

# **TRABAJO FINAL DE MÁSTER**

## **IA EN LA GESTIÓN DE CITAS MÉDICAS: INNOVANDO EN CONFIRMACIÓN Y REDUCCIÓN DE AUSENCIAS.**

### **Descripción breve**

La utilización de inteligencia artificial (IA) en la gestión de citas médicas tiene como objetivo mejorar la asignación de estas, disminuyendo los “non show” con el objetivo de mejorar la eficiencia de los servicios de salud ambulatorios

**Grupo 3**  
**Hernando Acevedo Aguilar**  
**Michelle Alexandra Chicaiza Anrrango**  
**Luis Marcelo Ortiz Carinao**  
**Sergio Valdueza Lozano**

**Entrega Parcial 04 junio 2024**

# Resumen ejecutivo del proyecto

Uno de los principales problemas de los servicios de salud tanto de España como de Latinoamérica son los tiempos de espera para consultas médicas de especialidades. En España para acceder a una atención especializada, los pacientes esperan una media de 79 días para ser atendidos en primera consulta con un rango entre 22 y 107 días (1). Mientras que el sistema de salud chileno enfrenta un desafío crítico se alcanzaron 61.191 prestaciones de especialidad retrasadas al 31 de diciembre del 2022, donde cada prestación promediaba un retraso de 156.5 días (2).

Esto se explica por múltiples factores, tales como la brecha de especialistas (3), y la **inasistencia a consultas médicas, que en algunos centros alcanza hasta el 20% del total de usuarios (4) (5)**.

Debido a esta problemática es que planteamos el desafío de integrar herramientas de inteligencia artificial con el objetivo de mejorar la asignación de citas médicas, mediante la predicción de una probabilidad de inasistencia (“non show”) con herramientas de machine learning, y en base a esta probabilidad de inasistencia ofrecer estas horas a pacientes en listas de espera con el fin de disminuir los retrasos en las consultas médicas de especialidades.

Para este proyecto se plantean una serie de objetivos e hitos clave en su desarrollo:

- 1) Se obtiene una base de datos con información de cada usuario del servicio de salud y una etiqueta target: asistió o no a su cita (““asiste” vs “non show”).
- 2) Se realiza un análisis de los datos, donde se identifican y preparan variables predictoras de inasistencia.
- 3) Se preparan los datos para su posterior uso con herramientas de machine learning mediante estandarización, balanceo de clases y técnicas de reducción de dimensionalidad.
- 4) Se entrena nodelos de machine learning, en donde previa separación de un conjunto de datos de prueba se valorará la capacidad predictora de inasistencia (“non show”) de estos modelos
- 5) En base a la información obtenida del punto anterior se elaborará un modelo de asignación de cupos de “overbooking” con el fin de minimizar el problema de que un “non show” signifique una pérdida de horas médicas, pero a su vez que evite de que haya más de un paciente asignado para la misma hora médica, evitando así los retrasos en las consultas médicas. Vale decir corresponde a un problema de optimizar una función.
- 6) Se elaborará una interfaz de usuario tipo chat boot que permitiría la interacción con los usuarios finales de este sistema y se modelará el proceso de integración entre modelos de machine learning, interfaz de usuario y bases de datos.

Para la obtención de datos se recopilan fuentes de la empresa y se procede a realizar un análisis de bases de datos de fuentes externas que se encuentren con la variable objetivo: “asiste”, “non show” etiquetada.

# Índice

Resumen ejecutivo del proyecto

Introducción

Núcleo del proyecto

Bibliografía

# Introducción

- Breve introducción de la empresa: sector al que pertenece alcance geográfico y actividad desarrollada.

MediAgenda Solutions, S.L. (empresa ficticia para fines académicos), es una empresa líder en el sector de la salud, especializada en la administración inteligente de agendas médicas para hospitales y centros médicos en España y Latinoamérica. Nuestra actividad se centra en gestionar las agendas de visita a los especialistas, con la misión de optimizar la asignación de horarios médicos, reducir los tiempos de espera y mejorar la calidad de la atención médica.

- Definición del problema planteado

En España, para acceder a una atención especializada, los pacientes esperan una media de 79 días hasta ser atendidos en primera consulta, con un rango entre 22 y 107 días (1).

Mientras que en Latinoamérica se enfrenta un desafío de salud crítico, particularmente en Chile, donde pese a tener patologías priorizadas por ley con tiempos de espera máximo (garantías explícitas de salud GES), al 31 de diciembre del 2022 se alcanzaron 61.191 garantías retrasadas, con un promedio de 156.5 días de retraso por garantía (2). Estos retrasos se explican por múltiples factores, como la brecha de especialistas (3) y la **inasistencia a consultas médicas, que en algunos centros alcanza hasta el 20%** (4) (5).

**La falta de asistencia a citas programadas**, o "No Show", es un obstáculo recurrente que entorpece la eficiencia operativa, genera desequilibrios en la programación de los médicos y prolonga las listas de espera para ser atendido por un especialista.

- Objetivo del proyecto según el reto planteado

Para abordar esta problemática, proponemos integrar herramientas de inteligencia artificial para mejorar la asignación de citas médicas mediante la predicción de la probabilidad de inasistencia ("No Show") usando técnicas de machine learning. Con base en esta probabilidad, se optimizarán las agendas médicas mediante un sistema de overbooking, similar al utilizado en otros sectores (aviación, hoteles, etc.), con el fin de ofrecer estas horas a pacientes en listas de espera y reducir así los retrasos en las consultas de especialistas.

Además, desarrollaremos una interfaz de usuario que facilite la concertación y confirmación de citas de forma eficiente y amigable. Esta herramienta permitirá agendar citas, generar recordatorios y recopilar información útil para mejorar las predicciones de asistencia con nuestro modelo de IA, la integración de estos módulos se desarrollará en este proyecto solo a nivel teórico.

- Presentación resumida de las fases en las que va a contar el proyecto

Para alcanzar los objetivos planteados, el proyecto se desarrollará en varias fases, con una primera fase que constituye el núcleo fundamental del mismo y consta de varias

subfases. Inicialmente, se desarrollará un modelo de IA para predecir la asistencia a citas médicas, comenzando con la recopilación de datos y un análisis exploratorio para identificar variables clave y crear características significativas. Se aplicará el particionamiento de los datos siguiendo distintos criterios para entrenar y evaluar los modelos, empleando algoritmos como regresión logística, árbol de decisión y redes neuronales, realizando una optimización de los modelos con búsqueda exhaustiva de los mejores parámetros. Los modelos se clasificarán según su rendimiento, seleccionando los más efectivos en cada particionamiento para su implementación en la gestión de overbooking. El proceso descrito anteriormente se respalda mediante el uso de Python como lenguaje de programación principal, aprovechando sus entornos de desarrollo como Google Colab, Visual Studio Code y Jupyter Notebook de Anaconda.

En la segunda fase, se utilizarán las predicciones de asistencia para optimizar las agendas médicas mediante la creación de cupos de overbooking, maximizando el uso de recursos sin generar tiempos de espera excesivos ni sobrecargar los servicios médicos. En la tercera fase, se desarrollará un Asistente Virtual basado en Procesamiento de Lenguaje Natural (NLP) para gestionar citas y recordatorios, mejorando la experiencia del usuario y optimizando el proceso de programación y seguimiento de citas médicas. Finalmente, en la cuarta fase, se refinará el sistema a nivel teórico, integrando la base de datos con los modelos predictivos y de optimización, y ajustando el modelo de IA para mejorar la precisión de las predicciones y la optimización obtenida.

- Resumen del resultado final obtenido

#### Fase 1: Predicción de asistencia a citas médicas

Modelo de IA para Predicción de No Shows: Desarrollamos y optimizamos un modelo de inteligencia artificial capaz de predecir la asistencia o no asistencia a citas médicas con una precisión significativa. Utilizamos algoritmos de machine learning como la regresión logística, árboles de decisión y redes neuronales. A través de técnicas de feature engineering, creamos variables predictoras adicionales y mejoramos la calidad de los datos. La precisión del modelo se validó mediante técnicas de validación cruzada y se lograron métricas de desempeño robustas.

#### Fase 2: Implementación de un sistema de overbooking

Sistema de Overbooking: Basándonos en las predicciones de asistencia obtenidas en la Fase 1, diseñamos un sistema de overbooking que permite maximizar el uso de los recursos médicos disponibles sin generar tiempos de espera excesivos ni sobrecargar los servicios médicos. Este sistema calcula el grado óptimo de overbooking para cada slot de cita médica, teniendo en cuenta la incertidumbre en las predicciones y otros factores ambientales. La implementación de este sistema permite reducir a nivel teórico los tiempos de espera para los pacientes y mejora la utilización de los recursos médicos.

#### Fase 3: Creación de un asistente virtual basado en NLP

Asistente Virtual para Gestión de Citas: Desarrollamos un asistente virtual utilizando técnicas de Procesamiento de Lenguaje Natural (NLP) para la gestión eficiente de citas y recordatorios. Este asistente permite a los pacientes solicitar, modificar y cancelar citas médicas a través de plataformas de mensajería como WhatsApp.

#### Fase 4: Refinamiento e integración del sistema

Integración y Mejora Continua: A nivel teórico, desarrollamos un proceso de refinamiento continuo que integra la base de datos de la interfaz creada para gestionar las citas y los recordatorios con los modelos predictivo y de optimización. Este proceso permite la actualización constante del modelo de IA con nuevas características relevantes y mejora la precisión de las predicciones de asistencia. Además, la integración del sistema de gestión de citas con la interfaz de usuario facilita una experiencia más fluida y eficiente para los pacientes.

Resultados clave:

- **Reducción de Tiempos de Espera:** La implementación del sistema de overbooking, junto con las predicciones precisas del modelo de IA, permite una reducción en los tiempos de espera para la atención médica, mejorando la satisfacción de los pacientes.
- **Optimización de Recursos Médicos:** La utilización eficiente de los recursos médicos a través del sistema de overbooking y la mejora en la programación de citas contribuye a un uso más efectivo y racional de los mismos, generando ahorros económicos y operativos.

# Núcleo del proyecto

## Definición del proyecto y análisis de viabilidad

### Descripción detallada de la empresa

Se plantea la creación de MediAgenda Solutions, S.L. como empresa líder en el sector de la salud, especializada en la administración inteligente de agendas médicas para hospitales y centros médicos en España y Latinoamérica.

Nos enfrentamos al desafío crítico de los largos tiempos de espera en la atención médica, una problemática que impacta tanto a los pacientes, quienes experimentan retrasos significativos en su atención, como a los profesionales de la salud, que deben hacer frente a agendas sobrecargadas y recursos limitados.

Un ejemplo ilustrativo de esta problemática se encuentra en el barómetro sanitario realizado en España por el Ministerio de Sanidad, Consumo y Bienestar Social (6) en colaboración con el Centro de Investigaciones Sociológicas. Según los datos recogidos en el informe correspondiente al año 2023, el 27.2% de los ciudadanos reportaron haber esperado "11 días o más" desde que solicitaron la cita hasta que fueron atendidos por el médico de familia, evidenciando así los prolongados tiempos de espera que afectan a la población.

SÓLO A QUIENES EN LOS ÚLTIMOS DOCE MESES HAN TENIDO ALGUNA CONSULTA CON UN/A MÉDICO/A DE CABECERA O DE FAMILIA DE LA SANIDAD PÚBLICA Y PASÓ MÁS DE UN DÍA HASTA QUE LES ATENDIERON PORQUE NO HABÍA CITA
ANTES
(1 o 3 en P6 y 3 en P6b)
(N=1.388)

#### Pregunta 6b01

¿Cuántos días?

1 día	0,1
2 días	11,4
3 días	12,7
4 días	6,8
5 días	7,3
6 días	1,8
7 días	18,5
Entre 8 y 10 días	12,4
11 y más días	27,2
No recuerda	1,7
N.C.	0,1
(N)	(1.388)
Media (días)	9,48
Desviación típica	9,47
(N)	(1.363)

**Figura 1. Tiempo de espera en días para consulta a médico de familia.**

Con el objetivo de abordar este problema, hemos centrado nuestros esfuerzos en el desarrollo de soluciones innovadoras que optimicen la gestión de citas médicas mediante la aplicación de tecnologías avanzadas, especialmente inteligencia artificial (IA). Nuestros avanzados algoritmos de machine learning analizan una amplia gama de datos, desde históricos médicos hasta patrones de comportamiento previos, para predecir con

precisión la probabilidad de inasistencia de los pacientes a sus citas médicas. Esta información se utiliza para ajustar de manera inteligente las agendas médicas, maximizando así la eficiencia de los recursos médicos disponibles y reduciendo los tiempos de espera para los pacientes.

Además de los desafíos asociados con los largos tiempos de espera en la atención médica, otro problema significativo que enfrentan tanto pacientes como profesionales de la salud es el alto porcentaje de absentismo en las citas médicas. Este fenómeno no solo genera costos económicos para los sistemas de salud, sino que también puede afectar negativamente la calidad y eficiencia de la atención médica.

Conscientes de esta problemática, MediAgenda Solutions, S.L. desarrollará una solución innovadora que aborde directamente el problema del absentismo. Nuestra plataforma utiliza algoritmos avanzados para enviar recordatorios personalizados a los pacientes días previos o incluso el mismo día de su cita médica, solicitando una confirmación de asistencia. Esto no solo ayuda a reducir el porcentaje de inasistencias, sino que también permite una mejor planificación de los recursos médicos y una atención más eficiente para aquellos pacientes que sí asisten a sus citas programadas.

La eficacia de las innovaciones tecnológicas en el ámbito médico se ha demostrado en términos de ahorro de tiempo tanto para los pacientes como para los profesionales de la salud, lo que se traduce en una reducción de costos. Estas innovaciones no solo mejoran la organización de los horarios y la reducción de tiempos de espera, sino que también permiten una actualización inmediata de la información del paciente y una mayor flexibilidad en la programación de citas.

En resumen, la inversión en tecnología aplicada a la gestión de citas médicas es crucial para optimizar los procesos, aumentar la satisfacción de los usuarios y avanzar hacia sistemas de salud más accesibles y sostenibles.

La empresa tiene su sede principal en Barcelona, España, desde donde coordina sus operaciones y desarrollo tecnológico. Sin embargo, su alcance abarca a nivel nacional e internacional, con el objetivo de ofrecer soluciones a los problemas de tiempos de espera en los servicios de salud en países de habla hispana, especialmente en España y Latinoamérica.

## Análisis interno y externo

Con el propósito de obtener una visión detallada y completa de la posición y perspectivas de MediAgenda Solutions, S.L. en el sector de la salud, se realizará un análisis tanto de los factores internos como externos que influyen en la empresa.

### *Análisis Interno*

**Innovación Tecnológica:** MediAgenda Solutions, S.L. se destaca por fomentar una cultura de innovación y colaboración, y se mantiene actualizada en las últimas tendencias y desarrollos en el campo de la inteligencia artificial y la tecnología de la salud.

**Equipo Multidisciplinario:** La empresa cuenta con un equipo altamente calificado y multidisciplinario, conformado por profesionales en áreas como inteligencia artificial, medicina, gestión de proyectos, diseño de experiencia de usuario (UX/UI) y desarrollo de

software con el objetivo de desarrollar y ofrecer una solución innovadora y eficaz para abordar el problema de los largos tiempos de espera en las consultas médicas especializadas. Esta diversidad de talentos permite una visión integral en el desarrollo de sus productos y servicios.

