

Desarrollo

Instalación de Drupal11

Utilizo DDEV para realizar la asignatura, con lo que los pasos iniciales para crear el proyecto son los siguientes:

```
ddev config --project-type=drupal11 --docroot=web
ddev start
ddev composer create-project "drupal/recommended-project:^11"
ddev composer require drush/drush
ddev drush site:install --account-name=admin --account-pass=dbe-PEC2 -y
ddev launch
```

La contraseña para el usuario *admin* es *dbe-PEC2*.

```
DBE-PEC2/drupal-site on main [?] via v8.4.11 took 3s
+ ddev drush site:install --account-name=admin --account-pass=dbe-PEC2 -y
You are about to:
+ DROP all tables in your 'db' database.

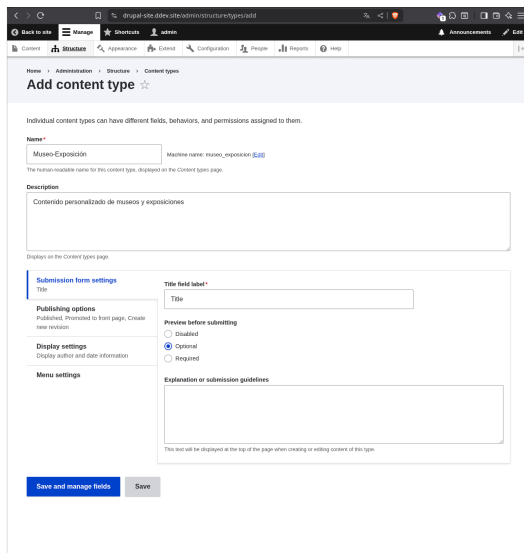
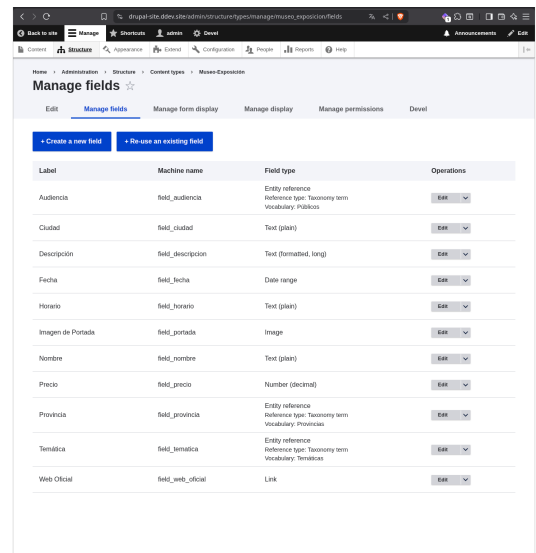
// Do you want to continue?: yes.

[notice] Starting Drupal installation. This takes a while.
[notice] Performed install task: install_select_language
[notice] Performed install task: install_select_profile
[notice] Performed install task: install_load_profile
[notice] Performed install task: install_verify_requirements
[notice] Performed install task: install_verify_database_ready
[notice] Performed install task: install_base_system
[notice] Performed install task: install_bootstrap_full
[notice] Performed install task: install_profile_modules
[notice] Performed install task: install_profile_themes
[notice] Performed install task: install_install_profile
[notice] Performed install task: install_configure_form
[notice] Performed install task: install_finished
[success] Installation complete. (Admin)
DBE-PEC2/drupal-site on main [?] via v8.4.11 took 4s
+ ddev launch
```

Ahora ya hay acceso al proyecto Drupal.

Creación de contenido personalizado

Primero nos logueamos como *admin*. Vamos *Manage>Structure>Content types* para crear el tipo de contenido y sus campos. Ciudades se ha cambiado por Provincias, ya que me resultaba más sencillo a la hora de crear términos.

