



# PROYECTO DE TITULACIÓN

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

Para obtener el Título de Ingeniero en Sistemas Computacionales

Alumno: Seth Aldanid Torres Delgado

Asesor: ING. Pedro García Guerrero

Co-Asesor: M.C. María Estela Vega Guerrero

## 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 mis asesores ING. Pedro García Guerrero, y a M.C. María Estela Vega Guerrero. que se dieron tiempo para poder guiarme en este proceso.*

*Agradezco a mi familia por tener la paciencia durante estos años de carrera. Especialmente durante estos meses pesados que tuvimos. Yo les doy todas mis gracias por ser los mejores padres y hermanas que alguien quisiera tener.*

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

- Seth Torres

# Índice

---

Agradecimientos .....	I
Índice .....	II
Índice de Tablas y Figuras .....	VIII
Capítulo 1: Introducción.....	1
1.1    Antecedentes.....	1
1.2    Planteamiento del Problema.....	2
1.3    Hipótesis.....	3
Objetivos .....	3
1.4    Generales.....	3
1.4.1    Específicos .....	3
1.5    Alcances y Metas.....	4
Capítulo 2: Marco Teórico y Metodológico .....	5
2.1    Antecedentes de la Investigación.....	5
2.2    Marco Metodológico.....	8
2.3    Justificación de Investigación.....	9
Capítulo 3: 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
Capítulo 4: Análisis .....	13
4.1 Requerimientos.....	13
4.1.1 Funcionales .....	13
4.1.2 No Funcionales.....	13
4.2 Diagrama de Caso Uso.....	15
4.3 Diagrama Entidad Relación.....	16
Capítulo 5: Diseño .....	17
5.1 Bosquejos.....	17
5.1.1 Bosquejo de Página Principal .....	18
5.1.2 Bosquejo de Página de Tienda .....	19
5.1.3 Bosquejo de Carrito .....	20
5.1.4 Bosquejo de Compra .....	21
5.1.5 Bosquejo de Login.....	22
5.1.6 Bosquejo de Admin .....	23

5.1.7	Bosquejo de Reportes.....	24
Capítulo 6:	Herramientas de Desarrollo .....	25
6.1	Visual Studio.....	25
6.2	Git.....	26
6.2.1	GitHub .....	26
6.3	Base de Datos.....	27
6.3.1	MySQL Workbench .....	27
6.4	PHP .....	28
6.4.1	Composer .....	28
6.5	Laravel .....	29
6.5.1	Livewire.....	29
6.5.2	JetStream.....	30
6.6	JavaScript.....	30
6.6.1	NPM.....	31
6.7	Materialize.....	31
Capítulo 7:	Desarrollo .....	32
7.1	Preparación Inicial .....	32
7.2	Instalación de Proyecto Laravel .....	32

7.3	Implementación de Plugins y Paquetes .....	32
7.3.1	Instalación de Livewire y JetStream .....	33
7.3.2	Instalación de Materialize .....	33
7.4	Desarrollando el Sistema.....	35
7.5	Creación de modelos, Controladores, Vistas .....	36
7.5.1	Creación de Modelos .....	36
7.5.2	Creación de Controles .....	37
7.5.3	Creación de Vistas.....	37
7.6	Rutas y Controladores .....	37
7.6.1	Controlador .....	38
7.7	Definición de Base de Datos .....	39
7.7.1	Modelos .....	40
7.7.2	Migraciones.....	41
7.7.3	Semillas.....	42
7.8	Definición de Vistas .....	43
7.8.1	Diseño grafico .....	43
7.8.2	Controlador de Vistas.....	45
	Capítulo 8: Implementación y Pruebas .....	46

8.1	Usuarios.....	46
8.1.1	Creación de Usuario .....	46
8.1.2	Modificación de Usuario.....	47
8.1.3	Eliminación de Usuario.....	48
8.2	Menú de Admin .....	49
8.2.1	Productos.....	50
8.2.2	Entradas.....	51
8.2.3	Salidas .....	52
8.2.4	Compras .....	52
8.2.5	Reportes.....	53
8.3	Página de Tienda.....	54
8.4	Hacer una compra.....	55
8.5	Blogs.....	57
8.5.1	Creación de Blog .....	58
8.5.2	Modificación de Blog .....	60
8.5.3	Eliminación de Blog .....	60
Capítulo 9:	Instalación a Hosting .....	61
9.1	Preparación de instalación.....	61

9.1.1	Compresión del sistema .....	61
9.1.2	Extracción de la base de datos.....	62
9.2	Hosting .....	63
9.2.1	Transferencia del Sistema .....	64
9.2.2	Instalación de la Base de Datos .....	65
9.3	Comprobación del Sistema en la web .....	67
	Capítulo 10: Conclusiones.....	68
	Bibliografías .....	70
	Glosario de Términos .....	76
	Anexos .....	85
	Código Fuente .....	85
	Hojas de Referencias .....	86
	Laravel .....	86
	HTML.....	87
	PHP.....	88
	MySQL .....	89
	Git .....	90

# ÍNDICE DE TABLAS Y FIGURAS

---

1.1 MAPA DE PLAZUELA .....	1
1.2 IMAGEN DE LA HUERTA .....	2
4.1 DIAGRAMA CASO USO.....	15
4.2 DIAGRAMA ENTIDAD RELACIÓN.....	16
5.1 BOSQUEJO DE PÁGINA PRINCIPAL .....	18
5.2 BOSQUEJO DE PAGINA DE TIENDA.....	19
5.3 BOSQUEJO DE PAGINA DE CARRITO.....	20
5.4 BOSQUEJO DE PAGINA DE COMPRA .....	21
5.5 BOSQUEJO DE PAGINA DE LOGIN.....	22
5.6 BOSQUEJO DE PAGINA DE ADMIN .....	23
5.7 BOSQUEJO DE PAGINA DE REPORTES .....	24
6.1 LOGO DE VISUAL STUDIO.....	25
6.2 LOGO DE GIT .....	26
6.3 LOGO DE GITHUB.....	26
6.4 LOGO DE MYSQL.....	27
6.5 LOGO DE PHP .....	28
6.6 LOGO DE COMPOSER .....	28
6.7 LOGO DE LARAVEL.....	29
6.8 MASCOTA DE LIVEWIRE .....	29
6.9 LOGO DE JETSTREAM .....	30
6.10 LOGO DE JAVASCRIPT .....	30
6.11 LOGO DE NPM .....	31

6.12 LOGO DE MATERIALIZE.....	31
7.1 IMAGEN DE ARCHIVO CSS QUE BUSCAR.....	34
7.2 IMAGEN DE VISUAL STUDIO.....	35
7.3 IMAGEN DE MENÚ DE TERMINAL.....	36
7.4 EJEMPLO DE BOTÓN .....	37
7.5 IMAGEN DE RUTAS .....	38
7.6 IMAGEN DE VISTA .....	38
7.7 IMAGEN DE LA VISTA.....	39
7.8 IMAGEN DE DEFINICIÓN DE BASE DE DATOS.....	39
7.9 IMAGEN DE MODELO .....	40
7.10 IMAGEN DE MIGRACIÓN .....	41
7.11 IMAGEN DE SEMILLA .....	42
7.12 PÁGINA DE MATERIAL DESIGN DE GOOGLE .....	43
7.13 PÁGINA DE GOOGLE FONTS.....	44
7.14 PÁGINA DE PIXABAY .....	44
7.15 IMAGEN DE CONTROLADOR.....	45
8.1 IMAGEN DE BOTÓN DE REGISTRO .....	46
8.2 IMAGEN DE REGISTRO DE USUARIO.....	46
8.3 IMAGEN DE USUARIO REGISTRADO .....	47
8.4 IMAGEN DE MODIFICACIÓN.....	47
8.5 NOMBRE ORIGINAL .....	48
8.6 NOMBRE NUEVO.....	48
8.7 NOMBRE CAMBIADO .....	48
8.8 MENÚ DE ELIMINACIÓN .....	48

8.9 ÁREA DE CONFIRMACIÓN .....	48
8.10 BOTÓN DE ADMIN .....	49
8.11 MENÚ DE ADMIN .....	49
8.12 IMAGEN DE CREACIÓN DE PRODUCTO .....	50
8.13 CONFIRMACIÓN DE CREACIÓN .....	50
8.14 NUEVO PRODUCTO.....	50
8.15 IMAGEN DE EDICIÓN DE PRODUCTO .....	51
8.16 PRODUCTO BORRADO .....	51
8.17 PAGINA DE ENTRADAS.....	51
8.18 IMAGEN DE SALIDAS .....	52
8.19 IMAGEN DE COMPRAS.....	52
8.20 IMAGEN DE PÁGINA DE PDF.....	53
8.21 IMAGEN DE SELECCIÓN DE REPORTES .....	53
8.22 IMAGEN DE RESULTADOS DE REPORTES .....	54
8.23 IMAGEN DE PÁGINA DE TIENDA .....	54
8.24 IMAGEN DE COMPRA DE PRODUCTO.....	55
8.25 IMAGEN DE PÁGINA DE CARRITO .....	56
8.26 IMAGEN DE PAGO DE PRODUCTO.....	56
8.27 IMAGEN DE BLOG DEL SISTEMA .....	57
8.28 IMAGEN DE CREACIÓN DE BLOG.....	58
8.29 IMAGEN DE RESULTADOS DE CREACIÓN DE BLOG.....	59
8.30 IMAGEN DE EDICIÓN DE BLOG .....	60
8.31 IMAGEN DE ELIMINACIÓN DE BLOG.....	60
9.1 COMPRESIÓN DEL SISTEMA .....	61

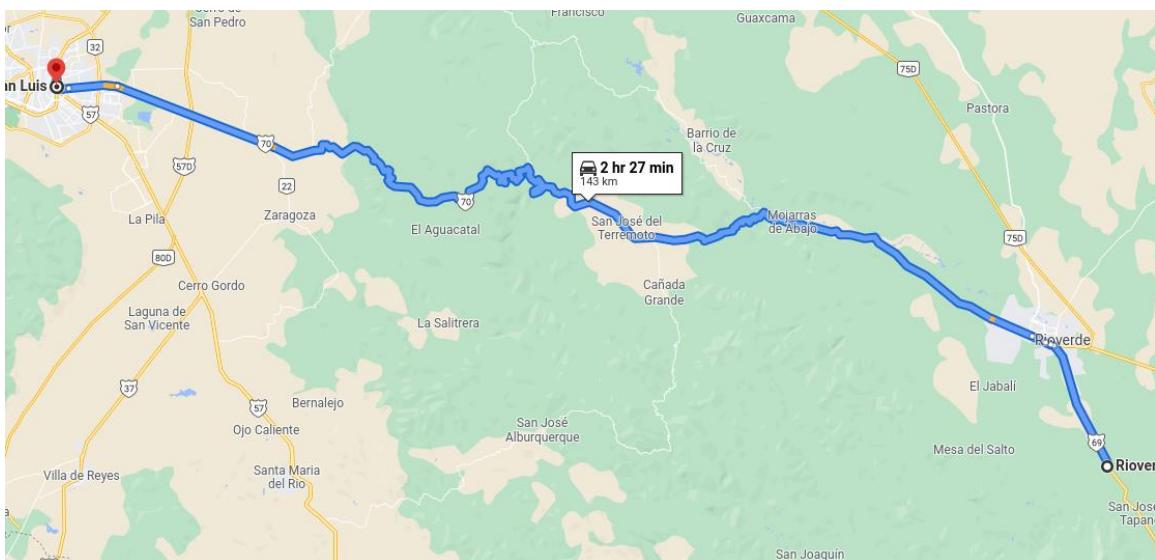
9.2 SISTEMA COMPRIMIDO.....	61
9.3 SELECCIÓN DE EXTRACCIÓN .....	62
9.4 CONFIGURACIÓN DEL ARCHIVO SQL.....	62
9.5 ARCHIVO SQL EXTRAÍDO .....	63
9.6 PAGINA DE HOSTINGER.COM .....	63
9.7 BOTONES DE ADMINISTRADOR DE ARCHIVOS .....	64
9.8 TRANSFIRIENDO EL SISTEMA.....	64
9.9 BOTONES DE BASE DE DATOS .....	65
9.10 SELECCIÓN DE BASE DE DATOS .....	65
9.11 PAGINA DE PHPMYADMIN .....	65
9.12 PAGINA DE IMPORTACIÓN .....	66
9.13 INSTALACIÓN EXITOSA.....	66



# Capítulo 1: INTRODUCCIÓN

## 1.1 ANTECEDENTES

**E**l 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 árboles frutales como: naranjas, limones, mangos, y papayas.



1.1 Mapa de Plazuela

Este negocio familiar se inició 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 Capulín.



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. Abecés 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 número 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 más alto para vender a máximo beneficio.

### 1.3 HIPÓTESIS

Si se puede hacer un sistema web, para difundir la información de la huerta, con la opción de vender productos. Se podrá mejorar los problemas de poder vender los limones de la huerta.

### OBJETIVOS

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

### 1.4 GENERALES

El objetivo es:

- Crear un sistema web para poder difundir información sobre la huerta de limones.
- Hacer ventas en línea del producto.
- Tener un control del producto al ser vendido.

#### 1.4.1 Específicos

Los objetivos específicos son:

- Al difundir información de la huerta, se dará información de otros medios en que la huerta está ligada como redes sociales, o email.
- Los clientes pueden comprar los productos y pueden poner opciones en cómo 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 meses 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 línea usando un hosting.

## Capítulo 2: MARCO TEÓRICO Y METODOLÓGICO

---

### 2.1 ANTECEDENTES DE LA INVESTIGACIÓN

**L**as 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 página 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 gráfico 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 página. 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 páginas web son programadas para Front End son HTML y CSS. HTML (Hyper Text Markup Language) es un lenguaje para crear márgenes y definir la estructura de la página usando etiquetas [6]. CSS (Cascade Style Sheets) es como cada etiqueta y elementos de HTML es definido y dibujado en una página. Como cambiar el color de texto, o definir ciertas etiquetas tamaños específicos dependiendo si la página 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 página de web Para hacer páginas más dinámicas. En si se puede usar para hacer programación y ejecución del lado del servidor [8][9]. MySQL es un 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 páginas 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 más viejo que PHP; originalmente creado como una versión web de Java, pero termino siendo algo más diferente. [8] [12].

