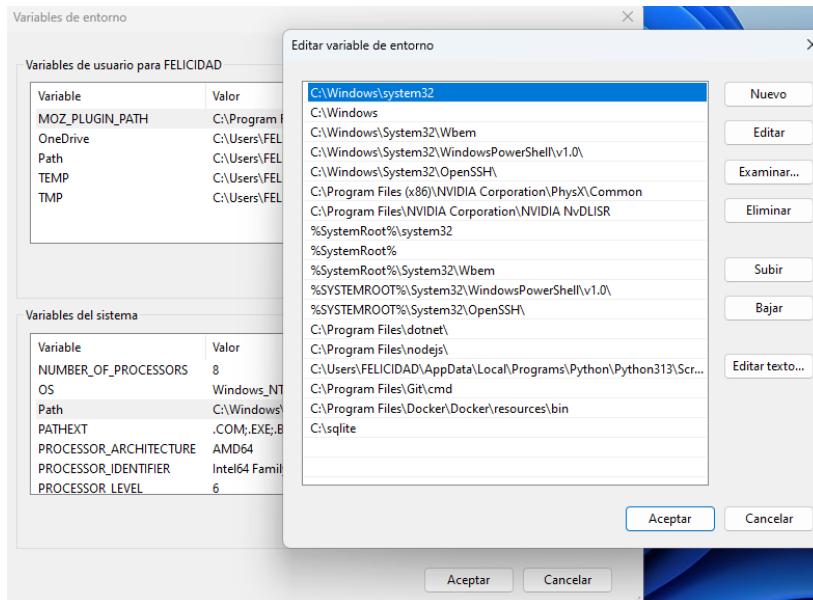


Para que funcione en cualquier terminal se agrega SQLite al PATH con los siguientes pasos:

- 💡 Preciona la tecla **win + R** y escribe **sysdm.cpl** o te vas a configuracion a sistemas, eliges “**Configuracion avanzada del sistema**”, aparecerá la ventana “**Propiedades del sistema**”, como se muestra en la imagen:



- ➡ Elige la pestaña “opciones avanzadas”, clic en “variables de entorno”.
- ➡ Buscas Path y haces clic en “Editar”, se elige la ruta donde se extrajo lo que se descargo, una vez agregado la ruta se acepta todo, como se muestra en la imagen:

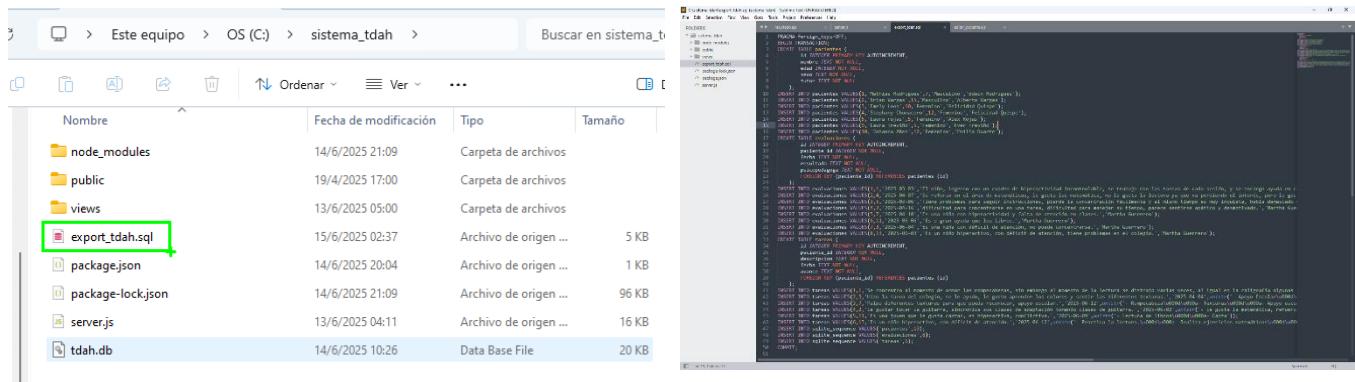


Se vuelve a ejecutar “`sqlite3 tdah.db`”, y se deberia entrar al prompt de SQLite, luego se ejecuta lo siguiente como se muestra en la imagen.

```
C:\sistema_tdah (5h 3m 53s)
sqlite3 tdah.db
SQLite version 3.50.1 2025-06-06 14:52:32
Enter ".help" for usage hints.
sqlite> .output export_tdah.sql
sqlite> .dump
sqlite> .exit

C:\sistema_tdah A Pair Ctrl I Dispatch Beta Ctrl Shift I
| exit → |
```

Se generara un archivo "export_tdhah.sql", contiene las instrucciones para crear las tablas que estan en tdah.db y para modificarlo se puede abrir ese archivo en visual studio code o Notepad++, en la imagen se muestra el archivo creado.



Se debe realizar los cambios con la sintaxis del MySQL y guardar con el nombre de "sistema_tdhah_mysql.sql", los cambios lo muestro a continuacion:

```

-- Crear tabla pacientes
CREATE TABLE IF EXISTS pacientes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
    edad INT NOT NULL,
    sexo ENUM('M', 'F') NOT NULL,
    telefono VARCHAR(15) NOT NULL
);

-- Insertar pacientes
INSERT INTO pacientes (nombre, apellido, edad, sexo, telefono)
VALUES ('Luis Rodriguez', 'Rodriguez', 25, 'M', '9876543210'),
       ('Maria Hernandez', 'Hernandez', 28, 'F', '9876543211'),
       ('Pedro Lopez', 'Lopez', 30, 'M', '9876543212'),
       ('Ana Gomez', 'Gomez', 22, 'F', '9876543213'),
       ('Juan Perez', 'Perez', 40, 'M', '9876543214'),
       ('Karla Diaz', 'Diaz', 20, 'F', '9876543215'),
       ('Laura Lopez', 'Lopez', 18, 'F', '9876543216'),
       ('Diego Sanchez', 'Sanchez', 27, 'M', '9876543217'),
       ('Carmen Ramirez', 'Ramirez', 35, 'F', '9876543218'),
       ('Eduardo Gutierrez', 'Gutierrez', 32, 'M', '9876543219');

-- Crear tabla evaluaciones
CREATE TABLE IF EXISTS evaluaciones (
    id INT AUTO_INCREMENT PRIMARY KEY,
    paciente_id INT NOT NULL,
    fecha DATE NOT NULL,
    resultado TEXT NOT NULL,
    paciente_nombre VARCHAR(50),
    paciente_apellido VARCHAR(50),
    paciente_edad INT NOT NULL,
    paciente_sexo ENUM('M', 'F') NOT NULL,
    paciente_telefono VARCHAR(15) NOT NULL
);

-- Insertar evaluaciones
INSERT INTO evaluaciones (paciente_id, fecha, resultado, paciente_nombre, paciente_apellido, paciente_edad, paciente_sexo, paciente_telefono)
VALUES (1, '2025-03-01', 'Es una paciente con hipertension y dolor de cabeza.', 'Luis', 'Rodriguez', 25, 'M', '9876543210'),
       (2, '2025-03-02', 'Es una paciente con diabetes tipo 2 y colesterol alto.', 'Maria', 'Hernandez', 28, 'F', '9876543211'),
       (3, '2025-03-03', 'Es una paciente con hipertension y dolor de cabeza.', 'Pedro', 'Lopez', 30, 'M', '9876543212'),
       (4, '2025-03-04', 'Es una paciente con hipertension y dolor de cabeza.', 'Ana', 'Gomez', 22, 'F', '9876543213'),
       (5, '2025-03-05', 'Es una paciente con hipertension y dolor de cabeza.', 'Juan', 'Perez', 40, 'M', '9876543214'),
       (6, '2025-03-06', 'Es una paciente con hipertension y dolor de cabeza.', 'Karla', 'Diaz', 20, 'F', '9876543215'),
       (7, '2025-03-07', 'Es una paciente con hipertension y dolor de cabeza.', 'Laura', 'Lopez', 18, 'F', '9876543216'),
       (8, '2025-03-08', 'Es una paciente con hipertension y dolor de cabeza.', 'Diego', 'Sanchez', 27, 'M', '9876543217'),
       (9, '2025-03-09', 'Es una paciente con hipertension y dolor de cabeza.', 'Carmen', 'Ramirez', 35, 'F', '9876543218'),
       (10, '2025-03-10', 'Es una paciente con hipertension y dolor de cabeza.', 'Eduardo', 'Gutierrez', 32, 'M', '9876543219');

-- Crear tabla turnos
CREATE TABLE IF EXISTS turnos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    paciente_id INT NOT NULL,
    descripcion TEXT NOT NULL,
    fecha DATE NOT NULL,
    hora TIME NOT NULL
);

