



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

TEMA: Avance de proyecto

PRESENTADO POR:

De La Mora Vazquez Victor Manuel

Noyola Barradas Juan Diego

Ruiz Alvarez Jose Eduardo

Toledo Perez Cristian Alejandro

Velazquez Gonzalez Jesus Alejandro

GRUPO: 10B

MATERIA: PWA

PROFESOR: Ray Brunett Parra Galaviz

Tijuana, Baja California, 14 de marzo del 2024

Investigación de recursos/Librerías para el diseño de la PWA en Laravel:

laravel-pwa:

Descripción: Laravel PWA es un paquete de Laravel desarrollado para ayudar a convertir aplicaciones web Laravel en PWAs fácilmente.

Características principales:

- Proporciona un archivo de manifiesto y un archivo de servicio para habilitar las funcionalidades de PWA.
- Puede ayudar a configurar la precarga de recursos y el almacenamiento en caché para mejorar el rendimiento.
- Facilita la instalación de la aplicación web como una aplicación independiente en dispositivos móviles y escritorios.
- Comunidad y soporte: Es importante revisar la comunidad y el soporte disponible para esta librería, incluyendo actualizaciones y mantenimiento.

PWA Builder:

Descripción: PWA Builder es una herramienta en línea que ayuda a generar un esqueleto básico para PWAs, incluyendo un manifiesto y un archivo de servicio.

Características principales:

- Proporciona una interfaz gráfica fácil de usar para generar los archivos necesarios para una PWA.
- Ofrece compatibilidad con diversas características de PWA, como notificaciones push y almacenamiento en caché.
- Es independiente del framework y se puede utilizar con cualquier aplicación web, incluyendo aquellas desarrolladas con Laravel.
- Flexibilidad y personalización: Aunque es fácil de usar, puede ser menos flexible en comparación con soluciones específicas para Laravel.

Workbox:

Descripción: Workbox es una biblioteca de JavaScript desarrollada por Google que facilita la creación de PWAs, especialmente en el manejo del almacenamiento en caché y la gestión de solicitudes.

Características principales:

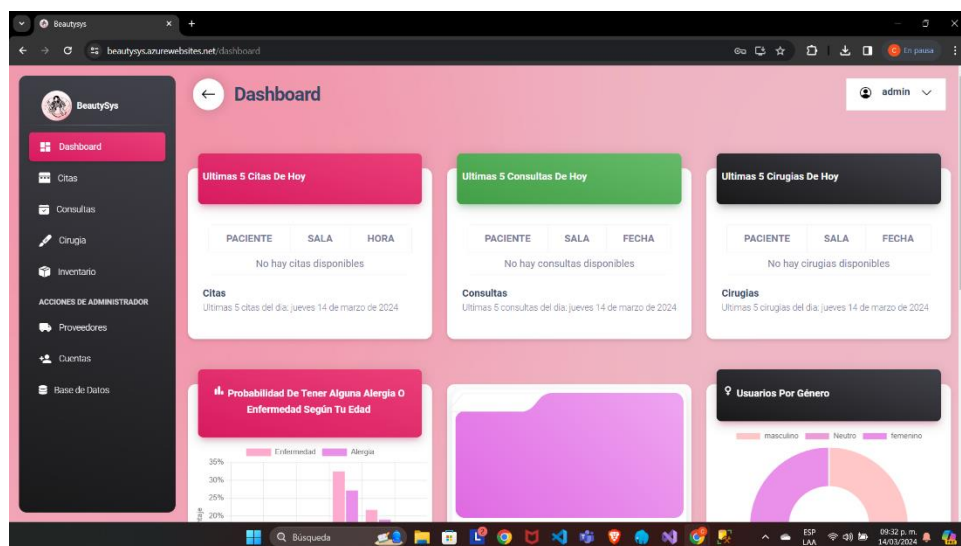
- Ofrece una amplia gama de estrategias de almacenamiento en caché para optimizar el rendimiento de la PWA.
- Proporciona herramientas para generar automáticamente el archivo de servicio necesario para la PWA.

- Es altamente configurable y se puede integrar fácilmente en aplicaciones Laravel existentes.
- Complejidad: Requiere un poco más de conocimientos técnicos para integrarlo correctamente en una aplicación Laravel, pero ofrece una gran flexibilidad y control.

