



**INDEXACIÓN DE DOCUMENTOS DIGITALES POR MEDIO DE  
LA EXTRACCIÓN AUTOMÁTICA DE NIT Y FIRMA**

Autor  
Jose Luis Henao Londoño

Universidad de Antioquia  
Facultad de Ingeniería  
Medellín, Colombia  
Julio 2020



# Indexación de documentos digitales por medio de la extracción automática de NIT y firma

Jose Luis Henao Londoño

Informe Final Práctica Académica Presentado Como Requisito Para Optar al Título  
de:  
Ingeniero Electrónico

Asesores (a):

Ricardo Andrés Velásquez  
Docente

Mauricio Cano Restrepo  
Gerente

Universidad de Antioquia  
Facultad de Ingeniería  
Departamento de Ingeniería Electrónica y Telecomunicaciones  
Julio 2020  
Medellín, Colombia

## Resumen

Actualmente Bancolombia se encuentra en el proceso de migrar su información documental a plataformas en la nube. Además, dentro de sus proyectos de desarrollo futuros Bancolombia busca utilizar nuevas tecnologías ofrecidas por empresas proveedoras de servicios tecnológicos a nivel mundial para convertir algunos procesos realizados de forma manual a automáticos.

Con base a estas perspectivas de desarrollo y exploraciones de nuevos servicios que están a la vanguardia de los avances tecnológicos como *Machine Learning*, *Deep Learning*, *Inteligencia Artificial*, se recibe la propuesta para desarrollar un proyecto el cual se resume de la siguiente manera: explorar, investigar, hacer pruebas y desarrollar una aplicación que se encargue de la identificación de la firma y el respectivo número de identificación de la persona a la que pertenece, tomando la información plasmada en un formulario escaneado llamado "tarjeta de firmas".

Como resultado del trabajo se desarrolló una aplicación que realiza la extracción automática del segmento donde se encuentra la firma y envía los fragmentos a un repositorio indexados con el respectivo número de identificación personal del firmante, utilizando el OCR de Amazon Textract, la librería de visión artificial OpenCV y el lenguaje de programación Python.

Por último, este proyecto cumplió con los objetivos propuestos, se siguieron los protocolos de seguridad en el uso del software, se cumplió con los requisitos establecidos para el desarrollo y con base a las pruebas realizadas se obtuvo una efectividad del 93.5% en el proceso de indexación de los segmentos de las firmas de forma automática.

## Contenido

### 1. Introducción

### 2. Objetivos

- 2.1. Objetivo general
- 2.2. Objetivos específicos

### 3. Marco teórico

- 3.1. Servicios en la nube
  - 3.1.1. Amazon Textract
  - 3.1.2. API Amazon Textract
  - 3.1.3. Amazon S3
- 3.2. Librería de visión artificial OpenCV
- 3.3. Lenguaje de programación Python
- 3.4. Formato JSON

### 4. Metodología

- 4.1. Acceso a los servicios de Amazon
- 4.2. Acceso a Amazon Textract
  - 4.2.1. Ensayos del funcionamiento del OCR de Amazon Textract
  - 4.2.2. Análisis del archivo JSON entregado
  - 4.2.3. Análisis de un bloque en el archivo JSON
- 4.3. Configuración de uso de la API de Amazon Textract
- 4.4. Trabajando desde Python con la API de Amazon Textract
  - 4.4.1. Enderezado de las imágenes
  - 4.4.2. Función para el llamado de la API
  - 4.4.3. Extracción de las coordenadas
  - 4.4.4. Segmentación del recuadro de la firma
  - 4.4.5. Extracción de los números de identificación personal
  - 4.4.6. Indexación de la imagen con el número de identificación personal

### 5. Resultados y análisis

- 5.1. Resultados procesamiento digital de imágenes con OpenCV
- 5.2. Resultados del enderezado de la imagen
- 5.3. Resultados del indexado de la imagen
- 5.4. Estructura final del proyecto
- 5.5. Costos de la implementación del proyecto

### 6. Conclusiones

### 7. Referencias bibliográficas

## 1. Introducción

La Gerencia de Servicios de Información Digital del Grupo Bancolombia, busca una solución a un proceso que se estaba quedando sin la capacidad para atender a la demanda. Los usuarios (que en este caso corresponde a otra área del banco) observan cómo el proceso utilizado para la validación de firmas, o el término utilizado en el banco como “visación de firmas”, ya no cuenta con la capacidad ni el soporte adecuado para el volumen de operaciones, algunos de los procedimientos son realizados de forma manual, por lo tanto, las consultas y los tiempos de ejecución son lentos, de allí surge la necesidad de buscar alternativas para agilizar o incluso automatizar dicho proceso.

Este proyecto plantea una solución basada en la innovación, la exploración de nuevas tecnologías y la automatización en el proceso de extraer información de documentos físicos mediante la utilización de Python, Amazon Textract y OpenCV.

Los resultados obtenidos en las pruebas realizadas a este proyecto mostraron un buen comportamiento y una confiabilidad de más del 90% en su funcionalidad, además cuenta con la tecnología y formó parte de su desarrollo los servicios especializados de Amazon Web Services (AWS). Esta plataforma digital brinda servicios muy completos y confiables para muchos desarrollos tecnológicos, para este proyecto se utilizó su OCR, ofrecido mediante la herramienta tecnológica llamada Amazon Textract la cual utiliza sistemas avanzados de inteligencia artificial para el reconocimiento de caracteres.

En este informe se describe el proceso de investigación, exploración de herramientas, el desarrollo del software de la aplicación y las pruebas realizadas para comprobar su funcionalidad. Se destaca el hecho de que el desarrollo total de este nuevo proceso de visación de firmas en el banco, se vio alterado por circunstancias complejas a nivel mundial que afectaron la economía y las rutinas normales de las empresas, por este motivo este proyecto se trabajó sobre un documento tentativo, sin embargo, el desarrollo contempló modificaciones que pudieran tener en un futuro las tarjetas de firmas.

## 2. Objetivos

### 2.1. Objetivo general

Desarrollar una aplicación que extraiga de forma automática el NIT y la firma de un documento digital utilizando *Amazon Textract* y *OpenCV*, de tal forma que permita disminuir el tiempo empleado en la tarea de cotejo de firmas de clientes en los procesos del grupo Bancolombia.

### 2.2. Objetivos específicos

1. Establecer las necesidades y requisitos funcionales para el proceso de indexación de documentos digitales por medio de la extracción automática de NIT y firma.
2. Desarrollar una aplicación siguiendo los lineamientos y las normas de seguridad de Bancolombia para el proceso de indexación, utilizando el servicio en la nube *Amazon Textract*, la librería de visión artificial *OpenCV* y lenguaje de programación *Python*.
3. Evaluar el desempeño de la aplicación mediante pruebas de concepto.

### 3. Marco teórico

#### 3.1 Servicios en la nube

Los servicios en la nube proporcionan acceso a herramientas de solución que se utilizan a través de internet, es decir programas que no están físicamente instalados en nuestro computador. Se trata de un nuevo paradigma producto de la evolución de la World Wide Web.

Entre los beneficios ofrecidos a nivel personal o laboral por los servicios en la nube encontramos: contenedores de grandes cantidades de información, emuladores de una computadora, herramientas entrenadas que brindan una respuesta a una necesidad en especial, entre otras. La capacidad de brindar solución a necesidades por parte de dichos servicios es innumerable, además la competencia por parte de los proveedores de servicios ofrezcan más y mejores opciones aumenta cada día.

A continuación, se da una descripción del proveedor de servicios de nube utilizado y de las herramientas disponibles en esta plataforma que fueron empleadas para el desarrollo de este proyecto.

**Amazon Web Services:** mejor conocido como AWS, "Es la plataforma en la nube más adoptada y completa en el mundo, que ofrece más de 175 servicios integrales de centros de datos a nivel global. Millones de clientes, incluyendo las empresas emergentes que crecen más rápido, las compañías más grandes y los organismos gubernamentales líderes, están utilizando AWS para reducir los costos, aumentar su agilidad e innovar de forma más rápida." (Amazon, *Informática en la nube con AWS*, 2020).

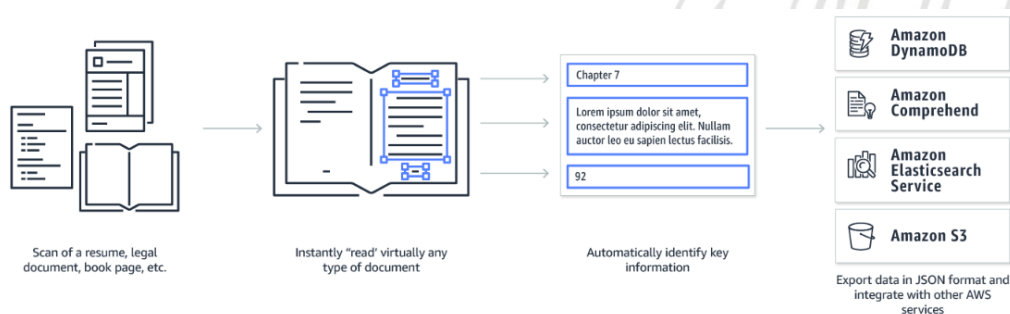
Amazon ofrece servicios de muy variadas características. En la actualidad una de las mayores necesidades entre las personas, tanto en el ámbito personal como laboral, es la necesidad de almacenar información de forma segura, confiable y accesible en cualquier lugar y momento. Las nuevas tecnologías ofrecen formas más eficientes e inteligentes de afrontar estos retos, uno de ellos es el análisis documental. La cantidad de documentos que muchas empresas requieren analizar es demasiada, y

es evidente el tiempo que necesita una persona para analizar y extraer información manualmente es mucho mayor comparado con el tiempo requerido por un sistema automático.

Adicional a estos servicios mencionadas, Amazon ha buscado también dar solución en otros aspectos como: transcripción de audio a texto, reconocimiento de imágenes, entrenamiento de modelos personalizados de *Inteligencia Artificial*, *Machine Learning* y *Deep Learning*, entre otros.

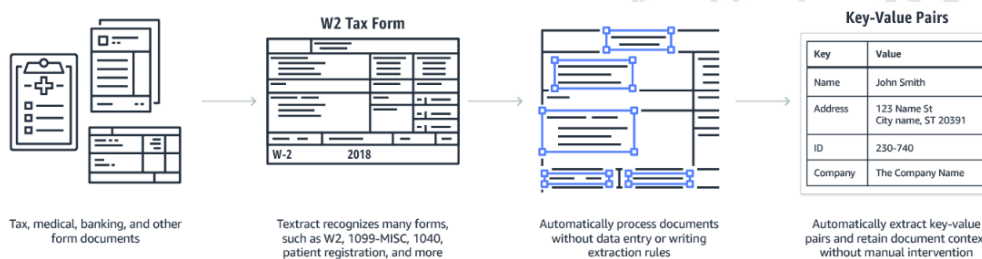
### 3.1.1 Amazon Textract

“Amazon Textract es un servicio que extrae automáticamente texto y datos de documentos escaneados. Amazon Textract no se limita al simple reconocimiento óptico de caracteres (OCR), sino que también identifica el contenido de campos en formularios e información almacenada en tablas” (Amazon, Amazon Textract, 2020). En la figura 3.1 se observa de manera ilustrativa la funcionalidad de la tecnología OCR ofrecida por la plataforma de Amazon mediante su herramienta Amazon Textract, la cual se utiliza para la detección automática de texto impreso ya sea un documento escaneado o una imagen.



**Figura 3.1.** OCR de Amazon Textract. Imagen tomada de: (Amazon Web Services I. o., 2020)

En la figura 3.2 se muestra un esquema detallando de manera general la funcionalidad de Amazon Textract, una de sus ventajas principales es que permite detectar los pares *key-value* (ejemplo nombre: Ana, donde nombre es *key* y Ana el *value*) en un documento, esto indica que la herramienta identifica si el formato presente en la imagen o documento es una tabla, formulario o un texto.



**Figura 3.2.** Identificación de formatos en una imagen. Imagen tomada de: (Amazon Web Services I. o., 2020)

Hoy en día, muchas empresas extraen datos de documentos o formularios de forma manual, este proceso que se torna lento y costoso cuando la cantidad de documentos a analizar es considerable, por lo tanto, se requiere más personal a cargo para lograr realizar la tarea en poco tiempo.

En otras empresas utilizan programas o software que debe ser configurado para cada proceso en particular, también representa una menor eficiencia cuando se debe procesar documentos con características diferentes.

Amazon Textract supera estos inconvenientes utilizando el aprendizaje automático para extraer de manera instantánea el texto en cualquier tipo de documento sin la necesidad de realizar cambios manuales ni utilizar código personalizado. Textract es una nueva Inteligencia Artificial (IA) que puede extraer texto de documentos, es un OCR (Reconocimiento Óptico de Caracteres) mejorado que permite mantener inclusive el formato del documento. El uso del OCR consiste en reconocer los caracteres escritos o impresos y digitalizarlos de modo que se pueda seleccionar y editar si es necesario.

### **3.1.2 API Amazon Textract**

La Interfaz de Programación de Aplicaciones (API) es un conjunto de definiciones y protocolos utilizados para desarrollar e integrar el software de las aplicaciones. Una API permite que un producto o servicio se comuniquen con otros sin la necesidad de saber cómo están implementados; esto simplifica el proceso de desarrollo de aplicaciones permitiendo ahorrar tiempo y dinero.

