



**UNIVERSIDAD TECNOLÓGICA DE NEZAHUALCÓYOTL**  
ORGANISMO PÚBLICO DESCENTRALIZADO DEL GOBIERNO DEL ESTADO DE MÉXICO

---

---

**DIVISIÓN DE INFORMATICA Y COMPUTACION**

**MEMORIA**

**QUE PARA OBTENER EL TÍTULO DE:**  
**Ing. En Desarrollo y Gestión Software**

Presenta  
**Villalba Mendoza Osvaldo** Matricula  
**232271007**

Proyecto  
**Auto Glass**

Empresa  
**Digital Jaguar RL S de C.V.**

Asesor Empresarial  
**Ing. Alonso Montiel Villar**

Asesora Académica  
**Ing. Erika Hernández Vázquez**

**septiembre 2022 – diciembre 2024**

**enero 2025**



**UNIVERSIDAD TECNOLÓGICA DE NEZAHUALCÓYOTL**  
Organismo Público Descentralizado del Gobierno del Estado de México  
División de Informática y Computación  
Ingeniería en Tecnologías de la Información y Comunicación

Ciudad Nezahualcóyotl, Estado de México a 19 de diciembre del 2024

**Constancia de Autorización de  
Examen Profesional**

Alumno (a): VILLALBA MENDOZA OSVALDO, con número de matrícula: 232271007, estudiante de la carrera Ingeniería en Desarrollo y Gestión de Software de la 7<sup>a</sup> Generación, se le autoriza presentar el proyecto Auto Glass, como parte de su Examen Profesional, para obtener el Título de Ingeniero en Desarrollo y Gestión de Software, el día **16 de enero 2025** de **09:00 a 10:00 horas**, con base a los lineamientos establecidos para el proceso de Titulación en esta Casa de Estudios. Quedando de común acuerdo para conformar el **Honorable Jurado** que dictaminará el resultado de su presentación, los ciudadanos:

Firma

Ing. Erika Hernández Vázquez

Secretario del Jurado

Ing. Alonso Montiel Villar

Representante empresarial

M. en C. Elvia María Hernández  
Espitia  
Vocal



## Constancia de autorización de presentación de Memoria

Nezahualcóyotl, Estado de México a 19 de Diciembre de 2024

Estimado Alumno **VILLALBA MENDOZA OSVALDO**, con número de matrícula **232271007**, estudiante de la carrera de Ingeniería en Desarrollo y Gestión de Software (ING DGS), de la generación 7, se le autoriza a presentar el proyecto denominado **Auto glass** con número de folio **ITIC20240301001** para concluir estadía profesional y obtener el Título de **Ingeniero en Desarrollo y Gestión de Software**, el día **16 de Enero de 2025** a las **09:00** horas, con base a los lineamientos establecidos para el proceso de Titulación de esta Casa de Estudios.

Quedando de común acuerdo para conformar el Honorable Jurado que dictaminaría el resultado de su presentación, a los ciudadanos:

Ing. Erika Hernández Vázquez  
Asesor Académico

Ing. Alonso Montiel Villar  
Asesor Empresarial

Elaboró: López Gutiérrez Sergio  
c.c.p. Expediente



UNIVERSIDAD TECNOLÓGICA DE NEZAHUALCÓYOTL  
CTO. UNIVERSIDAD TECNOLÓGICA S/N COL. BENITO JUÁREZ, NEZAHUALCÓYOTL, ESTADO DE MÉXICO  
CP 57000 TELÉFONOS: (01 55) 5716 9700

Página 1 de 1

19/DIC/2024 - 17:46:03

## **AGRADECIMIENTOS**

Agradezco al ingeniero Alonso Montiel Villar por su asesoramiento, aceptación y seguimiento del proyecto brindado por la empresa, así como por proporcionar las herramientas digitales necesarias para aplicar técnicas de desarrollo competitivas en el sector empresarial.

Extiendo mi gratitud a la ingeniera Erika Hernández Vázquez por su constante seguimiento y apoyo en la realización de esta memoria, destacando su continua revisión y consideración de cada aspecto desarrollado.

Finalmente, agradezco a Gabriel Licona Valdez por su inquebrantable apoyo durante la evaluación y por compartir sus valiosos puntos de vista a lo largo del desarrollo y redacción de este documento.

## DEDICATORIAS

A mi padre, Carlos Villalba Miguel, por inculcarme el valor de la perseverancia y otorgarme las herramientas necesarias para destacar. Gracias a su formación basada en los valores del respeto, la solidaridad y la empatía, así como su presencia y apoyo durante toda mi carrera, se ha convertido en uno de los pilares e inspiración para superar nuevos retos.

A mi madre, Patricia Concepción Mendoza Martínez, por su inquebrantable apoyo moral y su cuidado incansable de mis necesidades básicas. Por su constante preocupación por mi bienestar y salud, y por calmar mi espíritu en momentos de gran estrés. Su interés en mi desarrollo académico y profesional, así como sus palabras de ánimo y fortaleza, han sido fundamentales para alcanzar este logro.

A mi hermano, Uriel Villalba Mendoza, por sus invaluables recomendaciones y por sus constantes ovaciones que resaltaron mi habilidad en la informática. Gracias a su confianza en mis capacidades y su aliento incansable, me adentré en esta carrera con pasión y determinación, logrando culminarla con éxito. Su apoyo ha sido una fuente inagotable de inspiración y motivación en cada paso de este camino.

A mi hermana, Valeria Villalba Mendoza, por su inquebrantable interés y apoyo moral en cada paso de mi camino. Tu constante aliento y tu ejemplo de dedicación han sido una inspiración fundamental para mí. Espero que puedas verme como un ejemplo a seguir en tu propia formación, así como yo he aprendido a enfrentar los desafíos con determinación y pasión, inspirado por ti. Este logro también es un reflejo de la fortaleza y la inspiración que me has brindado, y deseo que te sirva de guía en tu propio camino.

A Perla Janet Davalos Ramírez, por su amor, paciencia y apoyo incondicional durante todo este viaje. Tu presencia en mi vida ha sido una fuente constante de motivación y fortaleza. Gracias por estar siempre a mi lado, brindándome ánimo en los momentos difíciles y celebrando conmigo cada pequeño logro.

## ÍNDICE TEMÁTICO

<b>RESUMEN</b> .....	10
<b>ABSTRAC</b> .....	11
<b>INTRODUCCION</b> .....	12
<b>PLATEAMIENTO DEL PROBLEMA</b> .....	13
<b>PROPUESTA DE SOLUCION</b> .....	14
<b>JUSTIFICACION</b> .....	15
<b>OBJETIVOS</b> .....	16
<b>CRONOGRAMA DE ACTIVIDADES</b> .....	17
<b>RESEÑA DE LA EMPRESA</b> .....	18
<b>Misión</b> .....	18
<b>Visión</b> .....	18
<b>Valores</b> .....	18
<b>METODOLOGIA (SRUM)</b> .....	19
<b>CAPITULO I: MARCO TEORICO</b> .....	20
<b>Visual Studio Code 1.95.3</b> .....	21
<b>Xampp 3.3.0</b> .....	21
<b>Postman 11.19</b> .....	22
<b>PHP 8.2</b> .....	23
<b>Javascript ES6</b> .....	23
<b>MySQL 5.7</b> .....	24
<b>Gitlab 17.5</b> .....	24
<b>CAPITULO II: INICIO</b> .....	25
<b>Visión del proyecto</b> .....	26
<b>CAPITULO III: PLANIFICACION Y ESTIMACION</b> .....	27
<b>Creación del backlog</b> .....	28
<b>Recolección de requisitos</b> .....	28
<b>Definición de Historias de Usuario</b> .....	28
<b>Priorización del backlog</b> .....	29
<b>Planificación del sprint</b> .....	29
<b>CAPITULO IV: IMPLEMENTACION</b> .....	31
<b>Sprint 1: Módulo de cotizaciones</b> .....	32
<b>Desarrollo</b> .....	32

<b>Desafíos y soluciones.....</b>	71
<b>Sprint 2: Módulo de trabajos.....</b>	71
<b>Desarrollo.....</b>	71
<b>Desafíos y soluciones.....</b>	86
<b>Sprint 3: Módulo de mapas.....</b>	86
<b>Desarrollo.....</b>	86
<b>Desafíos y soluciones.....</b>	91
<b>Sprint 4: Módulo de configuraciones.....</b>	92
<b>Desarrollo.....</b>	92
<b>Desafíos y soluciones.....</b>	104
<b>CAPITULO V: REVISION Y RETROSPECTIVA .....</b>	106
<b>Sprint 1: Módulo de cotizaciones .....</b>	107
<b>Sprint 2: Módulo de trabajos.....</b>	107
<b>Sprint 3: Módulo de mapas.....</b>	108
<b>Sprint 4: Módulo de configuraciones.....</b>	108
<b>CAPITULO VI: LANZAMIENTO .....</b>	109
<b>Entrega del incremento .....</b>	110
<b>Sprint 1: Módulo de cotizaciones .....</b>	110
<b>Sprint 2: Módulo de trabajos.....</b>	110
<b>Sprint 3: Módulo de mapas.....</b>	110
<b>Sprint 4: Módulo de configuraciones.....</b>	110
<b>CONCLUSIONES.....</b>	112
<b>RECOMENDACIONES .....</b>	114
<b>BIBLIOGRAFIA Y LITERATURA CONSULTADA.....</b>	116
<b>GLOSARIO .....</b>	117

## ÍNDICE DE FIGURAS Y PANTALLAS

<i>Figura 1 Logotipo del software visual studio code.....</i>	21
<i>Figura 2 Logotipo del software Xampp .....</i>	21
<i>Figura 3 Logotipo del software Postman .....</i>	22
<i>Figura 4 Logotipo del lenguaje php .....</i>	23
<i>Figura 5 Logotipo del lenguaje Javascript .....</i>	23
<i>Figura 6 Logotipo de MySQL.....</i>	24
<i>Figura 7 Logotipo de GitLab.....</i>	24
<i>Figura 8 Diagrama para la vista general de "Quotes" .....</i>	34
<i>Figura 9 Vista de recursos para el uso de datatables .....</i>	35
<i>Figura 10 Sección 1 cotización .....</i>	
<i>Figura 11 Sección 2 cotización.....</i>	36
<i>Figura 12 Vista de diseños de modals proporcionadas por el template .....</i>	37
<i>Figura 13 Vista de diseños de tabs proporcionadas por el template .....</i>	37
<i>Figura 14 Vista desde MySQL Workbench de una tabla estándar .....</i>	38
<i>Figura 15 Interfaz de MySQL Workbench más el modelo de proyecto Auto Glass ...</i>	40
<i>Figura 16 Script SQL generado desde MySQL Workbench. ....</i>	41
<i>Figura 17 Interfaz SQL de phpMyAdmin para creación de una nueva tabla. ....</i>	41
<i>Figura 18 Barra de navegación con las opciones hacia las tablas independientes... </i>	42
<i>Figura 19 Vista del panel principal para el datatable.....</i>	60
<i>Figura 20 Vista del modal para la creación de un nuevo registro.....</i>	61
<i>Figura 21 Vista del modal para ver los detalles de un registro.....</i>	61
<i>Figura 22 Vista del modal para editar un registro existente .....</i>	61
<i>Figura 23 Vista del script generado por MySQL Workbench para la tabla de ent_quotes.....</i>	62
<i>Figura 24 Vista del script ent_quotes junto con las relaciones con las tablas complementarias .....</i>	62
<i>Figura 25 Vista de un modal de nueva cotización .....</i>	63
<i>Figura 26 Vista de un modal de edición de una cotización creada.....</i>	64
<i>Figura 27 Pagina web de la biblioteca dropzone .....</i>	64
<i>Figura 28 Vista de dropzone con archivos subidos para una cotización nueva .....</i>	68
<i>Figura 29 Vista del modal para la búsqueda de un numero de parte .....</i>	69
<i>Figura 30 Vista del modal de búsqueda de numero de parte con validación de campos vacíos .....</i>	69
<i>Figura 31 Vista de un modal para la creación de enlaces de detalles de una cotización existente. ....</i>	70
<i>Figura 32 Vista de los detalles de una cotización existente .....</i>	70
<i>Figura 33 Vista principal del módulo Jobs.....</i>	72
<i>Figura 34 Vista de diferentes cards para analíticas.....</i>	73
<i>Figura 35 Vista de las 3 primeras pestañas del modal para Jobs .....</i>	73
<i>Figura 36 Vista de la última pestaña del modal junto con una vista previa del modal completo.....</i>	74

<i>Figura 37 Vista de pestañas para el modal de cotizaciones .....</i>	74
<i>Figura 38 Vista de modal para un enlace de pago junto con la vista previa de un recibo .....</i>	75
<i>Figura 39 Vista previa de un recibo proporcionado por el template .....</i>	75
<i>Figura 40 Diagrama de tablas a utilizar para módulo de trabajos .....</i>	76
<i>Figura 41 Vista del módulo independiente de estatus .....</i>	77
<i>Figura 42 Vista previa de las cards de analíticas dentro del módulo de trabajos .....</i>	77
<i>Figura 43 Apertura automática del modal para un trabajo .....</i>	79
<i>Figura 44 Vista del modal dentro de la pestaña de distribuidor.....</i>	80
<i>Figura 45 Sección de attachments para la carga de archivos.....</i>	81
<i>Figura 46 Vista de la última sección para ver el historial de actualizaciones .....</i>	84
<i>Figura 47 Vista inferior de la segunda sección del modal .....</i>	84
<i>Figura 48 Vista del modal de enlace de pago solicitando los campos obligatorios faltantes.....</i>	85
<i>Figura 49 Vista de recibo de pago de un trabajo.....</i>	86
<i>Figura 50 Vista de diagrama para el módulo de mapas .....</i>	87
<i>Figura 51 Vista de mapas con diferentes configuraciones proporcionadas por el template .....</i>	87
<i>Figura 52 Página oficial de la librería leaflet.....</i>	88
<i>Figura 53 Vista de mapa de usuario con sucursales existentes.....</i>	90
<i>Figura 54 Vista detalles de una sucursal.....</i>	91
<i>Figura 55 Diagrama de diseño para la vista de usuarios .....</i>	94
<i>Figura 56 Vista de edición de perfil dentro del template.....</i>	94
<i>Figura 57 Vista de los recursos de toast dentro del template.....</i>	95
<i>Figura 58 Vista de la tabla principal para los usuarios existentes .....</i>	95
<i>Figura 59 Modal de registro de nuevo usuario validando los datos.....</i>	96
<i>Figura 60 Alerta para usuario en caso de que el correo ya se encuentre previamente .....</i>	96
<i>Figura 61 Vista de diagrama para nuevo distribuidor .....</i>	97
<i>Figura 62 Vista para el registro de datos de un nuevo usuario con rol de distribuidor .....</i>	97
<i>Figura 63 Validación de datos sobre la creación de un nuevo registro .....</i>	98
<i>Figura 64 Mensaje toast para aviso de correo existente .....</i>	98
<i>Figura 65 Modal para creación de un enlace de detalles de usuario con mensaje toast .....</i>	98
<i>Figura 66 Vista detalles de un distribuidor .....</i>	99
<i>Figura 67 Modal para registro de un nuevo zip_code .....</i>	99
<i>Figura 68 Modal con listado de zip_codes .....</i>	100
<i>Figura 69 Modal para restablecimiento de contraseña con alerta para confirmación .....</i>	101
<i>Figura 70 Vista de inbox principal para pruebas desde mailtrap .....</i>	104
<i>Figura 71 Toast de confirmación de carga de imagen de perfil.....</i>	104

## RESUMEN

El proyecto denominado "Auto Glass", se desarrolló en la empresa Digital Jaguar, ubicada en el municipio de Nezahualcóyotl.

Busca solucionar las deficiencias operativas de CANO'S Auto Glass, quienes actualmente carecen de un sistema centralizado para la gestión de cotizaciones, seguimiento de tareas, administración de sucursales y personal (distribuidores, técnicos, vendedores, usuarios y empleados). Esto genera ineficiencias, errores, retrasos y dificulta la toma de decisiones, impactando negativamente la productividad y el servicio al cliente.

Para abordar estas problemáticas, se desarrolla un sistema empleando la metodología Scrum, que comprende las etapas de Planificación del Sprint, Scrum Diario, Revisión del Sprint y Retrospectiva del Sprint. El desarrollo se apoyará en herramientas como Codeigniter 3, MySql 5.7 para la base de datos y Ajax de jQuery 3.7.1 para peticiones.

El sistema incluirá módulos para automatizar las cotizaciones de parabrisas, gestionar el seguimiento del progreso de los trabajos, centralizar la información de las sucursales en un mapa interactivo y administrar usuarios, roles y permisos. Con la implementación de este sistema, se espera que CANO'S Auto Glass logre una mayor eficiencia operativa, optimice la toma de decisiones, fortalezca la seguridad de la información, mejore la atención al cliente y, en consecuencia, logre un crecimiento sostenible.

El sistema se convertirá en una herramienta esencial para la gestión diaria de la empresa, permitiendo una mejor asignación de recursos, un control más preciso del trabajo interno y una mayor capacidad de respuesta a las necesidades del cliente.

## ABSTRAC

The project called “Auto Glass” was developed in the company Digital Jaguar, located in the municipio de Nezahualcóyotl.

To solve the operational deficiencies of CANO'S Auto Glass, which currently lacks a centralized system for the management of quotations, task tracking, administration of branches and personnel (distributors, technicians, salespeople, users and employees). This generates inefficiencies, errors, delays and hinders decision making, negatively impacting productivity and customer service.

To address these issues, a system is developed using the Scrum methodology, which includes the stages of Sprint Planning, Daily Scrum, Sprint Review and Sprint Retrospective. The development will be supported by tools such as Codeigniter 3, MySql 5.7 for the database and Ajax from jQuery 3.7.1 for requests.

The system will include modules to automate windshield quotes, manage job progress tracking, centralize branch information on an interactive map and manage users, roles and permissions. With the implementation of this system, CANO'S Auto Glass is expected to achieve greater operational efficiency, optimize decision making, strengthen information security, improve customer service and, consequently, achieve sustainable growth.

The system will become an essential tool for the daily management of the company, enabling better resource allocation, more precise control of internal work and greater responsiveness to customer needs.

## INTRODUCCION

En el contexto corporativo contemporáneo, la eficacia en las operaciones y la rapidez en la respuesta son elementos de suma importancia para mantener la eficacia. CANO'S Auto Glass, compañía dedicada a la prestación de servicios de parabrisas, se afronta a retos debido a la falta de sistemas de gestión que simplifiquen la administración de procesos fundamentales como la cotización de parabrisas, el monitoreo de trabajos, la vista de sucursales y configuración de usuarios.

El propósito principal de este proyecto es la creación de un software completo que incluya funciones especializadas para satisfacer estas necesidades. Mediante cuatro módulos fundamentales—Cotizaciones, Trabajos, Mapas y Configuraciones—el sistema facilitará la automatización de procesos, la centralización de la administración operativa y la optimización de recursos, incrementando de esta manera la capacidad de respuesta de la compañía y su calidad de servicio.

La creación del software se realizó empleando la metodología ágil scrum, lo que facilitó la constante adaptación a las necesidades emergentes y la entrega continua de resultados. Esto, por consiguiente, propició la incorporación eficaz de las nuevas características, asegurando un efecto beneficioso tanto en la funcionalidad interna como en la satisfacción del cliente.

El presente documento registra el procedimiento integral de desarrollo e implementación del sistema, desde la detección del problema y la elaboración de la solución hasta la valoración de los resultados obtenidos. Este documento examina los desafíos vencidos, los avances alcanzados y las posibilidades futuras para seguir robusteciendo las operaciones de CANO'S Auto Glass.

## PLATEAMIENTO DEL PROBLEMA

El cliente CANO'S Auto Glass, presenta limitaciones en su capacidad para gestionar eficientemente ciertos procesos operativos clave. Entre estos problemas se encuentran la falta de un sistema especializado para la cotización de parabrisas, una deficiente visibilidad del progreso en las tareas asignadas al personal y la falta de herramientas adecuadas para la administración de sucursales y otros elementos importantes de la operación, como distribuidores, técnicos, personal de ventas, usuarios y empleados.

Estas carencias generan ineficiencias en la gestión diaria de la empresa, provocan retrasos en los tiempos de respuesta, errores en las cotizaciones, dificultades en el seguimiento del trabajo interno y una limitada capacidad para optimizar recursos. Además, la falta de un sistema centralizado para gestionar y visualizar información relacionada con las sucursales y el personal impide que la empresa tenga un control adecuado sobre sus operaciones, lo que afecta directamente a su productividad y capacidad de servicio al cliente.

Sin la incorporación de nuevas funcionalidades que cubran estas necesidades, la empresa continuará enfrentándose a dificultades para mejorar su eficiencia operativa.

## PROPUESTA DE SOLUCION

El proyecto consiste en la incorporación de funcionalidades especializadas para la solución de problemas en específico, obteniendo una mejora en la eficiencia y gestión operativa. Dentro de las funcionalidades planeadas a resolver se incluye un cotizado de costos de parabrisas (Quotes), que permitirá la cotización de parabrisas nuevos para distintos tipos de vehículos. El cálculo de la cotización se basará en las características y precios definidos por el cliente. Si se mantiene este flujo, se podrá realizar el registro y seguimiento de progresos (Jobs) que permitirá registrar los trabajos y hacer un seguimiento del avance de las tareas asignadas al personal interno de la empresa. Cada tarea podrá ser actualizada en función de su estado de avance. Otro punto en cuenta a tomar es la gestión y visualización de sucursales con el cual integrará un módulo para gestionar las sucursales del cliente, permitiendo configurar sus ubicaciones y visualizar todas ellas en un mapa interactivo. Para el tema de la gestión de diversos puntos como administrativos o usuarios se implementará un panel de configuración para gestionar detalles relacionados con distribuidores, técnicos ("tech"), personal de ventas ("sale"), usuarios y empleados. Esta funcionalidad ofrecerá flexibilidad para administrar y modificar la información de cada grupo.

Para garantizar la correcta implementación de las nuevas funcionalidades, se realizarán pruebas de integración que aseguren el funcionamiento conjunto de todos los módulos. También se optimizará el rendimiento para minimizar el consumo de recursos.

## JUSTIFICACION

La incorporación de funcionalidades especializadas, como el módulo de cotización de parabrisas, la gestión de tareas, la visualización de sucursales y la administración de usuarios, responde directamente a las problemáticas identificadas. Esta solución permitirá optimizar el tiempo y reducir errores mediante la automatización de procesos clave, como la generación de presupuestos. La gestión centralizada de la información y la visualización de datos en tiempo real facilitarán la toma de decisiones estratégicas y la asignación eficiente de tareas, lo que a su vez aumentará la productividad general. Además, la gestión unificada de usuarios, roles y permisos fortalecerá la seguridad de la información y permitirá un control de acceso granular. En consecuencia, se obtendrá una mayor satisfacción del cliente gracias a una capacidad de respuesta mejorada y una atención más eficiente. La solución propuesta no solo resolverá las dificultades actuales, sino que también fortalecerá la imagen empresarial y sentará las bases para un crecimiento sostenible basado en la eficiencia, la innovación y una sólida orientación al cliente.

## OBJETIVOS

### Objetivo general:

Optimizar la eficiencia operativa y la gestión comercial de la empresa mediante la incorporación de funcionalidades especializadas en su sistema actual que permitan mejorar la eficiencia operativa, optimizando los procesos de cotización de parabrisas, el registro y seguimiento de trabajos, la gestión de sucursales y la administración de distribuidores, técnicos, personal de ventas, usuarios y empleados. Asegurar una integración efectiva y un uso óptimo de los recursos, para mejorar su capacidad de respuesta al cliente, la toma de decisiones y el control sobre los procesos clave del negocio.

### Objetivos específicos:

- ✓ Automatizar el proceso de cotización de parabrisas para reducir los tiempos de respuesta, minimizar los errores y mejorar la experiencia del cliente.
- ✓ Implementar un sistema de gestión de tareas y seguimiento del progreso para optimizar la asignación de recursos, monitorizar el avance de las actividades e identificar cuellos de botella.
- ✓ Centralizar la gestión y visualización de las sucursales para facilitar la toma de decisiones estratégicas, optimizar la distribución de recursos y mejorar la comunicación interna.
- ✓ Fortalecer la seguridad y el control de acceso al sistema mediante la implementación de un módulo de administración de usuarios, roles y permisos.
- ✓ Mejorar la capacidad de análisis y generación de informes para obtener información estratégica sobre el desempeño del negocio y la toma de decisiones informadas.
- ✓ Incrementar la satisfacción del cliente mediante una mayor velocidad de respuesta, una información más precisa y una atención personalizada.

