

## Informe de Laboratorio 06

**Tema: Aplicacion Destinos Turisticos con Django**

Nota

Estudiante	Escuela	Asignatura
Ower Frank Lopez Arela olopeza@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: III Código: 20222083

Laboratorio	Tema	Duración
06	Aplicacion Destinos Turisticos con Django	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - A	Del 14 Junio 2023	Al 28 Junio 2023

### 1. Tarea

- Deberán replicar la actividad del video donde se obtiene una plantilla de una aplicación de Destinos turísticos y adecuarla a un proyecto en blanco Django.
- Luego trabajar con un modelo de tabla DestinosTuristicos donde se guarden nombreCiudad, descripcionCiudad, imagenCiudad, precioTour, ofertaTour (booleano). Estos destinos turísticos deberán ser agregados en una vista dinámica utilizando tags for e if.
- Para ello crear una carpeta dentro del proyecto github colaborativo con el docente, e informar el link donde se encuentra (compartir con el usuario CarloCorralesD) . El trabajo será revisado en laboratorios los 2 siguientes semanas hasta el miércoles 28 de junio y se presentará el trabajo final en Flipgrid.



## 2. URL de Repositorio Github

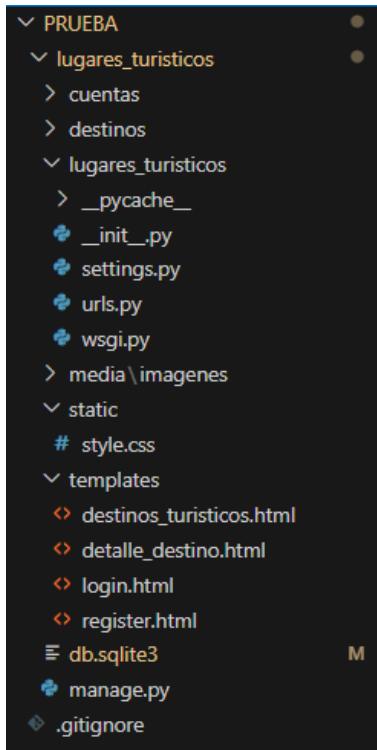
- URL para el laboratorio 06 en el Repositorio GitHub.
- <https://github.com/0werFrankLopezArela/LAB06-PWEB2.git>

## 3. Ejercicios

### 3.1. Estructura de laboratorio 06

- El contenido que se entrega en este laboratorio es el siguiente:

```
C: .
+-----lugares_turisticos
|-----cuentas
|     |-----migrations
|     +-----__pycache__
|     +-----__pycache__
|-----destinos
|     |-----migrations
|     +-----__pycache__
|     +-----__pycache__
|-----lugares_turisticos
|     +-----__pycache__
|-----media
|     +-----imagenes
|-----static
+-----templates
```



### 3.2. Página de lugares turísticos utilizando Django:

- Especificar paso a paso la creación de la pagina en su informe..
- Crear un video tutorial donde realice las operaciones CRUD (URL public reproducible online).
- Adjuntar URL del video en el informe.

#### Listing 1: CONFIGURACION DEL PROYECTO DJANGO

Crea un proyecto Django ejecutando los siguientes comandos en tu terminal:

```
$ django-admin startproject lugares_turisticos
$ cd lugares_turisticos
```

#### Listing 2: CREACION DE LA APLICACION

Ahora, crea una aplicación llamada "destinos" ejecutando el siguiente comando:

```
$ python manage.py startapp destinos
```

#### Listing 3: CONFIGURANDO LA BASE DE DATOS

En este proceso, se configura la base de datos predeterminada (sqlite3) en el archivo "mysite/settings.py". Se utiliza la configuración por defecto que viene con Django. Luego, se ejecuta el comando "python manage.py migrate." en la consola, estando en el directorio que contiene el archivo "manage.py". Este comando aplica todas las migraciones necesarias para la base de datos, creando las tablas correspondientes. Si todo va bien, se muestra en la consola una serie de mensajes que indican la ejecución exitosa de las migraciones. Después de este paso, se puede iniciar el servidor web para verificar el funcionamiento del sitio web.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

#### Listing 4: DEFINICION DEL MODELO DE TABLA

Abre el archivo models.py en la carpeta destinos y define el modelo DestinosTuristicos con los campos requeridos: nombreCiudad, descripcionCiudad, imagenCiudad, precioTour y ofertaTour.

```
# models.py

from django.db import models

class DestinosTuristicos(models.Model):
    nombreCiudad = models.CharField(max_length=100)
    descripcionCiudad = models.TextField()
```

```
imagenCiudad = models.ImageField(upload_to='destinos/')
precioTour = models.DecimalField(max_digits=8, decimal_places=2)
ofertaTour = models.BooleanField(default=False)

def __str__(self):
    return self.nombreCiudad
```