**Alianzas Estratégicas:** MediAgenda Solutions, S.L. ha establecido alianzas estratégicas con instituciones médicas, centros de salud y organizaciones del sector para colaborar en la implementación y mejora continua de sus soluciones. Estas alianzas fortalecen su posición en el mercado y les permiten acceder a una base de clientes potenciales más amplia.

### *Análisis Externo*

**Demandas en Crecimiento:** La creciente demanda de soluciones para reducir los tiempos de espera en los servicios de salud representa una oportunidad clave para MediAgenda Solutions, S.L. La necesidad de optimizar la gestión de citas médicas es un problema extendido en el sector, tanto en España como en Latinoamérica, lo que brinda un mercado potencialmente amplio para sus servicios.

**Competencia:** Aunque MediAgenda Solutions, S.L. es líder en la integración de inteligencia artificial en la gestión de citas médicas, enfrenta competencia de otras empresas que ofrecen soluciones similares o alternativas tradicionales. La capacidad de innovación y diferenciación será crucial para mantener su posición en el mercado.

**Regulaciones y Normativas:** La empresa opera en un sector altamente regulado, sujeto a normativas específicas en materia de protección de datos, seguridad y calidad de servicios de salud. Cumplir con estas regulaciones es fundamental para ganar la confianza de clientes y usuarios finales, así como para mantener la reputación y credibilidad de la empresa.

## **Explicación detallada del proyecto**

Para la ejecución de este proyecto se plantean 4 fases o retos que nos ayuden a conseguir el objetivo planteado:

- **Fase 1: Realizar un modelo de IA capaz de predecir la Asistencia o No Asistencia de los pacientes a las citas médicas.** Esta fase es esencial ya que sienta las bases para el desarrollo de soluciones efectivas en las etapas posteriores del proyecto.

- **Elección, Recolección y Preparación de datos.** Se realiza una búsqueda de una base de datos que contenga información sobre las citas médicas programadas en un servicio de salud, así como datos relacionados con los usuarios que las solicitan. Se elige una base de datos que incluye una etiqueta que indique si el paciente asistió o no a su cita médica (show – non show), ya que esta información es esencial para el entrenamiento y la evaluación del modelo de inteligencia artificial.
- **Ánalisis Exploratorio de los Datos (EDA).** Se realiza un exhaustivo EDA para comprender la estructura, la distribución y las relaciones dentro del conjunto de datos. Durante este proceso, se identifican y preparan las variables

predictoras relevantes para la predicción de inasistencia a las citas médicas. Esto implica la limpieza de datos para tratar inconsistencias o valores faltantes, así como explorar las variables presentes en la base de datos en busca de relaciones, tendencias o anomalías. Este paso es crucial para garantizar la calidad y la confiabilidad de los datos utilizados en el entrenamiento del modelo.

- **Feature Engineering.** El proceso de Feature Engineering desempeña un papel fundamental en la preparación de los datos para la construcción de modelos de inteligencia artificial. Durante esta etapa, transformamos los datos brutos del dataset en características significativas que permiten a los modelos aprender patrones y realizar predicciones precisas. Este proceso se divide en varias etapas clave que incluyen la extracción de características, la transformación de tipos de datos, la creación de nuevas características, la selección de características relevantes, y la normalización y manejo de los datos faltantes.

Durante la **extracción de características**, se identifican datos relevantes del dataset original, como la información demográfica de los pacientes y datos relacionados con el historial médico y las visitas anteriores, los cuales son importantes para predecir la asistencia o no asistencia a las citas médicas.

La **transformación de tipos** de datos garantiza que los datos estén en un formato adecuado para su procesamiento por parte de los modelos de inteligencia artificial. Además, se crean nuevas características que pueden incluir información climática, ubicación geográfica, y variables relacionadas con el historial del paciente, entre otras.

La **selección de características** se basa en la evaluación de su correlación con la variable objetivo (No-show), asegurando que solo se utilicen aquellas que tengan un mayor impacto en la predicción.

Finalmente, se aplican **técnicas de normalización, reducción de dimensionalidad, manejo de variables categóricas y balanceo de clases** para garantizar la calidad y eficacia del modelo de inteligencia artificial.

Este proceso de Feature Engineering sienta las bases para la construcción de modelos predictivos precisos en la gestión de citas médicas, permitiendo una mejor comprensión de los factores que influyen en la asistencia o no asistencia a las citas médicas.

- **Entrenar diferentes modelos de machine learning.** Se entrenarán diferentes modelos de machine learning, aplicando diversas técnicas y estrategias para asegurar una predicción precisa de la asistencia de los pacientes a sus citas médicas. Para el entrenamiento de los modelos de machine learning, se seguirá un enfoque metodológico detallado:
  - **Selección del conjunto de prueba:** Dado que no hay un conjunto de datos de prueba separado, se reservará una parte del conjunto original. Este subconjunto mantendrá la distribución original y no incluirá la etiqueta de asistencia ("No-show"). Solo se considerarán las últimas

citas de pacientes con múltiples citas.

- **Hipótesis y segmentación de datos:** Durante la preparación de los datos, se considerarán varias hipótesis para crear conjuntos de datos específicos que mejoren la capacidad predictiva de los modelos:
  - Conjunto de datos completo.
  - Pacientes sin condiciones médicas.
  - Pacientes de edad entre los 5 y 30 años.
  - Pacientes con citas programadas para otro día.
  - Pacientes de barrios con centro médico.
- **Entrenamiento de modelos de machine learning:** Para cada uno de los conjuntos de datos generados, se entrenarán varios modelos de machine learning. Entre los modelos seleccionados se incluyen regresión logística, árboles de decisión y redes neuronales. Durante el entrenamiento, se realizarán búsquedas de hiperparámetros y validaciones cruzadas para optimizar el rendimiento de cada modelo.
- **Evaluación y optimización:** Con el objetivo de mejorar la precisión y la eficacia de los modelos, se realizará una evaluación exhaustiva de su desempeño. Esto incluirá la *optimización de la precisión* para minimizar el overbooking, es decir, para reducir al mínimo la situación en la que se asignan más citas de las que puede manejar el médico, evitando así la sobrecarga de pacientes en determinados horarios o días. Asimismo, se buscará *optimizar el recall* para minimizar los casos en los que los pacientes que se predijo que asistirían finalmente no lo hacen, lo que ayudará a reducir los huecos libres en las agendas médicas y a maximizar la eficiencia en la utilización de los recursos disponibles.
- **Fase 2:** La fase 2 consiste en **implementar un Sistema de Overbooking en la Programación de Citas** para maximizar el uso de los recursos de los centros hospitalarios, sin excederse en los costos provocados por la propia implementación del sistema de overbooking, es decir, aquellos costos relacionados con los tiempos de espera de los pacientes o de horas extra realizadas por el personal sanitario.
  - **Estudio de Bibliografía relacionada.** Existe una extensa bibliografía desarrollada en torno a los costes provocados por la infrautilización o sobreutilización de recursos médicos, así como los provocados por una reducción en la calidad del servicio debida a los tiempos de espera de los pacientes. Hay muchas formas de medir los costes, son muchas las variables que impactan en dichos costes, y decenas de variantes a la hora de definir los métodos de asignación de citas para tratar de reducirlos.
  - **Elección de las hipótesis para la resolución del problema y fórmula para el cálculo del Coste del Sistema a minimizar.** Una vez estudiada la bibliografía existente, se procede a determinar las hipótesis que acoten los posibles escenarios a desarrollar, y que se ajusten al dataset elegido en la Fase 1 para el Modelo Predictivo de Asistencia, pues se necesitan dichos datos y sus respectivas predicciones para generar el modelo ML optimizador de overbooking. Dichas hipótesis servirán para definir y delimitar diferentes

casos de estudio, así como para generar restricciones que se tendrán que aplicar al modelo ML para restringir sus grados de libertad a la hora de optimizar los slots de overbooking.

De forma paralela, se elige la fórmula para el cálculo del Coste del Sistema, función que se desea minimizar para optimizar la asignación de citas médicas, teniendo en cuenta toda la bibliografía encontrada al respecto, y también buscando compatibilidad con las características presentes en el dataset escogido en la Fase 1.

- **Cálculo del Coste del Sistema en métodos tradicionales de asignación de citas médicas.** Una vez escogidas hipótesis y fórmula, se procede a realizar el cálculo aritmético puro de los Costes del Sistema usando escenarios tradicionales de asignación de citas, sin sistema de optimización alguno, sin y con diferentes grados de overbooking.

Cada uno de estos escenarios se ejecutará con millares de variaciones aleatorias en la selección de citas programables, todas provenientes del conjunto de pruebas de la Fase 1. Esto para garantizar robustez en el cálculo medio del Coste del Sistema de Asignación de Citas para cada escenario. Concretamente se usará una partición del 80% de los datos del set de pruebas de la Fase 1 porque es el mismo que se utilizará para entrenar el modelo ML de optimización.

- **Desarrollo de modelo de ML para el cálculo óptimo de overbooking.** Pura aplicación de IA para desarrollar un modelo ML que sea capaz de asignar el mejor slot a cada cita que reciba para programar, en estricto orden de recepción, minimizando el Coste del Sistema calculado según la fórmula previamente escogida, y de acuerdo a las restricciones a imponer en las hipótesis previamente establecidas.

El modelo se entrenará con la partición del 80% de los datos del set de pruebas utilizado en el punto anterior para el cálculo del Coste del Sistema con métodos tradicionales, y se evaluará con el 20% restante de la partición no usada. Se escogerá el modelo en función del mínimo absoluto encontrado.

- **Comparativa de Costes en la optimización del overbooking.** Se compararán los Costes obtenidos en todos los escenarios comentados, detallando las conclusiones.
- **Fase 3: Creación de un Asistente Virtual basado en el Procesamiento de Lenguaje Natural (NLP) que permita el contacto entre Centros de Salud y Pacientes para la gestión de citas y recordatorios.**

Las ventajas de tener un sistema de asignación de citas automático incluyen la disminución de errores humanos, la rapidez en la asignación de citas, la versatilidad de acceso, la disponibilidad las 24 horas, una plataforma unificada de citas y una experiencia de servicio óptima.

Un asistente virtual permite agilizar y optimizar la gestión de citas médicas, elección de fecha y horario de la cita, recepción de alertas de citas programadas,

disponibilidad de citas online y la consulta de citas agendadas tanto por los pacientes como por los profesionales de la salud.

En esta fase, se desarrollará un Asistente Virtual que facilitará la comunicación entre los Centros de Salud y los pacientes para la gestión de citas y recordatorios. El objetivo principal es mejorar la experiencia del usuario y optimizar el proceso de programación y seguimiento de citas médicas. El asistente se basará en técnicas de Procesamiento de Lenguaje Natural (NLP) utilizando Large Language Models (LLM) para comprender las consultas y solicitudes de los pacientes y proporcionar respuestas relevantes y precisas, integrándose a una interfaz de usuario que, preliminarmente, permitirá una interacción “text to text”.

### **Objetivos del Asistente Virtual**

- Gestión de citas:
  - Permite a los pacientes solicitar, modificar y cancelar citas médicas a través de una interfaz de usuario: WhatsApp.
  - Proporciona información sobre la disponibilidad de citas y los horarios de los médicos.
  - Permite confirmar las citas programadas
- Proporcionar información relevante:
  - Brinda información general sobre los servicios y especialidades médicas disponibles en los Centros de Salud.
  - Responde a preguntas frecuentes relacionadas con el proceso de atención médica.

### **Alcance del Asistente Virtual**

El Asistente Virtual podrá manejar consultas relacionadas con la programación, modificación y cancelación de citas médicas:

- Responde a preguntas generales sobre los servicios médicos, horarios de atención y disponibilidad de especialistas.
- Ofrece asistencia en la navegación del proceso de atención médica, como los pasos a seguir para programar una cita.
- Recopila información básica de los pacientes, como nombre, fecha de nacimiento y motivo de la consulta, para facilitar el proceso de programación de citas.

### **Generación de Preguntas Relevantes (Prompt Engineering)**

- Se creará un conjunto de preguntas clave para obtener la información necesaria de los pacientes, como el motivo de la consulta, preferencias de fecha y hora, y datos personales básicos.

- Se crearán asistentes con prompt de sistema que permitirán la personalización y la contextualización, para generar respuestas claras, concisas y relevantes.
- Se desarrollará un sistema de funciones en el asistente específicas para cada etapa del proceso de programación de citas y consulta de citas, incluyendo el agendamiento, confirmación y cancelación de citas.
- Se implementarán mecanismos de manejo de errores y aclaraciones para garantizar la comprensión adecuada de las respuestas de los pacientes.

### **Integración del Asistente en la plataforma de mensajería (WhatsApp)**

- Se utilizará la API de WhatsApp Business para integrar el Asistente Virtual en la plataforma de mensajería.
- Se desarrollarán funciones para recibir y procesar los mensajes entrantes de los pacientes.
- Se implementarán mecanismos de respuesta automatizada para proporcionar información y asistencia de manera oportuna.
- Se manejarán los flujos de conversación y las interacciones con los pacientes de manera eficiente y escalable.

La implementación del Asistente Virtual se realizará utilizando la API de OpenAI, aprovechando sus capacidades de procesamiento de lenguaje natural para crear de forma efectiva y precisa respuestas coherentes y relevantes para las consultas de los pacientes. Después de revisar diversas plataformas de chatbots como Dialogflow, Amazon Lex, Microsoft Bot Framework y developers OpenAI, se decide utilizar la API de OpenAI debido a su avanzada capacidad para generar respuestas naturales y precisas, así como su flexibilidad y facilidad de integración. Se emplearán instrucciones de sistema para establecer el contexto y el comportamiento general del asistente. Además, se utilizarán funciones específicas para tareas como la programación de citas y la proporción de información. También se aplicarán técnicas de retrieval para acceder a información relevante de la base de datos de citas médicas.