## CRONOGRAMA DE ACTIVIDADES

TAREA	ASIGNADO A	PROGRESO	INICIO	FIN	DÍAS
<b>Sprint 1: Modulo de cotizaciones</b>		100%	02-09-24	03-10-24	32
Inicio	Osvaldo Villalba	100%	02/09/2024	06/09/2024	5
Planificacion y estimacion	Osvaldo Villalba	100%	10/09/2024	11/09/2024	2
Implementación	Osvaldo Villalba	100%	12/09/2024	27/09/2024	16
Revision y retrospeciva	Osvaldo Villalba	100%	30/09/2024	01/10/2024	2
Lanzamiento	Osvaldo Villalba	100%	02/10/2024	03/10/2024	2
<b>Sprint 2: Modulo de trabajos</b>		100%	04-10-24	31-10-24	28
Inicio	Osvaldo Villalba	100%	04/10/2024	07/10/2024	4
Planificacion y estimacion	Osvaldo Villalba	100%	08/10/2024	10/10/2024	3
Implementación	Osvaldo Villalba	100%	11/10/2024	25/10/2024	15
Revision y retrospeciva	Osvaldo Villalba	100%	28/10/2024	29/10/2024	2
Lanzamiento	Osvaldo Villalba	100%	30/10/2024	31/10/2024	2
<b>Sprint 3: Modulo de mapas</b>		100%	31-10-24	15-11-24	16
Inicio	Osvaldo Villalba	100%	31/10/2024	01/11/2024	2
Planificacion y estimacion	Osvaldo Villalba	100%	04/11/2024	05/11/2024	2
Implementación	Osvaldo Villalba	100%	06/11/2024	11/11/2024	6
Revision y retrospeciva	Osvaldo Villalba	100%	12/11/2024	13/11/2024	2
Lanzamiento	Osvaldo Villalba	100%	14/11/2024	15/11/2024	2
<b>Sprint 4: Modulo de configuraciones</b>		100%	18-11-24	06-12-24	19
Inicio	Osvaldo Villalba	100%	18/11/2024	19/11/2024	2
Planificacion y estimacion	Osvaldo Villalba	100%	20/11/2024	21/11/2024	2
Implementación	Osvaldo Villalba	100%	22/11/2024	02/12/2024	11
Revision y retrospeciva	Osvaldo Villalba	100%	03/12/2024	04/12/2024	2
Lanzamiento	Osvaldo Villalba	100%	05/12/2024	06/12/2024	2

## RESEÑA DE LA EMPRESA

### Misión

Es impulsar la transformación digital de empresas y organizaciones a través de soluciones tecnológicas innovadoras y a medida. Nos dedicamos a brindar servicios de desarrollo de software, consultoría de TI y soluciones de analítica de datos, con el compromiso de potenciar la eficiencia, la competitividad y el crecimiento sostenible de nuestros clientes. Buscamos ser líderes en el mercado, destacando por nuestra excelencia, creatividad y enfoque centrado en el cliente. ¡Juntos, avanzamos hacia un futuro tecnológico más brillante y conectado!"

### Visión

Ser reconocidos como líderes en el mercado de soluciones tecnológicas, impulsando la excelencia y la innovación en cada proyecto. Nos proyectamos como el aliado estratégico que transforma la forma en que las empresas operan y crecen en el entorno digital. Con un enfoque centrado en el cliente y la pasión por la tecnología, aspiramos a ser agentes de cambio, impulsando el progreso y la competitividad de nuestros clientes en un mundo cada vez más conectado

### Valores

Nuestra dedicación a la innovación y excelencia tecnológica impulsa cada proyecto que emprendemos. Valoramos la colaboración estrecha con nuestros clientes, comprendiendo sus necesidades para ofrecer soluciones a medida que transformen sus operaciones y fomenten su crecimiento. Nuestros estamos comprometidos con la integridad y la calidad en cada fase del desarrollo nos convierte en socios confiables y de confianza. Somos agentes de cambio, en busca de la excelencia en un mundo digitalmente conectado."

## METODOLOGIA (SRUM)

El Desarrollo del Proyecto Auto Glass será realizado en conjunto con la metodología Scrum, la cual tiene la definición que es una metodología ágil para la gestión de proyectos, especialmente utilizada en el desarrollo de software. Se basa en un enfoque iterativo e incremental, lo que significa que el trabajo se divide en ciclos cortos llamados sprints, generalmente de dos a cuatro semanas.

Scrum permite a los equipos trabajar de manera colaborativa y adaptarse rápidamente a los cambios. Al final de cada sprint, se revisa el progreso y se ajusta el plan según sea necesario. Esto asegura que el producto evolucione de acuerdo con las necesidades del mercado.

Al usar esta metodología ágil, se necesita saber que las personas desempeñaran un rol en específico, los cuales pueden ser los siguientes:

- Product Owner: Define y prioriza las características del producto.
- Scrum master: Facilita el proceso y elimina obstáculos.
- Equipo de desarrollo: Trabaja en la creación del producto.

La forma en la cual suceden los actividades o eventos también deben surgir de la siguiente estructura:

- Sprint Planning: Una reunión que se lleva a cabo todos los días para planificar el día y sincronizar actividades.
- Sprint review: La revisión del trabajo concluido de un sprint.
- Sprint retrospective: Reflexión sobre el proceso y como mejorarlo.

Por último, esta metodología dispone de ciertas herramientas o artefactos los cuales ayudaran en la eficiencia del proyecto:

- Product backlog: Lista priorizada de todo lo que se necesita en el producto.
- Sprint backlog: Lista de tareas a completar en el sprint.
- Incremento: Resultado del trabajo durante el sprint.



## CAPITULO I: MARCO TEORICO

## CAPITULO I – MARCO TEORICO

Dentro del desarrollo web se tiene una exorbitante cantidad de herramientas y tecnologías con las que se pueden desarrollar y crear aplicaciones que den solución a diferentes problemáticas, basándonos en esta idea, este marco teórico mostrara a una manera breve la herramientas y tecnologías que satisfagan a el proyecto Auto Glass.

### Visual Studio Code 1.95.3



*Figura 1 Logotipo del software visual studio code*

Es un editor de código fuente desarrollado por Microsoft, diseñado para ser ligero, rápido y altamente personalizable. A diferencia de otros editores de código, como los IDE completos (Entornos de Desarrollo Integrados), VS Code se enfoca en ofrecer una experiencia ágil para la escritura y edición de código, sin sacrificar características avanzadas.

Este será el software principal con el cual se accederá al código fuente del proyecto, así mismo permitirá la modificación e implementación de las funcionalidades nuevas a crear.

### Xampp 3.3.0



*Figura 2 Logotipo del software Xampp*

Es uno de los servidores web multiplataforma más utilizados, que ayuda a los desarrolladores a crear y probar sus programas en un servidor web local. Fue desarrollado por Apache Friends y su código fuente nativo puede ser revisado o modificado por la audiencia.

Con el uso de esta herramienta podremos simular un espacio de área de trabajo interno, lo cual asimilara las características más similares a un entorno de producción, con la ventaja de no irrumpir dentro de este.

### **Postman 11.19**



*Figura 3 Logotipo del software Postman*

En sus inicios nace como una extensión que podía ser utilizada en el navegador Chrome de Google y básicamente nos permite realizar peticiones de una manera simple para testear APIs de tipo REST propias o de terceros.

Gracias a los avances tecnológicos, Postman ha evolucionado y ha pasado de ser de una extensión a una aplicación que dispone de herramientas nativas para diversos sistemas operativos como lo son Windows, Mac y Linux.

El uso de postman será más específico dentro del desarrollo del proyecto, ya que se utilizará a modo de comprobación de pruebas dentro del proceso de desarrollo de ciertos sprints extensos o con una funcionalidad particular.

## PHP 8.2



Figura 4 Logotipo del lenguaje php

PHP es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario. El PHP generalmente es definido como un lenguaje del lado del servidor. Esto significa que se aplica en la programación que tiene lugar en el servidor web responsable de ejecutar la aplicación o, más a menudo, en un sitio web.

## Javascript ES6



Figura 5 Logotipo del lenguaje Javascript

JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado.

## MySQL 5.7



Figura 6 Logotipo de MySQL

En cuanto a la definición general, MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor. RDBMS es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional.

## Gitlab 17.5



Figura 7 Logotipo de GitLab

Es una plataforma web para la gestión del ciclo de vida del desarrollo de software mediante el control de versiones de código fuente, sirviendo también como repositorio online para almacenar proyectos, el mantener diferentes versiones del código para pruebas individuales será una ventaja, además de tener un historial de cambios realizados a lo largo del proceso.



## CAPITULO II: INICIO

## CAPITULO II – INICIO

### Visión del proyecto

El proyecto de optimización del sistema de gestión de la empresa tiene como visión transformar la eficiencia operativa y la gestión comercial a través de la integración de funcionalidades especializadas que no solo mejoren los procesos internos, sino que también eleven la experiencia del cliente a un nuevo nivel. La implementación de estos módulos avanzados permitirá a la empresa responder con mayor rapidez y precisión a las necesidades del mercado, optimizando recursos y fortaleciendo su capacidad competitiva.

Este proyecto se enfoca en:

1. **Automatización de Cotizaciones:** Implementar un sistema que agilice el proceso de cotización de parabrisas, reduciendo los tiempos de respuesta y minimizando errores, lo que mejorará significativamente la experiencia del cliente.
2. **Gestión de Tareas y Seguimiento:** Desarrollar un sistema robusto para la gestión de tareas y el seguimiento del progreso, asegurando una asignación óptima de recursos, monitoreo constante del avance de las actividades e identificación rápida de cuellos de botella.
3. **Centralización de la Gestión de Sucursales:** Centralizar la administración de todas las sucursales para facilitar la toma de decisiones estratégicas, optimizando la distribución de recursos y mejorando la comunicación interna, lo cual es crucial para el crecimiento y la eficiencia de la empresa.
4. **Seguridad y Control de Acceso:** Fortalecer la seguridad mediante un módulo de administración de usuarios que gestione roles, garantizando un acceso controlado y seguro al sistema.

Con esta visión, se espera no solo optimizar los procesos operativos y comerciales, sino también crear un sistema integral que potencie la eficiencia, seguridad y capacidad de análisis de la empresa, posicionándola como líder en su sector.



## CAPITULO III: PLANIFICACION Y ESTIMACION

## CAPITULO III – PLANIFICACION Y ESTIMACION

### Creación del backlog

#### Recolección de requisitos

El primer paso para crear el backlog del producto fue recolectar todos los requisitos necesarios para el proyecto. Esto se realizó mediante reuniones con los stakeholders, análisis de la situación actual de la empresa y entrevistas con los usuarios finales. Se identificaron las funcionalidades clave que debían ser desarrolladas, enfocándonos en los objetivos generales y específicos del proyecto.

#### Definición de Historias de Usuario

Con los requisitos recopilados, se definieron historias de usuario que describen de manera detallada las necesidades y deseos de los usuarios. Cada historia de usuario se enfocó en un aspecto específico del sistema que debía ser desarrollado. Por ejemplo:

- **Historia de Usuario 3 (Módulo de Cotizaciones):** "Como cliente, quiero recibir una cotización precisa de un parabrisas, para decidir si proceder con la reparación."
- **Historia de Usuario 2 (Módulo de Trabajos):** "Como gerente, quiero asignar tareas a los técnicos y seguir su progreso, para asegurarme de que se completen a tiempo."
- **Historia de Usuario 1 (Módulo de Mapas):** "Como usuario, quiero ver todas las sucursales en un mapa interactivo, para poder localizar fácilmente la más cercana."
- **Historia de Usuario 4 (Módulo de Configuraciones):** "Como administrador, quiero gestionar las cuentas de usuario dependiendo del rol que desempeñe."

## Priorización del backlog

Una vez definidas las historias de usuario, se procedió a priorizar el backlog del producto. Se utilizó una matriz de priorización que consideraba el valor para el negocio, la urgencia y la complejidad de implementación. Las historias de usuario se ordenaron según su prioridad, asegurando que las funcionalidades más críticas fueran abordadas primero. La priorización se realizó en colaboración con los stakeholders y el Product Owner.

- Alta prioridad:
  - Implementar el sistema de cotizaciones (Módulo de Cotizaciones)
  - Gestión de tareas y seguimiento del progreso (Módulo de Trabajos)
- Media prioridad:
  - Desarrollar la visualización de sucursales en el mapa (Módulo de Mapas)
  - Administración de usuarios (Módulo de Configuraciones)
- Baja prioridad:
  - Mejoras en la interfaz de usuario
  - Funcionalidades adicionales solicitadas por los usuarios finales

## Planificación del sprint

Cada historia de usuario se desglosó en tareas más pequeñas y manejables que podrían ser completadas dentro de un sprint. Estas tareas se documentaron claramente en el backlog del producto.

- Tareas para el sprint 1 módulo de cotizaciones:
  - Diseño de la interfaz de usuario para la creación de cotizaciones.
  - Implementación de la lógica de cálculo de precios.
  - Integración del sistema de cotizaciones con la base de datos.
  - Pruebas y validación de la funcionalidad de cotizaciones.
- Tareas para el sprint 2 módulo de trabajos:
  - Diseño de la interfaz de usuario para la creación de trabajos.

- Implementación de la lógica para la actualización del progreso de cada tarea
- Integración del sistema de trabajos con la base de datos
- Pruebas y validación de funcionalidad de trabajos
- Tareas para el sprint 3 módulo de mapas:
  - Diseño de la interfaz de usuario para la creación de mapas.
  - Implementación del uso de API's para localización real dentro de un mapa
  - Carga de sucursales cercanas a la zona de la ubicación actual del usuario.
- Tareas para el sprint 4 módulo de configuraciones:
  - Diseño de la interfaz de usuario para la creación de configuraciones.
  - Diseño de la interfaz de usuario para la creación de cualquier rol de un usuario.
  - Implementación de carga de archivos y fotos de perfil
  - Implementación de encriptado de contraseña y opciones para su restablecimiento.
  - Integración del sistema de configuraciones con la base de datos.
  - Pruebas y validación de funcionalidad de trabajos.



## CAPITULO IV: IMPLEMENTACION

## CAPITULO IV – IMPLEMENTACION

### Sprint 1: Módulo de cotizaciones

#### Desarrollo

El módulo de cotizaciones se encarga de generar presupuestos detallados para los clientes, asegurando precisión mediante la validación de campos y el cálculo automático de costos. Cada cotización debe registrar los siguientes datos:

#### **Datos generales de la cotización:**

- quote\_name: Nombre de la cotización
- phone: Número telefónico
- vin: Clave VIN de un automóvil
- glass\_cost: Coste del cristal.
- others: Costes extra adicionales a los establecidos
- subtotal: Sumatoria de los conceptos establecidos y others
- tax: Impuesto aplicado dependiendo el estado
- total: Sumatoria del subtotal más el porcentaje de impuestos establecidos

#### **Datos generales del cliente:**

- first\_name: Nombre
- last\_name: Apellidos
- phone\_client: Número de teléfono del cliente
- alternative\_phone: Teléfono alternativo como casa u oficina
- email: Correo electrónico
- address: Dirección

#### **Datos generales de la póliza de seguro:**

- insuance\_carrier: Proveedor de seguro
- policy\_id: Numero de póliza
- refecal: Referencia de la póliza
- policity\_holder: Titular de la póliza
- policity\_address: Dirección del titular

- deductible: Deducible

**Detalles del trabajo:**

- nags\_description: Detalles del trabajo
- nags\_labior: Mano de obra
- list\_price: Precio listado
- price\_per\_hour: Precio por hora.

**Identificadores del vehículo y localización:**

- id\_make: Marca
- id\_model\_car: Modelo del vehículo
- id\_body: Tipo de carrocería
- id\_glass: Tipo de cristal
- id\_part\_number: Número de parte
- id\_price\_tier: Nivel de precios
- id\_calibration\_type: Tipo de calibración
- id\_ciudades: Ciudad
- id\_estados: Estado
- id\_calibration\_location: Ubicación de calibración
- id\_year: Año del vehículo
- id\_zip\_code: Código postal
- notes: Notas adicionales
- id\_tag: Etiqueta

Para ciertos campos como make, model\_car, body, glass, part\_number, price\_tier, calibration\_type, ciudades, estados, calibration\_location, year, zip\_code y tags, se recomienda un módulo independiente que permita al cliente agregar nuevas opciones de forma sencilla, facilitando una gestión flexible de futuras cotizaciones.

Se había mencionado que el módulo deberá de realizar un cálculo de costos, dicho calculo se deberán tomar las siguientes consideraciones para reflejar dentro del subtotal:

- La selección del código postal se aplicará un coste diferente

- El valor definido dentro de glass cost
- La opción seleccionada dentro de Price tier
- La opción seleccionada dentro de calibration type
- La opción seleccionada dentro de calibration location
- El valor definido dentro de others

Se realizarán validaciones en los campos esenciales para asegurar una cotización precisa y evitar errores dentro del sistema. Estas validaciones mejorarán la fiabilidad del módulo y optimizarán la experiencia del cliente al registrar sus datos.

El módulo de cotizaciones será un punto de partida para más procesos relacionados, como la asignación de una orden de trabajo, la cual necesita de una cotización previa, por esto mismo el módulo de cotizaciones deberá contar con la opción para generar una orden de trabajo al mismo tiempo de registrar una nueva cotización.

Se tendrá la opción para generar una búsqueda de un numero de parte específico, el cual permita un registro más rápido de una cotización o una orden de trabajo nueva.

El panel principal de cotizaciones contara con una opción para la creación de un enlace para mostrar detalles de una cotización realizada al cliente.

Dentro de un diagrama compartido por el cliente, se tiene la visión de mantener una tabla con detalles de cada cotización, botones de acción para la creación de cotizaciones, la creación de enlaces hacia una cotización existente, así como un buscador de numero de parte.

Quote No.	Date	Year	Model	Model	Job Tape	Quoted Price	Zip Code	Sale	Tag	Call Back
Q00001	06/30/22	2022	Silverado	1500	Windshield	285.00	92804	Antonio	Call Back	▼
Q00002	Agregar te	Agregar te	Agregar texto	Agregar te	Agregar te	▼				
Q00002	Agregar te	Agregar te	Agregar texto	Agregar te	Agregar te	▼				
Q00004	Agregar te	Agregar te	Agregar texto	Agregar te	Agregar te	▼				
Q00005	Agregar te	Agregar te	Agregar texto	Agregar te	Agregar te	▼				

Figura 8 Diagrama para la vista general de "Quotes"

Se tomará para la vista de la tabla de datos el uso de datatables las cuales nos proporcionan herramientas como un buscador, vista de cierto número de registros, selección, ver detalles, etc. Una opción para tomar será la datatable “Multilingual” la cual incluye la barra de búsqueda y la vista de un numero de registros más específico.

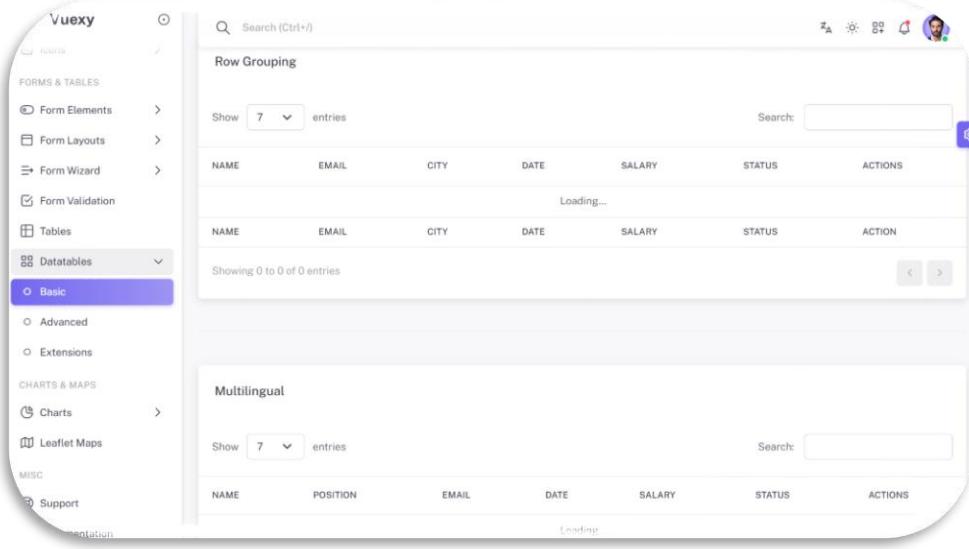


Figura 9 Vista de recursos para el uso de datatables

Para la creación, edición o ver los detalles de una cotización, se tiene un diagrama con el cual incluye todos los campos a integrar, el ordenamiento que llevará, así como la integración de pestañas para dividir la sección de archivos adjuntos con los datos de mayor relevancia, se observa para el apartado de archivos adjuntos que se podrá tener una vista previa del contenido subido. También se muestran elementos como valores de ejemplo en los apartados para la realización de cálculos y sumatorias dependiendo los valores seleccionados.

Cabe recalcar que para el tema de ver los detalles de una cotización los campos estarán disponibles para solo lectura, así como una vista individual de cada detalle de una cotización para que el usuario final pueda visualizar solo lo necesario de su cotización correspondiente.

Figura 10 Sección 1 cotización

Figura 11 Sección 2 cotización

El template nos proporciona diferentes diseños atractivos para el uso de un modal, hay desde la animación de mostrarse en horizontal o vertical, así como el tipo de contenido a manejar en su interior, en todos los modals que se nos proporcionan no hay alguno que incluya pestañas de navegación interna, por lo tanto, al realizar una búsqueda dentro de otros elementos se tiene el apartado “Tabs & Pils” en ella se tienen diseños variados al uso de pestañas, así que para el diseño del modal para una cotización se hará uso de ambos recursos para realizar una fusión entre ambos lo cual nos ayudara en tener todos los campos divididos y con una mejor organización.

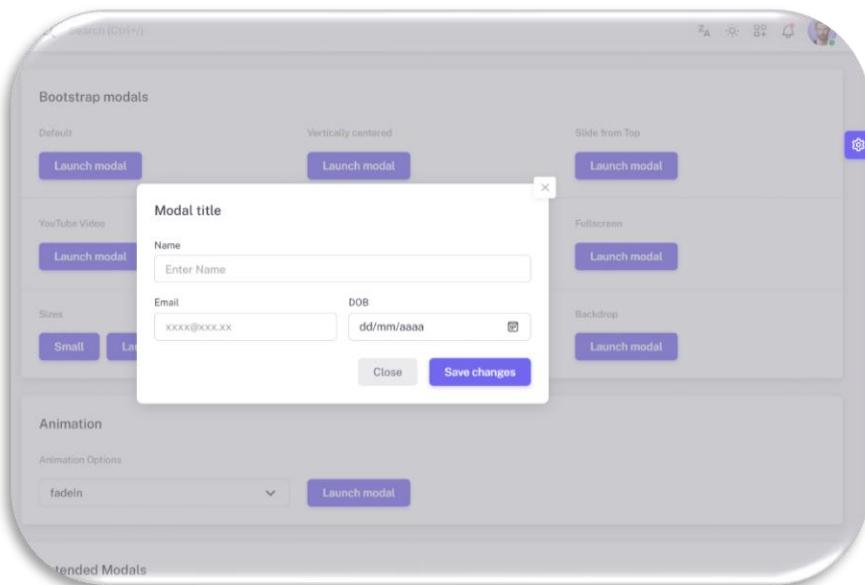


Figura 12 Vista de diseños de modals proporcionadas por el template

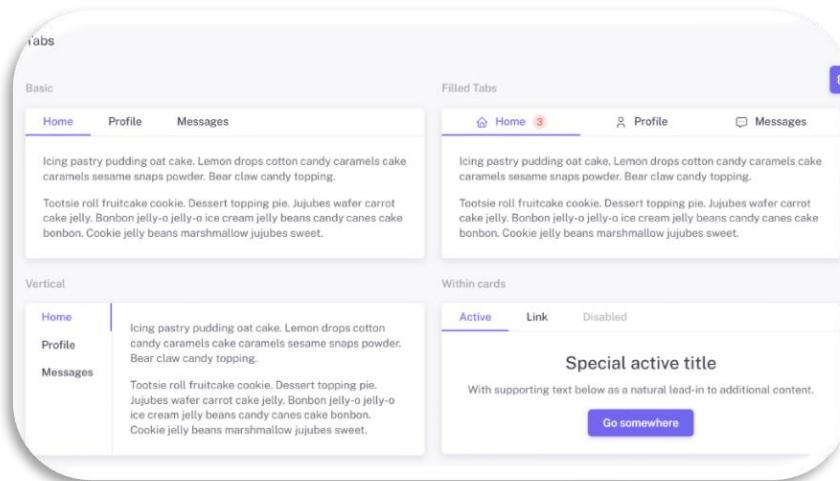


Figura 13 Vista de diseños de tabs proporcionadas por el template

La vista para los módulos independientes que también se usarán para la creación de una cotización se usara un datatable igual al principal, así como una vista de modal sin pestañas ya que en estos solo se tendrán las funcionalidades de un CRUD.

El proceso para la creación de un registro nuevo requiere información de los módulos independientes, por lo que se comenzara con el desarrollo de un módulo independiente general o “estándar”, dichos módulos solo contaran con las acciones de

creación, lectura, edición y eliminación, más una vista general de los campos en el datatable.

Cabe mencionar que, para el desarrollo de todas las tablas, en la base de datos se tiene una estructura esencial, su estructura es la siguiente:

- Id: El numero identificar ascendiente de cada registro, un número que no se repite.
- Name: El nombre o palabra clave que identificara de forma contextual el tipo de registro.
- Active: El estatus interno con el que se tiene (activo, inactivo, eliminado).
- Created\_by: Campo para registrar al usuario quien creo el registro.
- Created\_at: La fecha y hora en la cual se capturo la creación del registro.
- Updated\_by: Campo para capturar al usuario quien realizo la ultima edición del registro.
- Updated\_at: Fecha y hora correspondiente a la última actualización realizada.
- Deleted\_by: Campo correspondiente para agregar al usuario quien realizo la opción de eliminación de un registro.
- Deleted\_at: Campo correspondiente para registrar la fecha y hora para la eliminación de un registro.

cat_team	
id_team	INT
team_name	VARCHAR(155)
active	TINYINT
created_by	INT
created_at	DATETIME
updated_by	INT
updated_at	DATETIME
deleted_by	INT
deleted_at	DATETIME
Indexes	
•	PRIMARY

Figura 14 Vista desde MySql Workbench de una tabla estándar

Se tiene una lista de puntos clave a seguir para el desarrollo de cualquier modulo, siendo indispensable para el comienzo de cualquier tabla creada, los puntos clave son los siguientes:

1. Modelado de tabla con sus relaciones (en caso de ser necesario)
2. Creación de la tabla dentro de la base de datos local
3. Creación de la ruta para acceder con una url
4. Agregar opción en el menú (barra de navegación)
5. Carpeta Model con su respectivo archivo de modelo
6. Carpeta Controller con su respectivo archivo de controlador
7. Carpeta View con su respectivo archivo de vista
8. Configuración de nombres de etiquetas para el idioma ingles
9. Configuración de archivo Script para el proceso de peticiones de un CRUD, más la carga de datos de la datatables y acciones necesarias de interacción.
10. Agregar Script en “Footer.php”, archivo que realiza la validación de la sección en donde se encuentra para cargar el archivo javascript correspondiente.

