Reporte Servidor Local

Se va a utilizar Django CMS para el desarrollo del sitio web de *Ciencia Maskay* debido a los paquetes de python que ofrece y la facilidad para los diseñadores de contenido de agregar o modificar las diferentes páginas del sitio web.

El plan es primero correr el sitio en un servidor local y luego, usando el servidor brindado por la universidad, repetir o migrar todos los archivos creados a este servidor.

Se utilizó Linux OS, con distribución Ubuntu 22.05 JammyJellyfish.

Instalación de django cms y creación del proyecto

1. Instalar python3 y crear un entorno virtual.

En un directorio para el proyecto abrir el terminal y escribir,

```
sudo apt install python3-full #tener permisos de administrador#
pip install --upgrade pip
```

Luego,

```
python3 -m venv cienciaMaskay
source cienciaMaskay/bin/activate #escribir esto cada que se quiere entrar al entorno virtual#
```

Entonces, estamos dentro del entorno virtual.

2. Instalar django cms.

```
pip install djangocms-installer
```

3. Crear el proyecto.

```
pip install pytz
djangocms MASKAY #MASKAY es donde todas las dependencias van a estar instaladas#
```

Las configuraciones que se instalan son las que vienen predeterminadas. Esto se puede cambiar luego usando un editor de código como VS Code.

4. Crear superusuario o administrador del sitio web.

```
python manage.py createsuperuser
# Crear usuario y contraseña#
```

5. Correr el servidor.

```
cd MASKAY
python manage.py runserver
# Abrir el server haciendo clic en el link#
```

6. Terminar ejecución del servidor y del entorno virtual.

```
# Ctrl + C#
deactivate
```

Lo que se obtiene es un sitio web muy básico donde se puede añadir y/o modificar contenido, i.e, un manejador de contenido.

Ahora, usando una plantilla que se asemeje a la propuesta del sitio web de *Ciencia Maskay,* el objetivo es modificar la plantilla de acuerdo a nuestras necesidades usando el editor VS Code.

La plantilla que se utilizará es:
https://bootstrapmade.com/enno-free-simple-bootstrap-template/#download
Se procede a descargar la plantilla y extraer los archivos. Luego, abrir la carpeta MASKAY usando VS Code.

Modificación de la plantilla

1. Copiar todos los elementos de la carpeta *assets* del template, en la carpeta *static* del proyecto creado.
Esto permite que todos los archivos necesarios para la renderización de la plantilla estén en nuestro proyecto de django.
NOTA: Existen dos carpetas *static* en el directorio creado para el proyecto, se recomienda copiar los elementos en ambas carpetas.

2. Copiar el archivo *index.html* *en la carpeta* *templates* de nuestro proyecto.
3. Editar *index.html* y llevarlo al formato de django. Usar los siguientes scripts,

```
{% load cms_tags menu_tags sekizai_tags %}
{% load static %} #en el preámbulo#

{% render_block "css" %} #al final del block <head>#

{% cms_toolbar %} #inicio del bloque <body>#
{% render_block "js" %} #final del bloque <body>
```

4. Referenciar los elementos del archivo *index.html* a la carpeta *static*. Por ejemplo,

```
"{% static 'assets/vendor/bootstrap/js/bootstrap.bundle.min.js' %}"
```

NOTA: Verificar la correcta referencia de elementos en cada una de las secciones de los archivos del proyecto.

5. Eliminar secciones que no se van a utilizar, editar títulos, texto, imágenes y links.
NOTA: Para cambiar o editar una imagen, añadir antes la imagen a ambas carpetas *img* que se encuentran en las dos carpetas *static* antes mencionadas.
6. Para visualizar los cambios, guardar los ajustes realizados en VS Code y recargar el sitio web.

