



**UNIVERSIDAD  
DE ANTIOQUIA**

**DESARROLLO DE UN ALGORITMO MEDIANTE TÉCNICAS DE MACHINE LEARNING DE PROCESAMIENTO DEL LENGUAJE NATURAL QUE PERMITA LA GENERACIÓN DE TEXTOS RESUMEN DEL ESTADO DE LOS PACIENTES DE LA IPS NEUMOMED S.A.S. CON BASE EN INDICACIONES MÉDICOS OBTENIDOS DE LA BASE DE DATOS DE PACIENTES, PARA SU FUTURA IMPLEMENTACIÓN COMO UN SERVICIO DE LA HISTORIA CLÍNICA.**

Autor

**ANDRÉS SEBASTIÁN GONZÁLEZ BUITRAGO**

Universidad de Antioquia

Facultad de Ingeniería

Medellín, Colombia

2021



**DESARROLLO DE UN ALGORITMO MEDIANTE TÉCNICAS DE MACHINE LEARNING DE PROCESAMIENTO DEL LENGUAJE NATURAL QUE PERMITA LA GENERACIÓN DE TEXTOS RESUMEN DEL ESTADO DE LOS PACIENTES DE LA IPS NEUMOMED S.A.S. CON BASE EN INDICACIONES MÉDICOS OBTENIDOS DE LA BASE DE DATOS DE PACIENTES, PARA SU FUTURA IMPLEMENTACIÓN COMO UN SERVICIO DE LA HISTORIA CLÍNICA.**

**Andrés Sebastián González Buitrago**

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:

**Bioingeniero**

Asesores (a):

Jonathan Gallego Londoño, Bioingeniero, M.Sc Ingeniería Biomédica

Luisa Fernanda Fernandez Madrid, Bioingeniera.

Línea de Investigación:

Informática médica

Universidad de Antioquia

Facultad de Ingeniería

Medellín, Colombia

2021

# CONTENIDO

## INTRODUCCIÓN

1.1. Planteamiento del Problema	1
1.2. Objetivos	2
1.2.1 Objetivo General	2
1.2.3 Objetivos Específicos	2
2. MARCO TEÓRICO	4
3. METODOLOGÍA	10
4. RESULTADOS	18
4.1 Entrenamiento del modelo de machine learning	18
4.2. Evaluación de desempeño del algoritmo de machine learning	22
4.3. Implementación del modelo de machine learning	23
4.4. Construcción automática de resúmenes de consultas implementado el algoritmo	26
4.5. Evaluación manual de la calidad de los resúmenes generados y re-entrenamiento correctivo del sistema	27
4.6. Implementación de arquitectura cliente- servidor	27
4.7. Implementación de seguridad en el proyecto	28
5. CONCLUSIONES Y RECOMENDACIONES	32
6. REFERENCIAS	33

## RESUMEN

*La atención médica no suele ser un asunto limitado a la consulta o urgencia como evento único, sino que se extiende a muchos más aspectos. Aquí se va a destacar la importancia de un seguimiento médico adecuado y algunas de sus problemáticas y posibles rutas de soluciones en un mundo que avanza galopante a una integración de la tecnología en todos y cada uno de los aspectos de la vida. Trabajando sobre una problemática en la capacidad de seguimiento a muchos indicadores importantes en un paciente, ya que debido al alto volumen que puede llegar a tener un médico hoy en día y su limitado tiempo disponible para realizar una lectura profunda a la historia clínica. El proyecto se da en este contexto que la IPS Neumomed S.A.S ha identificado junto al equipo de TI, ideando así en una herramienta mediante el uso de técnicas de Machine learning que permite la construcción de textos resúmenes, que tengan información relevante y trazable del paciente en sus distintas etapas dentro de un programa médico especializado, haciendo de elemento contextualizar en su estado de salud y útil para tener ciertos criterios en cómo se puede proceder de la mejor forma y tener una certeza cuantificable en la efectividad del tratamiento seleccionado desde un principio. Esto da la posibilidad de poder evaluar en cómo seguirá el tratamiento, o una posible reestructuración del plan médico que lleve a mejores resultados, todo bajo un criterio especializado.*

**Palabras clave:** Procesamiento del lenguaje natural, Machine learning, seguimiento médico, árboles de decisiones.

## ABSTRACT

*Medical care is not usually a matter limited to the consultation or emergency as a single event, but extends to many more aspects. Here we will highlight the importance of adequate medical monitoring and some of its problems and possible solution routes in a world that is advancing galloping towards the integration of technology in each and every aspect of life. Working on a problem with the ability to monitor many important indicators in a patient, due to the high volume that a doctor can have today and his limited time available to perform a deep reading of the medical history. The project takes place in this context that the IPS Neumomed SAS has identified together with the IT team, thus devising a tool through the use of Machine learning techniques that allows the construction of summary texts, which have relevance and traceable information of the patient in its different stages within a specialized medical program, making it an element to contextualize your state of health and useful to have certain criteria on how you can proceed in the best way and have a quantifiable certainty in the effectiveness of the treatment selected from the beginning. This gives the possibility of evaluating how the treatment will continue, or a possible restructuring of the medical plan that leads to better results, all under specialized criteria.*

**Keywords:** Natural language processing, Machine learning, medical monitoring, decision trees.



## INTRODUCCIÓN

### 1.1. Planteamiento del Problema

La atención oportuna y de la mano de profesionales experimentados en el área de la salud es la condición ideal para cualquier tratamiento patológico, acompañado de información suficiente y un profundo estudio de las condiciones reportadas del paciente, permiten que el tratamiento sea el más adecuado [1], y por ende con una recuperación oportuna. Cumplir estas condiciones se hace complejo ya que un nivel de experiencia para un profesional de salud, toma una gran cantidad de tiempo y es poco el personal disponible en el campo con éstos estándares de formación, así mismo, la información de un paciente suele pasar de profesional a profesional y un estudio detallado se vuelve un proceso extenuante, el cual puede recaer en un segundo plano por la urgencia del tratamiento inmediato y la toma de decisiones para su manejo integral y adecuado, o en pacientes con un tratamiento de larga data, perder la trazabilidad de la condición del paciente [1]. Esta situación en donde el personal médico se encuentra sin opciones para el manejo de la información de un gran número de pacientes, hace que se planteen nuevas soluciones por medio del uso de tecnología de la información. Los algoritmos de Machine learning han sido puestos como una opción viable para el manejo y categorización de la información médica, aunque aún para el año 2019 era pequeño su aporte en el campo, haciendo necesario un cambio en los paradigmas de los sistemas de salud actuales para que se puedan adaptar a los beneficios que traería esta tecnología [1]. La IPS Neumomed S.A.S es una clínica especializada en sueño y el tratamiento de pacientes con afecciones respiratorias, esta institución cuenta con distintos programas médicos, los cuales tienen distintos criterios de inclusión, que van atados a condiciones patológicas, sintomatología, y objetivos deseados por los pacientes. Estos programas son: EPOC, SAHOS, ASMA, Rehabilitación pulmonar y Dejar de fumar, este último es la ejemplificación de lo antes mencionado. Cada uno de estos programas cuenta con un plan de terapias diseñadas para cada paciente según su condición, esto es dependiente del cluster diseñado por el equipo profesional. Actualmente la IPS cuenta con un número alto de pacientes, estas condiciones han evidenciado la falta de opciones de acceso para conocer indicadores e información global de los mismos, así un correcto

seguimiento en la mejora de la condición se hace un proceso lento al requerir una revisión manual al histórico con el que cuenta la institución. Por lo tanto se hace incierto generar indicadores de cuánto ha progresado en el programa su patología, y así conocer efectos de la terapia de forma cuantificable. Se ha propuesto en este proyecto la identificación de variables que puedan ser marcadas como etiquetas en la información que se recolecta del paciente mediante distintos formularios, permitiendo la trazabilidad de estas condiciones en el tiempo que lleve el paciente en el programa, mediante la implementación de algoritmos de Machine Learning (ML) de procesamiento del lenguaje natural (NLP) para generar textos resumen del paciente que contengan información médica relevante como indicadores numéricos o descriptivos del estado de salud del mismo. Esto con el fin de identificar la efectividad del plan de tratamiento en el programa y así tener una noción precisa de cuando sea necesario implementar cambios en él, sea porque el paciente tuvo la respuesta al manejo esperado y haya mejorado o no esté siendo efectivo [1].

