

Documentación DenunciasApp

Arquitectura de Software - Grupo D

1P - 2020

DenunciasApp es una aplicación desarrollada en Python con el framework Flask que permite a los usuarios realizar denuncias fotográficas de la contaminación existente en los caños de los barrios o sectores de las ciudades donde se encuentren, siendo conscientes que dicha contaminación puede ser dada por basuras o residuos que la misma comunidad genera.

Objetivo general

Desarrollar una aplicación de denuncias fotográficas sobre la contaminación presente en los caños de las áreas urbanas, haciendo uso de temáticas dadas en el curso de Arquitectura de Software, tales como patrones de arquitectura, patrones de diseño, implementación de frameworks, desarrollo del lado del servidor y utilización de contenedores a través de Docker.

Objetivos específicos

- Analizar e identificar los requerimientos funcionales de la aplicación.
- Identificar las entidades necesarias para la funcionalidad de la aplicación.
- Diseñar el modelado de las comunicaciones presentes en la aplicación.
- Desarrollar una aplicación atendiendo los estándares establecidos en el modelado inicial.

Esta aplicación está enfocada a la comunidad en general y específicamente a los ciudadanos afines con el cuidado del medio ambiente y las empresas reguladoras de desechos y residuos, aquellos que son conscientes del daño que se genera al planeta con la contaminación y mal manejo de las basuras.

Requerimientos funcionales

Requerimientos del usuario:

- El usuario debe registrarse en la aplicación.
- Una vez logueado, podrá observar las denuncias fotográficas más recientes sobre la contaminación en los caños de las áreas urbanas con su respectiva ubicación.
- El usuario registrará las denuncias fotográficas que considere pertinentes.

Requerimientos del sistema:

- La aplicación podrá ser utilizada en cualquiera sistema operativo.
- La aplicación deberá ser utilizada en un navegador.
- Los datos de usuario y denuncias serán almacenados en una base de datos.
- La aplicación mostrará las imágenes con su respectiva ubicación en el mapa.

Requerimientos no funcionales

- Obtener la información de los registros de las denuncias hechas por los usuarios.
- La aplicación será desarrollada para un entorno web.
- La aplicación debe ser responsive, para garantizar la adecuada visualización en múltiples computadoras y dispositivos móviles.

Metodología

La aplicación será desarrollada en el lenguaje de programación Python y junto con este se hará uso del micro-framework Flask, el cual está basado en el lenguaje mencionado anteriormente y gracias a su estructura, facilita el desarrollo de aplicaciones. En cuanto al procesamiento de los datos y/o información se utilizará MongoDB, que es una base de datos NoSQL, esto se refiere a que no se trabaja con un modelo relacional de entidades, sino que aquí se implementan colecciones como objetos y documentos. Una vez se desarrolle la aplicación, se hará necesaria la utilización de Docker, para desplegarla en contenedores.

En el desarrollo de la aplicación no sólo se tendrá en cuenta la utilización de herramientas de implementación a nivel general sino también a nivel interno, es decir, se utilizarán patrones de arquitectura y patrones de diseño.

Patrones de arquitectura

Los patrones de arquitectura se refieren a los esquemas de organización que son utilizados para estructurar un sistema a nivel general, es decir, se puede interpretar como un bosquejo de dicho sistema, en este caso, en la aplicación se implementará el patrón de arquitectura MVC (Model, View, Controller), el cual se basa en tres componentes fundamentales:

- **Modelo:** Corresponde a la manera en que se representarán los datos manejados dentro del sistema, es decir, generalmente aquí se realizan las consultas a la base de datos.
- **Vista:** Se centra en la parte visual del sistema, es decir, en la interfaz con la cual va a interactuar el usuario.
- **Controlador:** Se encarga de establecer una conexión entre el modelo y la vista, de tal manera que se gestione la información de la manera más pertinente para el usuario y se responda a sus peticiones.

Dentro del módulo Usuario se encontrará el archivo models.py en el que podemos visualizar la conexión a la base de datos y también views.py, en donde se encuentran todas las rutas que retornan a los html que vendría siendo todo lo que el usuario visualiza en la página y en app.py está la iniciación de cada módulo como intermediario de vista y controlador.

Patrones de diseño

Los patrones de diseño son soluciones para problemas típicos y recurrentes que nos podemos encontrar a la hora de desarrollar una aplicación. En nuestra aplicación se implementarán los siguientes patrones de diseño:

- **Factory:** Es un patrón de diseño creacional, su objetivo es proporcionar una interfaz para crear objetos en una superclase, pero permite que las subclases alteren el tipo de objetos que se crearán.

