

# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

## FACULTAD DE ARTES Y DISEÑO

DIVISIÓN DE EDUCACIÓN CONTINUA Y EXTENSIÓN ACADÉMICA

2024-1

### MEMORIA DE INVESTIGACIÓN - PRODUCCIÓN

---

DIPLOMADO

QUE PRESENTA

PROYECTO

RESPONSABLE ACADÉMICO

ASESOR METODOLÓGICO

CONCLUSIÓN DE MEMORIA/FIRMA DE AUTORIZACIÓN DEL RESPONSABLE ACADÉMICO



UNAM  
FACULTAD  
DE ARTES  
Y DISEÑO



DIVISIÓN  
DE EDUCACIÓN CONTINUA  
Y EXTENSIÓN ACADÉMICA



REDEC UNAM

## Resumen.

Empresa de ajuste a nivel global busca solucionar un problema recurrente en su trabajo diario, el cual consiste en que cada siniestro<sup>1</sup> que le es encomendado requiere, por lo menos, que una persona viaje al lugar en donde ocurre el incidente<sup>2</sup>, los incidentes que maneja la empresa son diversos y van desde una simple reparación de un contenido<sup>3</sup> hasta edificaciones. El traslado al punto indicado requiere de gastos en transporte, hospedaje, alimentación entre otros. Derivado de lo anterior se quiere dar la siguiente solución.

La creación de una aplicación web<sup>4</sup> y también móvil para el registro de los daños. El principal objetivo de esta será que el asegurado<sup>5</sup> pueda registrar sus daños y subir evidencia fotográfica de lo ocurrido, el proceso se describirá más adelante.

Se busca que la aplicación sea agradable y de fácil uso para el usuario final, para lograrlo se utiliza la metodología “*Design Thinking*”<sup>6</sup> vista en el presente diplomado, ayudará a facilitar el proceso de análisis y construcción de diseños que ayuden al objetivo principal.

El diseño principal enfatiza el perfil de la empresa a la que es desarrollada, la elección de elementos visuales también tiene coherencia y la estética de esta, mientras que la sección de experiencia del usuario incorpora principios de UX y técnicas de cómo se trabaja en el ámbito específico de las actividades de la empresa.

---

<sup>1</sup> Caso: acontecimiento que, por causar unos daños concretos previstos en la póliza, motiva la aparición del principio indemnizatorio

<sup>2</sup> Incidente: evento o exposición repetida a determinadas condiciones, que produce una lesión durante la vigencia de la póliza

<sup>3</sup> Contenido: bienes muebles propiedad del Asegurado

<sup>4</sup> Aplicación web: software que se ejecuta en el navegador web Le permiten acceder a funcionalidades complejas sin la necesidad de instalar o configurar un software

<sup>5</sup> Asegurado: Es la persona natural o jurídica, que está expuesta al riesgo que se asegura.

<sup>6</sup> Design Thinking: herramientas y procedimientos útiles para abordar el diseño centrado en el usuario y la creación de interfaces web efectivas

En cuanto a los aspectos técnicos, las tecnologías utilizadas son el uso del React<sup>7</sup>, HTML<sup>8</sup>, CSS<sup>9</sup> y JavaScript<sup>10</sup>, Para la base de datos, se utiliza un sistema de tipo NoSQL<sup>11</sup>

En resumen, este proyecto encapsula el proceso de investigación, diseño y desarrollo necesario para la creación de la aplicación web, subrayando la importancia de la simplicidad, la usabilidad y la adaptabilidad para lograr fortalecer el proceso de trabajo de la empresa anteriormente mencionada.

---

<sup>7</sup> React: biblioteca para interfaces de usuario web y nativas

<sup>8</sup> HTML: lenguaje de marcado estándar utilizado para crear y diseñar páginas web

<sup>9</sup> CSS: lenguaje de estilo que se utiliza para describir la presentación de un documento HTML

<sup>10</sup> JavaScript: lenguaje de programación de alto nivel que se utiliza para agregar interactividad y dinamismo a las páginas web

<sup>11</sup> NoSQL: son bases de datos que almacenan datos en un formato diferente al de las tablas relacionales (SQL).

## Índice

I. Introducción.....	4
II. Planteamiento del Problema. ....	5
III. Objetivo.....	6
IV. Marco Teórico.....	6
V. Desarrollo de Proyecto. Diseño Web.....	8
A. Objetivo Smart.....	8
B. Moodboard .....	8
C. Estructura del sitio.....	9
D. Wireframe.....	10
E. Wireflow.....	13
VI. Desarrollo del Proyecto. Desarrollo Web.....	16
A. Herramientas por utilizar durante el desarrollo .....	16
B. Instalaciones Necesarias.....	16
C. Programación del Maquetado.....	27
D. Resultados finales. ....	34
VII. Conclusiones.....	41
VIII. Referencias. ....	42

## I. Introducción.

En las tecnologías que usamos en la actualidad encontramos como un factor común el uso de múltiples herramientas digitales para hacer más productiva la vida diaria, no solo en la vida cotidiana sino en el aspecto laboral, una de las más comunes son las aplicaciones web que, a diferencia de una página web ordinaria (estática), nos da una mayor utilidad para poder hacer múltiples tareas. No basta solo con tener una herramienta funcional, en nuestros días la experiencia de usuario (UX<sup>12</sup>) es vital para que el uso de ellas no sea un obstáculo más, sino por el contrario, ayude a que la eficiencia<sup>13</sup> y la eficacia<sup>14</sup> sean un hecho, adicionalmente a esto la interfaz de usuario permite que mientras se utilice la dicha herramienta, el usuario tenga una agradable experiencia al usarla y quiera seguir usándola en cualquier momento.

Moverse al uso de las tecnologías de información para las empresas ha sido una constante, conforme pasan las décadas, las tareas en el trabajo han sido afectadas para bien, tanto en la rapidez como en el aspecto financiero. No solo las empresas tecnológicas optan por usar canales de información digital, sino que, las empresas que su ramo no es directamente lo tecnológico, también han optado por elegir estas herramientas para ayudarse en las necesidades que cada una tiene. Es por esto por lo que una empresa de ajuste<sup>15</sup> ha pedido la realización de una aplicación web y/o móvil que resuelva aspectos específicos en el proceso de ajuste, en la cual quiere tener un beneficio a gran escala.

A lo largo de esta memoria de investigación, se detalla minuciosamente el proceso mediante el cual se abordan las especificaciones de la empresa en cuestión. A medida que avanzamos en este recorrido, se desarrollará una comprensión profunda de los aspectos que necesitamos mejorar para alcanzar con éxito las metas establecidas. Este análisis exhaustivo no solo desvelará los desafíos presentes en el camino hacia la culminación de

---

<sup>12</sup> UX: Se refiere a la manera en la que un usuario interactúa con un producto o servicio digital

<sup>13</sup> Eficiencia: Capacidad de lograr los resultados deseados con el mínimo posible de recursos.

<sup>14</sup> Eficacia: Capacidad de lograr el efecto que se desea o se espera.

<sup>15</sup> Empresa de Ajuste: Conformada por ajustadores los cuales llevan a cabo el proceso de investigación de un siniestro con la finalidad de llegar a una indemnización de acuerdo con condiciones de póliza

nuestros objetivos, sino que también proporcionará una base sólida para identificar y perfeccionar aquellos elementos que son cruciales para el éxito del proyecto.

La creación del diseño y de la propia aplicación web buscará ayude significativamente al crecimiento de dicha empresa.

## **II. Planteamiento del Problema.**

Crear aplicaciones webs que cumplan con el propósito con el que fueron creadas y a su vez sea atractiva para los usuarios, implica enfrentar algunos desafíos significativos, a su vez, tener como referencia el diseño característico de una empresa supone contemplar más aspectos para el resultado final.

Esta memoria de investigación busca identificar y abordar dichos desafíos durante el proceso de diseño y construcción de la aplicación web, con el propósito de mejorar la eficiencia y la estética del producto final. Algunas de las interrogantes que se tienen incluyen las siguientes:

1. ¿Cómo influye el diseño de la interfaz en los primeros segundos de uso por parte del usuario?
2. ¿Existe un diseño o reglas específicas a las que debemos apegarnos de acuerdo con la empresa?
3. ¿Qué tiempo de uso debe de tener en promedio el usuario dentro del sistema?
4. ¿Cuál es el promedio de edad del usuario que va a utilizar esta aplicación web?
5. ¿Cuál es el proceso y/o logística que la empresa desea mejorar con la aplicación web?

Estas interrogantes ayudarán durante el proceso de investigación y desarrollo.

### III. Objetivo.

El objetivo principal es la creación de una aplicación web que minimice el tiempo y los costos del proceso de reclamación<sup>16</sup> de una empresa de ajuste, implementando una experiencia de usuario fluida y con los menores pasos para lograr la captura de evidencias<sup>17</sup> por parte del usuario, este a su vez debe encontrarse con una interfaz agradable y que le dé la confianza de utilizarlo en múltiples ocasiones.

Como objetivo secundario será la disminución del coste que genera este dicho proceso de reclamación, ya que, no es necesario que los ajustadores<sup>18</sup> se trasladen al punto de inspección<sup>19</sup>, sino que, el proceso que ellos realizaban recaiga en los mismos asegurados (clientes), es decir, que ellos genere sus propias evidencias.

### IV. Marco Teórico.

El diseño y desarrollo web son disciplinas esenciales en la era digital, enfocadas en la creación de interfaces interactivas y experiencias únicas para los usuarios, el diseño y desarrollo web han experimentado una transformación significativa, impulsada por avances tecnológicos y cambios en las expectativas de los usuarios.

La UI<sup>20</sup> se refiere a la selección de colores, tipografía y elementos visuales que refuercen la identidad de la página web y a su vez de la marca de la empresa a la que pertenece. Por su lado la UX se centra en la facilidad de navegación, usabilidad y accesibilidad. En conjunto buscan atacar las

---

<sup>16</sup> Reclamación: Se consideran reclamaciones las presentadas por usuarios de servicios financieros con la pretensión de obtener la restitución de su interés o derecho

<sup>17</sup> Evidencias: Certeza clara y manifiesta de la que no se puede dudar.

<sup>18</sup> Ajustadores: Un ajustador es un profesional independiente a la institución aseguradora, que actúa como experto imparcial al momento de ajustar un siniestro, investiga los hechos, comprueba y da seguimiento a la reclamación y valora las pérdidas o daños.

<sup>19</sup> Inspección: Examinar, reconocer atentamente.

<sup>20</sup> UI: User Interface es la capa visual y tangible a través de la cual los usuarios interactúan con un producto digital

necesidades del usuario y proporcionar soluciones intuitivas que mejoren la experiencia global de la página web.

El diseño responsivo garantiza que la página web sea accesible y funcional en dispositivos de diferentes tamaños, esto involucra dispositivos móviles como lo son los celulares o tableta y para dispositivos grandes, como lo son laptops y equipos de cómputo de escritorio. El uso de CSS empleando medidas de dispositivos y media queries<sup>21</sup>, optimiza la visualización en diversas plataformas.

En cuanto a las tecnologías de desarrollo que se utilizan para la creación de este proyecto se encuentra HTML5 CSS3 y Javascript, se utilizan a su vez el framework React que ofrece entre muchas cosas la reutilización de elementos y simplificación de código, implementación de bases de datos no relaciones.

Las herramientas que se emplean son el uso de Visual Studio Code como IDE de desarrollo, versionado en GIT.

---

<sup>21</sup> Media Queries: Las consultas de medios le permiten adaptar su sitio o aplicación dependiendo de la presencia o el valor de varias características y parámetros del dispositivo.



## V. Desarrollo de Proyecto. Diseño Web.

### A. Objetivo Smart.

En los próximos 4 meses, se creará un prototipo de aplicación web para un proceso administrativo de una empresa de ajuste, en la cual, se capture fotografías y videos, tanto las fotos como los videos deberán estar agrupados por un listado de ubicaciones (GPS) proporcionados por la empresa en cuestión. Deberá ser intuitivo para los usuarios y con el mínimo de interacciones (clics/taps), así mismo el tratado de la información será manejado con los estándares de confidencialidad más altos.

### B. Moodboard.

A continuación, se presenta el moodboard<sup>22</sup> construido a partir de los colores y el diseño de la empresa a la que será construida la aplicación web. Se eligieron así mismo las fuentes utilizadas cuentan con los estándares utilizados, las imágenes son elegidas para simplificar las actividades y usuarios a los que va dirigida la aplicación web.