Para el modelado de la tabla, se hará uso de MySql Workbench para abrir el diagrama de las tablas que se tengan del proyecto, la herramienta nos ofrecerá una interfaz clara y concisa para incluir nuevas tablas, realizar relaciones de una u otra así como generar un script SQL para ejecutar dentro de producción para ejecutar los cambios solicitados o dentro del entorno local para realizar las pruebas necesarias, este paso no se puede omitir y será indispensable para el punto de partida de los futuros módulos a realizar ya que dentro de este modelado, se ayudara a futuros desarrolladores en la comprensión y adaptabilidad más rápida al funcionamiento del proyecto.

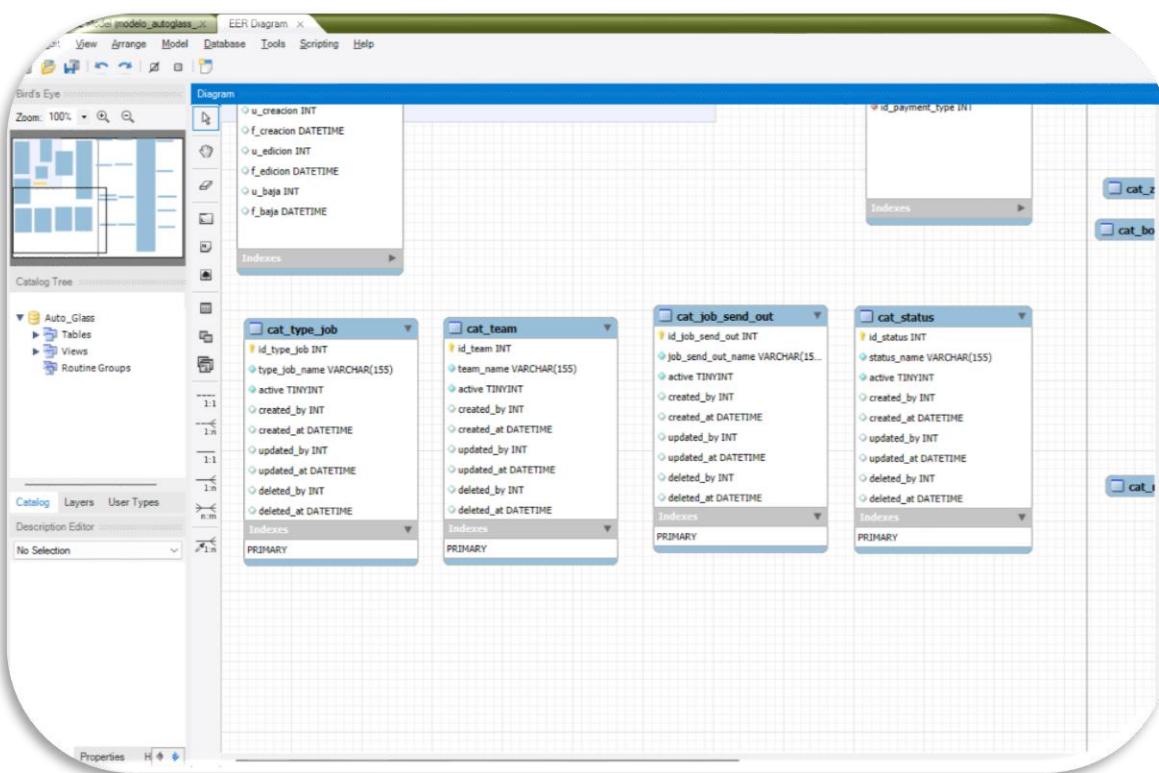
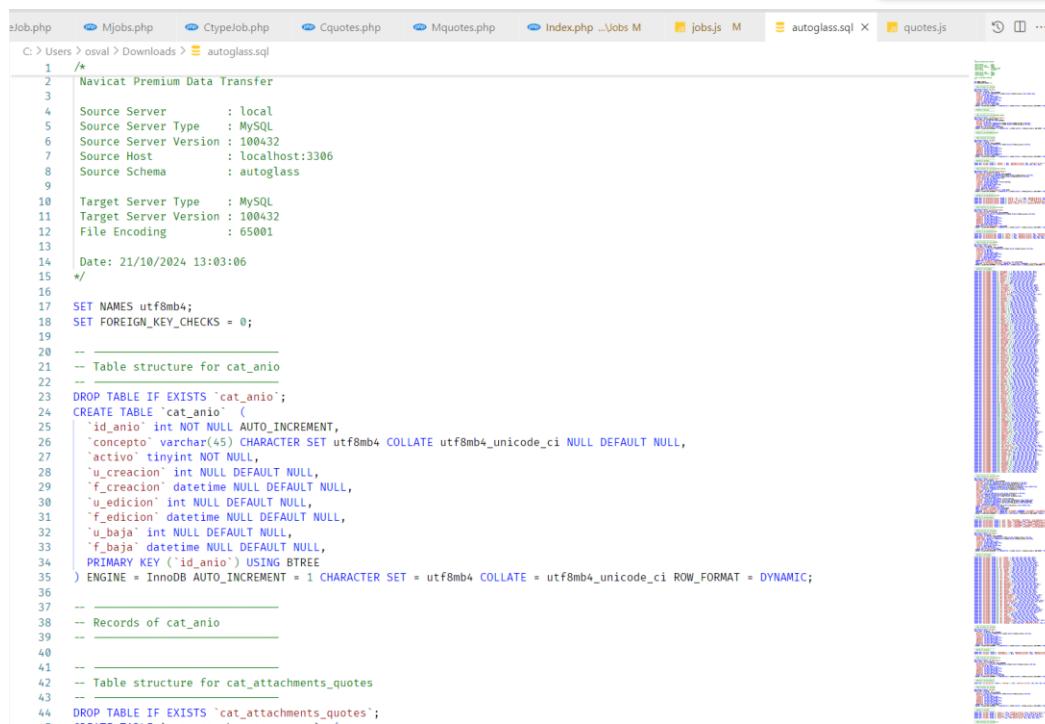


Figura 15 Interfaz de MySQL Workbench más el modelo de proyecto Auto Glass

La creación de la base de datos dentro del ambiente local se tomará de la exportación del modelo hacia un script con las instrucciones necesarias para la creación, dicho script se genera con toda la estructura de todas las tablas y sus relaciones, por lo que se necesitará buscar la tabla deseada, copiar el SQL generado y ejecutar dentro de phpmyAdmin u otro administrador de base de datos.



```

Job.php Mjobs.php Ctypejob.php Quotes.php Mquotes.php Index.php ...Jobs M autoglass.sql quotes.js ...
C: > Users > osval > Downloads > autoglass.sql

1 /*
2 Navicat Premium Data Transfer
3
4 Source Server      : local
5 Source Server Type : MySQL
6 Source Server Version : 100432
7 Source Host        : localhost:3306
8 Source Schema       : autoglass
9
10 Target Server Type : MySQL
11 Target Server Version : 100432
12 File Encoding      : 65001
13
14 Date: 21/10/2024 13:03:06
15 */
16
17 SET NAMES utf8mb4;
18 SET FOREIGN_KEY_CHECKS = 0;
19
20 --
21 -- Table structure for cat_anio
22 --
23 DROP TABLE IF EXISTS `cat_anio`;
24 CREATE TABLE `cat_anio` (
25   `id_anio` int NOT NULL AUTO_INCREMENT,
26   `concepto` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL,
27   `activo` tinyint NOT NULL,
28   `u_creacion` int NULL DEFAULT NULL,
29   `f_creacion` datetime NULL DEFAULT NULL,
30   `u_edicion` int NULL DEFAULT NULL,
31   `f_edicion` datetime NULL DEFAULT NULL,
32   `u_baja` int NULL DEFAULT NULL,
33   `f_baja` datetime NULL DEFAULT NULL,
34   PRIMARY KEY (`id_anio`) USING BTREE
35 ) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8mb4 COLLATE = utf8mb4_unicode_ci ROW_FORMAT = DYNAMIC;
36
37 --
38 -- Records of cat_anio
39 --
40
41 --
42 -- Table structure for cat_attachments_quotes
43 --
44 DROP TABLE IF EXISTS `cat_attachments_quotes`;

```

Figura 16 Script SQL generado desde MySQL Workbench.

Para la ejecución de este script y futuros, dentro del panel de SQL en phpmyadmin, se colocará la tabla a crear de manera local. Después de dar en continuar la tabla se insertará dentro de la base de datos local del proyecto.

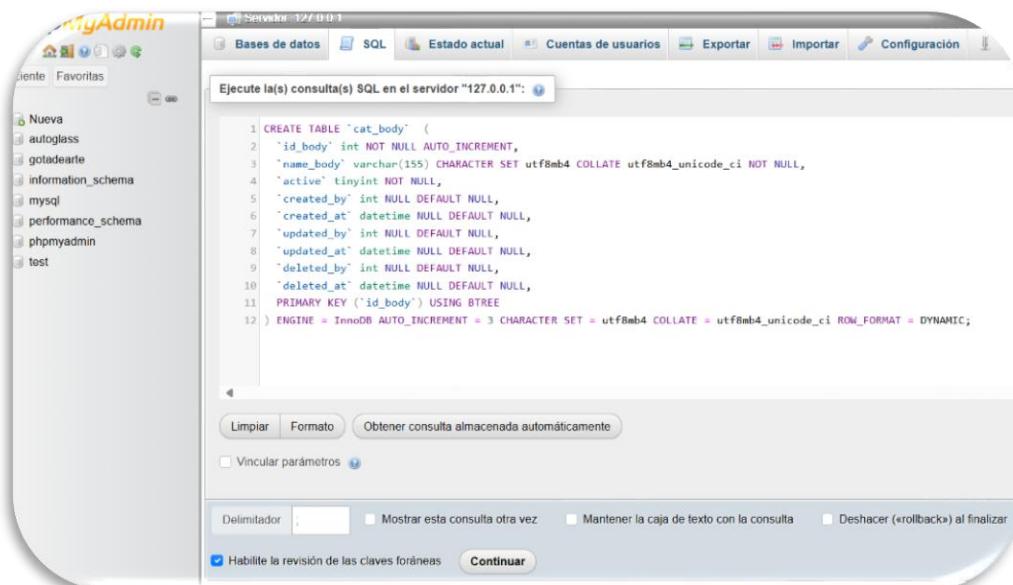


Figura 17 Interfaz SQL de phpMyAdmin para creación de una nueva tabla.

Para el paso 3 creación de una ruta y 4 opción en menú, la creación de la ruta, se retomará la explicación realizada dentro del capítulo III “Menús y Navegación” ya que el desarrollo de cada nueva ruta no requiere del uso de otra sintaxis específica o archivo diferente, lo mismo es para la opción integrada en la barra de navegación, se agregará el código HTML correspondiente para integrar una nueva opción con el uso de `base_url()` y la ruta en específico definida.

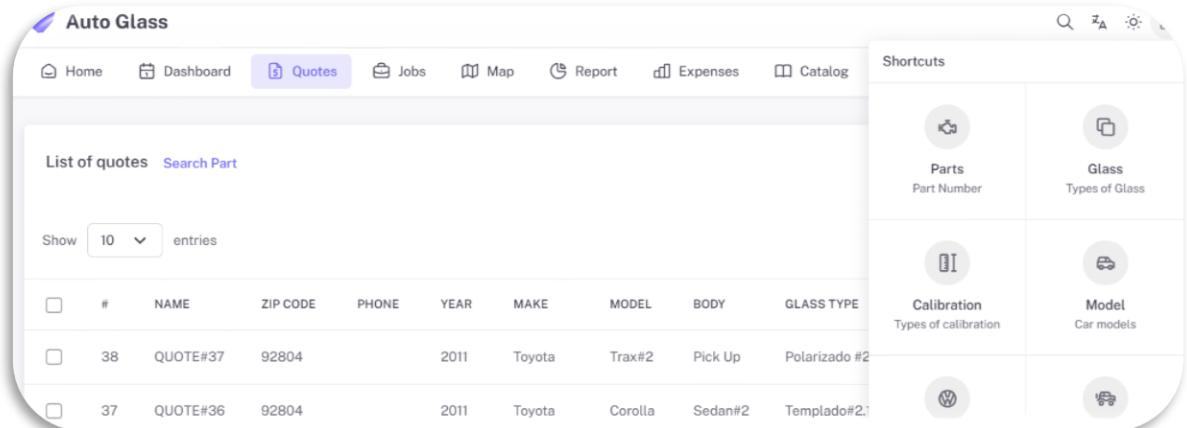


Figura 18 Barra de navegación con las opciones hacia las tablas independientes.

En la creación de un archivo de modelo, el framework de codeigniter ofrece un ordenamiento de archivos, al seguir este estándar, se tiene que, para cada archivo de modelo, se tiene dentro de un fichero independiente, así que una vez con el fichero creado, el nombre del archivo de modelo deberá llevar la siguiente sintaxis “*M (nombre de la tabla en plural).php*”. Con el archivo creado, este por mínimo deberá llevar 9 funciones o métodos que serán indispensables dentro del proceso de lectura, creación, edición, eliminación y cambio de estatus.

Tabla 1 Código correspondiente para obtener todos los datos de una tabla

Función “get_all_data”	
Código	Explicación
<pre>public function get_all_data () {     \$this-&gt;db-&gt;where_in ('active', array (0, 1)).</pre>	Obtiene todos los registros de la tabla cat_make (la cual se debe cambiar dependiendo de los datos de la tabla a obtener) donde la columna active es

<pre>\$query = \$this-&gt;db-&gt;get('cat_make'). return \$query-&gt;result_array (); }</pre>	igual a 0 o 1 y los devuelve como un array asociativo.
---	--

*Tabla 2 Código correspondiente para obtener todos los detalles de un registro*

Función “get_data”	
Código	Explicación
<pre>public function get_data(\$id) {     \$query = \$this-&gt;db-&gt;get_where     ('cat_make', ['id_make' =&gt; \$id]).     return \$query-&gt;row_array (); }</pre>	<p>Recibe un parámetro: La función recibe un parámetro \$id, que se espera sea el ID de un registro en la tabla cat_make.</p> <p>Consulta a la base de datos: Utiliza el método get_where de CodeIgniter para realizar una consulta en la tabla cat_make, buscando un registro donde la columna id_make sea igual al valor de \$id proporcionado.</p> <p>Obtiene el resultado: Al final, la función usa row_array () para devolver una fila de la tabla en forma de arreglo asociativo. Esto significa que devuelve un único registro con sus datos como un arreglo clave-valor, en caso de encontrarlo. Si no hay coincidencias, regresará null.</p>

*Tabla 3 Código correspondiente para insertar datos dentro de una tabla*

Función “create_register”	
Código	Explicación

<pre><code>public function create_register(\$nombre) {     \$data = array (         'Make_name' =&gt; \$nombre,         'active' =&gt; 1, // Por defecto, active         'Created_by' =&gt; \$this-&gt;session-             &gt;userdata('s_id_sesiones'),         'Created_at' =&gt; date ('Y-m-d H: i: s')     );     \$this-&gt;db-&gt;insert ('cat_make', \$data); }</code></pre>	<p>Recibe un parámetro: La función recibe un parámetro \$nombre, que es el nombre que se va a registrar en la columna make_name de la tabla cat_make.</p> <p>Crea un arreglo de datos (\$data): make_name se asigna el valor del parámetro \$nombre.</p> <p>active se establece en 1 (es decir, el nuevo registro estará activo por defecto).</p> <p>created_by toma el valor de la sesión del usuario actualmente activo (s_id_sesiones), lo que indica quién creó el registro.</p> <p>created_at guarda la fecha y hora actual en el formato Y-m-d H: i: s, indicando cuándo se creó el registro.</p> <p>Inserta en la base de datos: Con \$this-&gt;db-&gt;insert ('cat_make', \$data); CodeIgniter realiza una inserción en la tabla cat_make utilizando los datos del arreglo \$data.</p>
---	--

Tabla 4 Código correspondiente para desactivar un registro de una tabla

Función “deactivate_register”	
Código	Explicación
<pre><code>public function deactivate_register(\$id) {     \$data = array (         'active' =&gt; 0,</code></pre>	<p>Recibe un parámetro: La función recibe \$id, que corresponde al ID del registro que se quiere desactivar.</p> <p>Crea un arreglo de datos (\$data):</p>

<pre>'Updated_by' =&gt; \$this-&gt;session-&gt;userdata('s_id_sesiones'), 'updated_at' =&gt; date ('Y-m-d H: i:s') ); \$this-&gt;db-&gt;where ('id_make', \$id); \$this-&gt;db-&gt;update ('cat_make', \$data). return \$this-&gt;db-&gt;affected_rows () &gt; 0. }</pre>	<p>active se establece en 0, lo que significa que el registro quedará desactivado.</p> <p>updated_by guarda el ID del usuario que está realizando la actualización, obteniéndolo de la sesión activa (s_id_sesiones).</p> <p>updated_at guarda la fecha y hora actual (Y-m-d H: i: s), que indica el momento en que se realizó la actualización.</p> <p>Establece la condición (where): Con \$this-&gt;db-&gt;where ('id_make', \$id); se indica que se debe actualizar el registro en la tabla cat_make cuyo id_make coincida con el valor de \$id.</p> <p>Realiza la actualización (update): Con \$this-&gt;db-&gt;update ('cat_make', \$data); se actualiza el registro especificado, asignándole los valores de \$data.</p> <p>Devuelve un valor booleano: La función retorna true si la actualización afectó al menos una fila en la base de datos, lo que indica que el registro fue desactivado exitosamente. Si no se afecta ninguna fila, retorna false.</p>
---	---

Tabla 5 Código correspondiente para activar un registro de una tabla

Función “activate_register”	
Código	Explicación

<pre> public function activate_register(\$id) {     \$data = array (         'active' =&gt; 1,         'Updated_by' =&gt; \$this-&gt;session-             &gt;userdata('s_id_sesiones'),         'updated_at' =&gt; date ('Y-m-d H: i:s')     );      \$this-&gt;db-&gt;where ('id_make', \$id).     \$this-&gt;db-&gt;update ('cat_make', \$data).     return \$this-&gt;db-&gt;affected_rows () &gt; 0. } </pre>	<p>Recibe un parámetro: La función recibe \$id, que es el ID del registro que se quiere activar.</p> <p>Crea un arreglo de datos (\$data): active se establece en 1, lo que indica que el registro quedará activado.</p> <p>updated_by guarda el ID del usuario que realiza la activación, obteniéndolo de la sesión activa (s_id_sesiones).</p> <p>updated_at guarda la fecha y hora actuales (Y-m-d H: i: s), indicando cuándo se hizo la activación.</p> <p>Establece la condición (where): Con \$this-&gt;db-&gt;where ('id_make', \$id); se especifica que la actualización debe aplicarse solo al registro en cat_make cuyo id_make coincida con \$id.</p> <p>Realiza la actualización (update): Con \$this-&gt;db-&gt;update ('cat_make', \$data); se actualiza el registro, asignándole los valores en \$data.</p> <p>Devuelve un valor booleano: La función retorna true si la actualización afectó al menos una fila en la base de datos (lo que indica que el registro fue activado exitosamente) o false si no se afectó ninguna fila.</p>
--	--

Tabla 6 Código correspondiente para actualizar datos de un registro

Función “update_register”	
Código	Explicación
<pre>public function update_register (\$id,                                \$data) {     \$this-&gt;db-&gt;where ('id_make', \$id).     return \$this-&gt;db-&gt;update ('cat_make',                                \$data). }</pre>	<p>Recibe parámetros:</p> <p>\$id: el ID del registro que se desea actualizar.</p> <p>\$data: un arreglo asociativo que contiene los campos y sus nuevos valores que se deben actualizar en la base de datos.</p> <p>Establece la condición (where): Con \$this-&gt;db-&gt;where ('id_make', \$id); se especifica que la actualización debe aplicarse solo al registro en cat_make cuyo id_make coincida con \$id.</p> <p>Realiza la actualización (update): Con \$this-&gt;db-&gt;update ('cat_make', \$data); se ejecuta la actualización del registro con los datos proporcionados en \$data.</p> <p>Devuelve un valor booleano: La función retorna true si la actualización fue exitosa y false si ocurrió un error.</p>

Tabla 7 Código correspondiente para eliminar un registro de una tabla

Función “delete_register”	
Código	Explicación
<pre>public function delete_register(\$id) {</pre>	Asegura que el ID sea un entero: Con intval(\$id), convierte el valor de \$id a un número entero para evitar errores o inyecciones de SQL.

<pre> // Asegurarse de que el ID es un número         entero         \$id = intval(\$id);  // Verificar si el registro existe antes de         intentar eliminarlo         \$this-&gt;db-&gt;where ('id_make', \$id);         \$query = \$this-&gt;db-&gt;get('cat_make');          if (\$query-&gt;num_rows () &gt; 0)             \$data = array (                 'active' =&gt; 2,                 'Deleted_by' =&gt; \$this-&gt;session-                     &gt;userdata('s_id_sesiones'),                 'deleted_at' =&gt; date ('Y-m-d H:i:')             );          \$this-&gt;db-&gt;where ('id_make', \$id);         return \$this-&gt;db-&gt;update ('cat_make',             \$data);          // Realizar la eliminación         // \$this-&gt;db-&gt;where ('id_make', \$id).         // return \$this-&gt;db-&gt;delete('cat_make').         } else {             return false;         }     } </pre>	<p>Verifica la existencia del registro: Usa \$this-&gt;db-&gt;where ('id_make', \$id); y \$this-&gt;db-&gt;get('cat_make'); para obtener el registro con el id_make proporcionado. Verifica si existe comprobando que el número de filas (num_rows ()) sea mayor a 0. Si el registro no existe, devuelve false.</p> <p>Marca el registro como eliminado: Si el registro existe, crea un arreglo \$data que actualiza los siguientes campos: active se establece en 2, lo que indica que el registro está "eliminado" (una convención para inhabilitarlo sin borrarlo físicamente). deleted_by guarda el ID del usuario que realiza la "eliminación", obtenido de la sesión activa (s_id_sesiones). deleted_at guarda la fecha y hora actual (Y-m-d H: i: s) para registrar el momento de la eliminación.</p> <p>Realiza la actualización: Con \$this-&gt;db-&gt; where ('id_make', \$id); y \$this-&gt;db-&gt; update ('cat_make', \$data); se actualiza el registro en cat_make con los valores en \$data, y la función devuelve true si fue exitosa.</p> <p>Eliminación física comentada: En el código, hay una eliminación física</p>
--	--

	comentada (\$this->db->delete('cat_make') ;), lo que indica que originalmente podría haberse planeado borrar el registro físicamente, pero ahora solo se inactiva.
--	---

*Tabla 8 Código correspondiente para actualizar el estatus de varios registros de una tabla*

Función “update_status_products”	
Código	Explicación
<pre>public function update_status_products     (\$ids, \$status)     {         \$this-&gt;db-&gt;where_in ('id_make', \$ids);         \$this-&gt;db-&gt;update ('cat_make', ['active' =&gt; \$status]);     } }</pre>	<p>Recibe parámetros:</p> <p>\$ids: una lista (arreglo) de IDs (id_make) de los registros que se desea actualizar.</p> <p>\$status: el nuevo valor que se quiere asignar a la columna active para estos registros.</p> <p>Filtrá los registros (where_in): Usa \$this-&gt;db-&gt;where_in ('id_make', \$ids); para aplicar la actualización solo a los registros cuyo id_make esté dentro de la lista \$ids.</p> <p>Realiza la actualización: Con \$this-&gt;db-&gt;update ('cat_make', ['active' =&gt; \$status]); actualiza el valor de la columna active a \$status para todos los registros filtrados.</p>

*Tabla 9 Código correspondiente para eliminar varios registros de una tabla*

Función “delete_registers”	
Código	Explicación

<pre><code>public function delete_registers(\$ids) {     \$data = array (         'active' =&gt; 2,         'Deleted_by' =&gt; \$this-&gt;session-             &gt;userdata('s_id_sesiones'),         'deleted_at' =&gt; date ('Y-m-d H:i:')     );      \$this-&gt;db-&gt;where_in ('id_make', \$ids); //  <i>Asegúrate de usar where_in para-</i>  <i>arrays.</i>      \$this-&gt;db-&gt;update ('cat_make', \$data). }</code></pre>	<p>La función delete_registers prepara un conjunto de datos (\$data) que marca los registros como "eliminados" de manera lógica, estableciendo el estado active en 2. También incluye información sobre el usuario que realiza la eliminación (deleted_by) y la fecha y hora de esta acción (deleted_at)</p>
--	--

El archivo de controlador contendrá métodos con relación al uso de los que contiene, ya que, este archivo será el intermediario entre los archivos de la vista y el modelo, realizando subprocessos como cálculo de tax de ciertos valores obtenidos de la base de datos, cálculos de fecha y hora para enviar a la vista o actualizar a la base de datos. Los métodos que contendrá un módulo independiente son alrededor de 11 diferentes.

Tabla 10 Código correspondiente para inicializar modelos

Función “__construct”	
Código	Explicación
<pre><code>public function __construct () {     parent::__construct ();     \$this-&gt;load- &gt;model('Ntsdata/MakeCar/MakeCar');</code></pre>	<p>Inicializa el controlador y carga el modelo MakeCar para poder usarlo en las demás funciones del controlador.</p>

}	
---	--

Tabla 11 Código correspondiente para la carga inicial de la vista

Función “index”	
Código	Explicación
<pre>public function index () {     \$data['title'] = "Make Car.";      \$this-&gt;load-&gt;view     ('Ntsdata/Administrador/Layout/Header',      \$data).      \$this-&gt;load- &gt;view('Ntsdata/Administrador/Layout/Menu').     \$this-&gt;load-&gt;view ('Ntsdata/MakeCar/Index',      \$data).      \$this-&gt;load- &gt;view('Ntsdata/Administrador/Layout/Footer'). }</pre>	Carga la vista principal Index de la sección MakeCar, incluyendo encabezado, menú y pie de página. Establece el título "Make Car".

Tabla 12 Código correspondiente para la obtención de todos los datos de una tabla

Función “get_all_data”	
Código	Explicación
<pre>public function get_all_data () {     \$model_cars = \$this-&gt;MakeCar-         &gt;get_all_data ().      echo json_encode (array ('data' =&gt;         \$model_cars));</pre>	Obtiene todos los registros de MakeCar y los envía en formato JSON.

