

INSTITUTO TECNOLÓGICO SUPERIOR DE LERDO



Alumno: Miguel Ángel Carrillo Ortíz

Maestro: Ing. Jesús Salas Marín

Materia: Desarrollo de Aplicaciones Web

Carrera: Ingeniería en Informática

Semestre: "VI"

N° de Control: 182310823

Nombre de la actividad: Reporte de App Completa: Web Personal

Lugar: Gómez Palacio, Durango

Fecha: 20/Junio/2021

Template Tag_URL

En este paso, utilice dicho template para sustituir los enlaces, escritos en crudos, por enlaces que se crearan dinámicamente, para detectar valores.

```
<li><a href="{% url 'home' %}">Portada</a></li>
<li><a href="{% url 'about' %}">Acerca de</a></li>
<li><a href="{% url 'portfolio' %}">Portafolio</a></li>
<li><a href="{% url 'contact' %}">Contacto</a></li>
```

Lo verificamos



Ahora uniremos el front end con el Back-end, utilizaremos, el front-end, a la hora de fusionarlo hay que identificar los valores comunes, para trabajar con ellos. Utilizaremos el archivo index que nos proporciona el curso y lo editaremos con los templates adecuados.

```
<div class="collapse navbar-collapse" id="navbarResponsive">
  <ul class="navbar-nav ml-auto">
    <li class="nav-item">
      <a class="nav-link" href="{% url 'home' %}">Portada</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="{% url 'about' %}">Acerca de</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="{% url 'portfolio' %}">Portafolio</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="{% url 'contact' %}">Contacto</a>
    </li>
  </ul>
```

Agregamos un bloque de content en el apartado de contenido

```
<!-- Contenido -->
{% block content %}{% endblock %}
```

Y vamos a sustituir el texto al archivo original base.html, a la hora de abrir la página veremos un cambio

La página se ve así vacía por la falta de los ficheros estáticos

[MIKE CARRILLO](#) [MENÚ](#)

- [PORTADA](#)
- [ACERCA DE](#)
- [PORTAFOLIO](#)
- [CONTACTO](#)

ACERCA DE

BIOGRAFÍA

MIKE CARRILLO

Nacido en... lorem ipsum dolor sit amet, consectetur adipiscing elit. Saepe nostrum ullam eveniet pariatur voluptates odit, fuga anque ea nobis sit soluta odio.

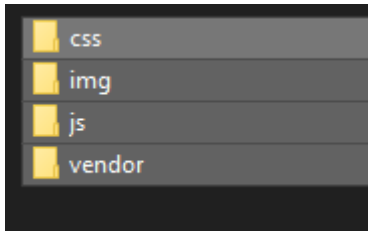
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Saepe nostrum ullam eveniet pariatur voluptates odit...

-
- -
 -

COPYRIGHT © 2021 - MIKE CARRILLO (INGENIERO INFORMÁTICO)

Para darle forma, crearemos un nuevo directorio static/core, en dicha página vamos a pegar las cuatro carpetas que tenemos en nuestros archivos de la práctica.

En la parte de la cabecera, tenemos un título, subtítulo y una imagen.



Para cargar un fichero estático, se usa el template {% load static %} asignando correctamente la ruta del fichero

```
<!-- Estilos y fuentes del template -->
{% load static %}
<link href="{% static 'core/vendor/bootstrap/css/bootstrap.min.css' %}" rel="stylesheet">
<link href="{% static 'core/vendor/font-awesome/css/font-awesome.min.css' %}" rel="stylesheet" type="text/css">
<link href="https://fonts.googleapis.com/css?family=Lora:400,700,400italic,700italic" rel="stylesheet" type="text/css">
<link href="https://fonts.googleapis.com/css?family=Open+Sans:300italic,400italic,600italic,700italic,800italic,400,300,600,700" rel="stylesheet" type="text/css">
<link href="{% static 'core/css/clean-blog.min.css' %}" rel="stylesheet">
```