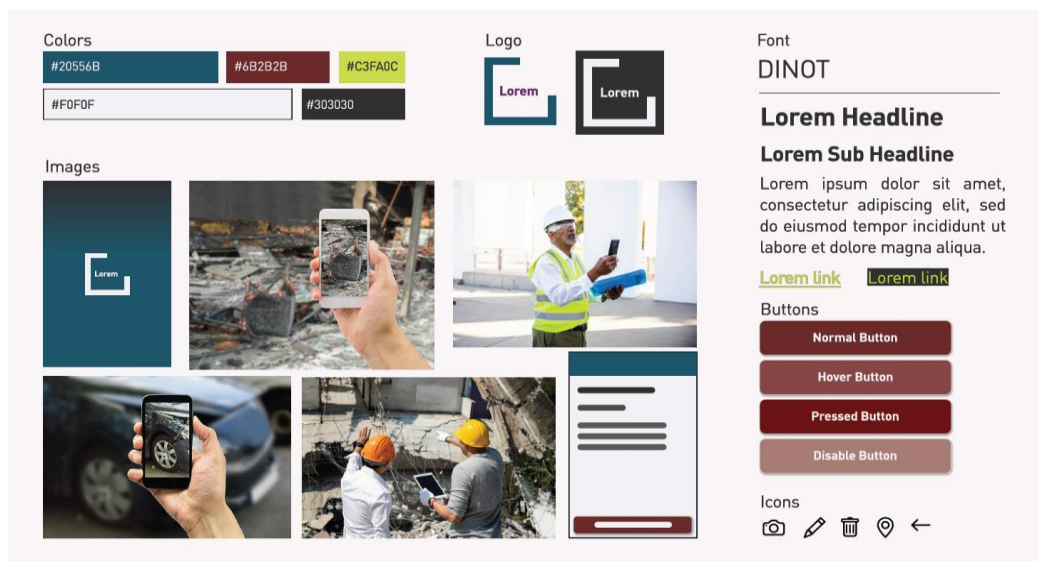


Imagen 1. Moodboard.

<sup>22</sup> Moodboard: Es una colección de imágenes, colores, tipografías y elementos visuales que capturan la esencia y la dirección estética de un proyecto

## C. Estructura del sitio.

En esta sección se define la estructura del sitio web, como es que el usuario va a interactuar en ella, se simplifica la estructura de la siguiente forma.

App y sitio web dirigido a una empresa privada encargada de ajustes en daños diversos, el objetivo es la toma de evidencia fotográfica de los mismos. La seguridad es esencial por esto la primera pantalla es un inicio de sesión, la segunda pantalla es un tipo splash que de una pequeña introducción a la app/web, a su vez servirá de loading en lo que se recibe la información desde el servidor, la siguiente pantalla muestra un listado de las tareas (ubicaciones geográficas) donde el usuario debe tomar evidencias, la cuarta pantalla es una galería de las fotos tomadas con opción de seguir agregando más o regresar a la pantalla de lista de tareas, la última pantalla la toma de evidencia y la corroboración de la misma, una vez tomada se regresará a la galería para observarla.

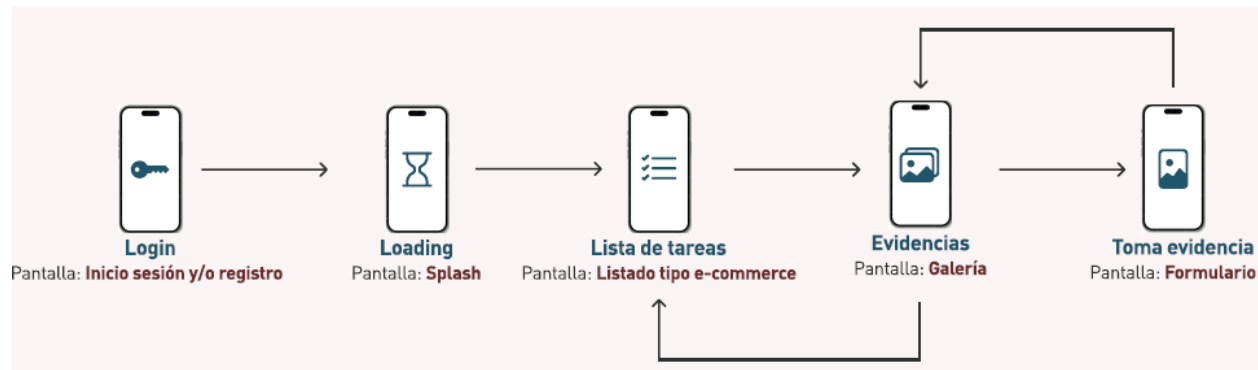


Imagen 2. Estructura del sitio.

## D. Wireframe.

Para representar como es que se verán las diferentes pantallas que se van a desarrollar para la página web tanto para vistas en celulares tablets y vistas de escritorio, se desarrollaron los siguientes wireframes<sup>23</sup>.

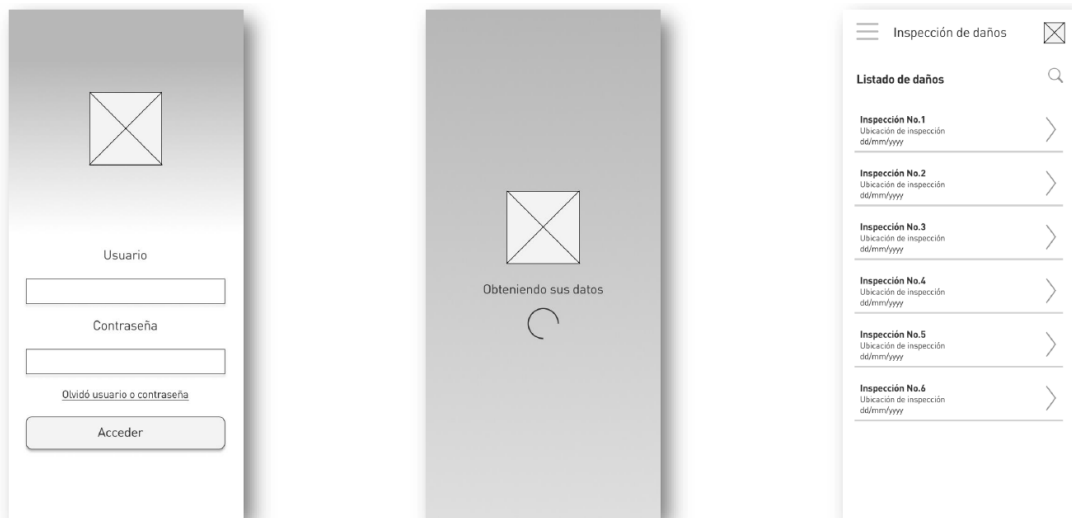


Imagen 3. Wireframe móvil.



Imagen 4. Wireframe móvil.

<sup>23</sup> Wireframes: son una representación gráfica de la disposición de los elementos en el sistema.

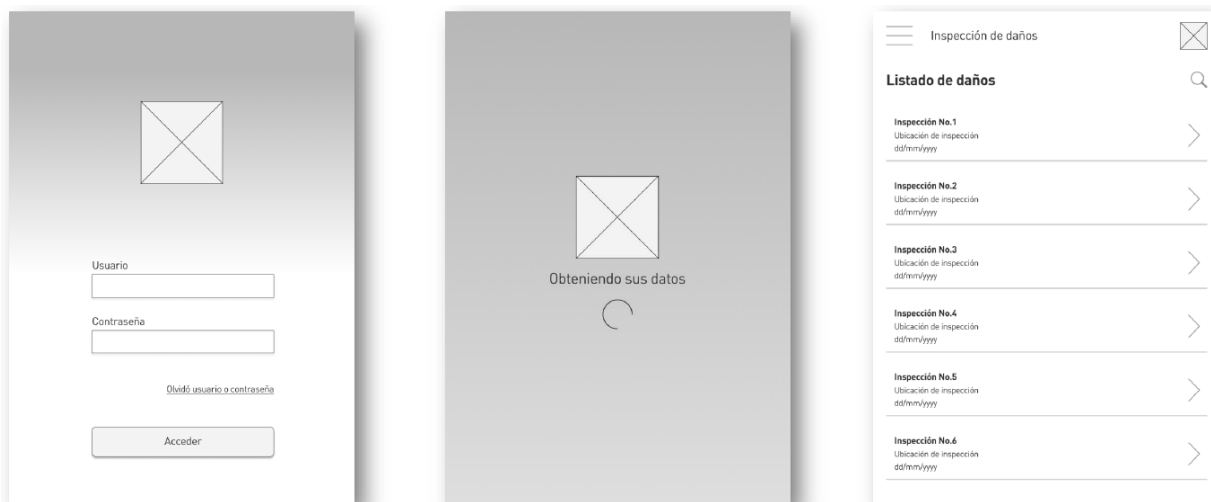


Imagen 5. Wireframe Tablet.

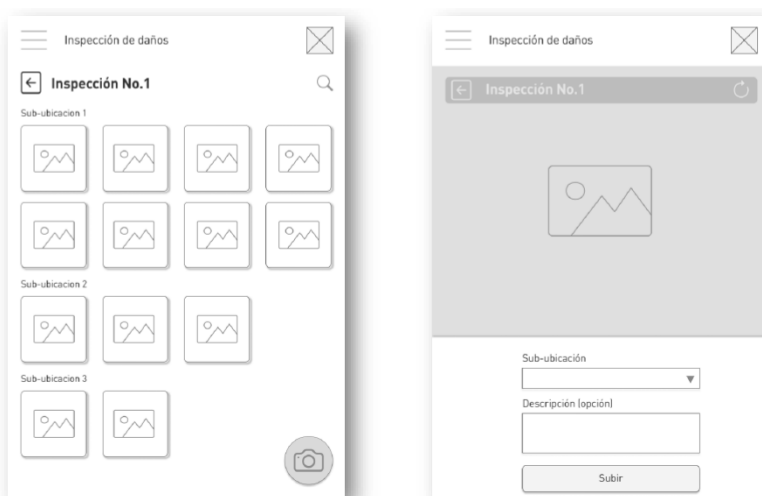


Imagen 6. Wireframe Tablet.

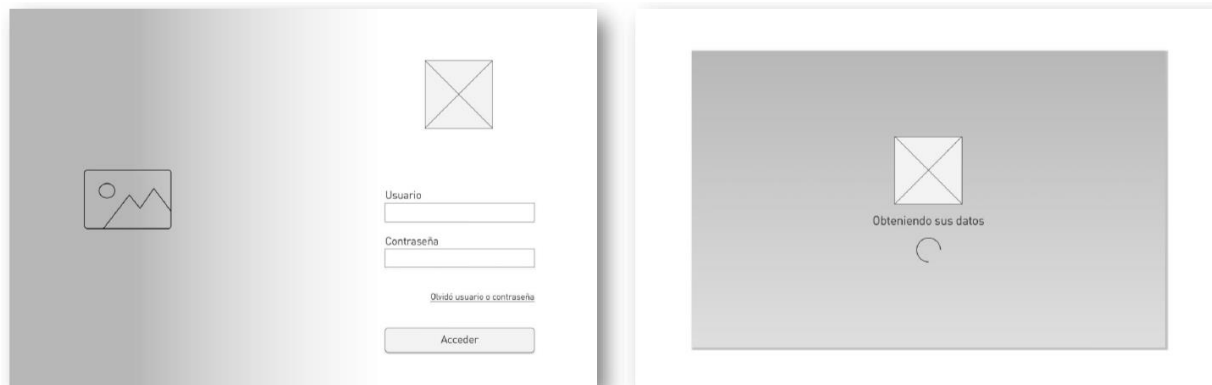


Imagen 7. Wireframe escritorio.

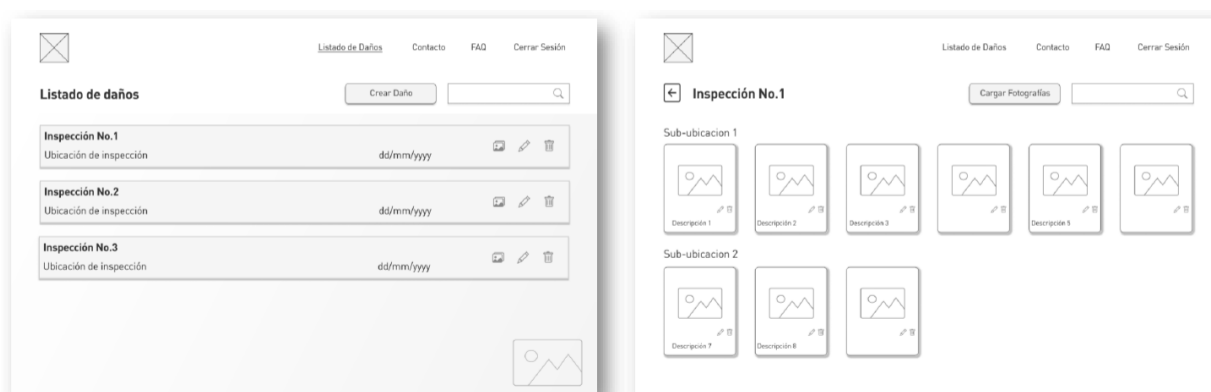


Imagen 8. Wireframe escritorio.



