**Trash Recolector**

**documentación de codigo**

Introduction

#### Description Breve del Problem a:

La subsecretaría de medio ambiente del municipio de Tandil nos contactó para que los ayudemos con el desarrollo de una aplicación que permita generar un mapa de la basura en la ciudad. Se necesita contar con un mecanismo que permita a los ciudadanos reportar lugares de la ciudad en los que se encuentre basura.

Herramientas utilizadas

Como primera instancia se definió y debatió, en base a los conocimientos en común de los integrantes, con que tecnología/s se iba a desarrollar el trabajo propuesto, y las pautas generales de desarrollo, para así todos estar trabajando de la misma manera y no tener problemas a la hora de lectura, fix de bugs, extensión de funcionalidades, etc., del código echo por cualquier integrante del equipo

En base a los conocimientos en común se decidió desarrollar la aplicación con la propuesta de diseño MVC (model-view-controller) y realizar una vista de la aplicación “movile-only”

#### Lenguajes de programación, de marcado y frameworks utilizados

* PHP
* JavaScript
* HTML 5
* CSS3
* Smarty
* MySQL

#### Otros Softwares de utilidad

* Xampp (se decidió mostrar una versión local de la aplicación)
* Editores de texto varios (Atom, VScode, NotePad ++, etc.)

*Clases Intervinientes en el Desarrollo del Sistema*

Clases Correspondientes al Model:

#### Form Model

La funcionalidad principal de esta clase es de guardar los datos provistos por el formulario principal de la aplicación (denuncia de ciudadano).

#### Funciones que intervienen en la clase

**createAComplaint**: Dado los datos provisto por el formulario la función se encarga de guardar dichos datos en la Base de datos de forma segura y utilizando buenas practicas

**uploadImage**: Dada una imagen provista por el formulario, esta función se encarga de guardarla en la base de datos

#### Admin Model

La funcionalidad principal de esta clase traer los datos de las denuncias para luego poder ser utilizadas en la sesión de administrador

#### Funciones que intervienen en la clase

**fetchMap**: Se encarga de traer las coordenadas de los registros de denuncia guardados

**fetchAllReports**: Retorna todos los reportes que contenga la base de datos, independientemente del estado de dichos reportes

Clases Correspondientes a la View:

#### Form View

Se encarga de la vista del formulario de denuncia, para mostrar la información del lado del cliente se utiliza el framework *Smarty*

#### Funciones que intervienen en la clase

**showForm**: Muestra el template asociado al formulario de denuncia principal

**successfulComplaint**: Dada una denuncia, en caso de ser exitosa se mostrara un template para dar seguridad al usuario de que su denuncia fue tomada

#### Admin View

Se encarga de la vista asociada al administrador, para esto es utilizado el framework *Smarty*

#### Funciones que intervienen en la clase

**showMap**: Muestra el mapa con la ruta de la basura para que el Administrador decida el mejor camino

**showReports**: Muestra todas las denuncias de los usuarios con sus respectivos detalles

Clases Correspondientes al Controller:

#### Form Controller

Se encarga de conectar la view y el model correspondientes al formulario de denuncia

#### Funciones que intervienen en la clase

**home**: Llama a la view y pide que muestre el formulario principal de denuncia

**successfulComplaint**: Se encarga de pedirle a la view que informe de cuando una denuncia es exitosa

**addComplaint:** Dado los datos ingresados en el formulario de denuncia, esta función se encarga de pasárselos al model para que luego sean guardados en la Base de Datos

#### Admin Controller

Se encarga de conectar la view y el model correspondientes al área de administrador

#### Funciones que intervienen en la clase

**map**: Le facilita los datos a la view para que así pueda mostrar en el mapa los lugares donde existen denuncias

**report**: Pide los reportes de basura al Model y se los pasa a la View para ser visualizados en el área de administrador