

**Auto generador de código (scaffolding) bajo el patrón modelo vista controlador - MVC,  
para desarrollos ágiles en lenguaje PHP usando el framework CodeIgniter**

**Duvan Alexander Angarita Sánchez**

**Universidad Nacional Abierta y a Distancia -UNAD**

**Escuela de Ciencias Básicas, Tecnología e Ingeniería – ECBTI**

**Ingeniería de Sistemas**

**Noviembre del 2019**

**Auto generador de código (scaffolding) bajo el patrón modelo vista controlador - MVC, ii  
para desarrollos ágiles en lenguaje PHP usando el framework CodeIgniter**

**Autor:**

**Duvan Alexander Angarita Sánchez**

**Trabajo de Grado**

**Presentado como requisito para optar al título de Ingeniero de Sistemas**

**Asesor:**

**Luis Fernando Arias Ramírez**

**Universidad Nacional Abierta y a Distancia -UNAD**

**Escuela de Ciencias Básicas, Tecnología e Ingeniería – ECBTI**

**Ingeniería de Sistemas**

**Noviembre del 2019**

**NOTA**

---

---

**FIRMA JURADOS**

**JURADO 1** \_\_\_\_\_

**JURADO 2** \_\_\_\_\_

**JURADO 3** \_\_\_\_\_

**JURADO 4** \_\_\_\_\_

Dedicado a mi esposa e hijos, ellos son la llama que enciende mi vida, y me dan fuerzas para seguir adelante con mis metas y logros, los amo.

## **Agradecimientos**

v

Agradezco a mis padres, hermanos, amigos y profesores que me han brindado su apoyo para poder llevar a cabo mis estudios, ustedes también hacen parte importante de este logro.

The use of agile development methodologies today has allowed the software industry to obtain products in a short time and of very good quality, but at the same time it has created a much higher workload for software developers, demanding Short term deliveries. For this reason, it is necessary to use frameworks and tools that allow to maintain the quality of development and speed to it.

For this reason, in this project a technological proposal for the development of web applications is presented through the PHP CodeIgniter framework using the MVC pattern, where CRUDS can be generated in a matter of seconds through the database tables, as well as the generation of base authentication and generation of tables from the same software, allowing a self-management and product growth from the inside out.

This project will be developed through the KANBAN agile development methodology, with which there will be a control of tasks to be performed to deliver a tool that allows software developers to focus their attention on a critical and important point which is the development of business logic.

**Key words:** Scaffolding, MVC, Model-View-Controller, PHP, CRUDS, KANBAN.

INTRODUCCIÓN .....	1
JUSTIFICACIÓN .....	2
1. TITULO .....	3
2. DEFINICIÓN DEL PROBLEMA .....	4
2.1. ANTECEDENTES DEL PROBLEMA .....	4
2.2. FORMULACIÓN DEL PROBLEMA.....	5
2.3. DESCRIPCION DEL PROBLEMA.....	5
2.4. SISTEMATIZACIÓN DEL PROBLEMA .....	6
3. OBJETIVOS .....	7
3.1. OBJETIVO GENERAL.....	7
3.2. OBJETIVOS ESPECÍFICOS .....	7
4. MARCO REFERENCIAL.....	8
4.1. MARCO TEORICO.....	8
4.2. MARCO CONCEPTUAL .....	9
5. DISEÑO METODOLÓGICO.....	13
5.1. TIPO DE INVESTIGACIÓN .....	13
5.2. MÉTODO .....	13
5.3. HIPÓTESIS.....	16
5.4. VARIABLES E INDICADORES.....	16
5.5. POBLACIÓN O MUESTRA.....	18
5.6. INSTRUMENTO Y TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN.....	19
6. ESQUEMA TEMATICO.....	21

6.1.	REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES .....	viii
	Listado de Requerimientos Funcionales .....	21
	Listado de Requerimientos No Funcionales .....	22
6.2.	DEFINICIÓN DE LA BASE DE DATOS .....	23
	Modelo Entidad Relación de la base de datos (MER) .....	23
	Diccionario de Datos.....	24
6.3.	PRERREQUISITOS PARA DESARROLLO .....	29
6.4.	CONFIGURACIÓN DE AZURE DEVOPS .....	31
6.5.	ESTRUCTURA DEL PROYECTO .....	34
6.6.	DESARROLLO Y CODIFICACIÓN DEL PROYECTO.....	37
	Modelos.....	37
	Controlador .....	41
	Vistas.....	67
7.	PERSONAS O PROPONENTES .....	72
	7.1. PROPONENTE PRIMARIO .....	72
	7.2. PROPONENTE SECUNDARIO.....	72
8.	RECURSOS DISPONIBLES .....	73
	8.1. RECURSOS FISICOS .....	73
	8.2. RECURSOS INSTITUCIONALES.....	73
	8.3. PRESUPUESTO.....	73
9.	RESULTADOS E IMPACTO ESPERADO .....	75
10.	CRONOGRAMA DE ACTIVIDADES .....	76
	CONCLUSIONES .....	78



LISTA DE REFERENCIAS ..... 79

## Lista de tablas

x

Tabla 1 Listado de requerimientos funcionales .....	21
Tabla 2 Listado de requerimientos no funcionales .....	22
Tabla 3 Descripción de tabla estado .....	24
Tabla 4 Descripción de tabla menú.....	24
Tabla 5 Descripción de tabla permisos .....	25
Tabla 6 Descripción de tabla rol .....	26
Tabla 7 Descripción de tabla usuario.....	27
Tabla 8 Recursos y gastos - Fuente: Este proyecto.....	73

## Tabla de Ilustraciones

xi

Ilustración 1 Patrón Modelo Vista Controlador.....	10
Ilustración 2 Tablero metodología KANBAN.....	14
Ilustración 3 Ranking Github Lenguajes .....	17
Ilustración 4 Ranking TIOBE Lenguajes.....	17
Ilustración 5 Ranking Frameworks PHP Google Trends.....	18
Ilustración 6 Modelo Entidad Relación de la base de datos. ....	23
Ilustración 7 Panel de administración XAMPP .....	30
Ilustración 8 Carpeta raíz del proyecto .....	31
Ilustración 9 Creación de proyecto Azure DevOps .....	32
Ilustración 10 Repositorio DevOps.....	32
Ilustración 11 Visual Studio Code Commit .....	33
Ilustración 12 Repositorio GIT del proyecto .....	33
Ilustración 13 Tablero KANBAN Azure DevOps .....	34
Ilustración 14 Estructura de proyecto CodeIgniter .....	35
Ilustración 15 Cronograma de actividades.....	76
Ilustración 16 Diagrama de Gantt Actividades.....	77

## INTRODUCCIÓN

El desarrollo de software es una de las ramas de la ciencia de la computación con más crecimiento y avance tecnológico en las últimas dos décadas, desde la cual han surgido nuevas formas, metodologías, arquitecturas entre otras características necesarias para entregar un buen producto al cliente, pero esto depende de diferentes factores como el factor técnico, el factor comercial y el factor económico. Uno de los factores técnicos más importantes a la hora de realizar desarrollos tecnológicos es la arquitectura ya que es la que soporta el adecuado comportamiento del sistema, pero existe un factor comercial que se ha vuelto fundamental a la hora de rentabilizar un desarrollo informático es el tiempo.

Por lo cual nos adentraremos en esta propuesta en analizar como la arquitectura y el tiempo convergen a la hora de ofrecer un sistema informático a la medida a través de la creación de andamios para la autogeneración de código y utilización de frameworks opensource existentes que permitan realizar desarrollos basados en arquitecturas estables ahorrando tiempos en diseños de arquitecturas y codificación, sin perder calidad en el producto.

Por último, identificaremos como estas herramientas permitirán simplificar la carga de trabajo a todo un grupo de desarrollo tecnológico (Gerente de proyecto, arquitecto de software, desarrolladores, administradores de base de datos, analistas de requerimientos y diseñadores gráficos).

## JUSTIFICACIÓN

La búsqueda continua de optimizar los procesos de desarrollo de software han motivado a los programadores informáticos, arquitectos de software y gerentes de proyectos a factorizar mejor el código, permitiendo incluso una autogestión y auto codificación desde el núcleo del software hacia afuera, con el único fin de reducir tiempos de respuesta y entrega muchos más cortos, por lo cual en este proyecto se plantea el desarrollo de un auto generador de código propio para el framework CodeIgniter que permita realizar desarrollos ágiles a través de la generación CRUD leyendo las tablas de base de datos y sus relaciones, así como la generación de autenticación y gestión base del sistema informático.

Este proyecto tecnológico permitirá a los desarrolladores de software agilizar procesos de producción de sistemas de información entre un 50 y 70 por ciento en tiempos con el fin de realizar entregas mucho más rápidas y así como dar cumplimiento de las metas planteadas en la metodología de desarrollos ágiles. Por lo cual es importante resaltar que este desarrollo también ira de la mano con el modelo de desarrollo planteado en el patrón MVC (Modelo Vista Controlador) el cual es uno de los más usados hoy en día, el cual facilita organizar estructuralmente la arquitectura del software y su código con una mayor legibilidad de este.

## **1. TITULO**

Auto generador de código (scaffolding) bajo el patrón Modelo Vista Controlador - MVC, para desarrollos ágiles en lenguaje PHP usando el framework CodeIgniter.

## **2. DEFINICIÓN DEL PROBLEMA**

### **2.1. ANTECEDENTES DEL PROBLEMA**

El uso de metodologías de desarrollos ágiles es una tendencia en crecimiento a la hora de desarrollar sistemas de información, esto ha llevado a realizar una mayor exigencia a los desarrolladores de software ya que el objetivo de dichas metodologías es la de realizar entregas mucho más rápidas de forma parcial para que el cliente certifique avances cada cierto tiempo. Esto ha llevado a los grupos de desarrollo de software a implementar la utilización de frameworks y andamios como complementos que ayudan a cumplir las metas planteadas el desarrollo informático. No obstante, no ha sido suficiente la utilización de estos frameworks y complementos como herramientas de apoyo, ya que algunas son muy costosas e incluso son muy difíciles de manejar o su estructura es mucho más compleja que el mismo lenguaje de desarrollo.

Si observamos de cerca el lenguaje PHP, el cual desde hace algunos años se ha posicionado como uno de los 10 lenguajes de desarrollo más usados a nivel mundial por ser de código libre, existen herramientas que simplifican su desarrollo pero que son costosas como lo son ScriptCase o PHPRunner; por otro lado, existen los framework que facilitan el desarrollo proporcionando el andamiaje inicial por lo cual se convierte en la forma más conveniente a la hora de realizar desarrollos tecnológicos. Teniendo esto presente, ahora un desarrollador se encuentra en la toma de decisión de cuál es el mejor framework de desarrollo para iniciar un proyecto, existen muchos factores, pero el principal es la simplicidad para la curva de aprendizaje, por esta característica el

framework más utilizado es CodeIgniter. Pero a este framework le hace falta una herramienta fundamental que incluso otros menos utilizados tienen y es el generador de CRUD, el cual permite generar la creación, lectura, actualización y eliminación de entidades de la base de datos a través de modelos.

## **2.2. FORMULACIÓN DEL PROBLEMA**

¿Cómo desarrollar un proyecto tecnológico base que contenga autogestión y autogeneración de código y que a su vez permita aplicar adecuadamente las metodologías de desarrollos ágiles reduciendo tiempos de entrega, pero sin perder calidad en la arquitectura de software?

## **2.3. DESCRIPCIÓN DEL PROBLEMA**

Hoy en día no existe un autogenerador de código para el framework CodeIgniter que permita un desarrollo de aplicaciones mucho más rápido y eficiente, teniendo en cuenta toda la estructura del modelo entidad relación de la base de datos permitiendo obtener las tablas y a su vez cada tipo de dato, relación y llaves foráneas para poder generar formularios en PHP para la listar, agregar, editar y eliminar tuplas. Esto es un complemento necesario para este framework ya que es muy potente y permite una curva rápida de aprendizaje, pero no cuenta con un scaffolding como lo tienen otros frameworks PHP como Yii, Laravel o Symfony.



## 2.4. SISTEMATIZACIÓN DEL PROBLEMA

- ¿Qué aspectos motivan a los grupos de desarrollo de software a trabajar en herramientas para la autogeneración de código?
- ¿Cuáles son los factores más importantes para tener en cuenta al momento de crear herramientas de reducción de tiempos en los desarrollos de software?
- ¿Cuáles herramientas y lenguajes de programación son los adecuados para poder desarrollar un proyecto tecnológico para la autogeneración de código?
- ¿Cómo un software de autogeneración de código brinda un beneficio a los clientes del producto final?

### **3. OBJETIVOS**

#### **3.1. OBJETIVO GENERAL**

Desarrollar una herramienta tecnológica para la autogeneración de código que permita agilizar los desarrollos de aplicaciones con el framework CodeIgniter para el lenguaje PHP.

#### **3.2. OBJETIVOS ESPECÍFICOS**

1. Identificar los requerimientos funcionales y no funcionales del proyecto.
2. Definir el Modelo Entidad Relación de la base de datos y sus tablas base.
3. Definir los prerequisites para llevar a cabo la ejecución del desarrollo.
4. Realizar la configuración del entorno de ejecución del proyecto teniendo presente la metodología KANBAN.
5. Identificar la estructura principal del framework CodeIgniter.
6. Desarrollar las funcionalidades que permita a través de la estructura de la base de datos generar los modelos, vistas y controladores por cada una de las tablas.
7. Desarrollar las funcionalidades que permitan la generación de permisos de lectura y escritura sobre la información así como la administración de autenticación base de los usuarios del sistema.

## 4. MARCO REFERENCIAL

### 4.1. MARCO TEORICO

Para poder realizar un desarrollo tecnológico hoy en día no basta con solo codificar, también es necesario tener presente la arquitectura de software a utilizar para que el comportamiento del nuevo sistema llene las expectativas a partir de su rendimiento, por lo cual la utilización de un marco de desarrollo más conocidos como frameworks, los cuales son considerados como aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta (Núñez & Ríos, 2017), por lo cual es una herramienta esencial para cualquier desarrollador, ya que le permite tener un andamiaje inicial para construir aplicaciones robustas teniendo presente que ya cuentan con una arquitectura definida. Estos frameworks tienen una distribución de arquitectura dependiendo de las necesidades del desarrollador, una de las arquitecturas más utilizadas son las que implementan el patrón Modelo Vista Controlador (MVC), el cual data de los años 70 y fue introducido en el lenguaje de programación SmallTalk. Según su arquitecto Trygve Reenskaug, "MVC fue creado como una solución al problema general de dar a los usuarios control sobre su información como se ve desde múltiples perspectivas". Trygve concibió MVC mientras trabajaba como científico visitante en el Laboratorio de Investigación Xerox Palo Alto (PARC), entre el verano de 1978 y 1979. Mientras estuvo en PARC, se centró en apoyar al LRG (grupo de investigación de aprendizaje), un equipo que estaba construyendo el DynaBook.

Con el uso del patrón MVC podemos mantener un código mucho más organizado permitiendo separar los datos de la lógica de negocio y así poder realizar una adecuada reutilización de código y separar la interacción del usuario (a través de las vistas) con los datos.

Teniendo ya un framework y una arquitectura definida, es indispensable contar con una metodología de desarrollo la cual define la forma en la que se debe trabajar, además debe estar a la altura de las necesidades tecnológicas de hoy en día, por lo cual las metodologías más usadas son las ágiles. Estas metodologías de desarrollos ágiles permiten tener un enfoque para la toma de decisiones basados en el desarrollo iterativo e incremental teniendo como base el trabajo colaborativo promoviendo así un proceso de toma de decisiones en corto plazo (Wikipedia, 2019). El método agile es un proceso que permite a un grupo de trabajo dar respuestas rápidas e impredecibles, crean oportunidades de evaluar la dirección de un proyecto durante el ciclo de desarrollo, (Gonçalvez, 2020).

