

A graphic on the left side of the slide. It features four overlapping horizontal bars of different colors: pink, orange, yellow, and blue. The text 'Agencia de Aprendizaje a lo largo de la vida' is written across these bars in white. The orange bar has a white arrow pointing to the right at its end.

Agencia de
Aprendizaje
a lo largo
de la vida

DJANGO

Reunión 39

Models - 1

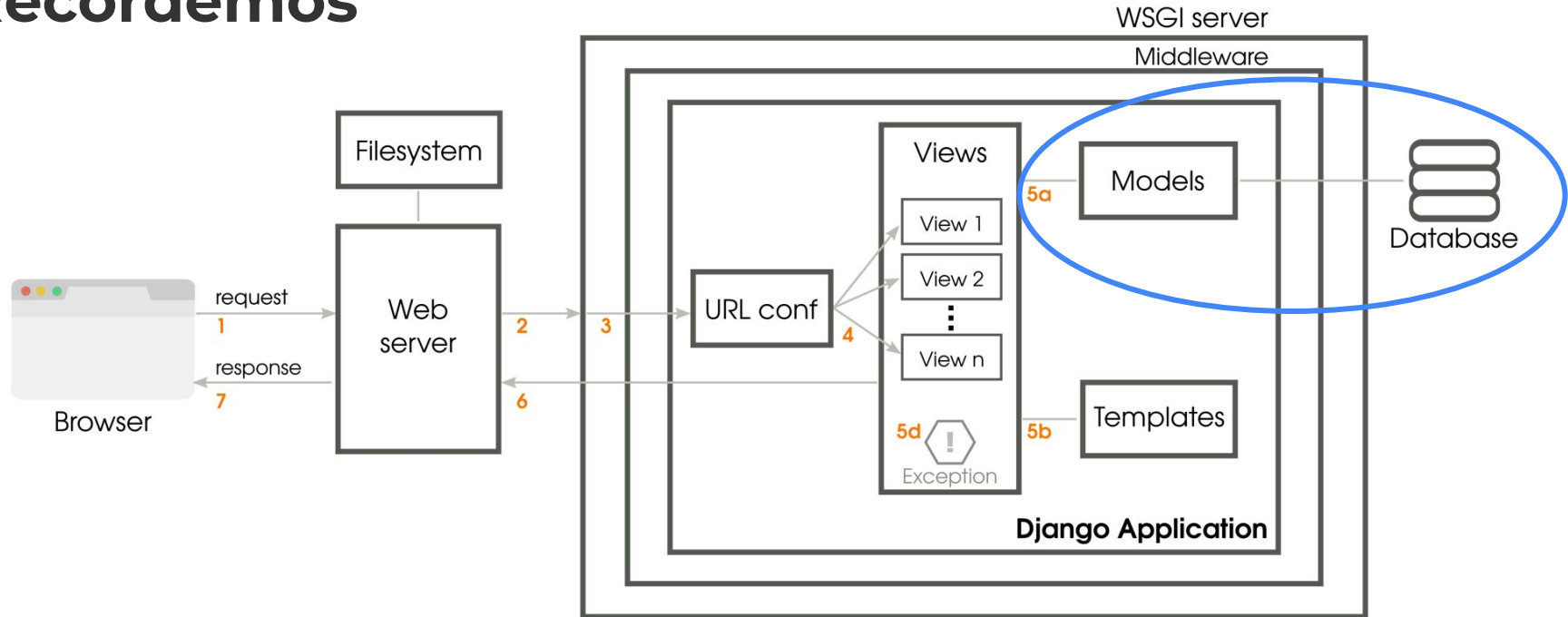
Les damos la bienvenida

Vamos a comenzar a grabar la clase

¿Qué es un Modelo?

Un modelo es la fuente única y definitiva de información sobre los datos en Django. Contiene los campos y comportamientos esenciales de los datos que se están almacenando. **Generalmente**, cada modelo se asigna a una tabla de base de datos.