Al igual las imágenes

```
<!-- Cabecera -->
<header class="masthead" style="background-image: url('{% static 'core/img/home-bg.jpg' %}')">
  <div class="overlay"></div>
  <div class="container">
```

Y Bootstrap y los Javascripts

```
<!-- Bootstrap y Javascripts -->
<script src="{% static 'core/vendor/jquery/jquery.min.js' %}"></script>
<script src="{% static 'core/vendor/bootstrap/js/bootstrap.bundle.min.js' %}"></script>
<script src="{% static 'core/js/clean-blog.min.js' %}"></script>
```

Y al reiniciar el servidor quedará así nuestra página web



Aquí Creamos la Cabecera dinámica



En todos los sitios tenemos la misma cabecera, lo que vamos hacer es cambiar la cabecera dependiendo del apartado de la página que entremos

Utilizaremos dos bloques, uno para la imagen y otro para el título y subtítulo

Para el fondo será el block background

```
<!-- Cabecera -->
<header class="masthead" style="background-image: url('{{% block background %}}{% endblock %}}')">
  <div class="overlay"></div>
  <div class="container">
    <div class="row">
      <div class="col-lg-8 col-md-10 mx-auto">
        <div class="site-heading">
```

Y este el bloque de headers

```
{{% block headers %}}{% endblock %}}
</div>
div>
```

Así quedarían asignados nuestros templates

```
{ % block background %}
  {% load static %}
  {% static 'core/img/home-bg.jpg' %}
{ % endblock %}

{% block headers %}
<h1>MIKE CARRILLO</h1>
<span class="subheading">INGENIERO INFORMÁTICO</span>
{% endblock %}
```

Un salto de línea que está creando un problema de sintaxis que se arregla, juntando todo en una sola línea así:

```
{% block background %}{% load static %}{% static 'core/img/home-bg.jpg' %}{% endblock %}}
```

Veremos que al actualizar la página web, nuestro front end estará fusionado con el back-end.



Ahora fusionaremos completamente el demás back-end para que las cabeceras coincidan con la página.

```
<!-- Contenido Principal -->
<div class="container">
  <div class="row">
    <div class="col-lg-8 col-md-10 mx-auto">
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Saepe nostrum ullam eveniet pariatur voluptates odit, fuga atque ea nobis sit soluta odio, adipisci quas exceptu</p>
      <br>
      <p><b>TELÉFONO:</b> +52 871 128 2616</p>
    </div>
  </div>
</div>
```

Se sustituyen los contenidos principales y se adaptan en nuestro código para poder utilizarlo en el código, quedaría así:

```
<!-- Contenido Principal -->
<div class="container">
  <div class="row">
    <div class="col-lg-3 col-md-4 offset-lg-1">
      
    </div>
    <div class="col-lg-7 col-md-8">
      <h2 class="section-heading">MIKE CARRILLO</h2>
      <p>Nacido en... lorem ipsum dolor sit amet, consectetur adipisicing elit. Saepe nostrum ullam eveniet pariatur voluptates odit, fuga atque ea nobis sit soluta odio.</p>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Saepe nostrum ullam eveniet pariatur voluptates odit...</p>
    </div>
  </div>
</div>
```


Ahora esto mismo se adaptará con los demás códigos archivos para tener la página así (acomodando los temple tags con las imágenes correspondientes).



MIKE CARRILLO

Nacido en... lorem ipsum dolor sit amet, consectetur adipiscing elit. Saepe nostrum ullam eveniet pariatur voluptates odit, fuga atque ea nobis sit soluta odio.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Saepe nostrum ullam eveniet pariatur voluptates odit...



