Retos de ciudad

Bogotá tiene 5 en Innovación

Reto de ciudad 1: Datos con historia

3. Informe de la implementación y operación de la solución en versión Producto Mínimo viable

Marzo 2025

Versión 3

OFICINA DE LABORATORIO DE CIUDAD-SECRETARÍA DISTRITAL DE PLANEACIÓN
CORPORACIÓN MALOKA DE CIENCIA, TECNOLOGÍA E INNOVACIÓN
IGG-INVERSIONES GUTIÉRREZ GARCÍA

#5 EN INNOVACIÓN

RETO DE CIUDAD 1

DATOS CON HISTORIA

1. Introducción **4**

2. Solución implementada **5**

2.1. Aspectos técnicos 6

2.2. Fases de despliegue 9

2.3 Máquina 1 9

2.4. Máquina 2 10

2.5. Máquina 3 10

2.6. Configuraciones específicas - Máquina 1 12

2.6.1. Instalación del servidor reverso nginx 12

2.6.2. Configuración servicio Keycloak 12

2.6.3. Configuración servicio Nodejs 13

2.6.4. Usuarios y Claves para acceder a la interfaz web 14

2.6.5. Configuración servicio API – backend 15

2.6.6. Servicio de Almacenamiento – MinIO – compatible con S3 y/o similares 17

2.7. Configuraciones específicas - Máquina 2 18

2.7.1. Instalación de Python – Versión 3.11 18

2.7.2. Api de Indexación 19

2.7.3. Servicio de Canalizaciones 20

2.7.4. Contenedor del Servicio de Canalizaciones 20

2.7.5. Base de Datos MySQL 21

2.7.6. Base de datos Postgres 22

2.7.7. Base de Datos – Mongo 23

2.7.8. Base de datos Neo4J 24

2.8. Configuraciones específicas - Máquina 3 25

2.8.1. Instalación de los drivers necesarios 25

3. Recomendaciones técnicas a nivel general **27**

4. Plan de soporte **28**

5. Aspectos funcionales **28**

6. Funcionalidades fuera del alcance **35**

7. Evaluación por parte de los usuarios **37**

8. Entregables finales **41**

8.1 Productos de desarrollo 41

8.2. Manuales 46

8.3. Formación 47

9. Anexos **49**

9.1. Configuración nginx 49
9.2. Configuración keycloak 53
9.3. Configuración angular y api backend 54
9.4. Configuración MinIO 55
9.5. Configuración servicio canalizaciones 56
9.6. Configuración Base de datos postgres 56
9.7. Configuración Base de datos Neo4j 57
10. Conclusiones **58**

1. Introducción

La estrategia de innovación abierta *Retos de Ciudad* invitó a personas de la academia, de organizaciones y de la comunidad en general, para presentar propuestas de desarrollo tecnológico basado en ciencia de datos que respondan al reto: *¿Cómo automatizar la recolección, análisis y categorización de la información cualitativa, gestionada por las diferentes dependencias de la Secretaría Distrital de Planeación (SDP), a través de un desarrollo tecnológico, para que esta información se pueda integrar efectivamente en la toma de decisiones y optimizar la planificación urbana?*

Esta estrategia se desarrolla en el marco del Convenio Especial de Cooperación No. 997-2024 suscrito entre la Oficina Laboratorio de Ciudad de la Secretaría Distrital de Planeación y la Corporación Maloka de Ciencia, Tecnología e Innovación, cuya metodología se divide en tres momentos: Convocatoria, Prototipado e Implementación.

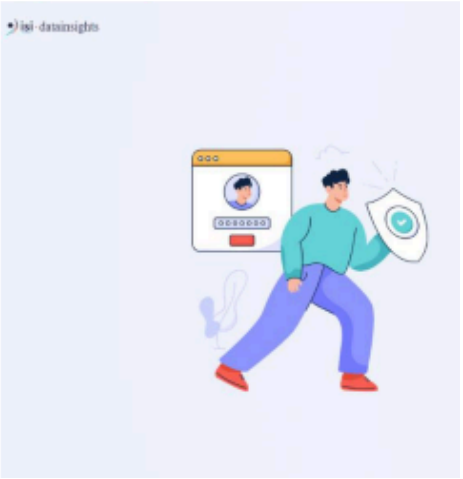
El presente documento hace parte de la última etapa, cuyo alcance es la construcción del MVP de la solución ganadora en la etapa de Prototipado, que fue propuesta por el equipo de IGG Inversiones Gutiérrez García. **Este documento presenta la información clave asociada al producto implementado, desde la configuración y accesos, la descripción de los productos entregados como el código, manuales, mockups; hasta el feedback de los usuarios finales, tras usar el producto mínimo viable (MVP)**.

Es importante recordar que el alcance de esta Etapa tiene como objetivo el desarrollo de un MVP (producto mínimo viable), y que este prototipo permitió validar sus posibilidades de ajuste y ampliación de desarrollo en futuras etapas y recopilar comentarios tempranos por parte de los usuarios

2. Solución implementada

La Secretaría Distrital de Planeación, a través de la Oficina Laboratorio de Ciudad, y la Corporación Maloka lanzaron el primer Reto de Ciudad: Datos con Historia. Como parte de esta estrategia de innovación abierta, se implementó una solución tecnológica desarrollada en el marco del reto, con el propósito de integrar la inteligencia artificial en el análisis de datos cualitativos y optimizar la toma de decisiones en la planeación urbana. Tras un proceso de convocatoria y prototipado, se seleccionó y desarrolló el producto mínimo viable (PMV) de la aplicación web ISI Data Insights, creada por el equipo IGG - Inversiones Gutiérrez García y CIA.

El objetivo de esta herramienta es mejorar la calidad, consistencia y análisis de la información cualitativa proveniente de diversas fuentes, fortaleciendo la toma de decisiones estratégicas en Bogotá. ISI Data Insights permite automatizar la clasificación y análisis de datos en temas clave .



Metodología de trabajo e impacto

El desarrollo de ISI Data Insights se realizó bajo la metodología Scrum, con seguimiento semanal y revisiones constantes. Se implementó un proceso de co-creación con usuarios finales de la SDP, quienes aportaron información clave para el diseño de la interfaz y funcionalidad de la aplicación.

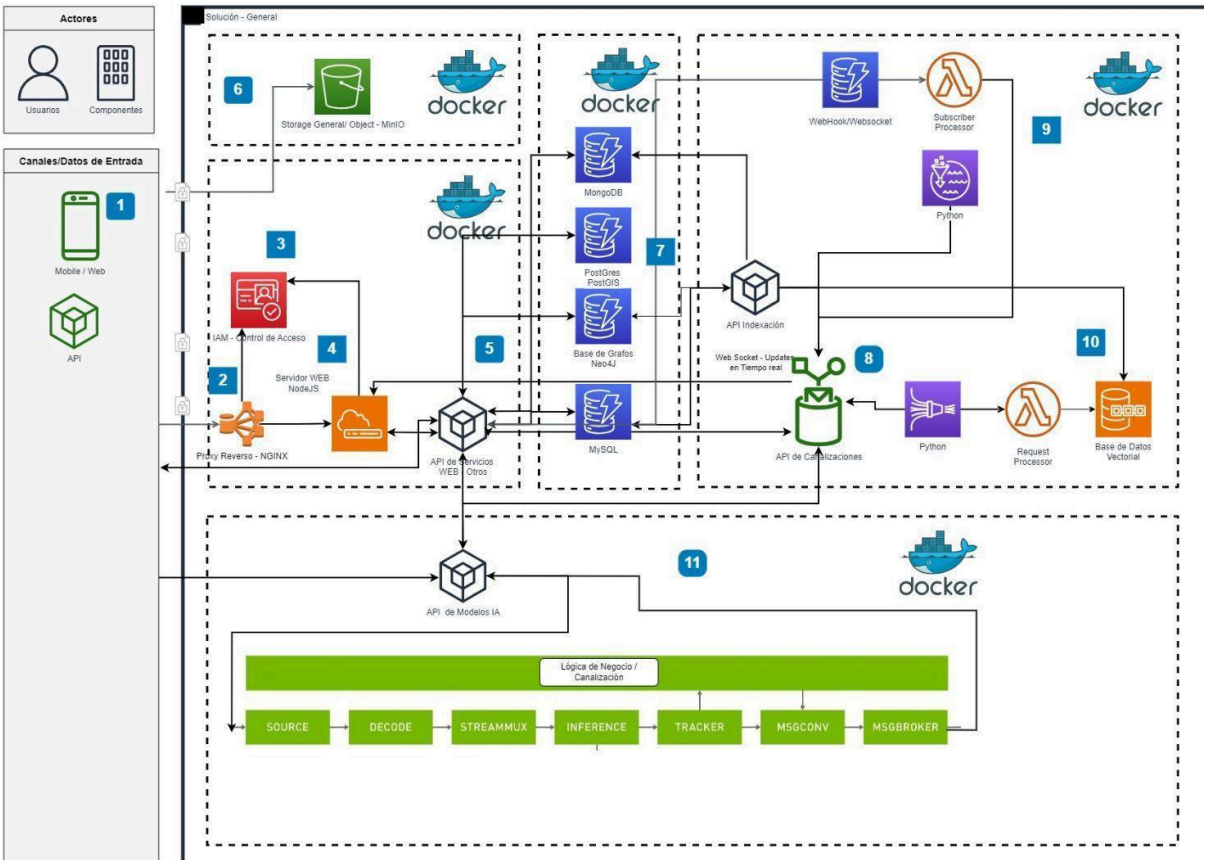
La herramienta cuenta con las siguientes características clave:

- Capacidad para analizar información no estructurada a partir de archivos en diversos formatos como Excel, PDF y Word.
- Validación de consistencia en las fuentes de datos.
- Implementación de modelos de inteligencia artificial para extraer patrones clave en la información.
- Visualización amigable de los resultados, facilitando el análisis y la toma de decisiones.

Entre los beneficios para los usuarios se destaca la reducción de tiempos en el análisis de información, mayor exactitud en los datos procesados y una optimización en la identificación de aportes ciudadanos relevantes para la planeación de la ciudad.

2.1. Aspectos técnicos

El alcance del desarrollo en relación a la arquitectura propuesta y herramientas implementadas se visualiza en la siguiente figura. Más adelante se describen en un alto nivel los diferentes componentes y/o servicios necesarios.



La arquitectura de la solución desde el punto de vista funcional está basada en módulos. Cada módulo está compuesto por uno o más contenedores basados en Docker o servicios que están corriendo en la máquina virtual. Esta solución utiliza tecnologías modernas y permite correr la solución on-premise o en cualquier proveedor de nube basado en contenedores. Esta arquitectura permite escalar de manera independiente cada uno de los módulos y/o de sus diferentes componentes de manera independiente. Los diferentes módulos se describen a continuación.

Módulo Front-End y Capa Media:

1. Acceso por un servidor web local que tiene un proxy reverso basado en Apache Nginx. Todos los otros dispositivos se comunican por medio de una API y/o un componente del contenedor del servidor web y/o del contenedor que tiene el servicio del Colas de mensajería abre una conexión remota.
2. Proxy reverso que permite escalar varios de los componentes de la solución planteada utilizando Apache Nginx. Control de Acceso a las diferentes aplicaciones / OAuth2 - JWT.
3. Los datos de autenticación estarán en la base de datos propia de KeyCloak.
4. Servicio WEB que tiene una aplicación basada en node.js que tendrá todos los casos de uso como autenticación, un CRUD básico de procesos de carga y transformación de archivos basado en formato PDF y/o EXCEL. También se implementarán todos los otros casos de uso necesarios para las 3 áreas de la Secretaría Distrital de Planeación y todos sus diferentes roles.
5. API de servicios implementada usando SpringBoot y java para atender todos los servicios de capa media necesarios por el servidor web, extraer los datos desde algún otro componente de la solución en caso de necesidad.