---

#### Listing 5: REALIZAR LAS MIGRACIONES Y APLICAR CAMBIOS

---

Ejecuta los siguientes comandos para crear las migraciones y aplicar los cambios en la base de datos:

```
$ python manage.py makemigrations
$ python manage.py migrate
```

---

#### Listing 6: CONFIGURACION DE ARCHIVOS ESTATICOS Y MEDIOS

---

Abre el archivo settings.py y agrega las siguientes configuraciones para los archivos estáticos y medios:

```
# settings.py

...
STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'static')

MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

...
```

---

#### Listing 7: CREACION DE VISTAS Y PLANTILLAS

---

Abre el archivo views.py en la carpeta destinos y define una vista llamada destinos\_turisticos que obtenga todos los destinos turísticos de la base de datos.

```
# views.py

from django.shortcuts import render
from .models import DestinosTuristicos

def destinos_turisticos(request):
    destinos = DestinosTuristicos.objects.all()
    return render(request, 'destinos_turisticos.html', {'destinos': destinos})
```

Luego, crea un archivo de plantilla llamado destinos\_turisticos.html en una carpeta llamada templates/destinos dentro de la aplicación destinos. Este archivo contendrá el código HTML y las etiquetas de Django necesarias para mostrar los destinos turísticos.

```
<!-- destinos_turisticos.html -->

{% for destino in destinos %}
```

```
<div>
    <h2>{{ destino.nombreCiudad }}</h2>
    <p>{{ destino.descripcionCiudad }}</p>
    
    <p>Precio: ${{ destino.precioTour }}</p>
    {% if destino.ofertaTour %}
        <p>Oferta especial!</p>
    {% endif %}
</div>
{% empty %}
    <p>No hay destinos turísticos disponibles.</p>
{% endfor %}
```

#### **Listing 8: CONFIGURACION DE LAS URL'S**

Abre el archivo urls.py en la carpeta lugares\_turisticos y configura las URL para la aplicación destinos.

```
# urls.py

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('destinos/', include('destinos.urls')),
]
```

Luego, crea un archivo llamado urls.py en la carpeta destinos y define la URL para la vista destinos\_turisticos.

```
# urls.py

from django.urls import path
from . import views

app_name = 'destinos'

urlpatterns = [
    path('', views.destinos_turisticos, name='destinos_turisticos'),
]
```

#### **Listing 9: EJECUCION DEL SERVIDOR**

Finalmente, ejecuta el servidor de desarrollo de Django con el siguiente comando:

```
$ python manage.py runserver
```

---

#### Listing 10: REGISTRAR EL ADMIN DE LA APLICACION

---

En el archivo admin.py se registró los modelos de la aplicación destinos para que puedan ser administrados a través de la interfaz de administración de Django.

```
from django.contrib import admin
from .models import DestinosTuristicos

admin.site.register(DestinosTuristicos)
```

---

#### Listing 11: REGISTRAR EL MODELS.PY DE LA APLICACION

---

Define un modelo llamado DestinosTuristicos que representa destinos turísticos en una base de datos. Este modelo tiene campos como el nombre de la ciudad, la descripción, la imagen, el precio del tour y una bandera para indicar si hay una oferta disponible. El campo de imagen permite subir y almacenar imágenes relacionadas con el destino turístico. El método `__str__()` devuelve el nombre de la ciudad como una representación legible del objeto DestinosTuristicos.

```
from django.conf import settings
from PIL import Image
from django.db import models

# Create your models here.
class DestinosTuristicos(models.Model):
    nombreCiudad = models.CharField(max_length=100)
    descripcionCiudad = models.TextField()
    imagenCiudad = models.ImageField(upload_to='imagenes/')
    precioTour = models.DecimalField(max_digits=8, decimal_places=2)
    ofertaTour = models.BooleanField(default=False)

    def __str__(self):
        return self.nombreCiudad
```

---

#### Listing 12: DEFINICION DE LAS URL'S

---

En el archivo urls.py, se importa el módulo path de django.urls y el módulo views del mismo directorio. Luego, se define una lista llamada urlpatterns que contiene dos rutas de URL. La primera ruta corresponde a la página principal y está asociada a la vista destinos\_turisticos, y se le asigna el nombre 'destinos\_turisticos'. La segunda ruta tiene un patrón dinámico `:int:destino_id`, que captura un entero como parámetro y está asociada a la vista detalle\_destino, y se le asigna el nombre 'detalle\_destino'. En resumen, este código establece las rutas de URL para acceder a diferentes vistas de la aplicación web, como la página principal y los detalles de un destino turístico específico.

```
# urls.py

from django.urls import path
from . import views

urlpatterns = [
    path('', views.destinos_turisticos, name='destinos_turisticos'),
    path('destino/<int:destino_id>/', views.detalle_destino,
         name='detalle_destino'),
```

]

### **Listing 13: CREAMOS LAS VISTAS DE NUESTRA APLICACION**

La primera vista, destinos\_turisticos, recupera todos los objetos de la clase DestinosTuristicos de la base de datos y los pasa a la plantilla destinos\_turisticos.html como contexto. Esta vista se encarga de mostrar una lista de destinos turísticos en la página principal de la aplicación.

