Módulo 6
Desarrollo de aplicaciones Web Progresivas (PWA)

Almacenamiento en una PWA

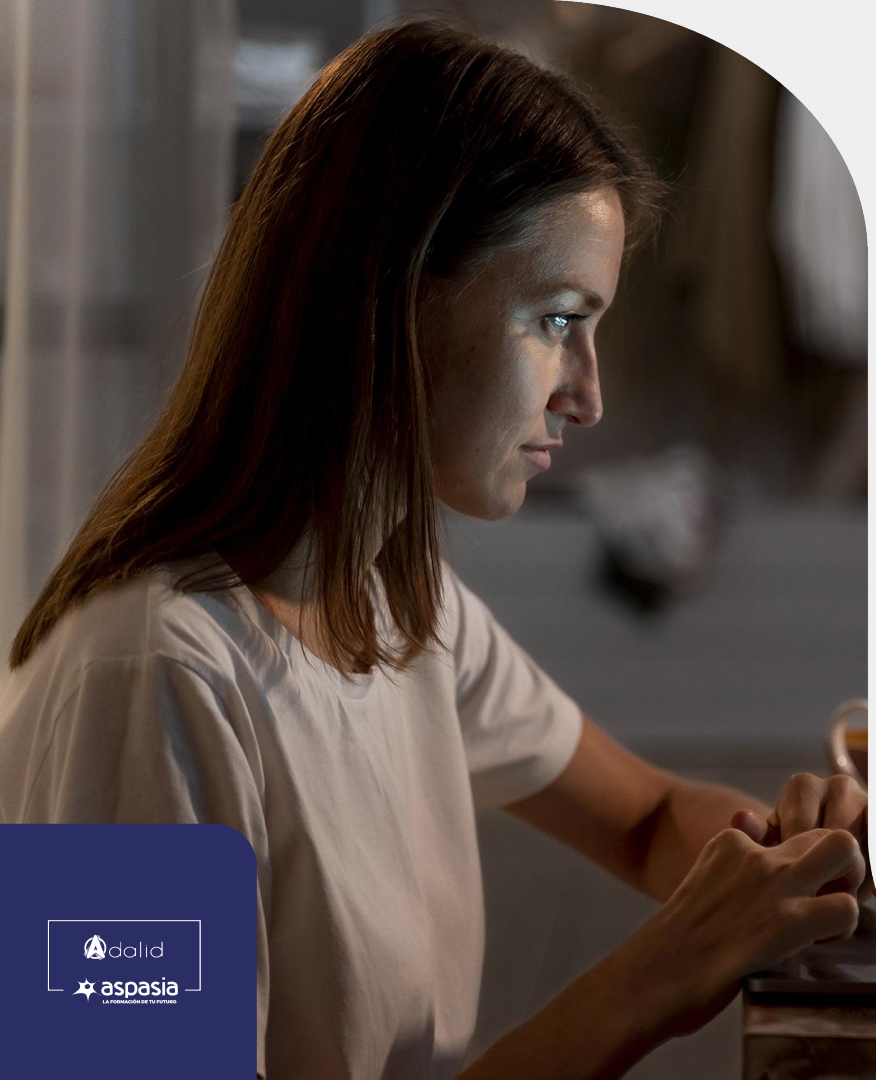
Módulo 6

AE 2.2

OBJETIVOS

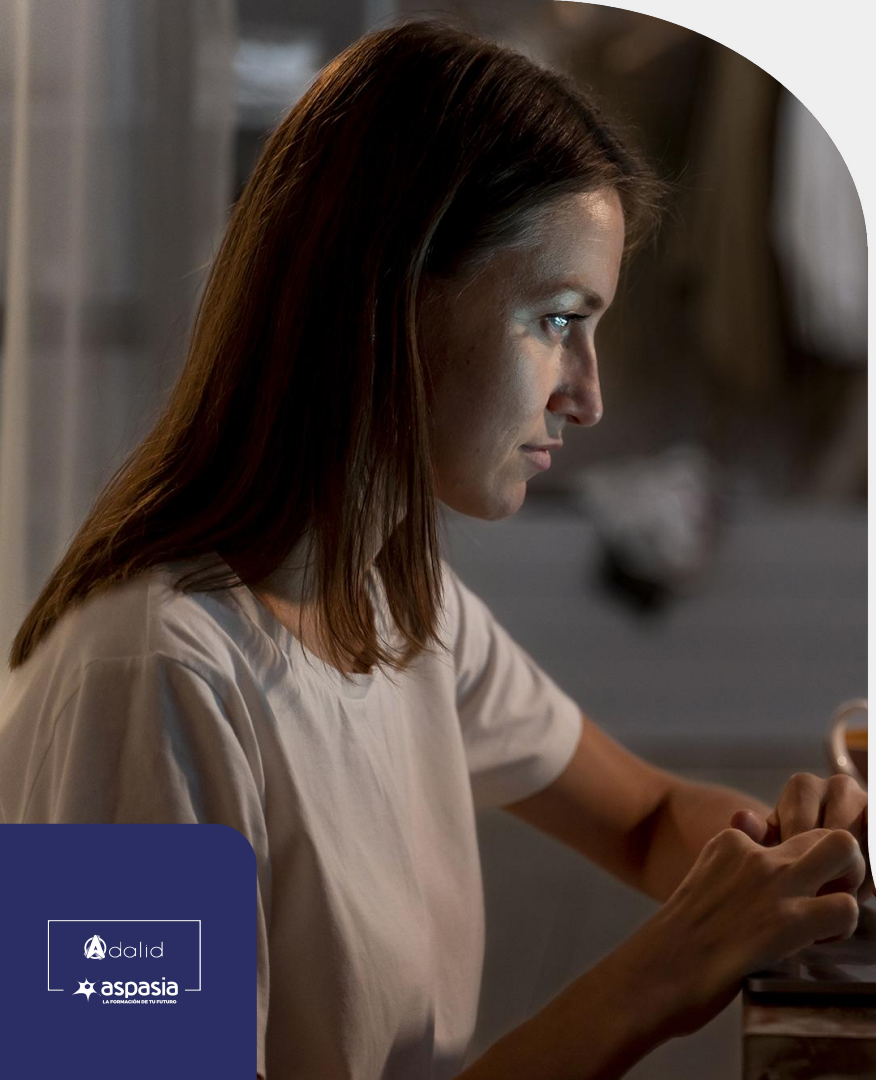
Entender como las PWA usan LocalStorage, IndexedDB y Service Workers para almacenar