Imagen 9. Wireframe escritorio.

## E. Wireflow.

El proceso de interacción de los componentes de cada página se encuentra ilustrados en sus correspondientes wireflows<sup>24</sup>.

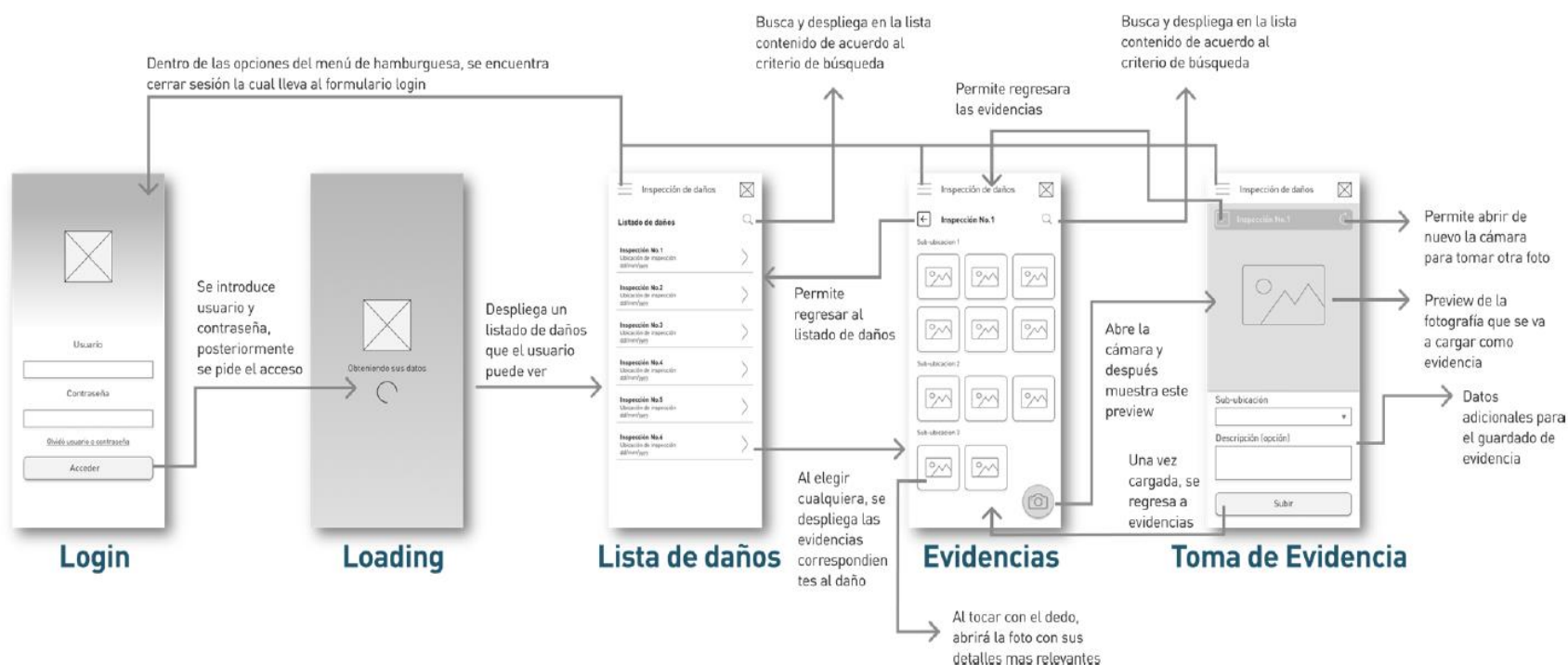


Imagen 10. Wireflow móvil.

<sup>24</sup> Wireflow: sirven para documentar el proceso que realizará el usuario a través de la página web,

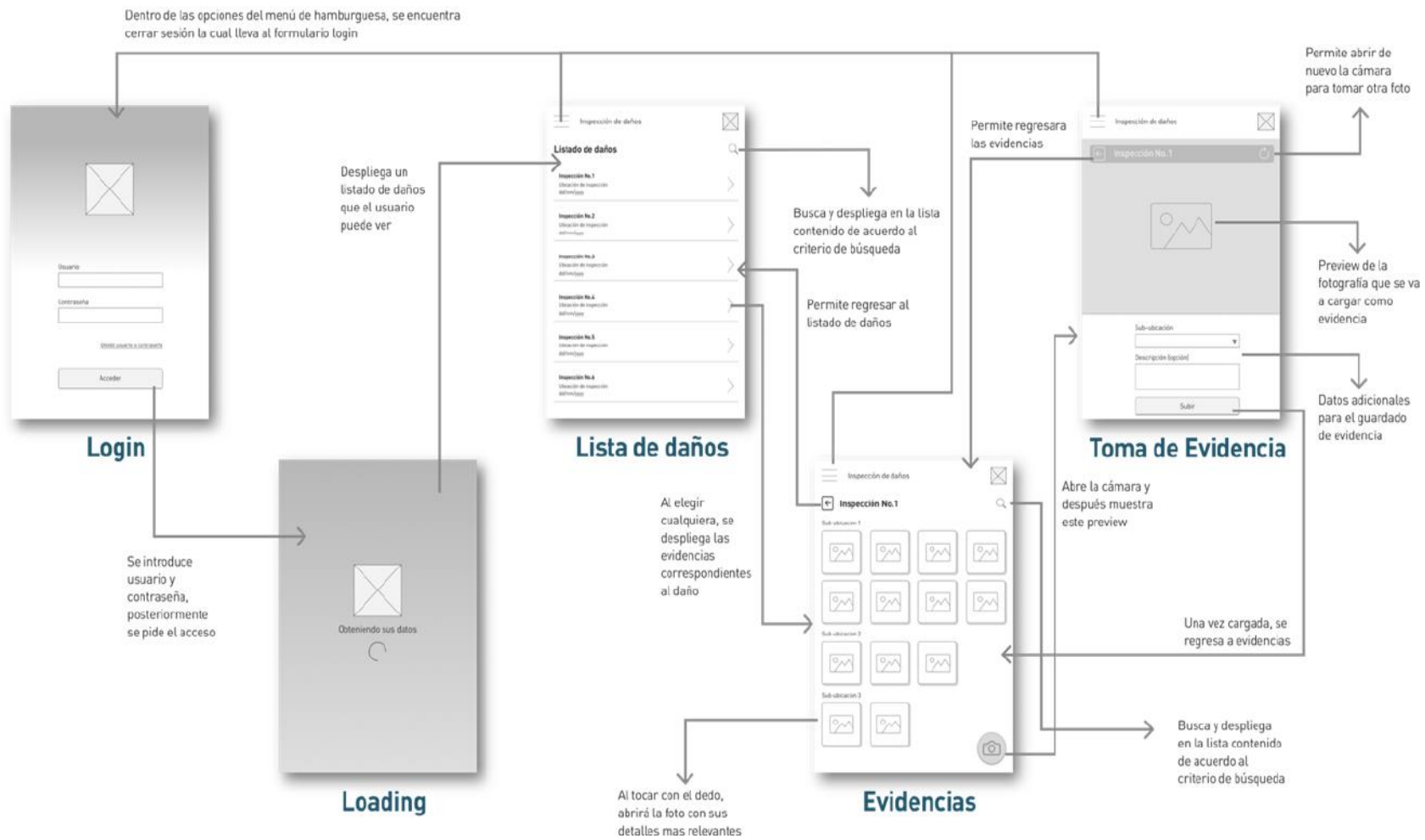


Imagen 11. Wireflow Tablet.

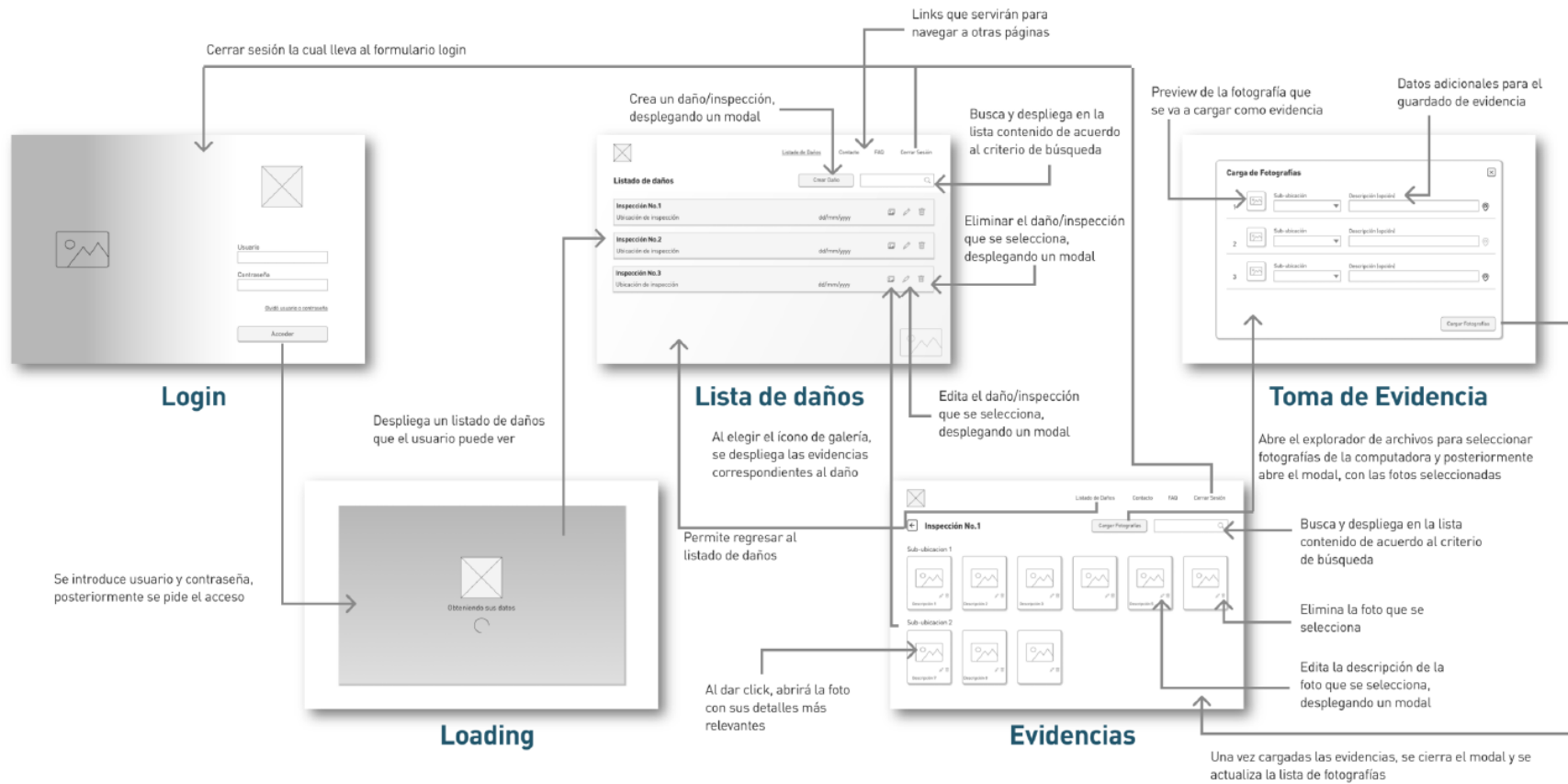


Imagen 12. Wireflow escritorio.



## VI. Desarrollo del Proyecto. Desarrollo Web

### A. Herramientas por utilizar durante el desarrollo.

Para el desarrollo de esta aplicación web, se necesitan las siguientes herramientas e instalaciones.

