



UNIVERSIDAD POLITÉCNICA DE SINALOA

PROGRAMA ACADÉMICO DE INGENIERÍA EN INFORMÁTICA

Tesina

“Bootstrap y Laravel, herramientas para el desarrollo de aplicaciones web”

Para obtener la acreditación de las estadías profesionales y contar con los créditos para el grado de Ingeniero en Informática.

Autor: Denzel Javier Ovando Ortega

Asesor: M.C. Gloria Irene Téllez Rodríguez

Asesor OR: Miguel Alonso Berrelleza Pérez

Mazatlán, Sinaloa 15 de diciembre del 2019



TATA CONSULTANCY SERVICES

Miércoles 18 de septiembre de 2019. Tlaquepaque, Jalisco

UNIVERSIDAD POLITÉCNICA DE SINALOA

DIRECCIÓN DE VINCULACIÓN, DIFUSIÓN Y EXT. UNIVERSITARIA.

Lic. Eduardo Crespo Campa,

Por medio de la presente me permito hacer constar que el alumno OVANDO ORTEGA DENZEL JAVIER (ID: 2016030030), de la carrera INGENIERIA EN INFORMÁTICA ha sido aceptado por nuestra empresa para la elaboración de sus estudios profesionales de último periodo, desarrollando actividades del ciclo de desarrollo de software para nuestros proyectos con clientes globales. Con un total de 600 horas durante el periodo comprendido desde el 2 de septiembre de 2019 hasta el 6 de diciembre de 2019.

Para los fines que al interesado convengan se extiende la presente, quedando a su disposición para atender las dudas que pueda tener a este respecto.

Saludos cordiales,



Mtro. Miguel Alonso Borrellaza Pérez
Academic Interface Programme Manager
TCS México



TATA CONSULTANCY SERVICES

6 de diciembre de 2019. Tlaquepaque, Jalisco

UNIVERSIDAD POLITÉCNICA DE SINALOA
DIRECCIÓN DE VINCULACIÓN, DIFUSIÓN Y EXT. UNIVERSITARIA.

Lic. Eduardo Crespo Campa,

Por medio de la presente me permito hacer constar que el alumno OVANDO ORTEGA DENZEL JAVIER (ID 2016030030), de la carrera INGENIERIA EN INFORMÁTICA ha concluido de manera satisfactoria su periodo de estudios profesionales de último periodo, desarrollando actividades del ciclo de desarrollo de software para nuestros proyectos con clientes globales. Con un total de 600 horas durante el periodo comprendido desde el 2 de septiembre de 2019 hasta el 6 de diciembre de 2019.

Para los fines que al interesado convengan se extiende la presente, quedando a su disposición para atender las dudas que pueda tener a este respecto.

Saludos cordiales



Mtro. Miguel Alonso Berrelteza Pérez
Academic Interface Programme Manager
TCS México



UNIVERSIDAD POLITÉCNICA DE SINALOA



C. OVANDO ORTEGA DENZEL JAVIER
Presente.-

Folio 2016030030-2019-14

Por medio de la presente me permito comunicarle que es de aceptarse el tema de tesina, el cuál se ha solicitado bajo el título:

"Bootstrap y Laravel, herramientas para el desarrollo de aplicaciones web"

mismo que usted desarrollará con objeto de dar lugar a los trámites conducentes para la acreditación de la asignatura de Estudios Profesionales de la Unidad Académica de:

Ingeniería en Informática

Así mismo. Le comunico que para el desarrollo de la citada tesina le ha sido asignado como director de la misma a: **Gloria Irene Tellez Rodriguez**, y como asesores a Luis Javier Mena Camare, **Ismaylia Saucedo Ugatde**, y **Alejandro Perez Pasten Borja**.

Sin otro particular por el momento, aprovecho la ocasión para enviarle un cordial saludo.

Atentamente
Dr. Rodolfo Ostos Robles

Director del Programa Académico de Ingeniería en Informática
INFORMATICA
Universidad Politécnica de Sinaloa



"2019, Año del Caudillo del Sur, Emiliano Zapata".

Carretera Municipal Libre Mazatlán Higueras Km. 3, Col. Genaro Estrada. C.P. 82199, Mazatlán, Sin. Tel (669) 1800695 y 96
www.upsin.edu.mx





UNIVERSIDAD POLITÉCNICA DE SINALOA



C. OVANDO ORTEGA DENZEL JAVIER
Presente.-

Folio 2016030030-2019-14

Por este conducto le envío un cordial saludo y aprovecho la ocasión para notificarle que el jurado que le fue asignado para evaluar la tesina desarrollada en las estadías profesionales denominada "Bootstrap y Laravel, herramientas para el desarrollo de aplicaciones web" y que después de ser revisada en reunión de sinodales, ante la Dirección de la Unidad Académica de Ingeniería en Informática, integrada por:

PRESIDENTE DEL JURADO: Gloria Irene Tellez Rodriguez

SINODAL: Luis Javier Mena Camare

SINODAL: Ismaylia Saucedo Ugalde

SINODAL: Alejandro Perez Pasten Borja

Ha decidido autorizar y aceptar la digitalización de la misma por el participante, conforme a la normatividad vigente y cumpliendo con los requisitos para tal caso.

Agradeciendo la atención a la presente, le reitero a Usted mi atenta consideración y respeto.

Dr. Rodolfo Ostos Robles

DIRECTOR DEL PROGRAMA ACADÉMICO DE INGENIERÍA EN INFORMÁTICA

Universidad Politécnica de Sinaloa



"2019, Año del Caudillo del Sur, Emiliano Zapata".



Carretera Municipal Libre Mazatlán Higueras Km. 3, Col. Genaro Estrada, C.P. 82199, Mazatlán, Sinaloa. Tel (669) 1800695 y 96
www.upsin.edu.mx

Agradecimientos

Mi tesina la dedico con todo mi cariño a todas las personas que han estado apoyándome con mi carrera desde que inicié y que fueron mi motivación para continuar con lo que ahora estoy terminando, mi carrera.

A mis amigos por haberme ayudado y motivado a seguir en mi carrera y ser parte de mi crecimiento como profesionista.

Gracias a todos.

Dedicatoria

Dedico este proyecto a:

Mis padres por mostrarme enseñarme que es valioso en la vida y por mostrarme el camino que es necesario para la superación.

Mis hermanos por brindarme su tiempo, su apoyo y su consejo.

Mis amigos por permitirme estar a su lado y por aconsejarme en todo lo bueno y lo malo que vivimos.

A mis profesores, que fueron y serán siempre mis segundos padres, que me enseñaron y aconsejaron en el camino profesional.

Índice

Agradecimientos	3
Dedicatoria	4
Resultados y discusiones	55
Conclusiones	56
Referencias bibliográficas	57
Glosario	59

Capítulo 1: Introducción

1.1 Antecedentes.....	9
1.2 Localización	9
1.3 Objetivos y prioridades de la empresa	10
1.4 Organigrama de la empresa	10
1.5 Visión	11
1.6 Misión	11
1.7 Planteamiento del problema	11
1.8 Propuesta de investigación	12
1.9 Objetivos	12
1.9.1 Objetivo general.....	12
1.9.2 Preguntas de investigación	13
1.10 Hipótesis	14
1.12 Relevancia	14

Capítulo 2: Marco teórico y referencial

Introducción	15
--------------------	----

2.1 Análisis de investigaciones relacionadas	15
2.2 ¿Qué es una Framework?	15
2.3 ¿Qué es una aplicación web?	15
2.4 ¿Qué es Laravel?	16
2.5 Manejo de MVC en Laravel	17
2.6 Manejo de datos en Laravel	18
2.7 Manejo de Modelos en Laravel	18
2.8 Manejo de Vistas en Laravel	19
2.9 Manejo de Layout con Blade en Laravel	20
2.10 ¿Qué es Bootstrap?	20
2.11 Características de Bootstrap	20
2.12 ¿Cómo se creó Boostrap?	21
2.13 Estructura	21
2.14 Comunidad de Boostrap	23
2.15 Laravel introducción básica	23
2.16 Instalación de Laravel	23
2.17 Instalación de Composer	24
2.18 Instalación de Laravel mediante composer	24
2.19 Estructura del proyecto Laravel	24
2.20 Funcionamiento básico	25
2.21 Rutas en Laravel	28
2.22 Rutas básicas en Laravel	28
2.23 Artisan en Laravel	28

2.24 Generación de código	28
2.25 Vistas	28
2.26 Platillas Blade	28
2.27 Bootstrap introducción básica	29
2.28 Inicio de Bootstrap	29
2.29 ¿Bootstrap es un framework?	29

Capítulo 3: Diseño y desarrollo

3.1 Diseño	40
3.2 Desarrollo	40
3.3 Herramientas para el desarrollo de Boostrap y laravel	41
3.4 Instalación de Composer	41
3.5 Productividad del ejemplo	50

Índice de imágenes y gráficas

Imagen 1.1 Google Maps	8
Imagen 1.2 Organigrama de Tata	10
Imagen 2.1 Vista de datos asociados	28
Imagen 2.2 Ejemplo Estructura básica 4	33
Imagen 2.3 Ejemplo Menú 5	34
Imagen 2.4 Ejemplo contenido 6	34
Imagen 2.5 Ejemplo layout 7	35
Imagen 2.6 Ejemplo componentes 8	36
Imagen 2.7 Ejemplo container 9	36
Imagen 2.8 Ejemplo navbar 10	36

Imagen 2.9 Ejemplo Responsive design 11	37
Imagen 2.10 Ejemplo meta 12	37
Imagen 2.11 Ejemplo El diseño responsivo 13	38
Imagen 3.1 Instalación de composer para Windows	43
Imagen 3.2 Pestaña de Composer Setup	44
Imagen 3.3 Selección de componentes Composer	45
Imagen 3.4 Selección de ruta de PHP	46
Imagen 3.4 Pasos de instalación, siguiente	47
Imagen 3.6 Inicializar composer	47
Imagen 3.7 Estructura de una clase en el modelo	46
Imagen 3.8 Estructura de una clase en la vista	46
Imagen 3.9 Estructura de la plantilla master.blade.php	46
Imagen 3.10 Estructura de una clase en la vista	47
Imagen 3.11 Estructura de la plantilla master.blade.php	48
Imagen 3.12 Estructura de una clase en el controlador	49
Imagen 3.13 Archivo /app/routes.php	50
Imagen 3.14 Funcionamiento de Laravel desde MVC	50
Imagen 3.15 Cookie inicio de Sesión	51
Imagen 3.16 Prueba de Stress	52
Grafica 3.1 Tiempo de respuesta en peticiones en framework, Laravel	54

CAPÍTULO 1: INTRODUCCIÓN

1.1 Antecedentes

El Grupo Tata es un conglomerado industrial indio fundado por Jamsetji Tata en 1868. En India, Tata es uno de los grupos económicos más respetados y antiguos. Su presencia se extiende a África, América, Asia y Europa.

El Grupo Tata está compuesto por 98 empresas que operan en siete sectores: Comunicaciones y sistemas de información, Ingeniería, Materiales, Servicios, Energía, Bienes de consumo y Químicos. El grupo factura unos 62 500 millones de dólares al año,¹ equivalentes al 5 % del producto interno bruto (PIB) de la India.

Entre sus empresas se cuentan: TCS (Tata Consultancy Services), la mayor empresa de software y servicios de Asia, Tata Tea, la mayor productora de té del mundo, Tata Steel, el segundo mayor productor de acero de la India, Tata Motors (antiguamente Tata Engineering and Locomotives Company Ltd (TELCO)), vehículos comerciales (el más grande en la India) y automóviles, Indian Hotels, división de hotelería y turismo del grupo.

1.2 Localización

Parque Industrial Edificio 3 Camino al Iteso 8900 Colonia, Pinar de la Calma, 45080 San Pedro Tlaquepaque, Jal.