Módulo Almacenamiento:

6. Servicios de almacenamiento de objetos basado en un protocolo compatible con S3 y/o similar. Se usó la herramienta de código abierto MinIO. En caso de que en el futuro se desee migrar a una solución de nube, se puede cambiar este servicio sin problema. Este servicio permite la replicación.

Módulo de Bases de Datos:

7. Servicios de base de datos. La solución contempla una base de datos no estructurada basada en MongoDB. También se contempla una base de datos postgres para persistir información de tipo geo-espacial (extensión postgis) y cualquier otro tipo de información estructurada y relacional necesaria como MySQL. También se contempla una base de datos basada en grafos como Neo4J o similares.

Módulo de Indexación de Documentos:

8. Servicio de API basada en protocolo REST que permite escalar todo el proceso de indexación con uno o más nodos respectivamente para realizar todos los trabajos de indexación de los diferentes documentos y/o casos de uso en batch. Este servicio es fundamental para implementar toda la lógica de negocio de los diferentes pipelines y/o canalizaciones de procesamiento de modelos de Inteligencia Artificial.
9. Componentes para poder hacer procesamiento. Este componente permitirá realizar búsquedas híbridas. Se fundamenta en la implementación de uno o más requerimientos y/o pasos en python de todas las canalizaciones de datos. Este componente permitirá notificar eventos de interés a la API del backend.
10. En esta sección se tiene el servicio de la base de datos vectorial basado en una tecnología como

MongoDB y/o similar. Este motor se encargará de persistir los sumergimientos (embeddings - por su nombre en inglés). Esto le permitirá a la solución realizar búsquedas semánticas, implementar varios temas de similitud semántica, entre otros.

Módulo de Modelos de Inteligencia Artificial:

11. Servicio que implementa una solución basada en la tecnología de nvidia, python, llamac++ y otros servicios para exponer uno o más modelos de Inteligencia Artificial de una manera sincrónica y/o asincrónica. Este módulo cuenta con un servicio de cola de mensajería para hacer replicación y/o atender solicitudes en batch de manera asíncrona. Servicios de canalización y flujos de trabajo para encadenar uno o más modelos en un flujo de trabajo con capacidad para escalar cada componente del módulo de manera independiente.

La implementación de la solución tecnológica en la infraestructura de la SDP tuvo varias fases. En un primer momento se tuvo una reunión con el área de la DTI, para explicarles la arquitectura inicialmente propuesta. Se acordaron aspectos a nivel de comunicación y cómo iban a ser alcanzadas las máquinas aprovisionadas. En la propuesta inicial se habían planteado 3 máquinas virtuales que fueron aprovisionadas en la nube privada de la entidad usando el proveedor Oracle para dicho fin. A nivel de espacio de disco duro, inicialmente no se realizó la configuración solicitada. Esto no tuvo impacto para la implementación inicial, pero sí es importante seguir monitoreando el consumo a nivel de almacenamiento de la solución. La fase de verificación de acceso a las 3 máquinas configuradas tuvo algunos percances. Inicialmente no se logró la comunicación esperada y fue necesario un acompañamiento en varias sesiones para lograr que la conectividad se lograra. IGG validó inicialmente el acceso por medio del protocolo ssh a las 3 máquinas virtuales utilizando las llaves privadas suministradas por la entidad. Seguidamente se procedió a verificar que se tuvieran implementadas las reglas de comunicación solicitadas. Algunas de dichas reglas no fueron configuradas de la manera solicitada y fue necesario ajustar políticas a lo largo de la implementación.

En una tercera fase, fue necesario revisar políticas de seguridad a nivel de comunicación entre las 3 máquinas virtuales, pues no se lograba habilitar la comunicación entre los nodos, protocolos y servicios necesarios. Durante el proceso de implementación fue necesario revisar estas políticas en un par de ocasiones. Más o menos en la mitad de la implementación y cuando ya se hizo necesario realizar pruebas con los usuarios finales, IGG notificó temas de alta latencia en la infraestructura utilizada para la solución. Se realizaron algunas reuniones con los profesionales de la DTI encargados de dicho tema, sin lograr identificar una solución. Se mencionaron configuraciones que se iban a realizar para validar si se podría mejorar la situación identificada, pero aparentemente y hasta el último día del contrato, en su fase de implementación, esta situación no se vio resuelta. Durante las sesiones en las que estuvo presente IGG, se trató de evidenciar que los temas de latencia no se debían a los servicios utilizados por la solución, sino que eran factores externos. En resumen, lo que se mostró por parte del equipo de IGG, es que cada cierto tiempo, las conexiones establecidas con los diferentes servicios se demoraban en promedio 36 segundos en terminar su ciclo y eso hacía muy lenta la respuesta de la solución implementada. Se validó con los diferentes logs disponibles, que

las peticiones no llegaban al proxy-reverso que se encargaba de realizar el enrutamiento de cada una de las solicitudes a los diferentes servicios. Se evidenció de los logs de dicho servicio, que la solución y los diferentes componentes que intervienen, estaban respondiendo de manera correcta y dentro de los tiempos esperados.

Finalmente, y luego de algunas pruebas iniciales se evidenció que la GPU aprovisionada para la solución, iba a ser muy pequeña. Los tiempos de respuesta de los diferentes modelos de IA utilizados en el prototipo mínimo viable, mostraron que en promedio se obtenían 1.8 tokens por segundo en la generación de inferencias y los tiempos de respuesta para el primer token eran casi del doble de los servicios que IGG utilizó para hacer todas las pruebas con los usuarios de las áreas finales. En esta línea, hay varias oportunidades futuras de mejoras, para reducir los tiempos de respuesta.

2.2. Fases de despliegue

La solución implementada corre en tres máquinas virtuales. Las fases del despliegue cubren los servicios de cara a los usuarios de la capa web (desplegados en la máquina 1), luego se tienen todos los servicios de bases de datos en la máquina 2, junto con otras API's necesarias para correr las canalizaciones de datos. Finalmente, en la máquina 3, se despliega todo lo relacionado con los modelos de Inteligencia Artificial. En la implementación actual, estas tres máquinas tienen las siguientes IPs fijas:

- Máquina 1 – FrontEnd y otros servicios – 10.5.4.115, el usuario con el que se realizó toda la implementación fue "opc" junto con la llave privada compartida por la SDP.
- Máquina 2 – Bases de Datos y otros servicios - 10.5.4.136, el usuario con el que se realizó toda la implementación fue "opc" junto con la llave privada compartida por la SDP.
- Máquina 3 – Modelos de IA - 10.5.4.80, el usuario con el que se realizó toda la implementación fue "ubuntu" junto con la llave privada compartida por la SDP.

Los sistemas operativos de las máquinas utilizadas respectivamente son:

2.3 Máquina 1

NAME="Oracle Linux Server"
VERSION="9.5"
ID="ol"
ID_LIKE="fedora"
VARIANT="Server"
VARIANT_ID="server"
VERSION_ID="9.5"
PLATFORM_ID="platform:el9"
PRETTY_NAME="Oracle Linux Server 9.5"

ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:oracle:linux:9:5:server"
HOME_URL="https://linux.oracle.com/"
BUG_REPORT_URL="https://github.com/oracle/oracle-linux"

ORACLE_BUGZILLA_PRODUCT="Oracle Linux 9"
ORACLE_BUGZILLA_PRODUCT_VERSION=9.5
ORACLE_SUPPORT_PRODUCT="Oracle Linux"
ORACLE_SUPPORT_PRODUCT_VERSION=9.5

2.4. Máquina 2

NAME="Oracle Linux Server"
VERSION="9.5"
ID="ol"
ID_LIKE="fedora"
VARIANT="Server"
VARIANT_ID="server"
VERSION_ID="9.5"
PLATFORM_ID="platform:el9"
PRETTY_NAME="Oracle Linux Server 9.5"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:oracle:linux:9:5:server"
HOME_URL="https://linux.oracle.com/"
BUG_REPORT_URL="https://github.com/oracle/oracle-linux"

ORACLE_BUGZILLA_PRODUCT="Oracle Linux 9"
ORACLE_BUGZILLA_PRODUCT_VERSION=9.5
ORACLE_SUPPORT_PRODUCT="Oracle Linux"
ORACLE_SUPPORT_PRODUCT_VERSION=9.5

2.5. Máquina 3

PRETTY_NAME="Ubuntu 22.04.5 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.5 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu

ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy

Todas las configuraciones necesarias a nivel de cada máquina se hicieron sobre dichas versiones. Luego de tener actualizados los OS de cada máquina, se instala docker:

sudo dnf update -y

sudo dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo

sudo dnf install docker-ce -y

sudo systemctl start docker
sudo systemctl enable docker
sudo systemctl status docker

sudo dnf install -y curl
curl -L https://github.com/docker/compose/releases/download/v2.21.0/docker-compose-linux-x86_64 -o /usr/local/bin/docker-compose

sudo curl -L
https://github.com/docker/compose/releases/download/v2.32.4/docker-compose-linux-x86_64 -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
sudo ls -al /usr/local/bin/docker-compose

sudo docker-compose --version

La versión final de Docker y Docker compose utilizadas respectivamente son:

- Máquina 1:

Docker version 27.5.1, build 9f9e405
Docker Compose version v2.32.4

- Máquina 2:

Docker version 27.5.1, build 9f9e405
Docker Compose version v2.32.4

- Máquina 3:

Docker version 27.5.1, build 9f9e405
Docker Compose version v2.32.4

2.6. Configuraciones específicas - Máquina 1

A nivel de la máquina 1, se realizaron las siguientes configuraciones particulares:

2.6.1. Instalación del servidor reverso nginx

sudo dnf install -y nginx
sudo systemctl enable --now nginx.service

sudo systemctl status nginx

La ruta de la configuración del nginx está en:

/etc/nginx/nginx.conf

Los logs de acceso y errores están ubicados respectivamente en:

/var/log/nginx/access.log
/var/log/nginx/error.log

2.6.2. Configuración servicio Keycloak

Para todo lo relacionado con los temas de autenticación y acceso de usuarios, se utilizó el servicio KeyCloak. La versión necesaria para que funcione la solución es: 24.0.2. No es necesario un archivo Dockerfile, pues toda la configuración necesaria se tiene en el archivo "Docker-compose.yml" mencionado en esta sección. Adicionalmente, se incluye en la sección de "Anexos", el contenido del archivo en cuestión.
La configuración a nivel del Docker compose para este servicio está disponible en la siguiente ruta:

/home/opc/varios/keycloak

sudo usermod -aG docker opc

Para validar que el servicio está activo se utiliza el siguiente comando:

docker ps -f name=keycloak

Para validar el estatus del servicio del Docker compose, se ejecuta:

cd /home/opc/varios/keycloak
docker compose ps

Para visualizer los logs del servicio:

cd /home/opc/varios/keycloak
docker compose logs -f

Usuario y Clave para acceder a la interfaz de administración:

usuario: admin
clave: password
host: iamisidi.sdp.gov.co
url complete: https://iamisidi.sdp.gov.co:8443

Se sugiere revisar toda la documentación en la url siguiente:
https://www.keycloak.org/archive/documentation-24.0.html

Todo lo relacionado con los certificados de seguridad, que son vitales para que el servicio funcione, está en la siguiente ruta:
cd /home/opc/varios/keycloak/certs

Se deben actualizar los certificados de seguridad cada vez que se venzan para que el sistema pueda operar.

2.6.3. Configuración servicio Nodejs

Para todo lo relacionado con los temas del servidor web que tiene la aplicación de angular se utiliza la versión del servidor nodejs: node:alpine que para efectos prácticos tiene la versión de node - v23.10.0. Dentro de la carpeta que se menciona a continuación, está el archivo Dockerfile respectivo para construir la imagen necesaria para el servicio. El puerto habilitado para el servicio es el 4200 que es el puerto por defecto de las aplicaciones web de angular.