Ahora voy a crear una segunda app llamada **Portfolio** con el comando:

```
>python manage.py startapp portfolio
```

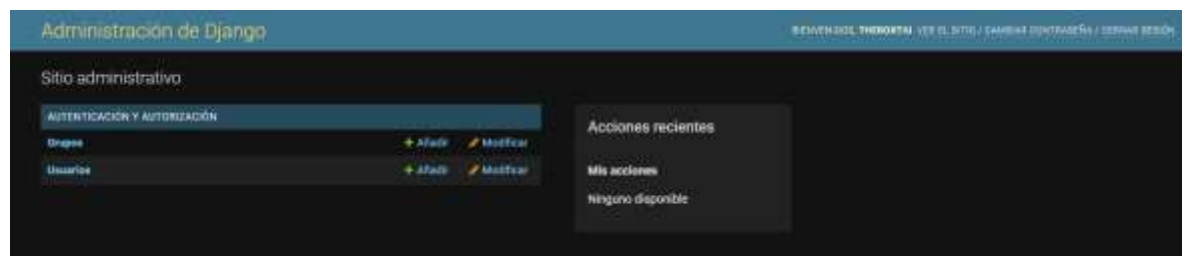
abriremos el archivo que se acaba de crear en la carpeta del mismo nombre que la app edicaremos el archivo **models.py**.

```
models.py x settings.py
1  from django.db import models
2
3  # Create your models here.
4  class Project(models.Model):
5      title = models.CharField(max_length=200)
6      description = models.TextField()
7      image = models.ImageField()
8      created = models.DateTimeField(auto_now_add=True)
9      updated = models.DateTimeField(auto_now=True)
```

Agregamos la app Portfolio al archivo **setting.py**.

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'core',
    'portfolio',
]
```

Ahora probamos entrar al administrador y veremos que nos dejará entrar con la cuenta anteriormente creada.



Ahora para que nuestro modelo de proyecto aparezca en el administrador hay que activarlo en el archivo admin.py agregamos la clase que acabamos de crear y verificamos en el administrador de Django.

```
from django.contrib import admin
from .models import Project

# Register your models here.
admin.site.register(Project)
```



Personalizando el administrador de Django.

Así se agrega un proyecto.

The screenshot shows the 'Añadir project' form. At the top, there's a message: 'Por favor, corrige los siguientes errores.' Below that, the 'Title' field has the value 'Prueba'. The 'Description' field has a placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec vitae efficitur eros, non mollis dui. Sed aliquam diam vel viverra imperdiet. Phasellus tincidunt lorem dui, vitae dictum turpis scelerisque vel. Sed a laoreet diam, eleifend semper nunc. Sed sagittis imperdiet sagittis. Praesent finibus odio vel ipsum dapibus, faucibus accumsan sapien aliquam. Sed ut enim a nisl dictum congue ornare tristique velit. Donec rhoncus sodales efficitur. Suspendisse potenti. Nullam fringilla urna mattis mi imperdiet pretium.' The 'Image' field has a file upload button with the text 'Elegir archivo' and the filename 'Screenshot_7.jpg'. At the bottom, there are three buttons: 'Guardar y añadir otro', 'Guardar y continuar editando', and 'GUARDAR'.


Editamos para que aparezca en plural de la siguiente manera.

```
from django.apps import AppConfig

class PortfolioConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'portfolio'
    verbose_name = 'Portafolios'
```

Y lo agrego en el archivo **settings.py**.

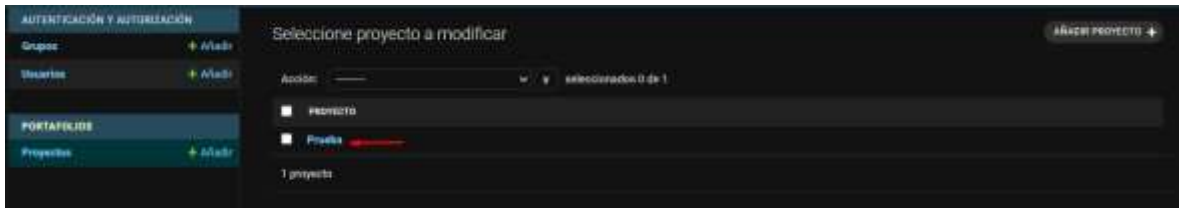
```
28 # SECURITY WARNING: don't run with debug turned on
29 DEBUG = True
30
31 ALLOWED_HOSTS = []
32
33
34 # Application definition
35
36 INSTALLED_APPS = [
37     'django.contrib.admin',
38     'django.contrib.auth',
39     'django.contrib.contenttypes',
40     'django.contrib.sessions',
41     'django.contrib.messages',
42     'django.contrib.staticfiles',
43     'core',
44     'portfolio.apps.PortfolioConfig',
45 ]
```



Al igual sorteamos los proyectos con esta configuración de más nuevos a más viejos.

