Documentación de pruebas realizadas a aplicativos 2025

Tabla de Contenido

Documentación de pruebas realizadas a aplicativos 2025 1
Tabla de Contenido 1
Introducción 1
Marco de Referencia 2
Pruebas realizadas para cambios en aplicativos 2
Estilos de capas en Geoserver para servicios OGC 2
1. Creación de estilos para capas a partir de imágenes .PNG mediante códigos SLD usando Filezilla y Udig para despliegue en Geoserver 2
2. Configuración entre Geoserver y Power BI para generación de Dashboards. 3
4. Alternativa: Crear Capas Personalizadas desde GeoServer 5
Pasos en GeoServer: 5
Creación de Sitios Web Estáticos 6
1. Prueba de creación de Web Map mediante desarrollo de sitio web estático usando GITHUB/GITLAB 6
Plantillas utilizadas como prueba para creación de producto mínimo viable de webmap para replicación de productos ESRI: 13
2. Prueba de creación de Story Map mediante desarrollo de sitio web estático usando GITHUB/GITLAB 13
Lecciones Aprendidas 13
Guía Paso a Paso para Crear un StoryMap en un Sitio Web Estático 14
Plantillas de storymaps creadas: 16
Conclusión 16
3. Prueba con desarrollo Formulario SIG mediante aplicativo suministrado por equipo de la Dirección Observatorio y Gestión del Conocimiento Cultural 16
4. Prueba creación de widget metadatos en Arcgis Online para los 35 experience builder publicados en Cultured-maps 18

Introducción

Este documento presenta los procesos y pruebas implementadas para funcionalidades de aplicativos que incluyan gestión de la información geográfica. Se

identifica a detalle cada uno de los procesos realizados en la construcción de Cultured-Maps mediante software libre, específicamente sobre la base de datos espacial SQL y servidor Geoserver donde reposa la información geográfica de la Secretaría de Cultura, Recreación y Deporte.

Marco de Referencia

Desde la Secretaría de Cultura Recreación y Deportes se tiene como objetivo unificar la información geográfica en un Geoportal o visor web que sea transversal a todas las entidades adscritas a la Secretaría, por lo tanto, se busca que esta nueva plataforma tenga una estructura que permita la convergencia de la información espacial que en general se maneja desde la Secretaría.

Esta plataforma Cultured-Maps construida con software libre va a contener la información debidamente estructurada, estandarizada, analizada, verificada y publicada para el público en general.

Cada actualización o mejora del aplicativo implica una serie de preparaciones y pruebas con el fin de obtener un resultado óptimo y sencillo en la interfaz de usuario del aplicativo. En este documento se explica el paso a paso de las pruebas realizadas y subproceso relevantes para la implementación y publicación de la información geográfica de la SCRD.

Pruebas realizadas para cambios en aplicativos

Estilos de capas en Geoserver para servicios OGC

1. Creación de estilos para capas a partir de imágenes .PNG mediante códigos SLD usando Filezilla y Udig para despliegue en Geoserver

Inicialmente se debe acceder a Filezilla y cargar las imagenes .png en la ruta /usr/share/geoserver/data_dir/styles

Posteriormente el código base creado mediante Udig es:

```
<?xml version="1.0" encoding="UTF-8"?>
<sld:StyledLayerDescriptor xmlns="http://www.opengis.net/sld"
   xmlns:sld="http://www.opengis.net/sld"
   xmlns:ogc="http://www.opengis.net/ogc"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   version="1.0.0">
 <sld:NamedLayer>
  <sld:Name>Artistas</sld:Name>
  <sld:UserStyle>
   <sld:Name>Artistas</sld:Name>
   <sld:FeatureTypeStyle>
    <sld:Rule>
     <sld:Name>artistas</sld:Name>
     <sld:PointSymbolizer>
      <sld:Graphic>
       <sld:ExternalGraphic>
                                                <sld:OnlineResource xlink:type="simple"
xlink:href="file:///usr/share/geoserver/data_dir/styles/ARTISTAS.png"/>
         <sld:Format>image/png</sld:Format>
       </sld:ExternalGraphic>
       <sld:Size>40</sld:Size>
      </sld:Graphic>
     </sld:PointSymbolizer>
    </sld:Rule>
   </sld:FeatureTypeStyle>
  </sld:UserStyle>
 </sld:NamedLayer>
</sld:StyledLayerDescriptor>
```

En el anterior código se debe reemplazar el nombre en negrilla de la imagen .PNG por el nombre de la imagen alojada en la carpeta /usr/share/geoserver/data_dir/styles

2. Configuración entre Geoserver y Power BI para generación de Dashboards.

¿Como crear un Tablero de Control en Power BI con DataFrames y Capas de GeoServer?

Power BI permite integrar datos geoespaciales y visualizarlos en mapas interactivos. Para utilizar una URL de GeoServer como la mencionada, puedes seguir estos pasos:

1. Configuración de la Capa WMS en Power BI
Power BI admite la integración de capas WMS en visualizaciones de mapas. Esto permite mostrar mapas base o capas específicas desde un servidor GeoServer.

Pasos:
Abrir Power BI Desktop:

Crea o abre un proyecto existente.
Agregar una Visualización de Mapa:

Selecciona la visualización Mapa o Mapa de ArcGIS desde el panel de visualizaciones.
Configurar la URL WMS:

Si utilizas el Mapa de ArcGIS:
Haz clic en el icono de engranaje para configurar.
Ve a la pestaña Agregar capas.
Selecciona WMS OGC y proporciona la URL de tu GeoServer:
bash

Copiar código
https://geoserver.scrd.gov.co/geoserver/Investigacion_Cultured_Maps/wms
En el campo Nombre de la Capa, ingresa el nombre completo de la capa que deseas usar, por ejemplo:
makefile
Copiar código
Investigacion_Cultured_Maps:Subindice_de_Ambiente_2023
Aplica los cambios.

2. Integración de DataFrames en Power BI
Si tienes datos en formato DataFrame (por ejemplo, en Python o pandas), puedes integrarlos en Power BI utilizando archivos intermedios o conectores directos.

Pasos para Exportar el DataFrame:
Guardar el DataFrame en un archivo compatible:

Desde Python, exporta el DataFrame a CSV o Excel:
python
Copiar código
import pandas as pd
df.to_csv('data.csv', index=False)

Guarda el archivo en una ubicación accesible para Power BI.
Importar el archivo a Power BI:

En Power BI, selecciona Obtener datos > Archivo > CSV/Excel.
Navega hasta el archivo exportado y cárgalo.
Visualizar los datos:

Usa visualizaciones como tablas, gráficos, o incluso mapas (si el DataFrame contiene datos espaciales como coordenadas).

3. Relación entre las Capas WMS y el DataFrame
Si tu DataFrame contiene datos geográficos (coordenadas, polígonos, o identificadores espaciales) que coinciden con la capa WMS, puedes integrarlos para crear un tablero dinámico.

Pasos para Relacionar los Datos:
Agregar Identificadores Espaciales al DataFrame:

Asegúrate de que el DataFrame tenga una columna que coincida con los identificadores de la capa WMS (por ejemplo, nombres de regiones, códigos únicos, etc.).
Unir Datos a la Capa WMS:

Configura el mapa en Power BI para que las regiones o puntos se coloreen según los datos del DataFrame.
Esto puede hacerse utilizando las columnas del DataFrame como valores para las capas.

Interactividad en el Tablero:

Configura interacciones entre el mapa y otras visualizaciones en el tablero para permitir filtros dinámicos.

4. Alternativa: Crear Capas Personalizadas desde GeoServer

Si necesitas más control sobre la capa utilizada en Power BI, puedes exportar los datos de GeoServer como archivos GeoJSON o CSV para utilizarlos directamente.

Pasos en GeoServer:

1. **Usar una URL de WFS para descargar datos**:

Configura una solicitud WFS desde GeoServer para obtener datos geográficos:
bash

Copiar código
```
https://geoserver.scrd.gov.co/geoserver/Investigacion_Cultured_Maps/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=Investigacion_Cultured_Maps:Subindice_de_Ambiente_2023&outputFormat=application/json
```

-
- Esto descargará los datos como GeoJSON.

