**Manual Técnico**

1. **Roles / integrantes**

Bricely Clariss Gutierrez Cadima – Team Leader – Desarrollo de la app Web – LARAVEL 12

Anett Azucena Garcia Sarzuri – Desarrollo de la app Móvil - FLUTTER

1. **Introducción:**

SEDES-TUBERCULOSIS-II es un proyecto enfocado en las transferencias de los pacientes que se cambiaron de un hospital a otro, ya sea de provincia de Chapare a Tiquipaya dentro de Cochabamba. Además, se busca hacer un seguimiento de las dosis que un paciente debe realizar. Por lo que se ha solicitado enviar videos grabados desde dispositivos móviles y que sean visualizados en la lista de pacientes. Eso como prueba tangible de que esta siguiendo su medicación correspondiente.

1. **Descripción del proyecto:**

Es importante recalcar que este proyecto consiste en una API BACKEND que consuma la aplicación WEB y MOVIL. La aplicación web es para la parte administrativa donde existe 3 roles: 1 Administrador, médicos y enfermeros.

El sistema se centra en registrar las transferencias de pacientesque se realiza entre hospitales(establecimientos).Para ello, se necesita poder realizar gestiones de las entidades establecimientos que son los hospitales y pacientes aparte.

Es importante que los médicos puedan visualizar los videos de los pacientes grabándose a sí mismos tomándose la medicación que sería como una prueba tangible de su medicación.

1. **Link al Video demostrativo YouTube (5 minutos máximo)**

**APP MOVIL FLUTTER:**

[**APP MÓVIL - FLUTTER.mp4**](https://univalleedu.sharepoint.com/:v:/s/PSII-1-SEDESTUBERCULOSISII/EYx5oc4HaFpNtsmY4B5_odcBVE5NPBTGMl4LVdjoUc5dRw?e=UVoztr)

**[WEB-LARAVEL.mp4](https://univalleedu.sharepoint.com/:v:/s/PSII-1-SEDESTUBERCULOSISII/ETphwYH1CjlNlVeizCUNEJ4BbDm88C84BJbqyvn-NVekCQ?e=4NJejy)**

1. **Listado de los Requisitos Funcionales del Sistema**

* *Registro de Usuario: El sistema debe permitir que el administrador registre empleados y pacientes. Mientras que, los enfermeros puedan registrar sólo pacientes. El manejo de credenciales para los empleados será mediante un nombre de usuario que haga referencia al establecimiento que corresponde*
* *Inicio de Sesión: Los usuarios deben poder iniciar sesión dependiendo de su rol. En caso de ser doctor o enfermero ingresaran mediante un nombre de usuario generado y una contraseña generada de 7 números.*
* *Transferencias de pacientes: Esto se realizará de establecimiento a establecimiento realizado por un usuario Doctor generalmente.*
* *Gestión de pacientes: Los pacientes serán registrados por doctores y enfermeros donde su nombre será el único dato de acceso a la app móvil para subir el video de la dosis que le toca.*
* *Generación de establecimientos y empleados: El Administrador podrá gestionar los establecimientos donde automáticamente genera 2 usuarios (médico y enfermero) por establecimiento.*
* *Subir videos como prueba tangible de cumplimiento de dosis: El paciente mediante la app móvil subirá las dosis que se han sido asignado por el establecimiento que corresponde y una vez completadas deberá deshabilitar su acceso a la app móvil.*

1. **Arquitectura del software:** *Explicación de la estructura y organización del software, incluyendo los componentes principales, las interacciones entre ellos y los patrones de diseño utilizados.*

6.1. Capas del Sistema

* Capa de Presentación (Frontend):
  + Esta capa se encarga de la interacción con el usuario final.
  + Será implementada como una aplicación web responsiva utilizando tecnologías como HTML5, CSS3, JavaScript y frameworks como Laravel.
  + Esta capa se comunica con el backend a través de servicios RESTful (API).
* Capa de Lógica de Negocio (Backend):
  + Aquí se implementan todas las reglas del negocio.
  + Expondrá servicios a través de una API RESTful, utilizando frameworks como Laravel 12.
  + Procesa solicitudes del frontend y se comunica con la base de datos.
* Capa de Datos (Base de Datos):
  + Responsable del almacenamiento persistente de la información.
  + Se utilizará una base de datos relacional MySQL.
  + La comunicación con esta capa será gestionada por el backend mediante ORM (Object Relational Mapping) como Sequelize, Entity Framework, o acceso directo por consultas SQL.

