Documentación Sistema de Contabilidad

Este documento tiene como propósito el comprender de mejor forma el desarrollo del sistema de contabilidad amexport, que actualmente se encuentra activo en el servidor Google Cloud Platform.

****** Herramientas y Tecnologías

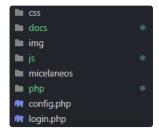
Las principales herramientas y tecnologías ocupadas para el desarrollo fueron las siguientes:

		≡ Uso	≡ Ejemplos en el sistema
1	HTML (LENGUAJE DE ETIQUETADO) FRONT-END	Lenguaje de etiquetado usado para la creación general de las vistas y formularios.	Todas las vistas, catálogos, formularios, tablas, etc.
2	PHP (LENGUAJE DE PROGRAMACIÓN) BACK-END	Lenguaje usado principalmente del lado del backend pero también del lado del front haciéndolo más dinámico.	Conexión con la base de datos y todas las consultas a la misma. Tablas con datos, vista detalle en los catálogos. Guardado de los formularios y autenticación de usuarios.
3	JavaScript (LENGUAJE DE PROGRAMACIÓN) FRONT-END	Se utilizó para junto con jquery y AJAX para el comportamiento dinamico de las páginas y el sistema en general.	Evita que se recarguen las páginas después del guardado o consulta de información, envía alertas, intermediario entre el front y el back. Botones con funciones, formularios y tablas dinámicas.
4	Bootstrap 5 (FRAMEWORK) FRONT-END	Framework implementado en el front para da mejor diseño al sistema en general.	Diseño de Navbars, elementos, botónes, tablas, modales, etc.

5	CSS (LENGUAJE DE ESTILOS EN CASCADA) FRONT-END	Se utilizó junto a Bootstrap para dar diseño a las interfaces.	Diseño más detallado para los elementos visuales, como el cambio de color en el fondo del login, inputs personalizados sin bordes, etc.
---	--	--	--

Administración del Proyecto

La jerarquía de carpetas usada es la siguiente:



Se utilizó un pseudo paradigma para el desarrollo basado en MVC, por lo que la mayoría de los módulos tienen:

- · Vista HTML con ciertos elementos PHP
- Archivo Script, usualmente AJAX para manejar las peticiones, los eventos asíncronos y la lógica del flujo de trabajo.
- · Consultas backend con PHP, mayormente inserciones a la base de datos en caso de los formularios y/o consultas de información a la mísma.

```
1 sequenceDiagram
 2
        participant Usuario
 3
        participant Vistas as vistas/formulario.php
        participant JS as js/guardar_Solicitudes.js
        participant Backend as modulos/guardar_solicitud.php
 5
 6
        participant MySQL
 7
 8
        Usuario->>Vistas: Interactúa con formulario
 9
        Vistas->>JS: Lanza AJAX
        JS->>Backend: POST con datos del formulario
10
        Backend->>MySQL: Ejecuta consultas SQL (CRUD)
11
12
        MySQL-->>Backend: Resultado (éxito o error)
13
        Backend-->>JS: Respuesta JSON
        JS-->>Usuario: Muestra éxito o error en pantalla
14
15
16
17
```

Las carpetas contienen lo siguiente:

- · Css: Hojas de estilos que se ocupan en todo el sistema. Se ocupó Bootstrap pero se personalizan algunos componentes como las tablas.
- Docs : Aquí se guardan los documentos subidos por el usuario al sistema
- Img: Las imágenes que ocupa el sistema como el logotipo o el favicon se encuentran aquí.
- Js : La mayor parte de la lógica del sistema se encuentra en esta carpeta la cual contiene los eventos AJAX para la asincronía de la página.
- Micelaneos: En esta carpeta se pueden encontrar archivos de apoyo como el que genera las encriptaciones HASH para las contraseñas.
- Php: Es la carpeta que almacena los archivos de backend en su mayoría, esta carpeta está dividida en dos, módulos y vistas. La parte de módulos se encarga de las consultas de inserción, consulta de datos y editado de información. En vistas se almacenan todos los archivos de las páginas visibles del sistema como formularios, tablas y catálogos.

🧩 Módulos

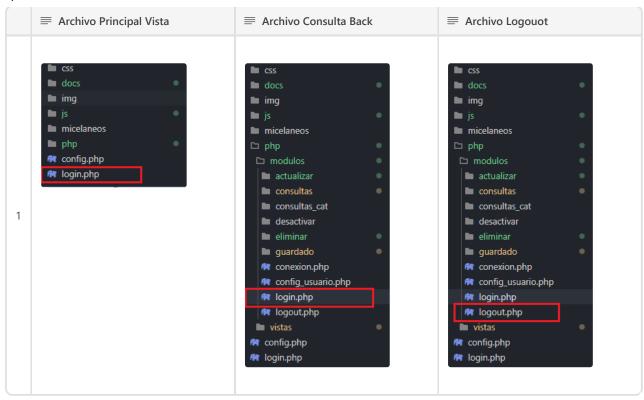


Login

Ya explicado el paradigma de desarrollo, se comenzará la explicación de cada uno de los módulos desarrollados, así como la explicación de su código respectivamente, comenzando con el Login.



