



## **PROYECTO DE TITULACIÓN**

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

**Que Presenta**

**Seth Aldanid Torres Delgado**

**Para obtener el Titulo de Ingeniero en Sistemas**

**Computacionales**

**Octubre – Diciembre 2021**

## **Agradecimientos**

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

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

*Agradezco a mi familia por tener la paciencia durante estos años de carrera. Especial durante tiempos bajos, y durante los cambios de escuelas. Cuando tuve que mover de ciudades, y los gastos que se hicieron. Yo les doy todas las gracias.*

*Seth*

**Indice General**

Agradecimientos.....	I
<b>Indice General.....</b>	II
<b>Indice de Tablas y Figuras.....</b>	V
Capítulo I: Introducción.....	1
1.1 Antecedentes.....	1
1.2 Planteamiento del Problema.....	3
1.3 Hipótesis.....	3
1.4 Objetivos.....	3
1.4.1 Generales.....	3
1.4.2 Específicos.....	4
1.5 Alcances y Metas.....	4
<b>Capitulo II: Marco Teórico y Metodológico.....</b>	6
2.1 Antecedentes de la Investigación.....	6
2.2 Marco Metodológico.....	10
2.3 Justificación de Investigación.....	10
<b>Capitulo III: La Propuesta.....</b>	11
3.1 <b>Presentacion.....</b>	11
3.2 Justificación.....	11
<b>Capitulo IV: Análisis.....</b>	12
4.1 Requerimientos.....	12

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

6.7 Materialize.....	21
<b>Capítulo VII: Desarrollo.....</b>	<b>22</b>
7.1 <b>Preparacion</b> Inicial.....	22
7.2 <b>Instalacion</b> de Proyecto Laravel.....	22
7.3 <b>Implementacion</b> de Plugins y Paquetes.....	22
7.3.1 <b>Instalacion</b> de Livewire y Jetstream.....	22
7.3.2 <b>Instalacion</b> de Materialize.....	23
7.4 Desarrollando el Sistema.....	24
<b>Capítulo VIII: Implementación y Pruebas.....</b>	<b>25</b>
<b>Capítulo IX: Resultados.....</b>	<b>26</b>
Bibliografías.....	27
Glosario de <b>Terminos</b> .....	30
Anexos.....	30
12.1 <b>Codigo</b> Fuente.....	30
12.2 Hojas de Referencias.....	31
12.2.1 Laravel.....	31
12.2.2 HTML.....	32
12.2.3 PHP.....	33
12.2.4 MySQL.....	34
12.2.5 Git.....	35

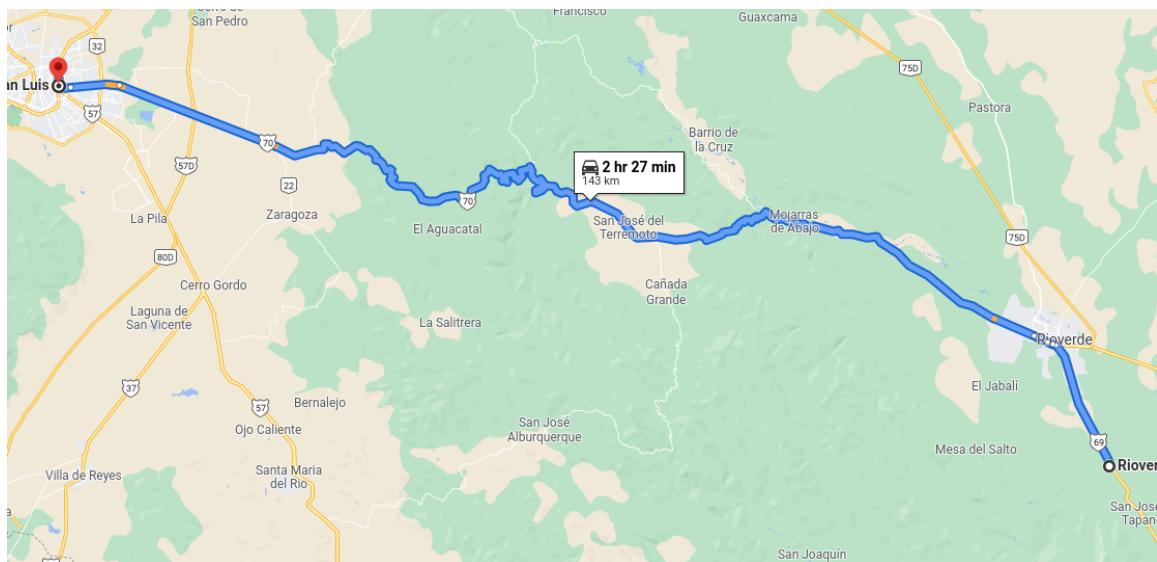
## Indice de Tablas y Figuras

1.1 Mapa de <b>direccion</b> del Pueblo.....	1
1.2 Imagen de la Huerta.....	2
4.1 Diagrama Caso Uso.....	14
4.2 Diagrama Base de Datos.....	15
5.1 Bosquejo de <b>Pagina</b> Principal.....	16
5.2 Bosquejo de Tienda.....	17
5.3 Bosquejo de Login.....	18
7.1 Imagen de Archivo CSS que Buscar.....	23
7.2 Imagen de Visual Studio.....	24
13.1 Hoja de Referencia Laravel.....	31
13.2 Hoja de Referencia HTML.....	32
13.3 Hoja de Referencia PHP.....	33
13.4 Hoja de Referencia MySQL.....	34
13.5 Hoja de Referencia Git.....	35

## Capítulo I: Introducción

### 1.1 Antecedentes

**E**l Pueblo de Plazuela, es un ejido de Rioverde SLP. El clima es **calido**, con muchas lluvias por ser un area montanosa. Aqui se dan a crecer arboles frutales como: naranjas, limones, mangos, y papayas.



