

“Rebel” sistema de control y gestionamiento de versiones de contenido para las equivalencias.

**Jesús Esteban Villalta Gonzalez**

**jesusesteban12321@gmail.com**

**Área de ingeniería en Sistemas**

**Programa de Ingeniería en Informática**

San Juan de los Morros, octubre de 2019

**Error! No table of contents entries found.**

“Rebel” sistema de control y gestionamiento de versiones de materias y equivalencias.

Jesús Esteban Villalta Gonzalez

jesusesteban12321@gmail.com

##### Resumen.

Este proyecto sera realizado he implementado para cumplir con las necesidades actuales de los estudiantes y profesores de nuestra Universidad Nacional Experimental Romulo Gallegos; y asi, mejorar la calidad y la seguriadad del sistema.

##### Palabras claves.

Seguridad, automatizar y mejorar.

##### Introducción.

Este sistema de información será implementado en laravel 5.8, es de código abierto para desarrollar aplicaciones y servicios web; fue creado en 2011 por Taylor Otwell, gran parte de laravel está formado por dependencias especialmente de Symfony, las características de laravel son:

* Sistema de Ruteo
* PeticionFluent
* Eloquent ORM
* Blade, motor de plantillas
* Patrón MVC
* Componentes de Symfony

En la comunidad de desarrolladores es muy considerado ya ha sido descargado más de 320.000 veces.

En este proyecto va a ser implementado por la causa de actualizar, mejorar y reforzar la seguridad del proyecto que se encuentra ya existente.

Los detalles más sobresalientes son no tener validaciones de correo, validaciones de formularios, validaciones de URLs, base de dato, entre otros. El efecto de implementar esta actualización es la mejora y automatizaciones, seguridad de los datos de la institución, validar el sistema vigente, agregar módulos funcionales al sistema, algunos errores técnicos y más estilo para que sea más agradable a la vista de los usuarios.

Se requiere actualizar y validad consultas para mayor seguridad; mejorar visivilidad de las vistas.

##### Metologia de investigación.

Descripción de las metodologías existentes y pertinentes.

El Modelo de prototipos, en Ingeniería de software, pertenece a los modelos de desarrollo evolutivo. El prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar muchos recursos.

Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.

###### Etapas.

* Comunicación.
* Plan rápido.
* Modelado, diseño rápido.
* Construcción del Prototipo.
* Desarrollo, entrega y retroalimentación.
* Entrega del desarrollo final.
* Recolección y refinamiento de requisitos.
* Desarrollo, evaluación del prototipo por el cliente.
* Refinamiento del prototipo.
* Producto de Ingeniería.

###### Ventajas.

También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.

Inconvenientes.

El usuario tiende a crearse unas expectativas cuando ve el prototipo de cara al sistema final. A causa de la intención de crear un prototipo de forma rápida, se suelen desatender aspectos importantes, tales como la calidad y el mantenimiento a largo plazo, lo que obliga en la mayor parte de los casos a reconstruirlo una vez que el prototipo ha cumplido su función. Es frecuente que el usuario se muestre reacio a ello y pida que sobre ese prototipo se construya el sistema final, lo que lo convertiría en un prototipo evolutivo, pero partiendo de un estado poco recomendado.

En aras de desarrollar rápidamente el prototipo, el desarrollador suele tomar algunas decisiones de implementación poco convenientes (por ejemplo, elegir un lenguaje de programación incorrecto porque proporcione un desarrollo más rápido). Con el paso del tiempo, el desarrollador puede olvidarse de la razón que le llevó a tomar tales decisiones, con lo que se corre el riesgo de que dichas elecciones pasen a formar parte del sistema final.

###### Selección y justificación de la metodología de trabajo.

Debido al desarrollo o etapas de esta metodología la he tomado en cuenta por el motivo de que el cliente no está muy seguro no tiene la certeza de cómo puede funcionar el sistema y no tener los suficientes datos o requerimientos para desarrollarlo, esta metodología es bastante factible ya que esta una método evolutivo que este se encarga de que el prototipo valla evolucionando según las necesidades o lo requerimientos expuesto por el cliente.

