

## Informe de Laboratorio 07

### Tema: Django plantillas Destinos turísticos

Nota

Estudiante	Escuela	Asignatura
Victor Gonzalo Maldonado Vilca vmaldonadov@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: III Código: 1702122

Tarea	Tema	Duración
07	Django plantillas Destinos turísticos	2 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 9 de abril de 2024	Al 5 de junio de 2024

## 1. Introducción

Django es un framework de desarrollo web en Python que facilita la creación rápida de aplicaciones web seguras y escalables. Ofrece un conjunto de herramientas integradas como ORM, administrador de Django y enrutamiento de URLs, lo que simplifica el desarrollo y mejora la seguridad de las aplicaciones web.

## 2. Objetivos

- Implementar una aplicación en Django utilizando una plantilla profesional.
- Utilizar una tabla de Destinos turísticos para leer y completar la página web.
- Utilizar los tags “if” y “for” en los archivos html para leer todos los registros de una tabla desde una base de datos.

## 3. Tarea

### Ejercicios Propuestos

1. Deberán replicar la actividad del video que se encuentra en el AV de Teoría (Django Tutorial for Beginners - Telusko <https://youtu.be/OTmQOjsl0eg>) donde se obtiene una plantilla de una aplicación de Destinos turísticos y adecuarla a un proyecto en blanco Django.

2. Luego trabajar con un modelo de tabla DestinosTuristicos donde se guarden nombreCiudad, descripciónCiudad, imagenCiudad, precioTour, ofertaTour (booleano). Estos destinos turísticos deberán ser agregados en una vista dinámica utilizando tags for e if.
3. Para ello crear una carpeta dentro del proyecto github colaborativo con el docente, e informar el link donde se encuentra.
4. Crear formularios de Añadir Destinos Turísticos, Modificar, Listar y Eliminar Destinos.
5. Eres libre de agregar CSS para decorar tu trabajo.
6. Ya sabes que el trabajo con Git es obligatorio. Revisa el avance de la teoría Django parte4

#### **4. Entregables**

- Informe hecho en Latex
- URL: Repositorio GitHub
- Link de Vídeo Explicativo (Youtube o flipgrid)

#### **5. Equipos, materiales y temas utilizados**

- Python
- Pip
- Django
- Entornos virtuales en python
- Proyectos de Django
- Aplicaciones en Django
- Modelos, Vistas, Templates y Formularios en Django
- Plantillas profesionales
- Tags para vistas dinámicas

#### **6. URL de Repositorio Github**

- URL del Repositorio GitHub
- [https://github.com/Victor-Gonzalo-Maldonado-Vilca/Lab07\\_Django.git](https://github.com/Victor-Gonzalo-Maldonado-Vilca/Lab07_Django.git)

#### **7. Link de Video**

- Link del Vídeo Explicativo
- **Youtube:** <https://www.youtube.com/watch?v=pFaY1LSbvb8>
- **flipgrid:** <https://flip.com/s/jUuZm7UfskSd>

## 8. Metodología

### 8.1. Creación del entorno de Trabajo

#### 8.1.1. Carpeta de trabajo

Listing 1: Creación del Directorio

```
mkdir lab07 && cd lab07
```

#### 8.1.2. Creación del Entorno Virtual

Listing 2: Creación Entorno virtual

```
virtualenv -p python3 .
```

#### 8.1.3. Activar entorno Virtual

Listing 3: Activar Entorno Virtual

```
Scripts/activate
```

#### 8.1.4. Instalar Django en el entorno Virtual

Listing 4: Instalar Django

```
pip install Django
```

#### 8.1.5. Creación carpeta del proyecto

Listing 5: Carpeta src

```
mkdir src && cd src
```

#### 8.1.6. Inicializar git

Listing 6: Inicializar git

```
git init
```

#### 8.1.7. Crear .gitignore

Seguimos el siguiente Repositorio <https://github.com/django/django/blob/main/.gitignore>

## 8.2. Creacion del proyecto y apps

### 8.2.1. Crear Proyecto

Listing 7: Crear proyecto

```
django-admin startproject telusko
```

### 8.2.2. Crear App

Listing 8: Crear App

```
python manage.py startapp travello
```

## 9. Desarrollo del trabajo

### 9.1. Seguimiento de la Actividad del Vídeo

*Todo las app prueba se encuentran en el repositorio GitHub*

- Generación de proyectos Pruebas.
- El video nos otorga las plantillas, imágenes, js , css, etc para empezar con el proyecto.
- Link donde se encuentran los recursos: <https://www.dropbox.com/sh/gf8x8xjwidq20cz/AABP1OsXmijnz0-yjxYj9edl=0>