Este módulo se apoya con la tabla usuarios de la base de datos amexportdb, y se ocupan las columnas Nombre y Hash para llenar el formulario con Usuario y Contraseña respectivamente. Las contraseñas, como se mencionó anteriormente, están encriptadas con HASH por lo que textualmente no se encuentran en la base de datos, esto es así por cuestión de seguridad, por lo que cada usuario deberá memorizar o apuntar su contraseña.



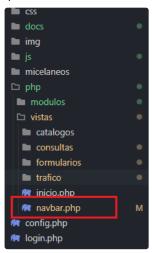
El archivo se encuentra en la carpeta raíz del proyecto, llamada Contabilidad. Este archivo funge como index en el servidor por lo que es el principal. En ese archivo se puede encontrar la parte visual del módulo, así como el script del envío del formulario y manejo de errores. El archivo login.php que está dentro de php/módulos/ es el encargado de hacer la verificación con password verify y retorna una respuesta al login de la raíz principal y dando inicio a las SESSION o sesiones, las cuales deben iniciar en todos los módulos para mayor seguridad del sistema.

```
1 <?php
2 session_start();
3
4 if (!isset($_SESSION['usuario_id'])) {
5     header('Location: /portal_web/Contabilidad/login.php'); // Ruta desde la raíz del servidor web
6     exit;
7 }
8
9</pre>
```

Este fragmento se coloca en el inicio de cada módulo. Verifica la sesión y redirige al login en caso de no haber una sesión activa.

Navbar

Se utilizaron dos barras de navegación, el diseño es una mezcla de Bootstrap y css.

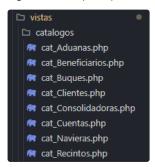


Se encuentra dentro de las vistas en php. Se importa de igual forma en cada uno de los archivos de vistas.

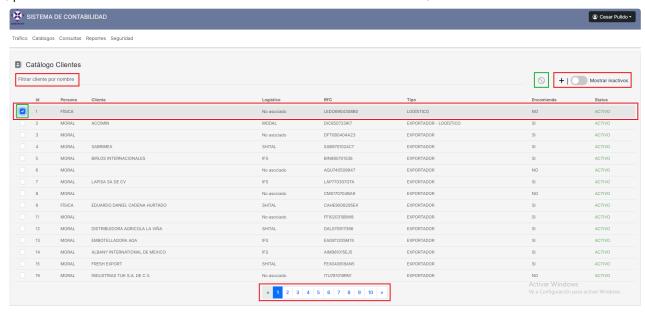
```
1 //Inclusión de la navbar en las vistas
2 <?php
3 include($_SERVER['DOCUMENT_ROOT'] . '/portal_web/Contabilidad/php/vistas/navbar.php');
4 ?>
5
6
```

Catálogos

Los catálogos se dividen en la tabla con información y la vista detalle en donde se pueden hacer cambios a la información de los registros. La carpeta que contiene las vistas de los catálogos es la siguiente:



Siguen la nomenclatura "cat_(Nombre)". A continuación, se detallarán más las funcionalidades que se encuentran dentro de los catálogos.



Las funciones que tiene cada formulario son las siguientes:

- Filtrado de información por nombre, dependiendo que valor recibe ese atributo, en este caso de "Clientes" sería razonSocial_exportador en la base de datos.
- Checkbox para desactivar un registro. Cada registro cuenta con un checkbox que al seleccionarlo aparece un nuevo botón en la parte superior derecha para cambiar su status en la base de datos a 0 o inactivo.
- Agregar un nuevo registro, representado con un "+" redirige al formulario del catálogo.
- Mostrar inactivos, este switch realiza la función que dice su nombre al mostrar en la tabla los registros con status 0 por que ciertos catálogos por defecto solo muestran los activos.
- Paginación, en la parte inferior se muestran los botones de paginación los cuales muestran un numero limitado de registros por página, por defecto son 15.

