



INTERFACES GRAFICAS SEGÚN REQUERIMIENTOS DEL PROYECTO

GA6-220501096-AA3-EV03

GUILLERMO ALIRIO CAMACHO JARAMILLO

FICHA: 2627058

TUTOR

REINALDO MARTINEZ LUNA

**SERVICIO NACIONAL DE APRENDIZAJE SENA
CENTRO DE FORMACIÓN GESTIÓN ADMINISTRATIVA
BOGOTÁ D.C FEBRERO – 2024**

Contenido

| | |
|--|----|
| INTRODUCCION | 3 |
| LISTA DE CHEQUEO | 4 |
| 1. REQUERIMIENTOS DEL SISTEMA..... | 5 |
| 1.1. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE | 5 |
| 1.2. Clientes | 5 |
| 1.3. Ordenes de Servicio | 6 |
| 2. APLICACION | 8 |
| 2.1. ARCHIVOS | 8 |
| 2.2. ARBOL DE DIRECTORIOS | 8 |
| 3. INTERFACES | 10 |
| 4. CONCLUSIONES | 16 |
| 5. BIBLIOGRAFIA | 17 |

INTRODUCCION

Una vez definidos los estilos a aplicar en el diseño de un prototipo de aplicativo de software, se debe proceder al diseño como tal del aplicativo, en el cual se apliquen dichos principios definidos de diseño con el contexto real de la aplicación. De esta manera se logra un acercamiento a un MVP Producto mínimo viable que tendrá la abstracción del aplicativo.

Esta actividad contiene la presentación y desarrollo del MVP a modo de diseño del producto en front-end del aplicativo para el control de las actividades de reparación de vehículos de un taller automotriz, partiendo de los requerimientos funcionales y del diseño del mockup.

LISTA DE CHEQUEO

| No. | VARIABLES/INDICADORES DE LOGRO | CUMPLE | | Observaciones |
|-----|--|--------|----|---------------|
| | | SI | NO | |
| 1. | Las interfaces presentadas muestran un diseño usable. | | | 30% |
| 2. | Las interfaces presentadas manejan buena composición de color. | | | 30% |
| 3. | Las interfaces están relacionadas con los <i>mockups</i> presentados anteriormente. | | | 30% |
| 4. | Las interfaces presentadas cumplen con los requerimientos del <i>software</i> a desarrollar. | | | 10% |

1. REQUERIMIENTOS DEL SISTEMA

1.1. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

A continuación, se detallan los requerimientos funcionales obtenidos de los stakeholders:

Agrupados por las siguientes entidades fundamentales. A partir de las estas entidades se inician las reglas de normalización de datos que llevan a creación de nuevas tablas secundarias y sus relaciones.

1. Clientes
2. Vehículos
3. Empleados
4. Ordenes de Servicio

1.2. Clientes

1. Debe Poderse inscribir dos tipos de Clientes: (Personas o Empresas).
2. Se almacenará información del cliente tal como: nombre tipo, ciudad. Dirección, email y número telefónico.
3. Un cliente puede estar asociado a muchas ordenes de servicio.
4. Un cliente puede estar asociado a muchas ordenes de servicio con un mismo vehículo o diferentes vehículos. A su vez cada orden de servicio solo puede contener un Cliente y Un vehículo.
5. Permitirá ingresar una imagen con el rostro del empleado.

2.2. Empleados:

1. Se almacenará información básica como nombre, apellidos, dirección, email y teléfono.
2. Cada empleado debe ser asignado a un tipo de rol, los cuales son: Lavador, Eléctrico, Administrador, Secretaria, Mecánico general y mecánico Diesel.
3. Deberá permitir ingresar información adicional como por ejemplo capacitaciones, certificaciones, habilidades, etc., en un campo de texto largo.