6.2. Componentes Principales

* Módulo de Usuarios: gestiona autenticación, roles y permisos.
* Módulo de Proyecciones: permite la generación de predicciones de ventas con base en datos históricos.
* Módulo de Reportes: genera gráficos e informes estadísticos.
* Módulo de Configuración: permite ajustar variables del sistema y parámetros de predicción.

6.3. Interacciones Entre Componentes

* El usuario accede al sistema mediante una interfaz web.
* El frontend envía solicitudes al backend a través de la API REST.
* El backend procesa la solicitud, aplica la lógica de negocio y consulta o actualiza datos en la base de datos.
* La respuesta es devuelta al frontend, que la presenta de forma amigable al usuario.

6.4. Patrones de Diseño Utilizados

* MVC (Modelo-Vista-Controlador): Este patron de diseño arquitectónico separa la presentación, la lógica del negocio y el acceso a los datos para facilitar el mantenimiento.
* Repositorio: abstrae el acceso a la base de datos para separar la lógica de negocio del acceso a datos.
* Inyección de Dependencias: permite un código más limpio y fácilmente testeable.

1. **Base de datos**
   1. **Diagrama completo y actual**
   2. **En el GIT una carpeta con la base de datos con script de generación e inserción de datos de ejemplo utilizados**
   3. **Script simple (copiado y pegado en este documento)**
2. **Listado de Roles más sus credenciales de todos los Admin / Users del sistema**

ADMINISTRADOR. – Tendrá acceso a toda la aplicación web.

MÉDICO. – Tendrá acceso a todo menos creación de establecimientos.

ENFERMERO. – Tendrá acceso sólo a la gestión de pacientes.

PACIENTE. – Tendrá acceso a la aplicación móvil.

1. **Requisitos del sistema:**

* Requerimientos de Hardware (mínimo): (cliente)

**Requerimientos de Hardware (mínimo): (cliente)**

Estos requisitos aplican a los equipos donde se usará la aplicación web y móvil:

* **Procesador:** Intel Core i3 o superior
* **Memoria RAM:** 4 GB mínimo
* **Almacenamiento:** 10 GB de espacio libre (para almacenar videos de pacientes, si se descargan)
* **Pantalla:** Resolución mínima 1366x768
* **Cámara (opcional):** Solo en caso de uso móvil en tablets para grabación directa (aunque normalmente el paciente graba con su propio celular).

**Requerimientos de Software: (cliente)**

**Sistema Operativo (PC):** Windows 10 o superior / Linux Ubuntu 20.04 o superior

**Navegador Web Compatible:**

* Google Chrome (recomendado)
* Mozilla Firefox
* Microsoft Edge

**Conexión a Internet:** Estable, mínimo 5 Mbps (para visualizar y cargar videos)

**App móvil (paciente):**

**Sistema operativo Android 8.0 o superior**

**Espacio disponible:** 200 MB (para almacenar temporalmente los videos hasta su subida)

* Requerimientos de Hardware (server/ hosting/BD)

Todavía no está en el host

* Requerimientos de Software (server/ hosting/BD)

