Contenido

Aplicación “pages”

1. Crear la aplicación “pages”
2. Crear el modelo

models.py

M5_sep2023 > code > webRestaurante > pages > models.py > ...

```python
from django.db import models

class Page(models.Model):
    title       = models.CharField(max_length=200, verbose_name="Títutlo")
    content     = models.TextField(verbose_name="Contenido")
    created     = models.DateTimeField(auto_now_add=True, verbose_name="Fecha de creación")
    updated     = models.DateTimeField(auto_now=True, verbose_name="Fecha de actualización")

    class Meta:
        verbose_name        = "Página"
        verbose_name_plural = "Páginas"
        ordering            = ['title']

    def __str__(self):
        return self.title
```

3. Registrar en el admin.py

admin.py

M5_sep2023 > code > webRestaurante > pages > admin.py > ...

```python
from django.contrib import admin
from .models import Page

class PageAdmin(admin.ModelAdmin):
    readonly_fields = ('created', 'updated',)
    ordering        = ('title',)
    search_fields   = ('title', 'content', )
    list_display    = ('title', 'created')

admin.site.register(Page, PageAdmin)
```

4. Modificar el archivo views.py
    a. Obtener el page_id
    b. Obtener el registro correspondiente
        i. Page.objects.get(id=page_id)
    c. Regresar el objeto page al template correspondiente

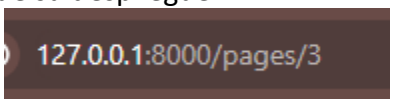
5. Crear el archivo urls.py
    a. .. path(‘<int:page_id>’, view…)

6. Modificar el archivo urls del proyecto

```python
from pages.urls import pages_urlpatterns

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include(core_urlpatterns)),
    path('blog/', include(blog_urlpatterns)),
    path('pages/', include(pages_urlpatterns)),
]
```

7. Crear el archivo page.html (tome como base el archivo sample.html de los archivos estáticos)

```
{% extends "core/base.html" %}
{% load static %}
{% block content %}

<section class="page-section about-heading">
    <div class="container">
      <div class="about-heading-content mbtm">
        <div class="row">
          <div class="col-xl-9 col-lg-10 mx-auto">
            <div class="bg-faded rounded p-5 forced">
              <h2 class="section-heading mb-4">
                <span class="section-heading-lower">{{page.title}}</span>
              </h2>
              <div class="section-content">
                <p>
                    {{page.content}}
                </p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>

{% endblock %}
```

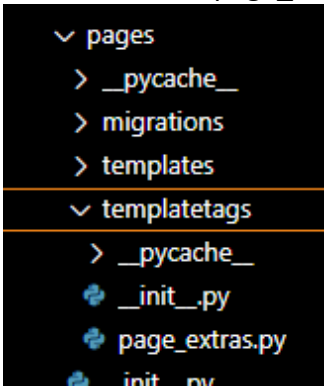
8. Agregar la aplicación en el archivo settings.py
9. Realizar las migraciones correspondientes
10. Entrar al Admin y registrar 3 páginas:
    a. , Aviso legal, Cookies
11. Verificar su funcionamiento
    http://127.0.0.1:8000/pages/1
12. Cambiar el verbose name de la aplicación
    a.
```
from django.apps import AppConfig


class PagesConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'pages'
    verbose_name = 'Páginas'
```
13. Verifique su despliegue
    a. 

Creación de Tags propios

En Django, los `templatetags` son una forma de añadir funcionalidad personalizada a tus plantillas. Pueden usarse para realizar tareas complejas, filtrar datos o modificar el contenido por ejemplo.

1. Crear la carpeta templatetags
2. Crear el archivo __init__.py para indicar que es un paquete

3. Crear el archivo page_extras.py



4. En el archivo page_extras.py

```python
from django import template
from pages.models import Page

# Registrarlo en la librería templates
register = template.Library()


@register.simple_tag
def get_page_list():
    pages = Page.objects.all()
    return pages
```

5. En el archivo base.html modificarlo

```
<p class="m-0 mbt">
  {% load page_extras %}
  {% get_page_list %}
</p>
```

6. Verificar que obtenga la información
7. Hacerlo dinámico

```
<p class="m-0 mbt">
  {% load page_extras %}
  {% get_page_list as page_list%}
  {% for page in page_list %}
    <a href="{% url 'pages:page' page.id %}" class='link'>{{page.title}} </a>
    {% if not forloop.last %}·{% endif %}
  {%endfor%}
</p>
```