



Agencia de
Aprendizaje
a lo largo
de la vida

DJANGO

Reunión 17

Django: Formularios - 2

Les damos la bienvenida

Vamos a comenzar a grabar la clase

Reunión 17

Django: Forms - 2

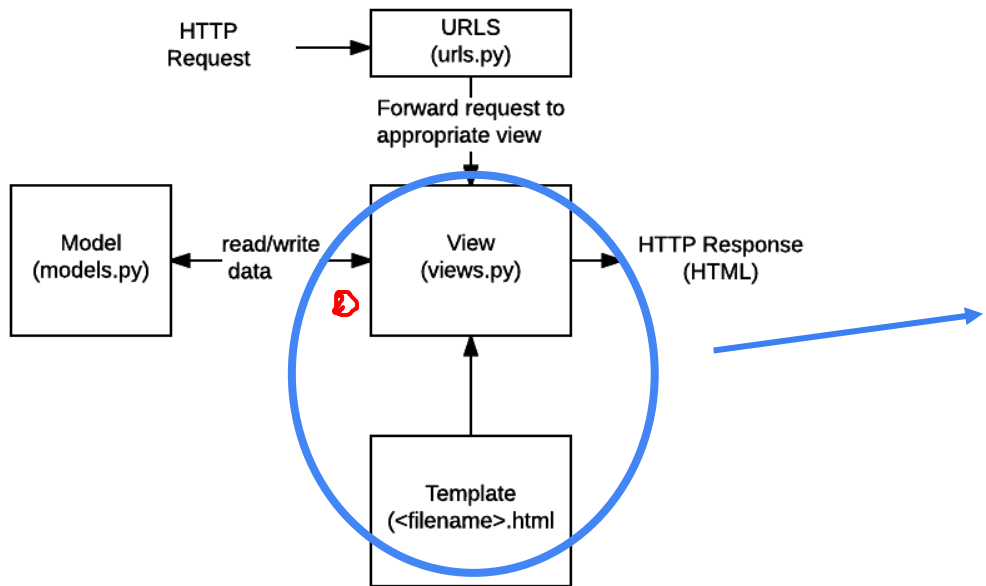
- Widgets
- Validaciones en formularios (is_valid, cleaned_data)
- Errores en formularios
- Mensajes

Reunión 18

Django: DB - 1

- Repaso general (relacionales, no relacionales, dbms, db, etc)
- Del modelo de clases al modelo de datos (der)
- PostgreSQL y PgAdmin
- Repaso DDL, DML.
- Repaso claves y relaciones.

Recordemos



Los formularios son clases que tienen su propia representación pero se pueden sobrescribir.

¿Qué es un Widget?

Un widget es la representación de Django de un elemento HTML de entrada. El widget maneja la representación del HTML y la extracción de datos de un diccionario GET/POST que corresponde al widget.

WIDGET != CAMPO FORMULARIO

Widgets por defecto

django.forms.widgets

NumberInput

class NumberInput

- `input_type: 'number'`
- `template_name: 'django/forms/widgets/number.html'`
- Renders as: `<input type="number" ...>`

TextInput

class TextInput

- `input_type: 'text'`
- `template_name: 'django/forms/widgets/text.html'`
- Renders as: `<input type="text" ...>`