Todavía no está en el host

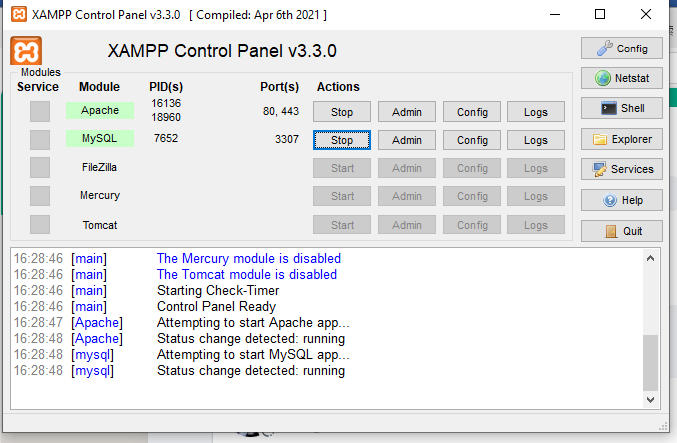
1. **Instalación y configuración:** *Instrucciones detalladas sobre cómo instalar el software, configurar los componentes necesarios y establecer la conexión con otros sistemas o bases de datos*
2. *COMPOSER Y XAMPP PARA LARAVEL (APP WEB Y API BACKEND)*

[*https://www.youtube.com/watch?v=NdcB3bNRV50*](https://www.youtube.com/watch?v=NdcB3bNRV50)

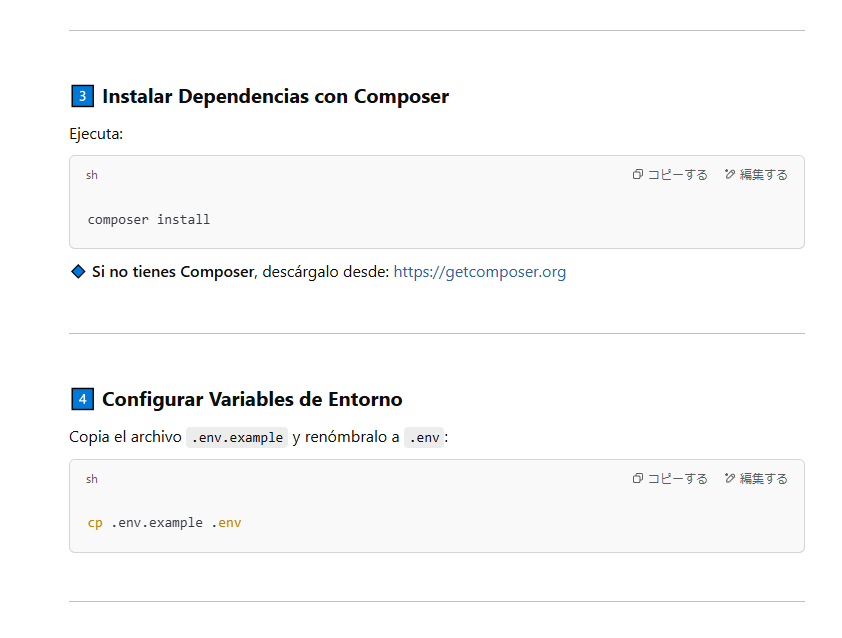
1. *VISUAL STUDIO CODE (Preparación del entorno)*

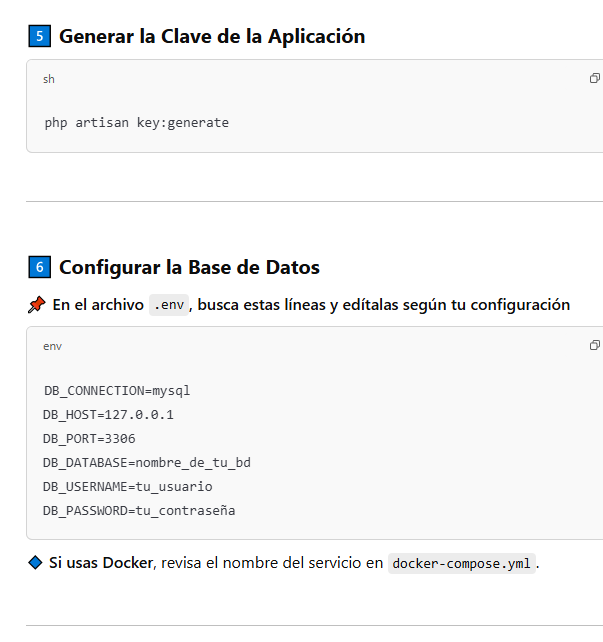
Todos estos comandos lo ejecutas dentro del proyecto de laravel en su terminal de Visual Studio Code. Ejecuta los sgts. Comandos:

PRIMERO: Debes tener el XAMPP abierto con Apache y Mysql corriendo. Si no lo tienes debes buscar un video para instalar Laravel 11 preferible.