## **1.2. Objetivos**

### **1.2.1 Objetivo General**

- Desarrollar un algoritmo mediante técnicas de ML de procesamiento del lenguaje natural que permita la generación de textos resumen del estado de los pacientes en un programa médico de la IPS Neumomed S.A.S, con base en indicadores médicos obtenidos de la base de datos de pacientes para su futura implementación como un nuevo servicio de la historia clínica.

### **1.2.3 Objetivos Específicos**

- Diseñar un algoritmo de procesamiento de lenguaje natural basado en técnicas de machine learning capaz de procesar la información de las historias clínicas de la IPS con la finalidad de generar textos resumen con información relevante de pacientes



dentro de un programa médico.

- Entrenar el algoritmo de generación de texto usando las bases de datos del programa médico de la IPS Neumomed S.A.S con la finalidad de obtener un texto lógico y coherente del estado del paciente.
- Implementar el algoritmo de generación de texto usando datos del mismo programa médico y distintos a los datos de entrenamiento para corroborar un correcto funcionamiento del modelo.
- Validar la calidad y cohesión del texto generado por el algoritmo mediante pruebas de testeo manual en conjunto con un equipo profesional de la IPS ó mediante un testeo automático usando librerías capaces de analizar la gramática y sintaxis de un texto y encontrar los errores cometidos.

## 2. MARCO TEÓRICO

En el contexto de la IPS, el programa médico busca que los pacientes con las enfermedades en las que se enfocan, realicen una serie de ejercicios con la finalidad de mejorar su calidad de vida y el autocontrol de su enfermedad. Éste programa está regido por un terapeuta, que se encarga de hacer un seguimiento adecuado al plan de tratamiento, que fue diseñado por un equipo médico experto. La IPS cuenta distintos programas médicos, cada uno para una distinta patología o rehabilitación, estos cuentan con una terapia específica, la cual se define por un conjunto de profesionales que se le asigna al paciente para el tratamiento de patología o necesidad que requiera según la sintomatología que reporta. Con esta información ya recolectada, se designa un plan de tratamiento, el cual se basa en un conjunto de ejercicios, medicamentos, meditaciones y otros cuidados que el paciente día a día debe tomar siguiendo indicaciones dadas, y con un seguimiento periodico. Éste seguimiento y mediante el uso de formularios que se van llenando en cada uno de los encuentros, permiten la creación de la base de datos que lleva a la implementación interna del servicio de historia clínica, ya construido mucha de esta información se puede declarar como indicadores en la base de datos, estos se definen como demarcaciones de variables clínicamente relevantes para la construcción de textos mediante el algoritmo propuesto en el proyecto. Estos indicadores servirán para el entrenamiento del algoritmo.

El ML es una tecnología que se basa en el uso de herramientas de la computación, donde se construyen modelos mediante el entrenamiento o aprendizaje del mismo. Una vez alcanzado un nivel de entrenamiento, suelen ser usados mediante máquinas virtuales con accesibilidad remota, como también se pueden trabajar de forma local, integrándose a macroprocesos en distintos campos de la industria. Haciendo posible, la integración de estos a aplicaciones de tipo web, lo que permite una globalización de la información y mayor accesibilidad a esta [2]. En la actualidad, uno de los mayores desafíos de las diferentes instituciones es automatizar procesos como los anteriormente mencionados. Siendo los algoritmos basados en inteligencia artificial, una de las herramientas de tipo software más implementadas en los últimos años. Siendo una de estas el procesamiento de lenguaje natural NLP (natural language processing), que se puede definir como la capacidad que tiene una computadora para comprender el lenguaje humano tal como se

habla. De igual modo, los algoritmos de NLP pueden entrenarse para que aprendan a reconocer y reconstruir textos que tengan una estructura definida, siendo este el caso de los resúmenes de las consultas médicas. Es decir, se puede entrenar un algoritmo por medio de técnicas de ML para que reciba los datos recolectados durante una consulta médica y posteriormente genere de manera automática el resumen de esta [3].

Ahora, por efectos de manejo de considerable cantidad de opciones o de información, muchos de los modelos desarrollado en ML se han orientado a un aprendizaje supervisado, el cual se describe como un método en donde las muestras de entrenamiento especifican los atributos a predecir, y se busca la asignación de dichos atributos a una entrada de datos nuevos sin atributos asignados, a su vez éstos algoritmos pueden dividirse en dos:

- Clasificación: las muestras de entrenamiento deben tener especificados al menos dos o más atributos, de esta forma el algoritmo, una vez entrenado, es capaz con una entrada aún sin atributo, clasificar esta nueva variable.
- Regresión: donde se quiere aprender a predecir el valor de un atributo de los datos, a partir de sus demás atributos.

El entrenamiento es una parte esencial en estos modelos, y muchas veces se hará necesario un re-entrenamiento según se vayan presentando distintas necesidades, como lo pueden ser la implementación de un nuevo atributo, o incluso el cambio estadístico de alguna variable puede hacer que los modelos se vuelvan obsoletos, estos debido a que estos modelos cuentan con parámetros y hiperparámetros que regulan el desempeño y la capacidad de aprendizaje del algoritmo, estos parámetros son determinados ya sean por métodos estadístico o incluso por iteración de ensayo y error [4 , 5]. En los diferentes lenguajes de programación que permiten la creación de modelos de ML existen librerías que tienen herramientas propias para ayudar a ajustar estos parámetros [6].

En este proyecto se consideraron distintos modelos de aprendizaje supervisado, tales como las redes neuronales, los árboles de decisiones y las regresiones logísticas. Debido a las



etiquetados correctamente, verdaderos negativos (TN) o sujetos no asignados a una etiqueta falsos positivos (FP) sujetos etiquetados en un grupo incorrecto y falsos negativos (FN) sujetos no asignados a la clase que pertenecen. Conociendo como se etiquetaron los sujetos se puede calcular la precisión(p) y recall(r) usando las ecuaciones 1 y 2.

$$p = \frac{TP}{TP + FP}$$

Ecuación 1: Cálculo de precisión (p)

$$r = \frac{TP}{TP + FN}$$

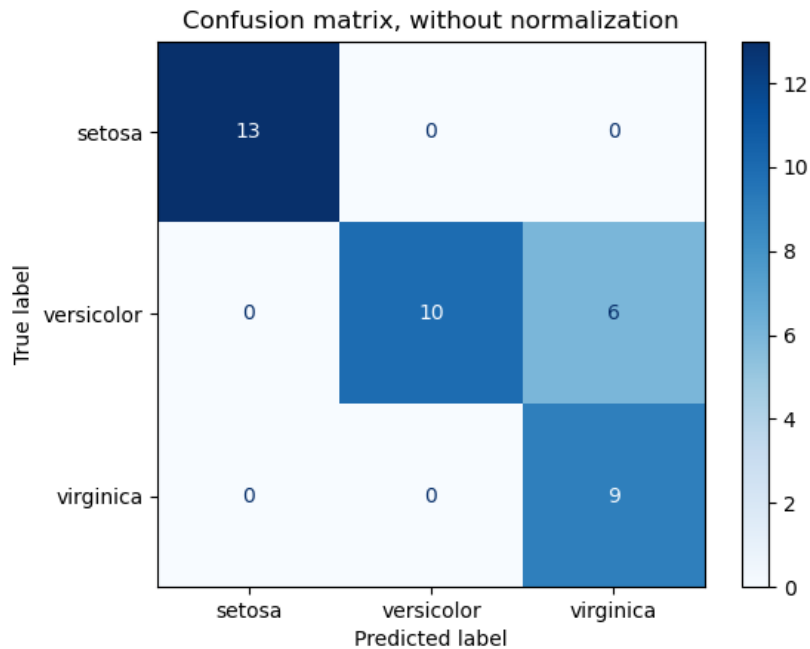
Ecuación 2: Cálculo de recall (r)