Muchas páginas modernas son creadas con Frameworks. Un Framework es una herramienta para ayudar en la creación rápida de páginas 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 módulo de Laravel para la creación de Login, Registros, Verificación de E-mail, y Autenticado 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 paginas 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 mas 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 diseñada para tener una planeación y producción de un proyecto con la intención de cambiar durante su desarrollo. Ejemplos serian que un cliente quiera una característica que antes no fue implementada. Con otras metodologías no sería posible porque ya se adelantó 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 está 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 más 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 para 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 líneas de tiempo [28]. Microsoft Visio es un programa para la creación de diagramas, formación de maquetas, planeación de flujos de trabajos en una forma gráfica [29].

## 2.2 MARCO METODOLÓGICO

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

## 2.3 JUSTIFICACIÓN DE INVESTIGACIÓN

Una gran ventaja es de tener una presencia en línea. 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 línea rara sus futuras ventas. [31].

Otra justificación es de tener una forma de vender tus productos en línea. Con una forma de vender en línea se puede incrementar el área en que se vende. Ejemplo serían vender de estado a estado, o de un país a otro. Si no es requerido tener una página especial, pero se puede hacer otros servicios como: WhatsApp, EBay, o Mercado Libre.

## Capítulo 3: LA PROPUESTA

---

### 3.1 PRESENTACIÓN

**S**e presenta una página web que está diseñado para difundir información de la huerta. La huerta también tendrá accesos a una página especial para tener un control de los productos que entran y salen. Los clientes tendrán una página para hacer compras de los productos.

### 3.2 JUSTIFICACIÓN

Casi todo negocio moderno tiene una página web, con conexiones a otras redes sociales. No todos, pero muchos tienen tiendas en línea. 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 línea en donde se puede comprar los productos de la huerta.

### 3.3 OBJETIVOS

#### 3.3.1 Objetivos Generales

Los objetivos del 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 e 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.

## Capítulo 4: ANÁLISIS

---

### 4.1 REQUERIMIENTOS

#### 4.1.1 Funcionales

**L**os requerimientos Funcionales son descripciones implícitas del comportamiento de un programa [32]. En sí 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 página para más información.
- Se puede comprar producto dentro del sistema.
- Tener control de entrada y salida de productos.

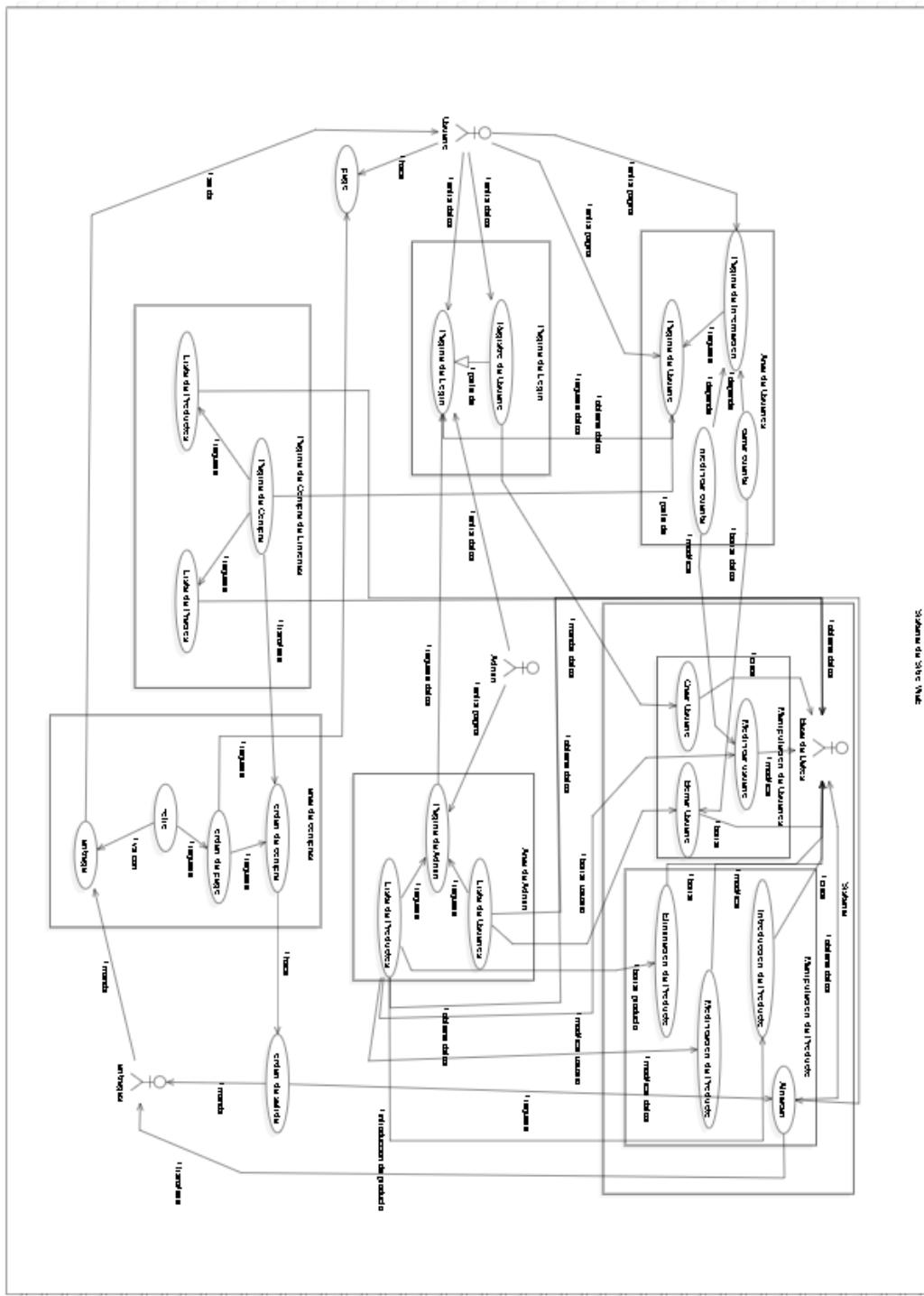
#### 4.1.2 No Funcionales

Los requerimientos 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 página de logeo se puede entrar usando credenciales de Google, o Facebook.

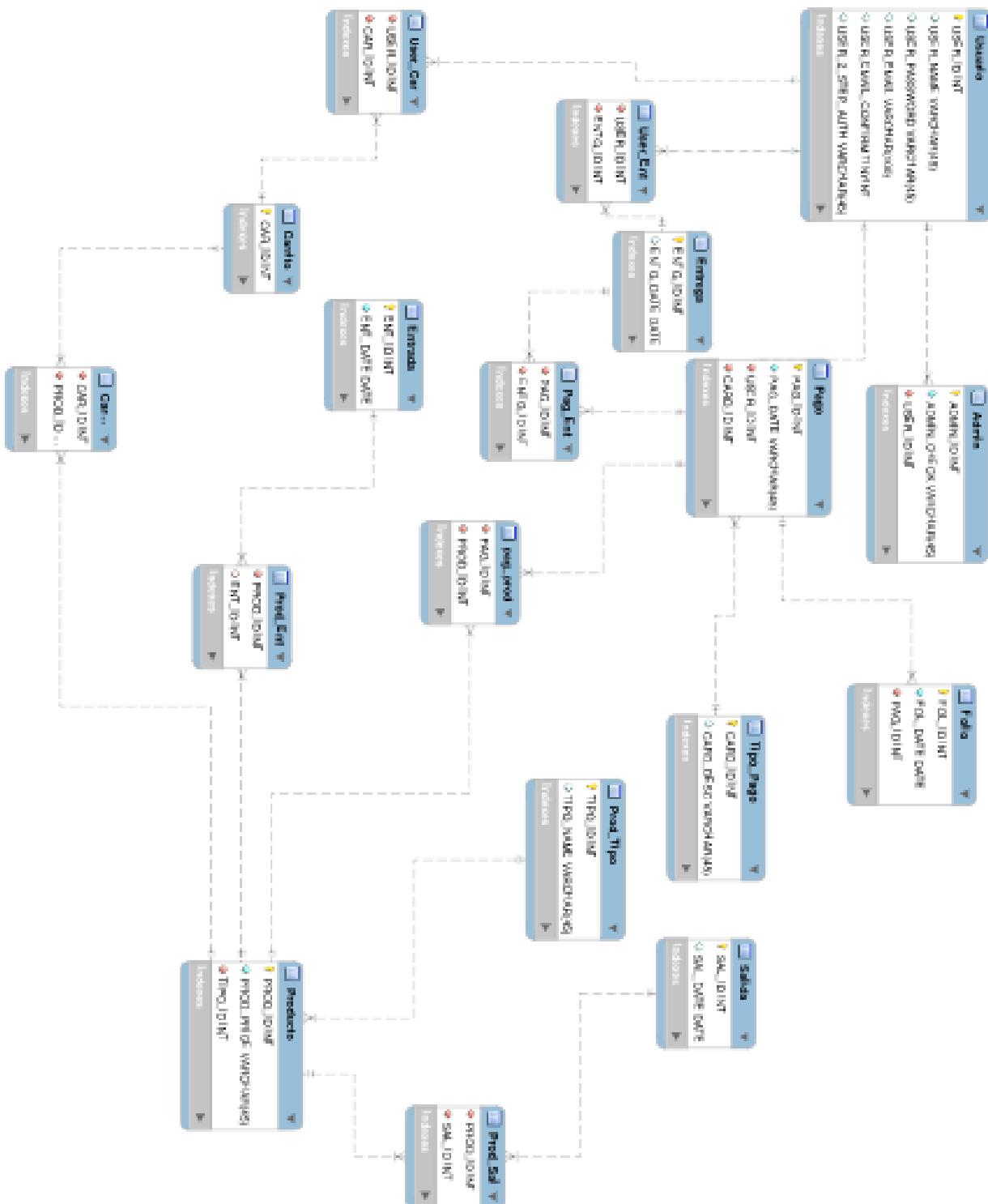
- Se hace la difusión de información con sincronización con otras redes sociales.
- La compra de producto se puede basar el precio en precios generales del mercado.
- Se puede elegir la forma de pago y el tipo de entrega en línea.

## 4.2 DIAGRAMA DE CASO USO



## 4.1 Diagrama Caso Uso

## 4.3 DIAGRAMA ENTIDAD RELACIÓN



## 4.2 Diagrama Entidad Relación

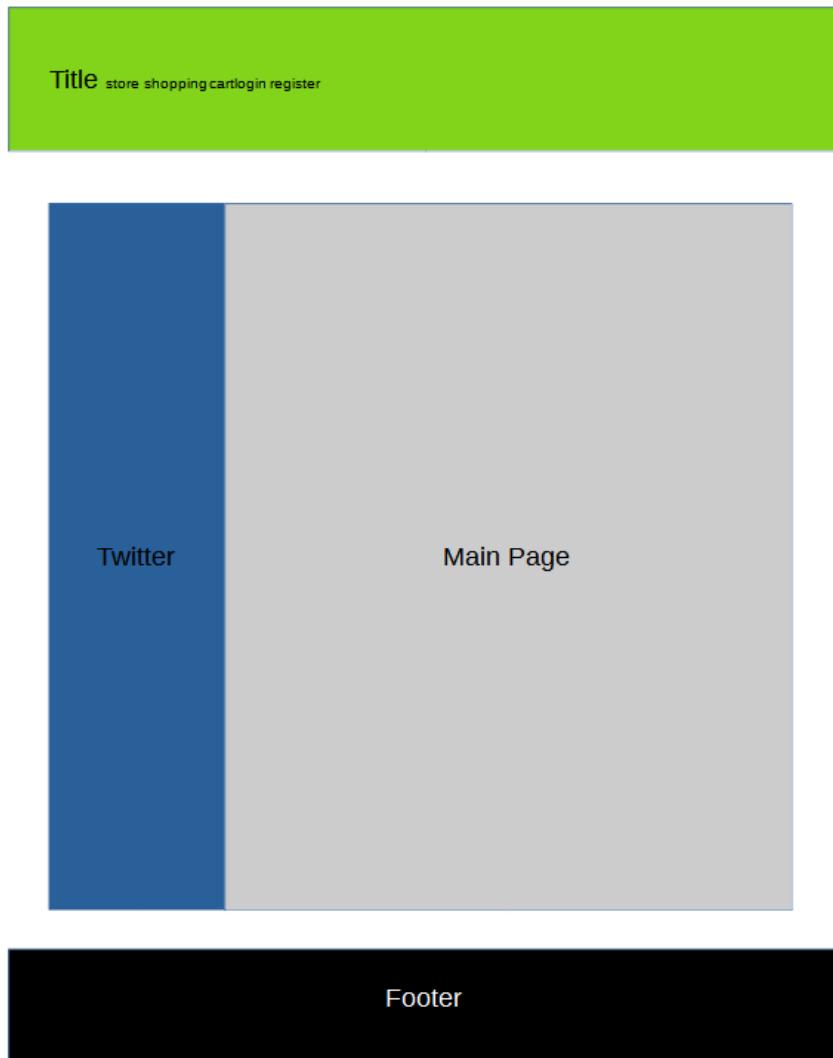
## Capítulo 5: DISEÑO

---

### 5.1 BOSQUEJOS

**U**n Bosquejo es una forma para demostrar las clases del sistema. Se demuestra una forma visual para que demostrar como el sistema se va a ver. Los bosquejos tienen un diseño simple y fácil de entender, para que el cliente tenga una visual antes de empezar con el diseño final en una página.