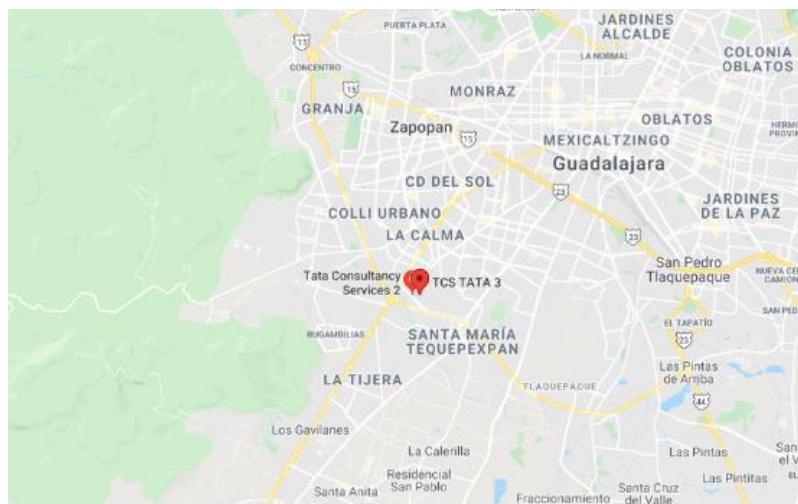


Imagen 1.1 Google Maps

1.3 Objetivos y prioridades de la empresa

Tata siempre ha sido una organización basada en valores. Estos valores continúan dirigiendo el crecimiento y el negocio de las empresas Tata.

Los cinco valores principales de Tata que sustentan la forma en que hacemos negocios son:

1. Integridad: Seremos justos, honestos, transparentes y éticos en nuestra conducta; todo lo que hacemos debe resistir la prueba del escrutinio público.
2. Responsabilidad: Integraremos principios ambientales y sociales en nuestros negocios, asegurando que lo que proviene de las personas vuelva a las personas muchas veces.
3. Excelencia: Nos apasionará alcanzar los más altos estándares de calidad, siempre promoviendo la meritocracia.
4. Pionero: Seremos audaces y ágiles, asumiendo valientemente los desafíos, utilizando una visión profunda del cliente para desarrollar soluciones innovadoras.
5. Unidad: Invertiremos en nuestra gente y socios, permitiéndonos un aprendizaje continuo y construiremos relaciones afectivas y colaborativas basadas en la confianza y el respeto mutuo.

1.4 Organigrama de la empresa

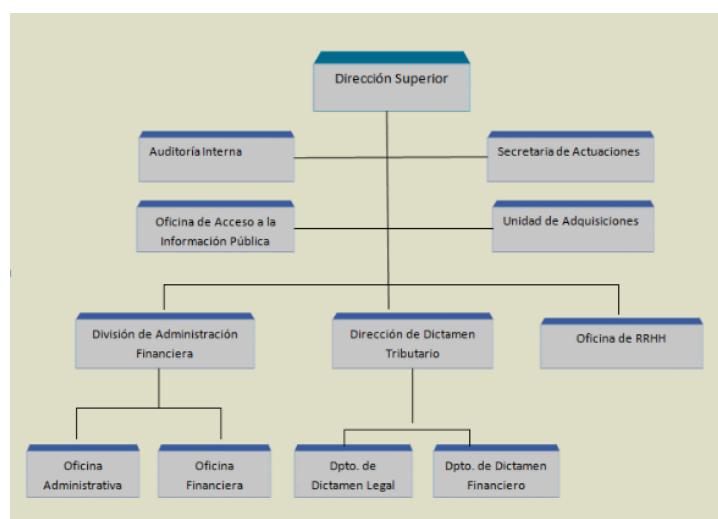


Imagen 1.2 Organigrama de Tata

1.5 Visión

El Tata Business Excellence Group (TBExG) promueve la excelencia empresarial en las empresas del grupo al incorporar los mejores procesos de su clase y facilitar el intercambio de las mejores prácticas. El TBExG facilita las evaluaciones TBEM en las empresas del grupo para evaluar los niveles de madurez de sus procesos y el rendimiento general.

Estas evaluaciones son realizadas por asesores capacitados y certificados que tienen en cuenta el contexto de la empresa al aplicar los criterios holísticos de TBEM dentro de la empresa y la industria.

1.6 Misión

Nuestra filosofía de gobernanza es garantizar una gestión justa, transparente, responsable y ética para proteger los intereses de todas las partes interesadas, incluidos los accionistas, empleados, clientes, proveedores, reguladores y la sociedad.

Como ciudadano corporativo responsable, Tata Sons sigue las leyes de la tierra en letra y espíritu. Tata Sons también va más allá del mero cumplimiento para destacar ciertos comportamientos y normas para las compañías operativas del grupo Tata.

1.7 Planteamiento del problema

Actualmente existen variedad de herramientas de desarrollo que permiten realizar proyectos de manera más rápida y sencilla, un punto central que encapsulan funcionalidades de bajo y alto nivel que tienen la capacidad de optimizar procesos, ya que estos usan los recursos de un lenguaje de programación, para hacer un rápido desarrollo, a esto se le llama Frameworks.

A pesar de las mejoras y el avance de las tecnologías, siempre existe la incertidumbre de cuáles son las mejores para la realización de nuevos proyectos.

No todos los Frameworks del desarrollo de lado del usuario que permiten la creación de elementos, diseño de estética y funcionalidad mejor llamado lenguajes front-end

siguen siendo actualizados, por lo cual se pueden encontrar algunos que solamente tuvieron mantenimiento por sus primeros años y después quedaron en el olvido.

También puede haber Frameworks que ayudan a la funcionalidad del procesamiento de datos del lado del servidor o mejor llamado back-end, estos mismos son aquellos que un usuario jamás podrá modificar, puesto que son partes esenciales, un núcleo que es vital para el correcto funcionamiento de funciones y procesos.

De los mejores que ahora se encuentran en el mercado, son Laravel y Bootstrap, Frameworks para el rápido desarrollo de aplicaciones webs.

1.8 Propuesta de investigación

La importancia de esta investigación es dar a conocer el uso de Frameworks que existen actualmente en el mercado, pero enfocando la investigación en Laravel y Boostrap que son herramientas de desarrollo para la rápida creación de aplicaciones web.

Existen desarrolladores que no tienen un amplio conocimiento de estos poderosos Frameworks, con esta investigación se muestran otros caminos para determinar que herramientas son convenientes para desarrollar aplicaciones web.

1.9 Objetivos

1.9.1 Objetivo general

- Clarificar la idea sobre Laravel y Boostrap.
- Mostrar la potencia de Laravel con ejemplos claros.
- Mostrar que aplicaciones actualmente usan Laravel, Boostrap y cómo las ha beneficiado.
- Mostrar el avance desde su lanzamiento hasta ahora.
- Comparar la velocidad de procesamiento de la información de Laravel sobre otros frameworks.

1.9.2 Preguntas de investigación

Antes y durante investigación mencionada se consideran las preguntas siguientes:

1. ¿Cuál es la mayor ventaja de la implementación del uso de bootstrap y laravel sobre las demás herramientas de software?
2. ¿Qué otra herramienta de software se le asemeja a las mencionadas en la pregunta anterior?
3. ¿Es fácil la utilización de una herramienta bootstrap y laravel para la restructuración de un software antiguo?
4. ¿Cuál es el porcentaje (aproximado) de rendimiento en líneas de código de la implementación del uso de bootstrap?

1.10 Hipótesis

Laravel llegó para demostrar, tanto a viejas por nuevas generaciones de desarrolladores, que las buenas prácticas, y por sobre todo los patrones de diseño, no son monstruos inalcanzables e inentendibles, y coloca en evidencia las ventajas y rendimientos que se logran cuando se coloca algo más que sólo código en una aplicación.

Con Bootstrap el beneficio de usar *responsive design* en un sitio web, es principalmente que el sitio web se adapta automáticamente al dispositivo desde donde se acceda. Lo que se usa con más frecuencia, y que a mi opinión personal me gusta más, es el uso de media queries, el cual es un módulo de CSS3 que permite la representación de contenido para adaptarse a condiciones como la resolución de la pantalla y si se trabajan las dimensiones del contenido en porcentajes, se puede tener una web muy fluida capaz de adaptarse a casi cualquier tamaño de forma automática.

1.11 Limitaciones y supuestos

A continuación, se extienden puntos a favor y en contra acerca de la estabilidad de Laravel.

Ventajas:

- Utiliza el motor de plantillas *Blade* para acelerar las tareas de compilación y los usuarios pueden incluir nuevas características de una manera sencilla.
- El "*Bundled modularity*" permite que el código sea reutilizado sin dificultades.
- ORM, técnica fácil de entender. Es por eso que, la creación de relaciones de las bases de datos es tan simple.
- Un CLI Artisan que comprende herramientas avanzadas para hacer tareas y migraciones.
- Documentación espléndida
- Característica adicional para el enrutamiento inverso.

Desventajas:

- Los desarrolladores avanzados suelen liarse extendiendo códigos y clases.
- No cuenta con una gran comunidad detrás, en comparación con otras plataformas.
- Muchos métodos incluidos en el enrutamiento inverso son complejos.
- Pero en general todo depende de cómo se administra el código, Laravel ofrece muchas posibilidades hablando de la implementación y herencias de clases.

1.12 Relevancia

La importancia de esta investigación es dar a conocer el uso de *Frameworks* que existen actualmente en el mercado, enfocando la investigación en Laravel y Bootstrap que son herramientas de desarrollo para la rápida creación de aplicaciones web.

Existen desarrolladores que no tienen un amplio conocimiento de estos poderosos *Frameworks*, con esta investigación se muestran otros caminos para determinar que herramientas son convenientes para desarrollar aplicaciones web.

CAPÍTULO 2. MARCO REFERENCIAL Y MARCO TEÓRICO

Introducción

Laravel ha sido un framework muy útil en los últimos años, cuenta ya con la versión 5.1 en la cual implementa nuevas características en base a la versión de PHP, pero con ello, también incorpora nuevas herramientas de rápido desarrollo llamadas Vendor como son Cake, Boostrap y otras que ayudan a la realización de aplicaciones webs.

2.1 Análisis de investigaciones relacionadas

2.2 ¿Qué es una Framework?

Un framework es un conjunto de código que puedes utilizar para hacer un sistema personalizado muy simple o muy complejo.

2.3 ¿Qué es una aplicación web?

Antes de continuar es necesario aclarar que es una aplicación web:

Las aplicaciones web reciben este nombre porque se ejecutan en internet. Es decir que los datos o los archivos en los que trabaja son procesados y almacenados dentro de la web. Estas aplicaciones, por lo general, no necesitan ser instaladas en el computador [1].

El concepto de aplicaciones web está relacionado con el almacenamiento en la nube. Toda la información se guarda de forma permanente en grandes servidores de internet y se envían a los dispositivos o equipos los datos requeridos en ese momento, quedando una copia temporal dentro del equipo.

Una vez aclarado el concepto de aplicaciones web, a continuación, se mencionan algunas de las aplicaciones más reconocidas.

- **Correo electrónico:** Servicios como Gmail y Yahoo! se ejecutan en tu buscador y realizan las mismas tareas de programas de correo electrónico como Microsoft Outlook.

Después de hacer el ingreso a un servicio de correo electrónico online, se puede utilizarlo inmediatamente, sin necesidad de instalar algún programa en el equipo. Los correos permanecerán almacenados en los servidores de tu proveedor del servicio [2].

- **Google Docs:** Es un conjunto de programas ofimáticos. Se ejecutan desde el navegador y funcionan de igual manera que los programas que hacen parte del paquete de Microsoft Office, ya que son utilizados para crear documentos, hojas de cálculo, presentaciones y más.

Los documentos se almacenan en la nube, lo que facilita compartir y editar los archivos simultáneamente con otras personas y puede hacerse desde cualquier dispositivo, en cualquier lugar a cualquier hora. Solo se debe tener una conexión a internet [3].

2.4 ¿Qué es Laravel?

Laravel es un nuevo y poderoso Framework PHP desarrollado por Taylor Otwell, que promete llevar al lenguaje PHP a un nuevo nivel [4].

Laravel, propone una forma de desarrollar aplicaciones web de un modo mucho más ágil. Por ejemplo, en Laravel opcionalmente se puede usar el patrón de diseño MVC (Modelo-Vista-Controlador) tradicional, donde al igual que otros frameworks PHP, el controlador es programado como una clase [5].