```
class Meta:
    verbose_name = "proyecto"
    verbose_name_plural = "proyectos"
    ordering = ["-created"]
```

Agregamos un método string, para cambiar el título del proyecto, nos devuelva el nombre del proyecto original.



También para extender los datos de lectura usaremos esta clase en el archivo **admin.py**.

```
# Register your models here.  
class ProjectAdmin(admin.ModelAdmin):  
    readonly_fields = ('created', 'updated')  
  
admin.site.register([Project, ProjectAdmin])
```

Tal y cómo se muestra aquí:

Modificar proyecto

Prueba

Título:

Descripción:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam sodales massa ut ex laoreet accumsan. Nam quam lacus, condimentum non lectus eget, rhoncus vehicula velit. Nunc tempor maximus lacus et malesuada. Sed venenatis tristique vestibulum. Nunc eleifend, dui a elementum placerat, urna nibh efficitur diam, non tristique nulla sem ut velit. Ut in feugiat lorem, eget scelerisque turpis. Morbi posuere ullamcorper posuere. Duis feugiat sollicitudin pulvinar. Maecenas vestibulum sapien elit, ac pellentesque nibh efficitur quis.

Imagen: Actualmente: [Screenshot_7.jpg](#)
Modificar: No se eligió ningún archivo

Ahora hay que crear una carpeta llamada Media en la raíz dado que Django no admite archivos media nativamente, crearemos la carpeta y en el archivo **settings.py** agregaremos estas líneas de código.

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC_URL = '/static/'

import os
#media files
MEDIA_URL = '/media'
MEDIA_ROOT = os.path.join(BASE_DIR, "MEDIA")

# Default primary key field type
# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Django por defecto no puede servir estos ficheros multimedia así que habrá que editar el archivo **models.py**.

```
# Create your models here.
class Project(models.Model):
    title = models.CharField(max_length=200, verbose_name="Título")
    description = models.TextField(verbose_name="Descripción")
    image = models.ImageField(verbose_name="Imagen", upload_to="projects")
    created = models.DateTimeField(auto_now_add=True, verbose_name="Fecha de Creación")
    updated = models.DateTimeField(auto_now=True, verbose_name="Fecha de Edición")
```

En la carpeta de la primera app llamada **core** lo pasaremos a la app llamada **portfolio** que acabamos de crear, también crearemos una nueva carpeta llamada **templates** en la carpeta portfolio y nos llevaremos el template portfolio a dicha carpeta.

```
from django.shortcuts import render
from .models import Project
# Create your views here.
def portfolio(request):
    projects = Project.objects.all()
    return render(request, "portfolio/portfolio.html", {'projects': projects})
```

```
▼ templates\portfolio
  portfolio.html
```

Ahora hay que redactar las URL para que en vez de que busque en core, busque en **Portfolio**

```
from django.contrib import admin
from django.urls import path
from core import views as core_views
from portfolio import views as portfolio_views

from django.conf import settings

urlpatterns = [
    path('', core_views.home, name="home"),
    path('about-me/', core_views.about, name="about"),
    path('portfolio/', portfolio_views.portfolio, name="portfolio"),
    path('contact/', core_views.contact, name="contact"),
    path('admin/', admin.site.urls),
]

if settings.DEBUG:
    from django.conf.urls.static import static
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Ahora vamos a realizar una lista de proyectos con este script.

```
from .models import Project

# Create your views here.
def portfolio(request):
    projects = Project.objects.all()
    return render(request, "portfolio/portfolio.html", {'projects': projects})
```

