

ENTORNOS WEB

TRABAJO FINAL



UCEL

UNIVERSIDAD DEL CENTRO EDUCATIVO
LATINOAMERICANO

Aressi, Juan

Descripción de la página

La página (actualmente) está compuesta por la página principal, la cual contiene información sencilla sobre la clínica y un formulario de contacto, y además un módulo de Log In para los distintos tipos de usuarios.

En un futuro, la idea sería implementar un módulo para que los pacientes puedan agendar sus turnos con los médicos.

A su vez, tiene un módulo de gestión, el cual está orientado a los administradores de la clínica, en donde pueden gestionar los pacientes, y en un futuro realizar la implementación para gestionar los médicos y los turnos. También se desea implementar un módulo similar para los médicos, en donde puedan gestionar sus turnos, días disponibles, etc.

Este sitio no está dirigido ni desarrollado para una clínica particular, sino que se hizo de un modo genérico para que pueda ser adaptado a un caso real.

Lista de objetivos

El objetivo principal es poder conocer y explorar a fondo las tecnologías que se están implementando en el desarrollo de esta página, y poder aprender a utilizarlas de la mejor manera posible, a lo que se le añade el objetivo de cumplir los requisitos de la materia de Entornos Web.

Usuarios a los cuales está destinado

La página está destinada a los pacientes de la clínica, a los médicos, y a los administradores de la clínica. Cada uno con sus propias funcionalidades.

Lista de funcionalidades que se van a implementar a futuro

En un futuro se desea implementar la funcionalidad de que los pacientes puedan agendar sus turnos con los médicos.

Que los médicos puedan gestionar sus turnos, días disponibles, etc.

Que los administradores puedan gestionar los médicos, pacientes y turnos.

Manejo de los estándares y como los manejamos sobre frameworks

Con respecto a los estándares de programación, para PHP se siguió el PSR2.

Con respecto a la validación de la estructura de la página, al ser una aplicación realizada en React.js, no existe una herramienta que valide con los estándares impuestos por W3C. Lo más cercano que pude realizar fue, utilizando la herramienta de desarrollo de Chrome, copié el contenido generado y lo pasé a la herramienta de validación de W3C, y me arrojó ciertos errores a los que se les realizó la corrección correspondiente.

Accesibilidad en la pagina

La página posee un alto contraste entre los textos y los fondos, lo que facilita la lectura de los mismos.

Los textos están en tamaño de letra adecuado, y se puede aumentar o disminuir el tamaño de la letra.

SEO

No he podido utilizar una herramienta para el análisis en etapas de desarrollo, pero en el caso que la página se encuentre en producción, hay una gran variedad de herramientas que permiten analizar la página y ver si cumple con los estándares de SEO.

Por mi parte he intentado seguir los estándares de SEO, como, por ejemplo, utilizar etiquetas semánticas, utilizar el atributo 'alt' en las imágenes, utilizar el atributo 'title' en los links, etc.

Optimización

Las páginas creadas con React.js, son optimizadas por defecto a la hora de ser compiladas, ya que se eliminan los espacios en blanco, se eliminan los comentarios, se minifican todos los archivos .css y .js en archivos únicos, etc.

Hosting y posible deploy

La página no cuenta con un hosting, ya que se encuentra en desarrollo, pero en el caso que se desee implementar, se ha analizado la posibilidad de utilizar Heroku, ya que es una plataforma que permite el deploy de aplicaciones web de manera gratuita.

En el caso que el proyecto se convierta en algo real, se considerará la opción de comprar un hosting, ya que la página no es muy pesada, y no requiere de un hosting muy potente.

Testing crossbrowser

Se hizo el testeo manual de las distintas funcionalidades de la página, en los navegadores Chrome, Firefox, Edge y Safari, y no se encontraron inconvenientes.

Se me ha recomendado una aplicación que no la utilicé, llamada BrowserStack, la cual permite testear la página en distintos navegadores y distintos dispositivos, y de esta manera poder detectar posibles inconvenientes.

Tecnologías utilizadas

Para el desarrollo de la página se utilizó React.js, ya que es una tecnología que permite crear aplicaciones web de manera rápida y sencilla, y que además permite crear Single Page Applications, lo que permite que la página sea más rápida y que no se recargue cada vez que se navega por ella.

Aquí también se implementaron herramientas adicionales, como jQuery, Bootstrap, paquetes de React.js, etc.

Para el backend se utilizó Laravel, el cual es un framework que permite crear APIs REST de manera dinámica y sencilla, lo cual facilita el desarrollo de la página y la comunicación con el frontend.

Para la base de datos se utilizó MySQL, ya que el sitio, la gestión de usuarios y turnos se consideró desde un primer momento en tablas relacionales, y es la única con la que he trabajado y sé manejar.

Guía de Instalación

Los requerimientos para instalar el proyecto son:

- Node.js 14.17.0 o superior
- npm 6.14.13 o superior
- git 2.30.1 o superior

- docker 20.10.17 o superior

Estas versiones son las que se han usado para el desarrollo del proyecto. Puede que funcione con versiones anteriores y/o posteriores, pero no se ha probado.

Paso a paso:

- 1) Clonar el repositorio localmente
`git clone git@github.com:JuanAressi/Proyecto-Final.git`
- 2) Instalar las dependencias
`cd Proyecto-Final/front`
`npm install`
- 3) Levantar los contenedores de docker
`docker-compose up -d`
- 4) Abrir una terminal en el contenedor 'proyecto-final_web'
`composer install`
`php artisan migrate:fresh --seed`

Ya se podría navegar el sitio en el puerto 3000 de localhost (<http://localhost:3000/>)
 Puede tardar unos minutos en cargar las primeras veces.

Si no carga, entrar en la consola y si tira errores de que no puede cargar los componentes:

- `docker-compose down`
- `docker system prune`
- `cd Proyecto-Final/front`
- `rm -rf node_modules`
- `npm install`
- `cd ..`
- `docker-compose up -d`

Datos de acceso de usuarios:

Administrador:

- Usuario: `admin@xmail.com`
- Contraseña: `admin`

Paciente:

- Usuario: usuario.prueba@xmail.com
- Contraseña: `Contraseña1`

Captura de las interfaces y mensajes de validación







