

Contenido

Services - CRUD Personalizado.....2

 CreateView2

 UpdateView.....5

 DeleteView ➔ Práctica6

Services - CRUD Personalizado

CreateView

1. Crear el archivo forms.py (ModelForm)

```
1 from django.forms import ModelForm
2 from services.models import Service
3
4 class ServiceForm(ModelForm):
5     class Meta:
6         model = Service
7         fields = ['title', 'subtitle', 'content', 'image']
```

2. En el archivo views.py

```
views.py M X
DiplomadoDAW > Modulo5 > Django > solutions > webrestaurante > services > views.py
1 from django.shortcuts import render, HttpResponseRedirect
2 from .models import Service
3 from .forms import ServiceForm
4
5 def service_list(request):
6     services = Service.objects.all()
7     return render(request, 'services/service_list.html', {'services':services})
8
9 def create(request):
10     if request.method == 'POST':
11         form = ServiceForm(request.POST, request.FILES)
12         if form.is_valid():
13             new_service = form.save()
14             return HttpResponseRedirect('/')
15     else:
16         form = ServiceForm()
17     return render(request, 'services/service_form.html', {'form':form})
```

3. En el archivo urls.py, agregar el menú correspondiente.

```
urls.py M X
DiplomadoDAW > Modulo5 > Django > solutions > webrestaurante > services > urls.py
1 from django.urls import path
2 from services import views
3
4 services_urlpatterns = ([
5     path('', views.service_list, name='service_list'),
6     path('create/', views.create, name='create'),
7 ], 'services')
8
```

4. Crear el archivo service_form.html

```

<> service_form.html U X
DiplomadoDAW > Modulo5 > Django > solutions > webrestaurante > services > templates > services > <> serv
1  {% extends "core/base.html" %}
2  {% block content %}
3  {% load static %}
4  {% include 'services/menu_services.html' %}
5
6  <div class="container">
7      <div class="row mt-3">
8          <div class="col-md-9 mx-auto text-light">
9              <div>
10                 <form action="" method='post' enctype='multipart/form-data'>
11                     {% csrf_token %}
12                     <table>
13                         | {{form.as_table}}
14                     </table>
15                     <br>
16                     <input type="submit" value='Crear servicio' />
17                 </form>
18             </div>
19         </div>
20     </div>
21 </div>
22
23 {% endblock %}

```

5. Crear el archivo menu_services.html

```

2  {% if request.user.is_staff %}
3  <nav class="navbar navbar-expand-lg navbar-dark py-lg-4 border border-dark" id="SecondNav">
4      <div class="container">
5          <span class="navbar-brand"><i>Servicios</i></span>
6          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#nep"
7              aria-controls="services_menu" aria-expanded="false" aria-label="Toggle navigation">
8              <span class="navbar-toggler-icon"></span>
9          </button>
10         <div class="collapse navbar-collapse" id="nep">
11             <ul class="navbar-nav mx-auto">
12                 <li class="nav-item">
13                     <a class="nav-link" href="{% url 'services:create' %}">Crear Nuevo Servicio</a>
14                 </li>
15             </ul>
16         </div>
17     </div>
18 </nav>
19 {% endif %}

```

a. Agregar el estilo en business-casual.css

```

86  #SecondNav {
87      | background-color: rgba(47, 23, 15, 0.5);
88  }

```

b.

6. En el archivo service_list.html

```

<> service_list.html M X
DiplomadoDAW > Modulo5 > Django > solutions > webrestaurante > serv
1  {% extends "core/base.html" %}
2  {% block content %}
3  {% load static %}
4  {% include 'services/menu_services.html' %}
5  {% for service in services %}
6      <section class="page-section">
7          <div class="container">

```

a.

7. Verificar su funcionamiento.

Práctica.

1. Modificar el modelo

You, 2 hours ago | 1 author (You)

```
class Service(models.Model):
    title = models.CharField(max_length=100, verbose_name="Título")
    subtitle = models.CharField(max_length=100, verbose_name="Subtítulo")
    content = CKEditor5Field(
        'Contenido', config_name='extends', blank=True, null=True)
    image = models.ImageField(verbose_name='Imágen', upload_to='services')
```

2. Modificar el formulario para agregar widgets y para agregar validaciones.

```
forms.py 1,0 X
ode > webrestaurante > services > forms.py > ...
1 from django.forms import ModelForm, TextInput, Textarea
2 from services.models import Service
3 from django_ckeditor_5.widgets import CKEditor5Widget
4
5
6 class ServiceForm(ModelForm):
7     class Meta:
8         model = Service
9         fields = ['title', 'subtitle', 'content', 'image']
10        widgets = {
11            'title': TextInput(attrs={'class': 'form-control', 'placeholder': 'Título'}),
12            'sub_title': TextInput(attrs={'class': 'form-control', 'placeholder': 'Título'}),
13            'content': CKEditor5Widget(config_name='extends'),
14        }
15
16    def clean(self):
17        super(ServiceForm, self).clean()
18        title = self.cleaned_data.get('title')
19        if len(title) < 5:
20            self.errors['title'] = self.error_class(['Mínimo 5 caracteres'])
21        return self.cleaned_data
```

