

INTEGRANTES

DAVID ORLANDO HERNANDEZ MURILLO

JOSUE REYES NATANAEL CAMARGO

JUAN CARLOS MAESTRE HERRERA

DOCENTE

JUAN ALEJANDRO CARRILLO JAIMES

BASES DE DATOS

MANUAL TECNICO

CONTENIDO

1. Objetivos ……………………………………………………………………………….

1.1. Objetivos Especificos ………………………………………………………..

2. Alcance ………………………………………………………………………………….

3. Requirimientos tecnicos ……………………………………………………………….

3.1. Requirimientos minimos de hardware………………………………………

3.2. Requirimientos minimos de software……………………………………….

4. Herramientas utilizadas para el desarrollo .….………………………………………

5. Instalacion ……………………………………………………………………………….

6. Configuracion ………………………………………………………………………...…

7. Analisis …………………………………………………………………………………..

8. Diseño de la arquitectura ………………………………………………………………

9. Desarrollo ………………………………………………………………………………..

10. Privilegios ………………………………………………………………………………

**1. OBJETIVOS.**

Proporcionar a los desarrolladores y administradores de la aplicación una guía detallada sobre la implementación, configuración y mantenimiento del sistema CRUD de estudiantes.

En general esto se diseno para guiar a un programador que este a cargo de este sistema.

1.1. OBJETIVOS ESPECIFICOS.

* Descripción de la estructura del código y base de datos.
* Evidencias de diseno de sistemas (Diagramas)
* Instrucciones para la configuración del servidor y despliegue.
* Procedimientos para solucionar problemas técnicos.
* Indicaciones para realizar actualizaciones o personalizaciones futuras.

**2. ALCANCE**

Este documento esta dirigido a personas que tengas un conocimineto previo en bases de datos y programacion por lo que no se incluye capacitación en lenguajes de programación específicos (se asume conocimiento previo).

**3. REQUERIMIENTOS TECNICOS.**

3.1. Requerimiento minimo de hardware

* Procesador: Dual-core 2.0 GHz o superior.
* Memoria RAM: 4 GB o más.
* Almacenamiento: 20 GB de espacio libre.
* Conexión a Internet estable.
* Dispositivo: Computadora o móvil con navegador web.
* Memoria RAM: 2 GB o más.
* Navegador: Google Chrome, Mozilla Firefox

3.2. Requerimientos minimos de software

* Sistema operativo: Windows, Linux, o macOS (cualquier versión compatible con el entorno web).
* Base de datos: Postgres
* Entorno de desarrollo: Node.js, Angular, Visual Studio, .Net

**4. HERRAMIENTAS PARA EL DESARROLLO**

* Editor de código: Visual Studio.
* Servidor de desarrollo: Node.js
* Base de datos: PostgreSQL.
* Gestión de dependencias: npm (Node Package Manager)
* Frameworks/Librerías:
  + Para el backend: (.net).
  + Para el frontend: (angular).
* Control de versiones: Git (con GitHub).
* Navegador para pruebas: Google Chrome o Firefox.
* Cliente para base de datos: pgAdmin (para gestionar PostgreSQL).

5. INSTALACION

Antes de iniciar la instalación, se tiene que contar con lo siguientes requisitos previos.

PostgreSQL instalado en tu equipo local.

pgAdmin configurado para gestionar la base de datos.

Un editor de código como Visual Studio

Un navegador web actualizado (Google Chrome, Firefox, etc.).

(Opcional) Git instalado para gestionar el código fuente.

PASO 1. Preparar el proyecto.

Tenemos en cuenta donde esta el nombre del proyecto.



PASO 2. CONFIGURAR LA BASE DE DATOS

* Abre PostgreSQL y crea una nueva base de datos (ejemplo: escuelaVault).
* Usa pgAdmin para importar o ejecutar el script SQL incluido en el proyecto para crear las tablas necesarias.
* Configura las credenciales de acceso en el archivo de configuración del proyecto (ejemplo: .env o config.js), incluyendo:
  + Nombre de la base de datos.
  + Usuario.
  + Contraseña.
  + Host (generalmente localhost).

PASO 3. INSTALAR DEPENDENCIAS DEL PROYECTO.

Abre una terminal en la carpeta del proyecto.



PASO 4. CONFIGURAR EL SERVIDOR WEB

* Asegúrate de que el servidor local esté configurado y funcionando correctamente.

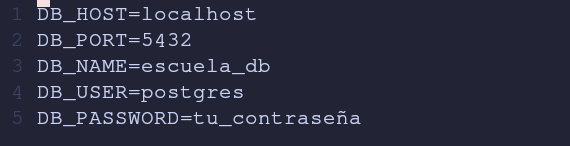
PASO 5. ACCEDER A LA APLICACION

* Abre un navegador web.
* Ingresa la dirección del servidor local (ejemplo: http://localhost:3000 o el puerto configurado).
* Verifica que la página de inicio carga correctamente y realiza pruebas básicas.