Aunque ambas ecuaciones muestran información relevante del modelo, son variables ciertamente limitadas, pero para un mejor panorama existen las F-score, que juntan en una sola medida la precisión y el recall. se calculan por medio de la ecuación 3[9].

$$F_{\beta} = (1 + \beta^2) \frac{pr}{r + \beta^2 p} = \frac{(1 + \beta^2)TP}{(1 + \beta^2)TP + \beta^2 FN + FP}$$

Ecuación 3: cálculo del valor F-score

Otra forma de evaluar el desempeño de un algoritmo o modelo de clasificación es la matriz de confusión, especialmente significativa en clasificaciones multiclase. Esta matriz relaciona los sujetos de una clase con las etiquetas asignadas a dichos sujetos y muestra los sujetos etiquetados de manera correcta situados en la diagonal principal de la matriz y qué etiquetas asignó a los sujetos que fueron mal clasificados como se muestra en la Figura 2 [10].



**Figura 2.** Ejemplo de matriz de confusión para clasificación de tipos de flores

Para la implementación de estos modelos, se usaron librerías disponibles y escritas en Python, el cual fue el lenguaje de programación en el cual se desarrolló todo el proyecto.

Las principales librerías usadas fueron:

- Scikit learn: Es una librería de licencia libre disponible en Python, desarrollada con un enfoque en rutinas de machine learning, simplificadas y de fácil levantamiento, además de poseer módulos de procesamiento de datos y análisis de modelos entrenados [11].
- Tensor Flow : Es una librería open source desarrollada por Google y lanzada en 2015, enfocada en dar soluciones de machine learning de alto y bajo nivel, especializada en redes neuronales y procesamiento tensorial, cuenta con módulos de análisis de resultados y monitoreo del entrenamiento [12].
- Pandas: Es una librería especializada en análisis de datos, la cual provee de mano de las librerías más comunes como Numpy, una cantidad impresionante de

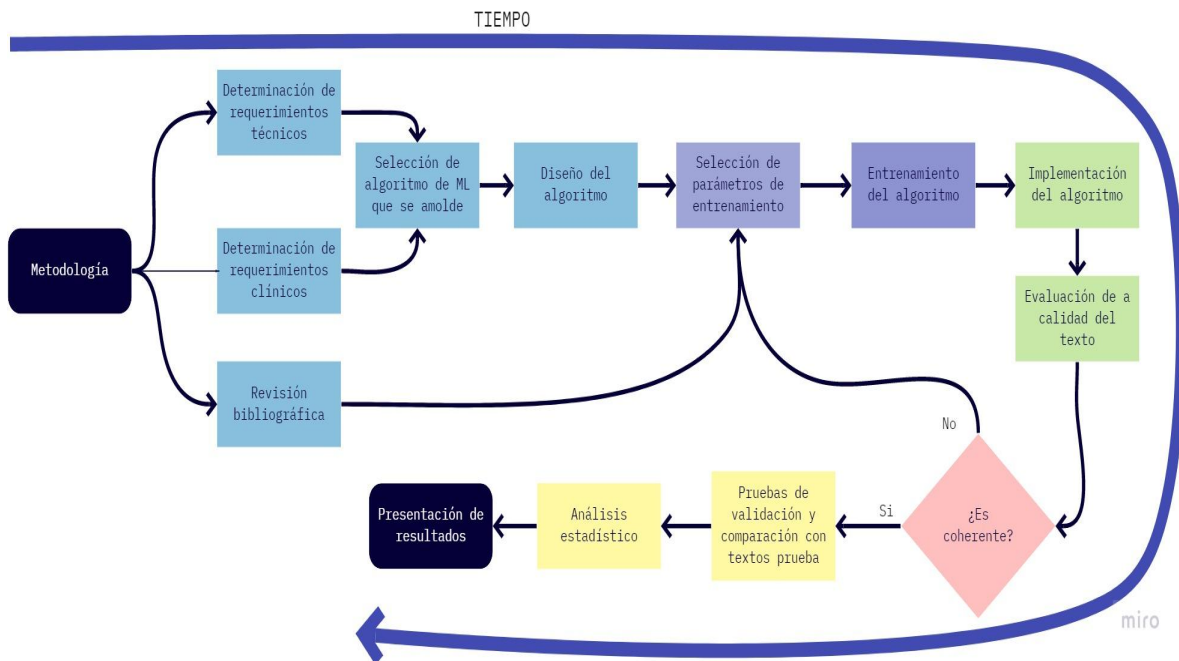
funcionalidades para el manejo e interpretación de una gran cantidad de datos relevantes en un dataset entregado [16].

Los modelos construidos a partir de las diferentes librerías se estructuran empleando un patrón de arquitectura de software. En dichos patrones se expresa una descripción de los subsistemas y componentes de un sistema de software que establece las relaciones entre ellos. Dichos subsistemas y componentes generalmente se especifican en diferentes vistas para mostrar las propiedades funcionales y no funcionales relevantes de un sistema de software, la definición de un patrón de arquitectura es el resultado principal de la actividad de diseño de software [13].

La arquitectura usada en este proyecto fue la de servidor-cliente, la cual es una arquitectura que se utiliza cuando el servidor y el cliente se conectan a una red. Aquí el servidor actúa como proveedor de servicios y el cliente es el consumidor del mismo. Normalmente el servidor está ubicado en internet, sin embargo, en los casos de manejo de información sensible el servidor se puede ubicar en una red de área local, garantizando que los usuarios externos no puedan acceder al servidor, pero los usuarios internos sí [14]. Una vez desarrollados los modelos, estos se pueden levantar en un servidor y funcionar como una aplicación (API) a la cual se consulta la predicción. En este caso se llevó a usar Flask, el cual es un framework enfocado al desarrollo web y especialmente al levantamiento de servidores de manera rápida y segura, donde en el mismo servidor pueden estar modelos y funciones, para poder tener un mejor manejo y acceso a ellos [15].

### 3. METODOLOGÍA

En la figura 3 se puede observar un mapa mental de la metodología que usó en el proyecto, como aclaración se especifica que los colores de cada recuadro van relacionados al objetivo específico que están respondiendo, azul claro para el primero y así siguiendo el flujo del tiempo y los colores.



**Figura 3.** Metodología del proyecto

#### 3.1. Diseño de algoritmo

**3.1.1. Revisión bibliográfica:** Se realizó una búsqueda bibliográfica en distintas fuentes acerca de investigaciones en ML similares, manejo de bases de datos, modelos de ML supervisados y procesamiento del lenguaje natural, todo desde el enfoque de desarrollo usando Python como lenguaje principal. Ésto con propósito de determinar de forma más coherente las herramientas necesarias en el proyecto.