Clase abstracta Widget

Proporciona
atributo **attrs**



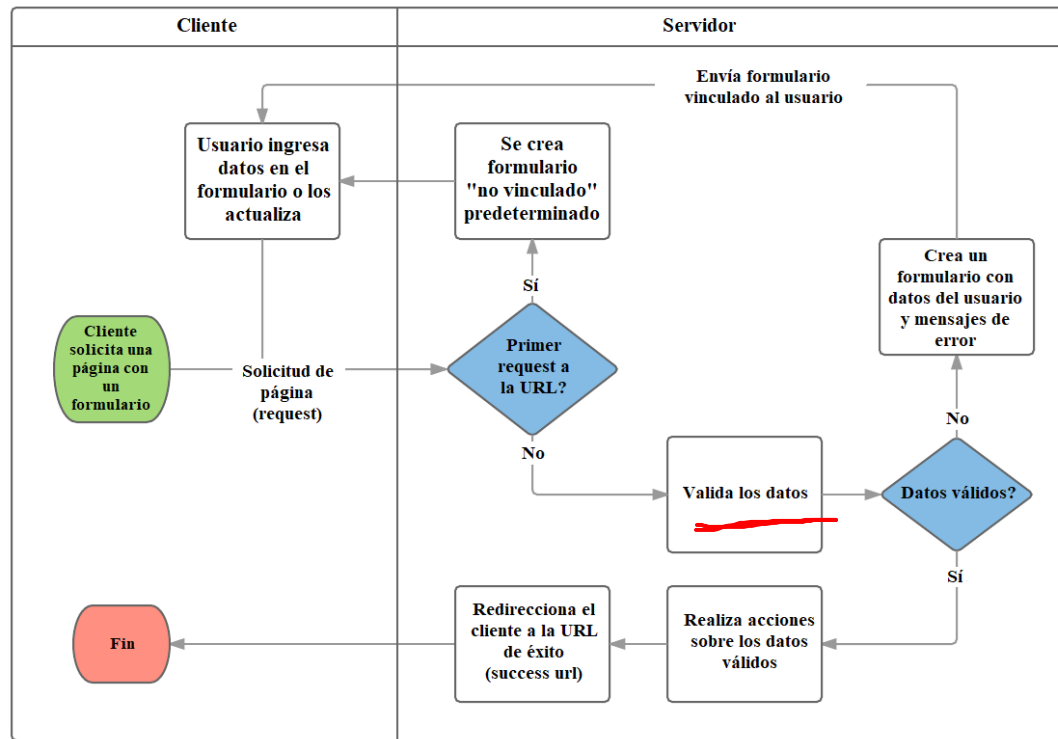
```
nombre = forms.CharField(widget=forms.TextInput(attrs={'class': 'clase_css'}))
```

Clase interna Media

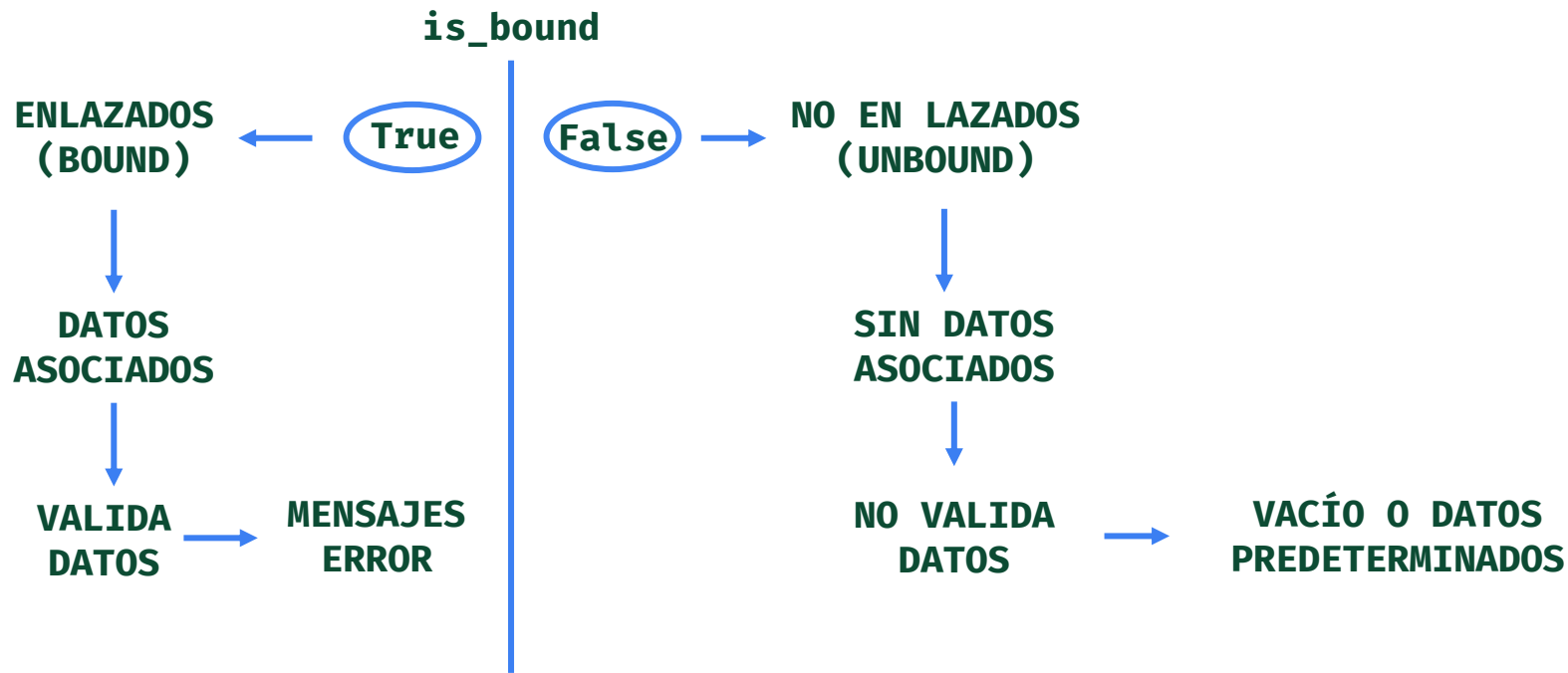
```
from django import forms

class CalendarWidget(forms.TextInput):
    class Media:
        CSS = { 'all': ('pretty.css',) }
        JS = ('animations.js', 'actions.js')
```