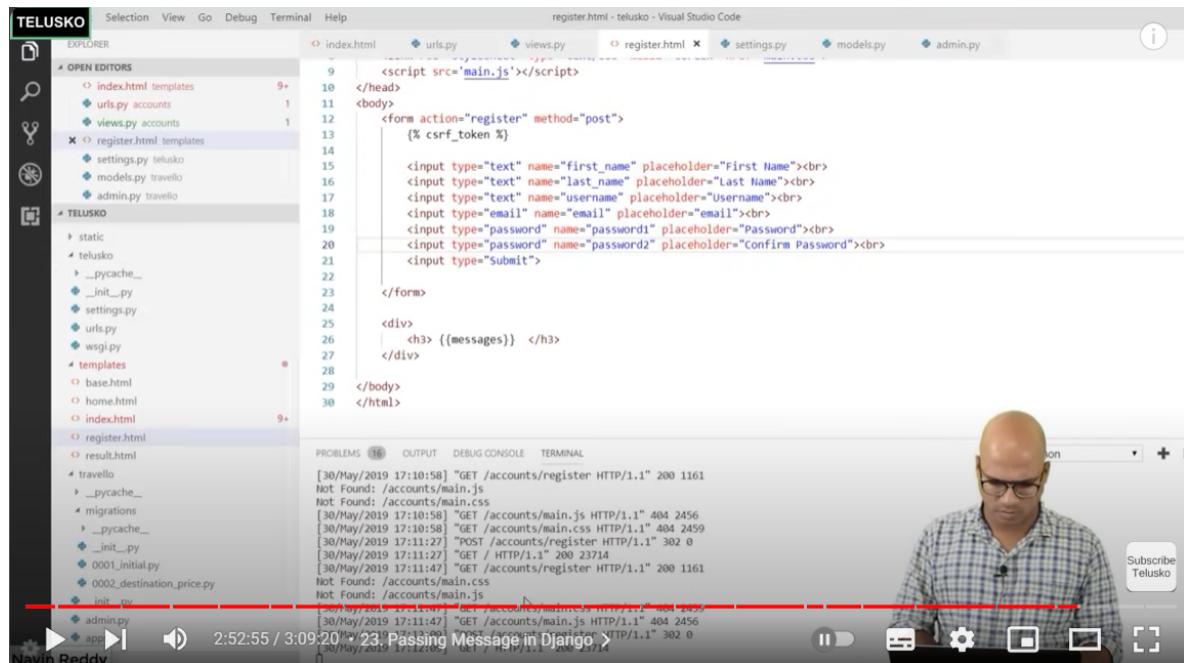


Figura 1: Imagen del vídeo

## 9.2. Configuración en settings.py

### 9.2.1. Configuración del Lenguaje y Time-Zone

```

108 | LANGUAGE_CODE = 'es'
109 |
110 | TIME_ZONE = 'America/Lima'
111 |
112 USE_I18N = True
113 |
114 USE_TZ = True
115

```

Figura 2: Internationalization

### 9.2.2. Configuración de archivos estáticos

```

120 STATIC_URL = '/static/'
121 STATICFILES_DIRS = [
122     os.path.join(BASE_DIR, 'static')
123 ]
124
125 STATIC_ROOT = os.path.join(BASE_DIR, 'assets')
126

```

Figura 3: Static Files

### 9.2.3. Configuración de archivos media

```

132 MEDIA_URL = '/media/'
133 MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
134

```

Figura 4: Media Files

## 9.3. Modelos

### 9.3.1. Modelo tabla DiseñoTuristico

- Descripción:** El modelo Destination representa destinos turísticos e incluye campos como el nombre de la ciudad (nombreCiudad), una descripción detallada de la ciudad (descripcionCiudad), una imagen representativa de la ciudad (imagenCiudad), el precio del tour (precioTour), y un indicador de si hay una oferta de tour disponible (ofertaTour).

Listing 9: Modelo Destination

```

class Destination(models.Model):
    nombreCiudad = models.CharField(max_length=100)
    descripcionCiudad = models.TextField()
    imagenCiudad = models.ImageField(upload_to='images/', default='default_image.jpg')
    precioTour = models.DecimalField(max_digits=10, decimal_places=2)

```

```

ofertaTour = models.BooleanField(default=False)

def __str__(self):
    return self.nombreCiudad

```

### 9.3.2. Modelo Para Eliminar Lugares Turísticos

- **Descripción:** El modelo Delete esta relacionado con la eliminación de ciudades, ya que solo tiene un campo para el nombre de la ciudad (nombreCiudad).

Listing 10: Modelo Delete

```

class Delete(models.Model):
    nombreCiudad = models.CharField(max_length=100)

    def __str__(self):
        return self.nombreCiudad

```

### 9.3.3. Modelo Para Modificar Lugares Turísticos

- **Descripción:** El modelo Modificar tiene un campo para el nombre de la ciudad (nombreCiudad), despues se enviara como argumento este campo a otra página web donde se podra realizar cambios al Lugar Turístico seleccionado.

Listing 11: Modelo Modificar

```

class Modificar(models.Model):
    nombreCiudad = models.CharField(max_length=100)

    def __str__(self):
        return self.nombreCiudad

```

## 9.4. Creación de los Formularios Añadir, Modificar, Listar y Eliminar

### 9.4.1. Formulario DestinationForm - Análogo de Añadir

- **Descripción:** El formulario DestinationForm está asociado con el modelo Destination y contiene campos para ingresar información como el nombre de la ciudad, su descripción, una imagen representativa, el precio del tour y la disponibilidad de ofertas.

Listing 12: Formulario Añadir

```

class DestinationForm(forms.ModelForm):
    class Meta:
        model = Destination
        fields = ['nombreCiudad', 'descripcionCiudad', 'imagenCiudad', 'precioTour',
                  'ofertaTour']

```

#### 9.4.2. Formulario ModificarForm - Análogo de Modificar

- **Descripción:** El formulario ModificarForm corresponde al modelo Modificar y tiene un campo para ingresar el nombre de la ciudad que se desea modificar.

Listing 13: Formulario Modificar

```
class ModificarForm(forms.ModelForm):  
    class Meta:  
        model = Modificar  
        fields = ['nombreCiudad']
```

#### 9.4.3. Formulario ModificarFormu - Análogo de Modificar

- **Descripción:** El formulario ModificarFormu es una variante del DestinationForm pero con otro propósito, este logrará que los datos se actualicen.

Listing 14: Formulario Modificar

```
class ModificarFormu(forms.ModelForm):  
    class Meta:  
        model = Destination  
        fields = ['nombreCiudad', 'descripcionCiudad', 'imagenCiudad', 'precioTour',  
                  'ofertaTour']
```

#### 9.4.4. Formulario DeleteForm - Análogo de Eliminar

- **Descripción:** El formulario DeleteForm está vinculado al modelo Delete y proporciona un campo para ingresar el nombre de la ciudad que se desea eliminar.

Listing 15: Formulario Eliminar

```
class DeleteForm(forms.ModelForm):  
    class Meta:  
        model = Delete  
        fields = ['nombreCiudad']
```

#### 9.4.5. Formulario Listar

- **Descripción:** No se requiere formularios para esta sección, solo tags for e if.