La segunda vista, detalle\_destino, recibe un parámetro destino\_id que representa la clave primaria de un destino turístico específico. Utilizando la función get\_object\_or\_404, se obtiene el objeto DestinosTuristicos correspondiente a ese destino\_id o se muestra una página de error si no se encuentra. Luego, se pasa el objeto destino a la plantilla detalle\_destino.html como contexto. Esta vista se encarga de mostrar los detalles de un destino turístico específico.

```
from PIL import Image
from django.shortcuts import render, redirect
from .models import DestinosTuristicos
from django.shortcuts import render, get_object_or_404

# Create your views here.

def destinos_turisticos(request):
    destinos = DestinosTuristicos.objects.all()
    return render(request, 'destinos_turisticos.html', {'destinos': destinos})

def detalle_destino(request, destino_id):
    destino = get_object_or_404(DestinosTuristicos, pk=destino_id)
    return render(request, 'detalle_destino.html', {'destino': destino})
```

### **Listing 14: CREAMOS EL PRIMER TEMPLATE, DESTINOS\_TURISTICOS.HTML**

**Encabezado y navegación:** En esta parte del código, se define el encabezado de la página y la sección de navegación. El encabezado incluye un logotipo representado por la etiqueta `<img>` con la URL de la imagen. Luego, dentro de la etiqueta `<nav>`, se crea una lista desordenada `ul` que contiene enlaces a diferentes páginas. Dependiendo del estado de autenticación del usuario, se muestran opciones diferentes. Además, hay un enlace representado por el ícono de un carrito de compras.

```
{% load static %}
{% static "images" as baseUrl %}

<!DOCTYPE html>
<html>
<head>
    <title>Lugares Tursticos</title>
    <link rel="stylesheet" href="{% static 'style.css' %}">
</head>
<body>

    <header>
        <div class="logo">
            
    </div>
    <nav>
        <ul>
            <li><a href="http://127.0.0.1:8000/" class="sub">Inicio</a></li>

            {%
                if user.is_authenticated %}
                    <li class="sub welcome-message">Hello,
                        <span>{{user.first_name}}</span></li>
                    <li><a href="cuentas/logout" class="sub">Salir</a></li>
                {% else %}
                    <li><a href="http://127.0.0.1:8000/cuentas/register"
                        class="sub">Registrarse</a></li>
                    <li><a href="http://127.0.0.1:8000/cuentas/login"
                        class="sub">Ingresar</a></li>
                {% endif %}

                <li><a href="#" class="carrito" data-counter="0">#128722;</a></li>
            </ul>
        </nav>
    </header>
```

**Listing 15: DESTINOS\_TURISTICOS.HTML: Contenido principal**

La sección principal de la página se encuentra dentro del bloque % block content %. Aquí se presenta información sobre lugares turísticos en Perú. Se muestra un encabezado con el título "LUGARES TURISTICOS" una sección dividida en dos columnas. La primera columna contiene una imagen utilizando la etiqueta con la URL de la imagen del lugar turístico. La segunda columna contiene varios párrafos que describen diferentes aspectos de los lugares turísticos en Perú.

```

{%
    block content %}
    <div class="turismo-peru1">
        <h2>LUGARES TURISTICOS</h2>
    </div>

    <div class="turismo-peru">
        <div class="content">
            <div class="image">
                
            </div>
            <div class="text">
                <p>Descubre las maravillas de Perú, un país repleto de bellezas naturales. Desde la imponente ciudadela de Machu Picchu, una de las nuevas maravillas del mundo, hasta la enigmática línea de Nazca, cada rincón te sorprenderá.</p>
                <p>En la costa peruana, disfruta de playas de arenas doradas y aguas cristalinas como Mancora y Punta Sal, ideales para relajarse y practicar deportes acuáticos. Explora Lima, con su gastronomía exquisita y miradores con vistas impresionantes al Océano Pacífico.</p>
                <p>En la sierra andina, explora la antigua capital inca de Cusco y el Valle Sagrado, con impresionantes ruinas y pintorescos pueblos. No te pierdas la joya arqueológica rodeada de montañas y misterio: Machu Picchu, que te transportará en el tiempo.</p>
            </div>
        </div>
    </div>
```

```

<p>En la selva amazónica, la biodiversidad te espera. Aventrate en un recorrido por el río Amazonas, descubre la flora y fauna en reservas naturales como Tambopata o Iquitos, y vive experiencias únicas en contacto con la naturaleza, como avistar delfines rosados y explorar senderos selváticos. </p>
<p>En el norte del país, explora las ruinas de Chan Chan y las pirámides de Túcume para adentrarte en la historia precolombina. Disfruta de hermosas playas como Huanchaco, famosa por sus caballitos de totora, y descubre Trujillo, con su arquitectura colonial y casas coloridas. </p>
<p>Perú te espera con los brazos abiertos para que descubras su cultura, historia y naturaleza impresionante. No pierdas la oportunidad de visitar estos lugares turísticos y vivir experiencias inolvidables!</p>
</div>
</div>
</div>
```

