



Universidad Tecnológica de Puebla

Organismo Público Descentralizado del Gobierno del Estado de Puebla

ORGANIZACIÓN Y DESARROLLO PARA FORTALECER EL DEPARTAMENTO DE SISTEMAS

QUE PARA OBTENER EL TÍTULO DE INGENIERÍA EN

DESARROLLO Y GESTION DE SOFTWARE

PRESENTA:

MARIA MONTSERRAT TLACHI ANZURES

UTP0009259

ASESORES DE ESTADÍA:

HECTOR DE SAMPEDRO POBLANO

ASESORES DE ESTADÍA EXTERNO:

FERNANDO GÚZMAN ORTIZ

DE ABRIL DE 2022



ÍNDICE

ÍNDICE DE IMÁGENES	3
INTRODUCCIÓN.....	4
CAPÍTULO I: ASPECTOS GENERALES Y MARCO CONTEXTUAL.	6
1.1 A CERCA DE LA EMPRESA.....	6
1.1.1 MISIÓN.....	6
1.1.2 VISIÓN	6
1.2 FILOSOFÍA DE LA EMPRESA.....	6
1.3 OBJETIVOS GENERALES	6
1.3.1 PLANTEAMIENTO DEL PROBLEMA	6
1.3.2 HIPÓTESIS O PROPUESTA DE SOLUCIÓN.....	7
1.4 OBJETIVOS DEL PROYECTO	7
1.4.1 OBJETIVO GENERAL	7
1.4.2 OBJETIVOS ESPECIFICOS	7
1.5 JUSTIFICACIÓN	8
1.6 ALCANCES Y LIMITACIONES	8
CAPÍTULO II: MARCO TEÓRICO	10
PhpMyadmin.....	10
Framework	10
HTML y XHTML	11
CSS.....	11
JavaScript, jQuery y AJAX.....	12
PHP.....	12
Modelo, Vista, Controlador, (MVC)	13
XAMPP.....	13
Sistema de login y registro de usuarios	13
Vistas	14
Template	14
PDO	14
CAPÍTULO III: DESARROLLO Y CASO PRÁCTICO DE LA EMPRESA	16
3.1 DESCRIPCIÓN DEL SISTEMA.....	17
3.2 DESARROLLO / PROGRAMACIÓN.....	18

3.2.1	CONSTRUCCIÓN DEL SISTEMA	18
3.3.2	SISTEMA DE LOGIN.....	18
3.2.2	INTERFAZ DE RESUMEN DE VENTAS (GRÁFICAS).....	23
3.2.3	INTERFAZ LISTA DE USUARIOS.....	23
3.2.4	INTERFAZ DE MÓDULO DE CLIENTES	32
3.2.5	INTERFAZ DE MÓDULO DE VENTAS.....	34
3.2.6	INTERFAZ DE MÓDULO PRODUCTOS.....	37
3.2.7	INTERFAZ DE MÓDULO VENTAS-PRODUCTOS.....	39
3.2.8	INTERFAZ DE MÓDULO CITAS.....	40
CONCLUSIÓN.....		45
BIBLIOGRAFIA.....		46

ÍNDICE DE IMÁGENES

Ilustración 2 Estructura del proyecto	18
Ilustración 3 Login	19
Ilustración 4 Código de la vista login	20
Ilustración 5 Gráficas y resumen.....	23
Ilustración 6 Agregar usuario.....	24
Ilustración 7 Mensaje de error.....	24
Ilustración 8 Ingreso de datos	25
Ilustración 9 Mensaje de éxito	25
Ilustración 10 Registros	26
Ilustración 11 Botones de archivos de descarga.....	26
Ilustración 12 Descargas	26
Ilustración 13 Descarga CSV	26
Ilustración 14 Descarga PDF.....	27
Ilustración 15 Tabla de registro y datos	28
Ilustración 16 Acción Ver.....	29
Ilustración 17 Actualización de datos	30
Ilustración 18 Mensaje "Actualización correcta"	30
Ilustración 19 Rectificación de cambios	31
Ilustración 20 Eliminar registro	31
Ilustración 21 Confirmado de registro eliminado	32
Ilustración 22 Evidencia de eliminado.....	32
Ilustración 23 Interfaz de clientes	33
Ilustración 24 Agregar cliente (modal)	33
Ilustración 25 Modal ver cliente.....	34
Ilustración 26 Interfaz de ventas	34
Ilustración 27 Modal "agregar venta"	35
Ilustración 28 Mensaje éxito (ventas).....	35
Ilustración 29 Modal "ver venta"	37
Ilustración 30 Interfaz productos	38
Ilustración 31 Modal "agregar producto".....	38
Ilustración 32 Interfaz Ventas-Productos.....	39
Ilustración 33 Modal Ventas-Productos	39
Ilustración 34 Interfaz Modulo Citas	40
Ilustración 35 Modal Agregar Cita.....	40
Ilustración 36 Citas en calendario.....	41
Ilustración 37 Botones de acciones.....	41
Ilustración 38 Mensaje de éxito citas.....	42
Ilustración 39 Eliminar cita	42

INTRODUCCIÓN

En esta tesina, relate el proceso de la realización del proyecto “Agenda digital de ventas”, para obtener el título de Ingeniero en Desarrollo y Gestión de Software, en Tecnologías de la Información.

Este proyecto será implementado en la empresa de “EasyCode”, la cual está encargada de venderla a sus clientes o para uso de la misma empresa.

Todas las compañías cuentan con procesos administrativos, estos llegan a ser tediosos o cansados y también no cuentan con un control detallado si no se llevan a cabo de la manera correcta esto puede afectar la economía de las empresas, algunas de ellas continúan con sus métodos administrativos manuales mientras algunas otras ya cuentan con herramientas tecnológicas, pero no sabiendo sacarles mayor provecho a estas, otras se han apegado en su totalidad a las tecnologías de la información y eso es lo que busca este proyecto de estadía, ayudar a ahorrar tiempo, ganar más dinero, ser adaptativo y ganar popularidad al implementar las nuevas tecnologías que están al alcance de todos.

EasyCode, cuenta con experiencia y busca mejorar día a día los procesos administrativos de sus clientes, es por ello, que este proyecto almacenará datos de los productos, ventas, clientes, citas entre otras características más, el sistema piloto se está desarrollando para una tienda de herramientas.

En el documento se podrán ver los puntos clave para este proyecto en diferentes capítulos, en cada uno de ellos se explica para que el uso de herramientas y parte del proceso del desarrollo del sistema para poder tener evidencia de lo que se realizó.

CAPÍTULO I

ASPECTOS GENERALES

CAPÍTULO I: ASPECTOS GENERALES Y MARCO CONTEXTUAL.

1.1 A CERCA DE LA EMPRESA

Trabajamos con pasión dando la mejor calidad en el servicio. Siempre con el deseo de seguir tomando nuevos retos para crecer en el mercado, buscando empresas en crecimiento orientadas al logro.

1.1.1 MISIÓN

Ser para nuestros clientes un aliado estratégico que les ayudara con el crecimiento de sus negocios, para posicionarlos en la mente de su público objetivo mediante estrategias afectivas y con una inversión funcional.

1.1.2 VISIÓN

Ser una Agencia Integral, de las mejores de Puebla, ofreciendo estrategias de comunicación efectivas que satisfagan totalmente las necesidades de todos nuestros clientes.

1.2 FILOSOFÍA DE LA EMPRESA

Easy Code está formada por un equipo de trabajo joven y altamente capacitado el cual posee frescura e innovación, teniendo como objetivo asegurarles a nuestros clientes un servicio vanguardista. Empleando a nuestro favor las tendencias de Marketing Digital, el uso de redes sociales, medios digitales y la captación de mercados jóvenes.

1.3 OBJETIVOS GENERALES

Nuestro objetivo es ofrecer los siguientes servicios en conjunto: Mercadotecnia, Diseño y Programación, con el propósito de otorgar una asistencia más completa, accesible y fácil.

1.3.1 PLANTEAMIENTO DEL PROBLEMA

De acuerdo a las necesidades de la empresa EasyCode se me encargó realizar una agenda digital para la organización y seguimiento de las ventas dentro de la empresa, para mejor organización de las ventas y clientes que frecuentan.

Las citas, las ventas, clientes etc., serán creadas para el funcionamiento de la aplicación y se crearán más en caso de ser necesarias para un completo funcionamiento de la aplicación. Estos módulos tendrán operaciones CRUD y se visualizarán los datos al iniciar la página para que el Administrador pueda manipular la información visualizada.

Las citas o ventas agendadas serán de fácil visualización para el administrador y se clasificarán de acuerdo a su importancia, en este caso con uso de colores, para que sea mucho más fácil ubicarlas al momento de filtrarlas. Se podrá ver la información de estas en el calendario para darle un mejor seguimiento.

1.3.2 HIPÓTESIS O PROPUESTA DE SOLUCIÓN

Se realizará una Aplicación Web que cumpla con los requisitos básicos para una agenda digital de ventas, implementando los complementos que requiera la empresa o el cliente. Haciendo uso de editores de código como Visual Studio Code; Se buscará estructurar la base del proyecto mientras que se utilizaran múltiples complementos o plugins para completar dicha tarea.

Durante el desarrollo, el asesor externo evaluara cada sección; de esta manera se ira probando y realizando correcciones.

Al final se realizarán pruebas en producción donde se verá el verdadero funcionamiento del sistema y al terminar, el CEO decidirá el rumbo del proyecto, si el sistema requiere hacer algún cambio o corrección, volverá a el área de desarrollo para hacer las modificaciones y volver a probar todo su funcionamiento.

1.4 OBJETIVOS DEL PROYECTO

1.4.1 OBJETIVO GENERAL

Durante el desarrollo de este proyecto se intentará realizar el mayor número de actividades posibles que sean asignadas por la empresa “EasyCode” para tratar de completar el proyecto de la Agenda Digital de Ventas. Para esto lo primero será la creación de una Base de Datos para utilizarla de forma local, con esto usaremos una plantilla básica de la empresa para crear el modelo, vista, controlador (MVC) para poder llamar los datos y mostrarlos en la Aplicación Web haciendo uso de PDO.

Igualmente se comenzará a crear módulos específicos para los administradores y asistentes para poder darles seguimientos tanto a las ventas o actividades que tienen asignados y ver sus ventas o citas reflejadas en un calendario digital donde podremos ver la información de dichas citas/ventas. Con esa información también se podrá evaluar a los empleados y ver su rendimiento.

1.4.2 OBJETIVOS ESPECIFICOS

Proyecto:

- Estructuración MVC para el manejo de los Datos y su vista.

- Se utilizará PHP/PDO para la extracción y manejo de la información de la Base de Datos
- Uso de un calendario digital para dar seguimiento a las citas de ventas asignadas a los clientes y sus fechas establecidas para poder visualizarlas en dicho calendario.

Personales:

- Aprender el uso de la herramienta de PHP con PDO
- Aprobar estadía
- Continuar con la experiencia participando en el desarrollo de esta página web

1.5 JUSTIFICACIÓN

Ante la necesidad de tener un mejor control en la facturación, cotización y servicio para los clientes, se requiere el desarrollo de una agenda digital de ventas. ya que este sistema se hace cargo de las distintas operaciones y control de recursos dentro de la empresa EasyCode.

Para cumplir con lo que la empresa pide, se trabajará con herramientas de desarrollo como: el framework de PHP que está basado en MVC, un servidor local que soporte bases de datos en MySQL (en este caso XAMPP), Visual Studio Code que es un editor de código.

Se generará una base de datos la cual estará conectada con nuestro proyecto de desarrollo, en el framework, lo primero será configurar las librerías que se utilizarán, el nombre del proyecto para poder correrlo en el localhost y agregar los datos para la conexión con la base de datos, se crearán las vistas que contendrán los formularios, las alertas, diseño en general y algunos archivos multimedia, se llamarán en el controlador para poder visualizarlas, validarlas y conectarlas con el modelo, y en el modelo se crearán las consultas necesarias para poder insertar datos en nuestra base de datos.

1.6 ALCANCES Y LIMITACIONES

Alcances

- Control de usuarios, manejo de tipos de usuarios.
- Control de las sesiones que permitirá ver quien está dentro del sistema.

Limitaciones

- El sistema será solo para los clientes y personal de la empresa.
- El sistema solo permitirá realizar operaciones establecidas.
- Para ingresar al sistema se necesitará de internet.

CAPÍTULO II

MARCO TEÓRICO

CAPÍTULO II: MARCO TEÓRICO

PhpMyadmin

PHPMYAdmin es un software de código abierto, diseñado para manejar la administración y gestión de bases de datos MySQL a través de una interfaz gráfica de usuario. Esta aplicación nos permitirá realizar las operaciones básicas en base de datos MySQL, como son: crear y eliminar bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar sentencias SQL, administrar claves de campos, administrar privilegios y exportar datos en varios formatos. La función de exportar datos se emplea muchas veces para realizar backups de la base de datos y poder restaurar esta copia de seguridad en el futuro a través de phpMyAdmin mediante la opción “importar”.

