USABILIDAD:

1.- El proyecto cuenta con un diseño intuitivo y amigable en cualquier tipo de dispositivos, desarrollado bajo el principio de First Mobile y aprovechando todas sus ventajas para brindarle una grata experiencia al usuario lograda en bajos tiempos de desarrollo.

2.- Fueron aplicados los principios de la accesibilidad en el desarrollo de software para que, a pesar de su pequeña dimensión y limitantes, pueda ser usado por personas con discapacidades que usan tecnologías de asistencia, es por ello el uso impecable de atributos aria en todo el código HTML, títulos de imágenes, botones e inputs, …

3.- Se implemento un manejo de errores sencillo pero eficiente para mitigar los errores y la frustración de los usuarios.

4.- Se mantiene una retroalimentación efectiva con el usuario para brindarle en todo momento la información y acompañamiento necesarios para que pueda llevar a cabo las funciones brindadas por el sistema.

SEO:

1.- Se realizo un uso de palabras claves relevantes en títulos, descripciones y contenido.

2.-El sistema es Mobile-friendly.

3.-Se implementó Lazy Loading para mejorar los tiempos de carga.

4.Las imágenes fueron optimizadas usando TinyPNG.

PERFORMANCE:

1.-Se tomaron las medidas para reducir el tiempo que tarda el sitio en mostrarse completamente.

2.-Se realizo una búsqueda para garantizar que no existen recursos bloqueantes del renderizado.

3.-La interacción con el usuario es rápida y efectiva.

4.-El código CSS y JavaScript fueron minificados.

ESCALABILIDAD:

1.- En Pro de la Escalabilidad se le dio al proyecto la siguiente estructura:

-> src:

🡪 components

🡪 images

🡪 redux

🡪 routes

🡪 services

🡪 test

🡪 views

La cual nos permite aprovechar las ventajas de trabajar con componentes que nos ofrece React, segmentando el código de manera eficiente y precisa para reutilizarlo y hacer que crezca en el momento en que se requiera sin complicaciones. Además, separando en secciones muy representativas toda la lógica del proyecto para que le resulte sencillo de comprender y ubicarse a cualquier desarrollador, y con estas separaciones estratégicas se logra que a la hora de mantenimiento o extensión del código que el o los desarrolladores tengan archivos puntuales con los que interactuar para cada tarea especifica que deban realizar, eliminando así los riesgos que conllevan los coloquialmente llamados códigos espaguetis que no existen en este proyecto ya que fue desarrollado bajo el principio de DRY (Don’t Repeat Yourself).

2.- Fue implementado redux para el manejo de estados, si bien es cierto el proyecto es muy pequeño y se pudo llevar el manejo del estado de alguna manera mas sencilla como usando el Context u otra librería, se quería mostrar el dominio de redux, y se pretendía dejar implementado para tener un gran paso adelantado para cuando se desee agrandar el proyecto.

3.- Se implemento el uso del preprocesador SASS para hacer uso de sus ventajas a la hora de declarar variables, mixins y manejo de medias querys, para de este modo a la hora de requerir algún cambio o darle continuidad al código se pueda realizar dichos ajustes de manera muy rápida y entendible.