### 5.1.1 Bosquejo de Página Principal



5.1 Bosquejo de Página Principal

En este bosquejo se muestra la página principal. En este caso se diseñó usando HTML y CSS como bases. La página principal se verá blogs y actualizaciones de otras páginas sociales que la huerta tenga. Se mirará el carrito de compra que tiene el usuario si tiene productos para comprar después.

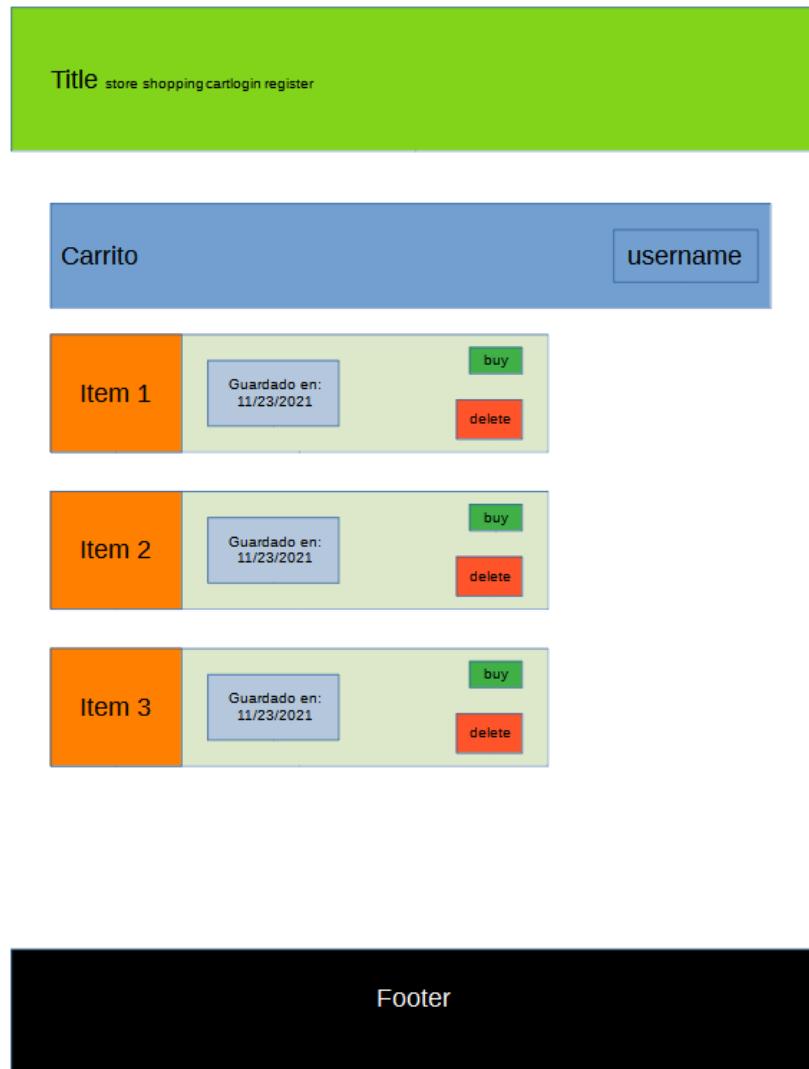
### 5.1.2 Bosquejo de Página de Tienda



#### 5.2 Bosquejo de Pagina de Tienda

En este bosquejo se muestran el interfaz en cómo se venden los productos. Se usarán 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 pasará al carrito para después pagar.

### 5.1.3 Bosquejo de Carrito

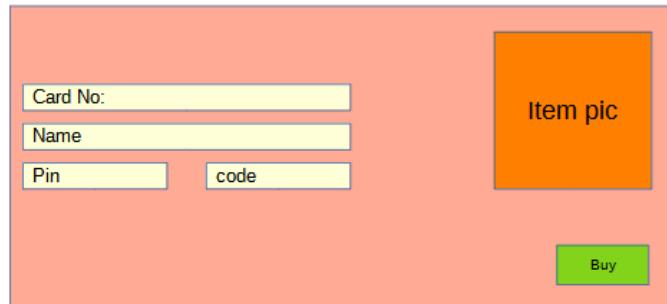


#### 5.3 Bosquejo de Página de Carrito

El Carrito es la página en donde el usuario tiene una lista de productos que van a comprar.

El cliente tendrá la opción de comprar y borrar los productos que no quieren.

#### 5.1.4 Bosquejo de Compra



##### 5.4 Bosquejo de Pagina de Compra

En la página de compra, el usuario da información de la tarjeta de crédito/débito. Con eso se finaliza la compra y el pedido es echo para el cliente.

### 5.1.5 Bosquejo de Login

A green rectangular placeholder for a login form. Inside, there is a white rectangular area labeled "Login". This area contains three input fields stacked vertically: "Username:", "Password:", and "E-Mail:". Below these fields is a green rectangular button labeled "enter".

5.5 Bosquejo de Pagina de Login

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

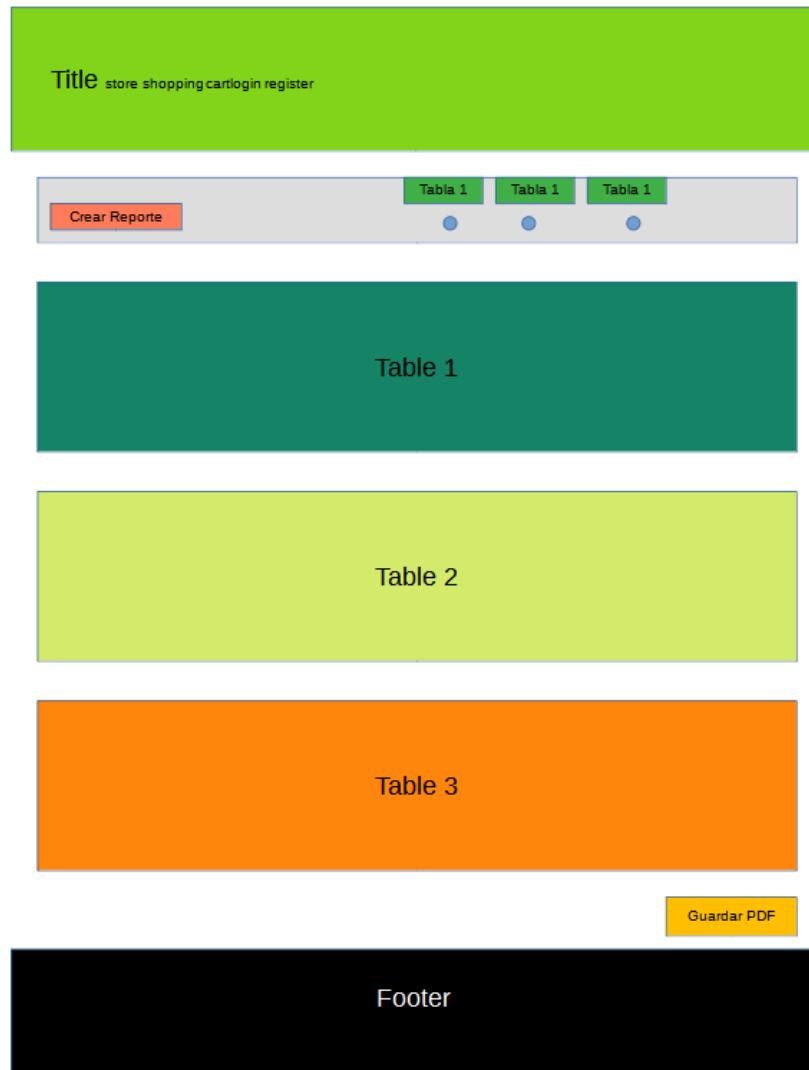
### 5.1.6 Bosquejo de Admin



#### 5.6 Bosquejo de Página de Admin

El Admin tiene acceso a esta página, para poder controlar detalles de todo el sistema. De aquí se pueden ver tablas de todo el sistema y poder agregar, editar, y borrar datos.

### 5.1.7 Bosquejo de Reportes



#### 5.7 Bosquejo de Pagina de Reportes

Los reportes son algo que el Admin puede hacer para tener un documento que se pueda guardar en PDF para archivar. El Admin puede elegir que datos se quieren guardar y como se van a imprimir en PDF.

## Capítulo 6: HERRAMIENTAS DE DESARROLLO

---

**L**as 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



6.1 Logo de Visual Studio

Microsoft Visual Studio es un editor de texto creado para programación general. Se puede instalar complementos 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. La función en este proyecto, es para tener un editor de texto con las funciones y control de versiones en un programa. Se puede buscar los archivos que se pueden modificar, correr programas en una terminal integrada y actualizar mi repositorio de Git con conexión a GitHub.

## 6.2 GIT



6.2 Logo de Git

Es un software para controlar versiones, cambios, y ramas de un proyecto de software. Se distribuye localmente, o en línea. Cada usuario tiene un clon del proyecto y al hacer cambios se puede hacer localmente, para después empujarlo al proyecto original [20]. Creado por Linus Torvalds para mantener el desarrollo del Kernel de Linux. Es un programa fácil de instalar y viene, usualmente instalado con casi todas las variantes de Linux que hay en el internet.

### 6.2.1 GitHub



6.3 Logo de GitHub

GitHub es una página que da el servicio de alojamiento de proyectos, para desarrolladores. Usan Git para controlar versiones y hacer ramas para múltiples usuarios [21]. Este es la página donde se guardará el proyecto en línea. Si por alguna razón tenga que mover de computadora, o usar otra computadora en otro lugar. Se puede descargar todo el proyecto y seguir sin tener que empezar de nuevo.

## 6.3 BASE DE DATOS

La base de datos que se está 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



6.4 Logo de MySQL

MySQL Workbench es un Software para administración, diseño, y mantenimiento de base datos visualmente [34]. Este es el servidor principal en donde se guardarán todos los datos de la página, incluyendo información de los clientes y las ventas de los productos.

## 6.4 PHP



6.5 Logo de PHP

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

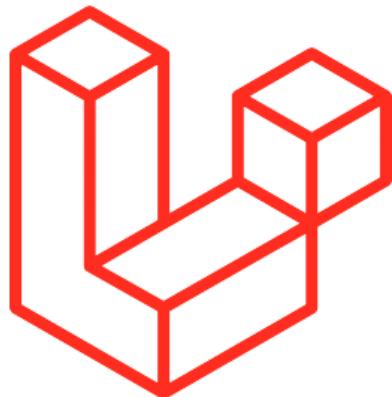
### 6.4.1 Composer



6.6 Logo de Composer

Composer es un administrador de dependencias de PHP. Se usa para instalar herramientas, o complementos para un proyecto [35]. Con Composer, se podrá correr un servidor PHP, o crear sistemas con solo comandos en un terminal.

## 6.5 LARAVEL



6.7 Logo de Laravel

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

### 6.5.1 Livewire



6.8 Mascota de Livewire

Livewire es un complemento de Laravel. Es para crear páginas web dinámicas sin tener que usar Frameworks de JavaScript completas [16].

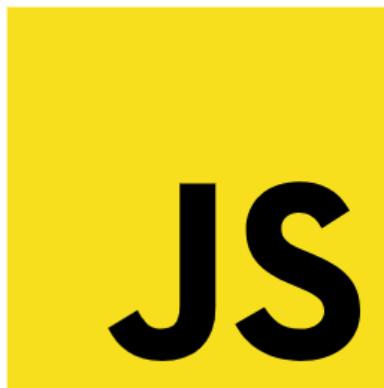
### 6.5.2 JetStream



6.9 Logo de JetStream

Es un complemento de Laravel con Livewire, para empezar páginas web más rápido. Tiene páginas hechas para Login, Registración, autenticación de 2 pasos [15].

## 6.6 JAVASCRIPT



6.10 Logo de JavaScript

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

### 6.6.1 NPM



6.11 Logo de 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



6.12 Logo de 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 página web. Con este Framework, le dará a todas las páginas una vista uniforme. Con un diseño moderno al par a otras páginas como: Google, Twitter, o Facebook. Con suficientes funciones para hacer animaciones y transiciones sin tener que escribir código.

# Capítulo 7: DESARROLLO

---

## 7.1 PREPARACIÓN INICIAL

**E**l 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 creó. 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 instalar Materialize a tu sistema es correr:

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

Después se busca el archivo: archivo\_de\_proyecto/resources/css/app.css. Dentro del archivo se introduce el texto:

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

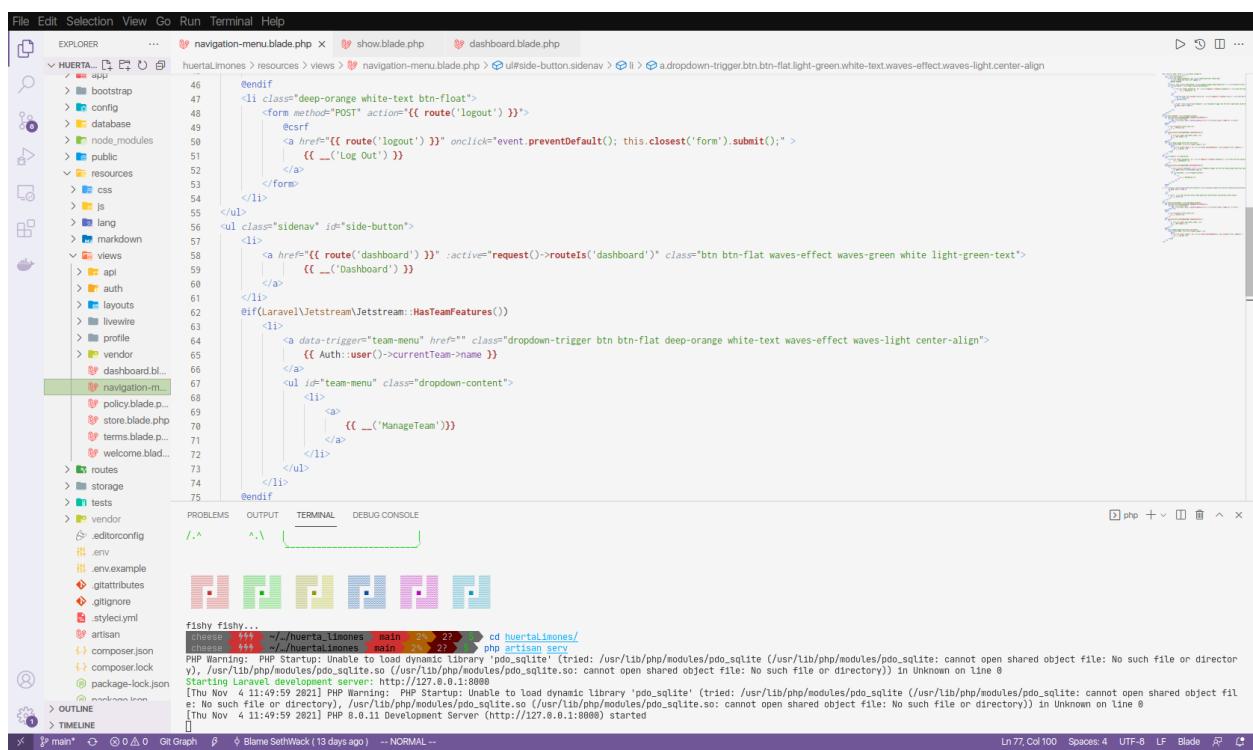
The screenshot shows a file explorer interface with the following structure:

- HUERTA\_LIMONES (selected)
- app
- bootstrap
- config
- database
- node\_modules
- public
- resources
  - css (selected)
  - app.css (highlighted)
  - js
  - lang
  - markdown
  - views
    - api
    - auth
    - layouts
    - livewire
    - profile
    - vendor
      - dashboard.bl...
      - navigation-m...
      - policy.blade.p...
      - store.blade.php
      - terms.blade.p...
      - welcome.blad...

