Aplicaciones Web Progresivas

13/05/2020

Grupo 2

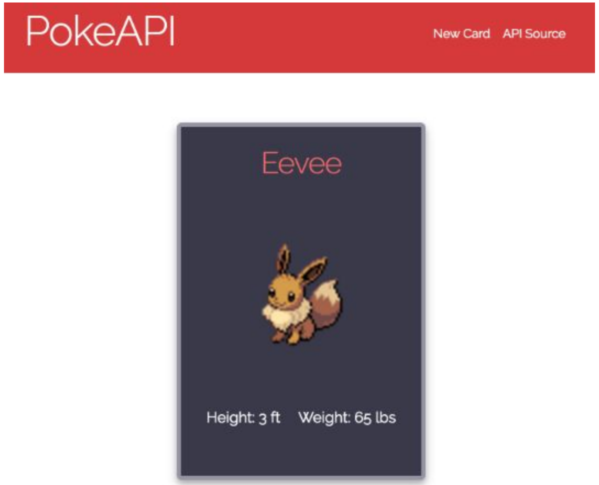
Paul Heinsohn
Sebastián Carreño

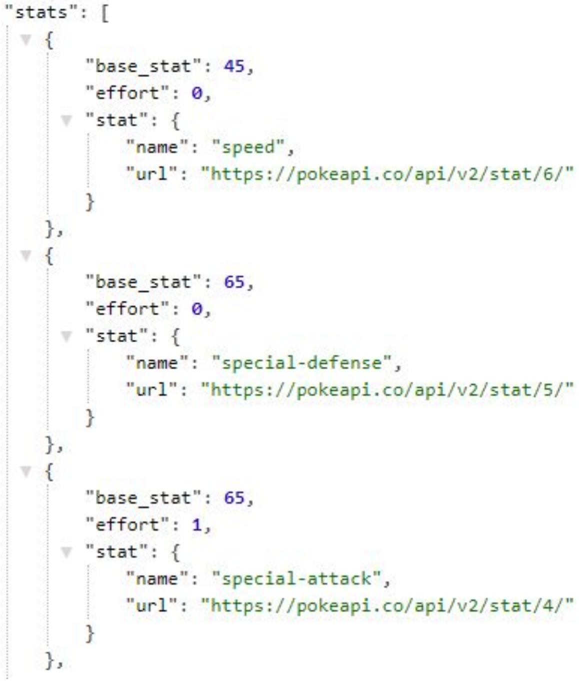
Profesor Jaime Navón

Pokémon Database DeMo

¿Por qué Pokémon?

