



UNIVERSIDAD DE SONORA

División de Ingeniería

Departamento de ingeniería industrial

Base de datos para clientes ALSO Montacargas,
Prácticas realizadas en ALSO Montacargas/Refacciones

PRESENTA:

JOEL ALFONSO PEREZ DIAZ

215206080

REPORTE DE PRACTICAS PROFESIONALES

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

TUTOR:

JOSE LUIS OCHOA HERNÁNDEZ

Enero 2020

ÍNDICE

INDICE DE ILUSTRACIONES	3
1. INTRODUCCIÓN	4
1.1. Explicación del proyecto	4
1.2 Objetivos generales	4
1.3 Línea Metodológica.	5
2. Descripción del contexto	6
2.1 Equipamiento e instalaciones	7
2.2 descripción de la normatividad o reglas de operación del programa o unidad receptora	9
2.3 Entorno	10
3. FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS	10
Herramientas Utilizadas	11
4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES REALIZADAS	14
Requisitos	15
CREACIÓN DE LA BASE DE DATOS	16
CREACIÓN DE LA MIGRACIÓN	17
CREACIÓN DEL MODELLO	19
CREACIÓN DE LAS RUTAS Y CONTROLADORES	20
CONTROLADORES	21
CONFIGURACIÓN DE BOOTSTRAP	21
GENERANDO LAS VISTAS	22
5-ANÁLISIS DE LA EXPERIENCIA ADQUIRIDA	25
5.1- Análisis general del programa, su diseño, desarrollo y organización.	25
5.2 Análisis de los objetivos del programa	25
5.3 Análisis de las actividades realizadas.	26
5.4 Análisis de la metodología utilizada	26
6- CONCLUSIONES Y RECOMENDACIONES	27
7 ANEXOS	28

ÍNDICE DE ILUSTRACIONES

Ilustración 2 1 ubicación de la unidad receptora	6
Ilustración 2 2 Instalaciones de la empresa	7
Ilustración 3 1 Controlador de la aplicación	13
Ilustración 4 1 Base de datos de clientes	17
Ilustración 4 2 migración de la base de datos	18
Ilustración 4 3 Atributos de tabla clientes	18
Ilustración 4 4 Atributos de tabla permisos.....	18
Ilustración 4 5 Creación del modelo	19
Ilustración 4 6 Modelo para roles	20
Ilustración 4 7 Modelo para permisos.....	20
Ilustración 4 8 instalación de Bootstrap.....	22
Ilustración 4 9 instalación de Vue.....	22
Ilustración 4 10 Vistas de clientes	23
Ilustración 4 11 Índex de clientes	23
Ilustración 4 12 Crear cliente.....	24
Ilustración 4 13 Vistas de la aplicación.....	24

ÍNDICE DE TABLAS

Tabla 2.1 1 Equipamiento de la empresa	9
Tabla 4 1 Diagrama de Gantt de actividades	16

1. INTRODUCCIÓN

1.1. Explicación del proyecto

Se realizó una aplicación web en donde se registran y clasifican todos los clientes y prospectos de la empresa ALSO MONTACARGAS ubicada en Salamanca 117, las Amapolas Hermosillo Sonora #8.

En la aplicación se pueden registrar, importar, filtrar, eliminar, exportar clientes/prospectos que se desean, este proyecto surge debido a la necesidad de la empresa de extender su mercado y crecer, no solo en Hermosillo si no en todo el estado y país. Anteriormente la empresa guardaba a sus clientes en hojas de Excel lo cual era muy confuso porque se creaban demasiados archivos e inseguro debido a la posible pérdida de datos, si bien contaban con un archivo en Google Drive donde se guardaban las empresas, existía la problemática que los vendedores repetían clientes, cualquiera con el link podía modificar los datos, no se pueden exportar datos lo cual era una tarea tediosa estar ingresando los datos uno por uno.

1.2 Objetivos generales

El objetivo principal es hacer que la empresa crezca, de manera que teniendo un registro de todos los clientes y/o prospectos, datos de ellos tales como teléfono, dirección, contacto etc. los vendedores puedan tener una idea de las empresas a las cuales puedan contactar para ofrecer los productos o para dar un seguimiento los negocios que ya son clientes. Además, de esta manera es mucho más fácil buscar prospectos porque se pueden filtrar según su giro (industrial, minero agrícola, etc.) o por otros atributos como la ciudad, estado etc.

Otro de los objetivos es dar un orden a todos los archivos los cuales se manejan en la empresa ya que estos son demasiados y pueden llegar a confundir a la asistente encargada del manejo de la base de datos ya que por lo regular se le pasaban los clientes en diferentes archivos y se encargaba de ingresarlos uno por uno.

1.3 Línea Metodológica.

El proyecto se desarrolló PHP utilizando el Framework Laravel. como gestor de base de datos se utilizó MySQL. Esto debido a que es más sencillo encontrar documentación en este lenguaje para el tipo de aplicación que se requería, a medida que se avanzaba en el proyecto existía un encargado de revisar los avances, Alan Flores, al revisar el proyecto el decidía si se quitan o agregan propiedades de este. Como por ejemplo agregar la opción de importar o exportar datos, creación de usuarios que puedan acceder a la aplicación, etc.,

El tiempo que se tomó en realizar el proyecto fueron dos meses, ya que siempre se agregan nuevos elementos, además de trabajar en otros proyectos como en la administración del CRM llamado "Gestino" en el cual se basó bastante para la creación de la aplicación ya que sirvió como ejemplo para tomar nuevas ideas.

Al terminar, el proyecto se subió a un servidor propio de la empresa, se explicó a los vendedores y asistentes cómo funcionaba se les preguntó si deseaban agregar nuevos elementos u opciones los cuales dieron sus opiniones.

2. Descripción del contexto

ALSO MONTACARGAS es una empresa que se dedica a la venta, renta y servicio de Montacargas, Patines así como refacciones de estos. Fundada en la ciudad de Hermosillo Sonora, ALSO inició sus operaciones en abril del 2007 por lo que cuenta casi 13 años en el mercado, actualmente se encuentra ubicada en la colonia Amapolas en la calle Salamanca #8 local 12, se busca actualmente expandirse a otras ciudades del mismo estado como Caborca, Obregón, Nogales etc.