1.3. Ordenes de Servicio

1. Cada ingreso de un vehículo al taller se registrará bajo un número de orden de servicio, el cual iniciará con los siguientes datos: Fecha Ingreso, Cliente, Vehículo, Estado de la orden, kilometraje de ingreso, y observaciones que se tomen del vehículo, las cuales consisten en las solicitudes del cliente.
2. En cada orden de servicio solo se puede ingresar un solo vehículo.
3. Cada orden de servicio solo se puede ingresar un solo cliente
4. Un cliente puede ingresar un vehículo que pertenezca a otro propietario. Existe el caso de compañías clientes que ingresan el mismo vehículo puesto que comparten ciertas responsabilidades con los mismos vehículos debido a pólizas o negociaciones. Por tanto, un mismo vehículo puede ser ingresado varias veces por diferentes clientes. La identificación de vehículos que ingresa un cliente estará determinada por las ordenes asignadas a dicho cliente.
6. Una orden de servicio puede contener a su vez registros de servicios que se prestan al vehículo referente, así como también registros de repuestos que se adicionan.
7. No se requiere administrar existencias de inventario de repuestos, sin embargo, si se desea llevar una trazabilidad de los servicios y repuestos vendidos en cada orden.
8. Tanto los servicios como los repuestos deben clasificarse para temas de analítica.
9. Los servicios que se prestan estarán categorizados tanto por el tipo de servicio como también por el sistema del vehículo al que corresponden.
10. Los repuestos o consumibles que se adicionen al vehículo deben estar especificados, de acuerdo a su presentación (unidad, docena, caja, galón, etc.).
11. Tanto para los servicios como para los consumibles se podrá definir el estado, esta definición corresponde a si el producto o servicio está desactivado porque no se prestará más o está activo para que permita adicionarse en las órdenes.
12. Una orden de servicio debe permitir generar una cotización a dicha orden.
13. Una orden de servicio debe permitir generar una factura enlazada a su número de orden, lo que significa un siguiente flujo.
14. Una orden de servicio debe contener un estado, el cual contendrá identificar si la orden está en ejecución, finalizada o facturada.
15. Deberá tener un panel que permita visualizar las ordenes activas en el taller o filtrar por estado y demás campos que se consideren relevantes como fecha, cliente y vehículo.
16. Se debe poder visualizar el estado de pagos de cada factura donde se identifique los pagos realizados, número de factura, medio de pago de pago, fecha y valor abonado.

17. Vehículos

Antes de ingresar un vehículo a una orden, este debe crearse en el sistema, por lo que se tendrá una base de datos de vehículos ingresados al taller. Los vehículos deben ingresarse con los

siguientes datos:

- Placa, Marca, línea, Año, Color, Tipo de Carrocería. Las marcas y tipos de carrocerías se almacenarán en listas para ser seleccionados en cada ingreso respecto al vehículo.
-
- No se requiere ingresar datos del propietario, puesto que solamente se seleccionará vehículos cada vez que ingresen al taller y estarán asociados en cada ingreso con el cliente que ingrese la orden como regla de negocio.

2. APLICACION

2.1. ARCHIVOS

Index.html: Contiene el formulario de logueo o log-in del proyecto.

Comprendiendo las funcionalidades CRUD como actividades con registros de datos tales como crear, leer, actualizar y eliminar, las cuales hacen parte del backend (las cuales no se incluyen aquí), pero se activan desde botones del front-end se presentan los siguientes módulos.

Módulos: Se estructuran los nombres de los módulos los cuales serán:

- clientes.html: Contiene las operaciones crud de clientes.
- vehiculos.html: contiene las operaciones crud de vehículos.
- gestion_os.html: Contiene las operaciones crud de ordenes de trabajo.
- funcionarios.html: contiene las operaciones crud de funcionarios.
- maestro_inventario.html: contiene las operaciones crud de inventario.
- maestro_servicios.html: contiene las operaciones crud de los servicios.

En esta actividad, se adicionan los módulos y campos reales al aplicativo, remplazando el mockup por una versión evolucionada del desarrollo, manteniendo los estilos definidos en actividad anterior. Adicionalmente se incluyen las imágenes de fondo de presentación del aplicativo en el formulario de login.

2.2. ARBOL DE DIRECTORIOS

El contenido se desarrolló con la siguiente estructura de directorios.

| CARPETA | CONTENIDO |
|--------------|--|
| CarRepairPro | Carpeta del Proyecto. Esta contiene las subcarpetas assets, y contiene los módulos del proyecto en html. |
| assets | Carpeta que contiene los componentes adicionales estáticos tales como imágenes, y código css |
| Img | Carpeta con contenidos tipo imágenes |
| Css | Carpeta que contiene el código css. |
| | |

