**Oficina de Tecnologías de la Información**

Versión 1.0 – agosto 2021

**Data Sandbox MinTIC  
Proyecto Guardian  
SIC  
2021**

Superintendencia de Industria y Comercio

Imagen que contiene reloj, objeto

Descripción generada automáticamente

Contenido

[***1***](#_heading=h.30j0zll) ***Descripción del problema 4***

[***2***](#_heading=h.1fob9te) ***Propuesta de solución 4***

[2.1](#_heading=h.hplxvut7t81o) Pre-procesamiento 4

[2.1.1](#_heading=h.3znysh7) Formato imágenes 4

[2.1.2](#_heading=h.2et92p0) resolución 4

[2.2](#_heading=h.tuqouizf8rxx) Carga de Datos 4

[2.3](#_heading=h.s9mkszp1eaz2) Extracción con Form Recognizer 5

[**2.3.1**](#_heading=h.j0xt266vg4hr) **Agrupación de la información 5**

[**2.3.2**](#_heading=h.gqabzdy54aqc) **Entrenamiento de los modelos 5**

[2.3.3](#_heading=h.3emlh7jyujqe) Extracción 5

[2.4](#_heading=h.ukjxqr298kc6) Post-procesamiento 5

[2.5](#_heading=h.dr2bofltajcy) Cargue al Synapse 5

[2.6](#_heading=h.1t84brkdowi2) Reporte 6

|  |  |  |
| --- | --- | --- |
| **NOMBRE DOCUMENTO** | Data Sandbox – Guardian | |
| **VIGENCIA** | 2021 | |
| **CREADO POR** | Julián Zabala - Javier Andres Arias Sanabria | **Fecha**: Noviembre, 2021 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Control de cambios** | | | | | |
| **Versión** | **Fecha** | **Descripción del cambio** | **Elaboró** | **Revisó** | **Aprobó** |
| 1.0 | 20/10/2021 | Creación del Documento | Julián Zabala - Javier Andres Arias Sanabria |  |  |

# Descripción del problema

La ley 100 de 1993 y la ley 1438 de 2011, confieren a la Comisión Nacional de Precios de Medicamentos y Dispositivos Médicos – CNPMDM, la formulación y regulación de la política de precios de medicamentos

La ley 1438 de 2011, faculta a la Superintendencia de Industria y Comercio para multar a cualquier entidad, agente o actor en la cadena de producción, distribución, comercialización u otras formas de intermediación de medicamentos, dispositivos médicos o bienes del sector salud, cuando infrinjan el régimen aplicable al control de precios de medicamentos o dispositivos médicos.

# La Superintendencia de Industria y Comercio tiene asignada la competencia de inspección, vigilancia y control del cumplimiento de los regímenes de control de precios establecidos para los siguientes productos:

# Agroquímicos: según las disposiciones del Ministerio de Agricultura y Desarrollo Rural bajo el régimen de libertad vigilada (reporte de precios).

# Compra de leche cruda: según las disposiciones del Ministerio de Agricultura y Desarrollo Rural bajo el régimen de libertad regulada (todas las personas naturales o jurídicas que adquieran leche cruda con fines industriales o comerciales deben aplicar el sistema de pago establecido por el Gobierno Nacional).

* Medicamentos y dispositivos médicos: según las disposiciones del Ministerio de Salud y Protección Social bajo dos (2) regímenes: (i) libertad vigilada (reporte de precios) y (ii) Control Directo (precio máximo).

La inspección, vigilancia y control se realizan actualmente mediante un proceso de análisis y comparaciones realizados por los expertos de la delegatura de metrología de la SIC en forma manual y puntual.

# Propuesta de solución

Este proceso tiene posibilidades de mejora:

La cantidad de medicamentos que se pueden vigilar está limitada por la capacidad de los expertos que revisan en SISMED

La cantidad de facturas que se pueden revisar se reduce a las que se piden cuando se encuentra una alerta, y en la cantidad que pueden revisar los expertos de la delegatura de forma manual

La cantidad de información disponible para revisar puede aumentar notoriamente si se cambia el proceso con la ayuda de herramientas de analítica

La revisión de las facturas se puede mejorar con herramientas de analítica para procesar un número mucho mayor mediante componentes para análisis de imágenes y texto

### Pre-procesamiento

En este proyecto se elaboró un script en Python con el fin de pre-procesar los archivos con los que se trabajó. El preprocesamiento consiste en recibir una carpeta con archivos que están en formato pdf y corresponden a las facturas de cada entidad, crear una carpeta para cada factura y en ella guardar por separado las páginas que contiene el pdf convertidas a formato jpg.