La API de Amazon Textract admite diferentes formatos de imágenes como escaneados, PDF y fotos (JPEG, JPG, PNG); es posible tomar las imágenes que están almacenadas en el repositorio Amazon S3 y pasarlas por la aplicación para extraer su contenido, luego toda esta información digitalizada o texto líquido puede ser de nuevo almacenada o pasar a otros servicios de aprendizaje automático también disponibles en AWS.

Mediante la documentación disponible para desarrolladores en la página oficial de Amazon Textract, se puede obtener los métodos y formas de hacer el llamado al servicio de API desde la nube o de manera local en una computadora personal. En la nube de Amazon puede ser corrido AWS Lambda, el cual “es un servicio de informática sin servidor que ejecuta código en respuesta a eventos y administra automáticamente los recursos informáticos subyacentes”. (Amazon, Características de AWS Lambda, 2020)

### **3.1.3 Amazon S3**

“Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes en el sector.” (Amazon, Amazon S3, 2020).

Este servicio puede ser usado por cualquier sector o cliente para el almacenamiento y protección de cualquier cantidad de datos. Estos pueden ser utilizados para diversos usos entre los cuales se encuentran procesos de copia de seguridad y restauración, operaciones de archivado, aplicaciones empresariales, análisis de *big data*, entre otros.



Amazon Textract puede analizar imágenes que se encuentran almacenadas en un *bucket* (contenedor de objetos de S3) de Amazon S3.

### 3.2 Librería de visión artificial OpenCV

“OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión artificial y aprendizaje automático de código abierto. OpenCV se creó para proporcionar una infraestructura común para aplicaciones de visión por computadora y para acelerar el uso de la percepción de máquinas en los productos comerciales” (OpenCV, 2020). OpenCV dispone de una gran cantidad de algoritmos optimizados, los cuales incluye un conjunto de algoritmos de visión por computador y aprendizaje automático. Estos algoritmos pueden ser utilizados para detectar y reconocer caras, identificar objetos, rastrear movimientos de cámara, rastrear objetos en movimiento, encontrar imágenes similares en una base de datos, etc. Según la página oficial de OpenCV, esta es una librería completamente libre y gratuita que permite ser usada libremente en distintos proyectos con propósitos de investigación y comerciales, siempre y cuando cumpla con las condiciones de la licencia. Las librerías pueden instalarse tanto bajo Linux como Windows.

OpenCV cuenta con una buena eficiencia computacional y con un fuerte enfoque en aplicaciones en tiempo real. Está escrito en C y C++ optimizados, y se puede ejecutar en GNU/Linux, Windows, Android, iOS y Mac OS. Hay un activo desarrollo en las interfaces de Python, Ruby, Matlab y otros lenguajes.

Con OpenCV se cuenta con diferentes funciones que permiten procesar imágenes, ya sea para resaltar o descartar zonas de esta para su posterior análisis o simplemente si se desea transformar la imagen original para mejorar el color, añadir algún efecto, mezclar la imagen, etc.

### 3.3 Lenguaje de programación Python

Python es un lenguaje de programación de propósito general poderoso y flexible, además es sencillo y fácil de aprender. Es un lenguaje de alto nivel, el cual permite procesar fácilmente todo tipo de estructuras de datos numéricos y de texto. El lenguaje fue creado a principio de los noventa por Guido van Rossum en los Países Bajos.

Este lenguaje toma características de lenguajes predecesores; por ejemplo, tiene tres formas de imprimir un valor de una variable: desde el entorno interactivo escribiendo su nombre (como en Basic), usando la función `print`, con concatenación de elementos (al estilo del `write` de Pascal) o bien con patrones de formato (al estilo del `printf` de C).

Python se desarrolla bajo una licencia de *open source* o código abierto aprobado por la OSI, por lo que se puede usar y ser distribuido libremente, incluso con fines comerciales.

A continuación, se resumen algunas de las características de Python:

1. Es un lenguaje interpretado, no compilado, usa tipado dinámico, fuertemente tipado.
2. Es multiplataforma, lo cual es ventajoso para hacer ejecutable su código fuente entre varios sistemas operativos.

3. Es un lenguaje de programación multiparadigma, el cual soporta varios paradigmas de programación como orientación a objetos, estructurada, programación imperativa y, en menor medida, programación funcional.
4. En Python, el formato del código (p. ej., la indentación) es estructural.

### 3.4 Formato JSON

“JSON (JavaScript Object Notation – Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo” (Introducción a JSON, s.f.).

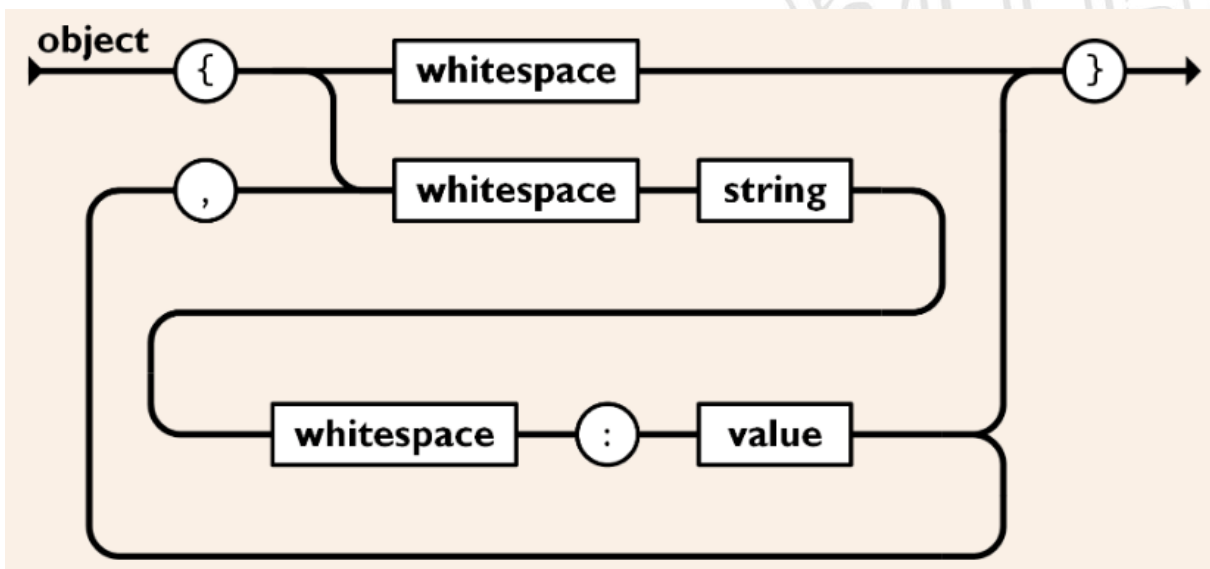
JSON es un formato de texto independiente del lenguaje pero que utiliza convenciones que son ampliamente conocidas por los programadores de lenguajes como C, C++, C#, Java, JavaScript, y muchos otros. Estas características hacen que JSON sea un formato ideal para el intercambio de datos.

Está constituido por dos estructuras:

1. Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, diccionario, tabla hash o un arreglo asociativo.
2. Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

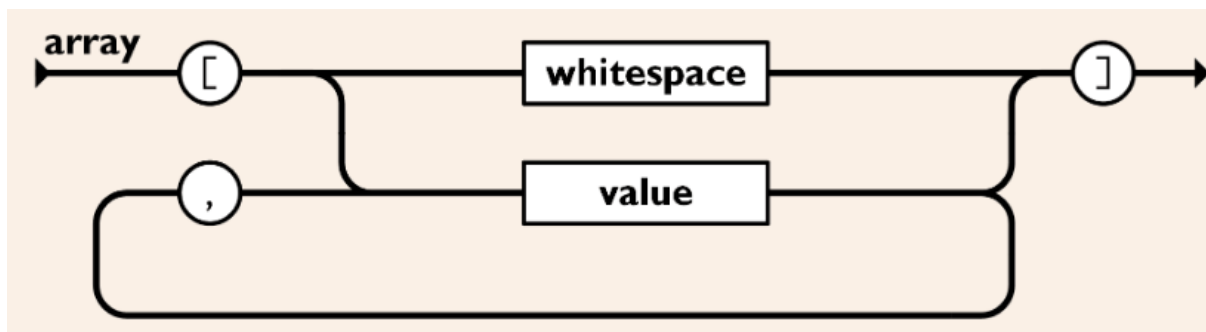
Un *objeto* es un conjunto desordenado de pares nombre/valor. Un *objeto* comienza con { (llave de apertura) y termine con } (llave de cierre). Cada nombre es seguido por : (dos puntos) y los pares nombre/valor están separados por , (coma).

En la figura 3.3 se observa el diagrama de la estructura de un objeto.



**Figura 3.3.** Estructura de un objeto en un archivo JSON. Imagen tomada de: (Introducción a JSON, s.f.)

Un arreglo es una colección de valores. Un arreglo comienza con [ (corchete izquierdo) y termina con ] (corchete derecho). Los valores se separan por , (coma). En la figura 3.4 se muestra el diagrama de la estructura de un arreglo



**Figura 3.4.** Estructura de un arreglo en un archivo JSON. Imagen tomada de: (Introducción a JSON, s.f.)

La teoría abordada en este informe sobre archivos JSON es una visión general de su estructura y de algunas características de la información que puede ser manejada mediante estos archivos, el formato de salida desde la API de Amazon Textract se puede ver en la sección 4.2.3

## 4. Metodología

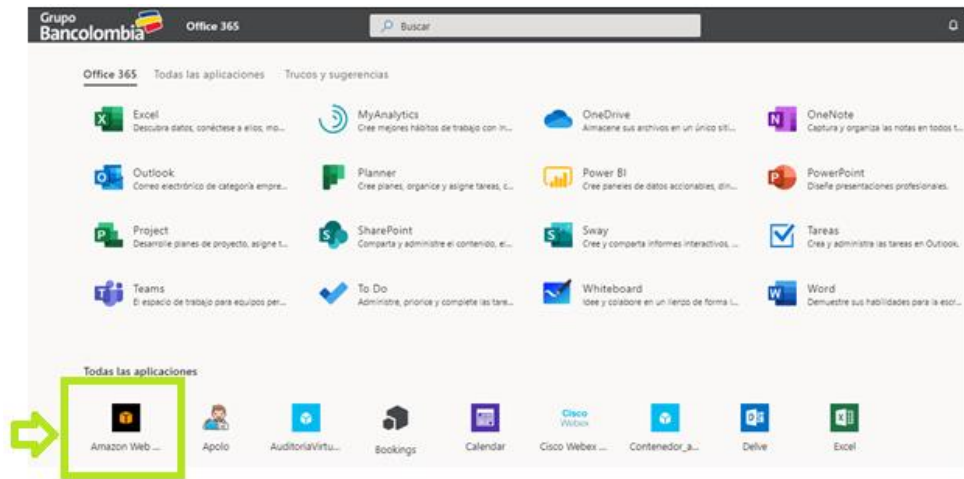
### 4.1 Acceso a los servicios de Amazon

Amazon ofrece un servicio llamado Identity and Access Management (IAM), lo cual permite tener un administrador de los servicios y recursos de AWS a nivel de empresa o de usuario. El Grupo Bancolombia cuenta con el administrador que otorga estos permisos, en el momento que alguno de sus colaboradores, por motivos exploratorios o a nivel de desarrollo requiere usar los servicios provistos por la plataforma. El usuario debe realizar una solicitud por medio de la red interna del banco para ser agregado y poder tener los permisos para acceder a cualquiera de las herramientas disponibles. La solicitud tarda alrededor de 3 días en hacerse efectiva y quedar inscrito como usuario, bajo el registro como empresa del Grupo Bancolombia. Una vez se encuentre autorizado ya puede hacer uso de los servicios disponibles en la plataforma de AWS; Amazon ofrece alrededor de 175 servicios y mediante el registro, una empresa o usuario tiene disponible todas las herramientas, es de anotar que solo se le cobra por las que sean utilizados.

### 4.2 Acceso a Amazon Textract

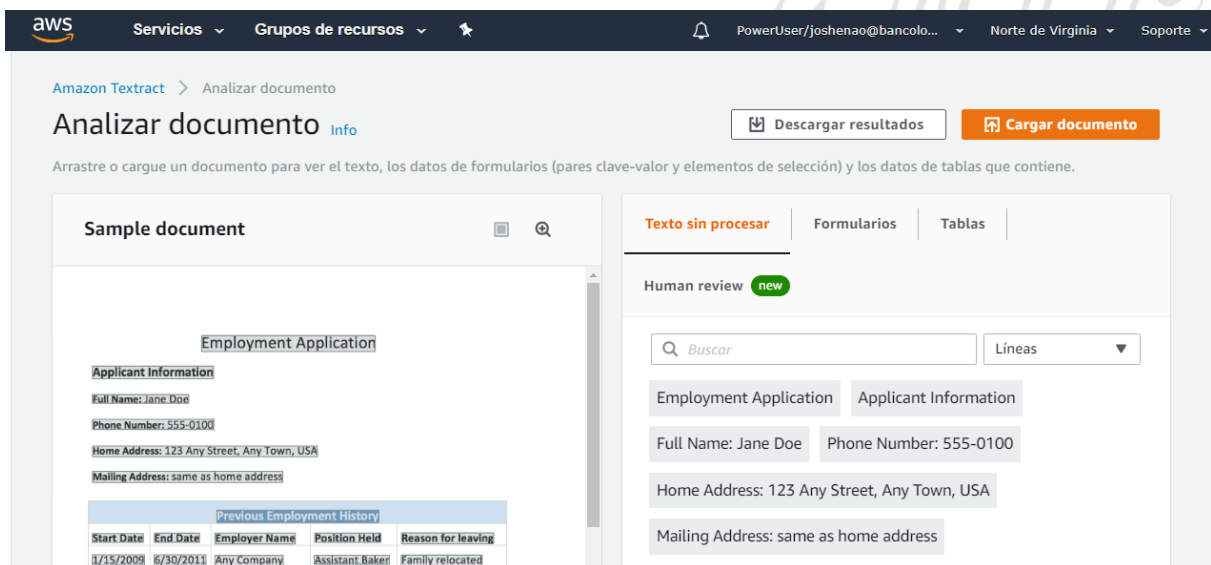
Una vez se tiene el acceso a la plataforma AWS, se puede hacer uso de los servicios en su versión de prueba. Para utilizarlos como servicios API se requiere seguir unos pasos los cuales se describen más adelante.