- PokeAPI

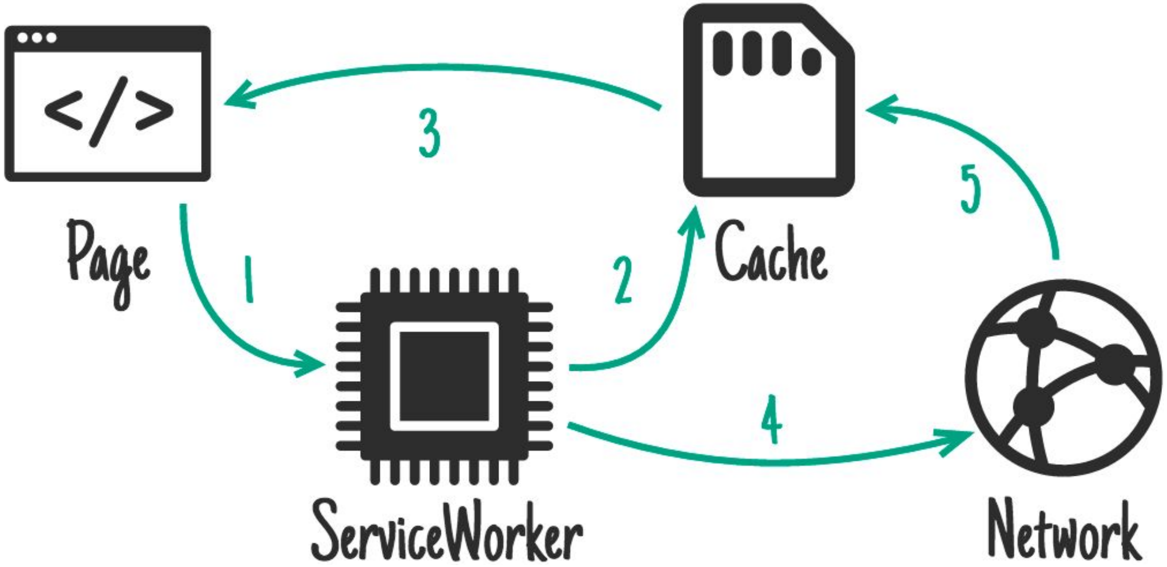




```
"stats": [
  {
      "base_stat": 45,
      "effort": 0,
      "stat": {
          "name": "speed",
          "url": "https://pokeapi.co/api/v2/stat/6/"
      }
  },
  {
      "base_stat": 65,
      "effort": 0,
      "stat": {
          "name": "special-defense",
          "url": "https://pokeapi.co/api/v2/stat/5/"
      }
  },
  {
      "base_stat": 65,
      "effort": 1,
      "stat": {
          "name": "special-attack",
          "url": "https://pokeapi.co/api/v2/stat/4/"
      }
  },
```

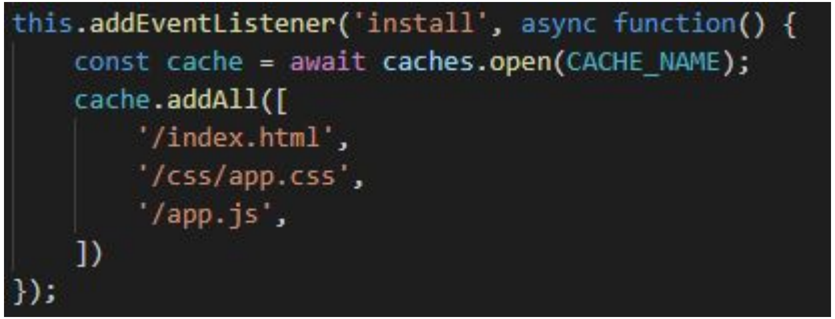
Función Offline

- Service Worker para caché



Implementación Caché

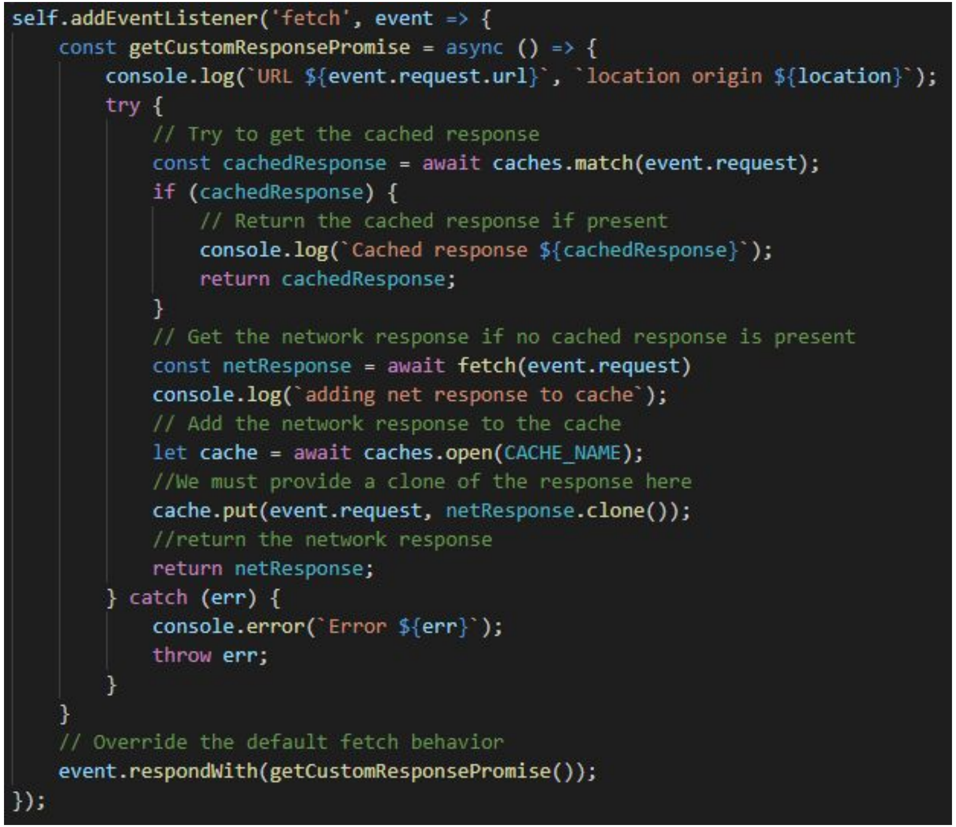
- Service Worker listener



```
this.addEventListener('install', async function() {
    const cache = await caches.open(CACHE_NAME);
    cache.addAll([
        '/index.html',
        '/css/app.css',
        '/app.js',
    ])
});
```

