TRABAJO DE INVESTIGACION PYTHON Y DJANGO

YAIR ALEXIS GUATAPO

ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN

CENTRO DE TELEINFORMÁTICA Y PROCUCCIÓN INDUSTRIAL

CTPI

SERVICIO NACIONAL DE APRENDIZAJE-SENA

POPAYÁN-CAUCA

2017

TRABAJO INVESTIGACION

YAIR ALEXIS GUATAPO

ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN

INGENIERO:

DIEGO PRADO

CENTRO DE TELEINFORMATICA DE PROCUCCIÓN INDUSTRIAL

CTPI

SERVICIO NACIONAL DE APRENDIZAJE-SENA

POPAYÁN-CAUCA

2017

TABLA DE CONTENIDO

Pag.

1. INTRODUCCIÓN…….……………………………………………………………4
2. TRABAJO INVESTIGACION………………………....……………..………...5-9

Vistas basadas en clases

Servidor web

1. BIBLIOGRAFIA………………………………………………………………….10

INTRODUCCION

En este trabajo encontraremos uno de los cambios más importantes que se han producido en las vistas. Las vistas ahora son clases en lugar de ser funciones. Si queremos podemos seguir usando funciones pero las clases pueden ser rapidas y fáciles de usar.

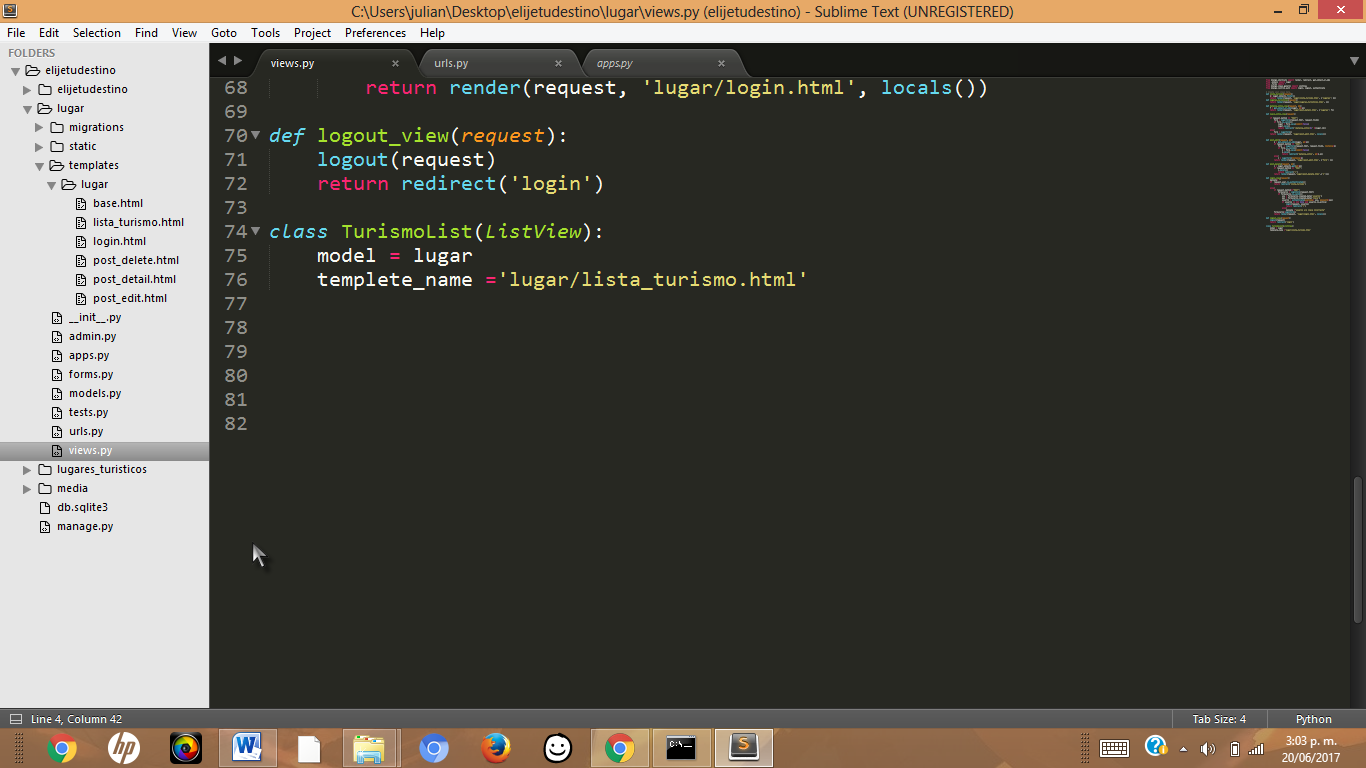
VISTAS BASADAS EN CLASES

Una vista es una invocable que toma una petición y devuelve una respuesta. Esto puede ser más que una función, y Django proporciona un ejemplo de algunas clases que se pueden utilizar como puntos de vista. Estos le permiten estructurar sus puntos de vista y la reutilización de código mediante el aprovechamiento de la herencia y mixins. También hay algunas vistas genéricas para tareas simples que vamos a llegar a más tarde, pero es posible que desee diseñar su propia estructura de puntos de vista reutilizables que se adapte a su caso de uso.