datos y mejorar la experiencia offline. Como con Lighthouse, se evalúa su rendimiento, instalación y optimización, logrando apps rápidas, confiables e intuitivas.

Chile
Adalid
aspasia
LA FORMACIÓN DE TU FUTURO



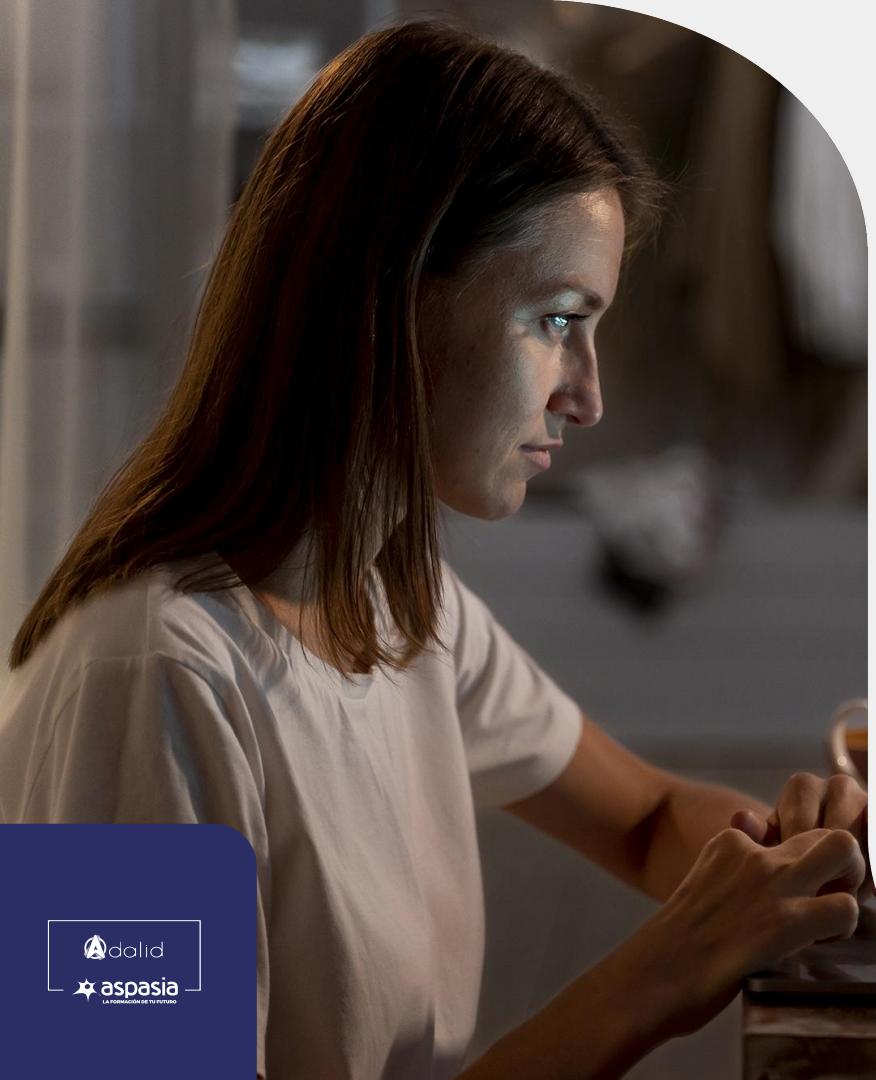
¿QUÉ VAMOS A VER?

