



PROYECTO DE TITULACIÓN

**DESARROLLO DE UNA APLICACIÓN WEB PARA UNA
HUERTA DE LIMONES EN PLAZUELA, RIOVERDE, S.L.P.**

Que Presenta

Seth Aldanid Torres Delgado

Para obtener el Titulo de Ingeniero en Sistemas

Computacionales

Octubre – Diciembre 2021

Agradecimientos

Quisiera agradecer a la universidad UDEP, por darme la oportunidad de estudiar y acabar mis estudios en esta institución. También daré gracias a mi escuela anterior la Universidad Tecnológica Superior de Rioverde por darme las herramientas para el éxito en esta carrera.

Agradezco a mi asesor ING. Pedro García Guerrero que se dio tiempo para poder guiarme en este proceso. Que me dio herramientas y funciones para poder hacer este proyecto mas eficiente durante estos 2 meses.

Agradezco a mi familia por tener la paciencia durante estos años de carrera. Especialmente durante estos dos meses pesados que tuvimos, se que no estuve para ayudar mucho. Yo les doy todos mis gracias por ser los mejores padres y hermanas que alguien quisiera tener.

Quisiera dar otros agradecimientos a mi perros y gatos: Tullio, Bernarda, Benito, Ana, Cecilia, Cody, Wilson, Chase, Cameron, Margarita, Armando, Satomi, Pancho, Concha, Sancho, Cacho, Ronnie, Alexander, Emma, Santiago, Pepita, Simba, y Porfirio. Incluyendo a los que ya no están aquí: Alf, Chilindrina, Ursula, Félix, Chato por darme compañía durante toda mi vida.

Seth Torres

Índice General

1 Capítulo I: Introducción.....	1
1.1 Antecedentes.....	1
1.2 Planteamiento del Problema.....	2
1.3 Hipótesis.....	3
1.4 Objetivos.....	3
1.4.1 Generales.....	3
1.4.2 Específicos.....	4
1.5 Alcances y Metas.....	4
2 Capítulo II: Marco Teórico y Metodológico.....	5
2.1 Antecedentes de la Investigación.....	5
2.2 Marco Metodológico.....	9
2.3 Justificación de Investigación.....	9
3 Capítulo III: La Propuesta.....	10
3.1 Presentación.....	10
3.2 Justificación.....	10
3.3 Objetivos.....	11
3.3.1 Objetivos Generales.....	11
3.3.2 Objetivos Específicos.....	11
4 Capítulo IV: Análisis.....	12
4.1 Requerimientos.....	12

Seth Torres	III
4.1.1 Funcionales.....	12
4.1.2 No Funcionales.....	12
4.2 Diagrama de Caso Uso.....	14
4.3 Diagrama Entidad Relación.....	15
5 Capítulo V Diseño.....	16
5.1 Bosquejos.....	16
5.1.1 Bosquejo de Página Principal.....	16
5.1.2 Bosquejo de Página de Tienda.....	17
5.1.3 Bosquejo de Login.....	18
6 Capítulo VI: Herramientas de Desarrollo.....	19
6.1 Visual Studio.....	19
6.2 Git.....	19
6.2.1 Github.....	19
6.3 Base de Datos.....	20
6.3.1 MySQL Workbench.....	20
6.4 PHP.....	20
6.4.1 Composer.....	20
6.5 Laravel.....	20
6.5.1 Livewire.....	21
6.5.2 Jetstream.....	21
6.6 JavaScript.....	21
6.6.1 NPM.....	21

6.7 Materialize.....	21
7 Capítulo VII: Desarrollo.....	22
7.1 Preparación Inicial.....	22
7.2 Instalación de Proyecto Laravel.....	22
7.3 Implementación de Plugins y Paquetes.....	22
7.3.1 Instalación de Livewire y Jetstream.....	22
7.3.2 Instalación de Materialize.....	23
7.4 Desarrollando el Sistema.....	24
7.5 Creación de modelos, Controladores, Vistas.....	25
7.5.1 Creación de Modelos.....	25
7.5.2 Creación de Controles.....	26
7.5.3 Creación de Vistas.....	26
7.6 Rutas y Controladores.....	27
7.6.1 Controlador.....	28
7.7 Definición de Base de Datos.....	28
7.7.1 Modelos.....	30
7.7.2 Migraciones.....	30
7.7.3 Semillas.....	32
7.8 Definición de Vistas.....	33
7.8.1 Diseño grafico.....	33
7.8.2 Controlador de Vistas.....	36
8 Capítulo VIII: Implementación y Pruebas.....	37

8.1 Usuarios.....	37
8.1.1 Creación de Usuario.....	37
8.1.2 Modificación de Usuario.....	38
8.1.3 Eliminación de Usuario.....	39
8.2 Menú de Admin.....	40
8.2.1 Productos.....	41
8.2.2 Entradas.....	43
8.2.3 Salidas.....	43
8.2.4 Compras.....	44
8.2.5 Reportes.....	44
8.3 Pagina de Tienda.....	46
8.4 Hacer una compra.....	47
8.5 Blogs.....	49
8.5.1 Creación de Blog.....	50
8.5.2 Modificación de Blog.....	51
8.5.3 Eliminación de Blog.....	52
9 Capítulo IX: Conclusiones.....	53
10 Bibliografías.....	54
11 Glosario de Términos.....	59
12 Anexos.....	60
12.1 Código Fuente.....	60
12.2 Hojas de Referencias.....	61

12.2.1 Laravel.....	61
12.2.2 HTML.....	62
12.2.3 PHP.....	63
12.2.4 MySQL.....	64
12.2.5 Git.....	65

Índice de Tablas y Figuras

Image1.....	1
1.1 Mapa de dirección de Plazuela.....	1
1.2 Imagen de la Huerta.....	2
4.1 Diagrama Caso Uso.....	14
4.2 Diagrama Base de Datos.....	15
5.1 Bosquejo de Página Principal.....	16
5.2 Bosquejo de Tienda.....	17
5.3 Bosquejo de Login.....	18
7.1 Imagen de Archivo CSS que Buscar.....	23
7.2 Imagen de Visual Studio.....	24
7.3 Imagen de menú de Terminal.....	25
7.4 Imagen de Rutas.....	27
7.5 Ejemplo de Boton.....	27
7.6 Imagen de Vista.....	28
7.7 Imagen de la vista.....	28
8.1 Imagen de definicion de Base de Datos.....	29
7.8 Imagen de Modelo.....	30
7.9 Imagen de Migracion.....	31
7.10 Imagen de Semilla.....	32
7.11 Página de Material Design de Google.....	33

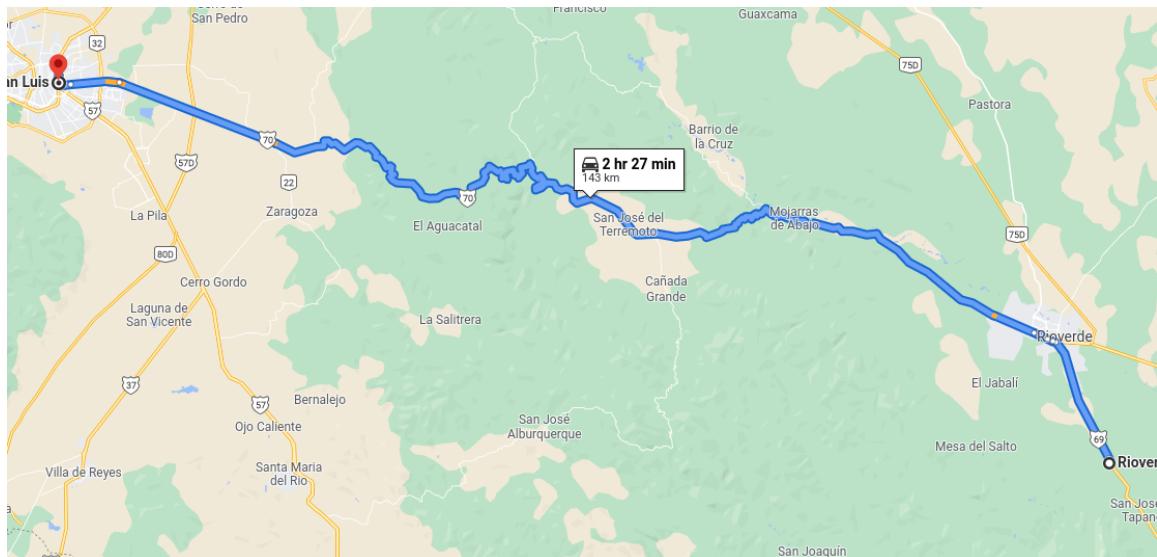
7.12 Página de Google Fonts.....	34
7.13 Página de Google Fonts.....	34
7.14 Imagen de controlador.....	36
8.1 Imagen de registro de usuario.....	37
8.2 Imagen de registro de usuario.....	37
8.3 Imagen de Usuario Registrado.....	38
8.4 Menú de Eliminación.....	38
8.5 Nombre original.....	39
8.6 Nombre nuevo.....	39
8.7 Nombre Cambiado.....	39
8.8 Menú de Eliminación.....	39
8.9 Área de confirmación.....	40
8.10 Botón de Admin.....	40
8.11 Menú de Admin.....	40
8.12 Creación de producto.....	41
8.13 Nuevo producto creado.....	41
8.14 Nuevo Producto.....	41
8.15 Imagen de Edición de Producto.....	42
8.16 Producto Borrado.....	42
8.17 Pagina de Entradas.....	43
8.18 Imagen de Salidas.....	43
8.19 Imagen de Compras.....	44

8.20 Imagen de pagina de PDF.....	45
8.21 Imagen de Selección de Reportes.....	45
8.22 Imagen de resultados de reportes.....	45
8.23 Imagen de pagina de Tienda.....	46
8.24 Imagen de compra de Producto.....	47
8.25 Imagen de pagina de Carrito.....	48
8.26 Imagen de pago de Producto.....	48
8.27 Imagen de Blog del sistema.....	49
8.28 Imagen de creación de Blog.....	50
8.29 Imagen de resultados de creación de Blog.....	51
8.30 Imagen de edición de Blog.....	51
8.31 Imagen de eliminación de Blog.....	52
13.1 Hoja de Referencia Laravel.....	61
13.2 Hoja de Referencia HTML.....	62
13.3 Hoja de Referencia PHP.....	63
13.4 Hoja de Referencia MySQL.....	64
13.5 Hoja de Referencia Git.....	65

1 Capítulo I: Introducción

1.1 Antecedentes

El Pueblo de Plazuela, es un ejido de Rioverde SLP. El clima es cálido, con muchas lluvias por ser un área montañosa. Aquí se dan a crecer arboles frutales como: naranjas, limones, mangos, y papayas.



1.1 Mapa de dirección de Plazuela

Este negocio familiar se inicio en el año 2019, con la construcción de una huerta de limones. El área de la zona media, donde esta Plazuela es conocida por sus naranjas. Los pueblos y ciudades en alrededor son: Rioverde, San Ciro, Ciudad Fernández, El Refugio, El Jabalí, Riachuelo, Gordo Blanco, El Caracol, El Capulin.



1.2 Imagen de la Huerta

Se decidió que se venderían limones porque, son un producto que están en constante demanda todo el año. El tipo de limón que se plantaron son limones persas sin semillas.

1.2 Planteamiento del Problema

El problema que tienen los jefes de la huerta es la baja distribución de limones. A veces se tiene que tirar la cosecha, porque no hay suficientes clientes para todos los limones que se venden. Hay también un problema de que no se da un numero exacto

de cuantos limones hay para vender en cada cosecha. Otro problema es que no se puede dar un buen tiempo para poder vender limones en tiempos en que el precio del limón es al mas alto para vender a máximo beneficio.

1.3 Hipótesis

Es posible crear un sistema web en la que se pueda tener una forma de controlar los productos que la huerta da todos los días. Se podrá dar una forma de vender en linea los productos, y tener una manera de conectar con los clientes si hay alguna pregunta que tengan. Esto incluirá los datos de teléfono, email, o mensaje directo, Este sistema se podrá tener la forma de hacer que los jefes o los empleados difundir noticias de la huerta, incluyendo formas de conectar con otras redes sociales. El sistema tendrá la forma de controlar las ventas que los clientes hacen, incluyendo hacer reportes de todo lo que se necesite.

1.4 Objetivos

Se darían los objetivos generales, y específicos para saber que es lo que se va hacer en este proyecto.

1.4.1 Generales

El objetivo es:

- crear un sistema web para poder difundir información sobre la huerta de limones.
- Hacer ventas en linea del producto.

- Tener un control del producto al ser vendido.

1.4.2 Específicos

Los objetivos específicos son:

- Al difundir información de la huerta, se dará información de otros medios en que la huerta esta ligada como redes sociales, o email.
- Los clientes pueden comprar los productos y pueden poner opciones en como se puede entregar el producto.
- El administrador puede introducir y modificar los productos en que se van a vender. Se puede cambiar el precio de los productos basados en precios generales.

1.5 Alcances y Metas

En si este proyecto usualmente duraría un tiempo de 6 mese a un año, pero por el corto tiempo de 8 semanas. Se podría hacer un sistema funcional en la superficie. Con la negativa que todos se haría a la rápida y sin tantas pruebas. Se tiene que tener un sistema base hecho con ciertos módulos creados. Se tiene que tener instalado la base de datos para poder hacer llamar datos a los módulos. Se va poder hacer ciertos módulos exclusivos al administrador, y para los clientes. Se va a poder acceder el sistema en linea usando un hosting.