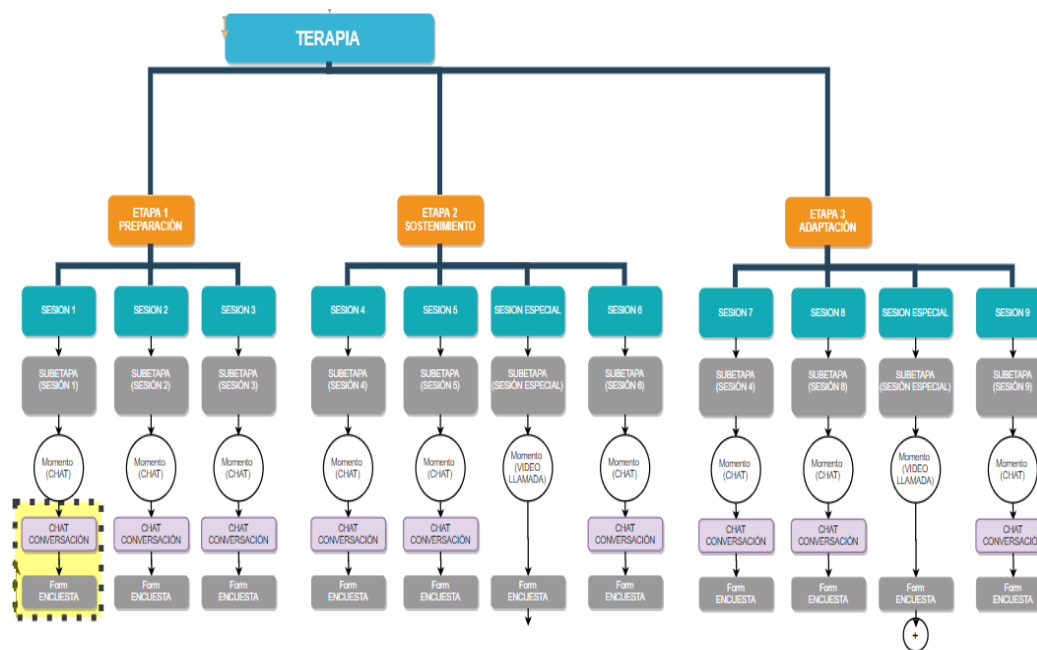


**3.1.2. Determinación de requerimientos técnicos:** Se determinó el funcionamiento y requerimientos básicos para la implementación del algoritmo en la base de datos de historias clínicas de la IPS, y se identificaron las problemáticas presentadas con soluciones que se intentaron implementar en el pasado, como lo han sido la seguridad y el consumo de recursos del proyecto. Los requerimientos técnicos que se definieron para el algoritmo se presentan en la Tabla 1.

*Tabla 1. Requerimientos técnicos*

R 01	Obtención y extracción de información relevante y cuantificable del paciente en el programa médico. Esta información se relaciona con indicadores clínicos específicos dentro del programa médico.
R 02	Generación automatizada de textos resumen. Esta función deberá generar textos resúmenes por medio de la aplicación de técnicas de machine learning
R 03	Creación de un servicio con potencial de escalabilidad e interoperabilidad entre otros servicios El servicio generado debe poder interactuar con los otros servicios existentes en el asistente virtual de Neumomed.
R 04	Creación de un servicio que sea consumible desde diferentes clientes El servicio debe poderse visualizar en cualquier equipo mientras este tenga un navegador y acceso a internet.
R 05	Calidad La funcionalidad del sistema debe ser de buena calidad para generar confiabilidad a los usuarios.
R 06	Rendimiento El servicio debe tener una adecuada velocidad de respuesta y el menor consumo de recursos posible
R 07	Mantenimiento El sistema estará en constante mantenimiento, verificación y actualización.
R 08	Restricciones -El servicio solo permite que solo los encargados de las áreas puedan ingresar -Una vez terminada la consulta los textos no pueden alterarse

**3.1.3 Determinación de requerimientos clínicos:** Junto con el equipo de TI y un profesional familiarizado con el proceso de selección de estos requerimientos, se hallaron puntos en común con trabajos anteriores y también distintos requerimientos nuevos que se podrían llegar a necesitar. Como resultado se consideró indispensable conocer la estructura básica de distintos programas médicos en la IPS y se estructuró un requerimiento basado en una terapia específica dentro de un programa, sus formularios y condiciones básicas de las variables que éstos requerían, y así se determinó la necesidad de hacer un seguimiento a variables de tipo numérica, presenten y trazables a lo largo de la terapia. En la Figura 4 se puede observar un diagrama básico de la estructura usada en una terapia.



**Figura 4.** Ejemplo de terapia dentro del programa seleccionado

A continuación, se presenta la estructura definida para el texto:

Título del evento médico

- Información general del paciente: este texto debe contener información de identificación del paciente, algunos de los elementos presentes en este son:
  - Nombre del paciente

- Edad
  - Género
  - Estado Civil
  - Profesión u Ocupación
  - Fecha de Ingreso
- Motivo de última consulta, enfermedad actual y programas actuales: se indica la razón por la cual el paciente asiste a la consulta, los síntomas y padecimientos manifestados por el paciente y la enfermedad base diagnosticada (EPOC, SAOS, asma, entre otras) así como una especificación del programa médico en el que se encuentra.
- Antecedentes: se refiere a todos los antecedentes patológicos del paciente, estos pueden ser de tipo:
- Familiares
  - Personales
  - Fisiológicos
  - Patológicos
  - Maritales
  - Quirúrgicos
  - Psicológicos
  - Psiquiátricos
  - Entre otros
- Examen físico general: se refiere al estado físico del paciente en el momento de la consulta, las variables incluidas en este son:
- Signos vitales
  - Peso
  - Talla
  - Medidas antropométricas
- Hallazgos en la consulta: se indican los hallazgos médicos tanto positivos como negativos pertinentes que permitan describir el estado de salud actual

del paciente, y sus variables de encuentro trazables a lo largo de la terapia.

- Concepto: opinión profesional emitida por el área asistencial tras considerar los hallazgos en la consulta.
- Observaciones: se incluyen las apreciaciones médicas particulares de cada casa

**3.1.4 Selección de algoritmo de ML que mejor se adapte:** Debido a la anterior implementación en la IPS y también por contar con distinta documentación acerca del desarrollo de software para éste tipo de aplicaciones en NLP, así como también aplicaciones en distintos campos de bioingeniería, ser código abierto, un buen soporte en el idioma objetivo e implementable en Python con todos los protocolos de seguridad de datos el modelo seleccionado fue un árbol de decisiones.

- *Diseño de algoritmo:* El algoritmo se basó en la librería scikit learn de python, como parámetros base se escogieron los que la librería recomienda para problemas de clasificación multiclase. El árbol de decisiones obtenido tras la implementación de la librería se ilustra en la Figura 5. En la lectura de este es posible notar que el número mínimo de decisiones que el algoritmo toma para generar el texto son 3 y el máximo son 6, esto para un manejo adecuado de variables extraídas de la historia clínica del paciente



**Figura 5.** representación gráfica del árbol de decisiones construido

### 3.2. Entrenamiento del algoritmo

**3.2.1 Selección de parámetros de entrenamiento:** en base en los antecedentes se definieron los parámetros de entrenamiento del modelo.

Para la selección de los parámetros relacionados con la profundidad del árbol y el número de divisiones, se realizó un ajuste paramétrico por medio de regresiones logísticas, ésta validación se realizó anteriormente, pero se aplicaron distintos criterios para la obtención de una precisión del 100% y posible ampliación de la profundidad de los árboles.

**3.2.2 Entrenamiento del algoritmo:** se crearon las matrices de entrenamiento del modelo por medio de un ciclo de condicionales y se determinó que estas fueran de tipo binario. A su vez, se implementó una automatización a este proceso, mediante un microservicio adicional, el cual con indicaciones básicas, generaba y entrenaba esta matriz con los parámetros exigidos desde un inicio.