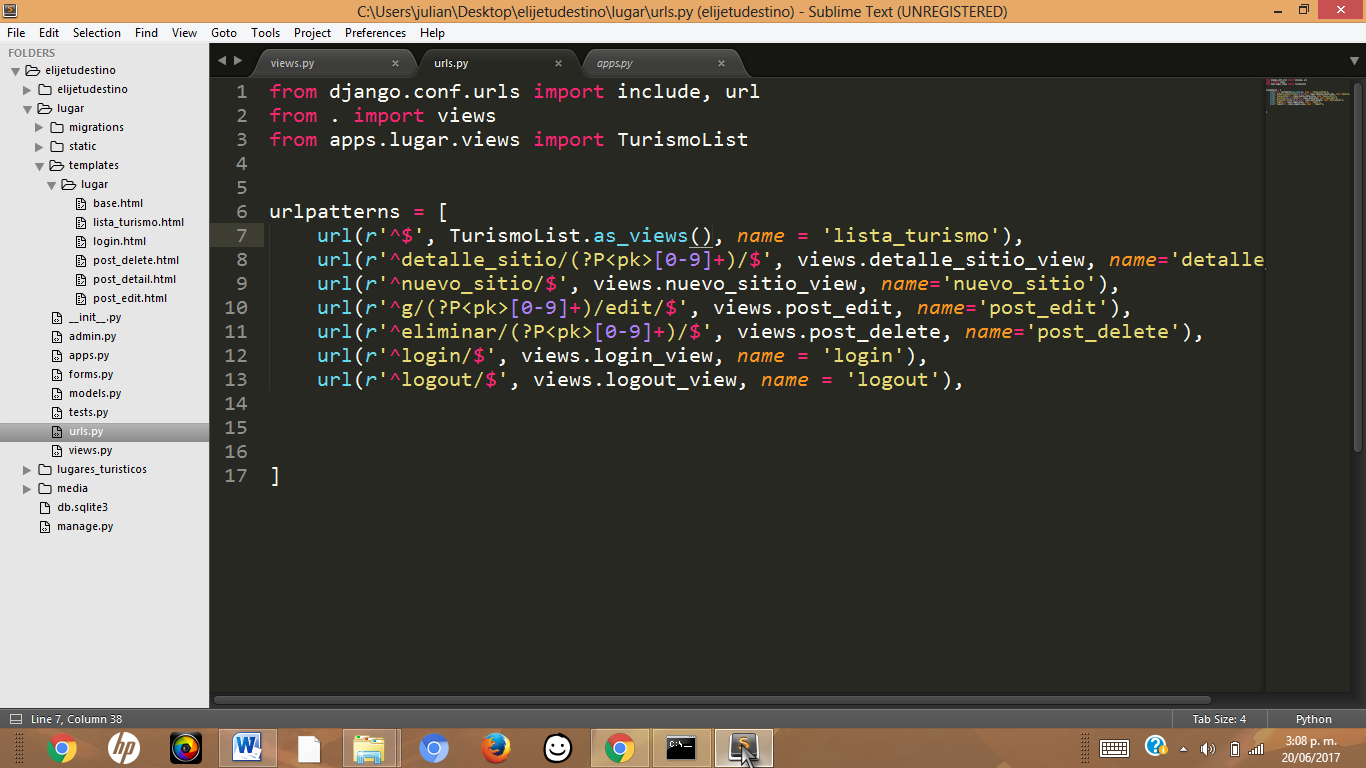
Django proporciona clases de vista de base que se adapte a una amplia gama de aplicaciones. Todas las vistas heredan de la [**View**](https://docs.djangoproject.com/en/1.11/ref/class-based-views/base/#django.views.generic.base.View) clase, que maneja la vinculación de la vista en la URL, despachar métodos HTTP y otras características simples. [**RedirectView**](https://docs.djangoproject.com/en/1.11/ref/class-based-views/base/#django.views.generic.base.RedirectView) es por una simple redirección HTTP, y [**TemplateView**](https://docs.djangoproject.com/en/1.11/ref/class-based-views/base/#django.views.generic.base.TemplateView)  se extiende la clase base para que sea también hacen una plantilla.

