

Proyecto Inmobiliaria.

Módulo 7 - Hito 1.

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.
- C. Instalación de los paquetes necesarios para la creación de un proyecto de Django
- D. Una aplicación de Django.

(3 Puntos)

Configuración de base de datos, instalaciones y migraciones.

```
requirements.txt
1 |
2 | asgiref==3.8.1
3 | Django==4.2
4 | psycopg2==2.9.9
5 | python-dotenv==1.0.1
6 | sqlparse==0.5.0
7 | typing_extensions==4.11.0
8 |

portal > settings.py > MEDIA_URL
84 DATABASES = {
85     'default': {
86         'ENGINE': os.getenv('ENGINE'),
87         'NAME': os.getenv('NAME'),
88         'USER': os.getenv('USERDB'),
89         'PASSWORD': os.getenv('PASSWORD'),
90         'HOST': os.getenv('HOSTDB'),
91         'PORT': os.getenv('PORT'),
92     }
93 }
```

▼	Tablas (13)
>	arriendos_inmueble
>	arriendos_solicitudarriendo
>	arriendos_usuario
>	auth_group
>	auth_group_permissions
>	auth_permission
>	auth_user
>	auth_user_groups
>	auth_user_user_permissions
>	django_admin_log
>	django_content_type
>	django_migrations
>	django_session

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

- A. Representación del modelo relacional de datos.
- B. Conexión a la base de datos
- C. Definición y manejo de llaves primarias en columnas foráneas

(3 Puntos)

Representación del modelo relacional de datos.

```
class Usuario(User):
    TIPO_USUARIO_CHOICES = [
        ('arrendatario', 'Arrendatario'),
        ('arrendador', 'Arrendador'),
    ]
    nombres = models.CharField(max_length=50)
    apellidos = models.CharField(max_length=50)
    rut = models.CharField(max_length=10, unique=True)
    direccion = models.CharField(max_length=100)
    telefono = models.CharField(max_length=13)
    tipo_usuario = models.CharField(max_length=20, choices=TIPO_USUARIO_CHOICES)
    correo_electronico = models.EmailField(unique=True, default='correo@correo.cl')

    def __str__(self):
        return f"{self.nombres} {self.apellidos}"

class Inmueble(models.Model):
    TIPO_INMUEBLE_CHOICES = [
        ('casa', 'Casa'),
        ('departamento', 'Departamento'),
        ('parcela', 'Parcela'),
    ]
    nombre = models.CharField(max_length=50)
    direccion = models.CharField(max_length=50)
    descripcion = models.CharField(max_length=50)
    imagen = models.ImageField(upload_to='img/inmuebles/')
    precio = models.DecimalField(max_digits=10, decimal_places=0)
    comuna = models.CharField(max_length=50, choices=COMUNAS_REGION_METROPOLITANA)
    disponible = models.BooleanField(default=True)
    m2_construidos = models.DecimalField(max_digits=10, decimal_places=2)
    m2_terreno = models.DecimalField(max_digits=10, decimal_places=2)
    cantidad_estacionamientos = models.PositiveIntegerField()
    cantidad_habitaciones = models.PositiveIntegerField()
    cantidad_banos = models.PositiveIntegerField()
    tipo_de_inmueble = models.CharField(max_length=12, choices=TIPO_INMUEBLE_CHOICES)
    propietario = models.ForeignKey(Usuario, on_delete=models.CASCADE)

    def __str__(self) -> str:
        return f"{self.nombre}"
```

```
class SolicitudArriendo(models.Model):
    arrendatario = models.ForeignKey(Usuario, on_delete=models.CASCADE)
    inmueble = models.ForeignKey(Inmueble, on_delete=models.CASCADE)
    mensaje = models.TextField(blank=True)
    def __str__(self):
        return f"Solicitud de {self.inmueble.nombre} por {self.arrendatario.nombres} {self.arrendatario.apellidos}"
```

Conección a la base de datos

```
DATABASES = {
    'default': {
        'ENGINE': 'os.getenv('ENGINE')',
        'NAME': 'os.getenv('NAME')',
        'USER': 'os.getenv('USERDB')',
        'PASSWORD': 'os.getenv('PASSWORD')',
        'HOST': 'os.getenv('HOSTDB')',
        'PORT': 'os.getenv('PORT')',
    }
}
```

Definición y manejo de llaves primarias en columnas foráneas

The screenshot displays the Django Admin interface for defining foreign key relationships. It shows two tables: 'arriendos_inmueble' and 'arriendos_solicitudarriendo'. For each table, the 'Restricciones' (Constraints) tab is selected, and the 'Clave Foránea' (Foreign Key) sub-tab is active. The interface lists the foreign key constraints with their names, the columns involved, and the referenced table.

