Propuesta de Cultured-Maps implementado con código abierto
02/2024

Contenido

Contenido 1
Introducción 1
Marco de Referencia 2
Componentes del sistema 3
1. Repositorio SIG 4
2. Publicación de Mapas 5
3. Historias en Mapas 8
4. Geo codificación de direcciones 9
Etapas de migración del sistema de información geográfico SIG 12
Desarrollos en software libre para visores geográficos por entidades adscritas 15
Facilidades de Laravel y PostgreSQL para un Geovisor: 16

Introducción

En este documento se presenta la propuesta de migración del geo visor *Cultured-maps* administrado por la Secretaría de Cultura, Recreación y Deporte **SCRD** a una estructura de software libre. La propuesta inicialmente está orientada en analizar los beneficios y los desafíos que puede tener dicha migración, paralelamente la propuesta contiene la metodología que en siete etapas permite implementar los sistemas de información geográfica SIG en una plataforma de código abierto, los siete pasos son; Evaluación, Diseño, Desarrollo, Pruebas, Implementación, Capacitación y documentación, Soporte y mantenimiento.

En cuanto a los beneficios de la migración a código abierto en el contexto actual, se catalogan de la siguiente manera; se evitan costos de licenciamiento, se tiene más independencia y control, mayor transparencia y seguridad, mejor interoperabilidad de la información (sin límite de licencias de usuarios), menor riesgo de desactualización de software y mayor desarrollo personalizado e innovador según necesidades específicas, además, el desarrollo comunitario es un plus importante ya que entidades adscritas a la

SCRD han hecho a la fecha importantes avances en el manejo de software libre siguiendo la promoción gubernamental de fomentar la Iniciativa Nacional de Software Libre y tener en cuenta lineamientos como el Acuerdo 279 de 2007.

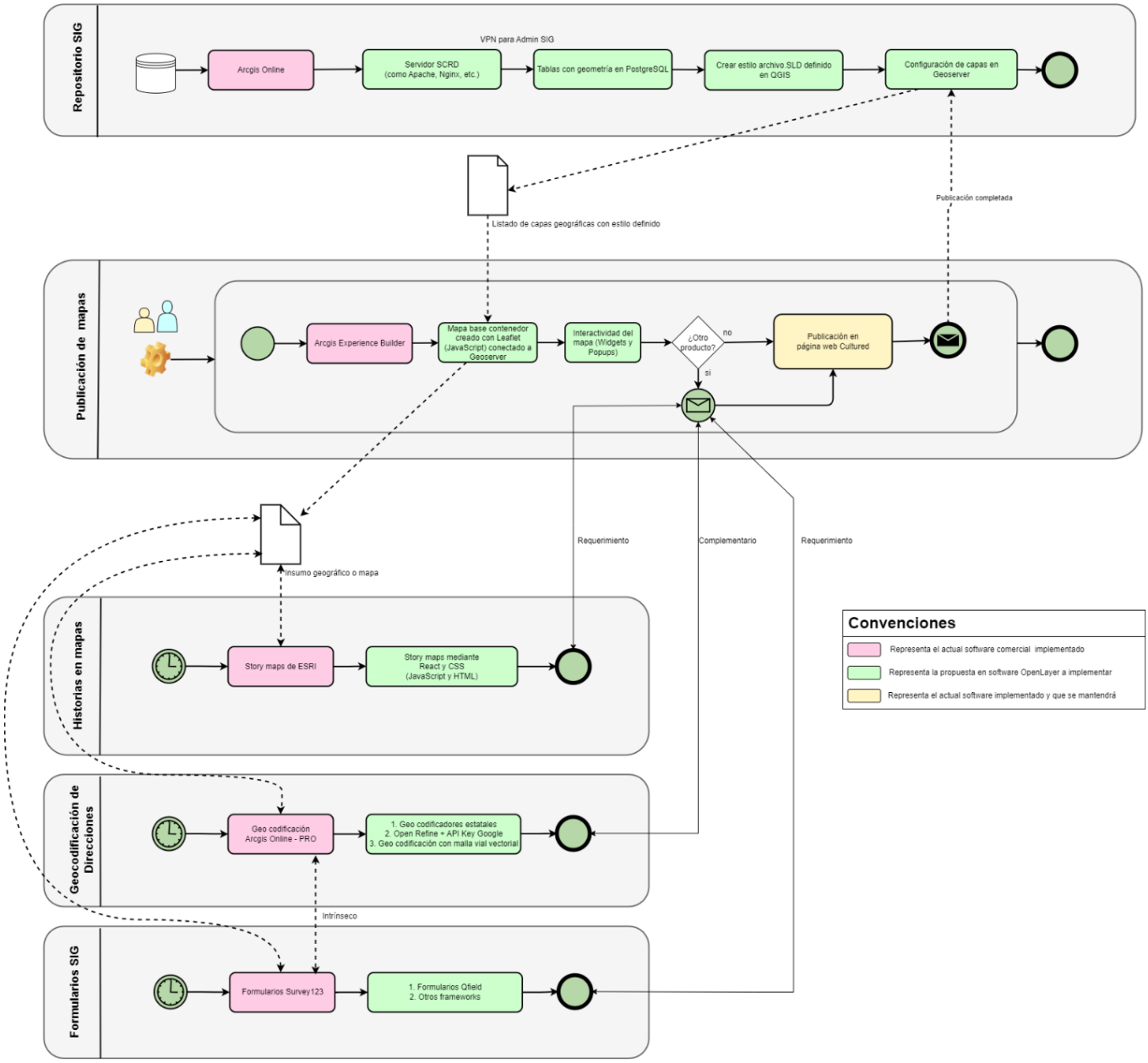
Por otra parte, los desafíos en esta implementación de software libre en el contexto actual son inicialmente tres; curva de aprendizaje de usuarios, tiempo de transición desde el software propietario, disponibilidad de aplicaciones o desarrollos específicos. Cabe resaltar que estos desafíos tienen una solución propuesta concertada de forma interdisciplinar en la entidad SCRD.