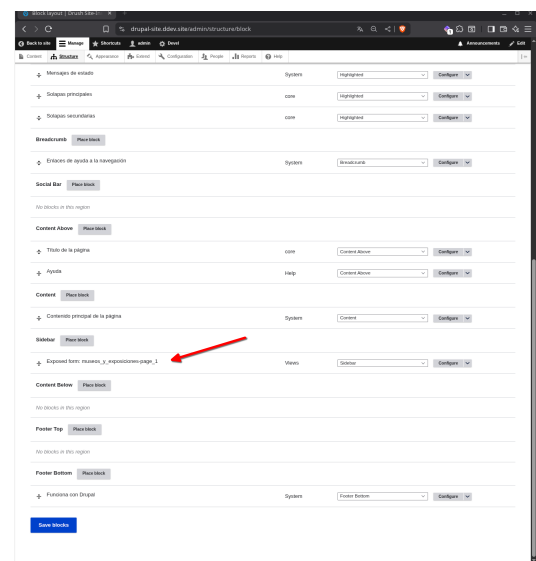
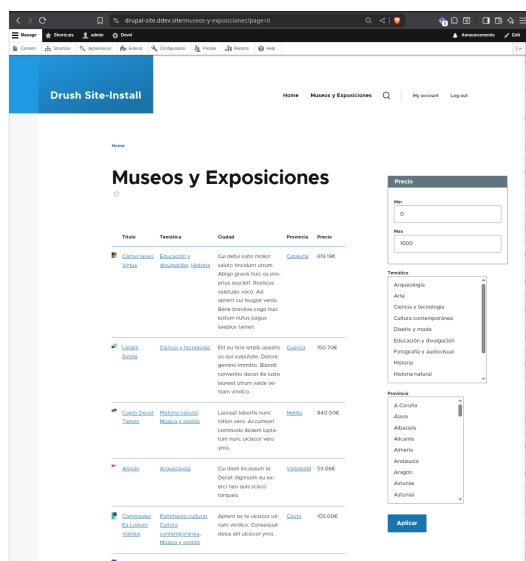
Label	Machine name	Field type	Operations
Audiencia	field_audiencia	Entity reference Reference type: taxonomy term Vocabulary: Públicos	Edit
Ciudad	field_ciudad	Text (plain)	Edit
Descripción	field_descripcion	Text (formatted, long)	Edit
Fecha	field_fecha	Date range	Edit
Horario	field_horario	Text (plain)	Edit
Imagen de Portada	field_portada	Image	Edit
Nombre	field_nombre	Text (plain)	Edit
Precio	field_precio	Number (decimal)	Edit
Provincia	field_provincia	Entity reference Reference type: taxonomy term Vocabulary: Provincias	Edit
Temática	field_tematica	Entity reference Reference type: taxonomy term Vocabulary: Temáticas	Edit
Web Oficial	field_web_oficial	Link	Edit

Se ha utilizado *Taxonomy Manager* para la entrada massiva de términos. Y para crear contenido automáticamente se ha utilizado *Devel_manager*. Los dos se han instalado a través de composer.

```
ddev composer require 'drupal/taxonomy_manager:^2.0'
ddev composer require drupal/devel
```

Creación de vista

Se ha creado la vista en forma de página con el filtrado de provincias, temática y rango de precios. Las opciones de filtrado quedaban arriba y para estilizar un poco la vista lo hemos configurado para ponerlo en el lateral.

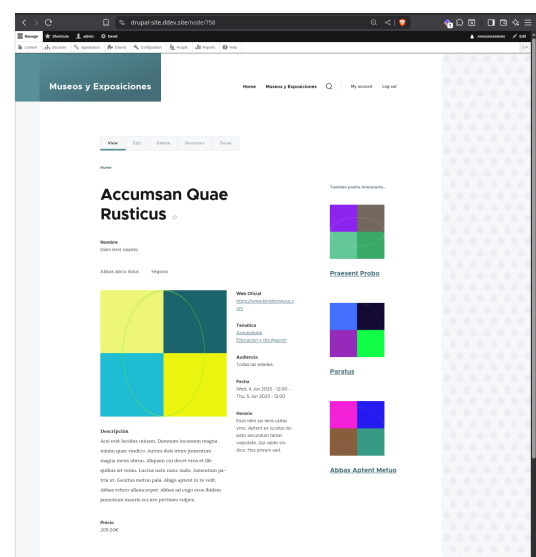


Para conseguirlo se ha separado el formulario (opciones de filtrado) del resto de la vista. El propio Drupal cuando lo separas te crea un bloque con los filtros que luego nos permite mover a la barra lateral del layout de la página.