Framework

Framework nos permite dar una arquitectura para que los softwares trabajen bajo un estándar y asegura que la contracción sea más rápida, es esquema que ayuda

a la programación a estructurar el código y a ahorrar tiempo y esfuerzos a los programadores.

HTML y XHTML

HTML, siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C.

Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo. El propio W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global".

Por convención, los archivos de formato HTML usan la extensión .htm o .html.

XHTML es el sucesor de HTML. XHTML, acrónimo inglés de eXtensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML.

XHTML tiene como objetivo avanzar en el proyecto del W3C de lograr una web semántica, donde la información, y la forma de presentarla estén claramente separadas. En este sentido, XHTML serviría únicamente para transmitir la información que contiene un documento, dejando para hojas de estilo y JavaScript su aspecto y diseño en distintos medios.

CSS

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML. En este último podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

1. Control centralizado de la presentación de un sitio web completo, con lo que se agiliza de forma considerable la actualización del mismo.
2. Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web remoto, con lo que aumenta considerablemente la accesibilidad.
3. Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre.
4. El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

JavaScript, jQuery y AJAX

JavaScript es un lenguaje interpretado utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java. Sin embargo, al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia. Es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM (Modelo de Objetos del Documento). JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias se van descargando junto con el código HTML.

jQuery es una biblioteca o Framework de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications).

Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Modelo, Vista, Controlador, (MVC)

Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

- Modelo
- Vista
- Controlador

Modelo: este componente se encarga de manipular, gestionar y actualizar los datos. Si se utiliza una base de datos aquí es donde se realizan las consultas, búsquedas, filtros y actualizaciones.

Vista: este componente se encarga de mostrarle al usuario final las pantallas, ventanas, páginas y formularios; el resultado de una solicitud. Desde la perspectiva del programador este componente es el que se encarga del frontend; la programación de la interfaz de usuario si se trata de una aplicación de escritorio, o bien, la visualización de las páginas web (CSS, HTML, HTML5 y JavaScript).

Controlador: este componente se encarga de gestionar las instrucciones que se reciben, atenderlas y procesarlas. Por medio de él se comunican el modelo y la vista: solicitando los datos necesarios; manipulándolos para obtener los resultados; y entregándose a la vista para que pueda mostrarlos.

XAMPP

XAMPP es un servidor independiente de plataforma de código libre. Te permite instalar de forma sencilla Apache en tu propio ordenador, XAMPP incluye además servidores de bases de datos como MySQL con sus respectivos gestores phpMyAdmin y phpSQLiteAdmin. Incorpora también el intérprete de PHP.

¿Para qué sirve?

XAMPP es una herramienta de desarrollo que te permite probar tu trabajo como páginas web en tu propio ordenador sin necesidad de tener que acceder a internet

Sistema de login y registro de usuarios

El programa login se encarga de la autenticación del usuario (comprobando que el nombre de usuario y contraseña sean correctos), y establece un entorno inicial para el usuario activando permisos para la línea serie e iniciando el intérprete de comandos. Una cuenta de usuario nos permite acceder a los servicios de un sistema. Por lo general nos autoriza el acceso. Aunque la autenticación no implica autorización automática.

Una vez que el usuario ha firmado, el sistema operativo asocia un identificador. Por ejemplo, un entero para referirse a él, en vez de utilizar el nombre de usuario. A esto se le conoce como identificador de usuario (user id) en los sistemas operativos

Los sistemas de cómputo se dividen en dos grupos basados en el tipo de usuarios que tienen:

- Sistemas monousuarios: Los cuales no manejan el concepto de varias cuentas de usuario.
- Sistemas multiusuario: Los cuales lo tienen.

También incluye recuperación de contraseña a través de correo electrónico y registrar usuarios en una base de datos.

Vistas

Las Vistas (views) son una parte de la arquitectura MVC. Estas son el código responsable de presentar los datos al usuario final. En una aplicación Web, las vistas son usualmente creadas en términos de templates que son archivos PHP que contienen principalmente HTML y PHP. Estas son manejadas por el componente de la aplicación view, el cual provee los métodos comúnmente utilizados para facilitar la composición y renderizado. Por simplicidad, a menudo nos referimos a los templates de vistas o archivos de templates como vistas.

Template

Template Method es un patrón de diseño de comportamiento que te permite definir el esqueleto de un algoritmo en una clase base y permite a las subclases sobrescribir los pasos sin cambiar la estructura general del algoritmo.

Los sistemas de templates nos permiten mejorar sensiblemente el código de las aplicaciones en general y de PHP en particular. Generalmente, cuando estamos aprendiendo PHP, posiblemente siguiendo los distintos tutoriales que encontramos en la web, nos enseñan que el código PHP se embebe dentro del código HTML. Esto está bien para comenzar con unos primeros pasos, pero a la hora de hacer aplicaciones complejas se queda muy corto.

Los sistemas de plantillas también nos permiten que la arquitectura de carpetas de un proyecto sea más ordenada, ya que generalmente colocaremos todas las vistas de la aplicación dentro de un directorio, organizadas por carpetas.

PDO

La extensión de Objetos de Datos de PHP (PDO por sus siglas en inglés) define una interfaz ligera para poder acceder a la base de datos en PHP. Cada controlador de base de datos que implementa la interfaz de PDO puede exponer características específicas de la base de datos, como las funciones habituales de la extensión.

PDO proporciona una capa de abstracción de acceso a datos, loque significa que independientemente de la base de datos que se esté utilizando, se emplean las mismas funciones para realizar consultas y obtener datos. PDO no proporciona

una abstracción de base de datos, no reescribe SQL ni emula características ausentes.

CAPÍTULO III

DESARROLLO Y CASO PRÁCTICO DE LA EMPRESA

CAPÍTULO III: DESARROLLO Y CASO PRÁCTICO DE LA EMPRESA

Este capítulo comprende la descripción del procedimiento, los instrumentos que fueron utilizados, para solucionar las áreas de oportunidad detectadas. Es decir, el

cómo pude dar solución y cuáles fueron las estrategias, propuestas y medidas de acción a tomar tanto por la empresa como yo, para la solución del problema que se detectó.

EASY CODE, ha desarrollado distintas páginas web, aplicaciones móviles, marketing en general y en este momento busca desarrollar una agenda digital de ventas, que como se sabe durante el lapso de pandemia se suspendieron actividades físicas y esto conllevó a quiebras o disgustos por no tener a la mano lo que el comprador necesita, desde ver los productos, como poder agendar una cita para atender dudas, así como los integrantes de la empresa, quienes atiendan este negocio no pueden estar expuestos y de alguna manera necesitan seguir laborando es así como se generó la solución a este problema; mediante una página web se subirán todos los productos de dicho negocio, al mismo tiempo realizar compras de manera digital y agendar citas para atender dudas o aclaraciones (estas dependen del negocio, pueden ser videollamada, telefónicas...).

3.1 DESCRIPCIÓN DEL SISTEMA.

El sistema está siendo codificado por secciones, es decir a un practicante le toco cierta parte y a mí una parte, la cual al final se decidirá si se une o se sigue programando por separado, esto con el objetivo de calificar nuestras habilidades y así ver las opciones de mejoras en cada uno.

Dicho esto, mencionaré las especificaciones que se me han indicado.

- Desarrollar mediante un template de admin, todas las tareas de un administrador (CRUD) en cada uno de los módulos.
- La página consta de estas secciones: el login, resumen de ventas (gráficos).
- El administrador tendrá acceso a estos módulos para realizar todas las tareas de CRUD (crear, leer, actualizar, borrar).
- Base de datos. Para la realización de este proyecto, la empresa creó la base de datos, nombrada “agenda digital”, esta fue diseñada con el lenguaje estructurado de consulta y la interfaz gráfica de PhpMyAdmin, en el cual se encuentran las siguientes tablas: “usuarios”, aquí se determinan que permisos tienen ya que son: administrador y asistente; En la tabla “productos”, aquí se guardan detalles de los productos, la tabla “ventas”, en esta se describe a quien se ha determinado la venta y otras características como el IVA y total, la siguiente tabla es “clientes”, la cual almacena información de los clientes, una tabla más es la de “ventas_productos” almacena cuantos productos se vendieron, costo y otras características y finalmente la tabla “citas” la cual guarda datos de las citas que agende el administrador o asistente.

3.2 DESARROLLO / PROGRAMACIÓN

3.2.1 CONSTRUCCIÓN DEL SISTEMA

Como se ha mencionado en el capítulo 2, una de las herramientas a ocupar para el desarrollo de este proyecto es el editor de Visual Studio Code, en este caso está presentándose la estructura del proyecto en el editor mencionado.

La estructura de este proyecto está enfocado al estilo de arquitectura MVC, se aprecian del lado izquierdo las carpetas con los archivos que conforman el modelo, vista y controlador.

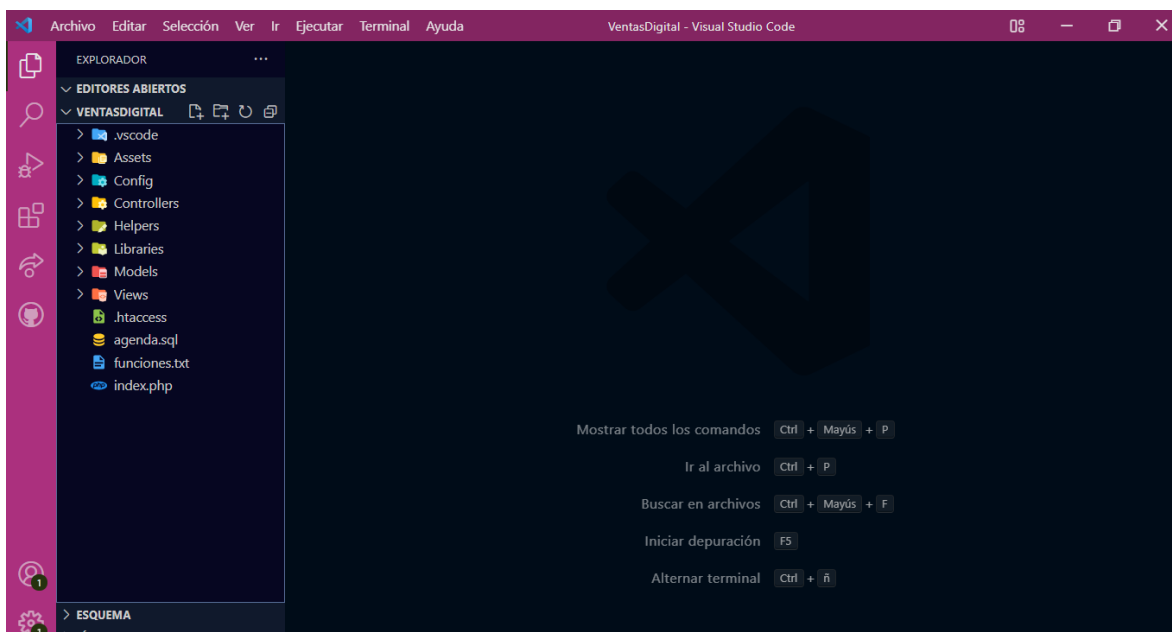


Ilustración 1 Estructura del proyecto

3.3.2 SISTEMA DE LOGIN

Para el desarrollo de la vista de login no se requirió algún estilo en específico ya que, a pesar de ser desarrollada para la empresa, aún no está destinada para un proyecto específico de ser así se hubieran asignado los colores y demás requerimientos; Sin embargo, se solicitó un login sencillo, el cual se pueda adaptar cualquier estilo de Bootstrap.

Agenda Digital

Correo

Usuario

Contraseña

Contraseña

☐ Recordarme [olvidaste tu contraseña?](#)

INICIAR SESION

Ilustración 2 Login

De fondo se asignó que fuera neutro, con la leyenda o título en este caso “Agenda digital” y los campos requeridos para iniciar sesión como administrador; son solo dos etiquetas las cuales pedirá el correo y contraseña, que previamente se les asignará a los usuarios.

Código

Vista

La interfaz es fácil de entender y manejar, la estructura de esta página está diseñada en HTML, la intención de diseñarla mediante HTML fue de poder ingresar el diseño acorde al tema, lo que viene siendo de la empresa, de esta forma hacer una interfaz amigable con el usuario empleando la usabilidad, donde lo más importante es que el usuario puede acceder sin ningún problema y de forma ágil.