1. Node js versión 16+
2. Git
3. Tener un repositorio en github
4. Un navegador instalado, preferencia Chrome
5. Visual Studio Code
6. Descargar repositorios vite (en la sección B se aborda más a detalle)
7. Instalación de Tailwind

### B. Instalaciones Necesarias.

A continuación, se muestra el proceso de las instalaciones necesarias.

1. Instalación de Node js.
  - a. Ir a la página oficial de node y descargar el archivo de instalación <https://nodejs.org/en/download>

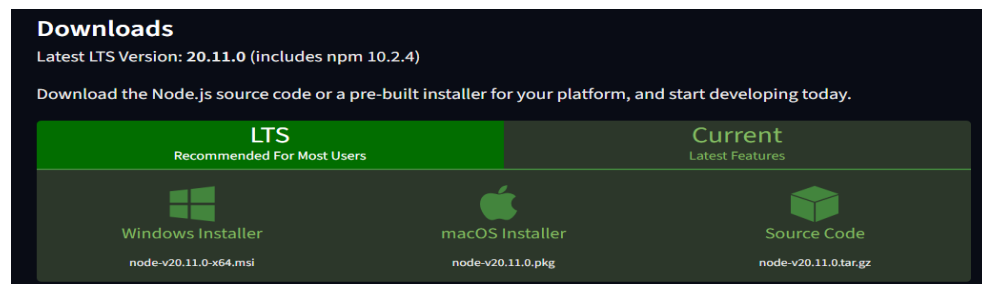


Imagen 13. Descarga de nodejs.

- b. Seguir el proceso y para corroborar que esté instalado abriendo una terminal y poniendo node -v y npm -v.

```
PS C:\Users\  
v16.18.0  
PS C:\Users\  
8.19.2  
PS C:\Users\  
> node -v  
> npm -v  
>
```

Imagen 14. Corroboración de instalación.

- c. En caso de no servir revisar que la ruta al instalador este dentro de las variables globales de entorno.

## 2. Instalar Visual Studio Code.

- a. Entrar a la página oficial de visual Studio Code y descargar el archivo ejecutable.

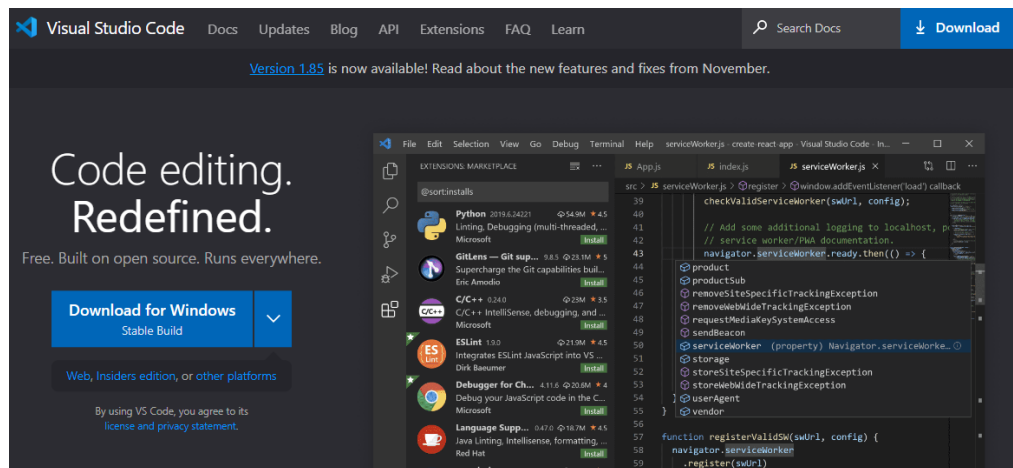


Imagen 15. Descarga de ejecutable VSC.

- b. Seguir las instrucciones.

### 3. Instalar Git

- a. Descargar el instalador desde la página oficial

<https://gitforwindows.org/>

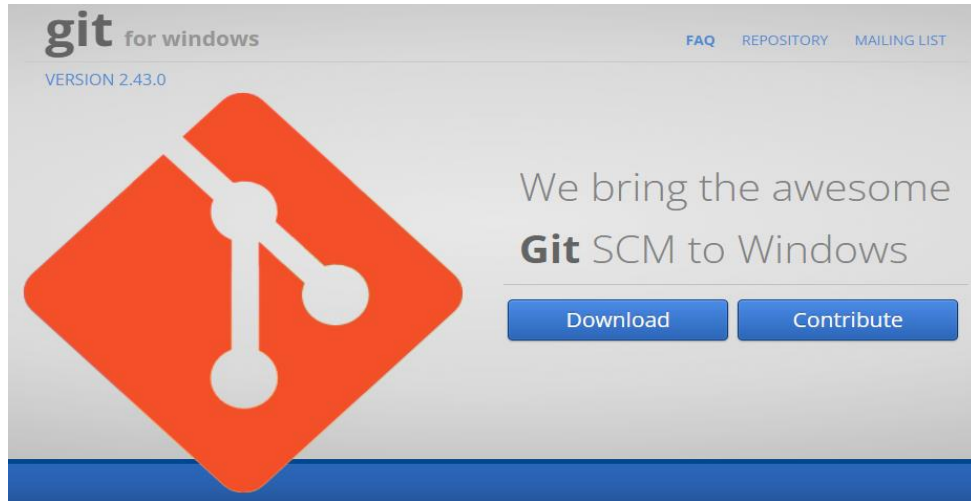


Imagen 16. Descarga de ejecutable Git.

- b. Seguir instrucciones y abrir git bash, esta será la principal consola que se manejará.

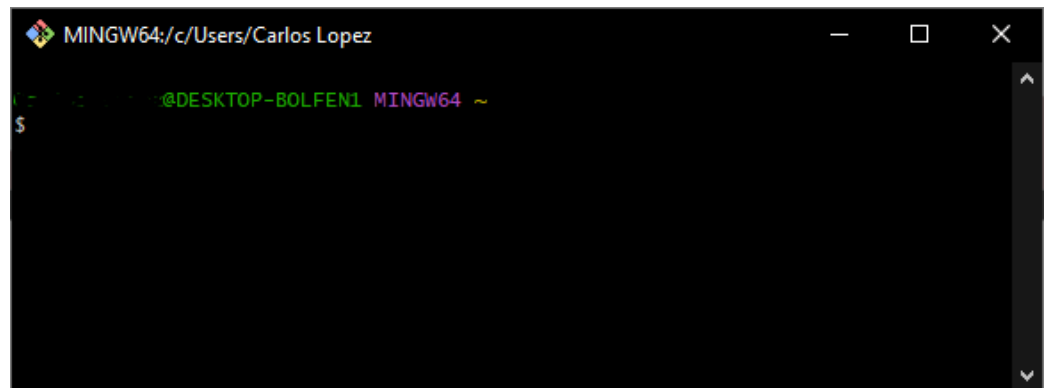


Imagen 17. Consola git bash.

#### 4. Crear una cuenta en github

##### a. Entrar a la página de registro de github

<https://github.com/signup>

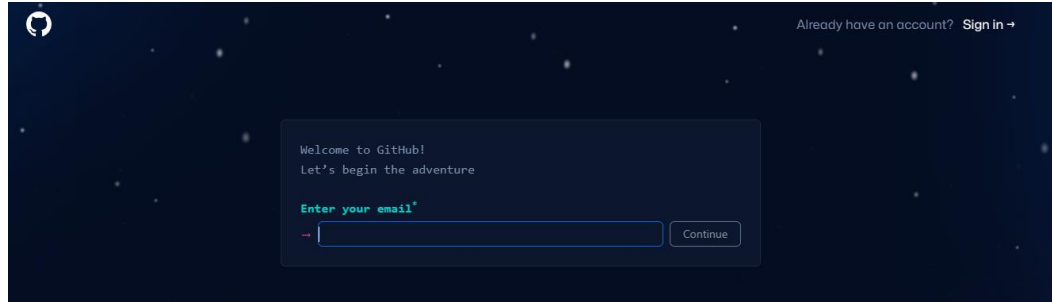


Imagen 18. Página de registro de github.

##### b. Seguir los pasos para crear la cuenta.

##### c. Crear un repositorio, para esto ir a la página

<https://github.com/new>

The image shows the "Create a new repository" form on GitHub. The title is "Create a new repository". Below it, a subtitle says "A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)". A note says "Required fields are marked with an asterisk (\*)". The form has two main sections: "Owner" and "Repository name", both with dropdown menus. Below these is a note: "Great repository names are short and memorable. Need inspiration? How about [redesigned-guide](#) ?". There is a "Description (optional)" text input field. Below that are two radio buttons: "Public" (selected) and "Private". The "Public" option has a subtext: "Anyone on the internet can see this repository. You choose who can commit." The "Private" option has a subtext: "You choose who can see and commit to this repository." Below the radio buttons is a section "Initialize this repository with:" with a checkbox "Add a README file". Below this checkbox is a subtext: "This is where you can write a long description for your project. [Learn more about READMEs.](#)". Below the "Add a README file" checkbox is a section "Add .gitignore" with a dropdown menu "gitignore template: None". Below this is a subtext: "Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)". Below the "Add .gitignore" section is a section "Choose a license" with a dropdown menu "License: None". Below this is a subtext: "A license tells others what they can and can't do with your code. [Learn more about licenses.](#)". At the bottom of the form, there is a note: "① You are creating a public repository in your personal account." and a green "Create repository" button.

Imagen 19. Formulario de alta de repositorio.

- d. Rellenar los campos con la información del proyecto a crear.
5. Clonar el proyecto a la computadora personal
- a. Una vez creado el repositorio, dirigirse a la parte superior derecha y desplegar la opción de “Code”, dentro de ella se puede copiar la dirección url del proyecto.

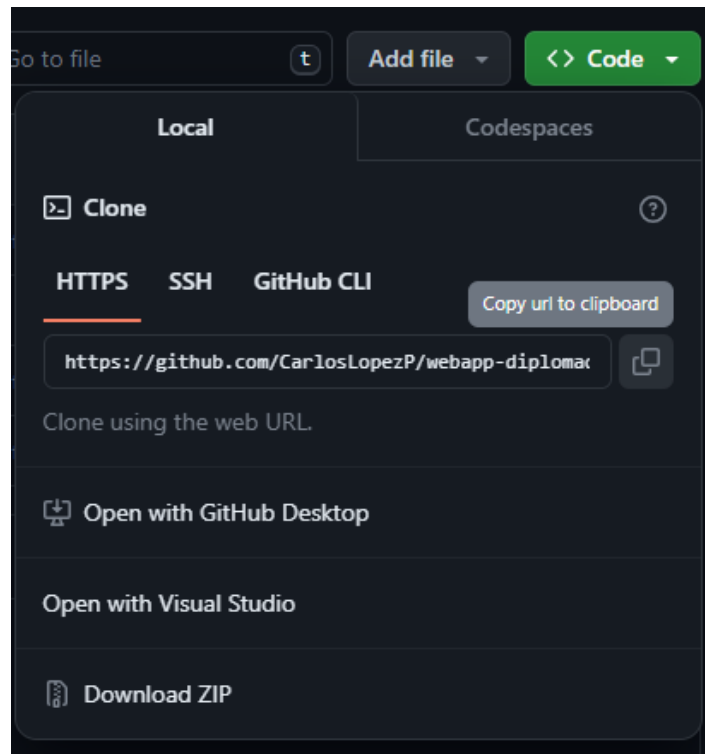
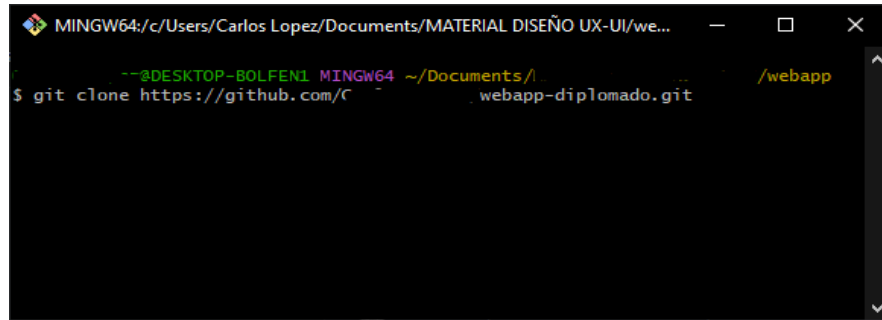


Imagen 20. Opción para copiar url del repositorio.

- b. Abrir dentro de la computadora personal una carpeta para alojar el código, en la misma carpeta abrir una terminal de git, colocar el código siguiente para clonar.