**LISTA DE LUGARES**

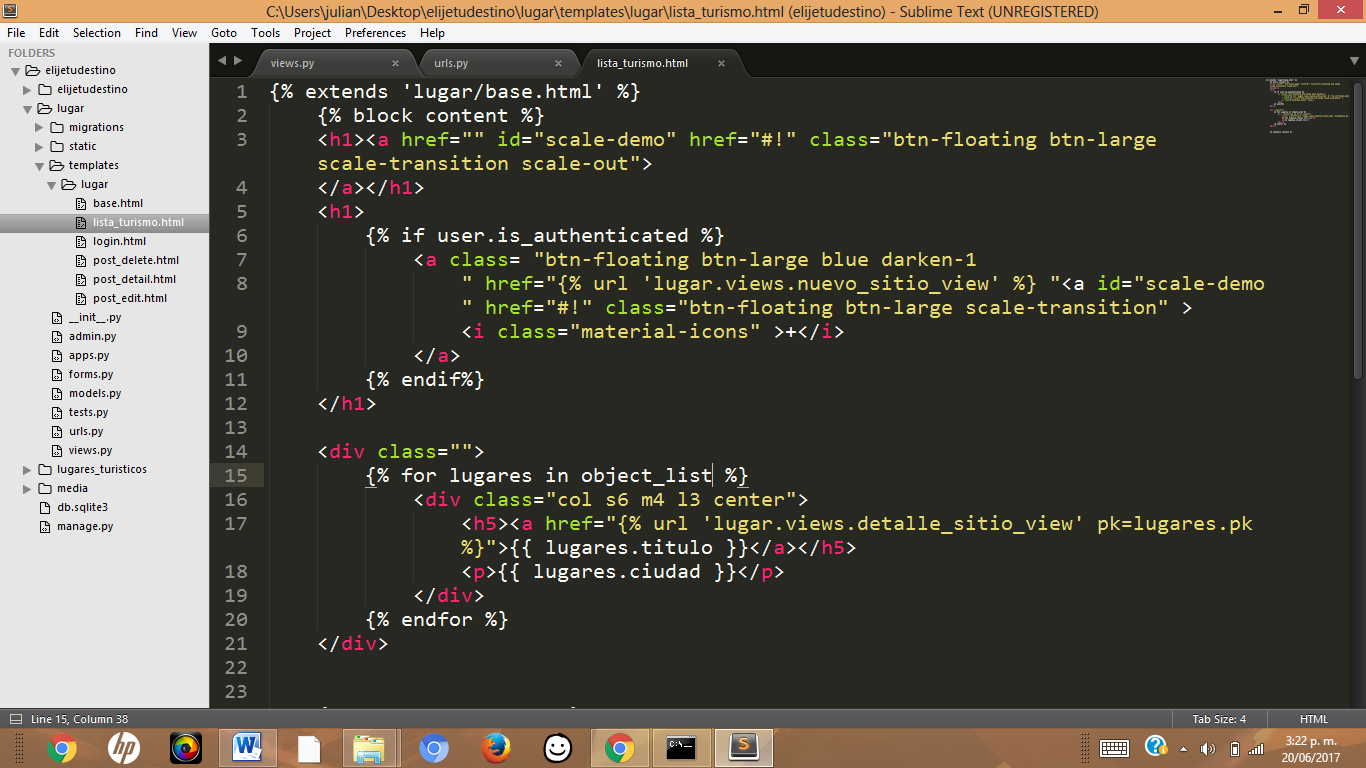
**Vista de la clase lista de sitios turísticos**