```

Una vez cambiado se debe cargar en MySQL, para ello se necesita descargar MySQL Workbench y el server, desde la pagina principal de MySQL <https://dev.mysql.com/downloads/mysql/>, como muestra la imagen:

Al querer descargar le aparece la siguiente imagen:

The screenshot shows the MySQL Downloads page. At the top, there's a message encouraging users to log in or register for a free account. Below this, a list of benefits for using Oracle web accounts is provided. Two prominent buttons are visible: 'Iniciar sesión »' (Sign In) and 'Regístrate »' (Register). A note about Oracle SSO authentication is present. At the bottom, a button labeled 'No, gracias, solo comienzo mi descarga.' (No, thanks, I'll start my download) is highlighted with a yellow border.

No, gracias, solo comienzo mi descarga.

Iniciar sesión »
con mi cuenta web de Oracle

Regístrate »
para obtener una cuenta web de Oracle

ORACLE © Oráculo 2025
Privacidad / No vender mi información | Términos de uso | Política de Marcas | Preferencias sobre cookies

Se debe elegir iniciar sesión o registrarse, pero se puede elegir lo que está enmarcado, una vez descargado instalarlas.

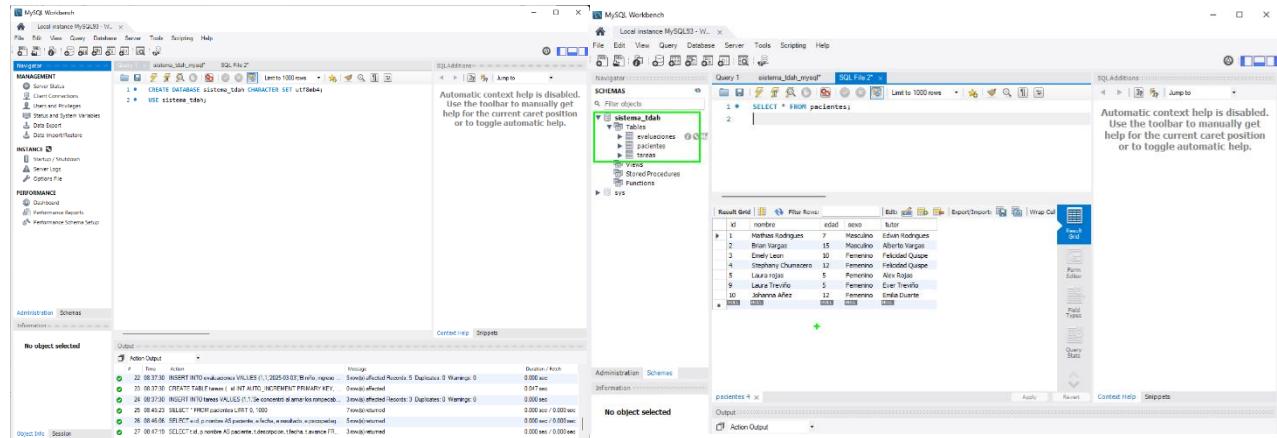
También se necesita instalar el MySQL Workbench se va a la siguiente página "<https://dev.mysql.com/downloads/workbench/>", se instala como la anterior.

The left side of the image shows the MySQL Workbench 8.0.42 download page. It features a 'MySQL Installer for Windows' section with a download link for 'Windows (x86, 32 & 64-bit), MySQL Installer MSI'. The right side shows the 'MySQL Workbench 8.0 CE - Setup Wizard' window titled 'Installing MySQL Workbench 8.0 CE'. The status bar indicates 'Please wait while the Setup Wizard installs MySQL Workbench 8.0 CE. This may take several minutes.' A progress bar is shown at the bottom.

Luego le aparece la siguiente pantalla: la primera no se sabe la contraseña por eso se instalo otra.

The screenshot shows the MySQL Workbench interface. The title bar says 'MySQL Workbench'. The main area displays the 'Welcome to MySQL Workbench' message, which describes the tool as the official graphical user interface for MySQL. It also includes links to 'Browse Documentation', 'Read the Blog', and 'Discuss on the Forums'. Below this, the 'MySQL Connections' section shows two local instances: 'Local instance mysql' and 'Local instance MySQL93', both connected to 'localhost:3306' as the root user.