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 SCRD, el proyecto tiene como nombre “Cultured - maps", 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, de una forma interoperable, estandarizada y que en esfuerzo conjunto permita tener la autonomía suficiente para gestionar la información sin depender de formatos propietario o terceros.

Actualmente la gestión de la información geográfica se realiza a través del software propietario Arcgis Online de la compañía ESRI, para la migración del sistema de información geográfico SIG a softwares de código abierto es necesario que la propuesta tenga en cuenta avances de desarrollo libre por parte de entidades Adscritas a la **SCRD** como la implementación por parte del Instituto Distrital de Patrimonio Cultural **IDPC** en visor geográfico **SISBIC** con software de código abierto que implementa lenguajes PHP y Javascript (librería Leaflet) mediante el framework Laravel 8, como servidor geográfico implementan Arcgis Server lo que en la propuesta actual de la SCRD esto cambia por un servidor geográfico libre que puede ser “Geoserver” o “MapServer”, la arquitectura del visor SISBIC se muestra más adelante en “Desarrollos en software libre para visores geográficos por entidades adscritas”.

Por otra parte, también se tiene a la fecha importantes avances de un geo codificador de direcciones libre que no requiere de licencias o “API keys” privadas, dicho desarrollo en geo codificación se ha trabajado desde el instituto **IDARTES** desarrollo que sigue en marcha actualmente y es parte fundamental en la creación de Formularios que permitan el diligenciamiento de SIG, de Historias en Mapas o en general de requerimientos de actualización de información geográfica según lineamientos establecidos en diagrama de flujo del proyecto OpenLayer el cual se presenta a continuación:



Componentes del sistema

En el marco del diagrama de flujo del proyecto OpenLayer se observa el flujo que el proceso de migración SIG debe tener en cuenta según cinco componentes principales:

1. Repositorio SIG
2. Publicación de Mapas
3. Historias en mapas

4. Geo codificación de direcciones
5. Formularios SIG

A continuación, se describirán los recursos necesarios, el tiempo y la propuesta general de despliegue para funcionamiento. Adicionalmente se tendrán en cuenta desarrollos nacionales e internacionales en cuanto al tema del Geovisor OpenLayer de Cultura.

1. Repositorio SIG

Actualmente el repositorio implementado es Arcgis Online, para migrar la información geográfica que allí reposa es necesario:

1. Uso de Servidor SCRD, gestionar el contenedor de servlets para lo cual se propone Apache Tomcat que es el contenedor de servlets más popular y usado, y ofrece una buena compatibilidad con los estándares de Java EE. La última versión estable es la 10.0.14. En resumen este viene a ser el Web Server.

Nota: Un contenedor de servlets es un componente que se encarga de gestionar las aplicaciones web escritas en Java, que se componen de servlets, páginas JSP, filtros, etc. Un contenedor de servlets se puede ejecutar de forma independiente, o se puede integrar con un servidor web, como Apache HTTP Server, Nginx, etc.

2. Descarga de PostgreSQL se propone que sea en versión 14 con extensión PostGIS versión 3.1 en donde se creará la base de datos espacial o las versiones de ambos más actuales y estables.

3. Creación de estilo definido en formato .SLD creado desde software QGIS versión 3.24.3 para cada capa incluida en la base de datos espacial SQL y recopilada en GeoServer en su versión 2.20.2.

4. Configuración de geoserver y de capas en servidor según formato predefinido para publicación de servicios OGC tipo WMS y WFS.

5. Inventario de capas con título, descripción y dirección de servicio WMS/WFS/OTRO manteniendo estructura Cultured-maps

Nota: Se propone **Geoserver** como servidor geoespacial. Otra opción puede ser **MapServer**.

2. Publicación de Mapas

La publicación actualmente depende de Arcgis Online en donde reposan las capas en sus diferentes formatos y que son publicadas en el portal Cultured mediante la aplicación Arcgis Experience Builder. Para publicar la información geográfica en su versión de código abierto es necesario lo siguiente:

1. Tener previamente las capas configuradas ya en **Geoserver** con estilo .sld definido y con la dirección url para consumir el servicio OGC.
2. Descargar e instalar la biblioteca de javascript denominada **Leaflet** en su versión 1.7.1.
3. Configurar el mapa con Leaflet y conectar a geoserver
4. Creación de widgets de interactividad del mapa específicamente filtros espaciales y alfanuméricos (“Cerca de mí” y “Filtro Localidad”)
5. Configuración de mapa en ventana de Cultured-Maps

Nota: Se propone **Leaflet** como biblioteca javascript de código abierto para mapas interactivos. Otra opción puede ser **OpenLayers.**

El desarrollo de widgets personalizados ha sido de interés anteriormente al momento de mostrar la información a los usuarios finales, estos permiten ordenar la información según su naturaleza o necesidades y mostrarla de una forma más dinamica según lo que necesite especificamente el usuario, filtrando la información, seleccionándola discriminadamente o simplemente si el usuario necesita dibujar algo en el mapa o insertar información geográfica de su propiedad para realizar análisis.

En base a lo anterior y con las herramientas actuales se avanzó en el desarrollo de estos widgets personalizados en la aplicación de ESRI denominada Arcgis Experience Builder como se evidencia en Documentación metodológica de código fuente por parte de la Oficina de Tecnología de la Información OTI – SCRD. En síntesis, esta aplicación la cual implementa librerías propias de ESRI tiene una documentación limitada, el soporte por parte de la compañía no tiene cobertura para estos requerimientos de desarrollo específicos.