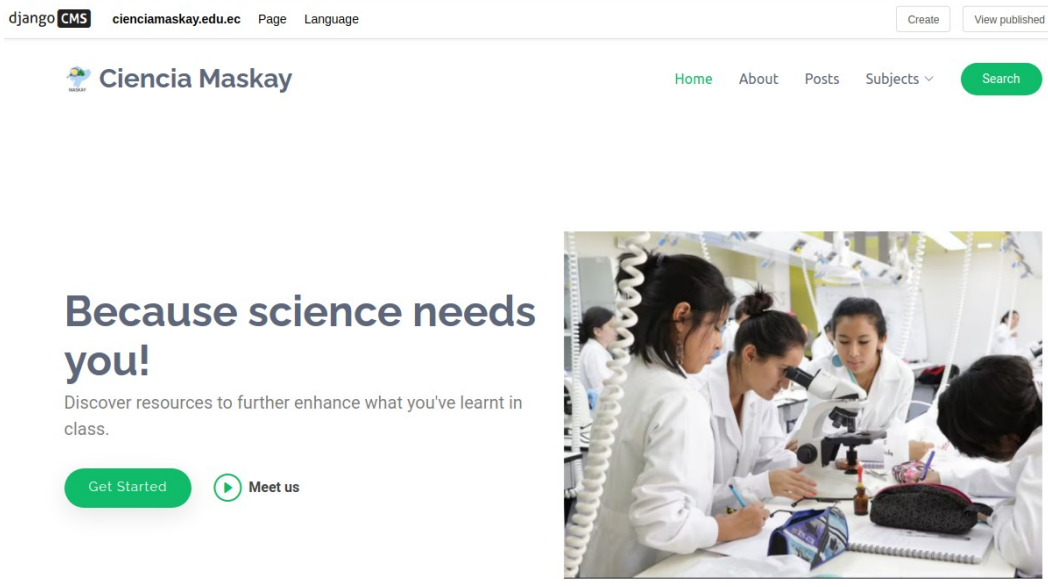


Figura 1. Primera versión del sitio web Ciencia MASKAY

Una vez realizada la primera versión del sitio web a través de la modificación de la plantilla, se puede empezar a usar *django cms* para editar el contenido del sitio web.

Editar el contenido del sitio web usando django cms

Ingresar con las credenciales de superusuario para realizar cambios en el sitio web.

1. Crear una página

En la interfaz de administrador, ir a la sección *Page* en el menú lateral. Hacer clic en el botón *Add Page* en la esquina superior derecha. Luego, completar el formulario con la siguiente información:

- *Título*: El título de la página (por ejemplo, "Inicio").
- *Slug*: La parte de la URL (por ejemplo, `/inicio/`).

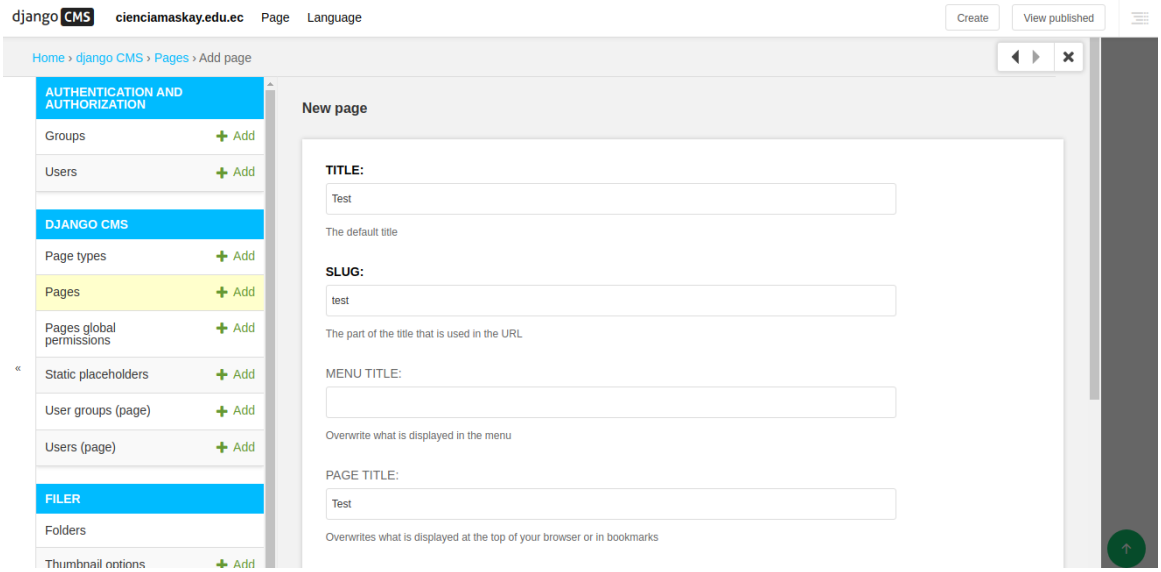


Figura 2. Menú para crear una página en Django CMS.

Luego, hacer clic en *Save and continue editing* para crear la página. Clic en *Advanced Settings* y seleccionar lo siguiente:

- *Template*: Seleccionar uno de los templates disponibles. (Se recomienda tener templates individuales: uno para la página de videos, uno para la página de los artículos, uno para la página principal,etc.)
- *Language*: Selecciona el idioma predeterminado para la página.

Finalizar haciendo clic en *Save*.