**Importamos la vita de clase TurismoList y cambiamos la url.**



**En el témplate cambiamos lugares por object\_list.**

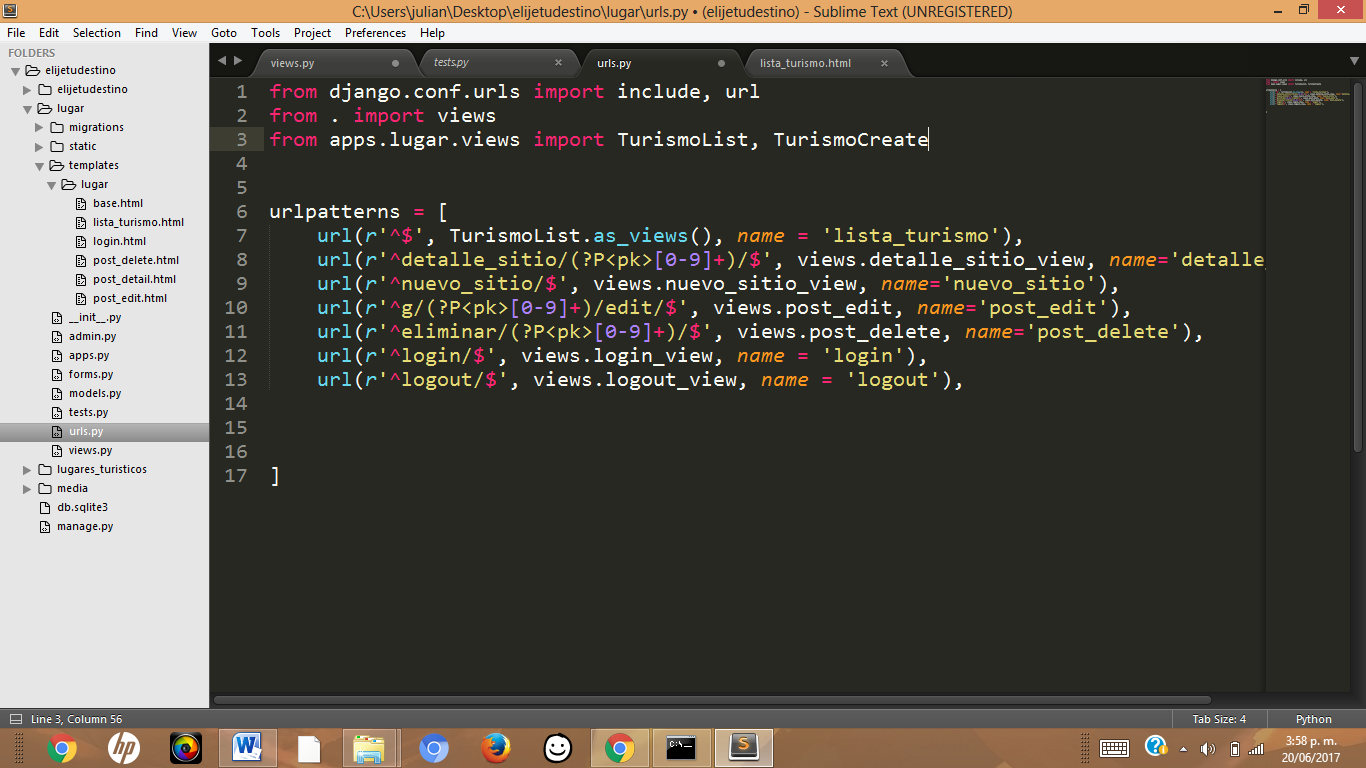


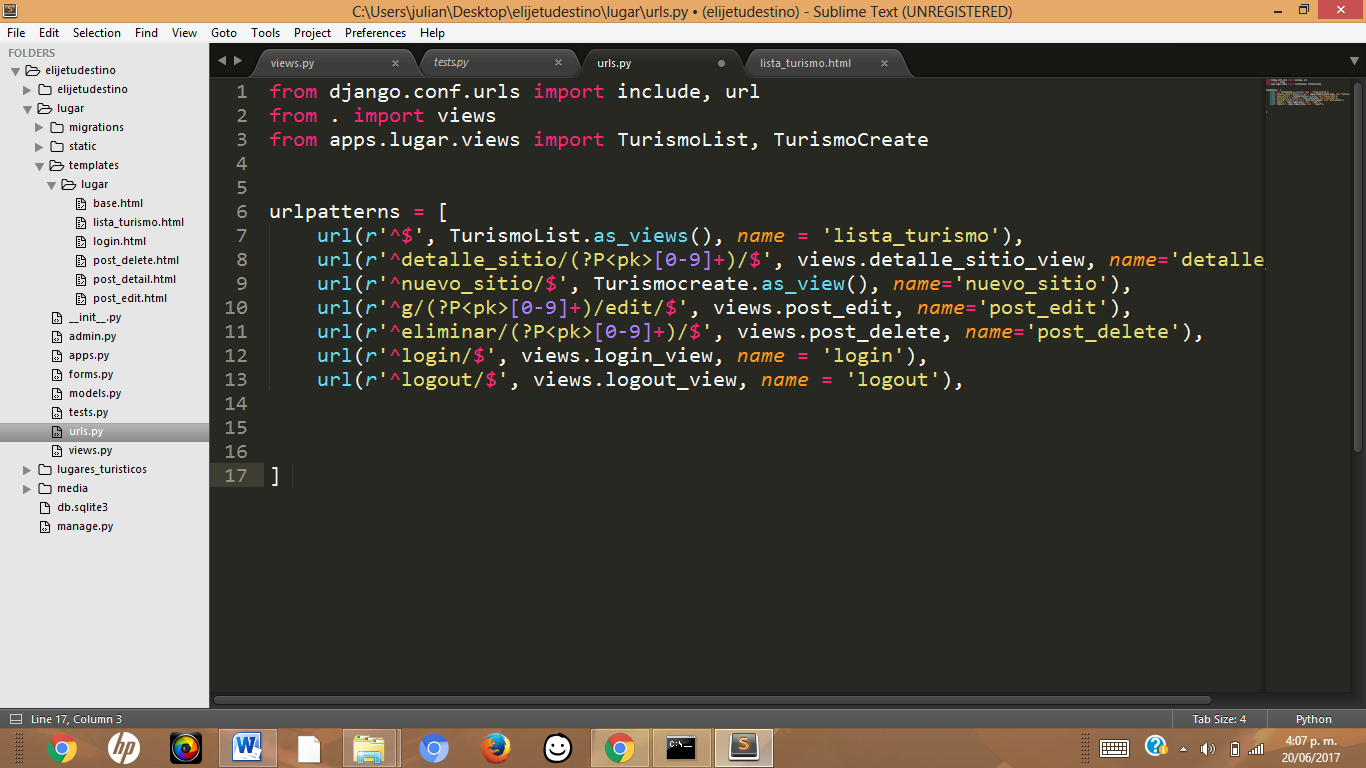