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 usará Microsoft Visual Studio.



```

File Edit Selection View Go Run Terminal Help
File Explorer ... navigation-menu.blade.php x show.blade.php x dashboard.blade.php
HUELTALIMONES > resources > views > navigation-menu.blade.php > ul#side-button.sidenav > li > a.dropdown-trigger.btn.btn-flat.light-green.white-text.waves-effect.waves-light.center-align
    <ul class="sidenav" id="side-button">
        <a href="{{ route('dashboard') }}" :active="request()->routeIs('dashboard')" class="btn btn-flat waves-effect waves-green white light-green-text">
            {{ __('Dashboard') }}
        </a>
    </ul>
    @endif

```

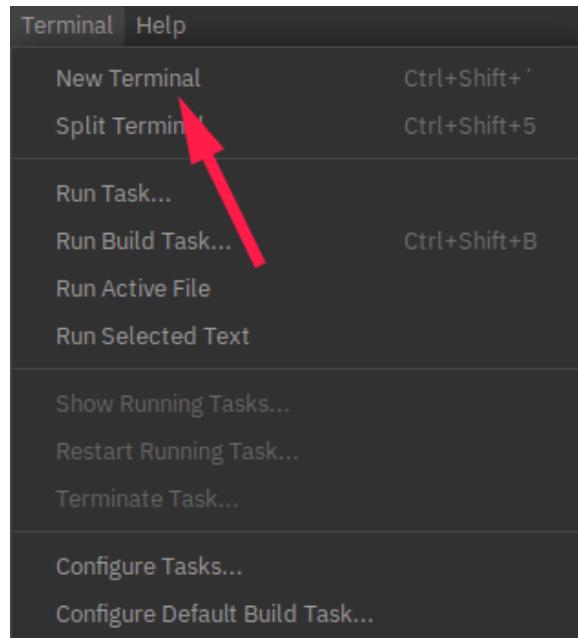
The screenshot shows the Microsoft Visual Studio interface with the following details:

- File Explorer:** Shows the project structure under "HUELTALIMONES". Files listed include bootstrap, config, database, node\_modules, public, resources (css, js, lang), markdown, views (api, auth, layouts, liveview, profile, vendor, dashboard.blade.php, navigation-menu.blade.php, policy.blade.php, store.blade.php, terms.blade.php, welcome.blade.php), routes, storage, tests, vendor, .editorconfig, .env, .env.example, gitattributes, gitignore, stylecilm, artisan, composer.json, composer.lock, package-lock.json, OUTLINE, and TIMELINE.
- Code Editor:** Displays the content of the "navigation-menu.blade.php" file. The code is a Blade template with PHP logic. It includes a conditional block `@if` for user authentication and another for team management. It uses Bootstrap classes like `btn`, `btn-flat`, `waves-effect`, and `waves-light` along with Material Design icons.
- Terminal:** At the bottom, a terminal window shows the command `php artisan serve` being run and the Laravel development server starting on port 8000.
- Status Bar:** Shows the current file is "main" (modified), Git Graph, Blame (SethWack 13 days ago), and other system information like Ln 77, Col 100, Spaces: 4, UTF-8, LF, Blade, and a refresh icon.

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

La ruta del sistema es como se define cuáles son las páginas que se van a ver al anclarlos en un link. 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>
```

7.4 Ejemplo de Botón

```

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);

```

7.5 Imagen de Rutas

### 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 página.

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

```
@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">
                    
                </div>
            </div>
        </div>
    </div>
```

*7.7 Imagen de la vista*

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

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

*7.8 Imagen de definición de Base de Datos*

### 7.7.1 Modelos

El modelo es una forma para conectar con la Base de Datos. Está 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'
            ]
        ];
    }
}
```

7.9 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');
        });
    }
}
```

7.10 Imagen de Migración

### 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.

```
DB::table('productos')->insert([
    'PROD_PRICE' => 40.00,
    'IMG_PATH' => 'lime6.jpg',
    'TIPO_ID' => 3,
    'PROD_AMMOUNT' => 20
]);
DB::table('users')->insert([
    'name' => 'ADMIN1',
    'email' => 'admin@gmail.com',
    'password' => Hash::make('qwerty_12345678'),
    'admin' => True
]);
DB::table('users')->insert([
    'name' => 'userTest',
    'email' => 'test@gmail.com',
    'password' => Hash::make('12345678'),
    'admin' => False
]);
DB::table('blogs')->insert([
    'BLOG_SLUG' => 'Test1',
    'BLOG_TITLE' => 'Test1 Title',
    'BLOG_DESC' => 'This is a description',
    'BLOG_TEXT' => 'This is a longer text: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et id est labore et dolore magna aliquante',
    'BLOG_IMG' => '1.jpg',
    'USER_ID' => '1'
]);
DB::table('carritos')->insert([
    'id' => 1,
]);
DB::table('user_cars')->insert([
    'USER_ID' => 1,
    'CAR_ID' => 1,
]);
```

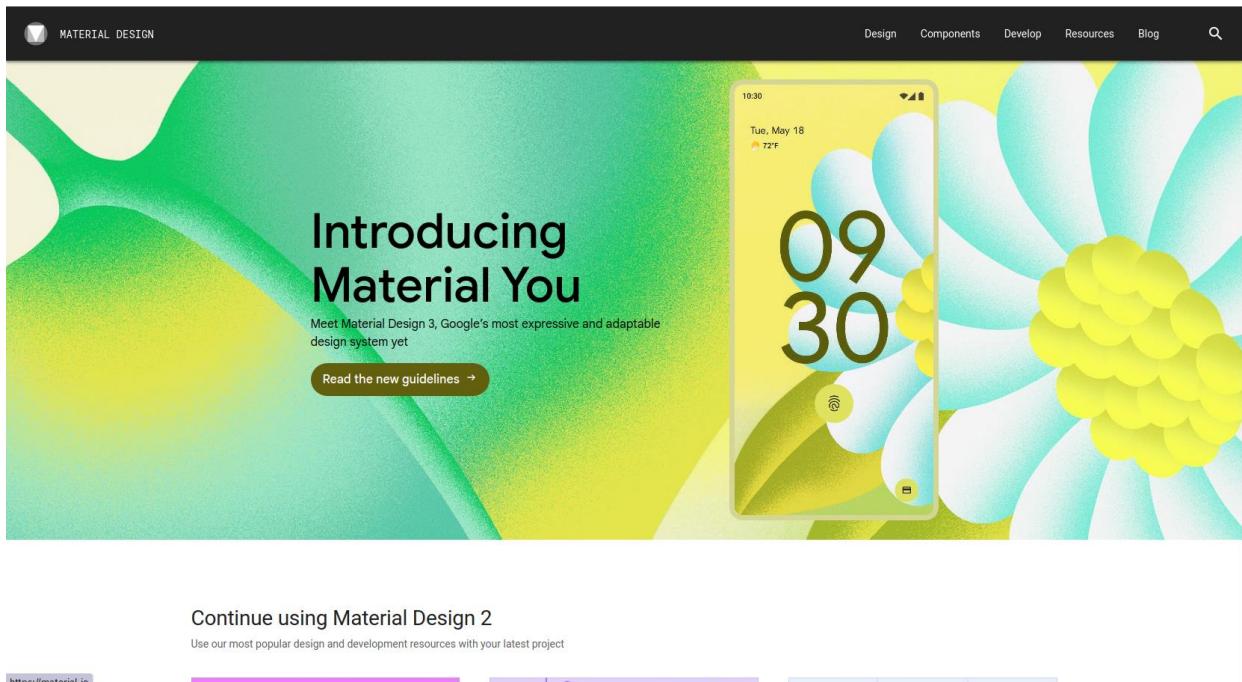
7.11 Imagen de Semilla

## 7.8 DEFINICIÓN DE VISTAS

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

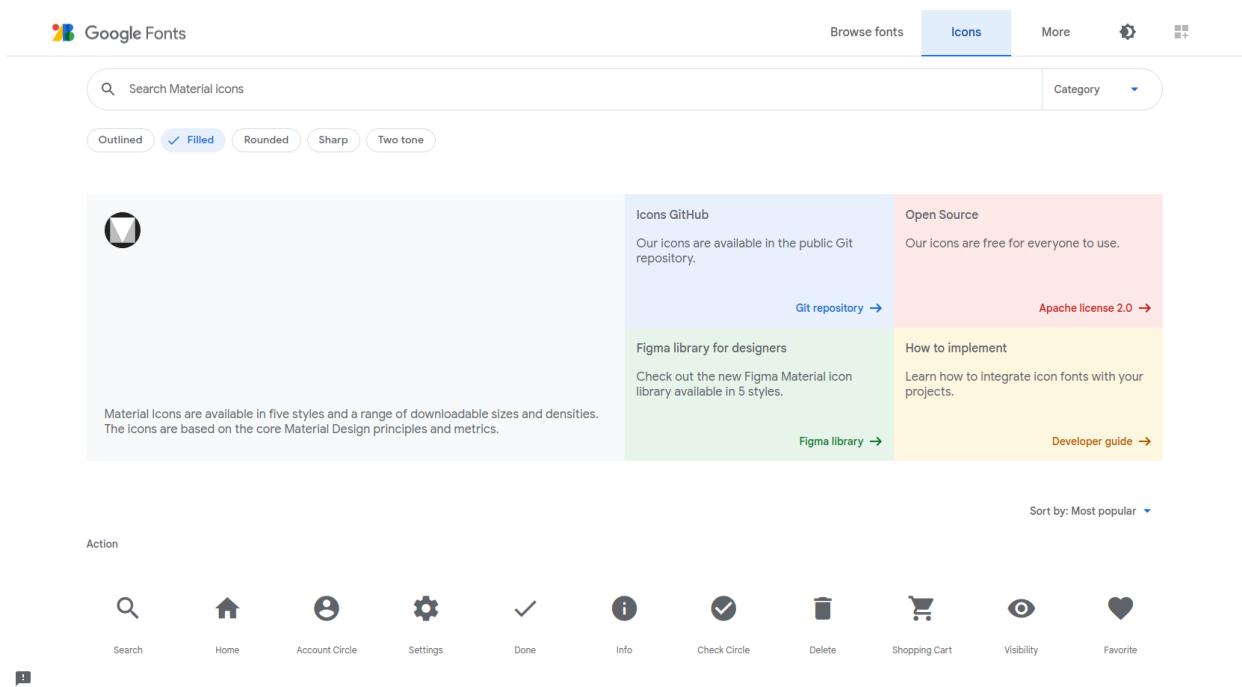
### 7.8.1 Diseño gráfico

La vista general es definida 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.



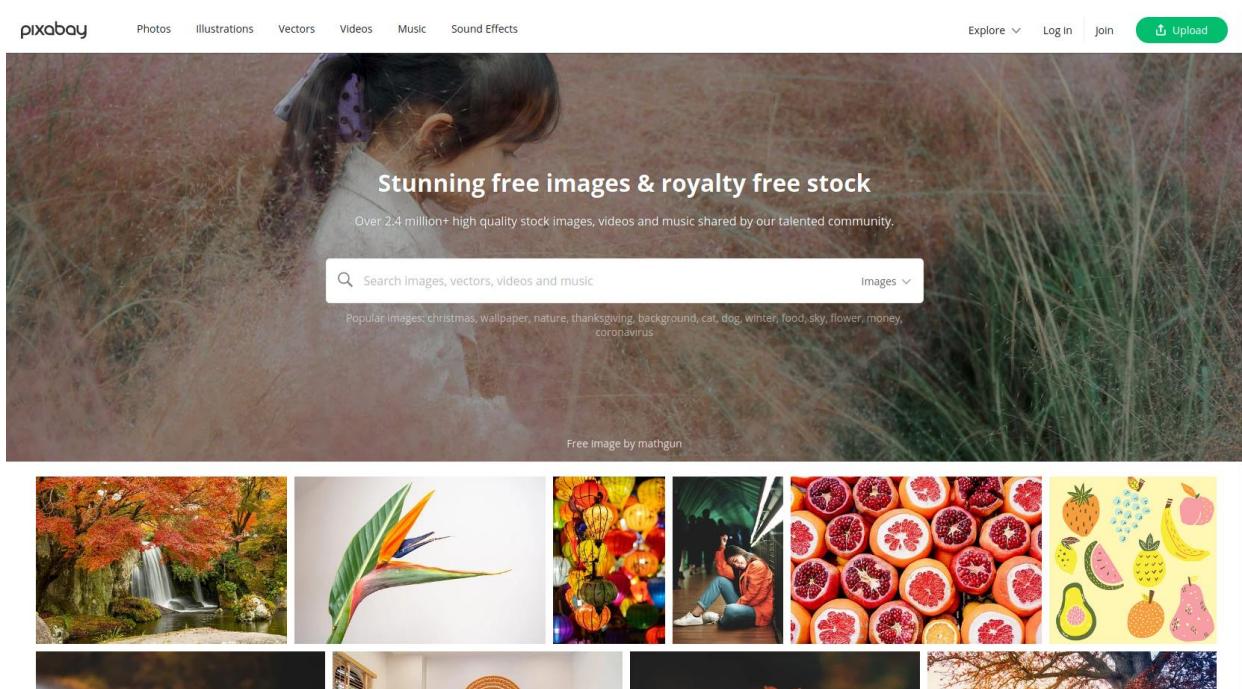
7.12 Página de Material Design de Google

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



7.13 Página de Google Fonts

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



7.14 Página de Pixabay

### 7.8.2 Controlador de Vistas

Para mandar llamar una vista primero se crea usando la terminal. Después en el controlador se define cuáles son las páginas 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 página.