La forma de ingresar a la plataforma de AWS en el banco es por medio de Office 365, se debe ingresar en el navegador y autenticarse con el usuario y contraseña personal como colaborador del Grupo Bancolombia, una vez allí se observa las herramientas disponibles de trabajo, entre esas opciones se encuentra Amazon Web Services (AWS) tal como se muestra en la figura 4.1



**Figura 4.1.** Acceso a Amazon desde la red interna del banco. Captura de pantalla.

Al seleccionar en el logotipo de AWS se direcciona a una interfaz la cual se llama “Consola de administración de AWS” allí se busca la herramienta utilizada para este proyecto, Amazon Textract, la cual permite hacer pruebas con este servicio como se muestra en la figura 4.2, en la capa gratuita se puede cargar un documento y mirar la respuesta que el servicio ofrece.

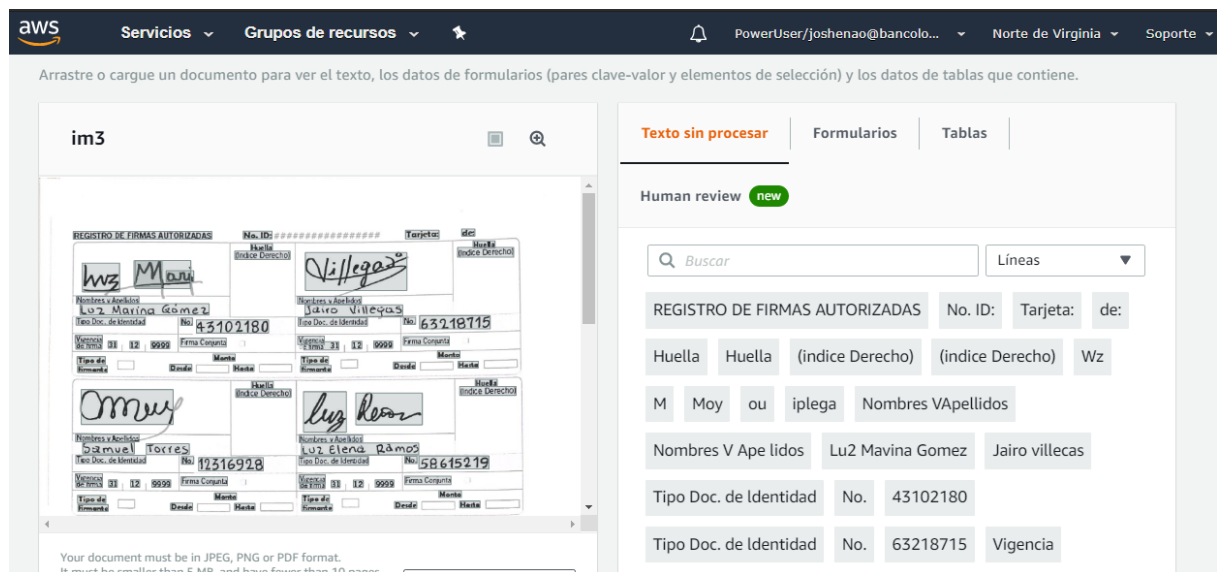


**Figura 4.2.** Capa gratuita de Amazon Textract. Imagen tomada de: (Amazon Web Services I. o., 2020)

Como se observa en el menú de la figura 4.2, este servicio tiene la opción de analizar diferentes documentos, como textos, tablas y formularios. Permite hacer un reconocimiento inteligente de la imagen, para identificar la clave-valor en caso de ser una tabla o un formulario y entrega el texto líquido en cualquiera de los dos casos, adicionalmente entrega el texto de cualquier imagen o documento de entrada.

#### 4.2.1 Ensayos del funcionamiento del OCR de Amazon Textract

En la figura 4.3 se observa una imagen escaneada cargada a la capa gratuita de Amazon Textract con la que se hizo la prueba a una de las tarjetas de firmas utilizadas para el desarrollo de este proyecto.



**Figura 4.3.** Prueba a una de las tarjetas de firmas en la capa gratuita de Amazon Textract. Captura de pantalla. (Datos personales ficticios)

La tarjeta de firmas es un documento en el cual sólo la firma se plasma de forma manuscrita, el resto de información se llena con letra de máquina. Sin embargo, en este ejemplo se llenó la tarjeta con datos ficticios y adicionalmente, la cédula y el nombre de la persona se hizo de forma manual, con el fin de exigir al máximo el reconocimiento de caracteres de Amazon Textract. También es importante anotar, el documento posee unas simetrías las cuales fueron fundamentales para el desarrollo del proyecto. En la figura 4.4 se observa una tarjeta de firmas con algunas de sus características, en la sección 4.4.3 se detallan otros aspectos de su diseño.

		Huella (Índice Derecho)
Nombres y Apellidos Luz Marina Gómez		
Tipo Doc. de Identidad No. 43102180		
Vigencia de Firma 31   12   9999	Firma Copiada	
Tipo de Fomento	Desde	Hasta

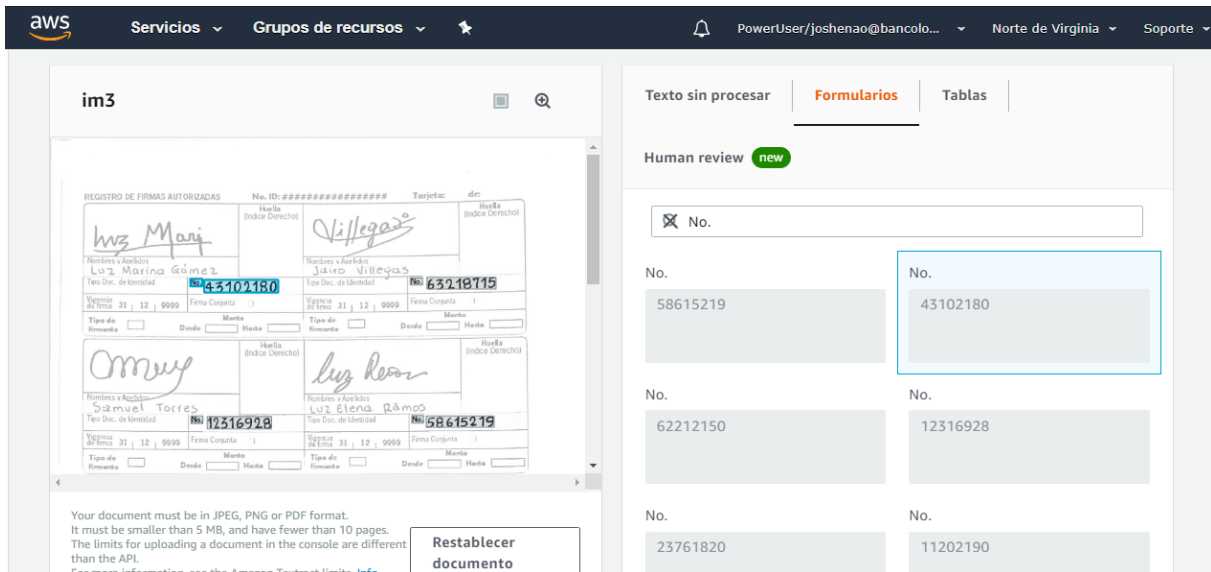
REGISTRO DE FIRMAS/AUTORIDADES		No. ID: *****	Tarjeta: de:
		Huella (Índice Derecho)	
Nombres y Apellidos Luz Marina Gómez		Nombres y Apellidos Javier Villegas	
Tipo Doc. de Identidad No. 43102180		Tipo Doc. de Identidad No. 63218715	
Vigencia de Firma 31   12   9999	Firma Copiada	Vigencia de Firma 31   12   9999	Firma Copiada
Tipo de Fomento	Desde	Hasta	Tipo de Fomento
		Huella (Índice Derecho)	
Nombres y Apellidos Soledad Torres		Nombres y Apellidos Luz Elena Ramos	
Tipo Doc. de Identidad No. 12316928		Tipo Doc. de Identidad No. 58615219	
Vigencia de Firma 31   12   9999	Firma Copiada	Vigencia de Firma 31   12   9999	Firma Copiada
Tipo de Fomento	Desde	Hasta	Tipo de Fomento
		Huella (Índice Derecho)	
Nombres y Apellidos Elena Rodríguez		Nombres y Apellidos Jaime Carmona	
Tipo Doc. de Identidad No. 62212150		Tipo Doc. de Identidad No. 23761820	
Vigencia de Firma 31   12   9999	Firma Copiada	Vigencia de Firma 31   12   9999	Firma Copiada
Tipo de Fomento	Desde	Hasta	Tipo de Fomento
		Huella (Índice Derecho)	
Nombres y Apellidos Ruth Álvarez		Nombres y Apellidos Carmenza Duque	
Tipo Doc. de Identidad No. 11202190		Tipo Doc. de Identidad No. 45219018	
Vigencia de Firma 31   12   9999	Firma Copiada	Vigencia de Firma 31   12   9999	Firma Copiada
Tipo de Fomento	Desde	Hasta	Tipo de Fomento

Scanned with CamScanner

Figura 4.4. Imagen de la tarjeta de firmas. Imagen propia. (Datos personales ficticios)

En la parte derecha de la figura 4.3 se observa el texto procesado hallado en la imagen, para este caso en particular, lo más importante es obtener un resultado confiable en la identificación de los números de documentos, ya que estos son la base para hacer la indexación del recuadro de las firmas en el momento en el cual se segmentan.

Una de las particularidades que ofrece Amazon Textract es la capacidad para hacer una identificación de la imagen y poder determinar si se trata de una tabla, un formulario o un texto plano; la tarjeta de firmas es un formulario. Se observa en la figura 4.5 algunos resultados del par clave-valor devueltos por la plataforma cuando se trata de este formato.

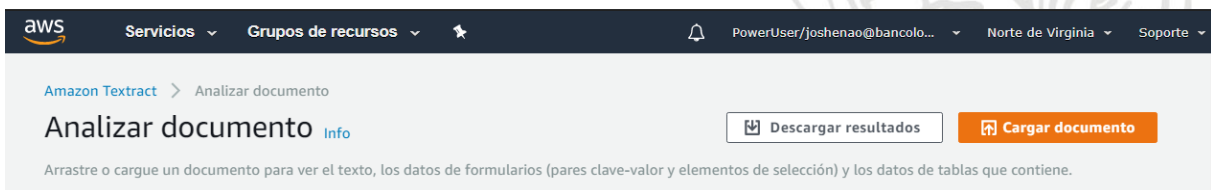


**Figura 4.5.** Identificación clave-valor por el OCR de Amazon. Captura de pantalla. (Datos personales ficticios)

Cuando se selecciona la opción formularios, se encuentra una barra de búsqueda en la cual se puede ingresar la clave que se requiere conocer, en esta prueba, se ingresó "No.", la cual, en la tarjeta de firmas, representa el número de identificación personal correspondiente a la firma que se encuentra en el mismo recuadro. A la derecha de la figura 4.5 se observa los demás pares clave-valor detectados, que corresponden a los otros números de identificación personal de las demás firmas presentes el documento.

Con esta prueba y otras más realizadas con diversas imágenes de la tarjeta de firmas, se pudo constatar la efectividad en el reconocimiento de números por parte del OCR de Amazon Textract. En cuanto a los caracteres alfabéticos se observó un gran margen de error en la identificación de las letras, en algunos se obtiene caracteres extraños y en otros una mezcla de mayúsculas y minúsculas que no están presentes en la imagen original. El hecho importante es que para este caso se necesitaba alta confiabilidad en la identificación de números, ya que son los datos importantes que se necesitan extraer para el proceso.

En la figura 4.6 se muestra una opción adicional cuando se carga un documento "Descargar resultados", al seleccionar esta opción la plataforma entrega una carpeta comprimida, la cual lleva como nombre el mismo de la imagen cargada, y dentro de esta hay tres archivos, como se muestra en la figura 4.7.



**Figura 4.6.** Opción de descarga de resultados de pruebas en Amazon. Captura de pantalla

Nombre	Tipo	Tamaño comprimido	Protegido ...	Tamaño	Relación	Fecha de modificación
apiResponse	Archivo JSON	525 KB	No	525 KB	0%	7/07/2020 1:12 p. m.
keyValues	Archivo de valores separa...	2 KB	No	2 KB	0%	7/07/2020 1:12 p. m.
rawText	Documento de texto	2 KB	No	2 KB	0%	7/07/2020 1:12 p. m.

**Figura 4.7.** Archivos entregados tras el análisis de la imagen. Captura de pantalla.

Como se puede observar son tres archivos de extensiones diferentes, primero, es un archivo JSON, segundo, un archivo de Microsoft Excel con una tabla en la que en cada celda se encuentra la clave-valor correspondientes hallados tras el análisis de la imagen, tercero, es un archivo de texto en el cual se encuentra todo el texto que Amazon Textract encontró tras el procesamiento.

