**DOCUMENTO DE ESTÁNDARIZACIÓN SISTEMA DE INFORMACIÓN DE LOS OBSERVATORIOS REGIONALES**

**CREADO POR:**

Fabián Andrés Chinchilla Blanco 164078

Yeilan CARNÉ

**VERSION 1.0**

**FECHA:** 30/08/2022

**Historial de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Fecha** | **Motivo de los cambios** | **Versión** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contenido

[Resumen ejecutivo 4](#_Toc113268739)

[Descripción del problema 5](#_Toc113268740)

# General

Idioma del proyecto: inglés.

Nombre del código del proyecto: sistemaobservatorioregional

Repositorio oficial: <https://github.com/fabianchin/Sistema-Observatorios-Regionales>

Cuando hay espacios en atributos de bases de datos, atributos de objetos o donde se diga explícitamente como en las controladoras se usara guion bajo “\_”.

# Formato de bases de datos

## Tablas

Estructura: singular, puede usar mayúsculas, no puede usar tildes ni ñ.

Idioma: inglés.

Formato: “tb\_obs\_<nombre de tabla>”

Ejemplo: "tb\_obs\_user"

## Campos o atributos.

### Normales

Estructura: Preferiblemente singular, pero hay excepciones donde apliquen. Siempre empiezan en minúscula, no puede usar tildes ni ñ.

Idioma: inglés.

Detalle: Lleva el nombre de la tabla previo al nombre, pero omitiendo “tb\_obs\_”.

Formato: “<nombre tabla>\_<nombre de atributo>”

Ejemplo 1: "user\_username", aquí “username” le pertenece a la tabla “tb\_obs\_user”

Ejemplo 2:

CREATE TABLE tb\_obs\_IndicatorDetail(

indicatorDetail\_id int,

indicatorDetail\_status int

)

### Constraint: Llaves foráneas y primarias

Base:

CREATE TABLE tb\_obs\_Indicator(

indicator\_id int,

indicator\_name varchar(75)

)

CREATE TABLE tb\_obs\_IndicatorDetail(

indicatorDetail\_id int,

indicatorDetail\_Indicator\_id int, (Esta hace referencia al id de otra tabla)

indicatorDetail\_status tinyint

)

#### Llaves primarias

Formato: “pk\_<nombre del atributo>”

Ejemplo:

CREATE TABLE tb\_obs\_Indicator(

indicator\_id int,

indicator\_name varchar(75),

CONSTRAINT pk\_indicator\_id PRIMARY KEY (indicator\_id)

)

Formato: “fk\_<nombre de atributo>”

Ejemplo:

CREATE TABLE tb\_obs\_Indicator(

indicator\_id int,

indicator\_name varchar(75),

CONSTRAINT pk\_indicator\_id PRIMARY KEY (indicator\_id)

)

CREATE TABLE tb\_obs\_IndicatorDetail(

indicatorDetail\_id int,

indicatorDetail\_indicator\_id int,

indicatorDetail\_status tinyint,

CONSTRAINT fk\_indicatorDetail\_indicator\_id FOREIGN KEY (indicatorDetail\_indicator\_id) REFERENCES Indicator (indicator\_id)

CONSTRAINT pk\_indicatorDetail\_id PRIMARY KEY (indicatorDetail\_id)

)

## Procedimientos almacenados

Estructura: No puede usar tildes ni ñ.

Idioma: inglés.

Detalle: Los nombres son compuestos por infinitivos.

Formato: “<infinitivo><detalle>”

Ejemplo 1: insertUser

Ejemplo 2: obtainAllUsers

# Laravel

## Clases

Por convención los modelos en laravel se crean en ingles, singular y con la primera letra en mayuscula. Importante es mantener los nombres de las bases de datos en los nombres de los objectos.

Ejemplo:

DataBase -> tb\_obs\_user

Laravel -> User

Para los atributos de la clase, mantener los mimos a como están en base de datos, aquellos atributos que no estén se manejaran en inglés, fuera de eso no tienen restricciones.

Ejemplo:

Base de datos

CREATE TABLE tb\_obs\_Indicator(

indicator\_id int,

indicator\_name varchar(75),

CONSTRAINT pk\_indicator\_id PRIMARY KEY (indicator\_id)

)

Laravel

class Indicator{

$indicator\_id;

$indicator\_name;

$index:

$status;

}

## Métodos

Estructura: No puede usar tildes ni ñ.

Idioma: inglés.

Detalle: Los nombres son compuestos por infinitivos, aunque puede haber excepciones de métodos específicos.

Formato: “<infinitivo><detalle>”

Ejemplo: beginAction(), insertForm(), getSomething(), etc.

## Controladoras

Estructura: No puede usar tildes ni ñ.

Idioma: inglés.

Formato: Controller<Modulo>\_<Nombre de la controladora>”

Ejemplo: ControllerPost\_MainView, ControllerDashboards\_FormView, ControllerUserProfile\_Profile, etc.

## En el file web.php

Donde se registran las rutas usar siempre un ->name("mismo nombre que url"); en caso de tener que afectar/cambiar URL del sitio, no es necesario reemplazar el llamado o envio de datos hacia ese archivo en cada lugar llamado, porque se mantiene o trabaja con el name (basicamente es alias).