**AGREGAR SITIO NUEVO**

**Vista de la clase agregar sitios turísticos**



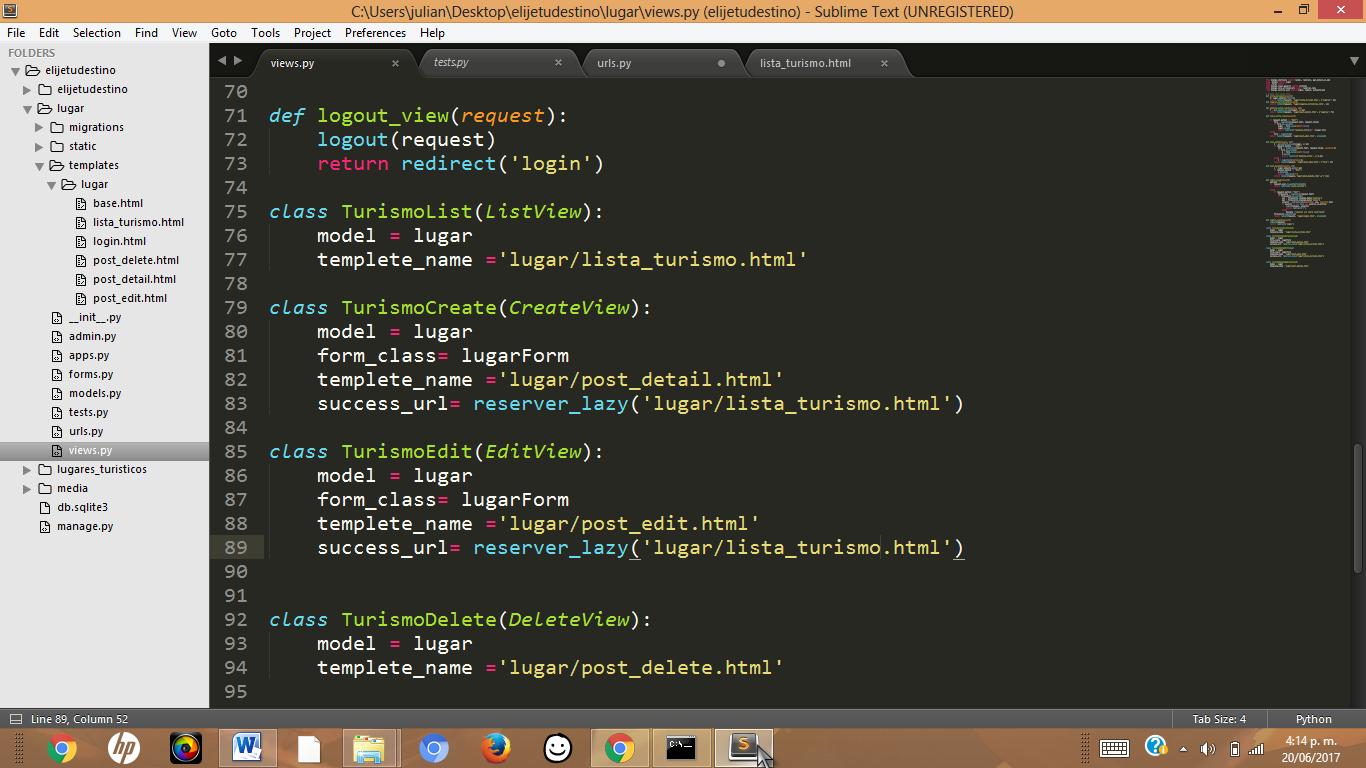
**Importamos la vita de clase TurismoCreate y cambiamos la url.**