La integración con la base de datos permitirá al Asistente Virtual acceder a información en tiempo real sobre la disponibilidad de citas, los horarios de los médicos y los detalles de las especialidades médicas. Esto garantizará que las respuestas proporcionadas a los pacientes sean precisas y actualizadas, como se puede ver en la figura 2:

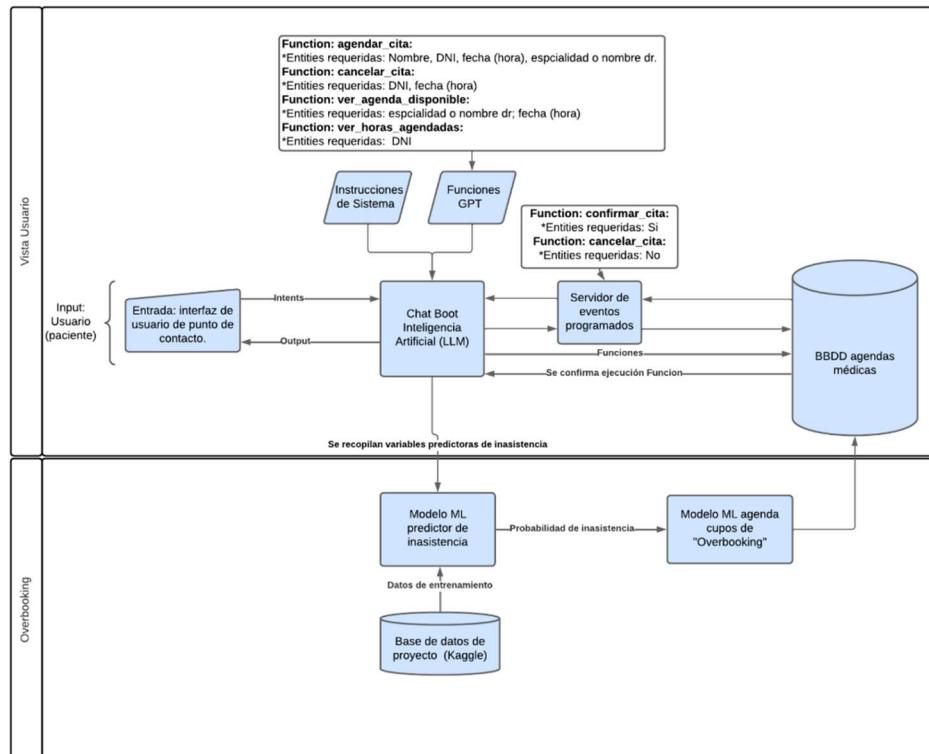


Figura 2. Diagrama de Flujo propuesta de Asistente Virtual.



Figura 3 Diagrama de flujo inexistencia de datos de identificación



Figura 4 Diagrama de flujo Asignación exitosa de cita



Figura 5 Diagrama de flujo Cancelación de cita

- **Fase 4:** Desarrollar, a nivel teórico, un proceso de refinamiento del sistema que integre la base de datos de la interfaz creada para gestionar las citas y los recordatorios a los pacientes (Fase 3) con los modelos predictivo y de optimización generados en las Fases 1 y 2.
  - Selección de posibles nuevas características relevantes provenientes de la aplicación.
  - Adhesión de estas nuevas características predictivas al dataset definitivo elaborado en la Fase 1.
  - Desarrollo fine-tuning del modelo de IA seleccionado en la Fase 1, con el objetivo de mejorar en precisión las predicciones de asistencia.
  - Integrar los nuevos resultados en modelo de IA creado en la Fase 2, analizando la optimización / disminución de “coste” obtenida.

- Fusionar este análisis de probabilidad de inasistencia y “overbooking” con una base de datos integrada a una interfaz de usuario.

## Fijación de los objetivos generales y específicos del proyecto

El proyecto, titulado "Aplicación de Inteligencia Artificial en la Gestión de Citas Médicas: Optimización y Reducción de Ausencias", tiene como objetivo general **mejorar la eficiencia y reducir los costos asociados con la gestión de citas médicas** mediante el uso de técnicas de Inteligencia Artificial (IA).

Para alcanzar el objetivo general mencionado anteriormente, se plantean los siguientes objetivos específicos:

**Desarrollo de modelos predictivos de ausencias:** Investigar y desarrollar modelos de IA capaces de predecir la no asistencia a las citas médicas con alta precisión. Estos modelos se basarán en datos específicos de pacientes y variables relevantes, buscando minimizar los falsos positivos y negativos.

**Comparación de algoritmos de aprendizaje automático:** Evaluar y comparar la eficacia de diferentes algoritmos de aprendizaje automático en la predicción del absentismo. Se optimizará el rendimiento de los algoritmos para adaptarse a las características específicas de los datos médicos, con el objetivo de seleccionar el más adecuado para el propósito del proyecto.

**Optimización de la métrica F1-Score:** Obtener la mejor F1-Score para el modelo de predicción, buscando un balance entre precisión y recall. Esto implica minimizar los falsos positivos (predicciones incorrectas de que un paciente no asistirá a una cita cuando sí lo hace) y los falsos negativos (predicciones incorrectas de que un paciente asistirá a una cita cuando no lo hace). Reducir estos errores mejora la eficiencia del modelo en la gestión de citas médicas y disminuye los costos asociados.

**Creación de un Asistente Virtual para gestión de citas:** Generar un Asistente Virtual con comunicación efectiva. Este asistente asignará citas con precisión, enviará recordatorios y facilitará la gestión a través de WhatsApp. También notificará cualquier cambio en las citas programadas.

**Entrenamiento de objetivos secundarios para el Asistente Virtual (*a nivel teórico*):** Se entrenará al Asistente Virtual para ofrecer información médica general, responder preguntas frecuentes sobre el proceso de atención médica, documentación requerida y protocolos de seguridad, así como proporcionar orientación sobre síntomas y medidas preventivas. Estas funciones, consideradas a nivel teórico, complementarán su tarea principal y mejorarán la experiencia del usuario.

**Desarrollo de un proceso de mejora continua:** Establecer un proceso teórico de mejora continua, donde el modelo predictivo y la información extraída por el Asistente Virtual se retroalimenten periódicamente. El objetivo es mejorar los resultados predictivos del modelo y reducir los costos asociados a inasistencias en la gestión de citas médicas.

# **Planificación**

## **Estimación de recursos económicos**

Para poder realizar esta estimación se espera a las decisiones tomadas por las directivas, posterior al estudio de la prefactibilidad.

## **Estimación de recursos materiales**

Para estimar los recursos materiales del asistente virtual se deben tener en cuenta los siguientes recursos:

- Un servidor o infraestructura en la nube, para alojar y ejecutar el programa del asistente virtual que se decida usar; debe tener capacidad suficiente para procesar y almacenar la data. Y una amplia capacidad de procesar un alto volumen de citas.
- Dispositivos de acceso como computadores - terminales, tablets o teléfonos inteligentes.
- Línea de acceso a Internet, ¿canal dedicado?
- Software licenciado para el programa - Incluye sistema operativo, Python, Visual Studio, u otras soluciones más robustas, que se decida adoptar; librerías y herramientas para los procesamientos.
- Otro servidor o infraestructura en la nube, para alojar y ejecutar los programas de prueba o nuevos desarrollos.
- Un sistema de gestión para almacenamiento de las bases de datos, y de copias de respaldo.
- El espacio físico para alojar los servidores y las personas responsables del proyecto.

## **Estimación de recursos humanos**

Las personas que harán parte del proyecto de asistente virtual se pueden dividir en dos momentos, planeación, desarrollo e implementación del proyecto y personal de funcionamiento y mantenimiento del asistente virtual.

- La primera parte debe contar con al menos un desarrollador de software.

Para todo el proyecto se necesitan personas de coordinación y soporte, como son las que siguen:

- Gerente del proyecto.
- Especialistas de seguridad de datos.
- Personal de soporte para hardware y software.

- Al menos un experto en IA, inicialmente puede ser medio tiempo.

De otro lado se debe tener en cuenta los implicados directamente con el proyecto que deben tener representación en todo el desarrollo del proyecto, al menos un médico, personal administrativo, personal de atención al cliente.

## Estimación de los recursos tiempo

Se ha estimado que para el estudio de prefactibilidad un mes.

En cuanto hace referencia a la planeación, desarrollo e implementación del proyecto: un mes y medio.

Para la parte de pruebas y ajustes un mes.

Lo anterior está condicionado a la cantidad de personas que puedan contratarse para este proyecto.

## Elaboración del cronograma del proyecto

Entrega	Fecha de inicio	Fecha de término	Contenido relevante
Entrega 1	23 Feb 2024	9 Abr 2024	Planteamiento del problema, posibles soluciones, análisis preliminar de bases de datos.
Entrega 2	19 May 2024	4 Jun 2024	Modelos de predicción de non show, algoritmo para optimización de agendas médicas, planteamiento de modelo de chatbot.
Entrega 3	5 Jun 2024	2 Ago 2024	Desarrollo final de modelos de predicción, optimización de algoritmos, desarrollo preliminar de chatbot.
Entrega final preliminar	19 Ago 2024	16 Sep 2024	Documento final del TFM, unificación de trabajo, conclusiones finales, Resumen Ejecutivo.
Entrega final	1 Oct 2024		Entrega final del proyecto.

A continuación, se detalla cada entrega para ofrecer información más específica sobre el trabajo realizado en cada una.

### Entrega 1

Fecha de inicio: 23 Feb 2024

Fecha de término: 9 Abr 2024

#### 1. Búsqueda de base de datos de non show de pacientes

##### a. *Planteamiento del problema y posibles soluciones:*

- Identificar y analizar las causas de no asistencia a citas médicas.
- Investigar posibles soluciones para mejorar la asistencia.

##### b. *Ánalysis preliminar de las bases de datos:*

- Recopilar y analizar datos sobre no asistencia a citas médicas.
- Identificar tendencias y patrones en los datos.

## Entrega 2

Fecha de inicio: 19 May 2024

Fecha de término: 4 Jun 2024

### 1. Desarrollo de modelos de predicción de non show (Fase 1):

- a. *Finalización del análisis exploratorio de datos (EDA)*: Analizar y visualizar los datos para comprender mejor su estructura y distribución.
- b. *Planteamiento de hipótesis con particionamiento del dataset*: Dividir el conjunto de datos en varios subconjuntos, cada uno representando una hipótesis subyacente sobre la predicción de asistencias a citas médicas.
- c. *Optimización de modelos de machine learning para predicción de inasistencia de pacientes*.
  - i. Búsqueda de mejores resultados de particiones: Evaluar y comparar diferentes particiones del dataset.
  - ii. Búsqueda de mejores modelos de machine learning y mejor red neuronal: Evaluar y comparar diferentes modelos de machine learning y redes neuronales.
  - iii. Optimización de hiperparámetros de modelos de machine learning: Ajustar los parámetros de los modelos para mejorar su rendimiento.
  - iv. Conclusiones iniciales comparando nuestros resultados con las bibliografías revisadas: Comparar los resultados obtenidos con los reportados en la literatura.

### 2. Planteamiento de algoritmo para optimización de agendas médicas según resultados de fase 1 (Fase 2):

- a. *Finalizar búsqueda bibliográfica sobre el tema*: Investigar y recopilar información sobre algoritmos de optimización de agendas médicas.
- b. *Selección de hipótesis y función de coste*: Determinar las principales hipótesis que delimiten el problema a resolver, así como la función de coste a calcular.

### 3. Planteamiento de modelo de chatbot de interacción con pacientes (Fase 3):

- a. *Iniciar búsqueda bibliográfica sobre el tema*: Investigar y recopilar información sobre chatbots médicos.
- b. *Definir el framework con el que elaborar el ChatBot*: Seleccionar un framework para desarrollar el chatbot.
- c. *Definir modalidad de interacción del ChatBot*. *De momento Text-to-Text*

## Entrega 3

Fecha de inicio: 4 Jun 2024

Fecha de término: 2 Ago 2024

### 1. Desarrollo final de modelos de predicción de non show (Fase 1):

- a. *Optimización de modelos de machine learning para cada grupo de usuarios*: Ajustar los modelos para cada grupo de usuarios.
- b. *Conclusiones de nuestros resultados con las bibliografías revisadas*: Comparar los resultados obtenidos con los reportados en la literatura.

### 2. Optimización de algoritmos para optimización de agendas médicas según resultados de fase 1 (Fase 2):

- a. *Analizar algoritmos tradicionales matemáticos puros*: Estudiar, comparar y calcular diferentes algoritmos matemáticos para optimizar agendas médicas.
  - b. *Desarrollar algoritmo de ML que optimice una función de reducción de coste*: Crear un algoritmo que minimice los costos de no asistencia.
  - c. *Comparar costes de algoritmos*: Evaluar y comparar los resultados de los algoritmos.
  - d. *Generar conclusiones*: Presentar los resultados y conclusiones sobre la optimización de agendas médicas.
3. **Desarrollo preliminar de ChatBot de interacción con pacientes (Fase 3):**
    - a. *Crear/inventar bases de datos*: Crear bases de datos de médicos por especialidad y ubicación, y agendas médicas con información de médico, especialidad, ubicación, fecha, hora y slot (libre o Id. Paciente).
    - b. *Crear funciones de búsqueda, lectura y escritura a dichas bases de datos*: Implementar funciones para interactuar con las bases de datos.
    - c. *Definir y crear flujos de conversación ChatBot - Pacientes*: Establecer y crear flujos de conversación para el ChatBot.
    - d. *Capacidades de asistencia médica*: Implementar funciones para proporcionar información básica sobre síntomas, condiciones y tratamientos. Además, hacer preguntas de seguimiento para entender mejor las necesidades del paciente. Capacidad de derivar al paciente a un profesional médico cuando sea necesario
    - e. *Probar ChatBot, y refinar*: Evaluar y mejorar el ChatBot.
    - f. *Generar conclusiones*: Presentar los resultados y conclusiones sobre el ChatBot.
  4. **Se realiza un planteamiento de una integración de los 3 puntos anteriores (Fase 4).**

#### **Entrega final preliminar**

Fecha de inicio: 19 Ago 2024

Fecha de término: 16 Sep 2024

1. **Elaboración y compilación del documento final del TFM:**
  - a. *Unificación de trabajo de las 3 fases*: Integrar los resultados de las tres fases.
  - b. *Conclusiones finales y desarrollo del Resumen Ejecutivo*.
  - c. *Revisión final y cita de bibliografías*.

#### **Entrega final:**

Fecha inicio: 17 Sep 2024

Fecha de término: 1 Oct 2024

1. **Desarrollo de presentación:**
  - a. Crear una presentación en PowerPoint que resuma los hallazgos y conclusiones del proyecto.
  - b. Incluir gráficos, visualizaciones y otros elementos para comunicar de manera efectiva los resultados.
  - c. Revisar la presentación para su entrega final.

## **Definición del alcance del proyecto**

Una vez determinada la necesidad de realizar el presente proyecto por la institución se ha procedido a realizar las actividades descritas. Y continuamos describiendo el alcance del asistente virtual.

**Recolección de datos**, se obtuvo una base de datos sobre la que se realizará el desarrollo del presente asistente virtual. En el que se están desarrollando actividades de limpieza de datos, errores y manejo de datos atípico.

**Integración con el sistema de gestión de citas**, que se va a ir realizando a medida que avanza el proyecto, que incluye la implementación de una interfaz de programación de aplicaciones (API - Application Programming Interfaces).

**Desarrollo del modelo del asistente virtual**, núcleo del presente trabajo que se ha venido diseñando y adaptando desde el inicio de este proyecto.

**Validación y pruebas**, a medida que se va avanzando con el proyecto, se evaluará el rendimiento del asistente virtual, usando datos de prueba y validación; pruebas de carga para verificar la escalabilidad y rendimiento, bajo diferentes condiciones de carga.

**Despliegue e implementación**, se realizará capacitación al personal involucrado en la asignación de citas, asistentes de atención al cliente, personas del área administrativa, médicos y personal de enfermería. Se realizará supervisión y soporte directo por un periodo de tiempo por definir.

**Entregables del proyecto**, se plantea realizar un documento de los requerimientos del sistema de asistencia virtual, el conjunto de los datos preprocesados, el modelo del asistente virtual entrenado y desplegado. Informe de las pruebas y validación, sistema de asistente virtual implementado y en funcionamiento, así como la documentación de soporte y capacitación.