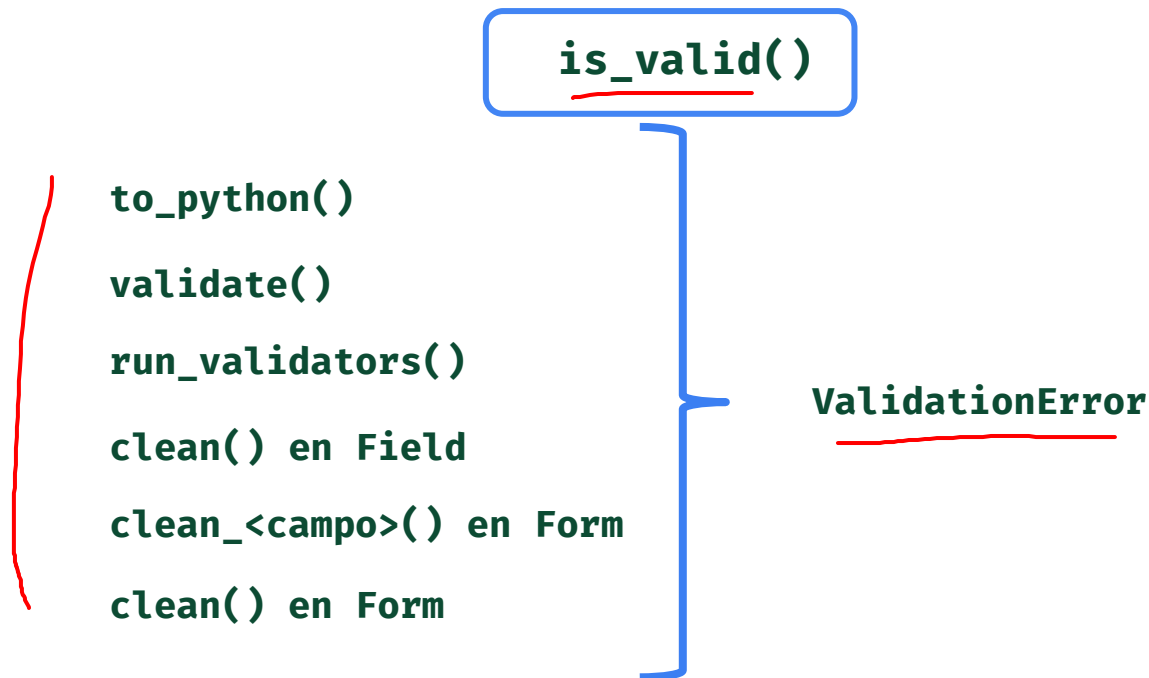

Recordemos el ciclo de los formularios



Validaciones en Formularios



Validaciones en Formularios



Mensajes en Django

INSTALLED_APPS → django.contrib.messages

MIDDLEWARE → 'django.contrib.sessions.middleware.SessionMiddleware'
'django.contrib.messages.middleware.MessageMiddleware'

'context_processors' → 'django.contrib.messages.context_processors.messages'.

DEBUG	debug
INFO	info
SUCCESS	success
WARNING	warning
ERROR	error

from django.contrib import messages
messages.add_message(request, messages.INFO, 'Hola Mundo.')
messages.info(request, 'Hola Mundo.')

Mensajes en Django

```
{% if messages %}  
<ul class="messages">  
  {% for message in messages %}  
    <li{% if message.tags %} class="{{ message.tags }}"{% endif %}>{{ message }}</li>  
  {% endfor %}  
</ul>  
{% endif %}
```

**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