**Listing 16: DESTINOS\_TURISTICOS.HTML: Viajes nacionales:**

En esta parte del código, se presenta otra sección con el título "VIAJES NACIONALES". Se utiliza la etiqueta `<main>` para delimitar el contenido principal de la página. Dentro de esta sección, se muestra una cuadrícula de destinos turísticos utilizando un bucle `% for destino in destinos %`. Cada destino se muestra como un cuadro que incluye el nombre de la ciudad, una imagen del lugar, el precio del tour y, en caso de haber una oferta, se muestra un texto indicando que está en oferta. Además, se proporciona un enlace "VER" que redirige al usuario a la página de detalle del destino específico.

```

<div class="turismo-peru1">
    <h2>VIAJES NACIONALES</h2>
</div>

<main>
    <div class="destinos-grid">
        {% for destino in destinos %}
            <div class="destino">
                <h2>{{ destino.nombreCiudad }}</h2>
                
                <p>Precio: {{ destino.precioTour }}</p>
                {% if destino.ofertaTour %}
                    <p class="oferta">En oferta!</p>
                {% endif %}
                <a href="{% url 'detalle_destino' destino.id %}" class="ver-btn">VER</a>
            </div>
        {% endfor %}
    </div>
</main>
```

**Listing 17: DESTINOS\_TURISTICOS.HTML: Visitar lugares y pie de página:**

La última sección del código presenta información sobre "VISITAR LUGARES" que muestra un mapa incrustado utilizando la etiqueta `<iframe>`. El mapa muestra la ubicación de los lugares turísticos mencionados anteriormente. Finalmente, en el pie de página, se muestra un texto de derechos de autor y créditos del creador del sitio web.

```

<div class="turismo-peru1">
    <h2>VISITAR LUGARES</h2>
</div>

<div class="turismo-peru3">
    <div align="center">
        <iframe
            src="https://www.google.com/maps/d/embed?mid=1rpFrIXZgjMaukWY1wfwj0BEow7D4sB4&ehbc=2E312F"
            width="640" height="480"></iframe> </div>
    </div>
    {% endblock %}

    <footer>
        <p> 2023 Lugares Tursticos. Todos los derechos reservados. Hecho por Ower Lopez</p>
    </footer>
</body>
</html>

```

**Listing 18: DETALLE\_DESTINO.HTML**

El código proporcionado extiende una plantilla llamada "destinos\_turisticos.html" que carga los archivos estáticos necesarios. Dentro del bloque de contenido, se crea una nueva sección para mostrar el detalle de un destino turístico específico.

Dentro de un contenedor centrado, se muestra la imagen del destino utilizando la URL proporcionada en la variable destino.imagenCiudad.url. A continuación, se muestra la descripción del destino, incluyendo el nombre de la ciudad y una descripción más detallada.

En la sección de precio y oferta, se muestra el precio del tour utilizando la variable destino.precioTour. Si el destino tiene una oferta disponible, se muestra el texto "¡En oferta!". Luego, se presenta un botón de compra con un enlace que redirige al usuario a la página de inicio de sesión.

```

{% extends 'destinos_turisticos.html' %}
{% load static %}

{% block content %}

<div class="centered-container">
    <div class="detalle-destino">
        <div class="imagen">
            
        </div>
        <div class="descripcion-precio">
            <div class="descripcion">
                <h2>{{ destino.nombreCiudad }}</h2>
                <p>{{ destino.descripcionCiudad }}</p>
            </div>
            <div class="precio-oferta">
                <div class="precio">
                    <p>Precio: {{ destino.precioTour }}</p>
                </div>
                <div>
                    {% if destino.ofertaTour %}
                        <p class="oferta">En oferta!</p>
                    {% endif %}
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </div>
        <div class="comprar-btn">
            <a href="http://127.0.0.1:8000/cuentas/login"
               class="comprar-btn">COMPRAR</a>
        </div>
    </div>
</div>
<% endblock %}

```

**Listing 19: LOGIN.HTML**

Se carga el directorio estático utilizando % load static % y se asigna la ruta base a la variable baseUrl utilizando % static 'images' as baseUrl %.

Dentro del `jhead`, se definen metadatos como la codificación de caracteres, la compatibilidad con Internet Explorer, el título de la página y la configuración de la vista inicial.

Se enlaza la hoja de estilo "style.css" utilizando % static 'style.css' %.