El **widget contenedor de capas** se puede observar en el Visor Geográfico Ambiental de Bogotá D.C en donde se observa el resultado deseado como se muestra a continuación:

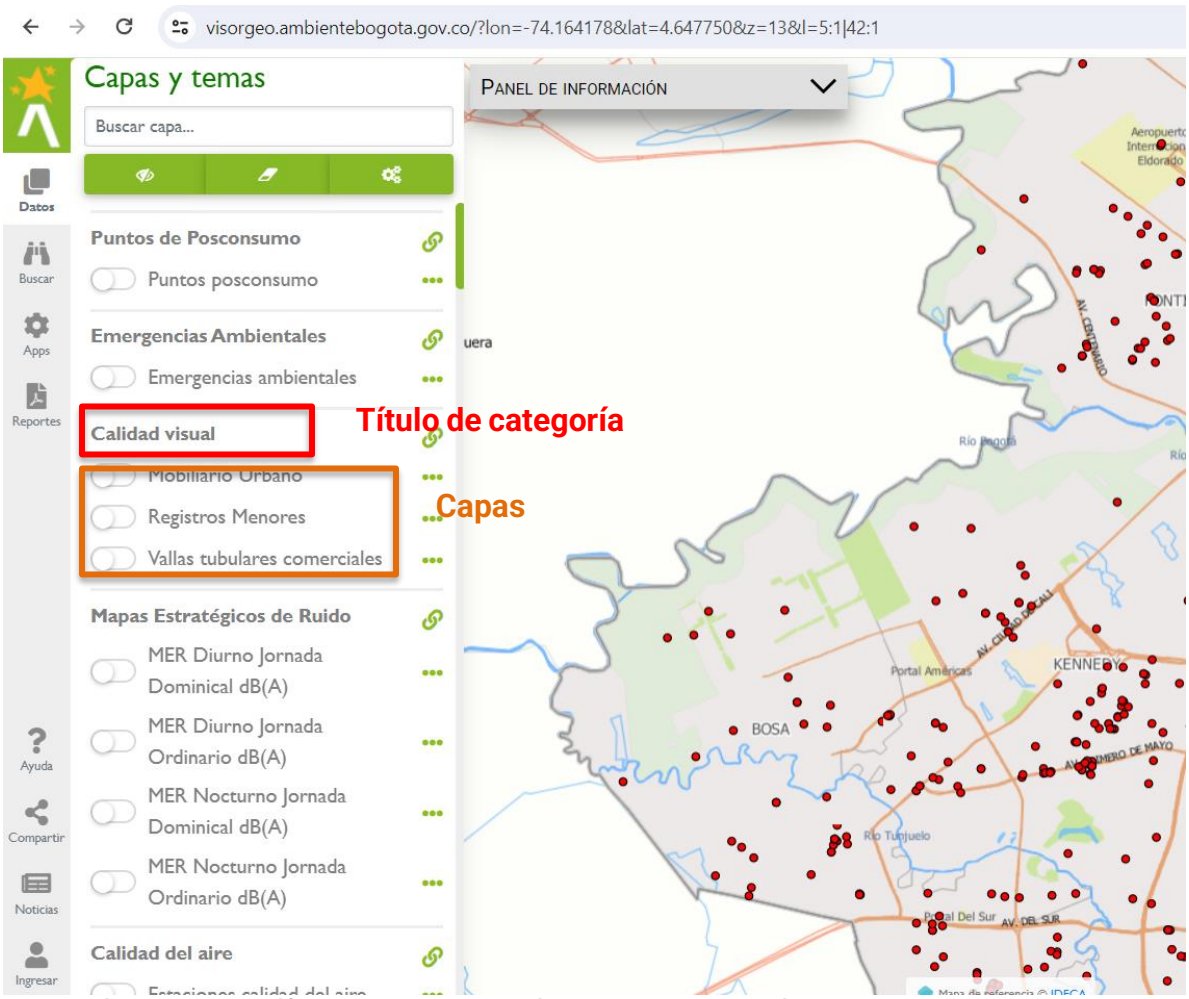


Ilustración 1: Visor geográfico Ambiental de Bogotá, implementando con código abierto según Manual de Usuario

Nota: ver Manual de Usuario del visor geográfico ambiental de Bogotá D.C.
Por otra parte, con el objetivo de visualizar en visores geográficos OpenLayer el **widget "Cerca de mí"**, se observa el geovisor de Cataluña, el cual tiene filtros espaciales y alfanuméricos como los que se necesitan, se muestran a continuación:

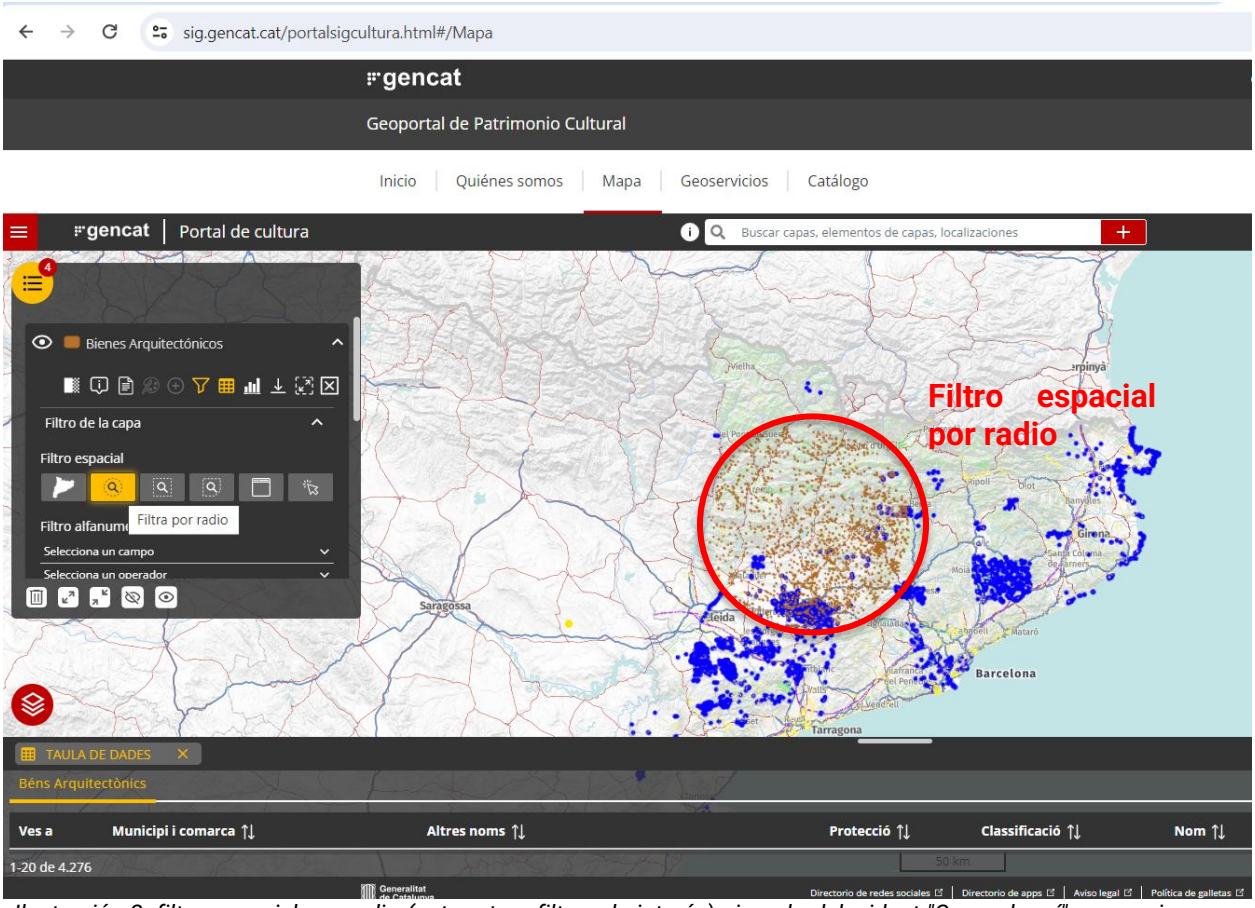


Ilustración 2: filtro espacial por radio (entre otros filtros de interés) ejemplo del widget "Cerca de mí" necesario para Cultured-maps y desarrollado en el [Portal de Cultura de Cataluña](#).

Como se observa en la anterior imagen se filtró la capa “Bienes Arquitectónicos” (color café) en un radio determinado. Otras funcionalidades de este visor geográfico se observan en: [Guía del Geoportal de Cultura](#).

En cuanto a contenido general también es llamativo por parte de la OTI el geovisor cultural de Perú como se muestra a continuación:

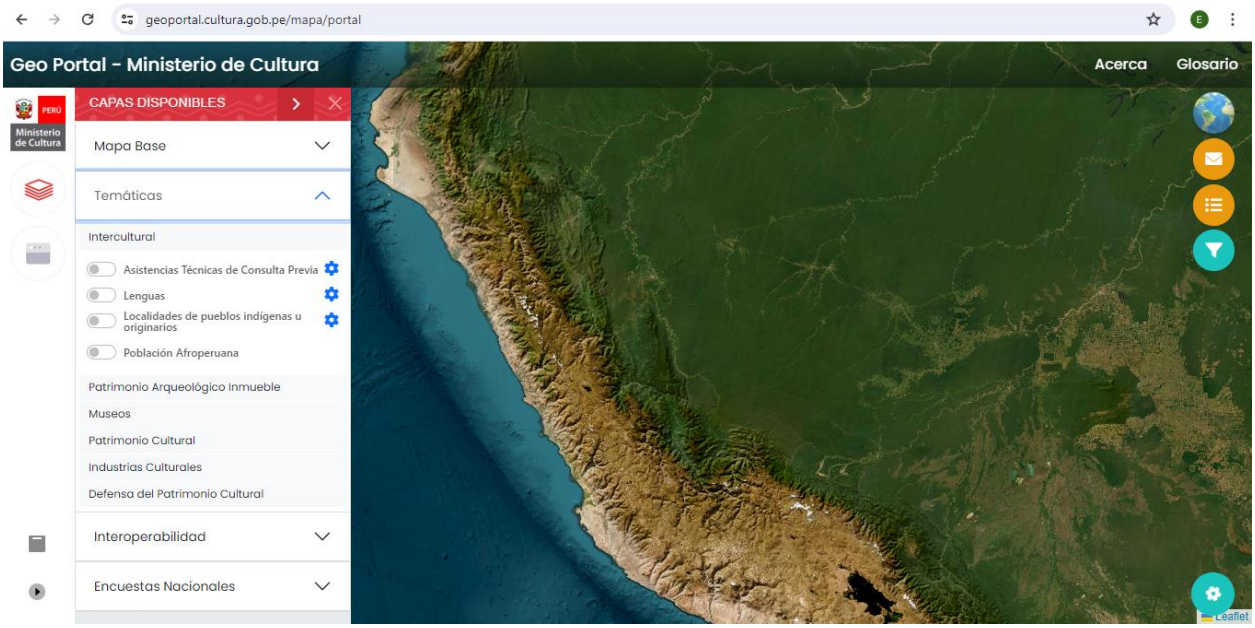


Ilustración 3: Geoportal cultura de Perú. Fuente: https://geoportal.cultura.gob.pe/mapa/portal

El visor geográfico para Cultura de Perú esta creado con la librería de Java Script propuesta denominada Leaflet.

3. Historias en Mapas

Actualmente para la publicación de historias en mapas se implementa los StoryMaps de ESRI, para el desarrollo en código abierto es necesario crear ciertas plantillas establecidas y manuales para edición puesto que estos "Story maps" son requerimientos recurrentes y cambiantes. Lo principal a tener en cuenta en estas plantillas es la relación de la información geográfica y la alfanumérica, la forma en como se imprime, los tamaños y los estilos de los textos.