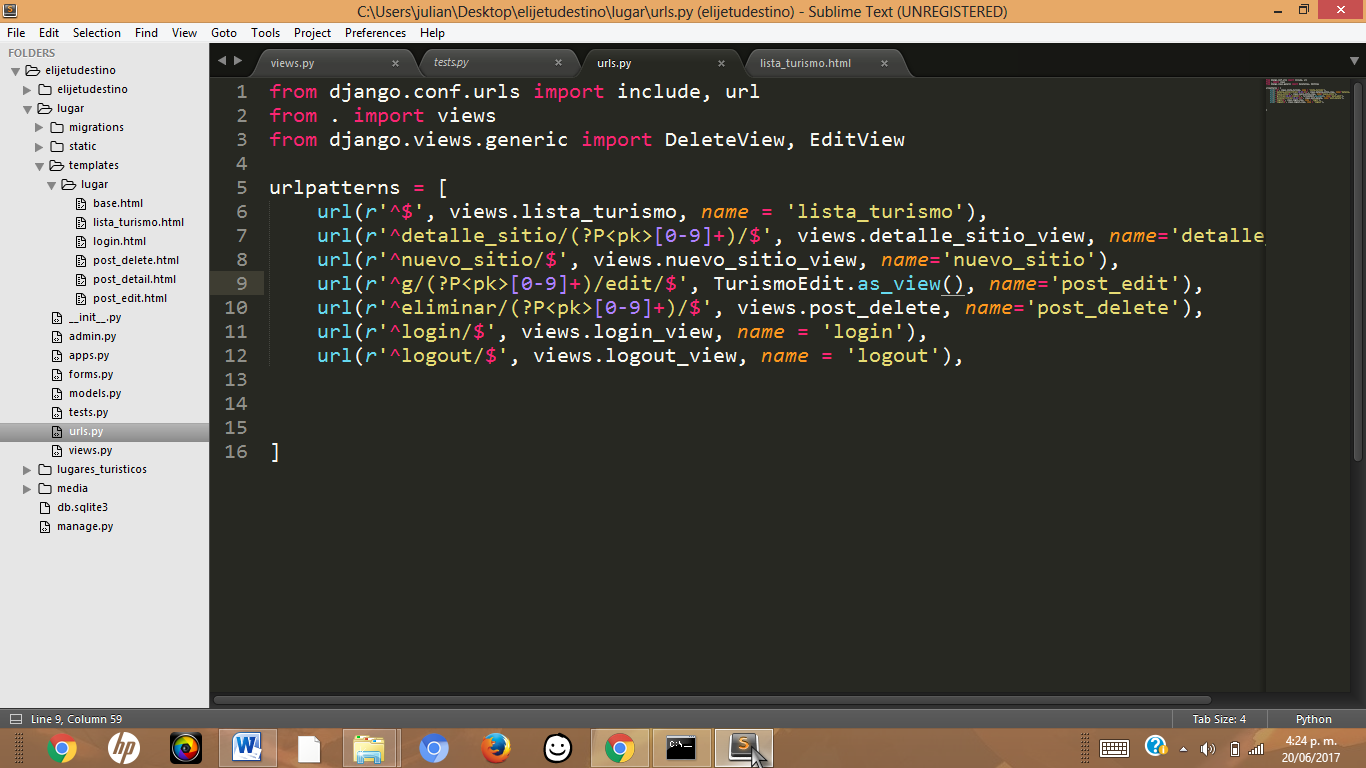


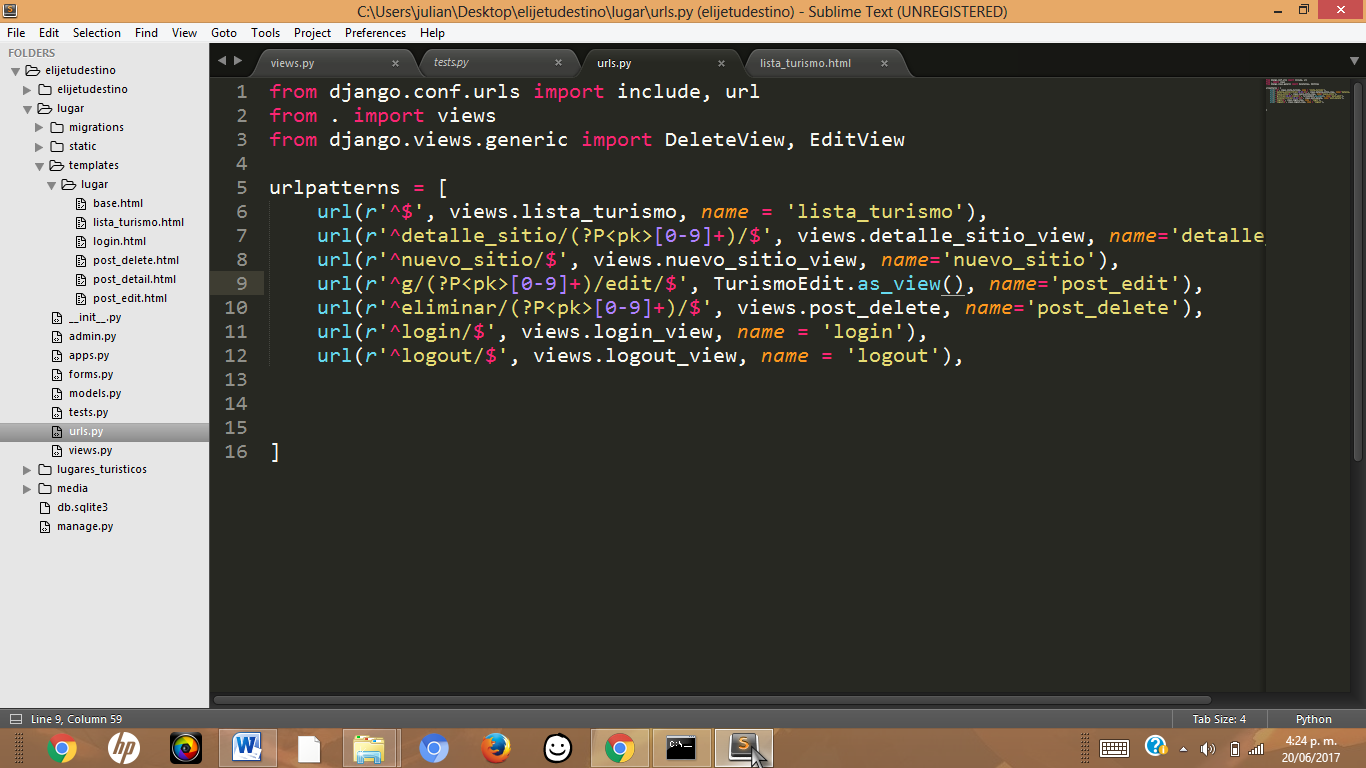
**EDITAR SITIO TURISTICO**

**Vista de la clase