## **4.2. MARCO CONCEPTUAL**

### **Metodologías Agiles**

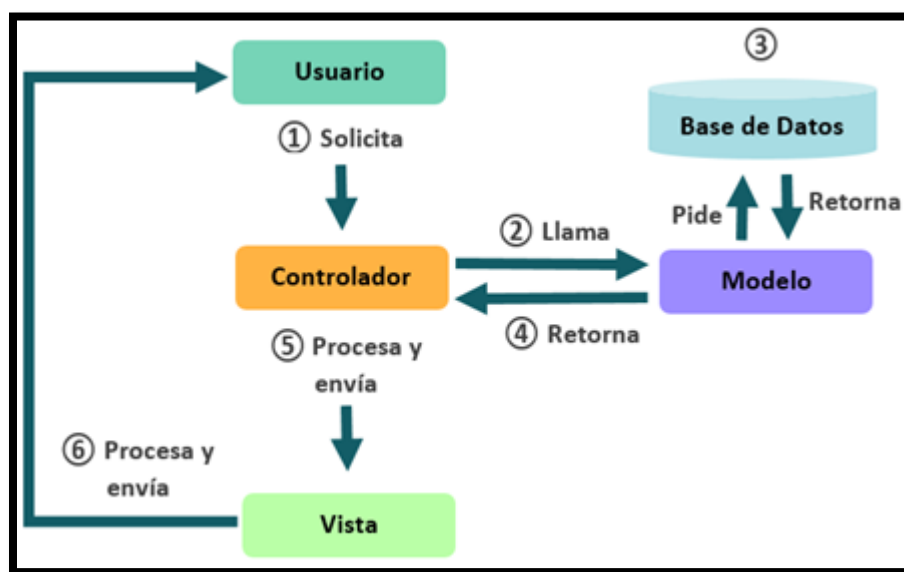
Las metodologías agiles son un enfoque de trabajo aplicado a proyectos, que permite a través de métodos organizar grupos interdisciplinarios o equipos de trabajo organizados con el objetivo de generar un producto a través de una serie de requerimientos de forma flexible e inmediata. Por otro lado mejoran la satisfacción del cliente ya que lo hace participe a lo largo del proyecto.

## KANBAN

El método Kanban para el trabajo del conocimiento y el trabajo de mantenimiento fue formulado por David J. Anderson en 2005. Anderson combinó elementos del trabajo de W Edwards Deming, Eli Goldratt, Peter Drucker y Taiichi Ohno. El método incorpora conceptos como sistemas “pull” (de arrastre), teoría de colas y flujo.

### Patrón Modelo Vista Controlador (MVC)

El patrón MVC consta de 3 componentes los cuales son el Modelo que es la representación de los conjuntos de datos almacenados en tablas con los cuales se trabaja. La Vista, la cual es la interfaz que permite al usuario tener el punto de interacción con el sistema a través de formularios, tablas y comportamientos. Y por último el Controlador, el cual es el encargado de transmitir las peticiones al modelo enviadas desde la vista por el usuario, ya que este es el encargado de ejecutar la logia de negocio.



*Ilustración 1 Patrón Modelo Vista Controlador*

## **Framework de Desarrollo**

Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software.

Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto. (Wikipedia, 2013).

## **CodeIgniter**

Es un framework para el desarrollo de aplicaciones en php que utiliza el MVC. Permite a los programadores Web mejorar la forma de trabajar y hacerlo a mayor velocidad. Al igual que cualquier framework está pensado para gente que tiene un dominio, al menos medio, del lenguaje de programación PHP. Siempre hay que controlar PHP «a pelo» para empezar a trabajar de forma eficiente con este framework (o cualquier otro). (Fontan, 2019).

## **CRUDS**

Es un acrónimo en ingles de (Create, Read, Update and Delete), que son las funciones básicas de persistencia de la base de datos (Crear, leer, actualizar y eliminar).

**Scaffolding**

El andamiaje es una técnica compatible con algunos marcos de modelo, vista y controlador, en los que el programador puede especificar cómo se puede usar la base de datos de la aplicación. El compilador o el marco de trabajo utiliza esta especificación, junto con plantillas de código predefinidas, para generar el código final que la aplicación puede usar para crear, leer, actualizar y eliminar entradas de la base de datos, tratando efectivamente las plantillas como un "andamio" sobre el cual construir Una aplicación más potente.

## **5. DISEÑO METODOLÓGICO**

### **5.1. TIPO DE INVESTIGACIÓN**

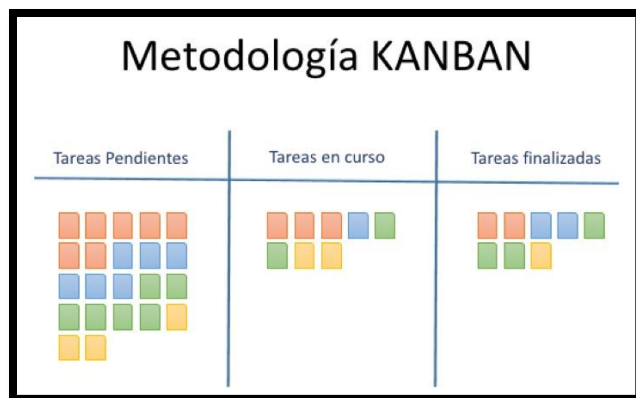
Cada vez que se realiza una investigación, es necesario tener presente la forma como se va a abordar la problemática, ya que es el punto de partida para poder dar un resultado adecuado. Para esto es necesario tener presente el tipo de investigación que se va a utilizar y así mismo clasificarlo de acuerdo con su función. Teniendo presente lo anteriormente expresado, el objetivo de este proyecto es generar un producto tecnológico y se abordara bajo una clasificación según su propósito, así que se utilizara una investigación aplicada en la subcategoría de investigación aplicada tecnológica, la cual sirve para generar conocimientos que se puedan poner en práctica en el sector productivo, con el fin de impulsar un impacto positivo en la vida cotidiana, (significados.com, 2019).

### **5.2. MÉTODO**

Las metodologías de desarrollo ágiles más usadas son SCRUM, eXtreme Programming, Kanban, Scrumban, Lean, Feature-Driven Development y est-Driven Development entre otras. Teniendo presente lo anterior, se puede deducir que existen muchas metodologías de desarrollo ágiles, pero ¿cuál es la adecuada para poder realizar el proyecto planteado? Para nuestro caso usaremos la metodología KANBAN, la cual tiene sus orígenes a finales de la década de los cuarenta y principios de los cincuenta empleados por Taiichi Onho en los procesos de producción que coordinaban en una cadena de montaje en Toyota. Kanban (palabra japonesa que se traduce como tablero o tarjeta de señalización) es una metodología que permite gestionar el trabajo intelectual,



teniendo énfasis en las entregas justo a tiempo mientras no se sobrecarguen los miembros del equipo.



*Ilustración 2 Tablero metodología KANBAN*

Teniendo presente las metodologías de desarrollo ágiles, se empleará para este proyecto la metodología KANBAN, la cual permite realizar un seguimiento de modo visual a través de tarjetas, donde se utiliza un tablero para representar las actividades por realizar, las actividades en proceso y las actividades ya realizadas. Con esta metodología se obtienen excelentes resultados en materia de gestión de proyectos ya que permite estimular el rendimiento del grupo de trabajo, mantener de forma organizada y simple las actividades a realizar, obtener una distribución equitativa de las tareas, además de poder ser asimilada de una forma sencilla y rápida.

Teniendo presente la metodología, se definirá como herramienta para análisis y control Azure DevOps, la cual no solo permite administrar a través de tableros KANBAN la actividad, sino que también nos suministra el alojamiento del código fuente en

repositorios Git. Todo esto pensado para poder obtener estadísticas y tiempos de desarrollos muy rápidos con entregas en cortos periodos de tiempo.

Después de haber identificado la metodología y la herramienta a utilizar, se definen cada una de las etapas del proyecto, las cuales se definirán bajo el sprint 1, iteración 1. En esta iteración se crearán las historias de usuarios, las cuales a su vez contendrán las actividades dispuestas en el cronograma, estas historias de usuario serán las etapas que deberán seguir para poder ejecutar en su totalidad el proyecto.

### **Identificación de la estructura del Framework CodeIgniter**

Permite conocer de manera detallada la estructura del framework CodeIgniter, carpetas y subcarpetas, recursos y configuraciones iniciales.

### **Análisis del desarrollo a realizar**

Analizar cada uno de los requerimientos funcionales y no funcionales del sistema para poder ejecutar la codificación.

### **Diseño del Modelo Entidad Relación de la base de Datos**

Identificar el motor de base de datos a utilizar, desarrollo de tablas bases, e inserción inicial de datos base.

### **Configuración inicial del proyecto**

Ejecutar las actividades de configuración de cadena de base de datos, recursos de carpetas, ruta base, variables y constantes e integración de plantilla AdminLTE.

### **Desarrollo de funcionalidades**

Realizar la ejecución de codificación del sistema de seguridad de la aplicación, así como codificación del núcleo del proyecto para la autogeneración de código a través de la lectura de la estructura de la base de datos.

### **Ejecución de pruebas**

Realizar pruebas de funcionalidad, seguridad, estrés al producto desarrollado con el objetivo de encontrar anomalías y así mismo realizar una autoalimentación y subsanar lo encontrado.

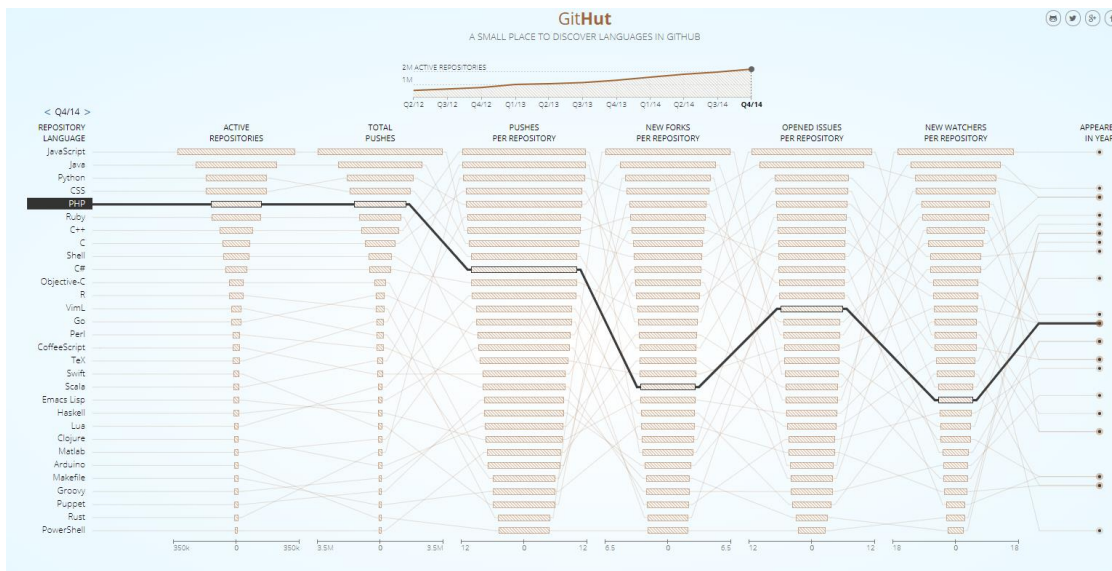
## **5.3. HIPÓTESIS**

Un autogenerador de código para el framework CodeIgniter permite una reducción mayor al 50 por ciento en tiempos de entrega de un proyecto tecnológico que utiliza una metodología agile.

## **5.4. VARIABLES E INDICADORES**

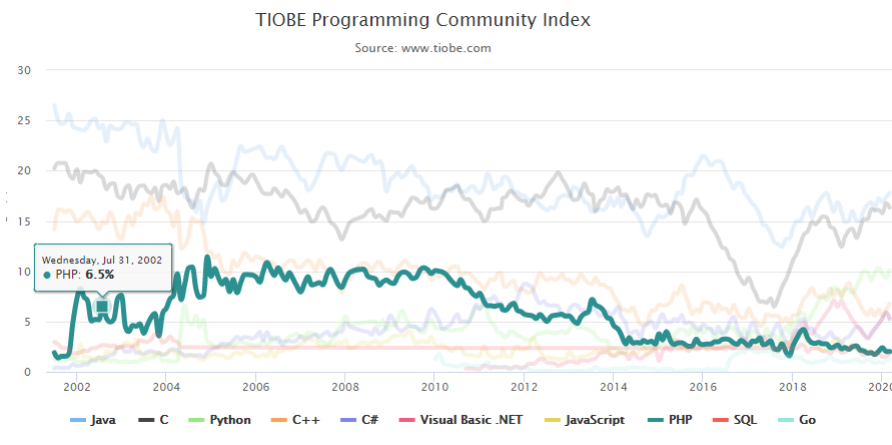
Para llevar a cabo este proyecto se tuvo presente el ranking de lenguajes de programación más usados en GitHub, el cual es el repositorio Git mas grande y usado en

el mundo donde se posiciona a PHP como el 5 lenguaje de programación mas usado, como lo muestran las siguientes graficas:



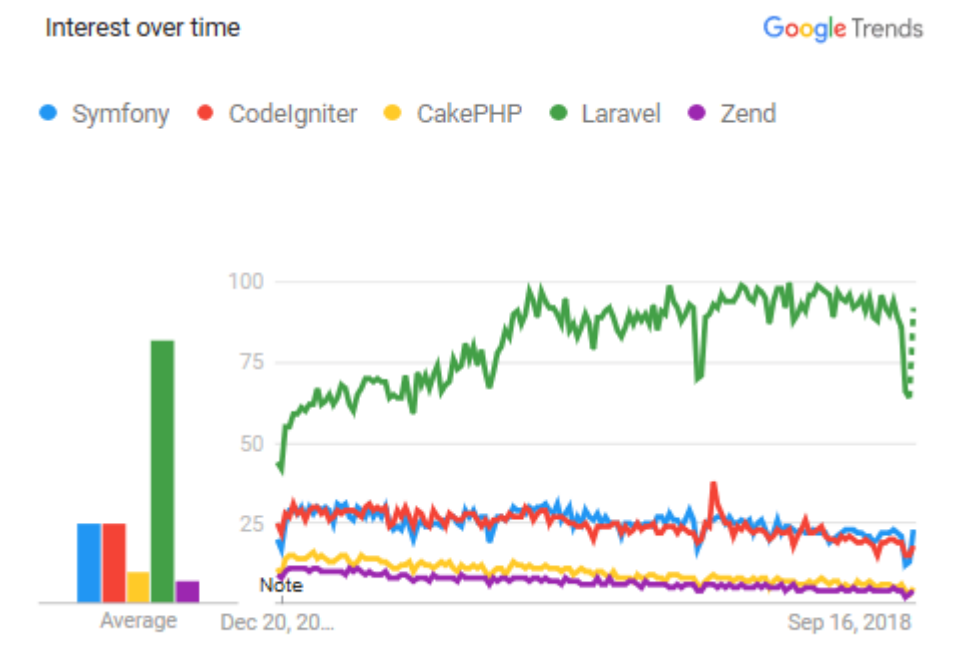
**Ilustración 3 Ranking Github Lenguajes**

De igual manera se utiliza el ranking publicado por TIOBE donde se posiciona a PHP como el 8 lenguaje de programación mas utilizado en el mundo, como lo muestra la siguiente gráfica:



**Ilustración 4 Ranking TIOBE Lenguajes**

También se tiene presente la grafica de frameworks de desarrollos mas usados en el mundo según Goolge Trends:



*Ilustración 5 Ranking Frameworks PHP Google Trends*

## 5.5. POBLACIÓN O MUESTRA

La población es el conjunto total de individuos, objetos o medidas que poseen algunas características comunes observables en un lugar y en un momento determinado, (Wigodski, 2010). Teniendo claro este concepto, para este proyecto la población son los desarrolladores PHP que usan CodeIgniter como framework de desarrollo. Como la muestra es un subconjunto representativo de la población, se tomarán los desarrolladores

PHP que usan CodeIgniter como framework de desarrollo en la Oficina de Sistemas del Ministerio de Comercio Industria y Turismo en el año 2018.

## **5.6. INSTRUMENTO Y TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN**

El instrumento de recolección es la herramienta utilizada para poder realizar el levantamiento de información, para este caso se utilizó la entrevista como instrumento en la cual se realiza una charla personalmente con una serie de entrevistados, específicamente desarrolladores PHP, donde se analiza la autogeneración de scaffolding con el framework CodeIgniter.

