

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 Mayaguez/
on/TD_python/M7/Proyecto_inmobiliario/Hito1
$ cd ..

Alondra@LAPTOP-LSLRMH15 MINGW64 ~/OneDrive - Universidad Mayaguez/
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

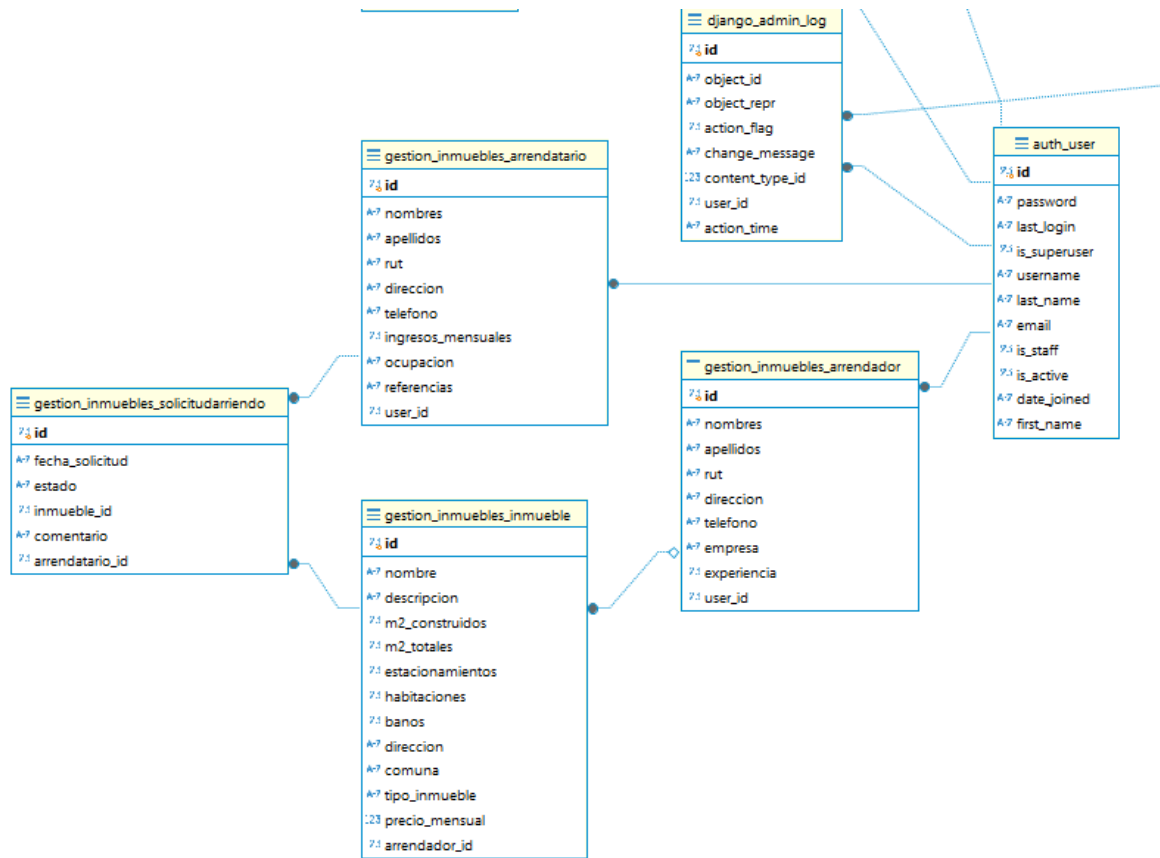
d. Una aplicación de Django.

```
Alondra@LAPTOP-LSLRMH15 MINGW64 ~/OneDrive - Universidad Mayaguez/
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.



```

1  from django.db import models
2  from django.contrib.auth.models import User
3
4  class Persona(models.Model):
5      """Abstract base model for common fields of Arrendador and Arrendatario."""
6      user = models.OneToOneField(User, on_delete=models.CASCADE)
7      nombres = models.CharField(max_length=100)
8      apellidos = models.CharField(max_length=100)
9      rut = models.CharField(max_length=12, unique=True)
10     direccion = models.CharField(max_length=200)
11     telefono = models.CharField(max_length=15)
12
13     class Meta:
14         abstract = True # This model is abstract and won't create a separate table.
15
16     def __str__(self):
17         return f"{self.nombres} {self.apellidos}"
18
19
20     class Arrendador(Persona):
21         """Model for property owners."""
22         empresa = models.CharField(max_length=100, blank=True, null=True)
23         experiencia = models.IntegerField(default=0, help_text="Years of experience as a property owner.")
24
25     def __str__(self):
26         return f"Arrendador: {self.nombres} {self.apellidos}"
27
28
29     class Arrendatario(Persona):
30         """Model for tenants."""
31         ingresos_mensuales = models.DecimalField(max_digits=10, decimal_places=2, help_text="Monthly income of the tenant.")
32         ocupacion = models.CharField(max_length=100, help_text="Current occupation of the tenant.")
33         referencias = models.TextField(blank=True, null=True, help_text="References from previous landlords.")
34
35     def __str__(self):
36         return f"Arrendatario: {self.nombres} {self.apellidos}"
37

```