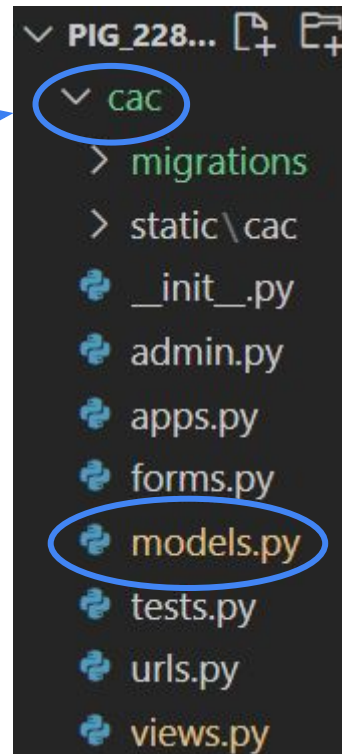
Recordemos



Creación de un Modelo

settings.py

```
INSTALLED_APPS = [  
    #...  
    'cac',  
    #...  
]
```



Creación de un Modelo

models.py

```
from django.db import models

class Estudiante(models.Model):
    nombre = models.CharField(max_length=100, verbose_name='Nombre')
    apellido = models.CharField(max_length=150, verbose_name='Apellido')
    email = models.EmailField(max_length=150, verbose_name='Email')
    dni = models.IntegerField(verbose_name='DNI')
```

Migración

python manage.py makemigrations

Migrations for 'cac':

cac\migrations\0001_initial.py

- Create model Estudiante

python manage.py migrate

Operations to perform:

Apply all migrations: cac

Running migrations:

Applying cac.0001_initial... OK

Creación de un Modelo

models.py

```
from django.db import models

class Estudiante(models.Model):
    nombre = models.CharField(max_length=100, verbose_name='Nombre')
    apellido = models.CharField(max_length=150, verbose_name='Apellido')
    email = models.EmailField(max_length=150, verbose_name='Email')
    dni = models.IntegerField(verbose_name='DNI')
```

migrations/0001_initial.py

```
python manage.py sqlmigrate public 0001
```

→ Vemos el sql sin ejecutar

```
migrations.CreateModel(
    name='Estudiante',
    fields=[
        ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
        ('nombre', models.CharField(max_length=100, verbose_name='Nombre')),
        ('apellido', models.CharField(max_length=150, verbose_name='Apellido')),
        ('email', models.EmailField(max_length=150, verbose_name='Email')),
        ('dni', models.IntegerField(verbose_name='DNI')),
    ],
),
```


Creación de un Modelo

postgresql

```
CREATE TABLE IF NOT EXISTS public.cac_estudiante
(
  id bigint NOT NULL DEFAULT nextval('cac_estudiante_id_seq'::regclass),
  nombre character varying(100) COLLATE pg_catalog."default" NOT NULL,
  apellido character varying(150) COLLATE pg_catalog."default" NOT NULL,
  email character varying(150) COLLATE pg_catalog."default" NOT NULL,
  dni integer NOT NULL,
  CONSTRAINT cac_estudiante_pkey PRIMARY KEY (id)
)
```

postgresql_psycopg2

```
data_types = {
    'BooleanField': 'boolean',
    'CharField': 'varchar(%(max_length)s)',
    'CommaSeparatedIntegerField': 'varchar(%(max_length)s)',
    'DateField': 'date',
    'DateTimeField': 'timestamp with time zone',
    'DecimalField': 'numeric(%(max_digits)s, %(decimal_places)s)',
    'DurationField': 'interval',
    'FileField': 'varchar(%(max_length)s)',
    'FilePathField': 'varchar(%(max_length)s)',
    'IntegerField': 'integer',
    'BigIntegerField': 'bigint',
    'NullBooleanField': 'boolean',
    'PositiveIntegerField': 'integer',
    'PositiveSmallIntegerField': 'smallint',
    'SmallIntegerField': 'smallint',
    'TextField': 'text',
    'TimeField': 'time',
    'UUIDField': 'uuid',
    .....
}
```

Django Shell (spoiler QuerySet)

python manage.py shell

Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

(InteractiveConsole)

```
>>> from cac.models import Estudiante
```

```
>>> Estudiante.objects.all()
```

```
<QuerySet []>
```

```
>>> Estudiante.objects.create(nombre="Alejandro", apellido="Perez", email="mail@mail.com", dni="123456")
```

```
<Estudiante: Estudiante object (1)>
```

```
>>> Estudiante.objects.all()
```

```
<QuerySet [<Estudiante: Estudiante object (1)>]>
```

```
queryset = Estudiante.objects.all()  
print(queryset.query)
```



?

**No te olvides de completar la
asistencia y consultar dudas**

Recordá:

- **Revisar la Cartelera de Novedades.**
- **Hacer tus consultas en el Foro.**

TODO EN EL AULA VIRTUAL