Por lo tanto, un Controlador es una clase PHP que dispone de métodos públicos que son el punto de entrada final de una petición HTTP (Request PHP) de la aplicación. Pero, Lavarel propone además una forma distinta y más directa de responder a la solicitud HTTP, que se verá a continuación

2.5 Manejo de MVC en Laravel

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones [6]. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Modelo vista controlador del patrón de arquitectura del software

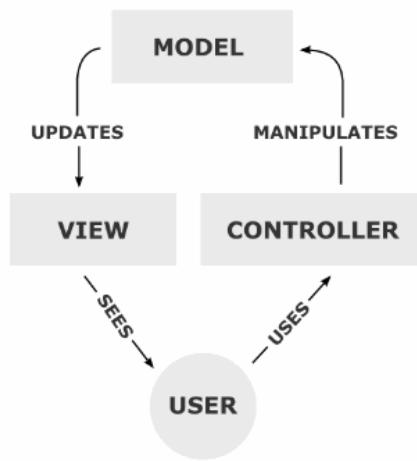


Imagen 2. MVC 1

De manera genérica, los componentes de MVC se podrían definir como sigue:

- El Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tantas consultas como actualizaciones. Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- El Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud de información (por ejemplo, editar un documento o un registro en una base de datos). Por

tanto, se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

- La Vista: Presenta el 'modelo' y los datos preparados por el controlador al usuario de forma visual. El usuario podrá interactuar con la vista y realizar otras peticiones que se enviarán al controlador.

Como en este ejemplo que se está declarando un modelo Libro que extiende todos los métodos de Eloquent, este mismo es el que será utilizado en el controlador para la interacción con la tabla definida en la variable \$table que es la que tomará el nombre de la tabla de la base de datos que se utilizará, como se muestra a continuación:

```
class Libro extends Eloquent {  
  
    public static $table = 'tb_libros';
```

Laravel “entrega la opción” de seguir usando la metodología tradicional MVC. Sin embargo, el framework propone una vía más rápida en PHP, la cual consiste en programar la interacción HTTP directamente como una función anónima asociada a una ruta, esto tiene la ventaja de reducir la cantidad de código, especialmente cuando sólo necesitamos incluir una funcionalidad, Natile, A. (2016).

2.6 Manejo de datos en Laravel

Laravel incluye una valiosa pieza de software, llamada Eloquent ORM. Este ORM se funda en patrón active record y su funcionamiento es en extremo sencillo.

Un ORM (Object Relational Mapper) en PHP es un software que permite tratar la capa de persistencia de los datos, como simples accesos a métodos de una Clase u Objeto en PHP.

La funcionalidad interna del ORM es mapear los objetos de PHP a las tablas en la base de datos, para el caso en que la persistencia de los datos de la aplicación es proporcionada por una DB [7].

2.7 Manejo de Modelos en Laravel

Para hacer la interacción con el modelo es necesario declarar una ruta la cual

```
Route::get('libro/listar', function() { // Funcionalidad });
```

permite la interacción con el mismo, de esta manera declaramos una ruta, NNutile, A. (2016).

2.8 Manejo de Vistas en Laravel

El manejo de las vistas es un punto demasiado importante si hablar de realizar una aplicación web, Laravel incluye de paquete un sistema de procesamiento de plantillas llamado Blade. Este sistema de plantillas, Blade favorece un código mucho más limpio en las Vistas, además de incluir un sistema de Caché que lo hace más rápido [8]. El sistema Blade de Laravel, permite usar una sintaxis más reducida en su escritura. Por ejemplo, en vez de pintar la vista usando el código PHP:

Código PHP

```
<!DOCTYPE html>  
  
<html lang="es">  
  
    <head>  
  
        <meta charset="UTF-8">  
  
        <title>@yield('titulo')</title>  
  
    </head>  
  
    <body>  
  
    </body>  
  
</html>
```

Lo cual, no es una gran ventaja sobre todo cuando siempre es posible usar una expresión resumida en PHP. No obstante, lo que, sí es una gran ventaja, es el modo en que Blade maneja los Layouts [9].

Los Layouts en los frameworks PHP, permiten organizar las vistas en PHP. En especial, todos los elementos estáticos en una aplicación web que no cambian entre peticiones HTTP, como lo son:

- ✓ Menús
- ✓ Header
- ✓ Footers.

Generalmente, a estos elementos se les denomina 'partial views' (vistas parciales).

2.9 Manejo de Layout con Blade en Laravel

Según la documentación de Laravel un Layout es:

El motor de creación de plantillas proporcionado con Laravel es simple pero potente. A diferencia de otros populares motores de plantillas de PHP, Blade no le impide usar código PHP simple en sus vistas.

Es así, que en todas las vistas de Blade se compilan en código PHP simple y se almacenan en la memoria caché hasta que se modifiquen, lo que significa que Blade agrega esencialmente una sobrecarga a su aplicación. Los archivos Blade View usan la extensión.blade.php y típicamente se almacenan en el directorio resources/views, NNutile, A. (2016).

En el render de una vista completa en Lavarel se usan dos archivos:

- ✓ Un sólo archivo, la Vista, con los elementos (partial views).
- ✓ Un código típico sería para una Layout HTML5 de Blade.

2.10 ¿Qué es Bootstrap?

Bootstrap fue desarrollado por Twitter, tiene como gran objetivo facilitar el diseño web sirviendo de guía para emplear buenas prácticas y estándares.

2.11 Características de Bootstrap

- ✓ Permite crear de manera fácil sitios web que son adaptables o responsivas para ajustarse a cualquier dispositivo y tamaños de pantalla.
- ✓ Consiguiendo un sitio web muy organizado visualmente.
- ✓ Posee un sistema de grillas en la cual maquetamos el sitio mediante columnas que lo hace bastante fácil.
- ✓ Se integra perfectamente con librerías de Javascript.
- ✓ Posee una comunidad muy grande y activa que lo respalda.
- ✓ Permite la utilización de Sass y Less para hacer aún más agradable la experiencia de desarrollo.

2.12 ¿Cómo se creó Boostrap?

Bootstrap fue creado por Mark Otto y Jacob Thornton ambos desarrolladores de Twitter, ellos buscaban un marco de trabajo para fomentar la consistencia entre las herramientas internas, las cuales llevaban a inconsistencias y a una gran carga de trabajo en su mantenimiento [10].

El primer desarrollo en condiciones reales ocurrió durante la primera “Semana de Hackeo” (Hackweek) de Twitter. Mark Otto mostró a algunos colegas como acelerar el desarrollo de sus proyectos con la ayuda de la herramienta de trabajo. Como resultado, decenas de temas se han introducido en el marco de trabajo. En agosto del 2011, Twitter liberó a Bootstrap como código abierto. En febrero del 2012, se convirtió en el proyecto de desarrollo más popular de GitHub.

2.13 Estructura

Bootstrap viene con una disposición de cuadrilla estándar de 940 píxeles de ancho. Alternativamente, se usan para un diseño de ancho-variable. Para ambos casos, la herramienta tiene cuatro variaciones para hacer uso de distintas resoluciones y tipos de dispositivos: teléfonos móviles, formato de retrato y paisaje, tabletas y computadoras con baja y alta resolución (pantalla ancha). Esto ajusta el ancho de las columnas automáticamente.

Hojas de estilo:

Bootstrap proporciona un conjunto de hojas de estilo que proveen definiciones básicas de estilo para todos los componentes de HTML. Esto otorga una uniformidad al navegador y al sistema de anchura, da una apariencia moderna para el formateo de los elementos de texto, tablas y formularios.

Componentes reusables:

Bootstrap contiene otra interfaz de elementos que comúnmente se usan. Incluye botones con características avanzadas como grupo de botones o botones con opción de menú desplegable, listas de navegación, etiquetas horizontales y verticales, ruta de navegación, paginación, capacidades avanzadas de miniaturas tipográficas, formatos para mensajes de alerta y barras de progreso.

Pugins de JavaScript:

Los componentes de JavaScript para Bootstrap están basados en la librería jQuery de JavaScript. Los plugins se encuentran en la herramienta de plugin de jQuery. Proveen elementos adicionales de interfaz de usuario como diálogos, tooltips y carruseles.

Tipo de licencia:

Siempre es importante conocer el tipo de licencia que tiene el framework a utilizar como parte de nuestra ética y profesionalismo. Bootstrap está publicado bajo la licencia “Apache 2 License” y está protegido por el siguiente copyright: “copyright 2013 Twitter” [11].

Entre los puntos clave que se tienen que conocer sobre la licencia de Boostrap, se mencionan las siguientes preguntas con sus respectivas repuestas:

¿Qué permite la licencia?

1. Descargar y usar gratuitamente Bootstrap entero o alguna de sus partes, tanto para uso personal, empresarial y/o comercial.

2. Utilizar e incluir Bootstrap en cualquier paquete o distribución de software que crees.

¿Qué prohíbe la licencia?

1. Redistribuir cualquier parte de Bootstrap sin mencionar quiénes son sus autores.
2. Utilizar cualquier elemento propiedad de la empresa Twitter de forma que parezca que Twitter apoya o tiene alguna relación con tu proyecto.
3. Utilizar cualquier elemento propiedad de la empresa Twitter de forma que parezca que se ha creado ese elemento de Twitter.

¿Qué te obliga la licencia?

1. Incluir una copia de la licencia de Bootstrap en cualquier distribución de software creada e incluya Bootstrap.
2. Indicar de forma clara quiénes son los autores de Bootstrap.

¿Qué no te obliga la licencia?

1. Incluir en el software el código fuente de Bootstrap o de cualquier variación de Bootstrap que se haya creado.
2. Hacer públicos los cambios y mejoras que se haya hecho sobre Bootstrap, aunque sus autores estarán encantados si se realiza.

2.14 Comunidad de Boostrap

Bootstrap posee una comunidad muy amplia que constantemente está aportando nuevas ideas y soluciones a problemas.

2.15 Laravel introducción básica

Laravel ha sido y sigue siendo unos de los frameworks con más funcionalidades para el desarrollo de aplicaciones webs, para ello aquí se mostrarán, los puntos principales que debemos conocer si queremos trabajar con Bootstrap.

2.16 Instalación de Laravel

Para la utilización de Laravel en primer lugar se necesita tener instalado un servidor Web con PHP mayor o igual a 5.5.9, MySQL y la extensión MCrypt de PHP. Una vez instalado el servidor se procede a instalar la utilidad Composer y por último la librería de Laravel [12].

2.17 Instalación de Composer

Composer es un gestor de dependencias para PHP. Esto quiere decir que permite descargar de sus repositorios todas las librerías y las dependencias con las versiones requeridas que el proyecto necesite.

2.18 Instalación de Laravel mediante composer

En la carpeta raíz del servidor web (/Applications/XAMPP/htdocs) o en otra ubicación, se ejecuta el siguiente comando:

```
$ composer create-project laravel/laravel [ project ] --prefer-dist
```

Se puede cambiar [project] por el nombre del proyecto a crear.

2.19 Estructura del proyecto Laravel

Al crear un nuevo proyecto de Laravel, se generará una estructura de carpetas y ficheros para organizar el código. Es importante que se conozca para qué funciona cada carpeta y donde tenemos que colocar nuestro código.

App: Contiene el código principal de la aplicación. Esta carpeta a su vez está dividida en muchas subcarpetas que se analizaran en la siguiente sección.

Config: Aquí se encuentran todos los archivos de configuración de la aplicación; base datos, cache, correos, sesiones o cualquier otra configuración general de la aplicación.

Database: En esta carpeta se incluye todo lo relacionado con la definición de la base de datos del nuevo proyecto. Dentro de ella podemos encontrar a su vez tres carpetas: factores, migrations y seeds.

Public: Es la única carpeta pública, la única que debería ser visible en nuestro servidor web. Todas las peticiones y solicitudes a la aplicación pasan por esta carpeta, ya que en ella se encuentra el index.php, este archivo es el que inicia todo el proceso de ejecución del framework. En este directorio también se alojan los archivos CSS, Javascript, imágenes y otros archivos que se quieran hacer públicos.

Resources: Esta carpeta contiene a su vez tres carpetas: assets, views y lang:

Resources/views: Este directorio contiene las vistas de la aplicación. En general serán plantillas de HTML que usan los controladores para mostrar la información. Hay que tener en cuenta que en esta carpeta no se almacenan los Javascript, CSS o imágenes, ese tipo de archivos se tienen que guardar en la carpeta public.