- Almacenamiento en una PWA.
- Cómo se administra el Almacenamiento Web en una PWA.
- El uso de LocalStorage y SessionStorage.
- Acerca de WebAssembly y el código precompilado.
- Bibliotecas de bases de datos con soporte para PWA(IndexedDB, PouchDB, RxDB, GunDB).



¿QUÉ VAMOS A VER?

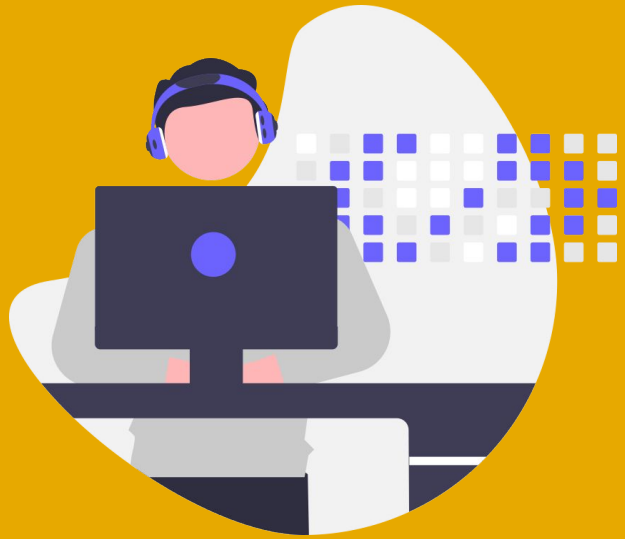
- Creando una PWA.
- Creando un proyecto PWA en ReactJs.
- Registrando un Service Worker.
- Personalizando un Service Worker.
- Navegando a través de una PWA.
- Implementando las distintas estrategias de almacenamiento en caché de Service Worker.
- Accediendo a periféricos del Sistema Operativo.
- Despliegue de una PWA.



¿QUÉ VAMOS A VER?

- Explorando el Estado de una PWA con Lighthouse.
- Instalando la extensión lighthouse en el navegador.
- Algunos aspectos relevantes de lighthouse (Rápido, instalable, optimizado para PWA).
- Ejecutando lighthouse (Análisis de carga de página)
- Revisando el informe de lighthouse.

Pongamos a prueba lo aprendido 😊!!!



Adalid aspasia

Ejercicio Guiado: Implementando localStorage en una PWA

En este ejercicio, vamos a mejorar la PWA que creamos anteriormente https://github.com/adalid-cl/ESPECIALIZACION_FRONTEND_M6_AE2 añadiendo persistencia de datos mediante localStorage, asegurando que la información del usuario **se mantenga disponible aunque cierre la aplicación o actualice la página**.

Ejercicio Guiado: Implementando localStorage en una PWA

1. Clona el repositorio si aún no lo tienes:

```
git clone https://github.com/adalid-cl/ESPECIALIZACION_FRONTEND_M6_AE2.git
cd ESPECIALIZACION_FRONTEND_M6_AE2
npm install
```

- Instala las dependencias necesarias:

```
npm install
```

Ejercicio Guiado: Implementando localStorage en una PWA

2. Modificar App.jsx para Usar localStorage

- Dentro de la Arrowfunction **App** elimina la constante que por defecto carga notas.
- Agrega una Arrowfunction que te permita **Cargar notas desde localStorage.**
- Agrega un **useEffect** que te permita **Guardar notas en localStorage cuando cambien.**
- Recuerda importar **useEffect.**

```
// Cargar notas desde localStorage
const getStoredNotes = () => {
  const savedNotes = localStorage.getItem("notes");
  return savedNotes ? JSON.parse(savedNotes) : [];
};

const [notes, setNotes] = useState(getStoredNotes());

// Guardar notas en localStorage cuando cambien
useEffect(() => {
  localStorage.setItem("notes", JSON.stringify(notes));
}, [notes]);
```

