**Manual Técnico**

1. **Roles / integrantes**

**Edwin Heredia Saravia (Team Leader)**

**Luis Mario Garcia Chambilla (Developer and Git Master)**

**Shanty Nicole Guthrie Saavedra (Dba)**

1. **Introducción:**

**El proyecto desarrollado para las sedes de Cochabamba en el programa de tuberculosis busca mejorar la gestión de datos de pacientes y personal de salud, fortaleciendo el seguimiento de tratamientos y la toma de decisiones mediante un sistema eficiente y accesible.**

1. **Descripción del proyecto:**

Este proyecto consiste en un sistema digital diseñado para gestionar de manera integral los datos relacionados con la tuberculosis en las sedes de Cochabamba. La plataforma permite:

1. **Gestión de Pacientes**: Registro, actualización y seguimiento de tratamientos de tuberculosis, incluyendo información médica.
2. **Control de Personal de Salud**: Monitoreo de roles, asignaciones y disponibilidad del personal que participa en el programa.
3. **Optimización de Sedes**: Coordinación entre las diferentes sedes para el intercambio y análisis de datos.
4. **Accesibilidad y Seguridad**: Implementación de un sistema seguro, accesible desde múltiples dispositivos, garantizando la privacidad y confidencialidad de los datos.
5. **Link al Video demostrativo YouTube (5 minutos máximo)**
6. **Listado de los Requisitos Funcionales del Sistema**

Gestión de Pacientes.

Gestión de Personal de Salud.

Gestión de Sedes.

Seguimiento de Tratamientos.

Accesibilidad por Niveles de Usuario.

Seguridad del Sistema.

1. **Arquitectura del software:**

La arquitectura del software desarrollada para el programa de tuberculosis en las sedes de Cochabamba está diseñada bajo una **arquitectura multicapa** que promueve la modularidad, mantenibilidad y escalabilidad del sistema.

* Componentes Principales y Organización

Capa de Presentación (Frontend)

Capa de Lógica de Negocio (Backend)

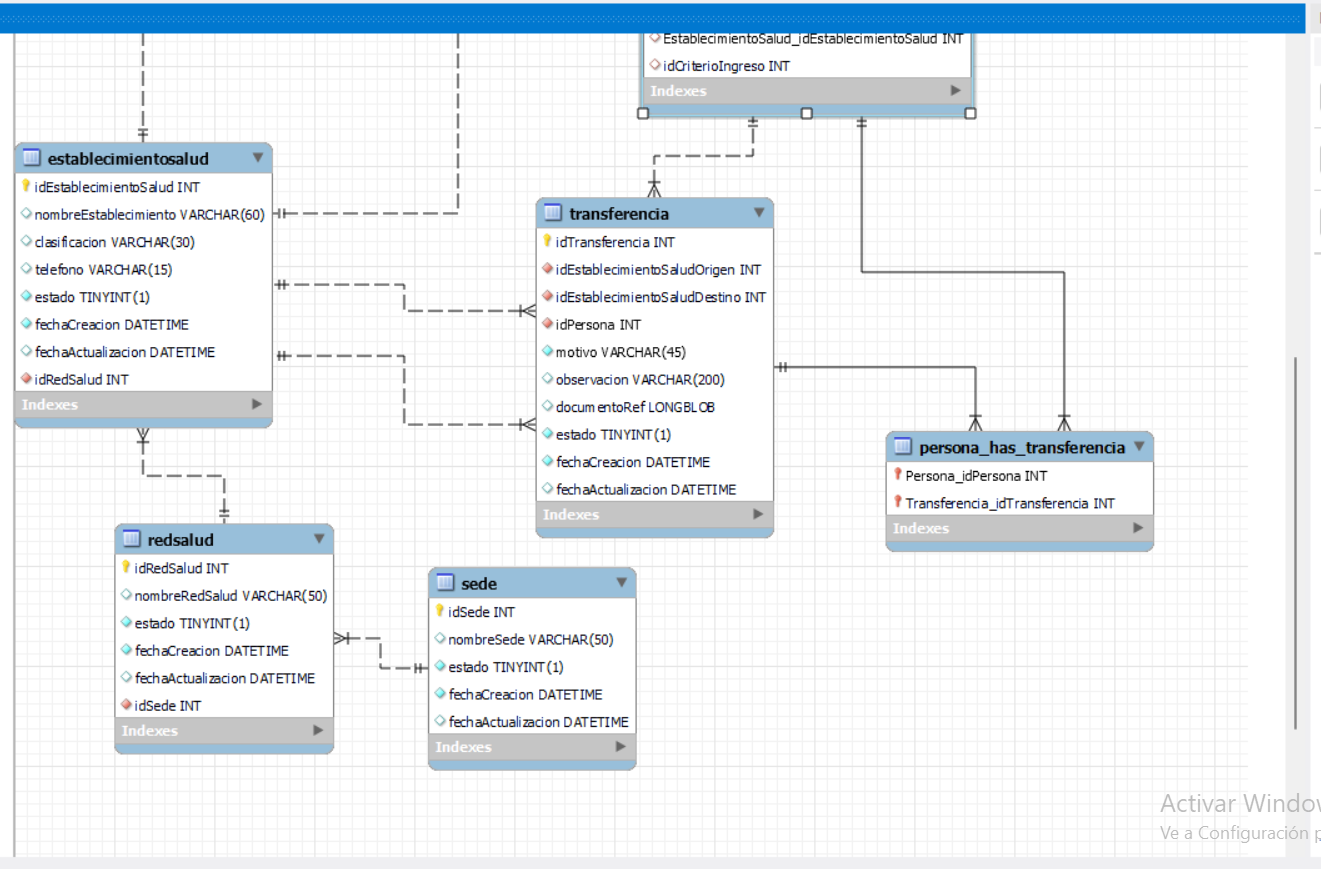
Capa de Datos (Base de Datos)

