

A graphic on the left side of the slide. It features four overlapping horizontal bars in purple, 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 right-pointing arrow shape at its end.

Agencia de  
Aprendizaje  
a lo largo  
de la vida

# DJANGO

## Reunión 24

Models - 4

# Les damos la bienvenida

Vamos a comenzar a grabar la clase

## Clase 23

### Django: Models - 3

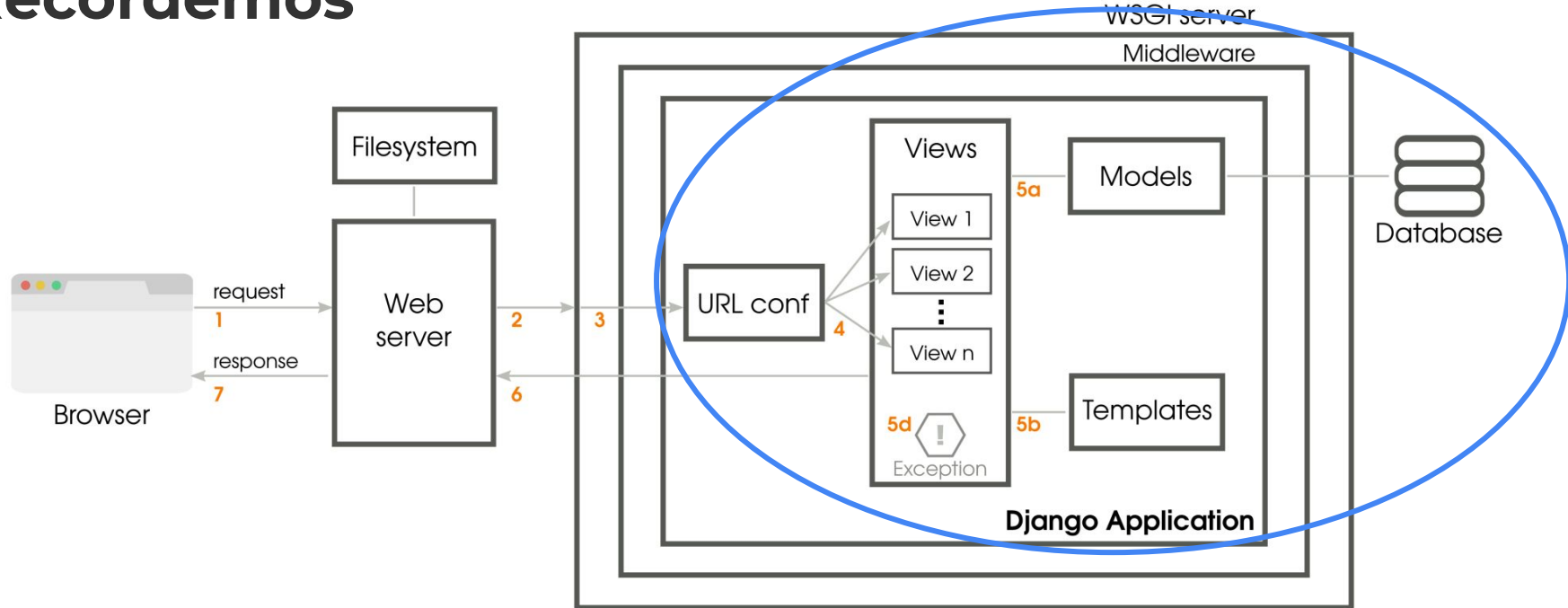
- Crud sobre modelos
- Integración de modelos con vistas y templates
- Vistas basadas en clases

## Clase 24

### Django: Models - 4

- Asociar un modelo a un formulario
- Seleccionando campos a utilizar
- Herencia de formularios


# Recordemos



# ModelForm

forms.py

```
class EstudianteForm(forms.ModelForm):  
    class Meta:  
        model = Estudiante  
        fields = '__all__'
```



Genera un formulario con cada uno de los campos del modelo (sin el id)

# ModelForm

```
class EstudianteForm(forms.ModelForm):  
    class Meta:  
        model = Estudiante  
        fields = ['nombre', 'apellido', 'dni']
```

Cada campo del modelo es  
convertido a un campo del form



Dos formas de seleccionar campos, una de  
las dos debe ir si o si o la opción `__all__`

```
class EstudianteForm(forms.ModelForm):  
    class Meta:  
        model = Estudiante  
        exclude = ['email']
```

# ModelForm

```
class EstudianteForm(forms.ModelForm):  
    class Meta:  
        model = Estudiante  
        exclude = ['email']  
  
    def clean_apellido(self):  
        apellido = self.cleaned_data['apellido']  
        if apellido.upper() == "MESSI":  
            raise ValidationError("Messi no es estudiante")  
  
        return apellido
```

Todas las validaciones de forms son  
válidas

# Herencia de Formularios

```
class EstudianteForm(forms.ModelForm):  
    class Meta:  
        model = Estudiante  
        fields = '__all__'  
  
class EstudianteFormValidado(EstudianteForm):  
    def clean_apellido(self):  
        apellido = self.cleaned_data['apellido']  
        if apellido.upper() == "MESSI":  
            raise ValidationError("Messi no es estudiante")  
  
        return apellido
```



# Sobreescribiendo formularios

```
class EstudianteForm(forms.ModelForm):  
    nombre = forms.CharField(error_messages={'required': 'PONE EL VALOOR'})  
  
    class Meta:  
        model = Estudiante  
        fields = '__all__'  
        widgets = {  
            'apellido': forms.Textarea(attrs={'cols': 20, 'rows': 20}),  
        }  
        error_messages = {  
            'email': {  
                'required': 'Y EL VALOR QUE ONDA?',  
            },  
        }  
    }
```

También depende de como se use en el Template

**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**