2. **Convertir GeoJSON a CSV o Excel**:

Usa Python para convertir el GeoJSON en un archivo más compatible con Power BI:
python
Copiar código
```
import geopandas as gpd
df = gpd.read_file('data.json')
df.to_csv('data.csv', index=False)
```

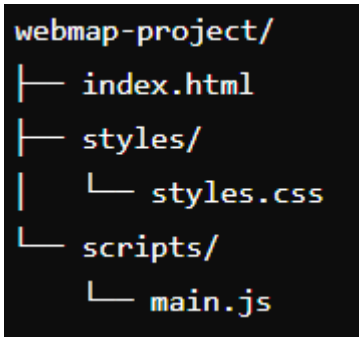
-

3. **Importar los datos a Power BI**:
   - Sigue los pasos mencionados para cargar un CSV.

Creación de Sitios Web Estáticos

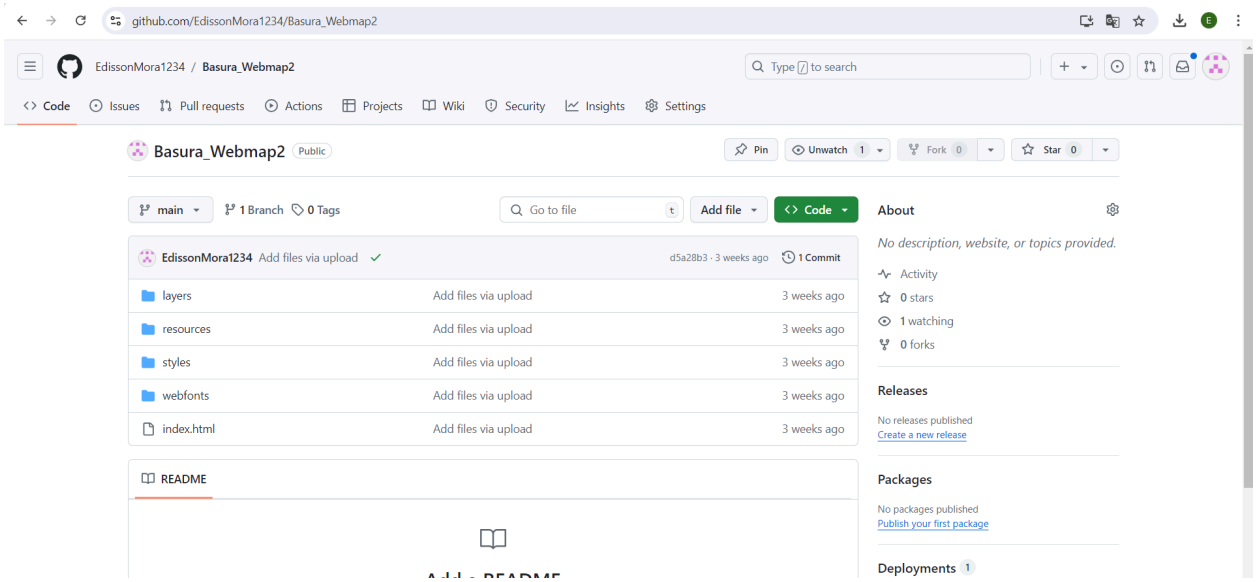
1. Prueba de creación de Web Map mediante desarrollo de sitio web estático usando GITHUB/GITLAB

Se propone crear una estructura de archivos todo en una misma carpeta como se muestra a continuación:



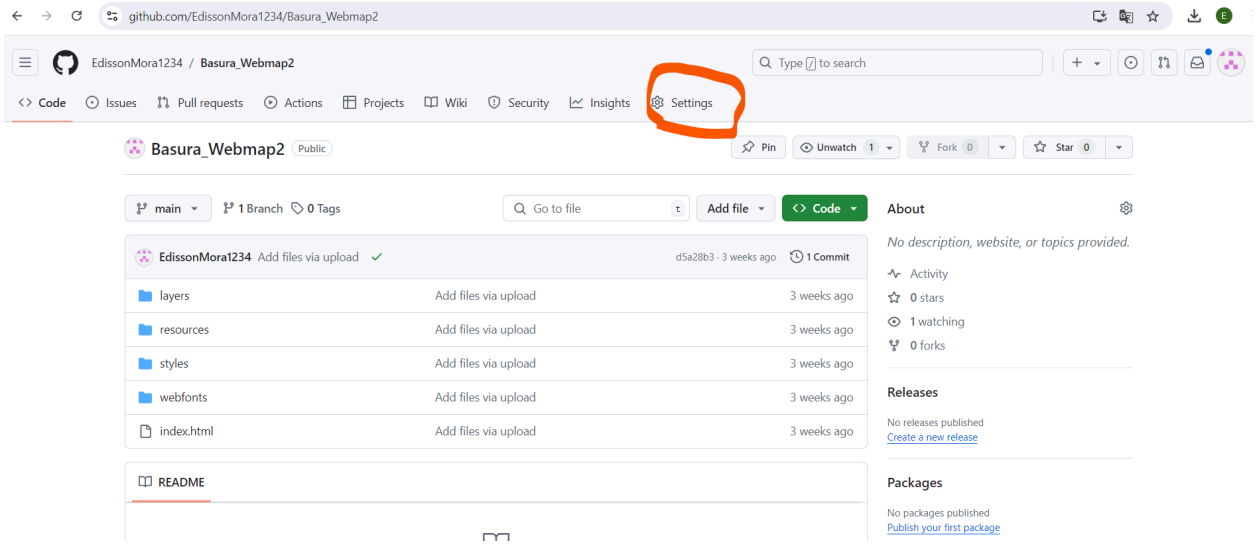
El WebMap que se crea consume capas WMS desde el geoserver de la SCRD, el contenido del código se crea de forma personalizada y para el despliegue debe

incluirse esto en un repositorio con las caracteristicas de GitHub Pages como se muestra a continuación:

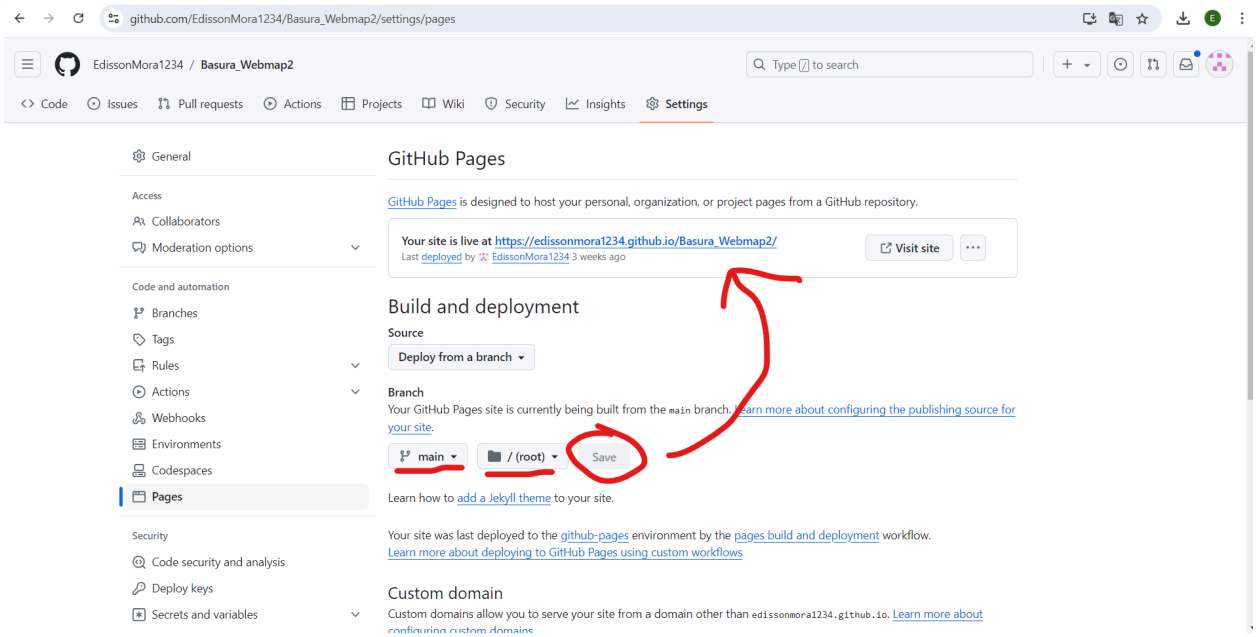


En la anterior imagen se muestra el repositorio de github llamado “Basura_Webmap2” en donde se alojan los archivos mencionados anteriormente.