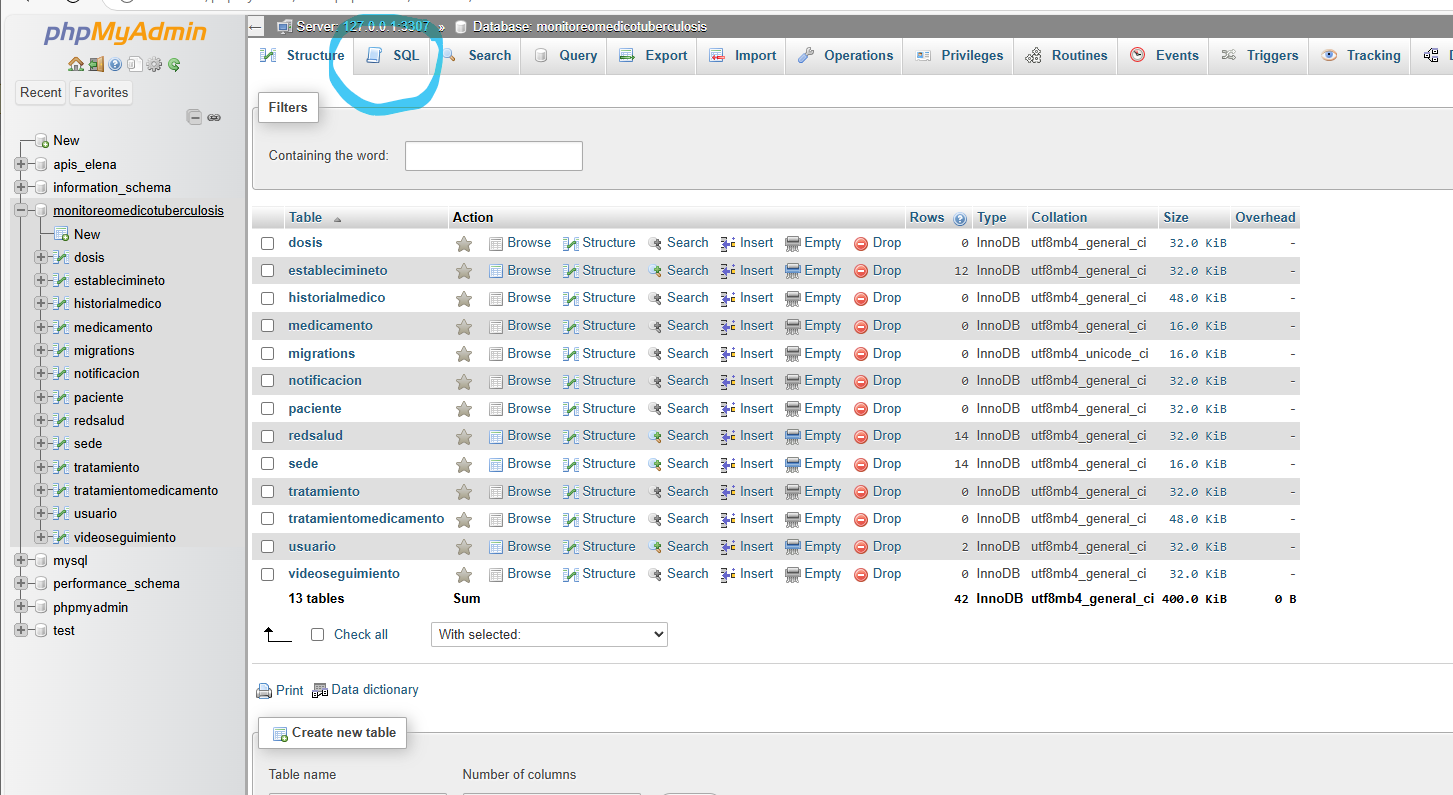


LUEGO SON ESTOS COMANDOS

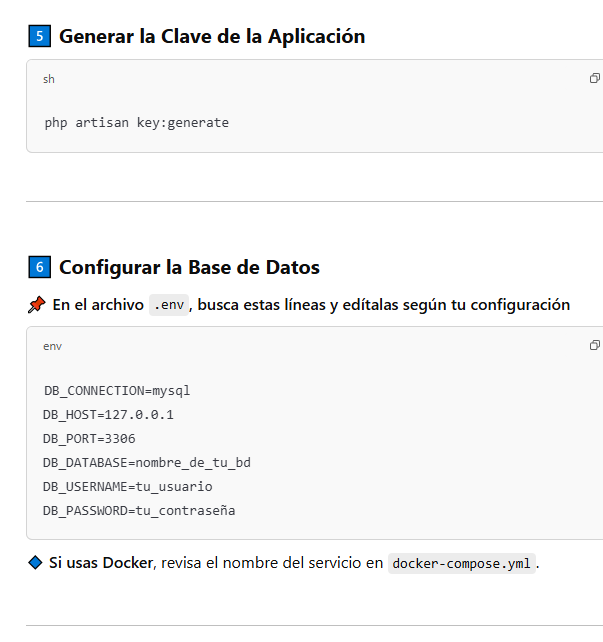




ANTES DEL PASO 6 DEBES TENER LA LA BDD DE JAIRO en Mysql phpmyAdmin. En la sección de “SQL” QUE ESTA MARCANDO, AHÍ COPIAS EL SCRIPT DE Jairo. Y también el script con los datos de ejemplo



LUEGO, RECIEN EL PASO 6.





*SIN EMBARGO, EL PROYECTO PARA QUE HAGA CONEXIÓN CON LA APP MOVIL DE MANERA LOCAL ES IMPORTANTE QUE LO EJECUTES CON ESTE COMANDO. SI ESTAS USANDO XAMPP:*

*php artisan serve --host=192.168.100.150 --port=8000*

*(Ej: 192.168.1.10 es la IP de la laptop/PC)*

1. *FLUTTER + EMULADOR EN ANDROID STUDIO (APP MÓVIL)*

[*https://www.youtube.com/watch?v=BTubOBvfEUE*](https://www.youtube.com/watch?v=BTubOBvfEUE)

*Para hacer correr el proyecto simplemente, debes ejecutar en la terminal:*

* 1. *flutter pub get*
  2. *flutter run*

En caso de que exista errores de dependencias o hay que actualizar los paquetes, ejecutar antes del “flutter run”: flutter pub upgrade

**PROCEDIMIENTO DE HOSTEADO / HOSTING (configuración)**

* **Sitio Web.**
* **B.D.**
* **API / servicios Web**
* **Otros** *(firebase, etc.)*

*Detalle DETALLADO paso a paso de la puesta en marcha en hosting, tanto para el sitio Web, API, B.D., etc.etc. (incluir scripts BD, Credenciales de acceso server, root BD, Admin, users clientes etc.)*

NO SE HA REALIZADO ESO TODAVÍA, DEBIDO A PROBLEMAS EXTERNOS.

1. **GIT :**

* Versión final entregada del proyecto.