### **Resultado de las entrevistas**

Como resultado se obtuvieron las siguientes observaciones:

- CodeIgniter es un framework de desarrollo PHP con una curvatura de aprendizaje muy corta, Por lo cual es muy ágil a la hora de desarrollar en PHP.
- CodeIgniter no cuenta un autogenerador de código que facilite el desarrollo de aplicaciones web.
- Muchos desarrolladores que han utilizado CodeIgniter han migrado a otros framework ya que no cuenta con fuertes herramientas de autogeneración de código como lo tienen otros framework PHP tales como Yii, Phalcon, Symfony, CakePHP entre otros.

- Otros framework PHP tienen buenas herramientas pero algunas son muy engorrosas ya que se debe utilizar la consola de comando para poder usarlas, lo que quita tiempo a un desarrollador nivel junior.

## 6. ESQUEMA TEMATICO

### 6.1. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

Los requerimientos funcionales y no funcionales de un sistema de información permiten al grupo de desarrollo poder entender el sistema tanto en su estructura funcional, como en sus comportamientos y entorno visual. Para este proyecto se definió como nomenclatura de requerimiento funcional RFU-0X, y para los requerimientos no funcionales RNF-0X.

Listado de Requerimientos Funcionales

*Tabla 1 Listado de requerimientos funcionales*

<b>Código</b>	<b>Descripción</b>
RF-01	El sistema debe permitir la administración (Listar, lectura, creación, edición y eliminación) de roles del sistema.
RF-02	El sistema debe permitir la administración (Listar, lectura, creación, edición y eliminación) de los usuarios del sistema.
RF-03	El sistema debe permitir la administración (Listar, lectura, creación, edición y eliminación) de los menús o rutas del sistema.
RF-04	El sistema debe permitir la administración de permisos de los roles sobre los menús o rutas del sistema.



RF-05	El sistema debe permitir obtener la información de las tablas de la base de datos como son campos, tipos de campos, amplitud, llaves primarias, llaves foráneas entre otra información.
RF-06	El sistema debe permitir crear los modelos a través de la información obtenida en el requerimiento RF-05.
RF-07	El sistema debe permitir crear los controladores a través de la información obtenida en el requerimiento RF-05.
RF-08	El sistema debe permitir crear las vistas (agregar, editar, índice y ver) a través de la información obtenida en el requerimiento RF-05.
RF-09	El sistema debe permitir la creación automática de las tablas base en la base de datos en la primera llamada al servidor.
RF-10	El sistema debe permitir la creación de un usuario administrador en la primera llamada al servidor.

#### Listado de Requerimientos No Funcionales

**Tabla 2** Listado de requerimientos no funcionales

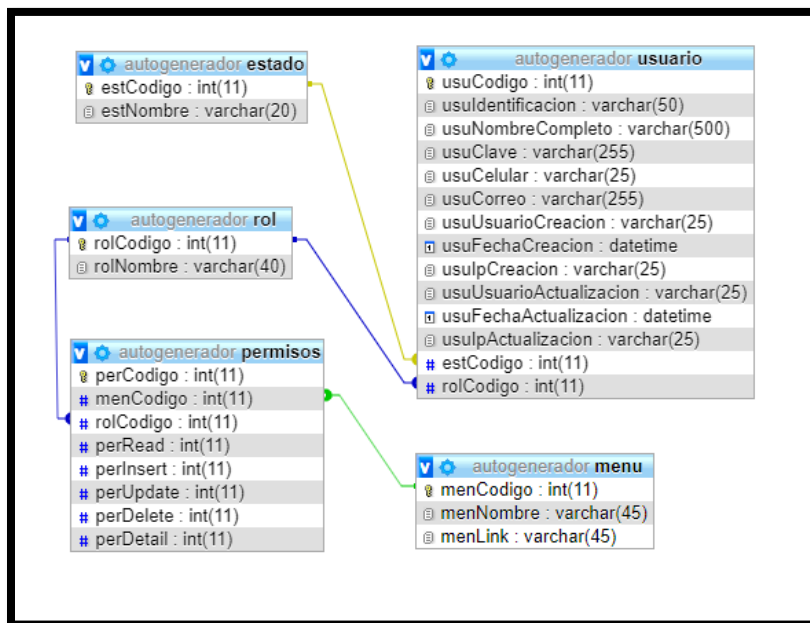
<b>Código</b>	<b>Descripción</b>
RNF-01	El sistema debe permitir el acceso solo a través de usuario y contraseña.
RNF-02	El sistema debe permitir cerrar sesión a los usuarios logueados.
RNF-03	El sistema debe permitir al usuario cambiar la contraseña.

RNF-04	El sistema debe permitir recuperar la contraseña a través de correo electrónico.
RNF-05	El sistema debe permitir su visualización en dispositivos de pantalla reducida como tabletas y smartphones.

## 6.2. DEFINICIÓN DE LA BASE DE DATOS

De la mano con el desarrollo del sistema de información, es necesarios definir las tablas base necesarias mínimas para que el sistema pueda funcionar adecuadamente, por lo cual, a través de los requerimientos funcionales y no funcionales se realiza un proceso de análisis que permite la estructuración de la base de datos.

Modelo Entidad Relación de la base de datos (MER)



*Ilustración 6 Modelo Entidad Relación de la base de datos.*

## Diccionario de Datos

*estado***Tabla 3 Descripción de tabla estado**

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
estCodigo ( <i>Primaria</i> )	int(11)	No			Identificador único del registro
estNombre	varchar(20)	No			Nombre del estado

*menú***Tabla 4 Descripción de tabla menú**

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
menCodigo ( <i>Primaria</i> )	int(11)	No			Identificador único del registro
menNombre	varchar(45)	Sí	<i>NULL</i>		Nombre del menú

menLink	varchar(45)	Sí	NULL		Enlace del menú
---------	-------------	----	------	--	-----------------

*permisos*

*Tabla 5 Descripción de tabla permisos*

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
perCodigo ( <i>Primaria</i> )	int(11)	No			Identificador único del registro
menCodigo	int(11)	Sí	NULL	menú -> menCodigo	Identificador foráneo de relación con tabla menú
rolCodigo	int(11)	Sí	NULL	rol -> rolCodigo	Identificador foráneo de relación con tabla rol
perRead	int(11)	Sí	NULL		Permiso de lectura

perInsert	int(11)	Sí	NULL		Permiso de inserción
perUpdate	int(11)	Sí	NULL		Permiso de edición
perDelete	int(11)	Sí	NULL		Permiso de eliminación
perDetail	int(11)	No			Permiso de ver detalle

### *rol*

**Tabla 6 Descripción de tabla rol**

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
rolCodigo ( <i>Primaria</i> )	int(11)	No			Identificador único del registro
rolNombre	varchar(40)	No			Nombre del rol

*usuario**Tabla 7 Descripción de tabla usuario*

Columna	Tipo	Nul o	Predeterminado	Enlaces a	Comentarios
usuCodigo ( <i>Primaria</i> )	int(11)	No			Identificador único del registro
usuIdentificacion	varchar(50)	No			Identificació n del usuario
usuNombreCompleto	varchar(500 )	No			Nombre del usuario
usuClave	varchar(255 )	No			Contraseña del usuario
usuCelular	varchar(25)	No			Numero de celular del usuario
usuCorreo	varchar(255 )	No			Correo electrónico del usuario

usuUsuarioCreacion	varchar(25)	No			Usuario que creo al usuario
usuFechaCreacion	datetime	No	0000-00-00 00:00:00		Fecha de alta del usuario
usuIpCreacion	varchar(25)	No			Ip de maquina donde se creó usuario
usuUsuarioActualizacion	varchar(25)	Sí	<i>NULL</i>		Ultimo usuario que actualizo información de usuario
usuFechaActualizacion	datetime	Sí	<i>NULL</i>		Ultima fecha de actualización de información de usuario

usuIpActualizacion	varchar(25)	Sí	<i>NULL</i>		Últimas Ip de maquina donde se actualizo información del usuario
estCodigo	int(11)	No		estado -> estCodigo	Identificador foráneo de relación con tabla estado
rolCodigo	int(11)	No		rol -> rolCodigo	Identificador foráneo de relación con tabla rol

### 6.3. PRERREQUISITOS PARA DESARROLLO

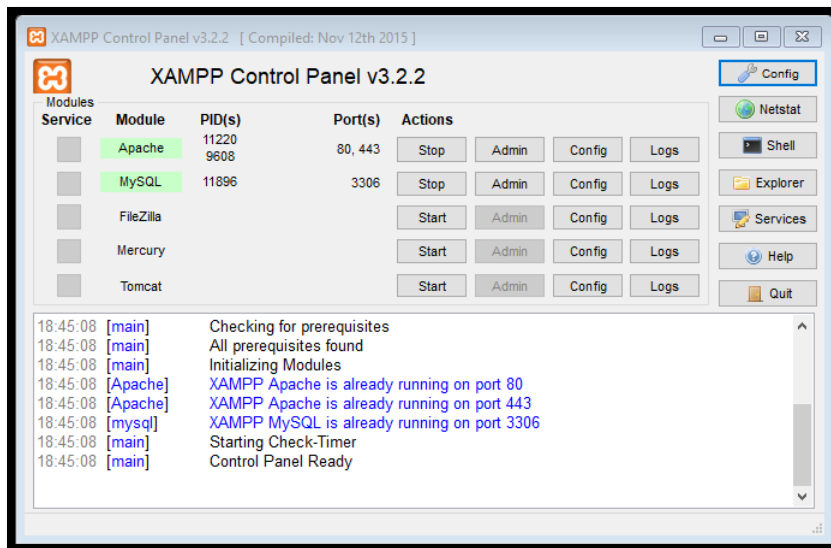
Para poder iniciar el desarrollo y la configuración del proyecto es necesario haber descargado e instalado una serie de herramientas los cuales se describirán a continuación:

1. Descargamos el instalador de Visual Studio Code desde el enlace <https://code.visualstudio.com/> y ejecutamos.



2. Descargamos el instalador de XAMPP desde el enlace

<https://www.apachefriends.org/es/index.html> y ejecutamos. Después de esto buscamos el panel de control de XAMPP en el menú de navegación y desde el panel activamos el servidor web apache y la base de datos MySQL.



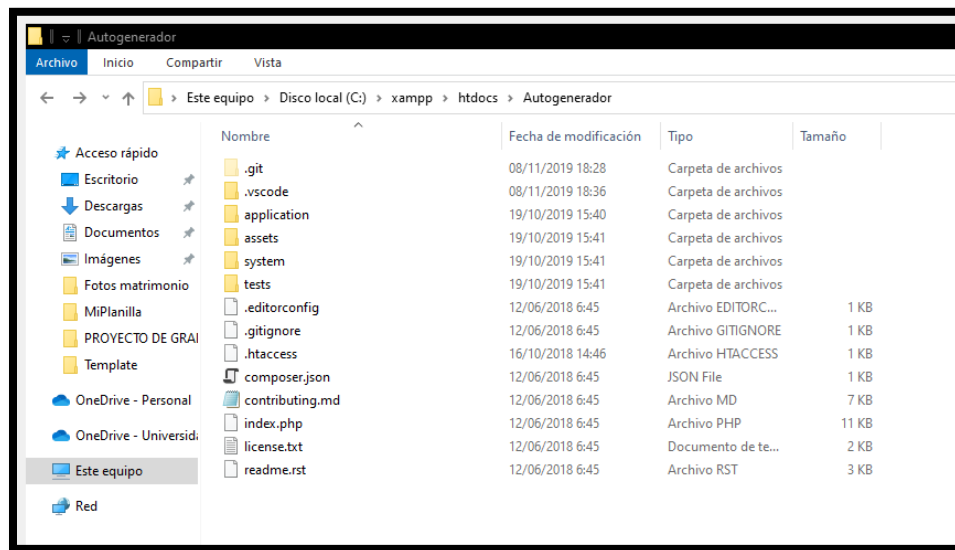
*Ilustración 7 Panel de administración XAMPP*

3. Descargamos el código base del framework desde el enlace

<https://api.github.com/repos/bcit-ci/CodeIgniter/zipball/3.1.11> y descomprimos.

Seguidamente vamos a la ruta del servidor apache instalado por XAMPP

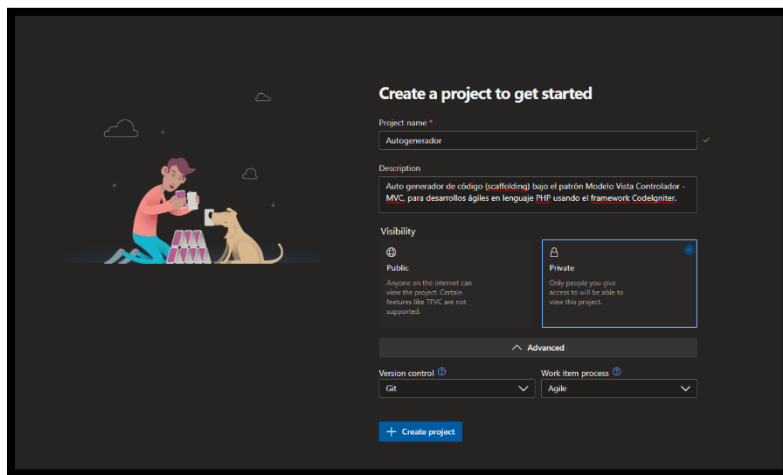
(C:\xampp\htdocs) creamos una carpeta con el nombre del proyecto y dentro de esta carpeta copiamos los archivos descomprimidos.



*Ilustración 8 Carpeta raíz del proyecto*

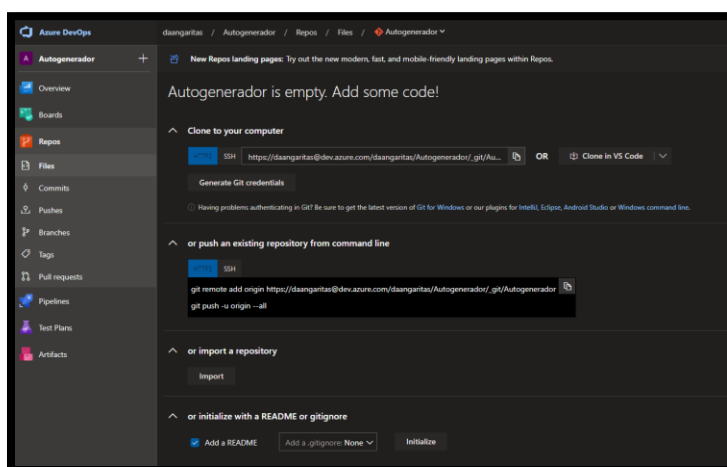
## 6.4. CONFIGURACIÓN DE AZURE DEVOPS

Azure DevOps es la herramienta utilizada para poder realizar el control de la ejecución del proyecto a través de metodologías ágiles. A continuación, se describirán los pasos para la configuración del entorno de trabajo.



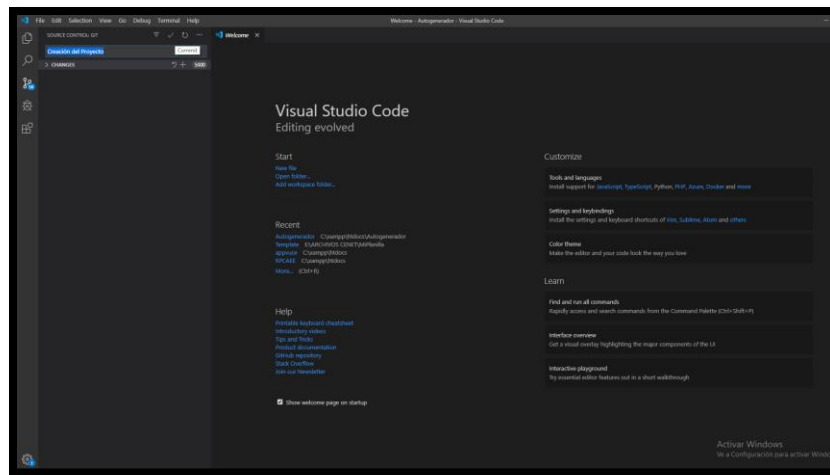
### *Ilustración 9 Creación de proyecto Azure DevOps*

Después de haber realizado la creación del espacio de trabajo, vamos al menú lateral y seleccionamos el enlace “Repos”, después de esto damos clic en el botón “Clone in VS Code”, para poder clonar el repositorio, damos permiso para poder abrir Visual Studio Code y seleccionamos la capeta raíz donde copiamos los archivos del framework CodeIgniter.