*1.1 Mapa de dirección del Pueblo*

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



### *1.2 Imagen de la Huerta*

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

## **1.2 Planteamiento del Problema**

El problema que tienen los jefes de la huerta es la baja distribución de limones.

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

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

### **1.3 Hipótesis**

En si los jefes de la huerta quieren una pagina para poder difundir información de la venta de limones. También quieren una forma para poder tener un control de clientes, y control de limones dados durante la cosecha. Implementar unas graficas y números de precios de el costo en kilo del limón en mes por mes para poder hacer predicciones en cuando vender a mejor beneficio.

### **1.4 Objetivos**

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

#### **1.4.1 Generales**

El objetivo es:

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

#### **1.4.2 Específicos**

Los objetivos específicos son:

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

#### **1.5 Alcances y Metas**

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

## Capítulo II: 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. [3]

En si una pagina web es un documento digital hecho en html, dados por servidores, y pueden ser leídos en un navegador de web [4]. 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 [5].

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 [6]. El back end es la programación que no se mira en la pagina. Es como un motor en la que todo se hace, como dar información a la base de datos o dar información que es solicitado del cliente [7].

Las **paginas** web son programadas para front end son HTML y CSS. HTML (Hyper Text Markup Language) es un lenguaje para crear margenes y definir la estructura de la **pagina** usando etiquetas [8]. CSS (Cascade Style Sheets) es como cada etiqueta y elementos de HTML es definido y dibujado en una **pagina**. Como

cambiar el color de texto, o definir ciertas etiquetas tamaños específicos dependiendo si la **pagina** es vista en una computadora o un teléfono celular [9].

Otros programas que son usadas para back end son como PHP, MySQL, y JavaScript. PHP es un lenguaje de scripting que se puede integrar a una **pagina** de web Para hacer paginas mas dinámicas. En si se puede usar para hacer programación y ejecución del lado del servidor [10][11]. MySQL es una lenguaje de base de datos, creado para tener un control y poder guardar información de un servidor. Con una base de datos se puede gestionar información [12][13]. JavaScript es otro lenguaje de scripting usado en **paginas** web. Es implementado y corre en el lado del cliente, en comparación de PHP. Era creada para tener un medio de interacción dinámica para un cliente. Es mas viejo que php; originalmente creado como una versión web de Java, pero **termino** siendo algo mas diferente. [10][14].

Muchas paginas modernas son creadas con frameworks. Un framework es una herramienta para ayudar en la creación rápida de **paginas** con diseños estandarizados [15]. Uno de los frameworks que se usara en este proyecto es Laravel. Laravel es un framework que usa php como su programación principal [16]. Laravel usa un numero de herramientas para otras funciones durante el desarrollo del proyecto como: Livewire, JetStream, y Tailwind. JetStream es un **modulo** de Laravel para la creación de Login, Registros, Verificación de E-mail, y Autenticador de 2 pasos para clientes [17]. Livewire es un framework fullstack que corre dentro de Laravel. Se usa para crear interfaces dinámicas [18]. Fullstack significa que es una aplicación de web que tiene herramientas de front end y back end [19]. Tailwind

es un framework de CSS creado para poder diseñar **páginas** modernas sin tener que escribir mucho código css [20].

Proyectos como estos requieren control de versiones para poder tener un desarrollo limpio y con menos defectos al salir al **publico**. 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 [21]. 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 [22]. 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 [23].

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

tiempo para crear un programa o aplicación[1]. Un método ágil común es el método SCRUM. Creado con la intención de hacer un marco en que se pueda hacer sprints (carreras) de corto tiempo, en que se hace lo que se va hacer primero, luego se hace una planeación rápida, luego se hace otro sprint para crear el producto. al final se hace juntas con el equipo y(o) clientes para dar retroalimentación para luego empezar de nuevo, hasta que sea satisfecha el proyecto que se esta haciendo [2].

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

## 2.2 Marco Metodológico

En estos tiempos con pandemias, y sin poder hacer contactos con gentes cercanas, es preferible tener una presencia en linea para hacer compras. Se ha visto que mas gente en estos tiempos son mas probables en comprar en linea que en tiempos pasados [33].

## 2.3 Justificación de Investigación

Una gran ventaja es de tener una presencia en linea. Se puede usar para tener un contacto con los clientes en una forma interactiva. Se puede tener una forma de servicio 24/7. Se puede tener un contacto para dar ciertos anuncios como tiempos de descuentos o cuando hay nuevos productos [34].

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

## **Capitulo III: La Propuesta**

### **3.1 Presentacion**

**S**e presenta una pagina web que esta disenado para difundir informacion de la huerta. La huerta tambien tendra acceso a una pagina especial para tener un control de los productos que entran y salen. Los clientes tendrán una pagina para hacer compras de los productos.