Para el despliegue del sitio web estático se debe activar la función de “Github Pages” el cual se activa como se muestra a continuación:



En configuración o Settings del repositorio como se muestra en la imagen anterior se debe dejar configurado lo siguiente:



En la imagen anterior se observa que al realizar la configuración subrayada en rojo y al guardar y refrescar se genera la URL señalada con la flecha roja que es: https://edissonmora1234.github.io/Basura_Webmap2/

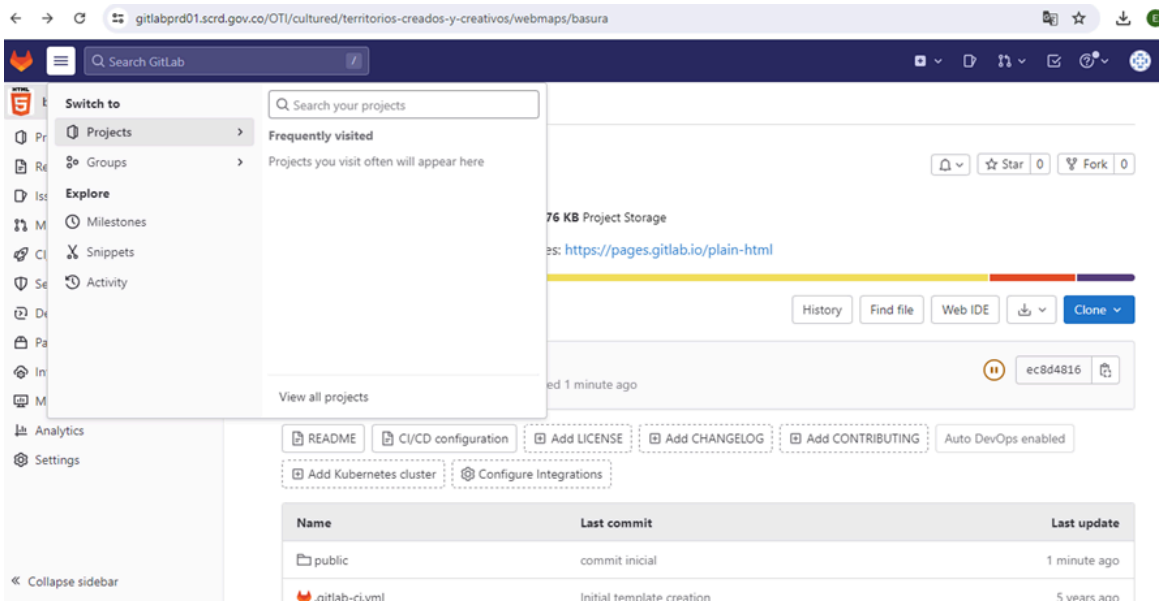
Ahora teniendo en cuenta que la SCRD no cuenta con el software Github sino con Gitlab el procedimiento es el siguiente:

Con el usuario y contraseña:

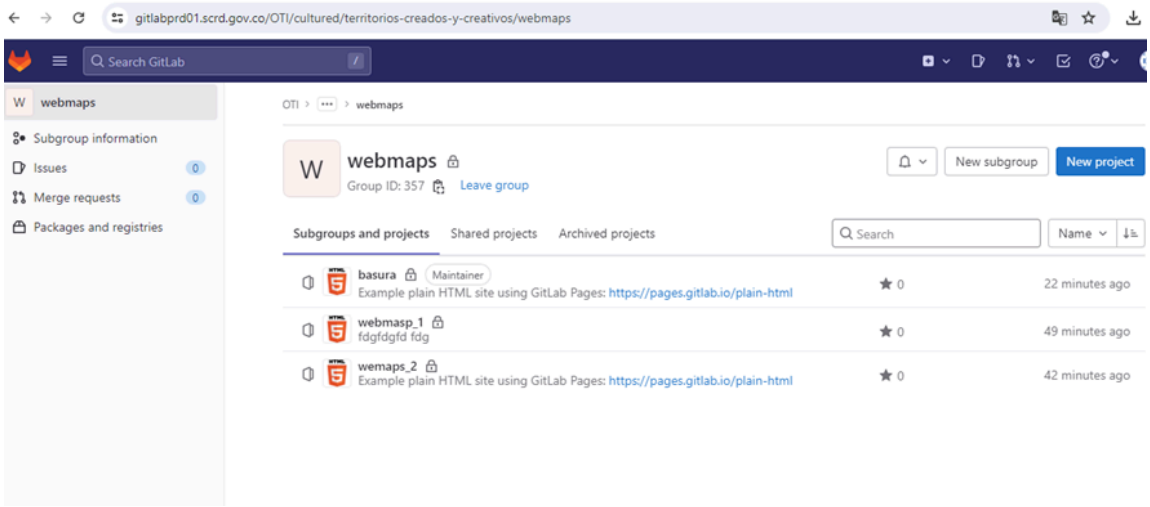
https://gitlabprd01.scrd.gov.co/users/sign_in?auto_sign_in=false

Usuario GIT: correo scrd
Contraseña: edisson0921

Se entra a Gitlab y se crea el repositorio

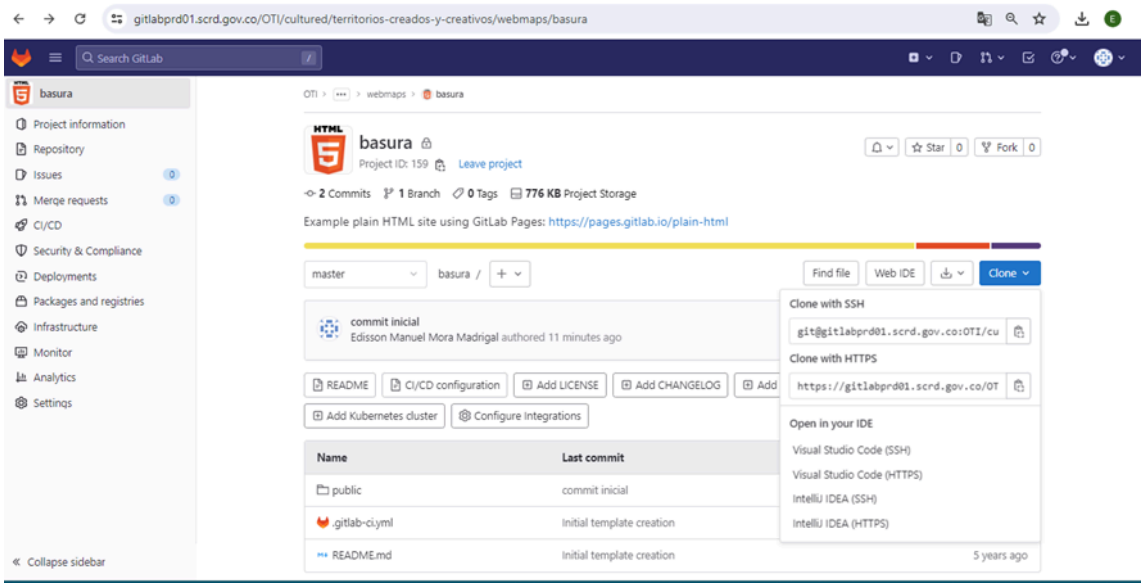


Se observan los proyectos compartidos:

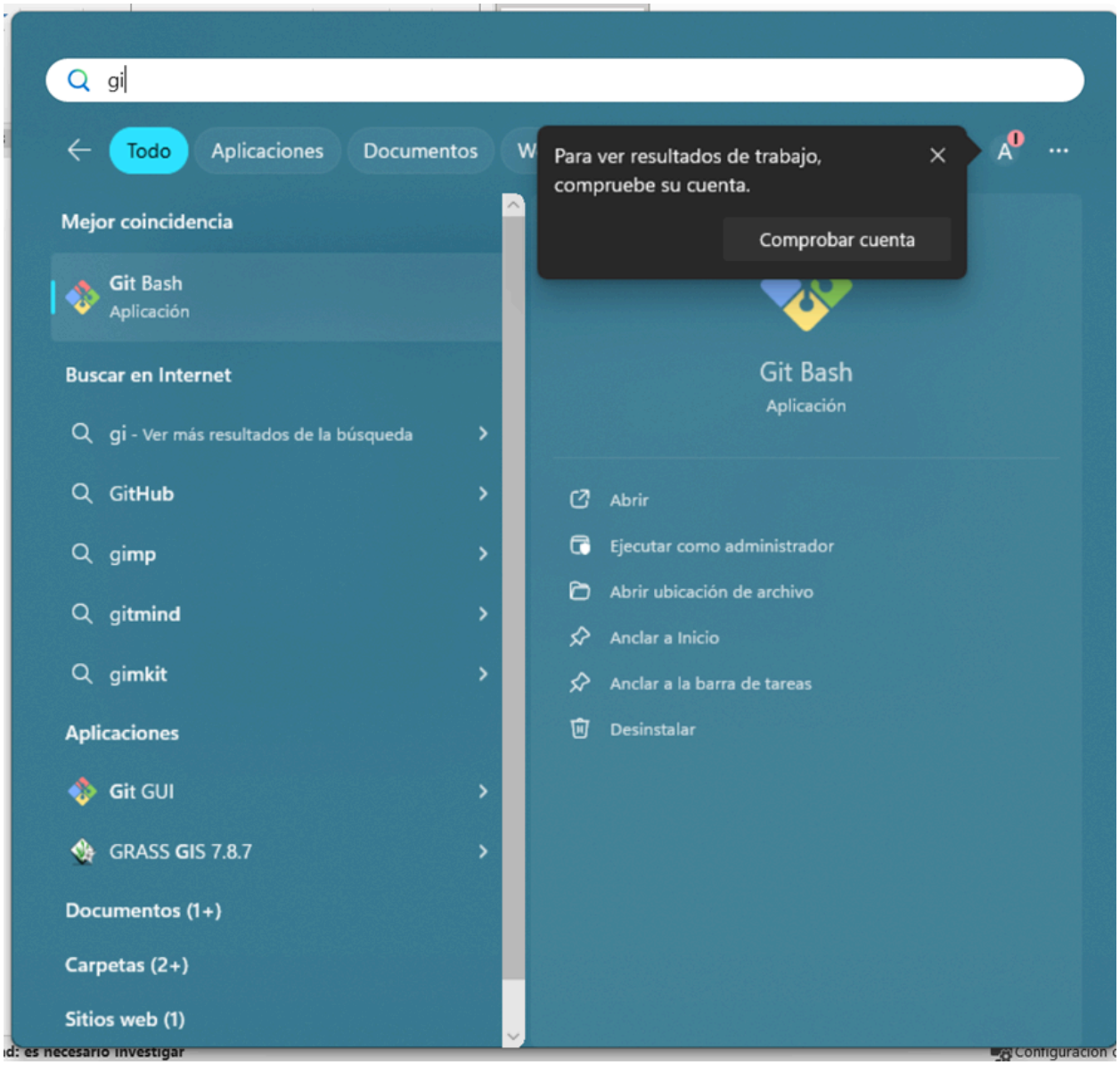


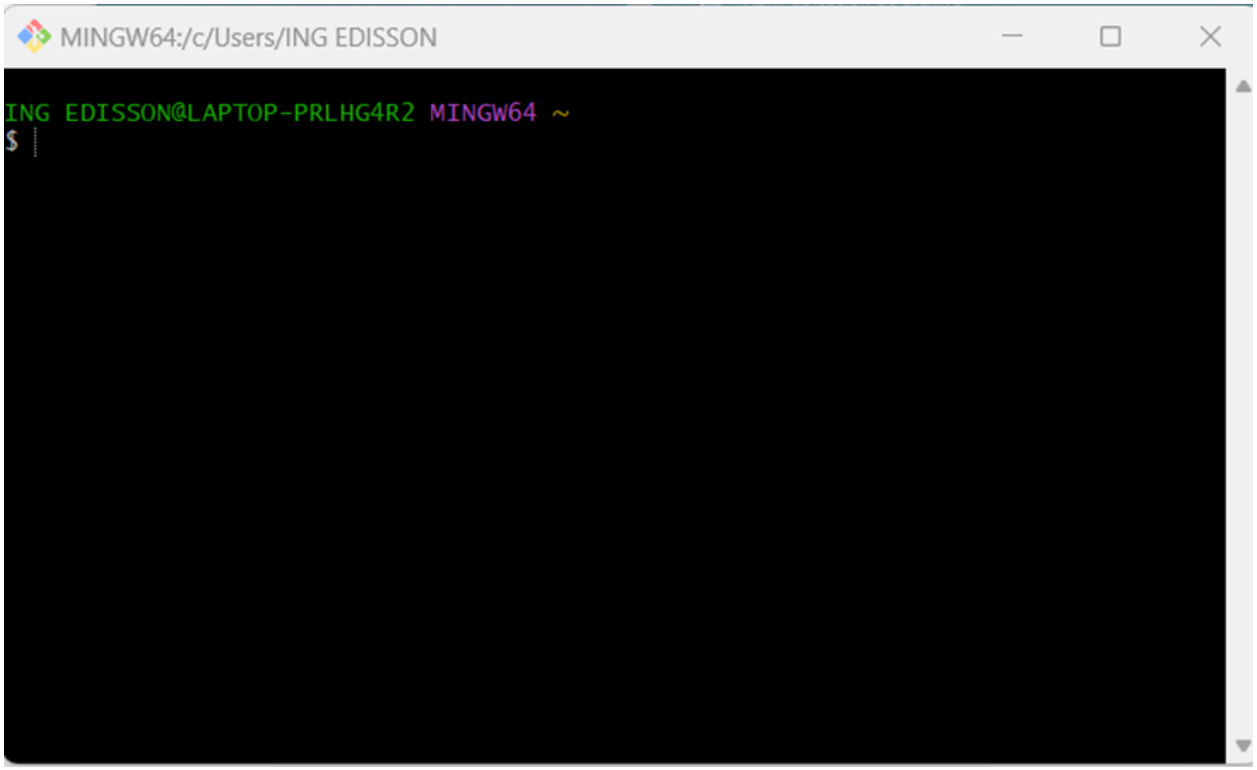
Se crea el repositorio y se clona por https en la terminal de gitbash:

git clone
https://gitlabprd01.scrd.gov.co/OTI/cultured/territorios-creados-y-creativos/webmaps/webmaps.git


gitlabprd01.scrd.gov.co/OTI/cultured/territorios-creados-y-creativos/webmaps/basura
Search GitLab
basura
Project information
Repository
Issues
Merge requests
CI/CD
Security & Compliance
Deployments
Packages and registries
Infrastructure
Monitor
Analytics
Settings
Collapse sidebar
OTI > webmaps > basura
basura
Project ID: 159
Leave project
Star 0
Fork 0
2 Commits
1 Branch
0 Tags
776 KB Project Storage
Example plain HTML site using GitLab Pages: https://pages.gitlab.io/plain-html
master
basura /
Find file
Web IDE
Clone
commit inicial
Edisson Manuel Mora Madrigal authored 11 minutes ago
Clone with SSH
git@gitlabprd01.scrd.gov.co:OTI/cu
Clone with HTTPS
https://gitlabprd01.scrd.gov.co/OT
README
CI/CD configuration
Add LICENSE
Add CHANGELOG
Add Kubernetes cluster
Configure Integrations
Open in your IDE
Visual Studio Code (SSH)
Visual Studio Code (HTTPS)
IntelliJ IDEA (SSH)
IntelliJ IDEA (HTTPS)
Name
Last commit
public
commit inicial
.gitlab-ci.yml
Initial template creation
README.md
Initial template creation
5 years ago

Posteriormente se abre en windows Gitbash y se abre la terminal:





Se añaden los cambios hechos o archivos añadidos desde windows con:

git add .

Y para cargar la información o empujar se usa el comando:

git push

Apunto a la carpeta que es:
C:\Users\ING EDISSON\Documents\SCRD\Proyectos_GIT

Se usa el código guardado en: C:\Users\ING EDISSON\Documents\SCRD\CONTRATO_430\gitlab

En la terminal se apunta a la carpeta anteriormente en la ruta, se clona el repositorio con la URL HTTPS, se apunta al repositorio o proyecto “basura”, se da git add, posteriormente se da commit inicial, puede pedir credenciales las cuales se indexan, nuevamente commit inicial con mensaje entre comillas dobles, y por último se empuja con git push origin master.