#### 4.2.2 Análisis del archivo JSON entregado

La información más completa y útil para el desarrollo del proyecto la entrega el archivo JSON, son datos fundamentales para el análisis del documento y poder ubicar las palabras y números de identificación personal dentro del documento.

Aspectos importantes devueltos en el archivo JSON:

1. Entrega cada palabra hallada en el documento de forma separada, también sí identifica que es una frase completa la entrega en otra sección de forma conjunta.
2. Devuelve las coordenadas de la ubicación de la palabra dentro de la imagen, un aspecto importante de este servicio es que las coordenadas se entregan de forma proporcional al tamaño de la imagen, por lo tanto, con un programa adecuado se pudo aprovechar este aspecto para recibir imágenes de distintas resoluciones y con diversas ubicaciones del documento de firmas dentro de la imagen.
3. Otra de las particularidades que brinda este servicio de Amazon, es la información que suministra de la correspondencia entre el par clave-valor cuando se trata de un formulario o una tabla, cada palabra en el JSON tiene un identificador, así cuando se requiere analizar el archivo para hacer una correlación entre la palabra en el documento y el valor que le corresponde, se debe buscar la relación entre ambos identificadores, para este proceso es necesario tener un programa que sea capaz de buscar estas relaciones e interconectarlas.

#### 4.2.3 Análisis de un bloque en el archivo JSON

A continuación, en la figura 4.8 se muestra un segmento de código con los parámetros que se deben configurar, el cual funciona tomando un elemento del repositorio de Amazon S3 y se procesa por Textract. En este proceso se utilizan dos servicios de la nube de AWS, uno es el almacenamiento de objetos (Amazon S3) y otro el OCR de Textract, se deben configurar los parámetros de acuerdo con los datos e información que se dispone, como nombre del *bucket* (contenedor de objetos de S3), nombre del archivo a procesar, lista de los tipos de análisis a realizar si es una tabla, un documento o un formulario. También hay otros parámetros opcionales que



se utilizan en caso de ser necesarios, uno de ellos es el parámetro *HumanLoopConfig* el cual se define en la documentación de la página oficial de Amazon Textract de la siguiente manera: "Configura el flujo de trabajo de revisión humana al que se enviará el documento si se cumple una de las condiciones. También puede establecer ciertos atributos de la imagen antes de la revisión." (Amazon, HumanLoopConfig, 2020). Para analizar el texto en un documento se utiliza la operación *AnalyzeDocument*, pasa un archivo de documento como entrada, el cual puede ser una matriz de bytes de imagen (bytes de imagen codificados en base64) o como un objeto de Amazon S3.

```
{
  "Document": {
    "Bytes": blob,
    "S3Object": {
      "Bucket": "string",
      "Name": "string",
      "Version": "string"
    }
  },
  "FeatureTypes": [ "string" ],
  "HumanLoopConfig": {
    "DataAttributes": {
      "ContentClassifiers": [ "string" ]
    },
    "FlowDefinitionArn": "string",
    "HumanLoopName": "string"
  }
}
```

**Figura 4.8.** Parámetros de entrada. Captura de pantalla tomada de: (Amazon Web Services I. a., 2020, pág. 106)

La respuesta obtenida es un archivo en formato JSON, con un bloque el cual suministra la información requerida (ver figura 4.9). Un bloque representa los elementos que se reconocen en un documento analizado dentro de un grupo de pixeles cercanos entre sí, el tipo de análisis depende de la operación solicitada, en esta respuesta se hizo la solicitud con la función *AnalyzeDocument*.

```

{
  "AnalyzeDocumentModelVersion": "string",
  "Blocks": [
    {
      "BlockType": "string",
      "ColumnIndex": number,
      "ColumnSpan": number,
      "Confidence": number,
      "EntityTypes": [ "string" ],
      "Geometry": {
        "BoundingBox": {
          "Height": number,
          "Left": number,
          "Top": number,
          "Width": number
        },
        "Polygon": [
          {
            "X": number,
            "Y": number
          }
        ]
      },
      "Id": "string",
      "Page": number,
      "Relationships": [
        {
          "Ids": [ "string" ],
          "Type": "string"
        }
      ],
      "RowIndex": number,
      "RowSpan": number,
      "SelectionStatus": "string",
      "Text": "string"
    }
  ],
  "DocumentMetadata": {
    "Pages": number
  }
}

```

**Figura 4.9.** Archivo JSON con la información de salida. Captura de pantalla tomada de: (Amazon Web Services I. a., 2020, pág. 107)

Después de haber analizado los datos entregados por Amazon Textract tras el procesamiento de una imagen, se puede detallar la información suministrada por cada uno de los bloques entregados dentro de un archivo JSON.

En la figura 4.10 se observa un ejemplo de los bloques con dicha información, se dan los detalles de la frase completa "REGISTRO DE FIRMAS AUTORIZADAS", correspondiente al título que lleva el documento procesado para el desarrollo de este proyecto, este título fue el punto de partida para la ubicación dentro de la

imagen; por cada una de las palabras o frases presentes en la imagen la plataforma entrega un bloque similar en el archivo JSON.

```
342 {
343   "BlockType": "LINE",
344   "Confidence": 99.88175964355469,
345   "Text": "REGISTRO DE FIRMAS AUTORIZADAS",
346   "Geometry": {
347     "BoundingBox": {
348       "Width": 0.25594469904899597,
349       "Height": 0.009842606261372566,
350       "Left": 0.06322410702705383,
351       "Top": 0.06845512241125107
352     },
353     "Polygon": [
354       {
355         "X": 0.06322410702705383,
356         "Y": 0.06845512241125107
357       },
358       {
359         "X": 0.3191688060760498,
360         "Y": 0.06845512241125107
361       },
362       {
363         "X": 0.3191688060760498,
364         "Y": 0.07829772680997849
365       },
366       {
367         "X": 0.06322410702705383,
368         "Y": 0.07829772680997849
369       }
370     ]
371   },
372   "Id": "a6e52254-4f0d-44af-8fb3-6b40ed1be59b",
373   "Relationships": [
374     {
375       "Type": "CHILD",
376       "Ids": [
377         "8dc9e578-3bd3-4a85-a2a7-7c617cdf3cb6",
378         "946d3bd1-950e-4aea-a1f3-94b3a2aad5d0",
379         "11ed8688-f65c-47d7-87de-ee9d563d5b47",
380         "8d7ec27e-1056-4169-8cf2-f6365a610dec"
381       ]
382     }
383   ]
384 },
```

Figura 4.10. Bloque de información en el archivo JSON. Captura de pantalla.

Los ítems entregados en el bloque son:

1. **"BlockType": "LINE"**: Representa un bloque de información de una línea de texto completa, en este caso, la frase corresponde al título del documento antes mencionado.
2. **"Confidence": 99.88175964355469**: es la probabilidad de que el resultado extraído de esta frase es correcto.
3. **"Text": "REGISTRO DE FIRMAS AUTORIZADAS"**: es el texto detectado en la imagen, en este caso corresponde al título de la tarjeta de firmas.
4. **"Geometry"**: en este ítem entrega las dimensiones del recuadro que bordea la frase completa, como se observa en la figura 4.11, el ancho("Width"),

alto("Height"), ubicación de la coordenada izquierda("Left") y de la coordenada derecha("Top").



Figura 4.11. Recuadro etiqueta de las palabras. Imagen propia.

5. **"Polygon"**: en este ítem, la información entregada son cada una de las coordenadas de las esquinas del recuadro de la figura 4.11, donde la primera coordenada es la parte superior izquierda y luego continua a la derecha dando la vuelta del recuadro en sentido de las manecillas del reloj.
6. **"Id"**: **"a6e52254-4f0d-44af-8fb3-6b40ed1be59b"**: esta es la identificación dada a esta frase en particular, esta información es necesaria cuando se requiere hacer una asociación entre un valor con otro, como en el caso del par clave-valor.
7. **"Relationships"**: acá se encuentran los Id de los bloques donde se encuentra la información de cada una de las palabras por separado que conforman la frase completa de este bloque.

### 4.3 Configuración de uso de la API de Amazon Textract

Para acceder a los servicios de Amazon textract como API se deben ejecutar unos procedimientos los cuales hacen posible la interacción con Amazon de forma remota.

A continuación, se listan los pasos realizados para utilizar los servicios de Amazon Textract mediante comunicación vía API:

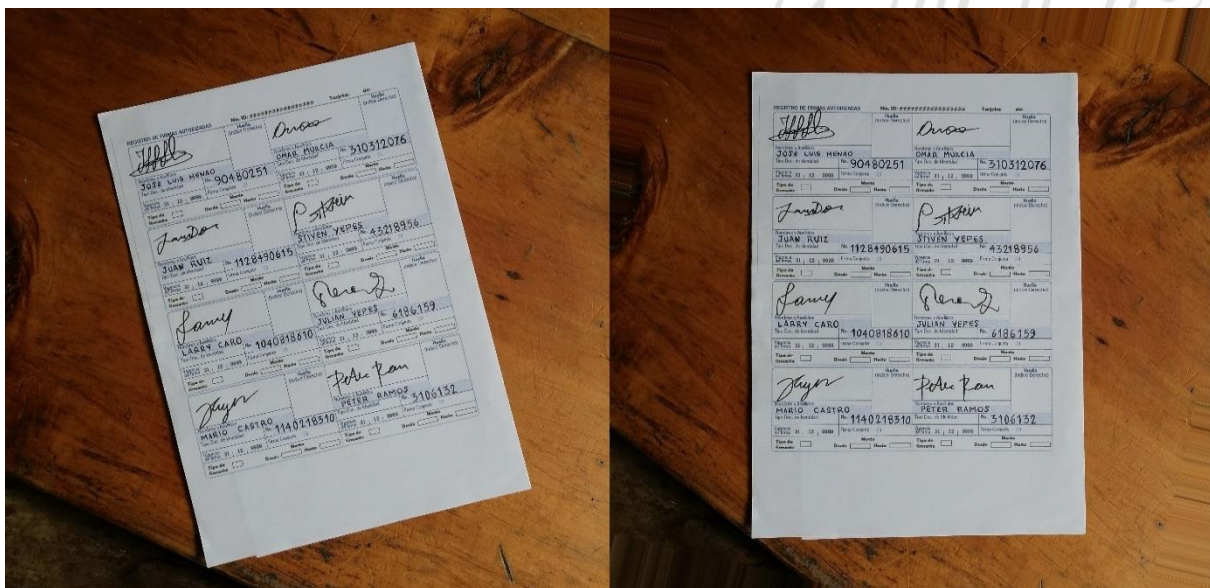
1. Instalar Node.js. "Node.js es un entorno JavaScript que nos permite ejecutar en el servidor, de manera asíncrona, con una arquitectura orientada a eventos y basado en el motor V8 de Google" (Seidor, 2020)
2. Instalar aws-azure-login lo cual según la documentación consultada define: "Si su organización usa Azure Active Directory para proporcionar inicio de sesión SSO en la consola de AWS, entonces no hay una manera fácil de iniciar sesión en la línea de comandos o usar la CLI de AWS. Esta herramienta arregla eso. Le permite usar el inicio de sesión normal de Azure AD (incluido MFA) desde una línea de comando para crear una sesión de AWS federada y coloca las credenciales temporales en el lugar adecuado para la AWS CLI y los SDK." (npm, 2020)
3. En las variables de entorno de Windows, más concretamente en "path", se coloca la ruta donde se encuentra el ejecutable de node.js
4. Seguir la serie de pasos descritos en la sección "Getting Your Tenant ID and App ID URI" en la página oficial de **npm** (Node Package Manager) disponible en: <https://www.npmjs.com/package/aws-azure-login>

5. Luego en la consola de comandos de Windows (cmd) se ingresa el siguiente comando: "aws-azure-login –configure" para realizar la configuración de la cuenta con los anteriores datos obtenidos.
6. Por último, para autenticarse se ingresa en el cmd de Windows el siguiente comando: "aws-azure-login", como respuesta te pedirá tu usuario autorizado para usar los servicios de Amazon, la contraseña, y el tiempo de uso de la herramienta, este servicio cuenta con un tiempo máximo de 2 horas antes de requerir volver a autenticarse.

## 4.4 Trabajando desde Python con la API de Amazon Textract

### 4.4.1 Enderezado de las imágenes

En el desarrollo de este proyecto se obtuvo buenos resultados en el procesamiento siempre y cuando, la tarjeta de firmas dentro de la imagen estuviera derecha. Observe en la figura 4.12 la comparación de dos formularios, a la izquierda muestra una tarjeta de firmas con un grado de inclinación de aproximadamente  $15^\circ$  con respecto a la horizontal de la imagen y a la derecha se encuentra la misma tarjeta luego de ser procesada por un programa adaptado para enderezarla.



**Figura 4.12.** Enderezado de los formularios en la imagen. Imagen propia. (Datos personales ficticios)

Para mejorar la funcionalidad de extracción y segmentación de las firmas del proyecto, se procedió a buscar la manera de hacer que la tarjeta de firmas estuviera lo más paralela posible a la horizontal de la imagen. Buscando documentación, se encontró un programa que cumplía esta tarea, fue necesario realizarle algunas modificaciones ya que cumplía la función si el ángulo de desviación era muy pequeño, además, devolvía errores cuando la imagen ingresada estaba derecha y no tenía desviación.