}	
---	--

Tabla 13 Código correspondiente para la obtención de datos de un registro en específico

Función “get_data”	
Código	Explicación
<pre>public function get_data(\$id) {     \$registro = \$this-&gt;MakeCar-         &gt;get_data(\$id);     echo json_encode(\$registro); }</pre>	Obtiene los detalles de un registro específico en MakeCar por su ID y los envía en formato JSON.

Tabla 14 Código correspondiente para la creación de un nuevo registro

Función “create_register”	
Código	Explicación
<pre>public function create_register () {     // Definir las reglas de validación     \$this-&gt;form_validation-&gt;set_rules         ('nombre', 'Nombre', 'required');      if (\$this-&gt;form_validation-&gt;run () ===         FALSE) {         // Devolver errores de validación         echo json_encode (array ('status' =&gt;             'error', 'errors' =&gt; validation_errors ()));     } else {         \$nombre = \$this-&gt;input-&gt;post('nombre');     } }</pre>	Valida y crea un nuevo registro en MakeCar. Si hay errores de validación, los devuelve en JSON; si se crea exitosamente, devuelve un estado de éxito.

<pre> \$this-&gt;MakeCar- &gt;create_register(\$nombre). echo json_encode (array ('status' =&gt; 'success')).</pre>	
---	--

Tabla 15 Código correspondiente para desactivar el estatus de un registro

Función “ <i>deactivate_register</i> ”	
Código	Explicación
<pre> public function deactivate_register(\$id) {     if (\$this-&gt;input-&gt;method () === 'post')         \$result = \$this-&gt;MakeCar-         &gt;deactivate_register(\$id).          if (\$result)             echo json_encode (array ('status' =&gt; 'success')).</pre> <p style="margin-left: 40px;">} else {</p> <p style="margin-left: 40px;">echo json_encode (array ('status' =&gt; 'error', 'message' =&gt; 'Failed to deactivate glass')).</p> <p style="margin-left: 40px;">}</p> <p style="margin-left: 40px;">} else {</p> <p style="margin-left: 40px;">show_error ('Forbidden', 403);</p> <p style="margin-left: 40px;">}</p> <p style="margin-left: 40px;">}</p>	<p>Cambia el estado de un registro específico a inactivo (desactiva), solo si el método HTTP es POST. Devuelve un mensaje de éxito o error en JSON.</p>

Tabla 16 Código correspondiente para activa el estatus de un registro

Función “activate_register”	
Código	Explicación
<pre>public function activate_register(\$id) {     if (\$this-&gt;input-&gt;method () === 'post')         \$result = \$this-&gt;MakeCar-&gt;activate_register(\$id);      if (\$result)         echo json_encode (array ('status' =&gt;             'success'));     } else {         echo json_encode (array ('status' =&gt;             'error', 'message' =&gt; 'Failed to activate             glass'));     } } else {     show_error ('Forbidden', 403); } }</pre>	Activa un registro específico (cambia el estado a activo), también solo si el método HTTP es POST. Devuelve un estado de éxito o error en JSON.

Tabla 17 Código correspondiente para actualizar los datos de un registro existente

Función “update_register”	
Código	Explicación
<pre>public function update_register(\$id) {     // Obtener los datos enviados desde el     formulario }</pre>	Actualiza los datos de un registro específico en MakeCar, incluyendo quién y cuándo se realizó la actualización.

<pre> \$nombre = \$this-&gt;input-&gt;post('nombre');  <i>// Preparar los datos para la actualización</i>  \$data = array (     'Name_make' =&gt; \$nombre,     'Updated_by' =&gt; \$this-&gt;session-&gt;userdata('s_id_sesiones'),     'updated_at' =&gt; date ('Y-m-d H:i:s') );  <i>// Intentar actualizar el registro en la base de datos</i>  if (\$this-&gt;MakeCar-&gt;update_register     (\$id, \$data))  \$response = array ('status' =&gt; 'success', 'message' =&gt; 'Registration successfully updated.');  } else {  \$response = array ('status' =&gt; 'error', 'message' =&gt; 'Error updating.');  }  <i>// Devolver la respuesta como JSON</i>  echo json_encode(\$response); } </pre>	<p>Devuelve un mensaje de éxito o error en JSON.</p>
---	--

Tabla 18 Código correspondiente para la eliminación de un registro

Función “ <i>delete_register</i> ”
------------------------------------

Código	Explicación
<pre> public function delete_register(\$id) {     \$response = array ('status' =&gt; false);      // Verificar si el ID es válido     if (\$id) {          // Llamar al método del modelo para         // eliminar el registro         if (\$this-&gt;MakeCar-&gt;delete_register(\$id))             \$response['status'] = true;         \$response['message'] = 'Record deleted                                 successfully';          } else {             \$response['message'] = 'Could not                                 delete record.';          }          } else {             \$response['message'] = 'ID del cristal                                 inválido.';          }      // Enviar la respuesta en formato JSON     echo json_encode(\$response); } </pre>	Marca un registro específico como "eliminado" en MakeCar, verificando primero la validez del ID. Devuelve un mensaje de éxito o error en JSON.

Tabla 19 Código correspondiente para la desactivación del estatus de varios registros

Función “deactivate_registers”	
Código	Explicación

<pre><code>public function deactivate_registers () {     \$ids = \$this-&gt;input-&gt;post('ids').         if (! empty(\$ids))             \$this-&gt;MakeCar- &gt;update_status_products (\$ids, 0).         echo json_encode (['status' =&gt;             'success']).     } else { echo json_encode (['status' =&gt; 'error',     'message' =&gt; 'No IDs provided']).     } }</code></pre>	<p>Desactiva múltiples registros en MakeCar de acuerdo con los IDs enviados, devolviendo un estado de éxito o error en JSON.</p>
--	--

Tabla 20 Código correspondiente para la activación de varios registros

Función “activate_registers”	
Código	Explicación
<pre><code>public function activate_registers () {     \$ids = \$this-&gt;input-&gt;post('ids').         if (! empty(\$ids))             \$this-&gt;MakeCar- &gt;update_status_products (\$ids, 1).         echo json_encode (['status' =&gt;             'success']).     } else { echo json_encode (['status' =&gt; 'error',     'message' =&gt; 'No IDs provided']).     } }</code></pre>	<p>Activa múltiples registros en MakeCar usando los IDs proporcionados, y devuelve el estado en JSON.</p>

}	
---	--

Tabla 21 Código correspondiente para la eliminación de varios registros

Función “ <i>delete_registers</i> ”	
Código	Explicación
<pre>public function delete_registers () {     {         \$ids = \$this-&gt;input-&gt;post('ids').             if (! empty(\$ids))                 if (is_array(\$ids))                     \$this-&gt;MakeCar-&gt;delete_registers(\$ids).                         echo json_encode (['status' =&gt;                             'success']).                             } else {                     echo json_encode (['status' =&gt; 'error',                         'message' =&gt; 'IDs is not an array']).                             }                             } else {                     echo json_encode (['status' =&gt; 'error',                         'message' =&gt; 'No IDs provided']).                             }                             }</pre>	Marca múltiples registros como "eliminados" en MakeCar usando una lista de IDs. Verifica que se reciba un arreglo y devuelve un estado de éxito o error en JSON.

Para la comunicación y solicitud de los métodos mencionados anteriormente, existe un archivo de javascript, el cual por medio de Ajax realiza las peticiones necesarias para la carga de un datatable, las acciones de un CRUD junto con peticiones extra como recarga interna de datos y envío en tiempo real. Este archivo supone un grado de conocimiento de Ajax y JavaScript debido a que se involucran alrededor de 13 funciones para el manejo de las múltiples peticiones, una explicación breve de cada una es la siguiente tabla.

Tabla 22 Funciones correspondientes para el funcionamiento de un módulo independiente

Archivo “make-car.js”	
Funciones	Resumen
dt_select_table.on('click', '.view-details', function ()	Despliegue del modal para mostrar detalles después de haber realizado una petición por Ajax para la obtención de datos del registro específico.
dt_select_table.on('click', '.deactivate-make-car', function ()	Despliegue de modal de alerta para la confirmación de desactivación de un registro, al confirmar la acción se ejecuta el cambio de estatus por medio de una petición de Ajax.
dt_select_table.on('click', '.activate-make-car', function ()	Ejecución de petición Ajax para el cambio de estatus del registro a un “activo”.
dt_select_table.on('click', '.item-edit', function ()	Petición Ajax para la obtención de datos de un registro para mostrar dentro del modal de editar registro.
\$(saveEditRecord).on('click', function ()	Obtención de datos del formulario dentro del modal para enviarlos por medio de Ajax al método de update del controlador.
\$(saveNewRecord).on('click', function ()	Envío de datos dentro de un modal de nuevo registro por Ajax hacia el método de creación de nuevo registro en el controlador.
dt_select_table.on('click', '.delete-record', function ()	Abre un modal de confirmación para eliminar un registro por medio de Ajax.

<code>\$(deactivateSelected'). on ('click', function ()</code>	Proceso para el cambio de estatus a un desactivado para un conjunto de diferentes registros seleccionados en el datatable.
<code>\$(activateSelected'). on ('click', function ()</code>	Proceso para el cambio de estatus a un activo para un conjunto de diferentes registros seleccionados en el datatable.
<code>\$(deleteSelected'). on ('click', function ()</code>	Proceso para el cambio de estatus a un eliminado para un conjunto de diferentes registros seleccionados en el datatable.

Una vez con los métodos del javascript definidos, la vista se tendrá solamente la sección para definir el cuerpo de la tabla, así como tres modals, uno para la creación, otro para la edición y un último para ver los detalles de un registro.

The screenshot shows the 'List of car make' page within the Auto Glass application. The page has a header with navigation links: Home, Dashboard, Quotes, Jobs, Map, Report, Expenses, Catalog, Settings, and a search bar. Below the header is a table titled 'List of car make'. The table has columns: '#', 'MAKE NAME', 'STATUS', and 'ACTIONS'. The data in the table is:

#	MAKE NAME	STATUS	ACTIONS
5	Tesla	Activate	⋮ ⚡
4	Volkswagen	Activate	⋮ ⚡
3	Nissan	Activate	⋮ ⚡
2	Chevrolet	Activate	⋮ ⚡
1	Toyota	Activate	⋮ ⚡

At the bottom of the table, it says 'Showing 1 to 5 of 5 entries'. There are also 'Previous' and 'Next' buttons at the bottom right.

Figura 19 Vista del panel principal para el datatable

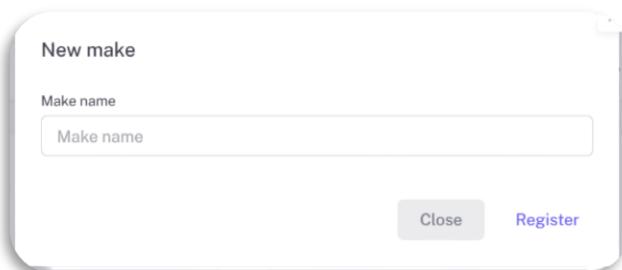


Figura 20 Vista del modal para la creación de un nuevo registro



Figura 21 Vista del modal para ver los detalles de un registro

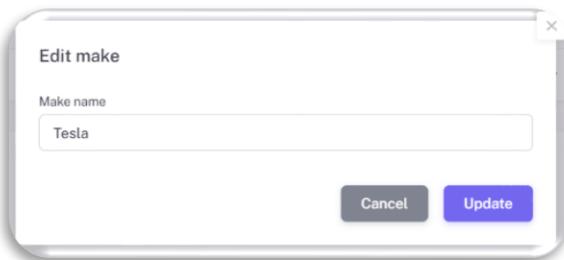


Figura 22 Vista del modal para editar un registro existente

Este proceso deberá ser aplicado en todos los módulos complementarios para que funcione uno más grande.

Una vez teniendo los módulos independientes necesarios para el funcionamiento, se creará el módulo correspondiente para cotizaciones como un módulo independiente, sin embargo, para la base de datos de realizaran las relaciones necesarias con las otras tablas, quedando un script de la siguiente manera.

Auto Glass – NTS DATA

## Capítulo IV – Implementación

```
TypeJob.php CQuotes.php Mquotes.php Index.php \Jobs M jobs.js M autoglass.sql X MakeCar.php Cmake.php M make-car.js quotes.js
C: > Users > oval > Downloads > autoglass.sql
937 CREATE TABLE `ent_quotes` (
938     `id_quotes` int NOT NULL AUTO_INCREMENT,
939     `quote_name` varchar(155) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL,
940     `active` tinyint NOT NULL,
941     `created_by` int NULL DEFAULT NULL,
942     `created_at` datetime NULL DEFAULT NULL,
943     `updated_by` int NULL DEFAULT NULL,
944     `updated_at` datetime NULL DEFAULT NULL,
945     `deleted_by` int NULL DEFAULT NULL,
946     `deleted_at` datetime NULL DEFAULT NULL,
947     `phone` int NULL DEFAULT NULL,
948     `vin` int NULL DEFAULT NULL,
949     `glass_cost` double NULL DEFAULT NULL,
950     `subtotal` double NULL DEFAULT NULL,
951     `tax` int NULL DEFAULT NULL,
952     `total` double NULL DEFAULT NULL,
953     `first_name` varchar(145) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL,
954     `last_name` varchar(145) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL,
955     `phone_2` int NULL DEFAULT NULL,
956     `alternative_phone` int NULL DEFAULT NULL,
957     `email` varchar(95) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL,
958     `address` varchar(145) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL,
959     `insuance_carrier` varchar(85) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL,
960     `policy_id` int NULL DEFAULT NULL,
961     `refecal` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL,
962     `policy_holder` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL,
963     `policy_address` varchar(145) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL,
964     `deductible` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL,
965     `nags_description` tinytext CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL,
966     `nags_labior` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL,
967     `list_price` varchar(45) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL,
968     `price_per_hour` double NULL DEFAULT NULL,
969     `id_make` int NOT NULL,
970     `id_model_car` int NOT NULL,
971     `id_body` int NOT NULL,
972     `id_glass` int NOT NULL,
973     `id_part_number` int NOT NULL,
974     `id_price_tier` int NOT NULL,
975     `id_calibration_type` int NOT NULL,
976     `id_ciudades` int NOT NULL,
977     `id_estados` int NOT NULL,
978     `id_calibration_location` int NOT NULL,
979     `id_year` int NOT NULL,
980     `others` int NULL DEFAULT NULL,
981 
```

Figura 23 Vista del script generado por MySQL Workbench para la tabla de ent\_quotes

C:\> Users\osval> Downloads\ autoglass.sql

```
937 CREATE TABLE `ent_quotes` (
 970   `id_model_car` int NOT NULL,
 971   `id_body` int NOT NULL,
 972   `id_glass` int NOT NULL,
 973   `id_part_number` int NOT NULL,
 974   `id_price_tier` int NOT NULL,
 975   `id_calibration_type` int NOT NULL,
 976   `id_ciudades` int NOT NULL,
 977   `id_estados` int NOT NULL,
 978   `id_calibration_location` int NOT NULL,
 979   `id_year` int NOT NULL,
 980   `others` int NULL DEFAULT NULL,
 981   `zip_code_2` int NULL DEFAULT NULL,
 982   `id_zip_code` int NULL DEFAULT NULL,
 983   `notes` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL,
 984   `id_tag` int NULL DEFAULT NULL,
 985   PRIMARY KEY (`id_quotes`) USING BTREE,
 986   INDEX `fk_ent_quotes_cat_year1`(`id_year` ASC) USING BTREE,
 987   INDEX `fk_ent_quotes_cat_make1`(`id_make` ASC) USING BTREE,
 988   INDEX `fk_ent_quotes_cat_model_car1`(`id_model_car` ASC) USING BTREE,
 989   INDEX `fk_ent_quotes_cat_body1`(`id_body` ASC) USING BTREE,
 990   INDEX `fk_ent_quotes_cat_glass1`(`id_glass` ASC) USING BTREE,
 991   INDEX `fk_ent_quotes_cat_part_number1`(`id_part_number` ASC) USING BTREE,
 992   INDEX `fk_ent_quotes_cat_price_tier1`(`id_price_tier` ASC) USING BTREE,
 993   INDEX `fk_ent_quotes_cat_calibration_type1`(`id_calibration_type` ASC) USING BTREE,
 994   INDEX `fk_ent_quotes_cat_ciudades1`(`id_ciudades` ASC) USING BTREE,
 995   INDEX `fk_ent_quotes_cat_estados1`(`id_estados` ASC) USING BTREE,
 996   INDEX `fk_ent_quotes_cat_calibration_location1`(`id_calibration_location` ASC) USING BTREE,
 997   INDEX `fk_ent_quotes_cat_zip_code`(`id_zip_code` ASC) USING BTREE,
 998   INDEX `fk_ent_quotes_cat_tags`(`id_tag` ASC) USING BTREE,
 999   CONSTRAINT `fk_ent_quotes_cat_body1` FOREIGN KEY (`id_body`) REFERENCES `cat_body`(`id_body`) ON DELETE NO ACTION ON UPDATE NO ACTION,
1000   CONSTRAINT `fk_ent_quotes_cat_calibration_location1` FOREIGN KEY (`id_calibration_location`) REFERENCES `cat_calibration_location`(`id_calibration_location`) ON DELETE NO ACTION ON UPDATE NO ACTION,
1001   CONSTRAINT `fk_ent_quotes_cat_calibration_type1` FOREIGN KEY (`id_calibration_type`) REFERENCES `cat_calibration_type`(`id_calibration_type`) ON DELETE NO ACTION ON UPDATE NO ACTION,
1002   CONSTRAINT `fk_ent_quotes_cat_ciudades1` FOREIGN KEY (`id_ciudades`) REFERENCES `cat_ciudades`(`id_ciudades`) ON DELETE NO ACTION ON UPDATE NO ACTION,
1003   CONSTRAINT `fk_ent_quotes_cat_estados1` FOREIGN KEY (`id_estados`) REFERENCES `cat_estados`(`id_estados`) ON DELETE NO ACTION ON UPDATE NO ACTION,
1004   CONSTRAINT `fk_ent_quotes_cat_glass1` FOREIGN KEY (`id_glass`) REFERENCES `cat_glass`(`id_glass`) ON DELETE NO ACTION ON UPDATE NO ACTION,
1005   CONSTRAINT `fk_ent_quotes_cat_make1` FOREIGN KEY (`id_make`) REFERENCES `cat_make`(`id_make`) ON DELETE NO ACTION ON UPDATE NO ACTION,
1006   CONSTRAINT `fk_ent_quotes_cat_model_car1` FOREIGN KEY (`id_model_car`) REFERENCES `cat_model_car`(`id_model_car`) ON DELETE NO ACTION ON UPDATE NO ACTION,
1007   CONSTRAINT `fk_ent_quotes_cat_part_number1` FOREIGN KEY (`id_part_number`) REFERENCES `cat_part_number`(`id_part_number`) ON DELETE NO ACTION ON UPDATE NO ACTION,
1008   CONSTRAINT `fk_ent_quotes_cat_price_tier1` FOREIGN KEY (`id_price_tier`) REFERENCES `cat_price_tier`(`id_price_tier`) ON DELETE NO ACTION ON UPDATE NO ACTION,
1009   CONSTRAINT `fk_ent_quotes_cat_tags1` FOREIGN KEY (`id_tag`) REFERENCES `cat_tags`(`id_tag`) ON DELETE NO ACTION ON UPDATE NO ACTION,
1010   CONSTRAINT `fk_ent_quotes_cat_zip_code` FOREIGN KEY (`id_zip_code`) REFERENCES `cat_zip_code`(`id_zip_code`) ON DELETE NO ACTION ON UPDATE NO ACTION
1011 ) ENGINE = InnoDB AUTO_INCREMENT = 27 CHARACTER SET = utf8mb4 COLLATE = utf8mb4_unicode_ci ROW_FORMAT = DYNAMIC;
```

*Figura 24 Vista del script ent\_quotes junto con las relaciones con las tablas complementarias*

Teniendo como punto de partida el módulo con las funciones generales, se realizarán los cambios en el siguiente orden:

- ❖ Reajuste de los 3 modals para que se encuentren bajo la vista de pestañas y con los campos necesarios correspondientes a una cotización.
- ❖ Funcionamiento de carga de archivo con el estilo “arrastrar y soltar” para subirlos o eliminarlos dentro del modal de creación y edición.
- ❖ Calculo dentro de los modals de creación y edición para el cambio del valor subtotal dependiendo la selección de los parámetros definidos en el análisis.
- ❖ Vista de los archivos para el modal de detalles de una cotización.
- ❖ Añadir las columnas definidas a mostrar dentro del datatable de cotizaciones.
- ❖ Creación de un nuevo modal para la búsqueda de un numero de parte y la creación de una nueva cotización o trabajo.
- ❖ Apartado de carga de archivos desde un modal de búsqueda de numero de parte.
- ❖ Nuevo modal para crear un enlace de acceso hacia una vista de detalles de una cotización creada.
- ❖ Validaciones dentro de los campos seleccionables los cuales estén relacionados con otras tablas para evitar errores dentro del sistema.

Figura 25 Vista de un modal de nueva cotización

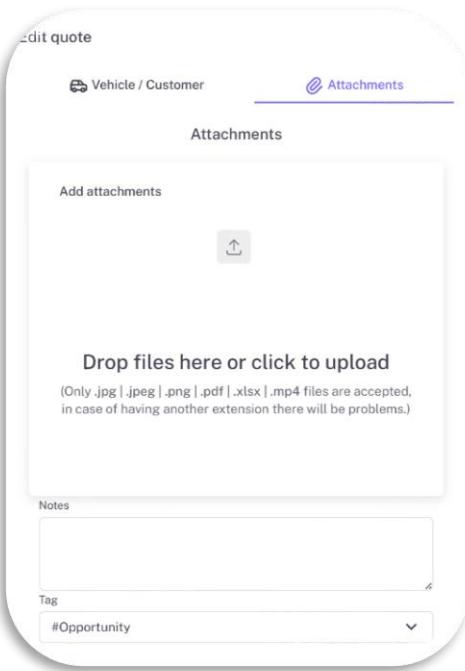


Figura 26 Vista de un modal de edición de una cotización creada

Los campos nuevos para que correspondan una cotización se agregan sin mayor dificultad, debido a que solo hay que seguir retomando el formato de envío que tiene un modal de un módulo independiente. La nueva funcionalidad para envío de archivos, se usará la biblioteca de dropzone, ya que este nos ofrece una integración para subir archivos arrastrando y soltando.

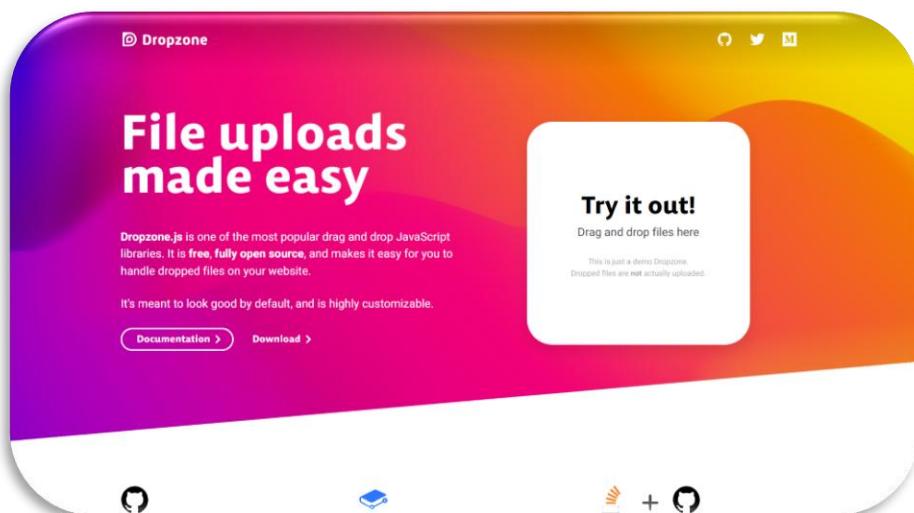


Figura 27 Pagina web de la biblioteca dropzone

Para el uso de este recurso, se deberá trabajar bajo la documentación que esta nos ofrece, para un uso correcto, se deberá tener dentro del mismo archivo javascript utilizado para la parte de peticiones Ajax del módulo, además deberá de definirse de forma diferente por cada apartado dentro de un modal diferente, su funcionamiento general es el siguiente:

Tabla 23 Explicación del uso de la librería de dropzone para cotizaciones

Dropzone para guardar archivo en modal nueva cotización	
Código	Explicación
<pre>if (dropzoneMultiNewQoute) {     const myDropzoneMultiNewQuote = new Dropzone (dropzoneMultiNewQoute,     previewTemplate: previewTemplateQuote,     parallelUploads: 1,     maxFilesize: 5,     addRemoveLinks: true,     acceptedFiles: '.gif,.jpg,.png,.pdf,.jpeg,.xlsx,.mp4,.mov' );     // Manejar subida exitosa en #dropzone-multi- new-qoute:     myDropzoneMultiNewQuote.on("success", function (file, response)     try {         const responseObj = JSON.parse(response).             tempFiles.push(name: file.name, path: responseObj.temp_file_path,             file_size: responseObj.file_size)).         \$("temp_files"). val (JSON.stringify(tempFiles));     } }</pre>	<p>Configuración de Dropzone:</p> <p>Crea una instancia de Dropzone asociada al elemento dropzoneMultiNewQoute.</p> <p>Configura parámetros como:</p> <p>previewTemplate: usa una plantilla HTML (previewTemplateQuote) para mostrar las vistas previas de archivos.</p> <p>parallelUploads: permite solo una subida simultánea.</p> <p>maxFilesize: limita el tamaño de los archivos a 5 MB.</p>

<pre> //console.log ("Archivos temporales:", tempFiles); } catch (error) {     console. error ("Error procesando la resuesta de Dropzone:", error); } };  // Manejar eliminación de archivo en #dropzone- multi-new-qoute: myDropzoneMultiNewQuote.on("removedfile", function (file) { //console.log ("Archivo eliminado:", file.name); let fileToRemove = tempFiles.find(f =&gt; f.name ==== file.name);  if (fileToRemove) let index = tempFiles.indexOf(fileToRemove); if (index &gt; -1) tempFiles.splice(index, 1); console. warn ("tempFiles updated:", tempFiles); \$("#temp_files"). val (JSON.stringify(tempFiles));  \$. ajax ({ url: base_url + 'Ntsdata/Quotes/Cquotes/delete_temp_file', // Ruta a tu controlador type: 'POST', </pre>	<p>addRemoveLinks: agrega un enlace para eliminar archivos.</p> <p>acceptedFiles: especifica tipos de archivos permitidos.</p> <p>Manejador de subida exitosa:</p> <p>Cuando un archivo se sube exitosamente (on("success")), el manejador intenta analizar la respuesta JSON enviada por el servidor.</p> <p>Si la respuesta contiene datos válidos, agrega un objeto al arreglo tempFiles con el nombre, la ruta temporal y el tamaño del archivo subido.</p> <p>Luego, actualiza un campo oculto (#temp_files) con el contenido de tempFiles en formato JSON para enviar todos los archivos</p>
---	--

<pre>         data: {temp_file_path: fileToRemove.path}, // Enviar la ruta del archivo a eliminar success: function (response) { //console.log ("Archivo eliminado en el servidor:", response); }, error: function (xhr, status, error) { //console.error ("Error al eliminar el archivo en el servidor:", error); } }); } } else { console.warn ("No se pudo encontrar el archivo en tempFiles:", file.name); } }); } </pre>	temporales en la próxima petición. Manejador de eliminación de archivos: Cuando se elimina un archivo de Dropzone (on("removedfile")), busca el archivo en tempFiles y lo quita del arreglo. Después de actualizar tempFiles y el campo oculto (#temp_files), realiza una petición AJAX POST a delete_temp_file en el controlador, enviando la ruta del archivo temporal para eliminarlo del servidor. Si el archivo no se encuentra en tempFiles, muestra una advertencia en la consola.
---	---

En el código de la tabla 4.43 es con relación a la configuración de la zona de arrastrar y soltar para una nueva cotización, esta estructura es bastante similar con el desarrollo de las 3 zonas faltantes, para la edición, detalles y búsqueda de parte solo se especifican los detalles.