A terminal window with a black background and white text. The title bar shows 'MINGW64: c:/Users/Carlos Lopez/Documents/MATERIAL DISEÑO UX/UI/we...'. The prompt is '--@DESKTOP-BOLFEN1 MINGW64 ~/Documents/'. The command entered is '\$ git clone https://github.com/C... webapp-diplomado.git'.

```
MINGW64: c:/Users/Carlos Lopez/Documents/MATERIAL DISEÑO UX/UI/we...
--@DESKTOP-BOLFEN1 MINGW64 ~/Documents/
$ git clone https://github.com/C... webapp-diplomado.git
```

Imagen 21. Código en terminal para clonar un repositorio.

- c. Una vez descargado se puede acceder al repositorio.
6. Crear un proyecto en Vite
- a. Abrir el proyecto que se acaba de descargar en una terminal de git.
  - b. Colocar el comando para crear un proyecto de Vite, se describe a continuación.

A terminal window with a black background and white text. The title bar shows 'MINGW64: c:/Users/Carlos Lopez/Documents/MATERIAL DISEÑO UX/UI/we...'. The prompt is '--@DESKTOP-BOLFEN1 MINGW64 ~/Documents/'. The command entered is '\$ npm init vite@latest'.

```
MINGW64: c:/Users/Carlos Lopez/Documents/MATERIAL DISEÑO UX/UI/we...
--@DESKTOP-BOLFEN1 MINGW64 ~/Documents/
$ npm init vite@latest
```

Imagen 22. Comando para crear proyecto en React.

- c. Seguir las instrucciones, seleccionando como framework React y como lenguaje Javascript.
- d. Entrar a la carpeta que se creó, con el comando "cd".

A terminal window with a black background and green text. The title bar shows 'MINGW64:/c/Users/C... z/Documents/MATERIAL DISEÑO UX-UI'. The prompt is 'MINGW64 ~/Documents/MATERIAL DISEÑO UX-UI' and the command entered is '\$ cd webapp-diplomado/'.

```
MINGW64:/c/Users/C... z/Documents/MATERIAL DISEÑO UX-UI
MINGW64 ~/Documents/MATERIAL DISEÑO UX-UI
$ cd webapp-diplomado/
```

Imagen 23. Moverse a la carpeta del proyecto.

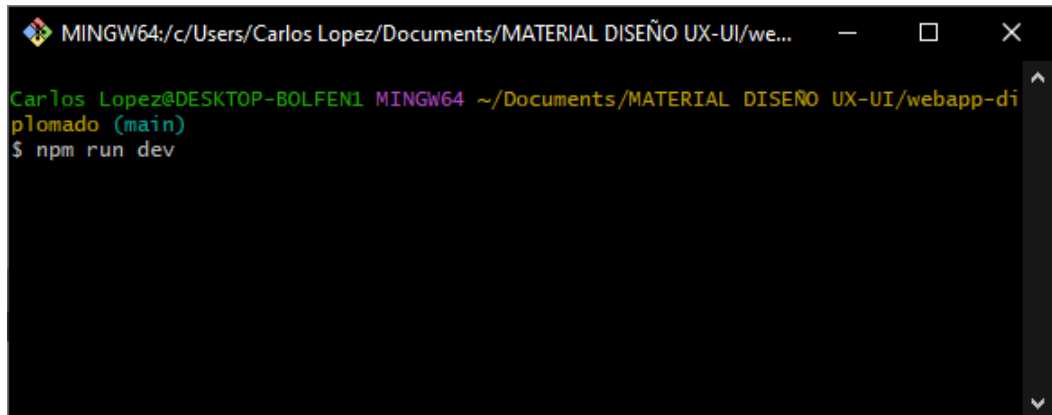
- e. Ejecutar el comando para instalar componentes y librerías necesarias para el funcionamiento correcto.

A terminal window with a black background and green text. The title bar shows 'MINGW64:/c/Users/C... Documents/MATERIAL DISEÑO UX-UI/we...'. The prompt is 'MINGW64 ~/Documents/MATERIAL DISEÑO UX-UI/webapp-diplomado (main)' and the command entered is '\$ npm install'.

```
MINGW64:/c/Users/C... Documents/MATERIAL DISEÑO UX-UI/we...
MINGW64 ~/Documents/MATERIAL DISEÑO UX-UI/webapp-diplomado (main)
$ npm install
```

Imagen 24. Comando para instalar librerías y/o componentes.

- f. Una vez concluida la instalación, levantar un pequeño servidor para poder visualizar el proyecto. Se realiza con el siguiente comando.

A terminal window with a black background and green text. The title bar shows 'MINGW64:/c/Users/Carlos Lopez/Documents/MATERIAL DISEÑO UX-UI/we...'. The prompt is 'Carlos Lopez@DESKTOP-BOLFEN1 MINGW64 ~/Documents/MATERIAL DISEÑO UX-UI/webapp-di'. The user has entered 'plomado (main)' and '\$ npm run dev'.

```
MINGW64:/c/Users/Carlos Lopez/Documents/MATERIAL DISEÑO UX-UI/we...
Carlos Lopez@DESKTOP-BOLFEN1 MINGW64 ~/Documents/MATERIAL DISEÑO UX-UI/webapp-di
plomado (main)
$ npm run dev
```

Imagen 25. Comando para correr el proyecto internamente.

- g. Ingresar la url que muestra la terminal, abrirlo en un navegador web.

A terminal window with a black background and green text. The title bar shows 'MINGW64:/c/Users/C... Documents/MATERIAL DISEÑO UX-UI/we...'. The prompt is 'Carlos Lopez@DESKTOP-BOLFEN1 MINGW64 ~/Documents/MATERIAL DISEÑO UX-UI/webapp-di'. The user has entered 'plomado (main)', '\$ npm run dev', and then '> registro-danhos-app@0.0.0 dev' and '> vite'. The output shows 'VITE v5.0.11 ready in 1334 ms' and the local URL 'http://127.0.0.1:5173/'.

```
MINGW64:/c/Users/C... Documents/MATERIAL DISEÑO UX-UI/we...
Carlos Lopez@DESKTOP-BOLFEN1 MINGW64 ~/Documents/MATERIAL DISEÑO UX-UI/webapp-di
plomado (main)
$ npm run dev

> registro-danhos-app@0.0.0 dev
> vite

VITE v5.0.11 ready in 1334 ms
➔ Local:   http://127.0.0.1:5173/
➔ Network: use --host to expose
```

Imagen 26. url generada.





Imagen 27. Página por defecto al correr un proyecto.

## 7. Instalar Tailwind en el proyecto.

- a. Abrir una línea de comandos en la raíz del proyecto y colocar el siguiente comando.

```
@DESKTOP-BOLFEN1 MINGW64 ~/Documents/MATERIAL DISEÑO UX/UI/webapp-diplomado (main)
$ npx tailwindcss init -p
```

Imagen 28. Comando para instalar Tailwind.

- b. Una vez concluida la instalación se observa que fueron agregados dos archivos.

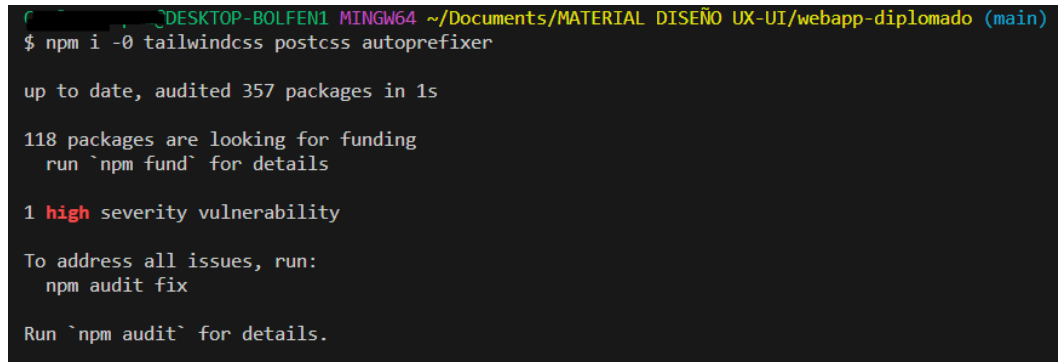
```
@DESKTOP-BOLFEN1 MINGW64 ~/Documents/MATERIAL DISEÑO UX/UI/webapp-diplomado (main)
$ npx tailwindcss init -p
Need to install the following packages:
  tailwindcss@3.4.1
Ok to proceed? (y) y

Created Tailwind CSS config file: tailwind.config.js
Created PostCSS config file: postcss.config.js

@DESKTOP-BOLFEN1 MINGW64 ~/Documents/MATERIAL DISEÑO UX/UI/webapp-diplomado (main)
$
```

Imagen 29. Resultados de la instalación de Tailwind.

- c. Configurar Tailwind para que funcione globalmente en el proyecto, en la misma ventana de consola ingresar el siguiente comando.



```
DESKTOP-BOLFEN1 MINGW64 ~/Documents/MATERIAL DISEÑO UX/UI/webapp-diplomado (main)
$ npm i -D tailwindcss postcss autoprefixer

up to date, audited 357 packages in 1s

118 packages are looking for funding
  run `npm fund` for details

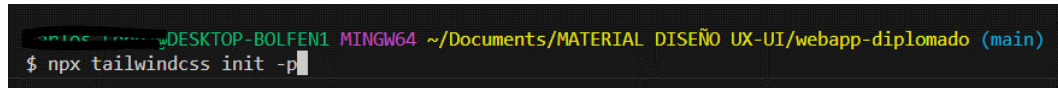
1 high severity vulnerability

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
```

Imagen 30. Comando para configurar el archivo Tailwind.

- d. Se ejecuta el siguiente comando una vez haya finalizado el paso anterior.



```
DESKTOP-BOLFEN1 MINGW64 ~/Documents/MATERIAL DISEÑO UX/UI/webapp-diplomado (main)
$ npx tailwindcss init -p
```

Imagen 31. Comando inicializar tailwind.

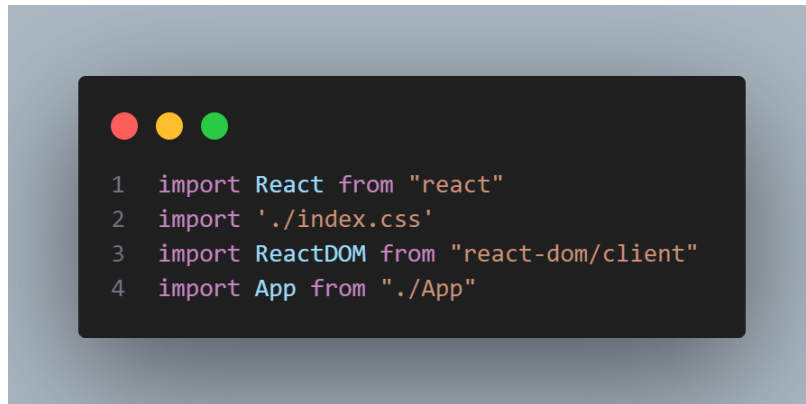
- e. Abrir el archivo index.css y modificar con lo siguiente,



```
1 @tailwind base;
2 @tailwind components;
3 @tailwind utilities;
```

Imagen 32. Modificación del archivo index.css.

- f. Modificar igualmente el main.jsx para que importe la plantilla de index.css.



```
1 import React from "react"
2 import './index.css'
3 import ReactDOM from "react-dom/client"
4 import App from "./App"
```

Imagen 33. Importar index.css.

- g. Abrir y modificar el archivo tailwind.config.js y agregar en “content” la regla para que todos los componentes utilicen tailwind.

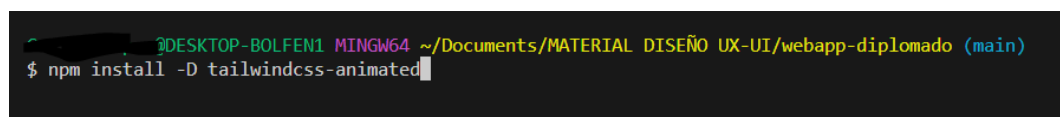


```
1 /** @type {import('tailwindcss').Config} */
2 export default {
3   content: ["/index.html", "/src/pages/**/*.jsx", "/src/*.jsx"],
4   theme: {
5     extend: {},
6   },
7   plugins: [
8
9   ],
10 }
11
12
```

Imagen 34. Modificación archivo tailwind.config.js.

## 8. Instalar tailwind-animated

- a. Insertar el comando para instalar la paquetería



```
$ npm install -D tailwindcss-animated
```

Imagen 35. Instalación de tailwindcss-animated.