<https://github.com/Univalle-PSII/sedes-tuberculosis-ii.git>

* Entrega compilados ejecutables

FOTOS

1. **Dockerizado** *Del Sitio WEB, de la Base de Datos*
   1. **Proceso de dokerizado, Configuración**
   2. **Como hacer Correr, Acceso credenciales:** 
      1. **base datos**
      2. **Roles Admin, User, etc**
      3. **Base de datos con datos válidos y legibles.**
2. **Personalización y configuración:** *Información sobre cómo personalizar y configurar el software según las necesidades del usuario, incluyendo opciones de configuración, parámetros y variables.*
3. **Seguridad**: *Consideraciones de seguridad y recomendaciones para proteger el software y los datos, incluyendo permisos de acceso, autenticación y prácticas de seguridad recomendadas.*

*PUNTOS CRITCOS:*

1. *Primeramente, de acuerdo a los requerimientos del cliente (Ing.Gaston Silva) se ha solicitado la visualización de contraseñas para el administrador argumentando que sería lo óptimo debido a rotación de empleados constante que existe dentro de los hospitales(establecimientos), realizando de una manera sencilla la captura de contraseñas en caso de olvido.*
2. *Segundo, se ha realizado la automatización de creación de credenciales (un usuario de rol enfermero y otro de médico) para cada hospital(establecimiento) que se cree. Las contraseñas y nombres de usuarios de podrán visualizar en la aplicación WEB desde el rol de ADMINISTRADOR.*
3. *Tercero, solo existe 1 administrador donde tiene acceso a todo. Por ejemplo, puede ver de todos los registros de todos los pacientes de Cochabamba. Y no por establecimiento como los médicos y enfermeros. La contraseña es 21061995. En caso de que se quiera reestablecer la contraseña seria dentro del directorio WEB donde esta alojado la app Web y API Backend (LARAVEL 12).*
4. *Cuarto, los pacientes para mayor facilidad de ingresar a la app móvil, simplemente ingresan con su carnet de identidad. Eso con la finalidad de usar correos electrónicos. El cliente busca que el paciente inicie sesión de la manera más sencilla posible.*

* Se esta manejando 2 tipos de autenticación: Autenticación mediante sesiones que LARAVEL 12 implementa para la parte WEB. Y se ha implementado TOKENS PERSONALES LARAVEL SANCTUM para la parte MOVIL. Esto con la finalidad de manejar mejor la seguridad debido a que el manejo de autenticación mediante sesiones es mas sencillo de implementar y también seguro. Tiene una seguridad estándar donde LARAVEL se encarga de casi todo. Y los tokens personales de tipo Bearer que son para la parte móvil también funcionan de una manera sencilla donde el token es validado desde el backend y verificado desde la base de datos.
* La generación de contraseñas para los usuarios médico y enfermero esta manejando su encriptación mediante Bcrypt. Es necesario manejar este tipo de encriptacion debido a que al iniciar sesión en la app web, Laravel espera a que sea de ese tipo de encriptación al momento de logearse un usuario.

1. **Depuración y solución de problemas:** *Instrucciones sobre cómo identificar y solucionar problemas comunes, mensajes de error y posibles conflictos con otros sistemas o componentes.*

¿CÓMO EJECUTAR EL PROYECTO DE MANERA LOCAL?

1. **Glosario de términos:** *Un glosario que incluya definiciones de términos técnicos y conceptos utilizados en el manual.*

| **Término** | **Definición** |
| --- | --- |
| **API (Application Programming Interface)** | Conjunto de definiciones y protocolos que permiten la comunicación entre diferentes aplicaciones. En este proyecto, la API es consumida por la app web y móvil. |
| **Backend** | Parte del sistema que gestiona la lógica, la base de datos y la seguridad. En este proyecto, está desarrollado en Laravel 12. |
| **Frontend** | Parte visual del sistema con la que interactúa el usuario. La app web y la app móvil conforman el frontend. |
| **CRUD** | Siglas de Create, Read, Update y Delete. Son las operaciones básicas que se realizan sobre los datos en un sistema. |
| **Token(Bearer Token)** | Cadena de texto que permite la autenticación segura entre aplicaciones móviles y servidores. Se utiliza en la app móvil mediante Laravel Sanctum. |
| **Laravel** | Framework de desarrollo web en PHP que facilita la creación de aplicaciones modernas y seguras. |
| **Flutter** | Kit de desarrollo de UI creado por Google para construir aplicaciones móviles, web y de escritorio desde una sola base de código. |
| **Firebase** | Plataforma de desarrollo de aplicaciones de Google que ofrece servicios como base de datos, autenticación y almacenamiento. |
| **Bcrypt** | Algoritmo de encriptación utilizado para proteger contraseñas. |
| **Docker** | Plataforma de contenedores que permite empaquetar y desplegar aplicaciones junto con todas sus dependencias. |
| **Sanctum** | Paquete de Laravel utilizado para autenticación basada en tokens, ideal para SPA (Single Page Applications) y aplicaciones móviles. |
| **Emulador** | Herramienta que simula un dispositivo móvil para pruebas de aplicaciones sin necesidad de un dispositivo físico. |