### Formato imágenes

Se utilizó la librería de Python llamada pdf2image para convertir las páginas de las facturas en imágenes, especificando el formato como JPG y la escala de colores blanco y negro.

### resolución

Para poder trabajar con las imágenes también fue necesario modificar su resolución para así disminuir también su peso. Se utilizó la misma librería pdf2image y el valor utilizado para la resolución fue 75 dpi.

Este pre-procesamiento se realizó en un equipo local ya que se debe tener instalado el programa Poppler. A continuación, se presenta el script utilizado para este proceso.

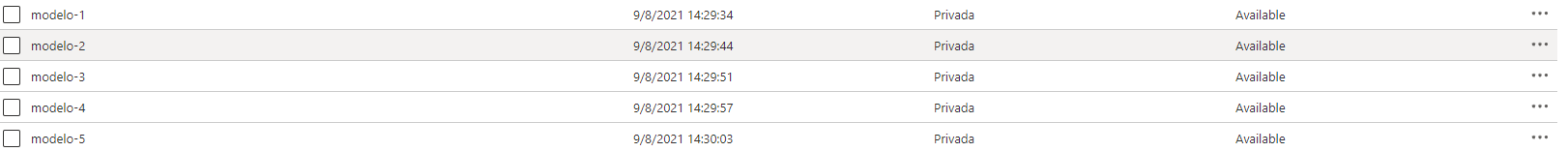
Texto

Descripción generada automáticamente

### Carga de Datos

En este proyecto utilizamos como lago de datos la herramienta de Microsoft Azure llamada Data Lake, en la cual guardamos los documentos que fueron preprocesados.

## Agrupación de la información

Antes de cargar la información se realizó una agrupación de las facturas en distintos modelos que tuvieran características similares, esto con el fin de garantizar buen rendimiento. Se utilizaron los Datos para entrenar modelos resultantes del post-procesamiento.

## Carga de los Datos

El cargue de los datos a la herramienta se realizó de forma manual y considerando una clasificación de 2 tipos:

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente- Datos para entrenar Modelos: Inicialmente se cargan las imágenes resultantes del pre-procesamiento que están en formato JPG (no se agrupan por carpetas, se cargan solo los JPG). En la misma carpeta queda guardada la información correspondiente al entrenamiento de cada uno de los modelos que se almacena en formato JSON y contiene el OCR de los documentos y los labels que fueron entrenados.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente- Datos para Ejecución de los modelos: Estos datos fueron cargados teniendo en cuenta la entidad a la que corresponden (las entidades están agrupadas en los modelos) y se cargan las carpetas con cada una de las páginas de las facturas luego de que hayan pasado por el pre-procesamiento.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

### Extracción con Form Recognizer

En este proyecto se utilizó la aplicación de Microsoft office llamada FormRecognizer para el reconocimiento de texto de los archivos preprocesados y así mismo para la extracción de los campos requeridos

## Entrenamiento de los modelos

Para realizar la extracción fue necesario entrenar los modelos para reconocer los campos que se busca extraer de cada factura, los modelos entrenados fueron 5 y luego con una de las estrategias que tiene la herramienta se diseñó un modelo compuesto que tiene en cuenta cada uno de los modelos individuales.

A continuación se muestra la pantalla de entrenamiento, a la cual se accede luego de configurar la conexión al blobstorage donde se encuentran los Datos para entrenar Modelo.

Interfaz de usuario gráfica

Descripción generada automáticamente

El primer paso para el entrenamiento consiste en ejecutar el layout para todos los documentos, luego se crean los Labels (etiquetas) de acuerdo a los campos que deben ser extraídos, para finalmente marcar o etiquetar los campos en la imagen y asociar cada etiqueta hecha al Label previamente definido. Una vez se hayan realizado todas las etiquetas correspondientes, se procede a entrenar el modelo seleccionando la tercera opción del menú que se ve a la izquierda en la pantalla de entrenamiento, a continuación se presenta un ejemplo de un modelo entrenado, con sus respectivos porcentajes de acierto.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Este proceso se realiza para los 5 modelos definidos en la carga de Datos, y al final, seleccionando la cuarta opción del menú en la pantalla de inicio se define el modelo “composed” que contempla la información de todos estos.