editar sitios turísticos**



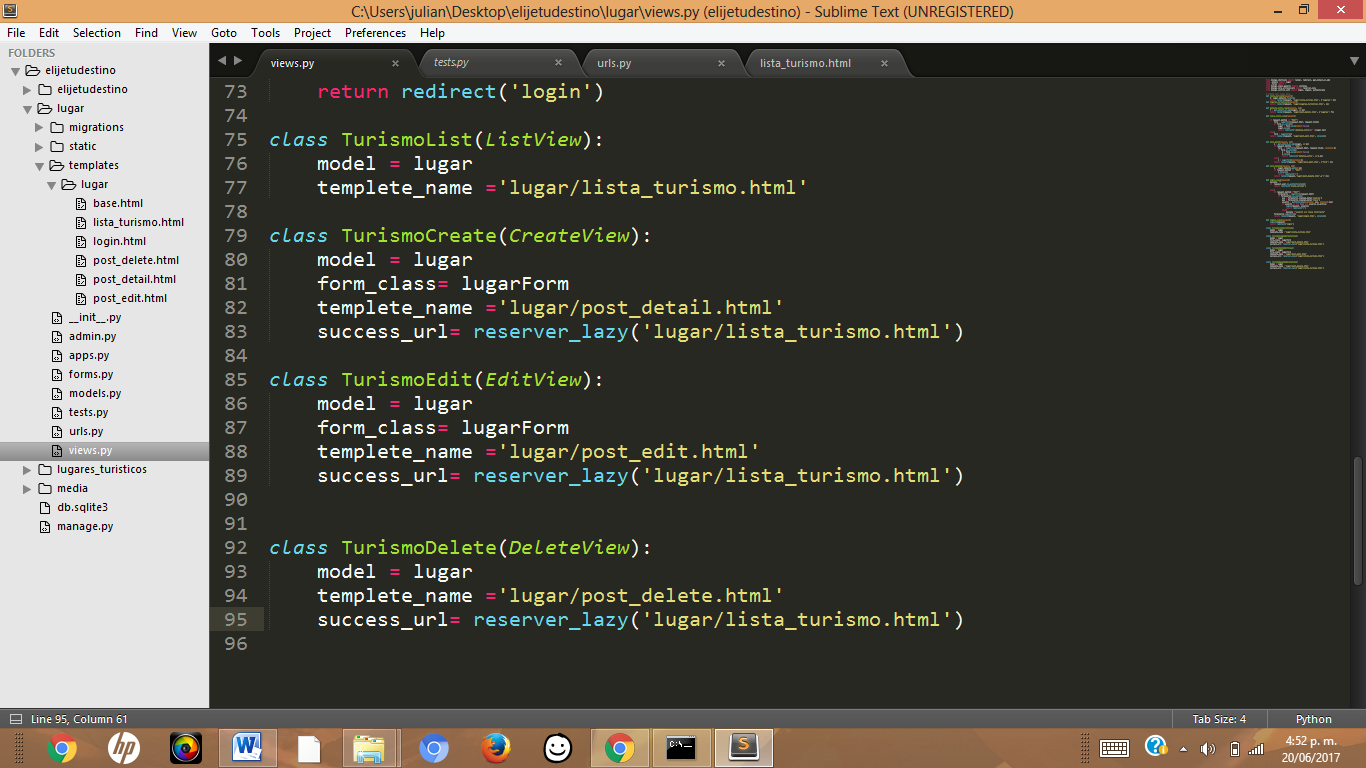
**Importamos la vita de clase TurismoEdit y cambiamos la url.**