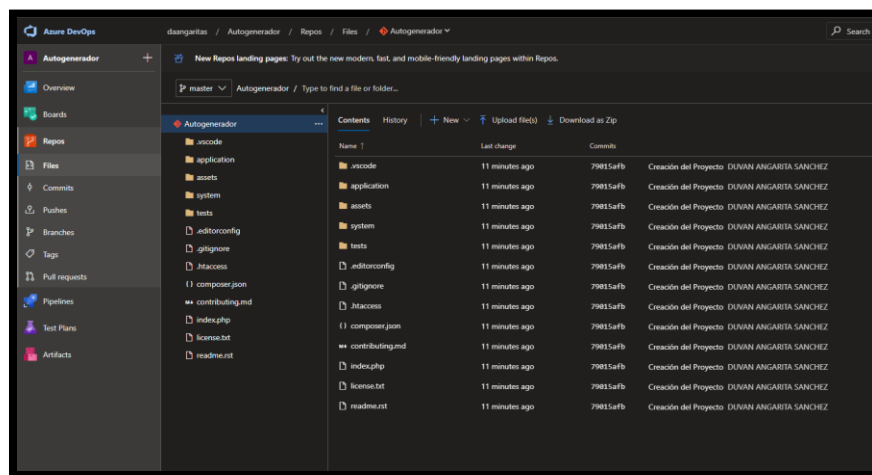
### *Ilustración 10 Repositorio DevOps*

Después de esto, en Visual Studio en el menú lateral, seleccionamos el Source Control, Digitamos un comentario y realizamos “commit” al proyecto.



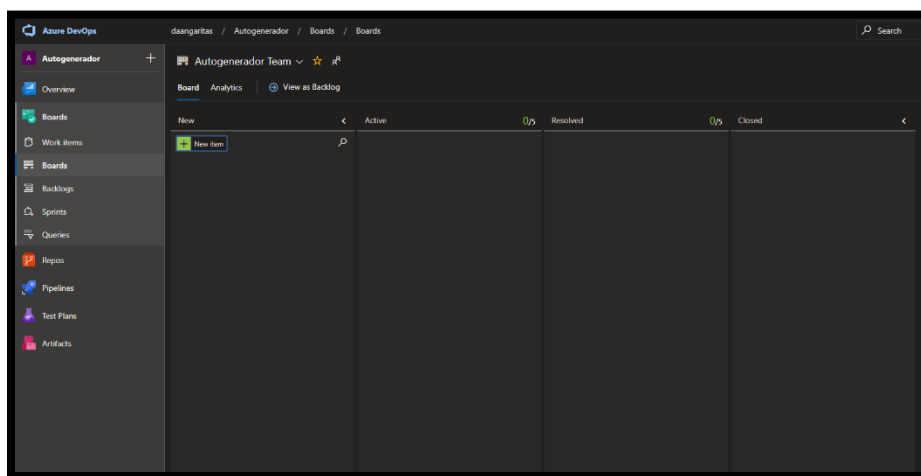
*Ilustración 11 Visual Studio Code Commit*

Seguidamente realizamos un “Push” al origen para cargar los archivos al repositorio.



*Ilustración 12 Repositorio GIT del proyecto*

Para configurar las actividades en el tablero KANBAN, es necesario ir a Azure DevOps, damos clic en el menú “Boards” y seleccionamos el submenú “Boards”, allí se nos mostrara el tablero vacío de la siguiente manera:



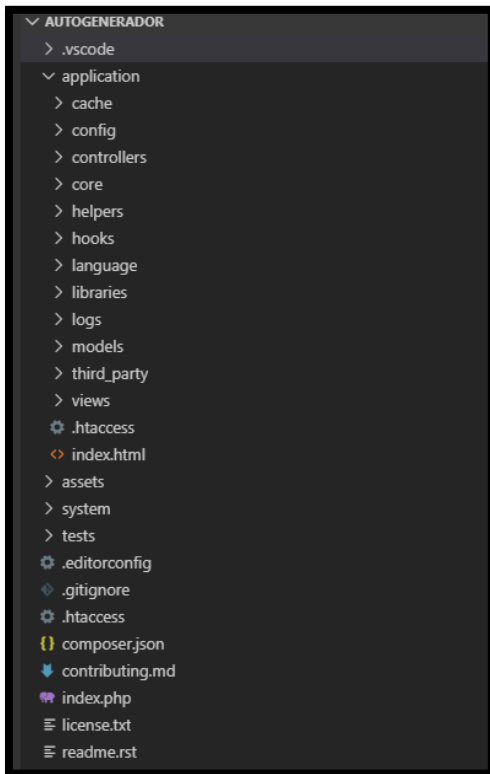
### ***Ilustración 13 Tablero KANBAN Azure DevOps***

Para agregar cada una de las actividades a realizar, se debe dar clic en el botón “New Ítem” y se diligencia la descripción. Estas actividades son las que se definieron en el cronograma de actividades.

## **6.5. ESTRUCTURA DEL PROYECTO**

Después de haber definido la utilización del framework CodeIgniter para lenguaje PHP, se debe tener presente el patrón Modelo Vista Controlador, el cual es usado en este framework para organizar la arquitectura del proyecto. La organización de las carpetas está estructurada para poder identificar cada uno de los recursos dependiendo de su funcionalidad en el sistema.

A continuación, se mostrará la forma en la que se encuentra estructurado el proyecto:



### *Ilustración 14 Estructura de proyecto CodeIgniter*

Las tres carpetas fundamentales del proyecto son:

**system:** En esta carpeta se encuentra el núcleo del framework CodeIgniter, es importante tener presente que no se puede manipular ninguno de los archivos de esta carpeta.

**assets:** En esta carpeta reposan los archivos de manipulación de entorno de usuario del sistema, tales como librerías de estilos, componentes JavaScript, entre otros.

**applications:** En esta carpeta es donde se lleva a cabo el desarrollo del proyecto como son controladores, modelos y vistas, así como configuraciones del proyecto y manipulación de librerías propias y de terceros agregadas al desarrollo. Esta carpeta a su vez se divide en las siguientes carpetas:

**cache:** En esta carpeta se almacena la información cache de la aplicación.

**config:** en esta carpeta están contenidas las clases de configuración del sistema como son las conexiones a base de datos, el uso de librerías, helpers, lenguajes, enrutamientos entre otros.

**controllers:** En esta carpeta están contenidos los controladores los cuales son las clases encargadas de determinar cómo deben ser manejadas las solicitudes http.

**core:** En esta carpeta se pueden disponer recursos para personalizar o modificar clases del núcleo del framework.

**helpers:** En esta carpeta se almacenan recursos que sirven como ayudantes al proceso del sistema creados por el usuario o por terceros.

**hooks:** En esta carpeta están contenidos recursos que son utilizados como tareas a ejecutar en un determinado momento, como son pre o post system (al principio de la ejecución del sistema, sin haber cargado nada o al finalizar el procedimiento de cargado del sistema), pre o post controller (antes o después de cargar el controlador) entre otros.

**language:** En esta carpeta se disponen los recursos de idiomas creados para cada proyecto.

**libraries:** En esta carpeta se disponen las librerías específicas implementadas en el desarrollo como son la integración de plantillas, la implementación de autorización o el consumo de servicios SOAP entre otras.

**logs:** En esta carpeta se almacenan los logs de auditoría configurados en el sistema.

**models:** En esta carpeta están contenidas las clases que interactúan con la base de datos llamadas modelos.

**Third\_part:** En esta carpeta se disponen las librerías específicas creadas por terceras partes.

**views:** En esta carpeta están contenidas las páginas web o fragmentos de ellas que permiten la interacción con el usuario.

## 6.6. DESARROLLO Y CODIFICACIÓN DEL PROYECTO

El Autogenerador de código está distribuido en tres partes fundamentales como son modelo, vista y controladores, en ello recae cada una de las funciones específicas para la creación de tablas y recursos del sistema.

Modelos

### *Listar tablas de la base de datos*

```
public function listarTablas()
{
    $db = $this->db->database;
    $resultado = $this->db->
>query("SELECT TABLE_NAME AS TABLAS FROM INFORMATION_SCHEMA.TABLES WHERE T
ABLE_SCHEMA = '".$db."'");

    return $resultado->result();
}
```

### *Listar campos de las tablas*

```
public function listarCapos($nombreTabla)
{
```



```

    $db = $this->db->database;
    $resultado = $this->db->query("DESCRIBE ".$nombreTabla);

    return $resultado->result();
}

```

### *Listar llaves foráneas de la base de datos*

```

public function listarForaneas($nombreTabla)
{
    $db = $this->db->database;
    $resultado = $this->db->
>query("SELECT TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME, REFERENCED_TABLE_
NAME, REFERENCED_COLUMN_NAME FROM INFORMATION_SCHEMA.KEY_COLUMN_USAGE WHER
E TABLE_NAME = '".$nombreTabla.'" GROUP BY CONSTRAINT_NAME");

    return $resultado->result();
}

```

### *Listar tablas dependientes de la base de datos*

```

public function listarDependientes($nombreTabla)
{
    $db = $this->db->database;
    $resultado = $this->db->
>query("SELECT TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME, REFERENCED_TABLE_
NAME, REFERENCED_COLUMN_NAME FROM INFORMATION_SCHEMA.KEY_COLUMN_USAGE WHER
E REFERENCED_TABLE_NAME = '".$nombreTabla.'"");

    return $resultado->result();
}

```

### *Crear tablas base*

```

public function CrearModelos()
{
    if(count($this->listarTablas()) == 0)
    {
        $db = $this->db->database;
    }
}

```

```

$this->db->query("CREATE TABLE IF NOT EXISTS `estado` (
  `estCodigo` int(11) NOT NULL,
  `estNombre` varchar(20) COLLATE utf8_spanish_ci NOT NULL,
  PRIMARY KEY (`estCodigo`)
)");

$this->db->query("CREATE TABLE IF NOT EXISTS `menu` (
  `menCodigo` int(11) NOT NULL AUTO_INCREMENT,
  `menNombre` varchar(45) COLLATE utf8_spanish_ci DEFAULT NU
LL,
  `menLink` varchar(45) COLLATE utf8_spanish_ci DEFAULT NULL
,
  PRIMARY KEY (`menCodigo`)
)");

$this->db->query("CREATE TABLE IF NOT EXISTS `rol` (
  `rolCodigo` int(11) NOT NULL AUTO_INCREMENT,
  `rolNombre` varchar(40) COLLATE utf8_spanish_ci NOT NULL,
  PRIMARY KEY (`rolCodigo`)
)");

$this->db->query("CREATE TABLE IF NOT EXISTS `usuario` (
  `usuCodigo` int(11) NOT NULL AUTO_INCREMENT,
  `usuIdentificacion` varchar(50) COLLATE utf8_spanish_ci NO
T NULL,
  `usuNombreCompleto` varchar(500) COLLATE utf8_spanish_ci N
OT NULL,
  `usuClave` varchar(255) COLLATE utf8_spanish_ci NOT NULL,
  `usuCelular` varchar(25) COLLATE utf8_spanish_ci NOT NULL,
  `usuCorreo` varchar(255) COLLATE utf8_spanish_ci NOT NULL,
  `usuUsuarioCreacion` varchar(25) COLLATE utf8_spanish_ci N
OT NULL,
  `usuFechaCreacion` datetime NOT NULL DEFAULT '0000-00-
00 00:00:00' ON UPDATE CURRENT_TIMESTAMP,
  `usuIpCreacion` varchar(25) COLLATE utf8_spanish_ci NOT NU
LL,
  `usuUsuarioActualizacion` varchar(25) COLLATE utf8_spanish
_ci DEFAULT NULL,
  `usuFechaActualizacion` datetime DEFAULT NULL ON UPDATE CU
RRENT_TIMESTAMP,
  `usuIpActualizacion` varchar(25) COLLATE utf8_spanish_ci D
EFAULT NULL,
  `estCodigo` int(11) NOT NULL,

```

```

        `rolCodigo` int(11) NOT NULL,
        PRIMARY KEY (`usuCodigo`),
        KEY `estCodigo` (`estCodigo`),
        KEY `rolCodigo` (`rolCodigo`),
        CONSTRAINT `USUARIO_ibfk_1` FOREIGN KEY (`estCodigo`) REFERENCE
RENCES `estado` (`estCodigo`),
        CONSTRAINT `USUARIO_ibfk_2` FOREIGN KEY (`rolCodigo`) REFERENCE
RENCES `rol` (`rolCodigo`)
    );

$this->db->query("CREATE TABLE IF NOT EXISTS `permisos` (
    `perCodigo` int(11) NOT NULL AUTO_INCREMENT,
    `menCodigo` int(11) DEFAULT NULL,
    `rolCodigo` int(11) DEFAULT NULL,
    `perRead` int(11) DEFAULT NULL,
    `perInsert` int(11) DEFAULT NULL,
    `perUpdate` int(11) DEFAULT NULL,
    `perDelete` int(11) DEFAULT NULL,
    `perDetail` int(11) NOT NULL,
    PRIMARY KEY (`perCodigo`),
    KEY `fk_menus_idx` (`menCodigo`),
    KEY `fk_rol_idx` (`rolCodigo`),
    CONSTRAINT `fk_menus` FOREIGN KEY (`menCodigo`) REFERENCES
`menu` (`menCodigo`) ON DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT `fk_rol` FOREIGN KEY (`rolCodigo`) REFERENCES `
rol` (`rolCodigo`) ON DELETE NO ACTION ON UPDATE NO ACTION
    );

$this->db-
>query("INSERT INTO `estado` (`estCodigo`, `estNombre`) VALUES (1, 'Activo
'), (2, 'Inactivo'), (3, 'Eliminado)");

$this->db-
>query("INSERT INTO `rol` (`rolCodigo`, `rolNombre`) VALUES (1, 'Administr
ador)");

$this->db-
>query("INSERT INTO `usuario` (`usuCodigo`, `usuIdentificacion`, `usuNombr
eCompleto`, `usuClave`, `usuCelular`, `usuCorreo`, `usuUsuarioCreacion`, `
usuFechaCreacion`, `usuIpCreacion`, `usuUsuarioActualizacion`, `usuFechaAc
tualizacion`, `usuIpActualizacion`, `estCodigo`, `rolCodigo`) VALUES

```

```

        (1, '123456789', 'Administrador', '40bd001563085fc35165329ea1f
f5c5ecbdbbbeeef', '3100000000', 'admin@admin.com', 'admin', '2019-09-
01 00:00:00', '', NULL, '2019-09-01 00:00:00', NULL, 1, 1)");

    }

    return true;
}

```

Controlador

### *Constructor*

```

public function __construct()
{
    parent::__construct();

    if (!$this->session->userdata('login') || $this->session-
>userdata('rol') != 'Administrador')
    {
        redirect(base_url());
    }

    $this->load->model('Scaffolding_model');
}

```

### *Llamado a página inicial (índice)*

```

public function index ()
{
    $data = array(
        'tablas' => $this->Scaffolding_model->listarTablas()
    );
    $this->template->set('title', 'Scaffolding');
    $this->template->set('subtitle', 'Scaffolding');
    $this->template-
>load('default_layout', 'contents', 'scaffolding/index', $data);
}

```

### *Llamado a página de resumen*

```

public function crear()
{
    $tabla = $this->input->post('tabla');

    $data = array(
        'campos' => $this->Scaffolding_model->listarCapos($tabla),
        'foraneas' => $this->Scaffolding_model-
>listarForaneas($tabla),
        'dependientes' => $this->Scaffolding_model-
>listarDependientes($tabla),
        'tabla' => $tabla
    );

    $this->template->set('title', 'Scaffolding Resumen');
    $this->template->set('subtitle', 'Scaffolding');
    $this->template-
>load('default_layout', 'contents', 'scaffolding/resumen', $data);
}

```

### *Llamado a funciones de generación de modelos, vistas y controladores*

```

public function generar()
{
    $tabla = $this->input->post('tabla');
    $nombreComponente = ucfirst(str_replace("auto_", "", $tabla));

    $campos = $this->Scaffolding_model->listarCapos($tabla);
    $llaves = $this->Scaffolding_model->listarForaneas($tabla);
    $dependientes = $this->Scaffolding_model-
>listarDependientes($tabla);

    $rutaModelo = $_SERVER['DOCUMENT_ROOT']."/Autogenerador/applicatio
n/models";
    $rutaVista = $_SERVER['DOCUMENT_ROOT']."/Autogenerador/application
/views";
    $rutaControlador = $_SERVER['DOCUMENT_ROOT']."/Autogenerador/appli
cation/controllers";
}