1. **Referencias y recursos adicionales:** *Enlaces o referencias a otros recursos útiles, como documentación técnica relacionada, tutoriales o foros de soporte.*

* **Documentación oficial de Laravel**:  
  <https://laravel.com/docs/12.x>
* **Documentación oficial de Flutter**:  
  https://docs.flutter.dev/
* **Documentación de Laravel Sanctum (autenticación móvil)**:  
  <https://laravel.com/docs/12.x/sanctum>
* **Guía oficial de Docker**:  
  https://docs.docker.com/get-started/
* **Tutorial Laravel + XAMPP (Instalación y configuración)**:  
  <https://www.youtube.com/watch?v=NdcB3bNRV50>
* **Tutorial Flutter + Android Studio (App móvil)**:  
  <https://www.youtube.com/watch?v=BTubOBvfEUE>
* **Repositorio del proyecto en GitHub**:  
  <https://github.com/Univalle-PSII/sedes-tuberculosis-ii.git>
* **Canal de soporte Laravel en español (Foro/Comunidad)**:  
  https://laracasts.com/discuss
* **Documentación sobre Bcrypt en Laravel**:  
  <https://laravel.com/docs/12.x/hashing>

1. **Herramientas de Implementación:**

* Lenguajes de programación:
* PHP
* DART (FLUTTER)
* Frameworks:
* APP WEB ES LARAVEL VERSIÓN 12
* APIs de terceros, etc.
* LARAVEL VERSIÓN 12

1. **Bibliografía**

* **Laravel Documentation – Laravel 12.x**  
  Laravel. (2025). *The PHP Framework for Web Artisans*. Recuperado de: <https://laravel.com/docs/12.x>
* **Flutter Documentation**  
  Google. (2025). *Flutter: Beautiful native apps in record time*. Recuperado de: https://docs.flutter.dev/
* **Laravel Sanctum**  
  Laravel. (2025). *API Token Authentication for SPAs and Mobile Applications*. Recuperado de: <https://laravel.com/docs/12.x/sanctum>
* **XAMPP - Apache Friends**  
  Apache Friends. (2025). XAMPP for Windows. Recuperado de: https://www.apachefriends.org/index.html
* **Firebase Documentation**  
  Google. (2025). Firebase Documentation. Recuperado de: https://firebase.google.com/docs
* **Hashing – Laravel 12.x**  
  Laravel. (2025). Hashing Passwords Securely. Recuperado de: <https://laravel.com/docs/12.x/hashing>
* **GitHub – Repositorio del Proyecto**  
  GitHub. (2025). Sedes Tuberculosis II - Proyecto final. Recuperado de: <https://github.com/Univalle-PSII/sedes-tuberculosis-ii.git>
* **YouTube – Instalación Laravel con XAMPP**  
  YouTube. (s.f.). Instalación Laravel desde cero + XAMPP. Recuperado de: <https://www.youtube.com/watch?v=NdcB3bNRV50>
* **YouTube – Configuración de entorno Flutter**  
  YouTube. (s.f.). Flutter - Primer proyecto móvil. Recuperado de: <https://www.youtube.com/watch?v=BTubOBvfEUE>