Se ingresa a “Local instance MySQL93”, te pide la clave, una vez ingresado crear la base de datos y hace la migracion con el archivo generado anteriormente con el nombre de “sistema_tdh_mysql.sql”, en la parte de abajo en la ventana “Output”, muestra si se genero correctamente los datos a la base de datos creado al principio con el nombre de “sistema_tdh”, tambien se puede hacer consultas sql en el “SQL File 2”, que se genero al ir a “File”, elige “New Query Tab”, a continuacion se mostrara las imagenes:



Con este archivo se hizo la migracion “sistema_tdh_mysql.sql”, una vez abierto se preciona el rayo amarillo que es el “Execute”, el codigo se muestra a continuacion:

```
USE sistema_tdh;
```

```
DROP TABLE IF EXISTS tareas;
```

```
DROP TABLE IF EXISTS evaluaciones;
```

```
DROP TABLE IF EXISTS pacientes;
```

```
-- Crear tabla pacientes
```

```
CREATE TABLE pacientes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    edad INT NOT NULL,
    sexo VARCHAR(20) NOT NULL,
    tutor VARCHAR(100) NOT NULL
);
```

```
-- Insertar pacientes
```

```
INSERT INTO pacientes VALUES
```

```
(1,'Mathias Rodrigues',7,'Masculino','Edwin Rodrigues'),
(2,'Brian Vargas',15,'Masculino','Alberto Vargas'),
(3,'Emely Leon',10,'Femenino','Felicidad Quispe'),
(4,'Stephany Chumacero',12,'Femenino','Felicidad Quispe'),
(5,'Laura rojas',5,'Femenino','Alex Rojas'),
(9,'Laura Treviño',5,'Femenino','Ever Treviño'),
(10,'Johanna Añez',12,'Femenino','Emilia Duarte');
```

-- Crear tabla evaluaciones

```
CREATE TABLE evaluaciones (
    id INT AUTO_INCREMENT PRIMARY KEY,
    paciente_id INT NOT NULL,
    fecha DATE NOT NULL,
    resultado TEXT NOT NULL,
    psicopedagoga VARCHAR(100) NOT NULL,
    FOREIGN KEY (paciente_id) REFERENCES pacientes(id)
);
```

-- Insertar evaluaciones

```
INSERT INTO evaluaciones VALUES
(1,1,'2025-03-03','El niño, ingreso con un cuadro de hiperactividad
incontrolable...','Martha Guerrero'),
(2,4,'2025-04-07','Se reforzó en el área de matemáticas, le gusta las
matemáticas...','Gabriela Garcia'),
(3,5,'2025-03-06','Tiene problemas para seguir instrucciones, pierde la
concentración...','Gabriela Garcia'),
(4,2,'2025-04-16','Dificultad para concentrarse en una tarea...','Martha
Guerrero'),
(7,3,'2025-06-04','Es una niña con déficit de atención, no puede
concentrarse.','Martha Guerrero');
```

-- Crear tabla tareas

```
CREATE TABLE tareas (
```

```

    id INT AUTO_INCREMENT PRIMARY KEY,
    paciente_id INT NOT NULL,
    descripcion TEXT NOT NULL,
    fecha DATE NOT NULL,
    avance TEXT NOT NULL,
    FOREIGN KEY (paciente_id) REFERENCES pacientes(id)
);

-- Insertar tareas
INSERT INTO tareas VALUES
(1,1,'Se concentró al armar los rompecabezas...','2025-03-03','-Rompecabezas de abecedario\n- Rompecabezas vocales\n-Lectura\n- Caligrafía'),
(2,5,'Hizo la tarea del colegio, le gustó aprender los colores...','2025-04-04','- Apoyo Escolar\n- Aprendiendo colores y texturas'),
(4,2,'Le gusta tocar la guitarra, clases sincronizadas con adaptación.','2025-06-02','- Refuerzo matemático\n- Lectura por gusto\n- Clases de guitarra');

```

La migracion fue un éxito.

- ✚ Se creo la base de datos sistema_tdah.
- ✚ Se importo los datos transformados desde SQLite.
- ✚ Se ejecuto las relaciones entre tablas sin errores.
- ✚ Se probó los registros, se ven correctamente con consultas SQL.
- ✚ Se verifico las claves foráneas y estructuras.

9. Como conectar la base de datos MySQL al Backend.

Se debe eliminar la base de datos “[tdah.db](#)” de SQLite para evitar confusiones, porque ya se migro a MySQL ya que es una base de datos externa. se crea un archivo app.js, en el cual se hará los cambios de server con la sintaxis de la base de datos MySQL el cambio es la siguiente:

[App.js](#)

```

require('dotenv').config() // para usar el archivo .env
const express = require('express');
const bodyParser = require('body-parser');