```
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;
}
```

7.15 Imagen de controlador

## Capítulo 8: IMPLEMENTACIÓN Y PRUEBAS

---

**E**l Framework de Laravel tiene una función incluida para hacer pruebas sin tener que implementar el sistema en línea. 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 páginas y el poder de crear Blogs en la página.

#### 8.1.1 Creación de Usuario

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



8.1 Imagen de botón de registro

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

Registration	
Name	cliente1
Email	cliente@gmail.com
Password	*****
Confirm Password	*****

8.2 Imagen de registro de usuario

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

Huerta Limones

HOME    STORE    CARRITO    CLIENTE

### 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 página de usuario.

#### Informacion de Perfil

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

	cliente
	cliente@gmail.com
<input type="button" value="GUARDAR"/>	

#### Actualizar Contraseña

Asegurate que tu contraseña sea larga y aleatoria para ser más seguro/a.

	Contraseña reciente
	Nueva contraseña
	Confirmar contraseña
<input type="button" value="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.

#### Autenticación de dos pasos

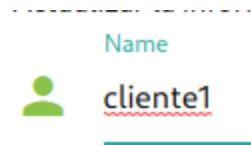
Anadir seguridad adicional a tu cuenta usando autenticación de dos pasos

Tu no tienes autenticación de dos pasos habilitada.

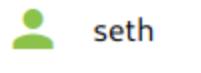
Cuando tienes autenticación de dos pasos, se dará una llave segura y aleatoria durante la autenticación que será dada en tu aplicación de Google en tu teléfono.

### 8.4 Imagen de Modificación

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



8.5 Nombre original



8.6 Nombre nuevo

HOME STORE

CARRITO

SETH

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

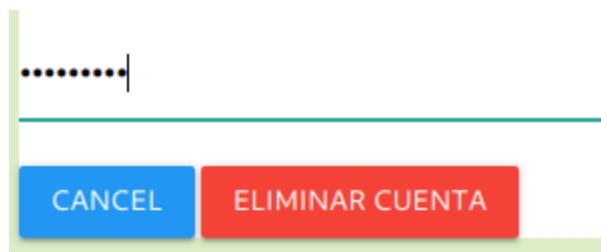
Elimina permanentemente tu cuenta.

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 página web.

The screenshot shows a dark blue sidebar on the left labeled "Admin side bar". At the top of the sidebar is the store name "Huerta Limones". To the right of the sidebar is a light green main content area. The sidebar contains the following menu items:

- USUARIOS
- PRODUCTOS
- ENTRADAS
- SALIDAS
- COMPRAS
- REPORTES

At the very top of the page, there is a navigation bar with links: HOME, STORE, CARRITO, ADMIN, and ADMIN1. The ADMIN1 link is highlighted with a green background and a pencil icon.

8.11 Menú de Admin

### 8.2.1 Productos

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

### Crear Producto

Arbolito

Cantidad  
\$ 50

Precio  
\$ 150

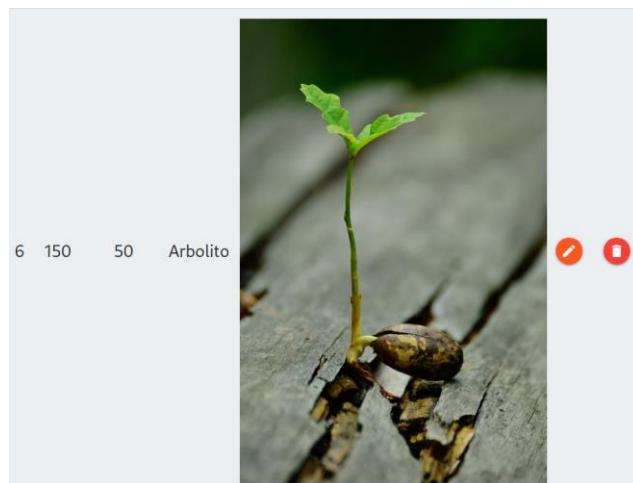
IMAGE sprout.jpg

CREAR BACK

8.12 Imagen de creación de producto



8.13 Confirmación de creación



8.14 Nuevo producto

La página 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 creadas a la introducción de un producto y borrados al ser borrados.

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

8.17 Pagina de Entradas

En si se pueden crear más 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.

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

8.18 Imagen de Salidas

Se pueden crear salidas si se venden productos afuera del sistema. Ejemplo será que se vendan limones en la huerta si se tiene que dar la actualización del producto.

### 8.2.4 Compras

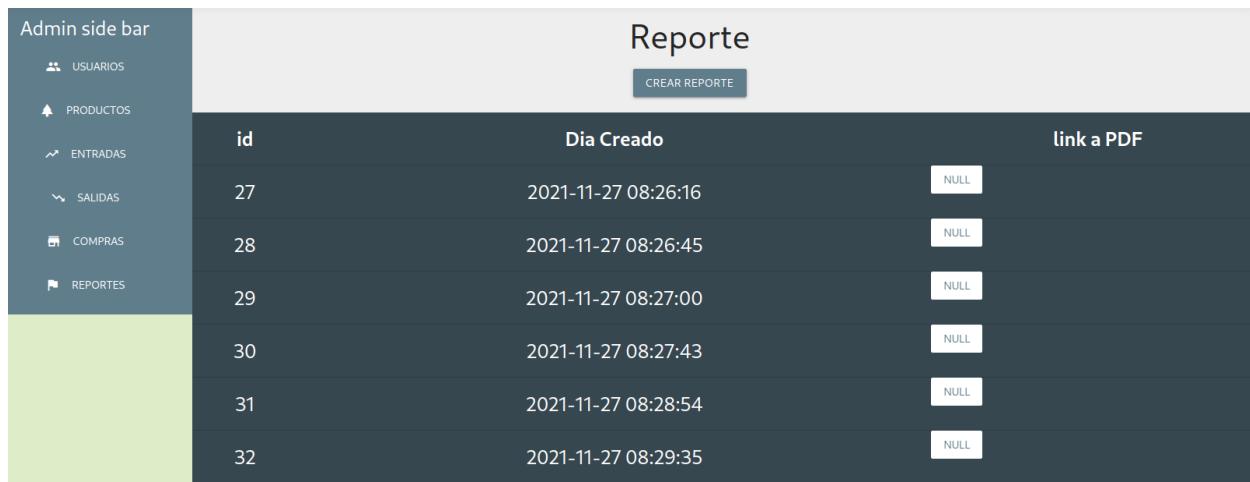
Las compras son una lista, de los clientes y sus productos vendidos en la página. Es preferido no borrar o modificar estas tablas. En este caso se van a crear los datos para las pruebas.

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 son para que el Admin pueda tener una visión general del sistema y tener imprenta si es requerido los reportes en PDF.



The screenshot shows a sidebar menu titled 'Admin side bar' with the following items: USUARIOS, PRODUCTOS, ENTRADAS, SALIDAS, COMPRAS, and REPORTES. The REPORTES item is highlighted with a green background. The main content area is titled 'Reporte' and contains a table with the following data:

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 página de PDF



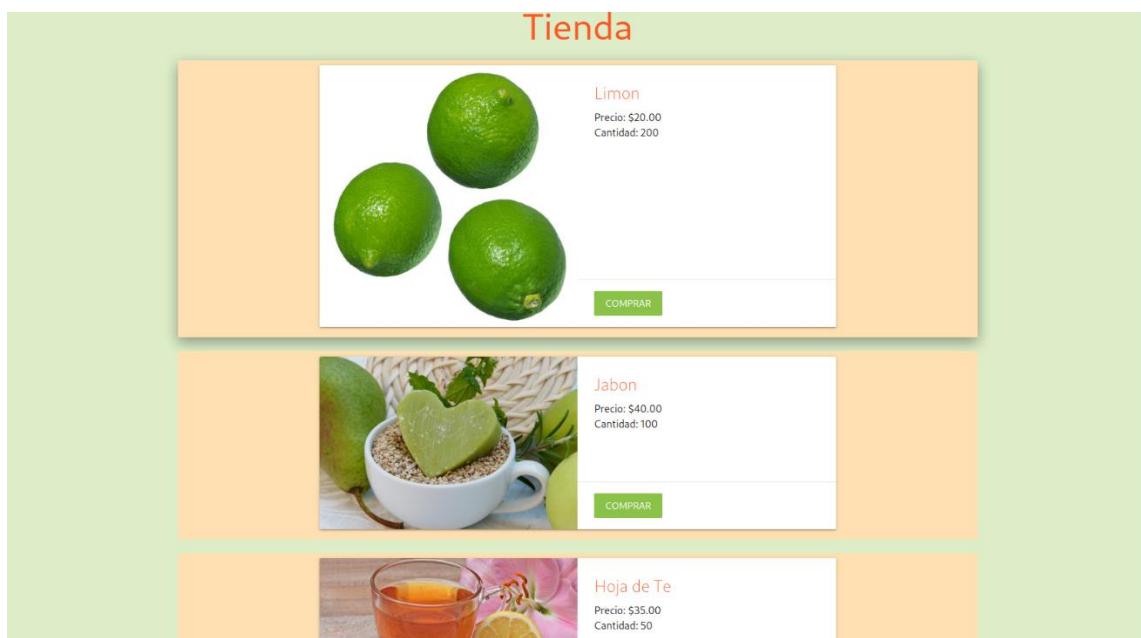
8.21 Imagen de Selección de Reportes

Admin side bar	<b>id</b>	<b>name</b>	<b>email</b>	<b>admin</b>
	1	ADMIN1	admin@gmail.com	1
	2	userTest	test@gmail.com	0
	6	virtual	virtual@gmail.com	0
	<b>Id</b>	<b>Usuario</b>	<b>Tipo de Pago</b>	<b>Producto</b>
	1	ADMIN1	MasterCard	Semilla
	<b>Id</b>	<b>Salida</b>		<b>Producto ID</b>
	2	2021-11-27		1
	<a href="#">MAKE PDF</a>			

8.22 Imagen de resultados de reportes

### 8.3 PÁGINA DE TIENDA

La página de Tienda es la página 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 página de Tienda

## 8.4 HACER UNA COMPRAS

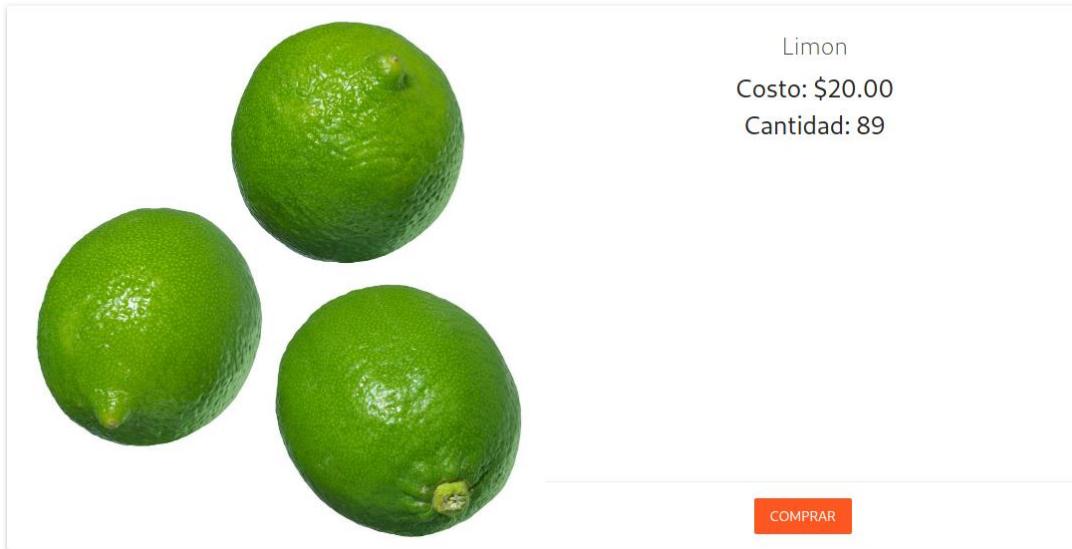
El usuario compra un producto al presionar el botón de comprar, Aquí se decide la cantidad que se quiere y luego es añadido al carrito.



8.24 Imagen de compra de Producto

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

## Carrito de ADMIN1



8.25 Imagen de página de Carrito

### Comprar Limon para: ADMIN1

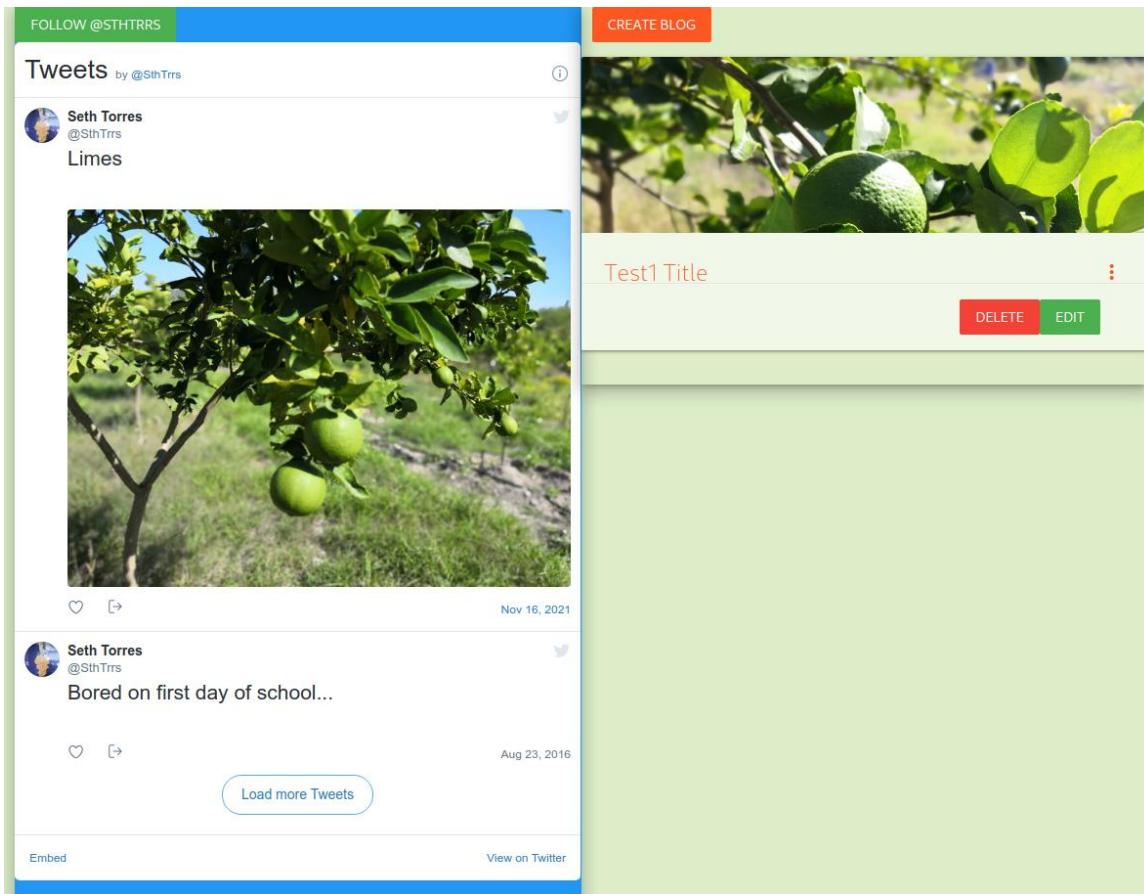
A screenshot of a product payment page. At the top, it says "Comprar Limon para: ADMIN1" and "Precio: 20". Below this is a form for entering payment information. The form fields include: "Tipo de Tarjeta" (dropdown menu), "Número de Tarjeta" (input field containing "12345623"), "Fecha de expiración" (calendar icon and input field), and "Nombre en la Tarjeta" (input field containing "cheese"). There is also a small logo of a person with a green checkmark next to the name field. At the bottom of the form is a red rectangular button with the word "COMPRAR!" in white capital letters.

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 será: 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 está 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 página. El botón de creación de blog, de modificación, y eliminación solo se pueden ver si eres un Admin.

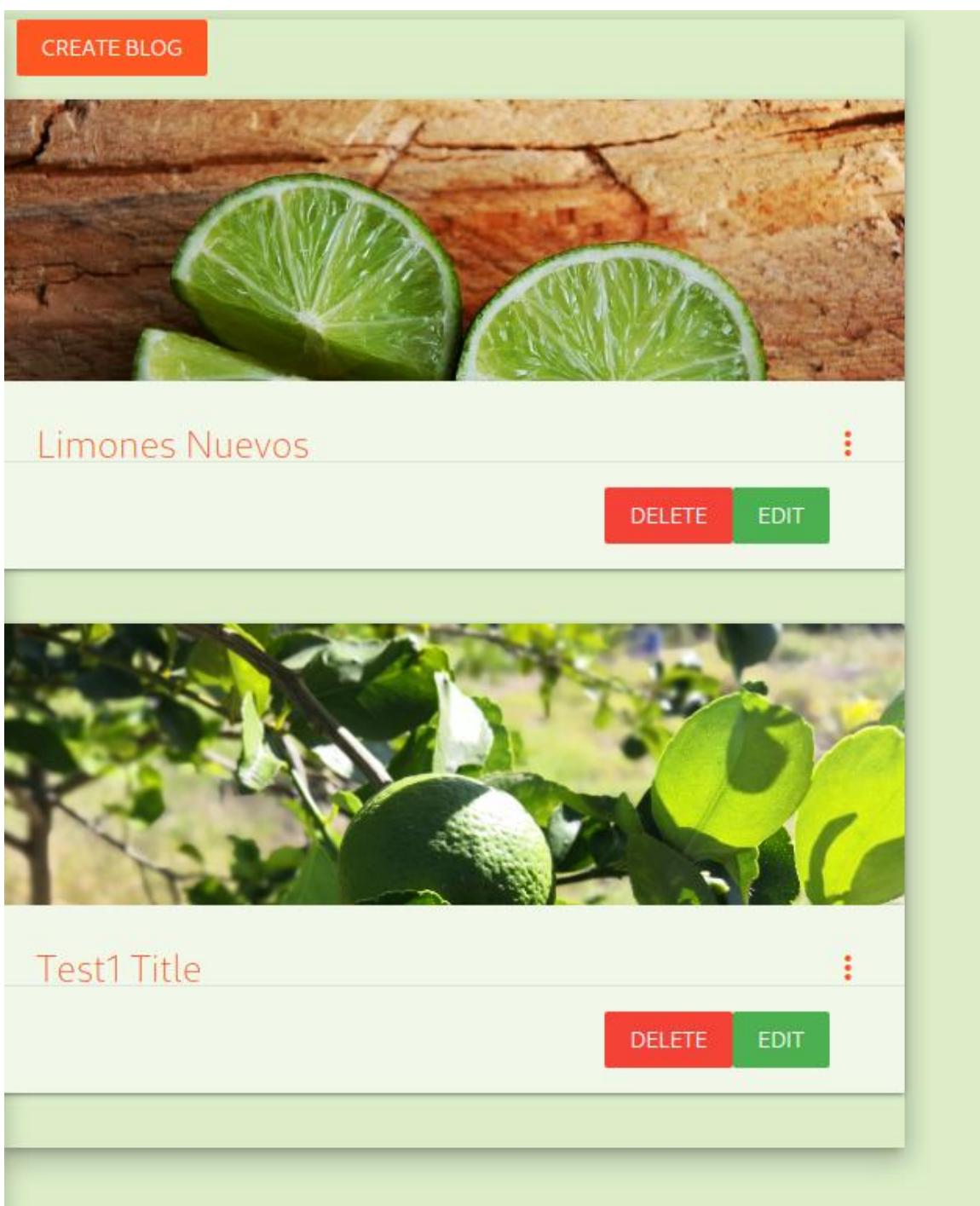
The screenshot shows a 'Crear Blog' (Create Blog) form. At the top, the title 'Crear Blog' is displayed in green. Below it, there are several input fields:

- 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 showing the file 'lime-2133091\_960\_720.jpg'.

At the bottom right of the form is a green 'SUBMIT' button with a white arrow icon.

8.28 Imagen de creación de Blog

Al crear un blog, con una imagen se la página se actualizará con el nuevo blog.



8.29 Imagen de resultados de creación de Blog

### 8.5.2 Modificación de Blog

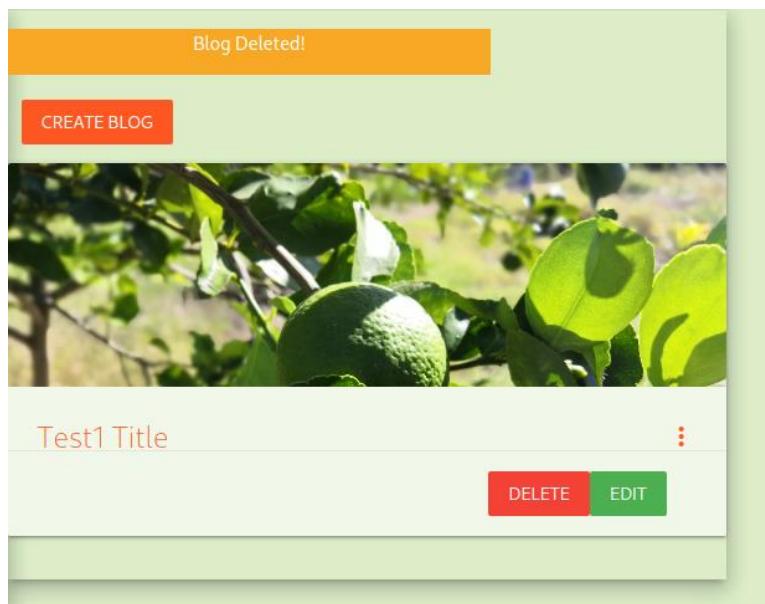
Al modificar el blog se va a una página de modificación, y con los datos del blog.



8.30 Imagen de edición de Blog

### 8.5.3 Eliminación de Blog

Se pueden borrar los blogs creados se requiere. Se da un mensaje que se borró el blog y la página se actualiza para mostrar los cambios.



8.31 Imagen de eliminación de Blog

## Capítulo 9: INSTALACIÓN A HOSTING

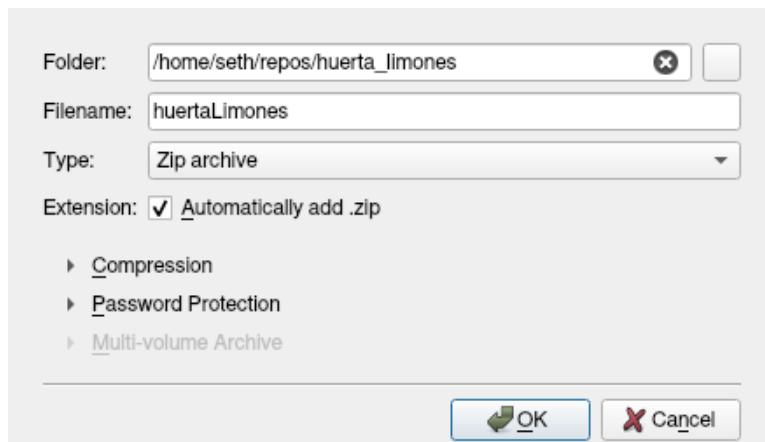
---

**E**l Hosting es un servicio de alojamiento de páginas web. Con eso se puede subir el proyecto al internet. Así todo cliente puede usar el sistema en línea con solo introducir el link de la página.

### 9.1 PREPARACIÓN DE INSTALACIÓN

#### 9.1.1 Compresión del sistema

Lo primero que se tiene que hacer es comprimir el sistema en un archivo, para fácil transporte del sistema.



9.1 Compresión del Sistema

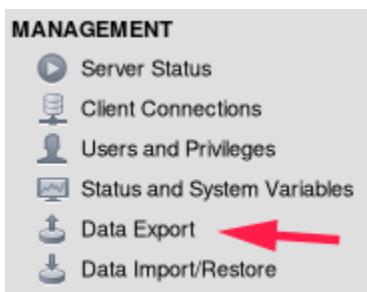


9.2 Sistema  
Comprimido

### 9.1.2 Extracción de la base de datos

La base de datos, se tiene que extraer del sistema MySQL. Se tiene que hacer esto para poder transferir al hosting, e instalar al sistema de MySQL en la web.

Lo primero que se tiene que hacer es hacer un “dump” o la extracción de la base de datos en un archivo.



9.3 Selección de extracción

The screenshot shows the MySQL Data Export configuration window. In the 'Tables to Export' section, the 'Schema' dropdown is set to 'Exp' and the 'Schema Objects' dropdown is set to 'Schema Objects'. Under 'Tables to Export', the 'huerta' schema is selected, while 'test' is unselected. In the 'Export Options' section, the 'Export to Self-Contained File' radio button is selected, and its value is '/home/seth/dumps/Dump20220203.sql'. Below it, the 'Create Dump in a Single Transaction (self-contained file only)' checkbox is checked. Red numbers 1, 2, and 3 are overlaid on the screen to indicate specific configuration points.