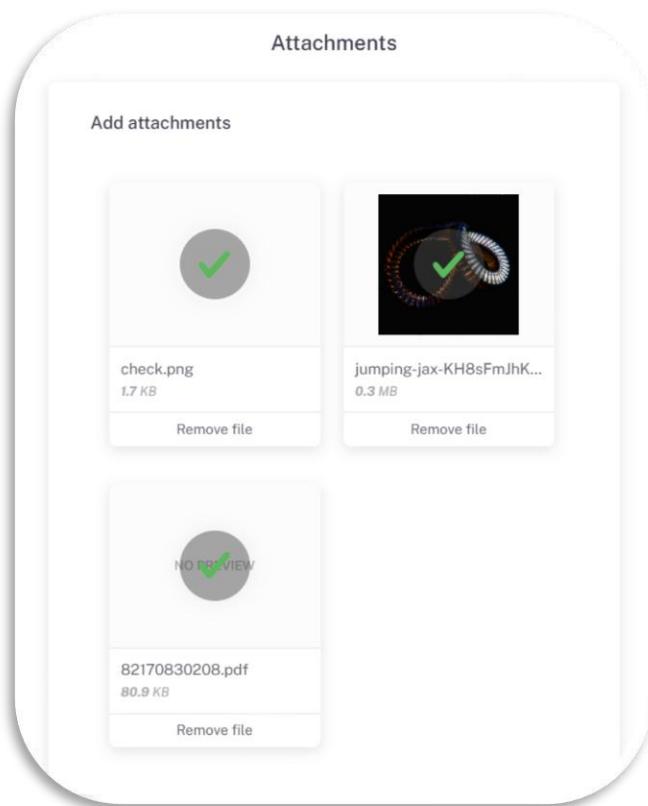


Figura 28 Vista de dropzone con archivos subidos para una cotización nueva

Para el modal de búsqueda por parte y creación de una cotización o trabajo se retomarán los elementos creados dentro del modal de creación de una nueva cotización para solo recuperar los elementos sugeridos en el análisis. Este modal contara de igual manera con una sección para subir archivos adjuntos, por el momento solo se dejará con la funcionalidad de crear una cotización mientras que para el próximo capítulo se trabajara el desarrollo para un trabajo asignado a un empleado. Los elementos propuestos dentro del análisis dan a notar que todos los campos necesarios será campos seleccionables por lo que estos incorporaran una búsqueda hacia los registros de los módulos independientes.

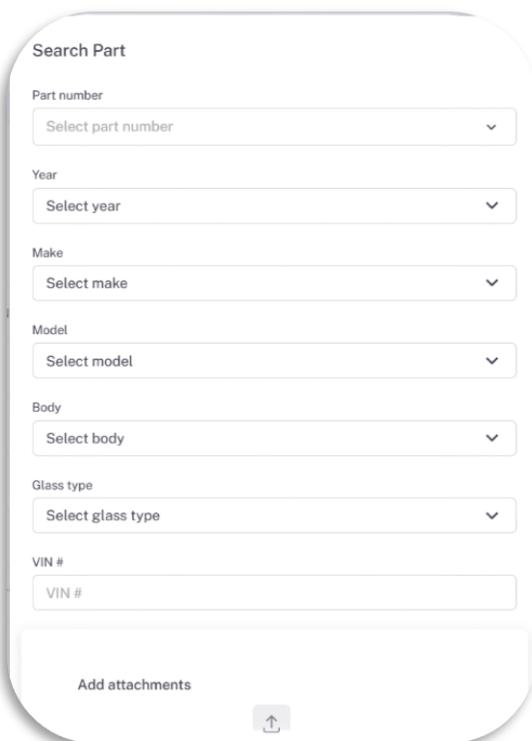


Figura 29 Vista del modal para la búsqueda de un numero de parte

Siguiendo con la estructura de este modal, se realizarán validaciones para corroborar que todos los campos hayan sido seleccionados previamente.

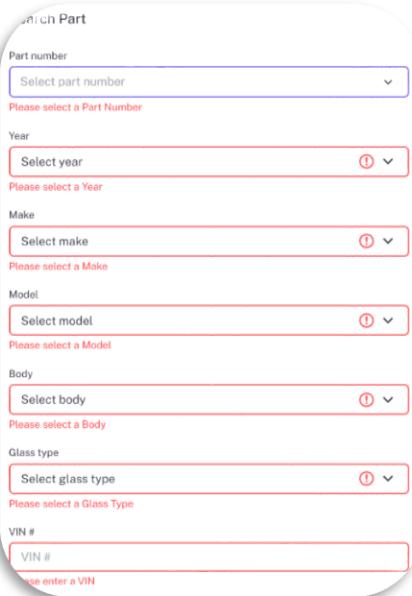


Figura 30 Vista del modal de búsqueda de numero de parte con validación de campos vacíos

El modal de para la creación de un enlace de detalles hacia una cotización realizada es más práctico, debido a que dentro de un modal se integrará una barra de búsqueda de la cotización correspondiente y el sistema generará un enlace el cual tendrá la opción de copiar con un botón y este se podrá compartir con sus usuarios finales.

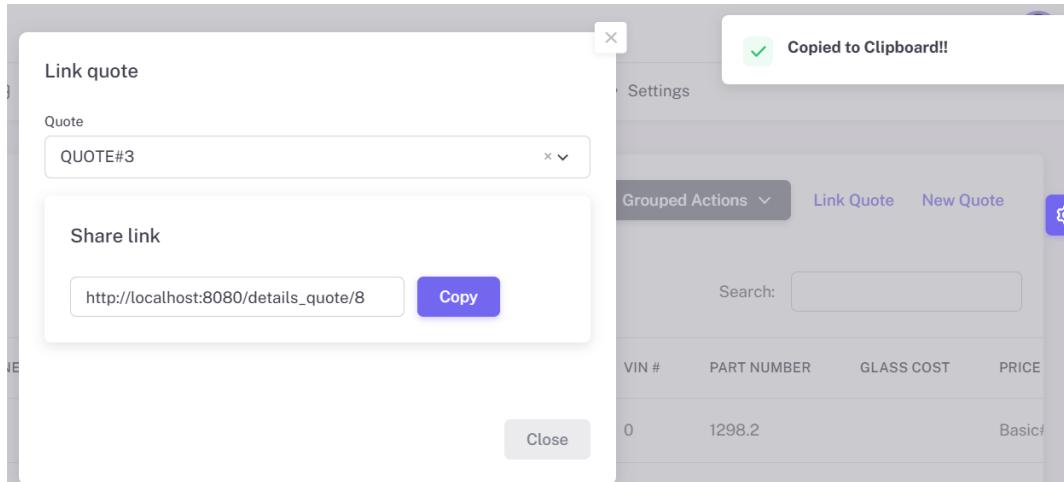


Figura 31 Vista de un modal para la creación de enlaces de detalles de una cotización existente.

Cuando el cliente tenga la cotización seleccionada y acceda al enlace, se tendrá desarrollada una nueva vista en la que el cliente podrá ver solo los detalles de dicha cotización.

Figura 32 Vista de los detalles de una cotización existente

La vista de esta sección se tomó en base al desarrollo del modal de detalles de una cotización, sin embargo, en este no se tendrá mayor acceso a detalles internos, por lo

que es una opción más fiable y segura de compartir con terceros para que puedan verlo lo más similar a lo que se tiene dentro del modal original.

### Desafíos y soluciones

- Desafío: Asegurar la precisión de los cálculos de precios.
  - Solución: Realizar pruebas exhaustivas de la lógica de precios y revisar los cálculos con el equipo de ventas.
- Desafío: Proveer una experiencia de usuario fluida y rápida
  - Solución: Optimizar las consultas a la base de datos y usar técnicas de carga dinámica para mejorar el rendimiento de la aplicación.

### Sprint 2: Módulo de trabajos

#### Desarrollo

Para el módulo de trabajos, se tiene como principal uso el mantener una gestión sobre el proceso y asignación de trabajo bajo una cotización previa, como punto importante no se podrá crear ninguna “Work order” sin una cotización hecha, por lo que la única forma de poder tenderla es mediante una cotización realizada.

El proceso por seguir dentro del módulo Jobs es el siguiente: Se mostrarán los datos del cliente, la información del vehículo, datos de la aseguradora que provienen de la cotización realizada, dichos datos solo estarán presentes para su lectura y consulta.

Se tendrá un apartado para el método de pago y distribuidor, en este apartado el cliente deberá capturar datos en relación con el distribuidor del vidrio correspondiente, así como el método del pago que usará el usuario, deberá de contar con opciones para mostrar una vista rápida de un enlace de método de pago y con este poder generar un recibo de pago.

Contará con un apartado para la carga de archivos, imágenes y documentos necesarios que serán parte de la evidencia de una “Work order”.

Como últimos datos en capturar, se tendrá un área para registrar las actualizaciones de estado de la “Work order”, así como tener una vista de historial de actualizaciones de todos los procesos que hayan transcurrido para tener un estatus de “Completed”, lo cual hace que la “Work order” haya finalizado en su captura.

Al igual que el capítulo pasado para el módulo de cotizaciones, se realizarán algunas operaciones y cálculo de “Tax”, para ello primero será necesario contar con los valores del costo del cristal calculado en una cotización y el precio anterior de una cotización, para que dentro de este módulo solo se realice una adición de gastos extra y tax para la obtención de un “Total due”.

Dentro de la acción para ejecutar un “Enlace Payment” será indispensable hacer uso de las validaciones de campos, ya que será datos indispensables para registrar para un recibo de pago, como dato extra, se podrá agregar una propina la cual será definida por el usuario.

Para la vista principal del módulo, además de hacer uso de un datatable, se deberá mostrar un conteo general de las “Jobs” terminadas, canceladas, abiertas y pendientes por pagar.

El diagrama creado previamente con las ideas y puntos clave, nos describe el uso de una vista de cards, para mostrar el conteo de Jobs, así como el uso de una datatable.

The screenshot shows the main view of the 'Jobs' module. At the top, there is a horizontal bar with five status indicators: 'Jobs Done' (0), 'Jobs Canceled' (0), 'Jobs Open' (0), 'Pending to Pay' (0), and '?????' (0). Below this is a table with the following columns: Work Order, Date, Customer, Year, Model, Make, Open, Part Number, Distributor, Tech, and Status. The table contains 8 rows, each with a 'Agregar' button followed by placeholder text like 'Aregar te', 'Aregar texto', etc. The 'Status' column for all rows shows a dropdown menu icon.

Work Order	Date	Customer	Year	Model	Make	Open	Part Number	Distributor	Tech	Status
Agregar	Agregar te	Agregar	Agregar	Agregar te	Agregar texto	▼				
Agregar	Agregar te	Agregar	Agregar	Agregar te	Agregar texto	▼				
Agregar	Agregar te	Agregar	Agregar	Agregar te	Agregar texto	▼				
Agregar	Agregar te	Agregar	Agregar	Agregar te	Agregar texto	▼				
Agregar	Agregar te	Agregar	Agregar	Agregar te	Agregar texto	▼				
Agregar	Agregar te	Agregar	Agregar	Agregar te	Agregar texto	▼				
Agregar	Agregar te	Agregar	Agregar	Agregar te	Agregar texto	▼				

Figura 33 Vista principal del módulo Jobs

Con el uso del template, el tema de la tabla ya se tiene contemplado en reutilizar las funcionalidades del datatable creada para módulos anteriores, realizando pocas modificaciones necesarias, sin embargo, para mostrar el conteo, se hará uso de “Cards”, de las cuales se tiene una amplia selección.

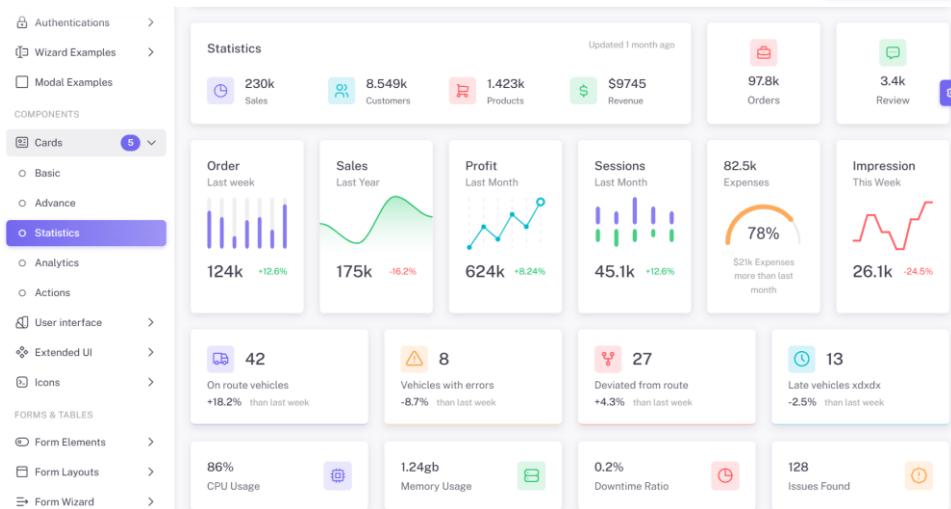


Figura 34 Vista de diferentes cards para analíticas

Mientras que para la vista de todos los datos que contendrá, se seguirá usando un modal con pestañas, para tener mejor organización de la información a capturar. Los datos se repartirán dentro de 4 secciones diferentes:

Customer/Vehicle, Distributor/Payment, Attachments y Updates.

Figura 35 Vista de las 3 primeras pestañas del modal para Jobs

The screenshot shows a software interface for managing work orders. On the left, there's a modal window titled 'WORK ORDER' with several tabs: 'Customer / Vehicle', 'Distributor / Payment', 'Attachments', and 'Updates' (which is highlighted with a red circle). Below these tabs are dropdown menus for 'Team', 'Tech', and 'Status'. A section labeled 'Job Send Out' contains two dropdown menus. To the right of these are five communication methods: Email (checked), Sms (unchecked), App (checked), WhatsApp (unchecked), and Others (unchecked). A 'Send' button is located at the bottom of this section. Below this is a list of status options with checkboxes: Completed, Cancelled, Tech Assigned, Tech Pending, Rescheduled, Pending Approval, Reserved, and Warranty. On the far right, a larger preview window shows detailed fields for 'Date', 'Time Frame', 'Address', 'City', 'State', 'Zip Code', 'Vehicle Information' (Year, Make, Model, Body), 'JobType', 'Calibration Type', 'Insurance Information' (Carrier, Policy ID, Referral #, Policy Holder, Policy Address, Deductible, NAGS Description, NAGS Labor, List Price, Price Per Hour), 'Distributor' (Part Number, Distributor Order Number, Distributor, Price Tier, Price, Tax, Total Due, Payment Type, Deductible, Deductible), and 'Attachments'.

Figura 36 Vista de la última pestaña del modal junto con una vista previa del modal completo

En cuestión de adaptar esta información a un modal con 4 pestañas, el trabajo se facilita debido a que para el módulo de cotizaciones se realizó uno con dos pestañas, por lo que se retomara de dicho modulo para solo hacer la implementación de los campos correspondientes.

The screenshot shows a 'New quote' modal. At the top, there are two tabs: 'Vehicle / Customer' (which is selected and highlighted with a blue underline) and 'Attachments'. Below the tabs are input fields for 'Name' (containing 'QUOTE#42'), 'Zip code' (with a dropdown menu showing 'Select zip c'), and 'Phone'. At the bottom of the modal, there is a section labeled 'Vehicle information'.

Figura 37 Vista de pestañas para el modal de cotizaciones

Por último, será necesario la creación de un nuevo modal para la gestión de un enlace de pago, en el cual se recuperarán los datos indispensables para la generación de un recibo de pago.

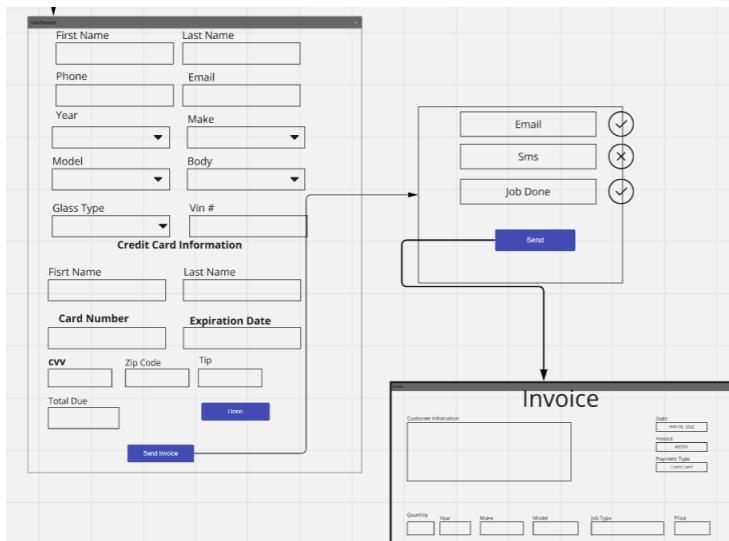


Figura 38 Vista de modal para un enlace de pago junto con la vista previa de un recibo

La vista del modal será a partir de uno ya existente como los que contiene un módulo independiente, haciendo las modificaciones correspondientes. Para el manejo del recibo de pago, el template nos ofrece un diseño sobre estos recibos, por lo que se optara por hacer uso de este. En relación con los datos de la tarjeta introducidos, de usaron archivos de validación del propio template para mostrar un formato más amigable y mayormente usado como la separación de 4 dígitos al escribir el número de tarjeta, la separación de fechas de expiración, así como solo aceptar el numero correspondiente a la clave CVV.

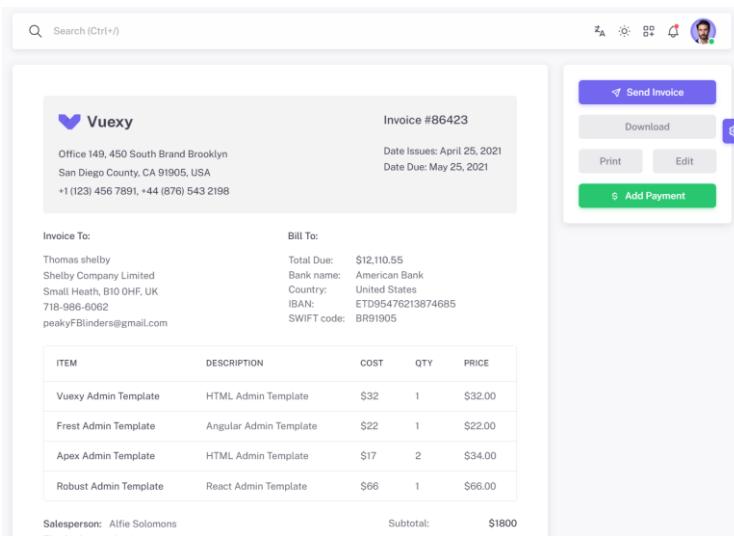


Figura 39 Vista previa de un recibo proporcionado por el template

La vista para validaciones será retomada de la realizada dentro del módulo de cotizaciones. Como punto a tomar, la creación de módulos independientes será también necesario para este módulo, los módulos independientes a realizar son los siguientes:

- Catálogo de equipos (cat\_team)
- Catálogo de envío de trabajo (job\_send\_out)
- Catálogo de estatus (cat\_status)
- Catálogo de Tipo de pago (cat\_payment\_type)
- Catálogo de técnicos (cat\_tech)
- Catálogo de distribuidores (cat\_distributor)
- Catálogo de tipo de trabajo (cat\_type\_job)

Así mismo, este módulo necesitará de dos tablas extra, catálogo de datos adjuntos para guardar la relación de rutas de archivos guardados y catálogo de actualización de estatus para guardar un historial de cambios de cada estatus que sufra la orden de trabajo.

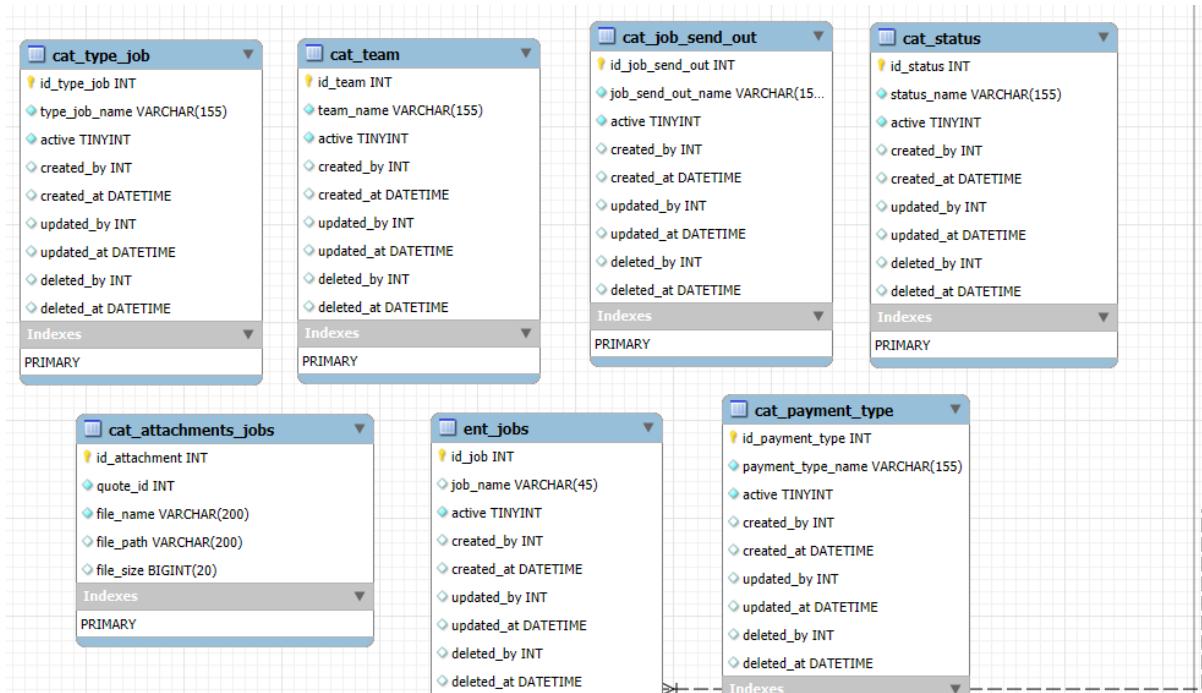


Figura 40 Diagrama de tablas a utilizar para módulo de trabajos

Se comenzará el desarrollo con la implementación de los módulos independientes, el proceso a seguir será el mismo que de ejecuto para el capítulo IV Modulo de cotizaciones. Con estos módulos independientes realizados, se continuará con el desarrollo en principio con el módulo de trabajos, que, a partir de su creación básica, se modificará a los requisitos solicitados.

List of Status			
		Grouped Actions ▾	
Show 10 ▾ entries		Search: <input type="text"/>	
#	NAME	STATUS	ACTIONS
9	Done	Activate	⋮ 0
8	Warranty	Activate	⋮ 0
7	Reserved	Activate	⋮ 0

Figura 41 Vista del módulo independiente de estatus

Se agregarán sobre la parte del datatable la vista correspondiente para las cards que llevarán el conteo de Jobs.