```

```
const db = require('./config/db'); // conexión MySQL

const app = express();

app.use(bodyParser.urlencoded({ extended: true }));

app.set('view engine', 'ejs');

// para que public sea accesible

app.use(express.static('public'));

// Configurar la carpeta 'views' para EJS

app.set('views', __dirname + '/views');

// Ruta principal

app.get('/', (req, res) => {

    res.send("¡BIENVENIDO AL SISTEMA DE GESTIÓN TDAH!");

});

// Muestra la pagina principal menu

app.get('/menu', (req, res) => {

    res.render('menu');

});

// Ruta para mostrar formulario de registro de pacientes

app.get('/registro', (req, res) => {

    res.render('registro');

});

// Guardar el registro en la base de datos, tabla pacientes

app.post('/guardar-paciente', async (req, res) => {
```

```
const { nombre, edad, sexo, tutor } = req.body;

try {

    const [resultado] = await db.query(
        'INSERT INTO pacientes (nombre, edad, sexo, tutor)
VALUES (?, ?, ?, ?)',
        [nombre, edad, sexo, tutor]
    );

    console.log("Paciente registrado con éxito, ID:",
resultado.insertId);

    res.render('resultado', {
        mensaje: "Paciente registrado satisfactoriamente",
        id: resultado.insertId,
        rutaEliminar: null
    });

}

} catch (err) {

    console.error("Error al insertar paciente:", err.message);
    res.render('resultado', { mensaje: "Error al registrar
paciente", id: null, rutaEliminar: null });

}

});

// Mostrar formulario para registrar evaluación (con lista de
pacientes)

app.get('/registro_evaluacion', async (req, res) => {

    try {
```

```
        const [pacientes] = await db.query("SELECT id, nombre  
FROM pacientes");  
  
        res.render('registro_evaluacion', { pacientes });  
  
    } catch (err) {  
  
        console.error("Error al cargar pacientes:", err.message);  
  
        res.send("Error al mostrar el formulario de evaluación.");  
  
    }  
  
});  
  
  
// Guardar evaluación  
  
app.post('/guardar-evaluacion', async (req, res) => {  
  
    const { paciente_id, fecha, resultado, psicopedagoga } =  
    req.body;  
  
    try {  
  
        const [resultadoQuery] = await db.query(  
  
            "INSERT INTO evaluaciones (paciente_id, fecha,  
resultado, psicopedagoga) VALUES (?, ?, ?, ?)",  
  
            [paciente_id, fecha, resultado, psicopedagoga]  
  
        );  
  
  
        console.log("Evaluación registrada con éxito, ID:",  
resultadoQuery.insertId);  
  
        res.render('resultado', {  
  
            mensaje: "Evaluación registrada satisfactoriamente",  
            id: resultadoQuery.insertId,  
            rutaEliminar: null  
        });  
    }  
});
```

```
        } catch (err) {  
            console.error("Error al insertar evaluación:", err.message);  
            res.render('resultado', { mensaje: "Error al registrar  
evaluación", id: null, rutaEliminar: null });  
        }  
    });  
  
// Mostrar formulario para registrar tarea (con lista de  
pacientes)  
app.get('/registro_tarea', async (req, res) => {  
    try {  
        const [pacientes] = await db.query("SELECT id, nombre  
FROM pacientes");  
        res.render('registro_tarea', { pacientes });  
    } catch (err) {  
        console.error("Error al cargar pacientes para tareas:",  
err.message);  
        res.send("Error al mostrar el formulario de tarea.");  
    }  
});  
  
// Guardar tarea  
app.post('/guardar-tarea', async (req, res) => {  
    const { paciente_id, descripcion, fecha, avance } = req.body;  
    try {  
        const [resultadoQuery] = await db.query(  
            "INSERT INTO tareas (paciente_id, descripcion, fecha,  
avance) VALUES (?, ?, ?, ?)",  
        );  
    } catch (err) {  
        console.error("Error al guardar tarea:", err.message);  
        res.send("Error al guardar la tarea.");  
    }  
});
```

```
[paciente_id, descripcion, fecha, avance]

);

console.log("Tarea registrada con éxito, ID:",
resultadoQuery.insertId);

res.render('resultado', {
  mensaje: "Tarea registrada satisfactoriamente",
  id: resultadoQuery.insertId,
  rutaEliminar: null
});

} catch (err) {
  console.error("Error al insertar tarea:", err.message);
  res.render('resultado', { mensaje: "Error al registrar tarea",
  id: null, rutaEliminar: null });
}

});

// Ruta para generar reporte de pacientes
app.get('/reporte', async (req, res) => {
  try {
    const [pacientes] = await db.query("SELECT * FROM
pacientes");
    res.render('reporte', { pacientes });
  } catch (err) {
    console.error("Error al consultar pacientes:", err.message);
    res.send("Error al generar el reporte.");
  }
})
```

```
});
```

```
// Ruta para reporte de evaluaciones con nombre del paciente
```

```
app.get('/reporte_evaluaciones', async (req, res) => {
```

```
    try {
```

```
        const [evaluaciones] = await db.query(`
```

```
            SELECT e.id, p.nombre AS paciente, e.fecha, e.resultado,  
            e.psicopedagoga
```

```
            FROM evaluaciones e
```

```
            JOIN pacientes p ON e.paciente_id = p.id
```

```
`);
```

```
        res.render('reporte_evaluaciones', { evaluaciones });
```

```
    } catch (err) {
```

```
        console.error("Error al consultar evaluaciones:",  
        err.message);
```

```
        res.send("Error al generar el reporte de evaluaciones.");
```

```
}
```

```
});
```

```
// Ruta para reporte de tareas con nombre del paciente
```

```
app.get('/reporte_tareas', async (req, res) => {
```

```
    try {
```

```
        const [tareas] = await db.query(`
```

```
            SELECT t.id, p.nombre AS paciente, t.descripcion, t.fecha,  
            t.avance
```

```
            FROM tareas t
```

```
            JOIN pacientes p ON t.paciente_id = p.id
```

```
        `);
        res.render('reporte_tareas', { tareas });
    } catch (err) {
        console.error("Error al consultar tareas:", err.message);
        res.send("Error al generar el reporte de tareas.");
    }
});

// Editar paciente
app.get('/editar_paciente/:id', async (req, res) => {
    const id = req.params.id;
    try {
        const [rows] = await db.query('SELECT * FROM pacientes WHERE id = ?', [id]);
        if (rows.length > 0) {
            res.render('editar_paciente', { paciente: rows[0] });
        } else {
            res.send("No se encontró el paciente especificado.");
        }
    } catch (err) {
        console.error("Error al cargar datos del paciente:",
err.message);
        res.send("Error al cargar el formulario.");
    }
});

app.post('/actualizar-paciente/:id', async (req, res) => {
```

```
const id = req.params.id;
const { nombre, edad, sexo, tutor } = req.body;
try {
    await db.query(
        'UPDATE pacientes SET nombre = ?, edad = ?, sexo = ?,
        tutor = ? WHERE id = ?',
        [nombre, edad, sexo, tutor, id]
    );
    res.redirect('/reporte');
} catch (err) {
    console.error("Error al actualizar el paciente:",
    err.message);
    res.send("Error al guardar los cambios.");
}
});

// Editar evaluación
app.get('/editar_evaluacion/:id', async (req, res) => {
    const id = req.params.id;
    try {
        const [rows] = await db.query('SELECT * FROM evaluaciones
        WHERE id = ?', [id]);
        if (rows.length > 0) {
            res.render('editar_evaluacion', { evaluacion: rows[0] });
        } else {
            res.send("No se encontró la evaluación especificada.");
        }
    }
});
```

```
    } catch (err) {  
        console.error("Error al cargar la evaluación:", err.message);  
        res.send("Error al cargar el formulario.");  
    }  
});  
  
app.post('/actualizar-evaluacion/:id', async (req, res) => {  
    const id = req.params.id;  
    const { fecha, resultado, psicopedagoga } = req.body;  
    try {  
        await db.query(  
            'UPDATE evaluaciones SET fecha = ?, resultado = ?,  
            psicopedagoga = ? WHERE id = ?',  
            [fecha, resultado, psicopedagoga, id]  
        );  
        res.redirect('/reporte_evaluaciones');  
    } catch (err) {  
        console.error("Error al actualizar la evaluación:",  
        err.message);  
        res.send("Error al guardar los cambios.");  
    }  
});  
  
// 🖍 Editar tarea  
app.get('/editar_tarea/:id', async (req, res) => {  
    const id = req.params.id;  
    try {
```

```
        const [rows] = await db.query('SELECT * FROM tareas
WHERE id = ?', [id]);

        if (rows.length > 0) {
            res.render('editar_tarea', { tarea: rows[0] });
        } else {
            res.send("No se encontró la tarea especificada.");
        }
    } catch (err) {
        console.error("Error al cargar la tarea:", err.message);
        res.send("Error al cargar el formulario.");
    }
});

app.post('/actualizar-tarea/:id', async (req, res) => {
    const id = req.params.id;
    const { descripcion, fecha, avance } = req.body;
    try {
        await db.query(
            'UPDATE tareas SET descripcion = ?, fecha = ?, avance = ?
WHERE id = ?',
            [descripcion, fecha, avance, id]
        );
        res.redirect('/reporte_tareas');
    } catch (err) {
        console.error("Error al actualizar la tarea:", err.message);
        res.send("Error al guardar los cambios.");
    }
});
```

```
});

// Rutas para eliminar un paciente en la tabla pacientes
app.get('/eliminar_paciente/:id', async (req, res) => {
    const id = req.params.id;
    const volver = req.query.volver || '/menu';
    try {
        const [rows] = await db.query('SELECT nombre FROM pacientes WHERE id = ?', [id]);
        if (rows.length === 0) {
            return res.render('resultado', { mensaje: "Paciente no encontrado", id: null, rutaEliminar: null });
        }
        const mensaje = `¿Estás seguro de que deseas eliminar al paciente ${rows[0].nombre}?`;
        const rutaEliminar = `/eliminar-paciente/${id}?volver=${volver}`;
        res.render('resultado', { mensaje, id, rutaEliminar, volverA: volver });
    } catch (err) {
        console.error("Error al buscar paciente:", err.message);
        res.send("Error al buscar el paciente.");
    }
});

app.post('/eliminar-paciente/:id', async (req, res) => {
    const id = req.params.id;
    const volver = req.query.volver || '/menu';
    try {
        const [rows] = await db.query('DELETE FROM pacientes WHERE id = ?', [id]);
        if (rows.affectedRows === 1) {
            res.redirect(volver);
        } else {
            res.send("Error al eliminar el paciente.");
        }
    } catch (err) {
        console.error("Error al eliminar paciente:", err.message);
        res.send("Error al eliminar el paciente.");
    }
});
```

```
try {  
    await db.query('DELETE FROM pacientes WHERE id = ?', [id]);  
    res.render('resultado', { mensaje: "Paciente eliminado  
correctamente", id, rutaEliminar: null, volverA: volver });  
} catch (err) {  
    console.error("Error al eliminar paciente:", err.message);  
    res.render('resultado', { mensaje: "Error al eliminar", id:  
null, rutaEliminar: null, volverA: volver });  
}  
});  
  
// Rutas para eliminar datos de un paciente en la tabla  
evaluaciones  
app.get('/eliminar_evaluacion/:id', async (req, res) => {  
    const id = req.params.id;  
    const volver = req.query.volver || '/menu';  
    try {  
        const [rows] = await db.query(`  
            SELECT e.id, p.nombre AS paciente  
            FROM evaluaciones e  
            JOIN pacientes p ON e.paciente_id = p.id  
            WHERE e.id = ?  
        `, [id]);  
        if (rows.length === 0) {  
            return res.render('resultado', { mensaje: "Evaluación no  
encontrada", id: null, rutaEliminar: null });  
        }  
    }  
});
```

```
        const mensaje = `¿Deseas eliminar la evaluación de
${rows[0].paciente}?`;

        const rutaEliminar = `/eliminar-
evaluacion/${id}?volver=${volver}`;

        res.render('resultado', { mensaje, id, rutaEliminar, volverA:
volver });

    } catch (err) {

        console.error("Error al buscar evaluación:", err.message);
        res.send("Error al buscar la evaluación.");

    }

});

app.post('/eliminar-evaluacion/:id', async (req, res) => {

    const id = req.params.id;

    const volver = req.query.volver || '/menu';

    try {

        await db.query('DELETE FROM evaluaciones WHERE id = ?',
[id]);

        res.render('resultado', { mensaje: "Evaluación eliminada
correctamente", id, rutaEliminar: null, volverA: volver });

    } catch (err) {

        console.error("Error al eliminar evaluación:", err.message);

        res.render('resultado', { mensaje: "Error al eliminar
evaluación", id: null, rutaEliminar: null, volverA: volver });

    }

});

// Rutas para eliminar datos de un paciente en la tabla tareas
```

```
app.get('/eliminar_tarea/:id', async (req, res) => {
    const id = req.params.id;
    const volver = req.query.volver || '/menu';
    try {
        const [rows] = await db.query(`SELECT t.id, p.nombre AS paciente
        FROM tareas t
        JOIN pacientes p ON t.paciente_id = p.id
        WHERE t.id = ?
        `, [id]);
        if (rows.length === 0) {
            return res.render('resultado', { mensaje: "Tarea no encontrada", id: null, rutaEliminar: null });
        }
        const mensaje = `¿Deseas eliminar la tarea de ${rows[0].paciente}?`;
        const rutaEliminar = `/eliminar-tarea/${id}?volver=${volver}`;
        res.render('resultado', { mensaje, id, rutaEliminar, volverA: volver });
    } catch (err) {
        console.error("Error al buscar tarea:", err.message);
        res.send("Error al buscar la tarea.");
    }
});

app.post('/eliminar-tarea/:id', async (req, res) => {
```

```

const id = req.params.id;
const volver = req.query.volver || '/menu';

try {

    await db.query('DELETE FROM tareas WHERE id = ?', [id]);

    res.render('resultado', { mensaje: "Tarea eliminada
correctamente", id, rutaEliminar: null, volverA: volver });

} catch (err) {

    console.error("Error al eliminar tarea:", err.message);

    res.render('resultado', { mensaje: "Error al eliminar tarea",
id: null, rutaEliminar: null, volverA: volver });

}

});

// Iniciar el servidor

app.listen(3000, () => {

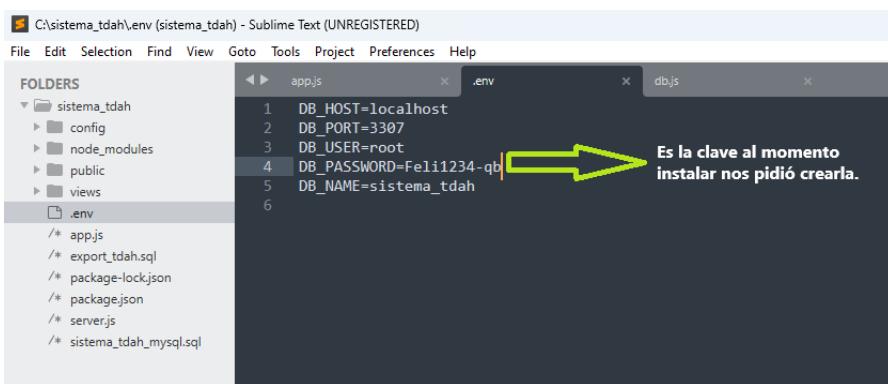
    console.log("Servidor ejecutándose en el puerto 3000");

});