La empresa cuenta con varios clientes ya que ofrece el mejor servicio a tiempo, con buenos resultados por lo que cada día va creciendo, entre las marcas de patines y montacargas que distribuye la empresa son RED, LIFT, ROYAL. UTILEV.

Ubicación

La empresa está ubicada en un local donde se encuentran varias bodegas, el espacio es amplio ya que en pueden ingresar varios Montacargas, patines, entre equipos y refacciones.



Ilustración 2 1 ubicación de la unidad receptora



Ilustración 2 2 Instalaciones de la empresa

También cuenta con refaccionaria propia con una amplia gama de productos originales, su sistema de almacenamiento permite resolver cualquier exigencia de manera oportuna.

2.1 Equipamiento e instalaciones

En cuanto a recursos la empresa contaba con pocos dispositivos.

Dispositivo/Software/Tecnología	Especificaciones	Uso
3 monitores AOC	Están conectados a un CPU de diferente marca, cada vendedor tiene su propia computadora de escritorio	Monitor
Paquete office 2013	<ul style="list-style-type: none"> • versión 2013 	<ul style="list-style-type: none"> • Almacena información de clientes y prospectos

		<ul style="list-style-type: none"> ● Elaboración de reportes ● Generación de gráficas para obtener resultados de cada vendedor
Servidor	<ul style="list-style-type: none"> ● Servidor HPE Proliant ● Compartido ● Sistema Operativo Linux ● 1.70 GHZ ● 96TB ● 16GB DDR4 	Almacenar información de la empresa
7 computadoras Lenovo	<ul style="list-style-type: none"> ● Lenovo M81 C15 SFF ● Disco duro de 500 GB ● 4GB DE RAM ● Algunas se les cambió por el monitor AOC 	Los vendedores las utilizan diariamente para cotizar, encontrar clientes, guardar archivos etc
Gestiono	<ul style="list-style-type: none"> ● CRP ● En proceso de ser implementado ● Proveedores en la ciudad de Monterrey 	<ul style="list-style-type: none"> ● Almacena inventario ● Operaciones de contabilidad ● Facturas

GotoMeeting	<ul style="list-style-type: none"> • Servicio web para reuniones en línea 	<ul style="list-style-type: none"> • Reuniones semanales con consultores para ver el estado de la empresa
Google Analytics		<ul style="list-style-type: none"> • Informa acerca de los posibles clientes en la página
Photoshop	<ul style="list-style-type: none"> • Versión 2016 	<ul style="list-style-type: none"> • Generación de carteles • Flyers

Tabla 2.1 1 Equipamiento de la empresa

2.2 descripción de la normatividad o reglas de operación del programa o unidad receptora

La norma de la empresa es distinta para cada empleado, cada uno de ellos cumple con tareas distintas pero todos tienen un propósito en común, el de brindar muy buen servicio al cliente en tiempo y forma. Entre las normas generales de la empresa está el uso de botas, zapato grueso ya que se maneja con materiales de gran volumen, el uso de pantalones y camisa formales ya cuenta con uniforme propio.

Algo destacable es que se rigen bajo la metodología o filosofía Gung-Ho es una forma de trabajo en la que la empresa alienta a sus trabajadores para generar desempeño, la empresa pone metas, entrega bonificaciones a quienes la logran y se trata a todos de manera igual lo que hace que se cree una buena comunicación de los trabajadores con cargos más altos.

Los vendedores operan de manera muy particular, se les entrega una lista prospectos de los cuales tienen que realizar 5 visitas, 10 llamadas y 15 correos, estas listas se sacan de archivos de Excel es por eso que se procedió hacer el proyecto de la base de datos, existe una asistente y 5 vendedores actualmente, la asistente es la

encargada de crear estas listas para posteriormente entregárselas a los vendedores, esta es la tarea que deben cumplir los vendedores en el transcurso del día. Al final del día estos vendedores anotan los resultados en un pizarrón. Una vez finalizada la semana se crean listas para ver el desempeño de cada uno así como el de la empresa en general. Es importante saber que anteriormente muchas de las empresas que se encontraban en los archivos se repetían lo que ocasiona problemas en los cientos ya que se les llamaba en más de una ocasión entre otras problemáticas.

Otra de las características de la empresa es que trabajan con un sistema llamado Monday, en él apuntan las actividades, oportunidades y desperdicios que tiene cada equipo de trabajo y se evalúan al final de la semana en una conferencia en línea con un consultor, KM solución.

2.3 Entorno

Una de las principales ventajas de la empresa es el hecho de que tiene una variedad de productos y servicios, estos, son dedicados a empresas que los utilizan constantemente y que no tienen otra opción, por esta razón la empresa ya cuenta con clientes fijos a los que se atiende de manera oportuna. Es una de las pocas empresas que ofrece este tipo de servicios por lo que existen muchas empresas y prospectos por visitar y convertir en clientes tanto en la ciudad como en el estado. Es por ello que la oportunidad de crecimiento es muy grande y es la razón por la que la empresa próspera y también es la razón por la que requiere muchas soluciones tecnológicas, una de ellas es la manera en la que manejan sus datos.

3. FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS

Para resolver la problemática de la empresa. El almacenamiento de datos de clientes y prospectos, se buscaron diversas alternativas ya que no era seguro ni práctico la manera en la que se realizaba. la cual, consistía en manejar diversos archivos de Excel y Word. Para esto, se optó por hacer una aplicación web ligada a una base de datos con diversos campos primordiales para que la empresa pueda contactar a sus clientes y prospectos y respaldados de manera segura.

La aplicación debería tener las siguientes características.