###### Implementación de la metodología de desarrollo.

Emplear el método de prototipo para la gestión de proyectos reporta numerosos e interesantes beneficios que compensan las dificultades de su implantación; generar un demo funcional lógicamente y estéticamente, para que el cliente se dé cuenta cómo puede marchar el proyecto, recolectar toda la información posible y plasmarla en el demo o prototipo, es necesario ya que esta es la base del sistema, utilizar herramientas de CASE o mejor conocidas como repositorios, ya que estoy implementando un método de prototipo evolutivo.

##### Diagnóstico Situacional.

###### Descripción del contexto de la situación problemática planteada.

El sistema esta siendo desarrollando con el objetivo y la finalidad de solventar el proble de la desactualizacion y la inseguridad del sistema antiguo, el sistema esta programado en una version desactualizada de CAKE y a su vez de php, en la que se tiene que solucionar o evitar la inyeccion por la url, validación de los formularios y mejoras de las vista para los usuarios, relacionar la base de datos para no repetir los datos.

###### Objetivos del proyecto

El objetivo principal es actualizaar a una nueva version de php y a un mejor framework, en esta nueva version del sistema es poder cumplir con las necesidades que a precentado los usuarios, que la información de los usuarios no sea corrompida ni extraida para su propia seguridad.

Procesos que se van a automatizar en la empresa

* El registro de las materias.
* El registros de los contenidos.
* La velocidad de los reportes.
* El diseño grafico y visual del sistema.
* El versionamiento de los contenidos.
* El rollback de los contenidos.
* La autentificacion y validacion de los usuarios.
* Capas de apceso o Middleware.
* Manejo de roles.
* Base de datos relacionada.

##### Determinacion, Instalacion y configuracion de la herramienta de desarrollo.

El framework Laravel tiene algunos requisitos del sistema. Todos estos requisitos son cubiertos por la máquina virtual Laravel Homestead, así que es altamente recomendable que uses Homestead como tu entorno local de desarrollo de Laravel.

Sin embargo, si no estás utilizando Homestead, deberás asegurarte de que tu servidor cumpla con los siguientes requisitos:

* PHP >= 7.0.0.
* OpenSSL PHP Extension.
* PDO PHP Extension.
* Mbstring PHP Extension.
* Tokenizer PHP Extension.
* XML PHP Extension.

Laravel utiliza Composer para administrar sus dependencias. Por lo que, antes de utilizar Laravel, deberás asegurarte de tener Composer instalado en tu sistema.

###### Mediante el instalador de Laravel

Primero, descarga el instalador de Laravel usando Composer:

***composer global require laravel/installer***

Una vez instalado, el comando laravel new creará una nueva instalación de Laravel en el directorio que especifiques. Por ejemplo, laravel new blog creará un directorio blog que contendrá una nueva instalación de Laravel con todas las dependiencias de Laravel ya instaladas:

laravel new blog

Mediante composer create-project

Alternativamente, también puedes instalar Laravel ejecutando el comando de Composer create-project en tu terminal:

***composer create-project --prefer-dist laravel/laravel blog***

##### Configuración

###### Archivo composer.json

Este archivo contiene un json para que los paquetes y requerimientos del sistema sean instalados para que este funcione como debe de ser, se debe ejecutar el comando

***Composer update***

###### Directorio público

Después de haber instalado Laravel, deberás configurar el documento raíz de tu servidor web para que sea el directorio public. El archivo index.php en este directorio funciona como controlador frontal (front controller) para todas las peticiones HTTP que entran a tu aplicación.

###### Archivos de configuración

Todos los archivos de configuración para el framework Laravel están almacenados en el directorio config. Cada opción está documentada, así que siéntete libre de revisar estos archivos y familiarizarte con las opciones disponibles para ti.

###### Permisos para directorios

Después de haber instalado Laravel, necesitarás congigurar algunos permisos. Los directorios dentro de storage y bootstrap/cache deberán tener permiso de escritura para tu servidor web o Laravel no va a funcionar. Si estás utilizando la máquina virtual Homestead, estos permisos ya están establecidos.

