Este Tutorial Esta Orientado a Inmobiliarias

I PARTE , Configuración

1. mkdir djimages , cd djimages , pipenv shell, pipenv install django

2. django-admin startproject djimages_project . (Que no se Te olvide el punto)

3. django-admin startapp inmuebles

4. Agregar la Nueva app:

```
   INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'inmuebles', #NEW
]
```

5. En Templates de settings realiza esta configuración(No se que tan importante será):

```
TEMPLATES = [
{
'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [],
#'DIRS': [os.path.join(BASE_DIR,'templates')], ← Cuando templates está en root
'APP_DIRS': True,
```

6. Al final de settings.py:(MEDIA FILES)

```
   STATIC_URL ='/static/'

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

II PARTE , Creación De Modelos:
1. El de Inmuebles y el de las Imágenes de los Inmuebles:

```
from django.db import models

class Inmueble(models.Model):
   codigo = models.CharField(max_length=100)
   tipo = models.CharField(max_length=50)
   ubicacion = models.CharField(max_length=1000)
   edo = models.CharField(max_length=1000,blank=True)
   precio = models.DecimalField(max_digits=7,decimal_places=2)
   description = models.TextField()
   image = models.FileField(blank=True)

   def __str__(self):
```

```
        template = '{0.codigo} {0.tipo} {0.ubicacion} {0.edo} {0.precio}'
        return template.format(self)

class InmuebleImage(models.Model):
    inmueble = models.ForeignKey(Inmueble, default=None,
on_delete=models.CASCADE)
    images = models.FileField(upload_to = 'images/')

    def __str__(self):
        return self.inmueble.codigo
```

2. python manage.py makemigrations inmuebles ← Para aplicar las migraciones directamente a la App

3. python manage.py migrate

III PARTE, Creación del View.py

```
from django.shortcuts import render, get_object_or_404

from .models import Inmueble,InmuebleImage

def inmueble_view(request):
    inmuebles = Inmueble.objects.all()
    return render(request, 'inmueble.html', {'inmuebles':inmuebles})

def detalle_view(request, id):
    inmueble = get_object_or_404(Inmueble, id=id)
    #Va a filtrar un inmueble en particular
    fotos = InmuebleImage.objects.filter(inmueble=inmueble)
    return render(request, 'detalle.html', {
                        'inmueble':inmueble,
                        'fotos':fotos
                       })
```

IV PARTE, Admin.py:

La interface de admin tiene la habilidad de editar modelos en la misma página como un modelo padre. A esto se le llama **inline(class StackedInline)**

```
from django.contrib import admin
from .models import Inmueble, InmuebleImage

class InmuebleImageAdmin(admin.StackedInline): ←--
    model = InmuebleImage

@admin.register(Inmueble)
class PostAdmin(admin.ModelAdmin):
    inlines = [InmuebleImageAdmin]

class Meta:
    model = Inmueble

@admin.register(InmuebleImage)
class InmuebleImageAdmin(admin.ModelAdmin):
    pass
```

V PARTE, Templates (3:36m):

1.En este Tutorial hay una particularidad , **colocó la carpeta template en root**
(Debe ser por eso la configuracion inicial en settings):

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
     'DIRS': [os.path.join(BASE_DIR,'templates')], #NEW
```

2.
creó tres html: base.html, inmueble.html, detalle.html

3. Edición del urls.py principal:

```
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static



urlpatterns = [
    path('admin/', admin.site.urls),
    path('',include('inmuebles.urls')),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

urls.py de la app:

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

urlpatterns = [
    path('', views.inmueble_view, name='inmueble'),
    path('<int:id>/', views.detalle_view, name='detalle')
]
```

4. Utilizar bootstrap para las tarjetas(con botón)

https://mdbootstrap.com/ --→ Jquery(La sección) → Menú de la izquierda → components → Cards

https://mdbootstrap.com/docs/b4/jquery/components/cards/

```
<div class="view overlay">
<img class="card-img-top" src="{{ inmueble.image.url }}"
alt="Card image cap">
<a href="#!">
<div class="mask rgba-white-slight"></div>
</a>
</div>
<!--Card content-->
<div class="card-body">
<!--Title-->
<h4 class="card-title">{{inmueble.tipo}}</h4>
```

```
<!--Text-->
<p class="card-text">Some quick example text to build on the card title and make up the bulk of the
card's content.</p>
<!-- Provides extra visual weight and identifies the primary action in a set of buttons -->
<a href="{% url 'detail' inmueble.id %}" class="btn btn-light-blue btn-md">Read more</a>
```

Que es inmueble.id ???

CAROUSEL

```
{% for p in fotos %}
<li data-target="#carousel-example-1z"
data-slide-to="{{forloop.counter0}}"  ←------ Que ????
class="{% if forloop.counter0 == 0 %} active {% endif %}">
</li>
{% endfor %}
```

VI PARTE CREATESUPERUSER

REVISA creo que media/images/ se crea solo