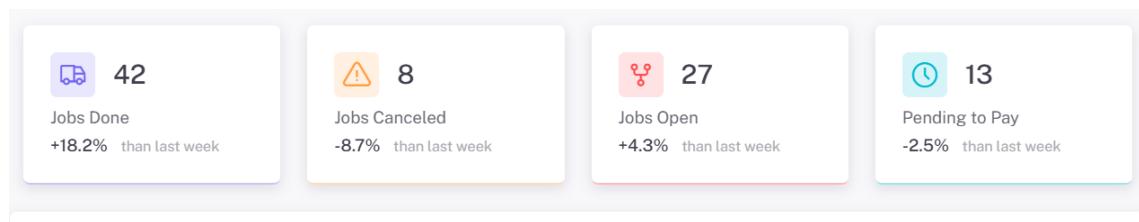


Figura 42 Vista previa de las cards de analíticas dentro del módulo de trabajos

Tabla 24 Fragmento de código correspondiente a una tarjeta

Código	Explicación
<pre>&lt;div class="card-body"&gt; &lt;div class="d-flex align-items-center mb-2"&gt;     &lt;div class="avatar me-4"&gt;         &lt;span class="avatar-initial rounded bg-label-primary"&gt;&lt;i class="ti ti-truck ti-28px"&gt;&lt;/i&gt;&lt;/span&gt;</pre>	<p>Parte de un componente de tarjeta que parece mostrar estadísticas o métricas de alguna clase.</p> <p>Crea una tarjeta con un ícono, un número destacado (42), una descripción breve (Jobs Done).</p>

<pre> &lt;/div&gt; &lt;h4 class="mb-0"&gt;42&lt;/h4&gt; &lt;/div&gt; &lt;p class="mb-1"&gt;Jobs Done&lt;/p&gt; &lt;p class="mb-0"&gt; &lt;span class="text-heading fw-medium me-2"&gt;+18.2%&lt;/span&gt; &lt;small class="text-muted"&gt;than last week&lt;/small&gt; &lt;/p&gt; &lt;/div&gt;</pre>	
--	--

Retomando algunos botones definidos dentro de cotizaciones para la creación de una nueva job, se les ha implementado funcionalidad sobre los mismos, haciendo que al querer crear una nueva job a partir de una cotización existente sea fácil y funcional, lo que se desarrolló dentro de esta funcionalidad fue la creación de peticiones Ajax para guardar los datos de una nueva cotización o actualización de los mismos, el extra para abrir el modal correspondiente dentro del módulo de trabajos parte desde agregar dentro de cotizaciones este código:

*"Window.location.href = base\_url + 'jobs?job\_id=' + response.job\_id,"*

Este define la acción para abrir el trabajo creado junto con la asignación del identificador único de su registro.

Una vez que hemos sido redirigidos hacia el módulo de trabajos, se usará el siguiente código de javascript para identificar si dentro de la url se tiene algún número de identificador de job:

*"Const queryString = window.location.search.*

*const urlParams = new URLSearchParams(queryString).*

*const jobId = urlParams.get('job\_id');*"

Estas tres líneas realizarán este proceso: Se hará una búsqueda dentro de la url con el objetivo de encontrar un valor que se encuentre asignado por *job\_id*.

Si se tiene algún valor encontrado se asignará a la constante declarada de jobId.

```
"if (jobId) {  
    mostrarModalTrabajo(jobId);  
    // Remover el parámetro job_id de la URL después de mostrar el modal  
    const newUrl = window.location.pathname;  
    window.history.replaceState({}, '', newUrl);  
}
```

Este otro fragmento incluye una validación, en donde si se obtuvo un valor jobId dentro de la url, este ejecutara la función correspondiente para hacer que el modal sea visible. Como detalle, una vez que se ejecuta la función para abrir el modal, el valor de jobId se elimina dentro de la url para que al momento de recargar la página no se vuelva a abrir.

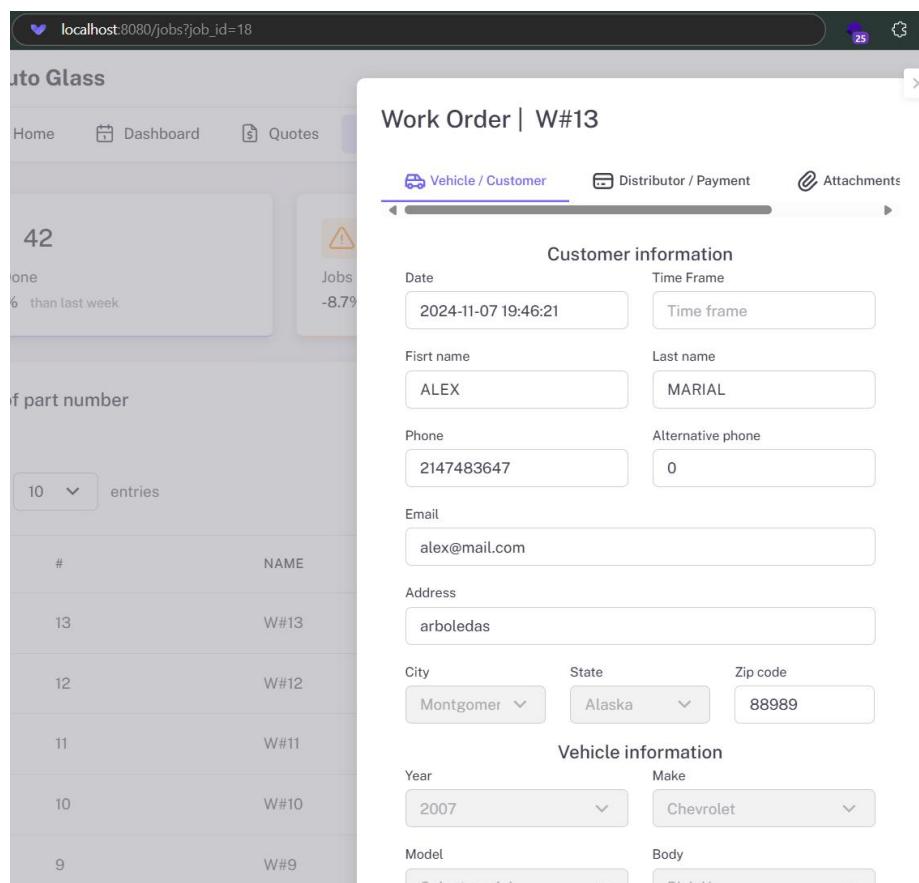


Figura 43 Apertura automática del modal para un trabajo

Como se aprecia dentro de la figura anterior, los datos dentro del modal ya se encuentran llenos con información, esto se debe que, dentro de la función para mostrar dicho modal, se hace la ejecución previa de un Ajax hacia “`url: base_url + 'Ntsdata/Jobs/Cjobs/get_data_by_quote/' + jobId,`” para la obtención de datos y asignar dentro del propio modal. Este proceso de obtención de datos y mostrarlos dentro de un modal es el que siempre se ocupara para todo modulo sea independiente o no. Este modal solo podrá actualizar los datos de un trabajo, para la carga de archivos se retomó la lógica utilizada para el módulo de cotizaciones por lo que su implementación se realizó en menor tiempo. Mientras que para la pestaña de detalles del distribuidor o detalles del método de pago se realizaron validaciones de campos vacíos, así como el cálculo de las entradas de datos correspondientes. Dentro de esta sección serán pocos los campos que serán solo de lectura y se hará uso de campos seleccionables para selección del técnico y distribuidor.

The screenshot shows a modal window titled "Work Order | W#13". At the top, there are three tabs: "Vehicle / Customer" (disabled), "Distributor / Payment" (selected), and "Attachments". The "Distributor" tab is active, displaying the following fields:

- Part number: 1298.2
- Glass cost: 66
- Distributor Order Number: (dropdown)
- Distributor: Select distributor (dropdown)
- Tech: Select tech (dropdown)
- Labor: 00,00
- Price tier: Recommend (dropdown)
- Price: 888
- Tax: 2
- Others: 2
- Total Due: 908.69
- Payment Type: Basic#2 (dropdown)
- Credit card information:
  - First name: Oscar
  - Last name: Meza
- Card number: 8978 9789 7978 9789
- Expiration date: 08/97

Figura 44 Vista del modal dentro de la pestaña de distribuidor

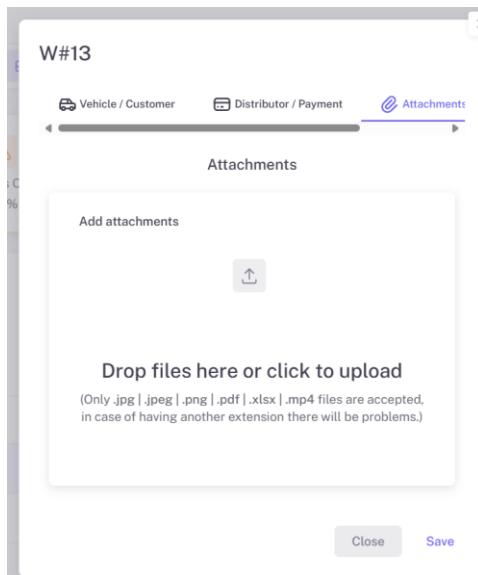


Figura 45 Sección de attachments para la carga de archivos

Una de las secciones que implicara más desarrollo debido a que no existe dentro de algún modulo existente hasta el momento, es la última pestaña que corresponde a las actualizaciones de estatus de trabajo, dentro de este se tendrán varios campos seleccionables, para la selección del equipo responsable, su el medio de trabajo, la selección del técnico responsable y su medio de trabajo, así como el propio estatus de avance. Debajo de estos campos tendrá un panel con una vista de línea del tiempo vertical con el que se podrá ver el historial de avance del trabajo.

Cuando se tengan datos dentro de updates, este actualizara dentro de la propia tabla los valores, así como lo agregara dentro de una tabla la cual guardara el historial de proceso.

Tabla 25 Tabla 25 5.15 Funcionalidad para agregar los valores de actualizaciones de trabajo dentro de un historial

Función “update_register” de Cjobs.php	
Código	Explicación
<pre>if (\$old_status_job === false)     \$error = \$this-&gt;db-&gt;error ().  \$response = array ('status' =&gt; 'error',     'message' =&gt; 'Database error.').     echo json_encode(\$response).</pre>	Verifica si hubo un error al consultar la base de datos para obtener el estado anterior de un trabajo. Si hubo un error, devuelve una respuesta JSON con un mensaje de error y termina la ejecución.

<pre>     return;      \$old_status = (\$old_status_job-&gt;num_rows () &gt; 0)? \$old_status_job-&gt;row ()-&gt;id_status: null; if (\$old_status === null    \$old_status!=     \$statusUpdateJob) {     \$job_data_update = array (         'Job_id' =&gt; \$id,         'Update_status' =&gt; \$statusUpdateJob,         'Id_team_update' =&gt; \$teamUpdateJob,         'Id_job_send_out_team_update' =&gt;             \$jobSendOutUpdate,         'Id_tech_update' =&gt; \$techUpdateJob,         'Id_job_send_out_tech_update' =&gt;             \$jobSendOutTechUpdate,         'Updated_by' =&gt; \$this-&gt;session-&gt;userdata('s_id_sesiones'),         'updated_at' =&gt; date ('Y-m-d H: i:s'),);     \$insert_result = \$this-&gt;db-&gt;insert     ('cat_updates_jobs', \$job_data_update); } </pre>	<p>Si no hubo error, obtiene el estado anterior del trabajo. Si el estado anterior es nulo o diferente del nuevo estado, inserta un nuevo registro en la tabla cat_updates_jobs con los detalles de la actualización.</p> <p>Registra los cambios de estado de un trabajo en la base de datos, asegurándose de no registrar un cambio si el nuevo estado es igual al anterior. También incluye manejo de errores básico para la consulta inicial que recupera el estado anterior del trabajo.</p>
--	---

Una vez con los datos almacenados en la tabla del historial, se utilizará javascript para iterar cada uno de los registros que tenga de actualización cada trabajo.

Tabla 26 Código de JS correspondiente a la generación de la vista de actualización de un trabajo

Archivo “Jobs.js”	
Código	Explicación
<pre> var timeline = \$("newRecordModal #timeline").     timeline. empty (); if (response.data. updates) </pre>	<p>Construye dinámicamente una línea de tiempo visual</p>

<pre> response.data. updates. forEach (function (update, index) {     var isLastUpdate = (index === response.data. updates.         length - 1);      var listItem = \$("&lt;li&gt;",         {             class: "timeline-item timeline-item-transparent " +             (isLastUpdate? "Border-transparent pb-0": "border-                 primary")});      var timelinePoint = \$("&lt;span&gt;",         {             class: "timeline-point                 timeline-point-primary" });      var timelineEvent = \$("&lt;div&gt;",         {             class: "timeline-event " +             (isLastUpdate ? "pb-0": "")));      var timelineHeader = \$("&lt;div&gt;",         {             class: "timeline-header" });              var title = \$("&lt;h6&gt;",                 {                     class: "mb-                 0"}).text(update.status_name); // Llamar a la función                 getStatusText              var date = \$("&lt;small&gt;",                 {                     class: "text-muted" }).text(update.updated_at);      var description = \$("&lt;p&gt;",         {             class: "mt-3 " + (isLastUpdate ?                 "mb-3": "")}).text(update.company_name    "");      timelineHeader.append(title).append(date);     timelineEvent.append(timelineHeader);     timelineEvent.append(description);      listItem.append(timelinePoint).append(timelineEvent);     timeline.append(listItem);});} } </pre>	<p>dentro de un modal, utilizando datos recibidos de una respuesta.</p> <p>Toma datos de una respuesta, itera sobre las actualizaciones y crea una línea de tiempo visual con información sobre cada actualización, incluyendo el estado, la fecha y el nombre de la empresa. La última actualización en la línea de tiempo recibe un estilo diferente para indicar que es la más reciente.</p>
---	---

Con este proceso realizado, solo basta abrir cualquier modal para editar y este agregara de manera dinámica los estatus correspondientes del trabajo a ver.

W#12

Filter: Distributor / Payment, Attachments, Updates

Updates

Team: Team ONE, Job Send Out: Sms

Tech: Elite Tech Services, Job Send Out: App

Status: Completed

Updates

- Pending Approval, 2024-11-04 12:02:17, Elite Tech Services
- Reserved, 2024-11-04 12:02:36, Elite Tech Services
- Rescheduled, 2024-11-04 12:02:56, Elite Tech Services
- Completed, 2024-11-04 14:07:59, Elite Tech Services

Figura 46 Vista de la última sección para ver el historial de actualizaciones

Continuando con el desarrollo del módulo de trabajos, en la sección 2 para el proceso de pago, se tiene un apartado con diferentes acciones, de las cuales se realizará el modal de un enlace de pago.

Link Payment, Run Credit Card, Save

Notes

Close, Save

Figura 47 Vista inferior de la segunda sección del modal

El modal para mostrar deberá de contener los datos ya existentes correspondientes a los datos del usuario y los datos de la tarjeta de crédito, en caso de contar con estos definidos, se hará uso con validaciones para mostrar de manera visual al usuario que

no puede continuar si no se llenan los campos necesarios, un dato que se mantiene nuevo es el de tip o propinas, este será el único campo opcional.

The screenshot shows a 'Link Payment' modal with two main sections: 'Link Payment' and 'Credit Card Information'. The 'Link Payment' section contains fields for First name, Last name, Phone, Email, Year, Make, Model, Body, Glass type, and VIN #. The 'Credit Card Information' section contains fields for First name, Last name, Card number, Expiration date, CVV, Zip code, and Tip. All fields are outlined in red, indicating they are required. Error messages are displayed below each field: 'Please enter a First Name' for First name, 'Please enter a Last Name' for Last name, 'Please enter a Phone Number' for Phone, 'Please enter an Email' for Email, 'Please enter a Card Number' for Card number, 'Please enter an Expiration Date' for Expiration date, 'Please enter a CVV' for CVV, and 'Please enter a Zip Code' for Zip code.

Figura 48 Vista del modal de enlace de pago solicitando los campos obligatorios faltantes

En el momento de tener todos los campos obligatorios definidos, se podrá continuar el proceso y ya sea terminarlo con la opción “Done” o directamente generar el recibo correspondiente, cabe mencionar que, para ambas opciones al dar clic, se agrega un último estatus el cual equivale a “completed”.

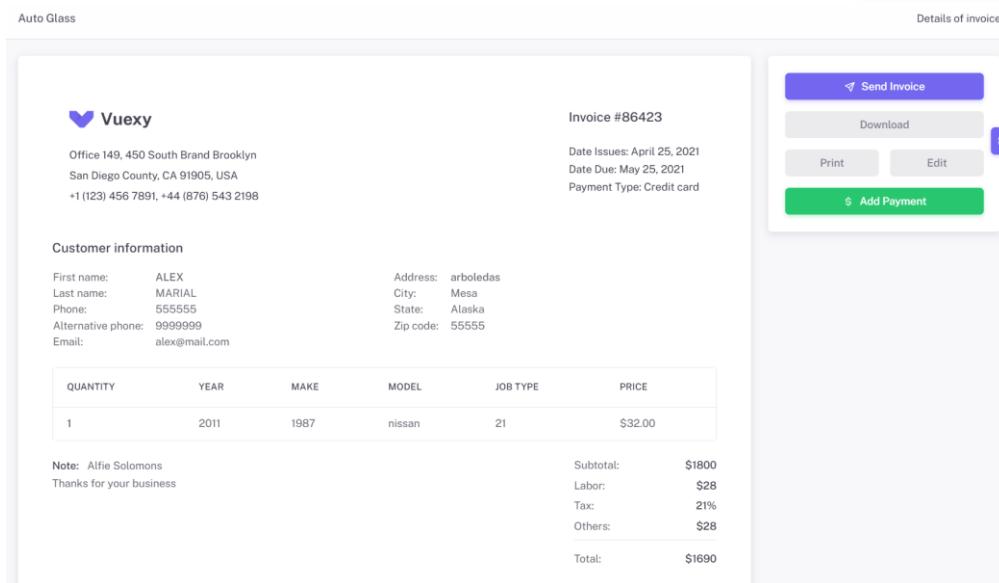


Figura 49 Vista de recibo de pago de un trabajo

## Desafíos y soluciones

- Desafío: Garantizar la integridad de los datos en el registro y seguimiento de trabajos.
  - Solución: Implementar validaciones de datos robustas tanto en frontend como en backend, además de pruebas unitarias y de integración.
- Desafío: Facilitar la entrada de datos rápida y eficiente para los técnicos.
  - Solución: Diseñar formularios intuitivos y optimizados para la entrada de datos rápida, incluyendo el auto relleno y validaciones en tiempo real.

## Sprint 3: Módulo de mapas

### Desarrollo

El módulo de mapas tiene el objetivo principal de mostrar sobre un mapa, las ubicaciones de todas las sucursales con las que cuenta el cliente, las cuales ayudaran a sus usuarios para una mejor ubicación y referenciar más fácil la ubicación.

Para este módulo, se tiene solo dentro del diagrama general la vista hacia un mapa del mundo con chinchetas, tratando de marcar alguna ubicación valida de una sucursal.

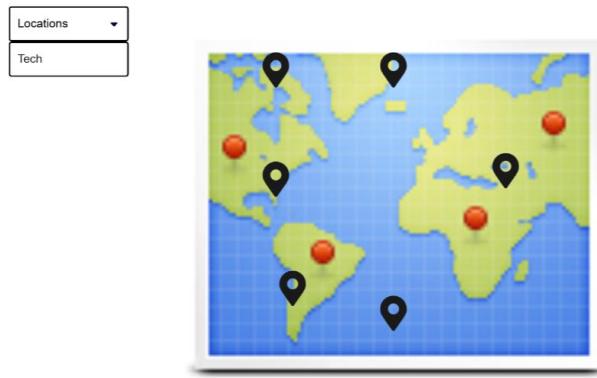


Figura 50 Vista de diagrama para el módulo de mapas

Reutilizando los recursos que nos ofrece el template, este cuenta con una librería de mapas llamada “leaflet map” el cual ofrece una vista de mapa bastante atractiva y en diferentes configuraciones.

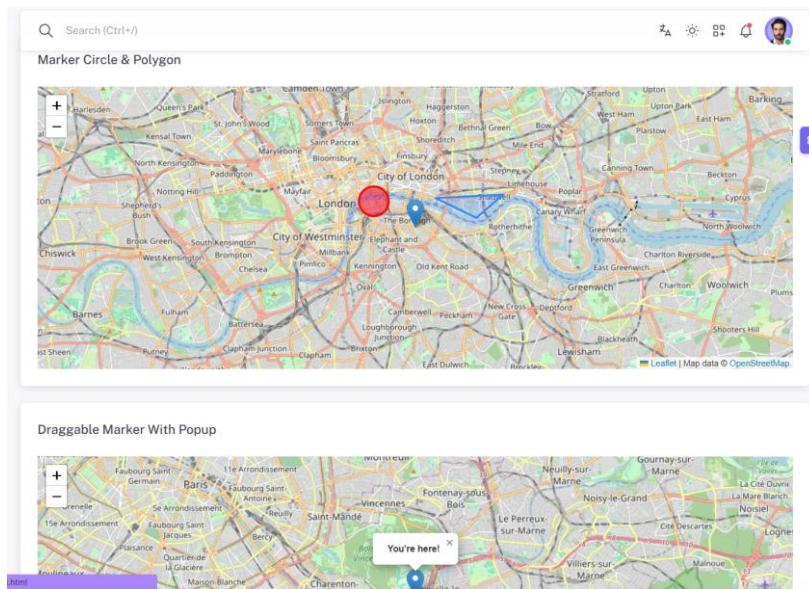


Figura 51 Vista de mapas con diferentes configuraciones proporcionadas por el template

Con base en los puntos clave mencionado en los mapas, se utilizará el mapa “Layer Control” el cual contiene diferentes chinchetas marcando diferentes ubicaciones.

Este módulo no tendrá una datatable o un modal para creación y edición, solo será la vista previa del mapa con las ubicaciones obtenidas desde la tabla de sucursales (cat\_branch), más específicamente a través de su latitud y longitud. Por lo que seguir los pasos para la creación de un módulo independiente como punto de partida no

aplica en este caso, así como también no contara con una tabla en específico. Para este módulo, se desarrollará la lógica para enviar la latitud y longitud correspondientes de todas las sucursales existentes, para que con ayuda de la librería de leaflet map se muestren con un marcador sobre el mapa.

En el desarrollo se colocará una vista con la siguiente etiqueta:

`<div class="leaflet-map-branch" id="map"></div>`

Con esto estamos con la clase que esta etiqueta corresponde a la vista de un mapa y con el id se retomara para añadirle la lógica de los lugares y ubicación actual del usuario.

Retomando los recursos disponibles dentro del template para un mapa con leaflet map, nos ofrece distintas funcionalidades ya incorporadas, sin embargo, al acceder a su página de documentación se mostrarán más usos e implementaciones a distintos proyectos, para este caso en particular, será necesario incorporar dos funcionalidades, la obtención de la ubicación y mostrar distintas ubicaciones fijas.

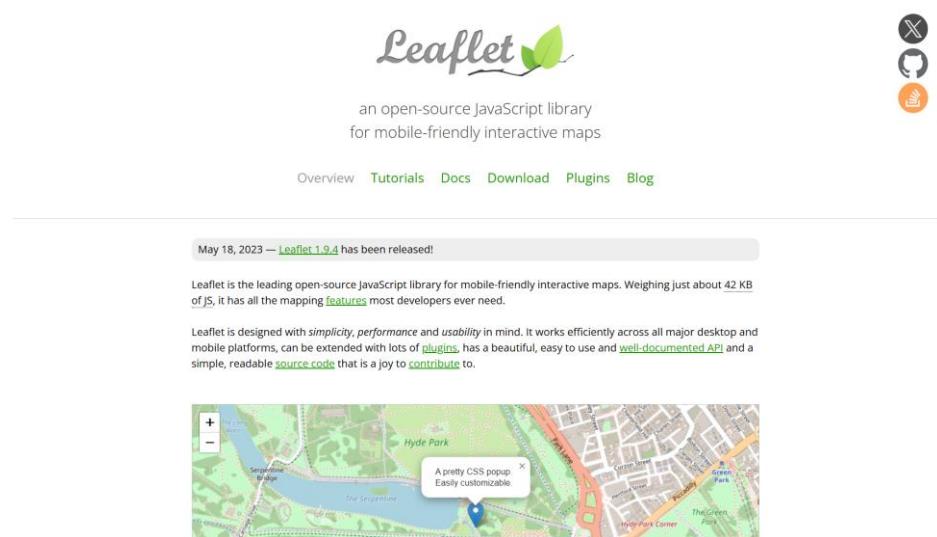


Figura 52 Página oficial de la librería leaflet

Después de una previa investigación para la implementación de cada funcionalidad, se desarrollará un script correspondiente que seguirá la siguiente lógica:

- 1. Crear y configurar el mapa:** Inicializar el mapa en el elemento HTML especificado y lo centra en unas coordenadas específicas con un nivel de zoom inicial.

2. **Agregar la capa de mosaicos (tiles):** Añade una capa de mosaicos de OpenStreetMap para proporcionar los datos visuales del mapa.
3. **Obtener datos de las sucursales:** Realiza una solicitud AJAX para obtener datos JSON de las sucursales desde el servidor. Si la solicitud es exitosa, procesa los datos recibidos y crea un grupo de marcadores para las sucursales.
4. **Agregar marcadores de sucursales al mapa:** Itera sobre cada sucursal y crea un marcador en su ubicación con un popup que muestra el nombre de la sucursal y un enlace a más detalles.
5. **Ajustar la vista del mapa a los marcadores:** Si hay sucursales, ajusta la vista del mapa para que todos los marcadores sean visibles.
6. **Localización del usuario:** Intenta localizar la posición del usuario y centrar el mapa en su ubicación. Si la ubicación se encuentra, añade un marcador y un círculo indicando la precisión. En caso de error, muestra un mensaje de alerta.

Para la petición de Ajax mencionada, esta se realizará a un archivo de controlador, que a su vez solicitará los datos de un archivo de modelo que consultará a la tabla de sucursales, obteniendo todos los registros en donde el valor de “activo” sea igual a 1 lo que corresponde a que solo se muestren las que sigan activas.

Tomando estos pasos en cuenta, obtendremos un mapa en el cual mostrara la ubicación actual del usuario con las ubicaciones de las sucursales obtenidas, cabe recalcar que tal mapa se cargara con la ubicación del usuario, dándole la posibilidad de arrastrar, aumentar o disminuir el zoom de este para poder visualizar las sucursales existentes.

Para la vista de las sucursales solo se mostrará una chincheta en la cual al tocar o dar clic desplegará un pequeño mensaje con información del nombre de la sucursal, como punto extra este contendrá un enlace para ver los detalles de esta solicitada.

