



**Universidad Nacional de la Matanza**

**Tecnicatura en Desarrollo WE**B

|  |  |
| --- | --- |
| Trabajo Final | 2017 |
| Proyecto de investigación Taller web 2 – PHP Laravel Framework | [Escribir el subtítulo del documento] |

**Caffi Juan Manuel – Escobar Nicolás – Kuz Brian**

1. Objetivos del trabajo práctico

Investigación y desarrollo de documentación del Framework Laravel basado en PHP

1. Detalle del Tema

**Introducción**

Laravel es un poderoso y versátil [Framework basado en PHP](http://laravel.com/) que fue desarrollado por Taylor Otwell, que promete llevar al lenguaje PHP a un nuevo nivel.

**Descripción:**

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

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) a nuestra aplicación.

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.

**Manejo de los 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.

En Laravel es opcional el uso de **Eloquent**, pues también dispone de otros recursos que nos facilitan interactuar con los datos, o específicamente la creación de modelos.

La forma de interactuar con los datos en un patrón de diseño MVC, es mediante la creación de Modelos. Los Modelos son clases en PHP que encapsulan toda el trabajo con los datos de una aplicación.  
Los modelos usan convenciones para que a Laravel se le facilite el trabajo y nos ahorre tanto líneas de código como tiempo para relacionar más modelos, las cuales son:

El nombre de los modelos se escribe en singular, en contraste con las tablas de la BD que se escriben en plural.

Usan notación UpperCamelCase para sus nombres.

Estas convenciones nos ayudan a detectar automáticamente las tablas, por ejemplo: el modelo User se encuentra en singular y con notación UpperCamelCase y para Laravel poder definir que tabla es la que está ligada a este modelo le es suficiente con realizar la conversión a notación underscore y plural, dando como resultado la tabla: users.

Y esto aplica para cuando queremos crear nuestros modelos, si tenemos una tabla en la base de datos con la que queremos trabajar que se llama user\_profiles, vemos que se encuentra con las convenciones para tablas de bases de datos (plural y underscore), entonces el modelo para esta tabla cambiando las convenciones seria: UserProfile (singular y UpperCamelCase).

**Vistas**:  
Las vistas en Laravel son la parte pública que el usuario de nuestro sistema va a poder ver y se escriben en HTML.  
Laravel usa unos archivos que se llaman plantillas o templates que suelen ser nuestros archivos principales, que tienen los segmentos de código que se repiten en más de una vista, como por ejemplo la barra de navegación, un menú de opciones, la estructura del acomodo de nuestro proyecto, etc. y como deben de estar prácticamente presentes en todos lados, no tiene sentido estar los repitiendo en todas las vistas. Por defecto Laravel contiene un template llamado app.blade.php, usualmente los templates contienen el head del HTML, las ligas del CSS del sistema y una sección exclusiva para los archivos Javascript.

Además de los templates, se cuentan con archivos que se llaman partials, estos archivos son pequeños segmentos de código que suelen ser usados comúnmente en partes del sistema en específico, como los formularios o secciones de mensajes, estos archivos surgen por el código que es más pequeño que repetimos mucho pero no es lo suficientemente grande como para considerarlo un template.

Esto hace que las vistas de cada parte del proyecto, que suelen ser llamadas por una ruta o controlador sean mucho más pequeñas que usando otro tipo de frameworks para desarrollo Web, y para poder unir todos estos archivos o piezas del rompecabezas usamos el motor de plantillas de Laravel llamado BLADE.

Blade nos provee de muchas ventajas (así como casi todo en Laravel), además de modularizar nuestras vistas de una forma sorprendente, también nos permite usar estructuras de control y variables de PHP directamente en ellas.

**Ventajas**:

* Reducción de costos y tiempos en el desarrollo y mantenimiento.
* Curva de aprendizaje relativamente Baja (en comparación con otros framework Php).
* Flexible y adaptable no solo al MVC Tradicional (Modelo vista controlador) sino que para reducir código propone usar “Routes with clousures”
* -Buena y abundante documentación sobre todo en el sitio oficial.
* Posee una amplia comunidad y foros.
* Es modular y con una amplio sistemas de paquetes y drivers con el que se puede extender la funcionalidad de forma fácil, robusta y segura.
* Hace que el manejo de los datos en Laravel no sea complejo; mediante **Eloquent** (que es un ORM basado en el patrón active record) la interacción con las bases de datos es totalmente orientada a objetos, siendo compatible con la gran mayoría de las bases de datos del mercado actual y facilitando la migración de nuestros datos de una forma fácil y segura. Otro punto es que permite la creación de consultas robustas y complejas.
* Facilita el manejo de ruteo de nuestra aplicación como así también la generación de url amigables y control de enlaces auto–actualizables lo que hace más fácil el mantenimiento de un sitio web.
* El sistema de plantillas Blade de Laravel, trae consigo la generación de mejoras en la parte de presentación de la aplicación como la generación de plantillas más simples y limpias en el código y además incluye un sistema de cache que las hace más rápidas, lo que mejora el rendimiento de la aplicación.
* Cuenta con una herramienta de interfaces de líneas de comando llamada Artisan que me permite programar tareas programadas como por ejemplo ejecutar migraciones, pruebas programadas, etc.

**Desarrollo**

Actualmente los integrantes del grupo 1 comenzamos a desarrollar un juego de investigación policial basado en el framework nombrado anteriormente en el cual, el usuario deberá recabar información sobre un caso que se le planteara luego de haberse registrado en la aplicación.   
El usuario deberá pasar por distintos niveles para lograr llegar hasta el último nivel, el cual será solamente accesible si pudo resolver los casos que se le presentaron anteriormente.   
El juego contara con un tablero de puntuación general el cual dará un ranking general de todos los usuarios registrados y este se modificara a medida que el usuario avance en el juego.

1. Bibliografía:

<https://laravel.com/docs/5.4>