### 9.5. Vistas

#### 9.5.1. Vista index

- **Descripción:** La vista index recupera todos los destinos de la base de datos utilizando el modelo Destination y los pasa al template index.html como parte del contexto para renderizar la página principal.

Listing 16: index

```
def index(request):  
    destinos = Destination.objects.all()  
    context = {'destinos': destinos}  
    return render(request, 'index.html', context)
```

### 9.5.2. Vistas about, contact y news

- **Descripción:** Las vistas about, contact y news simplemente renderizan las páginas correspondientes (about.html, contact.html y news.html) sin realizar ninguna operación adicional.

Listing 17: about

```
def about(request):
    return render(request, 'about.html');
```

Listing 18: contact

```
def contact(request):
    return render(request, 'contact.html');
```

Listing 19: news

```
def news(request):
    return render(request, 'news.html');
```

### 9.5.3. Vista admin

- **Descripción:** La vista admin maneja la lógica para la administración de destinos. Crea instancias de formularios para crear, eliminar y modificar destinos, recupera todos los destinos de la base de datos y procesa las solicitudes POST según la acción realizada por el usuario (crear, eliminar o modificar).

Listing 20: admin

```
def admin(request):
    crear_form = DestinationForm()
    eliminar_form = DeleteForm()
    modificar_form = ModificarForm()
    destinos = Destination.objects.all()

    if request.method == 'POST':
        if 'crear' in request.POST:
            crear_form = DestinationForm(request.POST, request.FILES)
            if crear_form.is_valid():
                crear_form.save()
                return redirect('admin')
        elif 'eliminar' in request.POST:
            eliminar_form = DeleteForm(request.POST)
            if eliminar_form.is_valid():
                nombre_ciudad = eliminar_form.cleaned_data['nombreCiudad']
                destino = get_object_or_404(Destination, nombreCiudad=nombre_ciudad)
                destino.delete()
                return redirect('admin')
        elif 'modificar' in request.POST:
            nombre_ciudad = request.POST.get('nombreCiudad')
            try:
                lugar = Destination.objects.get(nombreCiudad=nombre_ciudad)
                return redirect('modificar', ciudad=nombre_ciudad)
            except Destination.DoesNotExist:
                return redirect('admin')
```

```

context = {
    'crear_form': crear_form,
    'eliminar_form': eliminar_form,
    'modificar_form': modificar_form,
    'destinos': destinos,
}
return render(request, 'admin.html', context)

```

#### 9.5.4. Vista modificar

- **Descripción:** La vista modificar maneja la lógica para modificar un destino específico. Recupera el destino a modificar utilizando su nombre de ciudad, crea un formulario prellenado con los datos existentes del destino y guarda los cambios realizados por el usuario si el formulario es válido.

Listing 21: modificar

```

def modificar(request, ciudad):
    lugar = get_object_or_404(Destination, nombreCiudad=ciudad)
    form = ModificarFormu(instance=lugar)

    if request.method == 'POST':
        form = ModificarFormu(request.POST, instance=lugar)
        if form.is_valid():
            form.save()
            return redirect('admin')

    context = {
        'form': form,
    }
    return render(request, 'modificar.html', context)

```

## 9.6. URLs

### 9.6.1. urls.py del proyecto telusko

- **Descripción:** Este bloque de código configura las URL para el proyecto Django telusko, asignando las URL a vistas específicas y proporcionando la capacidad de servir archivos estáticos y de medios en modo DEBUG.

Listing 22: urls Telusko

```

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

if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

```

### 9.6.2. urls.py de la app travello

- **Descripción:** Este bloque de código definen las rutas accesibles en el sitio web y las asocian con las vistas correspondientes para manejar las solicitudes de los usuarios.

Listing 23: urls Travello

```
urlpatterns = [
    path('', views.index, name='index'),
    path('about/', views.about, name='about'),
    path('contact/', views.contact, name='contact'),
    path('news/', views.news, name='news'),
    path('admin/', views.admin, name='admin'),
    path('/modificar/<str:ciudad>/', views.modificar, name='modificar'),
]
```

## 9.7. Plantillas

*Las plantillas que se omitirán serán las presentadas en el video que se debió seguir, se centrar en la plantilla admin y modificar. Recordemos que las plantillas deben estar en la carpeta templates*

### 9.7.1. Plantilla admin.html

- **Descripción:** Esta línea carga los archivos estáticos (como CSS, JavaScript) para que puedan ser utilizados en la página.

```
{% load static %}
```

- **Descripción:** Define la estructura básica del documento HTML, incluyendo el título de la página y el enlace al archivo CSS para estilos.

```
<head>
    <title>Administrar Lugares</title>
    <link rel="stylesheet" type="text/css" href="{% static 'styles/styles.css' %}">
</head>
```

- **Descripción:** Formulario para agregar un nuevo lugar, con un campo para cada atributo del modelo Destination y un botón para guardar los datos.

```
<h1>Agregar Lugar</h1>
<form method="post" enctype="multipart/form-data">
    {% csrf_token %}
    {{ crear_form.as_p }}
    <button type="submit" name="crear">Guardar</button>
</form>
```

- **Descripción:** Formulario para eliminar un lugar existente, con un campo para seleccionar el nombre de la ciudad y un botón para eliminar el lugar.

```
<h1>Eliminar Lugar</h1>
<form method="post">
    {% csrf_token %}
    {{ eliminar_form.as_p }}
    <button type="submit" name="eliminar">Guardar</button>
</form>
```

- **Descripción:** Formulario para modificar un lugar existente, con un campo para seleccionar el nombre de la ciudad y un botón para redireccionar a modificar.html pasando como argumento la ciudad.

```
<h1>Modificar Lugar</h1>
<form method="post">
    {% csrf_token %}
    {{ modificar_form.as_p }}
    <button type="submit" name="modificar">Guardar</button>
</form>
```