- Base de datos en donde almacenar los clientes y/o prospectos.
- Login para los usuarios, ya sea los vendedores, la asistente o el encargado.
- Opción de administrador de crear nuevos usuarios, crear permisos, crear roles.
- Opción de exportar e importar datos.
- Crear, eliminar, modificar clientes operaciones básicas CRUD.
- Opción de guardar clientes VIP (clientes con más de 500 empleados)
- Filtrado por búsqueda, nombre, giro, contacto.
- Campos necesarios con la información primordial y paginados de 100

Para llevar a cabo dichas actividades se utilizaron diversas tecnologías y metodologías las cuales se explican a continuación:

Herramientas Utilizadas

HTML5/CSS

Se utilizan para hacer el diseño de la aplicación, las tablas, las posiciones de los botones, para ello que existe una librería o Framework muy popular llamado Bootstrap la cual hace más sencillo el diseño ya que tiene elementos predeterminados los cuales están basados en estándares para el buen diseño de una aplicación o página web.

PHP

Lenguaje de programación web para realizar el back-end de la aplicación, todas las operaciones lógicas de esta, tales como la conexión de la base de datos, las sentencias o Querys necesarios para mostrar determinados clientes, la versión que se utilizó de PHP es la última que existe, la versión 7.

Una de las decisiones que influyó en utilizar este lenguaje es existe mucha documentación para el tipo de sistema que se quería hacer, además utilizar la herramienta phpMyAdmin que es la que se utiliza para manejar MySQL a través de páginas web.

MySQL

Gestor de base de datos relacional, la cual es una de las más usadas al desarrollar una aplicación web, tiene contenido dinámico y es más sencilla de usar a comparación de otros gestores de base de datos , se optó por esta opción por su simplicidad, disponibilidad en gran cantidad de plataformas y sistemas, posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades, soporte físico, transacciones etc, además de ofrecer la posibilidad de generar claves foráneas que son características muy favorables a la hora de realizar un aplicación CRUD (create, read. update, delete).

Laravel

Framework de PHP que facilita la tarea al realizar este tipo de aplicaciones o páginas web ya que por lo regular son muy comunes entre las empresas, facilita la creación de acciones como la creación de un login, la conexión y migración de la base de datos.

Cuenta con soporte MVC (Modelo, Vista, Controlador) el cual funciona de la siguiente manera:

- **Modelo.** Ese controla el bade de datos llamada en este caso “Client” además de generar las consultas a esta para poder filtrar los datos, mostrarlos, eliminarlos etc. Para crear un modelo se usó la terminal escribiendo el siguiente comando:
php artisan make:model cliente
- **Vista.** Se controla en la que se presenta el contenido al usuario, aquí escribimos nuestro código HTML, CSS. Dentro de estos archivos podemos insertar código PHP, por ejemplo si queremos pasar datos a una tabla, a qué ventanas redireccionan ciertos botones etc., Se explica más a detalle en la siguiente sección.

- **Controlador.** Laravel nos ofrece un mecanismo que nos permite agrupar mejor las peticiones HTTP relacionadas para mejorar nuestro código, para generar un controlador utilizamos un comando en la terminal `-php artisan make:Controller ClientsController`, posteriormente en la clase que se crea guardamos métodos para enlazar rutas

```
class ClientsController extends Controller
{
    public function index()
    {
        // metodo para mostrar toda la base de datos
        // pide el permiso, muestra todos los clientes
        abort_unless(\Gate::allows('client_access'), 403);

        $clients = Client::all();

        return view('admin.clients.index', compact('clients'));
    }

    public function create()
    {
        abort_unless(\Gate::allows('client_create'), 403);

        return view('admin.clients.create');
    }
}
```

Ilustración 3 1 Controlador de la aplicación

WAMPSEVER

El uso de WAMP permite subir páginas HTML, además de poder administrar o gestionar datos en ellas, crea una especie de servidor el cual se usó para el proceso de desarrollo del proyecto, además permite el uso de los lenguajes de programación tales como PHP, el gestor de base de datos MySQL, apache como servidor web en el sistema operativo Windows sin necesidad de instalar uno por uno.

Bootstrap

Framework utilizado en el desarrollo web en CSS/Javascript que facilita el diseño de interfaces el cual ofrece múltiples herramientas haciendo que sea posible el diseño responsive y se faciliten tareas tales como el diseño de los botones, las posiciones, diseño de múltiples elementos de la página haciendo que la interfaz sea mucho más amigable.

Visual Studio Code

Editor de código que permite resaltar las sintaxis para que sea más sencillo escribir código, tiene varias características y ventajas tales como finalización inteligente de código, fragmentos y refactorización de código, es perdonable, atajos de teclado, es gratuito y de código abierto, además también permite crear archivos y carpetas y las muestra en un costado de la pantalla por lo que es fácil saber en qué carpetas se está trabajando y cuales se han modificado.

Github

Es una plataforma de desarrollo para trabajar en equipo, ideal para el desarrollo colaborativo, utiliza el sistema de control de versiones Git, se pueden guardar todas las versiones del proyecto desde su inicio y cada modificación por lo que si ocurre un error es fácil regresar o hacer ramificaciones para probar código, además tiene las siguientes características las cuales fueron muy útiles en el proyecto.

- Wiki para cada proyecto
- Página web para cada proyecto
- Gráfico para ver cómo los desarrolladores trabajan en sus repositorios y bifurcaciones del proyecto.
- Herramienta de trabajo colaborativo.

SASS

Esta herramienta fue muy útil porque brinda la posibilidad de convertir los CSS en algo más dinámico, permite trabajar mucho más rápido, su principal ventaja es la posibilidad de compartir los CSS en algo dinámico, permite trabajar mucho más rápido en la generación de Código con la posibilidad de crear algunas funciones, por ejemplo en este proyecto permitió usar variables que podemos manejar desde un sitio de forma sencilla y centralizada,

4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES REALIZADAS

Gráfico de Gantt de las actividades realizadas durante los meses de septiembre a diciembre del 2019

Actividades	20/09	27/09	03/10	17/10	02/10	25/10	04/11	01/12/19
1								
2								
3								
4								
5								
6								
7								
8								

Tabla 4 1 Diagrama de Gantt de actividades

Actividades

1. Análisis de los requerimientos
2. Creación de la base de datos
3. Creación de las migraciones
4. Creación del modelo
5. Configuración de las rutas
6. Creación de las vistas
7. Editar y actualizar la base de datos
8. Retroalimentación