```

10. Instalación de dotenv y creación del db.js y .env.

Se crea el archivo .env y se coloca el siguiente código.



```

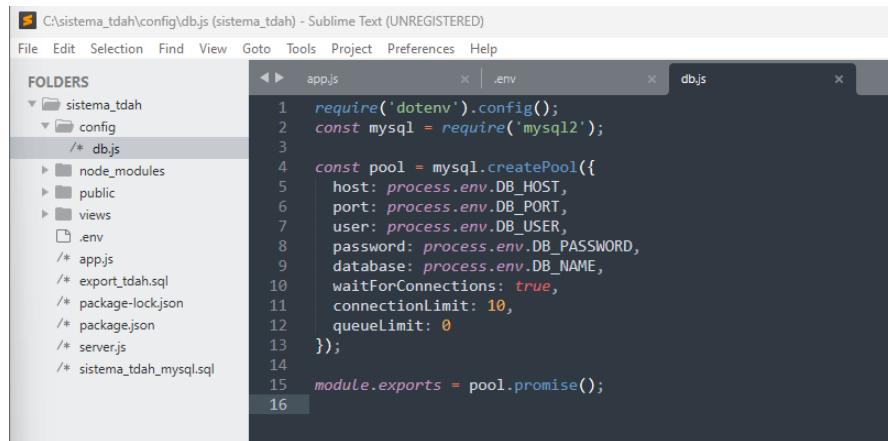
C:\sistema_tdah\.env (sistema_tdah) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  sistema_tdah
    config
    node_modules
    public
    views
    .env
      /* app.js
      /* export_tdah.sql
      /* package-lock.json
      /* package.json
      /* server.js
      /* sistema_tdah_mysql.sql
.app.js .env db.js
1 DB_HOST=localhost
2 DB_PORT=3307
3 DB_USER=root
4 DB_PASSWORD=Feli1234-qb| Es la clave al momento
5 DB_NAME=sistema_tdah
6

```

El punto 1 es la dirección del servidor MySQL ósea el localhost.