9.4 Configuración del archivo SQL

En la imagen anterior, se define las configuraciones que se tiene que hacer para extraer el archivo de MySQL. Lo primero se selecciona cual base de datos se va a extraer. Luego se

selecciona el lugar donde se va extraer. Y al final se confirma que tenga los comandos para crear la nueva base de datos en el sistema en la web. Despues se entra a la pagina de hosting para poder subir el sistema.

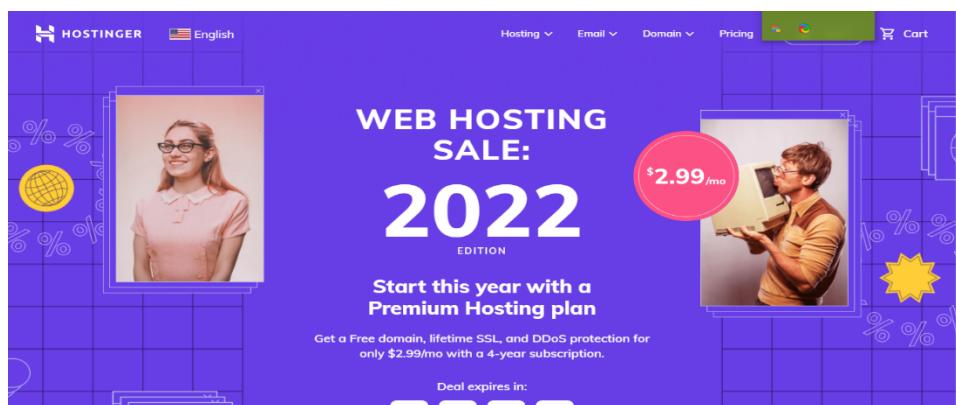


9.5 Archivo SQL extraído

## 9.2 HOSTING

Al tener el sistema listo, se va a mover todo al Hosting. Primero se tiene que tener un servicio de Hosting listo para tener tu página de internet. En este caso se va usar el servicio de Hostinger.

Se va a prestar un espacio a una cuenta para poder hacer pruebas por este momento, y luego se transfiere a uno independiente después. El administrador del sistema va hacer el nombre y configurar el servicio antes de transferir el sistema.



9.6 Pagina de hostinger.com

### 9.2.1 Transferencia del Sistema

Primero se accede al administrador de archivos del servidor. De ahí se transfiere el sistema a un folder.



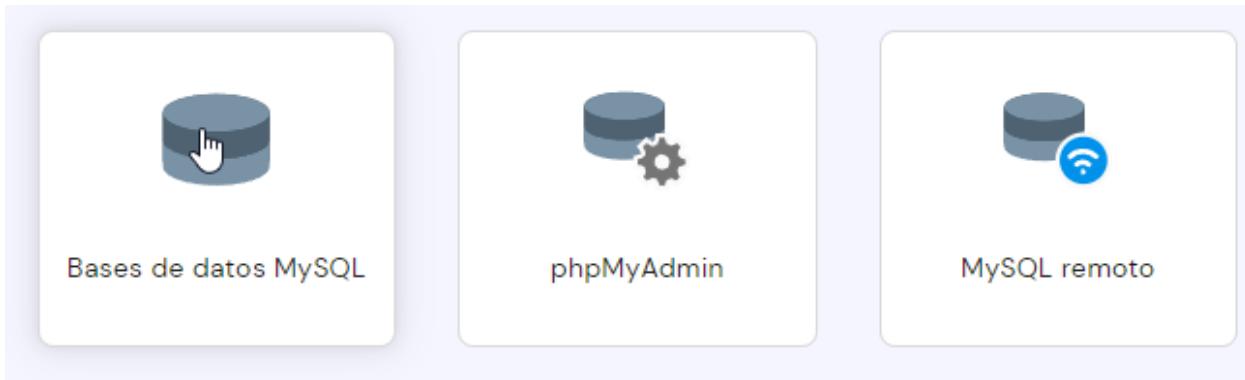
9.7 Botones de Administrador de Archivos

	Name	Size	Date	Permissions
	app		2022-02-17 01:50:00	drwxr-xr-x
	bootstrap		2022-02-17 01:50:00	drwxr-xr-x
	config		2022-02-17 01:50:00	drwxr-xr-x
	database		2022-02-17 01:50:00	drwxr-xr-x
	public		2022-02-17 01:50:00	drwxr-xr-x
	resources		2022-02-17 01:50:00	drwxr-xr-x
	routes		2022-02-17 01:50:00	drwxr-xr-x
	storage		2022-02-17 01:50:00	drwxr-xr-x
	tests		2022-02-17 01:50:00	drwxr-xr-x
	vendor		2022-02-17 01:51:00	drwxr-xr-x
	.editorconfig	0.3 kB	2022-02-17 01:50:00	-rw-r--r--
	.env	1.0 kB	2022-02-17 01:50:00	-rw-r--r--
	.env.example	0.9 kB	2022-02-17 01:50:00	-rw-r--r--

9.8 Transfiriendo el Sistema

### 9.2.2 Instalación de la Base de Datos

La Base de Datos, es trasferida usando el servicio de phpMyAdmin. Es un controlador de MySQL, similar a MySQL Workbench.



9.9 Botones de Base de Datos



9.10 Selección de Base de Datos

A screenshot of the phpMyAdmin interface. The left sidebar shows a tree view of databases, with 'u708081785\_huerta\_limones' selected. The main area displays the structure of this database, including tables like 'blogs', 'carritos', 'car\_prods', etc. A large table at the bottom lists these tables with their respective details. A purple button labeled 'Enter phpMyAdmin' is visible in the top right corner.

Table	Action	Rows	Type	Collation	Size	Overhead
blogs	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
carritos	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
car_prods	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
entradas	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
entregas	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
failed_jobs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
folios	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
migrations	Browse Structure Search Insert Empty Drop	24	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
pagos	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
pag_ents	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
pag_prods	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
password_resets	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
personal_access_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
productos	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
prod_ents	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
prod_sals	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
prod_tipos	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
reportes	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
salidas	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
sessions	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-

9.11 Página de phpMyAdmin

Al acceder la página se tiene que elegir la Base de Datos que se va usar. El administrador ya tendrá la Base creada anteriormente en este caso. Si no se tiene que hacer uno, con un usuario específico.

## Importing into the database "u708081785\_huerta\_limones"

### **File to import:**

File may be compressed (gzip, bzip2, zip) or uncompressed.

A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer:  huerta.sql (Max: 256MiB)

You may also drag and drop a file on any page.

Character set of the file:

### **Partial import:**



Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. (*This might be a good way to import large files.*)

Skip this number of queries (for SQL) starting from the first one:

### **Other options:**



Enable foreign key checks

### **Format:**

#### 9.12 Página de importación

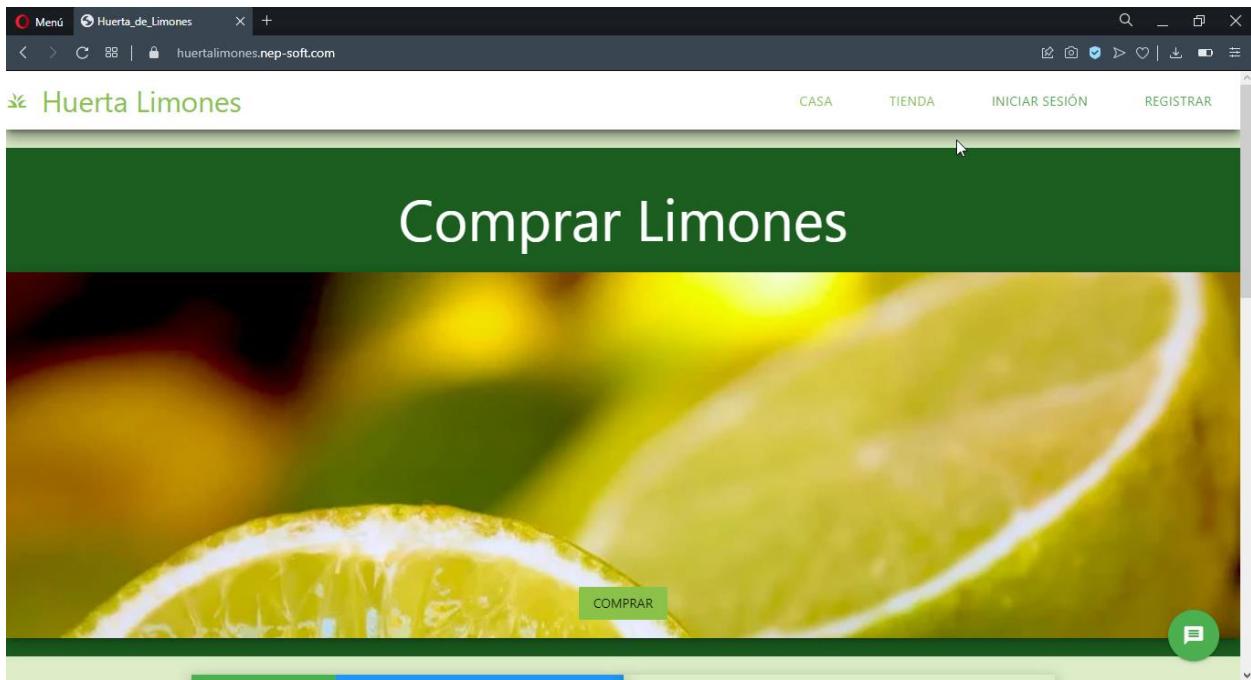
Se mueve a la página de importar. De ahí se elige el archivo que se creó para instalar la Base de Datos.

 Import has been successfully finished, 75 queries executed. (huerta.sql)

#### 9.13 Instalación exitosa

### 9.3 COMPROBACIÓN DEL SISTEMA EN LA WEB

Al acabar de instalar todo el Sistema en el servidor. El administrador tiene que crear el dominio, y configurar la conexión de los archivos para poder ver la página final en línea.



## Capítulo 10: CONCLUSIONES

---

Este proyecto tuvo que solucionar un problema que el cliente quiere resolver. Pero hay cosas que entre lo que quiere un cliente, y las intenciones de un programador pueden ser diferentes. Durante algunas discusiones entre cliente e ingeniero se tuvo que añadir, quitar, y expandir ciertos módulos del proyecto. Ejemplo será: El cliente quiso que se pueda vender otros productos que tengan que ser relacionados con los limones. Se tuvo que modificar ciertas partes de la Base de Datos y el módulo de productos para tener lo que el cliente quiere.

La intención es tener un sistema para mostrar a los clientes. Los jefes de la huerta les gusta el diseño de la página. Aunque van a tener que darse un tiempo de capacitación del sistema. El cliente espera que el sistema funcione, y que se tenga un buen uso en línea.

En sí, el desarrollo del sistema fue algo más fácil, que hacer todo de mano desde el principio. Una pregunta que se tenía es como tener una seguridad que la información que se está guardada sea segura. Pero con Laravel, y sus Plugins se tiene la capacidad de tener esa seguridad. Con Laravel se pudo acortar el tiempo del desarrollo del sistema, y así tener más funciones en corto tiempo.

Problemas principales fueron: encontrar una manera de demostrar el sistema, como si estuviera instalada a un servidor web. La solución fue que Laravel tiene una función que simula un servidor.

Para concluir, Este proyecto me enseño como tener una conversación entre cliente y desarrollador. También me dio la introducción al sistema Laravel, que sin ella me tardaría más tiempo en terminar el proyecto. Este Framework si es uno que usare en proyectos futuros, si tengo que hacer algo con desarrollo web. Esto si fue una experiencia, que me hizo crecer como desarrollador en general. Desde la planeación, e investigación. Hasta aprender métodos para desarrollar en sistemas web. Esto fue un tiempo que me canso mentalmente, pero me dio una prueba de cómo será el futuro de mi carrera como ingeniero en sistemas.

## BIBLIOGRAFÍAS

---

[1] DEFINICIÓN DE HUERTA [INTERNET]. DEFINICIÓN.DE. [CITED 2021OCT10].

AVAILABLE FROM: [HTTPS://DEFINICION.DE/HUERTA/](https://DEFINICION.DE/HUERTA/)

[2] WEB PAGE [INTERNET]. WEB PAGE DEFINITION. [CITED 2021OCT10]. AVAILABLE

FROM: [HTTPS://TECHTERMS.COM/DEFINITION/WEBPAGE](https://TECHTERMS.COM/DEFINITION/WEBPAGE)

[3] SERVER [INTERNET]. PAESSLER. [CITED 2021OCT10]. AVAILABLE FROM:

[HTTPS://WWW.PAESSLER.COM/IT-EXPLAINED/SERVER](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/](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/](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](https://WWW.W3SCHOOLS.COM/HTML/HTML_INTRO.ASP)

[7] CSS INTRODUCTION. [CITED 2021OCT10]. AVAILABLE FROM:

