Grupo 17

Proyecto de Software

Extension trabajo integrador

Integrantes del trabajo

Luis Emanuel Banega 16462/3

Pablo Valero 15985/3

Índice:

Introducción:

El presente informe aborda el desarrollo de un software diseñado específicamente para un club deportivo, con el objetivo de proporcionar una plataforma integral que satisfaga las necesidades tanto de los asociados del club como de los trabajadores involucrados en su gestión. Este software se compone de dos partes fundamentales: una sección privada, destinada a los trabajadores, y una sección pública, accesible para todos los usuarios.

La sección privada del software fue desarrollada utilizando Flask, un framework de Python que permite construir aplicaciones web de manera eficiente y escalable. Esta parte del sistema se enfoca en dar control sobre la administración de asociados, disciplinas, etc, a los diferentes encargados del sistema.

Por otro lado, la sección pública del software se construyó utilizando Vue.js, un framework de JavaScript para construir interfaces de usuario interactivas, el cual decidimos montar sobre la plataforma Vite, la cual crea un proyecto vue con una configuración estándar. Esta sección tiene como propósito ofrecer información relevante sobre el club a visitantes y potenciales asociados, tales como horarios de apertura, actividades disponibles, noticias y eventos destacados.

Durante el proceso de desarrollo, se realizó una exhaustiva investigación sobre las PWAs, una tecnología que permite a las aplicaciones web brindar experiencias similares a las de las aplicaciones móviles nativas. Esto permitió implementar características como notificaciones push y funcionalidad offline, mejorando así la experiencia de los usuarios.

Asimismo, se llevó a cabo un riguroso proceso de pruebas y verificación de la accesibilidad del software. Se utilizó la herramienta WAVE para garantizar que la aplicación cumpla con los estándares de accesibilidad web, de manera que todos los usuarios puedan acceder y utilizar el software de manera inclusiva.

Además se aplicaron las buenas prácticas en privacidad de manera correcta.

En resumen, este informe detalla el proceso de desarrollo de un software para un club deportivo, abarcando tanto su parte privada como pública. Se utilizaron tecnologías como Flask y Vue.js, se exploraron las PWAs y se verificó la accesibilidad con la herramienta WAVE, y por último se tuvieron en cuenta las buenas prácticas en privacidad. A lo largo del informe, se describirán en detalle los diferentes aspectos del desarrollo, así como los resultados obtenidos y las posibles mejoras a considerar.

Palabras Claves:

PWAs (por sus siglas en inglés “*Progressive Web Apps*”),
APIs (por sus siglas en inglés “Application Programing Interface”),

PWA y Tests:

Las PWAs son aplicaciones web que utilizan APIs y funciones emergentes del navegador junto a una estrategia tradicional de mejora progresiva para ofrecer una aplicación nativa como la experiencia del usuario para aplicaciones web multiplataforma.

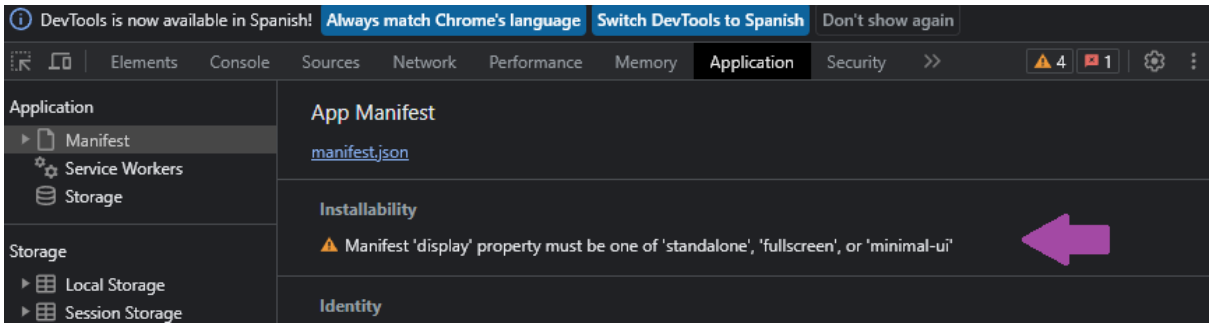
Estas PWA cuentan con una herramienta fundamental que son los Service Workers, los cuales son scripts que controlan la forma en que un navegador web maneja las solicitudes de red y el almacenamiento en caché de activos. Con la ayuda de ellos, los desarrolladores pueden crear páginas confiables y rápidas que también pueden funcionar sin conexión.