NOTA: Los pasos o script en la terminal están guardados en Manual Gitlab en el archivo de texto **pasos_terminal.txt**

Para crear el DNS o dirección URL a usar como iframe se debe enviar correo a soporte.sistemas@scrd.gov.co con el nombre del WebMap según estándar:

(Nombre Ubicación Geográfica)-WebMap(#Nombre Webmap)-Storymap-(Nombre del storymap si pertenece a storymap) - cultured.gov.co

El asunto debe ser Creación de DNS WebMap o StoryMap al correo mencionado de soporte sistemas de la SCRD.

Plantillas utilizadas como prueba para creación de producto mínimo viable de webmap para replicación de productos ESRI:

https://github.com/EdissonMora1234/Plantilla_Webmap_PMV_LEYENDA_Y_ATRIBUTOS_VERSION6

https://github.com/EdissonMora1234/Plantilla_Webmap_PMV_LEYENDA_Y_ATRIBUTOS_VERSION5

https://github.com/EdissonMora1234/Plantilla_Webmap_PMV_LEYENDA_Y_ATRIBUTOS_VERSION3

https://github.com/EdissonMora1234/Plantilla_Webmap_PMV_LEYENDA_Y_ATRIBUTOS_VERSION1

Las anteriores son las versiones necesarias para el despliegue de la plantilla final catalogada como producto mínimo viable.

2. Prueba de creación de Story Map mediante desarrollo de sitio web estático usando GITHUB/GITLAB

Este reporte detalla las lecciones aprendidas y proporciona una guía para la creación de un StoryMap en un sitio web estático utilizando GitHub Pages. La intención es simplificar el proceso para que cualquier persona, incluso con conocimientos básicos, pueda crear y alojar un StoryMap.

Lecciones Aprendidas

Durante la creación del StoryMap utilizando GitHub Pages, se aprendió la importancia de mantener una estructura de archivos organizada. Separar los recursos en carpetas específicas como `assets`, `css`, `js` y `lib` no solo facilita la gestión del proyecto sino que también hace que el código sea más legible y

manejable. Esto es esencial para proyectos colaborativos donde varios desarrolladores pueden estar trabajando simultáneamente.

Se descubrió que estructurar adecuadamente el HTML es clave para una navegación fluida. Utilizar elementos contenedores como `<div>` con identificadores únicos permite un control preciso del desplazamiento y la visualización de cada sección del StoryMap. Además, se demostró que es vital personalizar los estilos con CSS para mejorar la visualización de imágenes, videos e iframes dentro de las diapositivas, permitiendo que el contenido multimedia se adapte mejor a la presentación y mejore la experiencia del usuario.

El control del desplazamiento y la navegación a través del scroll del mouse fue otro aspecto crítico. Configurar adecuadamente la navegación vertical y limitarla a una sola barra de desplazamiento evita confusiones y mejora la usabilidad del StoryMap. Implementar botones de navegación, como el botón de "volver al inicio", también se identificó como una práctica beneficiosa para mejorar la accesibilidad y la experiencia general del usuario.

Se destacó la importancia de realizar pruebas continuas y de usar herramientas de desarrollo como Chrome DevTools para depurar y optimizar el código. Esta práctica es esencial para identificar errores y asegurar que el sitio web funcione correctamente en diferentes navegadores y dispositivos.

Otra lección clave fue la implementación de iframes para integrar StoryMaps y otros contenidos multimedia. Al trabajar con iframes, es importante ajustar sus dimensiones y asegurarse de que sean responsivos para garantizar una visualización adecuada en diferentes dispositivos y tamaños de pantalla.

Finalmente, se resaltó el valor de utilizar GitHub Pages como una plataforma gratuita y eficaz para alojar sitios web estáticos. Entender cómo configurar y desplegar un sitio en GitHub Pages es una habilidad valiosa que permite compartir y revisar rápidamente el trabajo en progreso.

Guía Paso a Paso para Crear un StoryMap en un Sitio Web Estático

Para comenzar, se debe configurar un repositorio en GitHub/Gitlab. Primero, se debe crear un nuevo repositorio en GitHub/Gitlab, asegurándose de seleccionar la opción para hacerlo público y agregar un archivo README. Luego, se debe clonar el repositorio en el entorno local utilizando Git Bash o cualquier terminal. Este paso inicial sienta las bases para todo el proyecto.

Una vez clonado el repositorio, es crucial crear una estructura de archivos bien organizada. En el directorio del repositorio, se deben crear carpetas como `assets` para imágenes y videos, `css` para hojas de estilo, `js` para archivos

JavaScript, y `lib` para bibliotecas adicionales. El archivo `index.html` debe colocarse en el directorio raíz. Dentro de `index.html`, se deben incluir enlaces a los archivos CSS y JavaScript y estructurar las diapositivas utilizando elementos `<div>` con clases e identificadores únicos.

Con la estructura básica en su lugar, se procede a diseñar y estilizar las diapositivas. Se deben definir estilos en el archivo `css/styles.css` para controlar la presentación de las diapositivas y el contenido multimedia. Por ejemplo, se puede usar CSS para ajustar el tamaño de imágenes, videos e iframes, asegurando que no ocupen más del 70% de la diapositiva y que mantengan una buena calidad visual. Además, se pueden aplicar estilos adicionales para agregar efectos como sombras y bordes redondeados.

La funcionalidad adicional se agrega mediante JavaScript en el archivo `js/script.js`. Aquí, se implementan scripts para manejar el desplazamiento entre diapositivas y actualizar la visualización del mapa según los parámetros definidos. Es importante asegurarse de que el desplazamiento sea suave y que la interfaz de usuario sea intuitiva.

Para integrar contenido multimedia y el StoryMap, se deben agregar imágenes, videos e iframes dentro de cada diapositiva utilizando etiquetas HTML. Se debe tener cuidado de aplicar clases o identificadores adecuados a estos elementos para permitir un control preciso sobre su estilo y presentación. Cuando se incorpora un StoryMap como un iframe, es fundamental ajustar su tamaño y responsividad para que se vea correctamente en cualquier dispositivo.

Después de completar la configuración y el diseño del sitio web, es esencial realizar pruebas locales para verificar la funcionalidad y la presentación del StoryMap. Abrir `index.html` en un navegador y utilizar herramientas de desarrollador ayudará a identificar y solucionar cualquier problema. Es recomendable ajustar los estilos y depurar el código según sea necesario para garantizar una experiencia de usuario fluida.

Finalmente, una vez que el sitio esté listo y probado localmente, se deben subir los cambios al repositorio de GitHub utilizando Git. Esto implica agregar los archivos al control de versiones, crear un commit con un mensaje descriptivo, y luego empujar los cambios al repositorio remoto. Después de esto, se puede configurar GitHub Pages para desplegar el sitio web. Esto se realiza en la configuración del repositorio en GitHub, seleccionando la opción "Pages" y eligiendo la rama de despliegue adecuada, generalmente `main` o `gh-pages`.

Con GitHub Pages activado, se debe verificar la URL proporcionada para asegurarse de que el sitio web se despliega correctamente y que todas las funciones del StoryMap están operativas.