2. Añadir plugins a la página web.

Las siguientes líneas de código permiten añadir plugins. Copiar y pegarlas en en el archivo *index.html* (template del homepage) donde necesitemos añadir o editar contenido usando. Hacer esto usando VS Code.

```
{% block content %}
        {% placeholder "content" %}
{% endblock content %}
```

Por ejemplo, si pegamos las líneas de código después de *</header/>*,

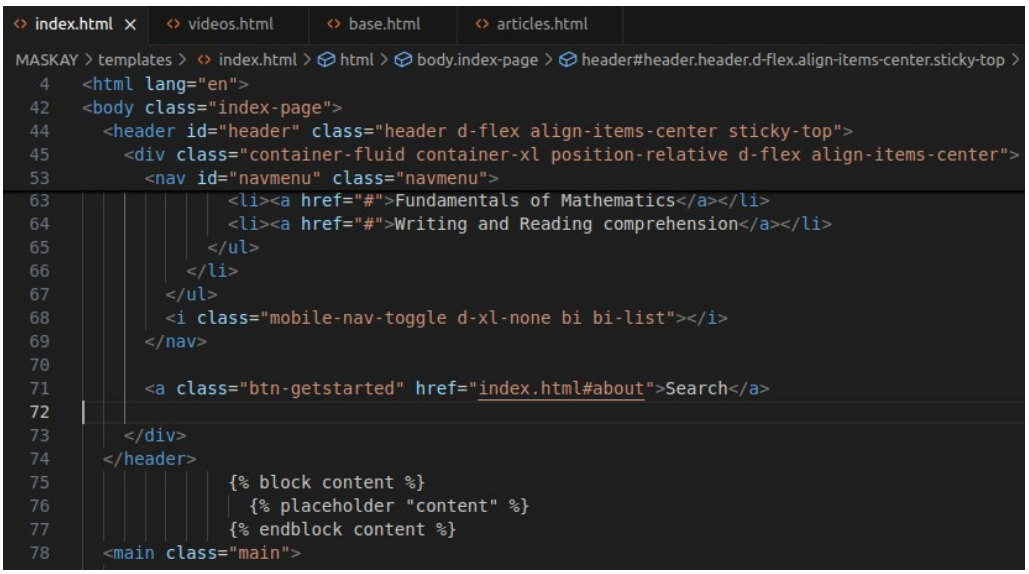


Figura 3. Añadir plugins usando VS Code.

y hacemos clic en la esquina superior derecha de nuestro sitio web, podremos añadir plugins después del título de la página. En este caso se usó un plugin de texto.

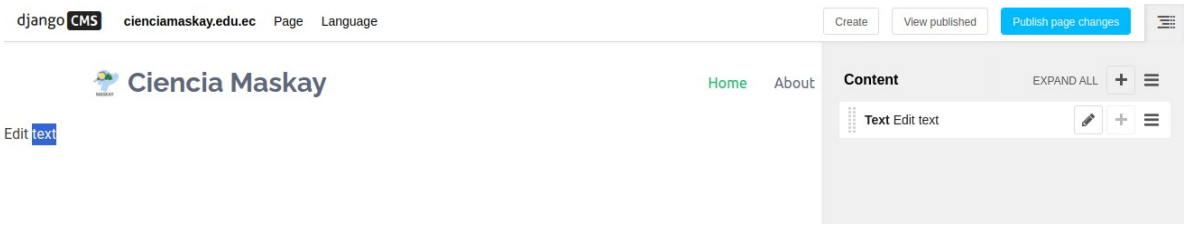


Figura 4. Edición de plugins desde django cms.

Ahora, si se quiere editar contenido ya publicado del sitio web, propio del template utilizado, se procede a reemplazar el texto por `{% placeholder "NAME" %}`. Entre comillas colocar el nombre con el cual se va a identificar el plugin para su posterior edición desde la pestaña de plugins. De igual manera, usar VS Code para hacer esto. Por ejemplo, se muestra los cambios realizados para editar la sección *About* del sitio web.

```
<!-- Section Title -->
<div class="container section-title" data-aos="fade-up">
  <h2>About</h2>
  <p>{% placeholder "About" %}</p>
</div><!-- End Section Title -->

<div class="container">

  <div class="row gy-4">
    <div class="col-lg-6 position-relative align-self-start" data-aos="fade-up" data-aos-delay="100">
      <img src="{% static 'assets/img/photo.jpg' %}" class="img-fluid" alt="">
      <a href="https://www.youtube.com/watch?v=ADpEXQZtFJQ" class="glightbox pulsating-play-btn"></a>
    </div>
    <div class="col-lg-6 content" data-aos="fade-up" data-aos-delay="200">
      <h3>{% placeholder "Section_title" %}</h3>
      <p class="fst-italic">
        {% placeholder "Section_description" %}
      </p>
      <ul>
        <li><i class="bi bi-check2-all"></i> <span>Objetivo 1</span></li>
        <li><i class="bi bi-check2-all"></i> <span>Objetivo 2</span></li>
        <li><i class="bi bi-check2-all"></i> <span>Objetivo 3</span></li>
      </ul>
    </div>
  </div>
```

Figura 5. Sección *About* del sitio web.

