

Requerimientos

1. Instalación de desarrollo, para esto el ambiente de desarrollo debe contar con las siguientes características:

- a. Una instalación de PostgreSQL ([link](#))
- b. Creación de un ambiente virtual de Python.

```
Alondra@LAPTOP-LSLRMH15 MINGW64 ~/OneDrive - Universidad May...  
on/TD_python/M7/Proyecto_inmobiliario/Hito1  
$ cd ..  
  
Alondra@LAPTOP-LSLRMH15 MINGW64 ~/OneDrive - Universidad May...  
on/TD_python/M7/Proyecto_inmobiliario  
$ source .venv/scripts/activate  
(.venv)
```

c. Instalación de los paquetes necesarios para la creación de un proyecto de Django

d. Una aplicación de Django.

```
Alondra@LAPTOP-LSLRMH15 MINGW64 ~/OneDrive - Universidad May...  
io/Curso full stack python/TD_python/M7/Proyecto_inmobiliario  
yecto_inmuebles  
$ python manage.py startapp gestion_inmuebles  
(.venv)
```

(3 Puntos)

2. Definición de modelo de datos para representación del problema utilizando el framework Django:

- a. Representación del modelo relacional de datos.

```

proyecto_inmuebles > gestion_inmuebles > models.py > SolicitudArriendo
1 from django.db import models
2 from django.contrib.auth.models import User
3
4 class Inmueble(models.Model):
5     TIPO_INMUEBLE_CHOICES = [
6         ('casa', 'Casa'),
7         ('departamento', 'Departamento'),
8         ('oficina', 'Oficina'),
9         ('local_comercial', 'Local Comercial'),
10        ('terreno', 'Terreno'),
11    ]
12    nombre = models.CharField(max_length=200)
13    descripcion = models.TextField(null=True, blank=True)
14    m2_construidos = models.DecimalField(max_digits=5, decimal_places=2)
15    m2_totales = models.DecimalField(max_digits=5, decimal_places=2)
16    estacionamientos = models.IntegerField(default=0)
17    habitaciones = models.IntegerField(default=1)
18    banos = models.IntegerField(default=1)
19    direccion = models.CharField(max_length=200)
20    comuna = models.CharField(max_length=100)
21    tipo_inmueble = models.CharField(max_length=50, choices=TIPO_INMUEBLE_CHOICES, default='casa')
22    precio_mensual = models.DecimalField(max_digits=10, decimal_places=2, default=0.00)
23    arrendador = models.ForeignKey('Profile', on_delete=models.CASCADE, null=False, blank=False, limit_choices_to={'tipo_usuario': 'arrendador'})
24
25    def __str__(self):
26        return self.nombre
27
28 class Profile(models.Model):
29     TIPO_USUARIO_CHOICES = [
30         ('arrendatario', 'Arrendatario'),
31         ('arrendador', 'Arrendador'),
32     ]
33     user = models.OneToOneField(User, on_delete=models.CASCADE)
34     nombres = models.CharField(max_length=100)
35     apellidos = models.CharField(max_length=100)
36     rut = models.CharField(max_length=12, unique=True)
37     direccion = models.CharField(max_length=200)
38     telefono = models.CharField(max_length=15)
39     tipo_usuario = models.CharField(max_length=20, choices=TIPO_USUARIO_CHOICES, default='arrendatario')
40
41     def __str__(self):
42         return f"{self.nombres} {self.apellidos} - {self.tipo_usuario}"

```

```

class SolicitudArriendo(models.Model):
    ESTADO_SOLICITUD_CHOICES = [
        ('pendiente', 'Pendiente'),
        ('aceptada', 'Aceptada'),
        ('rechazada', 'Rechazada'),
    ]
    inmueble = models.ForeignKey('Inmueble', on_delete=models.CASCADE, related_name='solicitudes')
    arrendatario = models.ForeignKey(Profile, on_delete=models.CASCADE, limit_choices_to={'tipo_usuario': 'arrendatario'})
    fecha_solicitud = models.DateTimeField(auto_now_add=True)
    estado = models.CharField(max_length=10, choices=ESTADO_SOLICITUD_CHOICES, default='pendiente')

    def __str__(self):
        return f"Solicitud de {self.arrendatario} para {self.inmueble} - {self.estado}"

```

b. Conexión a la base de datos

```

DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.sqlite3",
        "NAME": BASE_DIR / "db.sqlite3",
    }
}

```

c. Definición y manejo de llaves primarias en columnas foráneas

```

arrendador = models.ForeignKey('Profile', on_delete=models.CASCADE, null=False, blank=False, limit_choices_to={'tipo_usuario': 'arrendador'})

```

```
user = models.OneToOneField(User, on_delete=models.CASCADE)
```

```
arrendatario = models.ForeignKey(Profile, on_delete=models.CASCADE, limit_choices_to={'tipo_usuario': 'arrendatario'})
```

(3 Puntos)

3. Implementar operaciones en los modelos para la manipulación de datos utilizando el framework Django:

a. Crear un objeto con el modelo.

```
>>> arrendador_profile = Profile.objects.create( user=user, nombres='Nombre del Arrendador', apellidos='Apellido del Arrendador', rut='12345678-9', direccion='Calle Ficticia 123', telefono='987654321', tipo_usuario='arrendador')
>>> arrendador_profile = Profile.objects.get(user__username='arrendador123')
>>> print(arrendador_profile.nombres)
Nombre del Arrendador
```

b. Enlistar desde el modelo de datos.

```
>>> inmuebles = Inmueble.objects.all()
>>> for inmueble in inmuebles:
...     print(inmueble.nombre)
...
Departamento en Santiago
>>>
```

```
>>> arrendadores = Profile.objects.filter(tipo_usuario='arrendador')
>>> for arrendador in arrendadores:
...     print(f'{arrendador.nombres} {arrendador.apellidos}')
...
Nombre del Arrendador Apellido del Arrendador
>>>
```

c. Actualizar un registro en el modelo de datos.

```
>>> inmueble = Inmueble.objects.get(nombre='Departamento en Santiago')
>>> inmueble.precio_mensual = 350000
>>> inmueble.save()
>>> print(inmueble.precio_mensual)
350000
```

d. Borrar un registro del modelo de datos utilizando un modelo Django.

```
>>> arrendador_profile = Profile.objects.get(user__username='arrendador123')
n Santiago', descripcion='Departamento de 2 habitaciones y 2 baños', m2_construidos=60.5, m2_totales=80.0, estacionamientos=1, habitaciones=2, banos=2, direccion='Av. Libertador Bernardo O'Higgins 1000', comuna='Santiago', tipo_inmueble='departamento', precio_mensual=300000, arrendador=arrendador_profile)
>>> inmueble = Inmueble.objects.get(nombre='Departamento en Santiago')
>>> inmueble.delete()
(1, {'gestion_inmuebles.Inmueble': 1})
>>> inmuebles = Inmueble.objects.all()
>>> for inmueble in inmuebles:
...     print(inmueble.nombre)
...
>>>
```

```
>>> inmueble = Inmueble.objects.get(nombre='Departamento en Santiago')
>>> inmueble.delete()
(1, {'gestion_inmuebles.Inmueble': 1})
>>> inmuebles = Inmueble.objects.all()
>>> for inmueble in inmuebles:
...     print(inmueble.nombre)
...
>>>
```

(3 Puntos)

4. Documentar el proceso de acuerdo a las tareas asignadas. (1 Punto)