**Manual Técnico**

Nombre del Aplicativo o Proceso

Desarrollado por

Barranquilla - Colombia

**Tabla de Contenido**

[**Descripción General 3**](#_yjt0th5oo7fl)

[Descripción del Problema 3](#_gno1wsxv6ysc)

[Descripción de la Solución 3](#_rj216e4jgzzk)

[Tecnologías Utilizadas 3](#_r5l24uv1li94)

[**Arquitectura Implementada 4**](#_crq7uea3071j)

[Flujo de Proceso 4](#_aaslhqlzzy66)

[Diseño Base de Datos 4](#_myrtdlf53im7)

[**Descripción de Código Fuente 5**](#_wnov7jq5dued)

[**Paso a Paso para Producción 6**](#_q9721xq225rj)

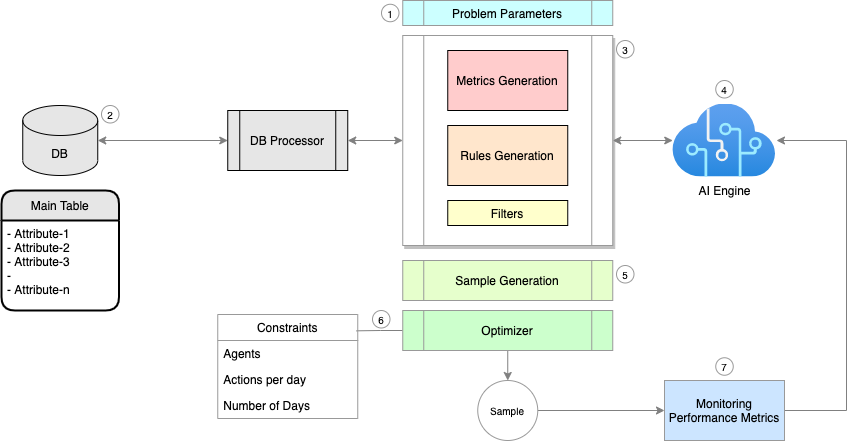
# Descripción General

La prestación de servicios públicos es una actividad que reviste alta complejidad dada la diversidad de los usuarios en términos de consumo, nivel socio-económico y cultura de pago. Las diferentes variantes en los precios, condiciones climáticas y ubicación geográfica lleva a un incremento en la facturación del servicio mensual. Lo anterior genera una cultura de impago y fraude por parte de los usuarios afectando el recaudo de las empresas comercializadoras de servicios y productos. Ante esta necesidad se propone el desarrollo de una herramienta para el análisis virtual de irregularidades (AVI) basado de inteligencia artificial que brinda una guía para la identificar aquellos usuarios con una alta probabilidad de irregularidad. En su arquitectura AVI brinda la posibilidad de integrar diferentes fuentes de datos provenientes de bases de datos relacionales a través de APIs de conexión y carga de archivos planos. Una vez establecida la conexión con las fuentes de datos, la herramienta proporciona al analista la posibilidad de caracterizar todas los atributos incluidos desde la base de datos del cliente a través de estadísticos que dan noción del comportamiento de consumo del usuario. AVI cuenta con funcionalidades que permiten adicionar reglas de clasificación cargadas por el analista que buscan enriquecer la base de conocimiento con la experiencia de los responsables en el proceso de detección de irregularidades. Asimismo AVI está basado en inteligencia artificial, por lo tanto, cuenta con un motor de reglas/base de conocimiento obtenidas a través de un proceso de entrenamiento (clasificación supervisada y no supervisada) con el objetivo de evaluar para cada cliente la probabilidad de fraude en el consumo. Adicionalmente se incluyen filtros con el fin de permitirle al analista delimitar, acorde a las características del usuario, la muestra objeto de estudio. Una vez se ha obtenido un conjunto de usuarios con alta probabilidad de irregularidad en el consumo, AVI brinda la opción al analista de optimizar las campañas de detección de fraude acorde con restricciones como: dimensionamiento operativo, número de días disponibles y/o acciones por día. Finalmente y para hacer seguimiento del desempeño de las reglas de detección y motor de inteligencia artificial, AVI le permite al analista evaluar la evolución de los modelos de clasificación a través de un tablero de indicadores. AVI se encuentra desarrollado en el lenguaje de programación Python haciendo uso del framework Dash.

En esta sección, se proporciona una visión general del problema resuelto, la solución propuesta y las tecnologías que se han utilizado en el proyecto. Se puedes mencionar el lenguaje de programación (Python), así como cualquier librería, framework o herramienta adicional que se haya utilizado para implementar las automatizaciones. Se describe brevemente cada tecnología y su propósito en el proyecto.

## Descripción del Problema

## Descripción de la Solución



El sistema de detección de irregularidades propuesto en la Figura. X. está conformado por los siguientes bloques funcionales.