### **3.2 Justificación**

## Capítulo IV: Análisis

### 4.1 Requerimientos

#### 4.1.1 Funcionales

Las requerimientos Funcionales son descripciones implícitas del comportamiento de un programa [30]. En si son las funciones que debe tener un programa. En este caso, el proyecto debe de tener estas funciones:

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

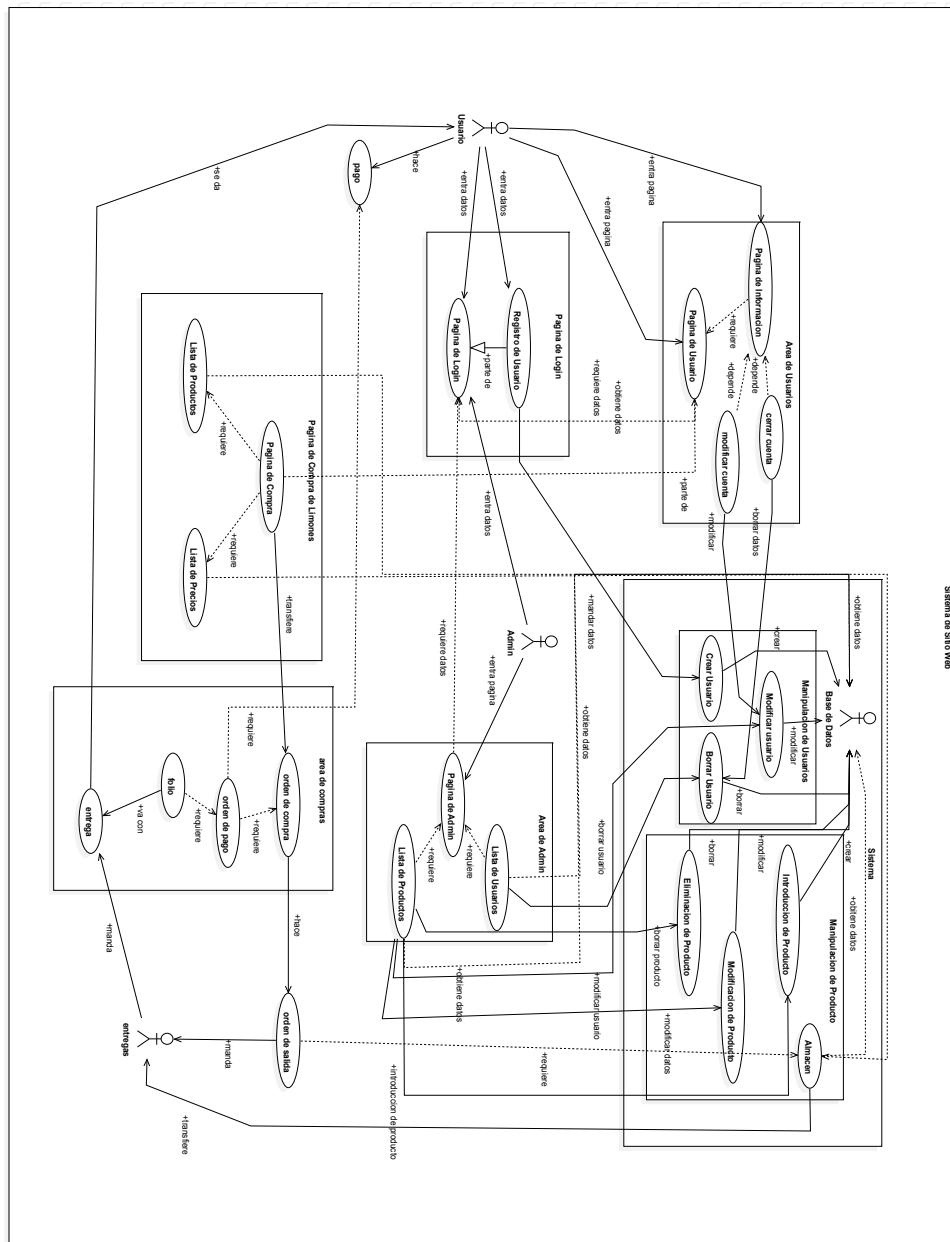
#### 4.1.2 No Funcionales

Los requerimiento no funcionales son requisitos que hacen restricciones en el diseño del proyecto [31]. 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.