[HTTPS://WWW.W3SCHOOLS.COM/CSS/CSS\\_INTRO.ASP](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](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/](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/](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](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](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/](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](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](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](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/](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](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](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/](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](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](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/](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/>

## Glosario de Términos

---

**Admin:** Persona que tiene el control total de una página, fórum, o software.

**Amazon:** Amazon.com, Inc. Es una empresa líder mundial de Internet y uno de los destinos más traficadas de Internet con más tráfico en todo el mundo.

**Back End:** La porción invisible de un software, o página web, que está al alcance más lejos del usuario.

**Base De Datos:** Es una colección de datos organizada, conectada lógicamente con otros datos usando tablas.

**Blog:** Un diario en línea en donde las personas pueden publicar entradas en el diario sobre sus experiencias personales y/o pasatiempos.

**Bosquejos:** Conjunto de instrucciones que se pasan a una base de datos.

**Cliente:** El rol de una aplicación o sistema informático que solicita y/o consume los servicios proporcionados por otro que tiene el rol de servidor.

**Composer:** Composer es una herramienta para la gestión de dependencias en PHP. Le permite declarar las bibliotecas de las que depende un proyecto.

**Compras:** La compra es la serie de fases por los que pasa un consumidor antes de decidirse a adquirir un producto.

**Controladores:** Una función de laravel en la que se define una clase que controla las rutas de una página web relacionada.

**CRUD:** Crear, Leer, Actualizar y Eliminar (CLAE, o CRUD en inglés) son las cuatro funciones básicas que los modelos deberían poder hacer, como máximo.

**CSS:** Las hojas de estilo en cascada (CSS) son un lenguaje de hojas de estilo utilizado para describir la presentación de un documento escrito en un lenguaje de marcado como HTML.

**Diagrama Caso Uso:** Un diagrama de caso de uso es una descripción de cómo una persona que realmente usa ese proceso o sistema logrará un objetivo.

**Diagrama De Entidad Relación:** Un diagrama de entidad relación (DER) muestra las relaciones de los conjuntos de entidades almacenados en una base de datos.

**Diseño Gráfico:** comunicación visual mediante una hábil combinación de texto e imágenes en anuncios, revistas, libros, etc.

**eBay:** eBay Inc. es una corporación multinacional estadounidense de comercio electrónico con sede en San José, California, que facilita las ventas de consumidor a consumidor y de empresa a consumidor a través de su sitio web.

**Email:** El correo electrónico, más comúnmente conocido como Email o E-mail desde 1993, es un método de intercambio de mensajes digitales de un autor a uno o más destinatarios.

**Framework:** El Framework, o marcos son software que son desarrollados y utilizados por los desarrolladores para crear aplicaciones.

**Front End:** La parte de un sistema de hardware o software que está más cerca del usuario.

**Fullstack:** Un desarrollador web de Fullstack (pila completa) es una persona que puede desarrollar software de cliente y servidor.

**Git:** En el desarrollo de software, Git es un sistema distribuido de control de versiones y gestión de código fuente con énfasis en la velocidad.

**GitHub:** GitHub es un servicio de alojamiento basado en la web para proyectos de desarrollo de software que utilizan el sistema de control de revisión de Git.

**Google Fonts:** Inicialmente lanzado en 2010 como Google Web Fonts, Google Fonts es una colección de interfaces interactivas de programación de aplicaciones que permiten a los usuarios utilizar fuentes web en su sitio web.

**HTML:** Hyper Text Markup Language es el lenguaje de marcado principal para crear páginas web y otra información que se puede mostrar en un navegador web.

**Huerta:** Un jardín o un área de tierra para el cultivo de árboles frutales o de nueces.

**Interfaz:** La parte de un programa informático que controla la forma en que un programa interactúa con un usuario; la forma de introducir y enviar datos, y la forma en que se presenta la información en un monitor de computadora.

**Internet:** Una gran red de numerosas computadoras conectadas a través de una serie de nodos principales de computadoras de alta velocidad que tienen canales de comunicación de alta velocidad entre los nodos principales, y numerosos nodos menores que permiten la comunicación electrónica entre millones de computadoras en todo el mundo.

**JavaScript:** JavaScript es un lenguaje de programación de computadoras interpretado. Originalmente se implementó como parte de los navegadores web para que los scripts del lado del cliente pudieran interactuar con el usuario, controlar el navegador, comunicarse de forma sincrónica y alterar el contenido del documento que se mostraba.

**JetStream:** Laravel JetStream es un nuevo andamio de aplicación para Laravel. Laravel JetStream reemplaza la interfaz de usuario de autenticación de Laravel heredada disponible para versiones anteriores de Laravel.

**Jira:** Jira es una aplicación de software utilizada para el seguimiento de problemas y la gestión de proyectos. La herramienta, desarrollada por la compañía de software australiana Atlassian, se ha convertido en ampliamente utilizada por los equipos de desarrollo ágil para rastrear errores, historias, y otras tareas.

**Laragon:** Laragon proporciona un entorno de desarrollo portátil, aislado, rápido y potente para PHP, Node.js, Python, Java, Go, Ruby para computadoras con Windows.

**Livewire:** Livewire es un marco de pila completa para Laravel que simplifica la construcción de interfaces dinámicas, sin abandonar la comodidad de Laravel.

**Login:** En seguridad informática, iniciar sesión (o iniciar sesión, iniciar sesión o iniciar sesión) es el proceso mediante el cual un individuo obtiene acceso a un sistema informático identificándose y atentándose.

**Maquetas:** Una descripción hipotética de una entidad o proceso complejo.

**Marco Metodológico:** El marco metodológico es la parte de nuestra investigación donde exponemos el método teórico y práctico utilizado para analizar el problema planteado o el tema que estamos tratando.

**Material Design:** Material Design es un lenguaje de diseño desarrollado por Google en 2014. Ampliando las "tarjetas" que debutaron en Google Now, Material Design utiliza más diseños basados en cuadrículas, animaciones y transiciones receptivas, relleno y efectos de profundidad como iluminación y sombras.

**Materialize CSS:** Materialize es básicamente un marco CSS que se utiliza para crear sitios web receptivos. El marco Materialize se basa básicamente en el concepto de diseño de materiales de Google.

**Mercado Libre:** Mercado Libre, Inc. es una empresa argentina con sede en Buenos Aires, constituida en Estados Unidos que opera mercados en línea dedicados al comercio electrónico y subastas en línea, incluyendo mercadolibre.com.

**Metodología:** Una metodología es un proceso científico dirigido a identificar el conjunto más adecuado de métodos para liderar investigaciones y analizar asuntos, con componentes específicos como paradigma, modelo teórico, fases, técnicas.

**Metodología Ágil:** La metodología Agile es una práctica que fomenta el desarrollo continuo y las pruebas a lo largo del ciclo de vida de desarrollo de software de un proyecto.

**Metodología De Desarrollo:** Una metodología / proceso de desarrollo de software, también conocido como el ciclo de vida de desarrollo de software es el conjunto de principios o reglas que dirige al equipo de desarrollo a la finalización exitosa del proyecto.

**Microsoft Project:** Microsoft Project es un programa de software de administración de proyectos, desarrollado y vendido por Microsoft, que está diseñado para ayudar a un gerente de proyecto a desarrollar un plan, asignar recursos a las tareas, realizar un seguimiento del progreso, administrar el presupuesto y analizar las cargas de trabajo.

**Migraciones:** Laravel Migrations es el método de control de versiones para la base de datos. Se hace para permitir que el equipo modifique la base de datos y la comparta entre dominios y plataformas.