Esto por el lado de la vista y el código se ve de esta manera, donde se pueden leer líneas de código construyendo el cuerpo de HTML y mando a llamar los respectivos estilos.

```

38 </head>
39 <body class="animstition">
40 <div class="page-wrapper">
41     <div class="page-content--bge5">
42         <div class="container">
43             <div class="login-wrap">
44                 <div class="login-content">
45                     <div class="login-logo">
46                         <a href="#"> Agenda Digital
47                         <img src="" alt="">
48                     </a>
49                 </div>
50                 <div class="login-form">
51                     <form action="" name="formLogin" id="formLogin" method="POST">
52                         <div class="form-group">
53                             <label>Correo</label>
54                             <input class="au-input au-input--full" type="text" id="txtEmail" name="txtEmail" p
55                         </div>
56                         <div class="form-group">
57                             <label>Contraseña</label>
58                             <input class="au-input au-input--full" type="password" id="txtpassword" name="txtp
59                         </div>
60                         <div class="login-checkbox">
61                             <label>
62                                 <input type="checkbox" name="remember">Recordarme
63                             </label>
64                         </div>

```

Ilustración 3 Código de la vista login

Controlador

El controlador contiene una lógica que actualiza el modelo y / o vista en respuesta a las entradas de los usuarios de la aplicación.

Aquí, se especifican las acciones, por ejemplo, cuando va a iniciar sesión valida que los datos solicitados estén correctos para acceder, manda esta información al modelo para que lo actualice y este se lo manda a la vista.

```

<?php
class Login extends Controllers{
    public function __construct(){
        parent::__construct();
        session_start();
        if(isset($_SESSION['login']))
        {
            header('Location: '.$base_url().'/dashboard');
        }
        parent::__construct();
    }
    public function login(){
        // $data['page_id'] = 1;
        $data['page_tag'] = "Login - Agenda digital";
        $data['page_title'] = "Login";
        $data['page_name'] = "login";

```

```

        $data['page_funtions_js'] = "funtions_login.js";

        $this->views->getView($this,"login", $data);
    }
    public function loginUser(){
        //dep($_POST);
        if($_POST){
            if(empty($_POST['txtEmail']) || empty($_POST['txtpassword'])){
                $arrResponse = array('status' => false, 'msg' => 'Error de
datos' );
            }else{
                $strUsuario = strtolower(strClean($_POST['txtEmail']));
                $strPassword = $_POST['txtpassword'];
                //hash("SHA256",$_POST['password']);
                $requestUser = $this->model->loginUser($strUsuario,
$strPassword);
                if(empty($requestUser)){
                    $arrResponse = array('status' => false, 'msg' => 'El
usuario o la contraseña es incorrecto.' );
                }else{
                    $arrData = $requestUser;
                    if($arrData['estatus'] == 1){
                        $_SESSION['id_usuario'] = $arrData['id_usuario'];
                        $_SESSION['login'] = true;

                        $arrData = $this->model-
>sessionLogin($_SESSION['id_usuario']);
                        $_SESSION['userData'] = $arrData;
                        //sessionUser($_SESSION['idUser']);

                        $arrResponse = array('status' => true, 'msg' =>
'ok');
                    }else{
                        $arrResponse = array('status' => false, 'msg' =>
'Usuario inactivo.');
```

Modelo

El modelo define qué datos debe contener la aplicación. Si el estado de estos datos cambia, el modelo generalmente notificará a la vista (para que la pantalla pueda cambiar según sea necesario) y, a veces, el controlador (si se necesita una lógica diferente para controlar la vista actualizada).

En pocas palabras, aquí se especifica los datos que deben contener los usuarios para tener acceso.

```
<?php

class LoginModel extends Mysql
{
    private $intIdUsuario;
    private $strUsuario;
    private $strPassword;
    private $strToken;

    public function __construct()
    {
        parent::__construct();
    }

    public function loginUser(string $usuario, string $password)
    {
        $this->strUsuario = $usuario;
        $this->strPassword = $password;
        //echo
        $sql = "SELECT id_usuario,estatus FROM usuarios WHERE
                correo = '$this->strUsuario' and
                password = '$this->strPassword' and
                estatus != 0 ";
        //exit
        $request = $this->select($sql);
        return $request;
    }

    public function sessionLogin(int $iduser){
        $this->intIdUsuario = $iduser;
        //BUSCAR ROLE
        $sql = "SELECT *
                FROM usuarios

                WHERE id_usuario = $this->intIdUsuario";
        $request = $this->select($sql);
        // $_SESSION['userData'] = $request;
        return $request;
    }
}
```


}

3.2.2 INTERFAZ DE RESUMEN DE VENTAS (GRÁFICAS)

Al momento de ingresar con los datos correspondientes se puede apreciar que es muy sencilla, debido a que solo aparecen las ventas y gráficas de las mismas.

Son pequeñas cartas donde muestran los miembros que hay, cuantos registros sucedieron a la semana y cuál ha sido el total de ingresos. Cada uno llevará a sus correspondientes módulos donde se especificará más a cerca del bloque, en este caso solo se mostrará esta sección para no alargar el documento.

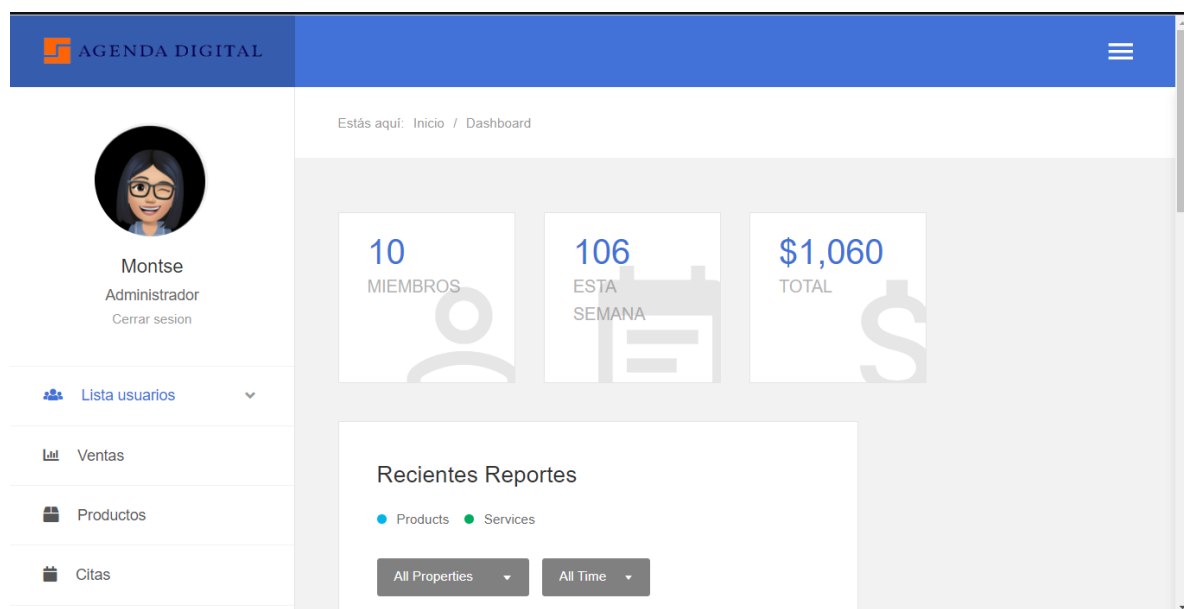


Ilustración 4 Gráficas y resumen

3.2.3 INTERFAZ LISTA DE USUARIOS

El módulo de la lista de usuarios está compuesto por una lista pequeña de clientes y usuarios.

La función de este módulo es que como administrador en cada uno de ellos podrá dar de alta a algún usuario y asignar que puede ser administrador o asistente.

La interfaz es clara, se aprecia un botón con la leyenda “Agregar Nuevo Usuario”, por ende, aquí cumple la función de crear.

La función y proceso para crear un usuario es el siguiente:

Al dar click en botón de agregar, sale un modal el cual pedirá los datos de este usuario para poder almacenarlos en la base de datos.

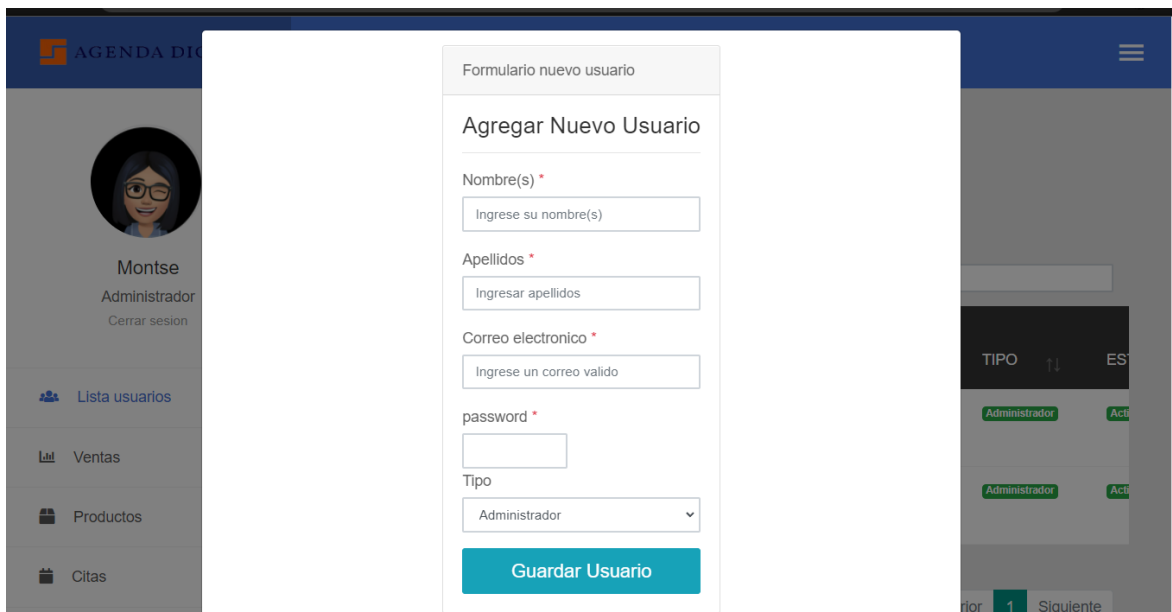
The image shows a web application interface with a dark blue header and sidebar. The sidebar on the left contains a user profile for 'Montse Administrador' and a menu with options: 'Lista usuarios', 'Ventas', 'Productos', and 'Citas'. The main content area displays a modal titled 'Formulario nuevo usuario' with the subtitle 'Agregar Nuevo Usuario'. The form contains several input fields: 'Nombre(s)' with placeholder 'Ingrese su nombre(s)', 'Apellidos' with placeholder 'Ingresar apellidos', 'Correo electronico' with placeholder 'Ingrese un correo valido', and 'password'. Below these is a 'Tipo' dropdown menu currently set to 'Administrador'. A teal 'Guardar Usuario' button is at the bottom of the form. In the background, a table with columns 'TIPO' and 'ES' is partially visible, showing rows for 'Administrador'.

Ilustración 5 Agregar usuario

Ahora, el sistema cuenta con una validación la cual consiste en que no pueden enviar datos vacíos, ya que si las etiquetas están vacías mandara un mensaje de error, esto con el fin de no guardar datos incompletos y crear un tráfico en la base de datos. El mensaje se muestra de esta manera:

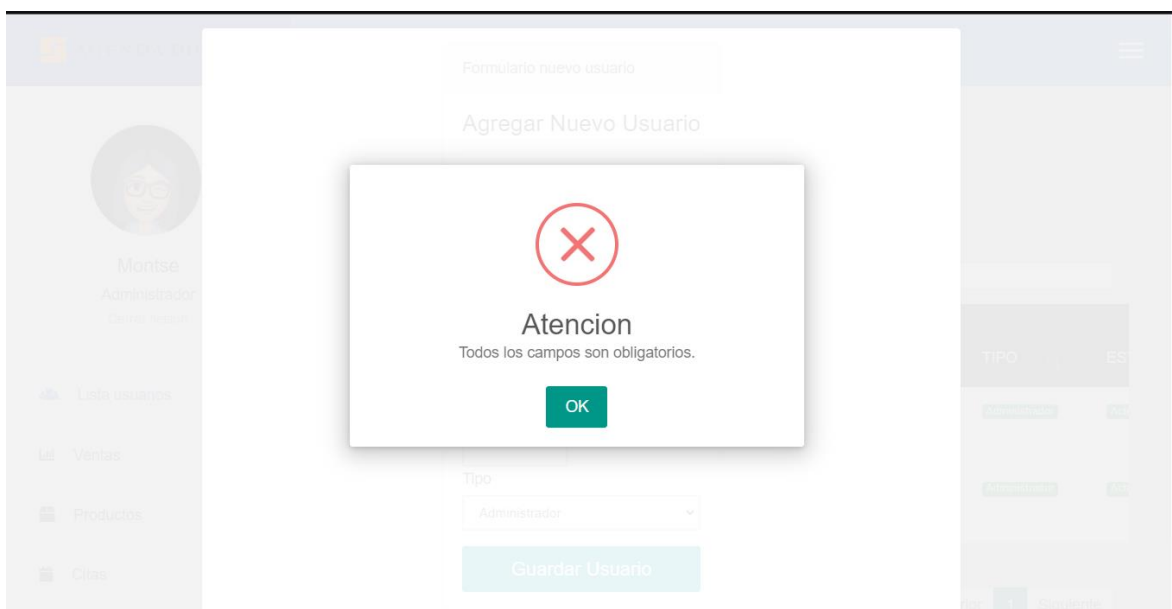


Ilustración 6 Mensaje de error

Seguido de llenar todos los datos, incluso haber seleccionado que tipo de usuario es podemos continuar con el proceso de agregar, en la imagen se muestra que los

datos están casi llenos, como se observa la caja de texto de password está vacía, pero en la programación se configuró para que si en caso de que al administrador no agregue contraseña se le asigne una por default.