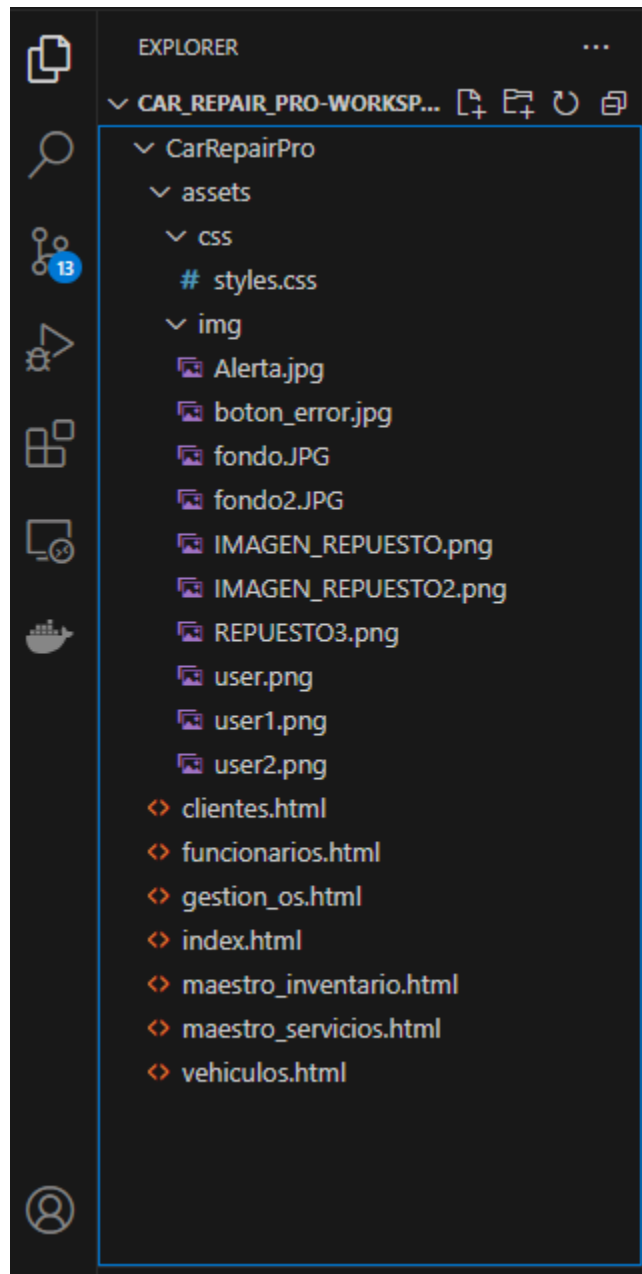


Ilustración 1 Carpetas del proyecto

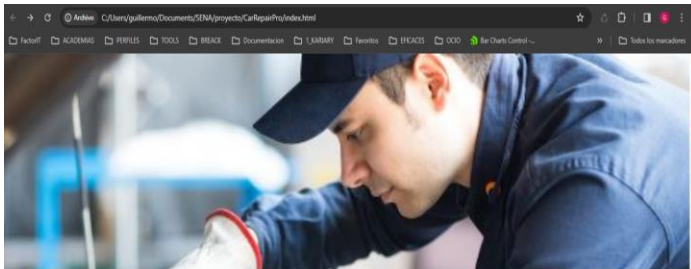
Styles.css: En la carpeta assets se encuentra el archivo styles.css, el cual contiene la configuración de estilos que se aplicará a todos los elementos, la cual consta de títulos, campos, botones, menú, fondos tipos de letras y tamaños, tamaños para responsive, etc. Los estilos se desarrollaron teniendo en cuenta los colores corporativos en un función de un aspecto minimalista.

Módulos: Los archivos correspondientes con los módulos para cada contenido se encuentran enumerados almacenados cada uno con su nombre de acuerdo a la funcionalidad (ej: clientes.html, funcionarios.html, etc).

3. INTERFACES

DESKTOP

Login



Gestión de Clientes

Gestión de Vehículos

RESPONSIVE

Login



Gestión de Clientes

Gestión de Vehículos

Gestion de Personal Interno

Gestión de Personal Interno (Movil)

Módulo de Inventario

Módulo de Inventario

Módulo de Servicios

Modulo de Servicios

Código HTML:

El elemento body contiene el siguiente código con la imagen de fondo anclada en la etiqueta img:

```
<body>
  <header>
    <div class="fondo">
      
    </div>
  </header>
  <div class="contenido">
    <div class="form-login">
      <form action="clientes.html">
        <div class="caja">
          <label for="usuario" class="label-login">Usuario</label>
          <input type="text" name="usuario" class="campo">
        </div>

        <div class="caja">
          <label for="password" class="label-login">Contraseña</label>
          <input type="password" name="password" class="campo">
        </div>

        <a href="">Olvidó su contraseña</a>

        <input type="submit" class="btn btn-submit" name="submit" value="Ingresar">
      </form>
    </div>
  </div>
</body>
```

Se mantiene la etiqueta “contenido” para almacenar el código del cuerpo del formulario y los componentes. Los elementos div nombrados con el estilo caja con las etiquetas y campos del formulario.

Se mantienen los estilos definidos del botón “Ingresar”, como un botón submit con estilos btn btn-submit.

Se mantienen los estilos definidos en el mockup para los campos.