**6. CONFIGURACION.**

La configuración de este sistema es sencilla, ya que está diseñada para cumplir con los requisitos de un proyecto de semestre. Sin embargo si quieres modifcar el localhost y cosas basicas de la bases de datos, puedes hacer lo sisguiente.

Edita los siguientes valores para reflejar tu configuración personalizada:



Si cambias el puerto o el host, asegúrate de ingresar la nueva dirección en tu navegador:



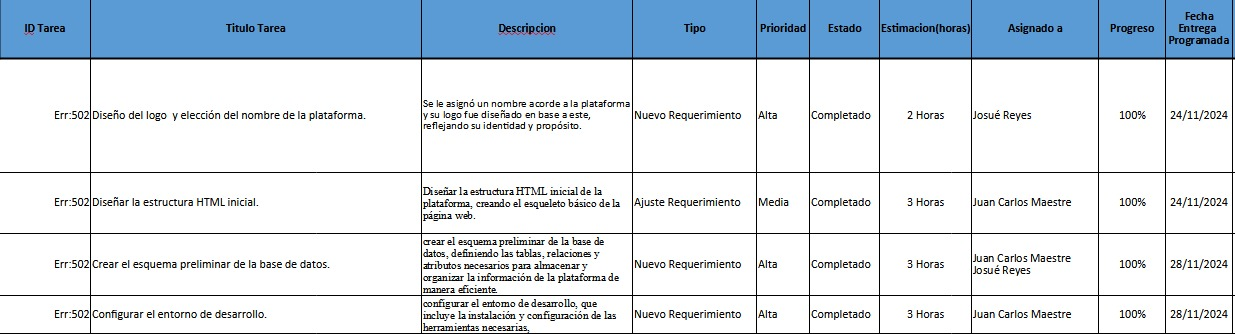
**7. ANALISIS**

El objetivo principal del sistema es facilitar la gestión de estudiantes en una escuela mediante una aplicación web que permita realizar las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) de manera sencilla y eficiente.

Requerimientos Funcionales

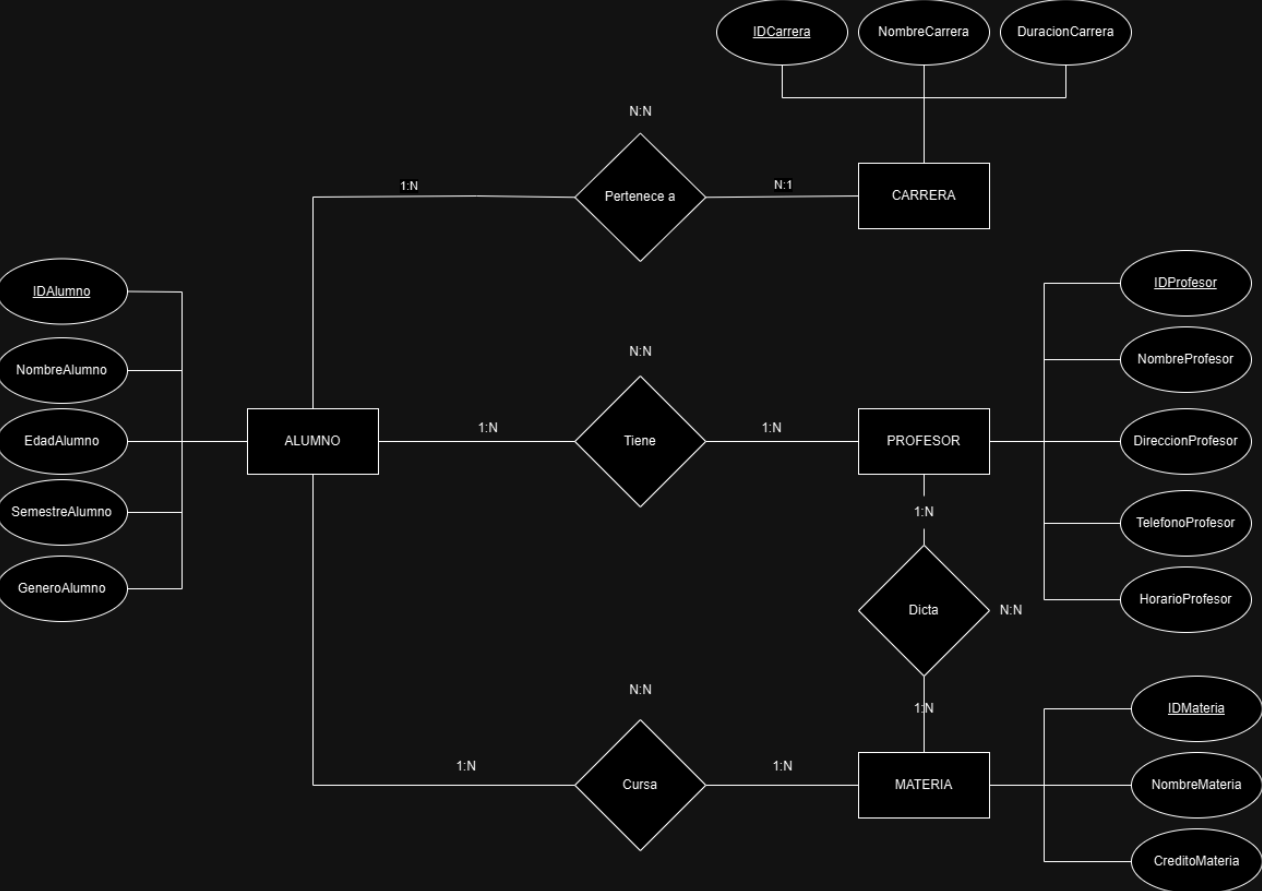
* Permitir la creación de nuevos registros de estudiantes.
* Visualizar la lista de estudiantes registrados.
* Actualizar la información de un estudiante existente.
* Eliminar registros de estudiantes de la base de datos.