En el `jheader`, se muestra un logotipo de la agencia de viajes utilizando la URL proporcionada. A continuación, se presenta una barra de navegación con enlaces a la página de inicio y a la página de registro.

Dentro del `jbody`, se muestra un encabezado con el título "INGRESAR" centrado en la página.

A continuación, se muestra un formulario de inicio de sesión. El formulario tiene un campo para ingresar el nombre de usuario y otro campo para ingresar la contraseña. Al enviar el formulario, los datos se envían a la ruta "login" utilizando el método POST. Se incluye % csrf\_token % para protección contra falsificación de solicitudes entre sitios.

Después del formulario, se muestra una sección para mostrar mensajes. Si hay mensajes disponibles, se iteran utilizando % for message in messages % y se muestran en un elemento `jh3`.

```

{% load static %}
{% static "images" as baseUrl %}

<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <meta http-equiv='X-UA-Compatible' content='IE=edge'>
    <title>Ingresar...</title>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <link rel="stylesheet" href="{% static 'style.css' %}">
</head>

<header>
    <div class="logo">
        
    </div>
    <nav>
        <ul >
            <li><a href="http://127.0.0.1:8000/" class="sub">Inicio</a></li>
            <li><a href="http://127.0.0.1:8000/cuentas/register"
               class="sub">Registrarse</a></li>
        </ul >
    </nav>
</header>

```

```

<li><a href="#" class="carrito" data-counter="0">#128722;</a></li>
</ul>
</nav>
</header>

<body>

<h1 class="titLoginRegister" align="center">INGRESAR</h1>

<form action="login" method="post">
    {% csrf_token %}
    <input type="text" name="username" placeholder="Usuario"><br>
    <input type="password" name="password" placeholder="Contraseña"><br>
    <input type="Submit">
</form>

<div>
    {% for message in messages %}
        <h3> {{message}} </h3>
    {% endfor %}
</div>
</body>
</html>

```

---

**Listing 20: REGISTER.HTML**

---

Se carga el directorio estático utilizando % load static % y se asigna la ruta base a la variable baseUrl utilizando % static "images" as baseUrl %.

Dentro del `jhead`, se definen metadatos como la codificación de caracteres, la compatibilidad con Internet Explorer, el título de la página y la configuración de la vista inicial.

Se enlaza la hoja de estilo "style.css" utilizando % static 'style.css' %.

En el `jheader`, se muestra un logotipo de la agencia de viajes utilizando la URL proporcionada. A continuación, se presenta una barra de navegación con enlaces a la página de inicio y a la página de inicio de sesión.

Dentro del `jbody`, se muestra un encabezado con el título REGISTRAR centrado en la página.

A continuación, se muestra un formulario de registro de usuarios. El formulario tiene campos para ingresar el nombre, apellidos, nombre de usuario, correo electrónico, contraseña y confirmación de contraseña. Al enviar el formulario, los datos se envían a la ruta register utilizando el método POST. Se incluye % csrf\_token % para protección contra falsificación de solicitudes entre sitios.

Después del formulario, se muestra una sección para mostrar mensajes. Si hay mensajes disponibles, se iteran utilizando % for message in messages % y se muestran en un elemento `jh3`.

```

{% load static %}
{% static "images" as baseUrl %}

<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <meta http-equiv='X-UA-Compatible' content='IE=edge'>
    <title>Registrarme..</title>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <link rel="stylesheet" href="{% static 'style.css' %}">
</head>

```

```

<header>
    <div class="logo" >
        
    </div>
    <nav>
        <ul >
            <li><a href="http://127.0.0.1:8000/" class="sub">Inicio</a></li>
            <li><a href="http://127.0.0.1:8000/cuentas/login" class="sub">Ingresar</a></li>
            <li><a href="#" class="carrito" data-counter="0">#128722;</a></li>
        </ul>
    </nav>
</header>

<body>

    <h1 class="titLoginRegister" align="center">REGISTRAR</h1>

    <form action="register" method="post">
        {% csrf_token %}

        <input type="text" name="first_name" placeholder="Nombres"><br>
        <input type="text" name="last_name" placeholder="Apellidos"><br>
        <input type="text" name="username" placeholder="Usuario"><br>
        <input type="email" name="email" placeholder="Correo"><br>
        <input type="password" name="password1" placeholder="Contraseña"><br>
        <input type="password" name="password2" placeholder="Confirmar Contraseña"><br>
        <input type="Submit">

    </form>

    <div>
        {% for message in messages %}
            <h3> {{message}} </h3>
        {% endfor %}
    </div>

</body>
</html>