La configuración a nivel del Docker compose para este servicio está disponible en la siguiente ruta:

/home/opc/varios/etls

Para validar que el servicio está activo se utiliza el siguiente comando:

docker ps -f name=etls-angular-app-1

Para validar el estatus del servicio del Docker compose, se ejecuta:

/home/opc/varios/etls
docker compose ps
docker compose ps angular-app

Para visualizer los logs del servicio:

cd /home/opc/varios/etls
docker compose logs -f angular-app

2.6.4. Usuarios y Claves para acceder a la interfaz web

url completa: https://isidi.sdp.gov.co

SPI:
Los usuarios con el rol1 y que representan a los analistas de la SPI (área de la entidad que se encarga de administrar y realizar el seguimiento de los planes de inversión), tienen acceso a la interfaz de carga de información respectiva para esa área. Adicionalmente estos usuarios, pueden listar los procesos de análisis agendados, tanto como ver los documentos respectivos y los detalles de los procesos particulares.

usuario: usuario1
clave: password

SPPPSE:
Los usuarios con el rol2 y que representan a los analistas de la SPPPSE (área de la entidad que se encarga de administrar y realizar el seguimiento a los planes de acción), tienen acceso a la interfaz de carga de información respectiva para esa área. Adicionalmente estos usuarios, pueden listar los procesos de análisis agendados, tanto como ver los documentos respectivos y los detalles de los procesos particulares.

Usuario: usuario2

Clave: password

OPDC:
Los usuarios con el rol3 y que representan a los analistas de la OPDC (área de la entidad que se encarga de administrar y realizar seguimiento a los aportes ciudadanos), tienen acceso a la interfaz de carga de información respectiva para esa área. Adicionalmente estos usuarios, pueden listar los procesos de análisis agendados, tanto como ver los documentos respectivos y los detalles de los procesos particulares.

Usuario: usuario3
Clave: password

Usuarios Admin:
Los usuarios con el rol4 y que representan a los ingenieros de la DTI, tienen acceso a la interfaz de carga de información respectiva para todas las áreas. Adicionalmente estos usuarios, pueden listar los procesos de análisis agendados, tanto como ver los documentos respectivos y los detalles de los procesos particulares. Finalmente pueden ver detalles de los procesos de canalizaciones que se han agendado junto con algunos logs que ayudan a resolver problemas en caso de necesidad.

Usuario: superusuario1
Clave: password

2.6.5. Configuración servicio API – backend

Para todo lo relacionado con los temas del servidor de la API del backend y de las ETLS se utiliza una aplicación de SpringBoot versión: 3.4.0. Adicionalmente, se usa Camel, en su versión: 4.9.0, SpringSecurity en su versión: 6.4.2. Todo el contenedor utiliza la versión de Java 17 de openjdk para su ejecución. La API del backend interactúa principalmente con la base de datos en MongoDB y la base de datos Postgres. En la carpeta que se menciona a continuación, se encuentra el archivo Dockerfile, que es necesario para poder construir la imagen asociada y así poder instanciar el contenedor para ejecutar el servicio respectivo. Existe un documento con el catálogo de servicios del backaend en donde se puede ampliar la información:
Catálogo de EndPoint - API Backend.xlsx

La configuración a nivel del Docker compose para este servicio está disponible en la siguiente ruta:
/home/opc/varios/etls

Para validar que el servicio está activo se utiliza el siguiente comando:

docker ps -f name=etls-springboot-app-1

Para validar el estatus del servicio del Docker compose, se ejecuta:

/home/opc/varios/etls
docker compose ps
docker compose ps springboot-app

Para visualizer los logs del servicio:

cd /home/opc/varios/etls
docker compose logs -f springboot-app

En la misma carpeta, se encuentra mapeados los logs internos a la carpeta "logs"

tail -f logs/sdp_backend.log

Antes de poder ejecutar el contenedor de la API del backend, se debe importar el certificado del sudominio iamisidi.sdp.gov.co debido a que no se encuentra publicado en un DNS disponible en el contenedor en el que se está ejecutando el servicio.

Se debe ingresar al contenedor e importar manualmente el certificado ya que la entidad certificadora no está disponible dentro de la máquina virtual de java en el contenedor.

docker exec -it etls-springboot-app-1 /bin/sh

dentro del contenedor se importa el certificado de IAM

sh-4.4# keytool -printcert -rfc -sslserver iamisidi.sdp.gov.co > iamisidi.crt
sh-4.4# ls -al

se hace el import al CA del java por default

keytool -importcert -file iamisidi.crt -alias iamisidi -storepass changeit -keystore /usr/java/openjdk-17/lib/security/cacerts -noprompt

se sale del contenedor y se vuelve a reiniciar contenedor para garantizar que se tomen los cambios de la CA

Se debe volver a iniciar el contenedor de la API del backend

2.6.6. Servicio de Almacenamiento – MinIO – compatible con S3 y/o similares

La configuración del servicio de almacenamiento de objetos basado en un protocolo compatible con S3 y/o similar se implementa utilizando la herramienta de código abierto MinIO en su versión: RELEASE.2025-02-07T23-21-09Z. Este servicio no requiere de un archivo Dockerfile, pues toda la configuración necesaria está en el archivo "Docker-compose.yml" que se encuentra en la ruta mencionada a continuación.

La configuración a nivel del Docker compose para este servicio está disponible en la siguiente ruta:

/home/opc/varios/minio

Para validar que el servicio está activo se utiliza el siguiente comando:

docker ps -f name=minio

Para validar el estatus del servicio del Docker compose, se ejecuta:

/home/opc/varios/minio
docker compose ps
docker compose ps minio

Para visualizar los logs del servicio:

cd /home/opc/varios/minio
docker compose logs -f minio

Usuarios y Claves para acceder a la interfaz web:

url completa: https://minisidi.sdp.gov.co

usuario: SDP_minAdmin
clave: SDP_mi1nio_04dmin

En esta interfaz de administración, se pueden crear usuarios, nuevos buckets, listar los ítems que se encuentran en cada bucket, crear llaves de acceso, revisar los logs del servicio, crear políticas de acceso, entre otras. Se invita al lector a revisar la siguiente url donde se encuentra toda la documentación completa:

https://min.io/docs/minio/linux/index.html

La ruta física del servidor en la que se persisten todos los elementos gestionados por el servicio está en /opt/minio-data . Se sugiere monitorear de manera continua la capacidad disponible en los discos duros para realizar la ampliación respectiva a nivel del sistema operativo y posiblemente para realizar tareas de backup de dicha ruta. El servicio de MinIO puede ser desplegado en cluster para garantizar una alta disponibilidad del servicio. Estos aspectos se mencionan en la url que se indicó previamente y en la que se encuentra toda una lista de buenas prácticas que pueden ser implementadas.

En la misma ruta:

/home/opc/varios/minio

Se encuentran las siguientes dos carpetas:

certs
minio_certs

Cada una tiene respectivamente los certificados de seguridad que deben ser actualizados. Sin estos certificados la solución no puede operar de manera correcta.

2.7. Configuraciones específicas - Máquina 2

A nivel de la máquina 2, se realizaron las siguientes configuraciones particulares:

2.7.1. Instalación de Python – Versión 3.11

Se valida la versión actual a nivel del OS:

python --version

En principio debería estar instalada la siguiente versión:

Python 3.9.21

Se hace la misma revisión del manejador de paquetes de Python PIP:

pip --version

Si no está instalada una versión de Python 3.11 o una versión con variantes en su menor versión, se procede a hacer la instalación del Python correcto así:

sudo dnf install -y python3.11

sudo dnf module install python311

sudo dnf install python3.11

sudo dnf search python3.11-pip

sudo dnf install -y python3.11-pip

Se verifica la instalación correcta de PIP así:

pip
pip-3.11

pip --version
pip3.11 –version

2.7.2. Api de Indexación

Este servicio permite invocar uno o más servicios para iniciar una o más de las canalizaciones configuradas en el sistema, específicamente en la base de datos en el servidor de mongodb. Estos servicios son invocados desde la API del backend de la solución. Es un script en Python que se encuentra en la ruta:

cd /home/opc/api

En caso de necesidad se instalan los paquetes adicionales requeridos como:

pip3.11 install flask
pip3.11 install flask_cors

Para validar si es servicio está corriendo, se puede utilizar la utilidad netstat y verificar si laAPI rest implementada en Python está corriendo en el puerto 5000 así:

netstat -tlnp | grep 5000

0 utilizando un filtro sobre los procesos activos así:

ps -aux | grep python3.11

Para lanzar el servicio se cambia la ruta a:

cd /home/opc/api

nohup python3.11 API.py > api.log 2>&1 &

Se puede verificar que todo está bien al hacer el tail del log a´si:

tail -f api.log

2.7.3. Servicio de Canalizaciones

Este servicio es fundamental para poder correr las diferentes canalizaciones implementadas en Python. Para realizar la configuración del servicio se realiza la instalación:

pip3.11 install zenml==0.73.0

Se puede verificar la instalación así:

zenml version

2.7.4. Contenedor del Servicio de Canalizaciones

Este servicio es fundamental para la operación satisfactoria de la solución. En caso de que no se encuentre disponible, se generaran timeouts en las operaciones invocadas desde el servicio de la API del backend y/o en la API de indexación. Estos errores impedirán que el flujo normal por el que deben pasar los diferentes ítems sea ejecutado de manera satisfactoria. La configuración de este servicio se encuentra en la ruta siguiente:

cd /home/opc/zenml/canalizaciones/zenml/

Corresponde a un Docker compose que despliega una base de datos MySQL versión 8.0 y al servidor de canalizaciones versión: 0.73.0.

Para validar que el servicio está activo se utiliza el siguiente comando:

docker ps -f name=zenml-zenml-1

Para validar el estatus del servicio del Docker compose, se ejecuta:

/home/opc/zenml/canalizaciones/zenml
docker compose ps
docker compose ps zenml

Para visualizer los logs del servicio:

cd / home/opc/zenml/canalizaciones/zenml
docker compose logs -f zenml

Usuarios y Claves para acceder a la interfaz web:

url completa: http://10.5.4.136:8080

usuario: sdpadmin
clave: sdpadminp4ss

En este servicio se pueden revisar todos los aspectos de las diferentes canalizaciones agendadas por la solución, revisar los pasos respectivos que conforman cada canalización, listar el contexto utilizado para la ejecución respectiva y acceder a los logs respectivos, en caso de que existan.

2.7.5. Base de Datos MySQL

Esta base de datos se utiliza para almacenar todos los datos de las canalizaciones respectivas. Se registra información sobre los diferentes pasos que conforman cada uno de las canalizaciones de datos utilizadas en la solución, tiempos de ejecución, variables de entorno, contexto de ejecución, entre otras. La configuración de este servicio se encuentra en la ruta siguiente:

cd /home/opc/zenml/canalizaciones/zenml/

Corresponde a un Docker compose que despliega una base de datos MySQL versión 8.0.

Para validar que el servicio está activo se utiliza el siguiente comando:

docker ps -f name=zenml-mysql-1

Para validar el estatus del servicio del Docker compose, se ejecuta:

/home/opc/zenml/canalizaciones/zenml
docker compose ps
docker compose ps mysql

Para visualizar los logs del servicio:

cd / home/opc/zenml/canalizaciones/zenml
docker compose logs -f mysql

usuario: root
clave: sdp_zenml_p4ssword

2.7.6. Base de datos Postgres

Esta base de datos se utiliza para persistir toda la información de los procesos en batch gestionados por el SDK SpringBatch. Estas tablas contienen información sobre los diferentes pasos para ejecutar los procesos de transformación y carga. La configuración de este servicio se encuentra en la ruta siguiente:

cd /home/opc/postgres/

Corresponde a un Docker compose que despliega una base de datos Postgres versión .
Para validar que el servicio está activo se utiliza el siguiente comando:

docker ps -f name=spring-batch-postgres

Para validar el estatus del servicio del Docker compose, se ejecuta:

/home/opc/postgres
docker compose -f docker-compose-postgres.yml ps
docker compose -f docker-compose-postgres.yml ps postgres

Para visualizar los logs del servicio:

/home/opc/postgres
docker compose -f docker-compose-postgres.yml logs -f postgres

usuario: admin
clave: admin

2.7.7. Base de Datos – Mongo

Esta base de datos corresponde al modelo de persistencia principal de la solución. La configuración de este servicio se encuentra en la ruta siguiente:

cd /home/opc/postgres/

Corresponde a un Docker compose que despliega una base de datos MongoDB versión mongodb-community-server:7.0.16-ubuntu2204.