Probar la Persistencia de localStorage

3. Ejecuta la aplicación

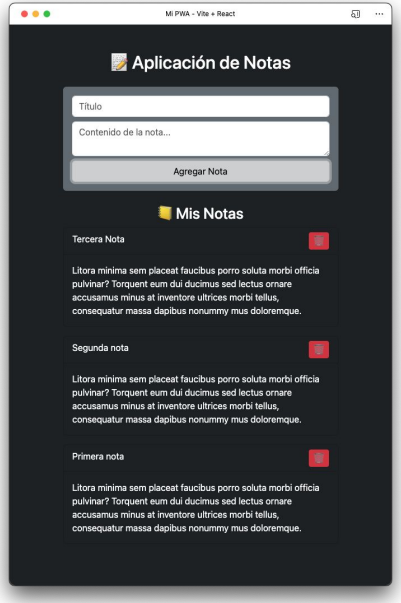
```
npm install
```

- **Añade algunas notas y recarga la página:**
  - Verás que las notas siguen ahí. 🎉
- **Elimina una nota y recarga nuevamente:**
  - Se mantendrán sólo las notas restantes. 💾

Probar la Persistencia de localStorage

Con esto, ya puedes agregar notas y asegurarte de que persistan incluso si cierras la aplicación.

Ahora, vamos a añadir una nueva funcionalidad. En esta ocasión, incorporaremos la posibilidad de agregar **imágenes** y **geolocalización**, lo que permitirá enriquecer aún más las funcionalidades de nuestra PWA.



Agregar Cámara y Ubicación a las Notas en la PWA

1. Modifica App.jsx para Soportar Imágenes y Ubicación

- En la función para agregar notas nuevas **addNote** busca la constante **newNote** y agrega dos argumentos nuevos **image y location.**

```
// Función para agregar una nueva nota
  const addNote = (title, content, image, location) => {
    const newNote = { id: Date.now(), title, content, image, location };
    setNotes([newNote, ...notes]);
  };
```

Agregar Cámara y Ubicación a las Notas en la PWA

2. Modifica NoteForm.jsx para Capturar Foto y Ubicación

- 2.1 En la constante **NoteForm** agregaremos la posibilidad de agregar imágenes y localización

```
const NoteForm = ({ addNote }) => {
  const [title, setTitle] = useState("");
  const [content, setContent] = useState("");
  const [image, setImage] = useState(null);
  const [location, setLocation] = useState(null);
```

Agregar Cámara y Ubicación a las Notas en la PWA

2. Modifica NoteForm.jsx para Capturar Foto y Ubicación

- 2.2 Agregamos la funcionalidad de capturar imágenes y guardarla en base64

```
// Capturar Imagen con la Cámara
  const handleCapture = (event) => {
    const file = event.target.files[0];
    if (file) {
      const reader = new FileReader();
      reader.onloadend = () => {
        setImage(reader.result); // Guardar la imagen
en base64
      };
      reader.readAsDataURL(file);
    }
  };
```

Agregar Cámara y Ubicación a las Notas en la PWA

2. Modifica NoteForm.jsx para Capturar Foto y Ubicación

- 2.3 Agregamos la funcionalidad para obtener la ubicación actual por medio de latitud y longitud

```
// Obtener Ubicación Actual
  const getLocation = () => {
    if ("geolocation" in navigator) {
      navigator.geolocation.getCurrentPosition(
        (position) => {
          setLocation({
            lat: position.coords.latitude,
            lon: position.coords.longitude,
          });
        },
        (error) => {
          console.log("Error obteniendo la ubicación:",
error);
        }
      );
    } else {
      console.log("Geolocalización no soportada en este
navegador.");
    }
  };
```

Agregar Cámara y Ubicación a las Notas en la PWA

2. Modifica NoteForm.jsx para Capturar Foto y Ubicación

- 2.4 Ahora modificaremos la funcionalidad para guardar notas

```
// Guardar Nota
  const handleSubmit = (e) => {
    e.preventDefault();
    if (!title.trim() || !content.trim()) return;

    addNote(title, content, image, location);
    setTitle("");
    setContent("");
    setImage(null);
    setLocation(null);
  };
```