```

**Listing 21: CUENTAS/VIEWS.PY - CREACION DE LAS VISTAS**

**Manejo del inicio de sesión:** En esta parte del código, se define la función login(request). Si la solicitud es de tipo POST, se obtienen el nombre de usuario y la contraseña del formulario utilizando request.POST. Luego, se utiliza auth.authenticate() para verificar si las credenciales son válidas. Si el usuario es autenticado correctamente, se inicia sesión utilizando auth.login() y se redirige al usuario a la página de inicio. Si las credenciales no son válidas, se muestra un mensaje de error utilizando messages.info() y se redirige al usuario de vuelta a la página de inicio de sesión. Si la solicitud no es de tipo POST, se renderiza la plantilla 'login.html' para mostrar el formulario de inicio de sesión.

```

from django.shortcuts import render, redirect
from django.contrib import messages
from django.contrib.auth.models import User, auth

```

```
# Create your views here.

def login(request):
    if request.method== 'POST':
        username = request.POST['username']
        password = request.POST['password']

        user = auth.authenticate(username=username,password=password)

        if user is not None:
            auth.login(request, user)
            return redirect("/")
        else:
            messages.info(request,'invalid credentials')
            return redirect('login')

    else:
        return render(request,'login.html')
```

**Listing 22: CUENTAS/VIEWS.PY - CREACION DE LAS VISTAS**

**Manejo del registro de usuarios:** En esta parte del código, se define la función register(request). Si la solicitud es de tipo POST, se obtienen los datos del formulario, como el nombre, apellidos, nombre de usuario, contraseñas y correo electrónico. Luego, se realizan algunas verificaciones, como comprobar si el nombre de usuario o el correo electrónico ya existen en la base de datos utilizando User.objects.filter(). Si alguno de los datos ya existe, se muestra un mensaje de error correspondiente y se redirige al usuario de vuelta a la página de registro. Si los datos son válidos, se crea un nuevo usuario utilizando User.objects.create\_user() y se redirige al usuario a la página de inicio de sesión. Si las contraseñas no coinciden, se muestra un mensaje de error y se redirige al usuario de vuelta a la página de registro. Si la solicitud no es de tipo POST, se renderiza la plantilla 'register.html' para mostrar el formulario de registro.

```
def register(request):

    if request.method == 'POST':
        first_name = request.POST['first_name']
        last_name = request.POST['last_name']
        username = request.POST['username']
        password1 = request.POST['password1']
        password2 = request.POST['password2']
        email = request.POST['email']

        if password1==password2:
            if User.objects.filter(username=username).exists():
                messages.info(request,'Username Taken')
                return redirect('register')
            elif User.objects.filter(email=email).exists():
                messages.info(request,'Email Taken')
                return redirect('register')
            else:
                user = User.objects.create_user(username=username, password=password1,
                                                email=email,first_name=first_name,last_name=last_name)
```

```
        user.save();
        print('user created')
        return redirect('login')

    else:
        messages.info(request,'password not matching..')
        return redirect('register')
    return redirect('/')

else:
    return render(request,'register.html')
```

---

#### Listing 23: CUENTAS/VIEWS.PY - CREACION DE LAS VISTAS

---

Manejo del cierre de sesión: En esta parte del código, se define la función logout(request). Al llamar a auth.logout(), se cierra la sesión del usuario y se redirige a la página de inicio.

```
def logout(request):
    auth.logout(request)
    return redirect('/')
```

---

#### Listing 24: CREAMOS LAS URLs DE NUESTRA APPS CUENTAS

---

El código proporcionado es una configuración de las URLs en Django. Se utiliza el módulo django.urls para importar la función path, y también se importa el módulo views que contiene las funciones definidas en el código anterior. Las URLs se definen en la variable urlpatterns, que es una lista de rutas de URL. Este código define las rutas de URL que corresponden a las funciones register, login y logout en el archivo views.py. Cada URL está asociada a una vista específica y se les asigna un nombre para su referencia en el código. Estas URLs permiten a los usuarios acceder a las páginas de registro, inicio de sesión y cierre de sesión en la aplicación web.

```
from django.urls import path
from . import views

urlpatterns = [
    path("register", views.register, name="register"),
    path("login",views.login, name="login"),
    path("logout",views.logout,name="logout"),
]
```

## 4. Mi Sitio de Lugares Turísticos:

- Demostracion de **LUGARES TURISTICOS DEL PERÚ**
- Ejecutamos en nuestra consola lo siguiente 'python manage.py runserver' para correr el servidor web. En el navegador escribimos la dirección <http://127.0.0.1:8000/>.
- **El destinos\_turisticos.html o pagina principal**

**LUGARES TURISTICOS**

The screenshot shows a collage of images from various Peruvian landmarks: Machu Picchu, a llama, a person playing a flute, a lake, and a person in traditional dress. To the right, there is descriptive text about the beauty of Peru, mentioning Machu Picchu, the coast, the Andes, and the Amazon.