Para el desarrollo se debe revisar en mesas de trabajo interdisciplinares:

1. Framework a implementar
2. Uso de librerías Javascript para mapas interactivos como Leaflet
3. Relación de información geográfica y textos alfanuméricos de la historia, relación directa
4. Establecimiento de manuales de edición para personalización o en su defecto plantillas para recreación de historias estandarizadas.

Nota: se necesita conocimiento en PHP, HTML y CSS además de experiencia en React o Leaftlet (bibliotecas Javascript para mapas interactivos).

4. Geo codificación de direcciones

La geo codificación de direcciones habitualmente surge según las siguientes necesidades:

1. En los formularios SIG es necesaria la **geo codificación puntual** para convertir una dirección (un texto con estructura de nomenclatura vial) en un punto con atributos predefinidos sobre un mapa, los atributos son información diligenciada en un formulario tipo Google forms. Una vez se diligencia el formulario al estar este conectado con una capa geográfica lo que hace es crear un punto u objeto de más en dicha capa y actualizarla.

   Para la **geo codificación puntual** se propone implementar un desarrollo como el geolocalizador Pandora Geolocalizador IDARTES o generar un desarrollo propio similar en un aplicativo utilizando por ejemplo una API KEY gratuita como la que suministra IDECA lo cual es fundamental para relaización de Formularios de diligenciamiento SIG.

2. La **geo codificación masiva** de direcciones, esta se propone realizarse a partir de proceso OpenRefine mediante API key de Google gratuita, o también desde un geocodificador estatal, por ejemplo el de IDECA y su API KEY.

5. Formularios SIG

Actualmente los formularios que se manejan para actualización de capas geográficas son los de **Survey Connect de ESRI**, para reemplazar esta herramienta por una OpenLayer es necesario tener en cuenta que se necesita de una geo codificación puntual, es decir, se necesita una actualización en tiempo en real y también un formulario tipo google forms que sea amigable con el usuario y que permita el envío de varios formularios a la vez.

A continuación, se observa el tipo de diligenciamiento necesario:

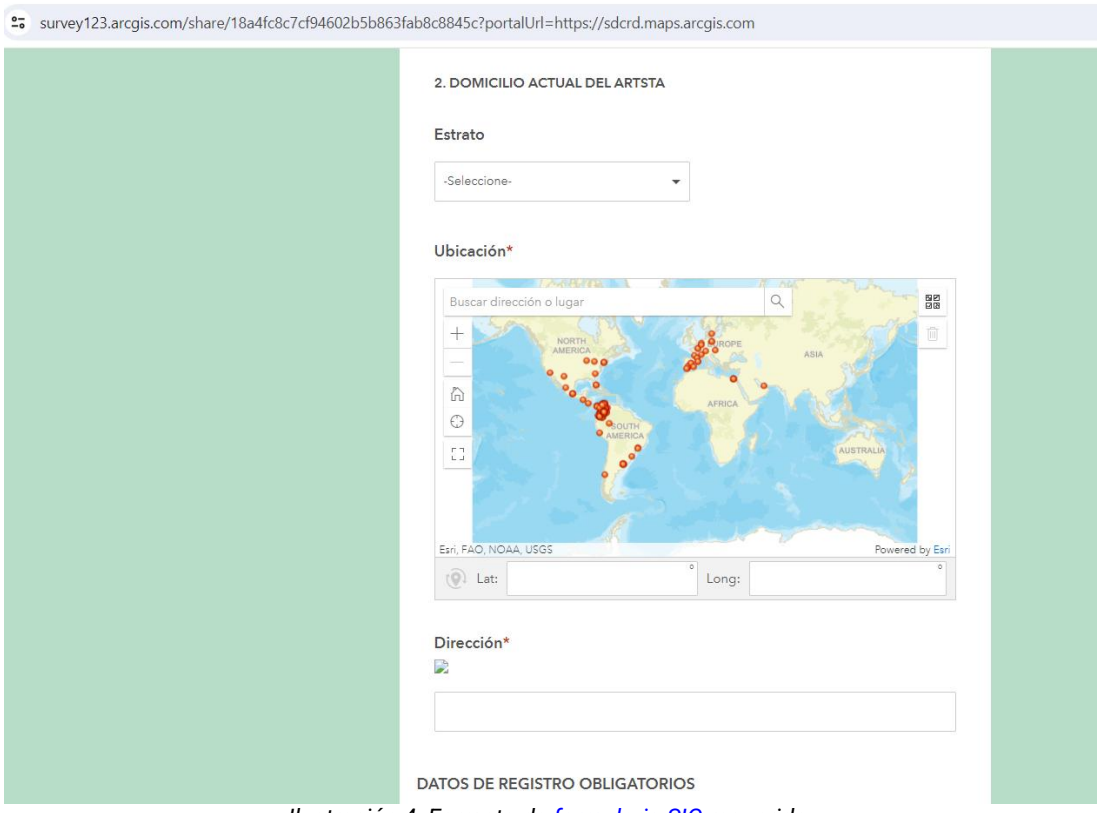


Ilustración 4: Formato de formulario SIG requerido.

1. Se propone revisar aplicación Qfield Cloud para implementación o un desarrollo web propio haciendo uso de librerías javascript para mapas dinámicos como como Leaflet o OpenLayer.