Agregar Cámara y Ubicación a las Notas en la PWA

2. Modifica NoteForm.jsx para Capturar Foto y Ubicación

- 2.5 Ahora agregamos el botón para capturar foto y el botón par aobtener la ubicación dentro del **return**.

```
{/* Botón para capturar foto */}
<div className="mb-2">
  <label className="form-label text-white">📷 Adjuntar Foto:</label>
  <input type="file" accept="image/*" capture="environment" className="form-control"
onChange={handleCapture} />
</div>

{/* Botón para obtener ubicación */}
<div className="mb-2">
  <button type="button" className="btn btn-light w-100" onClick={getLocation}>
    📍 Obtener Ubicación
  </button>
</div>
```

Agregar Cámara y Ubicación a las Notas en la PWA

3. Modifica NoteCard.jsx para Mostrar Imagen y Ubicación

- Agregamos dentro del **return** la posibilidad de visualizar las imágenes y las coordenadas.

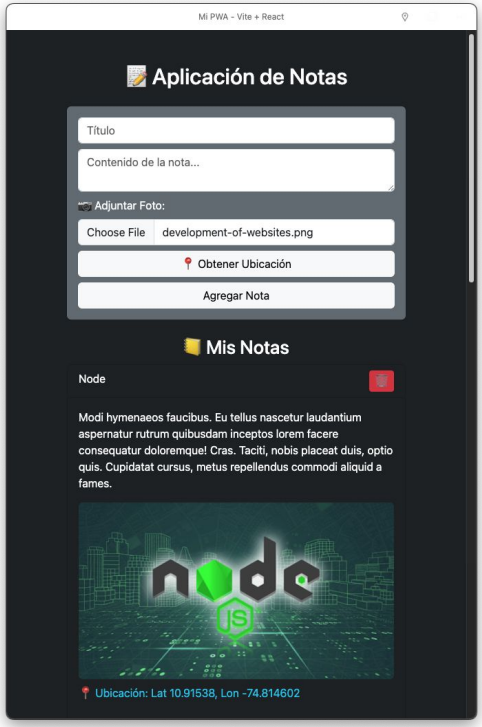
```
{/* Mostrar la imagen si existe */}
{note.image && (
  <img src={note.image} alt="Nota" className="img-fluid rounded mb-2" />
)}

{/* Mostrar la ubicación si existe */}
{note.location && (
  <p className="text-info">
    📍 Ubicación: Lat {note.location.lat}, Lon {note.location.lon}
  </p>
)}
```

Probar la Persistencia de localStorage

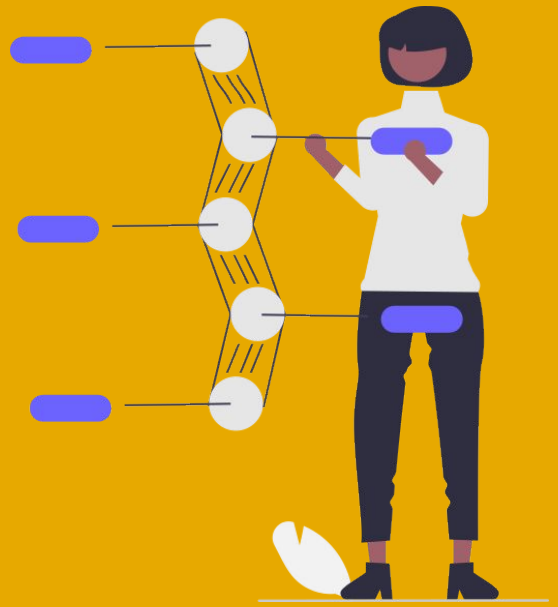
4. Ejecuta la aplicación

```
npm install
```

- **Crea una nota con foto y ubicación:**
  - Pulsa en "Adjuntar Foto" y toma una foto con tu cámara.
  - Pulsa en "Obtener Ubicación" para capturar tu posición.
  - Guarda la nota y verifica que la foto y ubicación se muestran correctamente.
- **Refresca la página y verifica que las notas siguen almacenadas.**



Resumen de lo aprendido 🧑‍🎓



Adalid aspasia

Resumen de lo aprendido

- Las PWA pueden almacenar datos con LocalStorage, SessionStorage e IndexedDB, entre otras opciones.
- Service Workers permiten el almacenamiento en caché y mejoran la experiencia offline.
- Lighthouse es una herramienta clave para auditar el rendimiento y optimización de una PWA.
- Una PWA bien construida debe ser rápida, confiable e instalable en múltiples dispositivos.

GRACIAS POR TU ATENCIÓN

Nos vemos en la próxima clase

Chile
Adalid
aspasia
LA FORMACIÓN DE TU FUTURO