Ilustración 7 Ingreso de datos

Al tener todos los campos completos y presionar el botón de guardar nuevo usuario, sale un mensaje de éxito como se muestra en la imagen.

Ilustración 8 Mensaje de éxito

Finalizando el proceso de agregar, en la tabla se puede visualizar que ya se ha agregado este nuevo usuario.

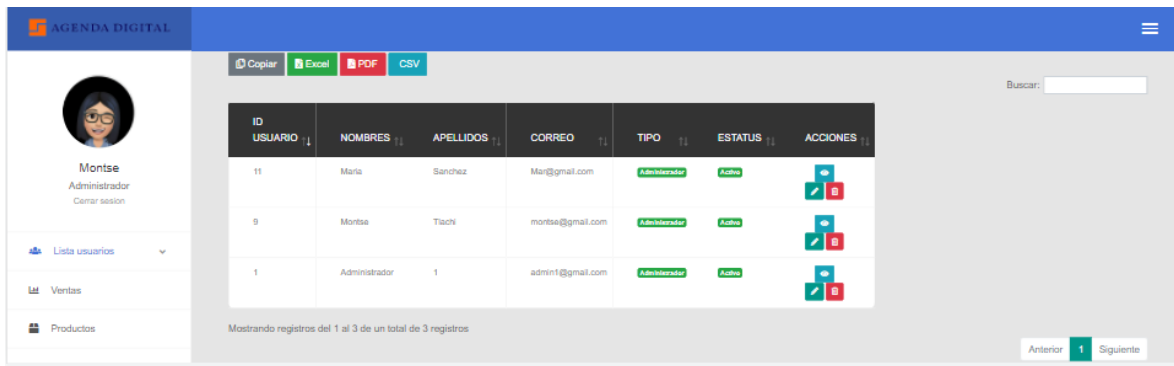


Ilustración 9 Registros

Botones de archivos de descarga.

Pasando al proceso de los botones que se puede observar una serie de botones las cuales al momento de dar click en automático se descargará la lista de usuarios en formato Excel, PDF y CSV, la opción de copiar solo copia el ultimo registro.



Ilustración 10 Botones de archivos de descarga

Al momento de dar click en cualquiera de ellos las descargas se hace en automático



Ilustración 11 Descargas

Y esta información se puede visualizar de acuerdo al formato en el que hayamos solicitado descargar.

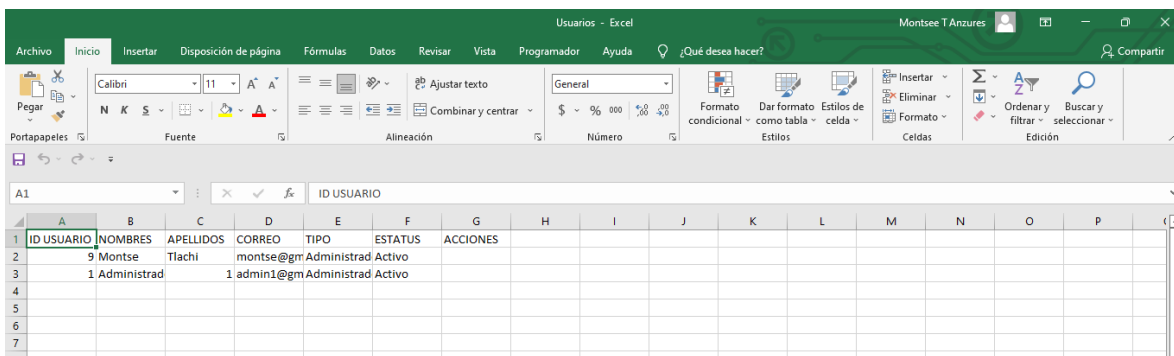


Ilustración 12 Descarga CSV

Usuarios

ID USUARIO	NOMBRES	APELLIDOS	CORREO	TIPO	ESTATUS	ACCIONES
9	Montse	Tlachi	montse@gmail.com	Administrador	Activo	
1	Administrador	1	admin1@gmail.com	Administrador	Activo	

Ilustración 13 Descarga PDF

El código que se puede destacar de esta sección es el de la funcionalidad de descargar los archivos ya que no es complejo, el código se muestra de esta manera y solo especifica que acción hacer. Aquí podemos modificar incluso que columnas no queremos descargar.

```
document.addEventListener('DOMContentLoaded', function() {
    tableUsuarios = $('#tablaUsuarios').dataTable({
        "aProcessing": true,
        "aServerSide": true,
        "language": {
            "url": "//cdn.datatables.net/plug-ins/1.10.20/i18n/Spanish.json"
        },
        "ajax": {
            "url": " " + base_url + "/Usuarios/getUsuarios",
            "dataSrc": ""
        },
        "columns": [
            { "data": "id_usuario" },
            { "data": "nombre" },
            { "data": "apellidos" },
            { "data": "correo" },
            { "data": "tipo" },
            { "data": "estatus" },
            { "data": "options" }
        ],
        'dom': 'lBfrtip',
        'buttons': [{
            "extend": "copyHtml5",
            "text": "<i class='far fa-copy'></i> Copiar",
            "titleAttr": "Copiar",
            "className": "btn btn-secondary"
        }, {
            "extend": "excelHtml5",
            "text": "<i class='fas fa-file-excel'></i> Excel",
```

```

        "titleAttr": "Exportar a Excel",
        "className": "btn btn-success"
    }, {
        "extend": "pdfHtml5",
        "text": "<i class='fas fa-file-pdf'></i> PDF",
        "titleAttr": "Exportar a PDF",
        "className": "btn btn-danger"
    }, {
        "extend": "csvHtml5",
        "text": "<i class='fas fa-file-csv'></i> CSV",
        "titleAttr": "Exportar a CSV",
        "className": "btn btn-info"
    }
  ],
  "resonsieve": "true",
  "bDestroy": true,
  "iDisplayLength": 5,
  "order": [
    [0, "desc"]
  ]
});

```

Tabla de registros

Debajo de estos botones ya se encuentra la tabla de registro de los usuarios, donde están algunos de los datos que se necesitan visualizar como el ID, NOMBRES, etc.

Se puede visualizar que en el tipo tenemos resaltado el tipo de usuario que es, si está activo será en verde en caso de que no solo lo marcará en rojo y finalmente los botones de las acciones que el administrador tiene sobre ellas donde mas adelante se especificara cada botón.




ID USUARIO ↑↓	NOMBRES ↑↓	APELLIDOS ↑↓	CORREO ↑↓	TIPO ↑↓	ESTATUS ↑↓	ACCIONES ↑↓
11	Maria	Sanchez	Mar@gmail.com	Administrador	Activo	  
9	Monise	Tlachi	montso@gmail.com	Administrador	Activo	  
1	Administrador	1	admin1@gmail.com	Administrador	Activo	  

Ilustración 14 Tabla de registro y datos

Acciones (RUD)

En este caso descartamos la acción de crear, ya que fue la primera en describir el proceso.

Ver

Cumple con la función de visualizar los datos del usuario de una manera más cómoda y formal puesto que está en un modal.

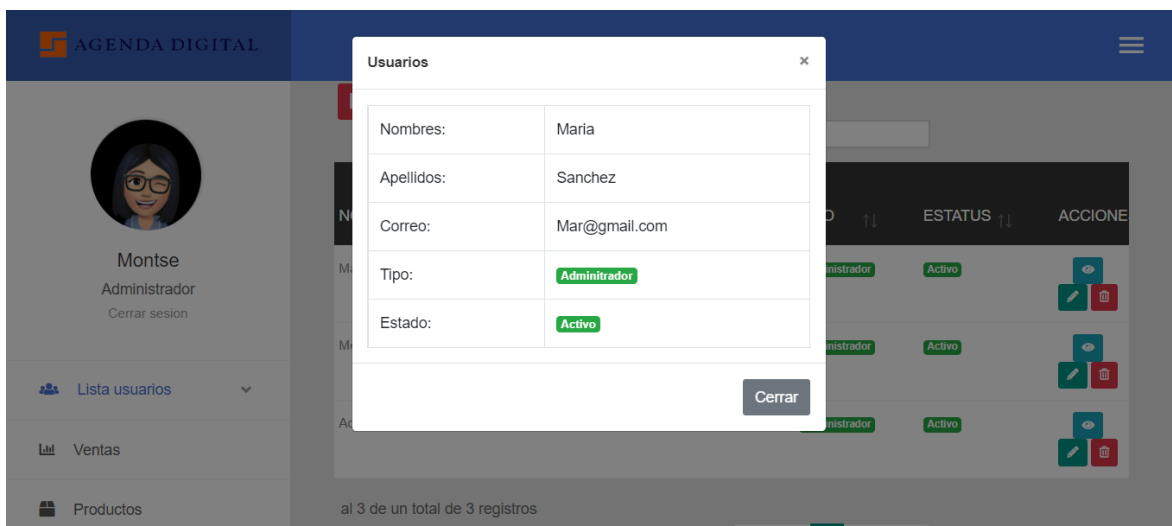


Ilustración 15 Acción Ver

Actualizar

El botón de actualizar está definido por un icono de un lápiz, al darle click este mostrará un modal con los datos del usuario que se desea modificar, por ejemplo, podemos modificar su nombre, apellidos o si queremos cambiar el tipo de usuario, no hay alguna validación ya que todos los datos pueden ser modificados por el administrador, una vez hecho los cambios procedemos a presionar el botón de actualizar usuario. (En este ejemplo se modificó el tipo de usuario a asistente)

Formulario nuevo usuario

Actualizar Usuario

Nombre(s) *

Maria Maria

Apellidos *

Sanchez

Correo electronico *

Mar@gmail.com

password *

....

Tipo

Asistente

Actualizar Usuario

Ilustración 16 Actualización de datos

Si los campos están llenos y se realizó la modificación correctamente se mostrará un mensaje de éxito, diciendo que los datos han sido actualizados correctamente

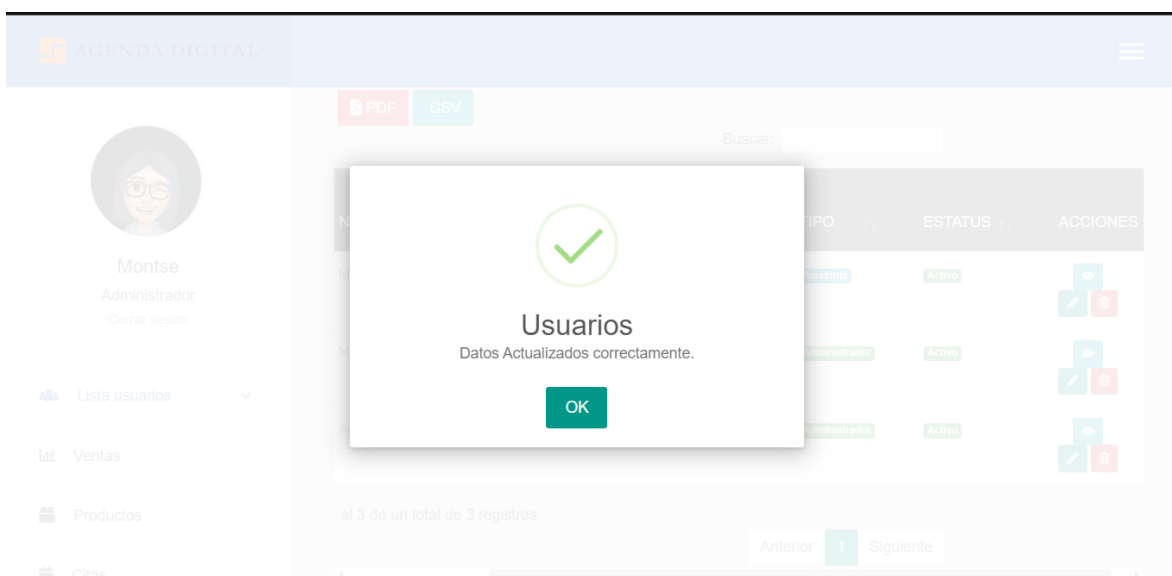


Ilustración 17 Mensaje "Actualización correcta"

Comprobamos que los datos se hayan modificado correctamente con el botón de ver. Y en efecto, en el apartado de tipo, ha cambiado a asistente incluso se nota la diferencia por el color que se le asigno.