Ahora, se puede editar el contenido desde la pestaña de plugins.

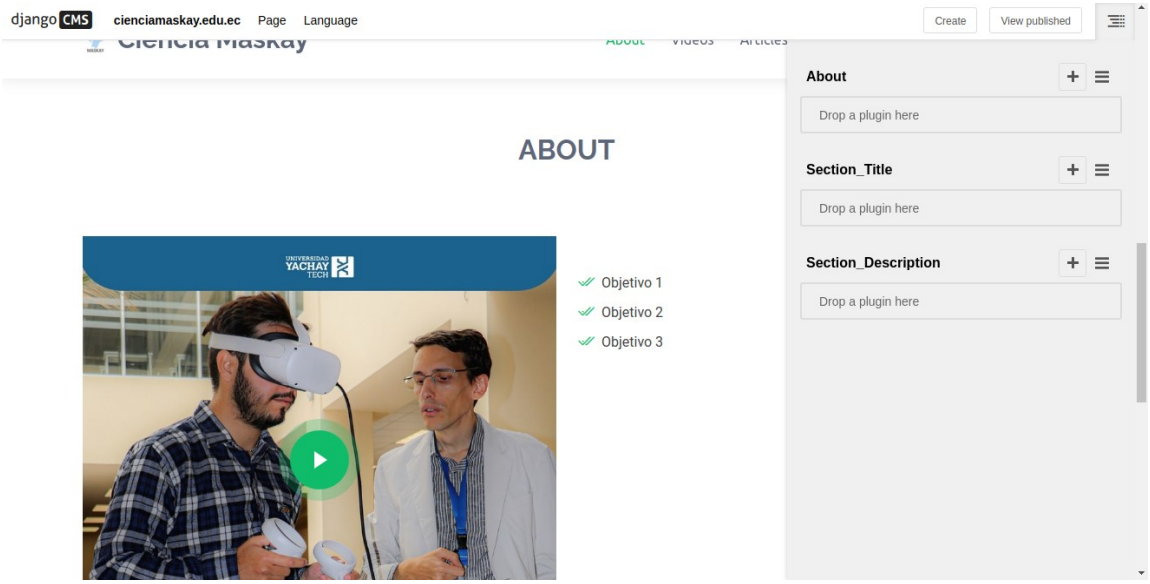


Figura 6. Editar contenido usando plugins.

NOTA: Se recomienda al programador añadir los tags de plugins en todas las instancias donde se quiera editar el contenido. De esta forma, los diseñadores del sitio web solo deben editar los plugins desde django cms.

NOTA: Para visualizar los cambios realizados en VS Code, guardar usando Ctrl + S. Luego, recargar el sitio web desde el navegador.

Anexos

Editar la página *Articles* desde django cms

1. Desde la pestaña de plugins en django cms, añadir plugins de texto para el título y el abstract del artículo. **NOTA:** Previamente, se debió haber añadido los tags de plugins desde VS Code.