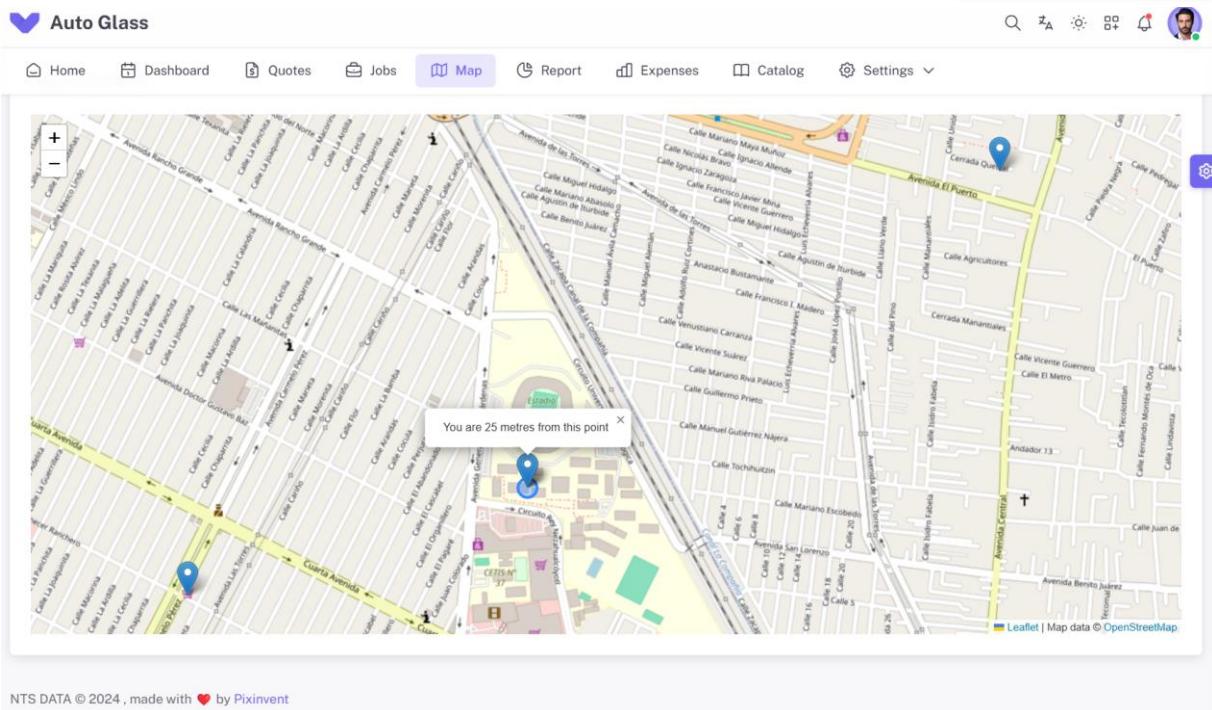


Figura 53 Vista de mapa de usuario con sucursales existentes

Para la vista de los detalles de cada sucursal, se organizará la con el diseño de menú sin opciones para un acceso público y sin opción de acceder a funcionalidades no autorizadas. Una vez teniendo en cuenta el diseño a usar, se utilizará de nuevo la lógica de obtención latitud y longitud que le corresponde a la sucursal por ver, por lo que la función de la petición por Ajax solo obtendrá todos los datos de la sucursal en base a un id proporcionado, que dicho id es enviado desde el mapa de todas las sucursales. La lógica será la siguiente:

#### 1. Inicialización del mapa detallado:

- Declarar variables:** Se declaran las variables detailMap y detailMapInitialized.
- Verificar inicialización:** Se verifica si detailMapInitialized es falso. Si es falso, se inicializa el mapa en un elemento HTML con el id 'detailMap', se centra el mapa en unas coordenadas específicas con un nivel de zoom de 16, se añade una capa de mosaicos de OpenStreetMap y se establece detailMapInitialized a verdadero para evitar volver a inicializar el mapa.

#### 2. Obtener y verificar coordenadas:

- **Parsear coordenadas:** Se convierten las coordenadas latitud y longitud de la respuesta en números flotantes.
- **Verificar validez:** Se verifica si las coordenadas no son NaN. Si las coordenadas son válidas, se centra el mapa en las coordenadas proporcionadas y se añade un marcador con un popup que indica la ubicación de la sucursal. Si las coordenadas no son válidas, se muestra un mensaje de error en la consola.

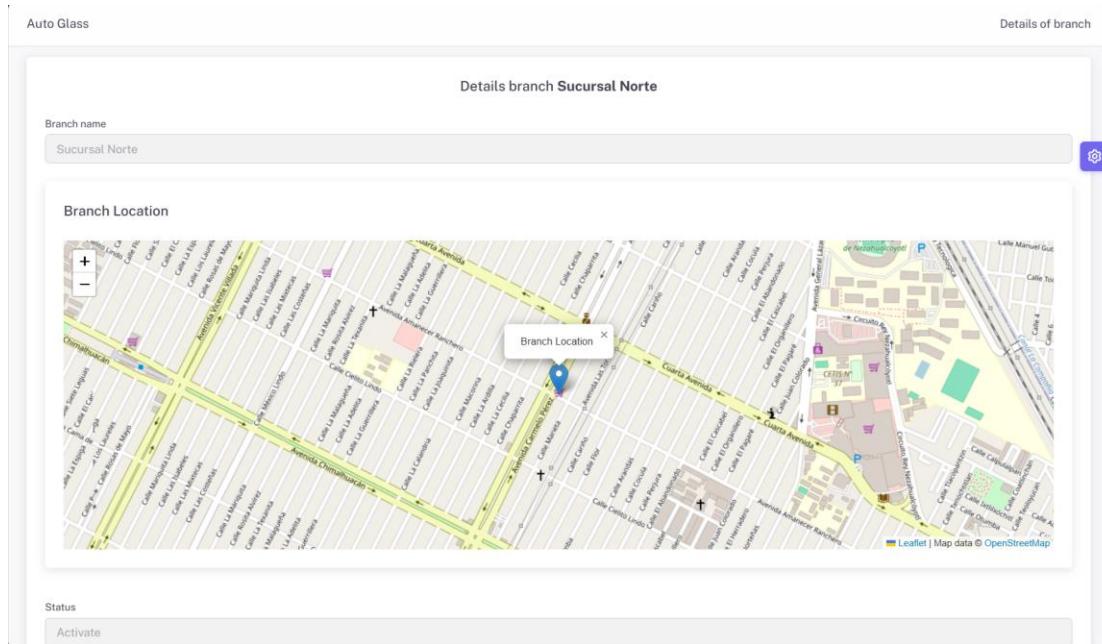


Figura 54 Vista detalles de una sucursal

Todos los cambios realizados se implementaron dentro de la rama SP5-1 dentro del repositorio al igual que ser realiza el procedimiento para enviar los cambios hacia la rama de desarrollo en espera de aprobación por el asesor empresarial.

### Desafíos y soluciones

- Desafío: Garantizar que el mapa fuera responsive y funcionara bien en dispositivos móviles.
  - Implementación: Utilizar un diseño adaptativo y realizar pruebas exhaustivas en diferentes dispositivos y tamaños de pantalla.
- Desafío: Recuperación del valor de id de sucursal para la vista detallada de una sucursal.

- Solución: Implementación de scripts para el envío y recepción de valores desde etiquetas con atributos “id” definidos.

## Sprint 4: Módulo de configuraciones

### Desarrollo

La funcionalidad de este módulo permitirá ver, editar y administrar los usuarios existentes dentro de la plataforma. También incluirá accesos directos para la creación de nuevos usuarios con roles específicos, destacando su ventaja sobre los módulos de tablas independientes gracias a la inclusión de campos especializados para cada tipo de rol.

Además, basándonos en las características destacadas de los sprints anteriores, este módulo deberá contar con las siguientes funcionalidades:

- Validaciones de campos para asegurar la integridad de los datos.
- Verificación de que los correos electrónicos no estén duplicados.
- Carga de archivos y la opción de subir una imagen de perfil.
- Vista de un modal para obtener un enlace con detalles específicos de un tipo de usuario.
- Lista de códigos postales (zip\_codes) disponible para consulta dentro de un modal.
- Encriptación de contraseña para su almacenamiento en la base de datos.
- Opción para restablecer la contraseña de un usuario y enviarla a su correo electrónico.

Se está mencionando que se registrarán diferentes tipos de usuario en base a un rol específico, por lo que, si se tendrá una tabla por cada tipo de rol de usuario existente, para ello la estructura de las tablas serán 2 principales: ent\_usuario y cat\_persona.

Dentro de la tabla ent\_usuario, se almacenarán los datos de:

- Correo\_electronico
- Id\_rol
- Contrasenia
- Id\_persona

- Ultimo\_inicio\_sesion

Mientras que para la tabla cat\_persona almacenara los datos personales de todos los usuarios de esta forma:

- Nombres
- Apellidos
- Teléfono
- Imagen\_perfil

Es indispensable tener al usuario registrado en ambas tablas, ya que en una solo se almacenan datos de acceso, mientras que sus datos personales estarán en una tabla separada para mayor seguridad al no tener todos los datos de los usuarios en una sola tabla.

Mientras que por las tablas que lleva cada tipo de usuario son las siguientes:

- Cat\_sale
- Cat\_distributor
- Cat\_tech
- Cat\_admin
- Cat\_employee
- Cat\_attachments\_distributor
- Cat\_attachments\_tech
- Cat\_attachments\_admin
- Cat\_attachments\_employee

Las tablas que llevan la palabra “attachments” hace referencia a que dentro de estas se almacenaran los datos de donde se almacenan los datos adjuntos de cada tipo de usuario.

Cada una de las tablas lleva ciertos datos específicos como comisiones, salarios, numero de banco, dependiendo el rol serán los datos específicos extra para almacenar.

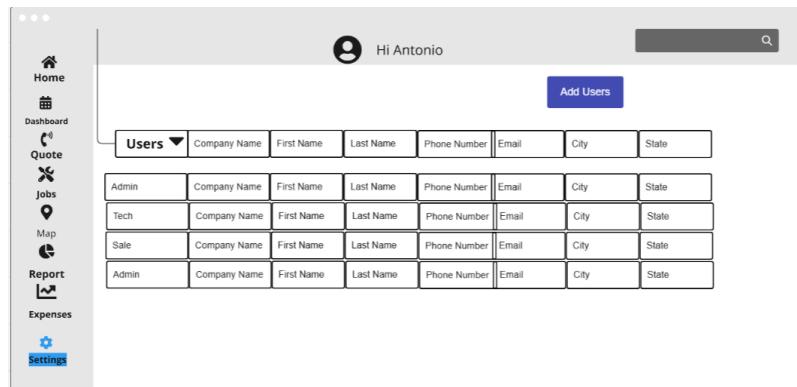


Figura 55 Diagrama de diseño para la vista de usuarios

Para la parte del diseño de tomar en base para la primera vista los elementos que usa una tabla independiente, así como la modificación de datos correspondientes dentro del modal para creación edición y lectura. Por lo tanto, se utilizarán recursos del template hacer la carga de una imagen de perfil y mostrarla.

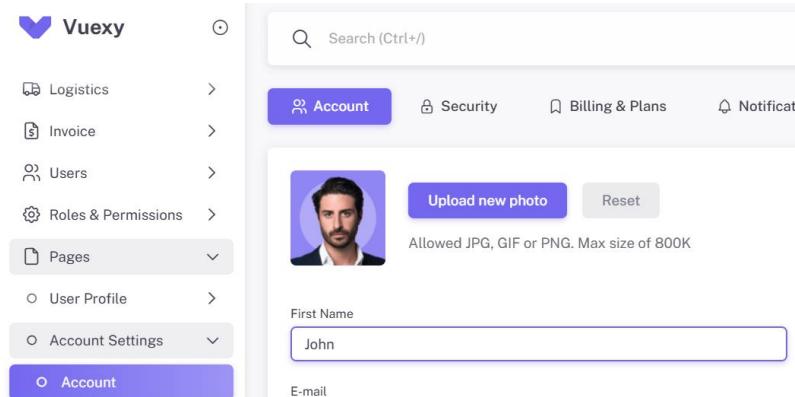


Figura 56 Vista de edición de perfil dentro del template

Se hará uso de los recursos del template para mostrar alertas personalizadas que indiquen al usuario sobre el registro exitoso de un usuario nuevo o la carga exitosa de una nueva imagen de perfil, debido a que dentro de cada vista no será utilizado un modal si no que solo será el formulario directamente sobre la vista principal, el uso de toastr permite hacer esta implementación, así como ofrece personalizar el estilo y contenido de cada uno como se muestra en la siguiente figura.

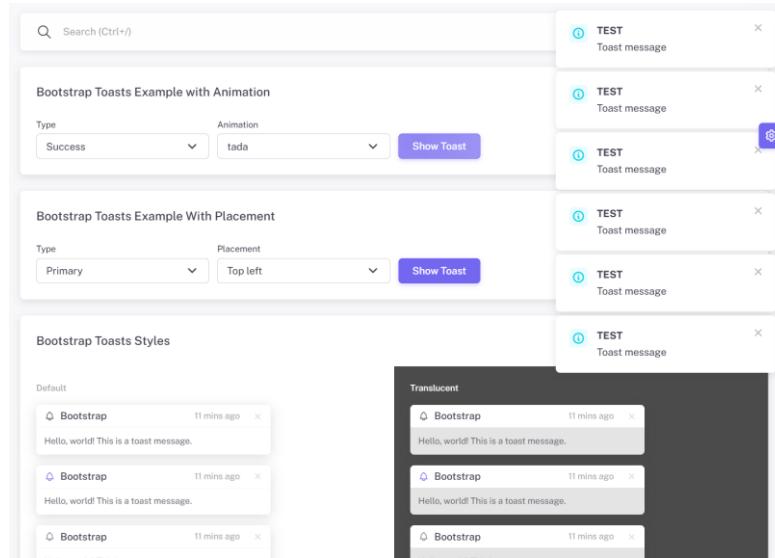


Figura 57 Vista de los recursos de toast dentro del template

Con estos recursos, se desarrollará en primer paso la vista de la tabla de usuarios, así como el modal para creación, edición y lectura de cada uno por medio del desarrollo de una tabla independiente, el proceso actualmente no demora más de 45 minutos debido a la estandarización de este proceso para hacerlo lo más ágil posible. El punto extra en agregar son las validaciones de campos para el correo electrónico, así como sus datos personales.

Show	10	entries	Search:				
	#	USER	ROLE	NAME	LAST NAME	STATUS	ACTIONS
<input type="checkbox"/>	65	manolo.distributor@mail.com	Distributor	Manolo	Floo	Activate	<span>⋮</span> <span>🔗</span>
<input type="checkbox"/>	64	paola.tech@mail.com	Tech	Paola	Juarez	Activate	<span>⋮</span> <span>🔗</span>
<input type="checkbox"/>	63	oscar.sale@mail.com	Sale	Oscar	Villanueva	Activate	<span>⋮</span> <span>🔗</span>
<input type="checkbox"/>	62	pedroadmin@mail.com	Administration	Pedo	picapiedra	Activate	<span>⋮</span> <span>🔗</span>
<input type="checkbox"/>	61	anna.employee@mail.com	Employee	Anna	Young	Activate	<span>⋮</span> <span>🔗</span>
<input type="checkbox"/>	60	karla.employee@mail.com	Employee	Karla	Villegas	Activate	<span>⋮</span> <span>🔗</span>
<input type="checkbox"/>	59	osvaldovillkdskllalba-02@hotmail.com	Employee	OSVALDO	VILLALBA MENDOZA	Activate	<span>⋮</span> <span>🔗</span>
<input type="checkbox"/>	58	pablo.employee@mail.com	Employee	Pablo	Deep	Activate	<span>⋮</span> <span>🔗</span>

Figura 58 Vista de la tabla principal para los usuarios existentes

Las validaciones de campos se realizan por medio de un script el cual detecta si el campo a validar se encuentra vacío o no cumple con los requisitos solicitados regresara un mensaje personalizado.

The screenshot shows a 'New User' registration form with several validation errors highlighted by red borders and error messages:

- User:** The 'User' field has a red border and the message "Please enter a email".
- Name:** The 'Name' field has a red border and the message "Please enter a Name".
- Phone:** The 'Phone' field has a red border and the message "Please select a City".
- City:** The 'City' dropdown has a red border and the message "Please select a City".
- Role:** The 'Role' dropdown has a red border and the message "Please select a Role".

At the bottom right are 'Close' and 'Register' buttons.

Figura 59 Modal de registro de nuevo usuario validando los datos

Existe una validación más, aunque esta se ejecutara después de validar los primeros campos, específicamente antes de registrar los datos del usuario dentro de las tablas correspondientes se realizará una validación sobre la existencia del correo para asegurar que este no se repita, ya que en caso de existir no se continuara el registro hasta cambiar el correo electrónico.

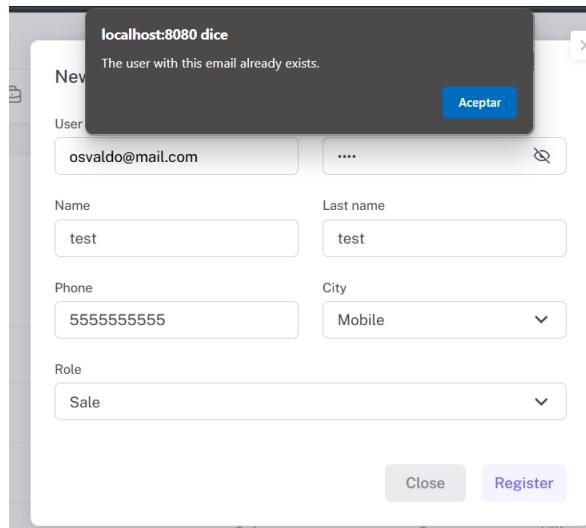


Figura 60 Alerta para usuario en caso de que el correo ya se encuentre previamente

Para la vista del registro de un solo tipo de usuario, el diseño será basado en relación con el diagrama que se tiene, cabe recalcar que no contendrá ninguna tabla debida que solo será el formulario de datos.

Figura 61 Vista de diagrama para nuevo distribuidor

En la figura anterior, se observa el contenido que incluirá un usuario de tipo distribuidor, en el cual además de retomar los campos necesarios de datos personales y usuario, registra datos de dirección, notas y archivos.

Para su desarrollo se reutilizarán las clases correspondientes que contenían los campos de entrada de las tablas independientes, por lo que su vista debe ser similar a las que se tienen para el ingreso de datos. Se agregará la opción con la que cuenta el template para la carga de una imagen de perfil.

Figura 62 Vista para el registro de datos de un nuevo usuario con rol de distribuidor

El formulario contara con validaciones para los campos vacíos o con información no correspondiente para evitar posible inconsistencia de datos.

The screenshot shows a 'Distributor data' form with several input fields and validation messages:

- Distributor:** A red border surrounds the 'Distributor name' field, and the placeholder 'Please enter Distributor Name' is visible.
- First name:** A red border surrounds the 'First name' field, and the placeholder 'Please enter a Name' is visible.
- Last name:** A red border surrounds the 'Last name' field, and the placeholder 'Please enter a Last Name' is visible.
- Email:** A red border surrounds the 'Email' field, and the placeholder 'Please enter an email address' is visible.

On the right side of the form, there is a vertical panel labeled 'Notes' with the sub-label '(On)' and a button 'Add a'.

Figura 63 Validación de datos sobre la creación de un nuevo registro

Como punto a retomar también será validado el correo electrónico para en caso de ya existir se muestre por medio de un toast que el correo ya existe.

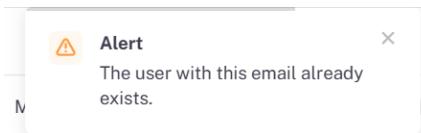


Figura 64 Mensaje toast para aviso de correo existente

Como punto extra, esta vista contará con un botón para crear un enlace de detalles de un distribuidor existente, por lo que se retomará la lógica del módulo de cotizaciones para copiar la url con un clic, sin embargo, el cambio será en la presentación del toast.

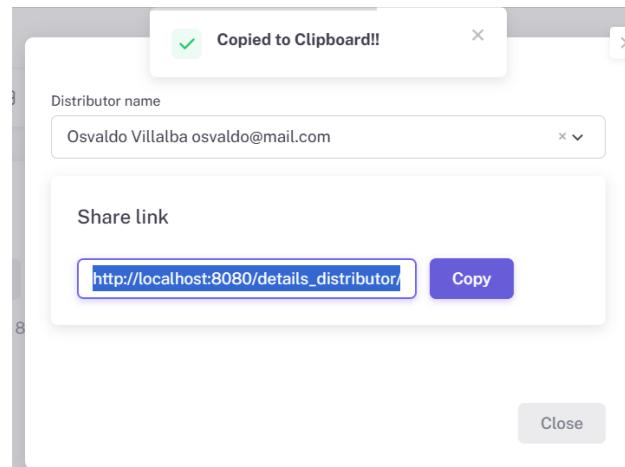


Figura 65 Modal para creación de un enlace de detalles de usuario con mensaje toast

Para la vista de detalles se retomará el mismo diseño de la creación del usuario, haciendo cambios en dejar los campos desactivados y un menú sin opciones.

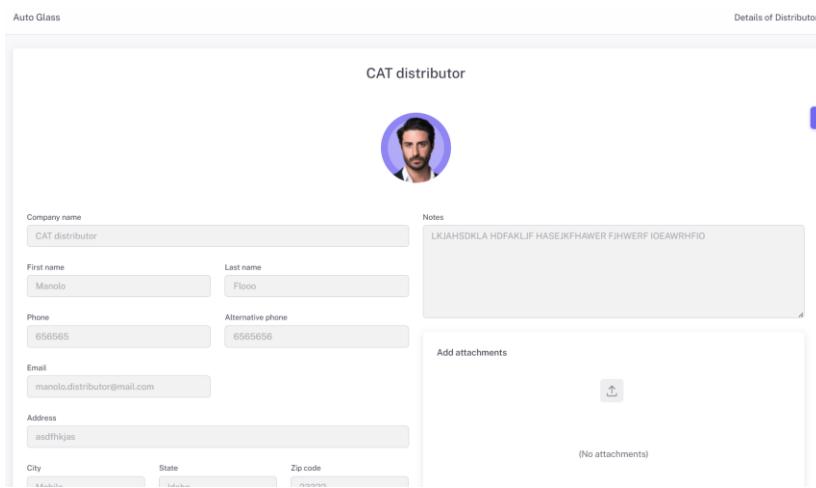


Figura 66 Vista detalles de un distribuidor

El desarrollo para los tipos de usuario restante retomara el mismo proceso, acoplándose solo los campos exclusivos de cada tipo de rol. En el caso de un usuario con rol tech, contara además del formulario base, con apartados para el registro de números bancarios, así como una nueva funcionalidad para registrar un nuevo código postal y acceder a una lista de los códigos postales existentes, el campo para registrar una contraseña y la funcionalidad para restablecerla por medio de correo electrónico.

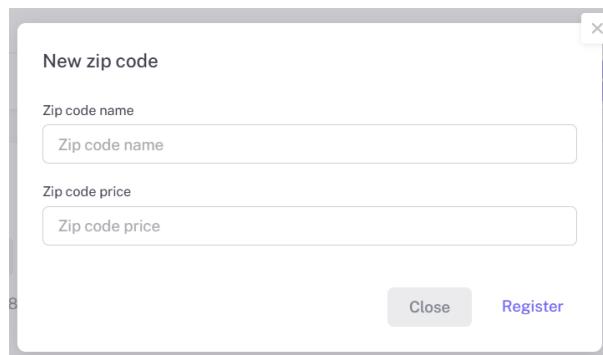


Figura 67 Modal para registro de un nuevo zip\_code

ZIP CODE	PRICE /LABOR	STATUS
92804	260.19	Active
92805	240.01	Active
77096	280	Active
78616	270	Active
121212221212	999	Active

Figura 68 Modal con listado de zip\_codes

Para el proceso de restablecimiento de contraseña se sigue la siguiente lógica:

1. **Evento de Click:** El script espera a que se haga clic en el botón primario (.btn-primary) dentro del modal #newResetPassTechModal.
2. **Obtención del ID de Usuario:** Cuando se hace clic, se obtiene el valor del campo de entrada #newResetPassTech, que contiene el ID del técnico cuya contraseña se desea reiniciar.
3. **Verificación del ID de Usuario:** Si el userId no está presente (es decir, no se ha seleccionado ningún técnico), se muestra un mensaje de error mediante un toast utilizando toastr.error y el proceso se detiene con return.
4. **Confirmación de Usuario:** Si el userId está presente, se muestra un cuadro de confirmación preguntando al usuario si realmente quiere reiniciar la contraseña del técnico.
5. **Petición AJAX:** Si el usuario confirma, se realiza una petición AJAX a la URL especificada (base\_url + 'Ntsdata/Settings/Csettings/reset\_tech\_password') con el método POST. Se envía el userId como dato en la petición.
6. **Manejo de la Respuesta del Servidor:**
  - **Éxito:**

- Si la respuesta del servidor indica éxito (`response.success`), se cierra el modal `#newResetPassTechModal` y se muestra un toast de éxito con `toastr.success`.
- **Error:**
  - Si la respuesta indica un error, se muestra un toast de error con el mensaje proporcionado por el servidor (`response.message`).

**7. Manejo de Errores de la Petición AJAX:** Si ocurre un error durante la petición AJAX (fallo de red, problema en el servidor, etc.), se muestra un toast de error general con `toastr.error`.