```

```

        $this->
>generarModelo($tabla, $nombreComponente, $rutaModelo, $campos, $llaves);

        $this->
>generarControlador($tabla, $nombreComponente, $rutaControlador, $campos,
$llaves, $dependientes);

        $this->
>generarVistas($tabla, $nombreComponente, $rutaVista, $campos, $llaves, $d
ependientes);

        redirect(base_url().$nombreComponente);
    }

```

### *Identificación de llave foránea*

```

public function esForanea($campo, $llaves)
{
    foreach($llaves as $llave)
    {
        if($llave->REFERENCED_COLUMN_NAME == $campo->Field)
        {
            return true;
        }
    }

    return false;
}

```

### *Obtener nombre de tabla relacionada*

```

public function obtenerTablaForanea($campo, $llaves)
{
    foreach($llaves as $llave)
    {
        if($llave->REFERENCED_COLUMN_NAME == $campo->Field)
        {
            return $llave->REFERENCED_TABLE_NAME;
        }
    }
}

```

### *Verificar si campo es llave primaria*

```
public function llavePrimaria($campos)
{
    foreach ($campos as $campo) {
        if($campo->Key == "PRI")
        {
            return $campo->Field;
        }
    }
}
```

### *Obtener Alias de la tabla*

```
public function obtenerAlias($tabla)
{
    $alias = str_replace("auto_", "", $tabla);
    return $alias;
}
```

### *Validar campos de auditoria*

```
public function validarCampo($campo)
{
    if(($campo->Key == 'PRI' && $campo->Extra == 'auto_increment') ||
    (strpos($campo->Field, 'UsuarioActualizacion') ||
    strpos($campo->Field, 'FechaActualizacion') ||
    strpos($campo->Field, 'IpActualizacion') ||
    strpos($campo->Field, 'UsuarioCreacion') ||
    strpos($campo->Field, 'FechaCreacion') ||
    strpos($campo->Field, 'IpCreacion')))
    {
        return false;
    }
    else
    {
        return true;
    }
}
```

*Generar modelos*

```

public function generarModelo($tabla, $componente, $ruta, $campos, $llaves)
{
    $cantidadLlaves = count($llaves);
    $contador = 1;
    $content = "<?php \n";
    $content .= "defined('BASEPATH') OR exit('No direct script access
allowed');\n";
    $content .= " \n";
    $content .= "class ".$componente."_model extends CI_Model\n";
    $content .= "{\n";
    $content .= "    public function listar()\n";
    $content .= "    {\n";

    if($cantidadLlaves > 1)
    {
        $content .= "        ".$this->db->select('s.*', ";
        foreach ($llaves as $llave) {
            if(strpos($llave->CONSTRAINT_NAME, 'fk'))
            {
                $contador++;
                $alias = $this->obtenerAlias($llave->REFERENCED_TABLE_NAME);
                $camposTablaReferencia = $this->Scaffolding_model->listarCapos($llave->REFERENCED_TABLE_NAME);
                $content .= $alias." ".$camposTablaReferencia[1]->Field." as ".$alias;
                if($contador < $cantidadLlaves)
                {
                    $content .= ", ";
                }
            }
        }
        $contador = 1;
        $content .= "    );\n";
        $content .= "        ".$this->db->from('".$tabla." s');\n";
        foreach ($llaves as $llave) {
            if(strpos($llave->CONSTRAINT_NAME, 'fk'))
            {

```



```

        $alias = $this->obtenerAlias($llave-
>REFERENCED_TABLE_NAME);
        $content .= "        $"."this->db->join('".'.$llave-
>REFERENCED_TABLE_NAME.'" ".$alias."', '"'.$alias.'".'".'$llave-
>REFERENCED_COLUMN_NAME.'" = s.'".'$llave->COLUMN_NAME.'"');\n";
    }
}
$content .= "        $"."this->db-
>where('s.estCodigo', 1);\n";
$content .= "\n";
$content .= "        $"."resultado = $"."this->db->get();\n";
}
else
{
    $content .= "\n";
    $content .= "        $"."resultado = $"."this->db-
>get('".'$tabla.'"');\n";
}

$content .= "\n";
$content .= "        return $"."resultado->result();\n";
$content .= "    }\n";
$content .= "\n";
$content .= "    public function guardar($"."data)\n";
$content .= "    {\n";
$content .= "        return $"."this->db-
>insert('".'$tabla.'"', $"."data");\n";
$content .= "    }\n";
$content .= "\n";
$content .= "    public function obtener($"."id)\n";
$content .= "    {\n";

if($cantidadLlaves > 1)
{
    $content .= "        $"."this->db->select('s.*, ";
    foreach ($llaves as $llave) {
        if(strpos($llave->CONSTRAINT_NAME, 'fk'))
        {
            $contador++;
            $alias = $this->obtenerAlias($llave-
>REFERENCED_TABLE_NAME);
            $camposTablaReferencia = $this->Scaffolding_model-
>listarCapos($llave->REFERENCED_TABLE_NAME);

```

```

        $content .= $alias.". ".$camposTablaReferencia[1]-
>Field." as ".$alias;
        if($contador < $cantidadLlaves)
        {
            $content .= ", ";
        }
    }
}
$contador = 1;
$content .= "');\n";
$content .= "        $"."this->db->from('".$tabla." s');\n";
foreach ($llaves as $llave) {
    if(strpos($llave->CONSTRAINT_NAME, 'fk'))
    {
        $alias = $this->obtenerAlias($llave-
>REFERENCED_TABLE_NAME);
        $content .= "        $"."this->db->join('".$llave-
>REFERENCED_TABLE_NAME." ".$alias."', '".$alias.'"."$llave-
>REFERENCED_COLUMN_NAME." = s."$llave->COLUMN_NAME."');\n";
    }
}

    foreach ($llaves as $llave) {
        if($llave->CONSTRAINT_NAME == "PRIMARY")
        {
            $content .= "        $"."this->db->where('s."$llave-
>COLUMN_NAME."', $"."id);\n";
        }
    }

    $content .= "\n";
    $content .= "        $"."resultado = $"."this->db->get();\n";
}
else
{
    foreach ($llaves as $llave) {
        if($llave->CONSTRAINT_NAME == "PRIMARY")
        {
            $content .= "        $"."this->db->where('".$llave-
>COLUMN_NAME."', $"."id);\n";
        }
    }
}
$content .= "\n";

```

```

        $content .= "        $"."resultado = $"."this->db-
>get('".$tabla."');\n";
    }
    $content .= "\n";
    $content .= "        return $"."resultado->row();\n";
    $content .= "    }\n";
    $content .= "\n";
    $content .= "    public function actualizar($"."id, $"."data)\n";
    $content .= "    {\n";
    foreach ($llaves as $llave) {
        if($llave->CONSTRAINT_NAME == "PRIMARY")
        {
            $content .= "        $"."this->db->where('".$llave-
>COLUMN_NAME."', $"."id);\n";
        }
    }
    $content .= "        return $"."this->db-
>update('".$tabla."', $"."data);\n";
    $content .= "    }\n";
    $content .= "}\n";
    $content .= "\n";
    $content .= "/* Fin del archivo ".$componente."_model.php */\n";

    $fp = fopen($ruta . "/" . $componente . "_model.php", "wb");
    fwrite($fp, $content);
    fclose($fp);
}

```

### *Generar controladores*

```

public function generarControlador($tabla, $componente, $ruta, $campos, $l
laves, $dependientes)
{
    $cantidadLlaves = count($llaves);
    $contador = 1;

    $content = "<?php \n";
    $content .= "defined('BASEPATH') OR exit('No direct script access
allowed');\n";
    $content .= "\n";
    $content .= "class ".$componente." extends CI_Controller\n";
    $content .= "{\n";

```

```

$content .= "    private $"."permisos;\n";
$content .= "    public function __construct()\n";
$content .= "    {\n";
$content .= "        parent::__construct();\n";
$content .= "\n";
$content .= "        $"."this->permisos = $"."this-
>backend_lib->control();\n";
$content .= "\n";
$content .= "        $"."this->load-
>model('".$componente."_model');\n";
    if($cantidadLlaves > 1)
    {
        foreach ($llaves as $llave) {
            if(strpos($llave->CONSTRAINT_NAME, 'fk'))
            {
                $contador++;
                $alias = $this->obtenerAlias($llave-
>REFERENCED_TABLE_NAME);
                $content .= "        $"."this->load-
>model('".ucfirst($alias)."_model');\n";
            }
        }
        $contador = 1;
    }
$content .= "    }\n";
$content .= "\n";
$content .= "    public function index()\n";
$content .= "    {\n";
$content .= "        $"."data = array(\n";
$content .= "            'permisos' => $"."this->permisos,\n";
$alias = $this->obtenerAlias($tabla);
$content .= "            '".$alias."' => $"."this-
>".$componente."_model->listar()\n";
$content .= "        );\n";
$content .= "        $"."this->template-
>set('title', 'Listado de ".$componente."');\n";
$content .= "        $"."this->template-
>set('subtitle', '".$componente."');\n";
$content .= "        $"."this->template-
>load('default_layout', 'contents' , '".$strtolower($componente)."/index',
$content .= "        $"."data);\n";
$content .= "    }\n";
$content .= "\n";

```

```

$content .= "    public function agregar()\n";
$content .= "    {\n";
$content .= "        $"."data = array(\n";
if($cantidadLlaves > 1)
{
    foreach ($llaves as $llave) {
        if(strpos($llave->CONSTRAINT_NAME, 'fk'))
        {
            $contador++;
            $alias = $this->obtenerAlias($llave->
>REFERENCED_TABLE_NAME);
            $content .= "            '$alias' => $"."this->
>ucfirst($alias)."_model->listar()";
            if($contador < $cantidadLlaves)
            {
                $content .= ", \n";
            }
        }
    }
    $contador = 1;
}

$content .= "\n        );\n";
$content .= "        $"."this->template-
>set('title', 'Agregar ".$componente."');\n";
$content .= "        $"."this->template-
>set('subtitle', '".$componente."');\n";
$content .= "        $"."this->template-
>load('default_layout', 'contents' , '".$strtolower($componente)."/agregar'
, $"."data);\n";
$content .= "    }\n";
$content .= "\n";
$content .= "    public function almacenar()\n";
$content .= "    {\n";
    foreach ($campos as $campo) {
        if($this->validarCampo($campo) && $campo->
>Field != "estCodigo")
        {
            $content .= "        $"."$campo->Field." = $"."this->input-
>post('".$campo->Field."');\n";
            if($campo->Null == "NO")
            {

```

```

        $content .= "                $"."this->form_validation-
>set_rules('".$campo->Field."', '".$campo->Field."', 'required');\n";
    }
}
}
$content .= "\n";
$content .= "                if ($"."this->form_validation->run())\n";
$content .= "                {\n";
$content .= "\n";
$content .= "                $"."data = array(\n";
foreach ($campos as $campo) {
    if($this->validarCampo($campo) && $campo->Field != "estCodigo")
    {
        $content .= "                '".$campo->Field."' => $"."$"."$campo->Field.",\n";
    }
    else if($campo->Field == 'estCodigo')
    {
        $content .= "                'estCodigo' => 1,\n";
    }
    else if(strpos($campo->Field, 'UsuarioCreacion'))
    {
        $content .= "                '".$campo->Field."' => $"."$"."this->session->userdata('identificacion'),\n";
    }
    else if(strpos($campo->Field, 'FechaCreacion'))
    {
        $content .= "                '".$campo->Field."' => date('Y-m-d h:i:sa'),\n";
    }
    else if(strpos($campo->Field, 'IpCreacion'))
    {
        $content .= "                '".$campo->Field."' => $"."$_SERVER['REMOTE_ADDR'],\n";
    }
    else
    {
        //Campo de actualizacion o llave primaria
    }
}

$content .= "                );\n";

```

```

        $content .= "\n";
        $content .= "            if($"."this->".$componente."_model-
>guardar($"."data))\n";
        $content .= "                {\n";
        $content .= "                    redirect(base_url().'".$componente."'")
;\n";
        $content .= "                }\n";
        $content .= "            else\n";
        $content .= "                {\n";
        $content .= "                    $"."this->session-
>set_flashdata('error', 'No se pudo guardar la informacion');\n";
        $content .= "                    redirect(base_url().'".strtolower($com
ponente)."/agregar');\n";
        $content .= "                }\n";
        $content .= "            }\n";
        $content .= "        else \n";
        $content .= "            {\n";
        $content .= "                $"."this->agregar();\n";
        $content .= "            }\n";
        $content .= "    }\n";
        $content .= "\n";
        $content .= "    public function editar($"."id)\n";
        $content .= "    {\n";
        $content .= "        $"."data = array(\n";
        $alias = $this->obtenerAlias($tabla);
        $content .= "            '".$alias."' => $"."this-
>".$componente."_model->obtener($"."id");
        if($cantidadLlaves > 0)
        {
            $content .= ",\n";
        }
        if($cantidadLlaves > 1)
        {
            foreach ($llaves as $llave) {
                if(strpos($llave->CONSTRAINT_NAME, 'fk'))
                {
                    $contador++;
                    $alias = $this->obtenerAlias($llave-
>REFERENCED_TABLE_NAME);
                    $content .= "                '$alias' => $"."this-
>".ucfirst($alias)."_model->listar()";
                    if($contador < $cantidadLlaves)
                    {

```

```

        $content .= ",\n";
    }
}
}
}
    $contador = 1;
}
$content .= "\n        );\n";
$content .= "        $"."this->template-
>set('title', 'Editar ".$componente."');\n";
    $content .= "        $"."this->template-
>set('subtitle', ' ".$componente."');\n";
    $content .= "        $"."this->template-
>load('default_layout', 'contents' , ' ".strtolower($componente)."/editar',
    $"."data);\n";
    $content .= "    }\n";
    $content .= "\n";
    $content .= "    public function actualizar()\n";
    $content .= "    {\n";
    $content .= "        $"."id = $"."this->input->post('id');\n";
    foreach ($campos as $campo) {
        if($this->validarCampo($campo) && $campo-
>Field != "estCodigo")
        {
            $content .= "            $"."$campo->Field." = $"."this->input-
>post(' ".$campo->Field."');\n";
            if($campo->Null == "NO")
            {
                $content .= "                $"."this->form_validation-
>set_rules(' ".$campo->Field."', ' ".$campo->Field."', 'required');\n";
            }
        }
    }
    $content .= "\n";
    $content .= "        if ($"."this->form_validation->run())\n";
    $content .= "        {\n";
    $content .= "\n";
    $content .= "            $"."data = array(\n";
    foreach ($campos as $campo) {
        if($this->validarCampo($campo) && $campo-
>Field != "estCodigo")
        {
            $content .= "                ' ".$campo-
>Field."' => $"."$"."$campo->Field.",\n";

```



```

    }
    else if(strpos($campo->Field, 'UsuarioActualizacion'))
    {
        $content .= "                ".$campo->Field.'" => $"."this->session->userdata('identificacion'),\n";
    }
    else if(strpos($campo->Field, 'FechaActualizacion'))
    {
        $content .= "                ".$campo->Field.'" => date('Y-m-d h:i:sa'),\n";
    }
    else if(strpos($campo->Field, 'IpActualizacion'))
    {
        $content .= "                ".$campo->Field.'" => $_SERVER['REMOTE_ADDR'],\n";
    }
    else
    {
        //Campo de actualizacion o llave primaria
    }
}
$content .= "                );\n";
$content .= "\n";
$content .= "                if($"."this->".$componente."_model->actualizar($"."id, $"."data))\n";
$content .= "                {\n";
$content .= "                    redirect(base_url().'".strtolower($componente)."'");\n";
$content .= "                }\n";
$content .= "                else\n";
$content .= "                {\n";
$content .= "                    $"."this->session->set_flashdata('error', 'No se pudo actualizar la informacion');\n";
$content .= "                    redirect(base_url().'".strtolower($componente)."/editar/'".$"."id);\n";
$content .= "                }\n";
$content .= "                }\n";
$content .= "                else \n";
$content .= "                {\n";
$content .= "                    $"."this->editar($"."id);\n";
$content .= "                }\n";
$content .= "            }\n";
$content .= "\n";