Plantillas de storymaps creadas:

Los storymaps se realizarán según las plantillas:

1. Plantilla_Storymap_Centrado_Imagenes
2. Plantilla_MapaDiapos_Con_Leyenda_Storymap
3. Plantilla_mapa_diapos_Storymap
4. Plantilla_generalStorymaps

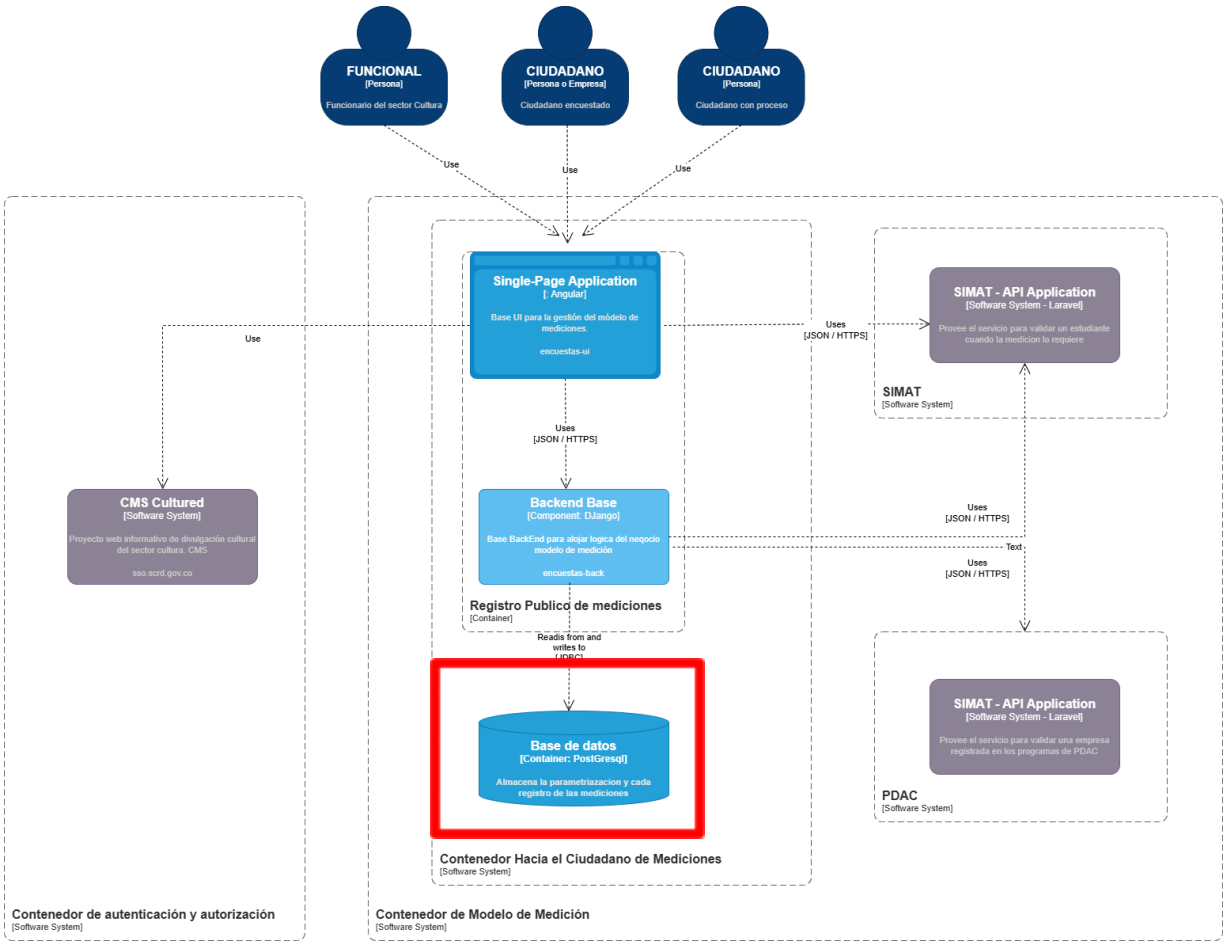
Con las anteriores plantillas se pueden recrear narrativas dinámicas utilizando información geográfica.

Conclusión

La creación de un StoryMap en un sitio web estático usando GitHub Pages requiere atención al detalle y una buena planificación. La clave está en mantener una estructura de archivos organizada, escribir código limpio y eficiente, y realizar pruebas continuas. Con estas prácticas, se puede asegurar una experiencia de usuario atractiva y funcional, facilitando la creación de StoryMaps para una amplia variedad de aplicaciones.

3. Prueba con desarrollo Formulario SIG mediante aplicativo suministrado por equipo de la Dirección Observatorio y Gestión del Conocimiento Cultural

Los formularios SIG fueron creados en colaboración con la Celula de Análitica a cargo del ingeniero Mauricio Ojeda Pepinosa <javier.ojeda@scrd.gov.co>. El desarrollo tiene la siguiente arquitectura principal:



El aplicativo crea formularios con preguntas que se crean por secciones y se van uniendo hasta obtener el formulario deseado, los formularios reciben archivos multimedia e información geográfica.

Inicialmente el formulario es configurado a partir de un dataset con atributos definidos que irá alimentando o actualizando el mismo formulario mediante se diligencia una y otra vez. La información de cada diligenciamiento se guarda en forma de filas de un dataset en formato SQL (base de datos relacional resaltada en rojo en la anterior imagen), los formularios pueden conectarse a la base de datos de Cultured la cual contiene todas las capas geográficas y datasets de la SCRD, los acceso a esta son:

Usuario	**Contraseña**	**bd**	**ip**	**puerto**

admimgeoportal	T3sTGpCulT	qa_cr_mods_geoportal	172.16.10.71	5432

Esta es la base de datos que contiene las capas con estilos en geoserver, por tanto este desarrollo permite replicar los formularios de Survey 123 de la licencia de Arcgis Online. Cabe mencionar que aún está pendiente el desarrollo de preguntas dependientes y carga masiva de contenido multimedia archivos/fotos, también está pendiente el manual de usuario para terminar de consolidar la documentación del uso del aplicativo Formularios SIG con enfoque en información geográfica.

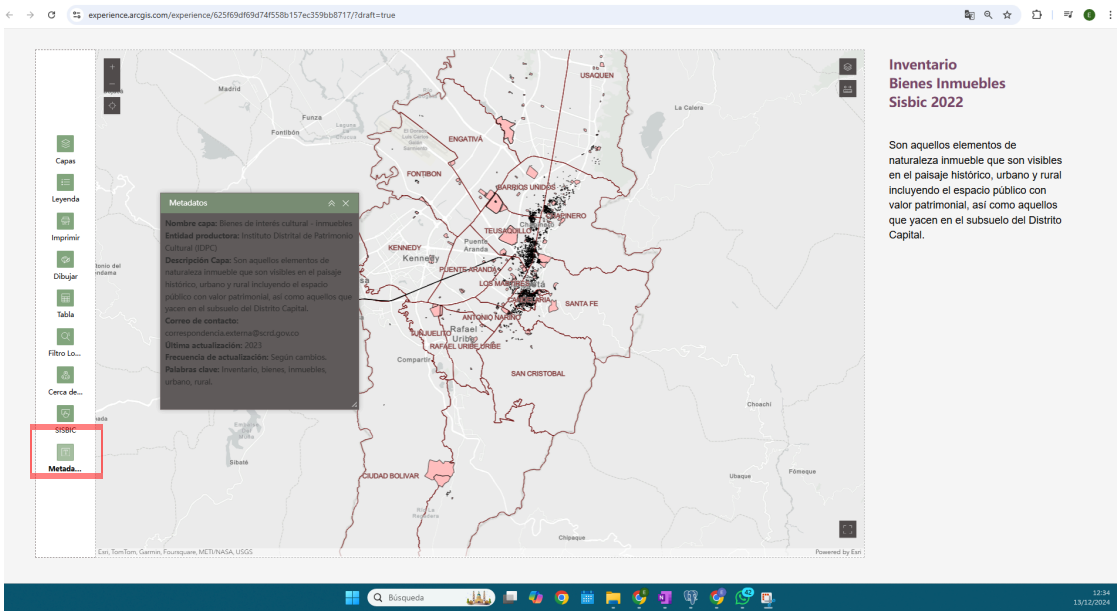
La creación de los formularios SIG se realizan en la página web que contiene el aplicativo: https://dev-analitica-mediciones-ui-cultured.scrd.gov.co/

Está pendiente hacer una prueba de creación de tabla postgreSQL + postGIS, con estilo definido en geoserver y servicio OGC y conectarla al aplicativo para actualización mediante diligenciamiento de formulario, adicional se debe probar migración de tabla ya creada de forma masiva y actualización en misma tabla de postgres.

4. Prueba creación de widget metadatos en Arcgis Online para los 35 experience builder publicados en Cultured-maps

Paso a paso para crear y configurar un widget "Metadatos" en ArcGIS Experience Builder

El widget "Metadatos" permite desplegar información descriptiva de las capas publicadas en ArcGIS Online dentro de una aplicación creada en Experience Builder. Este proceso incluye la creación, configuración y vinculación del widget con los atributos de las capas geográficas.