Implementación Caché

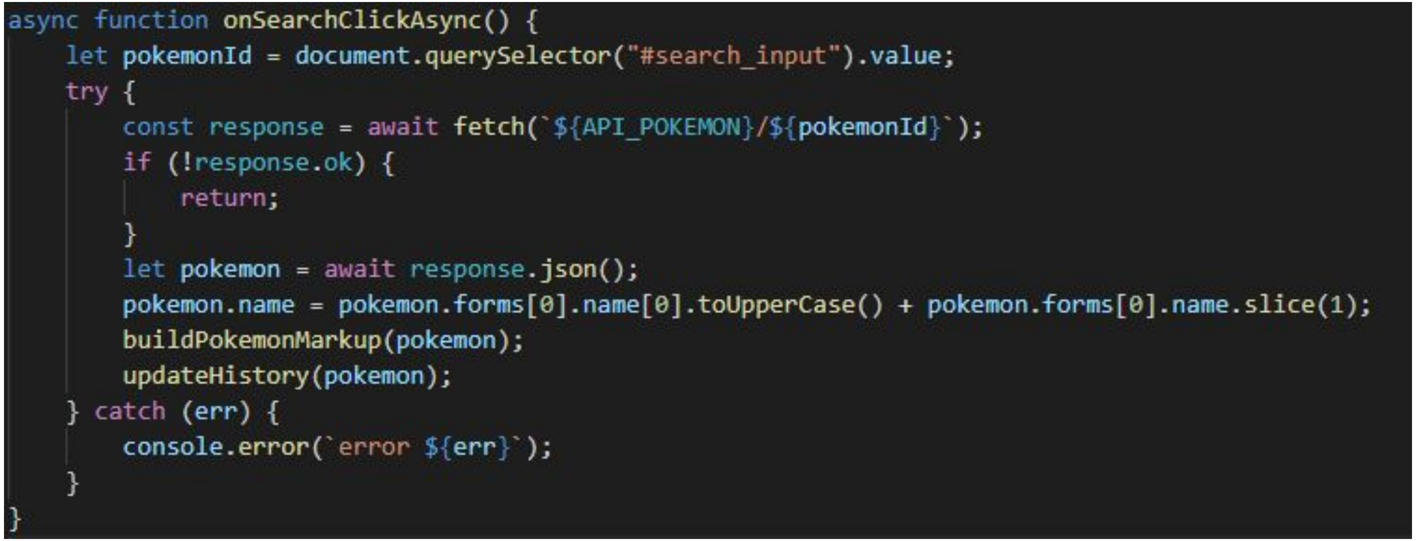
- SW fetch listener



```
self.addEventListener('fetch', event => {
    const getCustomResponsePromise = async () => {
        console.log(`URL ${event.request.url}`, `location origin ${location}`);
        try {
            // Try to get the cached response
            const cachedResponse = await caches.match(event.request);
            if (cachedResponse) {
                // Return the cached response if present
                console.log(`Cached response ${cachedResponse}`);
                return cachedResponse;
            }
            // Get the network response if no cached response is present
            const netResponse = await fetch(event.request)
            console.log(`adding net response to cache`);
            // Add the network response to the cache
            let cache = await caches.open(CACHE_NAME);
            //We must provide a clone of the response here
            cache.put(event.request, netResponse.clone());
            //return the network response
            return netResponse;
        } catch (err) {
            console.error(`Error ${err}`);
            throw err;
        }
    }
    // Override the default fetch behavior
    event.respondWith(getCustomResponsePromise());
});
```