### **3.3. Implementación del algoritmo**

**3.3.1. Implementación del algoritmo:** Una vez construido y entrenado el modelo, se procede a la implementación del algoritmo, usando datos simulados y posteriormente datos de un programa médico seleccionado para validar así los datos. Se hizo una validación tanto con librerías de Python como con el equipo TI de la IPS.

Además se declararon condiciones específicas para tener en cuenta acerca del tipo de datos simulados, un periodo de tiempo en el que se extrajo información de la historia clínica definida de 2018-2020 y también una restricción en la cantidad de variables a tratar como trazables. A partir de los textos extraídos de la lógica anterior, se construyeron nuevas plantillas de entrenamiento del modelo buscando que se conservara la estructura que se indicó desde el requerimiento clínico inicial.

**3.3.2. Evaluación de la calidad del texto:** Se hará una revisión de los textos construidos por el algoritmo, buscando fallos e incongruencias, esto se hará con la ayuda del personal de la IPS. En caso de no ser aprobado este paso, se someterá al código a una revisión y volviendo al paso 4.2.1.

Además se implementó el algoritmo entrenado usando postman para simular la interacción cliente-servicio con la que se diseñó el algoritmo. Se implementaron condiciones de autenticación y cifrado de la información en el algoritmo para garantizar la confidencialidad en el tratamiento de información sensible. Finalmente para su consumo se hizo uso del diseño básico de interfaz gráfica en la historia clínica digital de la IPS para consumir el algoritmo de machine learning.

### **3.4. Validación del algoritmo**

**3.4.1. Prueba de validación y comparación con textos pruebas:** se revisaron uno por uno los textos construidos por el algoritmo con el fin de identificar patrones de

fallos, faltas de ortografía o carencias de redacción en el texto.

La evaluación del texto se realizó por un equipo conformado por: 2 miembros del departamento de ingeniería, uno de medicina y uno de comunicaciones.

- Preguntas criterio de evaluación médica:
  - ¿El texto es coherente?
  - ¿El texto es técnicamente válido?
  - ¿Se incluyen todas las variables relevantes en el texto?
  - ¿El texto cumple con la estructura que se definió desde los requerimientos clínicos?
- Preguntas criterio de evaluación de ingeniería:
  - ¿El texto tiene fallos de construcción que estén inminente relacionados con el modelo?
  - ¿El texto presenta variables nulas o vacías?
  - ¿El texto cumple con la estructura que se definió desde los requerimientos clínicos?
- Preguntas criterio de evaluación de comunicaciones:
  - ¿El texto es de fácil lectura?
  - ¿El texto es entendible?
  - ¿El texto tiene buena ortografía?

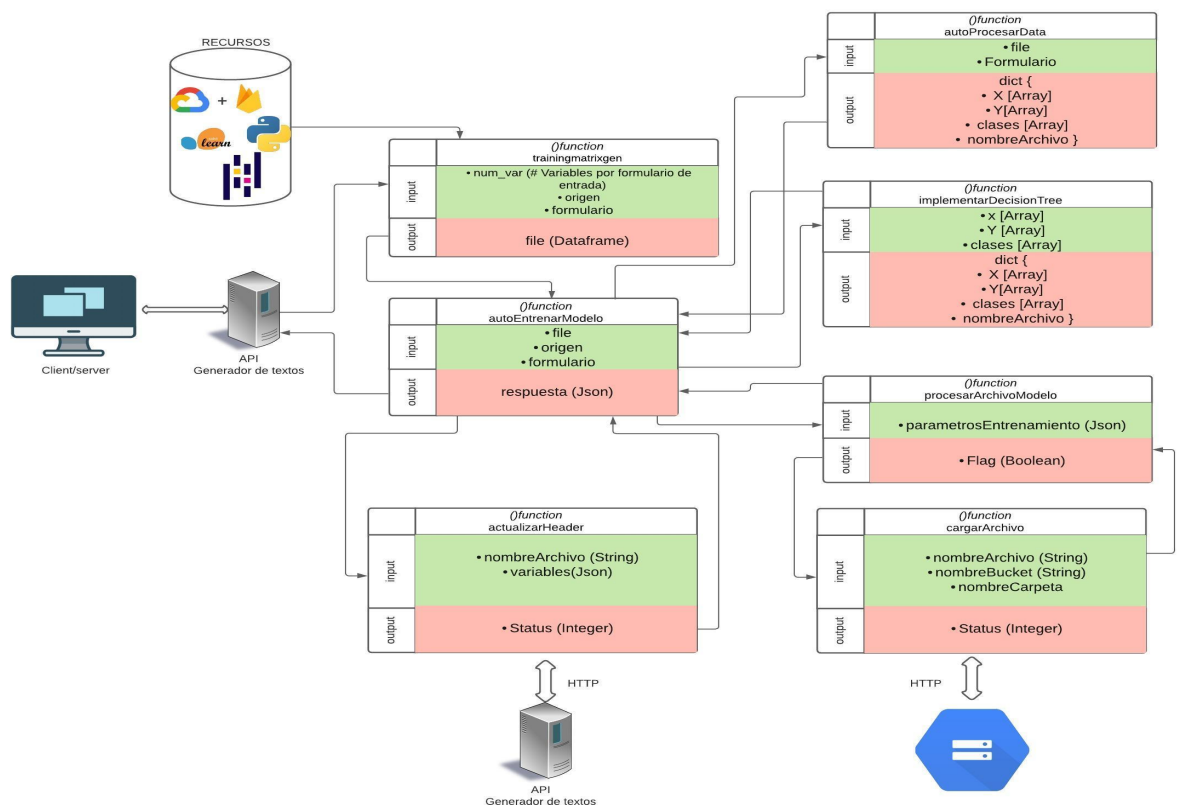
**3.4.2 Análisis estadístico:** se realiza un análisis estadísticos de los tiempos de ejecución del algoritmo en comparación con las soluciones que existían anteriormente en la IPS para la generación de textos. Adicionalmente se evalúa estadísticamente la asertividad gramatical de los textos generados por el algoritmo de machine learning diseñado en el presente proyecto. Esta revisión se realizó por medio de la implementación de la librería grammar-check de python, la cual se encarga de analizar la gramática y sintaxis de un texto y encontrar los errores cometidos.

## 4. RESULTADOS

### 4.1 Entrenamiento del modelo de machine learning

En la Figura 6 se detalla el diagrama UML de ejecución del algoritmo de entrenamiento del modelo actualizado, en este diagrama se representa el consumo de los microservicios e información relevante como:

- Las librerías necesarias para la implementación
- El flujo de consumo de microservicios, así como sus relaciones y requerimientos en entrada y salida
- Bases de datos implementadas para el almacenamiento
- Funciones intermedias de consumo de otros servicios y nombre del servicio consumido



**Figura 6.** Diagrama UML del consumo del algoritmo de entrenamiento del modelo



En la Tabla 2 se presenta un diccionario con las funciones representadas en el diagrama.