# Desarrollo del proyecto

## Fase 1. Predicción de Asistencia a Citas Médicas

### Elección, Recolección y Preparación de datos

Un buen modelo predictivo de asistencia a citas médicas requiere un set de datos con las siguientes características:

- Presencia de Variable Target indicando la Asistencia o No Asistencia a la cita médica.
- Presencia de Variables Predictoras diversas y heterogéneas con las que poder predecir si ocurrirá asistencia o no, contra más variables predictoras mejor.
- Gran número de muestras, necesarias para poder generar un modelo predictivo robusto y confiable.

Se hizo una búsqueda por internet y se escogió el siguiente dataset de Kaggle: “**Medical Appointment No Shows**” ([7](#)), publicado por Jonihoppen.

Dicho Dataset contiene 110.527 muestras (citas médicas) de 62.299 pacientes distintos, de los que se recogen hasta 13 características de las citas y pacientes mencionados, así como la imprescindible característica objetivo de asistencia o no la propia cita solicitada.

Se encontraron otros conjuntos de datos que fueron descartados por contener menor número de muestras, o peores características predictoras, o simplemente por no disponer de la variable objetivo de Show – NoShow.

Escogido el dataset, y antes incluso de realizar el pertinente Análisis Exploratorio de los Datos (EDA), revisamos la nomenclatura de las columnas para corregir errores topográficos en las mismas: Hypertension por Hipertension, Handicap por Handcap, y NoShow por No-show.

### Análisis Exploratorio de Datos (EDA)

La primera actividad de peso en el desarrollo de cualquier tarea de ML es realizar un buen Análisis Exploratorio de Datos para extraer toda la información posible de los mismos, y así diseñar la mejor estrategia de datos que le sirva posteriormente al modelo predictivo.

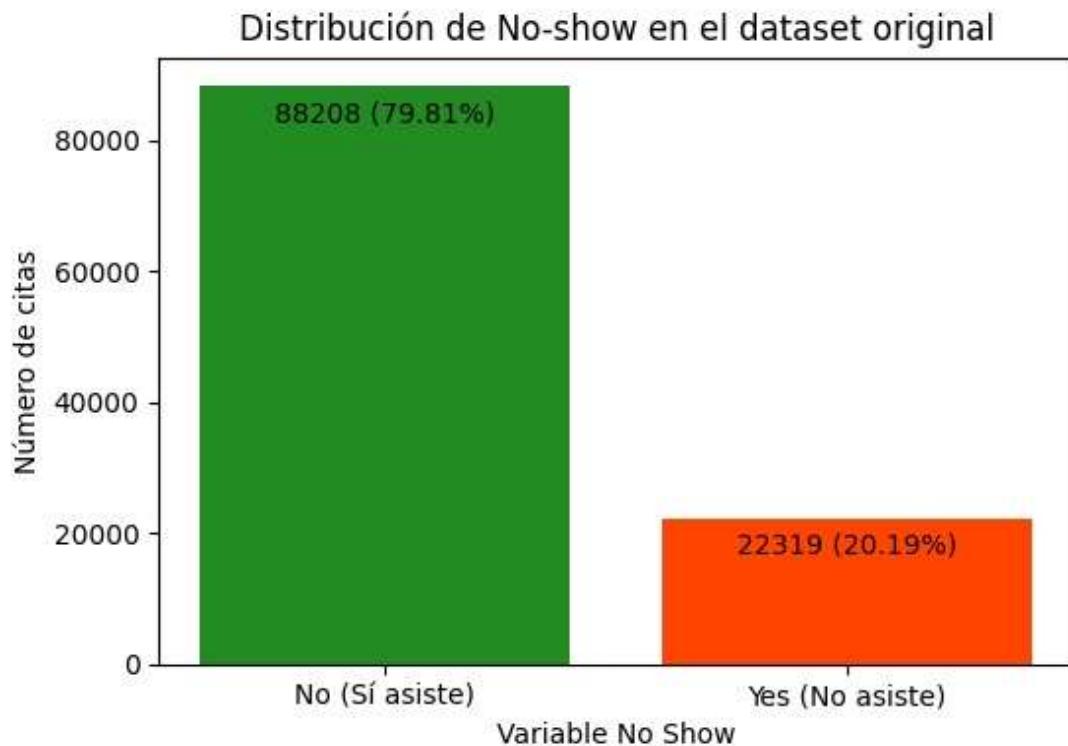
Primero se analiza la información general del dataset: tipos de datos (los cuales se modifican para poder trabajar la información que contiene cada variable), valores únicos, valores nulos, etc.

A continuación, se empieza analizando la variable target u objetivo, y posteriormente se seguirá con el resto de las 13 variables predictoras.

#### Variable Target “NoShow”

Sólo tiene 2 valores posibles: 0 (False) = Show o 1 (True) = No Show.

La distribución de valores indica que, como era de esperar, el dataset está fuertemente desbalanceado, revelando un 20.19% de citas sin asistencia.



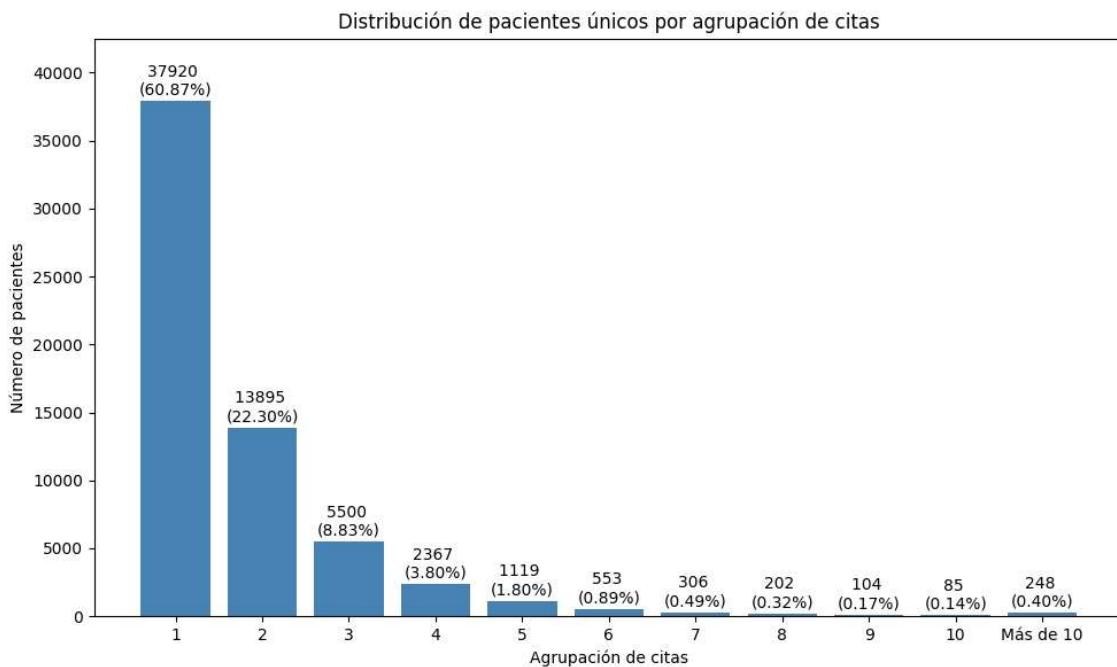
**Antes de entrenar el modelo se debe corregir este desbalanceo.**

**Variable Predictora: “PatientId”**

El dataset consta con información de 62.299 pacientes distintos, identificados con números enteros que van desde el 39217 hasta el 999981631772427.

Esto significa que el dataset contiene datos históricos de pacientes que realizan más de una solicitud de cita médica. Esta información puede ser muy relevante a la hora de determinar la probabilidad de asistencia de un paciente, pues en muchos casos ya se tendrá información relevante de su comportamiento en citas previas.

Aun así, tal y como se observa en la siguiente tabla, para el 60.87% de los pacientes sólo se tiene información de una única cita.



A simple vista no se observa ninguna relación o información de relevancia en el número de identificación del paciente, dato que nos confirmará posteriormente el cálculo de la matriz de correlación entre variables.

#### Variable Predictora: “AppointmentID”

Los valores únicos de esta variable son el total de muestras del dataset: 110.527, identificados con números enteros desde el 5030230 al 5790484.

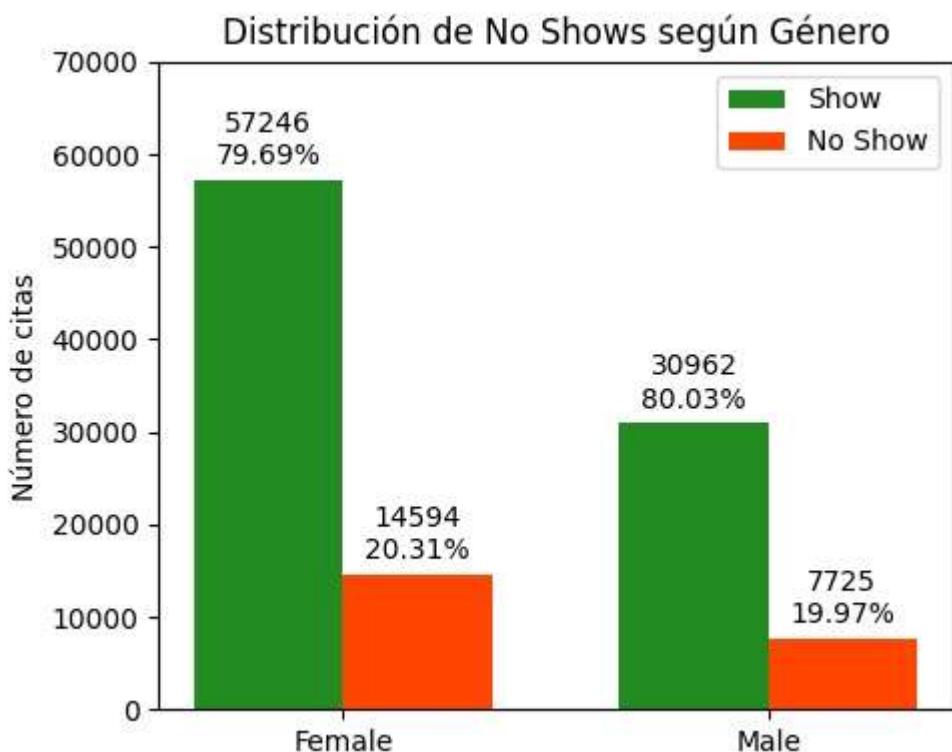
Tampoco parece haber mayor correlación numérica salvo con la asignación temporal, es decir, que es una identificación numérica correlativa en función de la fecha en la que se solicita la cita. La posible información de utilidad contenida en esa correlación temporal ya está contenida en otras variables más directas, como veremos a continuación.

#### Variable Predictora: “Gender”

Sólo se toman en cuante dos identificaciones de género: Female and Male, distribuidos en un 65% y 35% de los datos, respectivamente, por lo que esta variable está ligeramente desbalanceada, pero no se considera significativo.

Una de las comprobaciones importantes de esta variable, para garantizar coherencia en los datos, es que cada paciente tenga una identificación única e inequívoca del género asignado en cada una de sus correspondientes citas, y así es.

Respecto a la distribución de asistencia según género, las mujeres tienen una proporción mayor de No Shows (20.31%) respecto a los hombres (19.97%), pero la diferencia es mínima.

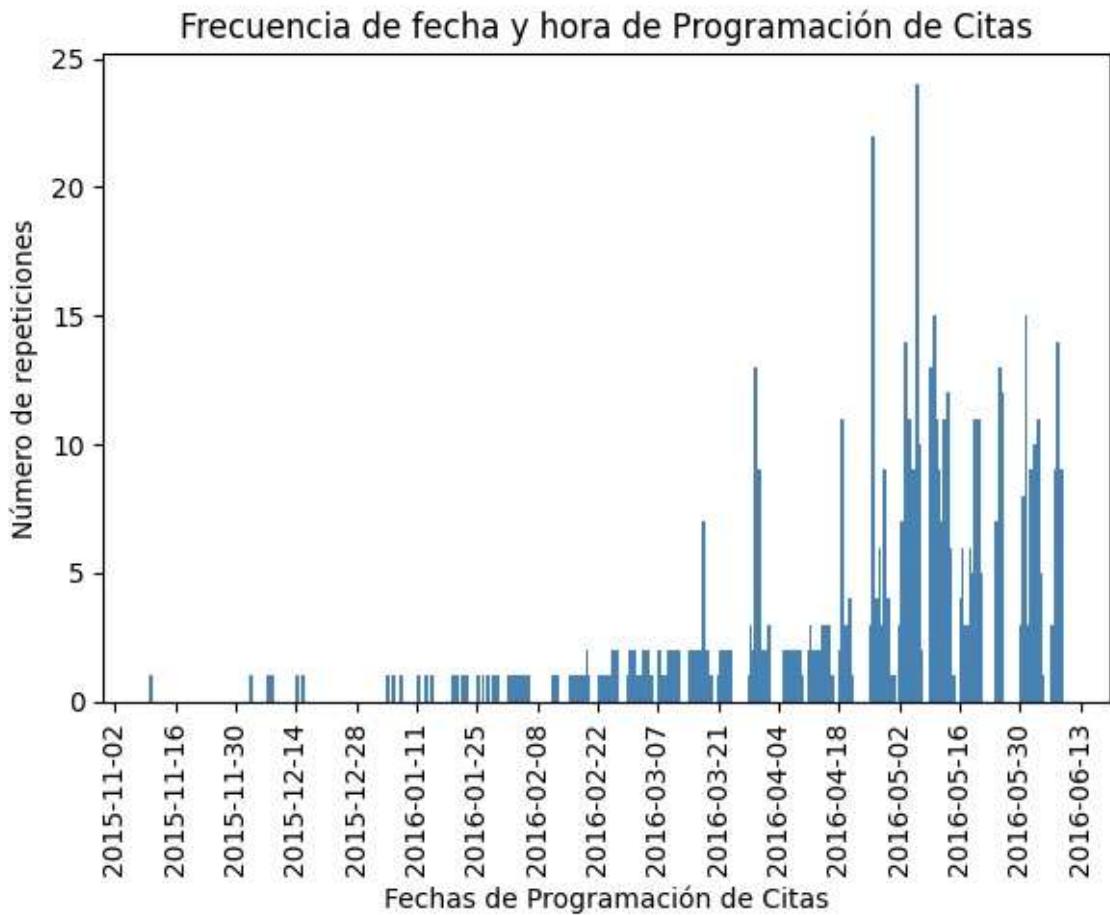


**Variable Predictora: “ScheduledDay”**

Esta variable muestra la fecha (día y hora) en la que se solicitó (**y programó**) la cita médica. Existen 103.549 fechas con día y hora distintos, por lo que hay días en los que se han solicitado citas exactamente en el mismo segundo (*pocas, pero las hay*).

El primer día que se solicitó una cita médica dentro de este conjunto de datos es el 10 de noviembre de 2015, a las 07:13:56, y la última el 8 de junio de 2016 a las 20:07:23.

El siguiente gráfico muestra como el grueso de fechas de solicitud de citas se concentra en los meses de abril, mayo y junio.

