8-01-2025

Juan Sebastián Martínez Lesmes

Manual técnico

1. **Lenguajes y Tecnologías Utilizadas**
   1. BackEnd:
      1. Lenguaje: JavaScript
      2. Framework: Express.js
      3. Descripción: Express es un framework web minimalista para Node.js que permite crear aplicaciones web y Apis de manera eficiente. En este proyecto, se utiliza para gestionar las solicitudes del servidor, manejar rutas y conectarse con la base de datos.
   2. FrontEnd:
      1. Lenguaje: JavaScript
      2. Framework/Librería: React.js
      3. Descripción: React.js es una biblioteca para construir interfaces de usuario dinámicas y reutilizables. En este proyecto, se utiliza para crear la parte visual de la aplicación, manejando componentes y el estado de la interfaz de usuario.
   3. Base de Datos
      1. Tecnología: MongoDB
      2. Herramienta de Gestión: MongoDB Compass
      3. Descripción: MongoDB es una base de datos NoSQL orientada a documentos, ideal para manejar datos no estructurados o semiestructurados. MongoDB Compass es una herramienta gráfica utilizada para gestionar y visualizar los datos de manera intuitiva.
   4. Control de Versiones (VC)
      1. Sistema Operativo: Windows
      2. Herramientas: Git y GitHub
      3. Descripción: Git es un sistema de control de versiones distribuido utilizado para rastrear cambios en el código fuente. GitHub es una plataforma basada en la nube para alojar repositorios de Git y colaborar en el desarrollo de software.
2. **Instalación y Configuración**
   1. Requisitos Previos
3. Node.js y npm: Descargar e instalar desde <https://nodejs.org/>.
4. Git: Descargar e instalar desde <https://git-scm.com/>.
5. MongoDB: Descargar e instalar desde <https://www.mongodb.com/try/download/community>.
6. MongoDB Compass: Descargar e instalar desde <https://www.mongodb.com/products/compass>.
7. Navegador Web: Chrome o Edle.

* **Clonación del Proyecto**

1. Abrir la terminal y ejecutar:   
   backend: git clone <https://github.com/SebastianMartinezLesmes/back_ToDoList.git>  
     
   fronted: git clone <https://github.com/SebastianMartinezLesmes/front_ToDoList.git>

* **Configuración del Servidor BackEnd**

1. Navegar al directorio del BackEnd:

cd back\_ToDoList

1. Instalar dependencias:

npm install

1. Configurar variables de entorno (archivo .env):

PORT=5000

MONGO\_URI= “mongodb://localhost:27017”

1. Iniciar el servidor:

npm run serve

* **Configuración del FrontEnd**
  1. Navegar al directorio del FrontEnd:

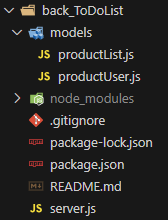
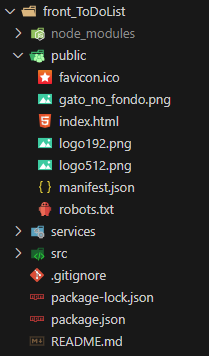
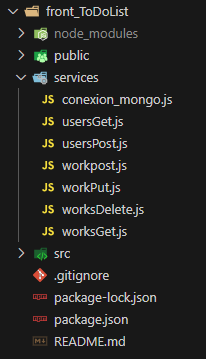
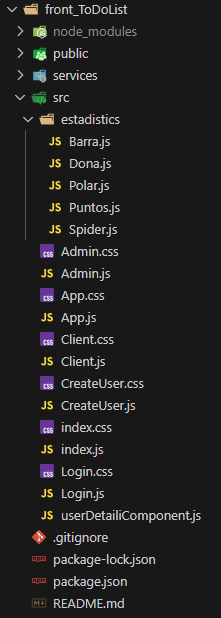
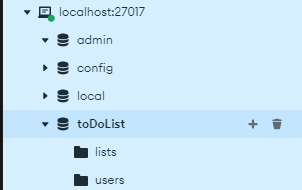
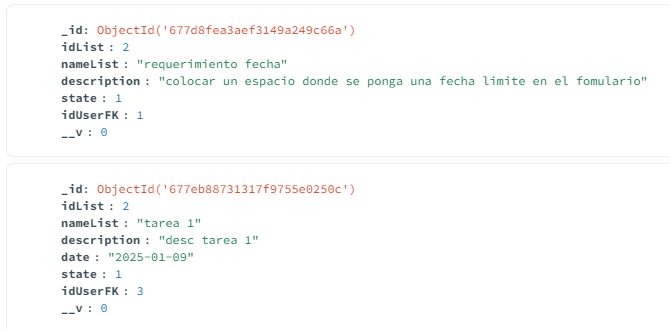
cd front\_ToDoList