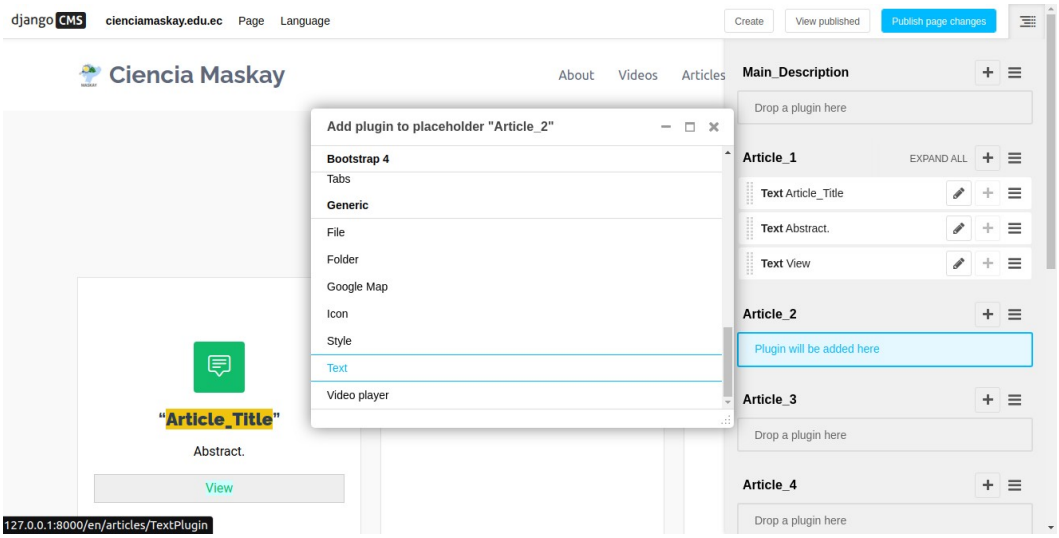
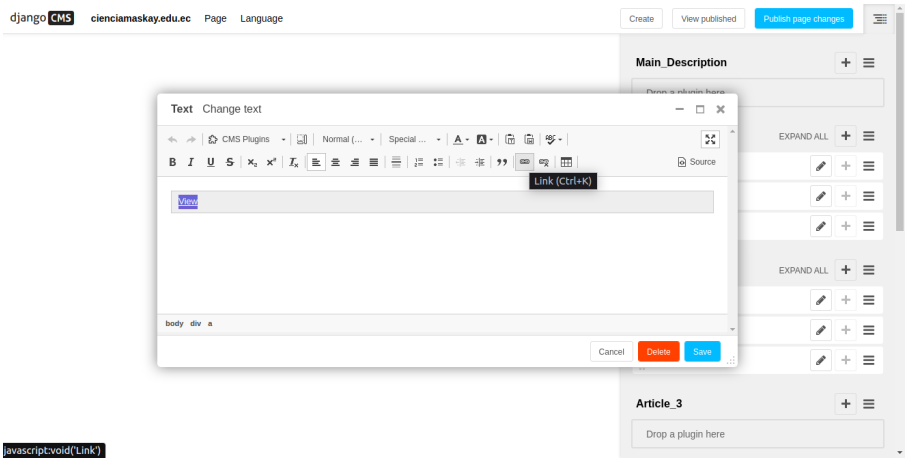
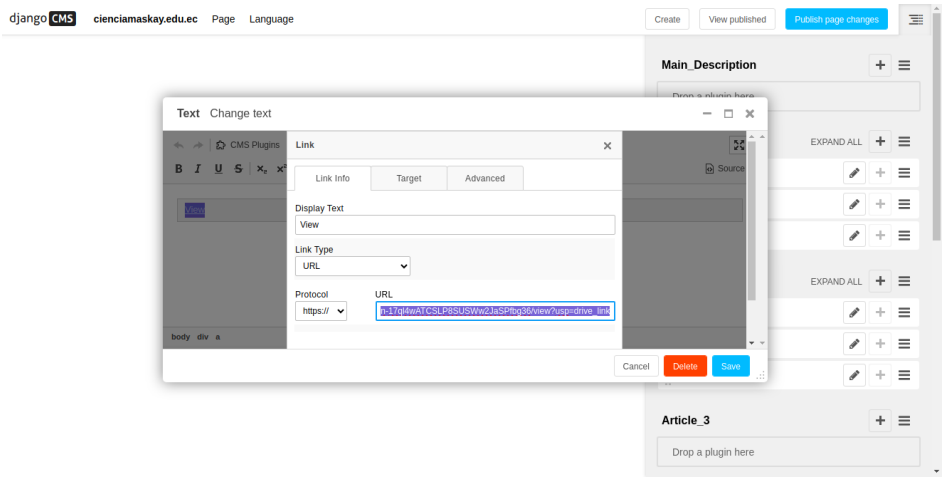


Figura 7. Añadir plugin genérico de texto.

2. Para esta primera versión, los artículos se acceden a través de un link que lleva a Google Drive, donde el artículo ha sido subido. Para esto, añadir otro plugin de texto. Al editarlo, usar la herramienta *Link.*



Figuras 8-9. Añadir plugin de link.



3. Una vez añdidos los plugins para un artículo, se puede copiar y pegar las mismas opciones para los siguientes artículos.