1. Gestor de base de datos: se plantea la posibilidad de integración con diferentes bases de datos que tengan relación con el proceso de caracterización del consumo del servicio a analizar. El objetivo de esta etapa es permitir la conexión de todas las fuentes de datos basados en modelos relacionales y no relacionales a través de una API que garantice independencia entre la herramienta y la base de datos del cliente. De igual manera, se contará con la opción de carga manual de datos históricos del usuario a través de archivos planos.
2. Gestor de reglas de decisión: en esta etapa se contempla la integración de dos funcionalidades: primero, reglas manuales basadas en la experiencia del usuario que permiten la integración de atributos, métricas y estadísticos relacionados con el comportamiento irregular de los clientes; segundo, reglas obtenidas a partir del motor de inteligencia artificial, las cuales son obtenidas a través de un proceso de entrenamiento que busca relacionar un conjunto de variables predefinidas con las características de un cliente de consumo irregular.
   1. Reglas manuales: este módulo da al analista la posibilidad de configurar cada una de las reglas que considera significativas en el proceso de detección de usuarios irregulares. A continuación, se describen las principales funcionalidades:
      * Generación de métricas: brinda al analista la alternativa de configurar diferentes estadísticos aplicados a los diferentes atributos extraídos de la base de datos de los clientes. Estas métricas se calcularán tomando como referencia la base de históricos de cada variable y estarán disponibles para ser consideradas en la construcción de las reglas por parte del analista.
      * Generación de reglas: proporciona al analista la funcionalidad de construcción de reglas relacionales con la base de datos histórica de atributos y métricas de manera manual; permitiendo identificar posibles clientes irregulares tomando como referencia la base de conocimiento del experto.
   2. Reglas basadas en el motor de inteligencia artificial: de manera automática este módulo proporciona al analista un modelo de clasificación inteligente capaz de establecer cuales clientes tienen una alta probabilidad de irregularidad en su consumo. Para esto, el motor de inteligencia computacional, a través de un proceso de entrenamiento exhaustivo, ensambla de manera óptima un conjunto de reglas que relaciona aquellos atributos y métricas que son significativos dentro del proceso de clasificación.
3. Filtros: dependiendo de las características de los usuarios, por ejemplo, tipo de cliente, estrato, ubicación geográfica, etc., la herramienta le brinda al analista la posibilidad de sesgar la muestra de clientes a analizar con el objetivo de focalizar las campañas a un grupo reducido gracias a los diferentes filtros disponibles.
4. Optimización de las campañas: dada a la posible limitación operativa de la empresa, esta herramienta integra un módulo de optimización no convencional que permite ajustar las campañas de detección dependiendo de las diferentes restricciones que establezca el analista. Lo anterior tiene como objetivo proporcionar al analista diferentes alternativas de abordaje de los diferentes clientes a visitar considerando una ruta óptima y ajustada a la disponibilidad operativa existente.
5. Tablero de desempeño del modelo de detección AVI: con miras al seguimiento del desempeño de cada una de las campañas de detección de irregularidades y su efectividad en la identificación de usuarios con fraude, la herramienta proporciona un tablero de indicadores que sirve de guía al analista para identificar la precisión de las reglas manuales y/o automáticas y su evolución en el tiempo clasificados por los diferentes filtros disponibles, lo anterior, con el fin de evaluar continuamente la necesidad de reentrenamiento o ajuste de cada una de las reglas disponibles.

## Tecnologías Utilizadas

# 

# Arquitectura Implementada

En esta sección, se expone la arquitectura de la solución de automatización. Se explica cómo se estructura y organiza el sistema en términos de componentes y su interacción. Se pueden utilizar diagramas o esquemas para visualizar la arquitectura y describir cada componente, su función y las relaciones entre ellos. En caso de tener bases de datos en la solución, se debe dar una descripción detallada de las tablas y su estructura.

## Flujo de Proceso

## Diseño Base de Datos

# 

# Descripción de Código Fuente

Aquí es donde se proporcionan detalles sobre el código fuente que se ha desarrollado. Se explica la estructura del proyecto, la organización de los directorios y archivos, y cómo se relacionan entre sí. Se describen los módulos o paquetes principales y su propósito. De ser necesario, se mencionan las convenciones de nomenclatura y estilo de codificación que se ha seguido, así como cualquier consideración especial en cuanto a la estructura del código.

# 

# Paso a Paso para Producción

En esta sección, se proporciona una guía paso a paso para llevar las automatizaciones a producción. Se describen los pasos necesarios para configurar el entorno de producción, incluyendo la instalación de dependencias, configuración de variables de entorno, integración con otros sistemas si corresponde, y cualquier otra tarea requerida para poner en marcha la solución de automatización.