Implementación Caché

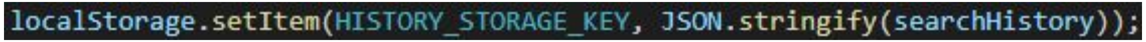
- Botón de búsqueda → fetch API



```
async function onSearchClickAsync() {
    let pokemonId = document.querySelector("#search_input").value;
    try {
        const response = await fetch(`${API_POKEMON}/${pokemonId}`);
        if (!response.ok) {
            return;
        }
        let pokemon = await response.json();
        pokemon.name = pokemon.forms[0].name[0].toUpperCase() + pokemon.forms[0].name.slice(1);
        buildPokemonMarkup(pokemon);
        updateHistory(pokemon);
    } catch (err) {
        console.error(`error ${err}`);
    }
}
```

Implementación Caché

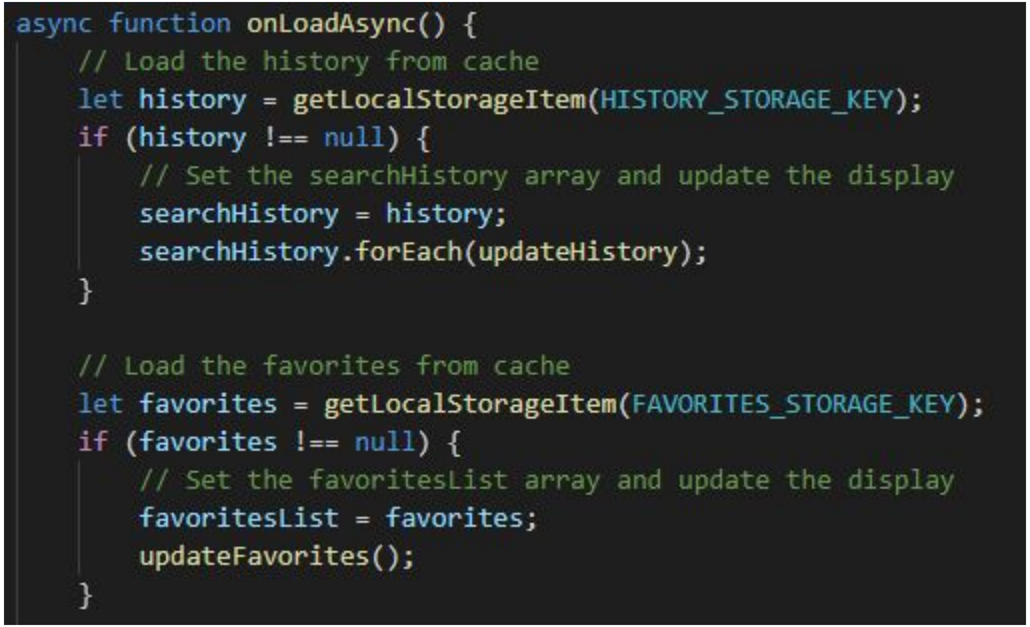
- Se almacena en local storage



```
localStorage.setItem(HISTORY_STORAGE_KEY, JSON.stringify(searchHistory));
```