Para iniciar la configuración del servicio se descarga la imagen a´si:

docker pull mongodb/mongodb-community-server:7.0.16-ubuntu2204

Para corre el contenedor con la imagen de MondoDB respectiva se ejecuta la siguiente instrucción:

Se deben diligenciar las siguientes dos variables de entorno:

environment variables

-e MONGO_INITDB_ROOT_USERNAME=mongoadmin
-e MONGO_INITDB_ROOT_PASSWORD=sdp_M$A4dmi1n

Luego se ejecuta la siguiente instrucción:

docker run --name mongodb -p 27017:27017 -d -v /opt/mongodb/data:/data/db -e MONGO_INITDB_ROOT_USERNAME_FILE=mongoadmin -e MONGO_INITDB_ROOT_PASSWORD_FILE=sdp_M$A4dmi1n mongodb/mongodb-community-server:7.0.16-ubuntu2204 --bind_ip_all –auth

Si se tienen configuradas las variables de entorno asociadas al usuario raíz de la DB, se puede ejecutar el servicio así:

docker run --name mongodb -p 27017:27017 -d -v /opt/mongodb/data:/data/db mongodb/mongodb-community-server:7.0.16-ubuntu2204 --bind_ip_all –auth

Para validar que el servicio está activo se utiliza el siguiente comando:

docker ps -f name=mongodb

Para visualizar los logs del servicio:

docker logs -f mongodb
docker logs -n 100 mongodb

usuario: mongoadmin
clave: sdp_M$A4dmi1n

Para hacer cualquier operación sobre la base de datos mongo, se puede utilizar el contenedor que tiene un cliente de línea de comando.

Se ingresa a la línea de comando del contenedor que está ejecutando la base de dato mongodb así:

docker exec -it mongodb /bin/sh

Una vez se está en la línea de comando, se crea una sesión a la base de datos utilizando el cliente mongosh así:

mongosh --port 27017 --authenticationDatabase "admin" -u "mongoadmin" -p

Luego se ingresa la clave: sdp_M$A4dmi1n

2.7.8. Base de datos Neo4J

Este servicio sirve para persistir los nodos y relaciones importantes utilizados en algunas de las visualizaciones utilizadas en la solución. La configuración de este servicio se encuentra en la ruta siguiente:

cd /home/opc/neo4j/

Corresponde a un Docker compose que despliega una base de datos Neo4j versión 5.25.1.

Para validar que el servicio está activo se utiliza el siguiente comando:

docker ps -f name=neo4j

Para validar el estatus del servicio del Docker compose, se ejecuta:

Cd /home/opc/neo4j/
docker compose ps
docker compose ps neo4j

Para visualizar los logs del servicio:

/home/opc/neo4j/

docker compose logs -f neo4j

Si se desea ver los logs de manera individual y con más detalle, en la misma ruta, se encuentran mapeados a la carpeta "logs" así:

debug.log
http.log
neo4j.log
query.log
security.log

También hay una carpeta de configuraciones adicionales "conf" en la que se encuentran parametrizados otros aspectos y/o variables importantes del servicio:

usuario: neo4j
clave: 123456789

2.8. Configuraciones específicas - Máquina 3

A nivel de la máquina 3, se realizaron las siguientes configuraciones particulares:

2.8.1. Instalación de los drivers necesarios

Se utiliza el siguiente comando, para validar la versión de la GPU así:

lshw

También se valida las características de la GPU así:

lspci

lshw -C display

Se instalan los siguientes paquetes:

sudo apt install ubuntu-drivers-common

Se validan los devices:

sudo ubuntu-drivers devices

Se realiza la instalación del driver siguiente:

sudo apt install nvidia-driver-570

Si toda la instalación se realiza de manera correcta, se puede ejecutar el comando siguiente y se debe validar toda la configuración de la GPU respectiva:

nvidia-smi

Se procede a instalar el servicio de API rest que se va a utilizar para los modelos de IA:

curl -fsSL https://ollama.com/install.sh | sh

Se valida la configuración del servicio a nivel del OS:

sudo cat /etc/systemd/system/ollama.service

Se valida la configuración de la IP y puerto del servicio:

sudo systemctl edit ollama.service

Se debe seleccionar: Environment="OLLAMA_HOST=0.0.0.0:11434"

Se habilitan los puertos en el firewall de la máquina en caso de necesidad...

Se realiza la actualización del servicio así:

sudo systemctl daemon-reload
sudo systemctl status ollama
sudo systemctl restart ollama

Se puede validar el log del servicio así:

journalctl -e -u ollama

Se puede validar el status del servicio así:

sudo systemctl status ollama

Para la configuración de los modelos de IA, se deben copiar las carpetas que se enviaron en los discos físicos en la siguiente ruta:

/usr/share/ollama/.ollama/models/

Se debe preservar la estructura tal cual para que todo funcione. El modelo más importante utilizado en la solución implementada es una variante del modelo de código abierto Llama3.3 de 70 billones de parámetros. El modelo fue cuantizado a 4 bits. Se utilizaron los comentarios de los distintos analistas para hacer un ajuste a los pesos del modelo.

3. Recomendaciones técnicas a nivel general

A nivel general se sugiere lograr revisar el esquema de comunicación que existe entre la infraestructura on-premise de la SDP y la nube privada en Oracle. A lo largo de la implementación del proyecto, se validaron temas de latencias muy grandes y posiblemente esto se deba a dicho esquema. También se recomienda que la SDP contemple un mejor esquema costo eficiente para poder desplegar los modelos de IA utilizados. Se sugiere conectar dicho despliegue con una integración continua para realizar MLDevOps que le permita a la entidad medir de manera continua los resultados de las inferencias de los diferentes modelos de AI, junto con un mejor modelo de despliegue que permita reducir los tiempos de respuesta de las diferentes inferencias. Adicionalmente se sugiere seguir revisando los criterios de calidad de datos junto con la continuación de la construcción de datos etiquetados para realizar nuevas iteraciones de entrenamiento de los modelos utilizados y seguir así mejorando la exactitud de las inferencias. Esta mejora continua, eventualmente permitirá destilar los entrenamientos en modelos de gran escala de menor complejidad, tratando de mantener al máximo los resultados a nivel de calidad.

4. Plan de soporte

Durante la fase de garantía del Mínimo Producto Viable implementado, se habilitará el correo electrónico siguiente soporte@igg.com.co para recibir cualquier requerimiento relacionado con posibles fallas en la herramienta. Adicionalmente se solicita poner en copia el correo siguiente: luisfgutierrez@igg.com.co. Tan pronto se reciba una comunicación, el equipo de soporte de IGG responde los correos y se asigna un caso de seguimiento. Se notificará a la menor brevedad posible el diagnóstico inicial de la incidencia y si aplica, se comunicarán los tiempos para lograr cualquier corrección respectiva, en caso de que aplique.

Al final de los 3 meses del soporte, se presentará un único informe con los diferentes requerimientos recibidos. Se visualizarán los tiempos de atención para cada incidente y las medidas implementadas. En caso de que se requiera realizar alguna modificación al código desplegado, se procederá a realizarla con su respectiva validación, dejando la evidencia de la solución al problema, en caso de que aplique.

No se considera ninguna garantía a nivel de los modelos de inteligencia artificial, pues hacer cualquier modificación a esos componentes es una tarea que implica un gran esfuerzo. Tal como se acordó con los usuarios, si durante los 3 meses siguientes se van a realizar ejercicios de ejecución de análisis reales, se asignarán unas horas de acompañamiento de nuestro equipo en caso de necesidad.

Todo el soporte de la solución depende de que la VPN configurada para el acceso a los diferentes servicios continúe operativa. Adicionalmente este soporte no podrá resolver ningún problema a nivel de latencia y velocidad en los diferentes procesamientos.

5. Aspectos funcionales

Gran parte de los datos disponibles provienen de fuentes estructuradas y no estructuradas como documentos técnicos, hojas de cálculo y otros archivos clave. En este proyecto, nos centraremos en la extracción de información de fuentes específicas proporcionadas, incluyendo el Plan de Desarrollo Distrital (PDD), el Plan de Ordenamiento Territorial (POT) y los programas, metas e indicadores del PDD contenidos en archivos Excel. Estas fuentes serán la base para los procesos de analítica necesarios para abordar los retos identificados.

Una vez que los datos han sido extraídos, implementaremos procesos avanzados como el reconocimiento de entidades nombradas. Este método permitirá identificar conceptos clave como nombres de personas, entidades, lugares, fechas y otros datos relevantes para los objetivos del proyecto. La información estructurada y sus metadatos serán almacenados en bases de datos especializadas, asegurando su disponibilidad para análisis posteriores.

En relación con los retos planteados, se diseñaron estrategias que maximizan la eficiencia y los resultados:

- Procesamiento de políticas públicas: Actualmente, la revisión manual de 6 políticas toma entre 1 y 2 semanas para una persona. Nuestra solución reducirá este tiempo en al menos un 60%, lo que liberará recursos para actividades estratégicas.
- Análisis de actas y documentos clave: Automatizamos la extracción y estructuración de información contenida en actas y documentos relevantes, permitiendo su visualización y análisis desde perspectivas más funcionales y orientadas a los objetivos del PDD.
- Alineación con necesidades de los usuarios: Abordaremos las expectativas de los diferentes actores mediante un enfoque analítico robusto que optimice los recursos disponibles, sin incluir inicialmente integraciones complejas como autenticación LDAP, visualización en ArcGIS o PowerBI. Esto asegura que las soluciones entregadas sean viables dentro del alcance establecido.

Soluciones proyectadas para las dependencias:
Subsecretaría de Políticas Públicas y Planeación Social y Económica:
Reconociendo la importancia de los datos para las políticas públicas de la ciudad y los desafíos asociados a su aprovechamiento, tras comprender las necesidades de la dependencia al inicio de la Etapa de Implementación, se proyectó un sistema de analítica que permita identificar errores de ortografía, información incompleta y duplicidad de aportes, definiendo la categoría correspondiente, evaluando su impacto, relacionándolo con indicadores y resultados, y generando reportes que evidencian alertas, logros y acciones de mejora. Sin embargo, dentro de los alcances del diseño del PMV (producto mínimo viable) se priorizaron las funciones que buscan facilitar la categorización de aportes en variables estratégicas y la generación automatizada de contribuciones alineadas con el plan de desarrollo y los ODS, quedando proyectadas para futuras etapas, el desarrollo de varias funcionalidades que surgieron de las necesidades identificadas y que pueden variar o aumentar con el uso cotidiano del PMV.

Oficina de Participación y Diálogo de Ciudad:
Entendiendo la importancia de la participación ciudadana y el impacto que tiene en la confianza pública, así como las necesidades particulares de la dependencia, se propuso optimizar la recolección, trazabilidad e intercambio de información mediante un sistema que depure aportes irrelevantes y evalúe dimensiones clave como relevancia estratégica, impacto, factibilidad, innovación y sostenibilidad. Con la intención de facilitar el seguimiento de los ciudadanos a sus aportes y garantizará su relación con áreas específicas de trabajo, se diseña una interfaz gráfica que simplifique la visualización categorizada y permita a los usuarios apropiarse del área de su interés.

Subsecretaría de Planeación de la Inversión:
Se proyectó incluir un sistema de análisis de texto, impulsado por inteligencia artificial, para identificar tendencias clave en la información recolectada por las entidades distritales, clasificándose por temáticas estratégicas. Este sistema pretende implementar un control de calidad basado en criterios definidos que validen coherencia, precisión y alineación temática, detectando inconsistencias y generando reportes sobre

tendencias, alertas y recomendaciones que aseguren datos útiles y confiables para la toma de decisiones. Para futuras etapas de desarrollo se pueden fortalecer las posibilidades del sistema de análisis de texto, teniendo en cuenta el alcance en el corto tiempo que se tenía para el desarrollo del PMV.