2 Capítulo II: Marco Teórico y Metodológico

2.1 Antecedentes de la Investigación

Las Huertas son áreas de terrenos grandes para la cultivación de frutas, verduras para consumo, usualmente para la venta de los productos. [1]

En si una pagina web es un documento digital hecho en html, dados por servidores, y pueden ser leídos en un navegador de web [2]. Un servidor es una computadora o un grupo de computadoras- echas especialmente para poder dar servicios o tener información como bases de datos para los clientes que estén conectados al servidor en una red [3].

Los sitios de internet tienen dos métodos de programación. Uno que es lo que el cliente ve, y otro en que no se ve. El front end, es lo que se ve y se interactúa del lado del cliente. Es el diseño grafico y la interfaz interactiva en la que se da, o pone información [4]. El back end es la programación que no se mira en la pagina. Es como un motor en la que todo se hace, como dar información a la base de datos o dar información que es solicitado del cliente [5].

Las paginas web son programadas para front end son HTML y CSS. HTML (Hyper Text Markup Language) es un lenguaje para crear margenes y definir la estructura de la pagina usando etiquetas [6]. CSS (Cascade Style Sheets) es como cada etiqueta y elementos de HTML es definido y dibujado en una pagina. Como

cambiar el color de texto, o definir ciertas etiquetas tamaños específicos dependiendo si la pagina es vista en una computadora o un teléfono celular [7].

Otros programas que son usadas para back end son como PHP, MySQL, y JavaScript. PHP es un lenguaje de scripting que se puede integrar a una pagina de web Para hacer paginas mas dinámicas. En si se puede usar para hacer programación y ejecución del lado del servidor [8][9]. MySQL es una lenguaje de base de datos, creado para tener un control y poder guardar información de un servidor. Con una base de datos se puede gestionar información [10][11]. JavaScript es otro lenguaje de scripting usado en paginas web. Es implementado y corre en el lado del cliente, en comparación de PHP. Era creada para tener un medio de interacción dinámica para un cliente. Es mas viejo que php; originalmente creado como una versión web de Java, pero termino siendo algo mas diferente. [8][12].

Muchas paginas modernas son creadas con frameworks. Un framework es una herramienta para ayudar en la creación rápida de paginas con diseños estandarizados [13]. Uno de los frameworks que se usara en este proyecto es Laravel. Laravel es un framework que usa php como su programación principal [14]. Laravel usa un numero de herramientas para otras funciones durante el desarrollo del proyecto como: Livewire, JetStream, y Tailwind. JetStream es un modulo de Laravel para la creación de Login, Registros, Verificación de E-mail, y Autenticador de 2 pasos para clientes [15]. Livewire es un framework fullstack que corre dentro de Laravel. Se usa para crear interfaces dinámicas [16]. Fullstack significa que es una aplicación de web que tiene herramientas de front end y back end [17]. Tailwind

es un framework de CSS creado para poder diseñar páginas modernas sin tener que escribir mucho código css [18].

Proyectos como estos requieren control de versiones para poder tener un desarrollo limpio y con menos defectos al salir al público. Se mantiene un control de quien y cuando se hicieron cambios o cuando se hicieron nuevos documentos antes de ser implementados. Si hay un problema se puede regresar a una versión estable y luego tratar de resolver el problema [19]. Git es un programa de control de versiones distribuida. Significa que cada persona que tiene un clon de un programa o proyecto es su propio control de versiones que son separados del proyecto original. Así que cada persona puede hacer pruebas sin tener que hacer una sincronización con el proyecto original para saber si funciona o no [20]. Cada proyecto se puede guardar en un servidor local o se puede subir en muchos sitios web para proyectos que usan git; Un ejemplo es Github, Github es un servicio web que hace la distribución de repositorios git más fácil de transferir entre desarrolladores en línea, sin tener que conectarse a un servidor especial o tener que estar presente a un edificio para hacer un trabajo [21].

Este proyecto se usará la metodología ágil. La metodología ágil es diseñado para tener una planeación y producción de un proyecto con la intención de cambiar durante su desarrollo. Ejemplos serían que un cliente quiera una característica que antes no fue implementada. Con otras metodologías no sería posible por que ya se adelanto el desarrollo de un programa. Mientras tanto con ágil la planeación está hecha para poder ser cambiada sin detener el desarrollo y posiblemente atrasar el

tiempo para crear un programa o aplicación[22]. Un método ágil común es el método SCRUM. Creado con la intención de hacer un marco en que se pueda hacer sprints (carreras) de corto tiempo, en que se hace lo que se va hacer primero, luego se hace una planeación rápida, luego se hace otro sprint para crear el producto. al final se hace juntas con el equipo y(o) clientes para dar retroalimentación para luego empezar de nuevo, hasta que sea satisfecha el proyecto que se esta haciendo [23].

Durante el desarrollo de un proyecto se usan herramientas para poder planear, crear maquetas, y tener un flujo de trabajo para hacer un desarrollo mas eficiente. Ejemplos serian StarUML, Laragon, Jira, Project, Visio. StarUML es un programa para modelar y crear diagramas de caso uso, maquetas de base de datos, diagramas de clases, entre otras cosas [24]. Laragon es un ambiente de desarrollo que es portable, rápido, y lleno de funciones para crear servidores virtuales en una computadora en casa [25]. Jira es una herramienta creada para equipos que usan metodologías ágiles. Es usado para el seguimiento de errores, gestionar tareas de desarrollo de software, y gestión de productos [26]. Project también conocido como Microsoft Project, es un programa para gestión de trabajos para proyectos. Usando tablas de Gantt se pueden crear las tareas del día a día de un trabajador que un gerente de trabajo usaría pare tener un control del flujo del desarrollo de software [27]. Una tabla de Gantt es una forma de dibujar los trabajos y fases un proyecto usando tablas y lineas de tiempo [28]. Microsoft Visio es un programa para la creación de diagramas, formación de maquetas, planeación de flujos de trabajos en una forma grafica [29].

2.2 Marco Metodológico

En estos tiempos con pandemias, y sin poder hacer contactos con gentes cercanas, es preferible tener una presencia en linea para hacer compras. Se ha visto que mas gente en estos tiempos son mas probables en comprar en linea que en tiempos pasados [30].

2.3 Justificación de Investigación

Una gran ventaja es de tener una presencia en linea. Se puede usar para tener un contacto con los clientes en una forma interactiva. Se puede tener una forma de servicio 24/7. Se puede tener un contacto para dar ciertos anuncios como tiempos de descuentos o cuando hay nuevos productos. En general muchos negocios ya están haciendo el brinco en linea rara sus futuras ventas. [31].

Otra justificación es de tener una forma de vender tus productos en linea. Con una forma de vender en linea se puede incrementar el área en que se vende. Ejemplo serian vender de estado a estado, o de un país a otro. En si no es requerido tener una pagina especial, pero se puede hacer otros servicios como: Whatsapp, Ebay, o Mercado Libre.

3 Capítulo III: La Propuesta

3.1 Presentación

Se presenta una pagina web que esta diseñado para difundir información de la huerta. La huerta también tendrá accesos a una pagina especial para tener un control de los productos que entran y salen. Los clientes tendrán una pagina para hacer compras de los productos.

3.2 Justificación

Casi todo negocio moderno tiene una pagina web, con conexiones a otras redes sociales. No todos, pero muchos tienen tiendas en linea. La razón por la que se sugiere hacer un sistema en la web es esta:

- Tener un lugar central en donde se puede difundir información y noticias de la huerta.
- Tener un control de los productos, incluyendo entradas, salidas, y ventas.
- Tener una tienda en linea en donde se puede comprar los productos de la huerta.

3.3 Objetivos

3.3.1 Objetivos Generales

Los objetivos de el proyecto son:

- Tener un sistema que se pueda difundir y dar noticias de la huerta.
- Tener un control de los productos incluyendo entradas, y salidas.
- Tener una forma de vender los productos usando el sistema.
- Poder tener un listado de los clientes y sus contactos.
- Hacer posible la forma de comunicarse con los clientes usando muchos métodos de comunicación.

3.3.2 Objetivos Específicos

Los objetivos específicos son:

- Crear un sistema diseñado para ser usado en un navegador web.
- Tener una forma de dar noticias o actualizaciones usando un sistema básico de blog.
- Crear una Base de Datos para controlar los datos del sistema.
- Poder conectar otras redes sociales y integrarlas con el sistema.
- Crear un sistema de venta de los productos.
- Tener una forma de creación, modificación, y eliminación de ciertas tablas de los sistemas.

4 Capítulo IV: Análisis

4.1 Requerimientos

4.1.1 Funcionales

Las requerimientos Funcionales son descripciones implícitas del comportamiento de un programa [32]. En si son las funciones que debe tener un programa. En este caso, el proyecto debe de tener estas funciones:

- Tiene que difundir información de la huerta.
- El cliente acceda pagina para mas información.
- Se puede comprar producto dentro del sistema.
- Tener control de entrada y salida de productos.

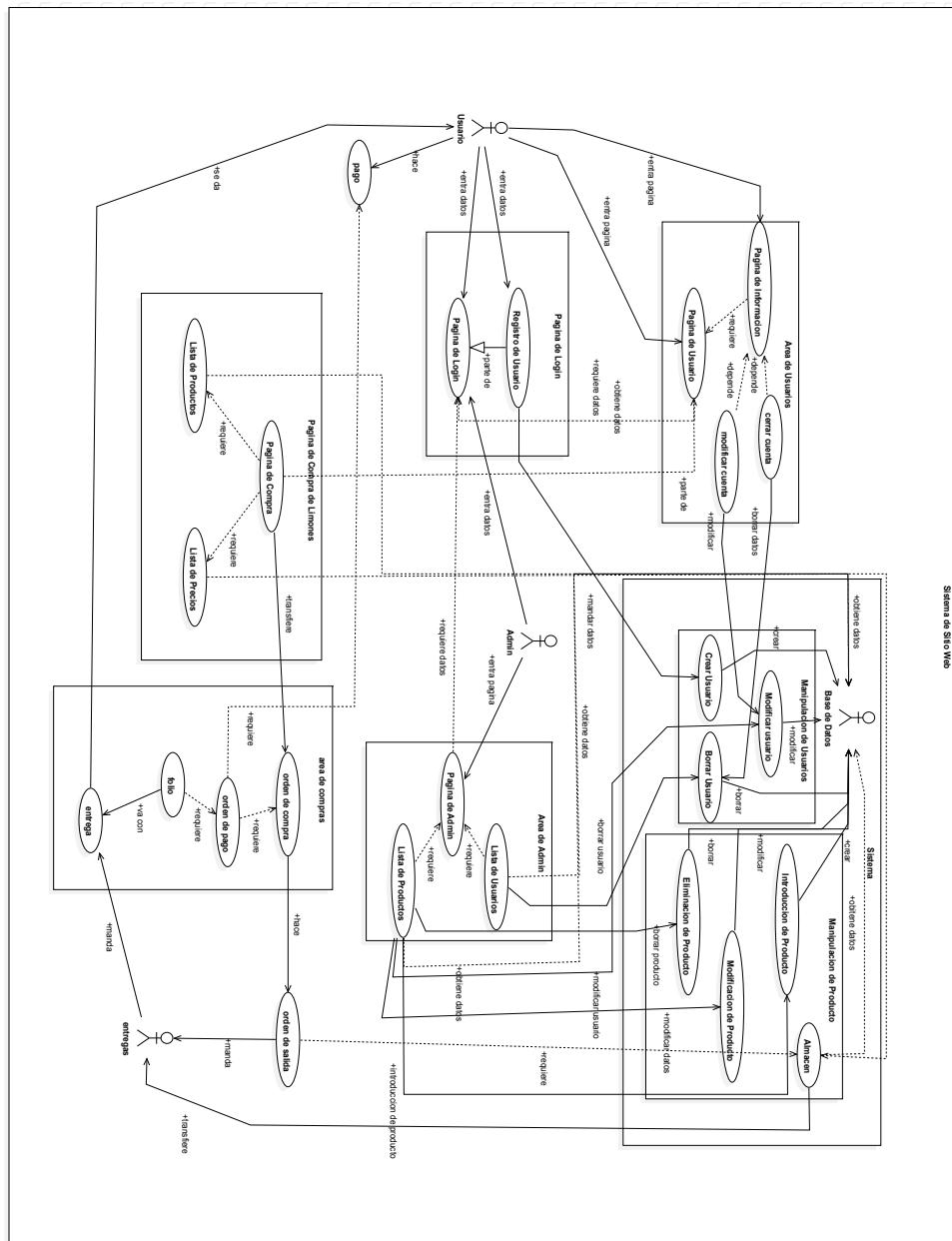
4.1.2 No Funcionales

Los requerimiento no funcionales son requisitos que hacen restricciones en el diseño del proyecto [33]. Ejemplos de requerimientos no funcionales serian:

- Tener que guardar usuarios en una base de datos.
- Requerirá un hosting para instalar el sistema web.
- La pagina de logeo se puede entrar usando credenciales de Google, o Facebook.
- Se hace la difusión de información con sincronización con otras redes sociales.