Requisitos

Para empezar a trabajar en el proyecto se requirió tener instalado:

VS CODE editor de texto para escribir y manipular código, WAMP para poder tener el servidor local de pruebas que soporte el lenguaje de programación PHP y el gestor

de base de datos MySQL. Y el Framework Laravel el cual facilita el desarrollo de este tipo de aplicaciones. para instalarlo se corre el siguiente comando en la consola ya sea de Visual Studio o el CMD:

```
composer create-project--prefer-distlaravel/laravel laravel6
```

Una vez ya ejecutado de manera exitosa se procedió a instalar npm con el siguiente comando, esto para desarrollar la interfaz de la aplicación, ya que más adelante se explica cómo se desarrolló la parte de diseño.

```
-Npm install
```

CREACIÓN DE LA BASE DE DATOS

En MySQL se creó una base de datos llamado como el nombre de la empresa “ALSO”. Para crearla fue necesario tener instalado el gestor de base de datos junto con un servidor local capaz de soportar aplicaciones en el lenguaje PHP, es por ese motivo que se instala WAMP, una vez ya instalado e inicializado ingresamos en el navegador “localhost” o “127.0.0.1” aparece lo siguiente:

ingresamos con el nombre de usuario y contraseña que por defecto en ambos casos es “root”.

Una vez ya creada la base de datos es necesario configurar nuestra aplicación para que identifique cual es la base de datos a la que debe migrar la información que ingresamos, para ello fue necesario ir al archivo que se creó al correr el comando. el cual llamamos “Laravel-Crud”. para crear la aplicación. En la carpeta con la extensión. env es se configuraron los datos necesarios, los cuales son el nombre de la base de datos que creamos en MySQL el cual es “ALSO”, el nombre de usuario el cual por defecto es “root” y la contraseña se dejó igual debido a que por defecto no se tenía ninguna. una vez ya creada la base de datos es necesario hacer las migraciones en la aplicación para que aparezca de la siguiente manera:

id	RazonSocial	Giro	Telefono	Direccion	Zona	Ciudad
1464	GEYMSA	Servicios	222	Puebla	Puebla	Puebla
2	COMPAÑIA INDUSTRIAL ASOCIADA SA DE CV	Industrial	5.28E+11	PORFIRIO DIAZ 411 SUR CENTRO	Apodaca	Apodaca
3	COMPAÑIA INDUSTRIAL ASOCIADA SA DE CV	Industrial	5.28E+11	PORFIRIO DIAZ 411 SUR CENTRO	Apodaca	Apodaca
4	GABINA ROMERO RIVEROS	Varios		AZABACHE MZ.42 LOTE 35 PRADERAS DEL POTRERO	Atotonilco De Tula	Atotonilco Tula
5	ELELE S.A. DE C.V.	Agrícola	66 21 28 03 23	Miramar esquina Alvarado 60 Centro	Bahía De Kino	Bahía De Kino
6	DELTA TRUCKING SERVICE S DE RL DE	Corporativo	6197342323	CAMINO ANTIGUO A TECATE 500	Baja California	Baja California

Ilustración 4.1 Base de datos de clientes

CREACIÓN DE LA MIGRACIÓN

Las migraciones son como el control de versión de la base de datos, permite que se comparta fácilmente un esquema de los datos.

Cómo creamos una aplicación de clientes, creamos una migración llamada “Clients”, en ella se ingresan los campos que tendrá la base de datos, es importante aclarar que en esta parte se especifican los campos, el tipo de valor, el tamaño, las propiedades etc.

También se crearon migraciones para usuarios, permisos y roles ya que son entidades aparte del cliente y cada uno tiene diferentes campos diferentes. Es necesario guardar los permisos, los usuarios que utilizarán la aplicación y los roles que se le dará a cada uno.

Para migrar las tablas con los campos y propiedades que deseamos se realiza ejecutando el siguiente comando:

```
php artisan make:migration create_clients_table --create=clients
```

```
php artisan make:migration create_useres_table --create=users
```

```
php artisan make:migration create_permissions_table --create=permissions
```

```
php artisan make:migration create_roles_table --create=roles
```

Una vez ya creadas las migraciones en el archivo aparece lo siguiente:

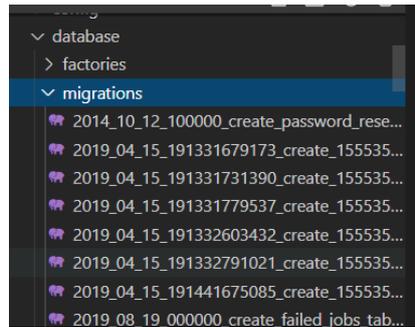


Ilustración 4 2 migración de la base de datos

Cada uno de los archivos es una tabla en la base de datos. una vez ya realizada la migración, en cada uno de los archivos creados (usuarios, permisos, roles, clientes) se escriben los scripts de las propiedades, campos o atributos que se requieren:

```

/**
 * Run the migrations el comando para correr la mi.
 *
 * @return void
 */
public function up()
{
    Schema::create('clientes', function (Blueprint $table) {
        $table->increments('id');
        $table->string('RazonSocial');
        $table->string('Giro');
        $table->string('Telefono');
        $table->string('Direccion');
        $table->string('Zona');
        $table->string('Ciudad');
        $table->string('Estado');
        $table->string('Contacto');
        $table->string('Correo');
        // $table->longText('description')->nullable();
        // $table->decimal('price', 15, 2)->nullable();
        $table->timestamps();
        $table->softDeletes();
    });
}

```

Ilustración 4 3 Atributos de tabla clientes

```

use Illuminate\Support\Facades\Schema;

class Create1555355612603PermissionRolePivotTable extends Migration
{
    public function up()
    {
        Schema::create('permission_role', function (Blueprint $table) {
            $table->unsignedInteger('role_id');
            $table->foreign('role_id')->references('id')->on('roles');
            $table->unsignedInteger('permission_id');
            $table->foreign('permission_id')->references('id')->on('perm');
        });
    }

    public function down()
    {
        Schema::dropIfExists('permission_role');
    }
}

```

Ilustración 4 4 Atributos de tabla permisos

La migración se realiza para guardar los cambios a la base de datos que se realiza en el código, de esa manera no es necesario hacer todos los cambios manualmente en la interfaz de MySQL en PHPMysqlAdmin.