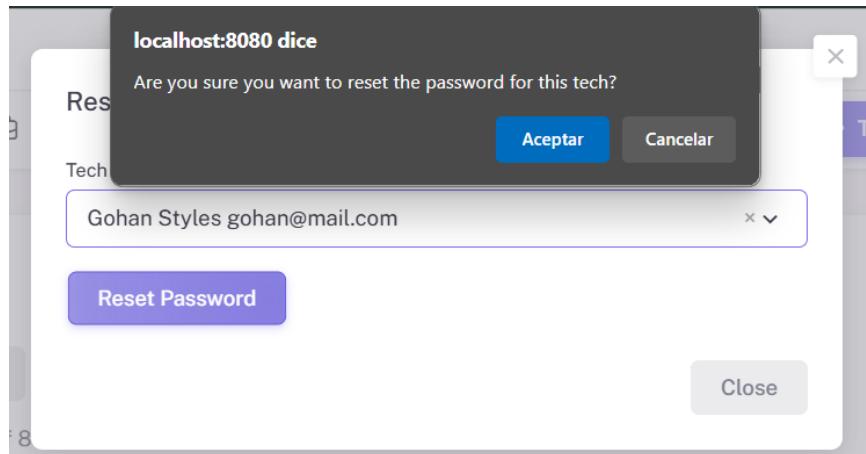


Figura 69 Modal para restablecimiento de contraseña con alerta para confirmación

La lógica para el proceso de restablecer la contraseña dentro del archivo de controlador debe tomar en cuenta los siguientes puntos:

1. **Recepción del ID de Usuario:**
  - El script comienza obteniendo el `id_usuario` enviado a través de una solicitud POST.
2. **Validación del ID de Usuario:**
  - Se verifica que el `userId` sea numérico y mayor que cero.
  - Si no cumple con estas condiciones, se devuelve una respuesta JSON indicando que el ID de usuario no es válido y el proceso se detiene.

**3. Consulta del Usuario:**

- Se consulta la base de datos para obtener los datos del usuario con el ID proporcionado.
- Si no se encuentra un usuario con ese ID, se devuelve una respuesta JSON indicando que el usuario no fue encontrado y el proceso se detiene.

**4. Generación de Contraseña Temporal:**

- Se genera una contraseña temporal utilizando el método generate\_random\_password.

**5. Actualización de la Contraseña:**

- La contraseña temporal se convierte en un hash utilizando password\_hash para mayor seguridad.
- Se actualiza la contraseña del usuario en la base de datos.
- Si la actualización falla, se devuelve una respuesta JSON indicando el fallo y el proceso se detiene.

**6. Envío de la Contraseña Temporal por Correo Electrónico:**

- Si la actualización de la contraseña es exitosa, se envía un correo electrónico al usuario con la nueva contraseña temporal.
- Si el envío del correo falla, se devuelve una respuesta JSON indicando el fallo en el envío del correo.
- Si el correo se envía exitosamente, se devuelve una respuesta JSON indicando que la contraseña se ha reiniciado exitosamente y se debe revisar el correo electrónico para la nueva contraseña.

Como método de prueba de envío de correos electrónicos se utiliza la plataforma mailtrap, la cual otorga un ambiente para realizar las pruebas de envío y recepción necesarias, así como tutoriales y configuraciones dependiendo la tecnología y framework a usar, en el caso de Codeigniter solicita mantener las siguientes configuraciones:

Tabla 27 Configuración para el envío de correos desde codeigniter

Función send_email	
Código	Explicación
<pre>\$config = array(     'protocol' =&gt; 'smtp',     'smtp_host' =&gt;     'sandbox.smtp.mailtrap.io',     'smtp_port' =&gt; 2525,     'smtp_user' =&gt; 'a83be05a6d92dd',     'smtp_pass' =&gt; '2b8bacd7b0422d',     'smtp_crypto' =&gt; 'tls',     'mailtype' =&gt; 'html',     'charset' =&gt; 'utf-8',     'wordwrap' =&gt; TRUE,     'crlf' =&gt; "\r\n",     'newline' =&gt; "\r\n",     'charset' =&gt; 'iso-8859-1', );</pre>	<p><b>Protocolo:</b> 'smtp' - Se define el protocolo para enviar correos.</p> <p><b>Servidor SMTP:</b> 'sandbox.smtp.mailtrap.io' - Especifica el servidor SMTP a utilizar.</p> <p><b>Puerto SMTP:</b> 2525 - El puerto en el que se comunicará con el servidor SMTP.</p> <p><b>Usuario SMTP:</b> 'a83be05a6d92dd' - Nombre de usuario para autenticarse en el servidor SMTP.</p> <p><b>Contraseña SMTP:</b> '2b8bacd7b0422d' - Contraseña para autenticarse en el servidor SMTP.</p> <p><b>Criptografía SMTP:</b> 'tls' - Utiliza TLS para asegurar la conexión.</p> <p><b>Tipo de Correo:</b> 'html' - Define que el correo será en formato HTML.</p> <p>Conjunto de Caracteres: 'utf-8' y 'iso-8859-1' - Especifica el conjunto de caracteres a utilizar.</p> <p><b>Ajuste de Palabras:</b> TRUE - Habilita el ajuste automático de líneas.</p> <p><b>Configuraciones de Línea:</b> Define los caracteres para salto de línea y nueva línea.</p>

Al realizar el proceso para el restablecimiento de una nueva contraseña, verificamos el envío de la información para el correo electrónico.

The screenshot shows the Mailtrap interface with the 'Inboxes' tab selected. The main area displays an email titled 'Temporary Password' from 'Reset password <support@ntsdatal.com.mx>' to 'anna.employee@mail.com'. The email was sent 'a few seconds ago'. Below the email, there is a modal window containing the temporary password 'B2E4nQRaIJ' and a message asking the user to log in and change their password as soon as possible. The sidebar on the left lists various sections: Home, Email API/SMTP, Email Testing, Inboxes (selected), Email Marketing, Contacts, Sending Domains, Templates, Billing, and Settings.

Figura 70 Vista de inbox principal para pruebas desde mailtrap

Este proceso elaborado para el tipo de usuario tech, se seguirá retomando a excepción de los modals para código postal. Los demás elementos se incluirán para el tipo de usuario sale, que, por cierto, el diseño desarrollado para un nuevo usuario de tipo sale será retomado de forma idéntica para el usuario de tipo administrador y empleado.

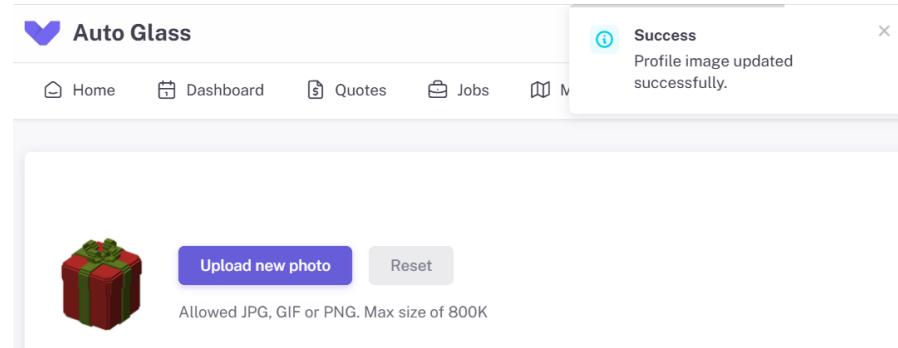


Figura 71 Toast de confirmación de carga de imagen de perfil

## Desafíos y soluciones

- Desafío: Implementación de restablecimiento de contraseña.
- Solución:
  - Validación de Datos: Implementar validaciones estrictas en el servidor para asegurar la integridad de los datos proporcionados.

- Seguridad: Utilizar enlaces de restablecimiento junto con métodos de encriptación robustos para las contraseñas.
- Experiencia del Usuario: Diseñar formularios intuitivos con feedback en tiempo real y mensajes de error claros.
- Desafío: Encriptado de contraseña.
- Solución:
  - Uso de Algoritmos Fuertes: Implementar algoritmos de encriptación seguros como password\_hash para almacenar las contraseñas de manera segura.
  - Actualización Continua: Mantenerse actualizado con las mejores prácticas y mejoras en seguridad para asegurar que las contraseñas estén protegidas contra ataques.
- Desafío: Subir imagen de perfil.
- Solución:
  - Validación y Seguridad: Validar el tipo de archivo y el tamaño antes de aceptar la subida para evitar archivos maliciosos.
  - Experiencia del Usuario: Proporcionar una interfaz fácil de usar para la subida de imágenes, incluyendo previsualizaciones y mensajes claros en caso de error.



## CAPITULO V: REVISION Y RETROSPECTIVA

## CAPITULO V – REVISION Y RETROSPECTIVA

El proyecto consiste en la incorporación de funcionalidades especializadas para la solución de problemas específicos, obteniendo una mejora en la eficiencia y gestión operativa. Las funcionalidades planeadas incluyen:

### **Sprint 1: Módulo de cotizaciones**

Objetivo: Implementar la funcionalidad de cotización en línea de parabrisas nuevos para distintos tipos de vehículos, basándose en las características y precios definidos por el cliente.

Presentación: El equipo demostró cómo los clientes pueden solicitar cotizaciones en línea y cómo se generan work orders.

#### **Feedback Recibido:**

- Positivo: El asesor empresarial hace hincapié a la facilidad para obtener cotizaciones rápidas y precisas.
- Mejoras sugeridas: Incluir opciones para personalizar las cotizaciones y añadir un historial de cotizaciones para los clientes, así como la generación de documentos pdf.

### **Sprint 2: Módulo de trabajos**

Objetivo: Crear un sistema de registro y seguimiento de trabajos, permitiendo registrar los trabajos y hacer un seguimiento del avance de las tareas asignadas al personal interno de la empresa. Cada tarea podrá ser actualizada en función de su estado de avance.

Presentación: Se presentó el sistema de registro de trabajos, mostrando cómo los técnicos pueden registrar el progreso y los supervisores pueden seguir el estado de los trabajos.

#### **Feedback Recibido:**

- Positivo: El asesor empresarial valoró la visibilidad del progreso en tiempo real.

- Mejoras sugeridas: Añadir alertas automáticas para tareas próximas a vencer y mejorar la interfaz de entrada de datos para los técnicos.

### Sprint 3: Módulo de mapas

Objetivo: Implementar la visualización interactiva de sucursales en un mapa, permitiendo configurar sus ubicaciones y visualizarlas en un mapa interactivo.

Presentación: Se mostró la visualización interactiva de sucursales en un mapa, permitiendo a los usuarios localizar rápidamente las sucursales.

#### ***Feedback Recibido:***

- Positivo: El asesor empresarial valoró la interfaz intuitiva y la rapidez en la carga del mapa.
- Mejoras sugeridas: Incluir filtros para buscar sucursales por servicios específicos y mejorar la precisión de las ubicaciones.

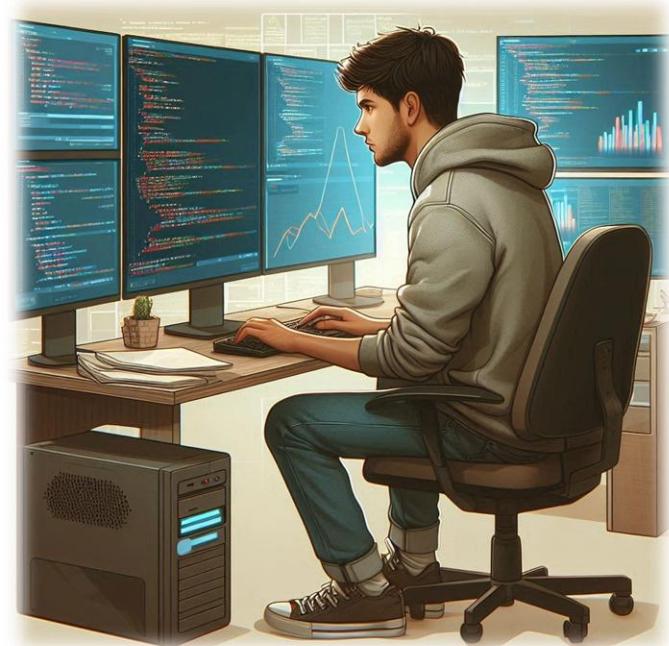
### Sprint 4: Módulo de configuraciones

Objetivo: Implementar un panel de configuración para gestionar detalles relacionados con distribuidores, técnicos ("tech"), personal de ventas ("sale"), usuarios y empleados, ofreciendo flexibilidad para administrar y modificar la información de cada grupo.

Presentación: Esta funcionalidad permitirá la gestión y visualización de sucursales, así como la administración de detalles administrativos o de usuarios.

#### ***Feedback Recibido:***

- Positivo: El asesor empresarial aceptó la interfaz intuitiva y las alertas personalizadas para una mejor comprensión.
- Mejoras sugeridas: Incluir una sección para asignación de permisos que tendrá cada tipo de rol para una mayor gestión y control de los usuarios.



## CAPITULO VI: LANZAMIENTO

## CAPITULO VI – LANZAMIENTO

### Entrega del incremento

#### Sprint 1: Módulo de cotizaciones

##### Entrega

- Funcionalidad de Cotización en Línea: Se entregó una funcionalidad que permitía a los clientes solicitar cotizaciones en línea y generar work orders con los precios detallados.
- Publicación: La funcionalidad se desplegó en el entorno de pruebas y se realizaron demostraciones a los representantes de ventas para validación.

#### Sprint 2: Módulo de trabajos

##### Entrega

- Sistema de Registro y Seguimiento de Trabajos: Se entregó un sistema que permitía a los técnicos registrar el progreso de los trabajos y a los supervisores seguir el estado de todos los trabajos en curso.
- Publicación: La funcionalidad se desplegó en el entorno de pruebas y se realizaron pruebas de aceptación por parte de los supervisores.

#### Sprint 3: Módulo de mapas

##### Entrega

- Implementación del Mapa Interactivo: Se entregó un mapa interactivo que mostraba las sucursales de la empresa. Los usuarios podían interactuar con el mapa para ver detalles de cada sucursal.
- Publicación: La funcionalidad se desplegó en el entorno de pruebas y se realizaron pruebas para validación de ubicación.

#### Sprint 4: Módulo de configuraciones

##### Entrega

- **Panel de Configuración:** Se implementó un panel de configuración para gestionar detalles relacionados con distribuidores, técnicos ("tech"), personal de ventas ("sale"), usuarios y empleados, ofreciendo flexibilidad para administrar y modificar la información de cada grupo.

- **Publicación:** La funcionalidad se desplegó en el entorno de pruebas y se realizaron demostraciones a los stakeholders para validación.

Para garantizar la correcta implementación de las nuevas funcionalidades, se realizaron pruebas de integración que aseguraron el funcionamiento conjunto de todos los módulos. Además, se optimizó el rendimiento para minimizar el consumo de recursos, mejorando así la eficiencia y la gestión operativa.

## CONCLUSIONES

La creación y puesta en marcha de las nuevas funcionalidades se realizó empleando una metodología ágil, lo que facilitó una adaptación rápida y adaptable a las variaciones y requerimientos que surgían a lo largo del proyecto. La selección de esta metodología resultó ser sumamente ventajosa, promoviendo la cooperación constante entre los distintos equipos y posibilitando entregas progresivas que se pudieron evaluar y modificar en cada sprint.

### Comparativa entre Planificación Inicial y Resultado Final

Desde el inicio de la planificación, la meta era incrementar la eficiencia y administración operativa a través de la incorporación de módulos particulares: cotizaciones, tareas, mapas y configuraciones. En el escenario práctico, el resultado final no solo alcanzó estas metas, sino que también sobrepasó las expectativas en diversas áreas, gracias a la retroalimentación constante durante las etapas de evaluación y confirmación.

- **Módulo de Cotizaciones:** Inicialmente diseñado para realizar cotizaciones en línea, finalmente incluyó la personalización de las cotizaciones y el historial, optimizando de manera significativa la experiencia del cliente.
- **Módulo de Trabajos:** Se expandió para incorporar advertencias automáticas y una interfaz optimizada para la entrada de datos, lo que mejoró la administración del trabajo cotidiano y la supervisión.
- **Módulo de Mapas:** Además de la visualización interactiva, se implementaron filtros para una búsqueda más exacta, optimizando así la eficiencia de la búsqueda.
- **Módulo de Configuraciones:** Ofreció una gestión flexible y detallada de usuarios y roles, lo cual no estaba originalmente previsto con tanto detalle.

### Valoración de Resultados Efectivos

- **Eficiencia Operativa Mejorada:** La implementación de estos módulos ha permitido a la empresa gestionar sus operaciones de manera más eficiente, reduciendo tiempos de respuesta y mejorando la calidad del servicio.
- **Experiencia del Usuario Mejorada:** La interfaz intuitiva y las funcionalidades adicionales han sido bien recibidas por los usuarios, quienes han expresado satisfacción con las mejoras introducidas
- **Flexibilidad y Adaptabilidad:** La capacidad de personalizar y ajustar configuraciones según las necesidades específicas ha permitido una mayor adaptabilidad a los cambios y requisitos del mercado.

### **Beneficios y Ventajas**

Entre los beneficios y ventajas del uso del nuevo software, se destacan:

- **Reducción de Errores:** La automatización de procesos y el seguimiento en tiempo real han reducido significativamente los errores humanos.
- **Aumento de Productividad:** La visibilidad del progreso y las alertas automáticas han permitido una gestión más proactiva, incrementando la productividad del personal.
- **Satisfacción del Cliente:** La capacidad de obtener cotizaciones rápidas y precisas, junto con una mejor gestión de las tareas y sucursales, mejorara la satisfacción general del cliente.

## RECOMENDACIONES

Para maximizar el rendimiento del software, se sugiere configurar lo siguiente, y se recomienda el conocimiento de lo siguiente:

- **Sesiones de Capacitación:** Organizar sesiones de capacitación iniciales y periódicas para asegurar que todos los usuarios comprendan cómo utilizar las funcionalidades del software de manera eficaz.
- **Manual de Usuario:** Proveer un manual de usuario detallado que incluya instrucciones paso a paso y respuestas a preguntas frecuentes.
- **Soporte Técnico:** Establecer un canal de soporte técnico accesible para resolver cualquier duda o problema técnico que pueda surgir durante el uso del software.

## Configuraciones y Conocimientos Tecnológicos

Para maximizar el rendimiento del software, se sugieren las siguientes configuraciones y conocimientos tecnológicos:

- **Configuración del Sistema:** Asegurarse de que los dispositivos que utilicen el software cumplan con los requisitos mínimos de hardware y software especificados.
- **Actualizaciones Regulares:** Mantener el software actualizado con las últimas versiones para aprovechar las mejoras y correcciones de errores.
- **Conocimiento Tecnológico Básico:** Capacitar a los usuarios en conocimientos tecnológicos básicos, como la navegación en interfaces web y el manejo de archivos digitales, para garantizar un uso fluido del software.

## Alcance Actual y Expansión Futura

Es importante reconocer el alcance actual del software y planificar su expansión futura para maximizar sus beneficios:

- **Alcance Actual:** El software actualmente abarca módulos de cotizaciones, trabajos, mapas y configuraciones, mejorando la eficiencia operativa y la experiencia del usuario.

- **Expansión Futura:** Para expandir el alcance del software, se pueden considerar las siguientes acciones:
  - **Integración con Otros Sistemas:** Evaluar la posibilidad de integrar el software con otros sistemas de la empresa para crear un ecosistema digital más cohesivo.
  - **Nuevas Funcionalidades:** Identificar y desarrollar nuevas funcionalidades que puedan atender necesidades emergentes de los usuarios o del mercado.
  - **Expansión Geográfica:** Considerar la expansión del uso del software a otras sucursales o regiones, asegurando que los ajustes y configuraciones sean adecuados para diferentes contextos operativos.

## BIBLIOGRAFIA Y LITERATURA CONSULTADA

- B., G. (03 de 09 de 2024). Hostinger Tutoriales. Obtenido de ¿Qué es MySQL? Explicación detallada para principiantes:  
<https://www.hostinger.mx/tutoriales/que-es-mysql#:~:text=En%20cuanto%20a%20la%20definici%C3%B3n%20general,%20MySQL%20es%20un%20sistema#:~:text=En%20cuanto%20a%20la%20definici%C3%B3n%20general,%20MySQL%20es%20un%20sistema>
- Chirinos, J. (28 de 03 de 2021). *MundoBytes.com*. Obtenido de Qué Es Xampp Usos, Características, Opiniones, Precios:  
<https://mundobytes.com/xampp/#:~:text=Xampp%20es%20uno%20de%20los%20servidores%20web%20multiplataforma%20m%C3%A1s%20utilizados,>
- Cuadrado, G. C. (22 de 07 de 2022). *OpenWebinars*. Obtenido de Visual Studio Code: Editor de código para desarrolladores:  
<https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/#:~:text=Visual%20Studio%20Code%20es%20un%20editor%20de%20c%C3%B3digo%20gratuito,%20multiplataforma#:~:text=Visual%20Studio%20Code%20es%20un%20editor%20de%20c%C3%B3digo%20gratuito>
- Javier. (29 de 01 de 2024). *Formadores IT*. Obtenido de ¿Qué es Gitlab y para qué sirve?: <https://formadoresit.es/que-es-gitlab-y-para-que-sirve/>
- MDN. (20 de 10 de 2024). *Mdn web docs*. Obtenido de ¿Qué es JavaScript?:  
[https://developer.mozilla.org/es/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript)
- Muradas, Y. (03 de 06 de 2019). *OpenWebinars*. Obtenido de Qué es Postman y primeros pasos: [https://openwebinars.net/blog/que-es-postman/#:~:text=Qu%C3%A9%20es%20Postman.%20Postman%20en%20sus%20inicios%20nace%20como%20una](https://openwebinars.net/blog/que-es-postman/#:~:text=Qu%C3%A9%20es%20Postman.%20Postman%20en%20sus%20inicios%20nace%20como%20una#:~:text=Qu%C3%A9%20es%20Postman.%20Postman%20en%20sus%20inicios%20nace%20como%20una)
- Souza, I. d. (09 de 03 de 2020). *RockContent*. Obtenido de Descubre qué es el lenguaje de programación PHP y en qué situaciones se hace útil:  
[https://rockcontent.com/es/blog/php/#:~:text=PHP%20es%20un%20lenguaje%20de%20programaci%C3%B3n%20destinado%20a%20desarrollar%20aplicaciones](https://rockcontent.com/es/blog/php/#:~:text=PHP%20es%20un%20lenguaje%20de%20programaci%C3%B3n%20destinado%20a%20desarrollar%20aplicaciones#:~:text=PHP%20es%20un%20lenguaje%20de%20programaci%C3%B3n%20destinado%20a%20desarrollar%20aplicaciones)

## GLOSARIO

### “A”

**Ajax:** Técnica de desarrollo web que permite la actualización de una parte de una página web sin necesidad de recargarla completamente.

**Arreglo:** Estructura de datos que almacena una colección de elementos del mismo tipo.

### “B”

**Backlog:** Lista priorizada de tareas pendientes por realizar en un proyecto, especialmente utilizado en metodologías ágiles como Scrum.

**Booleano:** Tipo de dato que tiene dos posibles valores: verdadero (true) o falso (false).

### “C”

**Controlador:** Componente en el patrón de arquitectura MVC (Modelo-Vista-Controlador) que maneja la lógica de la aplicación.

**Codeigniter:** Framework de desarrollo web en PHP que facilita la creación de aplicaciones dinámicas y robustas.

**CRUD:** Acrónimo que representa las operaciones básicas en bases de datos: Crear, Leer, Actualizar y Eliminar (Create, Read, Update, Delete).

### “D”

**Datatable:** Tabla de datos interactiva que permite la visualización, filtrado y manipulación de grandes conjuntos de datos.

### “F”

**Framework:** Conjunto de herramientas y librerías que proporcionan una estructura base para el desarrollo de aplicaciones de software.

**Feedback:** Es la información que se recibe sobre el desempeño o resultado de una acción o tarea. En informática, el feedback es crucial para el desarrollo y mejora continua del software. Puede provenir de usuarios, clientes, pruebas y evaluaciones internas, y se utiliza para identificar áreas de mejora, errores, y para ajustar el producto a las necesidades y expectativas de los usuarios.

### “I”

**Iterar:** Repetir un proceso o una serie de instrucciones en un bucle hasta cumplir una condición.

**“L”**

**Latitud:** Coordenada geográfica que especifica la posición norte-sur de un punto en la superficie de la Tierra.

**Longitud:** Coordenada geográfica que especifica la posición este-oeste de un punto en la superficie de la Tierra.

**Lanzamiento:** Se refiere a la distribución oficial de una nueva versión de un software o producto. Este proceso incluye la preparación del software para su distribución, la actualización de la documentación y, a menudo, la notificación a los usuarios sobre las nuevas características y mejoras. Un lanzamiento puede ser una nueva versión principal (major), una actualización menor (minor) o una corrección de errores (patch).

**“M”**

**Modal:** Ventana emergente en una interfaz de usuario que se superpone a la ventana principal y requiere una acción del usuario para cerrarla.

**Modelo:** Componente en el patrón de arquitectura MVC que maneja los datos y la lógica de negocio de la aplicación.

**“R”**

**Redireccionamiento:** Acción de enviar a un usuario desde una URL a otra diferente, generalmente por razones de mantenimiento o reestructuración de un sitio web.

**Retorna:** Acción de devolver un valor desde una función o método en programación.

**Retrospectiva:** Una retrospectiva es una reunión que forma parte de las metodologías ágiles (como Scrum) donde el equipo reflexiona sobre el trabajo realizado durante un sprint o un período específico. El objetivo es identificar lo que funcionó bien, lo que no funcionó y cómo pueden mejorar los procesos y prácticas para futuros sprints. Se enfoca en la mejora continua y la colaboración del equipo.

**Restablecimiento:** Se refiere a la acción de devolver un sistema, dispositivo o aplicación a su estado original o de fábrica. Esto puede incluir borrar datos, deshacer cambios recientes o restaurar configuraciones predeterminadas. Se utiliza a menudo para resolver problemas técnicos o preparar un dispositivo para un nuevo usuario.

**“S”**

**Script:** Conjunto de instrucciones de programación que se ejecutan en un entorno específico, como un navegador web.

**Scrum:** Metodología ágil para la gestión y desarrollo de proyectos, especialmente en el ámbito del software.

**Sintaxis:** Conjunto de reglas que definen la estructura correcta de las instrucciones en un lenguaje de programación.

**Sprint:** Periodo de tiempo en Scrum durante el cual se completan un conjunto de tareas específicas del proyecto.

**Stakeholders:** Partes interesadas o involucradas en un proyecto, como clientes, usuarios, y miembros del equipo.

**“T”**

**Tax:** Impuesto, carga financiera impuesta por el gobierno sobre productos, ingresos o actividades.

**Template:** Plantilla o estructura base que sirve de guía para crear otros documentos o componentes.

**“V”**

**Vista:** Componente en el patrón de arquitectura MVC que se encarga de presentar los datos y la interfaz al usuario.