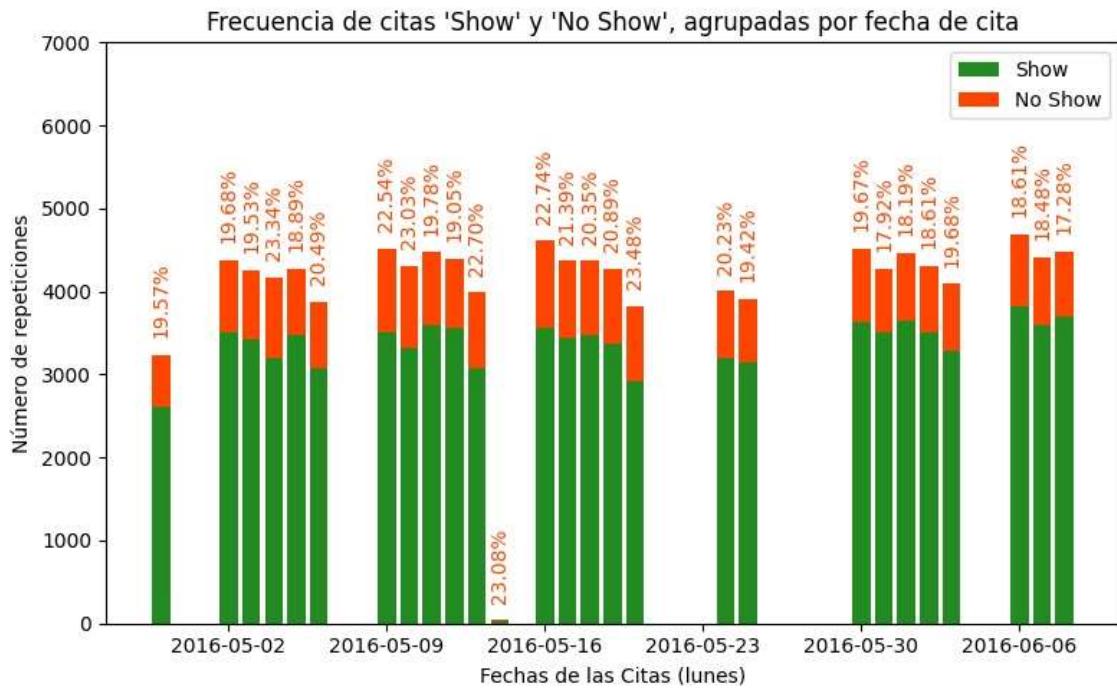


#### **Variable Predictora: “AppointmentDay”**

A diferencia de “ScheduledDay”, esta variable sólo guarda información del día en que se tiene la cita médica, sin la hora a la que está convocado cada paciente. Dentro de este dataset, existen 27 días distintos en los que se atienden consultas médicas, desde el 29 de abril al 8 de junio de 2016.

El hecho de contar con 2 variables Datetime relacionadas con la fecha en la que se solicita la consulta y la fecha en la que se atiende, nos permite estudiar posibles dependencias entre el lapso existente entre ambas fechas y la propia asistencia, característica que se estudiará durante el Feature Engineering.

También se prevé estudiar la posible correlación entre las asistencias y el día semanal en el que se atienden, pero tal y como se muestra en el siguiente gráfico de barras, las variaciones de asistencia según el día de la semana tampoco son exageradas.

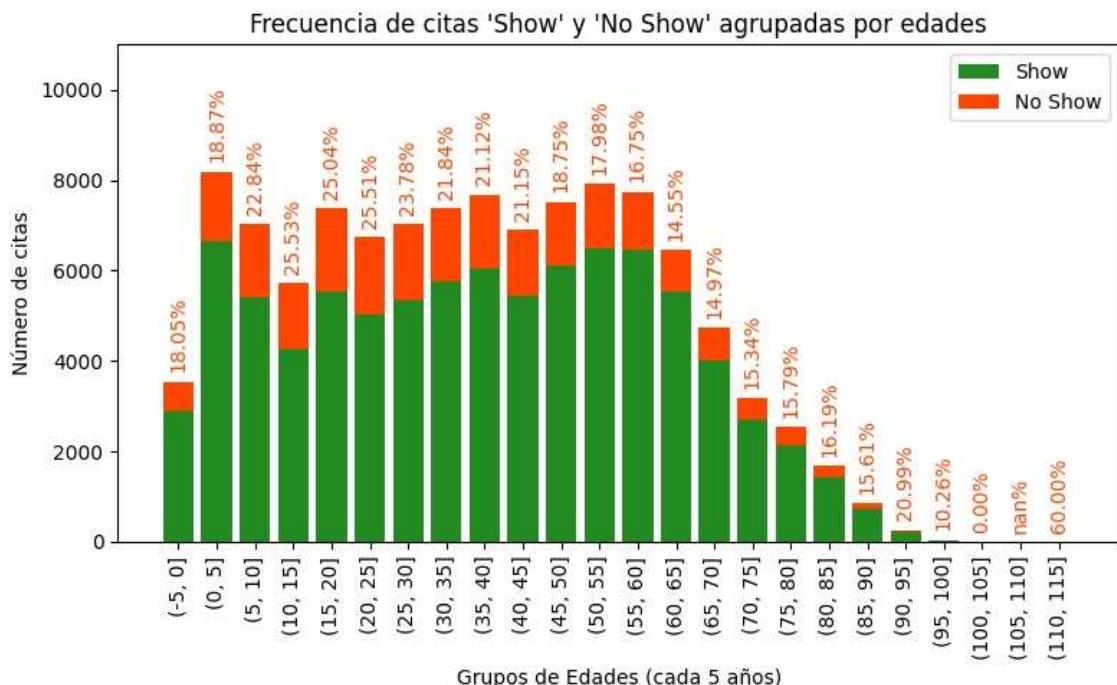


Las fechas indicadas en el gráfico corresponden a todos los lunes de la variable "AppointmentDay", y cada barra representa un día de la semana.

También se observa que el balanceo de los datos para cada día de la semana está bastante equilibrado, salvo para el único sábado del que se tiene constancia (14 mayo), cuyas citas se eliminaran del dataset final para evitar outliers que generen ruido al modelo.

#### Variable Predictora: "Age"

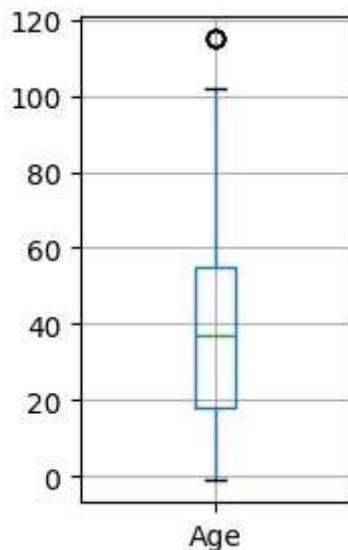
Respecto a la edad, el dataset comprende 104 edades distintas entre todos los pacientes, desde -1 a 115, según la siguiente distribución agrupadas las edades de 5 en 5:



Los pacientes que más faltan a las citas médicas son los comprendidos entre los 5 y los 30 años, así como los mayores de 110, pero de este último grupo hay muy pocos pacientes, por lo que no es representativo.

En esta variable se observa un severo desbalanceo en los pacientes de mayor edad, como es natural, pues el número de pacientes es mucho menor.

En esta variable es importante hacer un estudio más detallado de los outliers, utilizando incluso un gráfico boxplot (de caja y bigotes):



Las conclusiones para las edades outliers son las siguientes:

- -1: Sólo existe una cita para esta edad. Entendemos que es la madre atendiendo una consulta de ginecología.
- 0, 1, 2, ...: Existen 3539, 2273, 1618, ... citas para estas edades, las cuales no son desdeñables. Obviamente son citas en las que los pacientes (bebés) van acompañados por alguno de sus padres.
- 99: 1 sola cita
- 100: 4 citas correspondientes a 3 pacientes distintos.
- 102: 2 citas correspondientes a 2 pacientes distintos.
- 115: 5 citas correspondientes a un único paciente.

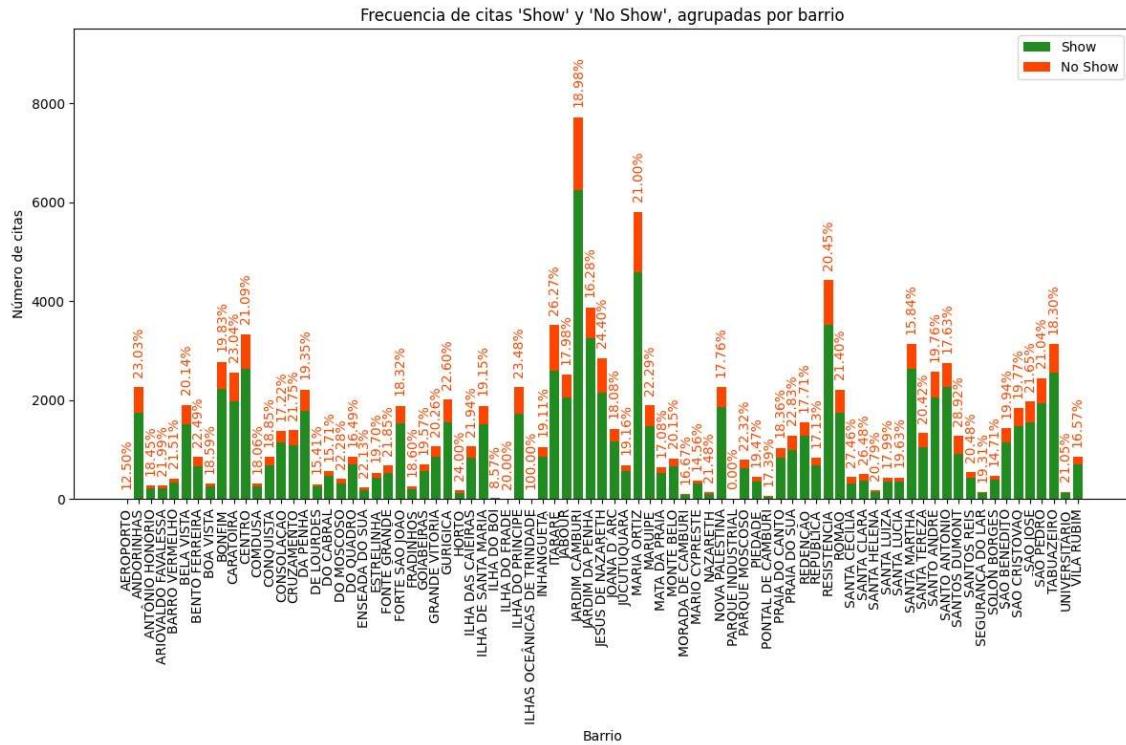
La edad es una información importante para pronosticar la asistencia o no las citas médicas, por lo que se deja toda la información intacta.

Sin embargo, se hacen un par de comprobaciones más para determinar la coherencia y robustez de los datos:

- Se comprueba que ningún bebé recién nacido presenta antecedentes de hipertensión, diabetes o alcoholismo (otras variables predictoras presentes en el dataset): OK.
- Se comprueba que los pacientes sólo tengan una edad asignada: Se detectan 1168 pacientes con más de una edad asignada, pero cuyas diferencias nunca son mayores de 1 año, por lo que se asume que son pacientes que han cumplido años entre una cita y la siguiente.

#### Variable Predictora: “Neighbourhood”

El dataset contiene información de 81 barrios distintos, todos correspondientes a la ciudad de Vitória, en Brasil.



El gráfico muestra un gran desbalanceo de datos en función del barrio considerado, y unas frecuencias de asistencia bastante dispares.

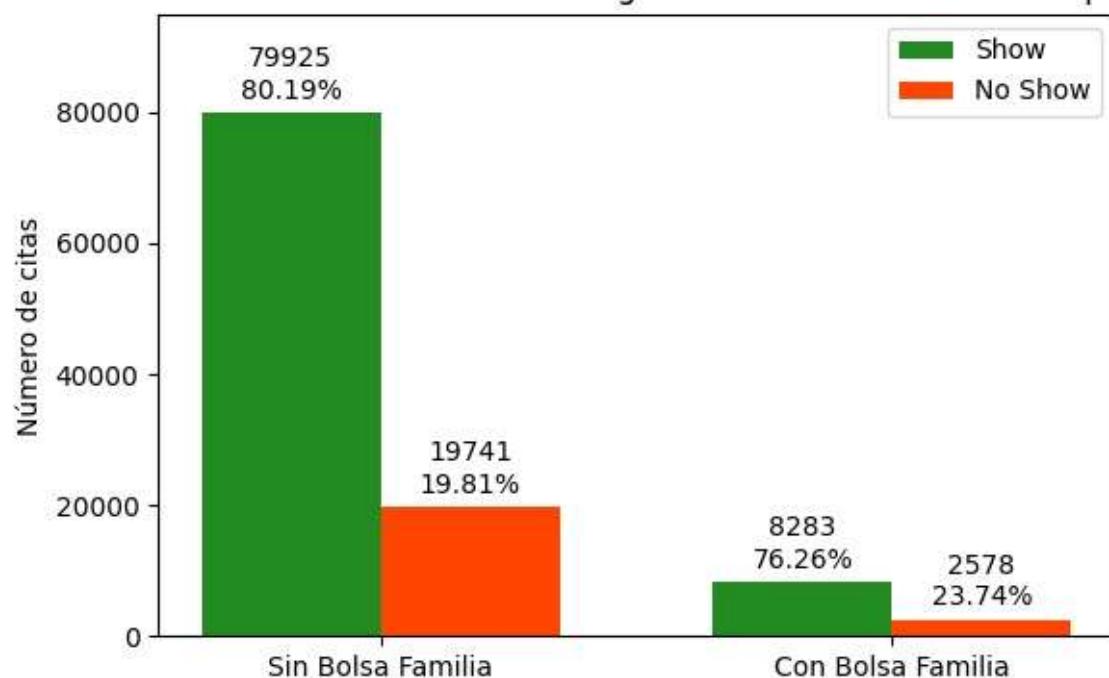
Con el objetivo de mejorar este desbalanceo y tener un modelo que haga unas predicciones más robustas, agruparemos los barrios por clúster de proximidad en el Feature Engineering.

#### Variable Predictora: “Scholarship”

Esta variable contiene información sobre unas becas de ayuda económica que se otorgan en Brasil, siendo una variable binaria indicando si el paciente que solicita la cita dispone de dicha beca o no.

Esta variable está fuertemente desbalanceada, pues sólo un 9.83% de las citas pertenecen a pacientes que disponen de la “Scholarship”, los cuales son los que exhiben un mayor porcentaje de No Show.

