

Agencia de
Aprendizaje
a lo largo
de la vida

DJANGO

Reunión 23

Models - 3

Les damos la bienvenida

Vamos a comenzar a grabar la clase

Clase 23

Django: Models - 3

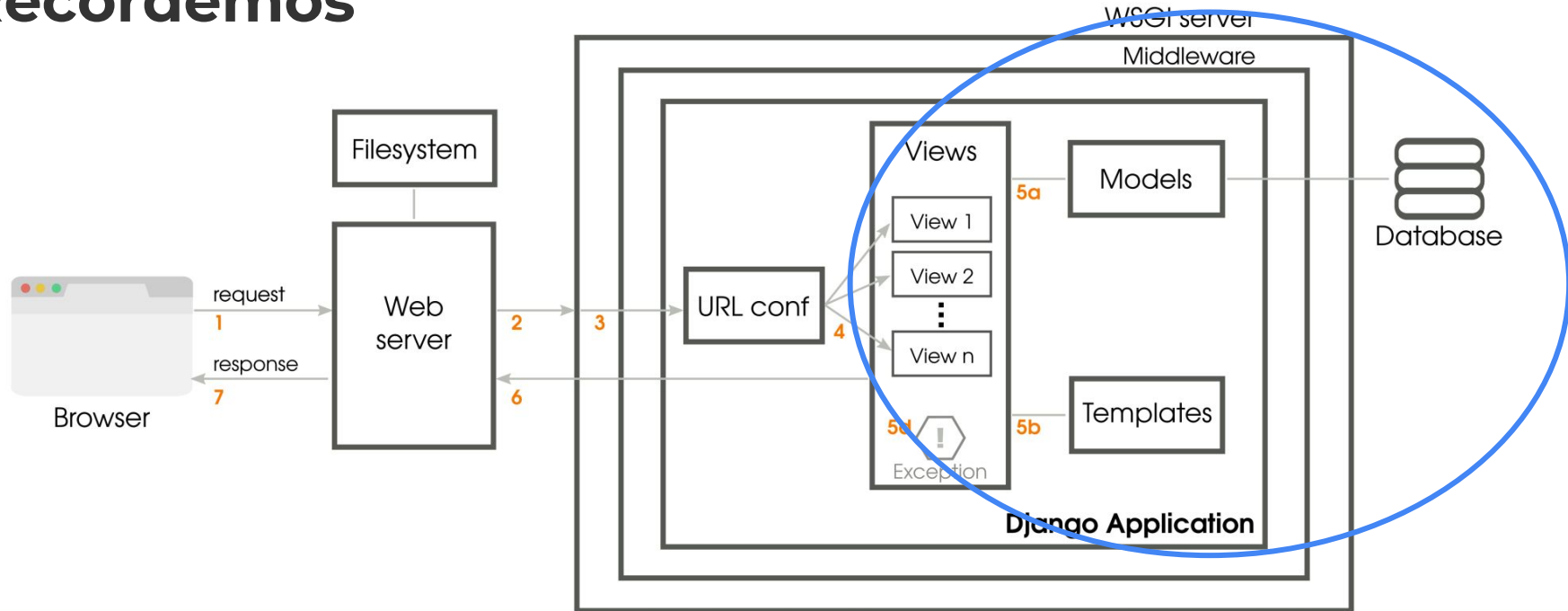
- Crud sobre modelos
- Integración de modelos con vistas y templates
- Vistas basadas en clases