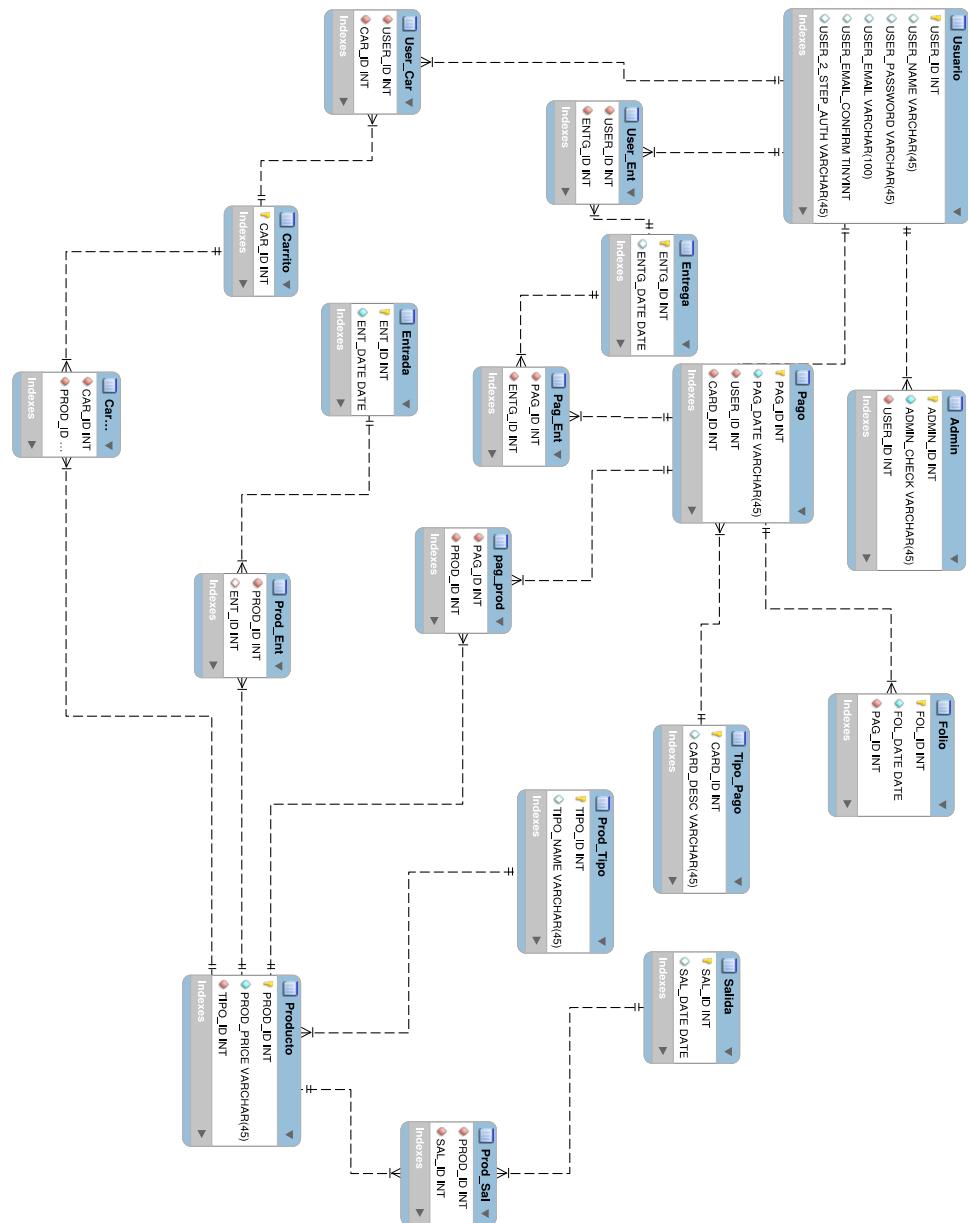
- La compra de producto se puede basar el precio en precios generales del mercado
- Se puede elegir la forma de pago y el tipo de entrega en linea.