- b. Abrir y modificar el archivo tailwind.config.js, en la sección de plugins.



Imagen 36. Modificación tailwind.config.js.

## C. Programación del Maquetado

Una vez instaladas las herramientas necesarias para el desarrollo se procede a empezar con el “maquetado” que es la codificación de elementos visuales que previamente se construyeron en los wireframes. Dar énfasis en que queden los elementos y colores tal cual se han propuesto con el moodboard es una tarea esencial.

Esta sección detallará los pasos clave involucrados. Asegurando la coherencia, la accesibilidad y la usabilidad en todo momento. Además, se examinarán las herramientas y técnicas utilizadas para optimizar el rendimiento y la compatibilidad multiplataforma de la aplicación web.

### 1. Planificación:

Analizando las diferentes pantallas podemos observar que elementos son repetitivos, estos van a ser codificados en archivos separados para que se pueda en distintas pantallas reutilizarlos, como elemento repetitivo, un ejemplo que se puede observar en el proyecto actual es el menú superior.

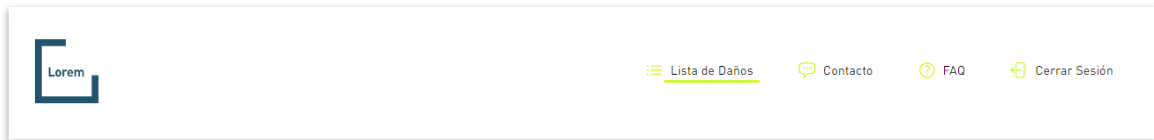


Imagen 37. Menú superior de la aplicación.

Otra parte de la planificación es la elección del tamaño de las fuentes, es clave identificar cual es el tamaño máximo que vamos a utilizar para los títulos más importantes y dependiendo eso, como es que las demás fuentes vas a verse modificadas en tamaño, dar atención a que se va a utilizar por lo menos 3 tipos de fuentes mayores una para vistas en desktop, la segunda para vistas de tablets y la tercera para vistas en celulares, a continuación se muestras los ejemplos de la aplicación da cada una de las fuentes utilizadas como títulos.

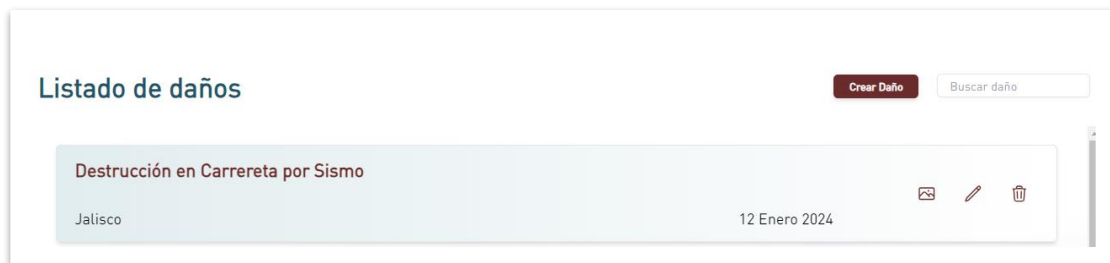


Imagen 38. Título “Listado de daños” para vista desktop.



Imagen 39. Título “Listado de daños” para vista Tablet.



Imagen 40. Título “Listado de daños” para vista móvil.

Para contar con un diseño responsivo, es decir que se pueda ver adecuadamente en múltiples dispositivos es crucial pensarlo en esta etapa de planeación, tanto que técnica va a ser utilizada, como que herramientas o frameworks son los más eficientes a la hora de realizar esta tarea que puede llegar a ser compleja.

Las técnicas que existen para lograr un diseño responsivo se encuentran las siguientes:

- Mobile First
- Desktop First

“Mobile First” trata de centrarse, como su nombre en inglés lo dice, primero desarrollar la versión móvil, mientras que la “Desktop First” busca por el contrario, enfocar los esfuerzos en la construcción de la versión para escritorio, es difícil decir cuál es la que trae más beneficios, ya que, por un lado realizar la versión móvil primero da un sentimiento de avance más rápido, porque los elementos que se utilizan en las versiones móviles buscan no tener demasiado contenido por el hecho de ser pantallas reducidas, de esta forma podemos tener un avance considerable en las primeras etapas de desarrollo aunque por el contrario, al dejar la vista de escritorio para el último puede ser cansado e incluso si no se tiene una buena planificación desde el wireframe, darse cuenta que elementos que

necesitamos para la lógica correcta de la página están en la versión de escritorio y se ausentan en la versión móvil.

Hablando del desarrollo “Desktop First” podemos encontrar como ventaja, que los elementos completos estarán ya reflejados en la maqueta desde etapas tempranas del desarrollo, esto también nos permite ver o analizar herramientas que quizá no se contemplaron en el diseño original, pero son cruciales para el correcto funcionamiento, este tipo de técnica no tiene sustanciales avances al principio y puede llegar a ser motivo de desesperación, pero por otro lado, deja las tareas más rápidas para el final del desarrollo.

Como se mencionó, ninguna es mejor que la otra y la elección de dicha técnica dependerá de la forma en que el desarrollador esté dispuesto a enfrentar la codificación.

Para este proyecto la técnica utilizada para el desarrollo será la de “Desktop First” ya que se espera abordar la programación más compleja desde un principio.

Por otro lado, el manejo responsivo puede hacerse entre diferentes formas que existen, de las siguientes 3 formas:

- “media queries”
- Vistas por pantalla
- Frameworks

Las “media queries” son reglas que se utilizan a nivel de css, estas marcan que tipo de diseños se van a utilizar dependiendo la medida de ancho de la pantalla que se está redimensionando en ese momento, esta técnica como ventaja se puede decir que en un solo archivo tenemos el control de cómo es que se verán las diversas vistas que se codifican, una de las desventajas es que el código puede llegar a no ser tan mantenible debido a que utilizar muchas reglas puede llegar a ser confuso para un desarrollador externo, sin embargo es una

de las mayormente aceptadas para el manejo responsivo, en la siguiente imagen se simplifica el proceso de esta técnica.



Imagen 41. Código de media.

Las vistas por pantalla son archivos separados que se codifican y por medio de lógica en el JavaScript se redirecciona a las ventanas correspondientes siguiendo el tamaño del dispositivo, como ventaja tiene que el código se vuelve más mantenible debido a que al ser elementos por separados, se puede acceder rápidamente para realizar cambios, la gran desventaja es que se duplica gran cantidad de código y el peso al momento de desplegar a producción es considerablemente más grande.

El uso de framework para ejecutar la tarea responsiva es más simple, ya que contiene reglas por defecto a las cuales se puede acceder mucho más rápido y con menos líneas de código, esa es su gran ventaja ya que en una misma línea de código se pueden delimitar los 3 tipos de vistas más populares, las cuales son vistas en escritorio, tablet y celulares. La desventaja de usar este tipo de herramientas es que se deben instalar de forma global, aunque no se usen todas las opciones, van a estar instalados en el proyecto, haciendo un poco más grande el proyecto para desplegar en producción.

Para el desarrollo de este proyecto se opta por usar la opción de framework ya que supone una gran ventaja por el tiempo de desarrollo y la posible escalabilidad que el proyecto requiere, como framework



se utilizará “Tailwind” como anteriormente se vio la instalación, nos ofrece gran variedad de uso para el maquetado y a su vez, para el diseño responsivo deseado, a continuación, se muestra en imagen un ejemplo de cómo se utiliza tailwind en el proyecto.

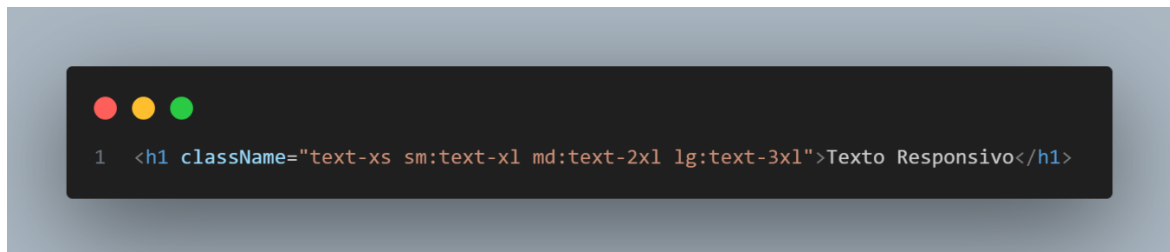


Imagen 42. Ejemplo del uso de Tailwind.

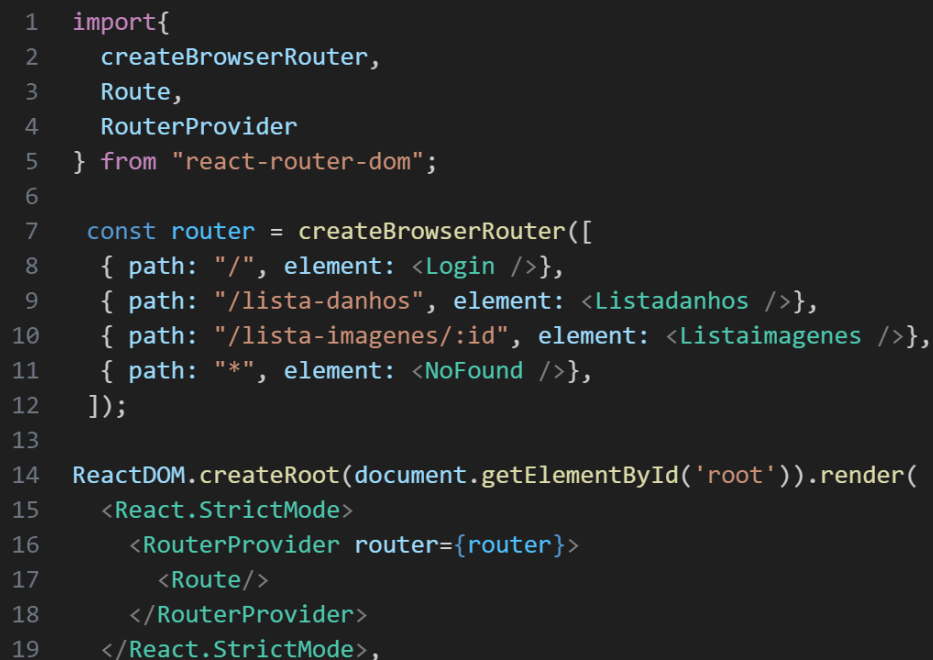
En el ejemplo anterior como defecto está el texto chico, la palabra reservada “sm” hace referencia a pantallas pequeñas y en el ejemplo anterior se le pone un valor de “text-xl” que es un texto un poco más grande, siguiendo el código las palabras reservadas “md” y “lg” hacen lo propio para dimensiones medianas y grandes respectivamente y se le asigna tamaños de textos cada vez más grandes.

## 2. Desarrollo:

Ya teniendo la planeación bien estructurada, se procede a la codificación, como primer paso se delimitan el número de páginas que se van a utilizar, incluyendo las que se van a reutilizar, esto ya tienes un proceso de análisis desde la planificación como ya se vio anteriormente, se procede a crear los archivos con el nombre declarativo de cada uno, esto es, palabras en concreto que a lo mucho en 2 expliquen de que se trata la vista, teniendo los diferentes archivos ya creados se plantea la posibilidad de tener una serie de rutas para tener todas las urls bien delimitadas, para esto se utiliza las propiedades de “Router” de React, a continuación, se explica cómo se declaran y como es que se utilizan, esto ayuda a tener todo controlado y saber a dónde es que se van a dirigir las rutas, también es

aconsejable que se tenga un archivo default el cual se despliegue cuando se quiera acceder a una url inexistente.

Para utilizar esta propiedad, debemos primero instalar la dependencia `react-router-dom` a través de `npm`, una vez teniéndolo se importa al archivo `main.jsx` como se muestra en la imagen de la línea 1 al 5, posteriormente se define una constante de tipo `createBrowserRouter` el cual contiene un arreglo de `path` y se definen los elementos (o vistas) a los que se refieren, dando énfasis en que se tiene una por defecto si es que se introduce una url equivocada, finalmente se utiliza `createRoot` para colocar todo el contenido en el `html`



```
1  import{
2    createBrowserRouter,
3    Route,
4    RouterProvider
5  } from "react-router-dom";
6
7  const router = createBrowserRouter([
8    { path: "/", element: <Login />},
9    { path: "/lista-danhos", element: <Listadanhos />},
10   { path: "/lista-imagenes/:id", element: <Listaimagenes />},
11   { path: "*", element: <NotFound />},
12  ]);
13
14  ReactDOM.createRoot(document.getElementById('root')).render(
15    <React.StrictMode>
16      <RouterProvider router={router}>
17        <Route/>
18      </RouterProvider>
19    </React.StrictMode>,

```

Imagen 43. Implementación de Rutas.

