

ABPro: Implementación Completa de PWA en la Web del Hospital

Contexto:

En este **Aprendizaje Basado en Proyectos (ABPro)**, los estudiantes deberán implementar una **PWA completa** para la **web del hospital**. El proyecto deberá hacer uso de **ReactJS** y aprovechar al máximo las características de **PWA**, incluyendo almacenamiento web avanzado, estrategias de **Service Worker**, y análisis de rendimiento y accesibilidad con **Lighthouse**. A su vez, se integrarán nuevas funcionalidades de **uso de periféricos del sistema operativo**, accesos a **APIs externas**, y una estrategia avanzada de despliegue en un servidor.

Duración: 2 horas

Requisitos:

1. Creación del Manifiesto y Configuración Inicial (1 punto)

- Crear el **archivo de manifiesto** de la aplicación que permita su instalación en dispositivos móviles:
 - Incluir el nombre, iconos adaptativos en varias resoluciones, tema de color y modo pantalla (fullscreen/standalone).
 - Asegurar que la aplicación sea reconocida como **PWA** y se pueda instalar.

Tip: Enfócate en optimizar los íconos y la estructura del manifiesto para dispositivos móviles.

2. Integración de Service Worker para Gestión Avanzada de Caché (1.5 puntos)

- Configurar un **Service Worker** avanzado:
 - Precaching para los recursos principales de la PWA (HTML, CSS, JS).
 - Implementar al menos tres estrategias de almacenamiento en caché (por ejemplo, **CacheFirst** para archivos estáticos, **NetworkFirst** para datos dinámicos, **Stale-While-Revalidate** para contenido mixto).
 - Implementar la gestión del ciclo de vida del **Service Worker**, garantizando la actualización de la caché cuando se publiquen nuevas versiones de la PWA.

Tip: Añade la capacidad de notificar al usuario cuando haya una nueva versión de la PWA disponible.

3. Acceso a Periféricos del Sistema Operativo (1.5 puntos)

- Implementar el acceso a al menos uno de los siguientes periféricos del dispositivo:
 - **Cámara:** Permitir la captura de imágenes o escaneo de documentos médicos dentro del sistema del hospital.
 - **Geolocalización:** Integrar una funcionalidad que permita obtener la ubicación del usuario para realizar un seguimiento de su localización o acceder a servicios cercanos.

Tip: Asegúrate de pedir permisos al usuario de manera adecuada y manejar errores en caso de que se niegue el acceso.

4. Consumo de API Externa para Datos Médicos (1.5 puntos)

- Integra una API externa que provea información relevante al hospital (por ejemplo, base de datos de doctores o medicamentos).
 - Usa **Axios** o **Fetch API** para consumir la API.
 - Mostrar los datos obtenidos de la API en la **PWA** utilizando componentes de **React**.
 - Implementar manejo de errores y alertas si la API no responde o se produce un fallo en la conexión.

Tip: Utiliza **async/await** y asegura que la PWA maneje correctamente las respuestas lentas de la API.

5. Pruebas de Rendimiento y Optimización con Lighthouse (1.5 puntos)

- Ejecuta **Lighthouse** para validar que la aplicación sea instalable, optimizada, y cumpla con los estándares de accesibilidad, rendimiento y seguridad.
 - Mejora las áreas señaladas por **Lighthouse** y optimiza el tiempo de carga de la aplicación.
 - Documenta los cambios realizados a partir de las sugerencias de **Lighthouse** y asegúrate de mejorar el puntaje general de la aplicación.

Herramientas a Utilizar:

- **ReactJS** como base de la aplicación.
 - **Service Worker** para la gestión del caché y funcionalidad offline.
 - **LocalStorage** o **IndexedDB** para el almacenamiento de datos.
 - **APIs externas** para la integración de datos dinámicos.
 - **Acceso a periféricos** mediante APIs del navegador.
 - **Lighthouse** para evaluar el rendimiento, accesibilidad y estado de la PWA.
-

Entrega:

- **Formato de entrega:**
 - Opción 1: Enviar un **enlace al repositorio de GitHub** con la PWA completamente funcional y los cambios optimizados basados en los informes de **Lighthouse**.
 - Opción 2: Entregar un archivo **ZIP comprimido** con el proyecto completo, incluyendo el manifiesto, **Service Worker** personalizado, y las pruebas realizadas con **Lighthouse**.