Resources/lang: En esta carpeta se guardan archivos PHP que contienen arrays con los textos del sitio web en diferentes lenguajes, solo será necesario utilizarla en caso que se desee que la aplicación se pueda traducir.

Resources/assets: Se utiliza para almacenar las fuentes de los assets tipo less o sass que se tendrían que compilar para generar las hojas de estilo públicas. No es necesario usar esta carpeta ya que se puede escribir directamente las hojas de estilo dentro de la carpeta public.

Bootstrap: En esta carpeta se incluye el código que se carga para procesar cada una de las llamadas al proyecto. Normalmente no es necesario que se modifique nada de esta carpeta.

Storage: En esta carpeta Laravel almacena toda la información interna necesarios para la ejecución de la web, como son los archivos de sesión, la caché, la compilación de las vistas, meta información y los logs del sistema. Normalmente tampoco se tiene que tocar nada dentro de esta carpeta, únicamente se suele acceder a ella para consultar los logs.

Tests: Esta carpeta se utiliza para los ficheros con las pruebas automatizadas. Laravel incluye un sistema que facilita todo el proceso de pruebas con PHPUnit.

Vendor: En esta carpeta se alojan todas las librerías y dependencias que conforman el framework de Laravel. Esta carpeta tampoco se tiene que modificar, ya que todo el código que contiene son librerías que se instalan y actualizan mediante la herramienta Composer.

Además, en la carpeta raíz también se puede encontrar dos ficheros muy importantes y que también son utilizados:

.env: Este fichero ya se ha mencionado en la sección de instalación, se utiliza para almacenar los valores de configuración que son propios de la máquina o instalación actual. Lo que permite cambiar fácilmente la configuración según la máquina en la que se instale y tener opciones distintas para producción, para distintos desarrolladores, etc. Importante, este fichero debería estar en el .gitignore.

Composer.json: Este fichero es el utilizado por Composer para realizar la instalación de Laravel. En una instalación inicial únicamente se especificará la instalación de un paquete, el propio framework de Laravel, pero se puede especificar la instalación de otras librerías o paquetes externos que añadan funcionalidad a Laravel.

2.20 Funcionamiento básico

El funcionamiento básico que sigue Laravel tras una petición web a una URL del sitio es el siguiente:

Todas las peticiones entran a través del fichero public/index.php, el cual en primer lugar comprobará en el fichero de rutas (app/Http/routes.php) si la URL es válida y en caso de serlo a qué controlador tiene que hacer la petición.

A continuación, se llamará al método del controlador asignado para dicha ruta. Como se ha visto, el controlador es el punto de entrada de las peticiones del usuario, el cual, dependiendo de la petición:

3. Accederá a la base de datos (si fuese necesario) a través de los "modelos" para obtener datos (o para añadir, modificar o eliminar).
4. Tras obtener los datos necesarios los preparará para pasárselos a la vista.

5. En el tercer paso el controlador llamará a una vista con una serie de datos asociados, la cual se preparará para mostrarse correctamente a partir de los datos de entrada y por último se mostrará al usuario.

A continuación, se incluye un pequeño esquema de este funcionamiento:

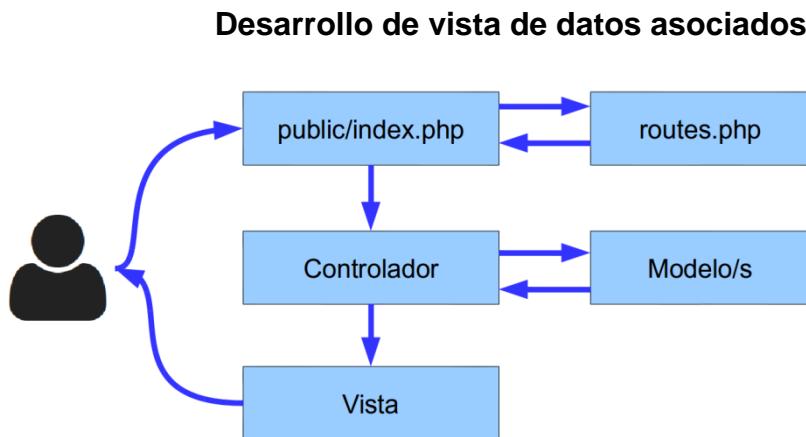


Imagen 2.1 Vista de datos asociados

2.21 Rutas en Laravel

Las rutas de la aplicación se tienen que definir en el fichero routes/web.php. Este es el punto centralizado para la definición de rutas y cualquier ruta no definida en este fichero no será válida, generando una excepción (lo que devolverá un error 404).

2.22 Rutas básicas en Laravel

Las rutas, además de definir la URL de la petición, también indican el método con el cual se ha de hacer dicha petición. Los dos métodos más utilizados y que se mencionaron son las peticiones tipo GET y tipo POST. Por ejemplo, para definir una petición tipo GET se tendría que añadir el siguiente código al fichero routes.php:

```
Route::get('/', function()
{
    return '¡Hola mundo!';
});
```

La documentación de Laravel cuenta con más tipos de rutas las cuales son necesarias para cada tipo de petición que se haga.

2.23 Artisan en Laravel

Laravel incluye un interfaz de línea de comandos (CLI, Command line interface) llamado Artisan. Esta utilidad permite realizar múltiples tareas necesarias durante el proceso de desarrollo o despliegue a producción de una aplicación, por lo que facilitará y acelerará el trabajo.

2.24 Generación de código

Una de las novedades de Laravel 5 es la generación de código gracias a Artisan. A través de la opción make se pueden generar diferentes componentes de Laravel (controladores, modelos, filtros, etc.) como si fueran plantillas, esto ahorrará mucho trabajo y se puede empezar a escribir directamente el contenido del componente. Por ejemplo, para crear un nuevo controlador se tiene que escribir: php artisan make:controller TaskController [13].

2.25 Vistas

Las vistas son la forma de presentar el resultado (una pantalla del sitio web) de forma visual al usuario, el cual podrá interactuar con él y volver a realizar una petición. Las vistas además permiten separar toda la parte de la presentación de resultados de la lógica (controladores) y de la base de datos (modelos). Por lo tanto no tendrán que realizar ningún tipo de consulta ni procesamiento de datos, simplemente recibirán datos y los prepararán para mostrarlos como HTML.

2.26 Platillas Blade

Laravel utiliza Blade para la definición de plantillas en las vistas. Esta librería permite realizar todo tipo de operaciones con los datos, además de la sustitución de secciones de las plantillas por otro contenido, herencia entre plantillas, definición de layouts o plantillas base, etc.

Los ficheros de vistas que utilizan el sistema de plantillas Blade tienen que tener la extensión .blade.php. Esta extensión tampoco se tendrá que incluir a la hora de referenciar una vista desde el fichero de rutas o desde un controlador. Es decir, se utilizará `view('home')` tanto si el fichero se llama `home.php` como `home.blade.php`.

En general el código que incluye Blade en una vista empezará por los símbolos @ o {{, el cual posteriormente será procesado y preparado para mostrarse por pantalla. Blade no añade sobrecarga de procesamiento, ya que todas las vistas son preprocesadas y cacheadas, por el contrario, nos brinda utilidades que ayudarán en el diseño y modularización de las vistas.

2.27 Bootstrap introducción básica

Bootstrap es un framework HTML, CSS y JS para crear interfaces de usuario web limpias y totalmente adaptables a todo tipo de dispositivos y tamaños de pantalla.

2.28 Inicio de Bootstrap

Lo desarrolló inicialmente Twitter en 2011, por eso también se conoce también como Twitter Bootstrap.

Aunque no se aprecie a simple vista, muchos sitios web de los que visitas están creados con Bootstrap.

2.29 ¿Bootstrap es un framework?

Un framework, es una colección de archivos HTML, CSS y JavaScript que contienen patrones de diseño con los que desarrollar un sitio web.

Esto evita perder tiempo reescribiendo código repetitivo una y otra vez. Además del ahorro de tiempo, realmente ayuda a escribir mejor código.

En Bootstrap estos archivos son las hojas de estilo CSS, las plantillas HTML (o snippets de código), los archivos JavaScript y la fuente de iconos Glyphicons.

2.30 ¿Qué no hace Bootstrap?

Bootstrap no es una aplicación para hacer páginas web, no se trata de un software con el que diseñar mediante una interface tipo Photoshop y que las páginas se hagan solas, aunque hay proyectos con los que puedes conseguir algo parecido. Bootstrap lo componen unos archivos que servirán de base para el desarrollo del sitio web. Se necesitan por lo tanto conocimientos de HTML, CSS y JavaScript [14].

2.31 Ventajas de usar Bootstrap en un proyecto

La más genérica es que permite simplificar el proceso de maquetación, sirviendo de guía para aplicar las buenas prácticas y los diferentes estándares. Aquí se mencionarán puntos a favor:

- 1) Se puede tener una web bien organizada de forma visual rápidamente: la curva de aprendizaje hace que su manejo sea asequible y rápido si ya sabes maquetar.
- 2) Permite utilizar muchos elementos web: desde iconos a desplegables, combinando HTML5, CSS y Javascript.
- 3) Sea lo que sea que se cree, el diseño será adaptable, no importa el dispositivo, la escala o resolución.
- 4) El grid system: maquetar por columnas nunca fue tan fácil. Además, son muy configurables.
- 5) Se integra muy bien con las principales librerías Javascript.
- 6) El haber sido creado por Twitter nos da ciertas garantías: está muy pensado y hay mucho trabajo ya hecho. Por lo tanto, hay una comunidad muy activa creando, arreglando cosas, ofreciendo plugins y mucho más.
- 7) Cuenta con implementaciones externas para WordPress, Drupal, etc.
- 8) Permite usar Less, para enriquecer aún más los estilos de la web.

2.32 ¿Porque Bootstrap?

Con la implementación de plantillas prediseñadas en HTML y CSS, los desarrolladores web no necesitan desarrollar una buena idea para una página web desde cero. Su estructura modular permite integrarlas fácilmente en cualquier documento HTML. Con esto, se eliminan muchas de las tediosas configuraciones

CSS, ahorrando una gran cantidad de tiempo y esfuerzo. Esto incluye, entre otros, los siguientes elementos:

- ✓ Botones
- ✓ Elementos de navegación
- ✓ Vista en miniatura
- ✓ Menús desplegables
- ✓ Avisos
- ✓ Historial
- ✓ Inclusión responsiva de vídeos

Otro aspecto importante es el Bootstrap grid o rejilla, sobre la que se construye el layout de una página web. Este sistema facilita la definición de los márgenes y el posicionamiento de los elementos que constituyen una página web distribuyendo los layouts en 12 columnas. En la organización de los elementos en esta cuadrícula se puede distinguir entre una gran variedad de tamaños de pantalla, ya sea de escritorio, smartphone, tablet o portátil. En cualquier caso, Bootstrap asegura un diseño web responsive.

Opcionalmente, los plugins de JavaScript basados en el framework jQuery ofrecen la posibilidad de integrar en el proyecto elementos de interfaz de usuario tan útiles como consejos de uso de la herramienta o cuadros de diálogo, así como de extender la funcionalidad de otros elementos ya integrados como, por ejemplo, la función autocompletar para los campos de búsqueda. Para poder utilizar Bootstrap con elementos de JavaScript, solo es necesario incluir la biblioteca jQuery en el documento HTML.

2.33 ¿Cómo instalar Bootstrap?

Debido a que Twitter lanzó Bootstrap como un software libre, su descarga y uso no tienen ningún coste. Aquellos desarrolladores web con más experiencia también se benefician del estatus open source de Bootstrap, pues pueden adaptar y modificar el framework de acuerdo a sus necesidades. En la siguiente guía se mostrarán las medidas necesarias para utilizar plantillas de Bootstrap en JavaScript y CSS.

