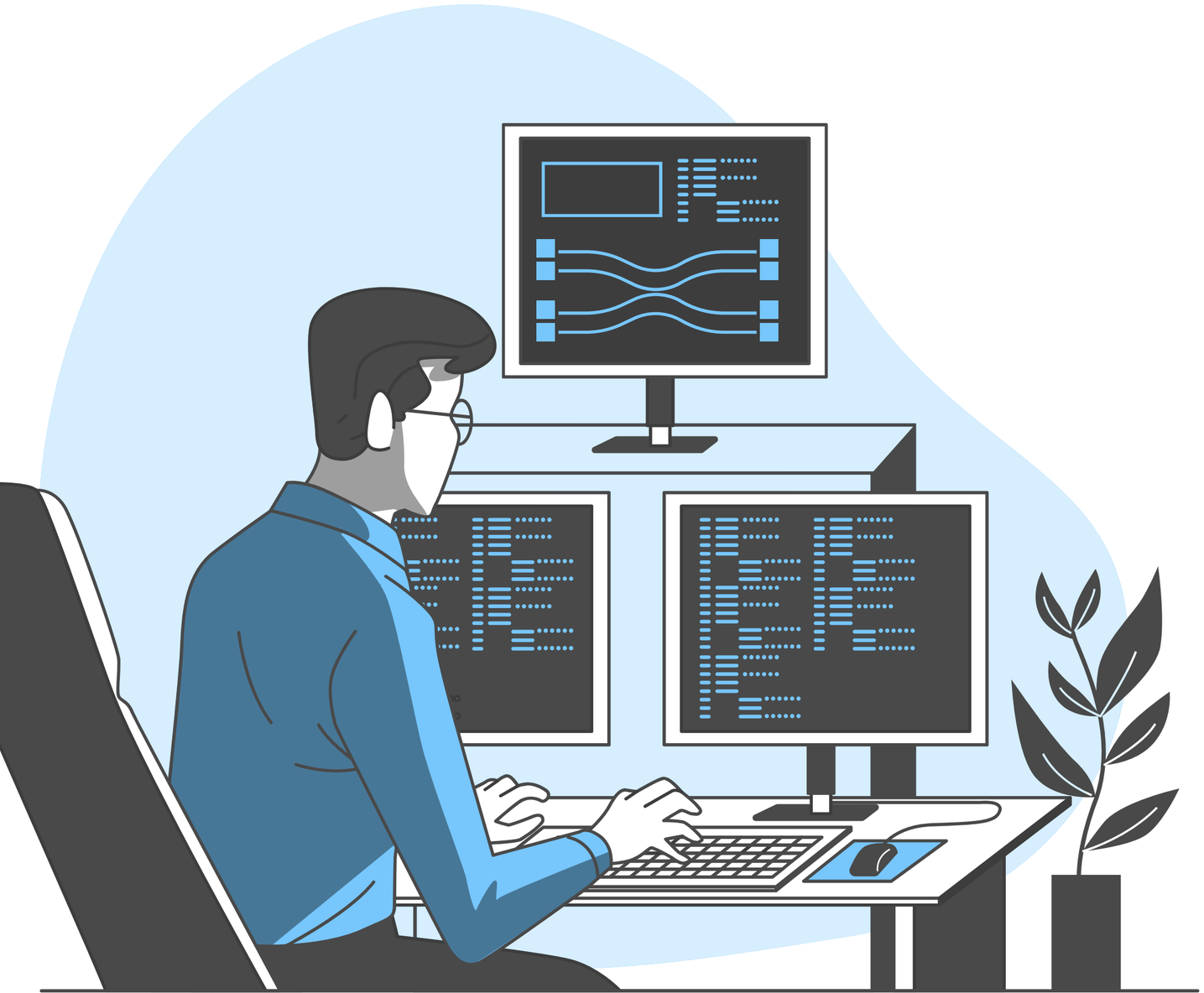


Tarea 2 Web Assembly y PWAs

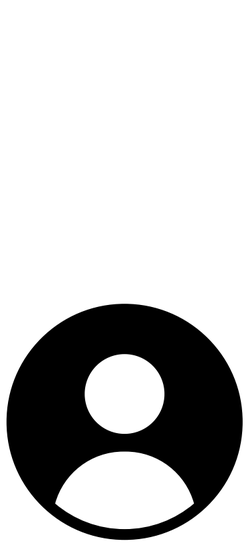
Grupo 9

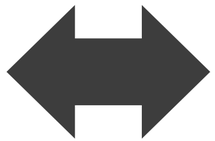
Demo

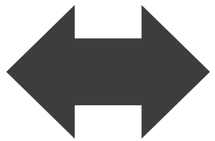
Arquitectura General

Usuario

Frontend

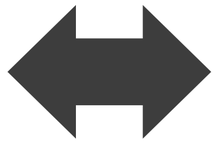






WASM

Procesamiento Imagénes



Herramientas utilizadas

Lenguaje eficiente y seguro para el procesamiento de imágenes.

Permite ejecutar código Rust en el navegador de manera nativa y rápida.

Bundler rápido para el frontend que optimiza la app y permite recargas instantáneas.

Transforma la app en una aplicación instalable, con soporte offline y experiencia tipo app nativa.

...

Implementación Frontend

Interfaz y UX

- Subida y previsualización de imagen
- Botones de filtros
- Muestra de imagen procesada
- Botón para descarga
- View Uploads
- Estilo responsive

Implementación Rust + WASM

Rust + WebAssembly

- Codigo Rust que aplica filtros