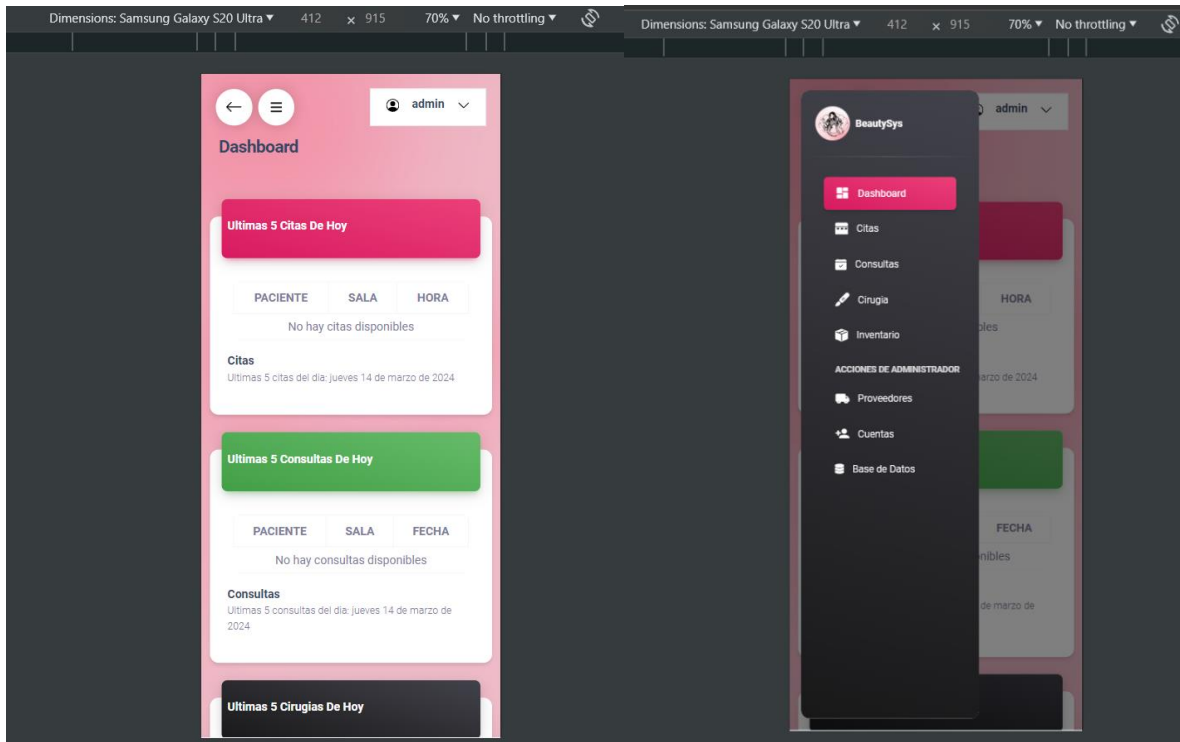
Ante estas opciones optamos por la implementación de laravel-pwa. en base a nuestros desarrollos, se tomaron en cuenta diferentes puntos y se llegó a la conclusión de que usar laravel-pwa simplifica la integración de características de PWA en Beautysys, aumenta su rendimiento y ofrece una experiencia de usuario más fluida, todo con el respaldo de una comunicación activa y soporte continuo.

Responsivo:

Durante la implementación de la PWA, se llevaron a cabo ajustes significativos en el diseño para garantizar la adaptabilidad de la aplicación a dispositivos móviles. Entre estos cambios, se incluyó la optimización del menú lateral originalmente diseñado para su visualización en escritorio.



Para abordar esta adaptación a dispositivos móviles, se implementó un botón específico que permite alternar la visibilidad del menú lateral según sea necesario.



