**Aplicación Web PWA de Registro de asistencia**

**Problemática**

A menudo no disponemos de un lugar accesible para registrar nuestra asistencia a clases para llevar un control cuando no podemos ingresar a la plataforma, muchos estudiantes recurren a notas en papel o ingresar a la plataforma donde no se puede ver de forma directa tu asistencia sin antes pasar por todos los módulos que tiene esta. Además, cuando no hay conexión a Internet se vuelve imposible ver nuestras asistencias y es frustrante cuando debes indicar que días asististe por si el profesor no registró nuestra asistencia bien en la plataforma.

**Objetivo General**

Construir una Aplicación Web progresiva (PWA) minimalista y de fácil uso, que permita a los estudiantes:

1. Registrar sus materias (días, horarios y profesor).
2. Marcar su asistencia sin necesidad de conexión, guardándola localmente.
3. Consultar, en cualquier momento, un resumen filtrable de sus asistencias.

**Objetivos Específicos**

* Ofrecer un flujo offline-first que garantice registro sin Internet.
* Mantener interfaz limpia y responsiva, adaptable a móvil y escritorio.
* Sincronizar datos automáticamente con la base de datos cuando se recupere conexión.
* Evitar distracciones: únicamente materias y asistencia, en un diseño minimalista.

**Alcance Funcional para el primer parcial**

* Gestión de Materias: CRUD de materias (nombre, días, horario, profesor).
* Registro de Asistencia: Marcado “Sí/No” para cada materia, guardado en cache.
* Resumen: Vista de historial; filtros por mes y semana; detecta falta de conexión y sincroniza luego.
* PWA: Instalable en dispositivos, funciona como app nativa (Service Worker + manifest).

**Alcance funcional para el segundo parcial:**

* Crear un módulo de login para que los usuarios puedan crear una cuenta, iniciar sesión y recuperar contraseña.
* Sincronizar sus datos a la base de datos cuando tengan internet
* Registrar usuarios y materias en la base de datos

**Tecnologías**

* **Frontend:** HTML5, CSS3 (responsive), JavaScript (vanilla, PWA APIs).
* **Backend:** Php, MySQL
* **PWA:** Service Worker para cache, manifest.json para instalación
* **IDE:** Visual Studio Code

**Módulos a implementar en este primer parcial**

1. **Módulo de Gestión de Materias (CRUD)**

* Formulario para crear y editar materias.
* Tabla dinámica para listar materias registradas.
* Botón para cancelar la edición y limpiar campos.

1. **Módulo de Registro de Asistencia**

* Identificación automática de materia activa según día y hora.
* Registro de asistencia con opciones "Sí" / "No" y botón "Confirmar".
* Validación para registrar sólo durante horas de clase.

1. **Módulo de Resumen de Asistencias**

* Tabla de resumen semanal y mensual de asistencias (✔/✘).
* Filtros dinámicos por mes y semana.
* Indicador de materias con baja asistencia.

1. **Módulo de Funcionalidad Offline (PWA)**

* Implementación de Service Worker para cacheo de archivos.
* Creación de manifest.json para instalación como app web.
* Funcionalidad completa sin conexión usando localStorage.

**Módulos a implementar en el segundo parcial**

1. **Autenticación de Usuarios (Login y Registro)**

* Registro de usuarios nuevos con validación de campos y contraseñas en PHP.
* Inicio de sesión con control de errores (usuario inexistente, contraseña incorrecta, cuenta bloqueada).
* Manejo de sesiones para mantener la sesión activa tras iniciar sesión correctamente.

1. **Validación y Protección del Acceso**

* Restricción de acceso a módulos (editar materias, asistencia, resumen) solo a usuarios autenticados.
* Redirección a la página de login si no hay sesión activa.
* Sanitización de datos ingresados para evitar inyecciones SQL o datos corruptos.

1. **Conexión con Base de Datos MySQL**

* Configuración de conexión PHP ↔ MySQL para insertar, actualizar y consultar datos de: Usuarios, Materias y Registros de asistencia.

1. **Sincronización entre localStorage y base de datos (modo offline/online)**

* Implementación de un módulo que detecte si hay conexión disponible y sincronice los datos almacenados localmente con la base de datos remota.
* Registro de pendientes cuando no hay conexión, y carga posterior alrestaurar internet.