- Es necesario visitar la página oficial de Bootstrap y descarga el pack “Bootstrap”, que contiene los archivos CSS y JavaScript (en versión compilada y reducida), así como las fuentes y el tema opcional.
- Se descomprimirá el archivo y se copiará el “Bootstrap File” en el directorio deseado.
- Se obtendrán las carpetas “CSS”, “fonts” y “js”. La carpeta “fonts” contiene más de 250 iconos en forma de tipografía de la fuente Glyphicons Halflings, que no tiene ningún coste para los usuarios de Bootstrap. Los archivos de las carpetas “CSS” y “js” contienen:
 - ✓ bootstrap.css: código CSS en formato legible
 - ✓ bootstrap.min.css: código CSS reducido. Se ha eliminado contenido irrelevante (como espacios) para facilitar su interpretación
 - ✓ bootstrap.css.map: contiene mapas de fuente CSS, que facilitan la búsqueda de datos originales en formato LESS
 - ✓ bootstrap-theme.css: archivos CSS de los temas prediseñados
 - ✓ bootstrap-theme.min.css: código reducido de los temas
 - ✓ bootstrap-theme.css.map: mapas de fuente de los temas
 - ✓ bootstrap.js: código JavaScript en formato legible
 - ✓ bootstrap.min.js: código JavaScript reducido

2.34 Estructura básica de un proyecto

Una vez que se ha descargado la distribución de Bootstrap desde la página principal del proyecto, se concentrará en crear un esqueleto básico sobre el que se trabajará:

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8">
    <title>Nuevo proyecto con Bootstrap 2.0</title>
    <meta name="description" content="Artículo en GenbetaDev sobre Bootstrap 2.0">
    <meta name="author" content="Ricardo Borillo">
    <!--[if lt IE 9]>
        <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
    <![endif]-->
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <style>
        body { padding-top: 60px; }
    </style>
    <link href="bootstrap/css/bootstrap-responsive.min.css" rel="stylesheet">
</head>
<body>
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
    <script src="bootstrap/js/bootstrap.min.js"></script>
</body>
</html>

```

Imagen 2.2 Ejemplo Estructura básica 4

De esta plantilla mínima, podemos destacar distintos aspectos. El primero de ellos es la inclusión de HTML Shim, el cual se encarga de que Internet Explorer reconozca los tags HTML5 y nos permita asignarles estilos.

Por otra parte, en la cabecera se debe incluir las CSS que distribuye Bootstrap y, al final de body para que la página se cargue más rápido, veremos el resto de recursos JavaScript que completan el despliegue.

Si nos centramos ya en contenido, podemos comenzar un ejemplo con una barra superior que me permita definir los menús generales del sitio:

```

<div class="navbar navbar-fixed-top">
  <div class="navbar-inner">
    <div class="container">
      <a class="brand" href="#">Genbeta Dev</a>
      <ul class="nav">
        <li class="active"><a href="#">Home </a></li>
        <li><a href="#about">Bio</a></li>
        <li><a href="#contact">Contacto</a></li>
      </ul>
    </div>
  </div>
</div>

```

Imagen 2.3 Ejemplo Menú 5

A continuación, simplemente añadimos algo de contenido al cuerpo de la página:

```

<div class="container">
  <h1>Ejemplo de Bootstrap para Genbeta Dev</h1>
  <p>Plantilla ejemplo para el desarrollo de un proyecto web.</p>
</div> <!-- /container -->

```

Imagen 2.4 Ejemplo contenido 6

2.35 Creación del layout

La definición del layout de la página es uno de los puntos fuertes de Bootstrap. La nueva disposición en base a 12 columnas permite una gran flexibilidad a la hora de crear nuestras presentaciones.



Imagen 2.5 Ejemplo layout 7

2.36 Componentes adaptativos

Para completar el diseño del layout, se puede elegir entre un diseño adaptado a 940 pixels o intentar que este pueda ser fluido.

Para ello Bootstrap nos ofrece varios recursos a tener en cuenta. Por una parte, la definición de las filas que se presentaron se declaraba con el class row. Si se desea hacer uso de la versión fluida, se utiliza un class distinto:

```
<div class="row-fluid">
  <div class="span4">...</div>
  <div class="span8">...</div>
</div>
```

Imagen 2.6 Ejemplo componentes 8

De igual forma, para definir las áreas principales de la página se puede optar por etiquetar el bloque principal con el class container como en los ejemplos, o de nuevo elegir la opción fluida:

```
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span2">
      <!--Sidebar content-->
    </div>
    <div class="span10">
      <!--Body content-->
    </div>
  </div>
</div>
```

Imagen 2.7 Ejemplo container 9

Por último, si queremos que la barra de herramientas principal se pueda adaptar a los cambios en el ancho de navegador o a la renderización en otros dispositivos de menor ancho, se aplican la siguiente construcción:

```

<div class="navbar navbar-fixed-top">
  <div class="navbar-inner">
    <div class="container">
      <a class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </a>
      <a class="brand" href="#">Genbeta Dev</a>
      <div class="nav-collapse">
        <ul class="nav">
          <li class="active"><a href="#">Home </a></li>
          <li><a href="#about">Bio</a></li>
          <li><a href="#contact">Contacto</a></li>
        </ul>
      </div>
    </div>
  </div>
</div>

```

Imagen 2.8 Ejemplo navbar 10

2.37 Responsive Design

Por último, si se desea asignar estilos específicos a los diseños en función de la resolución del dispositivo en el que se esté visualizando, Bootstrap ofrece un conjunto base de media queries con el que se puede construir expresiones del tipo:

```

// Smartphones
media (max-width: 480px) { /* ... */ }

// Tablets
media (max-width: 768px) { ... }

// Tablets apaisados y escritorio
media (min-width: 768px) and (max-width: 980px) { /* ... */ }

// Escritorio grande
media (min-width: 1200px) { .. }

```

Imagen 2.9 Ejemplo Responsive design 11

Para ello, es obligatorio incluir la siguiente etiqueta en la cabecera de la página, como se muestra en la imagen 2.7

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Imagen 2.10 Ejemplo meta 12

2.38 Responsive design

Se trata de redimensionar y colocar los elementos de la web de forma que se adapten al ancho de cada dispositivo permitiendo una correcta visualización y una mejor experiencia de usuario. Se caracteriza porque los layouts (contenidos) e imágenes son fluidos y se usa código media-queries de CSS3.

El diseño responsive permite reducir el tiempo de desarrollo, evita los contenidos duplicados, y aumenta la viralidad de los contenidos ya que permite compartirlos de una forma mucho más rápida y natural.



Imagen 2.11 Ejemplo El diseño responsivo 13

2.39 ¿En qué se basa?

Se basa en proporcionar a todos los usuarios de una web los mismos contenidos y una experiencia de usuario lo más similar posible, frente a otras aproximaciones al desarrollo web móvil como la creación de apps, el cambio de dominio o webs servidas dinámicamente en función del dispositivo.

En definitiva, el diseño web responsive se consolida como una de las mejores prácticas hoy en día en diseño web. Aunque como todas, tiene sus pros y contras, la web responsive es considerada hoy en día la mejor práctica posible en el diseño web.

2.40 Consejos de Responsive design

Uno de los grandes problemas que nos encontramos cuando hacemos una auditoría es que en los diseños responsive no se diferencia a nivel de servidor la carga de elementos. Esto es un gran problema en la carga de página ya que al hacer la web responsive se cargan exactamente los mismos recursos que para tu web de escritorio.

Para solucionarlo, trata de utilizar librerías que te permitan hacer una carga de página condicional, evitando cargar recursos innecesarios cuando la gente te visita desde móvil.

Evita efectos: Algunos de los efectos que usas en la web de escritorio no funcionarán en la web responsive. Esto es vital, ya que si no haces una comprobación multi dispositivo-plataforma puede que te encuentres que sorprendentemente el tiempo por visita desde móvil es extrañamente bajo. Esto pasa mucho, por ejemplo, cuando el “leer más” de una noticia aparece en el hover. El hover no existe en móvil por lo que el usuario no podrá continuar su visita.

Debemos de tener en cuenta todas estas cosas antes de lanzarte, así como el diseño responsive mejora la experiencia de usuario, como hemos visto, no siempre tiene que ser así.

2.41 Responsive design y los tamaños de las pantallas

Hay que tener mucho cuidado a la hora de diseñar. Se recomienda diseñar con el patrón mobile-first para evitar tener problemas de adaptación con las pantallas y evitar sobrecargar la página de elementos inútiles. La gran cantidad de dispositivos y pantallas que existen en el mercado hace que tengamos que tener muy en cuenta

tamaños de contenido fluidos y no estáticos para evitar que haya usuarios que no vean bien nuestro contenido.

CAPÍTULO 3. DISEÑO Y DESARROLLO

3.1 Diseño

La fase diseño en la construcción de software es muy importante, ya que es la parte fundamental de la aplicación en donde se conoce el funcionamiento del mismo.

Por motivos de privacidad, los ejemplos mostrados a continuación, son similares a los aplicados bajo la hipótesis planteada en el capítulo 1, ya que la empresa en la cual se utilizaron las herramientas Bootstrap y Laravel, no permiten bajo ningún motivo la utilización de material, código o trabajo generado en sus oficinas, bajo su dirección.

3.2 Desarrollo

La utilización de Laravel y Bootstrap tiene una gran competitividad en el mercado, no obstante, en el anexo 3.5 se puede observar la gran ventaja que tiene Laravel en comparación con otros frameworks. No obstante, para fines de estudio, se utilizará un sistema y analizaremos a detalle. Ya que sabemos cómo está estructurado el framework vamos a dar un ejemplo de cómo se desarrolla y estructura el código de nuestro sistema de una manera general con una tabla ficticia de una base de datos cualesquiera, se detallará las tablas reales del sistema en el anexo 4.3.

3.3 Herramientas para el desarrollo de Bootstrap y laravel

Las herramientas necesarias para utilizar laravel es Composer.

Composer es un manejador de dependencias, no un gestor de paquetes. Pero es cierto que trata con paquetes y librerías, la instalación siempre es local para cualquier proyecto, las librerías se instalan en un directorio por defecto (normalmente es /vendor).

Una vez que haya terminado la instalación del nuevo proyecto, nos creara una estructura de carpetas y archivos como se detalló en el capítulo 2.

3.4 Instalación de Composer

A continuación, se explica cómo instalar Composer y todo lo necesario para empezar a desarrollar Laravel en Windows. Previamente se debe tener instalado el

servidor web Apache mediante XAMPP o WAMPP como se menciona en el capítulo 2. y se utiliza XAMPP.

Instalar Composer

--version
You can install composer to a specific release by using the `--version` option and providing a target release.
example:
`curl -sS https://getcomposer.org/installer | php -- --version=1.0.0-alpha8`

Windows Installer
The installer will download composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.
Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed

Manual Download
If you prefer to download the phar manually, here are the available versions:

Latest Snapshot
[1.0.0-alpha8](#) (2014-01-06)
[1.0.0-alpha7](#) (2013-05-04)
[1.0.0-alpha6](#) (2012-10-23)
[1.0.0-alpha5](#) (2012-08-18)
[1.0.0-alpha4](#) (2012-07-04)
[1.0.0-alpha3](#) (2012-05-13)
[1.0.0-alpha2](#) (2012-04-03)
[1.0.0-alpha1](#) (2012-03-01)

Composer and all content on this site are released under the [MIT license](#).

Imagen 3.1 Instalación de composer para Windows

Composer es una herramienta para gestionar las dependencias en PHP. Permite declarar las librerías de las cuales el proyecto depende y se instala en el proyecto. Para instalar Composer en Windows se debe descargar de su página oficial y en la sección Windows Installer, clic en Composer-Setup.exe.

Composer Setup para instalación

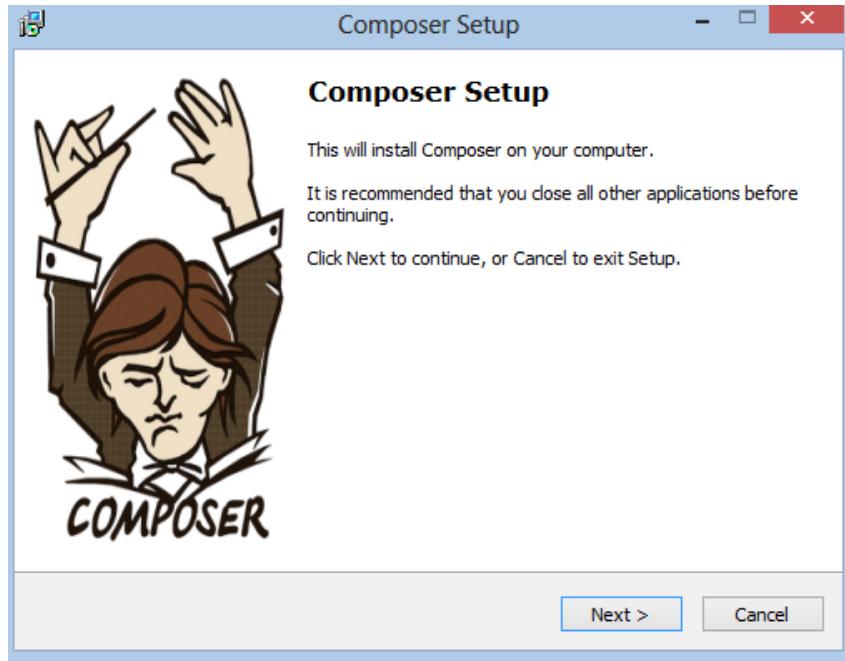


Imagen 3.2 Pestaña de Composer Setup

Una vez que la descarga finaliza, se debe ejecutar el instalador y hacer clic en Next. Si se requiere administrar los proyectos mediante el Explorador de Windows debe seleccionarse la opción «Install Shell Menus», aunque lo recomendable es la usar la línea de comandos.

Imagen de selección de componentes

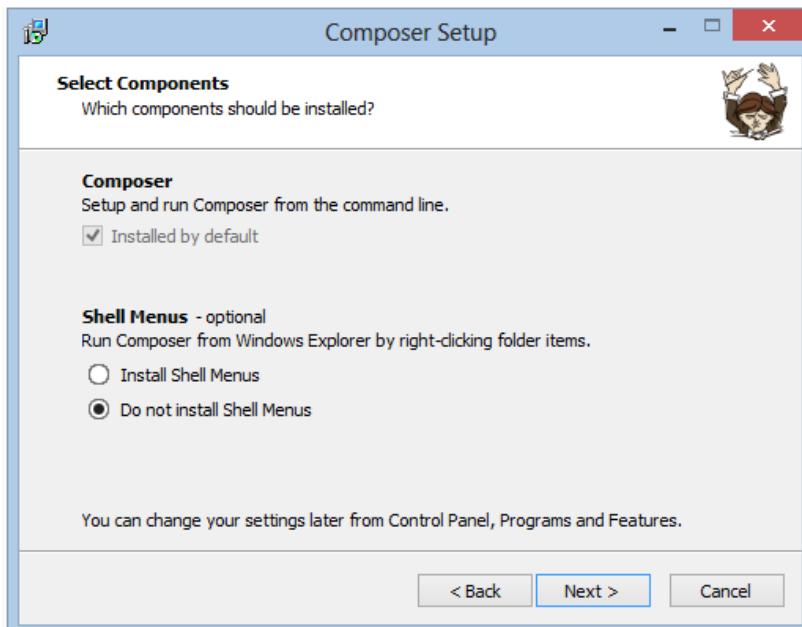


Imagen 3.3 Selección de componentes Composer

La ruta del ejecutable de PHP y como se usa XAMPP ejecutado desde PHP se encuentra en la ruta C:\xampp\php\ (si se usa WAMPP la ruta es C:\wamp\bin\php\php5.5.12) y se debe seleccionar php.exe, luego clic en Next.

Selección de ruta de PHP

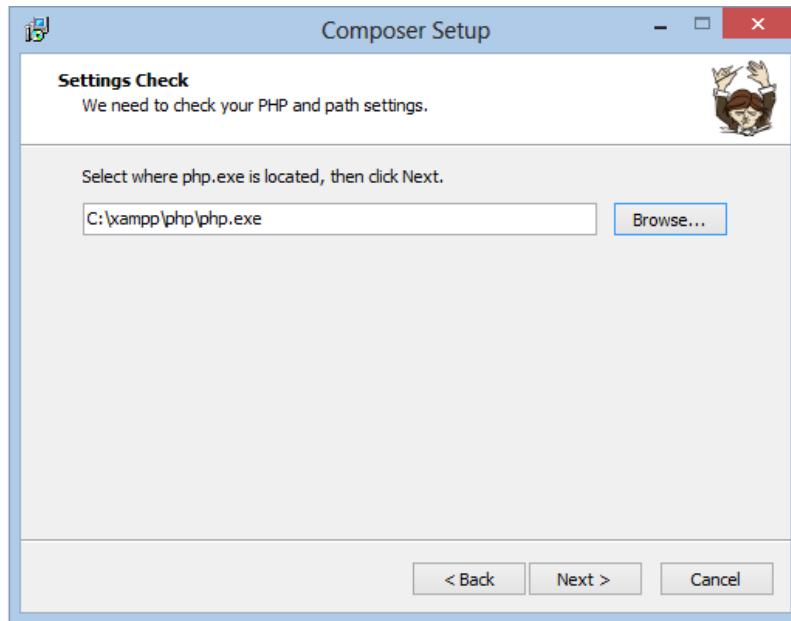


Imagen 3.4 Selección de ruta de PHP

En este punto, el instalador de Composer muestra la configuración de la instalación, simplemente dar clic a Install.

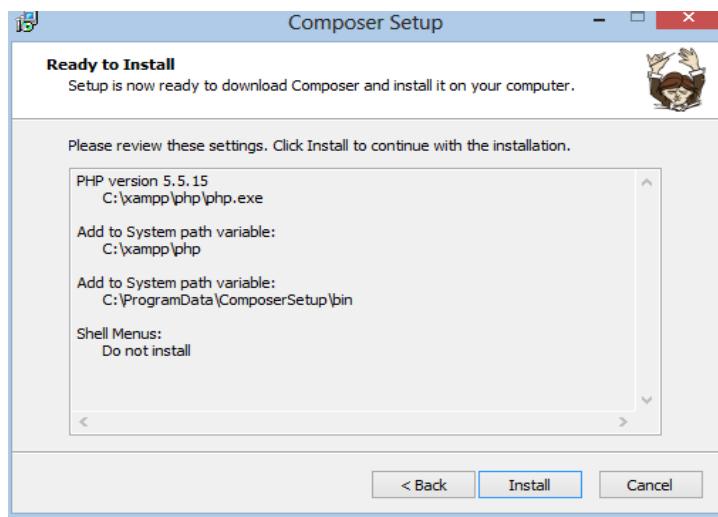


Imagen 3.5 Pasos de instalación, siguiente

Una vez instalado, aparecen otras donde simplemente se debe hacer clic en Next, y posteriormente en Finalizar. Esto permite trabajar desde consola escribiendo sólo php o composer sin necesidad de indicar la ruta del ejecutable. Para ver que todo este en orden se deben realizar dos pruebas, teclear lo siguiente:

CMD de windows