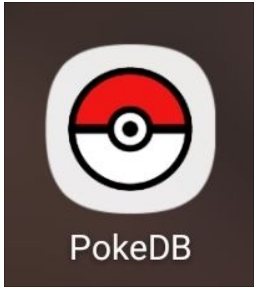
Implementación Caché

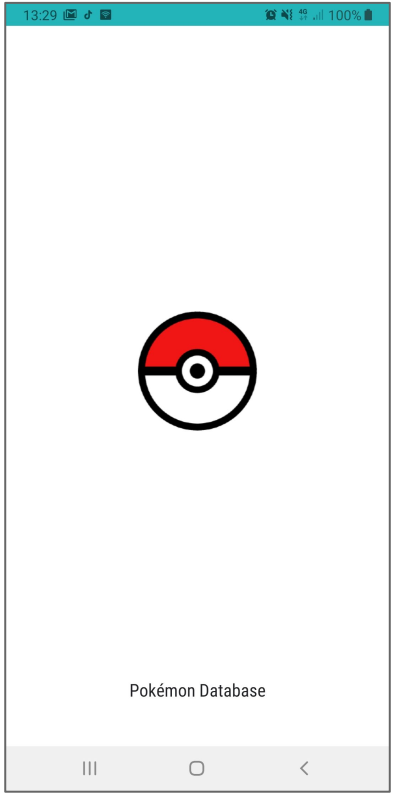
- Local storage persiste



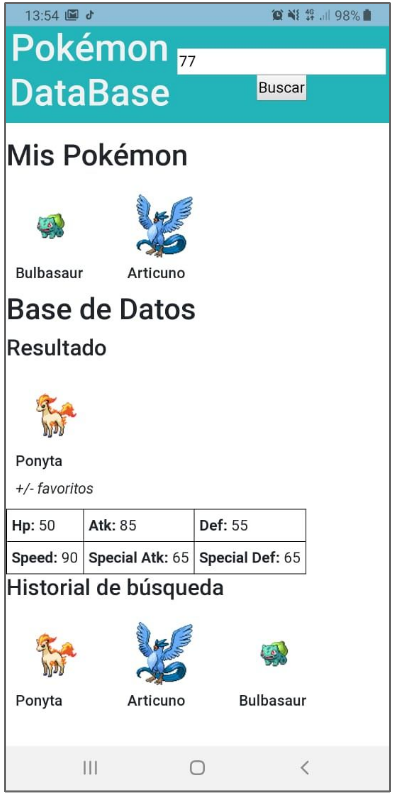
```
async function onLoadAsync() {
    // Load the history from cache
    let history = getLocalStorageItem(HISTORY_STORAGE_KEY);
    if (history !== null) {
        // Set the searchHistory array and update the display
        searchHistory = history;
        searchHistory.forEach(updateHistory);
    }

    // Load the favorites from cache
    let favorites = getLocalStorageItem(FAVORITES_STORAGE_KEY);
    if (favorites !== null) {
        // Set the favoritesList array and update the display
        favoritesList = favorites;
        updateFavorites();
    }
```