A continuación, en la figura 4.13 se observa el código final utilizado para adecuar la imagen, el código base fue tomado de (abulafia, 2018)

```

9 import numpy as np
10 import cv2
11
12 def estan_cercanos(a1, a2, error):
13     cases = np.unwrap([a2-error, a1, a2 + error])
14     return cases[0] <= cases[1] <= cases[2]
15
16 def enderezar(entrada):
17     # Leer la imagen
18     imagen = cv2.imread(entrada)
19     # Convertirla a gris y detectar bordes
20     gray = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
21     binaria = cv2.Canny(gray, 50, 150, apertureSize = 3)
22
23     # Usar la transformada de Hough para encontrar líneas
24     # en la imagen binarizada, con una resolución de medio
25     # grado (pi/720) y quedándose sólo con las líneas que
26     # alcancen puntuación de 1000 o más (que serán las
27     # más largas)
28     lineas = cv2.HoughLines(binaria, 1, np.pi/720, 1000)
29
30     # Recopillemos qué ángulos ha encontrado la transformada
31     # de hough para cada una de las líneas halladas
32     try:
33         angulos = []
34         for linea in lineas:
35             rho, theta = linea[0]
36             if rho < 0:
37                 theta = -theta
38
39             # Quedarse solo con las rayas próximas a la horizontal
40             # (con un error de +40 grados)
41             if not estan_cercanos(theta, np.pi/2, np.deg2rad(40)):
42                 continue;
43
44             angulos.append(theta)
45
46     # Ahora contemos cuántas veces aparece cada ángulo
47     from collections import Counter
48     veces = Counter(angulos)
49
50     # Quedémonos con los tres casos más frecuentes
51     frecuentes = veces.most_common(3)
52
53     # Y calculemos el promedio de esos tres casos
54     suma = sum(angulo*repeticion for angulo, repeticion in frecuentes)
55     repeticiones = sum(repeticion for angulo, repeticion in frecuentes)
56     angulo = suma/repeticiones
57
58     angulo = np.rad2deg(angulo - np.pi/2)
59     print("[INFO] angulo: {:.5f}".format(angulo))
60
61     # Ahora enderezemos la imagen, girando el ángulo detectado
62     (h, w) = imagen.shape[:2]
63     centro = (w // 2, h // 2)
64     M = cv2.getRotationMatrix2D(centro, angulo, 1.0)
65
66     girada = cv2.warpAffine(imagen, M, (w, h),
67                             flags=cv2.INTER_CUBIC, borderMode=cv2.BORDER_REPLICATE)
68
69     # Y volcamos a disco el resultado
70     cv2.imwrite("salida.jpg", girada)
71     print("Correcto")
72     return girada
73
74 except:
75     cv2.imwrite("error_salida.jpg", imagen)
76     print("Error ")
77
78 if __name__ == "__main__":
79     entrada = './inputs/im23.jpg'
80     enderezar(entrada)

```

Figura 4.13. Código utilizado para enderezar el formulario dentro de la imagen. Captura de pantalla

Este código funciona para documentos en los cuales se encuentre una tabla o en este caso un formulario, es decir, un documento en el cual se encuentren líneas rectas, ya que básicamente se encarga de buscar estas líneas y calcular el ángulo que forma con la horizontal de la imagen, para luego alinearlos. Se cambiaron unos parámetros de tal forma que tuviera un mayor rango de grados y no solo con desviaciones muy próximas a la horizontal, se configuró hasta 40° ya que si se aumentaba a más de 45° corría el riesgo de girar el documento 90° en la imagen. Otro inconveniente solucionado, fue cuando la imagen entraba al programa y no estaba desviada, o simplemente el formulario estaba bien ubicado, en este caso se obtenía un error en varias de las pruebas, por lo tanto, se le agrego un manejo de excepciones de Python, en tal caso, si al pasar por este programa ocurre algún error, lo que hace es devolver la imagen tal y como ingreso al programa.

#### 4.4.2 Función para el llamado de la API

A continuación, en la figura 4.14 se presenta el fragmento de código de la función con la cual se hace el llamado a la API de Amazon Textract:

```

7 import cv2
8 import boto3
9 from time import time #importamos la función time para capturar tiempos
10
11 def getResponse(file_name):
12     with open(file_name, 'rb') as file:
13         img_test = file.read()
14         bytes_test = bytearray(img_test)
15         print('Image Loaded', file_name)
16     # process using image bytes
17     client = boto3.client('textract', region_name='us-east-1')
18     #response = client.analyze_document(Document={'Bytes': bytes_test}, FeatureTypes=['FORMS'])
19     response = client.detect_document_text(Document={'Bytes': bytes_test})# , FeatureTypes=['FORMS'])
20
21     return response

```

Figura 4.14. Función para llamado a Amazon Textract. Captura de pantalla.

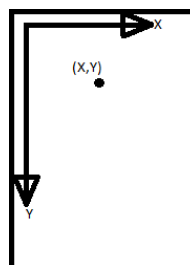
Se detallan las líneas principales de la función:

1. La librería **boto3**, permite a los desarrolladores de Python crear, configurar y administrar servicios de AWS.
2. La línea 17, `client = boto3.client('textract', 23egión_name='us-east-1')` son los parámetros requeridos para hacer el llamado de Textract.
3. Línea 19, `response = client.detect_document_text(Document={'Bytes': bytes_test})` en esta línea se hace un llamado a una función específica, en este caso, llama una función para detectar texto en un documento y le envía la imagen como un arreglo de bytes. También es posible tomar una imagen almacenada en el repositorio de Amazon S3, para efectos de prueba y desarrollo de este proyecto se utilizó imágenes almacenadas en el mismo PC de donde se hacía el llamado.
4. El parámetro **response** en la línea 19, es el archivo JSON con la información de respuesta entregada por Textract y la cual se analizó en el numeral 4.2.3

Se puede observar la línea 18 comentada, esta se utilizaba para el análisis de la imagen, la cual enviaba a la plataforma la orden de identificar qué tipo de documento estaba presente en la imagen, pero leyendo diferente documentación y analizando el tipo de análisis que se requería hacer, se constató que no era necesario utilizarla. Hacer el análisis del tipo de documento en la plataforma, por ejemplo, si es una tabla o un formulario, tiene un costo en dinero más alto que sólo el proceso de identificar el texto. Más adelante en la sección 5.5 se observarán los costos y las conclusiones de la utilización de estos servicios.

#### 4.4.3 Extracción de las coordenadas

Como se mencionó anteriormente, Amazon Textract devuelve las coordenadas de la etiqueta (boundingbox) que bordea la palabra o texto hallado en la imagen. Una de las características de esta respuesta es el valor de las coordenadas entregadas, son valores en proporción a la imagen. Por ejemplo, se supone el punto (x,y) de la siguiente gráfica, el llamado origen de coordenadas de una pantalla en un computador es la esquina superior izquierda.



El punto se encuentra en toda la mitad de la imagen en el eje  $x$  y en un cuarto de la imagen en la parte superior en el eje  $y$ , para este caso la respuesta será  $x = 0.5$  y  $y = 0.25$ , por lo tanto, la resolución o la cantidad de píxeles de la imagen que se envía a

la plataforma por medio de la API no es inconveniente para lograr un programa general que identifique las coordenadas sin importar la resolución de la imagen utilizada.

Seguidamente, como punto de referencia para poder ubicarse en la tarjeta de firmas dentro de la imagen, se buscó un factor característico que fuera confiable y fácilmente reconocible por el OCR de Amazon Textract, por lo tanto, con base en ensayos en la capa gratuita de la plataforma, se observó que el título del formulario "REGISTRO DE FIRMAS AUTORIZADAS" en todas las imágenes procesadas, el OCR daba una confiabilidad de más del 99% en su reconocimiento, siempre y cuando sus letras se encontraran legibles y sin rayones.

Se observa en la figura 4.15 el título del formulario y algunas características importantes que lo llevó a ser elegido como un punto de referencia.

REGISTRO DE FIRMAS AUTORIZADAS				No. ID: #####				Tarjeta: de:							
				Huella (Indice Derecho)											
Nombres v Apellidos Luz Marina Gómez								Nombres v Apellidos Jairo Villegas							
Tipo Doc. de Identidad				No. 43102180				Tipo Doc. de Identidad				No. 63218715			
Vigencia de firma 31   12   9999				Firma Conjunta <input type="checkbox"/>				Vigencia de firma 31   12   9999				Firma Conjunta <input type="checkbox"/>			
Tipo de firmante <input type="checkbox"/>				Monto Desde <input type="text"/> Hasta <input type="text"/>				Tipo de firmante <input type="checkbox"/>				Monto Desde <input type="text"/> Hasta <input type="text"/>			

Figura 4.15. Características y ubicación del título de la tarjeta de firmas. Imagen propia. (Datos personales ficticios)

Características importantes:

1. El título se encuentra en letra mayúscula y su tamaño es mayor que los otros caracteres dentro del documento, por lo tanto, es fácilmente reconocible sin errores.
2. Su posición en el documento lo lleva a que tenga menos probabilidad de ser alterado, por ejemplo, durante el proceso de firmado de algún cliente con su tipo de firma puede rayar el documento y hacer que induzca a un error al procesarlo por el OCR, esta ubicación está más exenta de este problema.

Inicialmente se necesitaba extraer las coordenadas del recuadro o boundingbox del título de la tarjeta de firmas, esta tarea se realizó con la función ilustrada en la figura 4.16

```

58     for block in blocks:
59         if block['BlockType'] == "LINE":
60             if 'REGISTRO DE FIRMAS AUTORIZADAS' in block['Text']:
61                 if 'Polygon' in block['Geometry']:
62                     key_map = block['Geometry']
63                     poligono = key_map['Polygon']
64                     polixy = poligono[3]

```

Figura 4.16. Extracción de las coordenadas del título. Captura de pantalla



Lo que hace el programa es buscar en los bloques de tipo "LINE" (línea de texto) en el archivo JSON la frase completa, precisamente el texto que corresponde al nombre del formulario. Es necesario anotar que en el desarrollo de este segmento no se tuvo en cuenta si ocurrían errores en el reconocimiento de este título, por lo tanto, se deja completamente la confianza al reconocimiento de caracteres por parte del OCR de Textract, adicionalmente es necesario que el documento se encuentre en buenas condiciones y no esté comprometida la legibilidad de cada uno de los caracteres que conforman el título.

La coordenada extraída es la numero 3 que se observa en la figura 4.17, se obtiene desde el archivo JSON mediante la línea 64 del programa mostrado en la figura 4.16. Este punto se tomó como base para luego obtener las medidas y hacer la segmentación de los recuadros de las firmas.



Figura 4.17. Coordenadas del boundingbox o recuadro del título. Imagen propia.

#### 4.4.4 Segmentación del recuadro de la firma

Una vez se tiene un punto de referencia y particularmente poder ubicar el formulario dentro de la imagen, se procede a definir los recuadros que van a demarcar el punto de corte del segmento donde se encuentra la firma.

A continuación, en la figura 4.18 se muestra la definición de la longitud del título como una unidad de medida de la imagen en cuestión, a partir de esta unidad se hizo una representación de las dimensiones requeridas tomando la longitud del título como base, este proceso se realizó con el fin de tener un punto de referencia y un sistema de medida de acuerdo con las dimensiones de la imagen de entrada, la cual está conformada de  $n \times m$  píxeles.

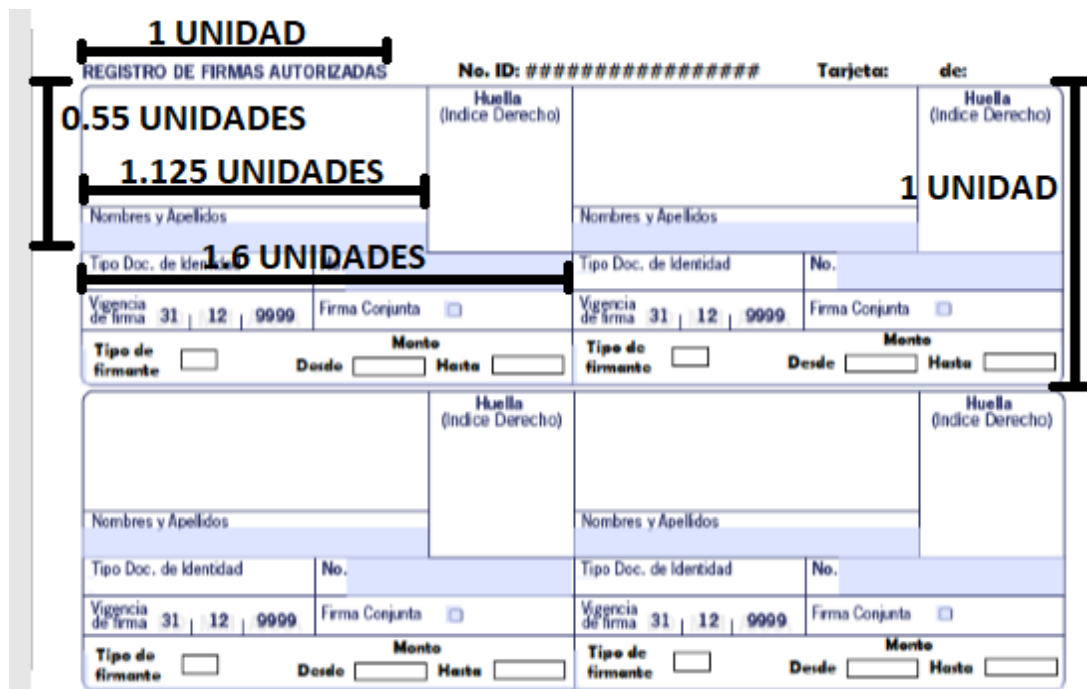


Figura 4.18. Longitud del título como unidad de medida de la imagen. Imagen propia

A continuación, se detallan los pasos del proceso de segmentación del recuadro de la firma:

1. Se toman las dimensiones de la imagen para luego obtener el valor en pixeles de la longitud del título. Se realizó con la siguiente línea de Python: " $m, n, ch = imagen.shape$ " donde  $m$  es la cantidad de filas o largo de la imagen,  $n$  número de columnas o ancho y  $ch$  el número de capas de la imagen, es decir, si la imagen está en RGB (Red, Green, Blue)
2. En la figura 4.19 se muestra la función que realiza el proceso para hallar la longitud del título, básicamente resta los puntos 1 – 0 de la figura 4.17, lo hace solo restando las coordenadas  $x$  de cada punto y luego el resultado final lo multiplica por el ancho de la imagen para obtener el valor en dimensiones de la imagen de entrada.