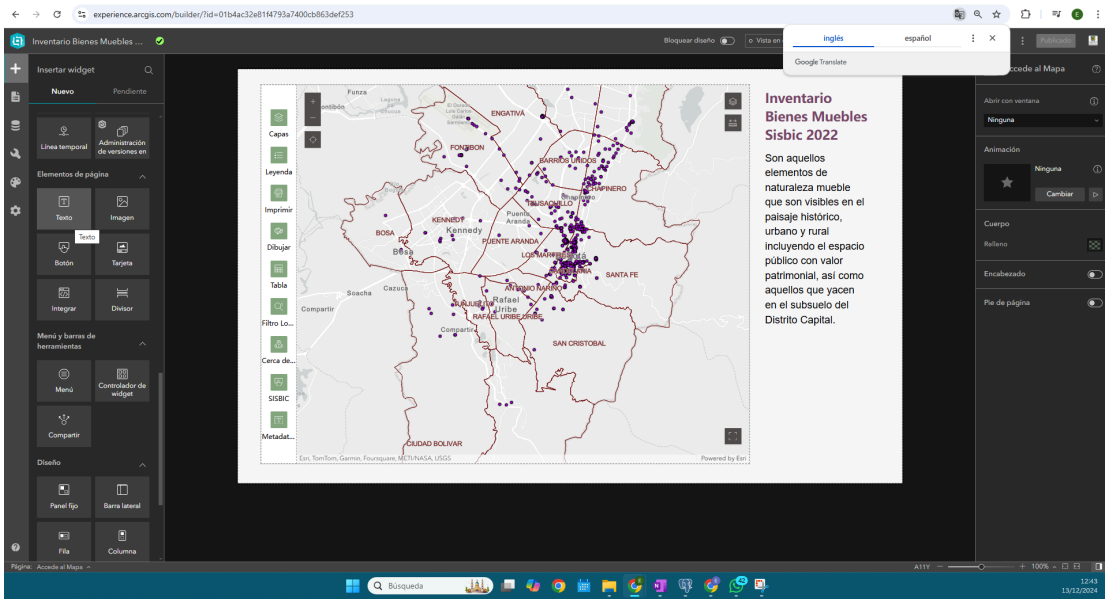
1. Preparación de los datos

Publicar las capas geográficas:
Asegúrese de que las capas geográficas a incluir en su Experience Builder estén publicadas en ArcGIS Online con los metadatos correctamente configurados y alojados en excel Resumen_capas_Cultured_2024.xlsx.

Verificar atributos:
Incluya en los metadatos de las capas información como el nombre de la capa, descripción, palabras clave, contacto, y otros detalles relevantes que se visualizarán en el widget.

2. Crear la aplicación en Experience Builder
Iniciar sesión en ArcGIS Online:
Acceda con su cuenta en ArcGIS Online de la SCRD.

Abrir Experience Builder:
Desde el menú principal, abra el entorno de Experience Builder. En la imagen se muestra el widget texto con el cual se mostraran los metadatos de la tabla excel mencionada anteriormente.



Crear un nuevo proyecto:
Seleccione "Crear nuevo proyecto" y elija una plantilla básica o personalizada según sus necesidades.

3. Añadir el widget "Metadatos"
Seleccionar el widget de texto:
En el panel de widgets, arrastre el widget de Texto y colóquelo en el lugar deseado dentro de la aplicación.

Configurar el widget de texto:
Haga clic en el widget para abrir el panel de configuración.
Personalice el diseño del texto (tamaño, estilo, color).
Escriba un título, por ejemplo: Metadatos de la Capa.

4. Configurar la interacción con las capas
Vincular el widget con datos:
Seleccione el botón Datos en el panel de configuración del widget de texto.
Conecte el widget al mapa web o a las capas de datos publicadas en ArcGIS Online.

Agregar expresiones dinámicas:
Inserte campos dinámicos que extraigan automáticamente los valores de los metadatos de cada capa.

Ejemplo:
css
Copiar código
Nombre capa: {layerName}

Entidad productora: {organization}
Descripción: {description}
Contacto: {contactEmail}
Última actualización: {lastUpdated}
Palabras clave: {keywords}

5. Estilizar el widget
Formato del texto:
Utilice el editor de texto para resaltar los títulos en negrita.
Ajuste el tamaño y el color para que sea legible.
Añadir un fondo:
En el panel de configuración, elija un color o imagen de fondo para el widget, si es necesario.

6. Probar y ajustar
Vista previa de la aplicación:
Use la opción Vista previa para comprobar cómo se despliegan los metadatos al interactuar con las capas.

Ajustes adicionales:
Modifique el diseño y la configuración si algún atributo no se visualiza correctamente.

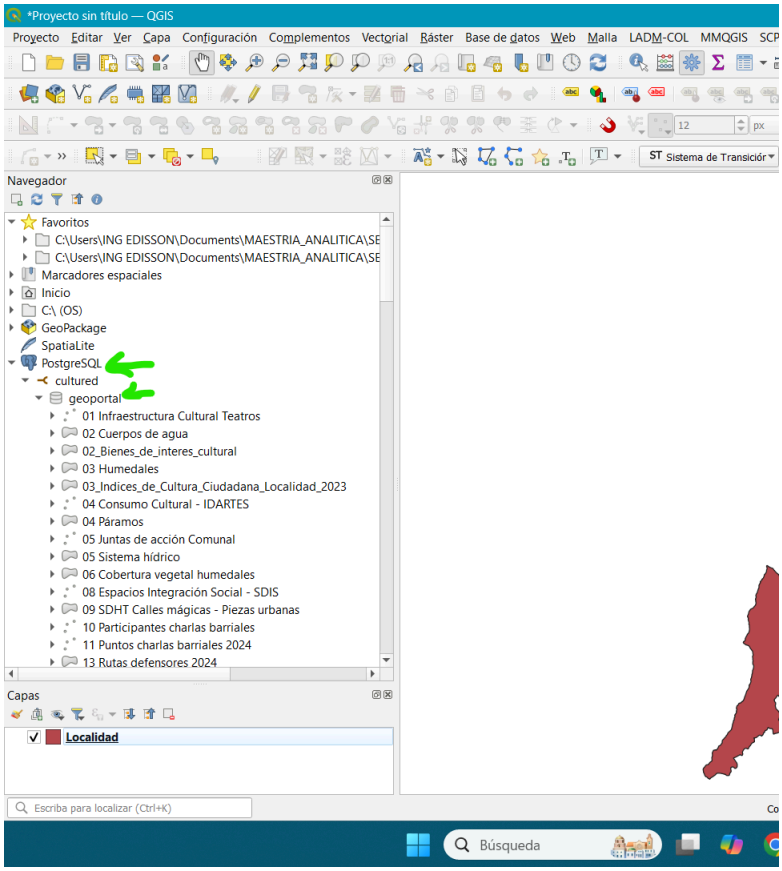
7. Publicar la aplicación
Guardar el proyecto:
Haga clic en Guardar para almacenar los cambios.

Publicar la experiencia:
Presione el botón Publicar para que la aplicación sea accesible a los usuarios finales.

Proceso para actualizar nombres de columnas en capas WMS publicadas desde GeoServer con base en PostgreSQL + PostGIS