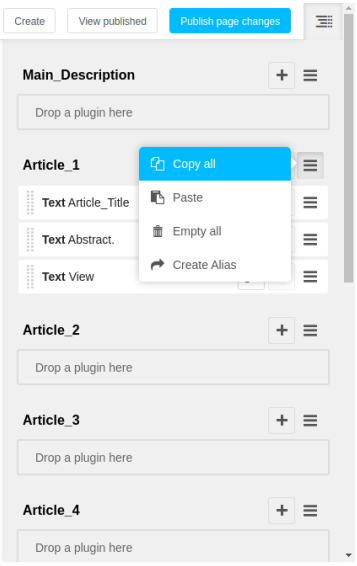


Figura 10. Copiar y pegar los plugins con las opciones preferidas.

Editar o añadir imágenes al sitio web

En esta primera versión, la edición de imágenes se realiza solo desde VS Code. Sin embargo, sí es posible añdir imágenes desde django cms.
Por ejemplo, para cambiar la imagen que aparece en la sección *About* realizar los siguientes pasos.
1. Guardar la imagen en la carpeta *MASKAY/static/assets/img*. Se recomienda usar imágenes de dimensiones similares.

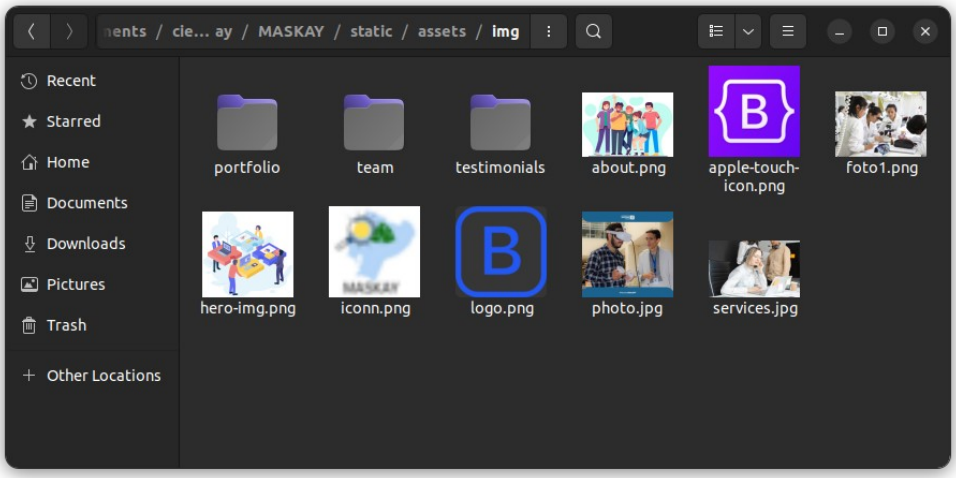


Figura 11. Path donde las imágenes son guardadas.

2. Editar el path de la imagen desde VS Code.

```
<!-- Section Title -->
<div class="container section-title" data-aos="fade-up">
  <h2>About</h2>
  <p>{% placeholder "About" %}</p>
</div><!-- End Section Title -->

<div class="container">

  <div class="row gy-4">
    <div class="col-lg-6 position-relative align-self-start" data-aos="fade-up" data-aos-delay="100">
      <img src="{% static 'assets/img/about.png' %}" class="img-fluid" alt="">
      <a href="https://www.youtube.com/watch?v=ADpEXQZtFJQ" class="glightbox pulsating-play-btn"></a>
    </div>
    <div class="col-lg-6 content" data-aos="fade-up" data-aos-delay="200">
      <h3>{% placeholder "Section_title" %}</h3>
      <p class="fst-italic">
        {% placeholder "Section_description" %}
      </p>
```

Figura 12. Editar el path de la imagen desde VS Code.

NOTA: Se puede también editar el link al que lleva la imagen desde VS Code (es la línea siguiente al path de la imagen).

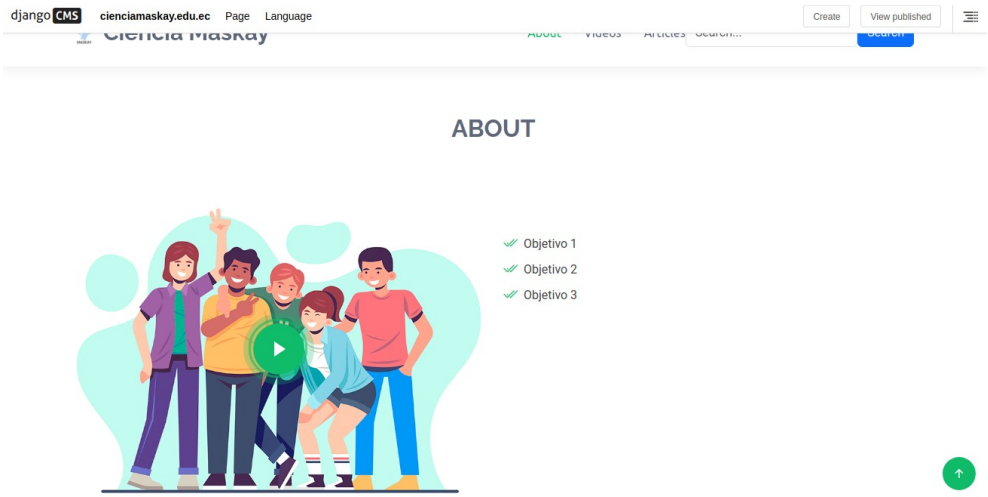


Figura 13. Cambio de imagen en el sitio web.