```

```

$content .= "    public function ver($"."id)\n";
$content .= "    {\n";
$content .= "        $"."data = array(\n";
$alias = $this->obtenerAlias($tabla);
$content .= "            '$alias.'" => $"."this-
>".$componente."_model->obtener($"."id)\n";
$content .= "        );\n";
$content .= "\n";
$content .= "        $"."this->template-
>set('title', 'Ver ".$componente."');\n";
$content .= "        $"."this->template-
>set('subtitle', '". $componente."');\n";
$content .= "        $"."this->template-
>load('default_layout', 'contents' , '".strtolower($componente)."/ver', $"
."data);\n";
$content .= "    }\n";
$content .= "\n";
$content .= "    public function eliminar($"."id)\n";
$content .= "    {\n";
$content .= "        $"."data = array(\n";
$content .= "            'estCodigo' => 3,\n";
foreach ($campos as $campo) {
    if(strpos($campo->Field, 'UsuarioActualizacion'))
    {
        $content .= "                '$campo->Field.'" => $"."this-
>session->userdata('identificacion'),\n";
    }
    else if(strpos($campo->Field, 'FechaActualizacion'))
    {
        $content .= "                '$campo->Field.'" => date('Y-m-
d h:i:sa'),\n";
    }
    else if(strpos($campo->Field, 'IpActualizacion'))
    {
        $content .= "                '$campo-
>Field.'" => $"."_SERVER['REMOTE_ADDR'],\n";
    }
    else
    {
        //Campo de actualizacion o llave primaria
    }
}
$content .= "        );\n";

```

```

        $content .= "\n";
        $content .= "        $" . "this->" . $componente . "_model-
>actualizar($" . "id, $" . "data);\n";
        $content .= "\n";
        $content .= "        redirect(base_url()." . strtolower($componente)
. "');\n";
        $content .= "    }\n";
        $content .= "}\n";
        $content .= "\n";
        $content .= "/* Fin del archivo " . $componente . ".php */\n";

        $fp = fopen($ruta . "/" . $componente . ".php", "wb");
        fwrite($fp, $content);
        fclose($fp);
    }

```

### *Generar vistas*

```

public function generarVistas($tabla, $componente, $ruta, $campos, $llaves
, $dependientes)
    {
        if(!is_dir(strtolower($componente)))
        {
            mkdir($ruta . "/" . strtolower($componente));
        }
        //Inicio Generar la vista Index
        $content = "<section class='content'>\n";
        $content .= "    <div class='box box-solid'>\n";
        $content .= "        <div class='box-body'>\n";
        $content .= "            <div class='row'>\n";
        $content .= "                <div class='col-md-12'>\n";
        $content .= "                    <?php if($" . "permisos-
>perInsert == 1): ?>\n";
        $content .= "                        <a href='<?php echo base_url()
?>' . strtolower($componente) . "/agregar' class='btn btn-primary btn-
flat'><span class='fa fa-plus'></span> Agregar " . $componente . "</a>\n";
        $content .= "                            <?php endif;?>\n";
        $content .= "                    </div>\n";
        $content .= "                </div>\n";
        $content .= "            <hr>\n";
        $content .= "        <div class='row'>\n";
        $content .= "            <div class='col-md-12'>\n";

```

```

        $content .= "                <table class='table table-
bordered table-hover datatable'>\n";
        $content .= "                <thead>\n";
        $content .= "                <tr>\n";
        foreach($campos as $campo)
        {
            if($this->validarCampo($campo) && $campo-
>Field != 'estCodigo')
            {
                $content .= "                <th>".$campo-
>Field."</th>\n";
            }
        }
        $content .= "                <th>Opciones</th>\n";
        $content .= "                </tr>\n";
        $content .= "            </thead>\n";
        $content .= "            <tbody>\n";
        $alias = $this->obtenerAlias($tabla);
        $content .= "                <?php if(!empty($".$alias
.")):?>\n";
        $content .= "                <?php foreach($".$ali
as." as $"."item): ?>\n";
        $content .= "                <tr>\n";
        foreach($campos as $campo)
        {
            if($this->validarCampo($campo) && $campo-
>Field != 'estCodigo')
            {
                $content .= "                <td><
?php echo $"."item->".$campo->Field." ?></td>\n";
            }
        }
        $content .= "                <td>\n";
        $content .= "                <div clas
s='btn-group'>\n";
        $content .= "                <a hr
ef='<?php echo base_url()?>".strtolower($componente)."/ver/<?php echo $"."
item->".$this->llavePrimaria($campos)." ?>' class='btn btn-
info'><span class='fa fa-search'></span></a>\n";
        $content .= "                <?php
if($"."permisos->perUpdate == 1): ?> \n";
        $content .= "                <a
href='<?php echo base_url()?>".strtolower($componente)."/editar/<?php ech

```



```

        foreach ($campos as $campo) {
            if($this->validarCampo($campo) && $campo->Field != 'estCodigo')
            {
                $content .= "                <p><strong>".$campo->Field."</strong><?php echo $".$this->obtenerAlias($tabla)."->".$campo->Field."</strong></p>\n";
            }
        }
        $content .= "                </div>\n";
        $content .= "            </div>\n";
        $content .= "        </div>\n";
        $content .= "    </div>\n";
        $content .= " </section>\n";

        $fp = fopen($ruta."/".strtolower($componente) . "/ver.php", "wb");
        fwrite($fp,$content);
        fclose($fp);

        //Fin Generar la vista Ver

        $content = "";

        //Inicio Generar la vista Agregar

        $content = "<section class='content'>\n";
        $content .= "    <div class='box box-solid'>\n";
        $content .= "        <div class='box-body'>\n";
        $content .= "            <div class='row'>\n";
        $content .= "                <div class='col-md-12'>\n";
        $content .= "                    <?php if($"."this->session->flashdata('error')): ?>\n";
        $content .= "                        <div class='alert alert-danger alert-dismissible'>\n";
        $content .= "                            <button type='button' class='close' data-dismiss='alert' arial-hidden='true'>\n";
        $content .= "                                &times;\n";
        $content .= "                            </button>\n";
        $content .= "                            <p><i class='icon fa fa-ban'></i><?php echo $"."this->session->flashdata('error'); ?></p>\n";
        $content .= "                        </div>\n";
        $content .= "                    <?php endif; ?>\n";

```

```

        $content .="                                <form action='<?php echo base_url(
)?>".strtolower($componente)."/almacenar' method='POST'>\n";

        foreach ($campos as $campo) {
            if($this->validarCampo($campo))
            {
                if($this->esForanea($campo, $llaves) && $campo->Field != "estCodigo")
                {
                    $content .="                    <div class='form-
group <?php echo !empty(form_error('".$campo->Field."'))? 'has-
error':''; ?>'>\n";
                    $alias = $this->obtenerAlias($this->obtenerTablaForanea($campo, $llaves));
                    $camposTablaReferencia = $this->Scaffolding_model->listarCapos($this->obtenerTablaForanea($campo, $llaves));
                    $content .="                    <label for='".$
$campo->Field."'>".$campo->Field."</label>\n";
                    $content .="                    <select name='
".$campo->Field."' id='".$campo->Field."' class='form-control'>\n";
                    $content .="                    <?php fore
ach($".$alias." as $"."item): ?>\n";
                    $content .="                    <optio
n value='<?php echo $"."item->".$campo->Field." ?>'><?php echo $"."item-
">".$camposTablaReferencia[1]->Field." ?></option>\n";
                    $content .="                    <?php endf
oreach; ?>\n";
                    $content .="                    </select>\n";
                    $content .="                    </div>\n";
                }
            }
            else
            {
                if(strpos($campo->Type, "varchar") && $campo->Field != "estCodigo")
                {
                    $content .="                    <div class='fo
rm-group <?php echo !empty(form_error('".$campo->Field."'))? 'has-
error':''; ?>'>\n";
                    $content .="                    <label for
='".$campo->Field."'>".$campo->Field."</label>\n";
                    $content .="                    <input typ
e='text' class='form-control' id='".$campo->Field."' name='".$campo->Field."' value='<?php echo set_value('".$campo->Field."') ?>'>\n";

```

```

        $content .= "                                <?php echo
form_error('".$campo->Field."', '<span class=\"help-
block\">', '</span>'); ?>\n";
        $content .= "                                </div>\n";
    }
    else if(strpos($campo->Type, "int") && $campo-
>Field != "estCodigo")
    {
        $content .= "                                <div class='fo
rm-group <?php echo !empty(form_error('".$campo->Field."'))? 'has-
error':''; ?>'>\n";
        $content .= "                                <label for
='".$campo->Field."'>".$campo->Field."</label>\n";
        $content .= "                                <input typ
e='number' step='1' class='form-control' id='".$campo-
>Field.'" name='".$campo->Field.'" value='<?php echo set_value('".$campo-
>Field.'"') ?>'>\n";
        $content .= "                                <?php echo
form_error('".$campo->Field."', '<span class=\"help-
block\">', '</span>'); ?>\n";
        $content .= "                                </div>\n";
    }
    else if(strpos($campo->Type, "date") && $campo-
>Field != "estCodigo")
    {
        $content .= "                                <div class='fo
rm-group <?php echo !empty(form_error('".$campo->Field."'))? 'has-
error':''; ?>'>\n";
        $content .= "                                <label for
='".$campo->Field."'>".$campo->Field."</label>\n";
        $content .= "                                <input typ
e='date' class='form-control' id='".$campo->Field.'" name='".$campo-
>Field.'" value='<?php echo set_value('".$campo->Field.'"') ?>'>\n";
        $content .= "                                <?php echo
form_error('".$campo->Field."', '<span class=\"help-
block\">', '</span>'); ?>\n";
        $content .= "                                </div>\n";
    }
    else if(strpos($campo->Type, "decimal") && $campo-
>Field != "estCodigo")
    {

```



```

        $content .= "                                <div class='fo
rm-group <?php echo !empty(form_error('".$campo->Field."'))? 'has-
error':''; ?>'>\n";
        $content .= "                                <label for
='".$campo->Field."'>".$campo->Field."</label>\n";
        $content .= "                                <input typ
e='number' step='.01' class='form-control' id='".$campo-
>Field.'" name='".$campo->Field.'" value='<?php echo set_value('".$campo-
>Field.'" ?>'>\n";
        $content .= "                                <?php echo
form_error('".$campo->Field."', '<span class=\"help-
block\">', '</span>'); ?>\n";
        $content .= "                                </div>\n";
    }
    else
    {
        if($campo->Field != "estCodigo")
        {
            $content .= "                                <div class
='form-group <?php echo !empty(form_error('".$campo->Field."'))? 'has-
error':''; ?>'>\n";
            $content .= "                                <label
for='".$campo->Field."'>".$campo->Field."</label>\n";
            $content .= "                                <input
type='text' class='form-control' id='".$campo->Field.'" name='".$campo-
>Field.'" value='<?php echo set_value('".$campo->Field.'" ?>'>\n";
            $content .= "                                <?php
echo form_error('".$campo->Field."', '<span class=\"help-
block\">', '</span>'); ?>\n";
            $content .= "                                </div>\n";
        }
    }
}
}
}

$content .= "                                <div class='form-group'>\n";
$content .= "                                <button type='submit' clas
s='btn btn-success btn-flat'>Guardar</button>\n";
$content .= "                                </div>\n";
$content .= "                                </form>\n";
$content .= "                                </div>\n";
$content .= "                                </div>\n";

```

```

$content .= "        </div>\n";
$content .= "    </div>\n";
$content .= "</section>\n";

$fp = fopen($ruta."/".strtolower($componente) . "/agregar.php", "wb
");
fwrite($fp, $content);
fclose($fp);

//Fin Generar la vista Agregar

$content = "";

//Inicio Generar la vista Editar

$content = "<section class='content'>\n";
$content .= "    <div class='box box-solid'>\n";
$content .= "        <div class='box-body'>\n";
$content .= "            <div class='row'>\n";
$content .= "                <div class='col-md-12'>\n";
$content .= "                    <?php if(\$"."this->session-
>flashdata('error')): ?>\n";
$content .= "                        <div class='alert alert-
danger alert-dismissible'>\n";
$content .= "                            <button type='button' clas
s='close' data-dismiss='alert' arial-hidden='true'>\n";
$content .= "                                &times;\n";
$content .= "                            </button>\n";
$content .= "                            <p><i class='icon fa fa-
ban'></i><?php echo \$"."this->session->flashdata('error'); ?></p>\n";
$content .= "                        </div>\n";
$content .= "                    <?php endif; ?>\n";
$content .= "                <form action='<?php echo base_url(
)?>".strtolower($componente)."/actualizar' method='POST'>\n";
$content .= "                    <input type='hidden' value='<?
php echo \$"."$alias.">".$this->llavePrimaria($campos)." ?>' name='id'>\n";
                foreach ($campos as $campo) {
                    if($this->validarCampo($campo))
                    {
                        if($this->esForanea($campo, $llaves) && $campo-
>Field != "estCodigo")
                        {

```

```

        $alias = $this->obtenerAlias($this-
>obtenerTablaForanea($campo, $llaves));
        $aliasTabla = $this->obtenerAlias($tabla);
        $camposTablaReferencia = $this->Scaffolding_model-
>listarCapos($this->obtenerTablaForanea($campo, $llaves));
        $content .= "                                <div class='form-
group <?php echo !empty(form_error('".$campo->Field."'))? 'has-
error':''; ?>'>\n";
        $content .= "                                <label for='".$
$campo->Field."'>".$campo->Field."</label>\n";
        $content .= "                                <select name='
".$campo->Field."' id='".$campo->Field."' class='form-control'>\n";
        $content .= "                                <?php fore
ach($".$alias." as $"."item): ?>\n";
        $content .= "                                <?php
if($"."item->".$campo->Field." == $".$aliasTabla."->".$campo-
>Field."): ?>\n";
        $content .= "                                <o
ption value='<?php echo $"."item->".$campo-
>Field." ?>' selected><?php echo $"."item->".$camposTablaReferencia[1]-
>Field." ?></option>\n";
        $content .= "                                <?php
else: ?>\n";
        $content .= "                                <o
ption value='<?php echo $"."item->".$campo-
>Field." ?>' ><?php echo $"."item->".$camposTablaReferencia[1]-
>Field." ?></option>\n";
        $content .= "                                <?php
endif; ?>\n";
        $content .= "                                <?php endf
oreach; ?>\n";
        $content .= "                                </select>\n";
        $content .= "                                <?php echo for
m_error('".$campo->Field."', '<span class=\"help-
block\">', '</span>'); ?>\n";
        $content .= "                                </div>\n";
    }
    else
    {
        if(strpos($campo->Type, "varchar") && $campo-
>Field != "estCodigo")
        {
            $alias = $this->obtenerAlias($tabla);

```

```

        $content .= "                                <div class='fo
rm-group <?php echo !empty(form_error('".$campo->Field."'))? 'has-
error':''; ?>'>\n";
        $content .= "                                <label for
='".$campo->Field."'>".$campo->Field."</label>\n";
        $content .= "                                <input typ
e='text' class='form-control' id='".$campo->Field.'" name='".$campo-
>Field.'" value='<?php echo !empty(form_error('".$campo-
>Field."'))? set_value('".$campo->Field."') : $".$alias."->".$campo-
>Field.'" ?>'>\n";
        $content .= "                                <?php echo
form_error('".$campo->Field."', '<span class=\"help-
block\">', '</span>'); ?>\n";
        $content .= "                                </div>\n";
    }
    else if(strpos($campo->Type, "int") && $campo-
>Field != "estCodigo")
    {
        $alias = $this->obtenerAlias($tabla);
        $content .= "                                <div class='fo
rm-group <?php echo !empty(form_error('".$campo->Field."'))? 'has-
error':''; ?>'>\n";
        $content .= "                                <label for
='".$campo->Field."'>".$campo->Field."</label>\n";
        $content .= "                                <input typ
e='number' step='1' class='form-control' id='".$campo-
>Field.'" name='".$campo-
>Field.'" value='<?php echo !empty(form_error('".$campo-
>Field."'))? set_value('".$campo->Field."') : $".$alias."->".$campo-
>Field.'" ?>'>\n";
        $content .= "                                <?php echo
form_error('".$campo->Field."', '<span class=\"help-
block\">', '</span>'); ?>\n";
        $content .= "                                </div>\n";
    }
    else if(strpos($campo->Type, "date") && $campo-
>Field != "estCodigo")
    {
        $alias = $this->obtenerAlias($tabla);
        $content .= "                                <div class='fo
rm-group <?php echo !empty(form_error('".$campo->Field."'))? 'has-
error':''; ?>'>\n";