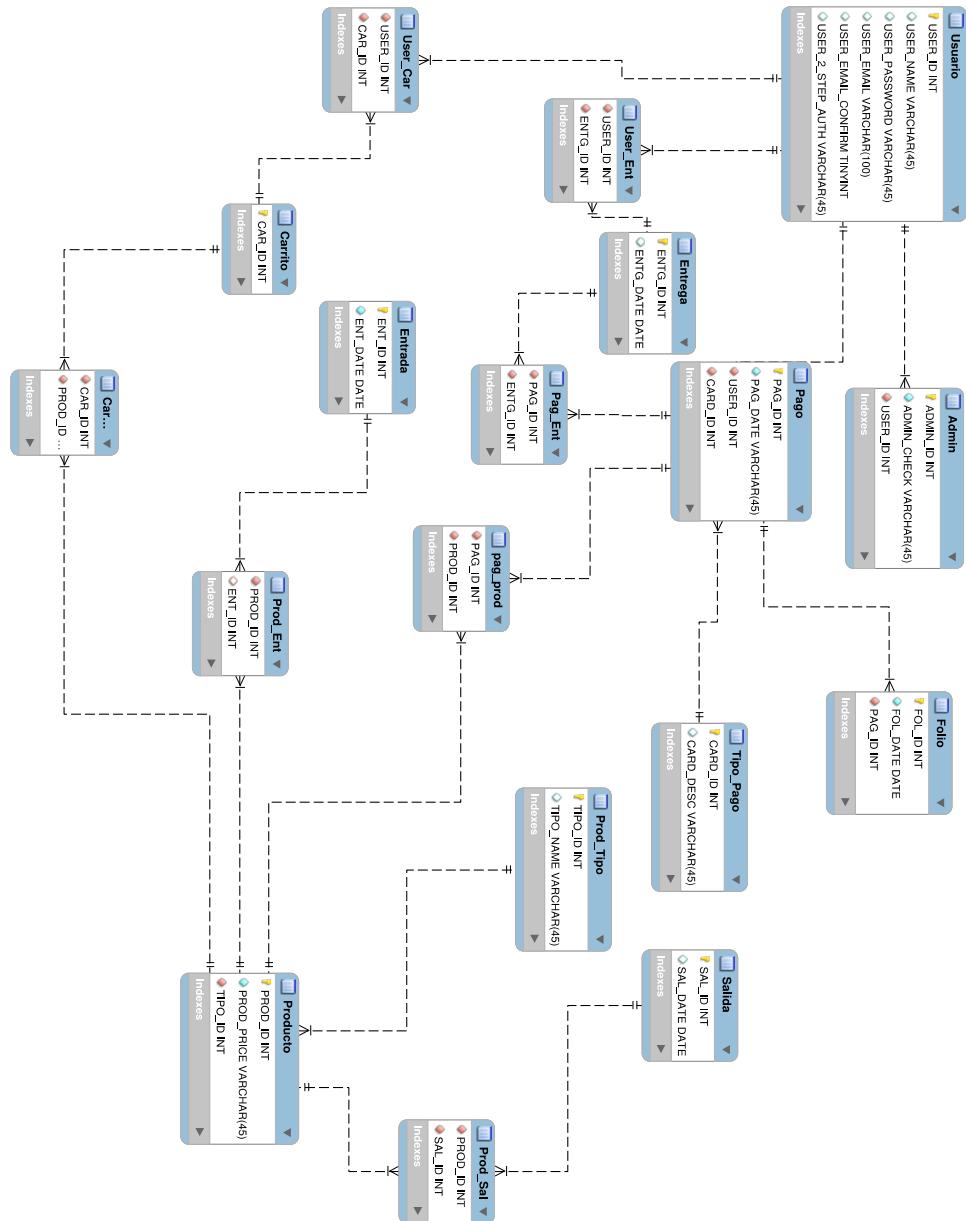
- La compra de producto se puede basar el precio en precios generales del mercado
- Se puede elegir la forma de pago y el tipo de entrega en linea.

4.2 Diagrama de Caso Uso



4.1 Diagrama Caso Uso

4.3 Diagrama Entidad Relación



4.2 Diagrama Entidad Relacion

5 Capítulo V Diseño

5.1 Bosquejos

Un Bosquejo es una forma para demostrar las clases del sistemas. Se demuestra una forma visual para que demostrar como el sistema se va a ver.

5.1.1 Bosquejo de Página Principal

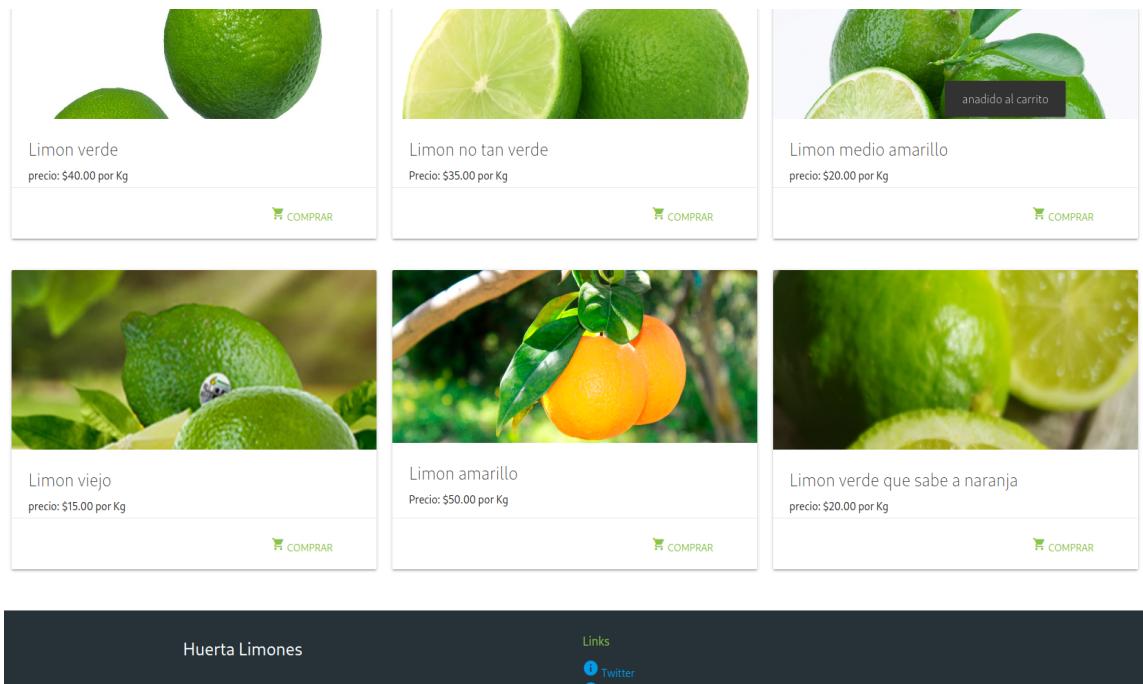


5.1 Bosquejo de Página Principal

En este bosquejo se muestra la pagina principal. En este caso se diseño usando HTML y CSS como bases. La pagina principal se vera blogs y actualizaciones de

otras páginas sociales que la huerta tenga. Se mirara el carrito de compra que tiene el usuario si tiene productos para comprar después.

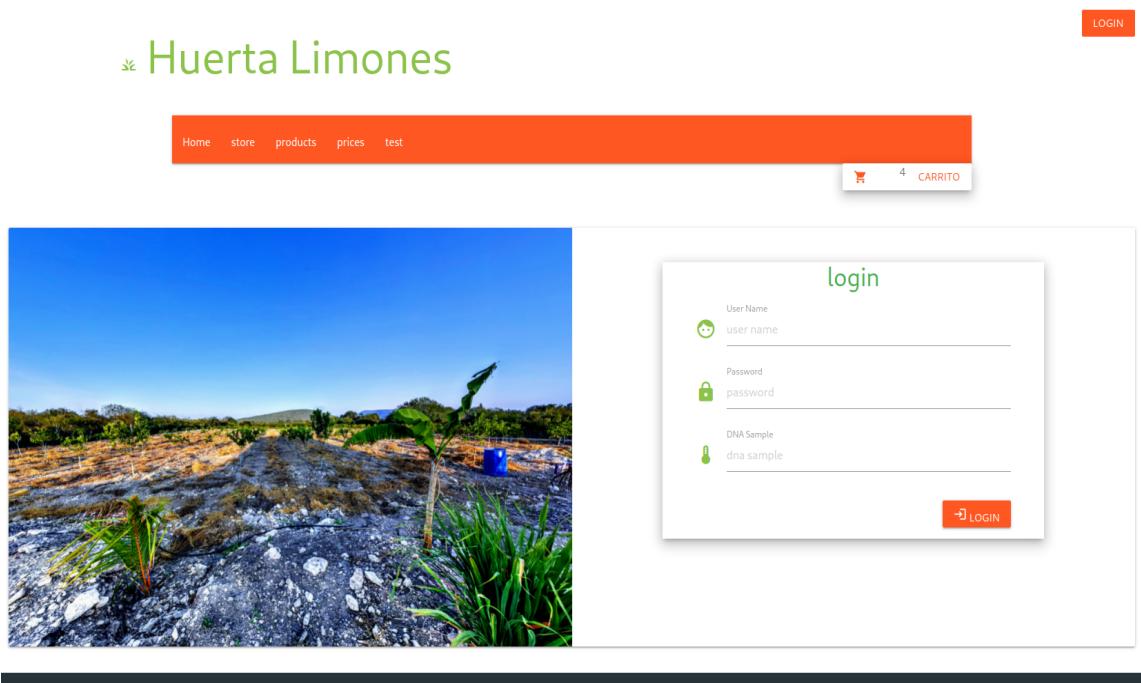
5.1.2 Bosquejo de Página de Tienda



5.2 Bosquejo de Tienda

En este bosquejo se muestran el interfaz en como se venden los productos. Se usaran programación PHP para hacer búsquedas en la base de datos, y demostrar los productos disponibles para vender. Se muestra una imagen del producto, con descripción y el precio. Al presionar el botón de comprar se pasara al carrito para después pagar.

5.1.3 Bosquejo de Login



5.3 Bosquejo de Login

La pagina de Login se usara una imagen de la huerta y un menú para entrar a la pagina de usuario.

6 Capítulo VI: Herramientas de Desarrollo

Las herramientas que se van usar para hacer el proyecto. Son lenguajes y software que se usara para tener el sistema hecho.

6.1 Visual Studio

Microsoft Visual Studio es un editor de texto creado para programación general.

Se puede instalar plugins y temas para tener un desarrollo rápido. Originalmente era un programa de paga, pero después fue hecha gratis y de origen abierta. Así que existen variaciones del mismo programa.

6.2 Git

Es un software para controlar versiones, cambios, y ramas de un proyecto de software. Se distribuye localmente, o en linea. Cada usuario tiene un clon del proyecto y al hacer cambios se puede hacer localmente, para después empujarlo al proyecto original [20].

6.2.1 Github

Github es una pagina que da el servicio de alojamiento de proyectos, para desarrolladores. Usan Git para controlar versiones y hacer ramas para múltiples usuarios [21].

6.3 Base de Datos

La base de datos que se esta usando es MySQL. Es un sistema de Base de Datos para guardar información y datos que son usados en servidores, haciendo consultas al sistema de Base de Datos [11].

6.3.1 MySQL Workbench

MySQL Workbench es un Software para administración, diseño, y mantenimiento de base datos visualmente [34].

6.4 PHP

PHP es un lenguaje de programación para paginas web. Se compilan en el lado del servidor, y luego es dado al cliente al tiempo de servir una pagina con php. Se puede usar, como en este caso para, crear frameworks en que construirá el sistema [9].

6.4.1 Composer

Composer es un administrador de dependencias de PHP. Se usa para instalar herramientas, o plugins para un proyecto [35].

6.5 Laravel

Es un framework para aplicaciones web. Es una base para crear paginas web rápido, y sin tantos problemas [14].

6.5.1 Livewire

Livewire es un plugin de Laravel. Es para crear paginas web dinámicas sin tener que usar frameworks de javascript completas [16].

6.5.2 Jetstream

Es un plugin de Laravel con Livewire, para empezar paginas web mas rápido. Tiene paginas hechas para login, registracion, autenticación de 2 pasos [15].

6.6 JavaScript

JavaScript es un lenguaje de programación diseñado para crear funciones complejas que HTML no puede hacer [36].

6.6.1 NPM

NPM es una función hecha con JavaScript, para manejar paquetes, descargar herramientas, correr paquetes sin descargar usando NPX, compartir código con otros usuarios, y actualizar código [37].

6.7 Materialize

Materialize es un framework de CSS basado en el diseño Material de Google [38]. Este es como se va a ver todos los componentes de la pagina web.

7 Capítulo VII: Desarrollo

7.1 Preparación Inicial

El sistema es creado usando Composer, para instalar el programa Laravel. Hay otras formas para instalar Laravel pero Composer es el que se va usar. El comando para instalar es:

```
$ composer global require laravel/installer
```

7.2 Instalación de Proyecto Laravel

Después se instala el proyecto de laravel usando el comando:

```
$ laravel new <nombre_de_proyecto>
```

después se mueve al archivo que se creo. Se puede correr un servidor local dentro del archivo usando el comando:

```
$ php artisan serv
```

7.3 Implementación de Plugins y Paquetes

Después de tener el sistema instalado se van a descargar e instalar paquetes requeridos y plugins.