- 2 Es el puerto de MySQL como se instaló 2 veces se cambió el puerto a 3307 ya que el anterior está con el puerto 3306.
- 3 Es el usuario ósea root.
- 4 Es la clave de acceso a MySQL.
- 5 Es el nombre de la base de datos ósea sistema_tdah.

Se el archivo “[db.js](#)”, dentro de config, que también se debe crear, con botón derecho “[New Folder](#)”, y se coloca “[config](#)”, y dentro de db.js debe estar el siguiente código:

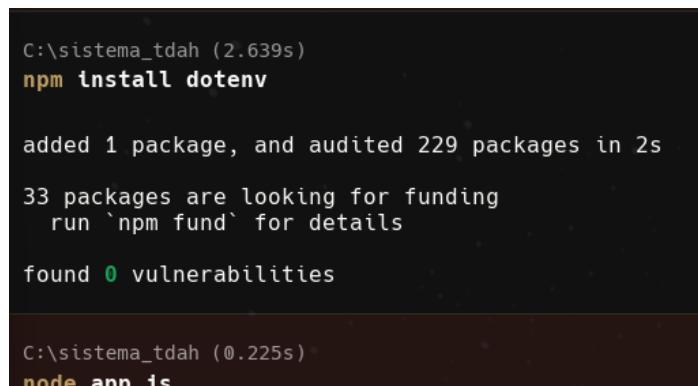


```
C:\sistema_tdah\config\db.js (sistema_tdah) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  sistema_tdah
    config
      db.js
        node_modules
        public
        views
        .env
      app.js
      export_tdah.sql
      package-lock.json
      package.json
      server.js
      sistema_tdah_mysql.sql
  app.js
  .env
  db.js
  index.html
  package.json
  server.js
  sistema_tdah_mysql.sql

app.js | .env | db.js

1  require('dotenv').config();
2  const mysql = require('mysql2');
3
4  const pool = mysql.createPool({
5    host: process.env.DB_HOST,
6    port: process.env.DB_PORT,
7    user: process.env.DB_USER,
8    password: process.env.DB_PASSWORD,
9    database: process.env.DB_NAME,
10   waitForConnections: true,
11   connectionLimit: 10,
12   queueLimit: 0
13 });
14
15 module.exports = pool.promise();
16
```

Por último, se instala “[dotenv](#)”, en el cmd, como muestra en la siguiente imagen:

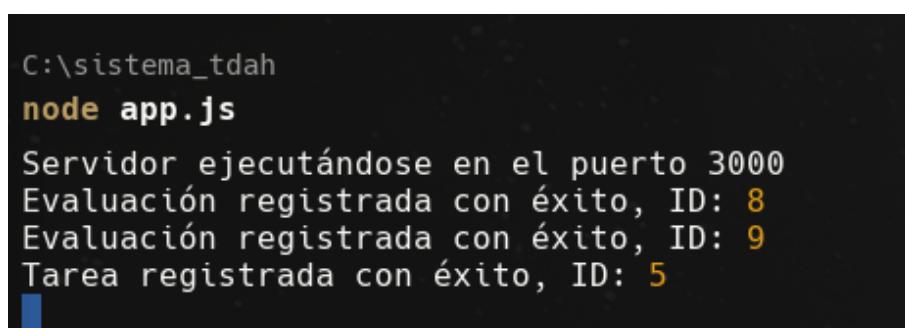


```
C:\sistema_tdah (2.639s)
npm install dotenv

added 1 package, and audited 229 packages in 2s
33 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities

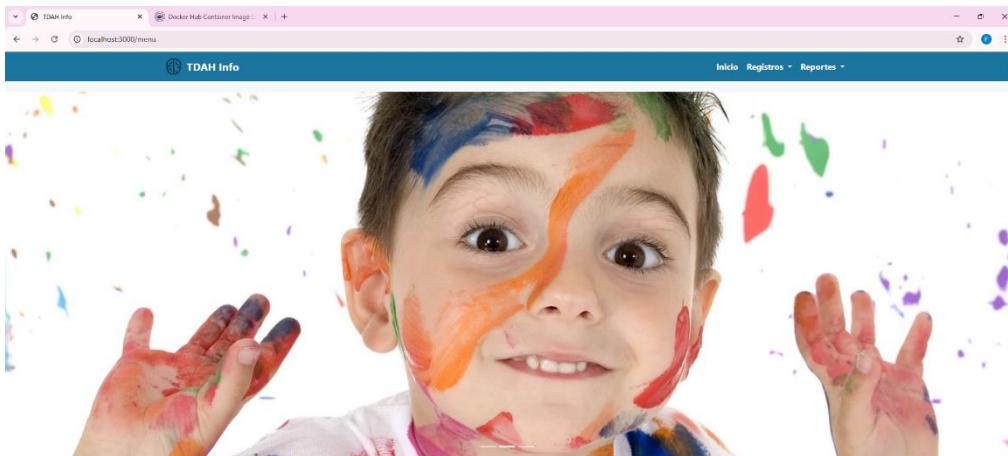
C:\sistema_tdah (0.225s)
node app.js
```

Por último, se ejecuta [node.app.js](#),y le mostrara lo siguiente:



```
C:\sistema_tdah
node app.js

Servidor ejecutándose en el puerto 3000
Evaluación registrada con éxito, ID: 8
Evaluación registrada con éxito, ID: 9
Tarea registrada con éxito, ID: 5
```