```

```

        $content .= "                                <label for
= '$campo->Field.'">"$campo->Field." :</label>\n";
        $content .= "                                <input typ
e='date' class='form-control' id=' '$campo->Field.'" name=' '$campo-
>Field.'" value='<?php echo !empty(form_error('$campo-
>Field.'"'))? set_value('$campo->Field.'"') : '$'.$alias.'"->"$campo-
>Field.'" ?>'>\n";

        $content .= "                                <?php echo
form_error('$campo->Field.'"', '<span class=\"help-
block\">', '</span>'); ?>\n";
        $content .= "                                </div>\n";
    }
    else if(strpos($campo->Type, "decimal") && $campo-
>Field != "estCodigo")
    {
        $alias = $this->obtenerAlias($tabla);
        $content .= "                                <div class='fo
rm-group <?php echo !empty(form_error('$campo->Field.'"'))? 'has-
error':''; ?>'>\n";
        $content .= "                                <label for
= '$campo->Field.'">"$campo->Field." :</label>\n";
        $content .= "                                <input typ
e='number' step='.01' class='form-control' id=' '$campo-
>Field.'" name=' '$campo-
>Field.'" value='<?php echo !empty(form_error('$campo-
>Field.'"'))? set_value('$campo->Field.'"') : '$'.$alias.'"->"$campo-
>Field.'" ?>'>\n";
        $content .= "                                <?php echo
form_error('$campo->Field.'"', '<span class=\"help-
block\">', '</span>'); ?>\n";
        $content .= "                                </div>\n";
    }
    else
    {
        if($campo->Field != "estCodigo")
        {
            $alias = $this->obtenerAlias($tabla);
            $content .= "                                <div class
='form-group <?php echo !empty(form_error('$campo->Field.'"'))? 'has-
error':''; ?>'>\n";
            $content .= "                                <label
for=' '$campo->Field.'">"$campo->Field." :</label>\n";

```



```

        <div class="col-md-12">
            <?php if($this->session->flashdata("error")): ?>
                <div class="alert alert-danger alert-
dismissible">
                    <button type="button" class="close" data-
dismiss="alert" arial-hidden="true">
                        &times;
                    </button>
                    <p><i class="icon fa fa-
ban"></i><?php echo $this->session->flashdata("error"); ?></p>
                </div>
            <?php endif; ?>
            <form action="<?php echo base_url()?>/scaffolding/
crear" method="POST">
                <div class="form-group">
                    <label for="tabla">Tablas: </label>
                    <select name="tabla" id="tabla" class="for
m-control">
                        <?php foreach($tablas as $tabla): ?>
                            <option value="<?php echo $tabla-
>TABLAS ?>"><?php echo $tabla->TABLAS ?></option>
                        <?php endforeach; ?>
                    </select>
                </div>
                <div class="form-group">
                    <button type="submit" class="btn btn-
success btn-flat">Ver resumen</button>
                </div>
            </form>
        </div>
    </div>
</div>
</section>

```

## Resumen

```

<section class="content">
    <div class="box box-solid">
        <div class="box-body">
            <form action="<?php echo base_url()?>/scaffolding/generar"
method="POST">

```

```

        <input type="hidden" name="tabla" value="<?php echo $t
abla ?>">
        <!--<div class="form-group">
            <label for="nombre">Nombre del componente: </label
>
            <input type="text" class="form-
control" id="nombre" name="nombre" required>
        </div-->
        <div class="form-group">
            <button type="submit" class="btn btn-success btn-
flat">Generar Scaffolding</button>
        </div>
    </form>
    <hr>
    <h3>Campos de la Tabla <?php echo $tabla ?></h3>
    <div class="row">
        <div class="col-md-12">
            <table class="table table-bordered table-
hover datatable">
                <thead>
                    <tr>
                        <th>Campo</th>
                        <th>Tipo</th>
                        <th>Es nulo</th>
                        <th>Llave</th>
                        <th>Extra</th>
                    </tr>
                </thead>
                <tbody>
                    <?php if(!empty($campos)):?>
                        <?php foreach($campos as $campo): ?>
                            <tr>
                                <td><?php echo $campo-
>Field ?></td>
                                <td><?php echo $campo-
>Type ?></td>
                                <td><?php echo $campo-
>Null ?></td>
                                <td><?php echo $campo-
>Key ?></td>
                                <td><?php echo $campo-
>Extra ?></td>
                            </tr>

```



```

                <?php endforeach; ?>
            <?php endif; ?>
        </tbody>
    </table>
</div>
</div>
</div>
</div>
</div>
<div class="box box-solid">
    <div class="box-body">
        <h3>Llaves de la Tabla <?php echo $tabla ?></h3>
        <div class="row">
            <div class="col-md-12">
                <table class="table table-bordered table-
hover datatable">
                    <thead>
                        <tr>
                            <th>Campo</th>
                            <th>Restriccion</th>
                            <th>Tabla referenciada</th>
                            <th>Campo referenciado</th>
                        </tr>
                    </thead>
                    <tbody>
                        <?php if(!empty($foraneas)):?>
                            <?php foreach($foraneas as $foranea):
?>
                                <tr>
                                    <td><?php echo $foranea-
>COLUMN_NAME ?></td>
                                    <td><?php echo $foranea-
>CONSTRAINT_NAME ?></td>
                                    <td><?php echo $foranea-
>REFERENCED_TABLE_NAME ?></td>
                                    <td><?php echo $foranea-
>REFERENCED_COLUMN_NAME ?></td>
                                </tr>
                            <?php endforeach; ?>
                        <?php endif; ?>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
</div>

```

```
        </div>  
    </div>  
</section>
```

Con el código anteriormente expuesto, se da la creación de tablas base, además de permitir los recursos necesarios de listar, crear, editar y eliminar información de una tabla creadas en la base de datos.

## **7. PERSONAS O PROPONENTES**

### **7.1. PROPONENTE PRIMARIO**

**DUVAN ALEXANDER ANGARITA SANCHEZ**, Investigador Principal

Técnico programador de software y Tecnólogo en análisis y desarrollo de sistemas de información con mas de 7 años de experiencia como desarrollador de software en lenguajes PHP y C# en entidades publicas y privadas como Policía Nacional de Colombia (Dirección Antinarcóticos y Oficina de Telemática), Ministerio de Comercio Industria y Turismo, IDEMIA Colombia, Grupo Componente Serviex e IT A la Medida entre otras. Ha realizado grandes aportes a proyecto de gran envergadura para el sector público como el sistema de perfilación de mercancías y pasajeros con destinos internacionales (Policía Nacional), el módulo de exportadores, cupos y reposición Plan Vallejo (Ministerio de Comercio Industria y Turismo).

### **7.2. PROPONENTE SECUNDARIO**

**LUIS FERNANDO ARIAS RAMIREZ**, Director de Proyecto

Ingeniero de sistemas e Ingeniero electrónico de la Universidad Nacional Abierta y a Distancia – UNAD, con especialización en pedagogía, profesor de cálculo, algebra, trigonometría y geometría analítica de la Universidad Agustiniiana y de la Universidad Nacional Abierta y a Distancia.

## 8. RECURSOS DISPONIBLES

### 8.1. RECURSOS FISICOS

Para el desarrollo de este proyecto fue necesario la utilización de un equipo de cómputo equipado con las herramientas y licencias necesarias, de igual manera se utilizó un equipo portátil para realizar pruebas. Por otro lado se requirió de un puesto de trabajo compuesto por escritorio y silla ergonómica de oficina acorde para desarrollar la actividad.

### 8.2. RECURSOS INSTITUCIONALES

La Universidad Nacional Abierta y A Distancia – UNAD, dispuso del Ingeniero Luis Fernando Arias Ramírez como Director de Proyecto, con el objetivo de orientar y dar pautas necesarias para que el proyecto se ejecutara de una forma adecuada y cumplimiento los lineamientos de la universidad.

### 8.3. PRESUPUESTO

El costo de los recursos necesarios para el desarrollo del proyecto fue de COP \$ 5.500.000 pesos colombianos los cuales se discriminan a continuación:

*Tabla 8 Recursos y gastos - Fuente: Este proyecto*

RECURSO	DESCRIPCIÓN	PRESUPUESTO
<b>Equipo</b>	Se requiere un desarrollador de software en lenguaje PHP con conocimientos en el framework CodeIgniter.	\$ 4.000.000
<b>Humano</b>		

<b>Equipos y Software</b>	Computador con características mínimas: 4 GB de RAM Procesador I3 2GB de almacenamiento XAMP 5.6	\$ 1.500.000
<b>TOTAL</b>		
		<b>\$ 5.500.000</b>

## 9. RESULTADOS E IMPACTO ESPERADO

Después de haber realizado las pruebas de funcionalidad se puede evidenciar que el software permite realizar las funciones para las que fue diseñado, crea correctamente las tablas base en la base de datos, así como el usuario administrador inicial. Por otro lado, crea los modelos, las vistas y los controladores dependiendo de la tabla de base de datos seleccionada para tal fin.

También fueron realizadas las pruebas de seguridad, las cuales permiten identificar que solo usuarios autenticados pueden realizar funciones en el sistema de acuerdo con los permisos otorgados. Así mismo permite un correcto funcionamiento y ejecución de autogeneración de código al usuario administrador únicamente.

Con la información anterior se puede deducir que el software pasa las pruebas de funcionalidad y de seguridad, así mismo se permite identificar que cuenta con un buen performance en la experiencia de usuario y permite una facilidad a la hora de desarrollar en PHP con el Framework CodeIgniter, reduciendo tiempos de desarrollo en un 60 por ciento utilizando metodologías de desarrollos ágiles.

## 10. CRONOGRAMA DE ACTIVIDADES

Tareas	Fecha de inicio	Fecha final	Días	Estado
<b>Identificación de la estructura del Framework CodeIgniter</b>	23/08/2019	24/08/2019	1	Finalizado
Identificar estructura de carpetas y rutas	23/08/2019	24/08/2019	1	Finalizado
<b>Análisis del desarrollo a realizar</b>	25/08/2019	27/08/2019	2	Finalizado
Definir requerimientos funcionales	25/08/2019	26/08/2019	1	Finalizado
Definir requerimientos no funcionales	26/08/2019	27/08/2019	1	Finalizado
<b>Diseño del Modelo Entidad Relación de la base de Datos</b>	28/08/2019	31/08/2019	3	Finalizado
Identificar tablas base del sistema	28/08/2019	29/08/2019	1	Finalizado
Identificar relaciones entre tablas	29/08/2019	30/08/2019	1	Finalizado
Diseñar MER	30/08/2019	31/08/2019	1	Finalizado
<b>Configuración inicial del proyecto</b>	01/09/2019	02/09/2019	1	Finalizado
Crear base de datos	01/09/2019	02/09/2019	1	Finalizado
Configurar conexión de base de datos en proyecto	01/09/2019	02/09/2019	1	Finalizado
Configurar ruta base; variables y constantes	01/09/2019	02/09/2019	1	Finalizado
<b>Integración de plantilla AdminLTE al proyecto</b>	03/09/2019	05/09/2019	2	Finalizado
Crear librería de integración de estilos	03/09/2019	04/09/2019	1	Finalizado
Crear Plantilla principal	04/09/2019	05/09/2019	1	Finalizado
<b>Desarrollo de la autenticación de usuario</b>	06/09/2019	11/09/2019	5	Finalizado
Crear plantilla de Login	06/09/2019	07/09/2019	1	Finalizado
Crear modelo Login	07/09/2019	08/09/2019	1	Finalizado
Crear controlador Login	08/09/2019	09/09/2019	1	Finalizado
Crear vista Login	09/09/2019	10/09/2019	1	Finalizado
Crear vista Perfil	10/09/2019	11/09/2019	1	Finalizado
<b>Desarrollo de funcionalidad para cerrar sesión</b>	12/09/2019	13/09/2019	1	Finalizado
Crear funcionalidad para cerrar sesión en Controlador Login	12/09/2019	13/09/2019	1	Finalizado
<b>Desarrollo de funcionalidad para cambiar contraseña</b>	14/09/2019	16/09/2019	2	Finalizado
Crear vista Cambiar clave	14/09/2019	15/09/2019	1	Finalizado
Cambiar vista Confirmación Cambiar clave	15/09/2019	16/09/2019	1	Finalizado
<b>Desarrollo de funcionalidad para restablecer contraseña</b>	17/09/2019	19/09/2019	2	Finalizado
Crear vista Restablecer Clave	17/09/2019	19/09/2019	2	Finalizado
<b>Desarrollo de CRUD's para roles de usuarios</b>	20/09/2019	25/09/2019	5	Finalizado
Crear modelo Rol	20/09/2019	21/09/2019	1	Finalizado
Crear controlador Rol	21/09/2019	22/09/2019	1	Finalizado
Crear vistas de rol(Index; Ver; Editar; Agregar)	22/09/2019	25/09/2019	3	Finalizado
<b>Desarrollo de CRUD's para estados</b>	26/09/2019	30/09/2019	4	Finalizado
Crear modelo Estados	26/09/2019	27/09/2019	1	Finalizado
Crear controlador Estados	27/09/2019	28/09/2019	1	Finalizado
Crear vistas de estados(Index; Ver; Editar; Agregar)	28/09/2019	30/09/2019	2	Finalizado
<b>Desarrollo de CRUD's para usuarios</b>	01/10/2019	05/10/2019	4	Finalizado
Crear modelo Usuarios	01/10/2019	02/10/2019	1	Finalizado
Crear controlador Usuarios	02/10/2019	03/10/2019	1	Finalizado
Crear vistas de usuario(Index; Ver; Editar; Agregar)	03/10/2019	05/10/2019	2	Finalizado
<b>Desarrollo de CRUD's para Menu</b>	05/10/2019	09/10/2019	4	Finalizado
Crear modelo Menu	05/10/2019	06/10/2019	1	Finalizado
Crear controlador Menu	06/10/2019	07/10/2019	1	Finalizado
Crear vistas de menu(Index; Ver; Editar; Agregar)	07/10/2019	09/10/2019	2	Finalizado
<b>Desarrollo de la librería de permisos</b>	09/10/2019	11/10/2019	2	Finalizado
Crear librería Backend para controlar autorización	09/10/2019	11/10/2019	2	Finalizado
<b>Desarrollo de CRUD's para permisos</b>	11/10/2019	15/10/2019	4	Finalizado
Crear modelo Permisos	11/10/2019	12/10/2019	1	Finalizado
Crear controlador Permisos	12/10/2019	13/10/2019	1	Finalizado
Crear vistas de permisos(Index; Ver; Editar; Agregar)	13/10/2019	15/10/2019	2	Finalizado
<b>Desarrollo del auto generador de Código</b>	15/10/2019	31/10/2019	16	Finalizado
Crear modelo Scaffolding	15/10/2019	20/10/2019	5	Finalizado
Crear controlador Scaffolding	20/10/2019	27/10/2019	7	Finalizado
Crear vistas de scaffolding(Index; Resumen)	27/10/2019	31/10/2019	4	Finalizado
<b>Pruebas de funcionalidad</b>	01/11/2019	07/11/2019	6	Finalizado
Realizar pruebas de funcionalidad	01/11/2019	04/11/2019	3	Finalizado
Realizar pruebas de seguridad	04/11/2019	07/11/2019	3	Finalizado