Home Page





Pokémon Database



Pokémon DataBase

Mis Pokémon

Bulbasaur Articuno

Base de Datos

Resultado

Ponyta

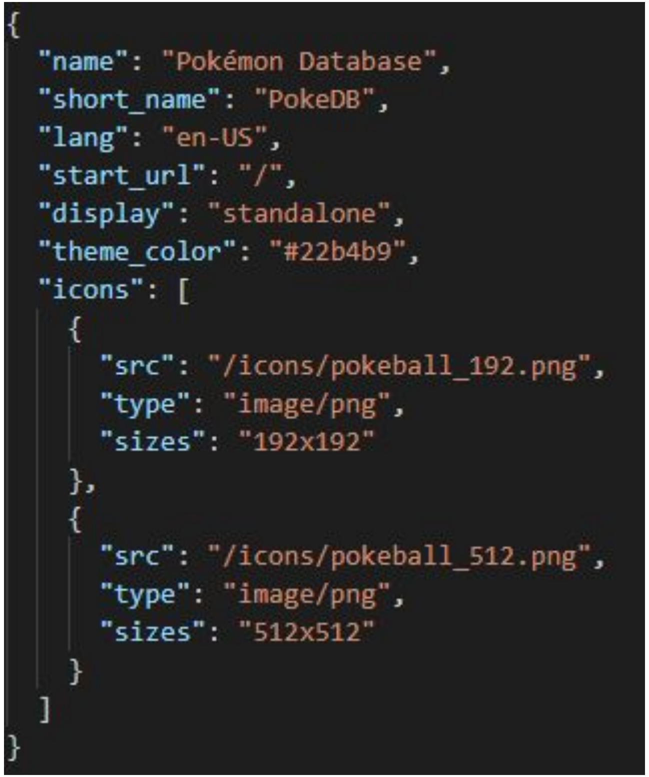
+/- favoritos

Hp: 50	**Atk:** 85	**Def:** 55
Speed: 90	**Special Atk:** 65	**Special Def:** 65

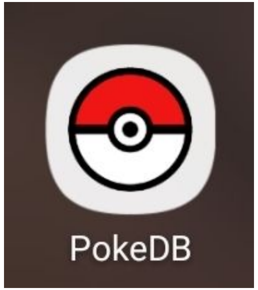
Historial de búsqueda

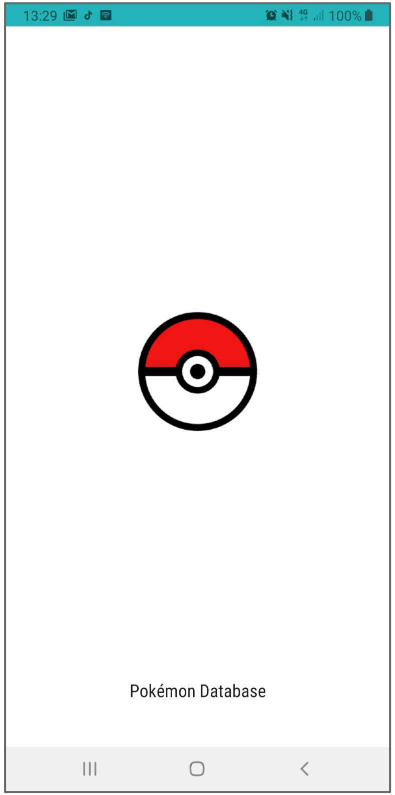
Ponyta Articuno Bulbasaur

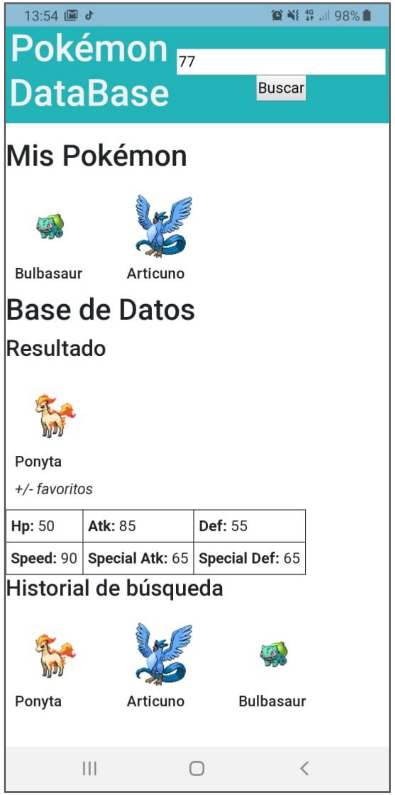
Manifest.json



```
{
  "name": "Pokémon Database",
  "short_name": "PokeDB",
  "lang": "en-US",
  "start_url": "/",
  "display": "standalone",
  "theme_color": "#22b4b9",
  "icons": [
    {
      "src": "/icons/pokeball_192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "/icons/pokeball_512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ]
}
```