- **Descripción:** Sección para mostrar los lugares existentes, incluyendo imágenes, nombres, descripciones, precios y ofertas.

```
{% if destinos %}
    <div class="row">
        <div class="col">
            {% for destino in destinos %}
                <div class="destination_item">
                    <div class="destination_image">
                        
                    </div>
                    <div class="destination_content">
                        <div class="destination_title">{{ destino.nombreCiudad }}</div>
                        <div class="destination_subtitle"><p>{{ destino.descripcionCiudad }}</p></div>
                        <div class="destination_price">$ {{ destino.precioTour }} - Oferta: {{ destino.ofertaTour }}</div>
                    </div>
                </div>
            {% endfor %}
        </div>
    {% endif %}
```

- **Descripción:** Enlace para regresar a la página principal (index.html) del sitio web.

```
<div class="main-link">
    <a href="{% url 'index' %}">Pagina Principal</a>
</div>
```

### 9.7.2. Plantilla modificar.html

- **Descripción:** Esta línea carga los archivos estáticos (como CSS, JavaScript) para que puedan ser utilizados en la página.

```
{% load static %}
```

- **Descripción:** Define la estructura básica del documento HTML, incluyendo el título de la página y el enlace al archivo CSS para estilos.

```
<head>
    <title>Modificar Lugar</title>
    <link rel="stylesheet" type="text/css" href="{% static 'styles/styles1.css' %}">
</head>
```

- **Descripción:** Formulario para modificar un lugar existente. Incluye un campo para cada atributo del modelo asociado al formulario y un botón para guardar los cambios.

```
<h1>Modificar Lugar</h1>
<form method="post" enctype="multipart/form-data">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit" name="crear">Guardar</button>
</form>
```

## 9.8. Estilos

*Los estilos fueron entregados por el video seguido, además los estilos de las anteriores plantillas solo son cambio de color, estos se encuentran en el repositorio GitHub*

### 9.8.1. Styles de admin.html

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f4f4f4;
}

h1 {
    color: #333;
    text-align: center;
    padding: 20px;
}

form {
    width: 50%;
    margin: 0 auto;
    padding: 20px;
    background-color: #fff;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
```

```
form button {  
    display: block;  
    width: 100%;  
    padding: 10px;  
    background-color: #5cb85c;  
    color: #fff;  
    border: none;  
    border-radius: 5px;  
    cursor: pointer;  
    font-size: 16px;  
}  
  
form button:hover {  
    background-color: #4cae4c;  
}  
  
.destinations {  
    padding: 20px;  
}  
  
.destination {  
    background-color: #fff;  
    border-radius: 8px;  
    margin-bottom: 20px;  
    overflow: hidden;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
    text-align: center;  
}  
  
.destination_image img {  
    width: 25%;  
    height: auto;  
    display: block;  
    margin-left: auto;  
    margin-right: auto;  
}  
  
.destination_content {  
    padding: 20px;  
    text-align: center;  
}  
  
.destination_title {  
    font-size: 20px;  
    font-weight: bold;  
    color: #333;  
}  
  
.destination_subtitle {  
    font-size: 14px;  
    color: #777;  
    margin-top: 10px;  
}  
  
.destination_price {
```

```
font-size: 18px;
color: #e67e22;
margin-top: 10px;
}

.container {
max-width: 1200px;
margin: 0 auto;
}

.section_title {
font-size: 28px;
margin-top: 20px;
}

.main-link {
text-align: center;
margin: 20px;
}

.main-link a {
color: #337ab7;
font-size: 18px;
text-decoration: none;
padding: 10px 20px;
border: 1px solid #337ab7;
border-radius: 5px;
transition: background-color 0.3s, color 0.3s;
}

.main-link a:hover {
background-color: #337ab7;
color: #fff;
}
```

#### 9.8.2. Styles de modificar.html

```
body {
font-family: Arial, sans-serif;
margin: 0;
padding: 0;
background-color: #f4f4f4;
}

h1 {
color: #333;
text-align: center;
padding: 20px;
}

form {
width: 50%;
margin: 0 auto;
padding: 20px;
background-color: #fff;
```

```
border-radius: 8px;  
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}  
  
form button {  
    display: block;  
    width: 100%;  
    padding: 10px;  
    background-color: #5cb85c;  
    color: #fff;  
    border: none;  
    border-radius: 5px;  
    cursor: pointer;  
    font-size: 16px;  
}  
  