- Compilación a .wasm, lo que permite ejecutar código en el navegador manteniendo un rendimiento cercano al nativo
- Comunicación JS ↔ WASM (uso de wasm-bindgen)

Ejemplo de filtro en Rust: grayscale

...

```rust
#[wasm_bindgen]
pub fn grayscale(image_data: &[u8]) -> Vec<u8> {
    let image = image::load_from_memory(image_data).expect("Failed to load image");
    let gray = image.grayscale();
    let mut buf = Cursor::new(Vec::new());
    gray.write_to(&mut buf, ImageFormat::Png).expect("Failed to write image");
    buff.into_inner()
}
```

- atributo wasm_bindgen
- recibe imagén en bytes
- aplica filtro grayscale()
- crea buffer con la imagén procesada
- extrae los bytes desde el buffer y los devuelve para pasar a JS

Funcionalidades PWA

...

Personalización

Colores, icono, splash screen, etc.

Service workers

Permite que la app funcione offline cacheando archivos y respondiendo con mayor velocidad.

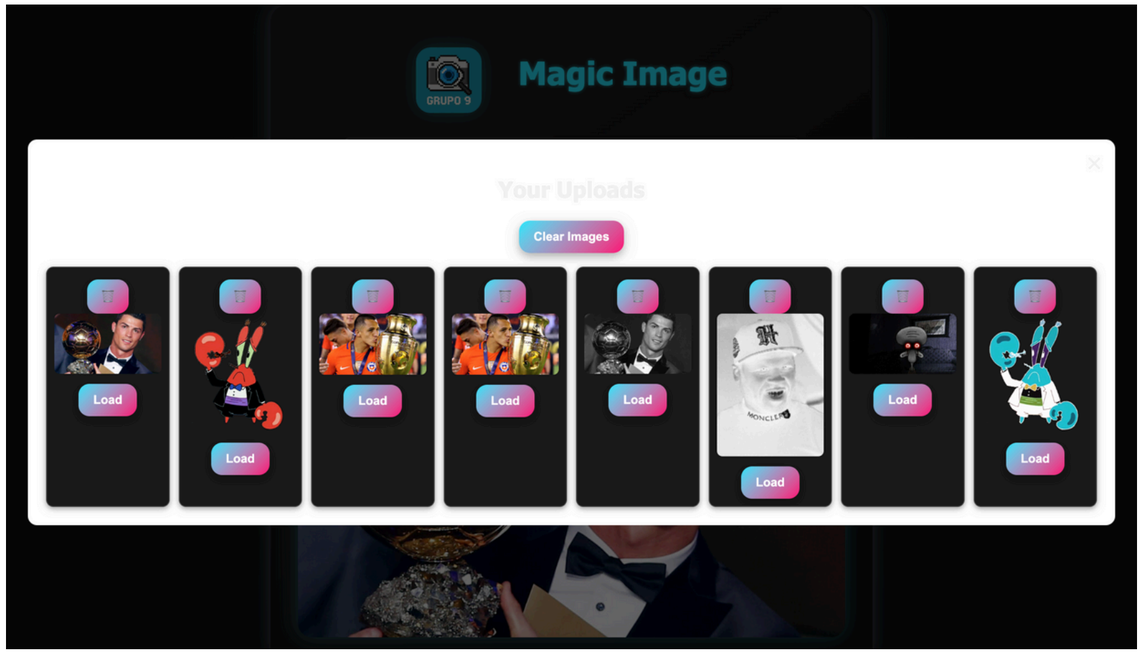
Guardar datos localmente en el navegador del usuario, incluso offline.