Laravel ya contiene tres migraciones “Users”, “Failed jobs” y “Passwords resets” Cuando se crean las migraciones contienen dos métodos up() y down(), el primero es para crear y actualizar tablas, columnas e índices y la segunda normalmente es usado para revertir operaciones hechas por el primer método,

Al correr el comando `-php artisan migrations` se guardan los cambios y aparecen en la base de datos en MySQL como se muestra en la siguiente imagen:

CREACIÓN DEL MODELO

Los modelos son uno de los componentes principales de este tipo de aplicaciones desarrolladas bajo el patrón MVC, el modelo hace la tarea de acceder a los datos y modificarlos, además de mantener la lógica y las reglas que deben cumplirse al trabajar con los datos, por tanto el tipo de acciones que vamos a solicitar de un modelo es obtener datos, insertarlos, modificarlos, además se realizó la validación de los datos para asegurarnos que tienen la forma necesaria de guardarlos.

Para la creación del modelo en la terminal se ejecutaron los siguientes comandos:

- `-php artisan make:model client`
- `-php artisan make:model users`
- `-php artisan make:model permissions, roles`

```
9  class Client extends Model
10 {
11     use SoftDeletes;
12
13     protected $dates = [ // variables las cuales el usuario
14                         // no puede acceder
15         'created_at',
16         'updated_at',
17         'deleted_at',
18     ];
19
20     protected $fillable = [ // el usuario úede modificar estos campos
21         'RazonSocial',
22         'Giro',
23         'Telefono',
24         'Direccion',
25         'Zona',
26         'Ciudad',
27         'Estado',
28         'Contacto',
29         'Correo',
30         'created_at',
```

Ilustración 4 5 Creación del modelo

Como podemos ver en la imagen, aquí se ingresaron los campos que existen y lo que el usuario puede hacer con ellas, la diferencia de lo que se hizo en migraciones es que en los modelos se especifican los campos y/o atributos con los que se trabajaran dentro de la aplicación así como el desarrollo de su lógica. mientras que en las migraciones se crea la base de datos la cual la aplicación está conectada mediante el archivo. `env`.

Lo mismo se hizo para los modelos de permiso, usuarios, roles.

```

class Permission extends Model
{
    use SoftDeletes;

    protected $dates = [
        'created_at',
        'updated_at',
        'deleted_at',
    ];

    protected $fillable = [
        'title',
        'created_at',
        'updated_at',
        'deleted_at',
    ];
}

```

Ilustración 4 7 Modelo para permisos

```

class Role extends Model
{
    use SoftDeletes;

    protected $dates = [
        'created_at',
        'updated_at',
        'deleted_at',
    ];

    protected $fillable = [
        'title',
        'created_at',
        'updated_at',
        'deleted_at',
    ];

    public function permissions()
    {
        return $this->belongsToMany(Permission::class)
    }
}

```

Ilustración 4 6 Modelo para roles

CREACIÓN DE LAS RUTAS Y CONTROLADORES

Los controladores en Laravel son clases que permiten agrupar acciones de una sección en particular, por ejemplo tenemos ClientsController y los métodos de este serian crear (), eliminar (), modificar().

Por otro lado las rutas se crean para gestionar datos en particular, podemos crear un controlador y definir cada una de sus acciones, por ejemplo en nuestro proyecto se hizo de la siguiente manera:

```
Route::redirect('/', '/login');
```

Al ingresar a la ruta, la primera pantalla que

se muestra es la de login, una vez ingresada las credenciales y confirmarse que son válidas se ingresa a la aplicación, la ruta para esto es la siguiente.

```

Route::group(['prefix' => 'admin', 'as' => 'admin.', 'namespace' => 'Admin', 'middleware' => ['auth']], function () {
    Route::get('/', 'HomeController@index')->name('home');

    Route::delete('permissions/destroy', 'PermissionsController@massDestroy')->name('permissions.mass Destroy');

    Route::resource('permissions', 'PermissionsController');
}

```

Las rutas se modificaron en el archivo web.php en la carpeta routes de nuestra aplicación.

Para generar las rutas de manera más fácil utilizamos el comando `-php artisan route:list`. Esto genera las listas de rutas en el archivo y ya se modifican las acciones necesarias como se hizo en el caso de la ventana de login para poder ingresar.

Los métodos GET permiten recuperar datos y mostrarlos en la interfaz, mientras que otros son métodos que nos permiten realizar diversas acciones.

CONTROLADORES

Una vez ya creadas las rutas se procedió a crear los controladores, estos nos permiten agrupar la lógica de las peticiones HTTP relacionadas, de esta forma organizamos mejor nuestro código.

Para generar un controlador corremos el comando `php artisan make:controller ClientController`

Realizamos lo mismo para usuarios, permisos y roles, hecho esto tenemos nuestros controladores en la carpeta `app/Http/Controllers` de nuestra aplicación donde podremos escribir nuestra lógica y consultas que ejecuta la aplicación realizada como muestra en la siguiente ilustración

Dentro de esta clase (en este caso `ClientController`), agregamos nuestros métodos públicos que después enlazamos a una ruta de la siguiente manera:

```
Route::get('/Clientes', 'ClientController@index')
```

CONFIGURACIÓN DE BOOTSTRAP

Para este proyecto se utilizó `vue` y `bootstrap` los cuales fueron de gran ayuda a la hora de realizar el diseño frontend de la aplicación. Afortunadamente `Laravel` por defecto usa `NPM` para instalarlos, estos se agregan en la carpeta `laravel/ui` ejecutando el siguiente comando:

```
-composer require laravel/ui --dev
```

```

→ laravel6 composer require laravel/ui --dev
Using version ^1.0 for laravel/ui
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Installing laravel/ui (v1.0.1): Downloading (100%)
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: laravel/tinker
Discovered Package: laravel/ui
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
→ laravel6