## 4.2 Diagrama de Caso Uso



#### *4.1 Diagrama Caso Uso*

### 4.3 Diagra Entidad Relacion



4.2 Diagrama Entidad Relacion

## Capítulo V Diseño

### 5.1 Bosquejos

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

#### 5.1.1 Bosquejo de pagina principal

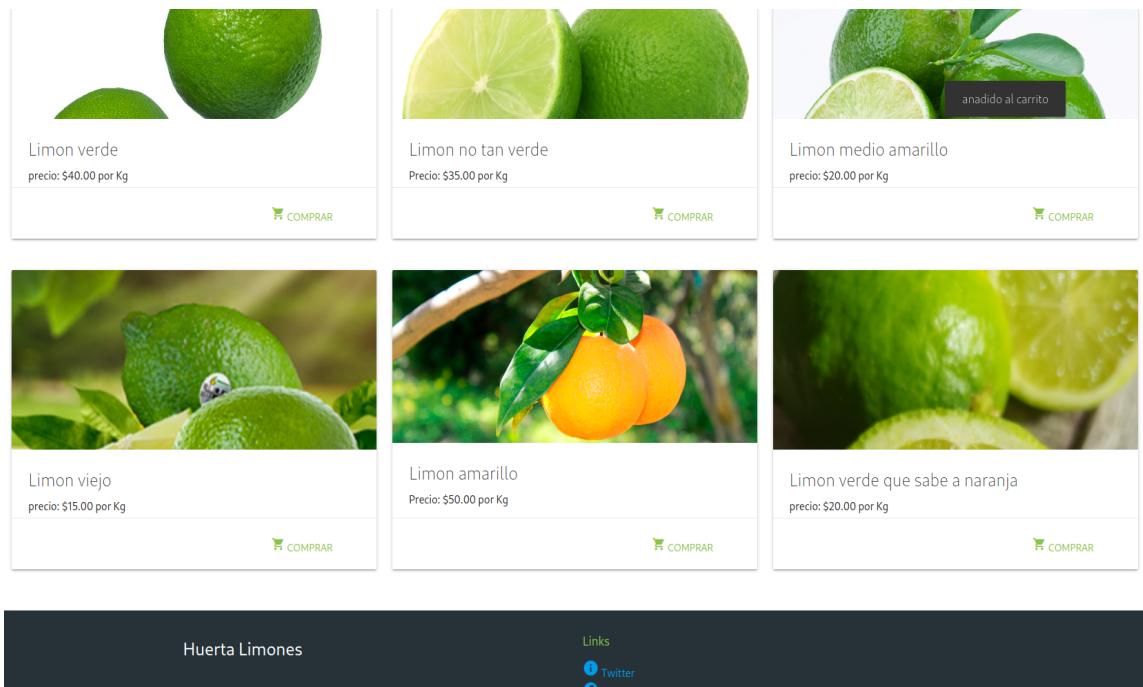


#### 5.1 Bosquejo de Pagina Principal

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

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

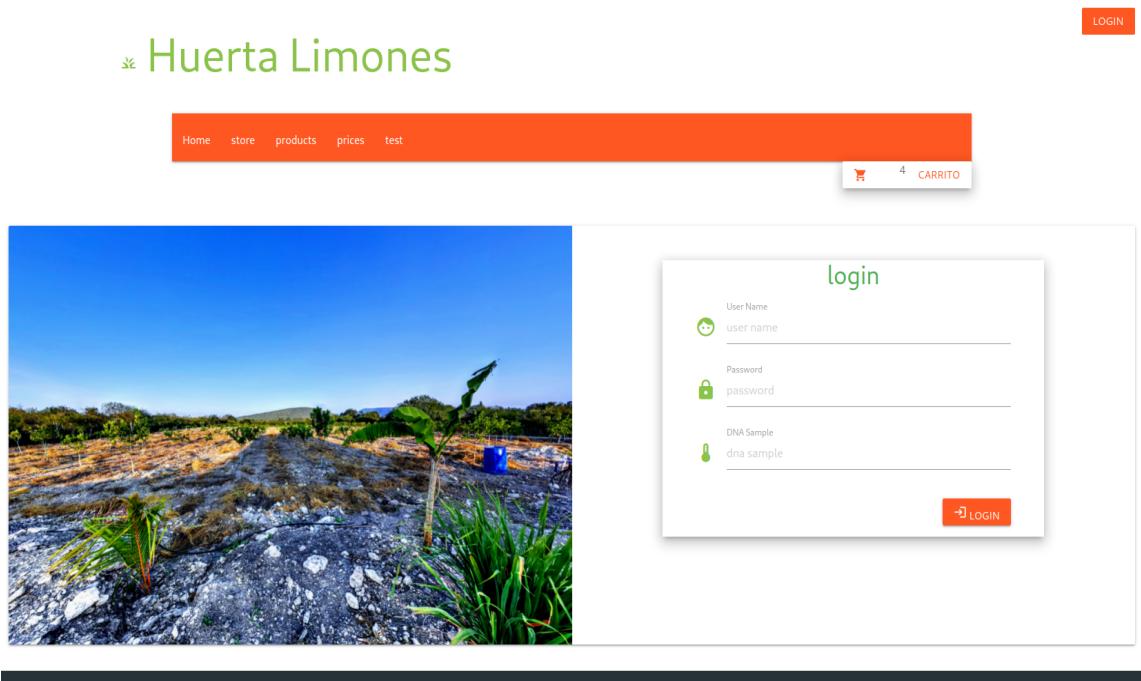
### 5.1.2 Bosquejo de Pagina de Tienda



### 5.2 Bosquejo de Tienda

En este bosquejo se muestran el interfaz en como se venden los productos. Se usaran programacion PHP para hacer **busquedas** en la base de datos, y demostrar los productos disponibles para vender. Se muestra una imagen del producto, con descripcion y el precio. Al precionar el boton de comprar se pasara al carrito para despues pagar.

### 5.1.3 Bosquejo de Login



#### 5.3 Bosquejo de Login

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

## Capítulo VI: 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

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

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

### 6.2 Git

Es un software para controlar versiones, cambios, y ramas de un proyecto de software. Se distribuye localmente, o en **linea**. Cada usuario tiene un clon del proyecto y al hacer cambios se puede hacer localmente, para despues empujarlo al proyecto original [22].

#### 6.2.1 Github

Github es una **página** que da el servicio de alojamiento de proyectos, para desarrolladores. Usan Git para controlar versiones y hacer ramas para **multiples** usuarios [23].

### 6.3 Base de Datos

La base de datos que se esta usando es MySQL. Es un sistema de Base de Datos para guardar información y datos que son usados en servidores, haciendo consultas al sistema de Base de Datos [13].

#### 6.3.1 MySQL Workbench

MySQL Workbench es un Software para administración, diseño, y mantenimiento de base datos visualmente [32].

### 6.4 PHP

PHP es un lenguaje de programacion para paginas web. Se compilan en el lado del servidor, y luego es dado al cliente al tiempo de servir una pagina con php. Se puede usar, como en este caso para, crear frameworks en que construira el sistema [11].

#### 6.4.1 Composer

Composer es un administrador de dependencias de PHP. Se usa para instalar herramientas, o plugins para un proyecto [33].

### 6.5 Laravel

Es un framework para aplicaciones web. Es una base para crear paginas web rapido, y sin tantos problemas [16].

### 6.5.1 Livewire

Livewire es un plugin de Laravel. Es para crear paginas web **dinamicas** sin tener que usar frameworks de javascript completas [18].

### 6.5.2 Jetstream

Es un plugin de Laravel con Livewire, para empezar **paginas** web **mas rapido**. Tiene **paginas** hechas para login, **registro**, **autenticacion**, **autenticacion** de 2 pasos [17].

## 6.6 JavaScript

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

### 6.6.1 NPM

NPM es una **funcion** hecha con JavaScript, para manejar paquetes, descargar herramientas, correr paquetes sin descargar usando NPX, compartir **codigo** con otros usuarios, y acutalizar **codigo** [37].