Aplicamos este patrón de diseño en nuestro código fuente al considerar cada módulo una app independiente y que en el archivo `app.py` se instancie todo.

- **Decoradores:** Es un patrón de diseño estructural, permite adjuntar nuevos comportamientos a los objetos al colocar estos objetos dentro de objetos de envoltura especiales que contienen los comportamientos.

Se implementó en nuestro código fuente en el archivo `decorators.py`, en el módulo `user` al envolver el decorador, lo cual logra ocultar que la función original ha sido decorada.

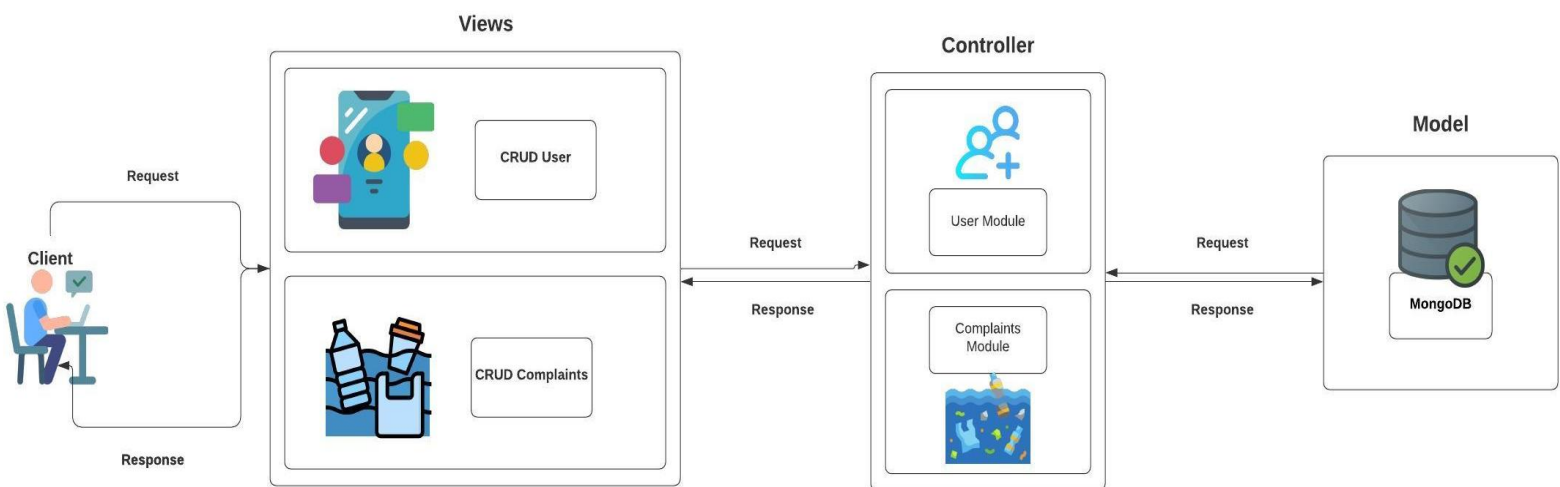
La aplicación se desarrollará a partir de Blueprint, esto consiste en separar la estructura general del proyecto a través de módulos, los cuales pueden contener cada uno los ficheros de acorde a la manera en que los desarrolladores lo deseen, en otras palabras, por medio de su utilización se puede hacer la instancia de un sólo objeto de Flask que haga uso de los diferentes Blueprint (Ficheros). Por consiguiente, los módulos que estarán presentes en la aplicación son los siguientes:

- **Usuario:** Este módulo está orientado a la creación y acceso de los usuarios, permitirá que exista un superusuario, el cual tiene los permisos necesarios para hacer control de usuarios, control de denuncias e imágenes.
- **Denuncia:** Este módulo se encargará de permitir ver los datos relacionados con las denuncias hechas por cada usuario. Se recibirán datos como imagen, datos de localización, descripción de la denuncia y nombre del usuario que la realiza.

Modelado de comunicaciones

A continuación, se podrá visualizar el modelado general de la aplicación, posteriormente se encontrará la descripción de cada uno de sus componentes y módulos.

Funcionamiento: Un cliente se conecta a las vistas de la aplicación, estas envían una petición que es recibida por el controlador, dependiendo del tipo de petición el controlador sabrá qué hacer. Por ejemplo: Si el cliente desea realizar un nuevo registro, el controlador recibe una petición POST que es enviada al controlador, este los recibe, le indica a la base de datos de los almacene y le da una respuesta al cliente que podrá observarse en las vistas. Existirá un superusuario que contará con permisos necesarios para el control de usuarios y denuncias.