```

Ilustración 4 8 instalación de Bootstrap

Una vez instalado se procedió a se crean las carpetas donde se *desarrollará* el front end para crear las vistas y la estructura de la aplicación por medio de los siguientes comandos.

```

// Generate basic scaffolding...
php artisan ui vue
php artisan ui react

// Generate login / registration scaffolding...
php artisan ui vue --auth
php artisan ui react --auth

```

Ilustración 4 9 instalación de Vue

GENERANDO LAS VISTAS

En esta parte del proyecto creamos las vistas o archivos blade en php, para ello fue necesario crear ir a resources/views/admin y crear carpetas para cada una de las entidades (clientes, usuarios, roles y permisos). Dentro de esta carpeta se crean los archivos para cada una de las acciones que realiza el CRUD, por ejemplo:

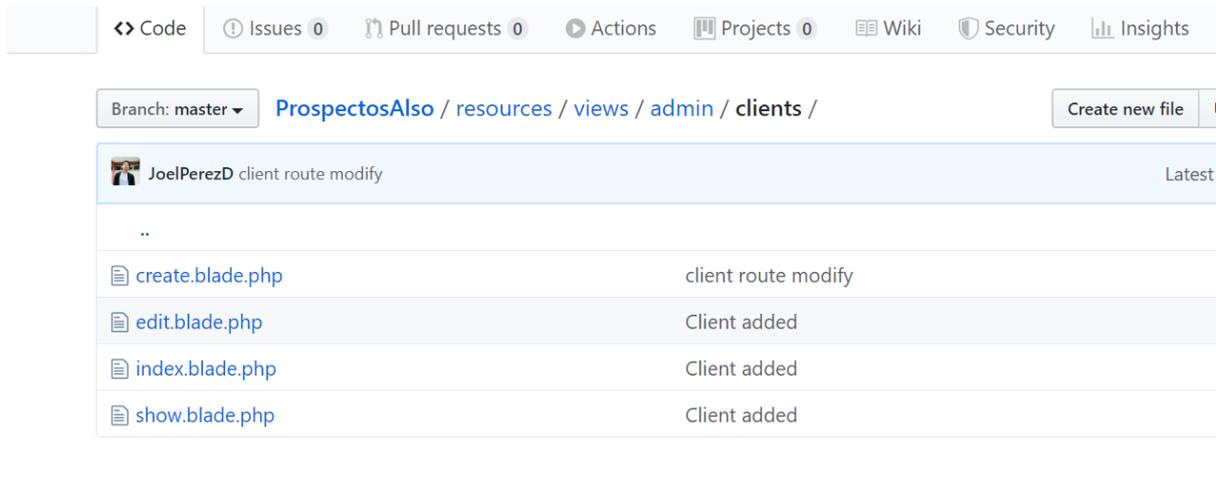


Ilustración 4 10 Vistas de clientes

En cada uno de los archivos se escribe código de las vistas de acuerdo con las acciones que el usuario llevará a cabo:

```

1  @extends('layouts.admin')
2  @section('content')
3
4  <div class="card">
5      <div class="card-header">
6          {{ trans('global.create') }} {{ trans('global.client.title_singular') }}
7      </div>
8
9      <div class="card-body">
10         <form action="{{ route('admin.clients.store') }}" method="POST" enctype="multipart/form-data">
11             @csrf
12             <div class="form-group {{ $errors->has('RazonSocial') ? 'has-error' : '' }}">
13                 <label for="RazonSocial">{{ trans('global.client.fields.RazonSocial') }}*</label>
14                 <input type="text" id="RazonSocial" name="RazonSocial" class="form-control" value="{{ old('RazonSocial', isset($
15                     @if($errors->has('RazonSocial'))
16                         <em class="invalid-feedback">
17                             {{ $errors->first('RazonSocial') }}
18                         </em>
19                     @endif
20                 <p class="helper-block">
21                     {{ trans('global.client.fields.RazonSocial_helper') }}
22                 </p>
23             </div>
24             <!-- -->
25             <div class="form-group {{ $errors->has('Giro') ? 'has-error' : '' }}">
26                 <label for="Giro">{{ trans('global.client.fields.Giro') }}*</label>
27                 <input type="text" id="Giro" name="Giro" class="form-control" value="{{ old('Giro', isset($client) ? $client->G
28                     @if($errors->has('Giro'))
29                         <em class="invalid-feedback">
30                             {{ $errors->first('Giro') }}
31                         </em>
32                     @endif
33             </div>

```

Ilustración 4 11 Índice de clientes

Como podemos observar, al dar clic sobre “crear cliente” se redirecciona a la ruta crear cliente, se despliega la ventana en donde se agregan los datos, estos se validan para mandarlos a la base de datos y guardarlos. Cada vista contiene métodos los cuales creamos en los controladores según las acciones que se ejecutarán, los controladores saben que vista se está ejecutando gracias a las rutas que configuramos anteriormente. el resultado de esta vista es el siguiente:

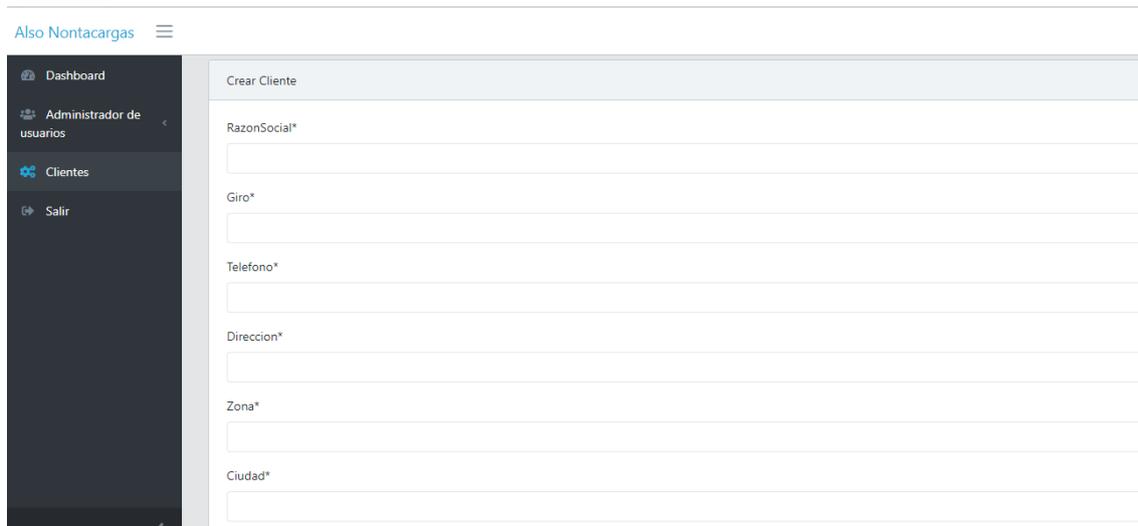


Ilustración 4 12 Crear cliente

Con la intención de no repetir código, ya que en todas las vistas se utilizan métodos realizados en JavaScript similares o contienen Bootstrap se separa el código en diversas carpetas.

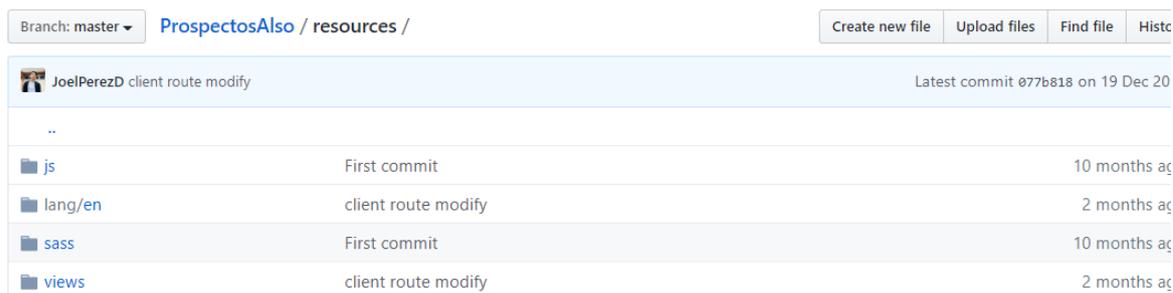


Ilustración 4 13 Vistas de la aplicación

La carpeta js contiene los componentes, archivos Bootstrap y vue ya que se utilizan en todas las ventanas de manera que se evita ser redundante con el código, la carpeta SASS simplifica el código en CSS, facilita el uso de las variables asignándoles nombre, en la carpeta views como vimos anteriormente contiene todas las ventanas, con código HTML, PHP métodos de Javascript que integran la parte “Frontend” de la aplicación web.

5-ANÁLISIS DE LA EXPERIENCIA ADQUIRIDA

5.1- Análisis general del programa, su diseño, desarrollo y organización.

Al iniciar mi programa de prácticas profesionales en la empresa ALSO no se contaba con un objetivo o proyecto a desarrollar para solucionar el problema con las bases que se contaban, eran demasiados datos y corrían el riesgo de perderlos o de que se repitieran datos ya que estos estaban en archivos xls en las computadoras de los vendedores, Al iniciar el programa se me asignaron diversas tareas como soporte técnico, ayuda a la creación de gráficas para determinar el progreso de los vendedores así como reportes o envío de diversos correos a clientes a los que se desean expandir como a modo de eMarketing , es en este punto donde identifiqué la problemática con las bases de datos de los clientes, estas bases de datos se utilizan para desarrollar lo que ellos llaman como “cinco-diez-quince”. el cual como anteriormente se ha explicado son llamadas, visitas y correos a clientes y prospectos. Al identificar el problema se lo comuniqué a los vendedores para posteriormente llegar a un acuerdo con las personas encargadas de la empresa a lo que ellos asintieron, a medida que iba avanzando ellos me daban sugerencias tales como importar y exportar datos para ingresarlos fácilmente o la posibilidad de crear varios usuarios y otorgarles permisos al mismo tiempo que seguía ayudando con otras tareas de la empresa.

La persona con la que más tuve comunicación es la asistente Karen Paredes y Alan Flores los cuales me dieron una serie de requisitos y evaluaron el proyecto al final de este.

5.2 Análisis de los objetivos del programa

Mi objetivo dentro de la empresa era facilitar las tareas que se llevaban referente a las TI, considero que este objetivo se cumplió ya que gracias a este proyecto los vendedores pueden identificar de manera más rápida a sus clientes o prospectos así como los datos de contacto y ubicación de estos, se evitan que existan casos en los que se pueda repetir de empresa y elimina la posibilidad de que la información se pierda fácilmente ya que se aloja en un servidor. Además es un proyecto el cual puede ir creciendo e ir adaptándose a las necesidades y objetivos de los vendedores y de la empresa.

5.3 Análisis de las actividades realizadas.

Las actividades que lleve a cabo fueron una gran experiencia de aprendizaje ya que constantemente aprendía sobre lenguajes de programación con los que había tenido poco contacto como PHP y JavaScript, utilice tecnologías como Laravel el cual facilita la realización de código y aprendí un poco más acerca de las bases de datos, relaciones y entidades.

Considero que gran parte de los conocimientos adquiridos en el transcurso de la carrera me ayudaron bastante a la hora de identificar problemas y cómo resolverlos, con que tecnología, que lógica utilizar y cómo llevar a cabo una metodología para cumplir con el objetivo de un proyecto.

5.4 Análisis de la metodología utilizada

La metodología utilizada en este proyecto fue parecida a una metodología en espiral ya que constantemente estaba teniendo comunicación con los miembros de la empresa con los que tenía contacto, estos me ayudaban o me daban sugerencias acerca de lo que se requiere, además de contar con determinados “tiempos de entrega” por así decirlo ya que cada vez que terminaba una tarea se la mostraba a la asistente o encargado de los vendedores, una vez ya presentado el resultado de la tarea se tomaba decisiones acerca de cambiar, continuar, mejorar o eliminar algo acerca de lo realizado.