```
1 php -v (tecla Enter)  
2 composer -version (tecla Enter)
```

Imagen 3.6 Inicializar composer

3.5 Utilización de Laravel mediante Composer

Para crear él modelo de la tabla demos debemos crear un archivo llamado Demo.php en la carpeta /app/models generalmente con el siguiente código de la imagen 2.6 del capítulo 2

Clase Modelo

```
<?php  
  
class Demo extends Eloquent {  
  
    /**  
     * The database table used by the model.  
     *  
     * @var string  
     */  
    protected $table = 'demos';  
  
}
```

Imagen 3.7 Estructura de una clase en el modelo

Esto así con todas las tablas del sistema, es decir tendremos los modelos:

User.php, Organization.php, Project.php, entre otras. Respetando el estándar de la primera letra en mayúscula y en singular.

En LARAVEL los modelos utilizan el Eloquent ORM, este se encarga de convertir a objetos cada uno de los atributos de la tabla, los modelos utilizan convenciones para saber qué modelo utiliza que tabla de la base datos, pero como esas convenciones están hechas para el idioma inglés es mejor decirle directamente que tabla debe usar en qué modelo, esto lo hacemos con las variables \$table.

La primera vista que se debe crear es index.blade.php en /app/views/demo/. Para esto primero se debe crear la carpeta demo dentro de /app/views, luego dentro de esta crear el archivo index.blade.php siguiendo este estándar para todos los modelos. En la imagen 3.3 se muestra cómo estará estructurada la vista.

Vista modelo DEMO

```
@extends('layouts.master')
@section('mensajes')
@if (Session::has('message'))
    <div class="alert alert-warning alert-dismissible"><button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
        {{ Session::get('message') }}
    </div>
@endif
@stop
@section('cuerpo')
<h1> Demos </h1>
<p> {{ link_to_action ('DemoController@create', ' Nuevo','',array("type"=>"button", "class"=>"btn btn-success")) }} </p>
@if($var>>count())
<div class="table-responsive">
    <table class="table table-striped">
        <thead>
            <tr>
                <th > campo 1</th>
                <th > campo 2</th>
                <th > campo n</th>
                <th > Ver </th>
                <th > Editar </th>
                <th > Eliminar </th>
            </thead>
        <tbody>
            @foreach($var as $item)
            <tr>
                <td > {{ $item->nombre }} </td>
                <td > {{ link_to('demo'.'>$item->id,'',array("type"=>"button", "class"=>"btn btn-default btn-sm glyphicon glyphicon-eye-open")) }} </td>
            </tr>
            ...
        </tbody>
    </table>
</div>
```

Imagen 3.8 Estructura de una clase en la vista

Todas las vistas van a ser en el formato de plantillas blade, el cual se hace referencia en el capítulo 2.

En la primera línea se aprecian como las vistas estarán extendidas en una plantilla principal /app/views/layouts/master.blade.php. En la imagen 3.7 se puede apreciar la estructura de master.blade.php

Estructura de plantilla

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>Sistema de Seguimiento de Proyectos ILALO V. 1.0</title>
    @section('css')
      {{ HTML::style('css/bootstrap.min.css'); }}
    @show
  </head>
  <body>
    @yield('menu')
    @yield('mensajes')
    @yield('cuerpo')
    @yield('footer')
    @section('js')
      {{ HTML::script('js/jquery-1.11.0.js'); }}
      {{ HTML::script('js/bootstrap.min.js'); }}
    @show
  </body>
</html>
```

Imagen 3.9 Estructura de la plantilla master.blade.php

La imagen 3.8 hace referencia a como cargará la estructura de html5, los estilos de twitter bootstrap y las librerías de JavaScript.

Para enlazar el modelo y la vista, el controlador, para esto en la carpeta de /app/controllers se debe crear el archivo DemoController.php que generalmente tendrá el código que se muestra en la imagen 3.9

Clase extendida a BaseController

```

<?php

class DemoController extends BaseController {
    protected $layout = 'layouts.master';
    //métodos CRUD demo
    public function index() {
        $var= Demo::all();
        return $this->layout->content = View::make('Demo.index', compact('var'));
    }
    public function show($id) {
        $var= Demo::find($id);
        return $this->layout->content = View::make('Demo.show', compact('var'));
    }
    public function create() {
        return $this->layout->content = View::make('Demo.save');
    }
    public function store() {
        $validation = Validator::make(Input::all(), Demo::$rules,Demo::$messages);

        if ($validation->passes())
        {
            $var= new Demo();

            $var->nombre=Input::get('nombre');
            $var->save();
            Session::flash('message', 'Estado creado con Exito');
            return Redirect::route('Demo.index');
        }
    }
}

```

Imagen 3.10 Estructura de una clase en el controlador

Todos los controladores deben extenderse de la clase BaseController, el nombre debe terminar en Controller y debe ser igual al nombre del archivo. Comúnmente deben ir los métodos CRUD de las diferentes tablas, como se muestra en la imagen 3.9 y se crea un nuevo objeto del modelo y este contiene las distintas funciones que realizan esas operaciones, por lo cual no es necesario realizar las típicas consultas SQL.

Por último, se debe crear una ruta, esta es la manera que el framework redirecciona hasta la acción y el controlador que se desea, existen diferentes tipos de rutas. Para crearse se debe abrir el archivo /app/routes.php y el código dependerá de la acción que deseemos, en la imagen 3.5 se visualiza un ejemplo de rutas.