La solución al alcance recién mencionada se materializó en los siguientes casos de uso:

- Login
- Vista de página inicial – reporte simplificado de tareas realizadas
- Carga de Documentos
- Visualización procesos de negocio – documentos cargados y procesados
- Visualización de resultados
- Visualización de resultados detallados
- Cierre de sesión

Los casos de uso mencionados anteriormente están asociados a los diferentes roles y/o áreas usuarias y presentan las siguientes descripciones:

Dependencia	Perfil del Usuario	Necesidades específicas identificadas	Casos de uso
Subsecretaría de Políticas Públicas y Planeación Social y Económica Dirección de Formulación y Seguimiento de Políticas Públicas	Especialista en políticas públicas / analista de datos	La Subsecretaría busca una solución innovadora que permita: ● Clasificar los aportes en variables estratégicas alineadas con los objetivos del plan de desarrollo y los ODS. ● Automatizar la generación de contribuciones estratégicas. ● Corregir errores de ortografía. ● Detectar información incompleta. ● Eliminar duplicidades. ● Relacionar indicadores con resultados concretos.	casos de uso: login, inicio de sesión, visita de página inicial – reporte simplificado de tareas realizadas, Carga de Documentos, visualización procesos de negocio – documentos cargados y procesados, visualización de resultados, visualización de resultados detallados .

Dependencia	Perfil del Usuario	Necesidades específicas identificadas	Casos de uso
		• Identificar alertas, logros y áreas de mejora. • Sugerir acciones para optimizar los resultados. • Evidenciar el impacto de las políticas públicas. • Relacionar los resultados con actividades específicas. • Todo esto mediante un sistema basado en analítica y tecnologías avanzadas para optimizar el análisis y el seguimiento de los objetivos estratégicos.	
Oficina de Participación y Diálogo de Ciudad	Facilitadores de participación ciudadana / gestores de información	La Oficina necesita un sistema que permita: • Mejorar la trazabilidad y el intercambio de datos relacionados con la participación ciudadana. • Ordenar, clasificar y agrupar los diferentes tipos de aportes según criterios definidos. • Analizar dimensiones como relevancia estratégica, impacto, factibilidad, innovación y sostenibilidad. • Producir datos categorizados y medibles que faciliten el seguimiento	casos de uso: login, inicio de sesión, visita de página inicial – reporte simplificado de tareas realizadas, Carga de Documentos, visualización procesos de negocio – documentos cargados y procesados, visualización de resultados, visualización de resultados detallados .

Dependencia	Perfil del Usuario	Necesidades específicas identificadas	Casos de uso
		ciudadano y su integración en áreas de trabajo específicas. • Permitir la visualización categorizada de la información. • Facilitar la apropiación de los datos por parte de los usuarios según sus áreas de interés. Estas funcionalidades estarán respaldadas por análisis avanzados y algoritmos que evitarán sesgos, garantizando datos transparentes y útiles para la toma de decisiones estratégicas.	
Subsecretaría de Planeación de la Inversión Dirección Distrital de Programación, Seguimiento a la Inversión y Plan de Desarrollo	Planificadores de inversión / analista de datos	La Subsecretaría de Planeación de la Inversión requiere una solución basada en inteligencia artificial para analizar y clasificar la información recolectada por las entidades distritales según temáticas estratégicas. El sistema deberá: ● Identificar tendencias clave en la información y clasificarlas de manera estratégica. ● Garantizar la calidad de los datos mediante controles	casos de uso: login, inicio de sesión, visita de página inicial – reporte simplificado de tareas realizadas, Carga de Documentos, visualización procesos de negocio – documentos cargados y procesados, visualización de resultados, visualización de resultados detallados .

Dependencia	Perfil del Usuario	Necesidades específicas identificadas	Casos de uso
		automatizados que validen coherencia, precisión y alineación temática. ● Detectar y reportar inconsistencias, generando alertas y recomendaciones para la toma de decisiones. ● Producir reportes detallados sobre tendencias, alertas de incoherencia y acciones sugeridas. La solución integrará análisis de datos cualitativos, categorización estratégica y herramientas de visualización, facilitando la evaluación de propuestas ciudadanas con base en criterios de relevancia estratégica, impacto social y factibilidad. Esto garantizará información confiable y útil para una toma de decisiones informada y efectiva en Bogotá.	

A continuación, se menciona de manera resumida, cómo cada uno de los diferentes casos de uso y/o funcionalidades implementadas, contribuye a los objetivos del proyecto.

Login:

Este caso de uso le permite a los diferentes usuarios ingresar a la solución desplegada de manera controlada y asumiendo un rol que le da acceso a ciertos recursos. Esta funcionalidad contribuye a los objetivos del proyecto, garantizando confidencialidad en la información, segmentación de las distintas funcionalidades a las que puede tener acceso un usuario y/o rol particular.

Vista de página inicial – reporte simplificado de tareas realizadas:
Esta funcionalidad le permite a los usuarios tener un resumen de los últimos procesos de analítica ejecutados en el sistema. Esta funcionalidad contribuye a los objetivos del proyecto, dando una visión general a nivel de los diferentes documentos asociados a los últimos procesos ejecutados. Muestra las fechas de dichas ejecuciones y permite una navegación rápida al detalle de cada uno de los procesos.

Carga de Documentos:
Esta funcionalidad le permite a cada rol particular (asociado a las diferentes subsecretarías), realizar la carga de la información fuente de entrada, con la que se realizan los diferentes procesos de analítica. En cada caso particular, le permite al usuario seleccionar valores y/o diligenciar campos importantes para aplicar una o más reglas de negocio propuestas en la implementación tanto como subir uno o más documentos de soporte que permiten enriquecer el contexto de información que será utilizado de manera posterior por los diferentes modelos utilizados en el proyecto. Esta funcionalidad contribuye a los objetivos del proyecto, permitiendo la integración de todas las fuentes de información externas que son vitales para todos los procesos de análisis implementados.

Visualización procesos de negocio – documentos cargados y procesados:
Este caso de uso le permite a cada usuario visualizar los diferentes procesos de análisis generados por el mismo usuario. Le permite realizar algunas operaciones de filtrado para identificar procesos en el histórico del sistema y poder visualizar el estado general de dichos procesos, la visualización de los distintos documentos de soporte, realizar la revisión de la información original cargada en el sistema y validar todos los aspectos generales a nivel de carga de información. También le permite a los usuarios iniciar y/o continuar un proceso de análisis cargado previamente en la solución. Esta funcionalidad contribuye a los objetivos del proyecto, permitiendo la visualización de todos los procesos agendados en el sistema junto con todos los documentos soportes respectivos. Le permite al usuario visualizar los diferentes estados de cada proceso para hacer un seguimiento a las diferentes fases de ejecución por las que pasa cada uno de los diferentes documentos.

Visualización de resultados:
Esta funcionalidad le permite a cada usuario ver un listado detallado de todos los registros asociado a un proceso de analítica. Le permite identificar en cada caso respectivo y asociado a los diferentes roles de cada subsecretaría, la visualización de las columnas más importantes por registro. Adicionalmente le permite a cada usuario la verificación de todas las canalizaciones y/o procesos de análisis por los que pasa cada uno de los items y/o registros asociados a un documento particular que forma parte de un proceso de análisis.

También le permite la visualización de los diferentes estados asociados a una o más reglas de negocio implementadas siguiendo los lineamientos y/o requerimientos acordados con las áreas usuarias finales. Finalmente, le permite acceder a los resultados de análisis detallados a nivel de cada registro. En esta misma funcionalidad, el usuario podrá utilizar uno o más filtros para segmentar los diferentes registros que conforman los documentos asociados a un proceso de análisis general. Entre los campos más importantes, están todas las columnas establecidas para cada uno de los formatos, junto con uno o más criterios calculados y finalmente uno o más conceptos importantes a nivel de negocio y/o asociados al contexto respectivo de toda la información procesada por la solución.

Visualización de resultados detallados:
Este caso de uso le permite a cada uno de los usuarios y/o roles, la visualización de los resultados generales construidos en la solución. Le permite de manera resumida y detallada, acceder a los diferentes criterios construidos con todos los usuarios respectivos que ayudan a lograr todos los objetivos específicos de la solución. A nivel de detalle, le permite a un usuario validar la aplicación de una o más inferencias asociadas a criterios particulares, a justificaciones, a clasificaciones, a puntuaciones, a conclusiones relevantes, a segmentación de información, a mediciones de calidad, entre otras. Las funcionalidades implementadas, le permiten a los diferentes usuarios, puntuar y dar ranking a una o más de las conclusiones generales identificadas en una o más de las diferentes fases de análisis a nivel de negocio. Adicionalmente le permite validar los criterios calculados asociados al ítem particular, junto con otras visualizaciones que aportan a la confirmación a nivel de coherencia, calidad, relevancia, entre otros. Esta funcionalidad contribuye a los objetivos del proyecto, permitiendo la visualización de todas las inferencias de los diferentes modelos de inteligencia artificial implementados en la solución. Finalmente, les permite a los usuarios que consumen la información, dar un feedback general de los resultados, que podrá ser utilizado en fases posteriores para seguir afinando la solución.

6. Funcionalidades fuera del alcance

En la etapa de prototipado, donde se propuso una solución al Reto planteado, el tiempo para comprender específicamente las necesidades de la SDP y de cada una de las dependencias era muy corto; sin embargo se hizo una propuesta que de manera general y acorde a la comprensión que se tenía en ese momento del Reto, pudiera evidenciar el potencial que tenían algunas herramientas para solventar los posibles requerimientos. A lo largo de toda la etapa de implementación de la solución, en el diálogo constante con los usuarios finales de las tres dependencias, surgieron nuevas claridades sobre las necesidades básicas (varias de las cuales se implementaron en el PMV) así como diversos requerimientos de funcionalidades que valdría la pena validar para etapas posteriores de desarrollo. A continuación, mencionamos algunas de las ideas de los usuarios de las tres dependencias:

Área SPPPSE:

- "Sería interesante poder realizar la visualización a nivel transversal de un producto particular en una política específica. Esto le (sic) permitiría a los analistas del área, verificar si corte a corte, las personas que están diligenciado la información fuente con la que trabaja el sistema están siguiendo las recomendaciones realizadas"
- "Sería interesante lograr construir algunos KPIs que midieran la calidad de los textos que se registran en las diferentes fuentes de datos. En este momento nos toca hacer un trabajo muy largo para poder validar y ejecutar este tipo de análisis..."
- "Quisiéramos poder visualizar la evolución de los criterios calculados a lo largo del tiempo..."
- "Quisiéramos que fuera más simple la identificación de los errores en las fuentes de datos de las cuales se está tomando la información para que el proceso de corrección fuera mucho más rápido..."

Área OPDC:

- "Nos sería de gran utilidad, poder tener un buscador geoespacial de aportes e ideas ciudadanas"
- "Nos gustaría poder modificar, ajustar y probar nuevos criterios de análisis para algunos instrumentos"
- "Sería interesante poder visualizar los diferentes temas que son de interés para una localidad, UPZ o una zona de la ciudad a lo largo del tiempo..."
- "Quisiéramos que los diferentes resultados generados por los modelos, (sic) se pudieran exportar para que la visualización de todo el análisis se pudiera hacer fuera del sistema..."
- "Sería increíble que (sic) en algún momento, los ciudadanos pudieran realizar la trazabilidad de sus aportes/ideas y validará si pudieron ser o no incorporadas dentro de los diferentes proyectos en la ciudad..."

Área SPI:

- "Nos gustaría que no nos tocara cargar la información (sic) sino que el sistema la buscara de manera automática en las fuentes originales..."
- "Sería deseable que pudiéramos ajustar los diferentes criterios de análisis a lo largo del tiempo..."
- "Nos gustaría que muchos usuarios pudieran ingresar al sistema para hacer el proceso de validación y verificación de los resultados más rápido..."
- "Nos gustaría poder integrar las salidas de los datos a los reportes de rendición de cuentas que nos toca hacer luego..."
- "Ojalá que los tomadores de decisiones, (sic) pudieran ingresar al sistema para verificar los datos fuentes de los diferentes análisis que realizamos..."