2. Se debe revisar desarrollo pendiente por parte de la SCRD en cuanto a estos formularios SIG.

6. Consideraciones generales de la infraestructura del visor geográfico

La arquitectura del backend se puede implementar como se muestra a continuación, partiendo de una base de datos espacial que hay que construir y gestionar desde Geoserver para publicarla en la web según los estándares OGC, esta información geográfica será editada y se le hará mantenimiento desde la aplicación QGIS de escritorio:

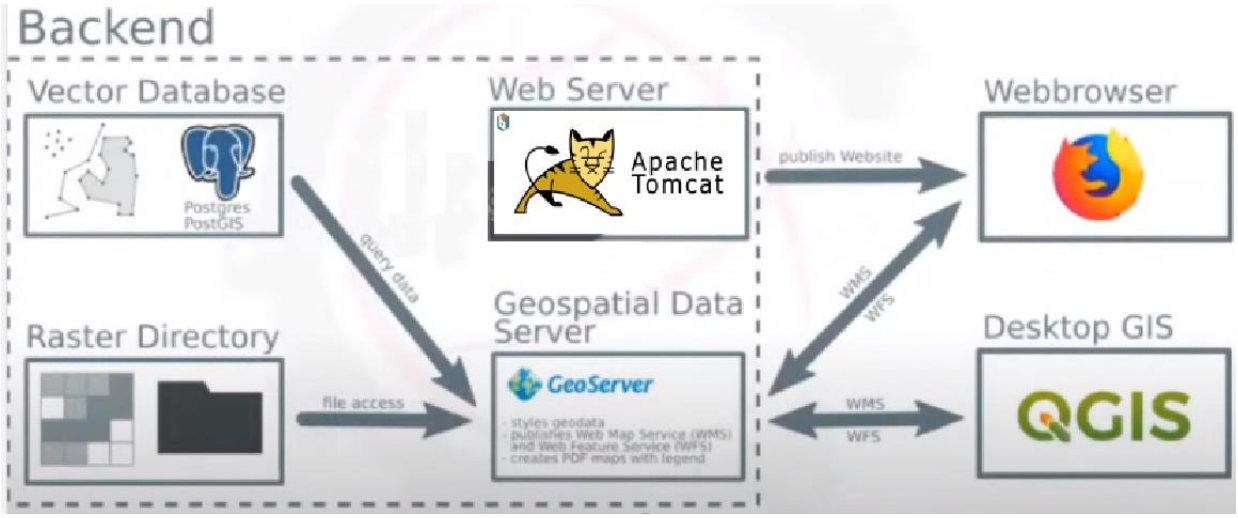


Ilustración 5: Arquitectura del backend del visor openlayer propuesto.

La etapa del mantenimiento y actualización de la información geográfica tendrá un flujo de trabajo que partirá desde el administrador SIG de la SCRD que desde su computador tendrá acceso al servidor en donde está contenida la base de datos espacial de tipo SQL y que es la información que es publicada según estándares OGC mediante Geoserver, lo mencionado se muestra a continuación:

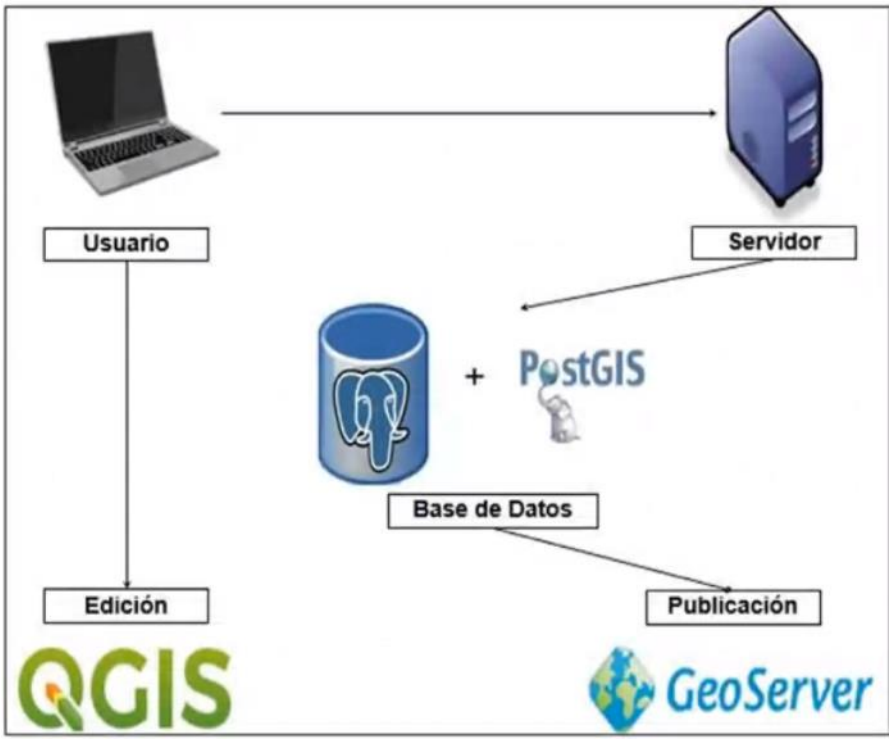


Ilustración 6: Solución a mantenimiento y actualización por parte del administrador SIG de la SCRD.

Cabe resaltar que es necesaria la creación de una VPN para que el administro SIG de la SCRD puesa tener acceso desde su computador a la base de datos espacial que reposa en el servidor estipulado.

En cuanto a los servicios según los estandares OGC que son creados meidante Geoserver, el proceso de trabajo es el siguiente:

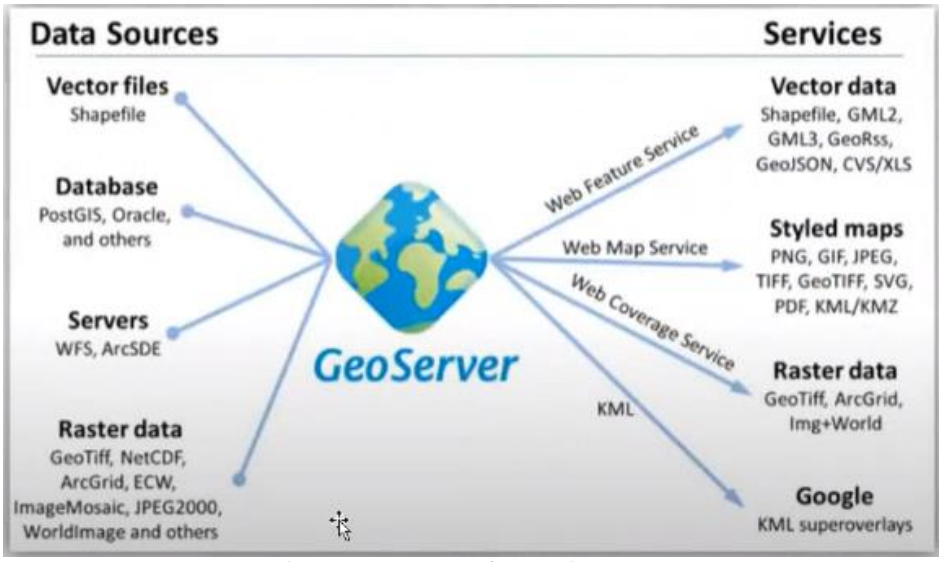


Ilustración 7: trabajo de configuración de Geoserver.