Añadir barra de búsqueda

1. En el Menú del sitio web, añadir las siguientes líneas de código para crear el Search Button. *NOTA:* Realizar este cambio en el menú de cada una de las páginas.

```
<form action="{% url 'search' %}" method="get" class="d-flex">
        <input type="text" name="q" class="form-control" placeholder="Search..."
         required>
        <button type="submit" class="btn btn-primary">Search</button>
</form>
```

2. Crear un archivo *views.py* dentro del proyecto y copiar las siguentes líneas de código para llamar a la página con los resultados de la búsqueda. *NOTA:* 'searchresults.html' es el template de la página con los resultados.

```
from django.shortcuts import render
from django.contrib.contenttypes.models import ContentType

def search(request):
        query = request.GET.get('q', '')
        results = []
        if query:
                # Search across specific models or content
                # Replace 'ModelName' with actual models you'd like to search
                results = ContentType.objects.filter(model__icontains=query)

        return render(request, 'searchresults.html', {'query': query, 'results': results})
```

3. En *urls.py* dentro del proyecto copiar el siguiente path.

```
urlpatterns = [
        # Other URLs...
        path('search/', search, name='search'),
 ]
```

4. Crear la página 'searchresults' usando el template 'searchresults.html'. Tomar en cuenta el header y footer de las demás páginas y añadir al *body* las siguientes líneas.

```
{% block content %}
<div class="container mt-4">
      <h2>Search Results for "{{ query }}"</h2>
      {% if results %}
          <ul>
            {% for result in results %}
                  <li><a href="{{ result.get_absolute_url }}">{{ result.title }}</a></li>
            {% endfor %}
          </ul>
      {% else %}
          <p>No results found.</p>
      {% endif %}
</div>
{% endblock %}
```

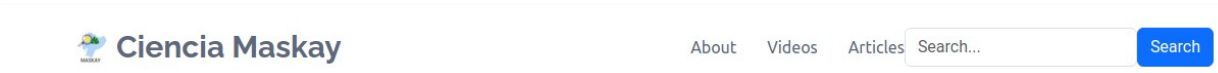


Figura 14. Search button en el menú principal.

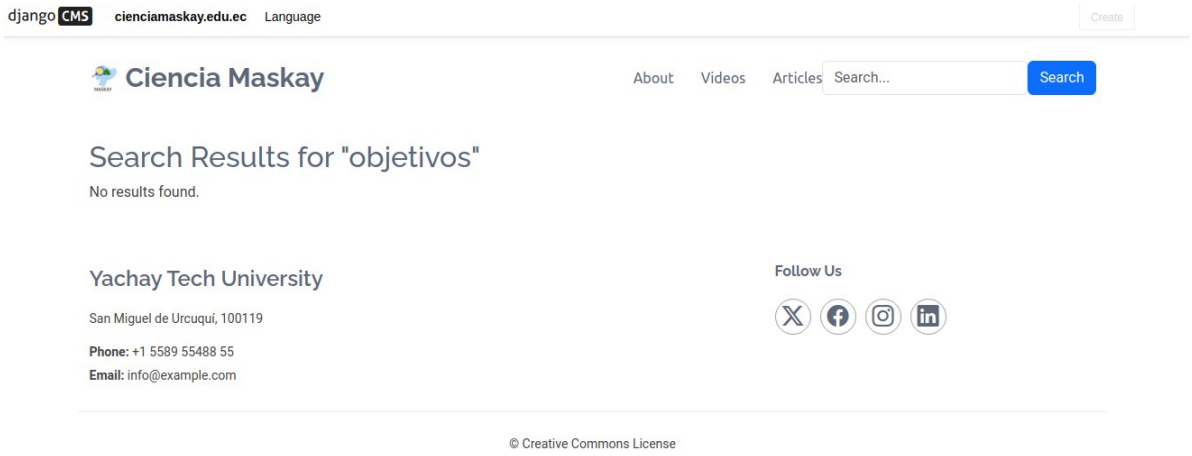


Figura 15. Paǵina de resultados de búsqueda.