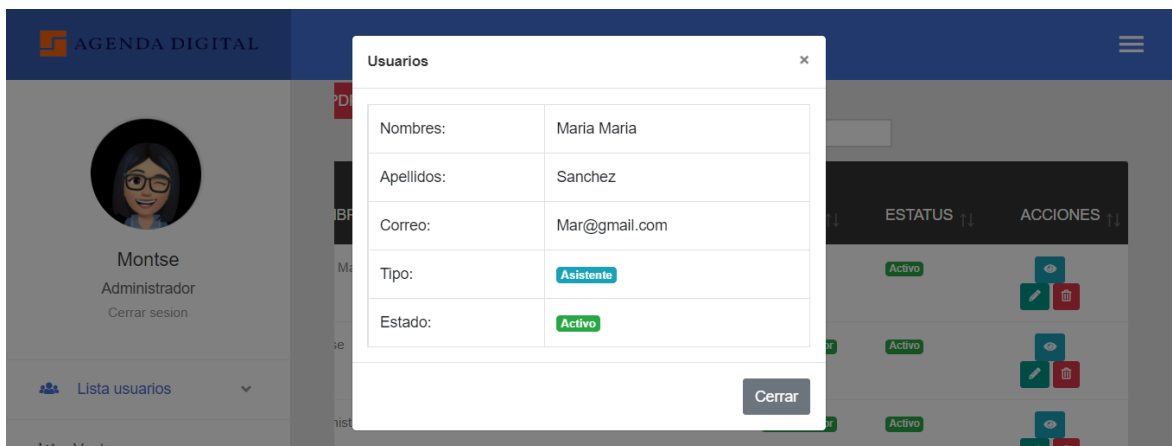


Ilustración 18 Rectificación de cambios

Eliminar

Finalmente tenemos el botón de eliminar, el cual no hay mucho que describir más que al darle click saldrá un mensaje con el icono de emergencia para asegurar que el usuario quiere eliminar algún registro.

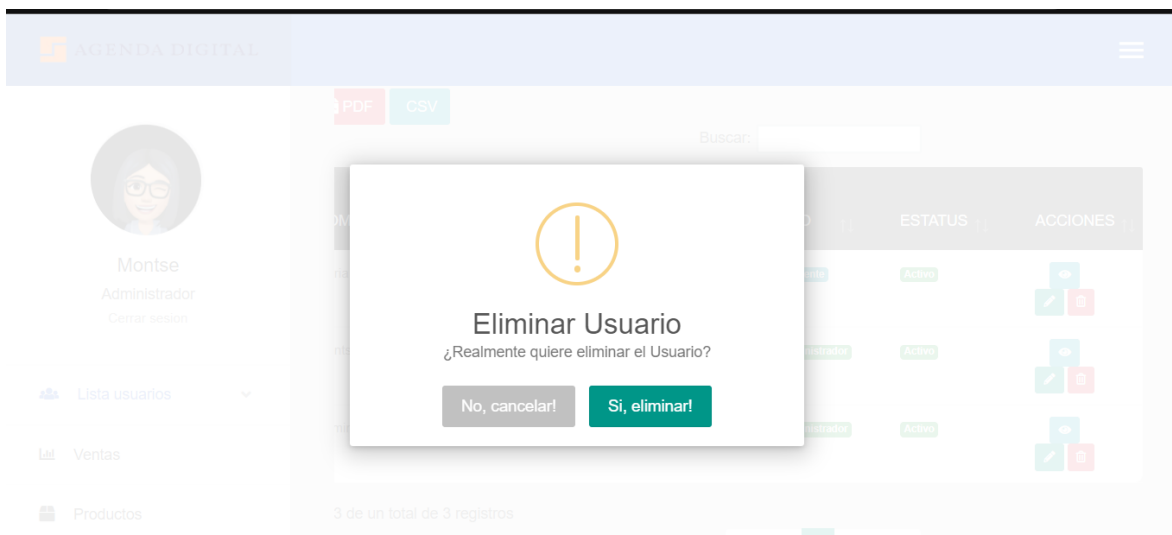


Ilustración 19 Eliminar registro

Una vez aceptado que se deseó eliminar este registro, nos indica que ya ha sido eliminado con el mensaje que se muestra en pantalla.

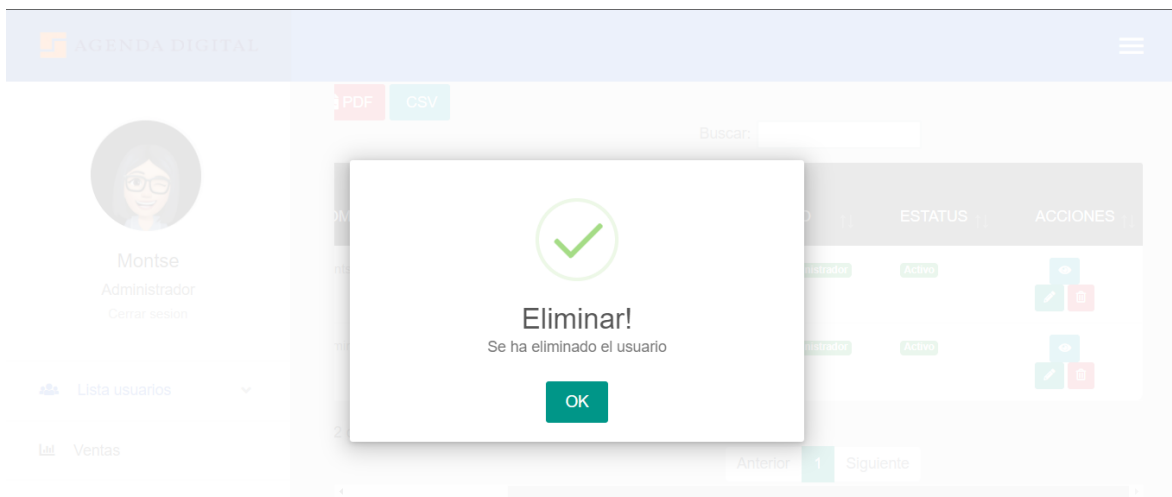


Ilustración 20 Confirmado de registro eliminado

Y evidentemente el registro en la tabla ya no aparece más.

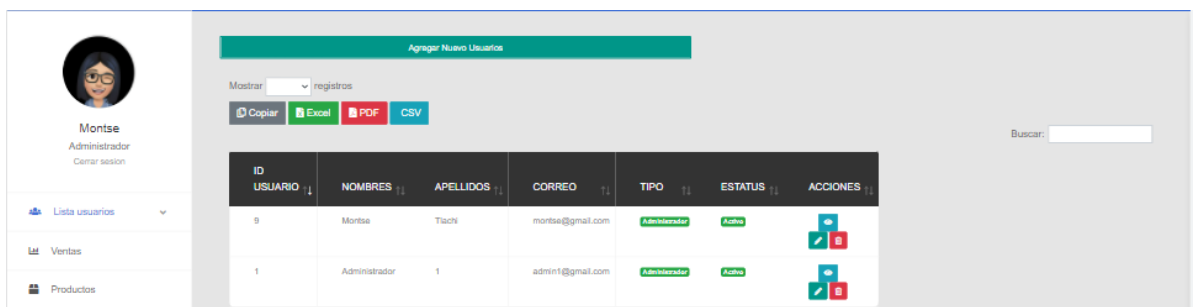


Ilustración 21 Evidencia de eliminado

3.2.4 INTERFAZ DE MÓDULO DE CLIENTES

El módulo de clientes no es tan distinto del módulo de usuarios, básicamente la interfaz y todo el entono será el mismo, ya que cada módulo para el administrador se realizarán las mismas acciones, pero en cada sección, lo único que cambiará será en la información que se requiera para dicho apartado.

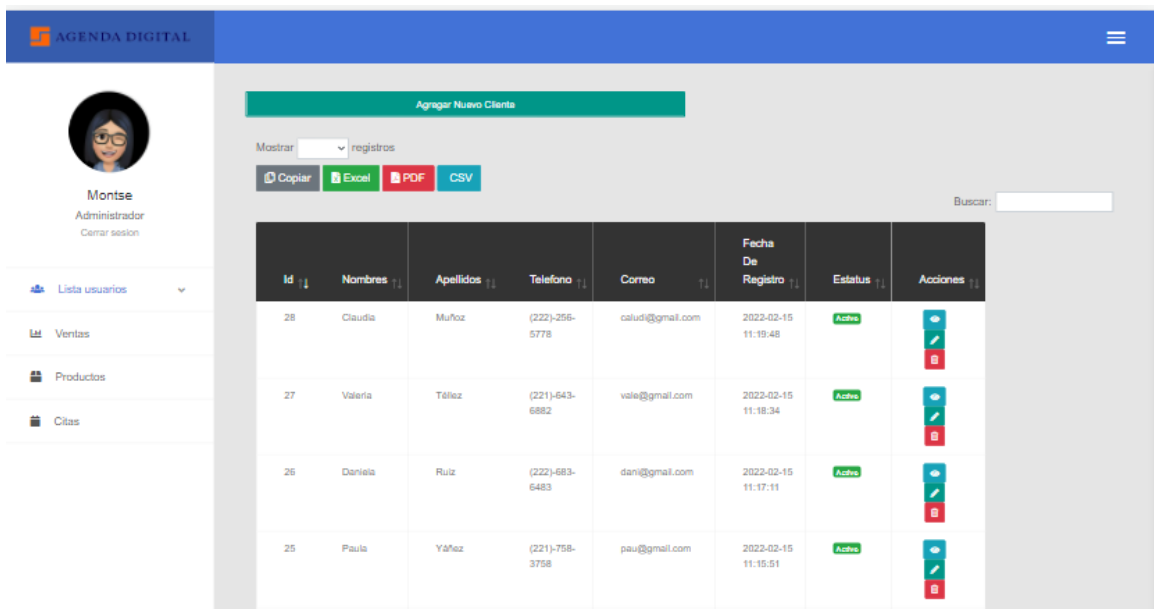


Ilustración 22 Interfaz de clientes

Lo diferente como se ha comentado, solo serán los datos requeridos, Podemos observar que no son los mismos datos que piden a los usuarios pues la información de los clientes es necesaria para otros módulos.

Por ejemplo, en la siguiente imagen se muestra el modal de agregar a un cliente, aquí algo a diferenciar es que son más los datos a guardar, así como el almacenamiento de una imagen.

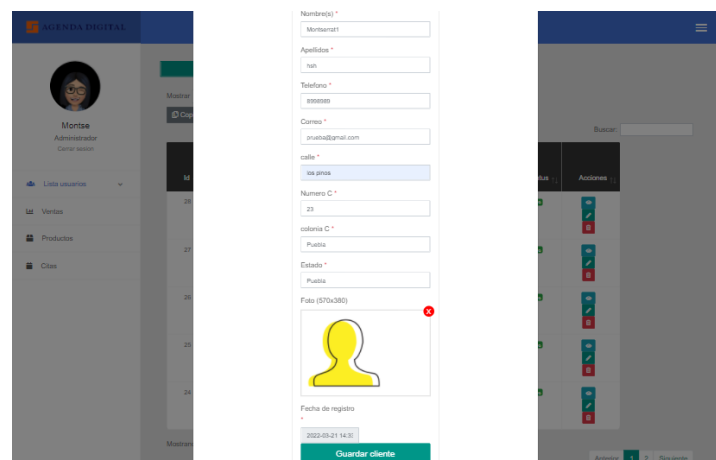


Ilustración 23 Agregar cliente (modal)

También al seleccionar el botón de ver cliente se observan con mayor detenimiento los datos del cliente, ya que en la tabla solo se muestran los necesarios pero el modal contiene toda información.