Para los títulos se adiciona la siguiente clase

```
.title {
  display: block;
  text-align: center;
  font-size: 30px;
  padding-top: 10px;
  color: #085394;
  font-family: Verdana, Tahoma, sans-serif;
}
```

Se adiciona estilos a las etiquetas del login:

```
.contenido .label-login{
  display: flex;
  margin-top: 5px;
  margin-bottom: 5px;
  text-align: center;
  font-size: 16px;
  font-weight: 500;
  font-family: Arial, Helvetica, sans-serif;
  color: rgb(61, 60, 60);
}
```

```
.campo{
  border: 1px solid rgba(240, 223, 223, 0.877);
  background-color: white;
  border-radius: 4px;
  padding: 4px;
  font-size: 20px;
```

A los botones se aplica el siguiente estilo que permite resaltarlos al ubicar el puntero sobre ellos:

```
.btn-submit:hover{
  background-color: #073e6e;
}
```

Para las etiquetas se crea la siguiente clase:

```
.imagen-form label{
  text-align: center;
  padding: 3px;
  background-color: darkgray;
}
```

| | | |
|--|--|--|
| <p>Para pantallas hasta de ancho: 346px:</p> <pre> /* hasta 346px */ @media screen and (max-width: 346px){ .caja { width: 250px; display: inline-block; margin: auto; } select{ width:250px; } textarea{ width:250px; } .form-principal{ display: block; } } </pre> | <p>Para pantallas hasta 768 px de ancho:</p> <pre> 295 /*-- hasta 768 */ 296 @media screen and (max-width: 768px){ 297 .contenido { 298 flex-direction: column; 299 } 300 301 aside { 302 width: 100%; 303 order:1; 304 height: auto; 305 } 306 307 .work-area{ 308 width:100%; 309 order:2; 310 } 311 .campos{ 312 display: block; 313 } 314 315 </pre> | <p>Para pantallas de más de 768 px de ancho:</p> <pre> 318 /* cuando pasa los 768px */ 319 @media screen and (min-width: 768px){ 320 .fondo img { 321 height:600px; 322 } 323 324 .form-principal{ 325 display: flex; 326 } 327 328 } 329 </pre> |
|--|--|--|

A nivel de estilos, se maneja los estilos compartidos del menú inicio con algunas modificaciones de adaptación por elementos adicionales como título, campos de diferente tipo de dato.

Se crea un elemento Aside, que contiene el menú y sus opciones de navegación.

El menú en formato de pantalla grande contiene el 20% de la pantalla:

```

aside{
  width:20%;
  background-color: #7f7f7f;
  height:100vh;
}

```

Algunos de los estilos que se adicionan al elemento aside son:

```
117 }
118
119 .contenido-aside .titulo-menu {
120     background-color: rgba(19, 16, 16, 0.473);
121     color: rgb(182, 176, 176);
122     font-size: larger;
123     padding: 10px;
124     font-weight: bold;
125     text-align: center;
126 }
127
128 .contenido-aside .linea-normal{
129     background-color: inherit;
130     font-size: large;
131     text-align: center;
132     padding: 10px 0 10px;
133     cursor: pointer;
134     line-break: auto;
135 }
136
137 .contenido-aside .linea-normal a{
138     text-decoration: none;
139     background-color: inherit;
140     color: inherit;
141     cursor: pointer;
142 }
143
144 .contenido-aside .linea-normal.padre{
145     background-color: rgba(19, 16, 16, 0.473);
146     color: rgb(182, 176, 176);
147 }
148
```

4. CONCLUSIONES

Al desarrollar esta actividad, se logra obtener las siguientes conclusiones.

El diseño de un mockup debe venir acompañado de una documentación que permita comprender su estructura así como destacar aspectos importantes que se consideren que deben tenerse en cuenta.

Es importante para el diseño del mockup tener en cuenta las funcionalidades del aplicativo de forma que sea eficaz para las labores que se requieren y eficiente para generar una experiencia aceptable por parte del cliente.

Las hojas de estilo en cascada permiten adicionar aspectos visuales que se adapten al gusto del cliente, así como diferentes tipos de pantalla, incrementando y mejorando la experiencia de usuario y usabilidad de la aplicación.

5. BIBLIOGRAFIA

Apellido, Inicial del nombre. (Fecha de publicación o actualización). Título del artículo o página.
Recuperado de URL

Ejemplo:

- SENA, Arquitectura de aplicaciones móviles nativas. Tomado de https://sena.territorio.la/content_mooc.php?idMateria=2186388&social=25323088
- SENA, Diseño de interfaces web. Tomado de: https://sena.territorio.la/content_mooc.php?idMateria=2186388&social=25323088
- W3Schools. Página de tutoriales prácticos. Tomado de <https://www.w3schools.com/css/>