



UNIVERSIDAD TECNOLÓGICA DE LA  
HUASTECA HIDALGUENSE

*Organismo Descentralizado de la Administración Pública Estatal*

**INGENIERÍA EN DESARROLLO Y GESTIÓN DE SOFTWARE**



**ASIGNATURA: GESTIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE**

**AUTORES:**

ANGEL JOSUE HERNANDEZ MANUEL 20210672

AGUSTINA HERNANDEZ MARTINEZ 20200700

JUAN CARLOS CIFUENTES CRUZ 20210654

**GRADO: 10° GRUPO: B**

**DOCENTE: ING. ANA MARIA FELIPE REDONDO**



# UNIVERSIDAD TECNOLÓGICA DE LA HUASTECA HIDALGUENSE

*Organismo Descentralizado de la Administración Pública Estatal*

## **I. Objetivo**

El objetivo de desarrollar esta PWA es proporcionar a los usuarios una plataforma integral que les permita, a través de una cuenta personal, contratar servicios de ambulancia de manera rápida y eficiente, gestionar el registro de citas médicas, y realizar donaciones para apoyar causas humanitarias, todo en un entorno accesible, seguro y conveniente. Estas funcionalidades mejorarán la interacción de los usuarios con los servicios esenciales y facilitarán un canal para apoyar causas importantes para la empresa Cruz Roja.

## **II. Alcance**

El alcance de la PWA incluye el desarrollo de una plataforma que permita a los usuarios, mediante la creación de una cuenta, acceder a diversas funcionalidades clave: la contratación de servicios de ambulancia en tiempo real, el registro y gestión de citas médicas, y la posibilidad de realizar donaciones a causas relacionadas con servicios de salud. La aplicación integrará sistemas de notificaciones para recordar citas y confirmar servicios, así como un portal seguro para gestionar transacciones y donaciones. Además, incluirá un sistema de perfiles de usuario donde podrán consultar el historial de servicios contratados, citas agendadas, y transacciones realizadas.

## **III. Equipo de trabajo**

El equipo de trabajo está conformado por los siguientes integrantes:

Ángel Josué Hernández Manuel – Líder del proyecto, Desarrollador y Documentador

Agustina Hernández Martínez – Líder de proyecto, Desarrollador y Documentador

Juan Carlos Cifuentes Cruz – Líder de proyecto, Desarrollador y Documentador



# UNIVERSIDAD TECNOLÓGICA DE LA HUASTECA HIDALGUENSE

*Organismo Descentralizado de la Administración Pública Estatal*

## IV. Riesgos Identificados

- Retrasos en la entrega del proyecto debido a tareas técnicas complejas:

**Medida de mitigación:** Establecer fechas límites intermedias (hitos) para asegurar el seguimiento del progreso y permitir ajustes tempranos.

- Falta de comunicación efectiva entre los miembros del equipo, lo que puede llevar a malentendidos:

**Medida de mitigación:** Implementar reuniones breves para revisar el estado del proyecto y asegurar que todos estén alineados con las tareas.

- Complejidad en la gestión del estado de la aplicación:

**Medida de mitigación:** Utilizar un sistema de gestión de estado como para mantener el estado centralizado y evitar problemas de sincronización entre componentes.

- Dependencia de servicios de terceros que pueden fallar o presentar limitaciones:

**Medida de mitigación:** Implementar soluciones alternativas, como una capa de seguridad propia, en caso de que el servicio de terceros falle.



# UNIVERSIDAD TECNOLÓGICA DE LA HUASTECA HIDALGUENSE

*Organismo Descentralizado de la Administración Pública Estatal*

## V. Estrategias de comunicación

- Canales de comunicación: Sesión en meet para la comunicación diaria y a discusión rápida de problemas. Establecer un proceso claro para la resolución de conflictos. Por ejemplo, en caso de un malentendido técnico, se podrían organizar sesiones de aclaración en las que todos los miembros participen y expongan sus puntos de vista para llegar a una solución.
- Utilizar la herramienta de Trello o cualquier herramienta de gestión de tareas para asignar, priorizar y hacer el seguimiento del estado de cada tarea.
- Utilizar un sistema de control de versiones como **Git** con ramas específicas para las plataformas si se requiere personalización por plataforma. Asegurarse de que las integraciones se realicen cuidadosamente y sean revisadas por miembros clave del equipo.