En la ilustración anterior se puede apreciar la versatilidad que ofrece el software Geoserver de código abierto para la publicación de capas geográficas, y sobre todo la compatibilidad que tiene con los otros software fundamentales para este trabajo que son PostgreSQL y QGIS además de su versatilidad al crear servicios con estandares OGC livianos y publicados mediante una dirección URL.

Etapas de migración del sistema de información geográfico SIG

Se presenta un plan de siete (7) etapas para migrar el actual sistema de información geográfica (SIG) denominado Culred-Maps a un geovisor construido con código abierto, al final se estipula el tiempo estimado según el cronograma. Esta migración nos permitirá obtener:

Flexibilidad: Adaptar el sistema a nuestras necesidades específicas.
Ahorro: Eliminar costos de licencia.
Comunidad: Acceder a una comunidad activa de desarrolladores.

Las 7 etapas de la migración son:

1. **Evaluación**: Análisis del sistema actual y definición de necesidades.

- Análisis del sistema SIG actual:
  - Funcionalidades
  - Datos
  - Usuarios
- Definición de necesidades y requisitos para el nuevo geovisor

2. **Diseño**: Planificación de la arquitectura del nuevo sistema.

- Diseño de la arquitectura del nuevo sistema:
  - Base de datos espacial
  - Servicios OGC
  - Visor web
  - Widgets personalizados
- Diseño de la interfaz de usuario del visor web

3. **Desarrollo:** Implementación de la base de datos, servicios OGC, visor web y widgets.

- Instalación y configuración de PostgreSQL/PostGIS en el servidor.
- Importación de los datos espaciales a la base de datos.
- Instalación y configuración de GeoServer en el servidor.
- Publicación de las capas como servicios OGC (WMS, WFS, WCS).
- Desarrollo del visor web con Leaflet u otra aplicación de código abierto.
- Implementación de funcionalidades básicas del visor:
  - Navegación por el mapa
  - Visualización de capas
  - Leyenda
  - Zoom
  - Búsqueda
- Creación de widgets personalizados:
  - Contenedor de capas personalizado por subcategorías
  - Filtros alfanuméricos
  - Filtros espaciales
  - Otras funcionalidades según necesidades

4. **Pruebas:** Validación del nuevo geovisor en diferentes escenarios.

- Pruebas exhaustivas del nuevo geovisor:
  - Funcionalidades
  - Rendimiento
  - Compatibilidad con diferentes navegadores
  - Seguridad

5. **Implementación:** Puesta en producción del nuevo sistema y migración de usuarios.

- Implementación del nuevo geovisor en producción.
- Migración de los usuarios al nuevo sistema.

6. **Capacitación:** Entrenamiento a los usuarios en el uso del nuevo geovisor.

- Capacitación a los usuarios de Direcciones y Subdirecciones de la SCRD sobre cómo usar el nuevo geovisor además de generación y suministro de manuales básicos para requerimientos básicos frecuentes, como revisión, edición espacial o de texto para añadir, eliminar o transformar la información geográfica.

7. **Soporte y Mantenimiento:** Asistencia técnica y actualizaciones del sistema.

- Soporte técnico para el nuevo sistema.
- Mantenimiento y actualizaciones de la base de datos, GeoServer, el visor web y los widgets.

Esta migración a código abierto permitirá fortalecer el SIG de la entidad, optimizar costos y aprovechar las ventajas del mundo OpenLayer.

Cronograma:

En cuanto al tiempo de ejecución del proyecto, se estipula un cronograma para la migración el cual en general, requiere de un proceso que se estima tomaría aproximadamente 6 meses. Los procesos a realizar y las consideraciones se encuentran en diagrama de flujo del proyecto OpenLayer.

Desarrollos en software libre para visores geográficos por entidades adscritas

Desarrollo de geovisor SISBIC en software libre por parte de entidad adscrita IDPC:

La estructura del geovisor SISBIC es la siguiente:

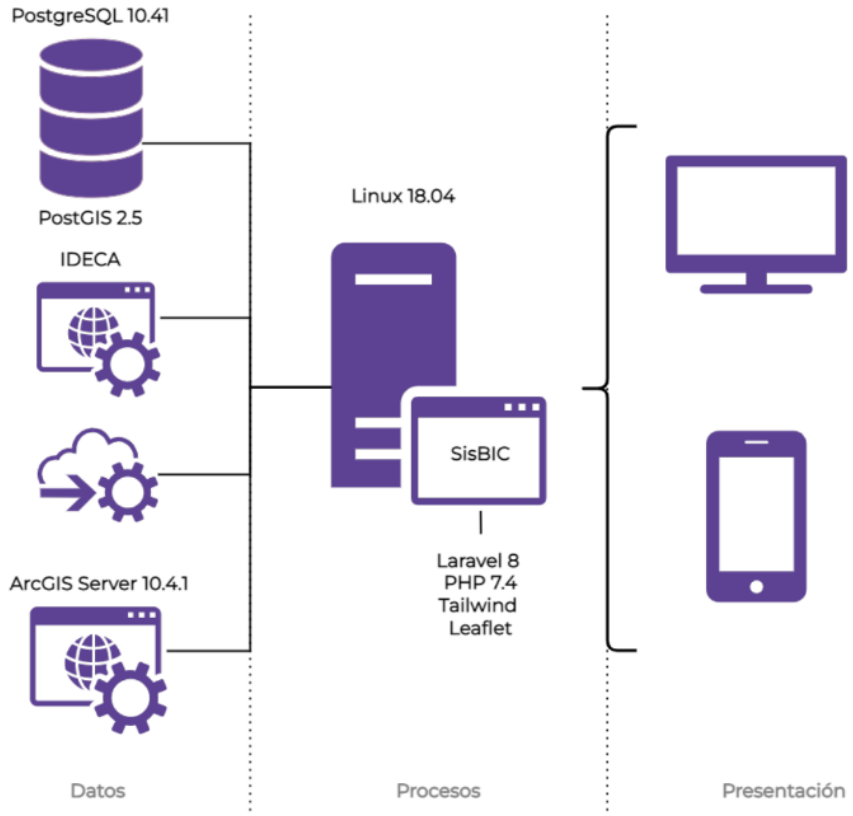


Ilustración 8:Arquitectura de visor geográfico SISBIC.

Como se observa **SISBIC** implementa bases de datos espaciales administradas mediante software **PostgreSQL** con extensión **POSTGIS**, la información geográfica la gestionan mediante ArcgisServer y también consumen servicios de capas libres de IDECA. En cuanto a la estructura, la información geográfica y en sí el visor geográfico se soporta en un servidor con sistema operativo Linux, el desarrollo de la aplicación web esta creado a partir de los frameworks Laravel, la aplicación web es creada mediante lenguajes de programación PHP, Tailwind (CSS) y Javascript (la librería Leaflet) está ultima acogida internacionalmente para mapas interactivos.

Respecto al tema de la **documentación**, el IDPC remite algunos enlaces prácticos para el acercamiento al tema desde Leaflet, PostgreSQL y Laravel.

Tutoriales de leaflet para inclusión de librería, mapas base y capas:
https://leafletjs.com/examples.html
https://bookdown.org/sammigachuhi/book-leaflet/first-leaflet-map.html

inclusión de geoservicios con Leaflet:
https://mappinggis.com/2016/06/anadir-servicio-wfs-leaflet/

Documentación de PostgreSQL:
https://www.postgresql.org/docs/current/

Documentación de PostGIS:
https://postgis.net/documentation/training/

Introducción a PostgreSQL y PostGIS:
https://github.com/psigcat/curso_postgis_1_intro

Integración PostgreSQL con Laravel PHP:
https://www.enterprisedb.com/postgres-tutorials/how-use-postgresql-laravel

Facilidades de Laravel y PostgreSQL para un Geovisor:

Ya que el equipo de la OTI-SCRD tiene importantes desarrollos web haciendo uso del framework Laravel, se observa que este framework tiene facilidades considerables con PostgreSQL el cual es el software de motor de base de datos en el cual se planea estructurar la base de datos espacial de Cultured-maps. Teniendo en cuenta lo anterior se revisa el valor agregado de juntar ambos programas:

Soporte Nativo de Laravel para PostgreSQL: Laravel tiene soporte nativo para PostgreSQL y te permite interactuar con la base de datos de manera eficiente mediante la extensión PostGIS:

Habilitar la extensión PostGIS en PostgreSQL: PostGIS proporciona funciones espaciales y geográficas que son esenciales para trabajar con datos geoespaciales.

Migraciones de Base de Datos: Se puede utilizar las migraciones de Laravel para gestionar y versionar la estructura de la base de datos. Esto facilita la creación y actualización de tablas espaciales.

Eloquent ORM: Laravel ofrece Eloquent, un ORM (Object-Relational Mapping) que simplifica la interacción con la base de datos. Se puede usar Eloquent para modelar y manipular datos geoespaciales.

GeoJSON y Laravel: Laravel facilita la conversión de datos a formato GeoJSON, que es comúnmente utilizado para representar datos geoespaciales en la web. Se puede convertir los datos a GeoJSON utilizando Eloquent y devolverlos a través de los controladores.

API RESTful: Se puede crear una API RESTful con Laravel para exponer datos geoespaciales. Se puede utilizar controladores y rutas para gestionar las solicitudes relacionadas con mapas y capas geográficas.

Leaflet o Mapbox para la Visualización: Se puede utilizar la biblioteca Leaflet o Mapbox en el lado del cliente para visualizar y manipular datos geoespaciales. Ambas bibliotecas son compatibles con Laravel y permiten la integración fácil de mapas interactivos en la aplicación.

Integración con Frontend: Se puede utilizar Laravel Mix o herramientas similares para compilar y gestionar los recursos frontales. Se puede integrar fácilmente bibliotecas como Leaflet y Vue.js para construir una interfaz de usuario interactiva.

Control de Acceso y Autenticación: Laravel proporciona herramientas para gestionar el control de acceso y la autenticación de usuarios. Se puede asegurar las capas geográficas y funciones específicas según la autenticación del usuario.

Despliegue con Laravel Forge o Envoyer: Se puede facilitar el despliegue de la aplicación Laravel utilizando herramientas como Laravel Forge o Envoyer. Estas herramientas simplifican la gestión de servidores y el despliegue continuo.

Pruebas y Mantenimiento: Se puede utilizar las capacidades de Laravel para realizar pruebas unitarias y de integración. Laravel también facilita la monitorización y el mantenimiento de la aplicación.
Al aprovechar las capacidades de Laravel y PostgreSQL con PostGIS, se puede desarrollar un geovisor robusto y escalable. Además, la comunidad activa de Laravel y PostgreSQL proporciona recursos y soporte valiosos para el proyecto.

Preparó	**Edisson Manuel Mora Madrigal** edisson.mora@scrd.gov.co	29/01/2024