Creación de página

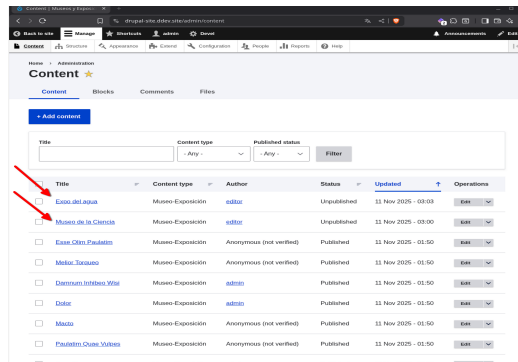
Utilizando *Layout Builder* se ha diseñado la presentación de los contenidos. Los textos largos (Descripción) se han trimmeado. Lo que para ver el texto completo (añadir función 'Leer más') requiere de complementos que no he querido añadir por no complicarlo más, ya que anteriormente instalando complemento el proyecto murió y tuve que rehacerlo de cero.

Además se ha creado el bloque de tres sugerencias por coincidencia de provincia y de temática.



Creación de roles

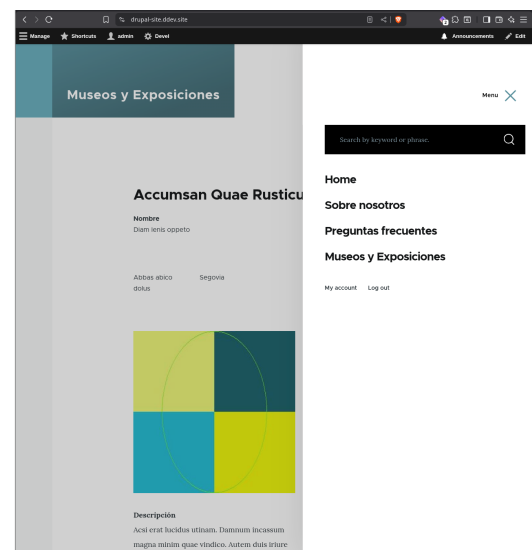
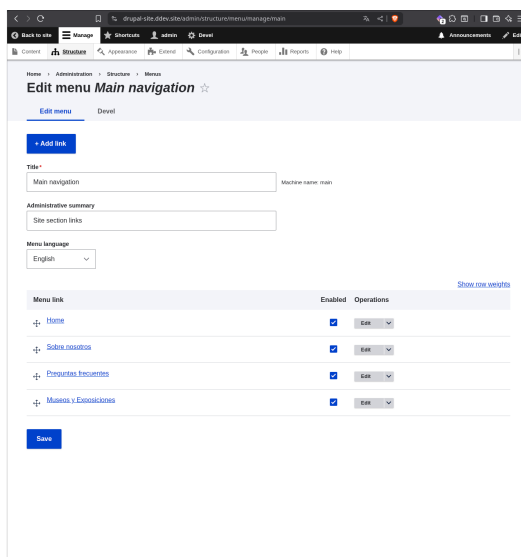
Se ha creado el rol de colaborador con los permisos solicitados: crear, editar y eliminar su propio contenido. Pero no era suficiente ya que no aparecía la opción para crear contenido, así que comparando con el rol ya definido Content editor, se han añadido las opciones de acceso a página de contenidos, poder visualizar contenido propio no publicado (por si se quiere editar contenido creado pero todavía no lo ha publicado un administrador) y uso de toolbar.



Se han creado las entradas de museo nuevas y se puede ver que están sin publicar. No he sabido encontrar el permiso para permitir publicar, en su defecto he hecho que todas las entradas de Museo-Exposición se creen como no publicadas y luego ya alguien con más rango la publicará. Quizás no es la solución más elegante.

Creación about y faqs

Pues relativamente sencillo, se crean dos contenido de página básica con título, contenido y alias URL. Se añaden al menú yendo a *Manage>Structure>Menus>Main Navigator* y si añaden los links con el nombre que queremos mostrar en el menú y el alias que hemos puesto.