## 6.7 Materialize

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

## Capítulo VII: Desarrollo

### 7.1 Preparacion 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 Instalacion de Proyecto Laravel

Despues se instala el proyecto de laravel usando el comando:

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

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

```
$ php artisan serv
```

### 7.3 Implementacion de Plugins y Paquetes

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

#### 7.3.1 Instalacion de Livewire y Jetstream

Para instalar Jetstream es el comando:

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

Luego se hace el comando:

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

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

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

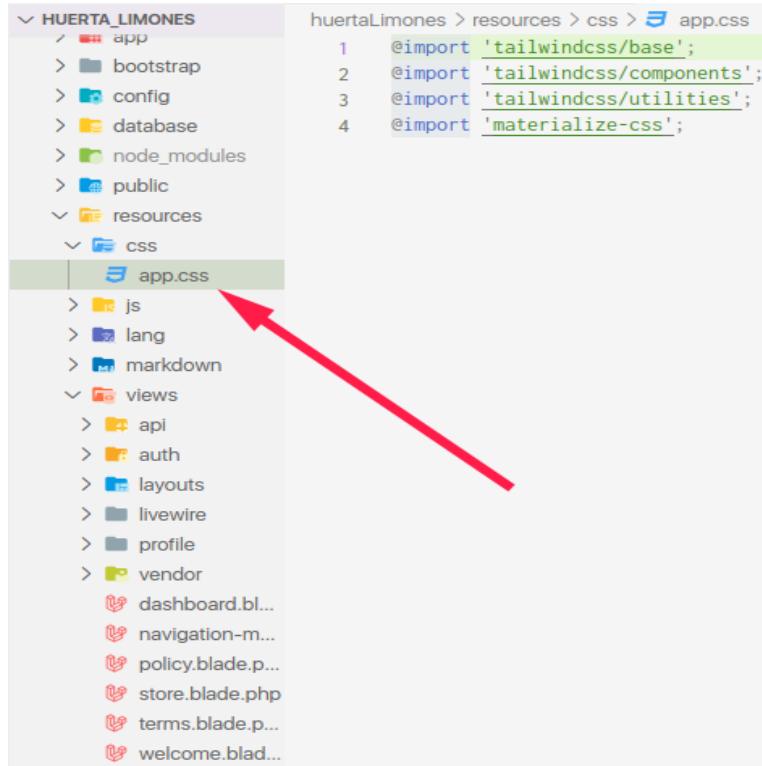
### 7.3.2 Instalacion de Materialize

Para **instalara** Materialize a tu sistema es correr:

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

**Despues** se busca el archivo: archivo\_de\_proyecto/resources/css/app.css. Dentro de el archivo se introduce el texto:

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



7.1 Imagen de Archivo CSS que Buscar

## 7.4 Desarrollando el Sistema

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

The screenshot shows a PhpStorm IDE interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbars:** EXPLORE, PROBLEMS, OUTPUT, TERMINAL, DEBUG CONSOLE.
- Left Sidebar (Project Tree):** Shows the project structure under "HUERTA\_LIMONES". Key components include:
  - bootstrap
  - config
  - database
  - node\_modules
  - public
  - resources
    - css
    - js
    - lang
    - markdown
    - views
      - api
      - auth
      - layouts
      - livewire
      - 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
      - .styling.yml
      - .artian
      - composer.json
      - composer.lock
      - package-lock.json
- Central Area:** Displays the content of the "navigation-menu.blade.php" file. The code includes logic for dropdown menus, CSRF protection, and user authentication.
- Bottom Status Bar:** Shows the current file path ("~/huerta\_limoness/main"), the terminal command ("cd huerta\_limoness"), and the PHP version ("PHP 8.0.1"). It also displays several terminal tabs with logs related to Laravel startup and PDO SQLite errors.
- Right Side:** A large, detailed file tree view showing the entire project structure.

## *6.1 Imagen de Visual Studio*

## **Capítulo VIII: Implementación y Pruebas**

## **Capítulo IX: Resultados**

## Bibliografías

- [1] 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>
- [2] 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>
- [3] Definición de Huerta [Internet]. Definición.de. [cited 2021Oct10]. Available from: <https://definicion.de/huerta/>
- [4] Web page [Internet]. Web Page Definition. [cited 2021Oct10]. Available from: <https://techterms.com/definition/webpage>
- [5] Server [Internet]. Paessler. [cited 2021Oct10]. Available from:  
<https://www.paessler.com/it-explained/server>
- [6] 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/>
- [7] 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/>
- [8] 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)

- [9] CSS introduction. [cited 2021Oct10]. Available from:  
[https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)
- [10] Nixon R, Carreño FM. Aprender PHP, mysql y javascript: Con jquery, CSS Y HTML5. 5th ed. Barcelona: Marcombo; 2019.
- [11] What is php? [Internet]. php. [cited 2021Oct10]. Available from:  
<https://www.php.net/manual/en/intro-whatis.php>
- [12] Silberschatz A, Korth HF, Sudarshan S. Database system concepts. Boston: McGraw-Hill; 2006.
- [13] What is mysql? A beginner-friendly explanation [Internet]. Kinsta®. 2021 [cited 2021Oct10]. Available from:  
<https://kinsta.com/knowledgebase/what-is-mysql/>
- [14] 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)
- [15] What is a web framework? [Internet]. GoodFirms. 2020 [cited 2021Oct10]. Available from: <https://www.goodfirms.co/glossary/web-framework/>
- [16] Installation [Internet]. Laravel. [cited 2021Oct10]. Available from: <https://laravel.com/docs/8.x>
- [17] # introduction [Internet]. Laravel Jetstream. [cited 2021Oct10]. Available from: <https://jetstream.laravel.com/1.x/introduction.html>
- [18] Livewire [Internet]. Laravel. [cited 2021Oct10]. Available from: <https://laravel-livewire.com/>