Captura de pantalla de un celular

Descripción generada automáticamente

### Extracción

Por último, se ejecutó con la ayuda de Databricks un script que permitiera consumir el servicio de FormRecognizer y ejecutar el modelo compuesto obtenido de los entrenamientos para los archivos denominados “Datos para Ejecución de los Modelos” cargados previamente al DataLake. Esta información se guardó en formato JSON en el mismo Datalake en una carpeta de resultados.

A continuación, se muestran unos fragmentos como ejemplo del JSON obtenido. El código se encuentra en el repositorio github.

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

### Post-procesamiento

En este proyecto se realizó un script también en databricks para hacer un post-procesamiento de los datos extraídos. Una vez ejecutado el modelo y cargada la información obtenida en formato JSON de las facturas en el Datalake se hace una limpieza de datos de cada uno de los campos, se calculan los campos adicionales que no están en la factura pero se requieren para el análisis, tales como descuento, diferencia con respecto al precio de control, entre otras, para finalmente consolidar la información en archivos de Excel. Esto para garantizar el posterior cargue a la base de datos. El código de este post-procesamiento se encuentra en el repositorio github.

### Cargue al Synapse

En este proyecto se utilizó como servicio de base de datos la herramienta [Azure Synapse Analytics](https://azure.microsoft.com/es-es/services/synapse-analytics/). La información fue cargada desde Azure Databricks.

* + 1. Modelo Base de Datos

Primero se diseñó la estructura del modelo de la base de datos, teniendo en cuenta los campos extraídos de las facturas. Se creó un modelo dimensional, que cuenta con las dimensiones correspondientes a las Entidades, los Clientes, las Fechas (que cuenta con una tabla de referencia llamada Festivos) y los Medicamentos. Para registrar los datos correspondientes a las líneas de las facturas se creó una tabla de hechos y a su vez, para tener mayor control de la información relacionada con una factura se creó la dimensión Factura. Adicionalmente se construyeron una serie de vistas que virtualizan la consulta de la información y permiten la generación de los informes. Esto puede evidenciarse en la siguiente imagen.

* + 1. Conexión y cargue de la información

Una vez definido el modelo de datos, se realizó la conexión al servicio de Azure Synapse Analytics mediante un script de Databricks utilizando también Apache Spark. Finalmente, se ajusta la precisión y los formatos de cada uno de los campos y se realiza el cargue de los datos que habían sido constituidos en los archivos de salida y guardados en la sección de resultados del DataLake. Este proceso es muy importante y se requiere de una alta precisión a la hora de consolidar los formatos de cada uno de los campos que van a ser cargados, deben ser exactamente iguales a los formatos definidos en el modelado de los Datos. El código se encuentra en el repositorio github.

### Reporte

En este proyecto definimos que para la visualización de los datos utilizaremos la herramienta Microsoft Power BI, se diseñaron tableros que resumen la información utilizada por la delegatura y análisis adicionales de los datos obtenidos de las facturas. Este reporte se consolida mediante la construcción de un cubo (Sistema de bases de datos solar) mediante el servicio de Azure Analysis Services el cual se alimenta de las vistas definidas en el modelado de datos.

Diagrama

Descripción generada automáticamente

En el cubo también se definieron unas medidas adicionales relacionadas con las facturas para así poder generar los reportes, estas medidas son: Cantidad de facturas, cantidad de productos, Codigo Cumple Precio Circular, Es medicamento Regulado. Una vez cargada la información, se pueden generar informes como el que se presenta a continuación, en el que mediante una serie de filtros se pueden hacer las consultas necesarias para analizar los resultados obtenidos.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

# Conclusiones

* La herramienta FormRecognizer aún está en desarrollo y para los casos aplicados no soluciona la necesidad de automatizar el proceso de revisión.
* Para obtener mejores resultados al entrenar los modelos con la herramienta FormRecognizer, se requiere que los formatos de las facturas tengan similitudes con respecto a la manera en la que se presentan los datos.
* Se pudo completar el ciclo completo (pre-procesamiento, entrenamiento, extracción, post-procesamiento, carga) para 2 entidades correspondientes a un modelo. Se realizó una prueba para otras 2 entidades, pero los resultados obtenidos no fueron consistentes.
* La duración del procesamiento de las facturas también es un factor que influye considerablemente, ya que, para algunos casos, realizando pruebas para paquetes de 20 facturas el tiempo de corrida alcanzaba las 9 horas.
* La agrupación de las entidades de acuerdo a su similitud para realizar el entrenamiento y el esquema de datos utilizados en el proyecto sirven de base para la automatización que sigue siendo el objetivo principal.