6- CONCLUSIONES Y RECOMENDACIONES

Las actividades que realicé dentro de mi programa de prácticas profesionales ayudaron a fortalecer los conocimientos adquiridos durante la carrera, de los cuales, algunos de ellos solo eran teóricos y los puse en práctica. Además, me ayudaron a comprender cómo son las empresas en el mundo exterior, las necesidades de estas y como un ingeniero en sistemas de información siempre puede aportar a las empresas ya que muchas de ellas tienen problemas y necesidades a atender.

Comprendí que en nuestra área como ingenieros en software no se limita a solo empresas de desarrollo, ya que todas las empresas necesitan de TI y es ahí donde podemos brindarles soluciones a problemas que inclusive ni se saben que se tienen y en muchas ocasiones son raíz de conflictos que se derivan de estos dichos problemas,

Como recomendación para futuros estudiantes que desean hacer sus prácticas en ALSO Montacargas o ya sea en la consultoría KM solución, si buscan una opción que les haga ganar experiencia en la área de desarrollo, sean claros desde el principio ya que las actividades que se realizan van más orientadas a soporte técnico o documentación es por ello que propuse este proyecto.

Y como recomendación para la empresa sugeriría que se valore más el trabajo de los de TI, existe muy buena comunicación y ambiente de trabajo pero se subestima el trabajo que se hace en esta área ocasionando muchas veces que exista presión debido a que se ignora la dificultad y tiempo que conlleva este.

7 ANEXOS

<https://fernando-gaitan.com.ar/laravel-5-parte-5-controladores-y-rutas/>

<https://appdividend.com/2019/09/12/laravel-6-crud-example-laravel-6-tutorial-for-beginners/>

<https://getbootstrap.com/docs/4.4/content/typography/>

<https://laravel.com/docs/6.x>

<https://www.phpmyadmin.net/>



Hermosillo, Sonora, México

13 de diciembre del 2019

Por medio de este presente se informa que el alumno: **Joel Alfonso Perez Diaz** con número de expediente **215206080** termino satisfactoriamente sus practicas profesionales en el periodo del **11 de septiembre al 13 de diciembre del 2019** en la empresa **ALSO MONTACARGAS** ubicada en ave Salamanca 117, las amapolas, Hermosillo Sonora. participando en el proyecto **BASE DE DATOS DE CLIENTES Y PROSPECTOS ALSO** cubriendo un total de 340 horas en el periodo ya mencionado.

Se extiende la presente carta de terminación de practicas profesionales para los fines que el interesado convenga.

Atentamente:

Héctor Villa

Ing. Héctor Villa

Gerente General



ALSO MONTACARGAS Y EQUIPOS S A DE C
Salamanca Esq Pintor #117 8
C P 83020 Hermosillo, Sonora
Tel (662) 210 4437 RFC AME 0704



UNIVERSIDAD DE SONORA

COORDINACIÓN DIVISIONAL DE INGENIERIA

PRÁCTICAS PROFESIONALES

FPP-4

DEPARTAMENTO: Ingeniería Industrial

UNIDAD REGIONAL CENTRO CAMPUS HERMOSILLO

REPORTE FINAL DE ACTIVIDADES

Periodo: Del 11 / Septiembre / 2019 al 13 / Diciembre / 2019

Cantidad de 340 Horas de un total de 340 Avance: 100 %

Nombre del practicante: Joel Alfonso Perez Diaz

Expediente: 215206080 Programa Educativo (Licenciatura): Ingeniería en sistemas de Información

Nombre del Programa/Proyecto: Creación de base de datos para clientes y prospectos de ALSO MONTACARGAS

Datos de la Unidad Receptora (Razón Social): ALSO MONTACARGAS

Responsable de la Unidad Receptora (Nombre/Puesto): Héctor Villa

Contacto: Teléfono/UR: 6622104437 Ext. 116 Celular: 6623003335

DESCRIPCIÓN GENERAL DE ACTIVIDADES

Creación de una base de datos en donde se manejan todos los clientes y prospectos de la empresa ALSO MONTACARGAS utilizando como gestor de base de datos MySQL y como lenguaje PHP y framework LARAVEL para la aplicación web. En la base de datos es posible filtrar las empresas por su nombre, giro, contacto, es posible ingresar con un usuario administrador el cual puede crear otras cuentas y determinar los roles que desempeñaran cada uno de ellos. Además la base de datos puede imprimirse en archivo csv o Excel si así se desea así como puede actualizar, eliminar o agregar clientes

Se genero indicadores de cada uno de los vendedores de la empresa en donde se muestra las ganancias por monto, cotización así como las llamadas, visitas, pérdidas, correos que han hecho cada uno.

Se realizaron graficas de los nuevos vendedores que ingresaron a la empresa, se lleva seguimiento diario y semanal

Se manejo CRM llamado Gestino para ingresar datos de la facturación de proveedores y clientes

Se maneja MailChimp para enviar correos de manera masiva, diseñar correos y flayers para ayudar a la empresa a conseguir más clientes, para realizar esto anteriormente se llevó un curso de E-Marketing

Prospectar clientes para ALSO montacargas buscando empresas en distintas bases de datos tales como las de la INEGI o de la secretaria de economía.

RETROALIMENTACIÓN (Comentarios del tutor)

En caso de requerirse, anexar reportes, formatos, diagramas que apoyen las actividades realizadas.

Para las Ingenierías deberá anexar **reporte técnico** en archivo electrónico ≤ 2 MB y carta de terminación de prácticas firmada por el responsable de la empresa.

Observaciones Generales:

 Nombre y firma del alumno	 Nombre y firma del tutor de prácticas profesionales UniSon.	 Nombre y firma del responsable de la unidad receptora Tel 16621104437 REC AME 07041
-------------------------------	---	--

Original entregar en físico al Coordinador o Responsable de Prácticas Profesionales de la carrera.

Copia para Tutor de Prácticas Profesionales y Copia alumno.

Enviar en PDF los documentos al coordinador/responsable de prácticas profesionales de la carrera.