- [19] What is full stack development ? [Internet]. GeeksforGeeks. 2019 [cited 2021Oct10]. Available from: <https://www.geeksforgeeks.org/what-is-full-stack-development/>
- [20] Rapidly build modern websites without ever leaving your HTML. [Internet]. Tailwind CSS. [cited 2021Oct10]. Available from: <https://tailwindcss.com/>
- [21] Atlassian. What is version control: Atlassian Git Tutorial [Internet]. Atlassian. [cited 2021Oct10]. Available from: <https://www.atlassian.com/git/tutorials/what-is-version-control>
- [22] 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>
- [23] 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/>
- [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] 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>
- [31] Requisitos no funcionales [Internet]. EcuRed. [cited 2021Oct15]. Available from: [https://www.ecured.cu/Requisitos\\_no\\_funcionales](https://www.ecured.cu/Requisitos_no_funcionales)

## Glosario de Terminos

## Anexos

### 12.1 Código Fuente

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

## 12.2 Hojas de Referencias

### 12.2.1 Laravel

**Cheatography**

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

<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');</code> <code>Route::resource('post', 'PostController', array('except' =&gt; 'show'));</code> <code>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>

## 12.2.2 HTML

<b>Cheatography</b>		HTML4 Cheat Sheet by Dave Child (DaveChild) via cheatography.com/1/cs/17/	
<b>HTML4 Document Outline</b>		<b>HTML4 Lists</b>	
<!DOCTYPE> Version of (X)HTML		<ol> Ordered list	
<html> HTML document		<ul> Unordered list	
<head> Page information		<li> List item	
<body> Page contents		<dl> Definition list	
<b>HTML4 Comments</b>		<dt> Definition term	
<!-- Comment Text -->		<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	
<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	
<b>HTML4 Text</b>		<tfoot> Table footer	
		<colgroup> Column group	
		<col> Column	
		<tr> Table row	
		<th> Header cell	
		<td> Table cell	
<b>HTML4 Images</b>		<b>HTML4 Objects</b>	
<img> Image		<object> Object	
<map> Image Map		<param> Parameter	
<area> Area of Image Map		<b>HTML4 Core Attributes</b>	
		class style	
		id title	
		Core Attributes may not be used with base,	

### 12.2.3 PHP

**Cheatography**

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 ...) array_filter (arr, function) array_flip (arr) array_intersect (arr1, arr2 ...) array_merge (arr1, arr2 ...) array_pop (arr) array_push (arr, var1, var2 ...) array_reverse (arr) array_search (needle, arr) array_walk (arr, function) count (count) in_array (needle, haystack)	clearstatcache () copy (source, dest) fclose (handle) fgets (handle, len) file (file) filemtime (file) filesize (file) file_exists (file) fopen (file, mode) fread (handle, len) fwrite (handle, str) readfile (file)	^ Start of string \$ End of string . Any single character (a b) a or b (...) Group section [abc] In range (a, b or c) [^abc] Not in range \s White space a? Zero or one of a a* Zero or more of a a*? Zero or more, ungreedy a+ One or more of a a+? One or more, ungreedy a{3} Exactly 3 of a a{3,} 3 or more of a a{6} Up to 6 of a a{3,6} 3 to 6 of a a{3,6}? 3 to 6 of a, ungreedy \ Escape character [:punct:] Any punctuation symbol [:space:] Any space character [:blank:] Space or tab
<b>PHP String Functions</b>	<b>PHP Date and Time Functions</b>	
crypt (str, salt) explode (sep, str) implode (glue, arr) nl2br (str) sprintf (fmt, args) strip_tags (str, allowed_tags) str_replace (search, replace, str) strpos (str, needle) strrev (str) strstr (str, needle) strtolower (str) strtoupper (str) substr (string, start, len)	checkdate (month, day, year) date (format, timestamp) getdate (timestamp) mktime (hr, min, sec, month, day, yr) strftime (formatstring, timestamp) strtotime (str) time ()	There's an excellent regular expression tester at: <a href="http://regexpal.com/">http://regexpal.com/</a>
	<b>PHP Regular Expressions Functions</b>	<b>Pattern Modifiers</b>
	ereg (pattern, str) split (pattern, str) ereg_replace (pattern, replace, str) preg_grep (pattern, arr) preg_match (pattern, str) preg_match_all (pattern, str, arr) preg_replace (pattern, replace, str) preg_split (pattern, str)	g Global match i * Case-insensitive m * Multiple lines s * Treat string as single line x * Allow comments and whitespace in pattern

---



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

Published 19th October, 2011.  
Last updated 29th February, 2020.  
Page 1 of 2.