Requerimientos No Funcionales

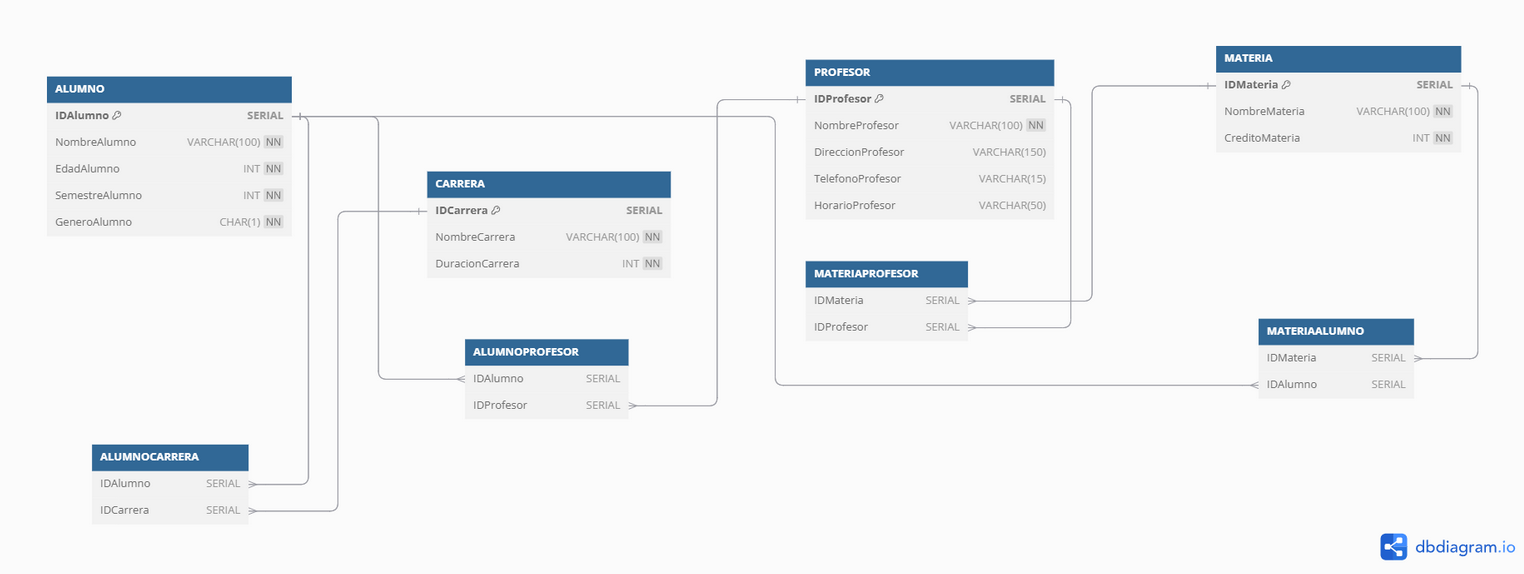
* Facilidad de uso: La interfaz debe ser intuitiva y accesible para usuarios con conocimientos básicos de computación.
* Portabilidad: El sistema debe ejecutarse en cualquier navegador moderno.
* Rendimiento: Las operaciones CRUD deben ejecutarse rápidamente con bases de datos de tamaño moderado.
* 

**8. DISEÑO DE LA ARQUITECTURA**

Modelo entidad-relacion.



Formalizacion.



**9. DESARROLLO**

**Diccionario de datos:**

**https://github.com/Natanael-28/SchoolVault/blob/main/DOCS/diccionario\_datos\_schoolVault.xlsx**

**Desarrollo Escuela**

**Codigogithub maestre**

**10. PRIVILEGIOS**

Todos los usuarios.