form button:hover {  
    background-color: #4cae4c;  
}
```

## 9.9. Ejecución

Usamos "python manage.py runserver"

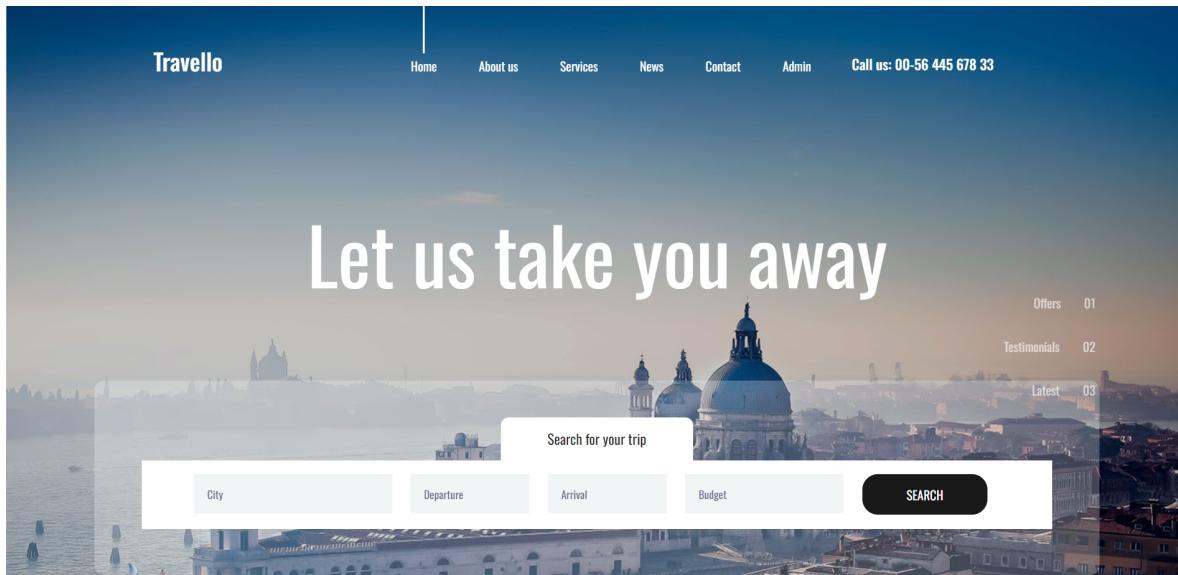


Figura 5: Home

Travello
Home    About us    Services    News    Contact    Admin    Call us: 00-56 445 678 33

## Popular Destinations



Arequipa  
Hermosa Ciudad Blanca

\$ 800,00 - Oferta: True



Lima  
Capital de Perú

\$ 900,00 - Oferta: False



Cusco  
La antigua capital del Imperio Inca, rica en historia.

\$ 900,00 - Oferta: True







Figura 6: Listar con tag for e if

Travello
Home    About us    Services    News    Contact    Call us: 00-56 445 678 33

## A few words about us

Pellentesque sit amet elementum ccumsan sit amet mattis  
 eget, tristique at leo. Vivamus massa.Tempor massa et  
 laoreet .Lorem ipsum dolor sit amet, consectetur adipiscing  
 elit.

Pellentesque sit amet elementum ccumsan sit amet mattis eget, tristique at leo.  
 Vivamus massa.Tempor massa et laoreet .Lorem ipsum dolor sit amet, consectetur  
 adipiscing elit. Sed eu laoreet ante, sollicitudin volutpat quam. Vestibulum posuere  
 malesuada ultrices. In pulvinar rhoncus lacus at aliquet. Nunc vitae lacus varius,  
 auctor nisi sit amet, consectetur mauris. Curabitur sodales semper est, vel faucibus  
 urna laoreet vel. Ut justo diam, sodales non pulvinar at, vulputate quis neque. Etiam  
 aliquam purus vel ultricies consequat.

[READ MORE](#)



Figura 7: About

Travello
Home
About us
Services
News
Contact
Call us: 00-56 445 678 33



### Categories

Travels	(09)
Organization	(12)
Tips & Tricks	(16)
Uncategorized	(22)

### Latest News



02 JUNE
Best tips to travel light
Pellentesque sit amet...

02 JUNE
Best tips to travel light

**02 JUNE**

**Best tips to travel light**

LIFESTYLE & TRAVEL