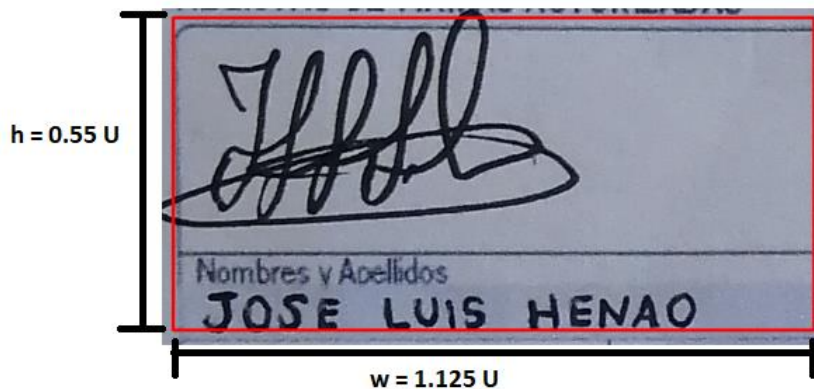
```

23  def PuntoInicial(poligono, n):
24      r1 = poligono[0]
25      r2 = poligono[1]
26      resta = (r2['X'] - r1['X']) * n
27
28      return resta

```

Figura 4.19. Función para obtener longitud del título. Captura de pantalla

3. Con base a esta dimensión inicial, ahora se tiene una unidad de medida en términos de cada imagen de entrada, ahora es posible definir el tamaño de la ventana que demarca la zona de corte del segmento, se observa en color rojo en la figura 4.20.



**Figura 4.20.** Demarcación de la zona de corte del segmento de firma. Imagen propia. (Datos personales ficticios)

Estas dimensiones están dadas con base en la unidad mostrada en la figura 4.18, este segmento corresponde a la primera firma de la parte superior izquierda

4. Una vez se tenía definida las dimensiones de la ventana que selecciona el segmento a recortar, ahora era necesario definir la ubicación de cada una de ellas en el espacio correspondiente a las otras firmas. Se definieron 8 coordenadas que son el máximo número de firmas en la página de la tarjeta de firmas.

A continuación, en la figura 4.21 se muestra el código utilizado para la ubicación de las coordenadas.

```

62     x = int (float (polixy['X'] * n ))
63     y = int (float (polixy['Y'] * m ))  #+ largo_ini * 0.025))
64     h = int (float(largo_ini * 0.55))
65     w = int (float(largo_ini * 1.125))
66     sepa_vert = int (float(largo_ini))
67     sepa_hori = int (float(largo_ini * 1.6))
68     x2 = x + sepa_hori
69     y2 = y + sepa_vert
70     y3 = y2 + sepa_vert
71     y4 = y3 + sepa_vert
72     coorde = [[x,y],[x2,y],[x,y2],[x2,y2],[x,y3],[x2,y3],[x,y4],[x2,y4]]

```

**Figura 4.21.** Código para ubicación de las coordenadas. Captura de pantalla

El código, define inicialmente los valores de  $x, y$  coordenadas del punto 3 mostrado en la figura 4.17, las dimensiones de la ventana de recorte  $h, w$  de la figura 4.20 y calcula los demás puntos coordenados sumando las distancias correspondientes especificadas en la figura 4.18. El resultado de ubicar las demás ventanas de recorte de cada uno de los segmentos de las firmas se observa en la siguiente figura 4.22.

REGISTRO DE FIRMAS AUTORIZADAS				No. ID: #####				Tarjeta: de:							
				Huella (Indice Derecho)								Huella (Indice Derecho)			
Nombres y Apellidos <b>JOSE LUIS HENAO</b>				Nombres y Apellidos <b>OMAR MURCIA</b>				Tipo Doc. de Identidad				No. <b>90480251</b>			
Vigencia de firma 31   12   9999				Firma Conjunta <input type="checkbox"/>				Vigencia de firma 31   12   9999				Firma Conjunta <input type="checkbox"/>			
Tipo de firmante <input type="checkbox"/>				Monto Desde <input type="text"/> Hasta <input type="text"/>				Tipo de firmante <input type="checkbox"/>				Monto Desde <input type="text"/> Hasta <input type="text"/>			
				Huella (Indice Derecho)								Huella (Indice Derecho)			
Nombres y Apellidos <b>JUAN RUIZ</b>				Nombres y Apellidos <b>STEVEN YEPES</b>				Tipo Doc. de Identidad				No. <b>1128490615</b>			
Vigencia de firma 31   12   9999				Firma Conjunta <input type="checkbox"/>				Vigencia de firma 31   12   9999				Firma Conjunta <input type="checkbox"/>			
Tipo de firmante <input type="checkbox"/>				Monto Desde <input type="text"/> Hasta <input type="text"/>				Tipo de firmante <input type="checkbox"/>				Monto Desde <input type="text"/> Hasta <input type="text"/>			
				Huella (Indice Derecho)								Huella (Indice Derecho)			
Nombres y Apellidos <b>LARRY CARO</b>				Nombres y Apellidos <b>JULIAN YEPES</b>				Tipo Doc. de Identidad				No. <b>1040818610</b>			
Vigencia de firma 31   12   9999				Firma Conjunta <input type="checkbox"/>				Vigencia de firma 31   12   9999				Firma Conjunta <input type="checkbox"/>			
Tipo de firmante <input type="checkbox"/>				Monto Desde <input type="text"/> Hasta <input type="text"/>				Tipo de firmante <input type="checkbox"/>				Monto Desde <input type="text"/> Hasta <input type="text"/>			
				Huella (Indice Derecho)								Huella (Indice Derecho)			
Nombres y Apellidos <b>MARIO CASTRO</b>				Nombres y Apellidos <b>PETER RAMOS</b>				Tipo Doc. de Identidad				No. <b>1140218310</b>			
Vigencia de firma 31   12   9999				Firma Conjunta <input type="checkbox"/>				Vigencia de firma 31   12   9999				Firma Conjunta <input type="checkbox"/>			
Tipo de firmante <input type="checkbox"/>				Monto Desde <input type="text"/> Hasta <input type="text"/>				Tipo de firmante <input type="checkbox"/>				Monto Desde <input type="text"/> Hasta <input type="text"/>			

Figura 4.22. Recuadros de recorte de los segmentos de las firmas. Imagen propia. (Datos personales ficticios)

Estas medidas predefinidas en el programa fueron posibles gracias a la simetría que posee la tarjeta de firmas, tal como se observó en la figura 4.4, ya que cada uno de los segmentos donde van los datos y la firma son iguales.

#### 4.4.5 Extracción de los números de identificación personal

Al recortar el recuadro de la firma, este segmento se convierte en una nueva imagen, el nombre que recibe es el número de identificación personal correspondiente de la persona quien firma, este proceso se le llama indexación. Para la identificación de estos números de documento dentro de la imagen también se utilizó el servicio de OCR de Amazon Textract.

Textract identifica estos caracteres y los envía en el archivo JSON de respuesta con su respectiva información, se observa en la figura 4.23 un segmento de ejemplo.

```
7797 "BlockType": "WORD",
7798 "Confidence": 99.16227722167969,
7799 "Text": "43102180",
7800 "Geometry": {
7801   "BoundingBox": {
7802     "Width": 0.13312627375125885,
7803     "Height": 0.016861259937286377,
7804     "Left": 0.290735125541687,
7805     "Top": 0.18114259839057922
7806   },
7807   "Polygon": [
7808     {
7809       "X": 0.290735125541687,
7810       "Y": 0.18114259839057922
7811     },
7812     {
7813       "X": 0.42386141419410706,
7814       "Y": 0.18114259839057922
7815     },
7816     {
7817       "X": 0.42386141419410706,
7818       "Y": 0.1980038583278656
7819     },
7820     {
7821       "X": 0.290735125541687,
7822       "Y": 0.1980038583278656
7823     }
7824   ]
7825 },
7826 "Id": "138498f6-b738-4a66-9c37-1adf56c737b1"
7827 },
```

**Figura 4.23.** Respuesta con la información de número de identificación. Captura de pantalla.

Detalles de la respuesta y captura del dato:

1. El bloque donde se encuentra el número de identificación personal es de tipo "WORD", es decir, no lo identifica como una línea de texto sino como una palabra.
2. Es necesario que este dato se escriba de forma que no tenga espaciamiento entre caracteres para poder procesarlo por el programa desarrollado.
3. En este bloque del archivo JSON también se encuentra información adicional que en este caso no es necesario procesar, como las coordenadas, confianza y el Id del bloque.
4. El código desarrollado para hallar estos números de documento dentro del archivo JSON, se basa en la búsqueda de números entre 6 y 10 dígitos de longitud, que es lo común en las cédulas de ciudadanía colombiana.

En la figura 4.24 se observa la función en Python encargada de realizar esta tarea:

```

30  def getNit(response):
31      blocks=response['Blocks']
32      texto = []
33      nit = []
34      for block in blocks:
35          if block['BlockType'] == "WORD":
36              texto.append(block['Text'])
37      for c in texto:
38          #z+= 1
39          if c.isdigit(): # == True:
40              if len(c)>5 and len(c)<11:
41                  nit.append(c)
42                  #cont.append(z)
43      return nit
44

```

**Figura 4.24.** Código para la búsqueda de números de documento. Captura de pantalla.

En resumen, esta función busca en el archivo JSON los bloques de tipo "WORD", lee los valores de tipo "Text" que están dentro de este bloque y los almacena en un arreglo, luego este arreglo se recorre en búsqueda de los números que cumplan la condición de tener entre 6 y 10 dígitos e igualmente los va almacenando en otro arreglo de tipo lista. Es de anotar que este código realizado no identifica dígitos los cuales estén separados por puntos, comas o espacios.

#### 4.4.6 Indexación de la imagen con el número de identificación personal

Una vez el recuadro de la firma está listo para recortar y los números de documento almacenados, se puede culminar el proceso de extraer e indexar la imagen. Para entender mejor el proceso se observa la figura 4.25 en la cual se encuentra el fragmento de código utilizado para este proceso, seguidamente se detalla su funcionamiento.

```

72  coorde = [[x,y],[x2,y],[x,y2],[x2,y2],[x,y3],[x2,y3],[x,y4],[x2,y4]]
73  nits = getNit(response)
74  for c in range(len(nits)):
75      idx = nits[c]
76      xd,yd = coorde[c]
77      cv2.rectangle(imagen,(xd,yd) , (xd + w, yd + h), (0, 0, 255), 2, cv2.LINE_AA)
78      crop = imagen[yd:yd + h, xd:xd + w]
79      cropped_image_path = "./recortes/%s.png" % idx
80      cv2.imwrite(cropped_image_path, crop)
81      cv2.imwrite("./recortes/completa.png", imagen)

```

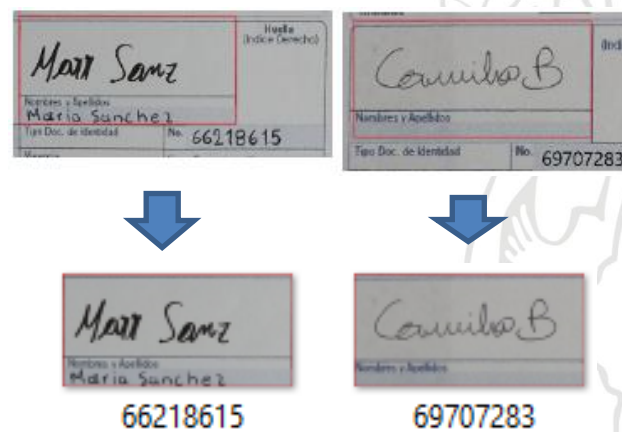
**Figura 4.25.** Segmento de código utilizado para el recorte e indexado. Captura de pantalla.

Para más claridad de esta parte del proceso el cual es de suma importancia en el proyecto, se detalla cada línea de código:

1. **Línea 72:** coordenadas predefinidas, a partir de las cuales se va a empezar a dibujar la ventana de recorte sobre el segmento de cada una de las firmas, con el ancho y alto definidos anteriormente.
2. **Línea 73:** variable con los números de identificación personal almacenados, tal como se mostró en el apartado 4.4.5.
3. **Línea 74:** este ciclo *for* se configuró de modo que el número de recortes de las firmas dependen de la cantidad de números de identificación personal detectados por Textract.

4. **Línea 75:** la variable *idx* va tomando cada número de identificación personal para asignárselo a cada recorte de la imagen correspondiente.
5. **Línea 76:** va leyendo cada una de las coordenadas del arreglo definido en la línea 72, los cuales se convierten en la coordenada de referencia para ubicar la ventana la cual selecciona cada segmento a recortar.
6. **Línea 77:** dibuja en la imagen original las líneas de recorte.
7. **Línea 78:** función para hacer el recorte de los recuadros.
8. **Línea 79:** dirección donde serán almacenadas las imágenes recortadas.
9. **Línea 80:** almacenamiento en disco de los segmentos recortados.
10. **Línea 81:** almacenamiento en disco la imagen completa con las líneas de recorte.