Ahora este script hará que tengamos contenido dinámico para cada proyecto.

```
{% block headers %}
<h1>Portafolio</h1>
<span class="subheading">Currículo</span>
{% endblock %}

{% block content %}
    {% for project in projects %}
        <div class="row project">
            <div class="col-lg-3 col-md-4 offset-lg-1">
                
            </div>
            <div class="col-lg-7 col-md-8">
                <h2 class="section-heading title">{{project.title}}</h2>
                <p>{{project.description}}</p>
                <!--<p><a href="http://google.com">Más información</a></p>-->
            </div>
        </div>
    {% endfor %}
{% endblock %}
```


Ahora si vemos nuestra página, sólo veremos que falta la imagen.



Prueba

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam sodales massa ut ex laoreet accumsan. Nam quam lacus, condimentum non lectus eget, rhoncus vehicula velit. Nunc tempor maximus lacus et malesuada. Sed venenatis tristique vestibulum. Nunc eleifend, dui a elementum placerat, urna nibh efficitur diam, non tristique nulla sem ut velit. Ut in feugiat lorem, eget scelerisque turpis. Morbi posuere ullamcorper posuere. Duis feugiat sollicitudin pulvinar. Maecenas vestibulum sapien elit, ac pellentesque nibh efficitur quis.

Ahora si agrego otro proyecto en el administrador de Django, podremos ver que también se agregará correctamente.



Flight

Sed pretium nec augue non maximus. Nunc porta sollicitudin tincidunt. Fusce ut ex urna. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris ultricies leo dui, non auctor risus fringilla vel. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Mauris porta facilisis quam, non imperdiet ligula venenatis ut. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Curabitur elementum leo sit amet imperdiet ullamcorper. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec pharetra et justo sit amet porttitor.



Prueba

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam sodales massa ut ex laoreet accumsan. Nam quam lacus, condimentum non lectus eget, rhoncus vehicula velit. Nunc tempor maximus lacus et malesuada. Sed venenatis tristique vestibulum. Nunc eleifend, dui a elementum placerat, urna nibh efficitur diam, non tristique nulla sem ut velit. Ut in feugiat lorem, eget scelerisque turpis. Morbi posuere ullamcorper posuere. Duis feugiat sollicitudin pulvinar. Maecenas vestibulum sapien elit, ac pellentesque nibh efficitur quis.

Ahora modificamos el template para mostrar un enlace más información que llevará a una URL establecida

Para eso usaremos el template if en el **portfolio.html**.

```
Migrations for 'portfolio':
portfolio\migrations\0002_auto_20210527_2219.py
- Change Meta options on project
- Alter field created on project
- Alter field description on project
- Alter field image on project
- Alter field title on project
- Alter field updated on project
Operations to perform:
Apply all migrations: portfolio
Running migrations:
Applying portfolio.0002_auto_20210527_2219... OK
(Web Personal) PS E:\UNIVERSIDAD\Semestre 6\Desarrollo Web\Web Personal\webpersonal> []
```

```
{% block content %}
{% for project in projects %}
    <div class="row project">
        <div class="col-lg-3 col-md-4 offset-lg-1">
            
        </div>
        <div class="col-lg-7 col-md-8">
            <h2 class="section-heading title">{{project.title}}</h2>
            <p>{{project.description}}</p>
            {% if project.link %}
            <p><a href="{{project.link}}">Más información</a></p>
            {% endif %}
        </div>
    </div>
{% endfor %}
```

Conclusión:

En este proyecto aprendí a utilizar las aplicaciones Django y Python los cuales a través del uso de su terminal me permitieron aprender hacer proyectos de desarrollo web, los cuales interactúan de una manera muy efectiva, logrando así que mi web personal fuera muy dinámica, a pesar de ser una página muy básica observe que puede llegar a ser una página tan compleja mientras que lo permita el código, espero llegar a hacer una página más robusta e interactiva.