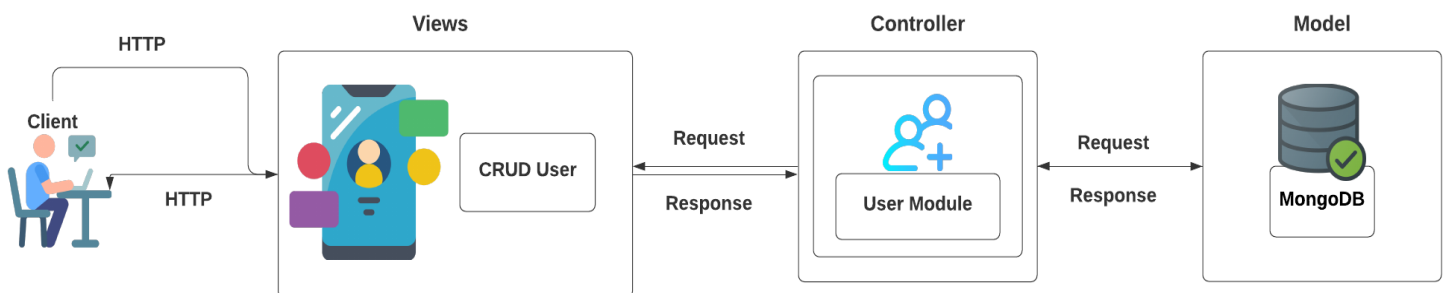
Descripción de componentes:

- **Cliente:** Cliente que se conecta a la aplicación y realiza peticiones.
- **Views:** Corresponden a las vistas de la aplicación, incluyendo tanto las de usuario como las de denuncia. Este componente está a su vez conformado por los CRUD de User y Complaints, el primero se encarga de realizar las operaciones CRUD relacionadas con los clientes, mientras que el segundo realiza estas mismas operaciones, pero con los datos de las denuncias. El cliente tendrá una conexión directa con este componente, ya que gestionará la interfaz correspondiente a cada petición.

- **Controller:** Son los controladores de la aplicación, se encuentra conformado por los módulos principales, los cuales corresponden a User Module y Complaints Module, estos serán los encargados de gestionar y responder a las peticiones que harán los clientes respecto a sus datos o a las denuncias, permitiéndoles visualizar estas acciones a través de la interfaz.
- **Model:** Hace referencia al modelo de la aplicación, se encuentra conformado por la base de datos MongoDB, la cual es una base de datos no relacional y que estará encargada de gestionar y almacenar los datos de la aplicación, es decir, los datos relacionados con las denuncias y los clientes.

Módulo usuario

Este módulo realiza operaciones CRUD relacionadas con los clientes, será el encargado de permitir la **creación y el acceso de los usuarios**, de tal manera que, realizando peticiones al **controlador**, este pueda gestionar una respuesta a través de la conexión con la base de datos **MongoDB**.



Los componentes principales de este módulo, como se visualiza en el gráfico son:

- **Cliente:** Cliente que se conecta a la aplicación y realiza peticiones.
- **CRUD User:** Corresponde a las operaciones CRUD relacionadas con los clientes, estas permiten la creación, el registro y la modificación de los datos suministrados. Al momento en el que el cliente realice una conexión con este componente, se habilitará la interfaz respectiva a cada una de las peticiones.

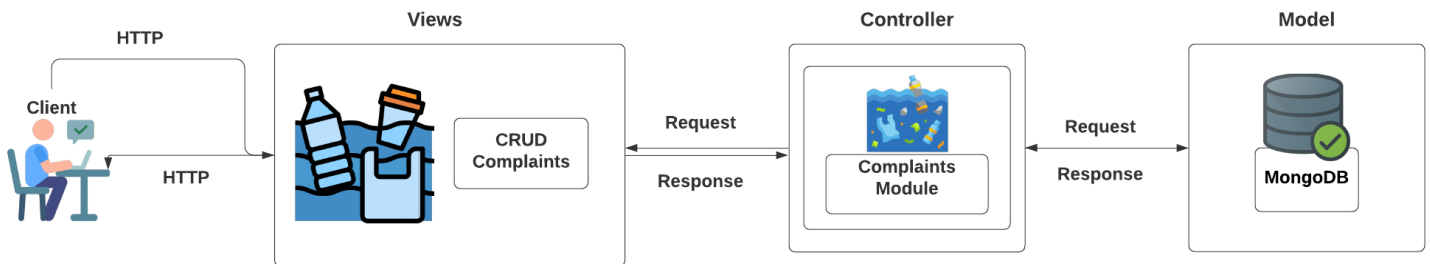
- **User Module (Módulo usuario):** Corresponde al controlador del módulo usuario, es decir, aquí se encuentra la lógica y el funcionamiento general de este módulo. Además, es el encargado de gestionar las peticiones, conexiones y respuestas que recibirá el cliente a través de la interfaz.
- **MongoDB:** Base de datos que podrá gestionar y almacenar toda la información pertinente respecto al cliente, tales como nombre, ciudad, e-mail y contraseña.