A continuación, en la figura 4.26 se observa un ejemplo de los recuadros con su respectiva indexación luego de pasar por el proceso.



**Figura 4.26.** Recuadros de la firma indexados por el número de identificación personal. Imagen propia. (Datos personales ficticios)

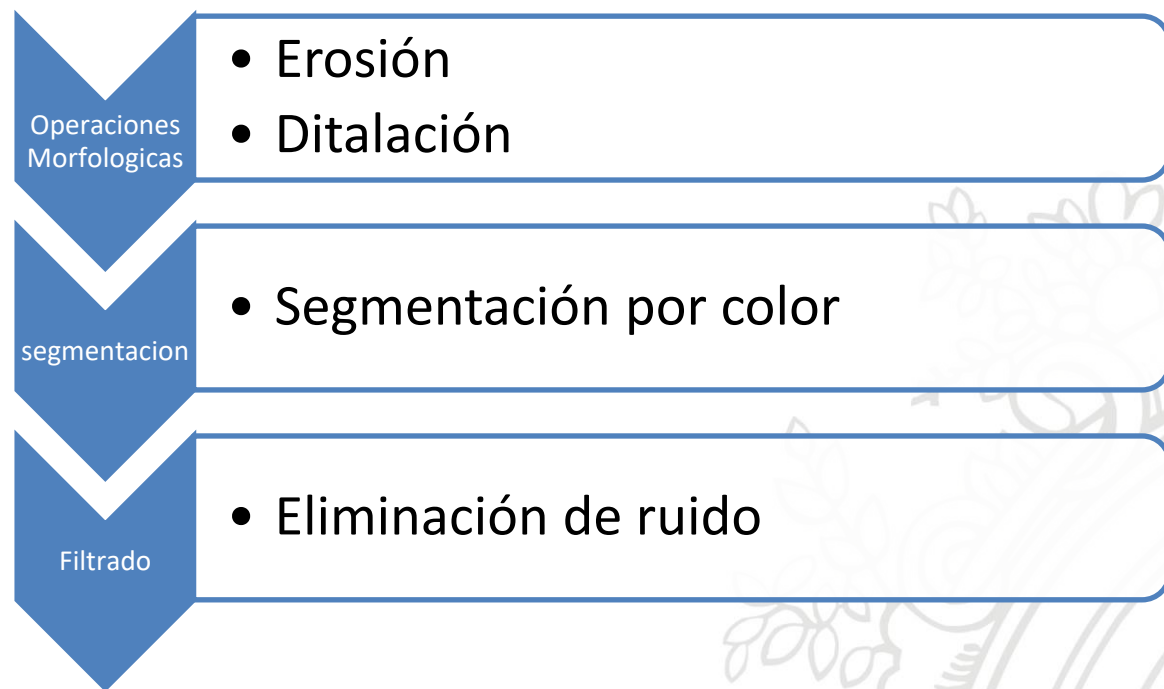
## 5. Resultados y análisis

### 5.1 Resultados procesamiento digital imágenes con OpenCV

En el inicio de este proyecto se realizaron diversas pruebas para hallar una solución a la necesidad planteada. Inicialmente se realizaron algunos ensayos utilizando únicamente OpenCV para hacer una segmentación de la firma, pero el resultado no era aplicable a casos generales y su nivel de confianza era de solo el 62.5% en el mejor de los casos.

A continuación, en el diagrama se muestra de manera general el procedimiento realizado, no se aborda de forma detallada ya que solo se utilizó como prueba en la búsqueda de un proceso funcional.

# Diagrama procesamiento de la imagen mediante OpenCV



A continuación, en las figuras 5.1 y 5.2 se observa el resultado de este proceso



Figura 5.1. Procesamiento de imágenes con OpenCV. Imagen propia. (Datos personales ficticios)





Figura 5.2. Resultado de segmentación por color con OpenCV. Imagen propia. (Datos personales ficticios)

### Análisis del resultado

1. Se asumieron condiciones ideales, partiendo de que la tarjeta de firmas era de color azul y las firmas de un tono negro muy oscuro, lo cual no siempre se cumple en condiciones reales.
2. Con este proceso se requería que las firmas fueran con letra grande en comparación de las demás letras del documento, lo cual es un proceso impredecible ya que las firmas son muy variadas y no tienen ningún tipo de formato o estándar.
3. A nivel de píxeles, el color azul de la tarjeta de firmas tiene muchas tonalidades o incluso puede estar compuesto por otros píxeles de colores diferentes, lo que lleva a tener mucho ruido en el procesamiento.
4. Como resultado se obtuvo una identificación parcial de las firmas y algunas otras ni siquiera fueron detectadas.

Con los resultados obtenidos en esta prueba utilizando procesamiento digital de imágenes con OpenCV, se llegó a la conclusión de que es un método el cual requiere un desarrollo más especializado, y por las características de las firmas puede ser más propenso a tener errores.

### 5.2 Resultados del enderezado de la imagen.

El proceso de enderezado de las imágenes es una de las fases fundamentales para el correcto funcionamiento de todo el proyecto, tal como se observó en el apartado 4.4.1

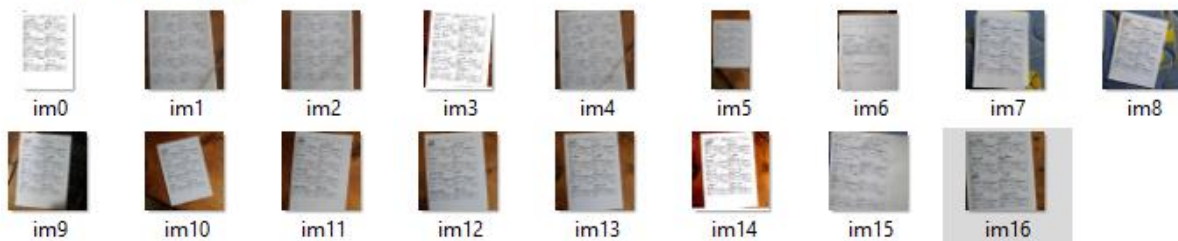
Para esta prueba se tomó una muestra de 17 imágenes con las tarjetas de firmas, en algunas esta tarjeta se encontraba desviada con respecto a la horizontal de la imagen, tal como se observó el ejemplo en la figura 4.12.

A continuación, en la tabla 5.1 se observan los datos de las muestras analizadas, seguidamente en la figura 5.3 se pueden observar las imágenes procesadas, sus entradas y sus respectivas imágenes de salida.

Número de muestras	Dimensión de imagen	Ángulo de desviación	Número de salidas correctas	Número de salidas incorrectas
2	4.3MP;10MP	0°	2	0
2	12MP;16MP	2°	2	0
3	12MP;8MP	3°	3	0
2	12MP;8MP	4°	2	0
3	12MP	5°	3	0
3	12MP	6°	3	0
2	12MP	14°	1	1
<b>Total 17</b>			<b>Total 16</b>	<b>Total 1</b>

Tabla 5.1. Resultado de la prueba de enderezado del formulario.

#### Imágenes de entrada



#### Imágenes de salida

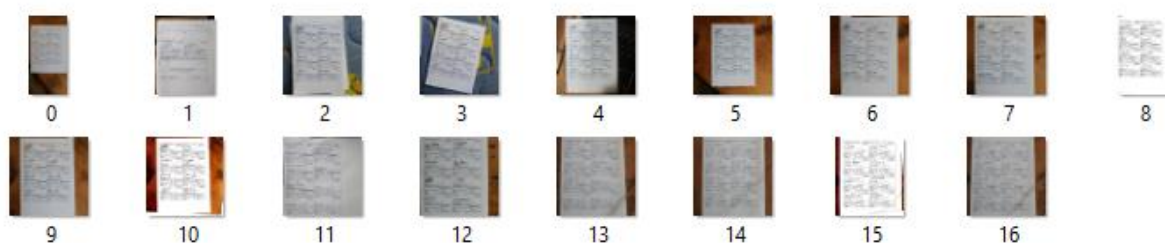


Figura 5.3. Imágenes de la prueba de enderezado de la tarjeta de firmas. Imagen propia. (Datos personales ficticios)

### Análisis del resultado

1. Se realizó la prueba tomándole una foto a cada una de las tarjetas de firmas disponibles, acomodándolas de tal manera que se asemejen a un caso real en el que se encuentren desviadas con respecto a la horizontal de la imagen.
2. De las 17 muestras procesadas se obtuvo una imagen catalogada como error, que no fue enderezada por el programa.

La hipótesis de este fallo es que la imagen tiene de fondo unas líneas, las cuales confunden el programa y no logra identificar correctamente el formulario, además su ángulo de desviación es bastante pronunciado lo cual lo hace más difícil.

3. Mediante esta prueba se evaluó la efectividad del programa, con el cual se obtuvo un error del 5.8%. Se tomaron imágenes en condiciones extremas como, por ejemplo: tarjetas de firmas a diferentes distancias, varias dimensiones para las imágenes, ruido exterior como objetos de colores, líneas, entre otros.

Una vez este proceso esté en funcionamiento, las tarjeas de firmas serán escaneadas en impresoras multifuncionales de muy buena calidad y los fondos de las imágenes no tendrán el ruido con el cual fueron hechas estas pruebas.

### 5.3 Resultados del indexado de la imagen

La indexación resultante de la nueva imagen con el segmento de la firma es el resultado final de este proyecto, luego de las imágenes haber pasado por el proceso de enderezado.

A continuación, en la tabla 5.2 se muestran los datos utilizados para la prueba de indexación, mediante la utilización de los servicios de la nube específicamente de Amazon Textract.

Cabe recordar que todas las pruebas realizadas fueron hechas con un diseño que aún no está implementado de la tarjeta de firmas, y el llenado de la información se realizó con datos ficticios. Para este proceso no era inconveniente hacer estas suposiciones, ya que lo importante era lograr un desarrollo el cual hiciera una identificación de la tarjeta como tal, extrajera de forma adecuada la información requerida y segmentara el recuadro tal como se necesitaba.

Número de tarjetas de firmas	Número de firmas en la tarjeta	Dimensión y tipo de imagen	Tipo de carácter de llenado de información	Número de firmas correctas	Número de firmas con errores
1	8	10MP; Fotografía	Manual	6	2
1	2	16MP; Fotografía	Manual	2	0
1	8	12MP; Fotografía	Manual	8	0
1	8	12MP; Fotografía	Manual	7	1
1	8	4.3MP; Escaneada	Manual	8	0

1	5	12MP; Fotografía	Manual	4	1
1	8	12MP; Fotografía	Manual	7	1
1	5	12MP; Fotografía	Máquina	5	0
1	8	12MP; Fotografía	Máquina	8	0
1	8	8MP; Escaneada	Máquina	8	0
1	8	12MP; Fotografía	Máquina	8	0
<b>Total 11</b>	<b>Total 76</b>			<b>Total 71</b>	<b>Total 5</b>

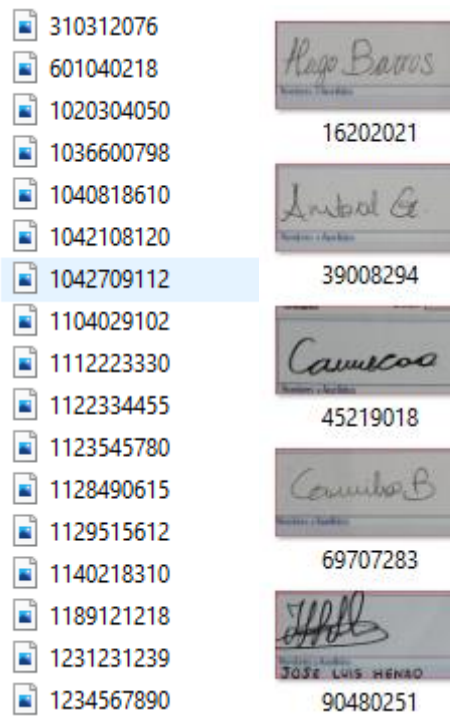
**Tabla 5.2.** Resultado de la prueba de indexación de la firma

### Análisis del resultado

- Las imágenes ingresadas en esta parte del proceso también fueron procesadas previamente por el programa de enderezado del formulario.
- En esta prueba se utilizaron diferentes características para diligenciar los datos de la tarjeta de firmas, por ejemplo:
  - Se tomaron imágenes escaneadas y fotografías desde un celular con diferentes de dimensiones o cantidad pixeles.
  - Algunas tarjetas de firmas se llenaron de forma parcial, con 5 firmas en vez de 8, para probar la efectividad del programa.
  - Algunos números de identificación se llenaron de forma manual y otros con caracteres de máquina con el fin de conocer la respuesta de Amazon Textract ante estas diversas características.
- Se obtuvo como resultado un error del 6.5%; un porcentaje tomando como total los datos escritos de forma manual y caracteres estructurados, en total 11 tarjetas de firmas y 76 firmas.
- El dato importante de esta prueba es el porcentaje de error obtenido con los caracteres estructurados, es decir, con letra de máquina. En este caso no se obtuvo ningún error y la indexación se realizó correctamente, sin fallos en ninguno de los caracteres ni tampoco errores en la relación entre la firma y su respectivo documento.
- Caso contrario ocurrió con las tarjetas de firmas diligenciadas de forma manual, acá se obtuvieron los siguientes errores:
  - Debido a que de forma manual el trazado de los dígitos de la cedula no son constantes, entonces algunos caracteres no los identificó correctamente, por tal motivo los números de identificación personal con los cuales se hacía la indexación eran erróneos en uno de sus dígitos.
  - En uno de los documentos de identificación el OCR de Textract detectó un espacio inexistente entre los dígitos, debido a esto hubo un error en la asignación del documento a la firma correspondiente.
  - Los demás errores presentes fueron debidos a la mala interpretación de algunos de los caracteres, es decir, los identificaba como un dígito cuando en realidad era otro.
  - En conclusión, con los datos escritos manualmente es muy propenso tener errores, pero lo importante de esta prueba es que permitió

obtener datos importantes acerca del funcionamiento del OCR de Amazon Textract.