Rutas para abrir el archivo en Laravel

```

Route::controller('gantt', 'GanttController');
Route::get('user/confirm/{code}', 'UserController@getConfirm');
Route::get('user/reset/{token}', 'UserController@getReset');
Route::controller('user', 'UserController');
Route::resource('organization', 'OrganizationController');
Route::resource('project', 'ProjectController');

```

Imagen 3.11 Archivo /app/routes.php

Resumiendo, la arquitectura de funcionamiento de LARAVEL se realiza de acuerdo como se observa en la imagen 3.10

Arquitectura de Laravel

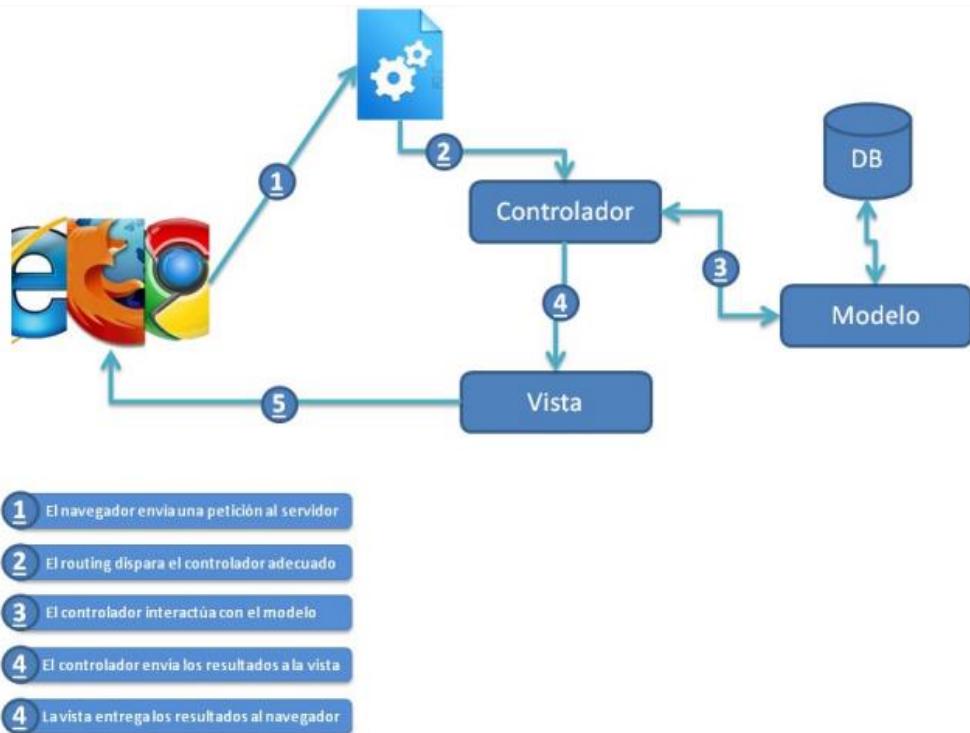


Imagen 3.12 Funcionamiento de Laravel desde MVC

3.5 Productividad del ejemplo

Para confirmar la productividad de laravel y bootstrap realizaremos una prueba de stress. Conocidas como “stress testing”, el objetivo de estas pruebas es obtener datos, sobre la carga del sistema, que ayuden a realizar el dimensionamiento del sistema. Esta prueba genera carga en el sistema hasta hacerlo colapsar, de esta manera se prueban los tiempos de respuesta ante un determinado número de conexiones y peticiones.

Utilizaremos la herramienta Apache Bench ya que es una utilidad que posee el servidor web Apache, que nos permite realizar pruebas de estrés simulando n número de conexiones y peticiones al sistema permitiendo medir el rendimiento del servidor, así como tener los tiempos de respuesta bajo los parámetros antes mencionados.

Para realizar la prueba de stress: Se debe ingresar a la aplicación e iniciamos la sesión.

Se necesita ir a Apache Bench la cookie con el identificador de sesión. Para averiguar el identificador de sesión, en Firefox está en Preferencias → Privacidad → Eliminar cookies de forma individual; tal cual como se observa en la imagen 3.7

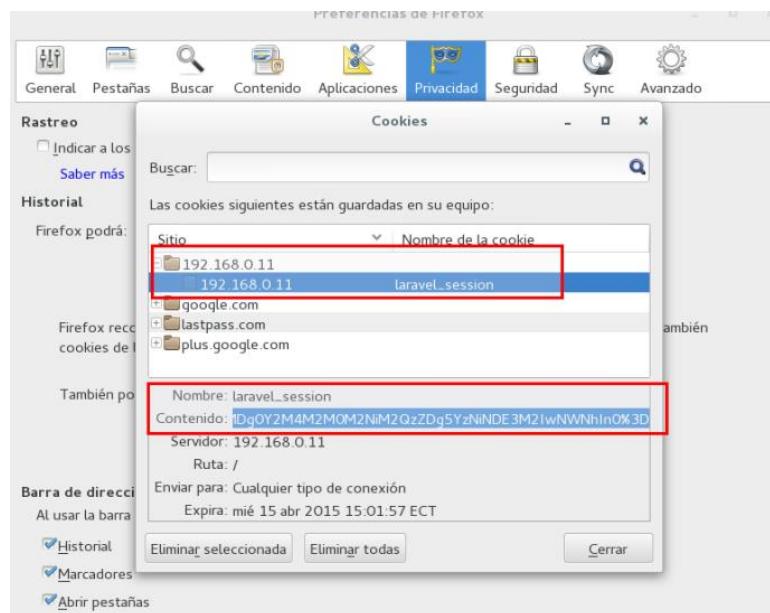


Imagen 3.13 Cookie inicio de Sesión

Se debe copiar el nombre de la cookie y su respectivo valor Y posteriormente abrir la terminal e ingresar el comando que se muestra en la imagen 3.8

```
ab -g grafica.data -n 14 -c 10 -C  
laravel_session=eyJpdiI6InpkQk1NbWp6V0E2RWxIbVZFeVhIVVVwcExsZThR  
MjNrZjJOM1gwZFVubTg9IiwidmFsdWUiOiJXM3AzWTNMVWhjTFhnU3hKMG  
JyZ2VEaktlUnkxUEtvY2tVa05tXC9RaEFiMjg1YWpoRkl0Z2drZ0dFR1NIUFZYY  
nFaTHIDTGVXRFV5TzM2cmh2bURcL013PT0iLCJtYWMiOjKODVjYThkMDhk  
ZDBkNWFjNjJmZjIINGjkZTgyY2E1ODkyMDg0Y2M4M2M0M2NiM2QzZDg5Y  
zNiNDE3M2IwNWNhIn0%3D http://192.168.0.11/ilalov1/public/
```

Imagen 3.14 Prueba de Stress

Donde:

- ab es el comando para iniciar Apache Bench.
- -g guarda los resultados en el archivo grafica.data, para poder dibujarlos luego.
- -n 14 indica que se harán de peticiones al servidor.
- -c 20 indica el número de conexiones.
- -C
- laravel_session=eyJpdiI6InpkQk1NbWp6V0E2RWxIbVZFeVhIVVVwcExsZThR
MjNrZjJOM1gwZFVubTg9IiwidmFsdWUiOiJXM3AzWTNMVWhjTFhnU3hKMG
JyZ2VEaktlUnkxUEtvY2tVa05tXC9RaEFiMjg1YWpoRkl0Z2drZ0dFR1
NIUFZYYnFaTHITGVXRFV5TzM2cmh2bURcL013PT0iLCJtYWMiOjKODVj
YThkMDhkZDBkNWFjNjJmZjIINGjkZTgyY2E1ODkyMDg0Y2M4M2M0M2Ni
M2QzZDg5YzNiNDE3M2IwNWNhIn0%3D pasa el identificador de sesión en
forma de cookie http://192.168.0.11/ilalov1/public/ es la dirección de la
aplicación

Se ingresaron estos parámetros puesto que el número de conexiones que actualmente ocupa el sistema es de 10, y el número de peticiones al servidor es un valor variable pero su tope máximo ha sido de 14 tal como se observa en la imagen 3.9.

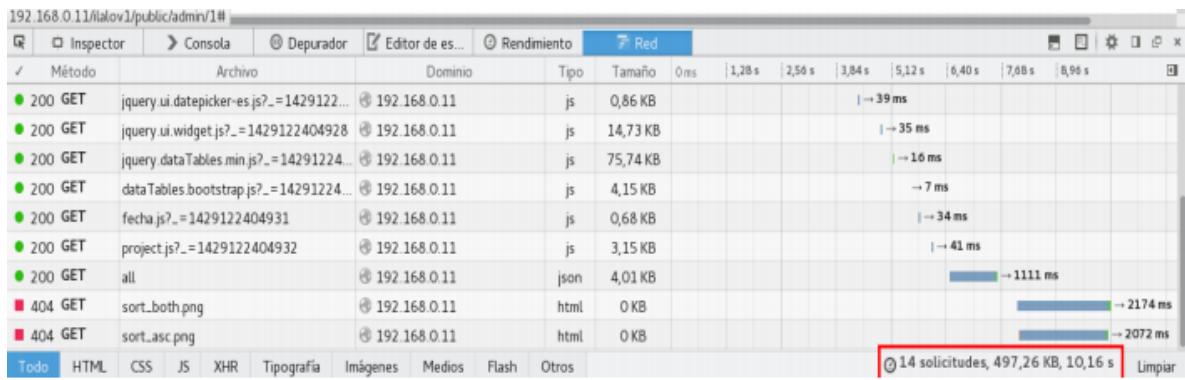


Imagen 3.15 Número máximo de peticiones al Servidor

Con la ejecución de la prueba se obtuvieron los siguientes resultados:

```
This is ApacheBench, Version 2.3 <$Revision: 1430300 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.0.11 (be patient)....done

Server Software:      Apache/2.4.7
Server Hostname:     192.168.0.11
Server Port:        80

Document Path:      /ilalov1/public/
Document Length:   5404 bytes

Concurrency Level:  10
Time taken for tests: 14.107 seconds
Complete requests:  14
Failed requests:    0
Write errors:       0
Total transferred:  85316 bytes
HTML transferred:  75656 bytes
Requests per second:  0.99 [#/sec] (mean)
Time per request:   10076.717 [ms] (mean)
Time per request:   1007.672 [ms] (mean, across all concurrent requests)
Transfer rate:      5.91 [Kbytes/sec] received

Connection Times (ms)
                  min  mean[+/-sd] median  max
Connect:        11932 1493.3 3004  3005
Processing:    887 4712 3428.1 4748 10237
Waiting:       862 4666 3422.4 4704 10209
Total:         889 6643 4284.0 7752 12229

Percentage of the requests served within a certain time (ms)

  50% 7752
  66% 9570
  75% 10238
  80% 10672
  90% 12010
  95% 12229
  98% 12229
  99% 12229
100% 12229 (longest request)
```

Imagen 3.16 Porcentaje de peticiones al servidor por segundo

Lo más relevante de estos resultados es lo siguiente:

Requests per second: peticiones atendidas por segundo durante la prueba.

Time per request (mean): tiempo medio que el servidor ha tardado en atender a un grupo de peticiones.

Time per request (mean, across all concurrent requests): tiempo medio que el servidor ha tardado en atender una petición individual.

Finalmente, para crear una gráfica del test utilizamos los datos generados en el archivo grafica.data.



Grafica 3.1 Tiempo de respuesta en peticiones en framework, Laravel

Resultados y discusiones

El software sufrirá modificaciones después que se encuentre en producción, estos ocurrirán debido a los errores que se encuentren durante su uso, el software deba adaptarse a nuevas reglas del negocio. Estos tendrán que darse bajo las normas y condiciones que Tata tengan dentro de sus estatutos en cuanto a los derechos de propiedad intelectual del software.

El sistema está diseñado en un ambiente web, pero este estará ejecutado de forma local, si en algún momento se desea migrarlo a la web se recomienda utilizar un servicio de hosting que tenga la misma arquitectura ya antes señalada.

Además, se recomienda el uso de navegadores web modernos, que estén siempre actualizados en sus últimas versiones estables, puesto que estos son más compatibles con las plantillas de diseño del sistema, de esta manera evitaremos errores en el funcionamiento.

Para finalizar, Laravel es muy cambiante y se recomienda no actualizar los distintos componentes que conforman el sistema, es decir: PHP, LARAVEL y Twitter Bootstrap, puesto que podría haber incompatibilidades entre estos.