Utilizado para manejar las notificaciones push

Librería idb para IndexedDB

IndexedDB es una BDD no relacional. Permite guardar objetos de Javascript en grandes cantidades y con distintos tamaños.

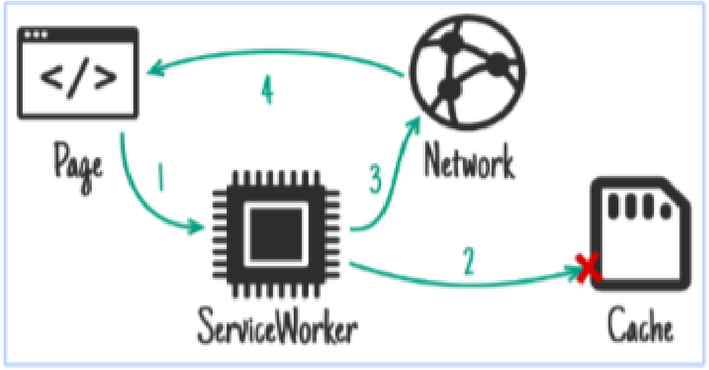
Guardado de imágenes y listas de filtros aplicados en tablas de IndexedDB. Esto da la posibilidad de cargar imágenes anteriores y revertir filtros individuales.



Evento de instalación de Service Worker

- Inicialización de cache para el funcionamiento del Service Worker
- Almacenamiento de data que se consulta a través de la URL

Funcionamiento Caché




```
self.addEventListener('install', (event) => {
    event.waitUntil(
      (async () => {
        try {
          const cache = await caches.open('static-v1');
          console.log('Service Worker: Caching Files');

          const base = '/2025-1-s1-g9-t2/';

          await cache.addAll([
            `${base}`,
            `${base}index.html`,
            `${base}app.js`,
            `${base}style.css`,
            `${base}camIcon192.png`,
            `${base}camIcon512.png`,
            `${base}camIcon512sinfondo2.png`,
            `${base}manifest.json`,
            `${base}iconLogo.png`
          ]);
        } catch (error) {
          console.error('Error caching files:', error);
        }
      })()
    );
  });
```