*Ilustración 15 Cronograma de actividades*

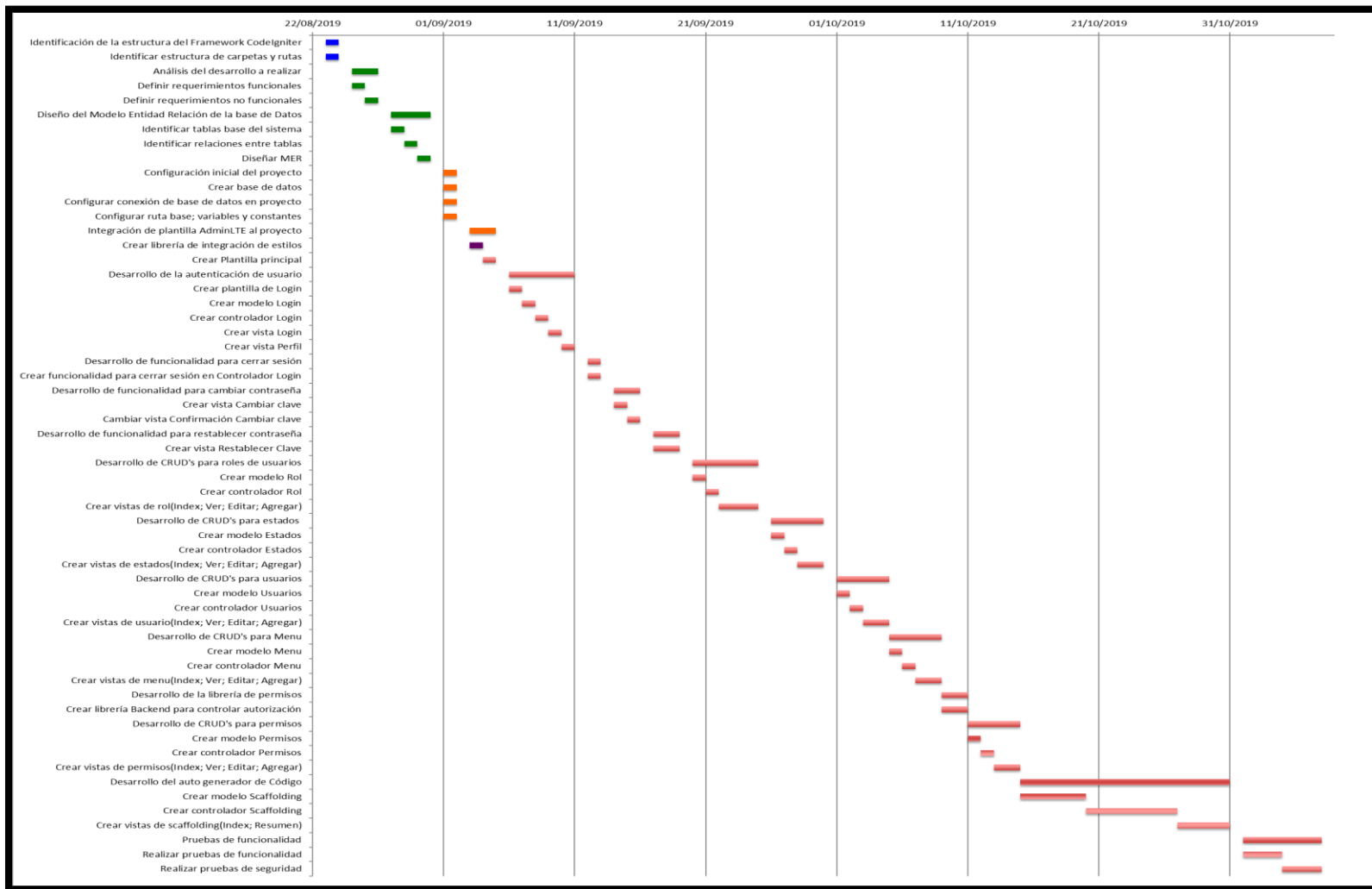


Ilustración 16 Diagrama de Gantt Actividades



## CONCLUSIONES

Las metodologías de desarrollo de software utilizadas hoy en día requieren una exigencia mayor que hace algunos años, ya que implementa una entrega mucho más corta pero del mismo modo demandan tener una arquitectura robusta que permita mantener un performance de la aplicación y una simultaneidad de peticiones elevada, por lo cual, al desarrollar la herramienta tecnológica de autogeneración de código en lenguaje PHP para el framework CodeIgniter, se pueden suplir estas necesidades para un grupo de desarrollo y así mismo poder brindar un complemento tecnológico para reducir los tiempos de desarrollo en un 60 por ciento teniendo siempre presente la calidad de los sistemas informáticos.

## LISTA DE REFERENCIAS

- Anónimo. (2013). La familia de normas ISO/IEC 25000. ISO 25000. Recuperado de <https://iso25000.com/index.php/normas-iso-25000>
- Anónimo. (2014). Kanban: origen y definición. Recuperado de [https://www.scrummanager.net/bok/index.php?title=Kanban: origen y definici%C3%B3n](https://www.scrummanager.net/bok/index.php?title=Kanban:_origen_y_definici%C3%B3n)
- Anónimo. (2016). Guía Digital. Desarrollo de un plan de pruebas. Recuperado de <http://www.guiadigital.gob.cl/articulo/desarrollo-de-un-plan-de-pruebas>
- Anónimo. (2016). Prueba de Software. Recuperado de [https://senastage.blackboard.com/bbcswebdav/courses/150752/Pruebas%20del%20Software\\_T%C3%A9cnicas.pdf](https://senastage.blackboard.com/bbcswebdav/courses/150752/Pruebas%20del%20Software_T%C3%A9cnicas.pdf)
- Anónimo. (2016). Guía Digital. Desarrollo de un plan de pruebas. Recuperado de <http://www.guiadigital.gob.cl/articulo/desarrollo-de-un-plan-de-pruebas>
- Anónimo. (2016). Guía Digital. Desarrollo de un plan de pruebas. Recuperado de <http://www.guiadigital.gob.cl/articulo/desarrollo-de-un-plan-de-pruebas>
- Bataller, A. (2016). Planificación de proyectos. La gestión de proyectos (pp. 27-42). Barcelona, ESPAÑA: Editorial UOC. Recuperado de <http://bibliotecavirtual.unad.edu.co:2077/lib/unadsp/reader.action?ppg=27&docID=11231227&tm=148279505652>
- Billage. (2016). Metodología Kanban: ventajas y características. Recuperado de <https://www.getbillage.com/es/blog/metodologia-kanban-ventajas-y-caracteristicas>
- Blanco, L. F. J. (2010). Sistema de gestión de la calidad, SGC, en el software. Diseño de procesos claves para el mejoramiento de la calidad en proyectos de software (pp. 7-10). La Habana, CU: D - Instituto Superior Politécnico José Antonio Echeverría. CUJAE. Recuperado de <http://bibliotecavirtual.unad.edu.co:2077/lib/unadsp/reader.action?ppg=15&docID=10609131&tm=1482794566128>
- Borrell, C. R. (2006). Importancia de la ingeniería de requerimientos dentro del ciclo de desarrollo de software. Tono: Revista Técnica De La Empresa De Telecomunicaciones De Cuba, S.A, (3), 52-56. Recuperado de <http://bibliotecavirtual.unad.edu.co:2051/login.aspx?direct=true&db=a9h&AN=34162106&lang=es&site=ehost-live>

- Cabot, S. J. (2013). Tendencias. Ingeniería del software (pp. 28-35). Barcelona, ES: Editorial UOC. Recuperado de <http://bibliotecavirtual.unad.edu.co:2077/lib/unadsp/reader.action?ppg=28&docID=10853350&tm=1482794744237>
- Caponi, M. De vera, D. (2016). Gestión de Software Informe sobre Evaluación de Productos. Recuperado de <http://www.fing.edu.uy/inco/cursos/gestsoft/Presentaciones/Evaluacion%20de%20Productos%20-%20G2/Evaluacion%20de%20Productos.pdf>
- Catalan Santiago, G. (09,03,2017). OVI - Unidad dos Pruebas de software. [Archivo de video]. Recuperado de <http://hdl.handle.net/10596/11652>
- Cordón-García, J. A., Martín-Rodero, H., & Alonso-Arévalo, J. (2009). Gestores de referencias de última generación: análisis comparativo de RefWorks, EndNote Web y Zotero. (Spanish). El Profesional De La Información, 18(4), 445-454. doi: 10.3145/epi.2009.jul.14. Recuperado de <http://bibliotecavirtual.unad.edu.co:2051/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-70349445311&lang=es&site=eds-live>
- Córdoba, M. (2011). Formulación y evaluación de proyectos, p.p. 271 - 285. Recuperado de <http://bibliotecavirtual.unad.edu.co:2460/lib/unadsp/reader.action?ppg=1&docID=3197583&tm=1511031252010>
- Dávila, A. (2012). Gestión de los Costes del Proyecto. Guía de los Fundamentos para la Dirección de Proyectos (GUÍA DEL PMBOK). uacm123.weebly.com. Recuperado de <https://uacm123.weebly.com/3-gestioacuten-de-los-costes-del-proyecto.html>
- Dávila, A. (2012). Gestión del alcance del proyecto. Guía de los Fundamentos para la Dirección de Proyectos (GUÍA DEL PMBOK). uacm123.weebly.com. Recuperado de <https://uacm123.weebly.com/1-gestioacuten-del-alcance-del-proyecto.html>
- Dávila, A. (2012). Gestión de la integración del proyecto. Guía de los Fundamentos para la Dirección de Proyectos (GUÍA DEL PMBOK). uacm123.weebly.com. Recuperado de <https://uacm123.weebly.com/7-getioacuten-de-la-integracioacuten-del-proyecto.html>
- Dávila, A. (2012). Gestión del tiempo del proyecto. Guía de los Fundamentos para la Dirección de Proyectos (GUÍA DEL PMBOK). uacm123.weebly.com.

- Recuperado de <https://uacm123.weebly.com/2-gestioacuten-del-tiempo-del-proyecto.html>
- Develapp. (2018). Metodologías ágiles en el desarrollo de software. Recuperado de <http://www.develapps.com/es/noticias/metodologias-agiles-en-el-desarrollo-de-software>
- Ferrer J. (2010). Metodología De La Investigación: TECNICAS DE LA INVESTIGACION Recuperado de <http://metodologia02.blogspot.com/p/tecnicas-de-la-investigacion.html>
- Fontán, M. (2012). CodeIgniter, un framework PHP para el desarrollo rápido de aplicaciones web. URL <https://www.adwe.es/codigo/codeigniter-framework-php-desarrollo-aplicaciones-web>.
- García, C. (2017). Ensayo de Gerencia de Proyectos (Project management). Monografias.com. Recuperado de <https://www.monografias.com/trabajos81/ensayo-gerencia-proyectos-project-management/ensayo-gerencia-proyectos-project-management.shtml>
- Gonçalvez, L. (2020). Que es la metodología Agile. Recuperado de <https://luis-goncalves.com/es/que-es-la-metodologia-agil/>
- Lozano, L. (2013). ESTÁNDAR DE CALIDAD DEL SOFTWARE. Recuperado de <http://estandarescalidadsoftware.blogspot.com/2013/09/isoiec-25000-esta-proporciona-una-guia.html>
- Marín, M. (2016). ISO 25000 Requisitos y Evaluación de Calidad de Productos Software (SQuaRE). SlideShare. Recuperado de <https://es.slideshare.net/MiguelAngelMarinNaranjo/iso-25000-67653960>
- Mendes, M. (2018). MVC (Model, View, Controller) Explicado. Recuperado de <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>
- Moreno C., F., Rebolledo S., L. A., & Marthe Z., N. (2010). Cómo escribir textos académicos según normas internacionales: APA, IEEE, MLA, Vancouver e Icontec (pp. 3-73). Barranquilla [Colombia]: Universidad del Norte. Recuperado de [http://bibliotecavirtual.unad.edu.co:2051/login.aspx?direct=true&db=e000xww&AN=710205&lang=es&site=ehost-live&ebv=EB&ppid=pp\\_iii](http://bibliotecavirtual.unad.edu.co:2051/login.aspx?direct=true&db=e000xww&AN=710205&lang=es&site=ehost-live&ebv=EB&ppid=pp_iii)
- Muñoz, R. C. (2009). Como elaborar y asesorar una investigación de tesis (pp. 24-40). Córdoba, AR: El Cid Editor | apuntes. Recuperado de

- <http://bibliotecavirtual.unad.edu.co:2077/lib/unadsp/reader.action?ppg=24&docID=10328246&tm=1481584091767>
- Núñez N., E.& Ríos N., D. (2017). Desarrollo de un marco de trabajo (framework) para el desarrollo de aplicaciones web en la Universidad Nacional de Costa Rica. Recuperado de <http://documentas.redclara.net/bitstream/10786/1287/1/131-17-4Desarrollo%20de%20un%20marco%20de%20trabajo%20%28framework%29%20para%20el%20desarrollo%20de%20aplicaciones%20web.pdf>
- Njenga, A. (2018). 10 Popular PHP frameworks in 2019. Recuperado de <https://raygun.com/blog/top-php-frameworks/>
- Palladino, E. (2014). Administración y gestión de proyectos. Administración y gestión de proyectos (pp. 25-34). Buenos Aires, AR: Espacio Editorial. Recuperado de <http://bibliotecavirtual.unad.edu.co:2077/lib/unadsp/reader.action?ppg=25&docID=11162736&tm=1482795267282>
- Pérez-Montoro, M. (2016). Gestión del conocimiento: orígenes y evolución (pp. 1-10). El Profesional De La Información, 25(4), 526-534. doi:10.3145/epi.2016.jul.02. Recuperado de <http://bibliotecavirtual.unad.edu.co:2051/login.aspx?direct=true&db=buh&AN=117565381&lang=es&site=ehost-live>
- QualityDev. (2015). Modelo Vista - Controlador. Recuperado de <https://sites.google.com/site/aunaris2/programacion/modelo-vista---controlador>
- Ramos, O. (2012). Plan de Pruebas. SlideShare. Recuperado de <https://es.slideshare.net/iguamba666/plan-de-pruebas>
- Scalone, F. (2006). Estudio Comparativo de los Modelos y Estándares de Calidad del Software. Recuperado de <http://laboratorios.fi.uba.ar/lsi/scalone-tesis-maestria-ingenieria-en-calidad.PDF>
- Shorelabs. (2019). ¿Por qué utilizar la metodología Kanban?. Recuperado de <https://kanbantool.com/es/metodologia-kanban>
- Significados.com. (2019). Tipos de Investigación. Recuperado de <https://www.significados.com/tipos-de-investigacion/>
- Strijker, D. (1999). Ruimtelijke verschuivingen in de EU-landbouw 1950 – 1992. Groningen: Ruimtelijke Economie Groningen. Recuperado de [https://www.waddenacademie.nl/fileadmin/inhoud/pdf/06-wadweten/Proefschriften/Thesis\\_Systemsma.pdf](https://www.waddenacademie.nl/fileadmin/inhoud/pdf/06-wadweten/Proefschriften/Thesis_Systemsma.pdf)

- Treviño, G. (2015). Los 10 lenguajes de programación más usados de la actualidad. Recuperado de <https://platzi.com/tutoriales/1631-java-basico/173-los-10-lenguajes-de-programacion-mas-usados-de-la-actualidad/>
- Universia. (2019). Los 10 lenguajes de programación más populares en la actualidad. Recuperado de <https://noticias.universia.com.ar/consejos-profesionales/noticia/2016/02/22/1136443/conoce-cuales-lenguajes-programacion-populares.html>
- Universidad Nacional Abierta y a Distancia. UNAD. Manual para usuarios Acceso a las Bases de Datos Bibliográficas Biblioteca. Recuperado de [http://biblioteca.unad.edu.co/images/Ingreso\\_a\\_bases\\_de\\_datos\\_desde\\_cv.pdf](http://biblioteca.unad.edu.co/images/Ingreso_a_bases_de_datos_desde_cv.pdf)
- Wigodski, J. (2010). Población y muestra. Recuperado de <http://metodologiaeninvestigacion.blogspot.com/2010/07/poblacion-y-muestra.html>
- Wikipedia. (2019). Desarrollo ágil de software. Recuperado de [https://es.wikipedia.org/wiki/Desarrollo\\_%C3%A1gil\\_de\\_software](https://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software)
- Wikipedia. (2019). Modelo–vista–controlador. Recuperado de <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>
- Wikipedia. (2019). Kanban (desarrollo). Recuperado de [https://es.wikipedia.org/wiki/Kanban\\_\(desarrollo\)](https://es.wikipedia.org/wiki/Kanban_(desarrollo))
- Yuni, J. A., & Urbano, C. A. (2014). Técnicas para investigar: recursos metodológicos para la preparación de proyectos de investigación. pp 7 – 26. Córdoba, Argentina: Editorial Brujas. Recuperado de <http://bibliotecavirtual.unad.edu.co:2051/login.aspx?direct=true&db=nlebk&AN=847671&lang=es&site=eds-live>