Distribución de No-Shows según existencia de 'Scholarship'

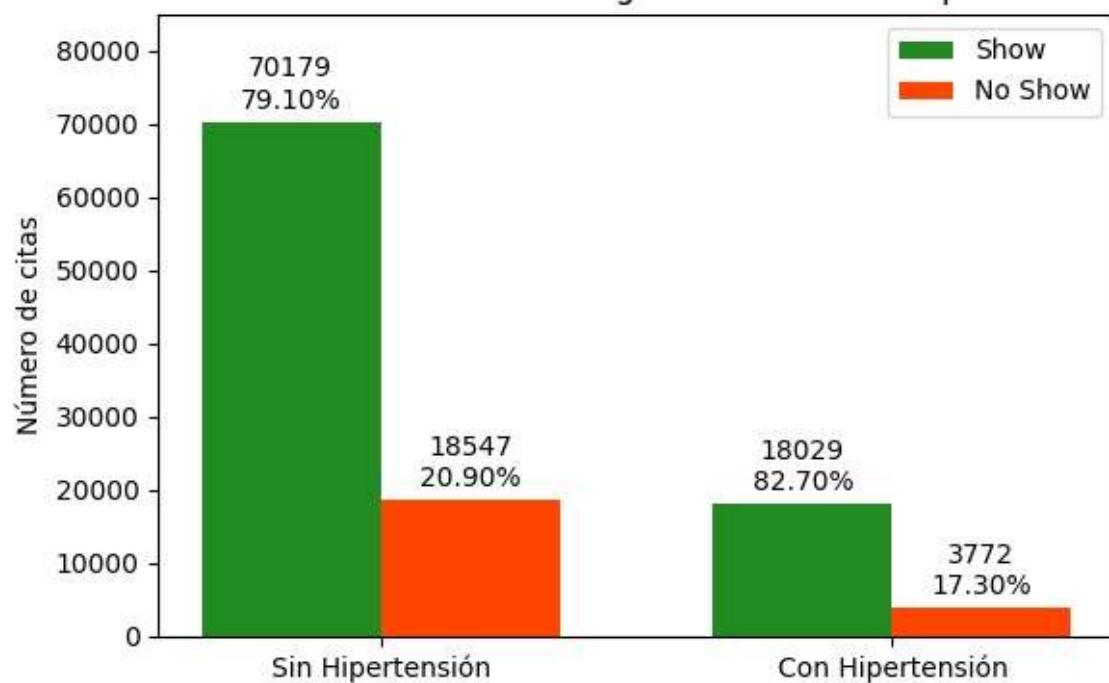


**Variable Predictora: "Hypertension"**

Esta variable también es una variable binaria para indicar si el paciente que acude a la cita tiene hipertensión o no.

Es una variable muy desbalanceada, con un 19.72% de las citas pertenecientes a pacientes que sufren de hipertensión, aunque son los que mejor índice de asistencia tienen.

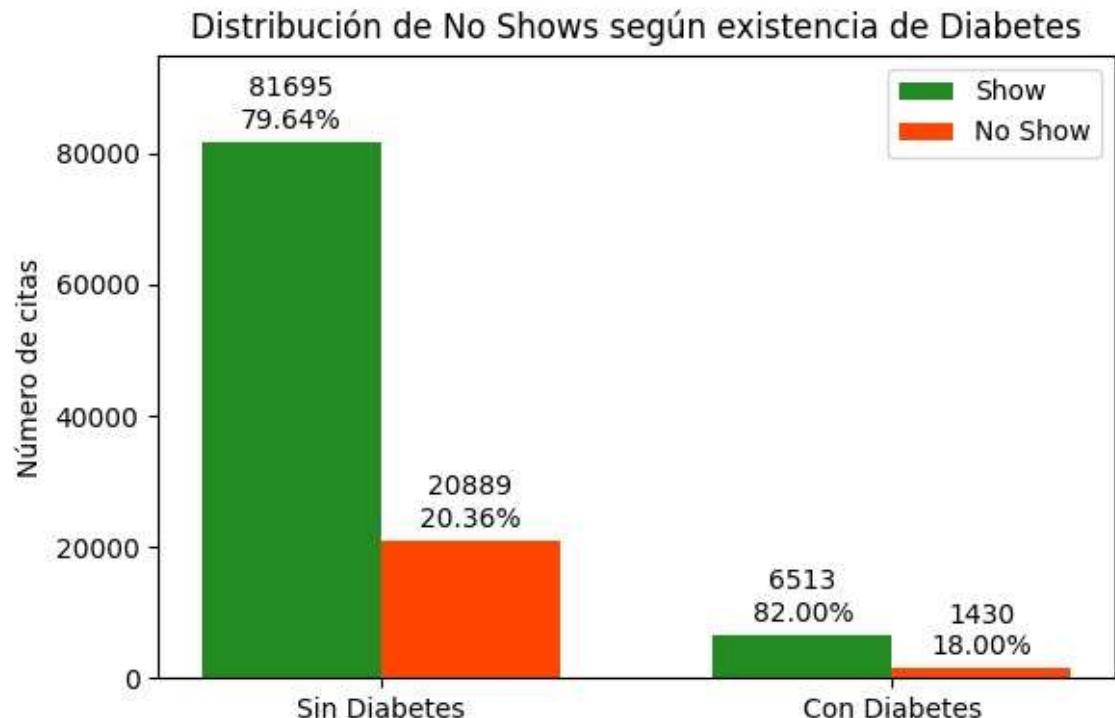
Distribución de No Shows según existencia de Hipertensión



#### Variable Predictora: “Diabetes”

Al igual que las dos variables anteriores, ésta también es binaria, indicando si el paciente sufre de diabetes o no.

Esta variable predictora también está fuerte desbalanceada, con tan sólo un 7.19% de las citas indicando que el paciente solicitante sufre de diabetes, y, al igual que pasaba con los pacientes que sufrían de hipertensión, son los que menos faltan a las citas programadas.



#### Variable Predictora: “Alcoholism”

Variable predictora binaria que indica si el paciente solicitante de la cita médica es alcohólico o no.

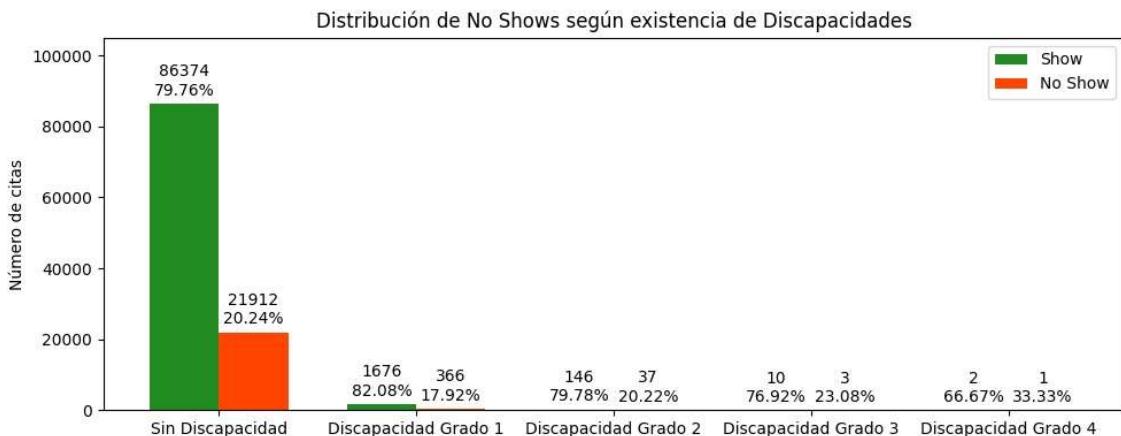
Esta variable está fuertemente desbalanceada, pues tan sólo un 3.04% de las citas del dataset corresponden a pacientes con alcoholismo. Aun así, tampoco es que existan diferencias relevantes en la frecuencia de asistencia a las citas médicas entre un grupo y otro.



#### Variable Predictora: “Handicap”

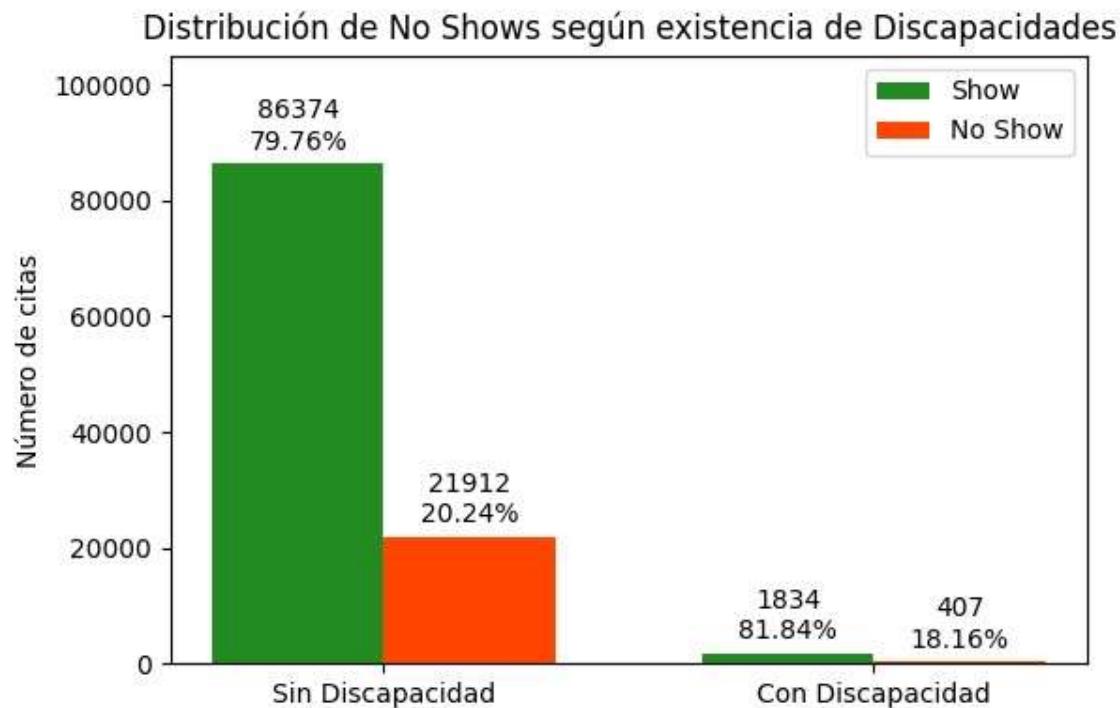
Esta variable indica si el paciente que solicita la cita tiene algún grado de discapacidad. Se miden hasta 4 grados de discapacidad, por lo que su valor varía entre 0 (ninguna discapacidad), 1 (grado de discapacidad 1), 2 (grado de discapacidad 2), 3 (grado de discapacidad 3) y 4 (grado de discapacidad 4).

Esta es la variable predictora más desbalanceada de todas: tan sólo el 1.8475% (2042 citas) corresponde a citas de pacientes con un grado de discapacidad 1, un 0.1656% (183 citas) corresponde citas de pacientes con un grado de discapacidad 2, un 0.0118% (13 citas) corresponden a un grado de discapacidad 3, y un ínfimo 0.0027% corresponden a 3 citas de 3 pacientes distintos con un grado de discapacidad 4.



Es por eso por lo que todas las citas que indican algún grado de discapacidad en el paciente se consideran outliers. Con tal de mejorar la robustez del modelo se procede a

agrupar todos los pacientes con discapacidad a un solo grupo, independientemente del grado de discapacidad médica.



Aun así, tal y como se observa en la gráfica anterior, no se consigue arreglar el desbalanceo, pues la suma de todas las citas con indicación de algún grado de discapacidad tan sólo representa el 2.0276%.

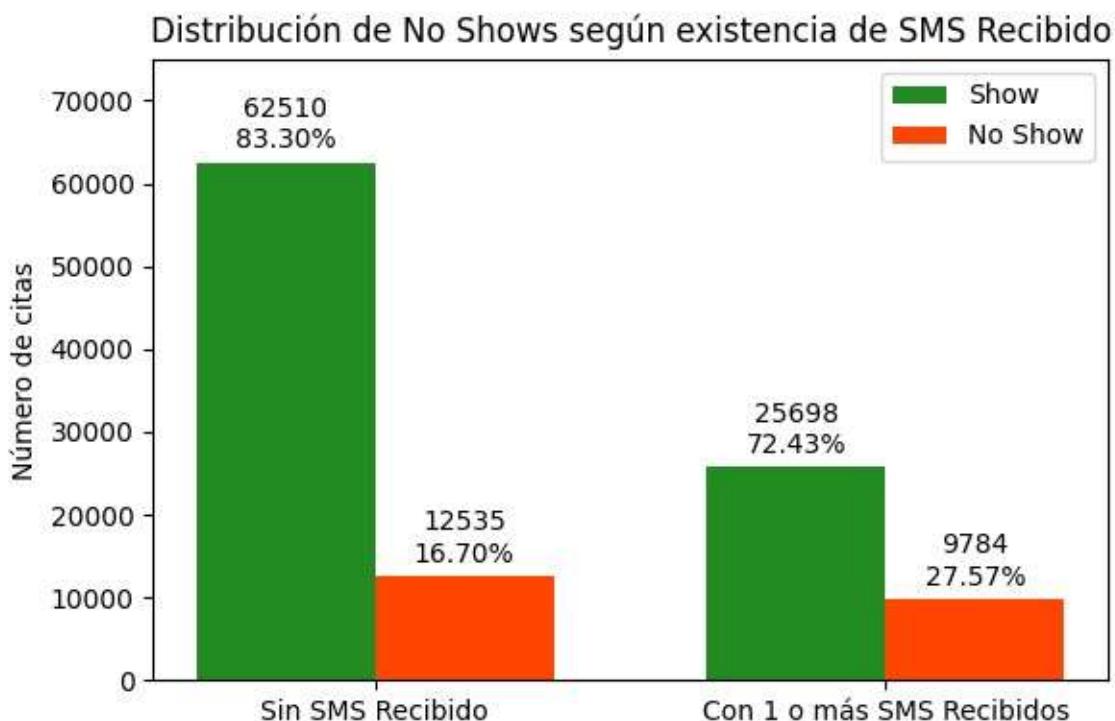
Es por ello por lo que, vistos también los desbalances existentes en todas las variables predictoras que describen la historia clínica de los pacientes, se trabajarán nuevas características predictoras en la fase de Featuring Engineering que aglutinen los pacientes con dolencias médicas, independientemente de cuál sea.

#### Variable Predictora: “SMS\_received”

Esta es la última variable predictora original del dataset, y se trata de una variable binaria que indica si el paciente que asistía a la cita médica recibió [0] o no [1] algún mensaje de texto recordatorio para dicha asistencia.

La variable está desbalanceada (*aunque no tanto como las que describían el historial médico de los pacientes*), pues tan sólo se enviaron mensajes de texto en un 32.10% de las citas contenidas en este dataset.

Curiosamente, y pareciendo “a priori” fuera de toda lógica, los pacientes que recibieron mensajes de texto recordatorios son los que muestran un porcentaje de No Show mayor.



Sin embargo, este comportamiento “fuera de lógica” guarda una correlación encubierta con el lapsus entre la fecha de programación de cita y la fecha de atención a la cita. Los mensajes de texto sólo se envían cuando dicho lapsus supera cierto margen (no se envían mensajes de textos si la cita se programa para los próximos días).

## Feature Engineering

La fase de Feature Engineering en este proyecto se enfoca en transformar y seleccionar las características más relevantes del conjunto de datos para mejorar el rendimiento de los modelos predictivos de IA, específicamente diseñados para disminuir las ausencias (no-shows) y mejorar la confirmación de citas. A continuación, se describen las acciones y transformaciones específicas que se han realizado:

### Creación de nuevas características

Para mejorar la capacidad predictiva sobre la asistencia a citas médicas, se han creado nuevas variables predictoras basadas en diferentes aspectos, desde patrones temporales hasta factores geográficos y meteorológicos. Estas variables se han creado para capturar aspectos clave que podrían influir en el comportamiento de los pacientes. A continuación, se describen en detalle estas nuevas características.