Descubre las maravillas de Perú, un país repleto de bellezas naturales. Desde la imponente ciudadela de Machu Picchu, una de las nuevas maravillas del mundo, hasta la enigmática línea de Nazca, cada rincón te sorprenderá. En la costa peruana, disfruta de playas de arenas doradas y aguas cristalinas como Máncora y Punta Sal, ideales para relajarse y practicar deportes acuáticos. Explora Lima, con su gastronomía exquisita y miradores con vistas impresionantes al Océano Pacífico. En la sierra andina, explora la antigua capital inca de Cusco y el Valle Sagrado, con impresionantes ruinas y pintorescos pueblos. No te pierdas la joya arqueológica rodeada de montañas y misterio: Machu Picchu, que te transportará en el tiempo. En la selva amazónica, la biodiversidad te espera. Aventúrate en un recorrido por el río Amazonas, descubre la flora y fauna en reservas naturales como Tambopata o Iquitos, y vive experiencias únicas en contacto con la naturaleza, como avistar delfines rosados y explorar senderos selváticos. En el norte del país, explora las ruinas de Chan Chan y las pirámides de Túcume para adentrarte en la historia precolombina. Disfruta de hermosas playas como Huanchaco, famosa por sus caballitos de totora, y descubre Trujillo, con su arquitectura colonial y casas coloridas. Perú te espera con los brazos abiertos para que descubras su cultura, historia y naturaleza impresionante. ¡No pierdas la oportunidad de visitar estos lugares turísticos y vivir experiencias inolvidables!

**VIAJES NACIONALES**

**VIAJES NACIONALES**

<b>PUNTA NEGRA</b>  Precio: 50.00 <b>¡En oferta!</b> <a href="#">VER</a>	<b>AREQUIPA</b>  Precio: 60.00 <a href="#">VER</a>	<b>MACHU PICCHU</b>  Precio: 150.00 <b>¡En oferta!</b> <a href="#">VER</a>
<b>CAÑON DEL COLCA</b> <td><b>LINEAS DE NAZCA</b> </td> <td><b>LAGO TITICACA</b> </td>	<b>LINEAS DE NAZCA</b> 	<b>LAGO TITICACA</b> 

VISITAR LUGARES

TOP 31 que ver en Perú imprescindibles ★  
 Vanessa Huett (Voyage Perou)  
 Este mapa se ha hecho con Google My Maps. Crea tu mapa

Cuernavaca Ilo Arequipa La Paz Cochabamba Santa Cruz de la Sierra

Datos de mapas ©2023 Google, INEGI Términos 200 km Combinaciones de fuentes

© 2023 Lugares Turísticos. Todos los derechos reservados. Hecho por Ower Lopez

- El detalle\_destino.html, esta pagina sirve para visualizar los lugares turisticos al hacer clic en VER

127.0.0.1:8000/destino/4/

**CAÑON DEL COLCA**  
 El Cañón del Colca es el lugar perfecto para ver a una de las aves más representativas de la biodiversidad en nuestro país, el cóndor andino. Puedes ver su vuelo desde el mirador de la Cruz del Cóndor y maravillarte con el tamaño de estas aves así como las imponentes vistas de las montañas alrededor.

Precio: 70,00      ¡En oferta!      COMPRAR

© 2023 Lugares Turísticos. Todos los derechos reservados. Hecho por Ower Lopez

127.0.0.1:8000/destino/3/

**MACHU PICCHU**

El Santuario Histórico de Machu Picchu se ubica en la provincia de Urubamba, Región Cusco. Comprende un extenso paisaje cultural y natural localizado en diversos ecosistemas, en el que también existen otros sitios arqueológicos conectados por caminos que conducen a la ciudad inca. Machu Picchu es el sitio arqueológico inca más sobresaliente debido a su creativo diseño urbano, la belleza de su arquitectura y el fino trabajo en piedra de sus construcciones. En su planificación se aprovechó notablemente la topografía de la cima de la montaña a la que transformó en una imponente llanura. Fue edificada en el siglo XV por el Inca Pachacutec, abandonada en los primeros años de colonia española y redescubierta en el año 1911. La arquitectura de esta ciudad inka, plasmada en sus 200 recintos, se mimetiza excepcionalmente con el maravilloso entorno natural al que se encuentra estrechamente ligado.

Precio: 150,00      ¡En oferta!      COMPRAR

© 2023 Lugares Turísticos. Todos los derechos reservados. Hecho por Ower Lopez

127.0.0.1:8000/destino/6/

**LAGO TITICACA**

De azul intenso y brillante, el Titicaca es el lago navegable más alto del mundo y un lugar sagrado para la extinta civilización Inca. Según cuenta la leyenda, fue en este lugar donde se creó el mundo, cuando el dios Viracocha salió del lago y creó el sol, las estrellas y las primeras personas.

Precio: 125,00      ¡En oferta!      COMPRAR

© 2023 Lugares Turísticos. Todos los derechos reservados. Hecho por Ower Lopez

- En el login.html, se aprecia el inicio de cuenta de un usuario

The screenshot shows a web browser window with the URL 127.0.0.1:8000/cuentas/login. The page has a dark header with the university's logo and navigation links for 'Inicio' and 'Registrarse'. The main content area is titled 'INGRESAR' and contains two input fields: 'Usuario' and 'Contraseña', followed by a 'Enviar' button.