Añadir navegación entre páginas creadas

Desde django cms escribir el nombre que aparecerá en el menú editando cada una de las páginas.
En el menú de *Pages*, hacer clic en el lápiz para editar las configuraciones de cada página.
Luego, editar la sección de *Menu Title*.

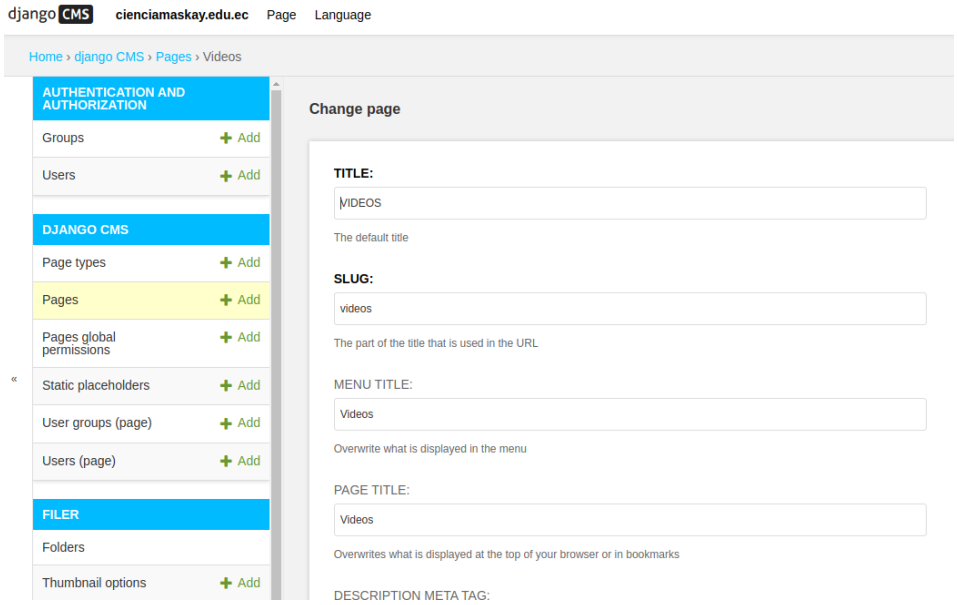


Figura 16. Edición del título de la página para el menú principal.

Desde VS Code, añadir la siguiente línea de código en el header de cada una de las páginas.

```
<body class="index-page">
  {% cms_toolbar %}
  <header id="header" class="header d-flex align-items-center sticky-top">
    <div class="container-fluid container-xl position-relative d-flex align-items-center">

      <a href="/#hero/" class="logo d-flex align-items-center me-auto">
        <!-- Uncomment the line below if you also wish to use an image logo -->
        <img src="{% static 'assets/img/iconn.png' %}" alt="" alt="">
        <h1 class="sitename">Ciencia Maskay</h1>
      </a>

      <nav id="navmenu" class="navmenu">
        <ul>
          <li><a href="#about">About</a></li>
          {% show_menu 1 100 100 100 %}
        </ul>
        <i class="mobile-nav-toggle d-xl-none bi bi-list"></i>
      </nav>

      <form action="{% url 'search' %}" method="get" class="d-flex">
        <input type="text" name="q" class="form-control" placeholder="Search..." required>
        <button type="submit" class="btn btn-primary">Search</button>
      </form>

    </div>
  </header>
```

Figura 17. Añadir la línea de código subrayada en VS Code.

Observaciones

Es importante evaluar la experiencia de los usuarios (diseñadores de contenido, estudiantes y profesores) para realizar mejoras en futuras versiones de la plataforma. Además, considerar la creación de una base de datos en el servidor para guardar los artículos y videos o almacenarlos en la nube.

Las próximas fases deberían centrarse en optimizar la adaptabilidad del sitio para distintos dispositivos, incorporar soporte para múltiples idiomas y alojar el sitio web en un servidor confiable.

Referencias

Getting started with django cms. *https://www.youtube.com/watch?v=NbsRVfLCE1U&t=102s*

django cms documentation. *https://djangocms.readthedocs.io/en/latest/*

Django CMS-Learn to create a website. *https://www.youtube.com/watch?v=Nlk2ml_jKTY&list=PLSPMgrv4IuJ4jRwXtja7nxYgifAnpZz6P*

django documentation. *https://docs.djangoproject.com/en/5.1/intro/*

django search bar tutorial. *https://learndjango.com/tutorials/django-search-tutorial*

Autor
Alejandro Silva
alejandro.silva@yachaytech.edu.ec