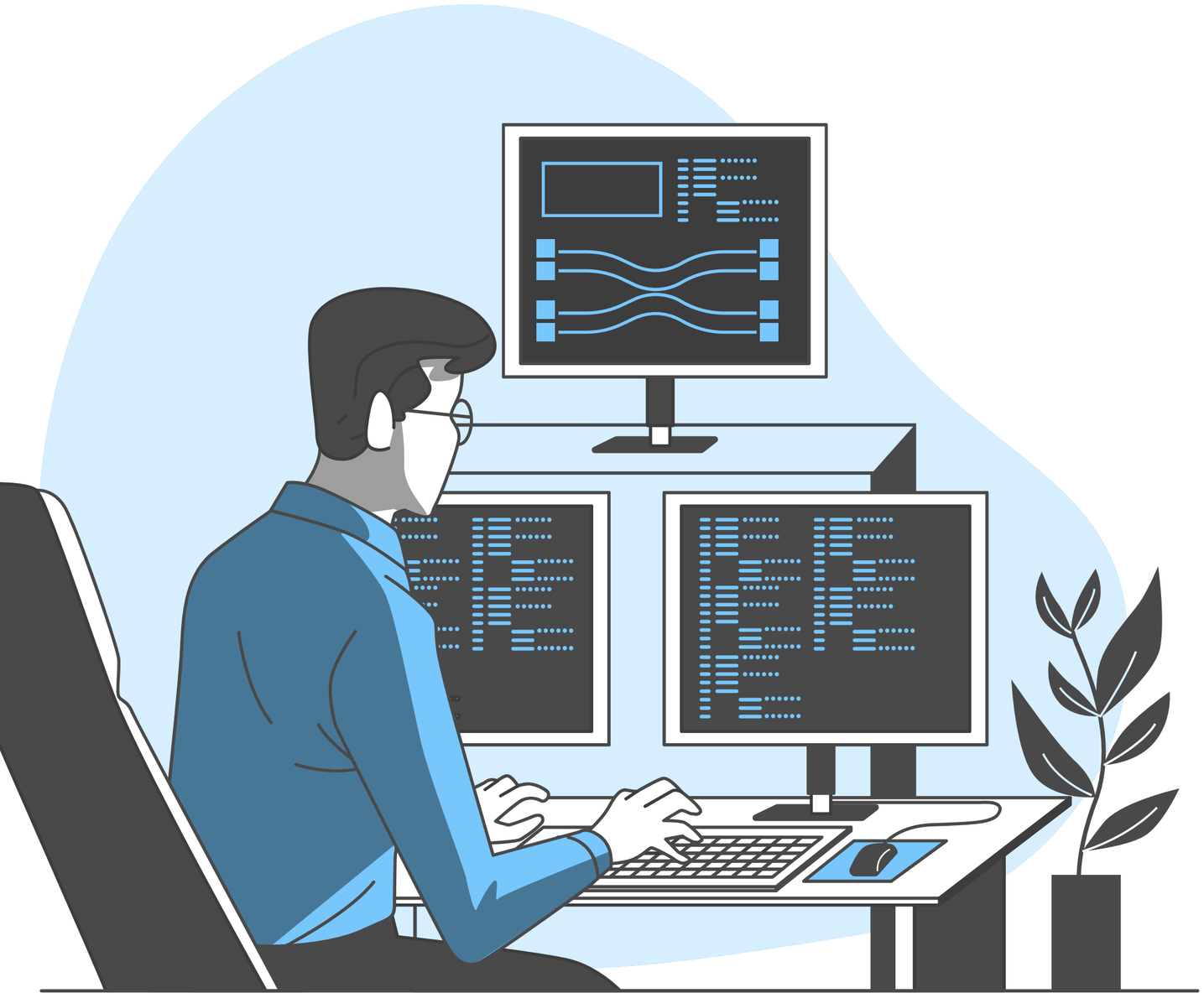
Firebase Cloud messaging

RequestPermission()

- Consulta y registra service worker para comunicación mediante notificaciones push
- Registra token de comunicación con servidor de Firebase

```
export function setupOnMessageHandler() {
  onMessage(messaging, (payload) => {
    console.log("Notificación en primer plano:", payload);

    if (Notification.permission === "granted") {
      new Notification(payload.notification.title, {
        body: payload.notification.body,
        icon: "/2025-1-s1-g9-t2/iconLogo.png",
      });
    }
  });
}
```



Muchas gracias!

Grupo 9