7.3.1 Instalación de Livewire y Jetstream

Para instalar Jetstream es el comando:

```
$ composer require laravel/jetstream
```

Luego se hace el comando:

```
$ php artisan jetstream:install livewire
```

Después que todo sea instalado se hace el comando final para hacer los cambios sean vistos por el sistema.

```
$ npm install && npm run dev
```

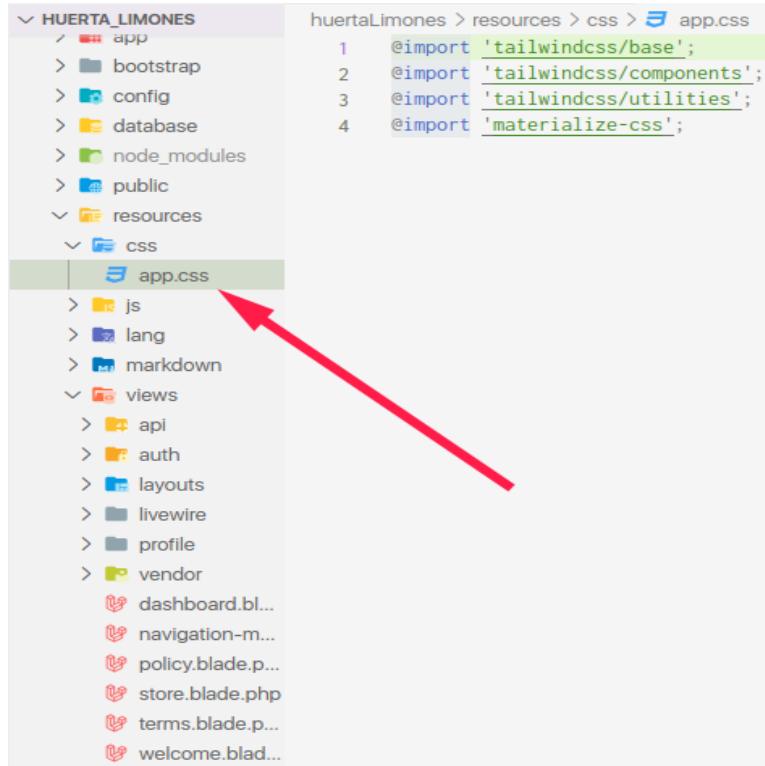
7.3.2 Instalación de Materialize

Para instalara Materialize a tu sistema es correr:

```
$ npm install materialize-css@next
```

Después se busca el archivo: archivo_de_proyecto/resources/css/app.css. Dentro de el archivo se introduce el texto:

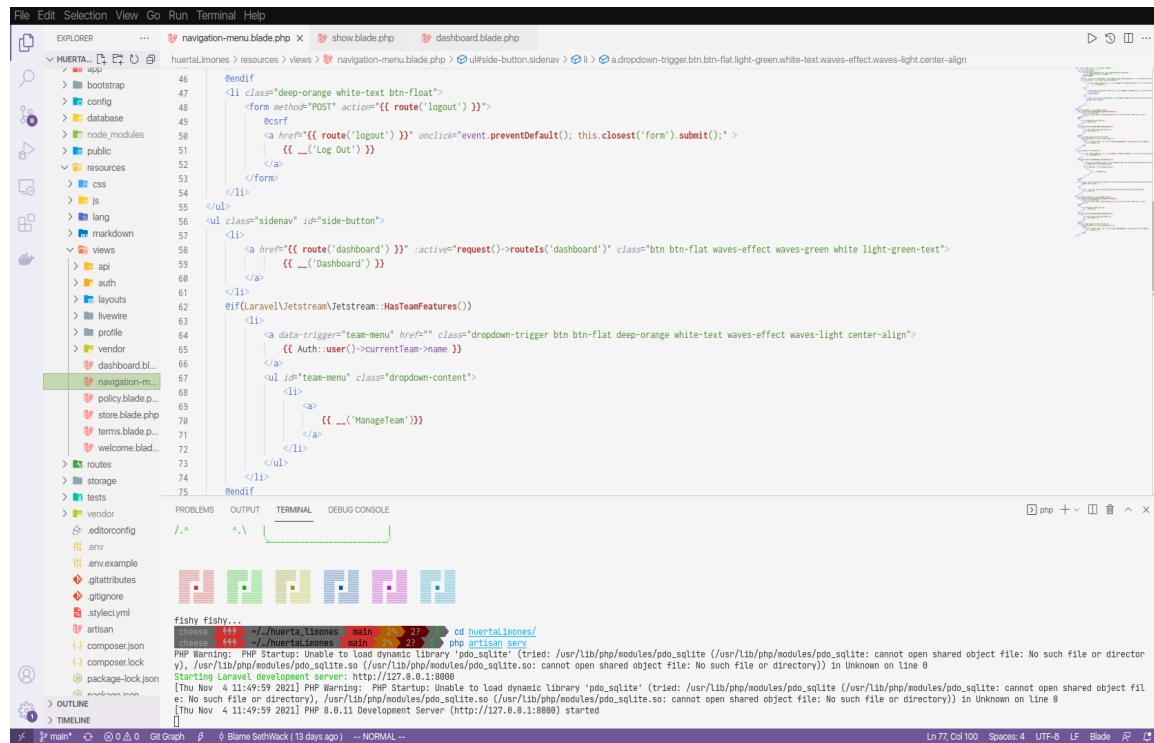
```
@import 'materialize-css';
```



7.1 Imagen de Archivo CSS que Buscar

7.4 Desarrollando el Sistema

Al crear un archivo se puede abrir un editor de texto para poder hacer el desarrollo del sistema. En este caso se usara Microsoft Visual Studio.



```

File Edit Selection View Go Run Terminal Help
File Explorer ... navigation-menu.blade.php show.blade.php dashboard.blade.php
HUELTALIMONES > resources > views > navigation-menu.blade.php > ul>li>a.dropdown-trigger.btn.btn-flat.light-green.white-text.waves-effect.waves-light.center-align
    <ul class="dropdown-trigger" action="{{ route('logout') }}>
        <form method="POST" action="{{ route('logout') }}>
            @csrf
            <a href="{{ route('logout') }}" onclick="event.preventDefault(); this.closest('form').submit();">
                {{ __('Log Out') }}
            </a>
        </form>
    </li>
<ul class="sidenav" id="side-button">
    <li>
        <a href="{{ route('dashboard') }}" active="request">>{{ route('dashboard') }}</a>
    </li>
</ul>
@if(Laravel\Jetstream\HasTeamFeatures())
    <li>
        <a data-triggers="team-menu" href="" class="dropdown-trigger btn btn-flat deep-orange white-text waves-effect waves-light center-align">
            {{ Auth::user()>>currentTeam->name }}
        </a>
        <ul id="team-menu" class="dropdown-content">
            <li>
                <a href="#">{{ __('ManageTeam') }}</a>
            </li>
        </ul>
    </li>
@endif

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```

fishy fishy... ~ /huertalimones main 2% 27% cd huertalimones/
fishy fishy... ~ /huertalimones main 2% 27% php artisan serve
PHP Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlite' (tried: /usr/lib/php/modules/pdo_sqlite: cannot open shared object file: No such file or directory, /usr/lib/php/modules/pdo_sqlite.so: cannot open shared object file: No such file or directory) in Unknown on line 0
Starting Laravel development server: http://127.0.0.1:8000
[Thu Nov 4 11:49:59 2021] PHP Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlite' (tried: /usr/lib/php/modules/pdo_sqlite (/usr/lib/php/modules/pdo_sqlite: cannot open shared object file: No such file or directory, /usr/lib/php/modules/pdo_sqlite.so (/usr/lib/php/modules/pdo_sqlite.so: cannot open shared object file: No such file or directory)) in Unknown on line 0
[Thu Nov 4 11:49:59 2021] PHP 8.0.11 Development Server (http://127.0.0.1:8000) started

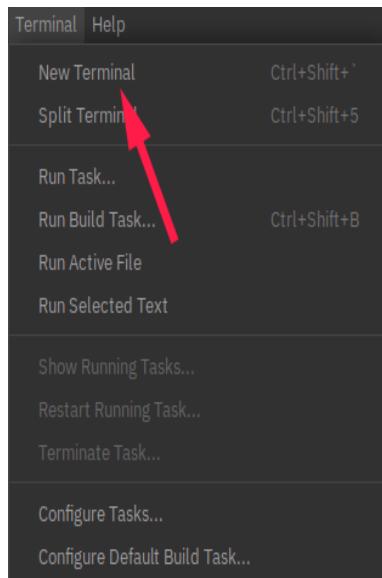
```

Ln 77, Col 100 Spaces: 4 UTF-8 LF Blade ⌂

7.2 Imagen de Visual Studio

7.5 Creación de modelos, Controladores, Vistas

Después de abrir el editor de texto, se abre una terminal



*7.3 Imagen de menú
de Terminal*

Este terminal se abriera en el fólder del proyecto. Después se puede usar el programa de php artisan, que es usado para crear todo lo necesario para trabajar con laravel.

7.5.1 Creación de Modelos

Los Modelos en Laravel, es un archivo que se conecta con la tabla de una base de datos. Todo que se usa base de datos, como: creación, edición, y borrar. Es conectado con el modelo. Para crear un modelo es:

```
$php artisan create:model <nombre_de_modelo>
```

7.5.2 Creación de Controles

Los controladores son los archivos que controlan las rutas de las páginas que un sistema web. Para crear un controlador es:

```
$php artisan make:controller <nombre_de_controlador> --resource
```

En este caso se usa la opción de ‘--resource’ para crear un archivo específico para el controlador de base de datos

7.5.3 Creación de Vistas

Una vista es un archivo en que la persona va a ver en la página web. Es donde el código html y php se va ir. Para crear una vista es:

```
$php artisan make:livewire <nombre_de_vista>
```

7.6 Rutas y Controladores

Las rutas del sistema es como se define cuales son las paginas que se van a ver al anclarlos en un link.

```
Route::get('/', function () {
|   return view('dashboard');
});

Route::middleware(['auth:sanctum', 'verified'])->get('/dashboard', function () {
|   return view('dashboard');
})->name('dashboard');

Route::resource('/blog', BlogController::class);
Route::resource('/store', StoreController::class);
Route::resource('/admin', AdminController::class);
Route::resource('/users', UsersController::class);
Route::resource('/productos', ProductController::class);
Route::resource('/entradas', EntradaController::class);
Route::resource('/salidas', SalidaController::class);
Route::resource('/compras', ComprasController::class);
Route::resource('/carrito', CarritoController::class);
```

7.4 Imagen de Rutas

La ruta es involucrada con el archivo controlador. Después se puede anclar con un botón o con un link.

```
<a href="/" class="btn btn-flat waves-effect waves-green white light-green-text">
|   Home
</a>
```

7.5 Ejemplo de Boton

7.6.1 Controlador

Después de hacer las rutas, se puede configurar el controlador para definir cuales archivos son los que se van a ver.

```
public function index()
{
    $productos = DB::table('productos')->get();
    $types = DB::table('prod_tipos')->get();
    $row_count = 0;
    return view('livewire.store')
        ->with('productos', $productos)
        ->with('types' , $types)
        ->with('row_count', $row_count);
}
```

7.6 Imagen de Vista

Con el controlador, se puede llamar un modelo, de base de datos para hacer consultas. Después se pueden guardar esa información a la vista de la pagina.

```
@foreach ($productos as $product)
    <div class="row hoverable orange lighten-4">
        <div class="col s2"></div>
        <div class="col s8">
            <div class="card horizontal white">
                <div class="card-image">
                    
                ...
```

7.7 Imagen de la vista

En este ejemplo, la tabla del producto es llamado y luego usado para mostrar en la vista de la pagina.

7.7 Definición de Base de Datos

La Base de Datos es la forma en que se van a guardar información de todo el sistema. En si es un grupo de tablas que son conectadas con cada uno usando llaves

para hacer consultas y obtener información que se requiere. Cuando se conecta a la Base de Datos, se tiene que conectar con el servidor. En la pagina de configuracion de laravel se puede definir en cual Base de Datos se tiene que conectar, incluyendo el nombre de la base y el usuario que se va a conectar.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=huerta
DB_USERNAME=serv
DB_PASSWORD=server_login_123
```

8.1 Imagen de definicion de Base

de Datos

Al hacer las migraciones esos datos se conectan con esa Base de Datos.

7.7.1 Modelos

El modelo es una forma para conectar con la Base de Datos. Esta diseñado para interactuar haciendo las creaciones, modificaciones y eliminación de las tablas.

```
namespace App\Models;

use Cviebrock\EloquentSluggable\Sluggable;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Blogs extends Model
{
    use HasFactory;
    use Sluggable;

    protected $fillable=[
        'BLOG_SLUG', 'BLOG_TITLE', 'BLOG_DESC', 'BLOG_TEXT', 'BLOG_IMG', 'USER_ID'
    ];

    public function user(){
        return $this->belongsTo(User::class);
    }

    public function sluggable(): array{
        return [
            'BLOG_SLUG' =>[
                'source' => 'title'
            ]
        ];
    }
}
```

7.8 Imagen de Modelo

7.7.2 Migraciones

Las migraciones son archivos que definen las tablas de una base de datos. Se pueden crear, modificar y borrar tablas usando migraciones. Para crear una migración es:

```
$php artisan make:migration <nombre_de_migración>
```

```
class CreateBlogsTable extends Migration
{
    /**
     * Schema table name to migrate
     * @var string
     */
    public $tableName = 'blogs';
    /**
     * Run the migrations.
     * @table Blogs
     *
     * @return void
     */
    public function up()
    {
        Schema::create('blogs', function (Blueprint $table) {
            $table->id();
            $table->string('BLOG_SLUG');
            $table->string('BLOG_TITLE');
            $table->longText('BLOG_DESC');
            $table->string('BLOG_TEXT');
            $table->string('BLOG_IMG');
            $table->timestamps();

        });
        Schema::table('blogs', function (Blueprint $table){
            $table->unsignedBigInteger('USER_ID');