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

### 13.3 Hoja de Referencia PHP

## 12.2.4 MySQL

<b>Cheatography</b>		MySQL Cheat Sheet	
		by Dave Child (DaveChild) via cheatography.com/1/cs/16/	
<b>MySQL Data Types</b>		<b>MySQL Type Conversion</b>	
CHAR	String (0 - 255)	BINARY 'string'	
VARCHAR	String (0 - 255)	CAST (expression AS datatype)	
TINYTEXT	String (0 - 255)	CONVERT (expression, datatype)	
TEXT	String (0 - 65535)		
BLOB	String (0 - 65535)		
MEDIUMTEXT	String (0 - 16777215)		
MEDIUMBLOB	String (0 - 16777215)		
LONGTEXT	String (0 - 4294967295)		
LONGBLOB	String (0 - 4294967295)		
TINYINT x	Integer (-128 to 127)		
SMALLINT x	Integer (-32768 to 32767)		
MEDIUMINT x	Integer (-8388608 to 8388607)		
INT x	Integer (-2147483648 to 2147483647)		
BIGINT x	Integer (-9223372036854-775808 to 922337203-6854775807)		
FLOAT	Decimal (precise to 23 digits)	AVG	MAX
DOUBLE	Decimal (24 to 53 digits)	BIT_AND	STD
DECIMAL	"DOUBLE" stored as string	BIT_OR	STDDEV
DATE	YYYY-MM-DD	COUNT	SUM
DATETIME	YYYY-MM-DD HH:MM:SS	GROUP_CONCAT	VARIANCE
TIMESTAMP	YYYYMMDDHHMMSS	MIN	
TIME	HH:MM:SS		
ENUM	One of preset options	<b>MySQL Mathematical Functions</b>	
SET	Selection of preset options	ABS	COS
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).		SIGN	SIN
		MOD	TAN
		FLOOR	ACOS
		CEILING	ASIN
		ROUND	ATAN, ATAN2
		DIV	COT
		EXP	RAND
		LN	LEAST
		LOG, LOG2, LOG10	GREATEST
		POW	DEGREES
		POWER	RADIANS
		SQRT	TRUNCATE
		PI	
		<b>MySQL String Functions</b>	
		ASCII	SUBSTRING
		ORD	MID
		CONV	SUBSTRING_INDEX
		BIN	LTRIM
		OCT	RTRIM
		HEX	TRIM
		CHAR	SOUNDEX
		CONCAT	SPACE
		CONCAT_WS	REPLACE
		<b>MySQL Control Flow Functions</b>	
		IF	NULLIF
		IFNULL	
		Published 19th October, 2011. Last updated 25th February, 2020. Page 1 of 2.	
		Sponsored by <a href="#">Readable.com</a> Measure your website readability! <a href="https://readable.com">https://readable.com</a>	
			
By <b>Dave Child</b> (DaveChild) <a href="http://cheatography.com/davechild/">cheatography.com/davechild/</a> <a href="http://aloneonahill.com">aloneonahill.com</a>			

## 13.4 Hoja de Referencia MySQL

## 12.2.5 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>
cd <i>project_dir</i>	git log	git remote -v
git init	<b>Short Format</b>	<b>Show information</b>
git add .	git log --pretty=short	git remote show <i>remote</i>
<b>From other repository</b>	<b>Patches</b>	<b>Add remote</b>
git clone <i>existing_dir new_dir</i>	git log -p	git remote add <i>path/url</i>
git clone <a href="https://github.com/user/repo.git">git://github.com/user/repo.git</a>	<b>Show file commits</b>	<b>Fetch changes</b>
git clone <a href="https://github.com/user/repo.git">https://github.com/user/repo.git</a>	git log <i>file</i>	git fetch <i>remote</i>
<b>Git - Local Changes</b>	<b>Show directory commits</b>	<b>Fetch + merge</b>
<b>Changed in working directory</b>	git log <i>dir/</i>	git pull <i>remote branch</i>
git status	<b>Stats</b>	<b>Publish local to remote</b>
<b>Tracked file changes</b>	git log --stat	git push <i>remote branch</i>
git diff	<b>Who changed file</b>	<b>Delete remote branch</b>
<b>Add changed files</b>	git blame <i>file</i>	git push <i>remote :branch</i>
git add <i>file1 file2 file3</i>	<b>Git - Merge/Rebase</b>	<b>Publish tags</b>
<b>Remove file</b>	<b>Merge branch into current</b>	git push --tags
git rm <i>file</i>	git merge <i>branch</i>	
git rmdir -r <i>(recursive under directory)</i>	<b>Rebase into branch</b>	
<b>See files ready for commit</b>	git rebase <i>branch</i>	
git diff --cached	git rebase <i>master branch</i>	
<b>Commit changes</b>	<b>Abort rebase</b>	
git commit	git rebase --abort	
git commit -m "My message"	<b>Merge tool to solve conflicts</b>	
git commit -a -m "My Message"	git mergetool	
<i>(tracked files only, auto add)</i>	<b>Conflicts against base file</b>	
<b>Change last commit</b>	git diff --base <i>file</i>	
git commit --amend	<b>Diff other users changes</b>	
<b>Revert changes to file</b>	git diff --theirs <i>file</i>	
git checkout -f <i>file</i>	<b>Diff your changes</b>	
<b>Revert changes (new commit)</b>	git diff --ours <i>file</i>	
git revert HEAD	<b>After resolving conflicts</b>	
<b>Return to last committed state</b>	git rebase --continue	
git reset --hard HEAD		
<b>C</b> 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 <a href="http://crosswordcheats.com">CrosswordCheats.com</a> Learn to solve cryptic crosswords! <a href="http://crosswordcheats.com">http://crosswordcheats.com</a>

## 13.5 Hoja de Referencia Git