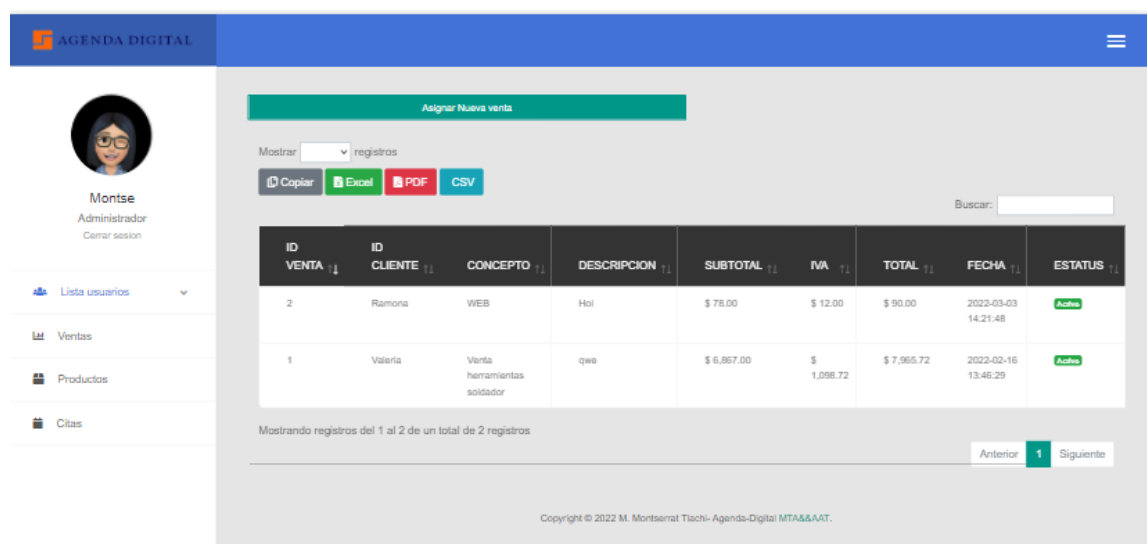
Nombres:	Montserrat1
Apellidos:	Hsh
Celular:	8998989
Correo:	Prueba@gmail.com
Calle:	Los Pinos
No de casa:	23
Colonia:	Puebla
Estado:	Puebla
Imagen:	
Fecha Registro:	2022-03-21 14:33:21
Estatus:	Activo

Ilustración 24 Modal ver cliente

3.2.5 INTERFAZ DE MÓDULO DE VENTAS

Continuando con el siguiente módulo, es de ventas el cual está destinado a los clientes, de cuantas ventas ha obtenido por eso en la tabla se muestra el ID del cliente que es con el cual el administrador podrá identificarlos (se ha puesto el nombre, pero, en la base de datos esta por numeración para un mejor control)



ID VENTA	ID CLIENTE	CONCEPTO	DESCRIPCION	SUBTOTAL	IVA	TOTAL	FECHA	ESTATUS
2	Ramona	WEB	Hoi	\$ 78.00	\$ 12.00	\$ 90.00	2022-03-03 14:21:48	Activo
1	Valeria	Venta herramientas soldador	qwe	\$ 6,867.00	\$ 1,098.72	\$ 7,965.72	2022-02-16 13:46:29	Activo

Ilustración 25 Interfaz de ventas

Un elemento a destacar de este módulo es que esta programado para calcular el IVA, y TOTAL. Como se observa hay tres etiquetas las cuales estan en untono gris ya que estas no pueden ser modificadas puesto que se ha programado que se calcule en automatico en cuanto el usuario ingrese la cantidad de subtotal.

Ilustración 26 Modal "agregar venta"

Algo importante que mencionar es que en cada módulo el mensaje llevará la leyenda del módulo, por ejemplo cuando se guarde o inserte una venta se nota la leyenda de "VENTAS" esto para asegurar el proceso y verificar estar en el módulo correcto .

Ilustración 27 Mensaje éxito (ventas)

En cuanto al código, se especificó que sea por procedimiento AJAX, este permite que un usuario de la aplicación web interactúe con una página web sin la interrupción que implica volver a cargar la página web. La interacción del sitio web ocurre rápidamente sólo con partes de la página de recarga y renovación.

```
var request = (window.XMLHttpRequest) ? new XMLHttpRequest() : new
ActiveXObject('Microsoft.XMLHTTP');
var ajaxUrl = base_url + '/Ventas/setVentas';
var formData = new FormData(formAddventas);
request.open("POST", ajaxUrl, true);
request.send(formData);
request.onreadystatechange = function() {
    if (request.readyState == 4 && request.status == 200) {
        var objData = JSON.parse(request.responseText);
        if (objData.status) {
            $('#modalformaddVentas').modal("hide");
            formAddventas.reset();
            swal('Ventas', objData.msg, 'success');
            tableVentas.api().ajax.reload()

        } else {
            swal("Error", objData.msg, "error");
        }
    }
}
```

Mensajes desde el controlador

```
{
    if($option == 1){
        $arrResponse=array('status' => true, 'msg'=>'Datos
Guardado');
    }else{
        $arrResponse = array('status' => true, 'msg' =>
'Datos Actualizados correctamente.');
```

```

        $arrResponse=array('status' => false, 'msg'=>'Datos
Repetidos');
    }else{
        $arrResponse=array("status" => false, "msg"=>'No es
posible guardar datos');
    }
}
echo json_encode($arrResponse,JSON_UNESCAPED_UNICODE);
}
die();
}

```

Finalmente, en nuestro modal ver datos de la venta se muestra toda la información coleccionada.

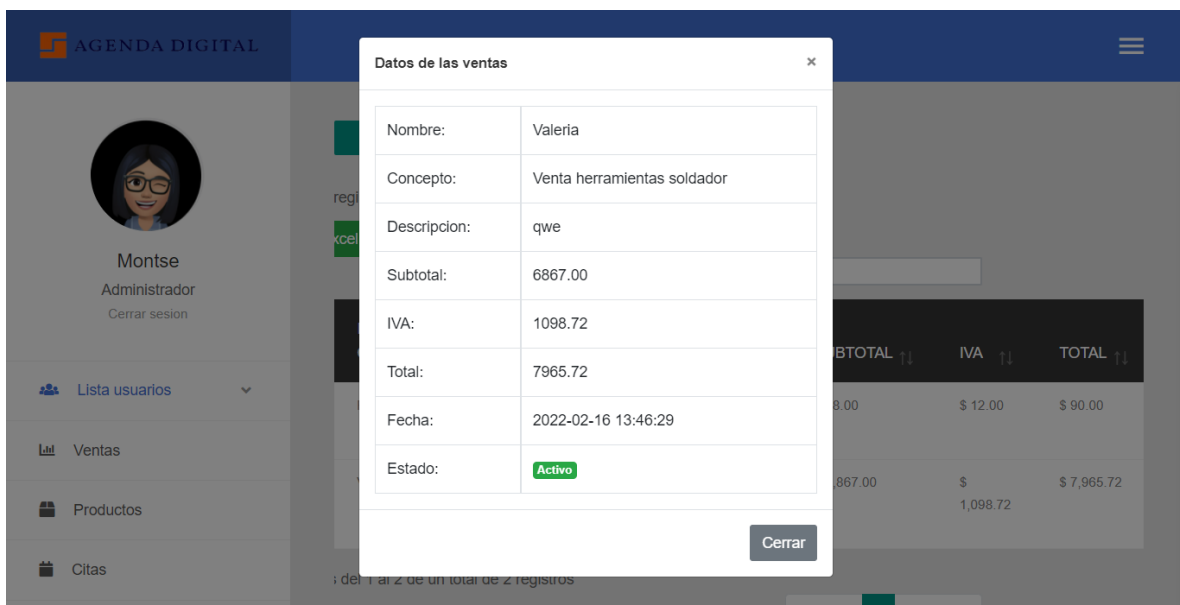


Ilustración 28 Modal "ver venta"

3.2.6 INTERFAZ DE MÓDULO PRODUCTOS

El siguiente modulo a describir es de productos, el cual es la misma dinámica que los módulos anteriores, se muestra la tabla de registros y cada uno de ellos tiene ciertas acciones que el administrador puede acceder (CRUD).

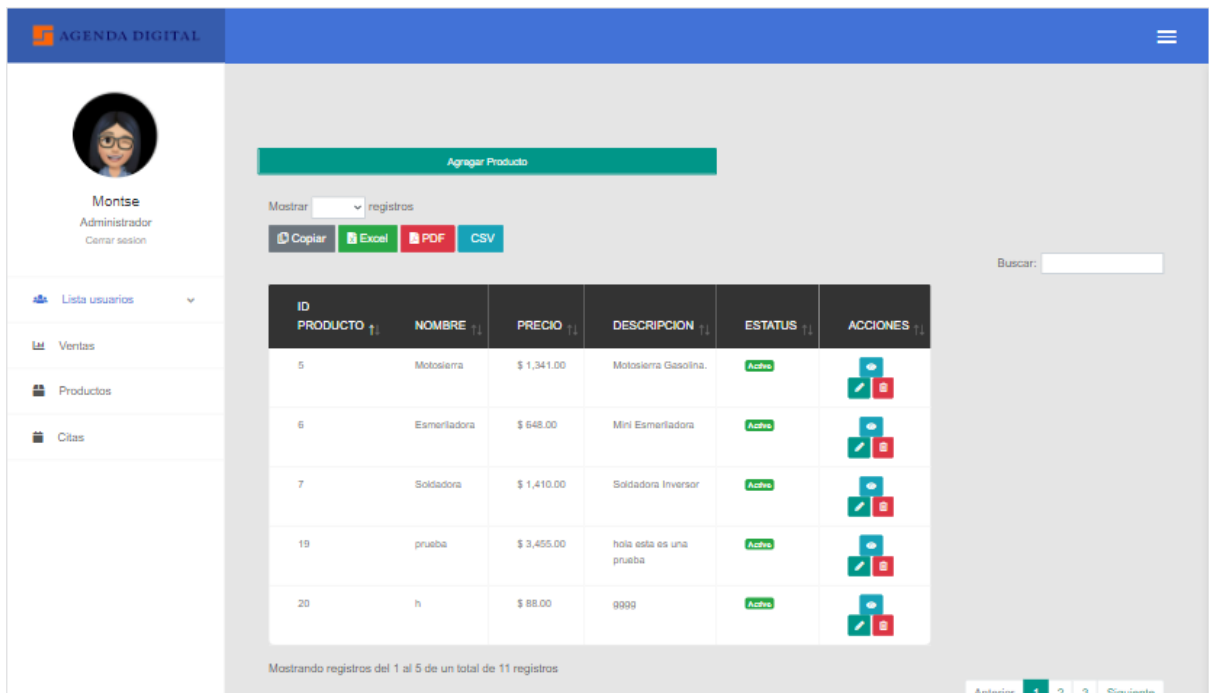


Ilustración 29 Interfaz productos

Esta sección, por sus características fue de las más sencillas en realizar ya que la información a almacenar es poca a comparación de los anteriores, esta se compone solo de tres cajas de texto las cuales requieren el nombre del producto, precio y una pequeña descripción.

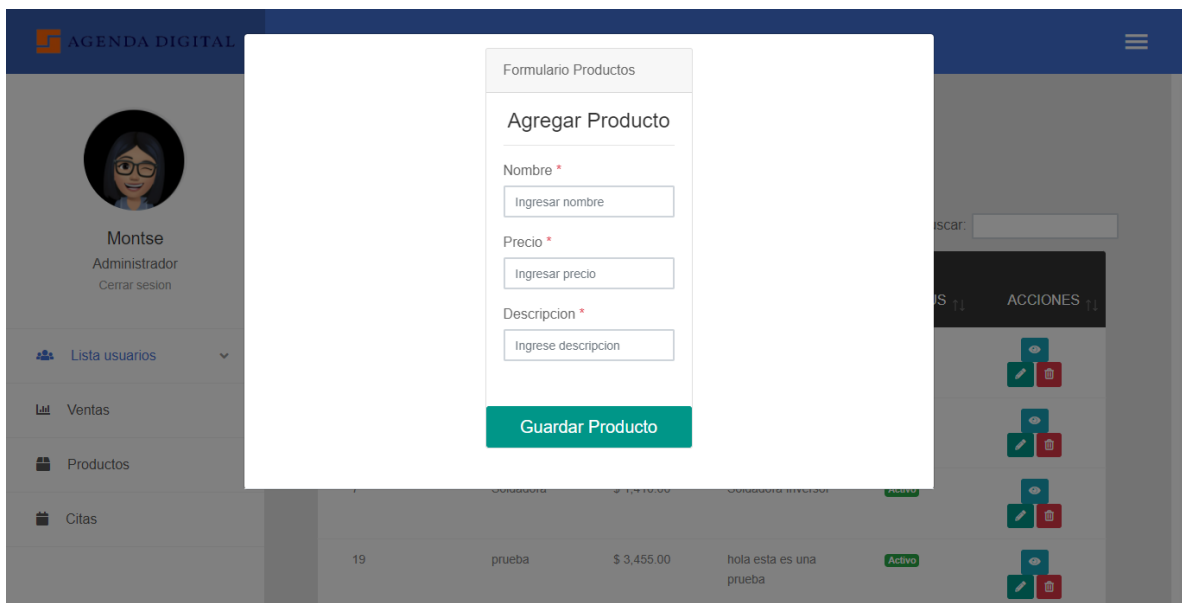
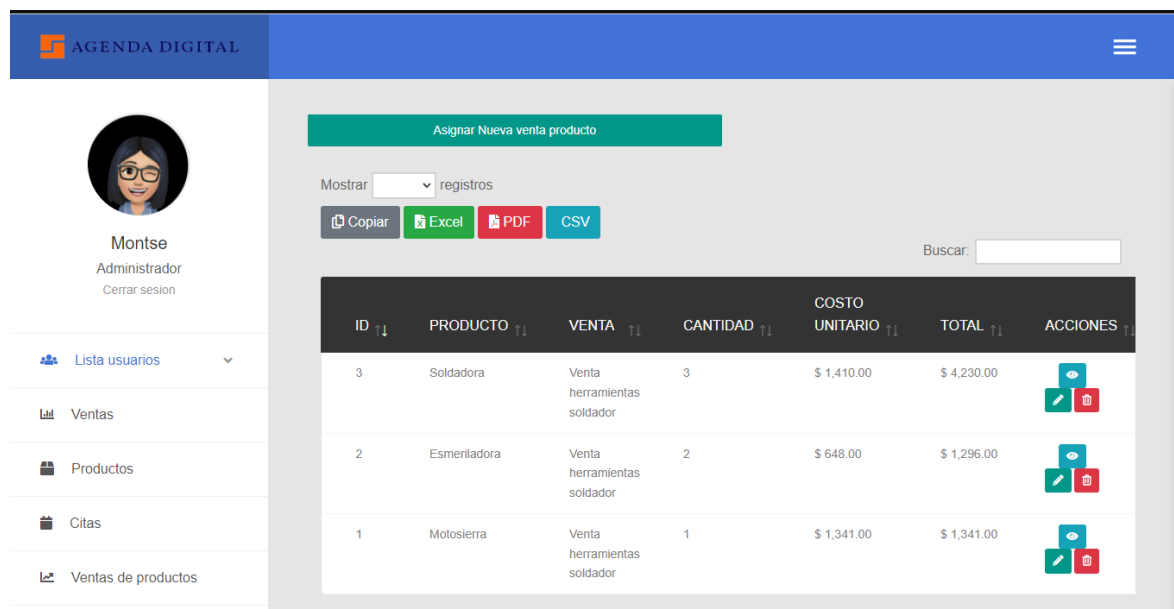


Ilustración 30 Modal "agregar producto"

Las acciones (ver, editar, eliminar) continúan con lo mismo, siguen siendo los mismos mensajes tanto de éxito como de advertencia para el eliminar.