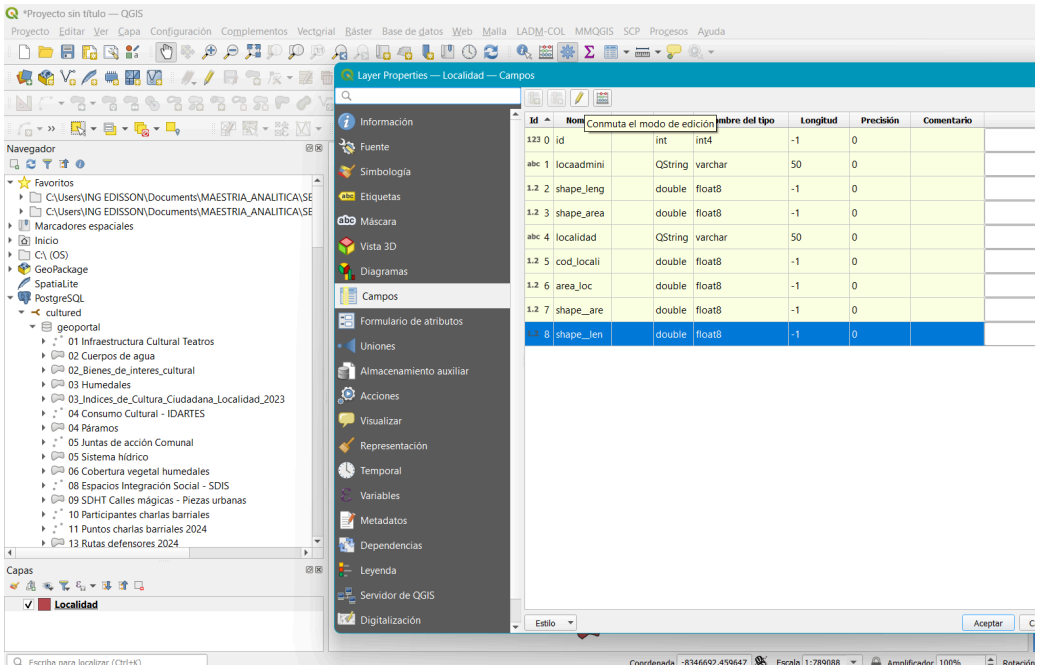
Para realizar la actualización de nombres de columnas en capas geográficas que están publicadas como servicios WMS desde GeoServer, conectadas a bases de datos PostgreSQL con extensión espacial PostGIS, se siguió el siguiente procedimiento técnico, utilizando QGIS como herramienta principal de edición:

1. **Conexión a la base de datos desde QGIS**
   Desde QGIS Desktop, se establece la conexión a la base de datos PostgreSQL/PostGIS correspondiente, utilizando las credenciales y parámetros de acceso autorizados.



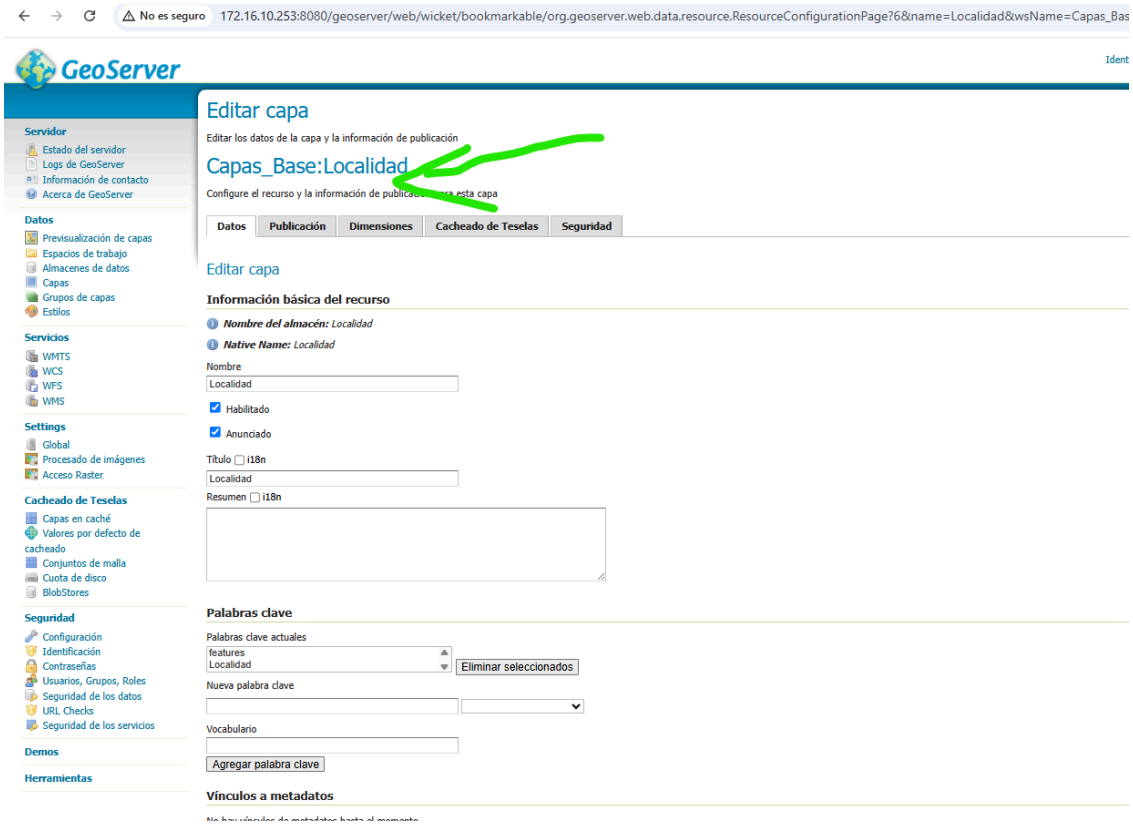
2. **Carga de la capa a modificar**
   Se carga en QGIS la capa geográfica cuyo esquema (estructura de columnas) requiere modificación.

3. **Edición del nombre de las columnas**
   Una vez cargada la capa, se accede a:
   `Propiedades de la capa > Campos`
   Desde allí, se seleccionan los campos (columnas) que se desean renombrar y se aplican los cambios de forma directa. Esta acción actualiza el esquema de la tabla en la base de datos.

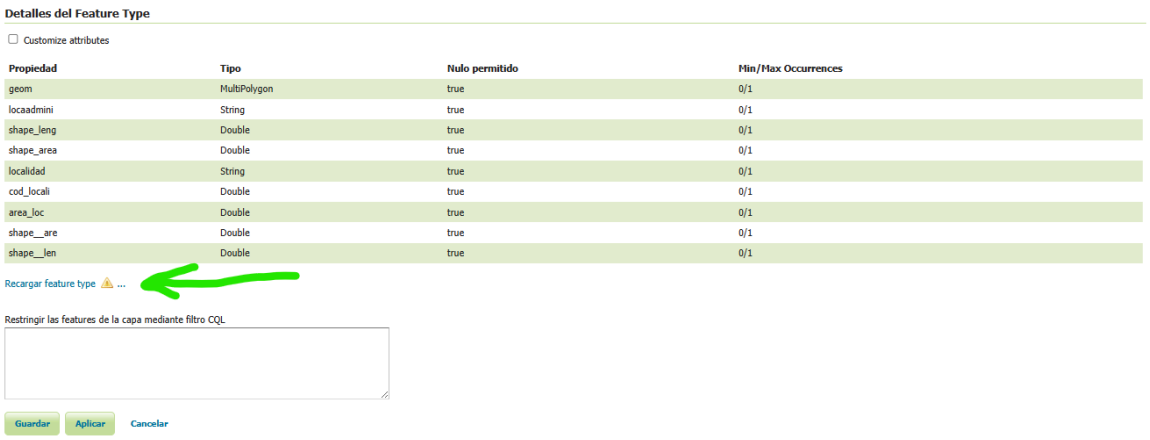


4. **Actualización en GeoServer**
   Luego, se ingresa a GeoServer y se accede al panel de administración de capas. Para cada capa modificada:

   - Se navega a **Capa > Feature Type > Detalles**.
   - Se actualizan manualmente los nombres de los atributos para reflejar los nuevos nombres de columna definidos en la base de datos.



Y específicamente se ajustan los atributos en la sección de la siguiente imagen:



5. **Consideraciones con los estilos (SLD)**
   Si el atributo renombrado es utilizado como criterio en un estilo (SLD), es **necesario actualizar también el archivo del estilo** para que no genere errores. El procedimiento es:

- Descargar el archivo SLD o copiar su contenido desde el editor de GeoServer.
- Abrirlo en un editor de texto (por ejemplo, Bloc de notas).
- Reemplazar todas las ocurrencias del nombre anterior del atributo por el nuevo nombre.
- Guardar el archivo y cargarlo nuevamente en GeoServer.

6. **Recomendación antes de cambiar nombres de columnas vinculadas a estilos**
   Para evitar errores en la visualización de las capas, se recomienda **desconectar temporalmente el estilo SLD personalizado** en GeoServer y asignar el estilo por defecto correspondiente (`point`, `line` o `polygon`) antes de realizar el cambio de nombre en los atributos. Esto previene fallos por referencias inválidas durante la actualización.

Preparó	**Edisson Manuel Mora Madrigal** edisson.mora@scrd.gov.co	30/06/2023