Pellentesque sit amet elementum ccumsan sit amet mattis eget, tristique at leo. Vivamus massa. Tempor massa et



02 JUNE
Best tips to travel light
Pellentesque sit amet...

Figura 8: News

Travello
Home
About us
Services
News
Contact
Call us: 00-56 445 678 33





02 JUNE
Best tips to travel light
Pellentesque sit amet...

### Get in touch with us

Pellentesque sit amet elementum ccumsan sit amet mattis eget, tristique at leo.  
 Vivamus massa. Tempor massa et laoreet. Pellentesque sit amet elementum ccumsan sit amet mattis eget, tristique at leo. Vivamus massa.

Your Name

**ADDRESS:**

1481 Creekside Lane Avila Beach, CA 931

Your E-mail

Subject

**PHONE:**

+53 345 7953 32453

Message

**EMAIL:**

SEND

Figura 9: Contact

### Agregar Lugar

NombreCiudad:

ImagenCiudad:  Ningún archivo seleccionado

PrecioTour:

OfertaTour:

**Guardar**

### Eliminar Lugar

NombreCiudad:

Figura 10: Admin - Formulario Agregar

### Eliminar Lugar

PrecioTour:

OfertaTour:

**Guardar**

### Modificar Lugar

NombreCiudad:

**Guardar**

Figura 11: Admin - Formulario Eliminar y Modificar

### Modificar Lugar

NombreCiudad:

DescripcionCiudad:

ImagenCiudad: Actualmente: [images/images\\_dZnpNG5.jpg](#)  
Modificar:  Ningún archivo seleccionado

PrecioTour:

OfertaTour:

Figura 12: Admin - Formulario Modificar

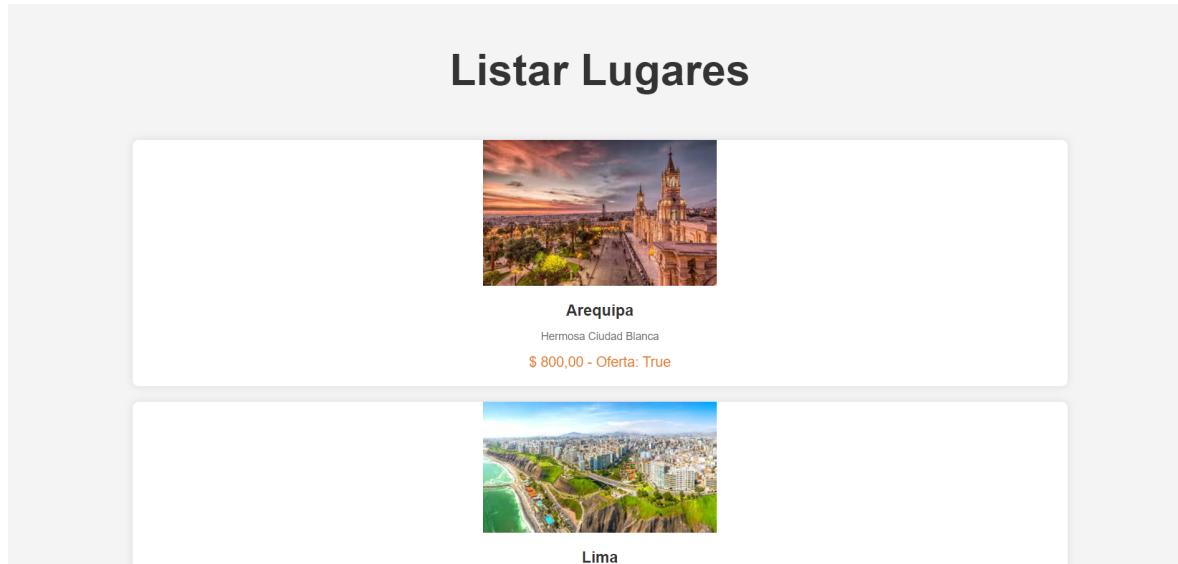


Figura 13: Admin - Listar 1

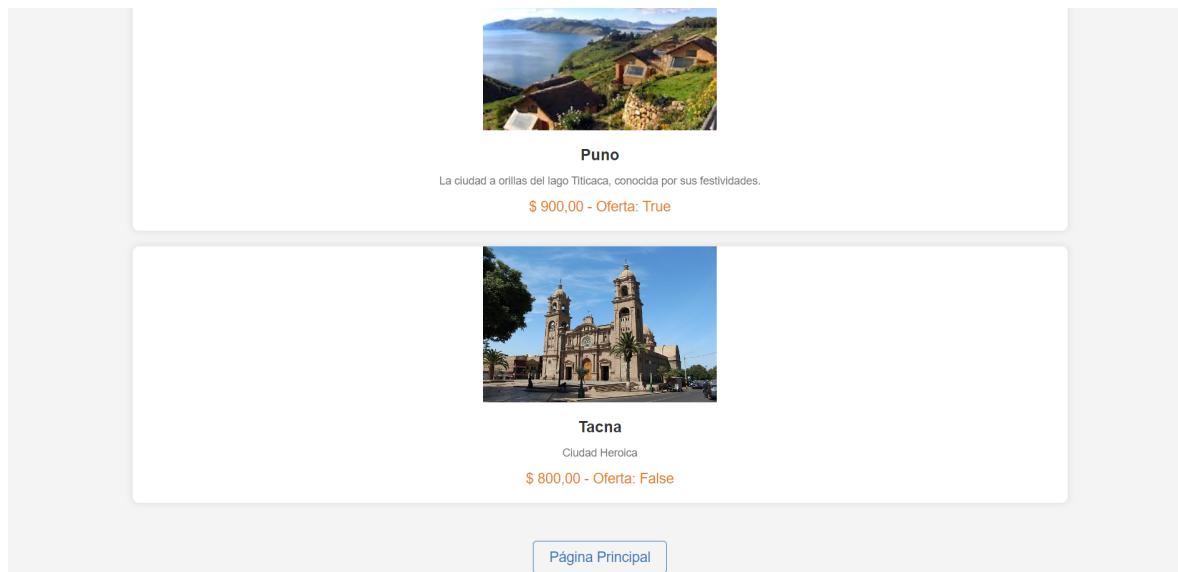


Figura 14: Admin - Listar 2

## 9.10. Uso de GitHub

### 9.10.1. Usuario de GitHub

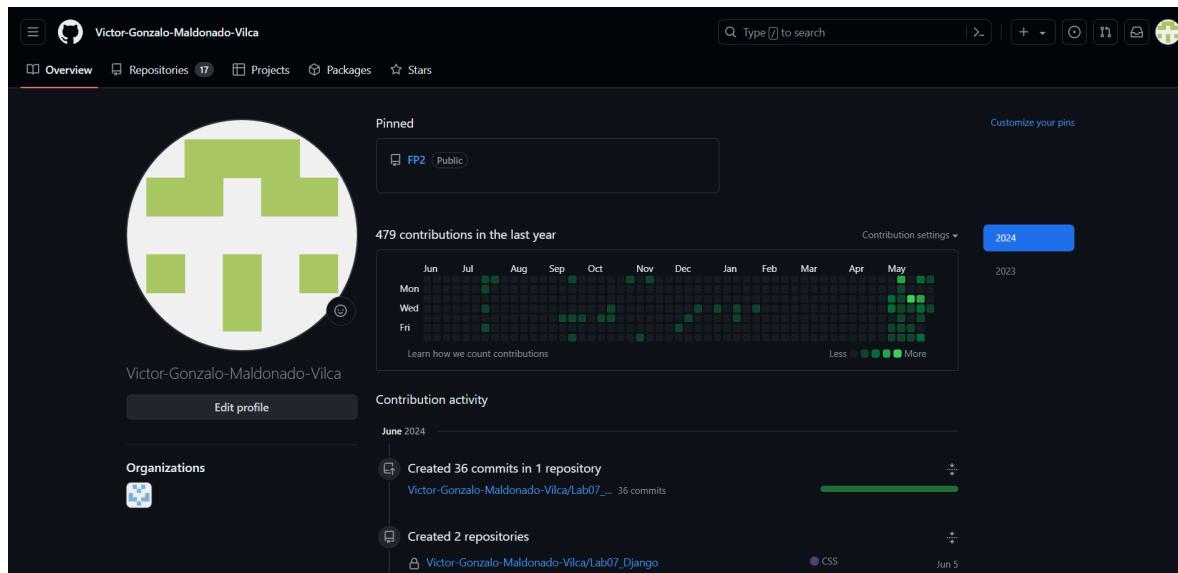


Figura 15: Usuario

### 9.10.2. Creación de un Nuevo Repositorio

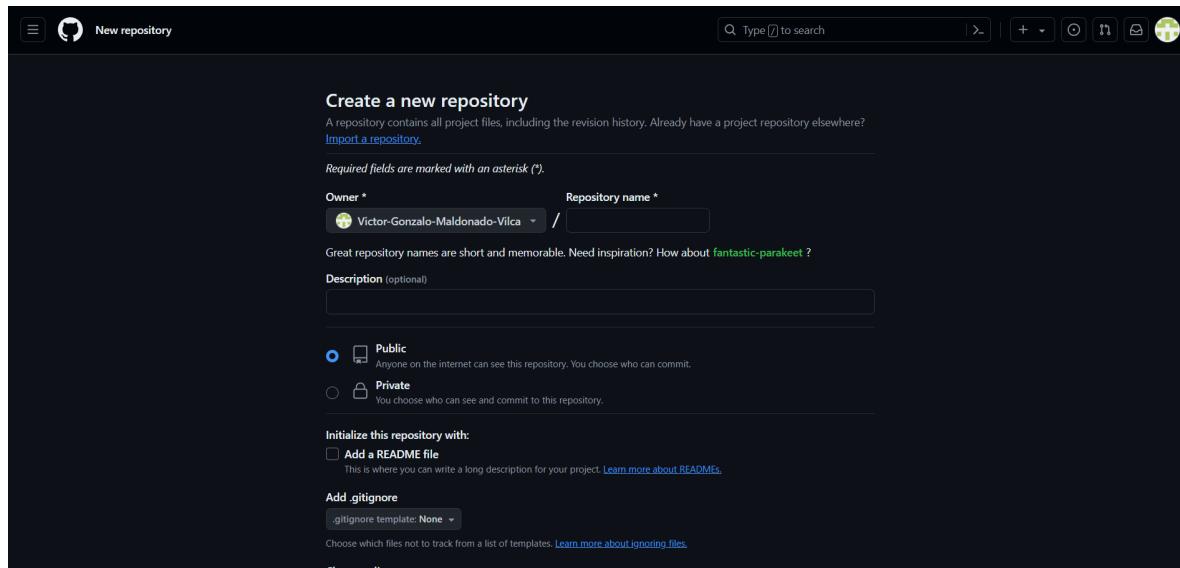


Figura 16: Nuevo Repositorio

### 9.10.3. Comandos de Configuración

