<codoacodo/>

DJANGO
Reunión 40

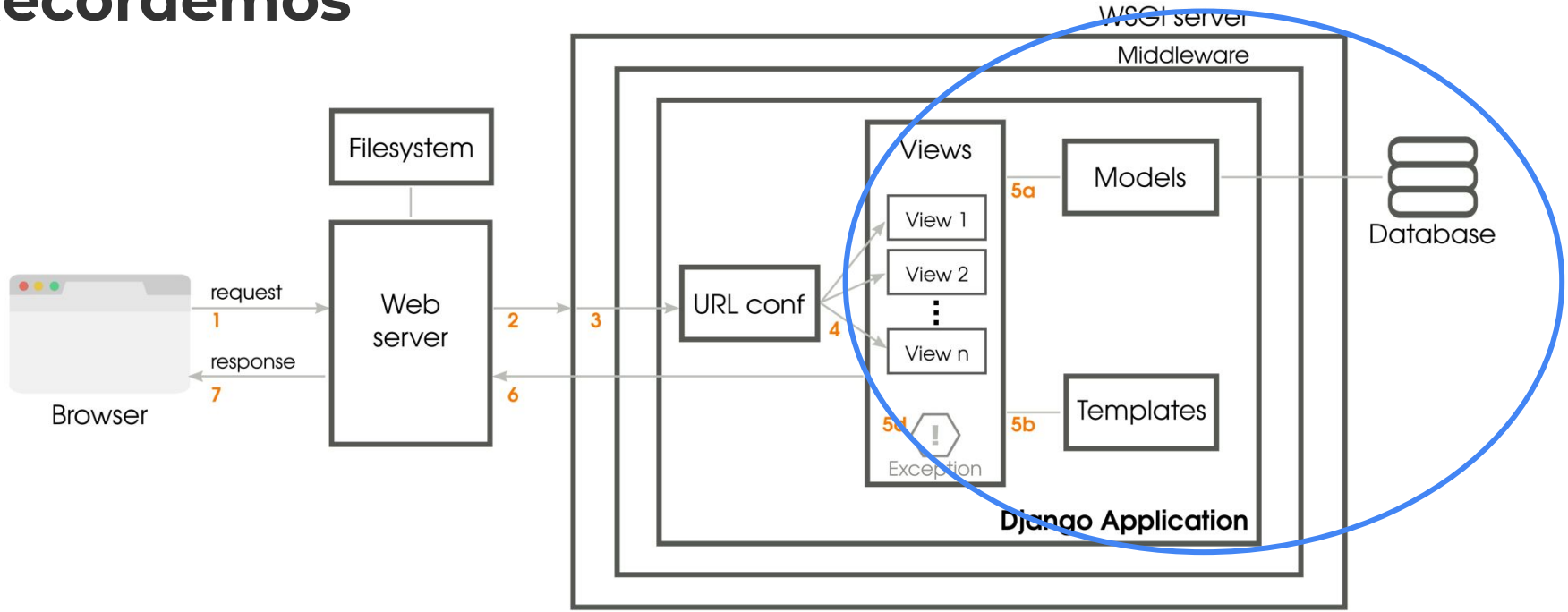
Models - 2

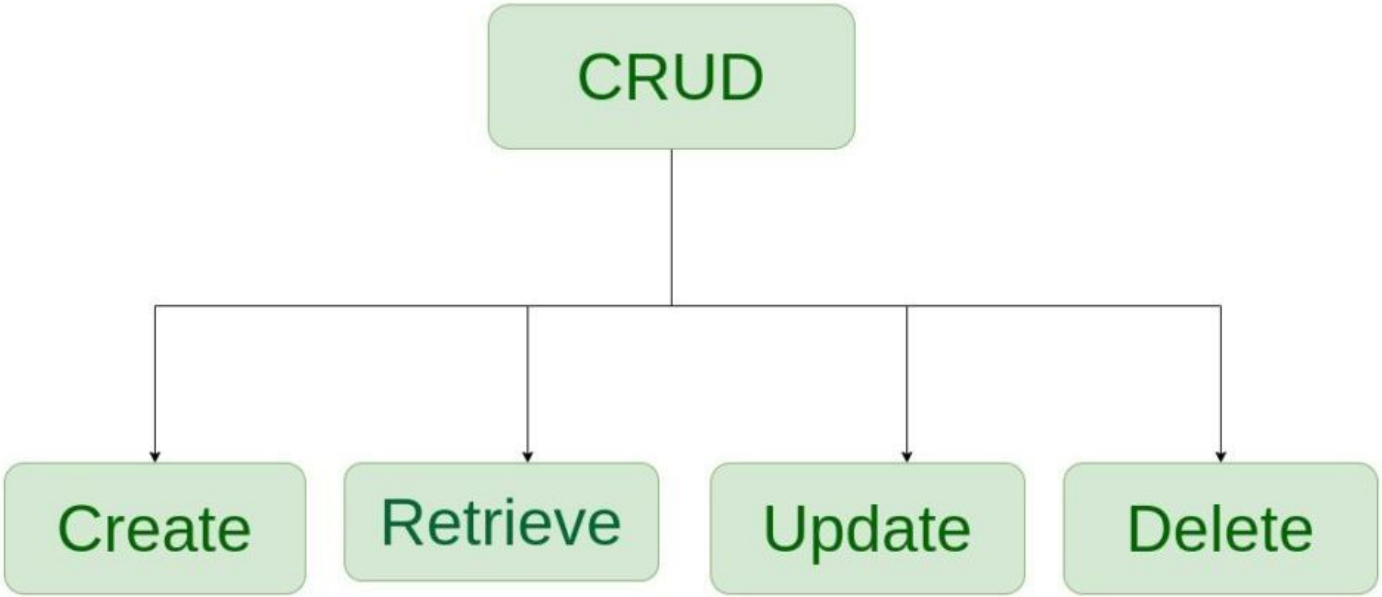
BA

Les damos la bienvenida

Vamos a comenzar a grabar la clase

Recordemos


WSGI server
Middleware
Filesystem
Views
Models
Database
View 1
View 2
View n
5a
request
1
response
7
Web
server
2
3
URL conf
4
6
5b
Templates
Exception
Browser
Django Application


CRUD
Create
Retrieve
Update
Delete

¿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.

Agencia de Aprendizaje a lo largo de la vida

Buenos Aires Ciudad
Educación

Vistas basadas en clase (CBV)

ListView

views.py

```
class CategoriaListView(ListView):
    model = Categoria
    context_object_name = 'listado_categorias'
    template_name = 'core/categorias_listado.html'
```

urls.py

```
path('categorias/listado', views.CategoriaListView.as_view(), name="categorias_listado"),
```

categorias_listado.html

```
    <ul>
        {% for categoria in listado_categorias %}
            <li>
                <p>{{categoria.nombre }}</p>
            </li>
        {% empty %}
            <p>Sin Categorías registradas</p>
        {% endfor %}
    </ul>
```

Vistas basadas en clase (CBV)

CreateView

views.py

```
class CategoriaCreateView(CreateView):
    model = Categoria
    template_name = 'core/alta_categoria.html'
    success_url = 'listado'
    fields = '__all__'
```

urls.py

```
path('categorias/alta', views.CategoriaCreateView.as_view(), name="alta_categoria"),
```

alta_categoria.html

```
    <h3>Datos de la nueva Categoria</h3>
    <form method="post" action="{% url 'alta_categoria' %}">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit">