El Service Workers es realmente importante ya que nos permite optimizar la retención de los usuarios.

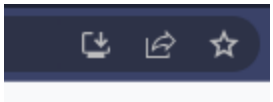
En nuestro caso, usamos la librería vite-plugin-pwa, la cual nos facilita el trabajo para nuestro proyecto montado en vite. Para realizar la configuración es necesario ir al archivo *vite.config.js*.

```
import { fileURLToPath, URL } from 'node:url'

import { defineConfig } from 'vite'
import vue from '@vitejs/plugin-vue'
import { VitePWA } from 'vite-plugin-pwa'

// https://vitejs.dev/config/
export default defineConfig({
  plugins: [vue(), VitePWA({
    manifest:{
      "name": "Club Socios App",
      "short_name": "Club",
      "icons": [
      {
          "src": "/img/icons/logo.png",
          "sizes": "any",
          "type": "image/png"
      }
      ],
      "start_url": "/",
      "display": "standalone",
      "background_color": "#ffffff",
      "lang": "en",
      "scope": "/"
  }
  })],
  resolve: {
    alias: {
      '@': fileURLToPath(new URL('./src', import.meta.url))
    }
  }
})
```

Como podemos ver, se importa la librería y se agrega en el array de plugins. Además, podemos configurar el archivo manifest, que va a contener la información relacionada con la aplicación, como puede ser el logo, el nombre, color de fondo, etc.
Cabe destacar que fue de gran ayuda la herramienta que dispone el navegador, la cual te irá indicando los errores que son necesarios corregir para que la aplicación se pueda instalar.



Finalmente, luego de realizar el proceso para transformar el proyecto en una PWAs, podremos observar que aparece en el navegador la opción para instalar la aplicación.



Los test son un proceso para verificar y validar la funcionalidad de un programa o una aplicación de software con el objetivo de garantizar que el producto de software esté libre de defectos.

En el proyecto, usamos la librería Pytest para el testing, la cual nos sirvió para poder realizar las pruebas sobre la api.

```python
def test_api_sport_by_id():
    response = client.get("/api/sport/1")
    assert response.status_code == 200
    json_sport_info = response.get_json()
    assert json_sport_info["division"] == "Sub 20"
    assert json_sport_info["instructors"] == "Carlos Gomez"
    assert json_sport_info["monthly_fee"] == "2000"
    assert json_sport_info["name"] == "Futbol"
    assert json_sport_info["schedule"] == "Lunes 15 a 16, Miercoles 14 a 15"
```

Aca podemos observar un ejemplo, donde se hace el request al endpoint, luego se chequea el status code, y finalmente se validan los datos del json de la respuesta.

Los tests se encuentran dentro de la carpeta *tests*, en el archivo *test_api.py*

Análisis de accesibilidad:

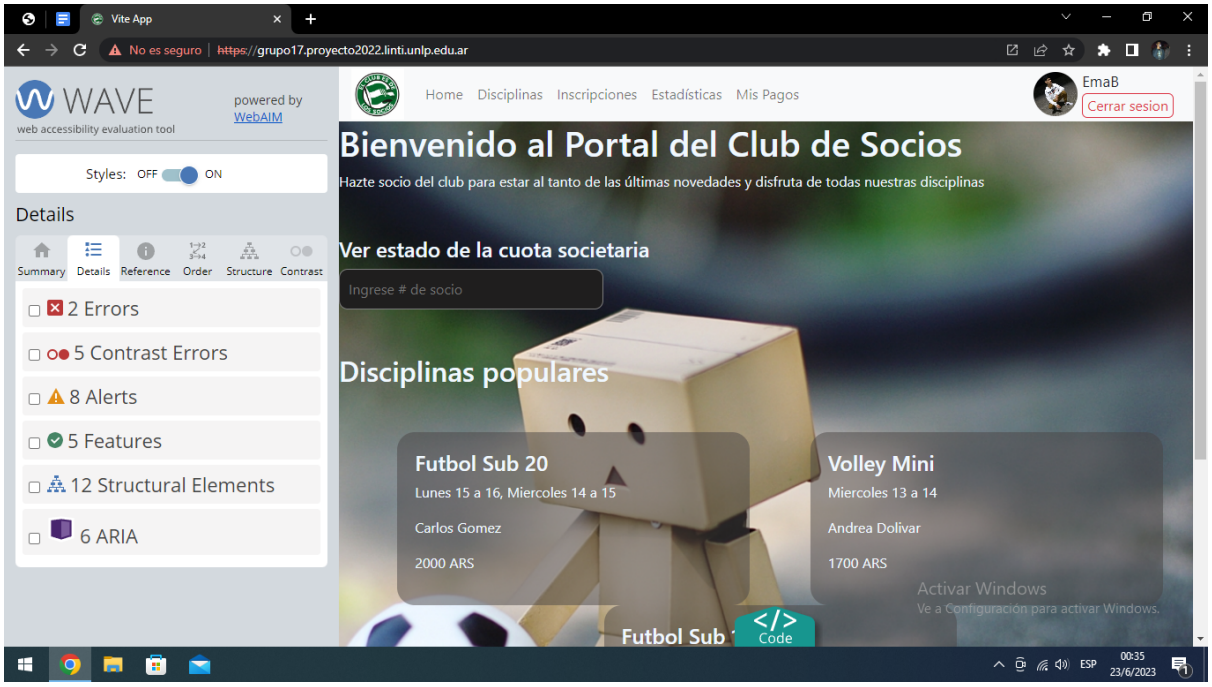
Tiene por fin analizar, estudiar y validar las páginas web con el objetivo de que no presenten problemas de accesibilidad y cumplan las pautas y directrices de accesibilidad existentes.