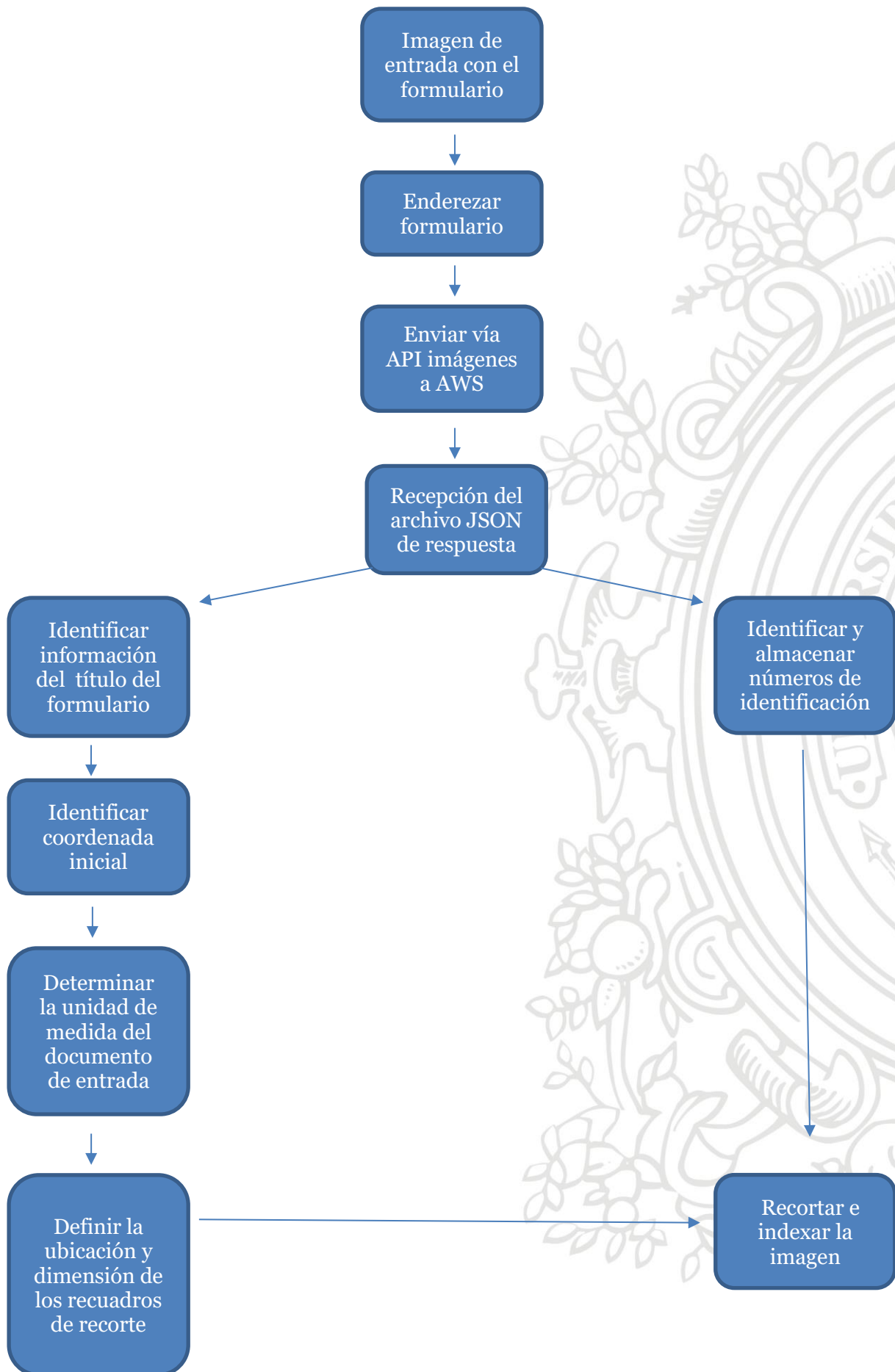
6. A continuación, en la figura 5.4 se observa el resultado final luego del proceso, se muestran dos columnas en las cuales se puede observar de dos vistas diferentes cómo es la apariencia de la respuesta obtenida.



**Figura 5.4.** Indexación de los segmentos de las firmas. Imagen propia. (Datos personales ficticios)

#### 5.4 Estructura final del proyecto.

A continuación, en el siguiente diagrama de flujo se observa un resumen de la estructura del programa en su versión final. Se detalla el proceso en forma general con el fin de ilustrar los pasos realizados.



## 5.5 Costos de la implementación del proyecto

En la búsqueda inicial de una solución para este proyecto se quería hacer un desarrollo totalmente *open-source*, por este motivo se realizaron las pruebas mediante el uso de OpenCV sin utilizar los servicios de AWS, tal como se observó en los resultados del apartado 5.1. Con la respuesta obtenida en estas pruebas se vio la necesidad de utilizar un recurso adicional para obtener los resultados requeridos, ya que con la utilización únicamente de OpenCV no se logró el cometido.

Tal como se vio en el apartado 3.2, la utilización de OpenCV es totalmente gratuita, igual ocurre con el lenguaje de programación Python, son elementos informáticos de los cuales se puede disponer con fines educativos o comerciales, por los servicios que debe pagarse es la utilización de las herramientas de AWS.

A continuación, se observan las figuras 5.5 y 5.6 en las cuales se encuentran los precios de los servicios de Amazon Textract, luego se hace un análisis de estos precios y la manera como se logró bajar los costos.

### API para analizar documentos para páginas con formularios

Región:

Mensual	OCR	Precio por página	Precio en efectivo por 1000 páginas
Primer 1 millón de páginas	Incluido	0,05 USD	50,00 USD
Más de 1 millón de páginas	Incluido	0,04 USD	40,00 USD

\*Los formularios son una recopilación de claves y valores independientes que se encuentran en la misma página (por ejemplo, W2).

**Figura 5.5.** Precios para análisis de un formulario. Captura de pantalla tomada de: (Amazon Web Services I. o., 2020 )

### API para detectar texto de un documento (OCR)

Región:

Mensual	Precio por página	Precio en efectivo por 1000 páginas
Primer 1 millón de páginas	0,0015 USD	1,50 USD
Más de 1 millón de páginas	0,0006 USD	0,60 USD

**Figura 5.6.** Precios para análisis de un documento. Captura de pantalla tomada de: (Amazon Web Services I. o., 2020 )

## Análisis del resultado

1. En la figura 5.5 se encuentran los precios de utilizar el servicio de Amazon Textract para el análisis de una imagen la cual contiene un formulario. Se recuerda, la funcionalidad de Amazon Textract en la cual identifica un formulario es buscar una relación entre la clave y el valor, tal como se describió en el apartado 3.1.1.
2. En las pruebas iniciales con Amazon Textract se estaba utilizando el servicio de reconocimiento de la imagen, para identificar si se trataba de un formulario o una tabla, ya que la tarjeta de firmas como tal es un formulario. Lo que se hacía era buscar el valor de la clave "No." El cual corresponde al número de identificación personal de quien firma.
3. De acuerdo con estos resultados y observando posteriormente los precios de Amazon por este servicio, se replanteó la forma como se realizaba la búsqueda, por lo tanto, se pudo bajar los costos al valor mostrado en la figura 5.6, se observa cómo se pasó de un precio de aproximadamente 0.05 dólares por página a 0.0015 dólares por página.
4. Para lograr bajar los costos, se realizó lo siguiente:
  - No se buscó una clave-valor, o *key-value* como se mencionó anteriormente, sino que se hizo una búsqueda de todos los números de identificación presentes en el formulario y se almacenaron en un arreglo, para luego hacer la indexación de las imágenes recortadas.
  - El proceso requerido al pasar por el OCR de Amazon Textract consistía en identificar el título de la tarjeta de firmas y extraer los números de documento, entonces se pudo omitir el análisis del tipo de formato presente en la imagen, si se trataba de un formulario o de una tabla.
  - Con base en el manual disponible de Amazon Textract, se observó que, con solo cambiar la función de llamado de la API desde el programa en Python se puede hacer un análisis del tipo de formato del documento en la imagen o solo hacer un reconocimiento de caracteres por parte del OCR. (En el apartado 4.4.2 se habló de estas funciones.)

## 6 Conclusiones

- Se desarrolló una aplicación funcional que se encarga de realizar la extracción de la firma desde un documento llamado tarjeta de firmas y lo indexa con el número de identificación personal del firmante.
- Para el desarrollo de este proyecto no se sugirió por parte de los proponentes la utilización de ningún servicio en especial, solo se pidió que este cumpliera con la funcionalidad solicitada, la utilización de servicios legales y dentro de lo posible buscar un desarrollo *open source* para lograr disminuir al máximo los costos del proceso. Todo el desarrollo y herramientas utilizadas fueron seleccionadas con base en la investigación y pruebas hechas a los servicios ofrecidos por las plataformas digitales.



- Se determinaron las necesidades y requisitos funcionales del proceso solicitado y con base a esto se estableció la estructura del proyecto y las herramientas adecuadas para cumplir con las tareas requeridas. Se optó entonces como base para este desarrollo el OCR disponible en Amazon Textract, la librería de visión artificial OpenCV y el lenguaje de programación Python.
- Debido a las restricciones en el uso de software dentro de la red interna y equipos del banco se requirió estudiar los protocolos establecidos para la descarga de librerías y programas o entornos de desarrollo, todo esto para evitar posibles amenazas a la seguridad informática de la empresa, lo cual es una prioridad por tratarse de una entidad bancaria.
- Las pruebas hechas a este proyecto utilizando un documento en formato real, permitió hacer análisis y obtener valores de error y efectividad en el desarrollo que fueron de 6.5% y 93.5% respectivamente. Se pudo observar el comportamiento acertado en la identificación de la información cuando el documento está diligenciado con caracteres estructurados, además estas pruebas permitieron determinar procesos más óptimos que se lograron alcanzar en este desarrollo, tratando siempre de optimizar recursos monetarios y de ejecución.
- Con la realización de las pruebas al enderezado de la imagen se pudo encontrar la probabilidad de error, que fue del 5.8%, además realizar algunas modificaciones importantes al programa, en este caso, las pruebas hechas permitieron identificar unos ajustes necesarios al ángulo de inclinación admitido por el programa para el formulario en la imagen, y también se pudo determinar que era necesario un manejo de excepciones para evitar el colapso en la ejecución del programa cuando la imagen no podía ser alineada.

## 7 Referencias bibliográficas

abulafia. (28 de Febrero de 2018). *Stackoverflow*. Obtenido de <https://es.stackoverflow.com/questions/141818/como-enderezo-una-imagen-en-python?answertab=votes#tab-top>

Amazon Web Services, I. a. (2020). *Amazon Textract: Developer Guide*.

Amazon Web Services, I. o. (2020 ). *Precios de Amazon Textract*. Obtenido de <https://aws.amazon.com/es/textract/pricing/>

Amazon Web Services, I. o. (2020). *Amazon Textract features*. Obtenido de Optical Character Recognition (OCR): <https://aws.amazon.com/es/textract/features/>

Amazon Web Services, I. o. (2020). *Calling Amazon Textract Synchronous Operations*. Obtenido de Documents Stored in an Amazon S3 Bucket : <https://docs.aws.amazon.com/textract/latest/dg/sync-calling.html>

Amazon, W. S. (2020). *Amazon S3*. Obtenido de <https://aws.amazon.com/es/s3/>

Amazon, W. S. (2020). *Amazon Textract*. Obtenido de <https://aws.amazon.com/es/textract/>

Amazon, W. S. (2020). *Características de AWS Lambda*. Obtenido de <https://aws.amazon.com/es/lambda/features/>

Amazon, W. S. (2020). *HumanLoopConfig*. Obtenido de [https://docs.aws.amazon.com/textract/latest/dg/API\\_HumanLoopConfig.html](https://docs.aws.amazon.com/textract/latest/dg/API_HumanLoopConfig.html)

Amazon, W. S. (2020). *Informática en la nube con AWS*. Obtenido de <https://aws.amazon.com/es/what-is-aws/>

Bancolombia, G. (2020). Obtenido de <https://www.grupobancolombia.com/wps/portal/acerca-de/informacion-corporativa/quienes-somos>

*Introducción a JSON*. (s.f.). Obtenido de <https://www.json.org/json-es.html>  
npm. (Abril de 2020). *aws-azure-login*. Obtenido de <https://www.npmjs.com/package/aws-azure-login>

OpenCV, t. (2020). *OpenCV About*. Obtenido de <https://opencv.org/about/>

Seidor. (2020). *¿Qué es Node.js y para qué sirve?* Obtenido de <https://www.drauta.com/que-es-nodejs-y-para-que-sirve#:~:text=Qu%C3%A9%20es%20Node.js%20y%20para%20qu%C3%A9%20sirve%3F,m%C3%A1s%20presente%20en%20el%20mercado.>