            $table->foreign('USER_ID')
                ->references('id')->on('users')
                ->onDelete('no action')
                ->onUpdate('no action');
        });
    }
}
```

7.9 Imagen de Migracion

7.7.3 Semillas

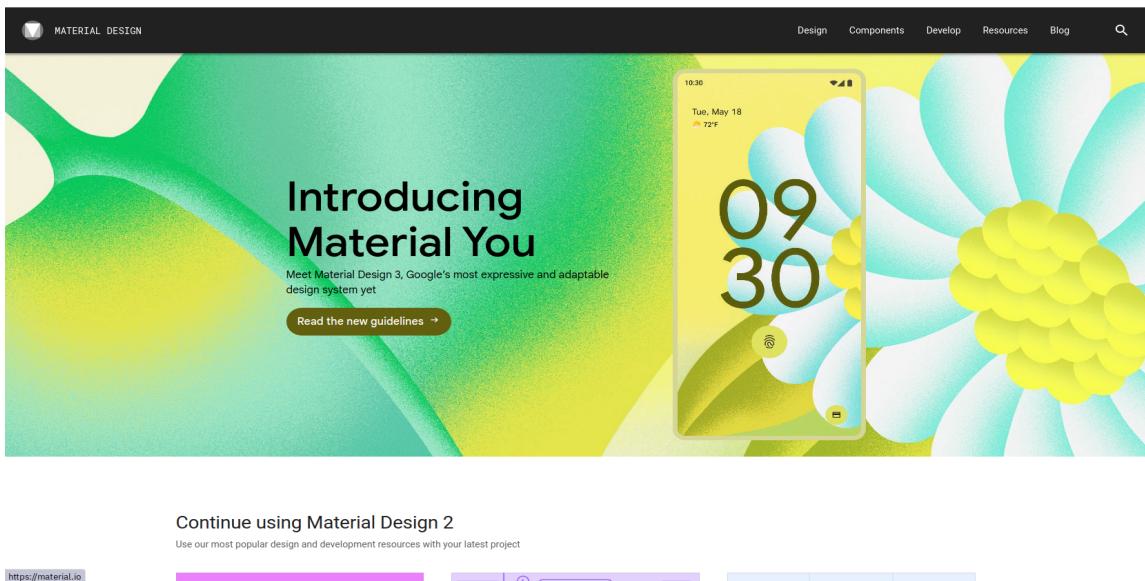
Un Seed o Semilla es una forma para introducir datos a la Base de Datos. Se pueden introducir datos que posiblemente son necesarios y es preferible tenerlos en el sistema.

7.8 Definición de Vistas

Las vistas son las páginas html que serán vistas por el usuario. Los controladores y las rutas mantienen cuales páginas serán vistas y que datos se van a pasar entre ellas.

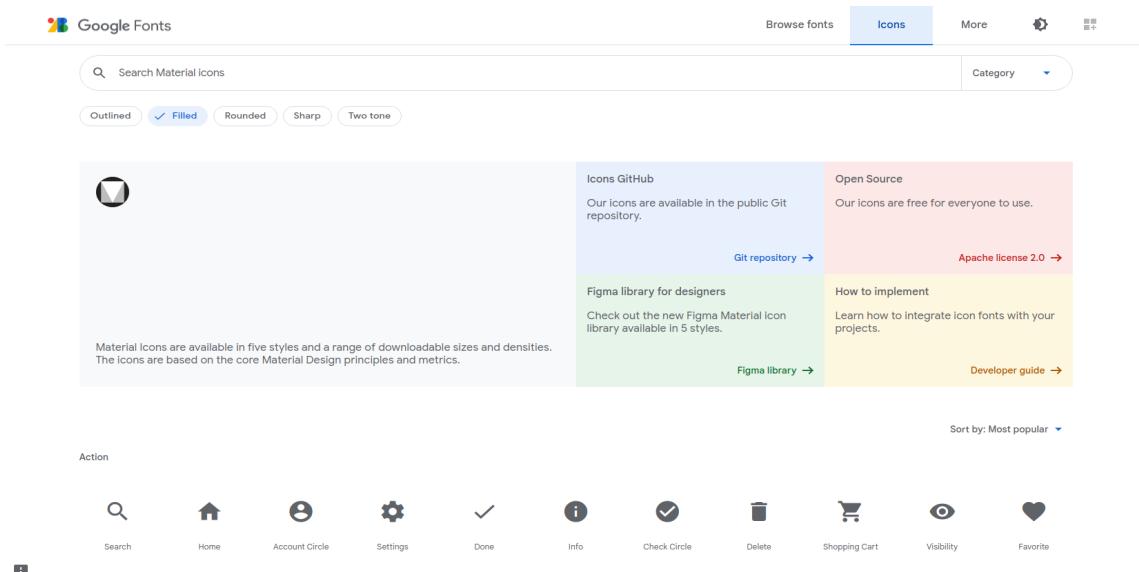
7.8.1 Diseño gráfico

La vista general es definido usando el materialize.css. Es un framework gráfico para tener páginas usando la filosofía de “material” de Google [39]. Así se tendrá un diseño uniforme, y con un fácil reconocimiento visual a páginas afiliadas a Google.



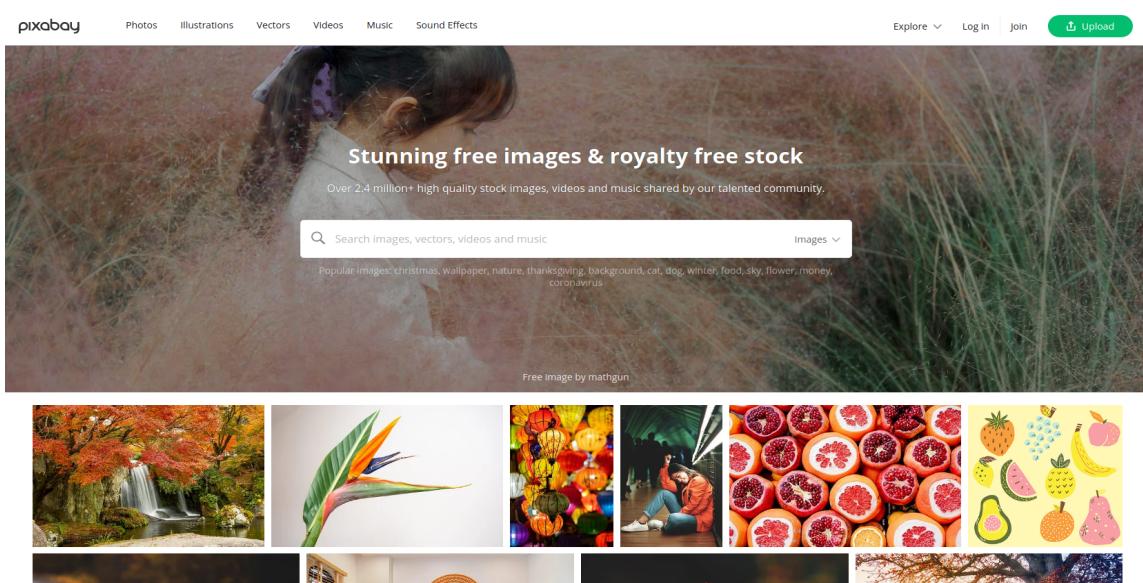
7.11 Página de Material Design de Google

Todos los iconos también son conseguidos en los repositorios de Google. Google da gratuitamente muchos de sus recursos visuales, incluyendo fuentes [40].



7.12 Página de Google Fonts

Imágenes de prueba son obtenidos usando la pagina Pixabay.com [41]. Esta pagina da imágenes gratuitas de alta definición y sin problemas de derechos de autor.



7.13 Página de Google Fonts

7.8.2 Controlador de Vistas

Para mandar llamar una vista primero se crea usando la terminal. Después en el controlador se define cuales son las paginas que se van a ver y como serán implementadas, incluyendo datos de una tabla de la Base de Datos o variables que se pueden usar en la pagina.

```
public function create()
{
    $productos = Producto::select()->get();
    return view('livewire.ent-make')
        ->with('productos', $productos);
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $request->validate([
        'ENT_DATE' => 'required',
        'PROD_ID' => 'required'
    ]);
    Entradas::create([
        'ENT_DATE' => $request->input('ENT_DATE')
    ]);

    $this->createProdEnt($request['PROD_ID']);
    return redirect('/entradas')->with('message', 'Entrada Creado!');
}

public function createProdEnt($prod){
    $getID = Entradas::latest()->first();
    Prod_ents::create([
        'ENT_ID' => $getID['id'],
        'PROD_ID' => $prod
    ]);
    return null;
}
```

7.14 Imagen de controlador

8 Capítulo VIII: Implementación y Pruebas

El framework de Laravel tiene una función incluida para hacer pruebas sin tener que implementar el sistema en linea. Se llama esa función con el comando:

```
$php artisan serv
```

8.1 Usuarios

El usuario tiene que poder crear, modificar, y borrar sus cuentas si quiere. Ciertos usuarios tienen el poder de Admin, que les da entrada a ciertas paginas y el poder de crear Blogs en la pagina.

8.1.1 Creación de Usuario

Para crear un usuario, el cliente tiene que acceder a la pagina de registro.



8.1 Imagen de registro de usuario

En la pagina se introduce el nombre, un email, y la contraseña.

Name	cliente1
Email	cliente@gmail.com
Password	*****
Confirm Password	*****
ALREADY REGISTERED?	
REGISTER	

8.2 Imagen de registro de usuario

Al introducir bien los datos, El usuario sera regresado a la pagina principal pero con el botón de usuario y la del carrito estará en la barra del menú.

8.3 Imagen de Usuario Registrado

De Aquí el usuario puede modificar sus datos de usuario o salirse de su cuenta, y poder acceder su carrito y ver los productos que se va a comprar.

8.1.2 Modificación de Usuario

El usuario puede modificar sus datos en la pagina de usuario.

Informacion de Perfil

Actualizar la informacion de tu cuenta y tu direccion de email.

cliente1

cliente@gmail.com

GUARDAR

Autenticación de dos pasos

Anadir seguridad adicional a tu cuenta usando autenticación de dos pasos

Tu no tienese autenticación de dos paso abilitada.

ABILITAR

Cuando tienes autenticación de dos paso, se dara una llave segura y aleatoria durante la autenticación que sera dado en tu aplicacion de Google en tu telefono.

Actualizar Contraseña

Asegurate que tu contraseña sea larga y alaatoria para se mas seguro/a.

Contraseña reciente

Nueva contraseña

Confirmar contraseña

GUARDAR

Sesiones del navegador

Administre y cierre la sesión de sus sesiones activas en otros navegadores y dispositivos.

Si es necesario, puede cerrar sesión en todas las demás sesiones de su navegador en todos sus dispositivos. Algunas de sus sesiones recientes se enumeran a continuación; sin embargo, esta lista puede no ser exhaustiva. Si cree que su cuenta se ha visto comprometida, también debe actualizar su contraseña.

CERRAR SESIÓN

8.4 Imagen de Modificación

Aquí el usuario puede cambiar su nombre, email, contraseña, activar el autenticación en dos pasos, y eliminar la cuenta.



8.5 Nombre original



8.6 Nombre nuevo



8.7 Nombre Cambiado

8.1.3 Eliminación de Usuario

El usuario puede borrar su cuenta si lo desea, al presionar el botón de eliminar cuenta.

Eliminar cuenta

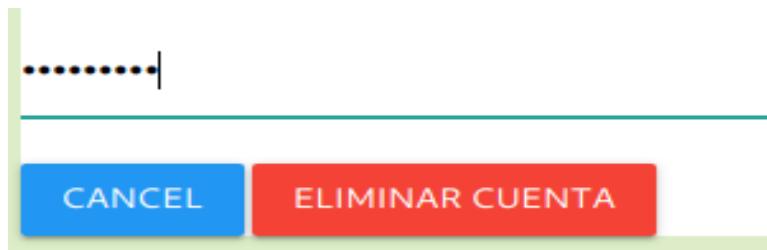
Elimina permanentemente tu cuenta.

Una vez que se elimine su cuenta, todos sus recursos y datos se eliminarán permanentemente. Antes de eliminar su cuenta, descargue cualquier dato o información que desee conservar.

ELIMINAR CUENTA

8.8 Menú de Eliminación

El usuario tendrá que introducir su contraseña para confirmar la eliminación.



8.9 Área de confirmación

8.2 Menú de Admin

El botón de Admin solo es expuesto si el usuario tiene el valor de Admin en su cuenta.



8.10 Botón de Admin

El menú de Admin es para poder ver ciertas tablas de los productos y usuarios en la pagina web.

A screenshot of an admin sidebar. The sidebar has a dark blue header with the text 'Admin side bar' and a back arrow icon. Below this, there is a list of menu items with small icons: 'USUARIOS' (user icon), 'PRODUCTOS' (triangle icon), 'ENTRADAS' (arrow icon), 'SALIDAS' (downward arrow icon), 'COMPRAS' (receipt icon), and 'REPORTES' (flag icon). The main content area is a light green color.

8.11 Menú de Admin

8.2.1 Productos

La pagina de productos es para introducir productos y modificar precios o el nombre del producto.

Crear Producto

Arbolito

Cantidad
\$ 50

Precio
150

IMAGE sprout.jpg

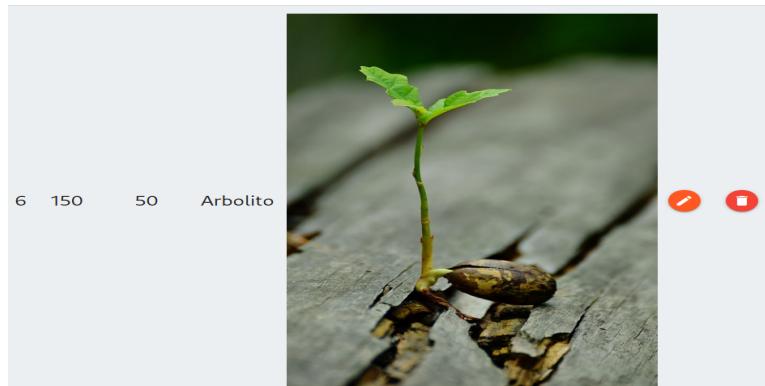
CREAR BACK



8.12 Creación de producto



8.13 Nuevo producto creado



8.14 Nuevo Producto

La pagina de modificación es similar a la de creación pero con los datos ya introducidos.



8.15 Imagen de Edición de Producto

Al borrar un producto se puede al presionar el botón de borrar.



8.16 Producto Borrado

8.2.2 Entradas

Las entradas de un producto son creados al la introducción de un producto y borrados al ser borrados.



The screenshot shows the Admin side bar on the left with the following menu items: USUARIOS, PRODUCTOS, ENTRADAS (which is expanded), SALIDAS, COMPRAS, and REPORTES. The main area is titled "Entradas" and contains a table with one row. The table has columns: Id, Entrada, Producto ID, and Delete. The data in the table is: Id: 2, Entrada: 2021-11-27, Producto ID: 10, and a red delete icon in the Delete column.

Id	Entrada	Producto ID	Delete
2	2021-11-27	10	

8.17 Pagina de Entradas

En si se pueden crear mas entradas si se quiere, y también borrar las entradas.

8.2.3 Salidas

Las salidas son similares. Explican cuando salieron productos que ya no están listos para ser vendidos.



The screenshot shows the Admin side bar on the left with the same menu items as before. The main area is titled "Salidas" and contains a green success message "Salida Creado!". Below it is a table with one row. The table has columns: Id, Salida, Producto ID, and Delete. The data in the table is: Id: 2, Salida: 2021-11-27, Producto ID: 1, and a red delete icon in the Delete column.

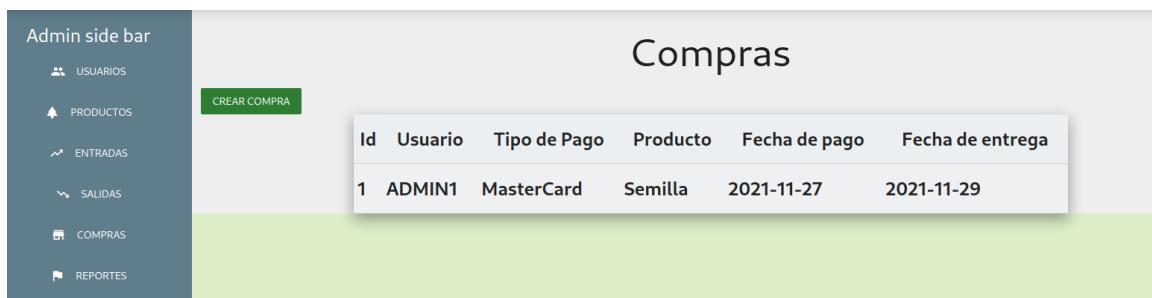
Id	Salida	Producto ID	Delete
2	2021-11-27	1	

8.18 Imagen de Salidas

Se pueden crear salidas si se venden productos afuera de el sistema. Ejemplo seria que se vendan limones en la huerta is se tiene que dar la actualización de el producto.

8.2.4 Compras

Las compras son una lista, de los clientes y sus productos vendidos en la pagina. Es preferido no borrar o modificar estas tablas. En este caso se van a crear los datos para las pruebas.



The screenshot shows a dark blue sidebar titled 'Admin side bar' with the following menu items: USUARIOS, PRODUCTOS, ENTRADAS, SALIDAS, COMPRAS (which is highlighted in green), and REPORTES. To the right of the sidebar, the main content area has a title 'Compras' and a green button labeled 'CREAR COMPRA'. Below this is a table with the following data:

	Id	Usuario	Tipo de Pago	Producto	Fecha de pago	Fecha de entrega
1	ADMIN1	MasterCard	Semilla	2021-11-27	2021-11-29	

8.19 Imagen de Compras

8.2.5 Reportes

Los reportes es para que el admin pueda tener una visión general del sistema y tener imprenta si es requerido los reportes en pdf.

Reporte			
	id	Dia Creado	link a PDF
	27	2021-11-27 08:26:16	NULL
	28	2021-11-27 08:26:45	NULL
	29	2021-11-27 08:27:00	NULL
	30	2021-11-27 08:27:43	NULL
	31	2021-11-27 08:28:54	NULL
	32	2021-11-27 08:29:35	NULL

8.20 Imagen de pagina de PDF

Admin side bar		Exportar Reporte
		CREAR REPORTE
<input checked="" type="checkbox"/> USUARIOS	✓ usuarios	
<input type="checkbox"/> PRODUCTOS	☐ productos	
<input type="checkbox"/> ENTRADAS	✓ Pagos	
<input type="checkbox"/> SALIDAS	☐ Entradas	
<input type="checkbox"/> COMPRAS	✓ Salidas	
<input type="checkbox"/> REPORTES	☐ Blogs	

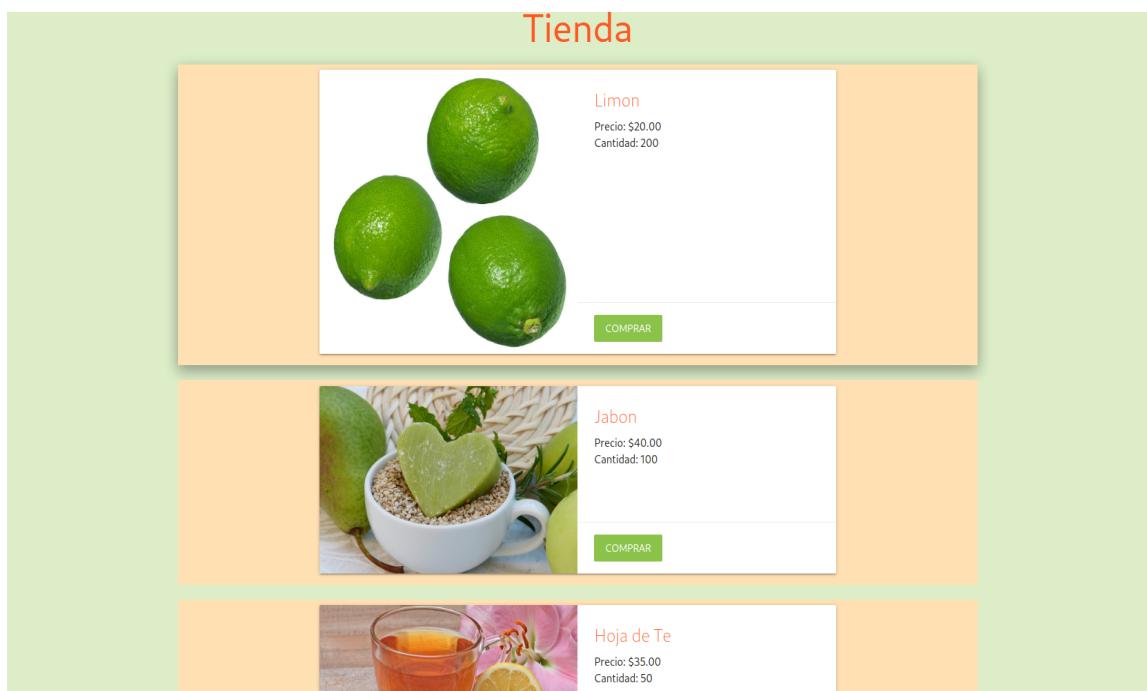
8.21 Imagen de Selección de Reportes

Admin side bar		Reporte de Usuarios			
		Reporte de Pagos			
		Reporte de Salidas			
<input checked="" type="checkbox"/> USUARIOS	id	name	email	admin	
	1	ADMIN1	admin@gmail.com	1	
<input type="checkbox"/> PRODUCTOS	2	userTest	test@gmail.com	0	
<input type="checkbox"/> ENTRADAS	6	virtual	virtual@gmail.com	0	
<input type="checkbox"/> SALIDAS	Id	Usuario	Tipo de Pago	Producto	Fecha de pago
<input type="checkbox"/> COMPRAS	1	ADMIN1	MasterCard	Semilla	2021-11-27
<input type="checkbox"/> REPORTES	Id	Salida			Fecha de entrega
	2	2021-11-27			2021-11-29
	Id	Salida			Producto ID
	2	2021-11-27			1
	MAKE PDF				

8.22 Imagen de resultados de reportes

8.3 Pagina de Tienda

La pagina de Tienda es la pagina principal del sistema. Es donde todos los productos serán listados. Aquí los productos se pueden comprar, y luego añadido al carrito de los clientes.



8.23 Imagen de pagina de Tienda

8.4 Hacer una compra

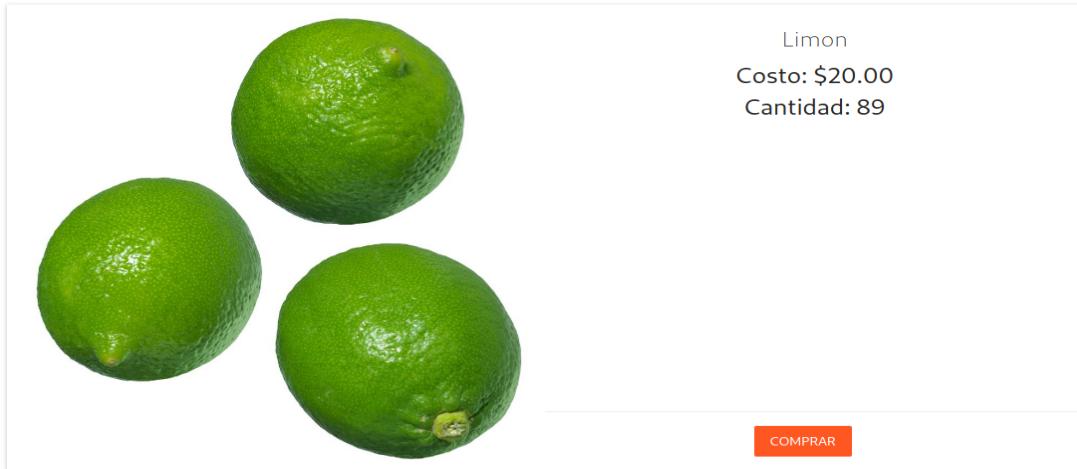
El usuario compra un producto al presionar el botón de comprar, Aquí se decide la cantidad que se quiere y luego es añadido al carrito.



8.24 Imagen de compra de Producto

En la pagina de carrito se mira una lista de los productos que se van a comprar. Al decidir de comprar, el usuario presiona el botón de comprar y luego es redirigido a la pagina de compras.

Carrito de ADMIN1



8.25 Imagen de pagina de Carrito

En la pagina de compras, el usuario añade sus datos de pago y luego se hace la compra.

Comprar Limon para: ADMIN1

Precio: 20

Tipo de Tarjeta

Numero de Tarjeta
123455623

Fecha de expiracion

Nombre en la Tarjeta
cheese

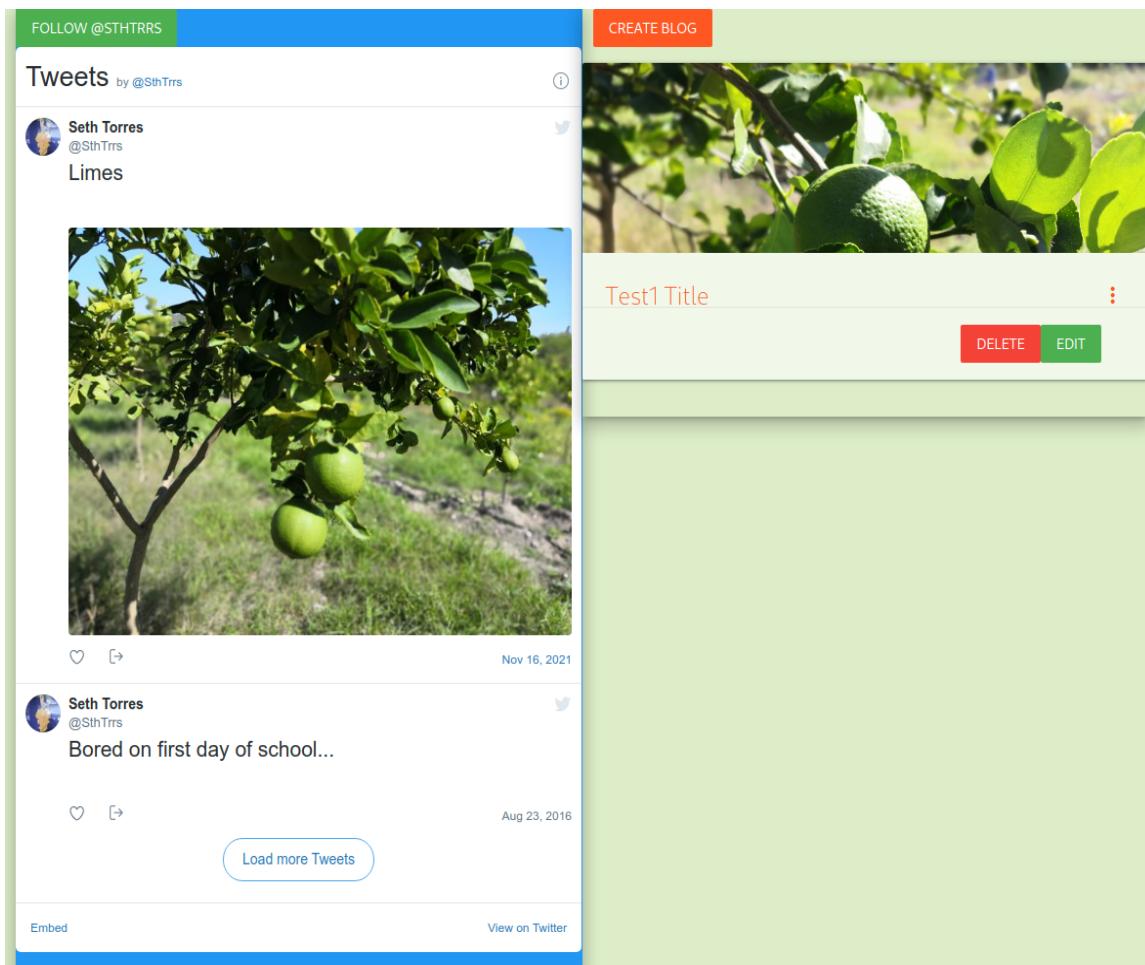
COMPRAR!

A screenshot of a payment form. It shows four lime images on the left. To the right, it displays the price "Precio: 20". Below that, there are fields for "Tipo de Tarjeta" (Credit Card Type), "Numero de Tarjeta" (Card Number) containing "123455623", "Fecha de expiracion" (Expiration Date), and "Nombre en la Tarjeta" (Name on Card) containing "cheese". At the bottom is a red "COMPRAR!" button.

8.26 Imagen de pago de Producto

8.5 Blogs

Los blogs es un sistema pequeño creado para que el admin haga actualizaciones de la huerta. Ejemplo seria: Si hay nuevos limones listos para vender, se puede poner una nota para dar la noticia.



8.27 Imagen de Blog del sistema

En este sistema esta integrado redes sociales, con un feed de Twitter, y links de otras redes sociales como Whatsapp, Facebook, y email.

8.5.1 Creación de Blog

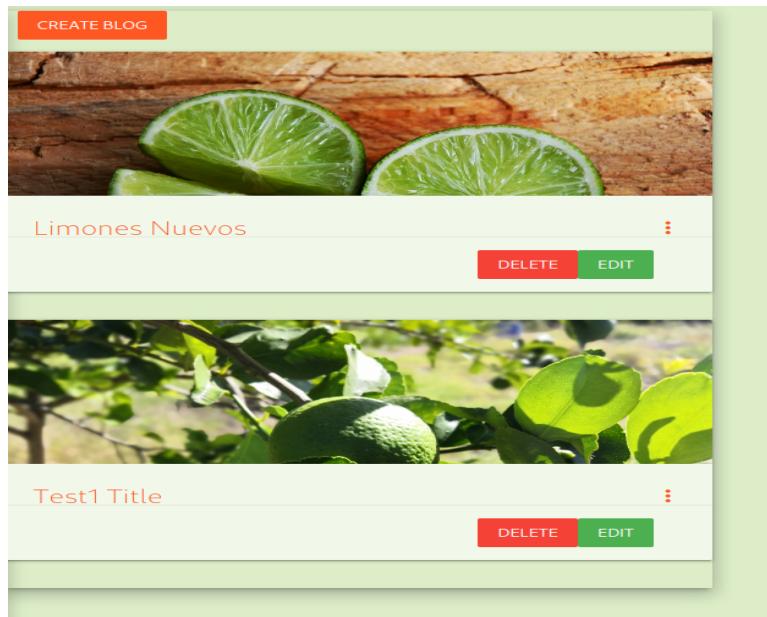