*Tabla 2. Diccionario de funciones*

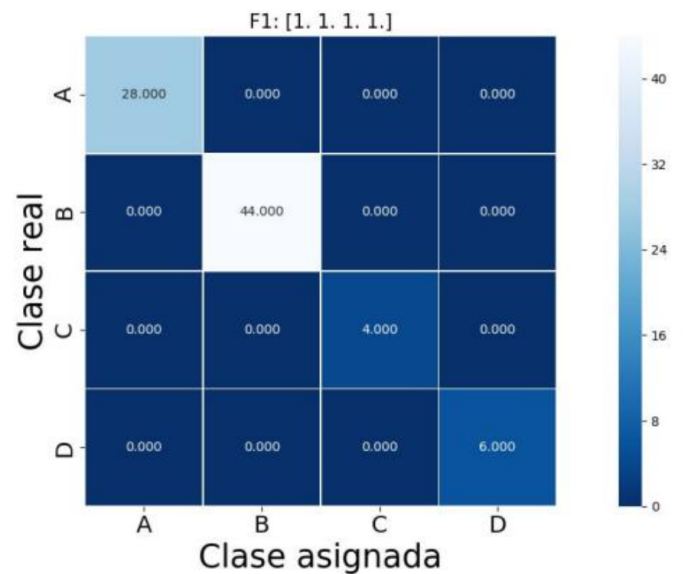
<b>Nombre</b>	<b>Descripción</b>	<b>Salida</b>
<i>trainingMatrixGen</i>	Función principal que construye las matrices de entrenamiento, según los requerimientos en la petición http hecha por el cliente, y hace los debidos consumos de microservicios	-Matriz entrenada -Ruta de formulario y programa
<i>autoEntrenarModelo</i>	Es la función que recibe la matriz de entrenamiento y otros requerimientos necesarios para la validación de formularios a entrenar, con indicaciones de variables a priorizar.	-Status -Message -Data: Ruta de almacenamiento del modelo
<i>autoProcesarData</i>	Limpia los datos que se hayan corrompido en el proceso de la petición y los ordena para su procesamiento.	-Matriz validada para la implementación del entrenamiento
<i>implementarDecisionTree</i>	Implementa el árbol de decisiones diseñado con	-Modelo de machine learning

	los parámetros definidos, tomando como entrada la matriz de entrenamiento y genera el modelo de machine learning por medio de la librería sklearn	
<i>procesarArchivosModelo</i>	Comprime el modelo generado mediante la librería pkl de python.	-Modelo comprimido en formato .pkl
<i>cargarArchivo</i>	Consume una API de Google Cloud Storage para almacenar el modelo comprimido y retorna a la función principal del algoritmo la ruta de almacenamiento	-Ruta de almacenamiento del modelo
<i>actulizarHeader</i>	Genera un registro en la base de datos de Neumomed para generar un historial de archivos.	-Status -Message

## 4.2. Evaluación de desempeño del algoritmo de machine learning

Ya entrenados los modelos, se implementaron distintos sets de validación con valores de salida ya conocidos, se implementó una matriz de confusión al modelo para determinar su exactitud y así evaluar posibles modificaciones a los parámetros para un reentrenamiento.

En la Figura 7, la matriz de confusión arrojó para el modelo seleccionado, altos valores de precisión, pues clasifica todos los datos en la categoría correspondiente, sin importar si el tamaño de la muestra es grande (muestra B) o pequeño (muestra C).



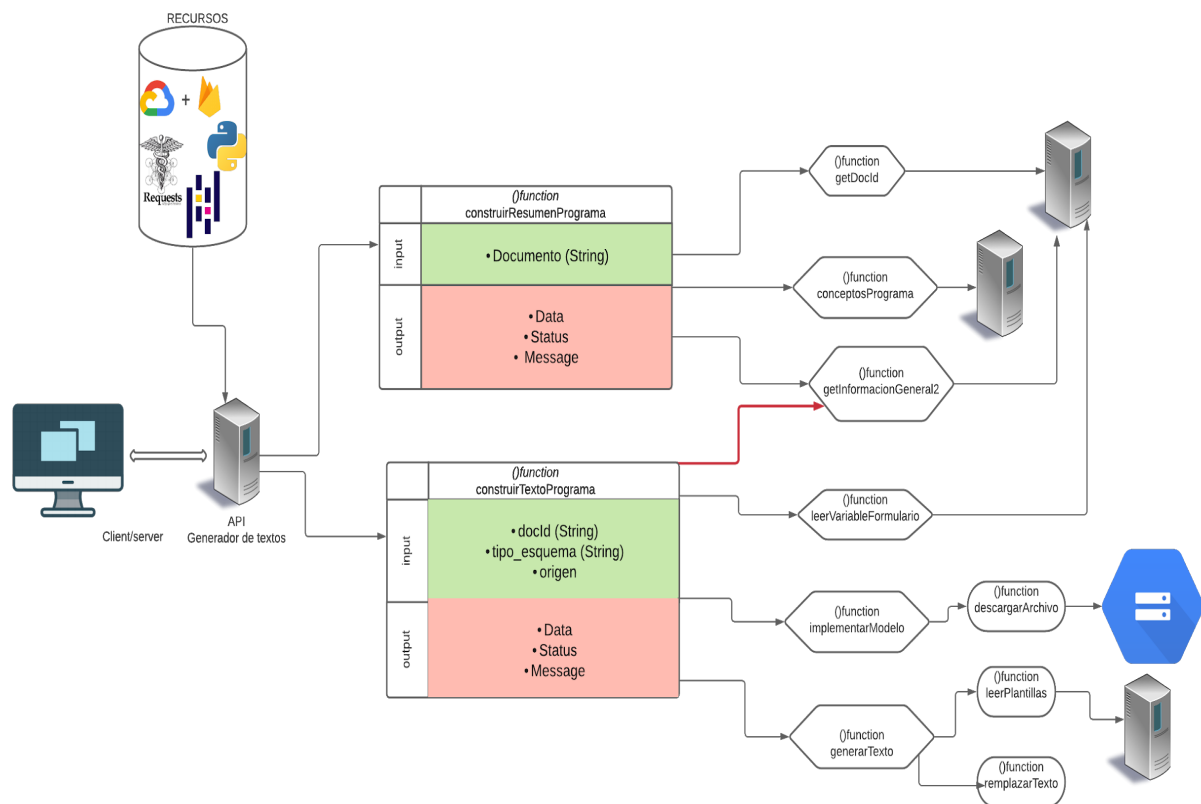
**Figura 7.** Matriz de confusión del modelo

Esto significa que el modelo está correctamente parametrizado y no es necesario reajustarlo ni re-entrenarlo al menos para este conjunto específico de datos y variables a manejar en esta consulta, cabe aclarar que una vez se diseñen nuevamente rutinas y nuevos formularios en un plan, se necesita nuevamente un entrenamiento que considere todas estas variables.

### 4.3. Implementación del modelo de machine learning en algoritmo de procesamiento de lenguaje natural

En la Figura 8 se ilustra el diagrama UML de ejecución del algoritmo que implementa el modelo entrenado, en este diagrama se especifica:

- El flujo de consumo de funciones en el algoritmo
- Las librerías empleadas
- Bases de datos implementadas
- Funciones principales del algoritmo, indicando parámetros de entrada y salida
- Funciones intermedias de consumo de otros servicios y nombre del servicio consumido



**Figura 8.** Diagrama UML de ejecución del algoritmo de implementación del modelo

En la Tabla 3 se presenta un diccionario con las funciones representadas en el diagrama.

*Tabla 3. Diccionario de funciones*

<b>Nombre</b>	<b>Descripción</b>	<b>Salida</b>
<i>construirResumenPrograma</i>	Esta función se encarga de generar un texto resumen introductorio del paciente, que incluye información general y el texto resumen del estado en el programa al que esté inscrito	Texto resumen general en el programa médico.
<i>getDocId</i>	Función para obtener el identificador del texto resumen de la última consulta.	docId
<i>conceptosPrograma</i>	Consulta el texto resumen de la última consulta del paciente en la I.P.S	Texto resumen última consulta
<i>getInformacionGeneral2</i>	Genera el texto resumen con información general	Texto resumen información general

<i>construirTextoPrograma</i>	Se encarga de generar el texto resumen del estado en el programa actual	Texto resumen en programa actual
<i>leerVariableFormulario</i>	Identifica las variables que se incluirán en el texto resumen	Vector de variables
<i>implementarModelo</i>	Implementa el modelo que se generó en el entrenamiento	-Status -Message -Data
<i>descargarArchivo</i>	Obtiene el archivo con el modelo desde el google cloud storage, para poderse implementar.	-Modelo en formato pkl
<i>generarTexto</i>	A partir de la implementación del modelo se hace el procesamiento del lenguaje natural para obtener el texto resumen	Texto resumen de programa actual

#### 4.4. Construcción automática de resúmenes de consultas implementado el algoritmo

En el momento de ejecución de una petición mediante Postman para simular un comando básico de cliente se obtuvo lo observado en la figura 9. Como respuesta se obtuvo una estructura de diccionario con los siguientes parámetros: status, message, data.