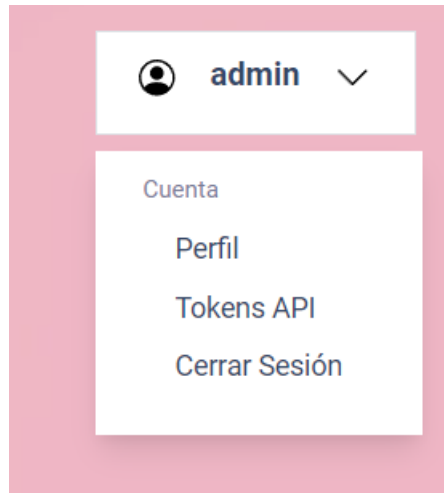
A continuación, se presenta el código utilizado para lograr esta funcionalidad de adaptación del menú lateral:

```

1 <main class="main-content position-relative max-height-vh-100 h-100 border-radius-lg">
2 <nav class="navbar navbar-main px-0 mx-2 shadow-none border-radius-xl" id="navbarBlur" data-scroll="true">
3 <div class="container-fluid py-1" style="display: inline-flex; justify-content: space-between;">
4 <div style="display: inline-flex; gap: 8px; align-items: center;">
5 <button id="backbutton" class="backbutton aside-menu" onclick="goBack();" type="button" style="width: 24px; height: 24px; border: 1px solid #000; border-radius: 50%; background-color: #f8f9fa; text-align: center; line-height: 24px; font-size: 0.8em; color: #000; cursor: pointer; margin-right: 10px;">
6 <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" fill="currentcolor" class="bi bi-arrow-left" viewbox="0 0 16 16">
7 <path fill-rule="evenodd" d="M15 8a.5.5 0 0 0-.5.5H2.5a2 2 0 0 0-2 2v6a2 2 0 0 0 2 2h9.5a.5.5 0 0 0 .5-.5" data-bbox="15 8 24 16" style="display: block; margin: 0 auto;"/>
8 </svg>
9 </button>
10
11 <div class="border-0 d-lg-none aside-menu backbutton" type="button" data-bs-toggle="offcanvas" data-bs-target="#sidenav-main" aria-controls="sidenav-main">
12 <svg xmlns="http://www.w3.org/2000/svg" width="16px" height="16px" data-bbox="15 8 24 16" style="display: block; margin: 0 auto;"/>
13 </div>
14 </div>
15
16 <div class="font-weight-bolder mb-0 px-2 d-none d-lg-block"><php echo session('activetab')></div>
17 </div>
18
19 <script>
20 function goBack() {
21 window.history.back();
22 }
23 </script>
24
25 <script>
26 document.addEventListener('DOMContentLoaded', function() {
27 var userRoles = @json(auth()->user()->roles);
28 var tabAdminStaff = document.getElementById("admin_dashboard");
29 var tabUser = document.getElementById("user_dashboard");
30
31 if (userRoles.includes('admin') || userRoles.includes('staff')) {
32 tabAdminStaff.click();
33 } else {
34 tabUser.click();
35 }
36 });
37 </script>
38
39 <div>
40 @livewire('navigation-menu')
41 </div>
42 </div>
43
44 <div class="font-weight-bolder mb-0 px-3 d-lg-none"><php echo session('activetab')></div>
45 </div>
46 @yield('content')
47 </main>

```

Asimismo, se adaptó el menú del perfil debido a que este tenía mal los estilos y las opciones salían mal acomodados



Manifest:

Al hacer uso de la librería laravel-pwa, esta nos genera dos archivos muy importantes: El archivo laravelpwa.php y el service worker. Laravelpwa.php es un archivo de configuración en el contexto de la biblioteca Laravel PWA. Se utiliza en proyectos de Laravel para definir la configuración de una aplicación web progresiva (PWA) y facilitar su implementación. Este archivo se encuentra típicamente en la carpeta config del proyecto Laravel.

La función principal del archivo laravelpwa.php es proporcionar un medio para configurar diferentes aspectos de la PWA, como el nombre de la aplicación, los iconos que se utilizarán, los colores de tema y las rutas que se deben cachear. Al definir estas configuraciones en el archivo laravelpwa.php, se facilita la personalización y adaptación de la PWA a las necesidades específicas de cada proyecto.

En cuanto a su función como manifest para el service worker, el archivo laravelpwa.php no actúa directamente como un manifest para el service worker, pero contribuye a la generación del archivo de manifiesto (manifest.json) que es esencial para el funcionamiento de una PWA y su service worker.

En nuestro proyecto hemos utilizado este archivo para definir características importantes de nuestra aplicación como lo son el nombre de la app, la descripción, iconos e imágenes a utilizar, atajos, etc.

Ps beautysys BEAUT-49-implementacion-pwa

laravelpwa.php x serviceworker.js app.php template.blade.php navigation-menu.blade.php show.blade.php web.php

```
1 <?php
2
3 return [
4     'name' => 'BeautySys Progressive Web App',
5     'manifest' => [
6         'name' => env('key: APP_NAME', default: 'BeautySys Progressive Web App'),
7         'short_name' => 'Beautysys App',
8         'start_url' => '/',
9         'background_color' => '#fff',
10        'theme_color' => '#00000000',
11        'display' => 'standalone',
12        'orientation' => 'any',
13        'status_bar' => '#fff',
14        'screenshots' => [
15            [
16                'src' => '/images/screenshots/landingPage.png',
17                'type' => 'image/png',
18                'form_factor' => 'wide',
19                'label' => 'Landing Page'
20            ],
21            [
22                'src' => '/images/screenshots/landingPage.png',
23                'type' => 'image/png',
24                'label' => 'Landing Page'
25            ]
26        ],
27        'description' => 'BeautySys: Sistema de Gestion y Administracion para clinica de cirugias esteticas.',
28        'icons' => [
29            '72x72' => [
30                'path' => '/images/icons/icon-72x72.png',
31                'purpose' => 'any'
32            ],
33            '96x96' => [
34                'path' => '/images/icons/icon-96x96.png',
35                'purpose' => 'any'
36            ],
37            '128x128' => [
38                'path' => '/images/icons/icon-128x128.png',
39                'purpose' => 'any'
40            ],
41            '144x144' => [
42                'path' => '/images/icons/icon-144x144.png',
43                'purpose' => 'any'
44            ],
45            '152x152' => [
46                'path' => '/images/icons/icon-152x152.png',
47                'purpose' => 'any'
48            ],
49            '192x192' => [
50                'path' => '/images/icons/icon-192x192.png',
51                'purpose' => 'any'
52            ],
53            '384x384' => [
54                'path' => '/images/icons/icon-384x384.png',
55                'purpose' => 'any'
56            ],
57            '512x512' => [
58                'path' => '/images/icons/icon-512x512.png',
59                'purpose' => 'any'
60            ]
61        ]
62    ]
63 ]
```