- En el register.html, se puede crear una nueva cuenta para un nuevo usuario

The screenshot shows a web browser window with the URL 127.0.0.1:8000/cuentas/register. The page has a dark header with the university's logo and navigation links for 'Inicio' and 'Ingresar'. The main content area is titled 'REGISTRAR' and contains six input fields: 'Nombres', 'Apellidos', 'Usuario', 'Correo', 'Contraseña', and 'Confirmar Contraseña', followed by a 'Enviar' button.

- Ahora veremos en la BD /admin, los lugares turísticos creados o para subir.

Administración de Django

Inicio > Destinos > Destinos turísticos

Empiece a escribir para filtrar...

**AUTENTICACIÓN Y AUTORIZACIÓN**

- Grupos [+ Añadir](#)
- Usuarios [+ Añadir](#)

**DESTINOS**

- Destinos turísticos [+ Añadir](#)

«

Selección de destinos turísticos a modificar

Acción: ----- Ir seleccionados 0 de 6

DESTINOS TURÍSTICOS  
 LAGO TITICACA  
 LINEAS DE NAZCA  
 CAÑÓN DEL COLCA  
 MACHU PICCHU  
 AREQUIPA  
 PUNTA NEGRA

6 destinos turísticos

ANADIR DESTINOS TURÍSTICOS +

- Aquí se aprecia nuestro models.py de nuestra base de datos

Administración de Django

Inicio > Destinos > Destinos turísticos > Añadir destinos turísticos

Empiece a escribir para filtrar...

**AUTENTICACIÓN Y AUTORIZACIÓN**

- Grupos [+ Añadir](#)
- Usuarios [+ Añadir](#)

**DESTINOS**

- Destinos turísticos [+ Añadir](#)

«

Añadir destinos turísticos

NombreCiudad:

DescripciónCiudad:

ImagenCiudad:  Ninguno archivo selec.

PrecioTour:

OfertaTour

**GUARDAR** **Guardar y añadir otro** **Guardar y continuar editando**

- Y aqui se aprecia como se rellena el espacio del models.py

The screenshot shows the Django Admin interface for modifying a tourism destination. The left sidebar shows navigation links for 'AUTENTICACIÓN Y AUTORIZACIÓN' (Groups, Users), 'DESTINOS' (Destinations), and 'Destinos turísticos'. The main content area is titled 'Modificar destinos turísticos' and shows a destination named 'LAGO TITICACA'. The 'NombreCiudad' field contains 'LAGO TITICACA'. The 'DescripciónCiudad' field contains a detailed description of Lake Titicaca. The 'ImagenCiudad' field shows a thumbnail of a lake image with the note 'Actualmente: imágenes/LAGO\_TITICACA.jpg'. The 'PrecioTour' field is set to '125.00'. A checkbox for 'OfertaTour' is checked. At the bottom are buttons for 'GUARDAR', 'Guardar y añadir otro', 'Guardar y continuar editando', and 'Eliminar'.

- Ademas, en la tabla de USUARIOS se puede visualizar los usuarios que crearon su cuenta

The screenshot shows the Django Admin interface for managing users. The left sidebar shows navigation links for 'AUTENTICACIÓN Y AUTORIZACIÓN' (Groups, Users), 'DESTINOS' (Destinations), and 'Destinos turísticos'. The main content area is titled 'Selección de usuario a modificar' and lists three users: 'GabrielaR', 'Luis1', and 'OwerFrank'. Each user has a checkbox next to their name, and columns for 'NOMBRE DE USUARIO', 'DIRECCIÓN DE CORREO ELECTRÓNICO', 'NOMBRE', 'APELLIDOS', and 'ES STAFF'. On the right, there is a 'FILTRO' sidebar with dropdown menus for 'Por es staff' (Todo, Sí, No), 'Por estado de superusuario' (Todo, Sí, No), and 'Por activo' (Todo, Sí, No).

## 5. Referencias

- [https://www.w3schools.com/python/python\\_reference.asp](https://www.w3schools.com/python/python_reference.asp)
- <https://docs.python.org/3/tutorial/>
- <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Models>
- [https://tutorial.djangogirls.org/es/django\\_models/](https://tutorial.djangogirls.org/es/django_models/)
- <https://pear.php.net/manual/en/standards.php>
- <https://docs.djangoproject.com/en/4.0/>
- <https://www.youtube.com/watch?v=M4NIIs4BM1dk>
- <https://pypi.org/>
- [https://pip.pypa.io/en/latest/user\\_guide/](https://pip.pypa.io/en/latest/user_guide/)
- <https://packaging.python.org/en/latest/tutorials/installing-packages/>