Haciendo una lectura de este, podemos evidenciar que el estado de la respuesta fue:

- Status: 200, lo que para el protocolo http indica 'ok', es decir que el algoritmo respondió correctamente a la petición.
- message: Este es el texto resumen de la consulta, indica que la información obtenida proviene del programa médico y consulta médica .
- data: texto, en el data se contiene la información objetiva del cliente, es decir sería lo que se visualice en pantalla, además de la información del programa o programas médico al que se encuentra inscrito.

```
{
  "data" : {
    "resumen": ["Paciente de sexo masculino, de 69 años de edad, con antecedentes de tabaquismo durante 38 años, consumia 2 cajetillas diarias, a lo largo del tratamiento ha presentado un índice de paquetes de 76, con riesgo alto e hipertension arterial por cerca de 20 años, controlada con Varsantan 160 mg, dosis 1 cada 24 horas. Actualmente se encuentra inscrito en los programas de EPOC, dejar de fumar. Se clasifica como GOLD 4 C actualmente. Ha presentado 4 exacerbaciones a lo largo de su estadía en el programa [2 años]. El último encuentro arrojó un índice en la prueba de EPWORTH de : 12 [+3], aumentando en +4 respecto a su ultima prueba y una fiebre objetiva de 38°C por 5 días. Su examen físico muestra las siguientes condiciones : Venturi 35% [+12], saturación de oxígeno de 92% [-4], frecuencia respiratoria 35 RPM [+9] " ,
    "programa" : ["dejar_de_fumar", "epoc"]
  },
  "message": "Mensaje resumen de programa",
  "status": 200
}
```

**Figura 9.** Consumo de la aplicación desde postman

#### **4.5. Evaluación manual de la calidad de los resúmenes generados y re-entrenamiento correctivo del sistema**

Los resultados generales de los criterios de evaluación médica, de ingeniería y comunicaciones fueron los siguientes:

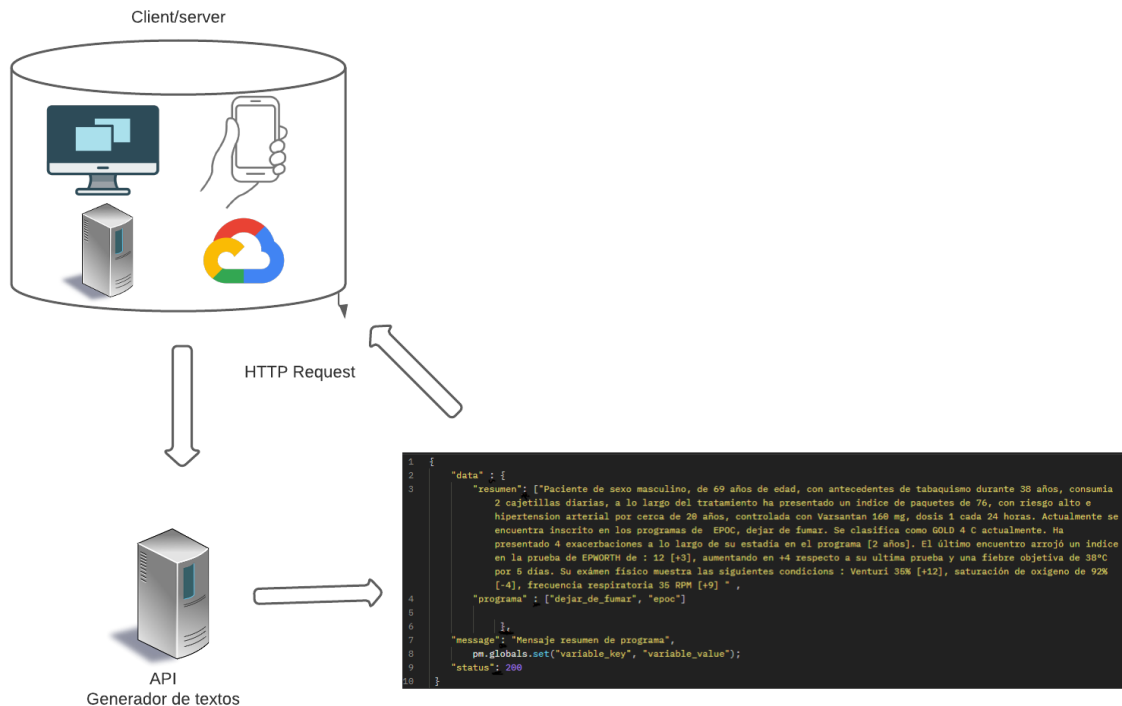
- Criterio de evaluación médica: cada texto introduce de manera acertada al contexto del paciente, sintomatología, antecedentes, medicamentos, patologías y diagnóstico, así como programa médico en el que se encuentra, y ciertas variables consideradas de importancia a las cuales se les hace una comparación con el encuentro inmediatamente anterior. No se evidencian fallas inminentes en su construcción
- Criterio de evaluación de ingeniería: los textos generados coinciden con lo esperado según las variables de los casos clínicos de los pacientes de prueba.
- Criterio de evaluación de comunicaciones: los textos presentan una estructura coherente, se detectaron cierta cantidad de errores en entonaciones y ciertos errores en el momento de tener más de un programa, pero se corrigieron estos errores.

#### **4.6. Implementación de arquitectura cliente- servidor**

En la Figura 10 se presenta el diagram UML de consumo del algoritmo en distintas plataformas, finalidad con la cual fue diseñado, este se creó como un servicio en el servidor y puede ser consumido por medio de peticiones HTTP desde cualquiera de los clientes existentes en la I.P.S que son:

- Aplicación web multiplataforma
- Servicios de interoperabilidad con Google
- Otros servicios internos de Neumomed





**Figura 10.** Diagrama UML arquitectura Cliente-Servidor

#### 4.7. Implementación de seguridad en el proyecto

Se mantuvo el anterior esquema de seguridad de Firebase para mantener protegidos los datos y reducir al mínimo el acceso a ellos, por medio de medidas de seguridad como:

- Restringir el acceso a una selección de usuarios que tienen un fin administrativo o clínico para acceder a los datos personales de los pacientes.
- Registrar el acceso de los usuarios a los sistemas que contienen información sensible.

- Firebase solo permite que accedan a los datos los usuarios que se encuentren autenticados.

En la Figura 11 se muestra la respuesta del algoritmo cuando se recibe una petición de un cliente no autenticado

```
1  {  
2  |   "status": 401,  
3  |   "message": "No es posible realizar este requerimiento"  
4  | }
```

**Figura 11.** Ejemplo de error de seguridad

#### **4.8. Consumo del algoritmo desde el frontend**

En la Figura 12 se puede visualizar el diseño actual de la interfaz gráfica disponible en la historia clínica digital. Se presenta al inicio de una exploración sencilla de formularios de información básica del paciente, seguido se presenta el texto resumen en la sección de “Programa respiratorio”, en él se especifican cierta cantidad de información esencial del paciente y medidas trazables del mismo a lo largo del número de consultas que haya hecho en el programa médico en el que se encuentra, una pequeña comparación de los valores en porcentaje de cambios de estas variables, al igual que el anterior trabajo en la IPS, se conservó la opción de poder editar estos textos para una mejor redacción según el criterio médico.