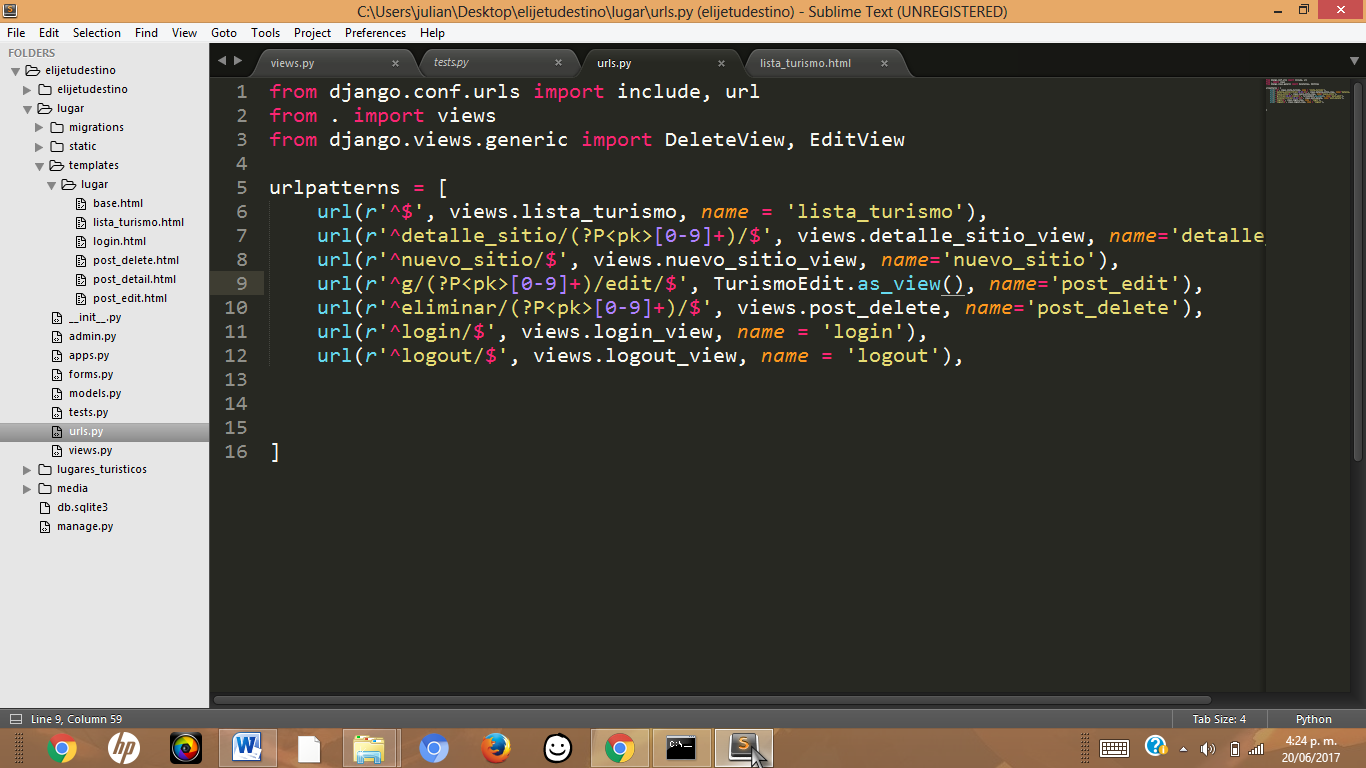


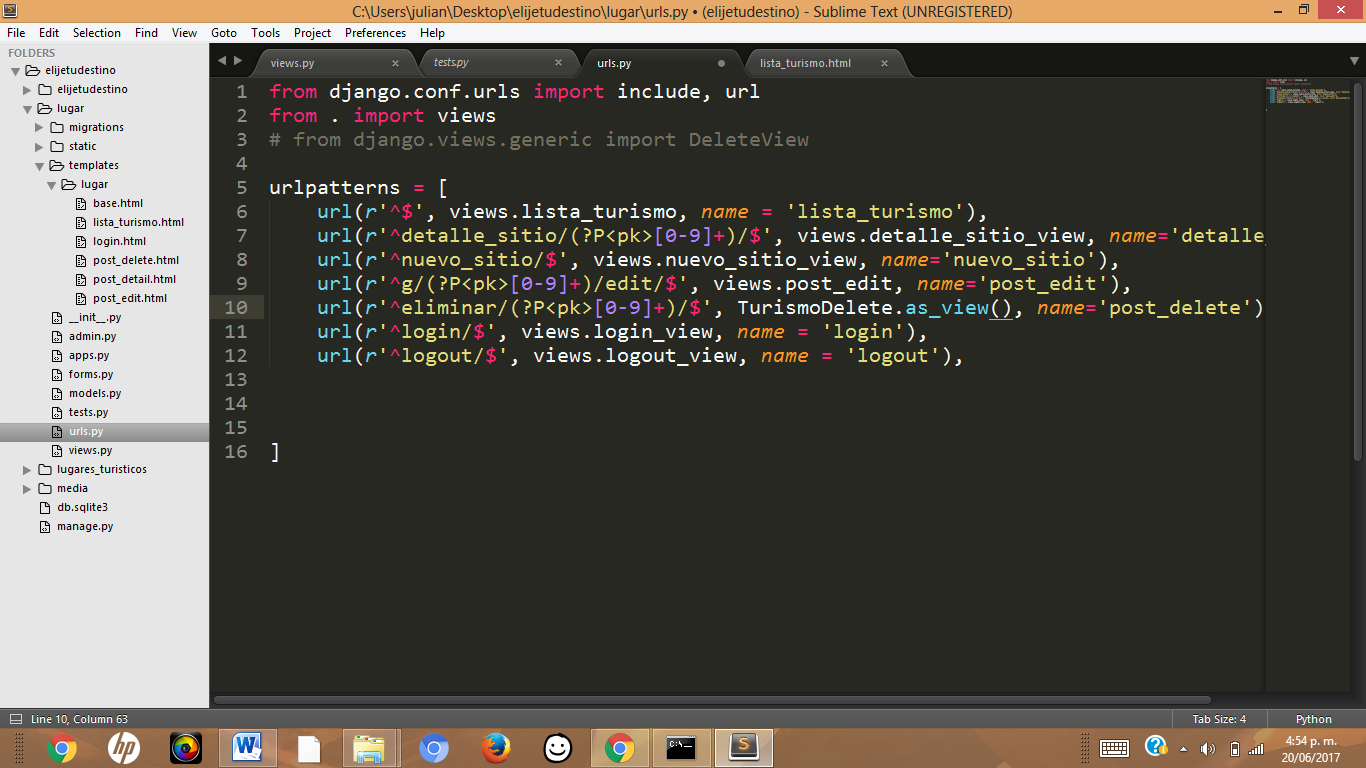
**ELIMINAR SITIO TURISTICO**

**Vista de la clase editar sitios turísticos**



**Importamos la vita de clase TurismoEdit y cambiamos la url.**



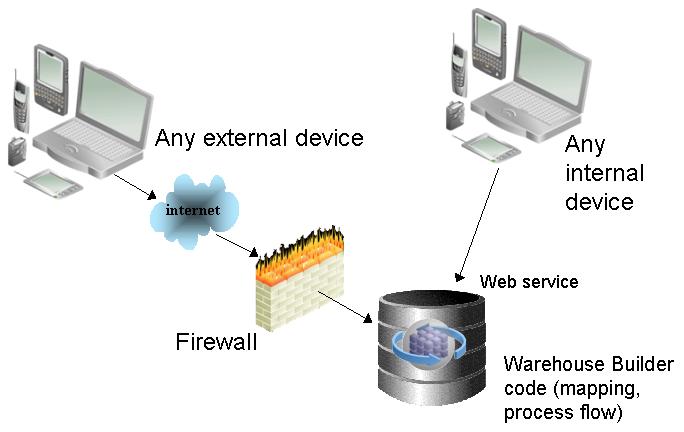


SERVIDOR WEB

Es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como internet.

De una manera más clara se podría decir que un web service es una función que diferentes servicios o equipos utilizan; es decir, solo se envían parámetros al servidor (lugar donde está alojado el web service) y éste responderá la petición. Entre algunas que se manejan de utilizar servicios webs en las aplicaciones destacan las siguientes:

* Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
* Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
* Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
* Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
* Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos. Las especificaciones son gestionadas por una organización abierta, la W3C, por tanto no hay secretismos por intereses particulares de fabricantes concretos y se garantiza la plena interoperabilidad entre aplicaciones.



BIBLIOGRAFIA

Curso Django- vistas basadas en clases

([https://docs.djangoproject. com/en/1.11/topics/class- based-views/](https://docs.djangoproject.com/en/1.11/topics/class-based-views/))