**MySQL:** MySQL es un sistema de gestión de bases de datos relacionales de código abierto que se ejecuta como un servidor que proporciona acceso multiusuario a una serie de bases de datos.

**Node.js:** Node.js es un entorno de tiempo de ejecución de JavaScript de código abierto, multiplataforma y Back End que se ejecuta en el motor V8 y ejecuta código JavaScript fuera de un navegador web.

**NPM:** Node Package Manager (NPM) es uno de los registros de software más grandes del mundo. Viene incluido con node.js, un entorno de servidor de código abierto.

**PHP:** Es un lenguaje de scripting de propósito general especialmente adecuado para el desarrollo web. Fue creado originalmente por el programador danés-canadiense Rasmus Lerdorf en 1994.

**Pixabay:** Pixabay.com es un sitio web internacional, copyleft y de uso gratuito para compartir fotos, ilustraciones, gráficos vectoriales, imágenes de películas y música.

**Producto:** Productos básicos ofrecidos a la venta.

**Redes Sociales:** Las redes sociales se refieren a los medios de interacción entre las personas en los que crean, comparten e intercambian información e ideas en comunidades y redes virtuales.

**Registración:** El acto de registrarse o registrarse para algo.

**Reportes:** Para relacionar detalles de un evento, o incidente; para contar, describir.

**Rutas:** El grupo Laravel Route permite al desarrollador enrutar todos los grupos. Esto ayuda a crear códigos limpios y quien se haga cargo del desarrollo del sitio web podrá seguir la rutina exacta.

**Scrum:** Scrum es un marco de desarrollo de software ágil iterativo e incremental para administrar proyectos de software y desarrollo de productos o aplicaciones.

**Semillas:** Laravel seeder utiliza para insertar varias filas de datos en la base de datos mediante un comando.

**Servidores:** Un servidor es un sistema que responde a las solicitudes a través de una red informática para proporcionar, o ayudar a proporcionar, un servicio de red.

**StarUML:** StarUML™ es una plataforma de modelado de software que soporta UML (Unified Modeling Language). Se basa en UML versión 1.4 y proporciona once tipos diferentes de diagrama, y acepta la notación UML 2.0. Apoya activamente el enfoque MDA (Model Driven Architecture) al admitir el concepto de perfil UML.

**Tabla De Gantt:** Una representación gráfica de las tareas y recursos necesarios para completar un trabajo o proyecto; puede mostrar rangos de posibles fechas de inicio y finalización y las relaciones entre las tareas; se utiliza para identificar cuellos de botella y asignar prioridades.

**Tailwind:** Tailwind CSS se describe a sí mismo como un marco CSS de utilidad primero. En lugar de centrarse en la funcionalidad del elemento que se está diseñando, Tailwind se centra en

cómo debe mostrarse. Esto hace que sea más fácil para el desarrollador probar nuevos estilos y cambiar el diseño.

**UML:** Abreviatura de Unified Modeling Language, es un lenguaje de modelado estandarizado que consiste en un conjunto integrado de diagramas, desarrollado para ayudar a los desarrolladores de sistemas y software a especificar, visualizar, construir y documentar los artefactos de los sistemas de software, así como para el modelado de negocios y otros sistemas que no son de software.

**Visio:** Microsoft Visio es una aplicación de diagramación y gráficos vectoriales y forma parte de la familia Microsoft Office. El producto se introdujo por primera vez en 1992, fabricado por Shapeware Corporation. Fue adquirida por Microsoft en 2000.

**Vistas:** Las vistas contienen el código HTML requerido por la aplicación, y es un método en Laravel que separa la lógica del controlador y la lógica de dominio de la lógica de presentación.

**Visual Studio:** Microsoft Visual Studio es un entorno de desarrollo integrado de Microsoft. Se utiliza para desarrollar programas informáticos, así como sitios web, aplicaciones web, servicios web y aplicaciones móviles.

**WhatsApp:** WhatsApp Messenger es una aplicación de mensajería móvil multiplataforma que le permite intercambiar mensajes sin tener que pagar por SMS.

## ANEXOS

---

### CÓDIGO FUENTE

Código fuente: [https://github.com/SethWack/huerta\\_limones](https://github.com/SethWack/huerta_limones)

## HOJAS DE REFERENCIAS

### Laravel

**Cheatography**

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

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

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>

## HTML

Cheatography		HTML4 Cheat Sheet by Dave Child (DaveChild) via cheatography.com/1/cs/17/	
<b>HTML4 Document Outline</b>	<!DOCTYPE> Version of (X)HTML	<b>HTML4 Lists</b>	
<html>	HTML document	<ol>	Ordered list
<head>	Page information	<ul>	Unordered list
<body>	Page contents	<li>	List item
<b>HTML4 Comments</b>	<!-- Comment Text -->	<dl>	Definition list
		<dt>	Definition term
		<dd>	Term description
<b>HTML4 Page Information</b>		<b>HTML4 Forms</b>	
<base>	Base URL	<form>	Form
<meta>	Meta data	<fieldset>	Collection of fields
<title>	Title	<legend>	Form legend
<link>	Relevant resource	<label>	Input label
<style>	Style resource	<input>	Form input
<script>	Script resource	<select>	Drop-down box
<b>HTML4 Document Structure</b>		<optgroup>	Group of options
<h1> to <h6>	Headings	<option>	Drop-down options
<div>	Page section	<textarea>	Large text input
<span>	Inline section	<button>	Button
<p>	Paragraph		
 	Line break		
<hr>	Horizontal rule		
<b>HTML4 Links</b>		<b>HTML4 Tables</b>	
<a href="">	Page link	<table>	Table
<a href="mailto:>">	Email link	<caption>	Caption
<a name="name">	Anchor	<thead>	Table header
<a href="#name">	Link to anchor	<tbody>	Table body
		<tfoot>	Table footer
		<colgroup>	Column group
		<col>	Column
		<tr>	Table row
		<th>	Header cell
		<td>	Table cell
<b>HTML4 Text</b>		<b>HTML4 Images</b>	
		<img>	Image
		<map>	Image Map
		<area>	Area of Image Map
		<b>HTML4 Objects</b>	
		<object>	Object
		<param>	Parameter
		<b>HTML4 Core Attributes</b>	
		class	style
		id	title
		Core Attributes may not be used with base,	

2 Hoja de Referencia HTML

## PHP

PHP Cheat Sheet		
by Dave Child (DaveChild) via cheatography.com/1/cs/2/		
<b>PHP Array Functions</b>	<b>PHP Filesystem Functions</b>	<b>Regular Expressions Syntax</b>
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/>		
<b>PHP String Functions</b>	<b>PHP Date and Time Functions</b>	<b>Pattern Modifiers</b>
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 (formating, 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)	<b>PHP Regular Expressions Functions</b>	
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 <b>Dave Child</b> (DaveChild) <a href="http://cheatography.com/davechild/">cheatography.com/davechild/</a> <a href="http://aloneonahill.com">aloneonahill.com</a>		
<p>Published 19th October, 2011.          Last updated 29th February, 2020.          Page 1 of 2.</p>		
<p>Sponsored by <b>Readable.com</b>          Measure your website readability!  <a href="https://readable.com">https://readable.com</a></p>		

## 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 (-922372036854-775808 to 92237203-6854775807)				
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 Mathematical Functions		MySQL Date and Time Functions		MySQL Control Flow Functions	
ABS	COS	DAYOFWEEK	DATE_SUB	IF	NULLIF
SIGN	SIN	WEEKDAY	ADDDATE	IFNULL	
MOD	TAN	DAYOFMONTH	SUBDATE		
FLOOR	ACOS	DAYOFYEAR	EXTRACT		
CEILING	ASIN	MONTH	TO_DAYS		
ROUND	ATAN, ATAN2	DAYNAME	FROM_DAYS		
DIV	COT	MONTHNAME	DATE_FORMAT		
EXP	RAND	QUARTER	TIME_FORMAT		
LN	LEAST	WEEK	CURRENT_DATE		
LOG, LOG2, LOG10	GREATEST	YEAR	CURRENT_TIME		
POW	DEGREES	YEARWEEK	NOW		
POWER	RADIANS	HOUR	SYSDATE		
SQRT	TRUNCATE	MINUTE	UNIX_TIMESTAMP		
PI		SECOND	FROM_UNIXTIME		
MySQL String Functions		PERIOD_ADD	SEC_TO_TIME		
ASCII	SUBSTRING	PERIOD_DIFF	TIME_TO_SEC		
ORD	MID	DATE_ADD			
CONV	SUBSTRING_INDEX				
BIN	LTRIM				
OCT	RTRIM				
HEX	TRIM				
CHAR	SOUNDEX				
CONCAT	SPACE				
CONCAT_WS	REPLACE				



By **Dave Child** (DaveChild)  
[cheatography.com/davechild/](http://cheatography.com/davechild/)  
[aloneonahill.com](http://aloneonahill.com)

Published 19th October, 2011.  
 Last updated 25th February, 2020.  
 Page 1 of 2.

Sponsored by **Readable.com**  
 Measure your website readability!  
<https://readable.com>

## Git

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