3.2.7 INTERFAZ DE MÓDULO VENTAS-PRODUCTOS

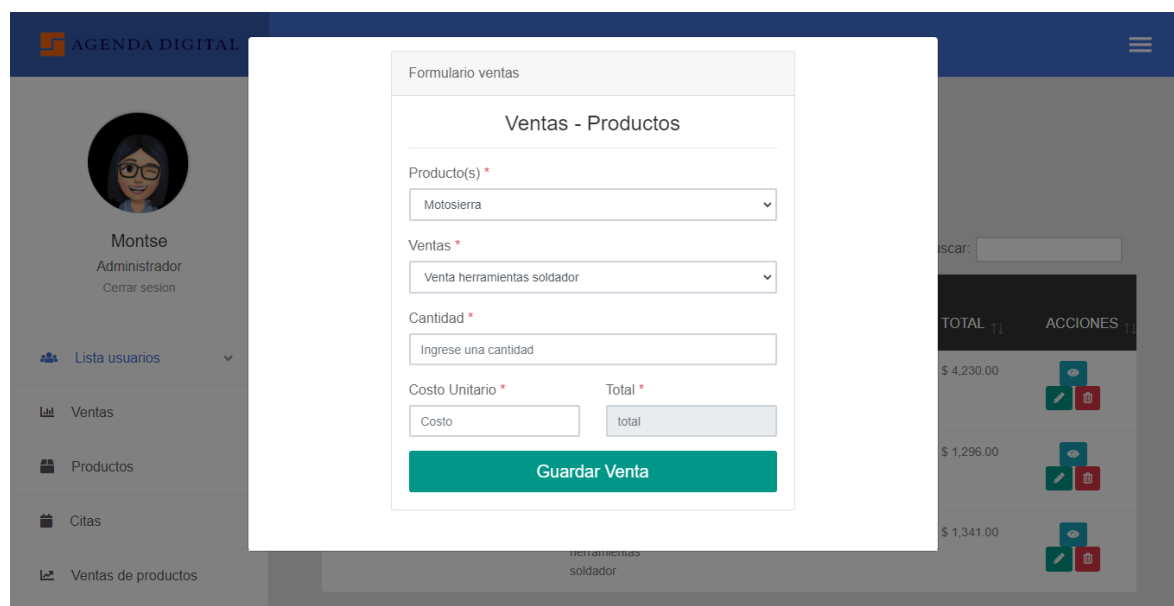
El apartado de ventas productos, se muestra en general las ventas de los productos que se han realizado, en la imagen podemos ver el registro de estas, con algunos ejemplos.



ID	PRODUCTO	VENTA	CANTIDAD	COSTO UNITARIO	TOTAL	ACCIONES
3	Soldadora	Venta herramientas soldador	3	\$ 1,410.00	\$ 4,230.00	
2	Esmeriladora	Venta herramientas soldador	2	\$ 648.00	\$ 1,296.00	
1	Motosierra	Venta herramientas soldador	1	\$ 1,341.00	\$ 1,341.00	

Ilustración 31 Interfaz Ventas-Productos

El modal de manera general, tiene este diseño. Se muestra los datos que solicita para poder guardar la información, cuenta con las mismas características que los demás modales, esta validado para que todos los recuadros estén completos al igual que el de ventas, calcula el total para agilizar las cuentas.



Formulario ventas

Ventas - Productos

Producto(s) *

Motosierra

Ventas *

Venta herramientas soldador

Cantidad *

Ingrese una cantidad

Costo Unitario *

Costo

Total *

total

Guardar Venta

Ilustración 32 Modal Ventas-Productos

3.2.8 INTERFAZ DE MÓDULO CITAS

En el caso del modulo de citas, la interfaz tiene un cambio total ya que esta muestra un calendario donde se mostrarán las citas que se tienen programadas por semana o por mes.

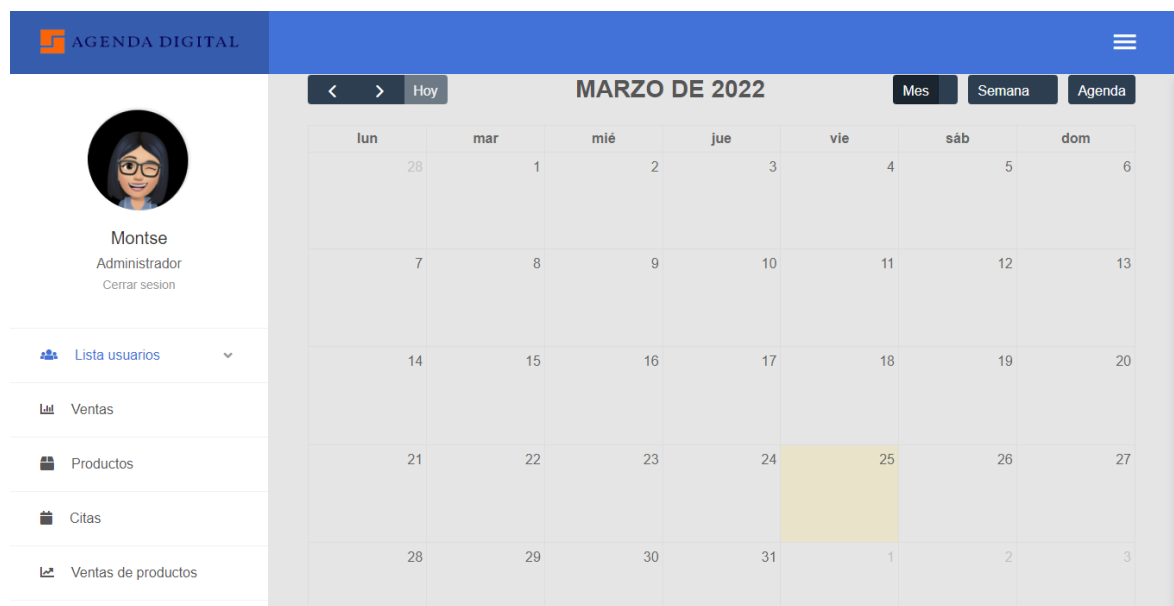


Ilustración 33 Interfaz Modulo Citas

Mientras que la interfaz del modal, para agregar una cita se vera de esta manera, esta se agrega dandole click al dia que se quiere agendar la cita, solo se debera llenar la informacion solicitada para poder guardar la informacion de manera exitosa.

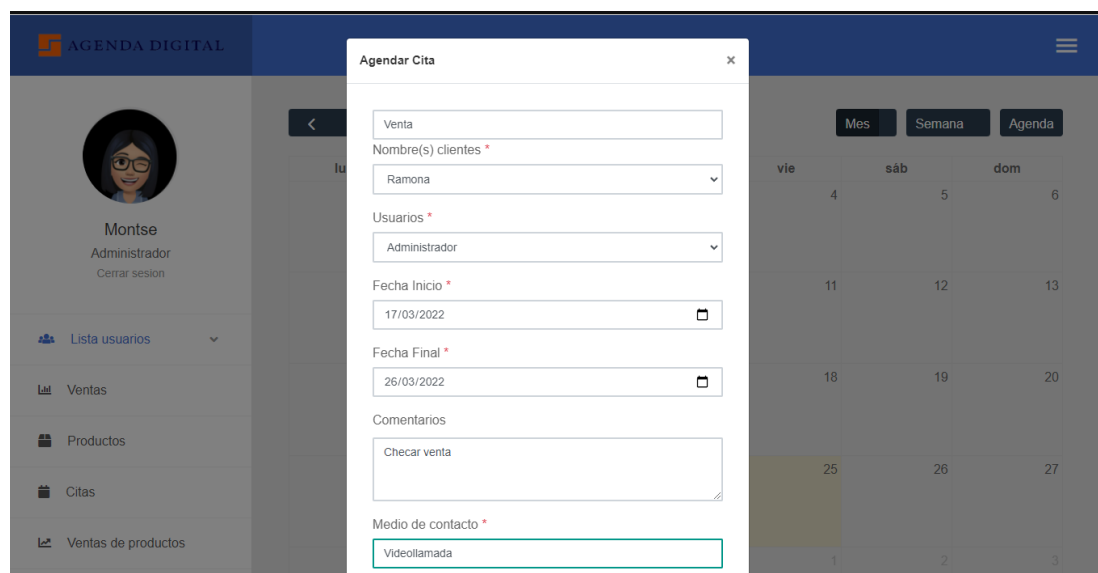


Ilustración 34 Modal Agregar Cita

Una vez agregada la cita, se podrá visualizar de manera general en el calendario

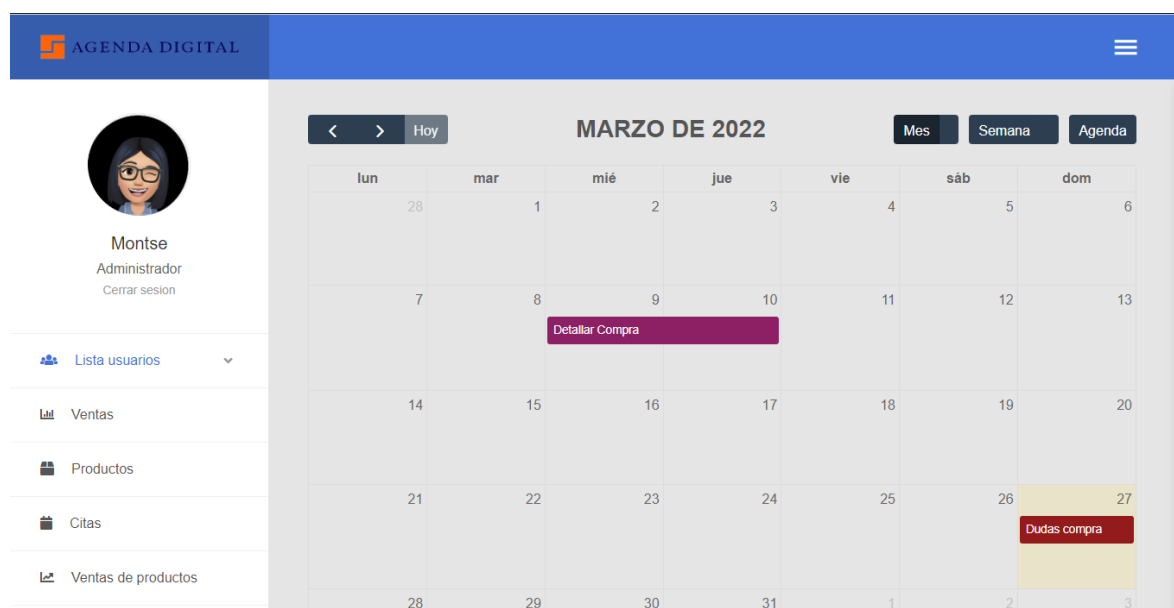


Ilustración 35 Citas en calendario

Para poder modificar o eliminar un evento, el modal cambiara agregándose un botón donde el usuario especificara la acción (eliminar o actualizar).