```
echo "# lab07_Django" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin https://github.com/Victor-Gonzalo-Maldonado-Vilca/Lab07_Django.git
git push -u origin master
```

#### 9.10.4. Implementación de Readme.md

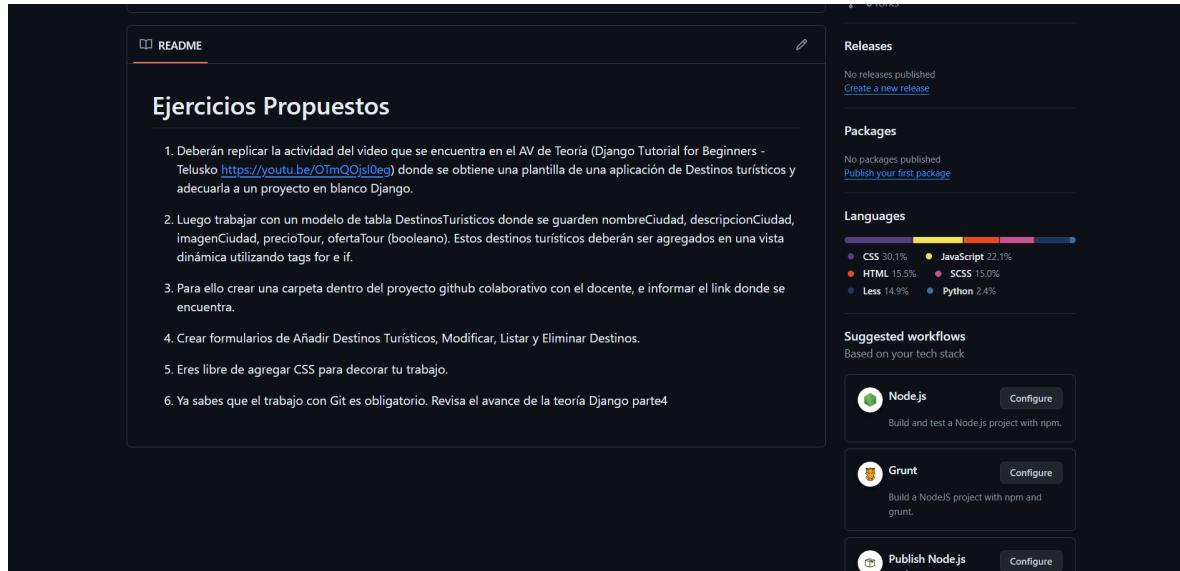


Figura 17: README.md

#### 9.10.5. Registro de cambios en mi código

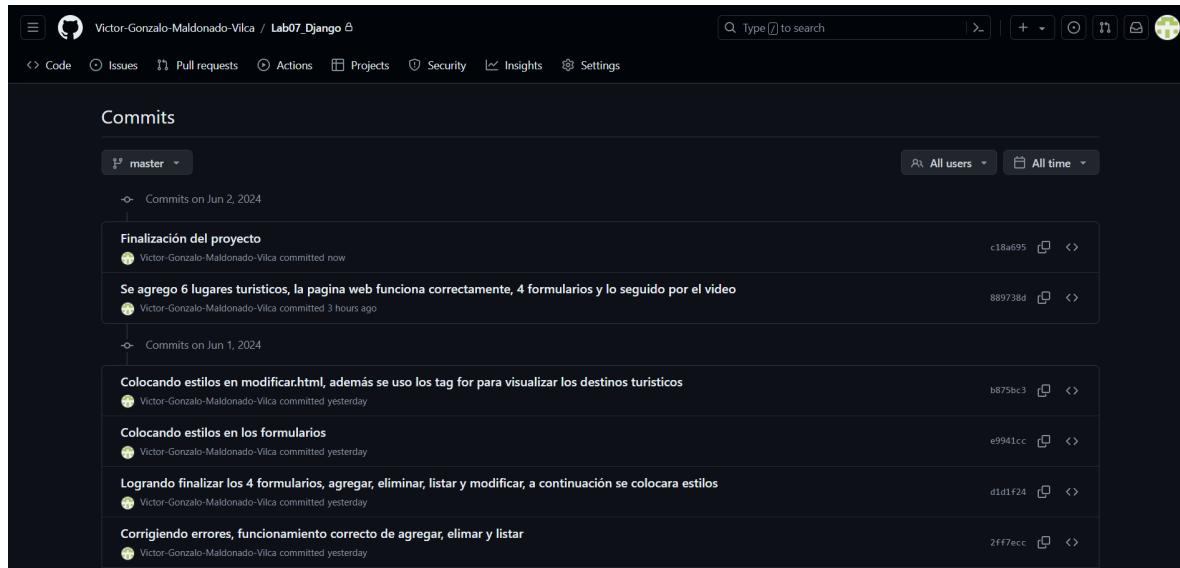


Figura 18: Commits

### 9.10.6. Repositorio

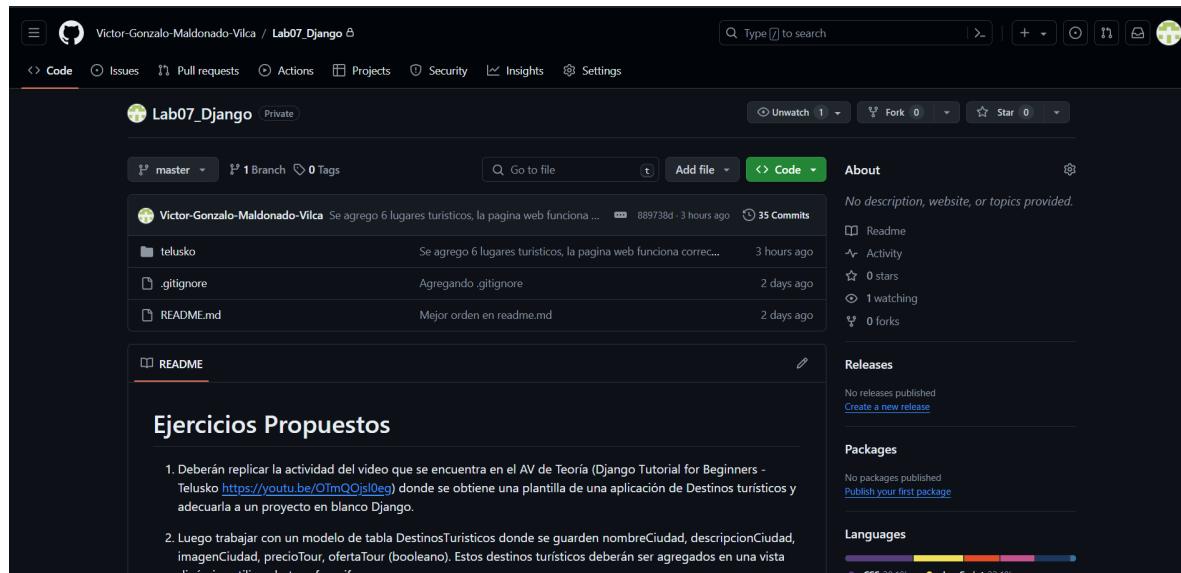


Figura 19: Repositorio

### 9.10.7. Proyecto compartido con el profesor de github

Figura 20: Compartir con el Docente

## 10. Recomendaciones

- Seguir las mejores prácticas de desarrollo de Django, como la separación de la lógica de negocios en vistas y la creación de plantillas reutilizables.
- Realizar pruebas unitarias y de integración para garantizar la calidad del código, y asegúrate de implementar medidas de seguridad en tu aplicación.
- Utilizar el sistema de plantillas de Django de manera efectiva para crear interfaces de usuario atractivas y funcionales.
- Implementar estrategias de caché y optimización de consultas para mejorar el rendimiento de la aplicación y reducir los tiempos de carga.

## 11. Conclusiones

- Django es un framework web potente y versátil que permite desarrollar aplicaciones web de manera rápida y eficiente.
- La arquitectura MVC (Modelo-Vista-Controlador) de Django ayuda a organizar el código de manera estructurada y modular, lo que facilita la mantenibilidad y escalabilidad de las aplicaciones.
- Con un buen entendimiento de Django y siguiendo las mejores prácticas de desarrollo, se pueden crear aplicaciones web robustas y de alto rendimiento.