beautysys > config > laravelpwa.php

```

53         '384x384' => [
54             'path' => '/images/icons/icon-384x384.png',
55             'purpose' => 'any'
56         ],
57         '512x512' => [
58             'path' => '/images/icons/icon-512x512.png',
59             'purpose' => 'any'
60         ],
61     ],
62     'splash' => [
63         '640x1136' => '/images/icons/splash.jpg',
64         '750x1334' => '/images/icons/splash.jpg',
65         '828x1792' => '/images/icons/splash.jpg',
66         '1125x2436' => '/images/icons/splash.jpg',
67         '1242x2208' => '/images/icons/splash.jpg',
68         '1242x2688' => '/images/icons/splash.jpg',
69         '1536x2048' => '/images/icons/splash.jpg',
70         '1668x2224' => '/images/icons/splash.jpg',
71         '1668x2388' => '/images/icons/splash.jpg',
72         '2048x2732' => '/images/icons/splash.jpg',
73     ],
74     'shortcuts' => [
75         [
76             'name' => 'Dashboard',
77             'description' => 'Initial Dashboard',
78             'url' => '/dashboard',
79             'icons' => [
80                 'src' => '/images/icons/icon-96x96.png',
81                 'purpose' => 'any'
82             ]
83         ]
84     ],
85     'custom' => []
86 ]
87 ];
88

```



• beautysys > config > laravelpwa.php

Services Worker:

Principal mente se generó un archivo service worker para comunicar nuestro proyecto aun cuando existan percances de internet y este mismo tengo la funcionalidad offline.

Se define una variable llamada `staticCacheName`, que utiliza la fecha y hora actual para asegurar un nombre de caché único cada vez que se instala una nueva versión de la PWA. La variable `filesToCache` contiene una lista de recursos estáticos que se almacenarán en caché para su uso sin conexión.

Cuando el Service Worker se instala por primera vez, este evento se activa. Durante la instalación, se crea una nueva caché (`staticCacheName`) y se añaden los archivos estáticos especificados (`filesToCache`) a esta caché.

Posterior mente el evento 'activate' se activa cuando el Service Worker se convierte en el Service Worker activo. Aquí, se eliminan todas las cachés antiguas relacionadas con esta PWA, excepto la caché actual (`staticCacheName`). Esto garantiza que solo estén presentes los reursos más recientes en la caché.

Cuando se hace una solicitud de recursos, este evento se activa. El Service Worker intenta recuperar la respuesta de la caché. Si el recurso está presente en la caché, se devuelve; de lo contrario, se realiza una solicitud de red para obtener el recurso desde el servidor. Si la solicitud de red falla debido a la falta de conexión a Internet, se devuelve un archivo offline predefinido.


```

1  var staticCacheName = "pwa-v" + new Date().getTime();
2  var filesToCache = [
3    '/offline',
4    '/css/app.css',
5    '/js/app.js',
6    '/images/icons/icon-72x72.png',
7    '/images/icons/icon-96x96.png',
8    '/images/icons/icon-128x128.png',
9    '/images/icons/icon-144x144.png',
10   '/images/icons/icon-152x152.png',
11   '/images/icons/icon-192x192.png',
12   '/images/icons/icon-384x384.png',
13   '/images/icons/icon-512x512.png',
14 ];
15
16 // Cache on install
17 self.addEventListener("install", event => {
18   this.skipWaiting();
19   event.waitUntil(
20     caches.open(staticCacheName)
21       .then(cache => {
22         return cache.addAll(filesToCache);
23       })
24   );
25 });
26
27 // Clear cache on activate
28 self.addEventListener('activate', event => {
29   event.waitUntil(
30     caches.keys().then(cacheNames => {
31       return Promise.all(
32         cacheNames
33           .filter(cacheName => (cacheName.startsWith("pwa-")))
34           .filter(cacheName => (cacheName !== staticCacheName))
35           .map(cacheName => caches.delete(cacheName))
36       );
37     })
38   );
39 });
40
41 // Serve from Cache
42 self.addEventListener("fetch", event => {
43   event.respondWith(
44     caches.match(event.request)
45       .then(response => {
46         return response || fetch(event.request);
47       })
48       .catch(() => {
49         return caches.match('offline');
50       })
51   );
52 });

```