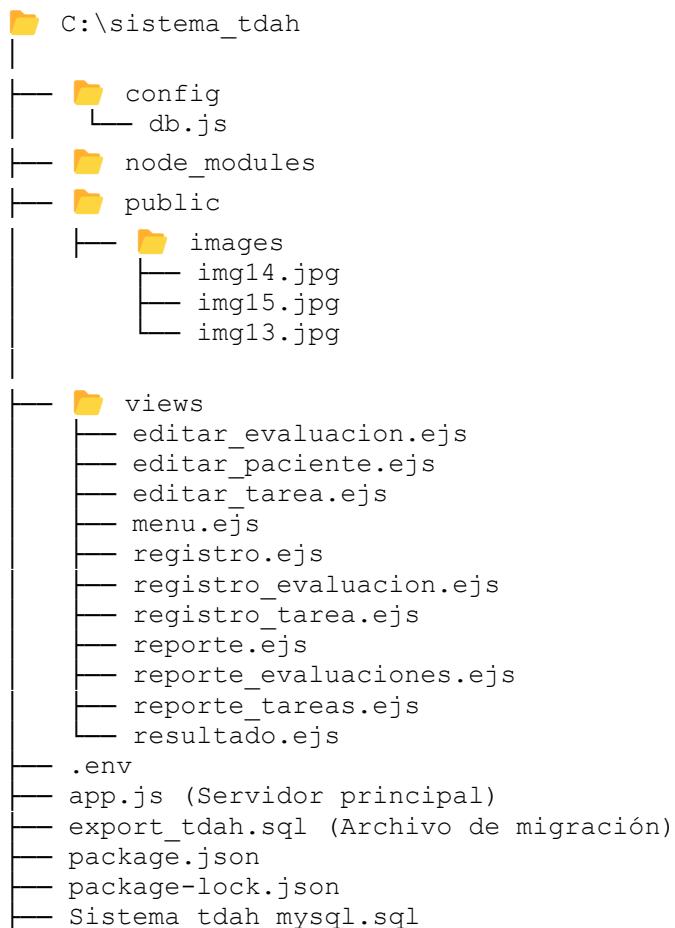


Una Nueva Perspectiva

El TDAH no es una limitación, es una forma diferente de percibir y experimentar el mundo. Cada persona con TDAH tiene talentos únicos y perspectivas valiosas que enriquecen.

Se hizo algunas pruebas, y funciona correctamente solo con otra base de datos, todos los puntos están detallados en el punto 6 “[Funcionamiento del sistema TDAH](#)”.

11. Estructura del proyecto.



12. Implementación del Docker compose.

Como la implementación y cambios del sistema TDAH lo hice con “[Sublime Text](#)”, es excelente para escribir código, pero no está diseñado para crear o administrar contenedores, porque no tiene soporte visual para Docker, por ello usare el “[Visual Studio Code](#)” ya que tiene extensiones especiales al que te permiten ver los contenedores, imágenes, puede levantar, detener o reiniciar los servicios, se puede acceder a la terminal y puedes ejecutar “[Docker-compose up](#)” para levantar el contenedor, tiene el autocompletado sugerencias para “[Docker-compose.yml](#), [Dockerfile](#)”, por esas ventajas, y la enseñanza del docente también fue en “[Visual Studio Code](#)”.

Se crea el archivo “[Docker-compose.yml](#)”, ya que es una herramienta fundamental para definir y ejecutar múltiples contenedores dentro de ese archivo estará lo siguiente como muestra la imagen:

```

version: '3.8'
services:
  backend:
    build: .
    container_name: node_backend
    ports:
      - "3000:3000"
    environment:
      - DB_HOST=mysql
      - DB_PORT=3306
      - DB_USER=root
      - DB_PASSWORD=Feli1234-qb
      - DB_NAME=sistema_tdah
    depends_on:
      - mysql
    volumes:
      - ./app
    networks:
      - red_tdah

  mysql:
    image: mysql:8.0
    container_name: mysql
    restart: always
    ports:
      - "3307:3306"
    environment:
      MYSQL_ROOT_PASSWORD: Feli1234-qb
      MYSQL_DATABASE: sistema_tdah
    volumes:
      - mysql_data:/var/lib/mysql
    networks:
      - red_tdah

volumes:
  mysql_data:
networks:
  red_tdah:

```

También se debe crear el archivo Dockerfile sin extensión, para crear un contenedor con Node.js, que va instalar las dependencias y podrá ejecutar “app.js”, como muestra en la imagen:

```

    sistema_tdah
    └── EXPLORADOR
        ├── EDITORES ABIERTOS
        │   └── Dockerfile
        └── SISTEMA_TDAH
            ├── config
            ├── node_modules
            ├── public
            └── views
                ├── editar_evaluacion.ejs
                ├── editar_paciente.ejs
                ├── editar_tareas.ejs
                ├── menu.ejs
                ├── registro_evaluacion.ejs
                ├── registro_tareas.ejs
                ├── reporte_evaluaciones.ejs
                ├── reporte_tareas.ejs
                ├── reportes.ejs
                └── resultado.ejs
            └── .env
            └── app.js
            └── docker-compose.yml
            └── Dockerfile
    └── Dockerfile
    └── export_tdah.sql
    └── package-lock.json
    └── package.json
    └── server.js
    └── sistema_tdah.mysql.sql

```

```

Dockerfile
FROM node:18
WORKDIR /app
COPY package*.json .
RUN npm install
COPY .
EXPOSE 3000
CMD ["node", "app.js"]

```

Ahora para levantar el contenedor se escribe lo siguiente en la terminal “`docker-compose up --build`”, este comando

Construye el contenedor del backend leyendo lo que está dentro del Dockerfile, luego inicializa MySQL usando la imagen oficial, crea y conecta los contenedores en la red, cuando se instala correctamente significa que el contenedor se construyó exitosamente, como se muestra en la siguiente imagen:

```

    sistema_tdah
    └── EXPLORADOR
        ├── EDITORES ABIERTOS
        │   └── Dockerfile
        └── SISTEMA_TDAH
            ├── config
            ├── node_modules
            ├── public
            └── views
                ├── editar_evaluacion.ejs
                ├── editar_paciente.ejs
                ├── editar_tareas.ejs
                ├── menu.ejs
                ├── registro_evaluacion.ejs
                ├── registro_tareas.ejs
                ├── reporte_evaluaciones.ejs
                ├── reporte_tareas.ejs
                ├── reportes.ejs
                └── resultado.ejs
            └── .env
            └── app.js
            └── docker-compose.yml
            └── Dockerfile
    └── Dockerfile
    └── export_tdah.sql
    └── package-lock.json
    └── package.json
    └── server.js
    └── sistema_tdah.mysql.sql

```

```

version: '3.8'
services:
  backend:
    build: .
    container_name: node_backend
    ports:
      - "3000:3000"
    environment:
      - DB_HOST=mysql
      - DB_PORT=3306
      - DB_USER=root
      - DB_PASSWORD=Feliz1234-qb
      - DB_NAME=sistema_tdah
    depends_on:
      - mysql
    volumes:
      - ./app
    networks:
      - red_tdah