```

class Inmueble(models.Model):
    TIPO_INMUEBLE_CHOICES = [
        ('casa', 'Casa'),
        ('departamento', 'Departamento'),
        ('oficina', 'Oficina'),
        ('local_comercial', 'Local Comercial'),
        ('terreno', 'Terreno'),
    ]
    nombre = models.CharField(max_length=200)
    descripcion = models.TextField(null=True, blank=True)
    m2_construidos = models.DecimalField(max_digits=5, decimal_places=2)
    m2_totales = models.DecimalField(max_digits=5, decimal_places=2)
    estacionamientos = models.IntegerField(default=0)
    habitaciones = models.IntegerField(default=1)
    banos = models.IntegerField(default=1)
    direccion = models.CharField(max_length=200)
    comuna = models.CharField(max_length=100)
    tipo_inmueble = models.CharField(max_length=50, choices=TIPO_INMUEBLE_CHOICES, default='casa')
    precio_mensual = models.DecimalField(max_digits=10, decimal_places=2, default=0.00)
    arrendador = models.ForeignKey(Arrendador, on_delete=models.CASCADE, null=True, blank=True, related_name='inmuebles')

    def __str__(self):
        return self.nombre

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(Arrendatario, on_delete=models.CASCADE, related_name='solicitudes')
    fecha_solicitud = models.DateTimeField(auto_now_add=True)
    estado = models.CharField(max_length=10, choices=ESTADO_SOLICITUD_CHOICES, default='pendiente')
    comentario = models.TextField(blank=True, null=True, help_text="Additional comments by the tenant.")

    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

```

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

```

```

arrendador = models.ForeignKey(Arrendador, on_delete=models.CASCADE, null=True, blank=True, related_name='inmuebles')

```

```

arrendatario = models.ForeignKey(Arrendatario, on_delete=models.CASCADE, related_name='solicitudes')

```

(3 Puntos)

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

framework Django:

a. Crear un objeto con el modelo.

```
<interactive console>
>>> from django.contrib.auth.models import User
>>> from app_name.models import Arrendador, Arrendatario, Inmueble, SolicitudArriendo
Traceback (most recent call last):
  File "<console>", line 1, in <module>
ModuleNotFoundError: No module named 'app_name'
>>> from gestion_inmuebles.models import Arrendador, Arrendatario, Inmueble, SolicitudArriendo
Traceback (most recent call last):
  File "<console>", line 1, in <module>
ImportError: cannot import name 'SolicitudArriendo' from 'gestion_inmuebles.models' (C:\Users\Alondra\OneDrive - Universidad Mayor\Escritorio\Curso full stack python\TD_python\7\Proyecto_inmobiliario\hitot\proyecto_inmuebles\gestion_inmuebles\models.py)
>>>                                     from gestion_inmuebles.models import Arrendador, Arrendatario, Inmueble, SolicitudArriendo
>>> user_arrendador = User.objects.create_user(username='arrendador_user', password='password123')
>>>
>>> user_arrendatario = User.objects.create_user(username='arrendatario_user', password='password123')
>>> arrendador = Arrendador.objects.create(
...     user=user_arrendador,
...     nombres="Juan",
...     apellidos="Perez",
...     rut="12345678-9",
...     direccion="Calle Falsa 123",
...     telefono="123456789",
...     empresa="Propiedades Perez",
...     experiencia=5
... )
>>>
>>> arrendatario = Arrendatario.objects.create(
...     user=user_arrendatario,
...     nombres="Maria",
...     apellidos="Gonzalez",
...     rut="98765432-1",
...     direccion="Avenida Siempreviva 742",
...     telefono="987654321",
...     ingresos_mensuales=2000.00,
```

b. Enlistar desde el modelo de datos.

```

... )
>>>
>>> inmueble = Inmueble.objects.create(
...     nombre="Casa en la playa",
...     descripcion="Casa con vista al mar, ideal para vacaciones.",
...     m2_construidos=120.50,
...     m2_totales=250.75,
...     estacionamientos=2,
...     habitaciones=3,
...     banos=2,
...     direccion="Playa Blanca 45",
...     comuna="Valparaíso",
...     tipo_inmueble="casa",
...     precio_mensual=750000,
...     arrendador=arrendador
... )
>>>
>>> solicitud_arriendo = SolicitudArriendo.objects.create(
...     inmueble=inmueble,
...     arrendatario=arrendatario,
...     estado="pendiente",
...     comentario="Interesado en arrendar por al menos 6 meses."
... )
>>> arrendadores = Arrendador.objects.all()
>>> for arrendador in arrendadores:
...     print(arrendador)
...
Arrendador: Juan Perez
>>> █

```

```

Arrendador: Juan Perez
>>> inmuebles = Inmueble.objects.all()
>>> for inmueble in inmuebles:
...     print(inmueble)
...
Casa en la playa
>>> █

```

c. Actualizar un registro en el modelo de datos.

```
>>> arrendador = Arrendador.objects.get(id=1)
>>> arrendador.nombres = "Juan Carlos"
>>> arrendador.direccion = "Nueva Direccion 456"
>>> arrendador.save()
>>> █
```

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

```
>>> arrendador = Arrendador.objects.get(id=1)
>>> arrendador.delete()
(3, {'gestion_inmuebles.SolicitudArriendo': 1, 'gestion_inmuebles.Arrendador': 1, 'gestion_inmuebles.Inmueble': 1})
>>> █
```

(3 Puntos)

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