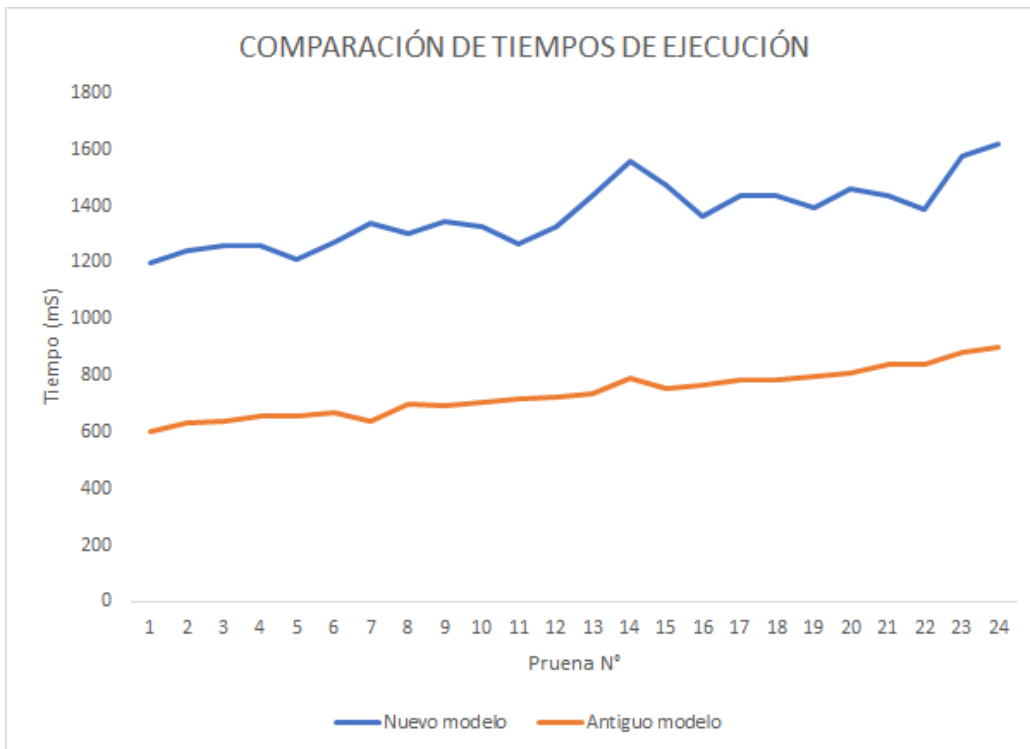


**Figura 12.** Interfaz gráfica de la historia clínica

#### 4.8. Tiempos de ejecución del algoritmo

En la Figura 13 se ilustra la comparación del tiempo de ejecución del modelo vs el tiempo de ejecución de otro algoritmo de generación de textos de la IPS. En este caso se especificó. Para esta prueba se usaron 24 casos clínicos que se seleccionaron en orden ascendente según la cantidad de variables que se llenaron en la consulta, es decir se muestra el comportamiento de ambos algoritmos cuando la cantidad de variables que deben incluirse aumenta.

Se puede apreciar claramente que el tiempo de ejecución del modelo es considerablemente mayor, siendo su máximo de 1570 ms, lo que significa una diferencia de 640 ms con el algoritmo existente lo que significa un aumento de aproximadamente 40.7% del tiempo de ejecución. Esto se debe principalmente al nuevo servicio llamado *trainingMatrixGen* el cual está automatizando todo un proceso que consistía en la creación manual de plantillas.



**Figura 13:** Tiempos de ejecución

## 5. CONCLUSIONES Y RECOMENDACIONES

Debido al crecimiento de la población con enfermedades respiratorias y además el contexto actual de pandemia en el que nos encontramos, el contar con un servicio de seguimiento en variables de interés a lo largo de un programa médico a un paciente es una herramienta que mejora notablemente la atención médica y permite una mejor visualización del progreso que este lleva. El algoritmo actual aunque aún se encuentra en un ambiente de pruebas, ha demostrado en sus fases de entrenamiento un avance en la capacidad de variables a procesar al aumentar la selectividad del datos de interés. Aunque el algoritmo se haya vuelto en procesamiento algo más complejo, el automatizar mucho del trabajo que antes era manual al momento de tratar las matrices de entrenamiento, siendo ahora proceso que toman entre 2-3 segundos y de ejecución única. Ahora, el proceso de revisión permitió que, comparando con la versión previa, se obtuvieron resultados similares en la calidad de los textos, ya que se siguieron los mismos requerimientos técnicos en ambos casos y la retroalimentación con el equipo de TI permitió reconocer y retomar correcciones previamente hechas. Respecto a la arquitectura cliente-servidor se comprobó que es adecuada para los algoritmos de procesamiento del lenguaje natural, pues permite que los textos generados sean consumidos desde múltiples clientes, tanto postman como cliente ficticio a ya un protocolo de comunicación establecido con la historia clínica de Neumomed, además esta arquitectura permite corregir errores en el algoritmo sin que signifique una refactorización completa del código debido a su estructura de microservicios. Y ya terminando, se propone como trabajo a futuro, además de las debidas repotenciaciones para poder implementar este servicio en la historia clínica, una posible integración con los servicios de análisis de sentimientos, el cual podría extraer de forma mucho más acertadas, las opiniones y percepciones de los propios pacientes y médicos en cómo se está llevando la terapia y funciona como una herramienta de apoyo en la priorización de una atención más oportuna.

## 6. REFERENCIAS

1. Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. *New England Journal of Medicine*, 380(14), 1347-1358.
2. McGhee, K. (2019). Machine Learning in Medicine. *Machine Learning*.
3. Ohno-Machado, L. (2011). Realizing the full potential of electronic health records: the role of natural language processing. *Journal of the American Medical Informatics Association*, 18(5), 539-539.
4. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260.
5. Cloud machine learning engine, [cloud.google.com](https://cloud.google.com/ml-engine/). (2019). Disponible en :<https://cloud.google.com/ml-engine/> [Accedido: 25- Feb- 2021].
6. Claesen, M., & De Moor, B. (2018). Hyperparameter Search in Machine Learning, 10–14. *arXiv preprint arXiv:1502.02127*.
7. DecisionTreeClassifier, documentacion libreria, 2018, <https://scikitlearn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html> [Accedido: 26- Feb- 2021]
8. Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural
9. Shung, K. P. (2018). Accuracy, precision, recall or F1. *Towards Data Science*.
10. Ruuska, S., Hämäläinen, W., Kajava, S., Mughal, M., Matilainen, P., & Mononen, J. (2018). Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle. *Behavioural processes*, 148, 56-62.

11. About Us, Documentacion scikitlearn, 2018 <https://scikitlearn.org/stable/index.html> [Accedido: 12- Ene- 2021]
12. Tensorflow, Google, 2018 <https://www.tensorflow.org/> [Accedido: 12- Ene- 2021]
13. García-Holgado, A., & García-Peñalvo, F. J. (2014, November). Architectural pattern for the definition of eLearning ecosystems based on Open Source developments. In *2014 International Symposium on Computers in Education (SIIE)* (pp. 93-98). IEEE.
14. Milosavljević, B., & Tešendić, D. (2010). Software architecture of distributed client/server library circulation system. *The Electronic Library*.
15. Flask, Documentacion Flask, 2020 <http://flask.pocoo.org/> [Accedido: 18- Marzo 2021]
16. McKinney, W. (2015). Pandas, python data analysis library. *see http://pandas.pydata.org.*