Conclusiones

Los desarrolladores describen Bootstrap como "HTML, CSS y JS simples y flexibles para componentes e interacciones de UI populares". Bootstrap es el marco HTML, CSS y JS más popular para desarrollar primeros proyectos móviles receptivos en la web. Por otro lado, Laravel se detalla como "Un marco PHP para artesanos web". Laravel es un marco de aplicación web con sintaxis expresiva y elegante.

El desarrollo debe ser una experiencia agradable y creativa para ser verdaderamente gratificante. Laravel intenta eliminar el dolor del desarrollo al facilitar las tareas comunes utilizadas en la mayoría de los proyectos web, como la autenticación, el enrutamiento, las sesiones y el almacenamiento en caché.

Conforme pasa el tiempo se desarrollan otros framework de desarrollo más agiles, y lo importante no es dominar todos, si no, obtener el mejor provecho de este y así vez poder implementarse en sistemas de gran escala.

Referencias bibliográficas

- Arias, M. A. (2014). Responsive Webs Responsive Design with Bootstrap. Obtenido de <https://books.google.com.mx/books?isbn=1495492095>
- Bean, M. (2015). Laravel 5 Essentials. Obtenido de <https://books.google.com.mx/books?isbn=1785283294>
- Bhaumik, S. (2015). Bootstrap Essentials. Obtenido de <https://books.google.com.mx/books?isbn=1784396338>
- Bradley Efron, . T. (1994). Bootstrap Methods and Their Application. Obtenido de https://books.google.com.mx/books?id=4aCDbm_t8jUC&pg=PA183&lpg=PA183&dq=bootstrap+reference+bibliographies&source=bl&ots=mY_mcV036&sig=_pvxznHDI7ipVMIs7LmW6M252h4&hl=es-419&sa=X&ved=0ahUKEwiTsbg2f7XAhUCSCYKHbjDBkQ6AEISDAE#v=onepage&q=bootstrap%20refere
- Bradley Efron, R. T. (1994). An Introduction to the Bootstrap. Obtenido de <https://books.google.com.mx/books?isbn=0412042312>
- Development, B. R. (s.f.). Bootstrap: Responsive Web Development. Obtenido de 2013: <https://books.google.com.mx/books?isbn=1449344607>
- Dockins, K. (2016). Design Patterns in PHP and Laravel. Obtenido de <https://books.google.com.mx/books?isbn=1484224515>
- Hall, P. (2013). The Bootstrap and Edgeworth Expansion. Obtenido de <https://books.google.com.mx/books?isbn=146124384X>
- Magno, A. (2013). Mobile-first Bootstrap. Obtenido de https://books.google.com.mx/books?id=P9xdAgAAQBAJ&printsec=frontcover&dq=Bootstrap&hl=es-419&sa=X&ved=0ahUKEwjCq8Sy3_7XAhWBKyYKHR1GA2UQ6wEIZTAH
- Nutile, A. (2016). Laravel 5.x Cookbook. Obtenido de <https://books.google.com.mx/books?isbn=1786469650>

Spurlock, J. (2013). Bootstrap: Responsive Web Development. Obtenido de <https://books.google.com.mx/books?isbn=1449344607>

Stauffer, M. (2016). Laravel: Up and Running: A Framework for Building Modern PHP Apps. Obtenido de <https://books.google.com.mx/books?isbn=1491936061>

Twitter-getbootstrap. (08 de 02 de 2008). getbootstrap. Obtenido de <https://getbootstrap.com/>

YILMAZ, H. İ. (2014). Laravel Design Patterns and Best Practices. Obtenido de <https://books.google.com.mx/books?isbn=1783287993>

Glosario

AJAX: Una presentación basada en los estándares XHTML y CSS - Utiliza el DOM para interactuar con la página web. Para el intercambio de datos se basa en XML y XSLT - Sigue una petición a un servidor de forma asíncrona utilizando XMLHttpRequest - Junta todo y lo presenta al usuario utilizando JavaScript.

Ancho de banda: El ancho de banda (bandwidth) es la cantidad de datos por unidad de tiempo que consume tu sitio. Generalmente se expresa en kilobits por segundo (kbps).

Aplicaciones web: Son las aplicaciones que los usuarios acceden a ellas mediante un servidor web a través de Internet o de una Intranet mediante un navegador web.

App / apps: App o apps son las mismas aplicaciones

Benchmarking: Proceso por el cual se observa que hace la competencia y como lo hace.

Blog: Blog (bitácora en castellano) es la abreviatura de weblog y es un sitio web periódicamente actualizado, que contiene artículos publicados por orden cronológico por uno o varios autores.

Bootstrap: Bootstrap es un framework desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web. Es un framework CSS que te permitirá crear sitios web y aplicaciones web totalmente responsive sin necesidad de complicados desarrollos.

Branding: Branding es un documento o conjunto de conceptos que recogen una marca.

Browser (navegador web): Browser, web browser o navegador es un programa de ordenador utilizado para ver páginas web e interactuar con ellas.

Captcha: Captcha es el acrónimo de Completely Automated Public Turing test to tell Computers and Humans Apart (Prueba de Turing pública y automática para

diferenciar máquinas y humanos). Se trata de una prueba desafío-respuesta utilizada en computación para determinar cuándo el usuario es o no humano.

Código HTML: El código HTML normalmente significa el lenguaje de marcas HTML que permite a los navegadores Web renderizar el contenido.

CMS: Content Management System, en castellano Sistema de Gestión de Contenidos, es una herramienta para gestionar el contenido, generalmente de una web, que separa los datos, el diseño y la interactividad para permitir a los autores proveer contenidos de una forma sencilla.

Cookie: Es un fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas.

CSS: CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación del sitio web.

Diseño: El diseño web es una actividad que consiste en la planificación, diseño e implementación de sitios web y páginas web.

Diseño Responsive: Es la forma de diseñar un tipo de página web en donde el diseño web se pretende que sea adaptable a cualquiera de las resoluciones de los dispositivos como celulares, tablets, monitores de pantallas de mesa, desde los cuales los usuarios puedan llegar a ingresar al sitio web.

DNS: Domain Name System o DNS (en español: sistema de nombres de dominio) es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada.

Editor de textos: Dentro del contexto de diseño web, es un editor que principalmente trabaja con la vista de código de una página web y por tanto con las etiquetas HTML o XML.

Engagement: Engagement es el compromiso o enganche

Favicon: El favicon o icono de favoritos es una pequeña imagen que está asociada con una página o sitio Web.

Framework: Un framwork es un esquema, un esqueleto o un patrón para el desarrollo o la implementación de una aplicación software como web. Sirve para apoyar el desarrollo de sofware como sitios web dinámicos, aplicaciones web y servicios web.

Fuente: Una colección de glifos de una tipografía, definiendo el tamaño, familia, peso, y estilo del texto.

HTML: Lenguaje de Marcas de Hipertexto (Hypertext Markup Language). Es el lenguaje utilizado para escribir páginas Web. Esta basado en SGML y recientemente se ha re-escrito para seguir las directrices de XML

HTTP: Protocolo de Transferencias de Hipertexto (Hypertext Transfer Protocol). Son el conjunto de reglas utilizado para el intercambio de información en la Web.

JavaScript: JavaScript es un Lenguaje de programación.

JOOMLA: Joomla es un sistema de gestión de contenidos o CMS que permite desarrollar sitios web dinámicos e interactivos.

jQuery: jQuery es una biblioteca multiplataforma de JavaScript, permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web

Laravel: Lavarel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. Su filosofía es desarrollar código PHP de forma elegante y simple

Marcador: En inglés bookmark, es la localización almacenada de una página web, de forma que puede volver a visitarse más adelante.

Pixel: Son puntos en un monitor de ordenador. La resolución de un monitor de ordenador se mide en puntos por pulgada o pixels por pulgada.

Programador web: Es persona con conocimientos de programación de computadores que aplica la lógica de la programación a crear páginas web usando uno o varios lenguajes de programación.

Script: Un script es una porción de código que es interpretado por los navegadores para realizar dinámica una página web

SEM: SEM son las campañas de anuncios de pago en buscadores.

Target Mercadeo: El target es el Público objetivo.

Trendings: Trendings son tendencias

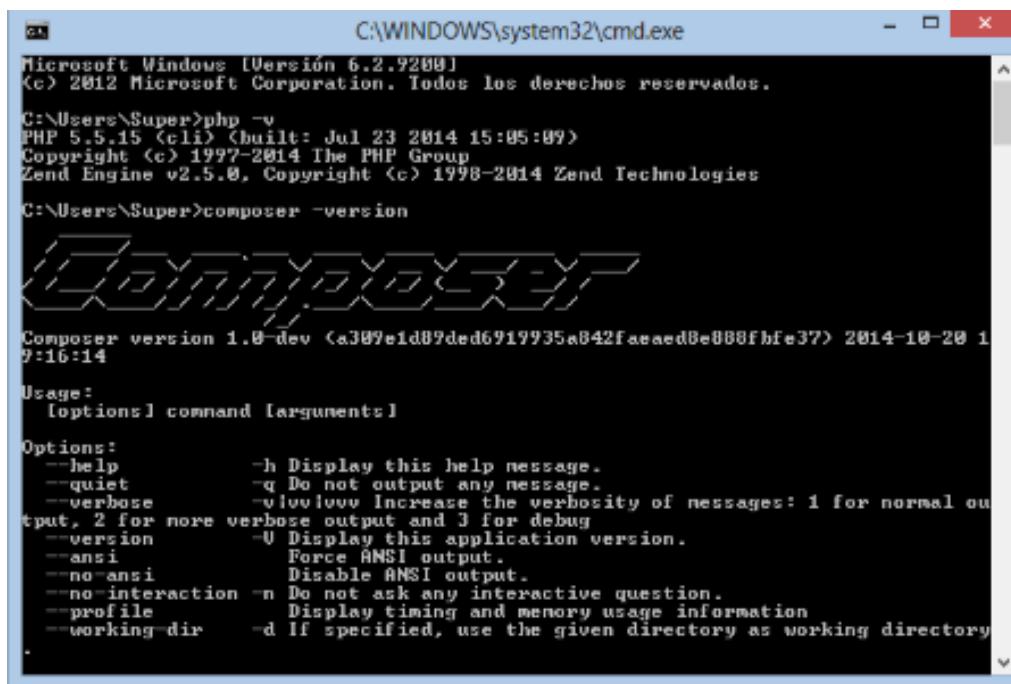
W3C: Es el acrónimo de World Wide Web Consortium. Es el grupo que determina los estándares y reglas detrás de las tecnologías Web.

Web browser: Browser, web browser o navegador es un programa de ordenador utilizado para ver páginas web e interactuar con ellas.

WordPress : WordPress es un sistema de gestión de contenidos o CMS enfocado a la creación de cualquier tipo de sitio web.

XML: Acrónimo de eXtensible Markup Language es un lenguaje de marcado para guardar información de una manera estructurada y descriptiva, por lo que nos permite que sea usado en aplicaciones como archivo de configuración, base de datos, etc.

Anexos



```
Microsoft Windows [Versión 6.2.9200]
(c) 2012 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Super>php -v
PHP 5.5.15 (cli) (built: Jul 23 2014 15:05:09)
Copyright (c) 1997-2014 The PHP Group
Zend Engine v2.5.0, Copyright (c) 1998-2014 Zend Technologies

C:\Users\Super>composer -version
The logo consists of a grid of small squares forming a stylized 'C' shape, with the word 'COMPOSER' written vertically next to it.

Composer version 1.0-dev (a389e1d89ded6919935a842faeaed8e888fbfe37) 2014-10-20 1
9:16:14

Usage:
  [options] command [arguments]

Options:
  --help           -h Display this help message.
  --quiet          -q Do not output any message.
  --verbose         -v|vv|vvv Increase the verbosity of messages: 1 for normal ou
tput, 2 for more verbose output and 3 for debug
  --version        -V Display this application version.
  --ansi            Force ANSI output.
  --no-ansi         Disable ANSI output.
  --no-interaction -n Do not ask any interactive question.
  --profile         Display timing and memory usage information
  --working-dir     -d If specified, use the given directory as working directory
```

Imagen 4.1 Version de composer, inicializada correctamente