* Interacciones entre los Componentes

1. **Base de datos**
   1. **Diagrama completo y actual**

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente



* 1. **En el GIT una carpeta con la base de datos con script de generación e inserción de datos de ejemplo utilizados**
  2. **Script simple (copiado y pegado en este documento)**

1. **Listado de Roles más sus credenciales de todos los Admin / Users del sistema**
2. **Requisitos del sistema:**

* Requerimientos de Hardware (mínimo): (cliente)

Procesador: Dual-Core de 2.0 GHz o superior.

Memoria RAM: Mínimo 4 GB.

Espacio en Disco: 500 MB de espacio libre para archivos temporales del navegador y almacenamiento de datos en caché.

Pantalla: Resolución mínima de 1366x768 píxeles.

Conexión a Internet: Velocidad mínima de 2 Mbps para la conexión con el servidor.

Navegador Compatible: Google Chrome, Mozilla Firefox, o Microsoft Edge (versiones actualizadas).

* Requerimientos de Software: (cliente)

PR-TUBERCULOSISWEB

Version minima de NodeJS (LTS) v20.18.0

Version minima de npm v10.8.2

* Requerimientos de Hardware (server/ hosting/BD)

Procesador: 8 núcleos (CPU x64 de 3.0 GHz o superior).

Memoria RAM: Mínimo: 8 GB. o 16 GB.

Espacio en Disco: 50 GB de almacenamiento SSD o 100 GB.

Conexión de Red: 10 Mbps (subida y bajada), recomendado 100 Mbps o superior.

* Requerimientos de Software (server/ hosting/BD)

Procesador: Mínimo: 4 núcleos (CPU x64 de 2.0 GHz). Recomendado: 8 núcleos (CPU x64 de 2.5 GHz o superior).

Memoria RAM: Mínimo: 8 GB para bases de datos pequeñas o medianas. Recomendado: 16 GB.

Espacio en Disco: Mínimo: 100 GB SSD. Recomendado: 200 GB.

Conexión de Red Interna: Mínimo: 1 Gbps para comunicación entre la API y la base de datos

Gestor de Base de Datos: Recomendado: MySQL 8.0

1. **Instalación y configuración:**

*PR-TUBERCULOSISWEB*

*Version minima de NodeJS (LTS) v20.18.0*

*Version minima de npm v10.8.2*

*DIRECTORIO CLIENT*

*cd client Cambiar de directorio*

*npm install Paquetes de dependencia en el directorio Client*

*npm start Para ejecutar el programa en el navegador web*

*DIRECTORIO SERVER*

*PASO 1: Verificar la conexion en la base de datos*

*PASO 2: Verificar la contraseña del servidor*

*PASO 3: node Conection.js Ejecutar el servidor*

1. **GIT :**

* Versión final entregada del proyecto.
* Entrega compilados ejecutables

1. **Personalización y configuración:** El sistema desarrollado para el programa de tuberculosis en las sedes de Cochabamba cuenta con opciones flexibles de personalización y configuración que permiten adaptarlo a las necesidades específicas de los usuarios y del contexto operacional.

Opciones de Configuración General:

Roles y Permisos.

Unidades de Salud Asociadas.

Configuración de Parámetros del Sistema:

Criterios de Filtrado.

Configuración de Variables Clave:

Base de Datos.

Endpoints de la API.

1. **Seguridad**:
   1. Permisos de Acceso y Roles:

Gestión de Sesiones.

* 1. Autenticación y Autorización:

Autenticación Segura.

Autorización.

Capacitación del Personal.

1. **Depuración y solución de problemas:**

Herramientas de Diagnóstico y Depuración:

Consola de Desarrollo.

Logs del Servidor.

Fallo en la autenticación. Credenciales incorrectas o token expirado. - Verifique las credenciales.

- Intente cerrar sesión y volver a iniciar sesión.

#### **Glosario de términos:** **A**

* **API (Application Programming Interface):**  
  Conjunto de definiciones y protocolos que permite la comunicación entre diferentes aplicaciones o componentes de software.
* **Autenticación:**  
  Proceso mediante el cual se verifica la identidad de un usuario, generalmente mediante credenciales como nombre de usuario y contraseña.
* **Autorización:**  
  Proceso que determina los permisos o accesos que tiene un usuario autenticado dentro del sistema.