Rutas del Módulo Usuario:

- **/user:** Será la ruta principal del módulo.
- **/user/newregister ['GET','POST']:** Ruta de creación de un cliente
- **/user/login ['GET','POST'] :** Ruta de acceso del cliente.
- **/user/logout login ['GET'] :** Ruta para cerrar sesión.
- **/user/edit ['POST','GET']:** Ruta de edición de los datos del cliente.

Módulo denuncias

Será el módulo encargado de permitir al cliente tener **acceso o visualización de las denuncias** hechas por los demás usuarios a través de una conexión con el **controlador**, el cual se conectará a su vez a la base de datos **MongoDB** para gestionar y almacenar datos como imágenes, localización, descripción de la denuncia y cliente al cual pertenece ese registro.



Los componentes principales de este módulo son los siguientes:

- **Client:** Cliente que se conecta a la aplicación y realiza peticiones.
- **CRUD complaints:** Corresponde a las operaciones CRUD relacionadas con las denuncias, estas permiten su creación y modificación, además a través de una interfaz, permitirá que el cliente visualice el mapa con los registros de denuncias existentes.
- **Complaints Module (Módulo de denuncias):** Será el controlador del módulo de denuncias, será capaz de gestionar las peticiones, conexiones y respuestas que recibirán del cliente acerca de las denuncias realizadas o que desea registrar.
- **MongoDB:** Base de datos que podrá crear, registrar, eliminar y modificar datos relacionados con la denuncia fotográficas, tales como imagen, localización, descripción de la denuncia y cliente que realiza la denuncia.

Rutas del Módulo Denuncias:

- **/complaint:** Ruta del módulo denuncia.
- **/complaint/newcomplaints ['GET','POST']:** Ruta que permite la creación y registro de las denuncias.
- **/complaint/edit ['GET','POST']:** Ruta que permite modificar los datos registrados de la denuncia (Si el usuario ha iniciado sesión).
- **/complaint/map ['GET']:** Ruta que permite ver cada denuncia hecha en el mapa.

Colecciones de la base de datos MongoDB

Al implementar una base de datos NoSQL como lo es MongoDB es propicio destacar que en lugar de entidades aquí se trabaja con colecciones y documentos, lo primero corresponde al conjunto de características de un objeto y lo segundo es la manera de visualizar de manera conjunta esas características o atributos de una misma colección, además, MongoDB provee un Id automático para cada uno de los objetos que se creen, esto facilita las consultas que se quieran realizar en la base de datos. De acuerdo a esto, las colecciones y documentos existentes en la implementación de la aplicación se visualizan a continuación.

USUARIO

```
{ "id" : "Hash generado por Mongo",  
  "nombre" : "String",  
  "apellido" : "String",  
  "ciudad" : "String",  
  "e-mail" : "String",  
  "contraseña" : "String-Hash"  
}
```

DENUNCIA

```
{ "id" : "Hash generado por Mongo",  
  "e-mail" : "String",  
  "ciudad" : "String",  
  "titulo" : "String",  
  "descripcion" : "String",  
  "imagen" : file,  
  "ubicacion" : [ coordenadas ]  
}
```

MAPA

```
{ "id" : "Hash generado por Mongo",  
  "e-mail" : "String",  
  "ciudad" : "String",  
  "imagen" : file,  
  "ubicacion" : [ coordenadas ]  
}
```

Diseño (Vistas)

- Vista principal



Bienvenido a Denuncias App

Esta página fue creada con el fin de ver y registrar las denuncias relacionadas con los caños contaminados en la ciudad de Cartagena.



- Vista de Inicio de sesión



- Vista de registrar usuario

Registrar

Nombre de Usuario

Ingresar su usuario

Nombre

Ingresar su nombre

Apellido

Ingresar su apellido

Ciudad

Ingresar su ciudad

Correo

Ingresar su correo

Contraseña

Ingresar su contraseña

Ingrese nuevamente la contraseña

Ingresar nuevamente su contraseña

Ingresar