- Django proporciona una amplia gama de funcionalidades integradas, como autenticación de usuarios, administración de contenido y seguridad, que facilitan el desarrollo de aplicaciones completas y seguras.
- La documentación detallada y los recursos de aprendizaje disponibles hacen que sea relativamente fácil para los desarrolladores aprender y dominar Django, lo que acelera el proceso de desarrollo y reduce los errores.

### 11.1. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumplió con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 1: Niveles de desempeño

<b>Puntos</b>	Nivel			
	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
<b>2.0</b>	0.5	1.0	1.5	2.0
<b>4.0</b>	1.0	2.0	3.0	4.0

Tabla 2: Rúbrica para contenido del Informe y demostración

	Contenido y demostración	Puntos	Checklist	Estudiante	Profesor
<b>1. GitHub</b>	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
<b>2. Commits</b>	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	4	
<b>3. Código fuente</b>	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
<b>4. Ejecución</b>	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
<b>5. Pregunta</b>	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
<b>6. Fechas</b>	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
<b>7. Ortografía</b>	El documento no muestra errores ortográficos.	2	X	2	
<b>8. Madurez</b>	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	4	
<b>Total</b>		20		20	

## 12. Referencias

- <https://docs.djangoproject.com/en/5.0/>
- <https://docs.github.com/es>
- <https://git-scm.com/doc>
- <https://www.youtube.com/watch?v=0TmQ0jsl0eg>
- <https://www.dropbox.com/sh/gf8x8xjwidq20cz/AABP10sXmijnz0-yjxYj9ebsa?dl=0>