Al momento de codificar cada una de las vistas, se aconseje estructurar el código como lógica y debajo de ella el diseño en formato `jsx`, por otro lado, los recursos de `css` que sean globales irían en un

archivo a nivel del routing, donde se pueda utilizar en cualquier archivo, y si es necesario para código css especial, crear e importar uno independiente por vista, lo que hace que el código sea mantenible y claro.

En cuanto a las recomendaciones de formularios están tener claros los conceptos de usabilidad, lo cual significa, tener y estar consiente de todos los posibles escenarios que el usuario final pueda llegar a pasar, como por ejemplo insertar cosas vacías cuando no debería o no mostrar al usuario en caso de que esté en un , de igual manera el mostrar mensajes cuando se realicen tareas correctas y en todo momento darles las herramientas de poder cancelar alguna acción o regresar a un paso anterior si es que el usuario lo desea.

## **D. Resultados finales.**

A continuación, se muestran los resultados en vista del proyecto, tanto para dispositivos escritorio, tablet y móvil.

La lógica o procedimiento que debe realizar el usuario es el siguiente:

- Iniciar sesión con su cuenta.
- Los llevará a un listado de daños.
- Si se desea crear uno nuevo se pulsará el botón crear daño o en versión móvil, el botón circular flotante del lado inferior derecho.
- Para visualizar la galería de casa daño dar click en el icono de imagen.
- Dentro de la página de imágenes se pueden agregar más a cada sección con el botón cargar foto o en la vista de móvil en la parte inferior derecha un botón flotante con el ícono de una cámara fotográfica.
- Abrirá un explorador para seleccionar fotos, una vez seleccionadas aparecerá una ventana flotante que servirá para ver en que subsección se quiere agregar.

## 1. Móvil.



Imagen 44. Login.



Imagen 45. Lista de daños.



Imagen 46. Crear Inspección.

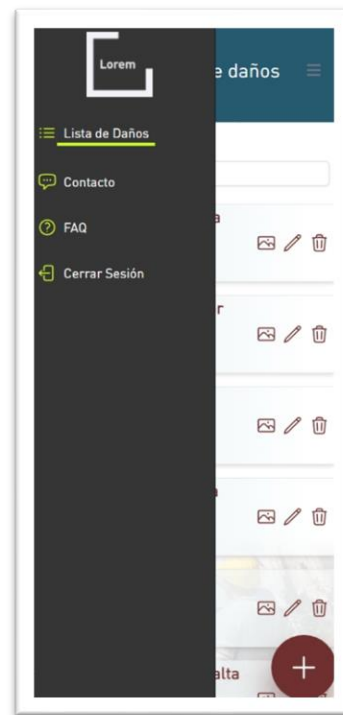


Imagen 47. Menú Hamburguesa.



Imagen 48. Galería imágenes.

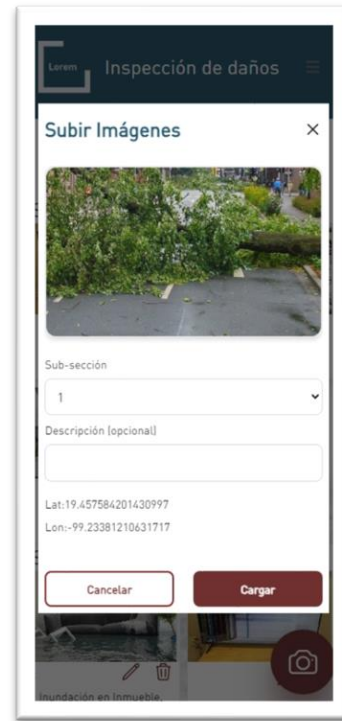


Imagen 49. Carga de imágenes.

## 2. Tablet.

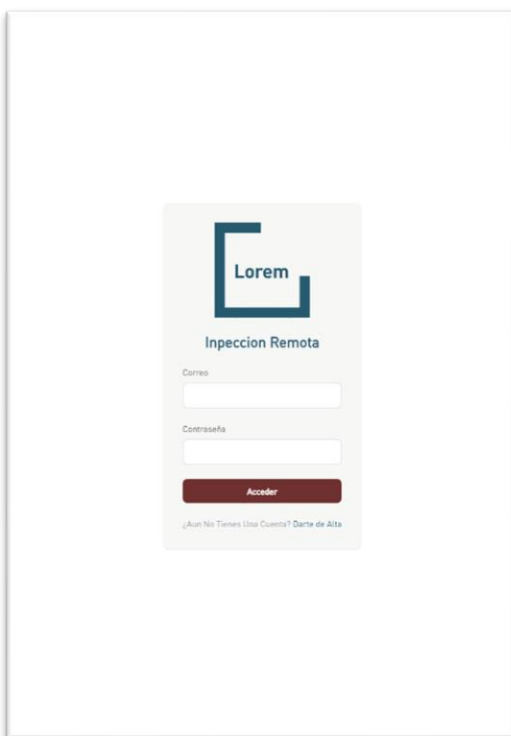


Imagen 50. Login (Tablet).



Imagen 51. Lista de daños (Tablet).

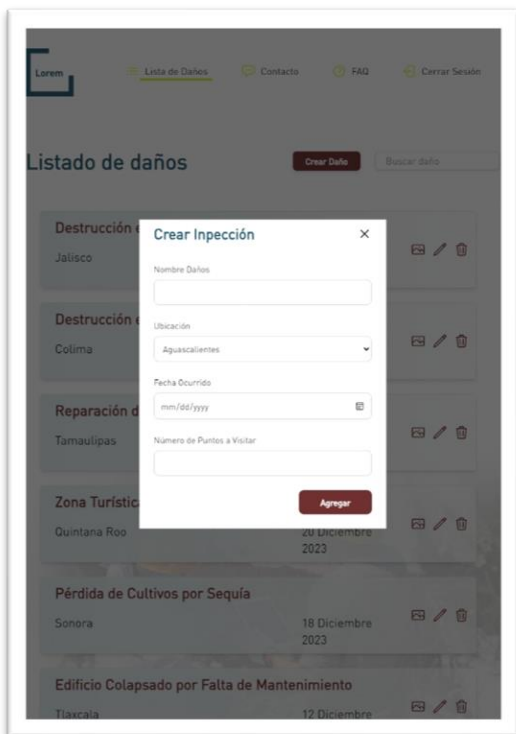


Imagen 52. Crear Inspección.

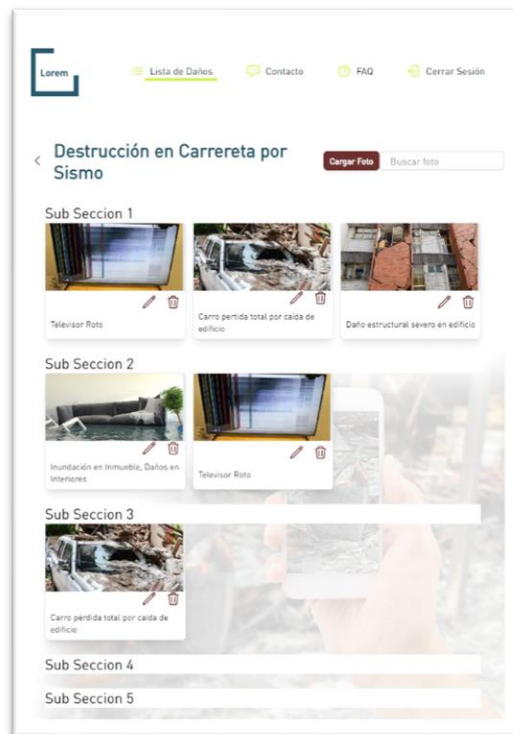


Imagen 53. Galería imágenes.

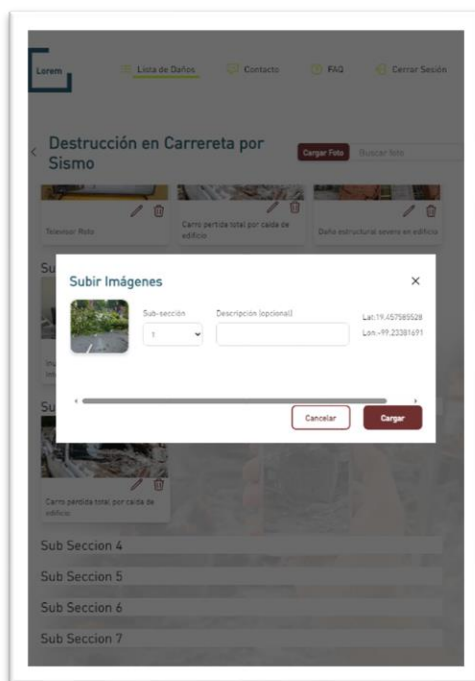


Imagen 54. Carga de imágenes (Tablet).

### 3. Escritorio.

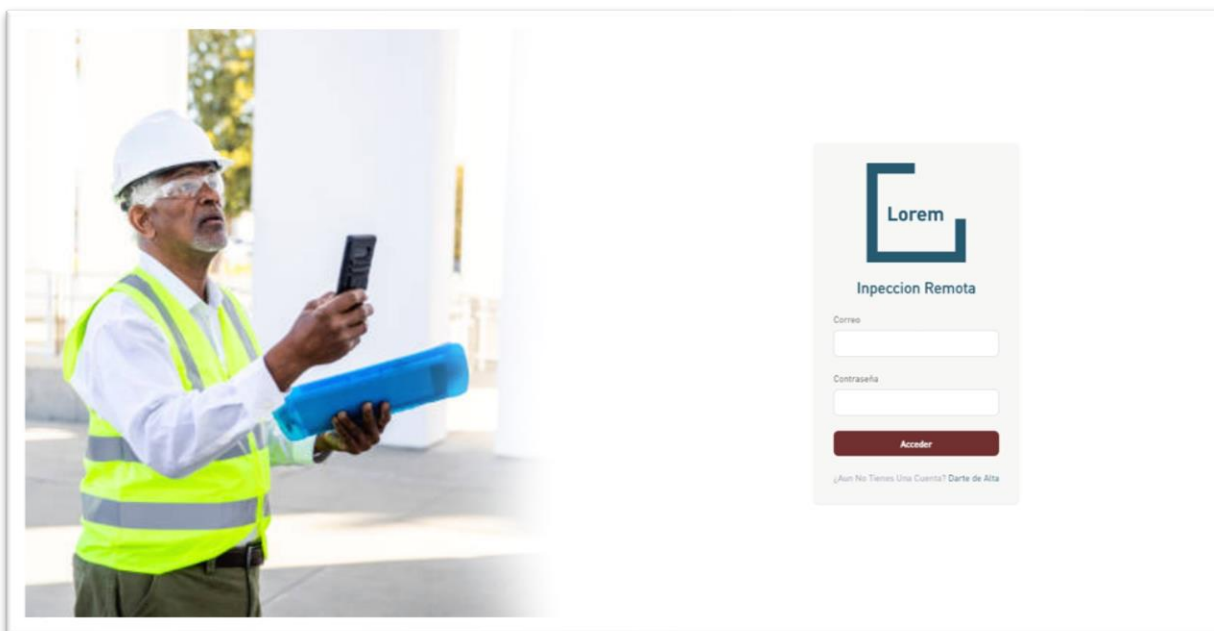


Imagen 55. Login (Desktop).