Filtrado de Información

Carpeta: /php/módulos/consultas_cat/tabla_clientes.php. Esta es la ruta en la que se encuentran las consultas para catálogos, solo cambia el nombre del archivo .php la ruta es la misma para todos los catálogos.

```
1 <?php
 2 include once( DIR . '/../conexion.php');
 3
 4 // Número de registros por página
 5 $registrosPorPagina = 15;
 6
  // Determinar el número de la página actual
 8 $paginaActual = isset($_GET['pagina']) ? (int)$_GET['pagina'] : 1;
   $inicio = ($paginaActual - 1) * $registrosPorPagina; // Índice de inicio para la consulta
10
11 // Consulta para obtener los clientes (solo 20 registros por página)
12 $filtro = isset($_GET['filtro']) ? trim($_GET['filtro']) : '';
  $filtroLike = '%' . $filtro . '%';
14
  $sql = "SELECT id01clientes exportadores, tipoClienteExportador, nombreCorto exportador,
   logistico_asociado, rfc_exportador, tipo_cliente, statusEcomienda_exportador, status_exportador
16
           FROM 01clientes_exportadores WHERE status_exportador = 1";
```

```
17
   if ($filtro !== '') {
       $sql .= " AND nombreCorto_exportador LIKE :filtro";
19
20
21
22 $sql .= " LIMIT :inicio, :registrosPorPagina";
23
24
   $stmt = $con->prepare($sql);
25
26 // Bind del filtro si se usa
   if ($filtro !== '') {
28
       $stmt->bindValue(':filtro', $filtroLike, PDO::PARAM_STR);
29
   }
30
31 $stmt->bindValue(':inicio', $inicio, PDO::PARAM_INT);
32 $stmt->bindValue(':registrosPorPagina', $registrosPorPagina, PDO::PARAM_INT);
33 $stmt->execute();
34 $clientes = $stmt->fetchAll(PDO::FETCH_ASSOC);
35
36 // Consulta para contar el total de registros
37  $stmtTotal = $con->prepare("SELECT COUNT(*) FROM 01clientes_exportadores");
38 $stmtTotal->execute();
39 $totalRegistros = $stmtTotal->fetchColumn();
40
   // Calcular el total de páginas
   $totalPaginas = ceil($totalRegistros / $registrosPorPagina);
43
   // Calcular el bloque de páginas a mostrar (de 10 en 10)
45 $inicioBloque = floor(($paginaActual - 1) / 10) * 10 + 1;
   $finBloque = min($inicioBloque + 9, $totalPaginas);
47 ?>
48
49
50
```

Tabla

La tabla con información se muestra más abajo, para este ejemplo se usará la tabla buques. Esta tabla es la que se importa en el archivo cat_Buques. Internamente ya lleva la opción para el evento del checkbox como se ve en la línea 13. La consulta que se hizo previamente trae los datos y se llaman dentro de las etiquetas td. Los estilos de las columnas de estas tablas se encuentran en el archivo style.css en la carpeta css.

```
9
      </thead>
10
      11
         <?php if ($buques): ?>
12
            <?php foreach ($buques as $buque): ?>
13
               '../../modulos/consultas_cat/detalle_buques.php?id=<?php echo $buque['Id']; ?>';}" style="cursor:
   pointer;">
14
                  <input class="form-check-input mt-1 chkBuque" type="checkbox" value="<?php</pre>
15
   echo $buque['Id']; ?>" aria-label="Checkbox for following text input">
16
17
                  <?php echo $buque['Id']; ?>
18
                  <?php echo $buque['Nombre']; ?>
               19
20
            <?php endforeach; ?>
21
         <?php else: ?>
22
            No se encontraron registros
23
         <?php endif; ?>
24
      25
```

Paginación

El siguiente código muestra el fragmente en el que se muestra la paginación del catálogo buques. Es similar para todos los demás catálogos.

```
<nav aria-label="Page navigation example" class="d-flex justify-content-center">
2
     3
        ">
          <a class="page-link" href="?pagina=<?php echo $paginaActual - 1; ?>" aria-
  label="Anterior">
5
             <span aria-hidden="true">&laquo;</span>
6
          </a>
        7
9
        <?php for ($i = $inicioBloque; $i <= $finBloque; $i++): ?>
10
          ">
             <a class="page-link" href="?pagina=<?php echo $i; ?>"><?php echo $i; ?></a>
11
12
          13
        <?php endfor; ?>
14
15
        ">
          <a class="page-link" href="?pagina=<?php echo $paginaActual + 1; ?>" aria-
16
  label="Siguiente">
             <span aria-hidden="true">&raquo;</span>
17
18
          </a>
        19
20
     21 </nav>
```

Desactivación

Como se vio previamente en el fragmento donde se genera la tabla, también lo hacen los checkboxs y la lógica para eso se encuentra en la carpeta /Js/desactivar_Buque.js en este caso.

```
1 document.addEventListener('DOMContentLoaded', function () {
 2
       // Inicializar tooltips
 3
       var tooltipTriggerList = [].slice.call(document.querySelectorAll('[data-bs-toggle="tooltip"]'));
       tooltipTriggerList.map(function (tooltipTriggerEl) {
 5
            return new bootstrap.Tooltip(tooltipTriggerEl);
 6
       });
 7
 8
       const btnDesactivar = document.getElementById('btnDesactivar');
 9
       const checkboxes = document.querySelectorAll('.chkBuque');
10
11
       // Mostrar/ocultar botón según selección
12
       function actualizarBoton() {
13
            const anyChecked = Array.from(checkboxes).some(cb => cb.checked);
            btnDesactivar.style.display = anyChecked ? 'inline-block' : 'none';
14
15
       }
```

Este fragmento muestra cómo se maneja el cambio de estado del botón que se muestra cuando se selecciona un checkboxs. En ese mismo archivo js se encuentra la petición ajax.

Cabe aclarar que esta misma estructura se ocupa para todos los catálogos.