###### Clave de la aplicación

Lo siguiente que debes hacer después de instalar Laravel es establecer la clave de tu aplicación a una cadena aleatoria. Si instalastes Laravel mediante Composer o el instalador de Laravel, esta clave ya ha sido establecida por el comando php artisan key:generate.

Típicamente, esta cadena debe tener una longitud de 32 caracteres. La clave puede ser establecida en el archivo de entorno .env. Si no has renombrado el archivo .env.example a .env, deberás hacerlo ahora. Si la clave de la aplicación no está establecida, ¡las sesiones de usuario y otros datos encriptados no serán seguros!

###### Configuración adicional

Laravel casi no necesita de configuración adicional. ¡Eres libre de empezar a desarrollar! Sin embargo, puede que quieras revisar el archivo config/app.php y su documentación. Contiene varias opciones como timezone y locale que es posible que desees ajustar en tu aplicación.

También puede que quieras configurar algunos componentes adicionales de Laravel, como:

* Cache
* Base de Datos
* Sesionen

##### 

##### Desarrollo de la aplicacion

###### Descripción

Rebel sistema de informacion para el control de versiones de contenidos de la universidad romulo gallegos, esta desarrollado en Laravel este sistema esta conformado, por un login una interfas para registrarse que en esta vista se precenta un formulacio que tiene un algoritmo para reconocer los correos y poder seleccionar unado es un profesor o un bachiller de la universidad romulo gallego, posteriormente le llega un e-mail con un codigo de verificacion al correo ingresado para poder verificar la existencia de ese correo y asi poder apceder al sistema, el sistema costa por 5 roles que son los roles de administrador, estudiante, profesor, coordinador y adminitrador curricular. Por la parte del administrador estan los modulos de universidad, area, carrera, materia, contenido y usuarios; el administrador es el que puede validar a los usuarios y a los profesores que se estan registrando para poder tenir un buen manejo de los usuarios, tiene privilegios de manejar los modulos universidad, area, carrera y materias. El rol profesor este solo puede ver los contenidos que son de su catedra y descargarlos, el rol de coordinador, este rol es el que puede agregar nuevos contenidos y editar los contenidos de la catedra asignada, este es el que puede versionar los contenidos editandolos o haciendo un rollback para regresar a contenidos antiguos, para que estos esten vigente necesitan la autentificacion del adminitrador curricular que esta es la funcion de este rol. El rol del estudiante es poder descargar la equivalencia corespondiente a su tiempo en que este aprovo la materia y con el contenido respectivo.

###### Requerimientos funcionales del proyecto

El respectivo versionamiento, minipulacion y verificacion de los contenidos, verificacion de usuarios, base de datos relacionada, interfaces mas atractivas.

###### Requerimientos no funcionales del sistema de informacion.

Modulo universidad no es necesario porque el sistema solo seria para la universidad romulo gallego.

##### Fase de diseño

Se comienza elaborando un prototipo del producto final: qué aspecto tendrá, cómo funcionará. Para muchas interfaces de usuario, este modelo puede resultar tan simple como unos dibujos con lápiz y papel o tan complejo como el propio código operativo final. Para interfaces de hardware o estaciones de trabajo, el modelo puede consistir en maquetas de espuma, caucho, cartón o cartulina. Cuanto más próximo se encuentre el prototipo al producto real, mejor será la evaluación, si bien se pueden obtener magníficos resultados con prototipos de baja fidelidad.

##### Fase de codificación.

###### Requerimientos de desarrollo

1. Para poder desarrollar en laravel 5.5 los requerimientos son:
   1. PHP >= 7.0.0.
   2. OpenSSL PHP Extension.
   3. PDO PHP Extension.
   4. Mbstring PHP Extension.
   5. Tokenizer PHP Extension.
   6. XML PHP Extension.
2. Editor de texto ***SublimeText3.***
3. Servidor local xampp.
4. Controlador de versiones git.
5. Repositorio Bitbucket.

##### Fase de Prueba

\*Elaboración y ejecución del plan de prueba

\*Analisis del resultado

##### Concluciones

##### Recomendaciones

##### Referencias

\*Se utilizara el sistema de referencia de la IEEE\*

##### Anexos