General:

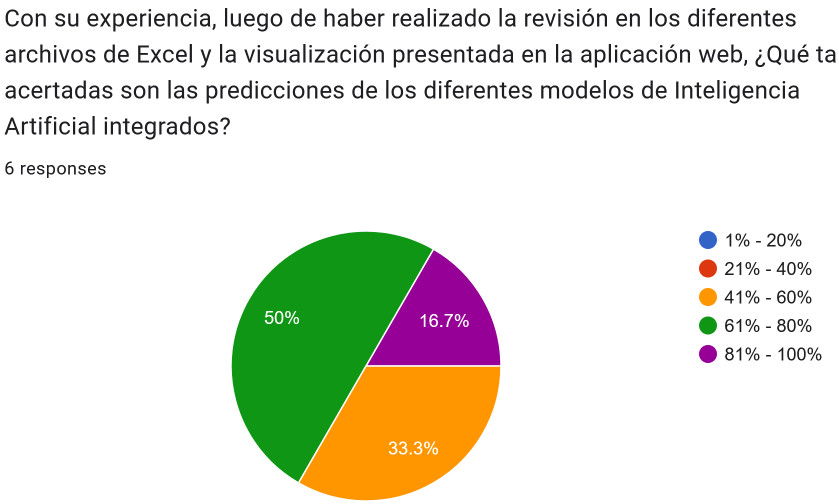
- Hay oportunidades de incluir nuevas visualizaciones de datos para tener una mejor aproximación a los diferentes insights.

- A nivel de la visualización de tópicos, se proponen nuevos modelos que tengan en cuenta información jerárquica para poder ir desagregando los aspectos más relevantes de algunos aportes ciudadanos.
- A nivel de mejoras en el reconocimiento de temas de calidad, se propone maneras de visualizar las mejoras e incorporación de recomendaciones realizadas para cada una de las entidades que generan la información fuente del sistema.
- Tener la oportunidad de realizar la integración con el directorio activo de la SDP para evitar tener que hacer gestión de aprovisionamiento de los usuarios.
- Tener la oportunidad de considerar entrenar nuevos modelos que logren preservar al máximo la calidad de las inferencias y que requieran menos complejidad para su ejecución, implementación y tiempos de latencia.
- Tener la oportunidad de integrar verdaderos procesos de negocio modelados usando tecnologías como BPMN, para que eventualmente se puedan paralelizar de mejor manera todas las reglas de negocio que en este momento se tienen implementadas y se pueda reducir el tiempo de espera para la obtención de los análisis.
- Incluir temas de analítica del uso de la solución, para garantizar que los usuarios gastan el tiempo adecuado dentro de la solución y que eventualmente se pueda rastrear mejoras en el proceso general a nivel de negocio.
- Incluir nuevos casos de uso de IA, en particular generativa, para comenzar a proponer llevar los "insights" identificados y asociarlos a los procesos de generación de los documentos finales que son generados por cada una de las diferentes dependencias.
- Integrar mecanismos que permitan controlar la calidad de las predicciones de los diferentes modelos para realizar mejora continua a toda la parte de analítica.

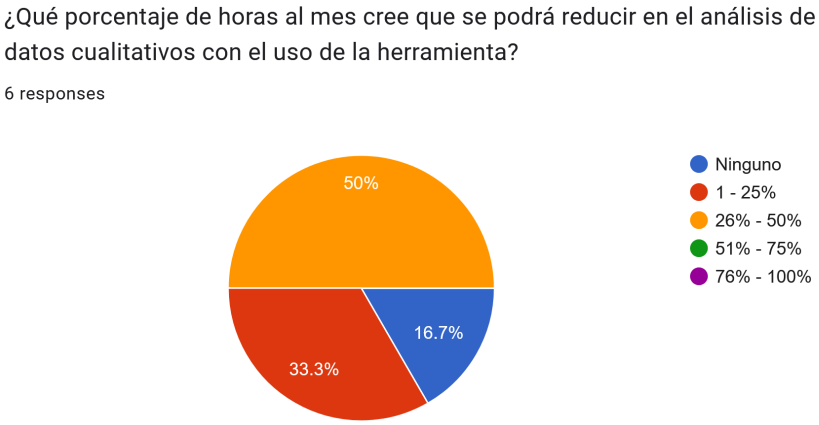
7. Evaluación por parte de los usuarios

Todas las semanas se realizaron pruebas técnicas y se verificaban las funcionalidades de cara a los usuarios. Se realizaron procesos de prueba de los diferentes servicios de canalización, etls, bases de datos, api's de la solución junto con la interfaz de usuario. A nivel general se trató de tener en cuenta al máximo posible, todas las sugerencias y nuevos requerimientos identificados por las diferentes áreas usuarias finales.

Para tener un feedback de los usuarios finales se implementó un formulario que fue diligenciado por los usuarios finales que participaron en el proceso. Se plantearon nueve preguntas alineadas con aspectos asociados al proceso de negocio, de usabilidad, medición de impacto y opiniones generales de los usuarios, entre otros. Dos funcionarios de cada una de las dependencias dieron respuesta a estas preguntas. A continuación se exponen las respuestas



Se confirma que en más del 67% de los casos los usuarios consideran que la exactitud de las predicciones de los modelos de inteligencia artificial utilizados tiene una buena exactitud. Hacia el futuro, esta tarea de seguir validando la calidad de las inferencias se debe seguir revisando de manera más rigurosa y continua.

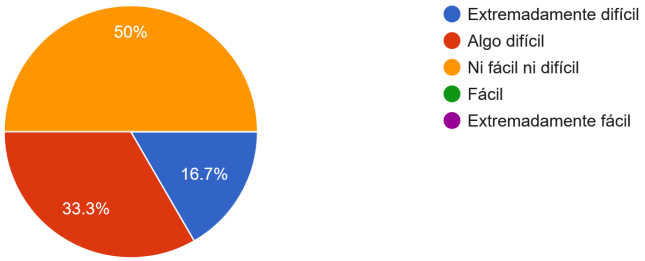


Se evidencia que más del 80% de los usuarios considera que se tendrán reducciones de uso de horas mensuales importantes en sus procesos relacionados con el análisis de la información que posiblemente ahora será realizado en una primera fase por la solución propuesta. Esto es un número importante y se considera que incluso se pueden realizar mejores ahorros al seguir escalando la solución para afinar aún más los criterios implementados y adicionando nuevas funcionalidades que libere más tiempo por parte de los analistas que harán uso de la herramienta.

A nivel de usabilidad, se identifican posibles mejoras que tendrán que ser analizadas de manera conjunta con cada uno de los usuarios encuestados:

Tras haber interactuado con la aplicación, ¿Cuál considera usted que es el nivel de complejidad en el uso de la herramienta?

6 responses



El 33% de los usuarios quisiera tener flujos de experiencia de usuario más simples. A medida que se realicen nuevas integraciones con los sistemas que tienen las fuentes de datos de entrada, muy seguramente estas experiencias serán aún más simples para los usuarios finales y se podrán mejorar el nivel de satisfacción en este sentido. Eso se ha considerado como posibles mejoras futuras.

Con el estado actual de la herramienta, ¿Qué otros usos podría tener para optimizar el análisis de los instrumentos que tiene a cargo la dependencia?

6 responses

Habilitar la base de datos de los aportes ciudadanos analizados a través del aplicativo para consulta general por todas las dependencias permitiría optimizar el análisis de los instrumentos a cargo de la dependencia. Esto facilitaría el acceso a la información sistematizada, promoviendo un uso más eficiente de los insumos ciudadanos en la toma de decisiones y en la formulación de políticas públicas.

Vemos un uso reducido, dado que ante los volúmenes de información manejados por la oficina, no se aprecian reducciones de tiempo u otros usos de la herramienta.

En nuestra dirección los únicos dátos cualitativos que se manejan son los de reportes de seguimiento a las políticas, por lo que no tendría algún uso adicional.

Insumo para consolidación de avance de programas generales

Frente a la pregunta Cómo escalar la solución, los usuarios sugieren lo siguiente:

- La herramienta permite agilizar el análisis cualitativo de las propuestas ciudadanas dentro del marco lógico de un instrumento de planeación específico. Esto facilita la correlación de los aportes ciudadanos con los objetivos planteados, optimizando la sistematización de la información y fortaleciendo la toma de decisiones basadas en la participación ciudadana.
- No se refleja impacto en reducción de tiempos de análisis o calidad de análisis de información.
- Podrá ayudar a los profesionales de la dirección a disminuir el tiempo de lectura y análisis de la información cualitativa y de enfoques de los reportes de la política pública.
- Impacto medio considerando que se debe realizar validaciones de calidad, previo al cargue de la información en la herramienta

- Apoyo en la consolidación de la información de seguimiento cualitativo, optimizando los tiempos de revisión, generando alertas de inconsistencias en la información y destacando los puntos clave.
- El impacto es bastante importante porque aporta al análisis de la información cualitativa, sin embargo, se deberían optimizar los tiempos de respuesta en el procesamiento.

Con relación a pedirles describir el impacto que aporta la herramienta de cara a las necesidades de la Secretaría Distrital de Planeación", los resultados mencionados por los usuarios son:

- Dado que la participación y el diálogo ciudadano es un proceso estratégico y transversal en la Entidad, la herramienta fortalece la capacidad de análisis de la Oficina para desarrollar estrategias de participación que involucren propuestas ciudadanas. Además, contribuye a optimizar el flujo de procesos en la formulación de instrumentos de planificación para la Ciudad, avanzando en la garantía de una mejor integración de los aportes ciudadanos en la toma de decisiones.
- Puede ser un semillero de desarrollo de herramientas en el corto plazo.
- Podrá mejorar los tiempos de respuesta en el análisis y clasificación de la información.
- Medio con relación a que en el plazo inmediato no se visualiza claramente la implementación de la misma
- Generar un análisis más rápido de la información solicitada, enfocado en temas específicos de seguimiento de políticas en datos cualitativos.
- El impacto es bastante importante porque aporta al análisis de la información cualitativa, sin embargo, se deberían optimizar los tiempos de respuesta en el procesamiento.

A la última pregunta sobre la satisfacción general y la relación directa con las expectativas iniciales, los usuarios mencionaron lo siguiente:

- Parcialmente. Si bien la herramienta facilita el análisis cualitativo de las propuestas ciudadanas, presenta limitaciones en su diseño gráfico, lo que afecta la visualización de los datos. Además, la legibilidad y accesibilidad al repositorio de los aportes analizados es compleja y no facilita el contraste global, y el proceso de análisis requiere el mismo o incluso más tiempo que hacerlo de forma manual.
- No, el resultado no cumple con las expectativas
- Si, pero se hace necesario un levantamiento de requerimientos y desarrollo más específico para cada dirección con el fin de que pueda ser utilizado por todos los usuarios internos.
- Teniendo en cuenta que comencé a hacer parte del proceso cerca de la última etapa, no es posible hacer mención hacia expectativas previas
- Dado el corto tiempo de desarrollo, la fase inicial del proyecto cumple con las expectativas. No obstante, es necesario ajustar la herramienta para que sea más fácil de usar, de modo que se pueda aprovechar adecuadamente y satisfacer las necesidades de la Dirección en cuanto a la gestión de información cualitativa
- Se cumplieron parcialmente porque no se finalizó el procesamiento del archivo de la SPI

A nivel general, se identifica que de manera general para la gran mayoría de usuario se logró alinear sus expectativas iniciales. Obviamente y como se menciona en diferentes partes de los documentos, aún hay muchas cosas que pueden y deberían ser mejoradas, para lograr aún una mejor satisfacción de las expectativas iniciales de los usuarios.