La evaluación de accesibilidad web se puede realizar de forma automática o manual. Para ello se emplean diversas herramientas de revisión.

La evaluación automática nos permite realizar una evaluación rápida, nos ayuda a tener una primera impresión de la accesibilidad de una página web, pero no proporciona un análisis definitivo y fiable, ya que puede no detectar errores importantes o señalar errores que realmente no existen (falsos positivos).

Solamente un análisis manual puede ofrecer un análisis completo y fiable de la accesibilidad web de una página.

Nosotros hicimos la evaluación de accesibilidad gracias al validador WAVE Evaluation tool y en líneas generales el proyecto respetaba bien las condiciones.



Este es un ejemplo de una de las vistas de la app pública como es el home, podemos apreciar distintas secciones donde nos detallaran que es lo que pasa como errores, alertas, etc.

En este caso el resultado fue positivo ya que las alertas son opcionales y los 2 errores que se encontraron no eran relevantes con el funcionamiento para el público en general.

Analisis y reflexion:

Si hablamos de las buenas prácticas en privacidad, nuestra aplicación sigue un diseño seguro en cuanto a calidad, confidencialidad, etc.

Algunos ejemplos claros pueden ser el uso de autenticación por token, la encriptación de todas las credenciales, entre otros.

Aun así, nos quedan pendientes algunos principios de privacidad como es el tema del consentimiento o finalidad, ya que en ningún momento mostramos un mensaje indicando al usuario cuál es el objetivo de la recolección de sus datos personales.

En conclusión durante todo el proceso de desarrollo, se tuvieron en cuenta los aspectos de privacidad y se buscaron integrar buenas prácticas en este ámbito. Se aplicaron medidas de seguridad para proteger la información personal de los asociados y trabajadores del club, asegurando que solo aquellos con los permisos adecuados pudieran acceder a ciertos datos

sensibles. Se implementaron también controles de acceso y autenticación para garantizar la privacidad y evitar el acceso no autorizado.

Resumen:

El objetivo de este proyecto fue mejorar la funcionalidad y la calidad de un sistema existente mediante la incorporación de nuevas características. En primer lugar, se implementaron PWAs sobre la aplicación Vue, lo que permitió que la aplicación fuera accesible desde múltiples dispositivos y plataformas, brindando a los usuarios una experiencia más fluida y envolvente.

Además, se llevó a cabo una fase de pruebas exhaustivas utilizando Pytest, un marco de pruebas en Python. Esto garantizó la estabilidad y el correcto funcionamiento de las nuevas funcionalidades agregadas, así como la detección temprana de posibles errores o fallos en el sistema.

Para asegurar la accesibilidad del sistema, se realizó un análisis exhaustivo utilizando Wave Evaluation Tools. Este proceso permitió identificar y abordar barreras de accesibilidad, mejorando así la experiencia de uso para usuarios con discapacidades o dificultades en la navegación.

También se llevó a cabo un análisis detallado de las buenas prácticas en privacidad, asegurándose de que el sistema cumpliera con los estándares y regulaciones actuales de protección de datos.

Conclusión:

Este proyecto logró el objetivo de incorporar nuevas funcionalidades al sistema, incluyendo PWA con Vite y tests con Pytest, y además se mejoró la accesibilidad. Estas mejoras aseguraron una experiencia de usuario más completa, confiable y accesible.