Solo los usuarios de admin tienen el poder de crear blogs en la pagina. El botón de creación de blog, de modificación, y eliminación solo se pueden ver si eres un admin.

The screenshot shows a web-based form titled "Crear Blog" (Create Blog). The form has a light green header and a light orange body. It contains the following fields:

- Title:** A text input field containing "Limones Nuevos".
- Description:** A text input field containing "Llegaron nuevos limones! Listos para los días festivos navideños de diciembre."
- description:** A text input field containing "limones para la fiesta navidena."
- FILE:** A file upload input field with the value "lime-2133091_960_720.jpg".
- SUBMIT:** A green button with a white arrow icon and the text "SUBMIT".

8.28 Imagen de creación de Blog

Al crear un blog, con una imagen se la pagina se actualizara con el nuevo blog.



8.29 Imagen de resultados de creación de Blog

8.5.2 Modificación de Blog

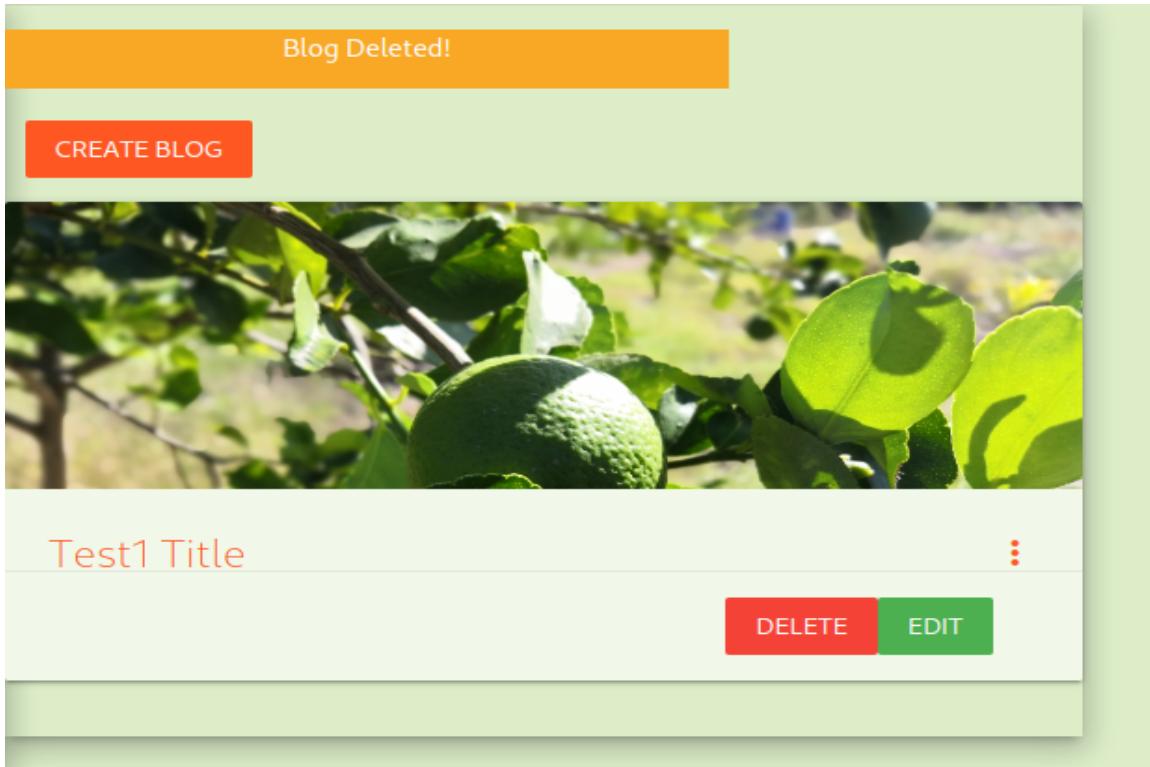
Al modificar el blog se va a una pagina de modificación, y con los datos del blog.

A screenshot of a blog editing interface titled 'Editar Blog'. The main area contains a single blog post. The title field is filled with 'Limones Nuevos'. The description field contains the text 'Llegaron nuevos limones! Listos para los días festivos navideños de diciembre.' The summary field contains the text 'limones para la fiesta navidena.'. At the bottom left is a green 'EDITAR' button with a pencil icon.

8.30 Imagen de edición de Blog

8.5.3 Eliminación de Blog

Se pueden borrar los blogs creados se se requiere.



8.31 Imagen de eliminación de Blog

9 Capítulo IX: Conclusiones

10 Bibliografías

- [1] Definición de Huerta [Internet]. Definición.de. [cited 2021Oct10]. Available from: <https://definicion.de/huerta/>
- [2] Web page [Internet]. Web Page Definition. [cited 2021Oct10]. Available from: <https://techterms.com/definition/webpage>
- [3] Server [Internet]. Paessler. [cited 2021Oct10]. Available from: <https://www.paessler.com/it-explained/server>
- [4] What is a front end (in a website) - definition & development [Internet]. What is a Front End (In a Website) - Definition & Development. [cited 2021Oct10]. Available from: <https://airfocus.com/glossary/what-is-a-front-end/>
- [5] What is a back end (in a website)? definition & faqs [Internet]. What Is A Back End (In A Website)? Definition & FAQs. [cited 2021Oct10]. Available from: <https://airfocus.com/glossary/what-is-a-back-end/>
- [6] HTML introduction [Internet]. Introduction to HTML. [cited 2021Oct10]. Available from: https://www.w3schools.com/html/html_intro.asp
- [7] CSS introduction. [cited 2021Oct10]. Available from: https://www.w3schools.com/css/css_intro.asp
- [8] Nixon R, Carreño FM. Aprender PHP, mysql y javascript: Con jquery, CSS Y HTML5. 5th ed. Barcelona: Marcombo; 2019.
- [9] What is php? [Internet]. php. [cited 2021Oct10]. Available from: <https://www.php.net/manual/en/intro-whatis.php>

- [10] Silberschatz A, Korth HF, Sudarshan S. Database system concepts. Boston: McGraw-Hill; 2006.
- [11] What is mysql? A beginner-friendly explanation [Internet]. Kinsta®. 2021 [cited 2021Oct10]. Available from: <https://kinsta.com/knowledgebase/what-is-mysql/>
- [12] What is javascript? - learn web development: MDN [Internet]. Learn web development | MDN. [cited 2021Oct10]. Available from: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- [13] What is a web framework? [Internet]. GoodFirms. 2020 [cited 2021Oct10]. Available from: <https://www.goodfirms.co/glossary/web-framework/>
- [14] Installation [Internet]. Laravel. [cited 2021Oct10]. Available from: <https://laravel.com/docs/8.x>
- [15] # introduction [Internet]. Laravel Jetstream. [cited 2021Oct10]. Available from: <https://jetstream.laravel.com/1.x/introduction.html>
- [16] Livewire [Internet]. Laravel. [cited 2021Oct10]. Available from: <https://laravel-livewire.com/>
- [17] What is full stack development ? [Internet]. GeeksforGeeks. 2019 [cited 2021Oct10]. Available from: <https://www.geeksforgeeks.org/what-is-full-stack-development/>
- [18] Rapidly build modern websites without ever leaving your HTML. [Internet]. Tailwind CSS. [cited 2021Oct10]. Available from: <https://tailwindcss.com/>

- [19] Atlassian. What is version control: Atlassian Git Tutorial [Internet]. Atlassian. [cited 2021Oct10]. Available from: <https://www.atlassian.com/git/tutorials/what-is-version-control>
- [20] Mijacobs. What is Git? - azure DevOps [Internet]. Azure DevOps | Microsoft Docs. [cited 2021Oct11]. Available from: <https://docs.microsoft.com/en-us/devops/develop/git/what-is-git>
- [21] Brown K. What is github, and what is it used for? [Internet]. How. How-To Geek; 2019 [cited 2021Oct11]. Available from: <https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/>
- [22] Mijacobs. What is agile? - azure DevOps [Internet]. Azure DevOps | Microsoft Docs. [cited 2021Oct10]. Available from: <https://docs.microsoft.com/en-us/devops/plan/what-is-agile>
- [23] Mijacobs. What is Scrum? - azure DevOps [Internet]. ? - Azure DevOps | Microsoft Docs. [cited 2021Oct10]. Available from: <https://docs.microsoft.com/en-us/devops/plan/what-is-scrum>
- [24] What is staruml [Internet]. What is StarUML - StarUML Documentation. [cited 2021Oct11]. Available from: https://documentation.help/StarUML/what_is_staruml.htm
- [25] Leokhoa. Documentation [Internet]. Laragon. 2019 [cited 2021Oct11]. Available from: <https://laragon.org/docs/>
- [26] Jira [Internet]. ProductPlan. 2021 [cited 2021Oct11]. Available from: <https://www.productplan.com/glossary/jira/>

- [27] Santos JMD. Microsoft Project Pros & Cons 2021 [Internet]. Project. 2021 [cited 2021Oct11]. Available from: <https://project-management.com/the-pros-and-cons-of-using-microsoft-project-software/>
- [28] Gantt.com. [cited 2021Oct11]. Available from: <https://www.gantt.com/>
- [29] Austin. What is Microsoft Visio and what does it do? [Internet]. groovyPost. 2021 [cited 2021Oct11]. Available from: <https://www.groovypost.com/reviews/microsoft-visio-explained/>
- [30] Advocacy Oof. The effects of the COVID-19 pandemic on small businesses [Internet]. SBA's Office of Advocacy. 2021 [cited 2021Nov27]. Available from: <https://advocacy.sba.gov/2021/03/02/the-effects-of-the-covid-19-pandemic-on-small-businesses/>
- [31] Tandon M. Benefits of having an online business or moving business online [Internet]. CedCommerce Blog. 2021 [cited 2021Nov27]. Available from: <https://cedcommerce.com/blog/benefits-of-having-an-online-business/>
- [32] Pmoinformatica.com. ¿Qué es un requerimiento funcional? [Internet]. La Oficina de Proyectos de Informática. [cited 2021Oct15]. Available from: <http://www.pmoinformatica.com/2018/05/que-es-requerimiento-funcional.html>
- [33] Requisitos no funcionales [Internet]. EcuRed. [cited 2021Oct15]. Available from: https://www.ecured.cu/Requisitos_no_funcionales
- [34] Ian. What is MySQL Workbench? [Internet]. Database.guide. 2020 [cited 2021Nov27]. Available from: <https://database.guide/what-is-mysql-workbench/>

- [35] Soni S. What is composer for PHP and how to install it [Internet]. Code Envato Tuts+. Envato Tuts; 2020 [cited 2021Nov27]. Available from: <https://code.tutsplus.com/tutorials/what-is-composer-for-php-and-how-to-install-it--cms-35160>
- [36] What is javascript? - learn web development: MDN [Internet]. Learn web development | MDN. [cited 2021Nov27]. Available from: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- [37] Node.js. What is NPM? [Internet]. Node.js. 2011 [cited 2021Nov27]. Available from: <https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/>
- [38] About [Internet]. Materialize. [cited 2021Nov27]. Available from: <https://materializecss.com/about.html>
- [39] What is material design? [Internet]. The Interaction Design Foundation. [cited 2021Nov27]. Available from: <https://www.interaction-design.org/literature/topics/material-design>
- [40] Google G. Google Fonts [Internet]. Google fonts. Google; [cited 2021Nov27]. Available from: <https://fonts.google.com/>
- [41] FAQ - pixabay [Internet]. [cited 2021Nov27]. Available from: <https://pixabay.com/service/faq/>

11 Glosario de Términos

12 Anexos

12.1 Código Fuente

Código fuente: https://github.com/SethWack/huerta_limones

12.2 Hojas de Referencias

12.2.1 Laravel

Cheatography

Laravel Cheat Sheet
by bernattorras via cheatography.com/19670/cs/2647/

Artisan	Routes
php artisan routes	Ruta simple Route::get('/',function(){ return View::make('hello'); });
php artisan controller:make UserController	Ruta amb paràmetres Route::get('posts/{id}',function(\$id){ return View::make('post.single')->with('id', \$id); });
// Migrations	Ruta Controlador + mètode Route::get('post/all', array('uses' => 'PostController@all', 'as' => 'post.all'));
php artisan migrate:create create_users_table	Ruta nominal Route::get('post/{id}', array('uses' => 'PostController@single', 'as' => 'get.post.single'))->where('id', '[1-9][0-9]*');
php artisan migrate:create create_users_table --create=users	Ruta + validació RegEX Route::get('post/{id}', array('uses' => 'PostController@single', 'as' => 'get.post.single'))->where('id', '[1-9][0-9]*');
php artisan migrate	Ruta POST Route::post('post', array('uses' => 'PostController@create', 'as' => 'post.post.create'));
php artisan migrate:rollback	Ruta Resource Route::resource('post', 'PostController');
php artisan migrate:refresh	Route::resource('post', 'PostController', array('except' => 'show'));
// Seed	Route::resource('post', 'PostController', array('only' => 'show'));