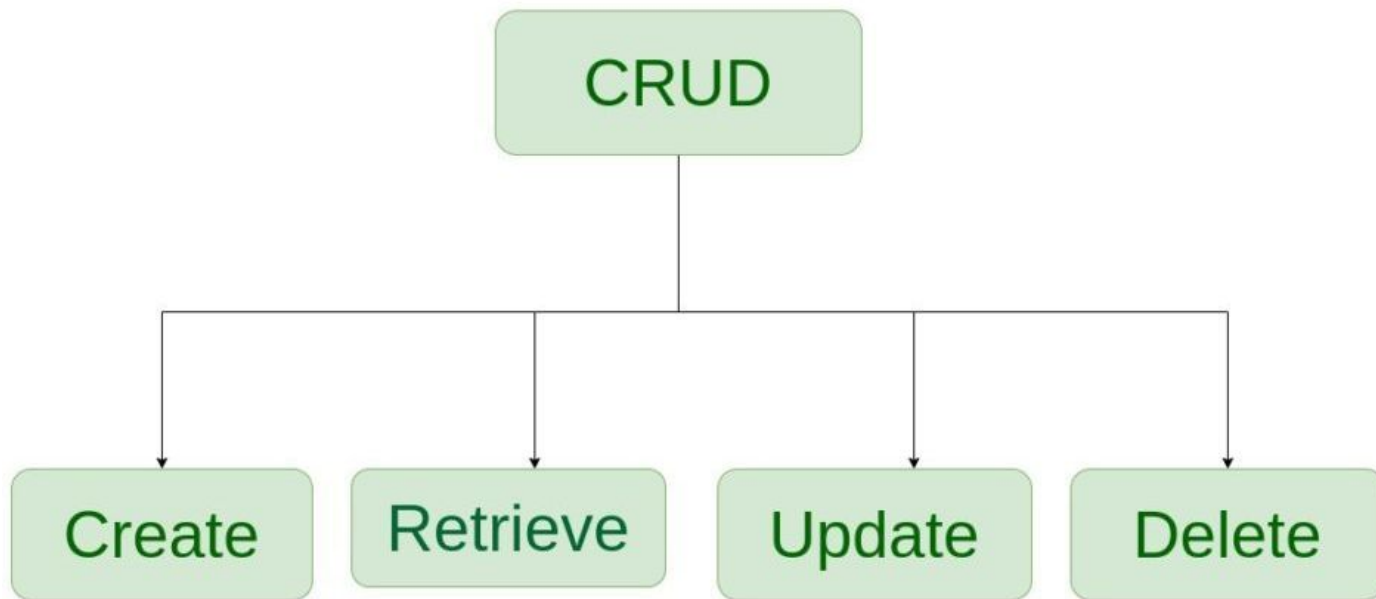
Clase 24

Django: Models - 4

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

Recordemos





Retrieve

urls.py

```
path('administracion/estudiantes', views.estudiantes_index, name='estudiantes_index'),
```

views.py

```
def estudiantes_index(request):  
    estudiantes = Estudiante.objects.all().order_by('dni')  
    return render(request, 'cac/administracion/estudiantes/index.html', {'estudiantes': estudiantes})
```

index.html

```
{% for estudiante in estudiantes %}  
  <tr>  
    <td>{{estudiante.id}}</td>  
    <td>{{estudiante.nombre}}</td>  
    <td>{{estudiante.apellido}}</td>  
    <td>{{estudiante.dni}}</td>  
    <td>{{estudiante.email}}</td>  
  </tr>  
{% endfor %}
```

Create

urls.py

```
path('administracion/estudiantes/nuevo/', views.estudiantes_nuevo, name='estudiantes_nuevo'),
```

views.py

```
def estudiantes_nuevo(request):  
    if request.method == "POST":  
        formulario = EstudianteForm(request.POST)  
        if formulario.is_valid():  
            nombre = formulario.cleaned_data['nombre']  
            apellido = formulario.cleaned_data['apellido']  
            dni = formulario.cleaned_data['dni']  
            email = formulario.cleaned_data['email']  
            nuevo_estudiante = Estudiante(nombre=nombre, apellido=apellido, email=email, dni=dni)  
            nuevo_estudiante.save()  
            return redirect('estudiantes_index')  
        else:  
            formulario = EstudianteForm()  
    return render(request, 'cac/administracion/estudiantes/nuevo.html', {'formulario': formulario})
```

Create

nuevo.html

```
<div class="card">
  <div class="card-header">Creacion de estudiante</div>
  <div class="card-body">
    <form action="{% url 'estudiantes_nuevo' %}" method="post" enctype="multipart/form-data">
      {% csrf_token %}
      {% include 'cac/administracion/form_campos.html' %}
    </form>
  </div>
</div>
```


Sobreescribiendo modelo

models.py

```
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.BigIntegerField(verbose_name='DNI')

    def __str__(self):
        return f"DNI: {self.dni} - {self.apellido}, {self.nombre}"

    def save(self, *args, **kwargs):
        if self.apellido.upper() == 'MESSI':
            raise ValueError("Messi es un maestro, no puede ser estudiante")
        else:
            super().save(*args, **kwargs)

    def delete(self, *args, **kwargs):
        if self.apellido.upper() == 'ALMADA':
            raise ValueError("No se puede eliminar, tiene mucho que aprender")
        return super().delete(*args, **kwargs)
```

**Ejemplo completo en proyecto
integrador grupal**

Opcionales (ver documentación)

¿Qué son las Vistas Basadas en Clases?

Las vistas basadas en clases proporcionan una forma alternativa de implementar vistas como objetos de Python en lugar de funciones. No reemplazan las vistas basadas en funciones, pero tienen ciertas diferencias y ventajas en comparación con las vistas basadas en funciones:

- La organización del código relacionado con métodos HTTP específicos (GET, POST, etc.) puede abordarse mediante métodos separados en lugar de bifurcaciones condicionales.
- Las técnicas orientadas a objetos, como mixins (herencia múltiple), se pueden usar para factorizar el código en componentes reutilizables.

Vistas basadas en clase (CBV)

Forma tradicional Django

url.py

```
path('administracion/estudiantes', views.estudiantes_index, name='estudiantes_index'),
```

views.py

```
def estudiantes_index(request):  
    estudiantes = Estudiante.objects.all().order_by('dni')  
    return render(request, 'cac/administracion/estudiantes/index.html', {'estudiantes':  
    estudiantes})
```

Vistas basadas en clase (CBV)

Forma tradicional Django

url.py

```
path('administracion/estudiantes/nuevo/', views.EstudiantesView.as_view(),  
     name='estudiantes_nuevo'),
```

Otras vistas en proyecto
integrador grupal

ListView

```
class EstudiantesListView(ListView):  
    model = Estudiante  
    context_object_name = 'estudiantes'  
    template_name = 'cac/administracion/estudiantes/index.html'  
    ordering = ['dni']
```

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