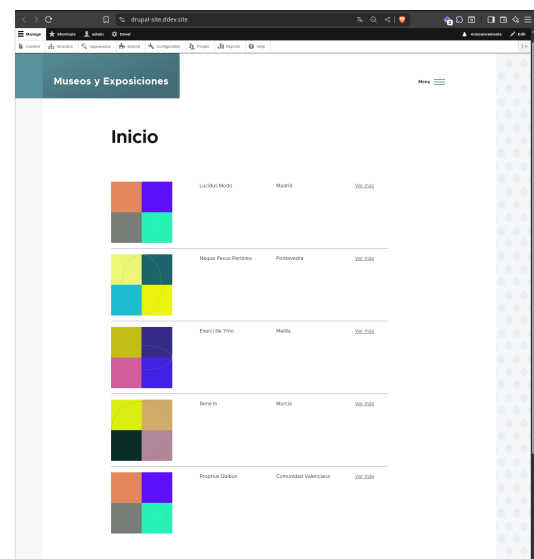
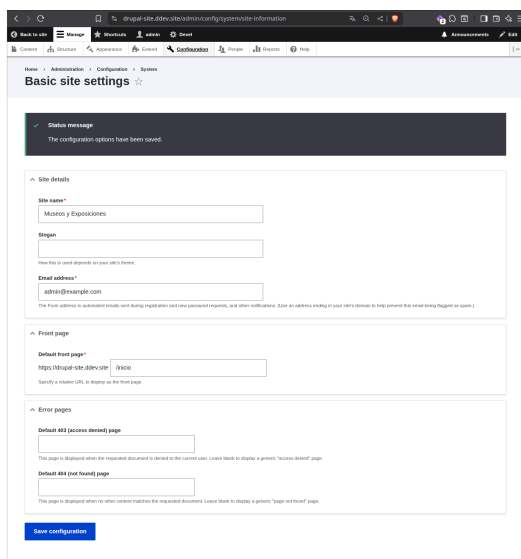


Como ya no me cabe tanto menú se acorta a botón Menú, en la parte superior derecha. De ahí se despliega una barra lateral con todos los accesos.

Creación de página de inicio

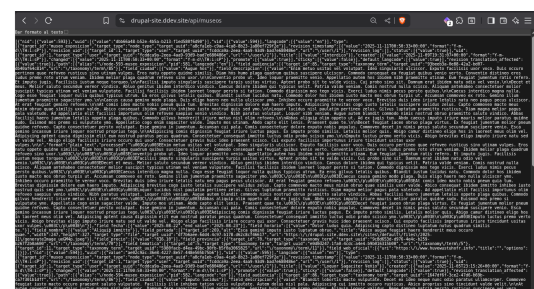
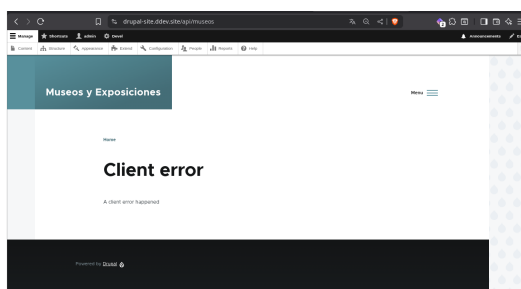
Pues igual que se ha hecho anteriormente con las Vistas y los filtrados contextuales, se hace algo parecido. Aplicamos una vista, seleccionamos orden aleatorio, ordenamos por tabla, seleccionamos los campos deseados, linkamos únicamente el enlace de detalles y ponemos el paginador con 5 elementos fijos. También añadimos ruta e incorporamos la vista al menú de navegación.

Por último nos queda configurar a Drupal para que cambie la página de inicio por defecto. Vamos a *Configuration>System>Basic site settings* y aquí cambiamos la front page por nuestro path de la vista.



Creación de API REST

Se instala el RESTful Web Services junto con Serializer. Entonces desde Vista se puede crear la API. Como si hicieramos una vista: paginador full con 10 ítems por página. Se prueba la API en <https://drupal-site.ddev.site/api/museos>. Y la primera en la cara.



Por lo visto es cosa del formato de la vista. En *Format>Serializer settings* marcamos JSON como formato. Y voilà. Ahora no salta error pero sinceramente así no puedo saber si realmente ha funcionado.

Despliegue a servidor UOC

Pues ya sólo queda subirlo a un servidor y comprobar que funciona. Primero copia a github. Limpiar caché. Desmarcamos opciones de optimización. Cerramos Drupal. Guardamos en un zip el proyecto y también la base de datos exportada.

Importando base de datos al servidor UOC. Éxito! Descomprimido el proyecto. Modificado el settings.php. Lanzamos <https://eimtcm2.techlab.uoc.edu/~oromeraa/pec2/web/....>
Error! Hay que añadir el hash-salt ya que he trabajado con DDEV.

Y listo! Web desplegada!