## VI. Metodología Ágil 2

La metodología ágil seleccionada fue Scrum, el cual es una metodología para la gestión de proyectos en donde nos permitirá obtener un resultado rápido en entornos muy cambiantes, mantiene un marco de trabajo ágil a través del cual, cada integrante del equipo podrá abordar problemas complejos adaptativos y entregar los productos de forma eficiente y creativa, de igual manera esta metodología permite el trabajo colaborativo. La metodología comprende de 4 fases, el cual en cada fase se realizan diferentes funciones para terminar con éxito cada fase.

- **Planificación:** En esta fase el equipo establecerá las tareas prioritarias del proyecto a desarrollar para la PWA como los objetivos y la lista de lo más importante para el Product Backlog (Sprint). Estableciendo las prioridades según el valor de cada entregable.
- **Ejecución del Sprint:** En esta fase el equipo trabajara en los elementos seleccionados del Sprint. Desarrollar los diseños e implementar la configuración del entorno de desarrollo para la PWA, las funcionalidades enfocándonos en cumplir los objetivos de cada uno de los Sprint de acuerdo a la planificación inicial.
- **Revisión del Sprint:** Al finalizar las actividades establecidas en el Sprint, se realizan la revisión del trabajo completado. Se verificará si los elementos cumplen con los criterios de aceptación.
- **Retrospectiva del Sprint:** Después de la revisión, se analiza el Sprint completado para identificar áreas de mejoras en el desarrollo o que es lo que funciono durante el Sprint que no.



## VII. Herramienta de versionamiento

Para el manejo de versionamiento se eligió GitHub.

Consideramos que dentro del desarrollo del proyecto el control de versiones es esencial para asegurar el trabajo colaborativo y poder trabajar en diferentes funcionalidades sin riesgo de sobrescribir el trabajo de otros. Esta herramienta nos permitirá registrar y revisar cada cambio hecho en el código, con la posibilidad de volver a versiones anteriores si surge algún problema. Esto proporciona un entorno más seguro y organizado para el desarrollo. Al desarrollar la aplicación, los integrantes del equipo podremos trabajar en distintas características o resolver bugs de manera simultánea sin afectar el código principal (branch main o master). Los pull requests permiten revisar el código de manera colaborativa antes de integrarlo al proyecto principal, lo que ayuda a garantizar la calidad y a mantener la coherencia en el estilo de codificación. GitHub se integra fácilmente con plataformas de Integración Continua y Entrega Continua (CI/CD). Estas integraciones permiten ejecutar automáticamente pruebas unitarias, de integración y de interfaz de usuario cada vez que se realiza un cambio en el código. Además, GitHub permite automatizar el proceso de generación. GitHub permite una gestión avanzada de ramas, lo que facilita a los equipos trabajar en diferentes funcionalidades, mejoras o correcciones de errores sin afectar el código principal hasta que todo esté probado y listo para ser fusionado. GitHub ofrece una forma organizada de gestionar estas ramas, permitiendo que los desarrolladores trabajen en distintas características o correcciones de forma simultánea sin interferencias, hasta que los cambios estén listos para ser integrados en la rama principal. Facilita la automatización de pruebas y despliegue, lo cual es crucial para asegurar la calidad continua en un entorno de desarrollo rápido como el de React.



## **VIII. Estrategia de versionamiento**

Para el desarrollo de nuestra PWA, hemos optado por utilizar un flujo de trabajo centralizado para el control de versiones. Esta estrategia se adapta perfectamente a las necesidades de un equipo pequeño como el nuestro, compuesto por tres integrantes.