* 1. Instalar dependencias:

npm install

* 1. Iniciar la aplicación (Port = 3000):

npm start

1. **Estructura del proyecto** 
   1. **BackEnd  
      **
   2. **FontEnd**   
      ****  
   3. **Data Base:  
        
      Modelo lists:  
        
      Modelo users:   
      **
2. **Ejecución y pruebas**
   1. **BackEnd:** por medio de Thunder Client se probaron los endPoints
      1. **/getUser:** retornacorrectamente datos en JSON
      2. **/getList:** retornacorrectamente datos en JSON
      3. **/createUser:** crea un usuario (ROL: client)
      4. **/createList:** crea una tarea con los campos requeridos
      5. **/deleteList:** borra una tarea de la lista
      6. **/updateList/:id:** cambia el estado de una tarea (pendiente 🡪 completada)
      7. **/updateListdata:** camia la información de una tarea
      8. **/mongoose.connect(‘mongodb://localhost:27017’):** conecta correctamente con la base de datos corre el BackEnd en el puerto 5000
   2. **FrontEnd:** 
      1. **Login:**
         1. Manipulación de datos faltantes
         2. consulta y comparación con la DB
      2. **Crear Usuario:**
         1. Manipulación de datos
         2. Campo email como un email (xxxAxx.com)
         3. Campo contraseña creación de contraseña segura y parámetros indicativos
      3. **Direccionamiento de páginas:** 
         1. Direccionamiento a otras páginas
         2. Protección de las rutas de navegación por medio de verificación de usuario
      4. **Función toma de datos usuarios:**
         1. Por medio de consultas http con fetch y axios se realizaron métodos de extracción de datos (Get) para los usuarios
      5. **Función toma de datos tareas:**
         1. Por medio de consultas http con fetch y axios se realizaron métodos de extracción de datos (Get) para las tareas
      6. **Cambio de estado:**
         1. Por medio de peticiones (put) http con fetch y axios se alteraro el estado de las tareas (Pendiente 🡪 completada)
      7. **Edición de información:**
         1. Por medio de peticiones (put) http con fetch y axios se alteraro la información de las tareas (nameList, description, date)
      8. **Eliminación de tareas:**
         1. Por medio de peticiones (delete) http con fetch y axios se elimino la tarea de la base de datos

* + 1. **Filtros:**
       1. En la intrfas del administrador se crearon filtros para que en la sección de usuario se encontraran a los usuarios por medio del nombre, estado y su rol en el sistema.
       2. En la interfas de administrador se crearon filtros para que en la sección de usuario mostrara a los usuarios dependiendo de su nombre, tareas completadas, tareas pendientes
    2. **Graficas:** 
       1. Se crearon graficas para el análisis del estado de los usuarios y de su cumplimiento de tareas
       2. Se crearon diferentes graficas para facilitar el análisis por medio de diferentes perspectivas

1. **Problemáticas y soluciones:**
   1. Conexión con la base de datos
      1. Tener instalado y configurado mongoDB compass
      2. Error de ejecución: <https://youtu.be/y1l1UI_UdQI?si=9VNunqqqQlAAUez9>
      3. Llamar correctamente los campos de la base de datos (No Relacional)
   2. Ejecución del BackEnd
      1. Tener instaladas las dependencias
      2. Correr la base de datos antes de ejecutar para una correcta conexión
   3. Ejecución del FrontEnd:
      1. Tener instalada node.js
      2. Tener instaladas las dependencias
      3. Ejecutar el BackEnd antes de correr el FrontEnd para correcta comunicación de los endPoints