Pokémon DataBase

Mis Pokémon

Bulbasaur Articuno

Base de Datos

Resultado

Ponyta

+/- favoritos

Hp: 50	**Atk:** 85	**Def:** 55
Speed: 90	**Special Atk:** 65	**Special Def:** 65

Historial de búsqueda

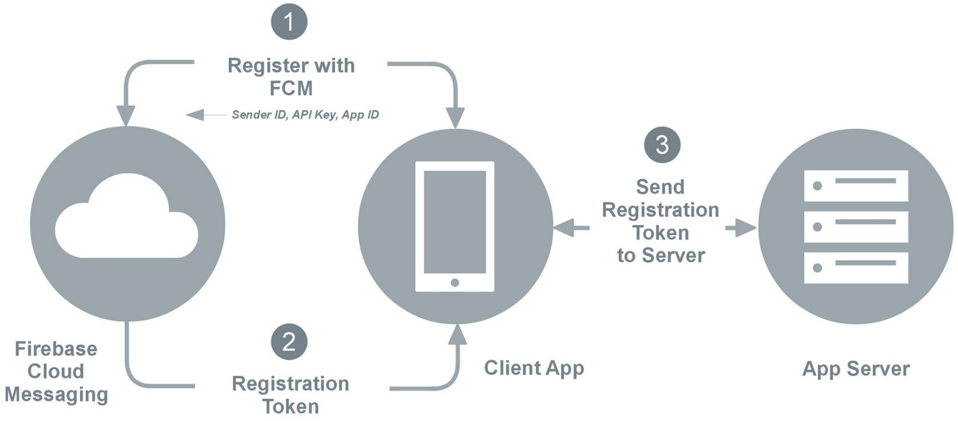
Ponyta Articuno Bulbasaur

Deploy con Firebase

- Firebase Hosting
- Sencillo
- Alta compatibilidad con los servicios que ofrecen

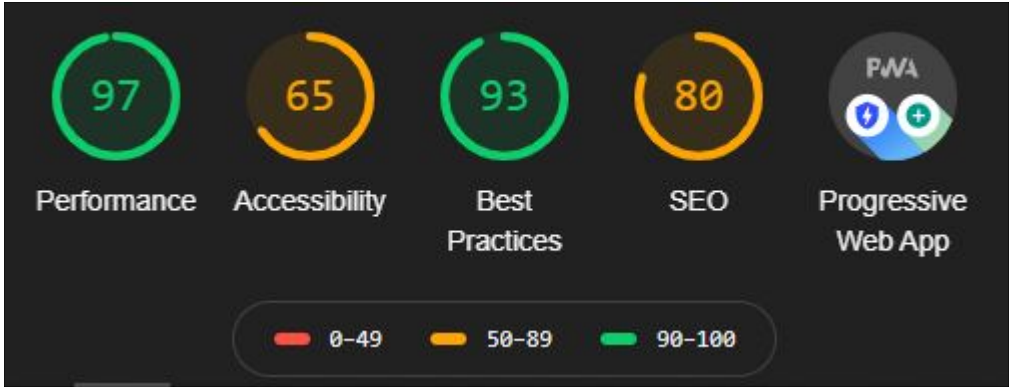
Notificaciones con Firebase

- Push API
- Firebase Cloud Messaging
- firebase-messaging-sw.js
    - Se despliegan los mensajes Push en el background
    - Se consigue el Registration Token del cliente



Anexos

- Desempeño Lighthouse


97
Performance
65
Accessibility
93
Best Practices
80
SEO
PWA
Progressive Web App
🟥 0-49
🟧 50-89
🟩 90-100