La url del formulario utilizado para recolectar los comentarios de los usuarios es:
https://docs.google.com/forms/d/1ezH6oKid-o1DtcgQ547r6jGxlRxAJtqgzi8lmNmeGng/edit?ts=67d98cb4

8. Entregables finales

8.1 Productos de desarrollo

A continuación, se describe el disco duro/USB que será entregado con todo el material necesario para la implementación de la solución. El disco duro contiene las siguientes carpetas en la raíz:

- Front-End
- ETLs-APIBackEnd
- Keycloack
- Storage
- API-Indexación
- Servicio-Canalizaciones
- DB-Postgres
- DB-Neo4j
- DB-Mongo
- Contenedores
- Canalizaciones
- IA
- Manuales

- API-Indexación
- Canalizaciones
- Contenedores
- DB-Mongo
- DB-Neo4j
- DB-Postgres
- ETLs-APIBackEnd
- Front-End
- IA
- Keycloack
- Manuales
- Servicio-Canalizaciones
- Storage

La carpeta "Front-End" contiene un zip con el nombre "front_end-0.3.7". Este archivo contiene la versión final del MVP que está desplegada en los servidores de la SDP. Contiene el material presente en la máquina 1, en la ruta siguiente:

/home/opc/varios/front_end

El contenido de esa carpeta es el siguiente:

angular.json
deps
Dockerfile
e2e
.env_example
generate-config.js
.gitignore
.npmrc
package.json
package-lock.json
README.md
src
tsconfig.app.json
tsconfig.json
tsconfig.spec.json

Este contenido se utiliza para construir la imagen de un contenedor encargado de ejecutar un servidor nodejs y la aplicación web angular que ya se describió en secciones previas.

front_end-0.3.7

La carpeta "ETLs-APIBackEnd" contiene un zip con el nombre "etls-0.3.7". Este archivo contiene la versión final del MVP que está desplegada en los servidores de la SDP. Contiene el material presente en la máquina 1, en la ruta siguiente:

/home/opc/varios/etls

El contenido de esa carpeta es el siguiente:

docker-compose-postgres.yml
docker-compose.yml
Dockerfile
.env
.env_example
.gitignore
logs
pom.xml
PostgresDockerInit.md
README.md
src

Este contenido se utiliza para construir la imagen de un contenedor encargado de ejecutar un servidor de SpringBoot y las diferentes ETLs necesiarias para procesar las diferentes fuentes de información. La carpeta en la USB contiene una copia del archivo "docker-compose.yml" que es necesario para construir la imagen del contenedor. También se visualiza un archivo de texto "env_etls_producción.txt" que corresponde al contenido del archivo en la máquina 1 con el nombre ".env". Este archivo tiene las variables de entorno necesarias para configurar gran parte de los servicios. Para realizar la generación de los dos contenedores necesarios, se debe configurar una variable de entorno con la versión del TAG de los contenedores. En producción se encuentran las imágenes para la versión 0.3.7 como se puede validar:

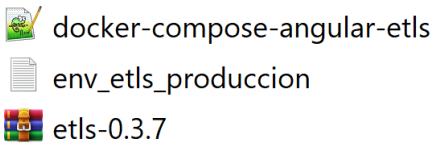
Ejecutar en la máquina 1:

docker image ls

con el resultado siguiente:

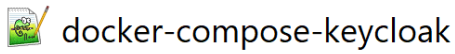
REPOSITORY	TAG
sdp.angular-app	0.3.7
sdp.springboot-app	0.3.7

Estas dos versiones, corresponden a las últimas versiones funcionales de la solución implementada. A continuación, el contenido de la carpeta "ETLs-APIBackEnd":



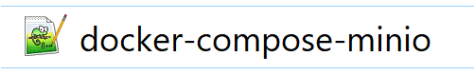
La carpeta "Keycloack" contiene un archivo con extensión .yml que permite configurar el docker-compose que sube el servicio de IAM utilizando en la solución. Este archivo contiene la versión final necesaria para que el MVP implementado en las máquinas de las SDP pueda operar. Su contenido se encuentra en los anexos de este documento también. Se recuerda que la ruta de este docker-compose es (máquina1):

/home/opc/varios/keycloak



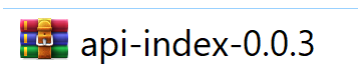
La carpeta "Storage" contiene un archivo con extensión .yml que permite configurar el docker-compose que sube el servicio de storage utilizando en la solución. Este archivo contiene la versión final necesaria para que el MVP implementado en las máquinas de las SDP pueda operar. Su contenido se encuentra en los anexos de este documento también. Se recuerda que la ruta de este docker-compose es (máquina1):

/home/opc/varios/minio



La carpeta "API-Indexación" contiene un archivo zip "api-index-0.0.3" con el contenido de una api rest desplegada implementada en python. Este archivo contiene la versión final necesaria para que el MVP implementado en las máquinas de las SDP pueda operar. Se recuerda que la ruta de este zip es (máquina2):

/home/opc/api



La carpeta "Servicio-Canalizaciones" contiene un archivo con extensión .yml que permite configurar el docker-compose que sube el servicio de canalizaciones utilizando en la solución. Este archivo contiene la versión final necesaria para que

el MVP implementado en las máquinas de las SDP pueda operar. Su contenido se encuentra en los anexos de este documento también. Se recuerda que la ruta de este docker-compose es (máquina2):

/home/opc/zenml/canalizaciones/zenml


docker-compose-canalizaciones

La carpeta "DB-Postgres" contiene un archivo con extensión .yml que permite configurar el docker-compose que sube el servicio de base de datos de Postgres utilizando en la solución. Este archivo contiene la versión final necesaria para que el MVP implementado en las máquinas de las SDP pueda operar. Su contenido se encuentra en los anexos de este documento también. Se recuerda que la ruta de este docker-compose es (máquina2):

/home/opc/postgres


docker-compose-postgres

La carpeta "DB-Neo4j" contiene un archivo con extensión .yml que permite configurar el docker-compose que sube el servicio de base de datos de Neo4J utilizando en la solución. Este archivo contiene la versión final necesaria para que el MVP implementado en las máquinas de las SDP pueda operar. Su contenido se encuentra en los anexos de este documento también. Se recuerda que la ruta de este docker-compose es (máquina2):

/home/opc/neo4j


docker-compose-neo4j

La carpeta "DB-Mongo" contiene un archivo zip "dump-0.1" con un dump en mongo con la base de datos necesaria para iniciar una nueva copia. Este archivo contiene la versión final necesaria para que el MVP implementado en las máquinas de las SDP pueda operar. Se recuerda que la ruta en la que se puede utilizar este dump es (máquina2) y se debe importar al contenedor que ejecuta la DB mongo para poder restaurar la DB


dump-0.1

La carpeta "Contenedores" contiene dos archivos con extensión .tar que contienen la versión 0.3.7 generada para los dos contenedores del "front-end" y la "ETL-ApiBackEnd" que ya fueron mencionados previamente. Estos archivos pueden ser utilizados en la máquina 1 para volver a correr parte de los servicios necesarios para que el MVP implementado en las máquinas de las SDP pueda operar. Se recuerda que, en caso de necesidad, se debe consultar la documentación oficial de "Docker" para poder utilizar estas imágenes.


contenedor-sdp.angular-app-0.3.7


contenedor-sdp.springboot-app-0.3.7

La carpeta "Canalizaciones" contiene un archivo zip "canalizaciones-0.4.6" con el contenido de todos los scripts en python necesarios para ejecutar una o más de las canalizaciones necesarias para analizar la información cualitativa. Este archivo contiene la versión final necesaria para que el MVP implementado en las máquinas de las SDP pueda operar. Se recuerda que la ruta de este zip es (máquina2):

/home/opc/pipelines/pipelines/zen_client/zenml


canalizaciones-0.4.6

La carpeta "IA" contiene una carpeta con el nombre "models" que tiene a su vez dos carpetas a saber "blobs" y "manifests". La carpeta "blobs" contiene los pesos de los modelos utilizados en la solución implementada. Estas carpetas contienen la versión final necesaria para que el MVP implementado en las máquinas de las SDP pueda operar. Se recuerda que la ruta de este zip es (máquina3):

blobs

manifests


sha256-53a87df39647944ad2f0a3010a1d4a60ba76a1f8d5025bb7e76986e966d28ab6


sha256-56bb8bd477a519ffa694fc449c2413c6f0e1d3b1c88fa7e3c9d88d3ae49d4dcb


sha256-948af2743fc78a328dcb3b0f5a31b3d75f415840fdb699e8b1235978392ecf85


sha256-4824460d29f2058aaf6e1118a63a7a197a09bed509f0e7d4e2efb1ee273b447d


sha256-bc371a43ce90cc42fc9abb0d89a5959fbae91a53792d4dcd9b51aa48bd369b06


sha256-c7091aa45e9be6c15e1e5c8d5489d47f18183bf5077b3d3697924e1d18ad1b2a

8.2. Manuales

A nivel técnico, ya se describieron todos los procedimientos para configurar cualquiera de las máquinas virtuales utilizadas para implementar la solución desplegada. También se comentaron los diferentes servicios, contenedores y bases de datos.

La carpeta "Manuales" contiene un archivo en extensión PDF, con el manual de usuario para las áreas finales. Se recuerda que este manual comprende 4 roles principales que pueden utilizar la solución implementada.

Manual de Usuarios Finales

El manual de usuario final según cada uno de los 4 roles, se incluye como un anexo a este documento. Contiene los flujos para realizar los casos de uso principales, a saber:

- Login
- Vista de página inicial – reporte simplificado de tareas realizadas
- Carga de Documentos
- Visualización procesos de negocio – documentos cargados y procesados
- Visualización de resultados
- Visualización de resultados detallados
- Cierre de sesión

8.3. Formación

El Plan de formación de transferencia consiste en una sesión en la que se va a presentar la arquitectura final de la solución y se revisarán todos los servicios, contenedores y bases de datos necesarias para que la solución implementada pueda funcionar. Adicionalmente se revisarán varios procedimientos necesarios para hacer la revisión de los diferentes logs de todos los servicios y temas a nivel de certificados de seguridad.

La metodología y cronograma del plan de formación va a consistir en una sesión virtual en la que se mostrará cómo acceder a todos los servicios necesarios para que la solución pueda operar. Se utilizará una presentación que se comparte como un anexo a este documento en la que se ve la revisión de todos los componentes importantes de la arquitectura, algunas preguntas frecuentes y soluciones para problemas simples. Se espera que algunos de los asistentes a dicha capacitación, puedan realizar algunos ejercicios de consulta a los diferentes logs, tanto como ver el status de algunos servicios, revisar los logs y en general poder hacer algunas actividades de soporte posterior. También se validarán todos los accesos a las diferentes herramientas administrativas de cada servicio. Se revisarán los logs para explicar los flujos de información principales de la solución, entre otros.

Temas Propuestos

1. Vista y Explicación General de la Arquitectura: Revisión general de los componentes, máquinas físicas, servicios, puertos, entre otros.
2. Dominios y Subdominios Necesarios en las máquinas / Reglas de Firewall: Revisión de los registros necesarios en cada máquina para que los servicios funcionen.
3. Proxy Reverso - NGINX: Revisión general de la configuración del servicio, certificados de seguridad, parámetros necesarios, revisión del tráfico.
4. KeyCloak – IAM: Solución utilizada para la gestión de identidades. Aprovisionamiento de usuarios, generación de tokens basados en el protocolo OAUTH.
5. Servidor WEB: Revisión general de la configuración del servicio. Parámetros necesarios, revisión de los logs.

6. Servidor API BackEnd: Revisión general de la configuración del servicio. Parámetros necesarios, revisión de los logs.
7. Servicio de Almacenamiento – Blob Storage - MinIO: Revisión general de la configuración del servicio, certificados de seguridad, parámetros necesarios, revisión del tráfico.
8. Bases de Datos – Mongo, Postgres, Neo4j y MySQL: Revisión general de la configuración del servicio. Parámetros necesarios, revisión de los logs.
9. API de Indexación: Revisión general de la configuración del servicio. Parámetros necesarios, revisión de los logs.
10. Servicio de Canalizaciones: Revisión general de la configuración del servicio. Parámetros necesarios, revisión de los logs.
11. API modelos IA: Revisión general de la configuración del servicio. Parámetros necesarios, revisión de los logs.