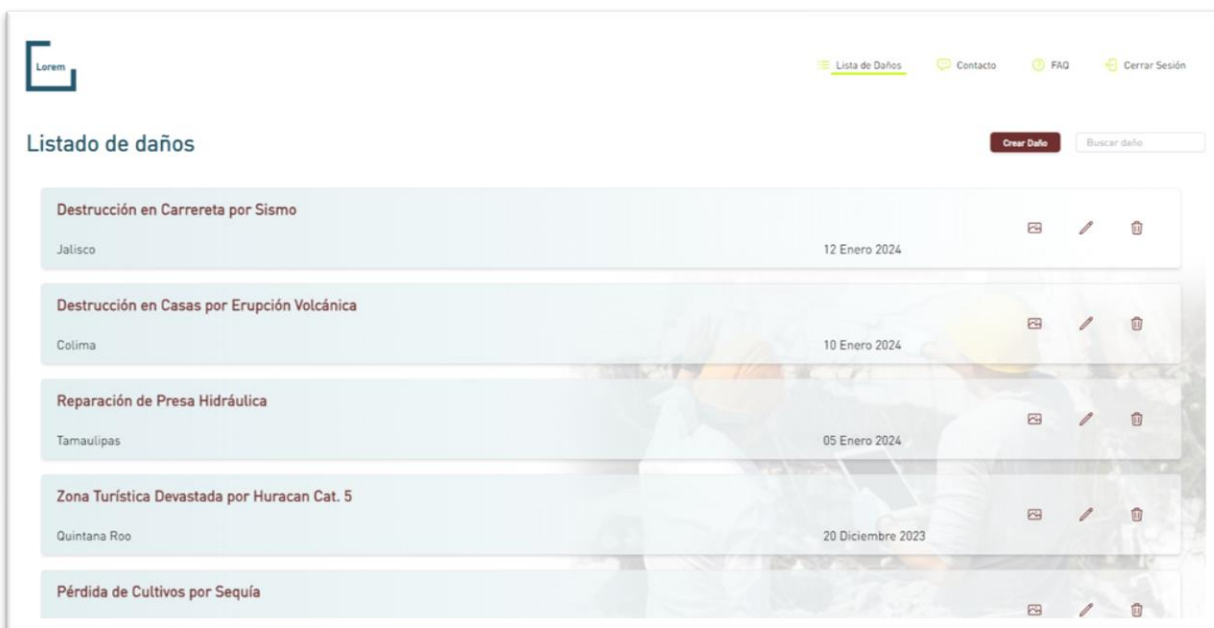


Imagen 56. Lista de daños (Desktop)

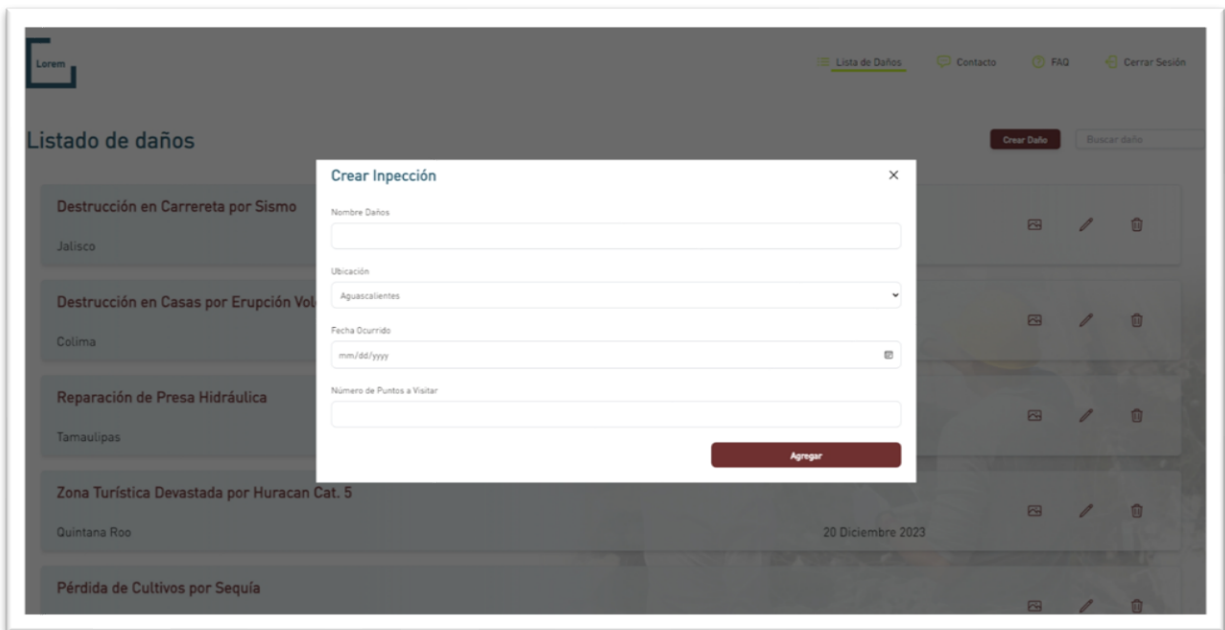


Imagen 57. Crear Inspección (Desktop).

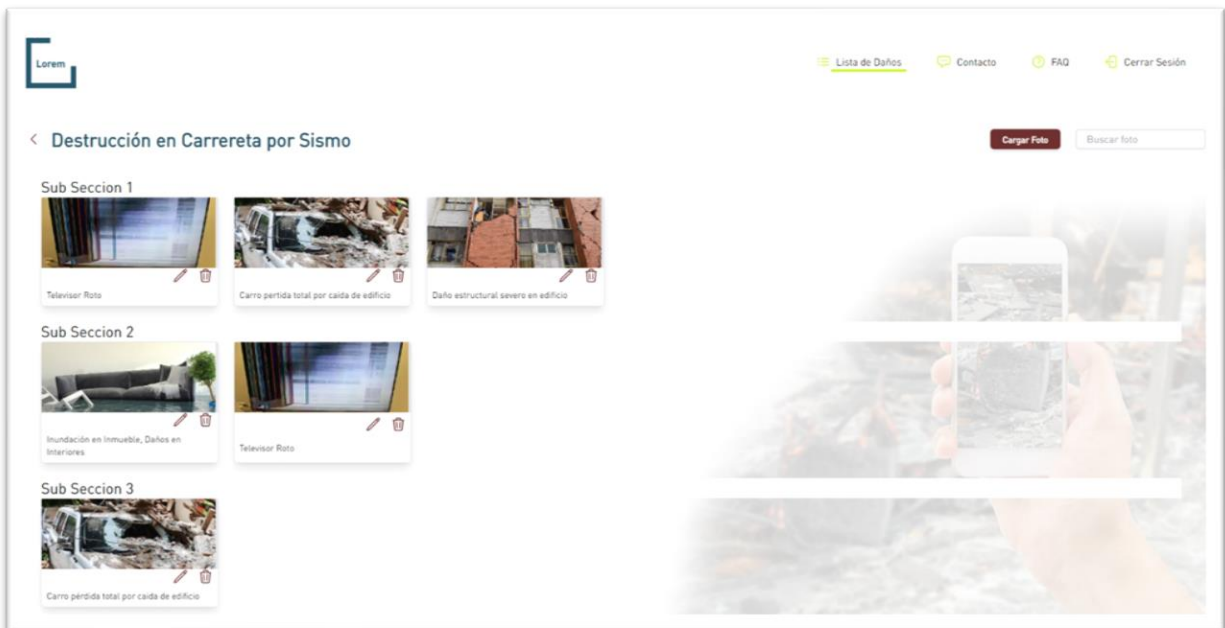


Imagen 58. Galería imágenes (Desktop).



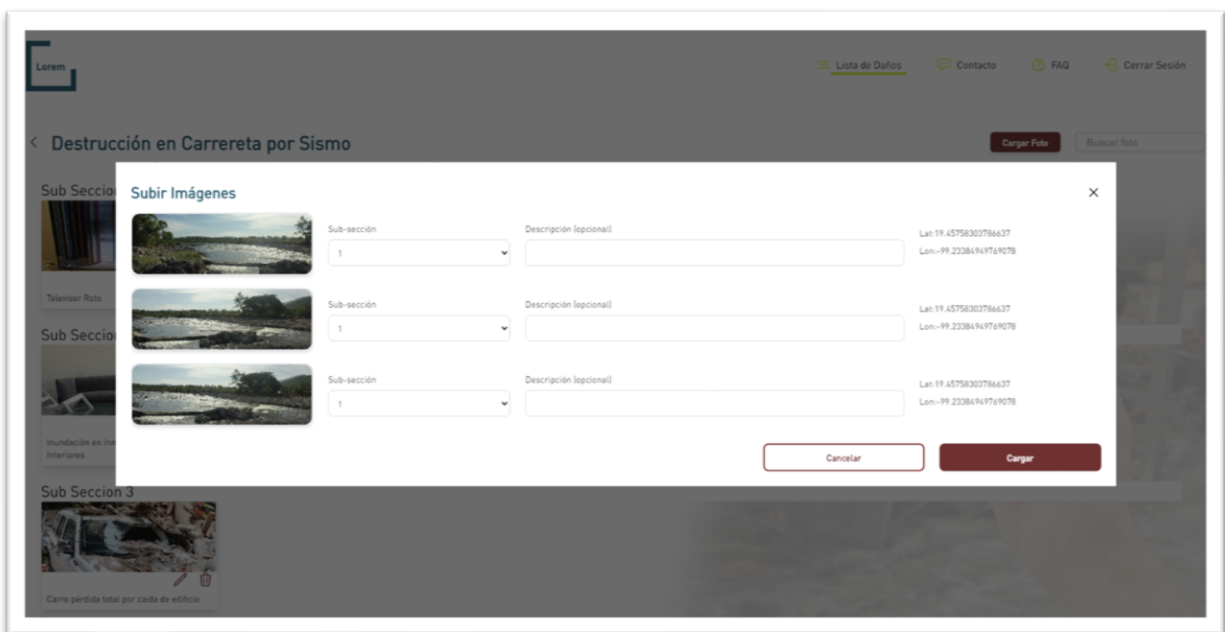


Imagen 59. Carga de imágenes (Desktop).

## VII. Conclusiones.

Durante el proceso de desarrollo de esta aplicación web, se han desarrollado nuevas habilidades y conocimientos que se pusieron en práctica en cada una de sus etapas de creación, se pueden englobar en dos principales ramas, las cuales son diseño y programación.

En cuanto al diseño, una herramienta valiosa que no conocía son los moodboards los cuales contienen información como fuentes, colores y distribución de estos, esto en la práctica es bueno porque en mi experiencia tratar de adivinar el color o el tamaño ya en la programación llega a ser tardado y los resultados al final difieren del diseño oficial; Además de esta herramienta me pareció interesante la teoría del color, es decir, que colores se usan para cada tipo de propósito y/o usuario. Otras de las herramientas que me ayudaron en la creación del diseño fueron las zonas que el usuario ve en los primeros segundos, así también como las zonas que, dependiendo del tipo de dispositivo, más son accesibles, como lo es la ley de Fitt y a su vez el tiempo en que el usuario tiene para recordar ciertas imágenes (ley de Miller).

Hablando ahora de la programación, el aprender un nuevo framework de desarrollo como lo es React fortalece el área de conocimiento y refuerza conceptos que anteriormente tenía de JavaScript, también usar otro framework como lo es Tailwind que últimamente es tendencia en el desarrollo de aplicaciones web, nos pone un peldaño más alto en el mercado laboral, conocer que también hay herramientas o procesos de desarrollo para el front como lo son “mobile first” y “desktop first”, dan un panorama mayor para la toma de decisiones a la hora del desarrollo en la vida profesional, de igual manera se han fortalecido mis conocimientos de css al usar adecuadamente las herramientas grid y flex.

En conclusión, del presente diplomado he obtenido habilidades para poder diseñar de manera más eficiente, interfaces que sean útiles para el cliente final, analizando paso a paso sus necesidades y tener en mente cuales son los estándares en la generación de interfaces, así también he aprendido herramientas nuevas para poder mostrar, documentar y desarrollar estas interfaces para el usuario y darle la mejor experiencia a la hora de utilizarlas.

## VIII. Referencias.

- Amazon Web Services (2023) <https://aws.amazon.com/es/what-is/web-application/>
- BBVA (2023) <https://www.bbva.mx/educacion-financiera/>
- Chubb (2023) <https://www.chubb.com/>
- Diplomado en seguros. CONDUSEF. (2023) <https://educacionfinanciera.condusef.gob.mx/DiplomadoSeguros/>
- Diplomado UX/UI Creando experiencias digitales significativas (2023) Facultad de Artes y Diseño
- Eloquent JavaScript 3rd edition (2018) Marijn Haverbeke.
- El economista.es (2020) <https://www.eleconomista.es/>
- Glosario de Seguros. (2023) <https://www.segurossura.com.co/paginas/glosario.aspx>
- MDN (2023) <https://developer.mozilla.org/>
- MDN Web Docs – CSS (2023) <https://developer.mozilla.org/es/docs/Web/CSS>
- MDN Web Docs - HTML (2023) <https://developer.mozilla.org/es/docs/Web/HTML>
- Mongo DB (2023) <https://www.mongodb.com/nosql-explained>
- React (2023) <https://es.react.dev/>
- Real Academia Española (2023) <https://www.rae.es/>