```

```

PS C:\sistema_tdah> docker-compose up --build
-> extracting sha256:3ee69d1a10514e19f12262a4ea5a5ad1d1a0ca7a62108adcfa060208
-> extracting sha256:37927a9091b1b200b07279c6811bf645408208e4a0a373792194050 0.5s
-> extracting sha256:7952f47a4443652b995cc1a195ed249f5976319efdbce159f2963878 2.2s
-> extracting sha256:7291a0231100100000000000000000000000000000000000000000000000000 0.1s
-> extracting sha256:c744f344fdec4c7514437a463f7373553345e5c0808 0.0s
-> extracting sha256:c9b30c3f169665521f0ac0521f0a15501f1fd10ac1c2f0b10350fc28 1.9s
-> extracting sha256:369fb58989d8071df4637e1d13491d00e7bf3a32768c876bd2309b3 0.1s
-> extracting sha256:461077a7fb7e40d34a37d6a195k4d167720d777f572ec50a1f0c41 0.0s
-> [backend] Internal build build context 1.3s
-> [backend] Transferring context: 23.19MB 0.0s
-> [backend] 3/5 COPY package*.json .
-> [backend] 4/5 RUN npm install 0.4s
-> [backend] 5/5 COPY . . 0.7s
-> [backend] Exporting to image 3.1s
-> Exporting layers... 1.1s
-> Exporting manifest sha256:7baf96321910f5a72d57b76d65677908be7a799e92d2d8d389 0.0s
-> Exporting config sha256:45e721d40019a2d7ed1d17b7e73516eaa1c1e19f4e94582 0.0s
-> Exporting attestation manifest sha256:a9e38909a6cbe1f90e62e2bb1ff4842e7ff18d1 0.0s
-> Exporting manifest list sha256:057c52c7ea3c76ef7951a70fa70d1ccf7c7e97899c7 0.0s
-> Naming to docker.io/library/sistema_tdah-backend:latest 0.0s
-> Unpacking to docker.io/library/sistema_tdah-backend:latest 1.1s
-> [backend] Resolving provenance for metadata file 0.0s
[+] Building 1/1
  < backend > Built 0.0s

```

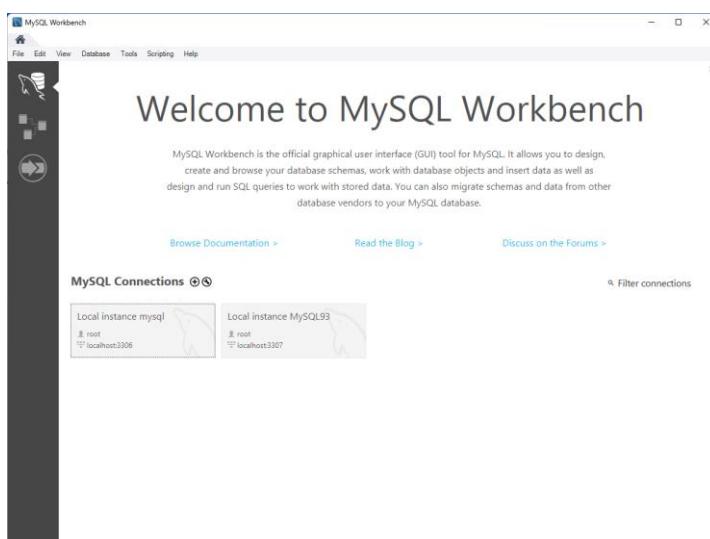
Una vez construido el contenedor, se debe iniciar los contenedores con el siguiente comando “`docker-compose up`”, esto levantara el backend y MySQL, con los cambios realizados en el “`app.js`”, debe mostrar el “`menu.ejs`”, muestra que se esta ejecutando correctamente que se levantó en contenedor, se muestra en la siguiente imagen:

```

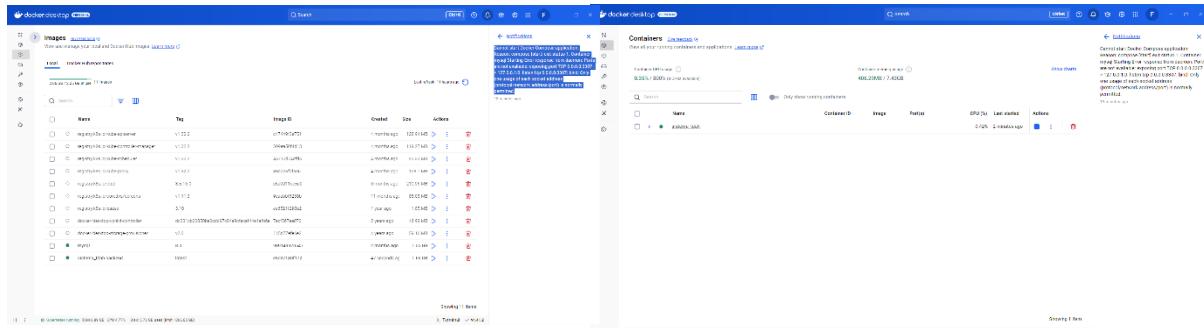
sistema_tdah
$ docker-compose up
Creating network "sistema_tdah_default" with the default driver
Creating service sistema_tdah_mysql
Creating service sistema_tdah_backend
Creating service sistema_tdah_red_tdah
Creating service sistema_tdah_mysql_data
Creating service mysql
Creating service mysql
Creating service backend
Attaching to mysql, node_backend
Gracefully stopping... (press Ctrl+C again to force)
Error response from daemon: Ports are not available: exposing port TCP 0.0.0.0:3307 -> 127.0.0.1:0: listen tcp 0.0.0.0:3307: bind: only one usage of each socket address (protocol/network address/port) is normally permitted.
$ docker ps
CONTAINER ID        IMAGE               COMMAND             STATUS              PORTS

```

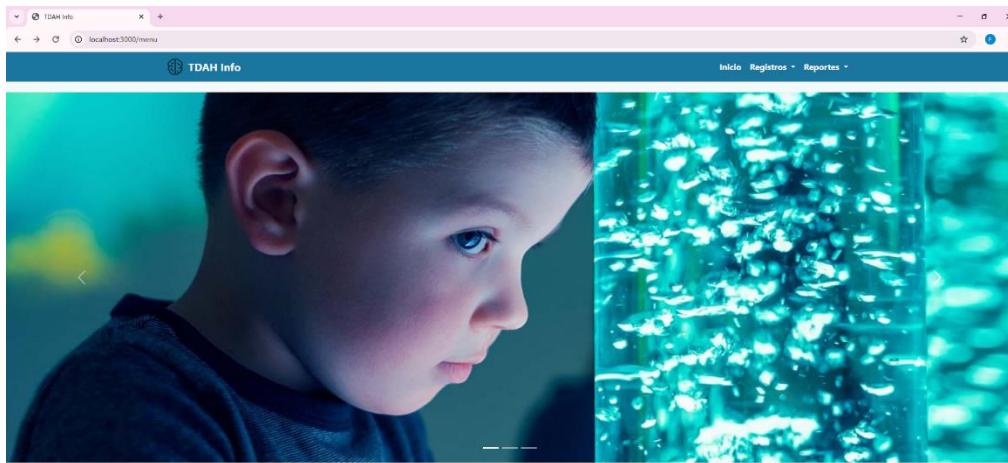
En este caso hay conflicto con los puertos ya que tenia instalado el MySQL anteriormente como muestra la imagen.



No pude solucionar ya que hice pruebas con todo lo aprendido en clase, el contenedor funciona como se muestra en la siguiente imagen:



Tanto el contenedor como las imágenes funcionan correctamente, se va al navegador y ejecuta <http://localhost:3000>, le aparecerá el backend, como muestra la siguiente página.



Una Nueva Perspectiva

El TDAH no es una limitación, es una forma diferente de percibir y experimentar el mundo. Cada persona con TDAH tiene talentos únicos y perspectivas valiosas que enriquecen