php artisan generate:seed posts	Filtres
php artisan db:seed	Route::get('post/create', array('uses' => 'PostController@create', 'as' => 'post.create', 'before' => 'auth'));
php artisan migrate:refresh --seed	Grups
php artisan db:seed --class=PostsTableSeeder	Route::group(array('before' => 'auth'), function(){ // Route:: ... });
// Generators	Prefixs
php artisan generate:resource post --fields="title:string, body:text"	Route::group(array('prefix' => 'admin'), function(){ // Route:: ... });
php artisan generate:pivot categories users	
Migrations	Blade functions
...	@if(count(\$posts) @foreach(\$posts as \$post) <p>{{ \$post->title }}</p> @endforeach @endif
public function up(){ Schema::create('users', function(Blueprint \$table){ \$table->increments('id'); \$table->integer('role'); \$table->string('email')->unique(); \$table->string('password', 60); \$table->rememberToken(); \$table->timestamps(); }); } public function down(){ Schema::drop('users'); } ...	
Seeds (faker)	
...	
User::create(['email' => \$faker->email(), 'password' => \$faker-> md5()]); ...	



By **bernattorras**
cheatography.com/bernattorras/

Published 9th October, 2014.
Last updated 13th November, 2014.
Page 1 of 2.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

12.2.2 HTML

HTML4 Document Outline		HTML4 Lists		HTML4 Language Attributes	
<!DOCTYPE>	Version of (X)HTML		Ordered list	dir	lang
<html>	HTML document		Unordered list	Language Attributes may not be used in base, br, frame, frameset, hr, iframe, param or script elements.	
<head>	Page information		List item		
<body>	Page contents	<dl>	Definition list		
HTML4 Comments		<dt>	Definition term	HTML4 Keyboard Attributes	
<!-- Comment Text -->		<dd>	Term description	accesskey	tabindex
HTML4 Page Information		HTML4 Forms		HTML4 Events	
<base>	Base URL	<form>	Form	Window Events	
<meta>	Meta data	<fieldset>	Collection of fields	onload	onunload
<title>	Title	<legend>	Form legend	Form Events	
<link>	Relevant resource	<label>	Input label	onblur	onreset
<style>	Style resource	<input>	Form input	onchange	onselect
<script>	Script resource	<select>	Drop-down box	onfocus	onsubmit
HTML4 Document Structure		<optgroup>	Group of options	Keyboard Events	
<h1> to <h6>	Headings	<option>	Drop-down options	onkeydown	onkeyup
<div>	Page section	<textarea>	Large text input	onkeypress	
	Inline section	<button>	Button	Mouse Events	
<p>	Paragraph			onclick	onmouseout
 	Line break	<table>	Table	ondblclick	onmouseover
<hr>	Horizontal rule	<caption>	Caption	onmousedown	onmouseup
HTML4 Links		<thead>	Table header		
	Page link	<tbody>	Table body		
	Email link	<tfoot>	Table footer		
	Anchor	<colgroup>	Column group		
	Link to anchor	<col>	Column		
HTML4 Text		<tr>	Table row		
		<th>	Header cell		
		<td>	Table cell		
HTML4 Images		HTML4 Objects		HTML4 Core Attributes	
			Image	class	style
		<map>	Image Map	id	title
		<area>	Area of Image Map	Core Attributes may not be used with base,	

12.2.3 PHP

PHP Cheat Sheet by Dave Child (DaveChild) via cheatography.com/1/cs/2/		
PHP Array Functions	PHP Filesystem Functions	Regular Expressions Syntax
array_diff (arr1, arr2 ...)	clearstatcache ()	^ Start of string
array_filter (arr, function)	copy (source, dest)	\$ End of string
array_flip (arr)	fclose (handle)	. Any single character
array_intersect (arr1, arr2 ...)	fgets (handle, len)	(a b) a or b
array_merge (arr1, arr2 ...)	file (file)	(...) Group section
array_pop (arr)	filemtime (file)	[abc] In range (a, b or c)
array_push (arr, var1, var2 ...)	filesize (file)	[^abc] Not in range
array_reverse (arr)	file_exists (file)	\s White space
array_search (needle, arr)	fopen (file, mode)	a? Zero or one of a
array_walk (arr, function)	fread (handle, len)	a* Zero or more of a
count (count)	fwrite (handle, str)	a*? Zero or more, ungreedy
in_array (needle, haystack)	readfile (file)	a+ One or more of a
<hr/>		
PHP String Functions	PHP Date and Time Functions	Pattern Modifiers
crypt (str, salt)	checkdate (month, day, year)	g Global match
explode (sep, str)	date (format, timestamp)	i * Case-insensitive
implode (glue, arr)	getdate (timestamp)	m * Multiple lines
nl2br (str)	mktime (hr, min, sec, month, day, yr)	s * Treat string as single line
sprintf (fmt, args)	strftime (formatstring, timestamp)	x * Allow comments and whitespace in pattern
strip_tags (str, allowed_tags)	strtotime (str)	
str_replace (search, replace, str)	time ()	
strpos (str, needle)	<hr/>	
strrev (str)	PHP Regular Expressions Functions	
strstr (str, needle)	ereg (pattern, str)	
strtolower (str)	split (pattern, str)	
strtoupper (str)	ereg_replace (pattern, replace, str)	
substr (string, start, len)	preg_grep (pattern, arr)	
<hr/>		
	By Dave Child (DaveChild) cheatography.com/davechild/ aloneonahill.com	Published 19th October, 2011. Last updated 29th February, 2020. Page 1 of 2.
<hr/>		
Sponsored by Readable.com Measure your website readability! https://readable.com		

13.3 Hoja de Referencia PHP

12.2.4 MySQL

MySQL Data Types		MySQL Type Conversion		MySQL String Functions (cont)	
CHAR	String (0 - 255)	BINARY 'string'		LENGTH	REPEAT
VARCHAR	String (0 - 255)	CAST (expression AS datatype)		CHAR_LENGTH	REVERSE
TINYTEXT	String (0 - 255)	CONVERT (expression, datatype)		BIT_LENGTH	INSERT
TEXT	String (0 - 65535)			LOCATE	ELT
BLOB	String (0 - 65535)			INSTR	FIELD
MEDIUMTEXT	String (0 - 16777215)			LPAD	LCASE
MEDIUMBLOB	String (0 - 16777215)			RPAD	UCASE
LONGTEXT	String (0 - 4294967295)			LEFT	LOAD_FILE
LONGBLOB	String (0 - 4294967295)			RIGHT	QUOTE
TINYINT x	Integer (-128 to 127)				
SMALLINT x	Integer (-32768 to 32767)				
MEDIUMINT x	Integer (-8388608 to 8388607)				
INT x	Integer (-2147483648 to 2147483647)				
BIGINT x	Integer (-9223372036854-775808 to 922337203-6854775807)				
FLOAT	Decimal (precise to 23 digits)				
DOUBLE	Decimal (24 to 53 digits)				
DECIMAL	"DOUBLE" stored as string				
DATE	YYYY-MM-DD				
DATETIME	YYYY-MM-DD HH:MM:SS				
TIMESTAMP	YYYYMMDDHHMMSS				
TIME	HH:MM:SS				
ENUM	One of preset options				
SET	Selection of preset options				
Integers (marked x) that are "UNSIGNED" have the same range of values but start from 0 (i.e., an UNSIGNED TINYINT can have any value from 0 to 255).					
MySQL Grouping Functions					
AVG	MAX				
BIT_AND	STD				
BIT_OR	STDDEV				
COUNT	SUM				
GROUP_CONCAT	VARIANCE				
	MIN				
MySQL Mathematical Functions					
ABS	COS				
SIGN	SIN				
MOD	TAN				
FLOOR	ACOS				
CEILING	ASIN				
ROUND	ATAN, ATAN2				
DIV	COT				
EXP	RAND				
LN	LEAST				
LOG, LOG2, LOG10	GREATEST				
POW	DEGREES				
POWER	RADIANS				
SQRT	TRUNCATE				
	PI				
MySQL String Functions					
ASCII	SUBSTRING				
ORD	MID				
CONV	SUBSTRING_INDEX				
BIN	LTRIM				
OCT	RTRIM				
HEX	TRIM				
CHAR	SOUNDEX				
CONCAT	SPACE				
CONCAT_WS	REPLACE				
MySQL Date and Time Functions					
DAYOFWEEK	DATE_SUB				
WEEKDAY	ADDDATE				
DAYOFMONTH	SUBDATE				
DAYOFYEAR	EXTRACT				
MONTH	TO_DAYS				
DAYNAME	FROM_DAYS				
MONTHNAME	DATE_FORMAT				
QUARTER	TIME_FORMAT				
WEEK	CURRENT_DATE				
YEAR	CURRENT_TIME				
YEARWEEK	NOW				
HOUR	SYSDATE				
MINUTE	UNIX_TIMESTAMP				
SECOND	FROM_UNIXTIME				
PERIOD_ADD	SEC_TO_TIME				
PERIOD_DIFF	TIME_TO_SEC				
	DATE_ADD				
MySQL Control Flow Functions					
IF	NULLIF				
IFNULL					

13.4 Hoja de Referencia MySQL

12.2.5 Git

Git Cheat Sheet by Sam Collett (SamCollett) via cheatography.com/196/cs/104/		
Create Git Repository	Git - History	Git - Remote Update / Publish
From existing directory	Show all commits	List remotes
cd project_dir	git log	git remote -v
git init	Short Format	Show information
git add .	git log --pretty=short	git remote show remote
From other repository	Patches	Add remote
git clone existing_dir new_dir	git log -p	git remote add path/url
git clone git://github.com/user/repo.git	Show file commits	Fetch changes
git clone https://github.com/user/repo.git	git log file	git fetch remote
	Show directory commits	Fetch + merge
	git log dir/	git pull remote branch
Git - Local Changes	Stats	Publish local to remote
Changed in working directory	git log --stat	git push remote branch
git status	Who changed file	Delete remote branch
Tracked file changes	git blame file	git push remote :branch
git diff		Publish tags
Add changed files	Git - Merge/Rebase	git push --tags
git add file1 file2 file3	Merge branch into current	
	git merge branch	Git - Branching/Tagging
Remove file	Rebase into branch	List branches
git rm file	git rebase branch	git branch
git rm dir/-r (recursive under directory)	git rebase master branch	Switch to branch
See files ready for commit	Abort rebase	git checkout branch
git diff --cached	git rebase --abort	Create new branch
Commit changes	Merge tool to solve conflicts	git branch new
git commit	git mergetool	Create branch from existing
git commit -m "My message"	Conflicts against base file	git branch new existing
git commit -a -m "My Message"	git diff --base file	Delete branch
(tracked files only, auto add)	Diff other users changes	git branch -d branch
Change last commit	git diff --theirs file	Tag current commit
git commit --amend	Diff your changes	git tag tag-name
Revert changes to file	git diff --ours file	
git checkout -- file	After resolving conflicts	
Revert changes (new commit)	git rebase --continue	
git revert HEAD		
Return to last committed state		
git reset --hard HEAD		
<hr/>		
 By Sam Collett (SamCollett) cheatography.com/samcollett/ www.texotela.co.uk		
<hr/> Published 23rd November, 2011. Last updated 1st June, 2014. Page 1 of 1.		
<hr/> Sponsored by CrosswordCheats.com Learn to solve cryptic crosswords! http://crosswordcheats.com		

13.5 Hoja de Referencia Git