Nombre	Columnas	Referenced Table
arriendos_inmueble_propietario_id_5f	(propietario_id) -> (user_ptr_id)	public.arriendos_usuario
arriendos_solicitudarriendo_arrendatario_id_5f	(arrendatario_id) -> (user_ptr_id)	public.arriendos_usuario
arriendos_solicitudarriendo_inmueble_id_65f	(inmueble_id) -> (id)	public.arriendos_inmueble

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

- A. Crear un objeto con el modelo.
- B. Enlistar desde el modelo de datos.
- C. Actualizar un registro en el modelo de datos.
- D. Borrar un registro del modelo de datos utilizando un modelo Django.

(4 Puntos)

Crear un objeto con el modelo.

```
>>> usuario_marcos = crear_usuario(
...     nombres="Marcos",
...     apellidos="Jose",
...     rut="98765435",
...     direccion="Calle San Martín 123",
...     telefono="+56912345678",
...     tipo_usuario="arrendador",
...     correo_electronico="marcos.jose5@ejemplo.com",
...     password="11223344",
...     username="marcosjose5"
... )
Usuario creado con éxito: marcosjose5
>>> inmueble = crear_inmueble(
...     nombre="Departamento en Providencia",
...     direccion="Av. Andrés Bello 1500",
...     descripcion="Amplio departamento de 3 dormitorios",
...     imagen="departamento.jpg",
...     precio=1200000,
...     comuna="Providencia",
...     disponible=True,
...     m2_construidos=75,
...     m2_terreno=0,
...     cantidad_estacionamientos=1,
...     cantidad_habitaciones=3,
...     cantidad_banos=2,
...     tipo_de_inmueble="departamento",
...     propietario=usuario_marcos
... )
>>> []
```

Enlistar desde el modelo de datos.

```
(env) andrea0690@MacBook-Pro-de-Andrea portal_inmobiliario % python3 manage.py shell
Python 3.9.6 (default, Feb 3 2024, 15:58:27)
[Clang 15.0.0 (clang-1500.3.9.4)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from arriendos.services import *
>>> enlistar_inmuebles()
ID: 9
Nombre inmueble: Casa
Nombre inmueble: casa
Propietario: Serena Gutierrez

-----
ID: 10
Nombre inmueble: Casa2
Nombre inmueble: casa
Propietario: Serena Gutierrez

-----
ID: 11
Nombre inmueble: Casa exclusiva
Nombre inmueble: casa
Propietario: Serena Gutierrez

-----
ID: 13
Nombre inmueble: Departamento en Providencia
Nombre inmueble: departamento
Propietario: Marcos Jose

-----
>>> 
```

Actualizar un registro en el modelo de datos.

```
(env) andrea0690@MacBook-Pro-de-Andrea portal_inmobiliario % python3 manage.py shell
Python 3.9.6 (default, Feb 3 2024, 15:58:27)
[Clang 15.0.0 (clang-1500.3.9.4)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from arriendos.services import *
>>> actualizar_campo_inmueble(9, "nombre", "Casa Hermosa")
"Campo 'nombre' actualizado exitosamente a 'Casa Hermosa' en inmueble 'Casa Hermosa'"
>>> 
```

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

```
(env) andrea0690@MacBook-Pro-de-Andrea portal_inmobiliario % python3 manage.py shell
Python 3.9.6 (default, Feb 3 2024, 15:58:27)
[Clang 15.0.0 (clang-1500.3.9.4)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from arriendos.services import *
>>> eliminar_inmueble(10)
"El inmueble '10' ha sido eliminado con éxito."
>>> 
```