#### **B**

* **Base de Datos (BD):**  
  Conjunto organizado de datos que se almacena y gestiona mediante un sistema de gestión de bases de datos (DBMS).
* **Backend:**  
  Parte del software que opera en el servidor, encargada de manejar la lógica de negocio, la comunicación con la base de datos y las interacciones con el frontend.

#### **C**

* **Cliente:**  
  Dispositivo o aplicación que interactúa con el servidor para solicitar datos o servicios.
* **CRUD:**  
  Acrónimo de **Crear, Leer, Actualizar y Eliminar**, las operaciones básicas que se pueden realizar en una base de datos.

#### **D**

* **Depuración:**  
  Proceso de identificación y corrección de errores o defectos en el software.
* **Despliegue:**  
  Proceso de publicación de una aplicación o sistema en un entorno donde pueda ser utilizado por los usuarios finales.

#### **E**

* **Endpoint:**  
  Punto de entrada de una API donde se reciben solicitudes y se procesan las respuestas.
* **Estado:**  
  Condición o situación actual de un componente o recurso en el sistema. Por ejemplo, un usuario puede tener el estado "activo" o "inactivo".

#### **F**

* **Frontend:**  
  Parte del software que interactúa directamente con los usuarios, generalmente la interfaz gráfica o visual.

#### **H**

* **Hosting:**  
  Servicio que permite alojar aplicaciones o sitios web en servidores accesibles desde Internet.

#### **I**

* **Interfaz de Usuario (UI):**  
  Medio por el cual el usuario interactúa con un sistema, generalmente compuesto por elementos visuales como botones, formularios y menús.

#### **L**

* **Logs:**  
  Registros que contienen información detallada sobre las actividades realizadas por un sistema, útiles para depuración y auditorías.

#### **P**

* **Parámetro:**  
  Valor que se pasa a una función o API para personalizar su comportamiento.
* **Permiso:**  
  Configuración que define lo que un usuario o rol puede o no puede hacer en el sistema.

#### **R**

* **Respaldo (Backup):**  
  Copia de seguridad de los datos o configuraciones del sistema para garantizar su recuperación en caso de pérdida o falla.
* **Responsividad:**  
  Capacidad de una interfaz de usuario para adaptarse a diferentes tamaños y resoluciones de pantalla.

#### **S**

* **Servidor:**  
  Computadora o programa que proporciona servicios o datos a otros dispositivos denominados clientes.
* **SQL (Structured Query Language):**  
  Lenguaje estándar para gestionar y manipular bases de datos relacionales.

#### **T**

* **Token:**  
  Código único utilizado para identificar y autenticar a un usuario o aplicación en un sistema.

#### **U**

* **Usuario Final:**  
  Persona que utiliza directamente el sistema para llevar a cabo tareas específicas.

#### **V**

* **Variable:**  
  Espacio en la memoria del sistema que almacena un valor que puede cambiar durante la ejecución del programa.

1. **Referencias y recursos adicionales:**

Documentación de Node.js

Página oficial de Node.js con información detallada sobre su instalación, uso y características.

https://nodejs.org/es/docs/

Documentación de React.js

Guía oficial sobre React.js, una librería para construir interfaces de usuario.

https://reactjs.org/docs/getting-started.html

Documentación de Axios

Información sobre Axios, la librería de JavaScript utilizada para realizar peticiones HTTP.

https://axios-http.com/docs/intro

Documentación de SQL (MySQL / PostgreSQL)

Guías y tutoriales para aprender a usar SQL y manejar bases de datos.

MySQL Documentation

PostgreSQL Documentation

Documentación de Express.js

Documentación para aprender sobre Express, un framework de Node.js para desarrollar aplicaciones web y APIs.

https://expressjs.com/

1. **Herramientas de Implementación:**

* Lenguajes de programación:

JavaScript (ES6+)

HTML5.

SQL (MySQL).

* Frameworks:

React.js

Node.js

Express.js

Bootstrap 5

* APIs de terceros, etc:

Axios

JWT (JSON Web Token)

Cloud Storage (Ej. Amazon S3 o Google Cloud Storage).

* Otras Herramientas

Git/GitHub

Postman

1. **Bibliografía**

 **MDN Web Docs (Mozilla)** <https://developer.mozilla.org/>

 **React Documentation** https://reactjs.org/docs/getting-started.html

 **Node.js Documentation** https://nodejs.org/en/docs/

 **Express.js Documentation** <https://expressjs.com/>

 **Axios Documentation** <https://axios-http.com/>

 **JWT.io** <https://jwt.io/>

 **Bootstrap Documentation** https://getbootstrap.com/docs/5.0/getting-started/introduction/

 **Postman Documentation** https://www.postman.com/docs/

 **MySQL Documentation** <https://dev.mysql.com/doc/>

 **GitHub Docs** <https://docs.github.com/en>

 **"Pro React" por Cassio de Sousa Antonio** https://www.apress.com/gp/book/9781484244509

 **"Node.js Design Patterns" por Mario Casciaro** https://www.packtpub.com/product/nodejs-design-patterns-third-edition/9781800563487