El flujo de trabajo centralizado se basa en el uso de un único repositorio central donde la rama principal (main) contiene siempre la versión estable del código. Todos los desarrolladores realizan cambios directamente en esta rama, asegurando que:

- El código siempre esté listo para ser desplegado en producción.
- La simplicidad del flujo de trabajo facilita la colaboración directa entre los miembros del equipo, sin la necesidad de gestionar múltiples ramas o flujos complejos.

Como herramienta de control de versiones, hemos elegido GitHub, que nos permitirá trabajar de manera colaborativa mediante el uso de pull requests para revisión de código, garantizando la calidad del desarrollo antes de realizar cualquier integración. Además, GitHub facilita la integración continua (CI), lo que asegura que las pruebas se ejecuten de manera automática con cada cambio, manteniendo la estabilidad del proyecto.



## IX. Estrategia de despliegue

Para el despliegue de la PWA, implementaremos la estrategia Canary, con el objetivo de reducir riesgos al momento de lanzar nuevas versiones del software. Con esta estrategia, se despliega la nueva versión a un subconjunto limitado de usuarios antes de hacerlo disponible para todos. Esto nos permitirá:

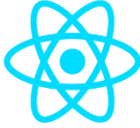









- **Minimizar riesgos:** La versión nueva es probada por un pequeño grupo de usuarios, lo que nos permite identificar posibles errores sin afectar a toda la base de usuarios.
- **Recoger feedback temprano:** El comportamiento de la nueva versión puede monitorearse en tiempo real, permitiendo ajustes rápidos y efectivos.
- **Rollback fácil:** Si se detectan problemas, se puede revertir a la versión anterior de manera sencilla, minimizando el impacto en la experiencia del usuario.

Este enfoque es ideal para nuestra PWA, ya que permite un despliegue controlado y gradual, asegurando que el sistema siga siendo estable y funcional mientras introducimos nuevas características. Utilizaremos herramientas de monitoreo y seguimiento para asegurar un proceso de despliegue eficiente y sin interrupciones significativas.





## X. Entornos de desarrollo

Capa de Presentación	Capa de negocio	Capa de Datos
 React JS  tailwindcss 	   Nodemailer Google APIs  Abstract	 Microsoft SQL Server™  HOSTINGER  Vercel

### 10.1 Capa de presentación

En la capa de presentación se eligieron algunas herramientas como las siguientes:

- **React** : Esta herramienta se utilizará para el desarrollo de la PWA, ya que nos ayuda por la forma de trabajo, permite una fácil integración con Service Workers, que son fundamentales para convertir una web app en una PWA.
- **TailwindCSS**: Nos permitirá crear los estilos de diseños personalizados, para hacer accesible la interfaz de usuario.
- **Zustand**: Este es una librería el cual nos ayudara a manejar la gestión de estado de la aplicación.



# UNIVERSIDAD TECNOLÓGICA DE LA HUASTECA HIDALGUENSE

*Organismo Descentralizado de la Administración Pública Estatal*

## 10.2 Capa de negocio

- **Node js:** Este entorno de ejecución de JavaScript permitirá desarrollar el backend o la lógica del servidor.
- **Express:** Facilitara la creación de servidores y APIs RESTful usados en el proyecto de la parte de backend.
- **Nodemailer:** Esta Api, nos permitirá con él envío de correo para la recuperación de contraseña.
- **Google Api's:** Se utilizará para el inicio de sesión con Google.
- **Abstract:** Este servicio que ofrece la API se utilizará para realizar validaciones de datos, como la validación de correos electrónicos, números de teléfono, entre otros. Útil para garantizar la calidad de los datos ingresados por los usuarios.

## 10.3 Capa de datos

- **SQLserver:** El tipo de base de datos diseñados es relacional, por esa razón se utilizada esta herramienta para la gestión de datos.
- **Vercel:** En esta herramienta es en donde actualmente se manejan la gestión de nuestras Api's.
- **Hostinger:** El servicio de hosting que será contratado para tener todo en la nube y poder desplegar las aplicaciones.