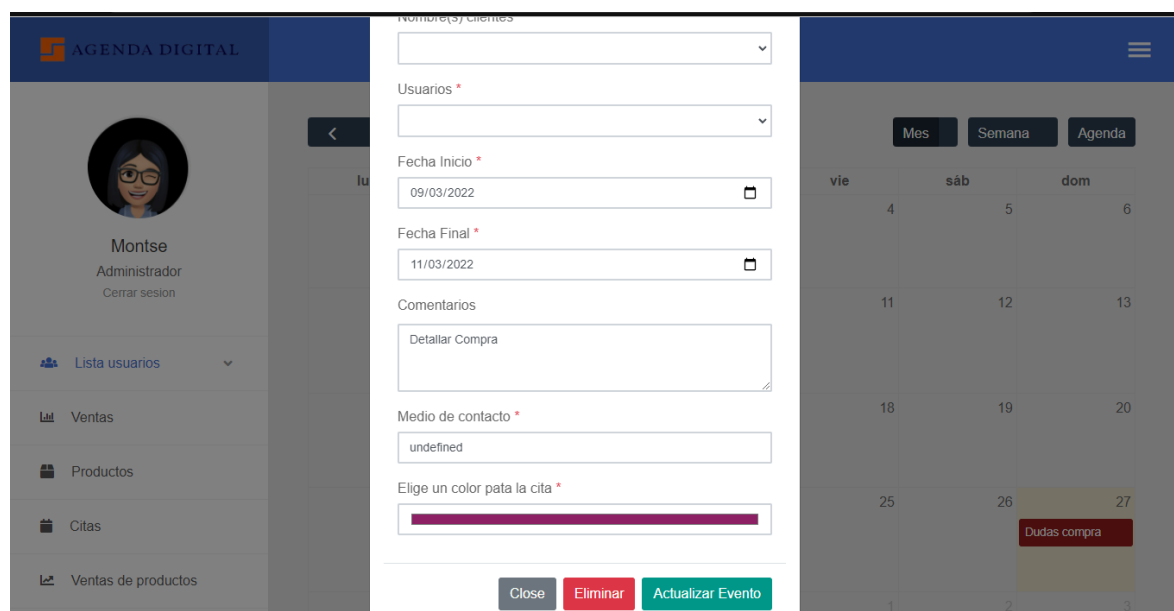


Ilustración 36 Botones de acciones

Al agregar o modificar los datos de la cita, el mensaje se mostrara de esta manera indicando que se han guardado los datos.

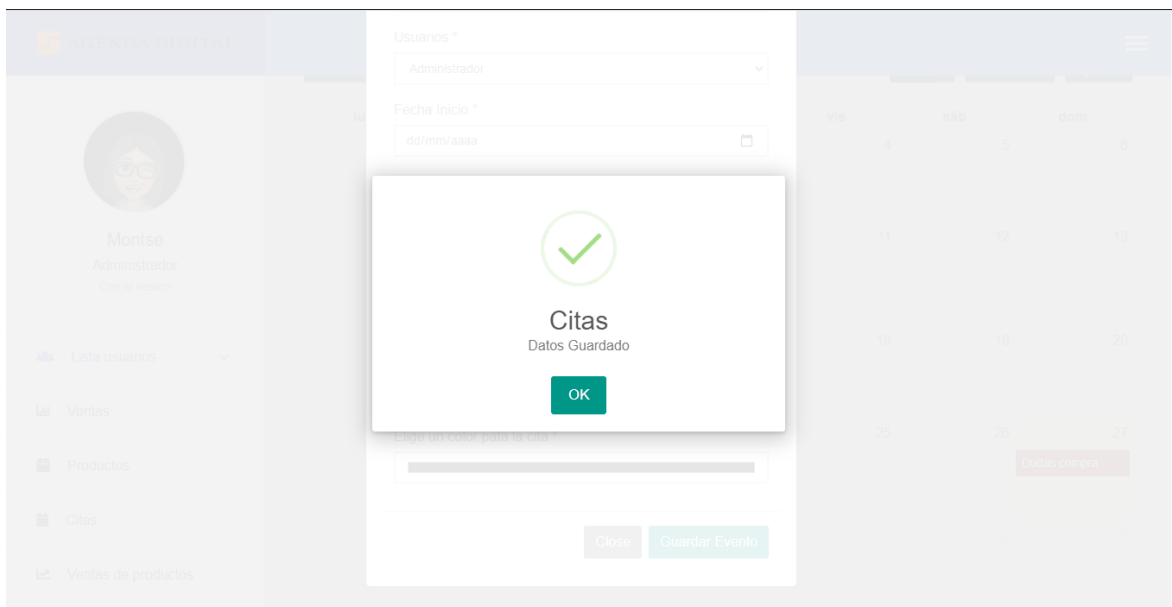


Ilustración 37 Mensaje de éxito citas

De lo contrario, en caso de que el usuario quiera eliminar la cita este arrojará un mensaje con la advertencia de eliminar la la cita.

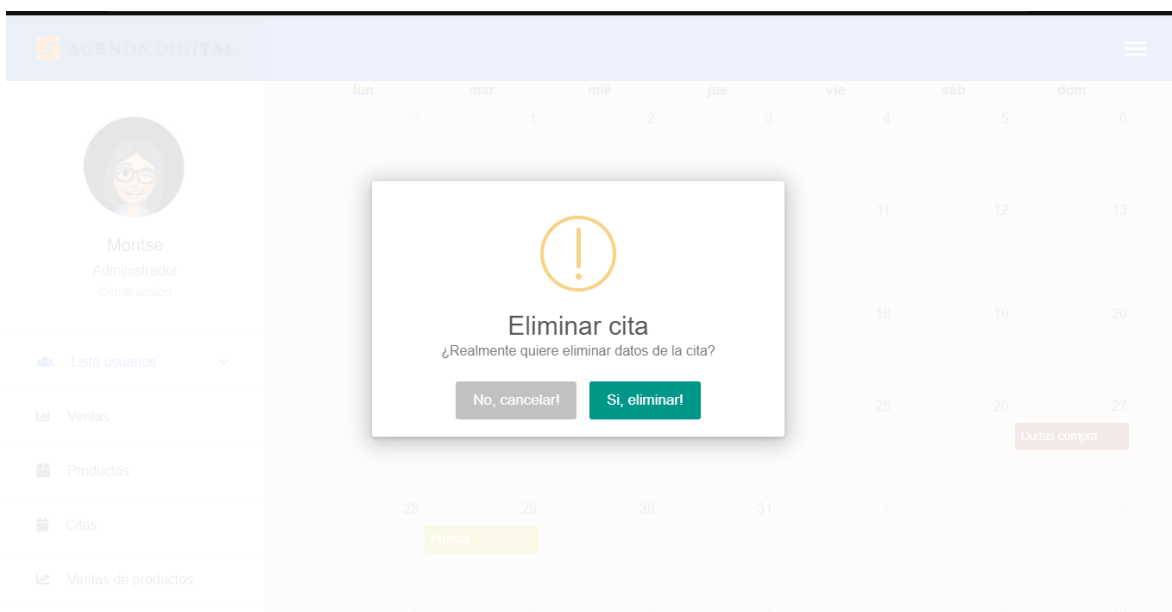


Ilustración 38 Eliminar cita

En cuanto al código, como ejemplo de la función del calendario indica en que formato se mostrará (Día, mes, año).

```
document.addEventListener('DOMContentLoaded', function() {
    var calendarEl = document.getElementById('calendar');
    var calendar = new FullCalendar.Calendar(calendarEl, {
        initialView: 'dayGridMonth',
        locale: 'es',
        headerToolbar: {
            left: 'prev,next,today',
            center: 'title',
            right: 'dayGridMonth, timeGridWeek, listWeek'
        },
        events: base_url + "/Citas/listar",
        dateClick: function(info) {
            frm.reset();
            document.getElementById('idcitas').value = '';
            document.getElementById('exampleModalLabel').textContent =
'Agregar evento';
            document.getElementById('btnAccion').textContent = 'Guardar
Evento';

            eliminarc.classList.add('d-none');
            document.getElementById('fechacitasI').value = info.dateStr;
            myModal.show();

        },
        eventClick: function(info) {
            console.log(info);
            document.getElementById('exampleModalLabel').textContent =
'Modificar evento';
            document.getElementById('btnAccion').textContent = 'Actualizar
Evento';

            eliminarc.classList.remove('d-none');
            document.getElementById('idcitas').value = info.event.id;
            document.getElementById('listidclientesc').value =
info.event.extendedProps;
            document.getElementById('listidusuarios').value =
info.event.extendedProps;
            document.getElementById('fechacitasI').value =
info.event.startStr;
            document.getElementById('fechacitasF').value =
info.event.endStr;
            document.getElementById('comentarios').value = info.event.title;
            document.getElementById('medioC').value =
info.event.tipo_contacto;
```

```
        document.getElementById('color').value =  
info.event.backgroundColor;  
        myModal.show();  
    }  
});
```

Se puede ver como algo extenso, pero solo está indicando en las funciones el como será el formato y como se vera el modal para poder agregar la cita de manera correcta mandando a llamar todas las variables que se tienen que llenar para que la base de datos la almacene.

Hasta ahora, ese el último módulo que se ha desarrollando y cumpliendo el objetivo de que sea una agenda digital pues la intención de una agenda es llevar el control de algún evento, en este caso las citas y el control de los productos, clientes y ventas.

Para poder ver el código, se esta almacenado y llevando el control con una herramienta de control de versiones, en este caso Git Hub, la cual pondré el link donde está el proyecto para que puedan apreciar la estructura completa del proyecto en la parte de la programación.

<https://github.com/Montse-14/VentasDigital.git>

CONCLUSIÓN

Agilizar el proceso de servicio en la parte administrativa, es el principal motivo por el cual se desarrolló este proyecto, con este sistema se evitará pérdida de tiempo respecto al recolectar información ya que esta puede ser almacenada en la nube o en el servidor que la empresa utilice, mientras que el control de todo está mejor administrado. Al mismo tiempo poder visualizar las citas es de gran ayuda para el usuario ya que así se organiza de mejor manera. En el caso de las ventas, se considera que es importante porque así se lleva un mejor registro de la mercancía y al mismo tiempo un mejor control del dinero que ingresa por la venta de los productos, tener reportes mensuales de manera digital y de manera instantánea y no esperar varios días para que se genere.

Con las herramientas utilizadas como el template, el lenguaje de programación en este caso PHP las cosas se agilizaron un poco ya que afortunadamente la mayor parte conocía y sabía cuáles eran las mejores técnicas para poder darle seguimiento a las tareas que se me asignaron, pero esto no difiere que no deje de estudiar pues las tecnologías siempre van avanzando y más en nuestra área, incluso tuve que hacer algún repaso para poder aclarar el proceso y no hacerlo tedioso.

Este proceso de estadía y la realización del proyecto ha superado mis expectativas puesto que no es lo mismo desarrollar un proyecto escolar a un proyecto del cual depende tu trabajo, me quedo con más conocimientos, la universidad solo fue parte del proceso pues es muy diferente pero no difícil, también quedo satisfecha por aprovechar los conocimientos que obtuve dentro de la universidad, con esto puedo decir que los objetivos planteados se han cumplido.

La implementación del sistema, no ha finalizado pues solo se ha documentado parte del proceso, ya que este seguirá en mantenimiento, para implementación y pruebas posteriores.

La información que se llevó a cabo en este documento fue fundamental para lograr el correcto desarrollo el sistema, con la idea de buscar

Finalmente, considero que esta tesina no solo es prueba de lo que se realizó si no, también espero que sirva de apoyo para nuevos procesos, mejoras, incluso un apoyo para quienes están adentrándose al mundo de la programación, diseño web u otras relacionadas, porque esta carrera se complementa de algunas otras y así es como se crean proyectos grandes y reconocidos. Principalmente a los nuevos estudiantes de la Universidad Tecnológica de Puebla, que vean cual ha sido el proceso, quizá cuando ellos implementen algo habrá una gran diferencia de como se ha programado este sistema y como ellos lo harían o que cambiarían, adaptándose a las nuevas tecnologías.

BIBLIOGRAFIA

- *PHP: ¿Qué es PHP? - Manual.* (s. f.). php. Recuperado 28 de enero de 2022, de <https://www.php.net/manual/es/intro-what-is.php>
- *¿Qué es un framework y para qué se utiliza?* (2018, 11 enero). Orix Systems. <https://www.orix.es/que-es-un-framework-y-para-que-se-utiliza>
- Barbettini, N. (s. f.). *Fundamentos de MVC El pequeño libro de ASP.NET Core*. aspnetcoremaster. <https://aspnetcoremaster.com/little-aspnetcore-book/chapters/mvc-basics/>
- Fernández, Y. (2019, octubre 30). *Qué es Github y qué es lo que le ofrece a los desarrolladores*. Xataka.com; Xataka. <https://www.xataka.com/basics/que-github-que-le-ofrece-a-desarrolladores>