LINEAMENTOS PARA SISTEMAS.

1. Skills del Sistema.
2. Declaración de Clases y/o Controladores
   * Prefijo para declarar Controladores
3. Declaración Métodos y/o Funciones
   * Sintaxis y Prefijo para creación de métodos en las clases y/o controladores
4. Declaración de Variables
   * Prefijo de variables referente a su tipo de dato.
5. Declaración de Modelos
   * Sintaxis y/o prefijos para la creación de modelos.
6. Declaración de Vistas.

* Sintaxis y/o prefijos para la creación de vistas.

1. Creación de migraciones
   * Creación de migraciones con el prefijo establecido.
2. Creación de Seeder
   * Creación de seeders con sus prefijos de trabajo

**1. Skills del Sistema.**

Los Skills requeridos, para los sistemas de desarrollo de diferentes clientes son los siguientes.

* **Laravel 5.6**  versión mas actual
* **PHP 7.1.0** versión mas actual
* **MySQL 5.6** versión mas actual
* **Apache/2.4.25 versión mas actual**
* **Metodología MVC.**
* **POO (Programación Orientado Objetos.)**
* **Composer**
* **GIT Control de versiones.**
* **Bootstrap 3.3.7**
* **JavaScript POO**
* **Jquery 3.3.1**
* **Vue.js 2.0**

**2. Declaración de Clases y/o Controladores**

Para declarar una clase o controlador se tiene que colocar el nombre del modulo y con la palabra Controller, en CamelCase y en plural haciendo referencia al modulo que se debe programar: **ModulosController.**

Se deben de declarar los Controller dentro de la carpeta de Development, se puede utilizar un comando dentro de la terminal ejecutando esta linea **php artisan build:cotroller ModulosController.**

Esto te genera un Controller y modelo con el nombre que le proporciones.

**3. Declaración de Métodos y/o Funciones**

Dentro de un controlador y/o clase se deben general los métodos con su modificador de acceso y su nombre que se le debe asignar, en caso de ser un acceso privado debe declararse de la siguiente manera:

**private function \_metodo\_privado( Request $request ){}**

en caso de que el modificador de acceso es protegido debe de declararse de la siguiente manera

**protected fucntion \_metodo\_protegido( Request $request ){}**

cuando es publico se declara de la siguiente manera

**public function metodo\_publico( Request $request ){}**

Como se puede observa los métodos se declarar con un guión bajo para su mayor legibilidad.

**4. Declaración de Variables**

Cuando se declaren variables se deberán colocar su tipo de variable, en caso de propiedades de una clase deben declarar dependiendo a su modificador de acceso.

#propiedades

private $\_str\_propiedad = “”;

private $\_int\_propiedad = 0;

private $\_double\_propiedad = 0.0;

protected $\_str\_propiedad = “”;

protected $\_int\_propiedad = 0;

protected $\_double\_propiedad = 0.0;

public $str\_propiedad = “”;

public $int\_propiedad = 0;

public $double\_propiedad = 0.0;

**variables**

$str\_variable = “”;

$int\_variable = 0;

$double\_variable = 0.0;

**5. Declaración de Modelos**

Cuando se declaran los modelos dentro del proyecto se colocaran dentro de una carpeta que se encuentra en el sistema, llamada **Model/Development/** donde se declarara con la caracteristica siguiente:

**Prefijo “Sys” seguido el nombre del modulo en plural, con el prefijo Model y CamelCase. “SysNombreModulosModel”**

Se puede utilizar el comando **php artisan build:controller NombreModulosController** el cual se crea el controlador y el modelo en sus respectivas carpetas donde se debera trabajar.

**6. Declaración de Vistas**

Al declarar las vistas se deben de declarar dos por cada modulo en la carpeta Development con la estructura siguiente:

Minúsculas

Nombre del modulo un archivo “modulos.blade.php”

Nombre del modulo con guión bajo para los modales “modulos\_edit.blade.php”

El archivo de edición es donde contendrá los modales o la parte de edición del portal.

**7. Creación de Migraciones**

Cuando se decide crear una tabla se debe crear una migracion con el comando de artisan de esta manera:

**php artisan make:migration create\_sys\_nombre\_tablas\_table –create=sys\_nombre\_tabla**

Siempre se debe crear la tabla con el prefijo “sys” y en plural y si el nombre es largo utilizar guiones bajo, para que concuerde con el modelo que se emplee y se desee trabajar.

**8. Creación de Seeder**

Para la declaración de los seeder, para insertar registros a la base de datos se debe realizar con el comando:

**php artisan make:seeder SysNombreModeloSeeder**

Deberá llevar el prefijo “Sys“

Nombre del modelo que se definió