    </form>
```

Vistas basadas en clase (CBV)

DetailView

views.py

```
class CategoriaDetailView(DetailView):
    model = Categoria
    template_name = 'core/categoria_detalle.html'
```

urls.py

```
path("categorias/detalle/<pk>/", views.CategoriaDetailView.as_view(), name="categoria_detalle"),
```

alta_categoria.html

```
    <h2>Categoria</h2>
    <p>{{object.nombre}}</p>
```

Vistas basadas en clase (CBV)

UpdateView

views.py

```
class CategoriaUpdateView(UpdateView):
    model = Categoria
    template_name = 'core/categoria_actualizar.html'
    fields = '__all__'

    def get_success_url(self):
        return reverse('categoria_detalle', kwargs={'pk': self.object.pk})
```

urls.py

```
path("categorias/actualizar/<pk>/", views.CategoriaUpdateView.as_view(), name="categoria_actualizar"),
```

alta_categoria.html

```
    <h3>Datos de la nueva Categoria</h3>
    <form method="post" action="{% url 'alta_categoria' %}">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit">
    </form>
```

Vistas basadas en clase (CBV)

DeleteView

views.py

```python
class CategoriaDeleteView(DeleteView):
    model = Categoria
    template_name = 'core/categoria_borrar.html'
    success_url = reverse_lazy('categorias_listado')
```

urls.py

```python
path("categorias/borrar/<pk>/", views.CategoriaDeleteView.as_view(), name="categoria_borrar"),
```

alta_categoria.html

```
    <h2>Borrar Categoria</h2>
    <form method="post">
        {% csrf_token %}
        <p>Estás seguro de querer borrar: "{{ object }}"?</p>
        {{ form }}
        <input type="submit" value="Confirm">
    </form>
```

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