3. Modificar el html que contiene el formulario

```
1 {% extends "core/base.html" %}
2 {% block content %}
3 {% load static %}
4 {% include 'services/menu_services.html' %}
5 <div class="container">
6     <div class="row mt-3">
7         <div class="col-md-9 mx-auto text-light">
8             <div>
9                 <form action="" method='post' enctype='multipart/form-data' >
10                     {% csrf_token %}
11                     <table>
12                         | {{form.as_table}}
13                     </table>
14                     <br>
15                     {{form.media}}
16                     <input type="submit" value='Crear servicio'>
17                 </form>
18             </div>
19         </div>
20     </div>
21 </div>
22 {% endblock %}
```

4. Modificar el archivo views.py

```

views.py M X
DiplomadoDAW > Modulo5 > Django > solutions > webrestaurante > services > views.py
1 from django.shortcuts import render, HttpResponseRedirect
2 from .models import Service
3 from .forms import ServiceForm
4
5 def service_list(request):
6     services = Service.objects.all()
7     return render(request, 'services/service_list.html', {'services':services})
8
9 def create(request):
10     if request.method == 'POST':
11         form = ServiceForm(request.POST, request.FILES)
12         if form.is_valid():
13             new_service = form.save()
14             return HttpResponseRedirect('/')
15         else:
16             return render(request, 'services/service_form.html', {'form':form})
17     else:
18         form = ServiceForm()
19         return render(request, 'services/service_form.html', {'form':form})

```

UpdateView

1. Crear la el método update

```

24 def update(request, service_id):
25     service = Service.objects.get(id=service_id)
26     if request.method == 'POST':
27         form = ServiceForm(request.POST, request.FILES, instance=service)
28         if form.is_valid():
29             form.save()
30             return HttpResponseRedirect('/services')
31     else:
32         form = ServiceForm(instance=service)
33     return render(request, 'services/service_update_form.html', {'form':form})

```

2. Modificar el archivo urls.py

```

6 path('create/', views.create , name='create'),
7 path('update/<int:service_id>', views.update , name='update'),
8 ], 'services')

```

3. Ya que el código es muy similar al código del archivo service_form.html, crear el archivo "service_formulario.html" con el contenido del formulario, de la siguiente manera.

```

<> service_formulario.html U X
DiplomadoDAW > Modulo5 > Django > solutions > webrestaurante > services > templates > services > <> service_formulario.html
1 <div class="container">
2   <div class="row mt-3">
3     <div class="col-md-9 mx-auto text-light">
4       <div>
5         <form action="" method='post' enctype='multipart/form-data'>
6           {% csrf_token %}
7           <table>
8             {{form.as_table}}
9           </table>
10          <br>
11          {{form.media}}
12          <input type="submit" value={{btn_description}} />
13        </form>
14      </div>
15    </div>
16  </div>
17 </div>

```

4. Modificar el archivo "service_form.html"

```

<> service_form.html U X
DiplomadoDAW > Modulo5 > Django > solutions > webrestaurante > services > templates > services > <> service_form.html
1 {% extends "core/base.html" %}
2 {% load static %}
3 {% include 'services/menu_services.html' %}
4 {% block content %}
5   {% include 'services/service_formulario.html' with btn_description="'Crear servicio'" %}
6 {% endblock %}
7

```

5. Crear el archivo service_update_form.html

```

<> service_update_form.html U X
DiplomadoDAW > Modulo5 > Django > solutions > webrestaurante > services > templates > services > <> service_update_form.html
1 {% extends "core/base.html" %}
2 {% load static %}
3 {% include 'services/menu_services.html' %}
4 {% block content %}
5   {% include 'services/service_formulario.html' with btn_description="'Actualizar servicio'" %}
6 {% endblock %}

```

6. Modificar el archivo service_list.html

```

20 <div class="bg-faded p-5 rounded">
21   {{service.content|safe}}
22   <br>
23   {% if request.user.is_staff %}
24     <a href="{% url 'services:update' service.id %}">Ver detalle</a>
25   {% endif %}
26 </div>

```

7. Verificar su funcionamiento

DeleteView ➔ [Práctica](#)

Verifique que pasa si accede de manera directa a la página de creación o de actualización sin estar logeado como administrador

Es necesario:

```
3 from .forms import ServiceForm
4 from django.contrib.admin.views.decorators import staff_member_required
5
6 @staff_member_required()
7 def update(request, service_id):
8     service = Service.objects.get(id=service_id)
9     if request.method == 'POST':
```