9. Anexos

9.1. Configuración nginx

```
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"'
                                        'rt=$request_time  uct="$upstream_connect_time"  uht="$upstream_header_time"
urt="$upstream_response_time"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 4096;
    client_max_body_size 20M;
    client_body_buffer_size 20M;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
    # for more information.
```

```
    include /etc/nginx/conf.d/*.conf;

    server {
        listen       80;
        listen       [::]:80;
        server_name  _;
        root         /usr/share/nginx/html;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        error_page 404 /404.html;
        location = /404.html {
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
        }
    }

# Settings for a TLS enabled server.

    server {
        listen       443 ssl http2;
        listen       [::]:443 ssl http2;
        server_name  _;
        root         /usr/share/nginx/html;

        ssl_certificate "/etc/pki/nginx/server.crt";
        ssl_certificate_key "/etc/pki/nginx/private/server.key";
        ssl_session_cache shared:SSL:1m;
        ssl_session_timeout  10m;
        ssl_ciphers PROFILE=SYSTEM;
        ssl_prefer_server_ciphers on;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        error_page 404 /404.html;
            location = /40x.html {
        }

        error_page 500 502 503 504 /50x.html;
            location = /50x.html {
```

```
    }
}

server {
    listen          443 ssl;
    server_name     isidi.sdp.gov.co;
    root            /usr/share/nginx/html;
    ssl_certificate     /etc/pki/nginx/server.crt;
    ssl_certificate_key /etc/pki/nginx/private/server.key;

    location /ng-cli-ws {
        proxy_pass http://localhost:4200/ng-cli-ws;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    location / {
         proxy_pass http://localhost:4200;
    }
}

server {
    listen          443 ssl;
    server_name     apiisidi.sdp.gov.co;
    root            /usr/share/nginx/html;
    ssl_certificate     /etc/pki/nginx/server.crt;
    ssl_certificate_key /etc/pki/nginx/private/server.key;

    location /ws {
        proxy_pass http://localhost:8080/ws;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    location / {
        proxy_read_timeout 600;
        proxy_pass http://localhost:8080;
        proxy_hide_header Access-Control-Allow-Origin;
```

```
    add_header Access-Control-Allow-Origin https://isidi.sdp.gov.co always;
  }
}

server {
  listen          443 ssl;
  server_name     minisidi.sdp.gov.co;
  root            /usr/share/nginx/html;
  ssl_certificate    /etc/pki/nginx/server.crt;
  ssl_certificate_key /etc/pki/nginx/private/server.key;

  location / {
    proxy_pass https://minisidi.sdp.gov.co:9090;
    #proxy_hide_header Access-Control-Allow-Origin;
    #add_header Access-Control-Allow-Origin https://isidi.sdp.gov.co always;
  }
}

server {
  listen          443 ssl;
  server_name     neoisidi.sdp.gov.co;
  root            /usr/share/nginx/html;
  ssl_certificate    /etc/pki/nginx/server.crt;
  ssl_certificate_key /etc/pki/nginx/private/server.key;

  location / {
    proxy_pass http://10.5.4.136:7474;
    proxy_set_header Authorization "Basic bmVvNGo6MTIzNDU2Nzg5";
  }
}

server {
  listen          443 ssl;
  server_name     pipeisidi.sdp.gov.co;
  root            /usr/share/nginx/html;
  ssl_certificate    /etc/pki/nginx/server.crt;
  ssl_certificate_key /etc/pki/nginx/private/server.key;

  location / {
    proxy_pass http://10.5.4.136:8080;
  }
}

server {
```

```
    listen          443 ssl;
    server_name     apican.sdp.gov.co;
    root            /usr/share/nginx/html;
    ssl_certificate     /etc/pki/nginx/server.crt;
    ssl_certificate_key /etc/pki/nginx/private/server.key;

    location / {
      proxy_pass http://10.5.4.136:5000;
    }
  }

}
```

9.2. Configuración keycloak

```
volumes:
 postgres_data:
  driver: local

services:

 keycloak:
  image: quay.io/keycloak/keycloak:24.0.2
  container_name: keycloak
  restart: always
  environment:
   KEYCLOAK_ADMIN: admin
   KEYCLOAK_ADMIN_PASSWORD: password
   KC_HOSTNAME: iamisidi.sdp.gov.co
   KC_HTTPS_CERTIFICATE_FILE: /opt/bitnami/keycloak/certs/fullchain.pem
   KC_HTTPS_CERTIFICATE_KEY_FILE: /opt/bitnami/keycloak/certs/privkey.pem
  volumes:
   - ./certs:/opt/bitnami/keycloak/certs
  ports:
   - 8000:8080
   - 8443:8443
  command:
   - start
  networks:
   - mybridge
```

```
networks:
  mybridge:
    driver: bridge
```

9.3. Configuración angular y api backend

```
services:
  angular-app:
    image: sdp.angular-app:${TAG}
    build:
      context: ./../front_end
      dockerfile: Dockerfile
    ports:
      - "4200:4200"
    env_file:
      - .env
    command: >
      sh -c "
        node generate-config.js &&
        ng serve --host 0.0.0.0 --disable-host-check --configuration=production
      "
    networks:
      - app-network

  springboot-app:
    image: sdp.springboot-app:${TAG}
    build:
      dockerfile: Dockerfile
    ports:
      - "8080:8080"
    extra_hosts:
      - "iamisidi.sdp.gov.co:10.5.4.115"
      - "minisidi.sdp.gov.co:10.5.4.115"
      - "apiisidi.sdp.gov.co:10.5.4.115"
      - "pipeisidi.sdp.gov.co:10.5.4.136"
      - "neoisidi.sdp.gov.co:10.5.4.136"
    env_file:
      - .env
```

```
    networks:
      - app-network

networks:
  app-network:
    driver: bridge
```

9.4. Configuración MinIO

```
services:
  minio:
    container_name: minio
    image: minio/minio:RELEASE.2025-02-07T23-21-09Z
    environment:
      MINIO_ROOT_USER: SDP_minAdmin
      MINIO_ROOT_PASSWORD: SDP_mi1nio_04dmin
      MINIO_SERVER_URL: 'https://minisidi.sdp.gov.co:9000'
      MINIO_SERVER_HOST: 'minisidi.sdp.gov.co'
      MINIO_DOMAIN: 'minisidi.sdp.gov.co'
      MINIO_BROWSER_REDIRECT_URL: 'https://minisidi.sdp.gov.co:9090'
    volumes:
      - /opt/minio-data:/minio_data
      - /home/opc/minio/certs:/certs
      - /home/opc/minio/minio_certs/certs:/root/.minio/certs
    extra_hosts:
      - "minisidi.sdp.gov.co:127.0.0.1"
    ports:
      - "9090:9090"
      - "9000:9000"
    command: minio server /minio_data --console-address ":9090"
    healthcheck:
      test: ["CMD", "curl", "-f", "https://minisidi.sdp.gov.co:9000/minio/health/live"]
      interval: 30s
      timeout: 20s
      retries: 3

networks:
  default:
    name: minio
```

9.5. Configuración servicio canalizaciones

```
services:
  mysql:
    image: mysql:8.0
    ports:
      - 3306:3306
    environment:
      - MYSQL_ROOT_PASSWORD=sdp_zenml_p4ssword
    volumes:
      - /opt/mysql-data:/var/lib/mysql
  zenml:
    image: zenmldocker/zenml-server:0.73.0
    ports:
      - "8080:8080"
    environment:
      - ZENML_STORE_URL=mysql://root:sdp_zenml_p4ssword@host.docker.internal/zenml
      ## INFO por default - DEBUG
      - ZENML_LOGGING_VERBOSITY=INFO
      - ZENML_SERVER_AUTO_ACTIVATE=1
      - ZENML_DEFAULT_USER_NAME=sdpadmin
      - ZENML_DEFAULT_USER_PASSWORD=sdpadminp4ss
    links:
      - mysql
    depends_on:
      - mysql
    extra_hosts:
      - "host.docker.internal:host-gateway"
    restart: on-failure
```

9.6. Configuración Base de datos postgres

```
services:
  postgres:
    image: postgres:latest  # Usa la imagen oficial de PostgreSQL
    container_name: spring-batch-postgres  # Nombre del contenedor
    environment:
      POSTGRES_USER: admin  # Usuario de la base de datos
      POSTGRES_PASSWORD: admin  # Contraseña del usuario
```

```
    POSTGRES_DB: springBatchDB  # Nombre de la base de datos
  ports:
   - "5433:5432"  # Mapeo de puertos (host:contenedor)
  volumes:
   - /opt/postgres-data:/var/lib/postgresql/data
   - ./dump-spring-batch.sql:/docker-entrypoint-initdb.d/init.sql  # Script SQL para inicialización
  restart: always  # Reiniciar el contenedor si falla
```

9.7. Configuración Base de datos Neo4j

```
services:
 neo4j:
  image: neo4j:5.25.1
  container_name: neo4j
  ports:
   - "7474:7474"  # Interfaz web de Neo4j
   - "7687:7687"  # Puerto Bolt para conexiones
  volumes:
   - "/opt/neo4j-data:/data"  # Cambia al directorio donde se almacenarán los datos
   - "./logs:/logs"
   - "./conf:/conf"
  environment:
   - NEO4J_AUTH=neo4j/123456789  # Configura usuario y contraseña
   - NEO4J_PLUGINS=["apoc", "graph-data-science"]
   - dbms.security.procedures.unrestricted=apoc.,gds.
   - dbms.security.procedures.allowlist=apoc.meta.data,gds.*
   - dbms.logs.http.enabled=true
  networks:
   - project_network

networks:
 project_network:
  driver: bridge
```

10. Conclusiones

La expectativa en esta etapa era lograr mejorar la consistencia y análisis de la información cualitativa proveniente de diversas fuentes, implementando una solución tecnológica integral con un alcance de Producto mínimo viable (PMV) para la Secretaría Distrital de Planeación (SDP). Esta solución debe ayudar a fortalecer la toma de decisiones estratégicas alineadas con los objetivos de desarrollo urbano, optimizando tiempos de procesamiento, y reduciendo horas dedicadas a tareas manuales.

En este sentido, durante la etapa de Implementación se destaca que se logró llegar al nivel de producto esperado, con un producto que permite automatizar la recolección, clasificación y análisis de datos, resaltando también que el equipo de IGG fue constante con la metodología que propuso, y sincronizó el avance del proceso con los encuentros con personas del equipo técnico de la SDP y de Maloka, priorizando la información, necesidades e ideas que dieron los usuarios finales desde la Etapa de Prototipado, hasta las pruebas de usuarios. En este proceso también se pudo documentar el feedback de los usuarios tras utilizar la solución, lo que además de evidenciar el nivel de desarrollo del PMV implementado, también constituye un insumo para la evolución del prototipo.

En relación a los objetivos específicos propuestos para la solución, se dejaron bases de desarrollo que se podrán ajustar, cambiar o ampliar en etapas posteriores, tras usar el producto y evaluarlo por parte de los usuarios de las tres dependencias, incluyendo las nuevas observaciones que surjan y las propuestas de funcionalidades que quedaron documentada en esta etapa. Algunos de los principales avances desde esta perspectiva son:

- Definición de variables estratégicas acordadas con las áreas usuarias finales de la información, que permiten desarrollar un sistema que clasifique los reportes cualitativos.
- Implementación de herramientas que analizan los insumos cualitativos.
- Optimización de reportes para su análisis estratégico

En esta última etapa los principales retos tuvieron que ver con la comprensión de la magnitud del producto desarrollado, y poder integrar el PMV a la infraestructura de la SDP para la implementación. Durante el proceso con los usuarios, también se debía enfatizar el alcance en relación a la expectativa de los usuarios, en gran medida porque no hay conocimiento sobre las posibilidades que tienen las herramientas de inteligencia artificial, o el desarrollo de software.