#### *Variables temporales*

Se han creado nuevas variables predictoras basadas en las fechas de solicitud (ScheduledDay) y asistencia a la cita (AppointmentDay). Se parte de la hipótesis de que el día de la semana en que se programa una cita médica puede afectar la probabilidad de asistencia, dado que ciertos días pueden presentar una mayor disponibilidad o disposición por parte de los pacientes para asistir. Así mismo, se plantea que el tiempo transcurrido

entre la programación de la cita hasta la fecha real de la misma podría influir en la probabilidad de asistencia

<b>Nueva variable</b>	<b>Descripción</b>	<b>Variable original</b>
<i>App_DayOfWeek</i>	Día de la semana en que se atiende la cita médica. Valores posibles: 'Friday' 'Wednesday' 'Monday', etc.	AppointmentDay
<i>Time_SchDay_to_AppDay</i>	Tiempo (en segundos) transcurrido entre la fecha de solicitud y la fecha de la cita.	AppointmentDay, ScheduledDay
<i>Days_since_last_App</i>	Tiempo (en días) transcurrido entre las citas consecutivas por paciente.	AppointmentDay

### **Variables geográficas**

Se han introducido variables predictoras adicionales basadas en la información geográfica de los barrios, incluyendo su ubicación (8) y la presencia de centros de salud en ellos (9). La hipótesis subyacente sugiere que la proximidad geográfica a los centros médicos y la disponibilidad de atención médica en el barrio pueden influir en la probabilidad de asistencia a las citas médicas.

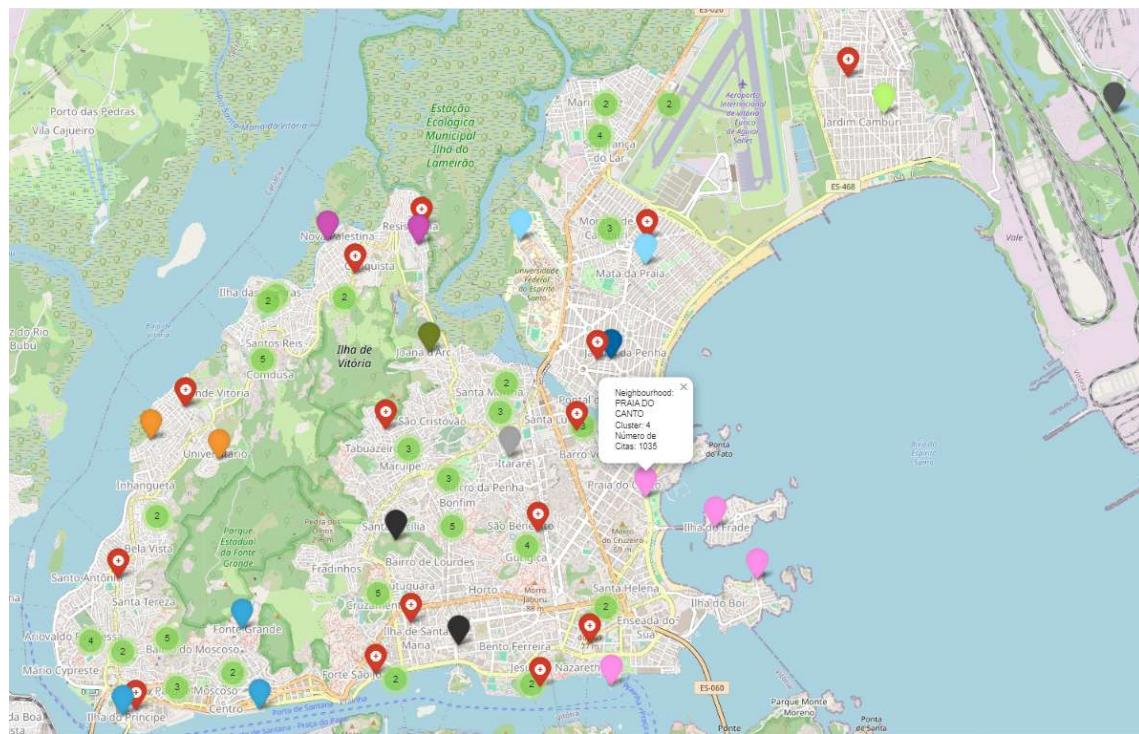
<b>Nueva variable</b>	<b>Descripción</b>	<b>Variable original</b>
<i>Neigh_Cluster</i>	Etiquetas de clúster asignadas a cada barrio mediante el modelo de K-Means. Esto permite agrupar los barrios según su proximidad geográfica.	LatitudeNeigh, LongitudeNeigh
<i>Health_Centre</i>	Identificación binaria que especifica la presencia (1) o ausencia (0) de Centro Médico en cada barrio.	

Se ha utilizado el algoritmo **K-Means**, una técnica de aprendizaje no supervisado, para entender cómo la proximidad geográfica y la disponibilidad de servicios de salud afectan la tasa de ausencias a las citas. El proceso se llevó a cabo de la siguiente manera:

1. **Selección de características:** Para agrupar los barrios, se seleccionaron las coordenadas geográficas (latitud y longitud) de cada barrio. Estas características permiten agrupar los barrios en función de su proximidad geográfica.

2. **Determinación del número de clústers:** Se decidió agrupar los barrios en 12 clústers. Esta decisión se basó en la distribución geográfica y la cantidad total de barrios, buscando un equilibrio entre la granularidad de los grupos y la manejabilidad de los datos.
  3. **Aplicación de K-Means:** Utilizando el algoritmo K-Means, se asignó a cada barrio una etiqueta de clúster que indica a qué grupo pertenecía.
  4. **Ajuste manual de la clusterización:** Tras la clusterización inicial, se realizó un ajuste manual para balancear mejor el número de citas médicas entre los clústers. Este ajuste consistió en mover ciertos barrios entre clústers para asegurar una distribución más equitativa y efectiva de los recursos médicos.

Para facilitar la comprensión y comunicación de los resultados, se creó un mapa interactivo utilizando Folium. Este mapa muestra los barrios agrupados por clústers, con diferentes colores para cada uno, y señala la ubicación de los centros de salud en rojo. Además, el mapa incluye información sobre el número de citas médicas por barrio, lo que permite identificar de manera clara y accesible la distribución geográfica de los barrios, los recursos de salud disponibles y la carga de trabajo en cada área.



## *Variables meteorológicas*

Otra hipótesis plantea que la asistencia a la cita médica esté fuertemente influenciada por las condiciones meteorológicas del día en cuestión, como la temperatura, la velocidad del viento y la lluvia. Para investigar esta suposición, se procedió a extraer la información meteorológica de la ciudad de Vitoria para el mes de mayo de 2016, obtenida de Weather and Climate (10). Posteriormente, se integraron al conjunto de datos las nuevas variables predictoras: ‘Temperature’, ‘WindSpeed’, ‘Precipitation’.

## *Variables del historial médico*

Bajo la premisa de que la combinación de las condiciones médicas de un paciente puede tener relevancia en la predicción de la asistencia a las citas médicas, se han introducido variables adicionales basadas en el historial médico de los pacientes.

Nueva variable	Descripción	Variable original
<i>Number_Health_Conds</i>	Número total de condiciones médicas que tiene cada paciente. Valores posibles: 0, 1, 2, 3, 4.	Hypertension, Diabetes, Alcoholism, Handicap
<i>Presence_Health_Conds</i>	Identificación binaria que especifica la presencia (1) o ausencia (0) de condiciones médicas por paciente.	<i>Number_Health_Conds</i>

Además, para determinar la probabilidad de asistencia del paciente, se han considerado las siguientes estadísticas:

Nueva variable	Descripción
<i>Prior_Apps_byPatient</i>	Número de citas previas de cada paciente.
<i>Prior_NoShows_byPatient</i>	Indica la cantidad de veces que un paciente no asistió a citas previas.
<i>Prob_NoShow_byPatient</i>	Porcentaje de no asistencia del paciente, basado en su historial de citas.

Es importante señalar que el porcentaje de no asistencia se inicializa para cada paciente con el valor promedio de no asistencia en todo el conjunto de datos. Posteriormente, este valor se actualiza según el historial específico de cada paciente a medida que tiene citas médicas.

**IMPORTANTE:** En el fichero jupyter habría que volver a calcular los valores de **NUMBER\_NO\_SHOWS / NUMBER\_SAMPLES** porque se han eliminado registros.

#### Transformación de tipos de datos

Durante el proceso de preparación de los datos, se realizaron varias transformaciones para garantizar la coherencia y la adecuación de los tipos de datos. A continuación, se detallan las principales transformaciones llevadas a cabo:

**Conversiones de tipo de datos numéricos.** Se ajustaron los tipos de datos numéricos, como enteros o flotantes, para garantizar la consistencia en las operaciones matemáticas y el análisis estadístico. Esto incluyó la conversión de variables como el identificador del paciente (*PatientId*) y el género (*Gender*).

**Manipulación de fechas y horas.** Las variables relacionadas con fechas y horas, como las fechas de programación (*ScheduledDay*) y las fechas de las citas médicas

(*AppointmentDay*), se convirtieron al formato de fecha y hora adecuado para facilitar su manipulación y análisis.

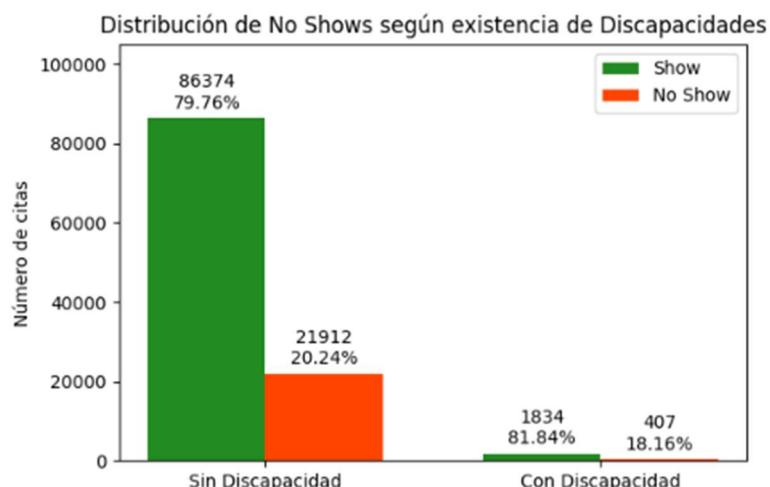
**Codificación de variables categóricas.** Se utilizó LabelEncoder para convertir variables categóricas, como el género (*Gender*), día de la semana de la cita (*App\_DayOfWeek*) y el barrio (*Neighbourhood*), en valores numéricos. Esto se realizó para permitir el procesamiento de algoritmos de aprendizaje automático que requieren datos numéricos como entrada.

Además, la **variable target 'NoShow'** fue transformada a tipo booleano para facilitar su manipulación y análisis durante el modelado predictivo: 0 (sí asiste), 1 (no asiste).

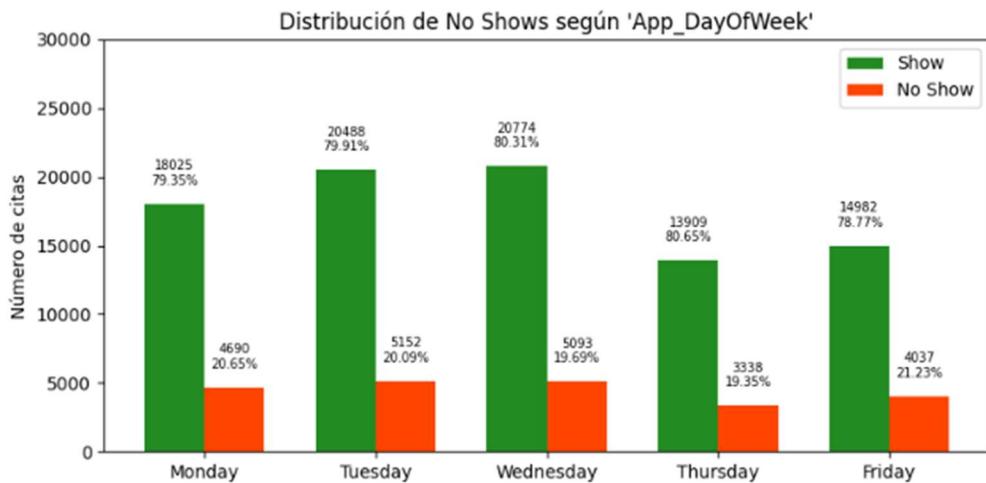
#### Tratamiento de valores atípicos y missing values

Durante el análisis de datos, se identificaron valores atípicos y algunos registros que requirieron un tratamiento especial. A continuación, se detallan las acciones tomadas:

**Condición de discapacidad:** La variable ‘*Handicap*’ originalmente incluía valores de 0 a 4, indicando niveles de discapacidad. Sin embargo, el análisis exploratorio de datos (EDA) reveló que los niveles superiores a 0 no eran relevantes. Por lo tanto, se simplificó la variable para que solo indique si un paciente tiene discapacidad (1) o no (0).



**Eliminación de citas programadas para los sábados:** La variable ‘*App\_DayOfWeek*’ mostró que solo había 39 citas programadas para los sábados. Para evitar posibles sesgos debido al pequeño tamaño de esta muestra y reducir el ruido en el modelo, se decidió eliminar estas citas del conjunto de datos.



**Gestión de valores negativos en el tiempo entre programación y cita:** Se identificaron 5 registros en la variable ‘Time\_SchDay\_to\_AppDay’ con valores negativos, indicando que la fecha de programación era posterior a la fecha de la cita. Estos casos se corrigieron asignando un valor de 0, asumiendo que el paciente programó la cita el mismo día de la consulta.

PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	App_DayOfWeek	Time_SchDay_to_AppDay	Age	Neighbourhood	Scholarship	Hypertension	Diabetes	Alcoholism	Handicap	SMS_received	NoShow
72362	3787481966821	M	2016-05-04 06:50:57	2016-05-03	Tuesday	-111057	7	TABUAZEIRO	0	0	0	0	0	0	True
27033	7839272661752	M	2016-05-10 10:51:53	2016-05-09	Monday	-125513	38	RESISTÊNCIA	0	0	0	0	1	0	True
55226	7896293967868	F	2016-05-18 14:50:41	2016-05-17	Tuesday	-139841	19	SANTO ANTÔNIO	0	0	0	0	1	0	True
64175	24252258389979	F	2016-05-05 13:43:58	2016-05-04	Wednesday	-135838	22	CONSOLAÇÃO	0	0	0	0	0	0	True
71533	998231581612122	F	2016-05-11 13:49:20	2016-05-05	Thursday	-568160	81	SANTO ANTÔNIO	0	0	0	0	0	0	True

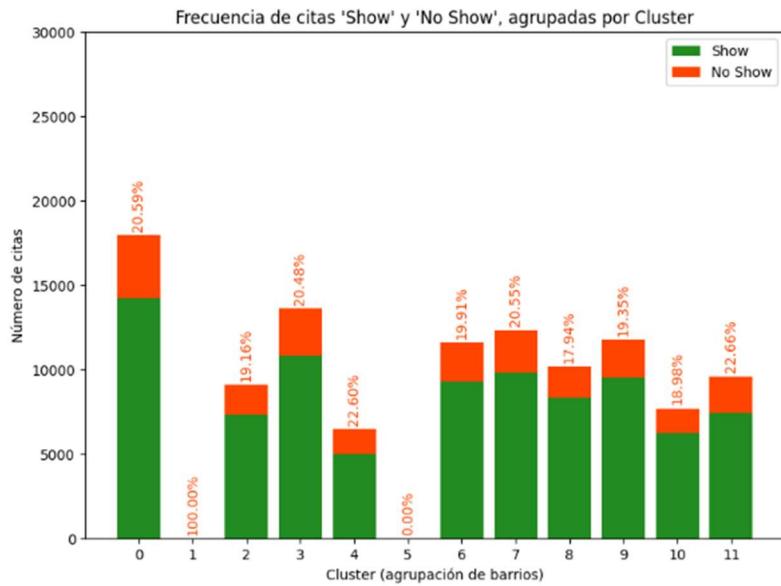
```
[ ] # Corrección a 0's de todos los 'Time_SchDay_to_AppDay' negativos:
med_app_FE['Time_SchDay_to_AppDay'][med_app_FE['Time_SchDay_to_AppDay'] <= 0] = 0
```

**Asignación de valor -1 a pacientes sin citas previas:** En la variable ‘Days\_since\_last\_App’, los pacientes sin citas previas recibieron un valor de -1, indicando la falta de datos previos para calcular el tiempo transcurrido.

```
[ ] # Cálculo e inserción de la variable del tiempo entre citas de un mismo paciente
med_app_FE.insert(loc = 7,
                   column = 'Days_since_last_App',
                   value = med_app_FE.groupby('PatientId')['AppointmentDay'].diff().dt.days)

# Asignación -1 a todos las citas de pacientes que no han tenido citas previas
med_app_FE['Days_since_last_App'].fillna(-1, inplace=True)
```

**Ajuste de clústeres en ‘Neigh\_Cluster’:** Se detectaron clústeres con solo 1 o 2 muestras (específicamente los clústeres 1 y 5). Estos clústeres se eliminaron y se reorganizaron los restantes, reduciendo el número de clústeres de 12 a 10 para eliminar los menos representativos.



```
[ ] # Eliminación de los 2 clusters con una o dos cita:
indexes = med_app_FE.loc[(med_app_FE['Neigh_Cluster'] == 1) | \
                           (med_app_FE['Neigh_Cluster'] == 5)].index.tolist()

med_app_FE.drop(indexes, inplace = True)

# Corremos todos los valores de los cluster del 0 al 9:
for i in range(2, 12):
    if i < 6:
        med_app_FE.loc[med_app_FE['Neigh_Cluster'] == i, 'Neigh_Cluster'] = i-1
    else:
        med_app_FE.loc[med_app_FE['Neigh_Cluster'] == i, 'Neigh_Cluster'] = i-2
```

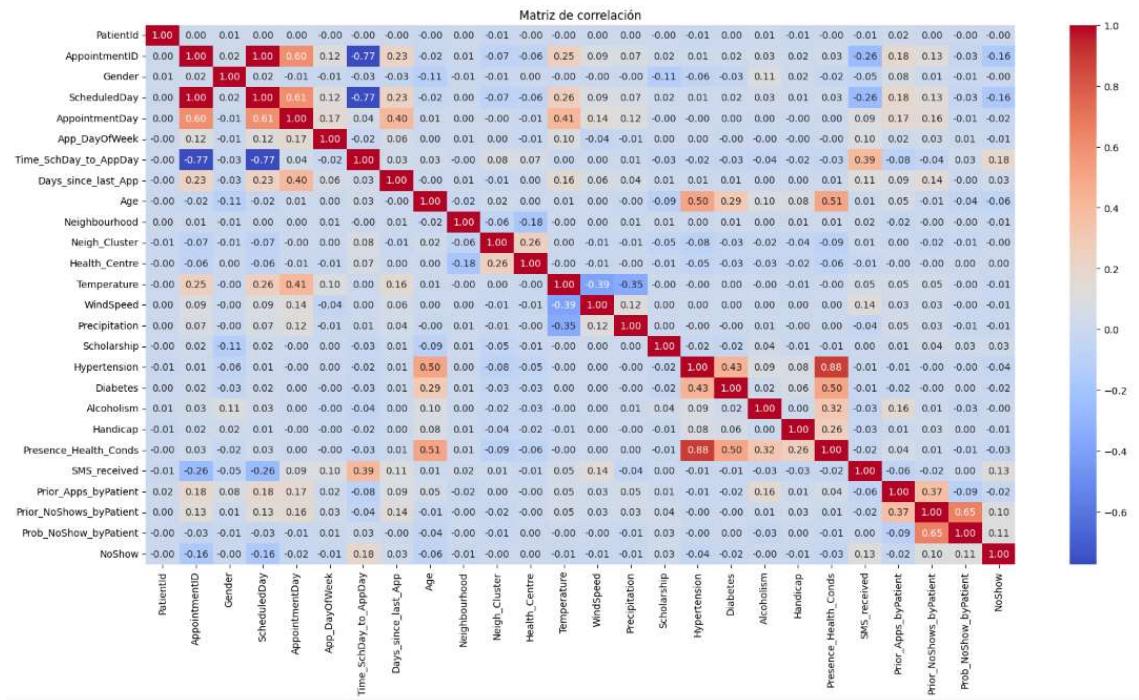
Es importante señalar que en el dataset original no se encontraron valores faltantes. Estas medidas aseguraron la integridad y coherencia de los datos utilizados para el análisis posterior.

### Validación de características

El análisis de correlación se utilizó para identificar relaciones lineales entre las características y la variable objetivo, así como entre las propias características. Esto incluye:

**Matriz de correlación:** Se calculó la matriz de correlación de Pearson para todas las características numéricas, lo que permitió identificar características redundantes que podrían ser eliminadas o combinadas.

**Correlación con la variable objetivo:** Se analizaron las correlaciones entre cada característica y la variable objetivo ('NoShow') para identificar las características con mayor poder predictivo.



## Selección de características

En esta sección, se presenta una comparación detallada entre la estructura del dataset original y el dataset resultante después de aplicar técnicas de ingeniería de características (Feature Engineering).

### Dataset Original

El dataset inicial contiene una serie de variables básicas y diversas para la gestión de citas médicas. Estas variables proporcionan una base inicial que, aunque útil, requiere refinamiento para mejorar su capacidad predictiva.

### Dataset tratado

Tras aplicar técnicas de ingeniería de características, el dataset se modificó significativamente. Se añadieron nuevas variables predictoras y se ajustaron los tipos de datos de varias de las variables originales. En la imagen incluida, se destacan en color rojo las modificaciones y las nuevas variables añadidas. Esta transformación es fundamental para aumentar la capacidad del modelo de IA para identificar patrones y tendencias relacionadas con la asistencia a citas médicas.

### Dataset Final

Finalmente, se realizó una selección exhaustiva de características para crear el dataset final. Este conjunto de datos, que se utilizará para la normalización, escalado, reducción de dimensionalidad y particionamiento, incluye solo las variables más relevantes y útiles para la predicción de asistencia a citas médicas. En la imagen, las variables eliminadas se muestran en gris, mientras que las variables seleccionadas se destaca en negrita.

**DATASET ORIGINAL**

PatientId	float64
AppointmentID	int64
Gender	object
ScheduledDay	object
AppointmentDay	object
Age	int64
Scholarship	int64
Hypertension	int64
Diabetes	int64
Alcoholism	int64
Handicap	int64
SMS_received	int64
NoShow	object

**DATASET TRATADO**

PatientId	int64
AppointmentID	int64
Gender	int32
ScheduledDay	datetime64[ns]
AppointmentDay	datetime64[ns]
App_DayOfWeek	int32
Time_SchDay_to_AppDay	int64
Days_since_last_App	int64
Age	int64
Neighbourhood	int32
Neigh_Cluster	int64
Health_Centre	int64
Temperature	float64
WindSpeed	float64
Precipitation	float64
Scholarship	int64
Hypertension	int64
Diabetes	int64
Alcoholism	int64
Handicap	int64
Presence_Health_Conds	int64
SMS_received	int64
Prior_Apps_byPatient	int64
Prior_NoShows_byPatient	int64
Prob_NoShow_byPatient	float64
NoShow	bool

**DATASET FINAL**

1 PatientId	int64
2 AppointmentID	int64
3 Gender	int32
4 ScheduledDay	datetime64[ns]
5 AppointmentDay	datetime64[ns]
6 App_DayOfWeek	int32
7 Time_SchDay_to_AppDay	int64
8 Days_since_last_App	int64
9 Age	int64
10 Neighbourhood	int32
11 Neigh_Cluster	int64
12 Health_Centre	int64
13 Temperature	float64
14 WindSpeed	float64
15 Precipitation	float64
16 Scholarship	int64
17 Hypertension	int64
18 Diabetes	int64
19 Alcoholism	int64
20 Handicap	int64
21 Presence_Health_Conds	int64
22 SMS_received	int64
23 Prior_Apps_byPatient	int64
24 Prior_NoShows_byPatient	int64
25 Prob_NoShow_byPatient	float64
26 NoShow	bool

## Iteraciones y refinamiento

Durante el proceso de Feature Engineering para el proyecto de predicción de asistencia a citas médicas, se llevaron a cabo varias iteraciones y refinamientos para optimizar las características utilizadas en el modelo predictivo. Estas iteraciones se basaron en los hallazgos del análisis exploratorio de datos, así como en la retroalimentación de la validación del modelo y los cambios en los requisitos del proyecto.

### Iteraciones

- Se realizaron iteraciones en la selección y creación de características en respuesta a los patrones observados durante el análisis exploratorio de datos. Por ejemplo, se exploraron diferentes formas de representar el historial médico de los pacientes para capturar de manera efectiva su impacto en la asistencia a las citas.
- Se llevaron a cabo ajustes en las características temporales, como el tiempo transcurrido entre la programación de la cita y la fecha real de la misma, para capturar con mayor precisión la influencia de los factores temporales en la probabilidad de asistencia.
- Se exploraron técnicas avanzadas de transformación de variables geográficas, como la segmentación espacial, para capturar la influencia de la ubicación geográfica en la asistencia a las citas médicas.

### Refinamiento

- Se refinaron las características seleccionadas en función de su capacidad predictiva y su interpretabilidad. Por ejemplo, se realizaron ajustes en las características basadas en el historial médico de los pacientes para equilibrar la representación de diferentes condiciones médicas.
- Se evaluaron y refinaron las técnicas de imputación de valores faltantes y manejo de valores atípicos para mejorar la calidad de los datos y reducir el impacto de datos erróneos en el modelo.

## Normalización y escalado de características

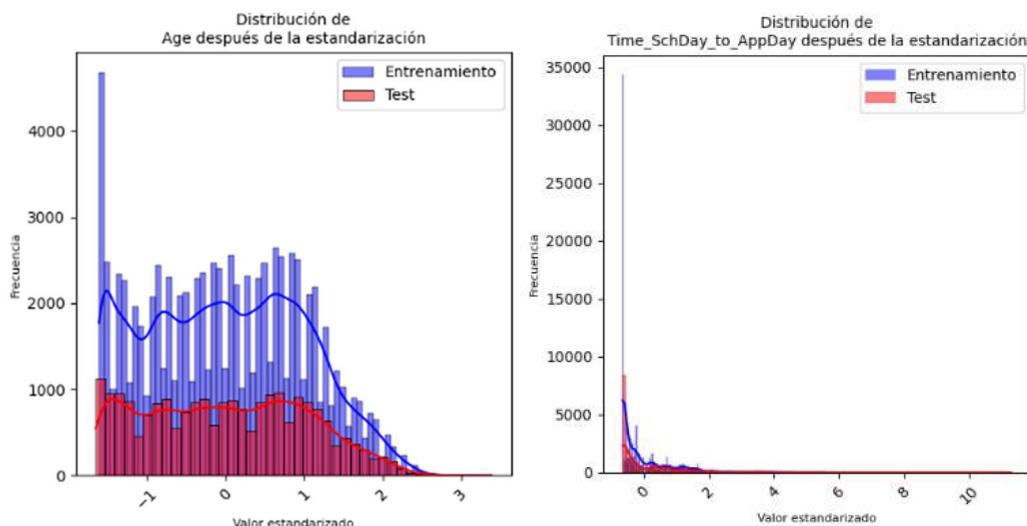
La estandarización y el escalado de características son procesos fundamentales en el preprocesamiento de datos para muchos algoritmos de aprendizaje automático. En nuestro proyecto de gestión de citas médicas, estos pasos son esenciales para asegurar que las características de entrada estén en una escala comparable, facilitando así la convergencia de los modelos y mejorando su capacidad predictiva.

Para llevar a cabo la estandarización de nuestras bases de datos, hemos utilizado la clase **StandardScaler** de la biblioteca scikit-learn. Este proceso implica ajustar el escalador utilizando exclusivamente los datos de entrenamiento, permitiendo que el escalador "aprenda" de estos datos. Luego, aplicamos la misma transformación a los datos de prueba utilizando el mismo escalador ajustado. De esta manera, garantizamos que la distribución de cada característica tenga una media de 0 y una desviación estándar de 1.

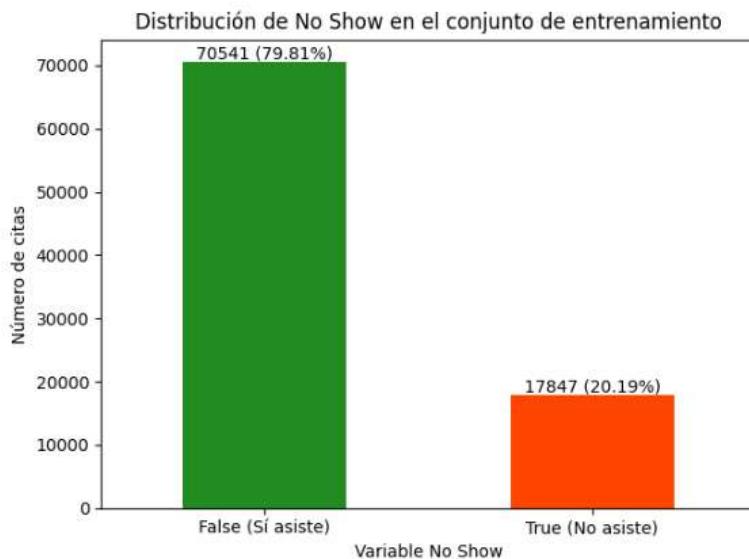
```
# Se crea una instancia de StandardScaler
scaler = StandardScaler()

# Se ajusta el escalador a los datos de entrenamiento y se transforman
X_train_scaled = scaler.fit_transform(X_train_set)
# Se transforman los datos de test utilizando el mismo escalador que se ajustó a los datos de entrenamiento
X_test_scaled = scaler.transform(X_test_set)
```

En el contexto de nuestra gestión de citas médicas, la estandarización es especialmente relevante debido a la diversidad de las características que consideramos. Por ejemplo, características como la edad del paciente, el tiempo transcurrido entre la fecha de solicitud y la fecha de la cita. La estandarización garantiza que las variaciones de las variables no afecten negativamente en el rendimiento de nuestros modelos, al tiempo que preserva la integridad de los datos originales.



Para abordar el desbalanceo de clases en nuestro conjunto de datos de entrenamiento, hemos aplicado dos técnicas de **Data Augmentation** de manera independiente: SMOTE-ENN y ADASYN.



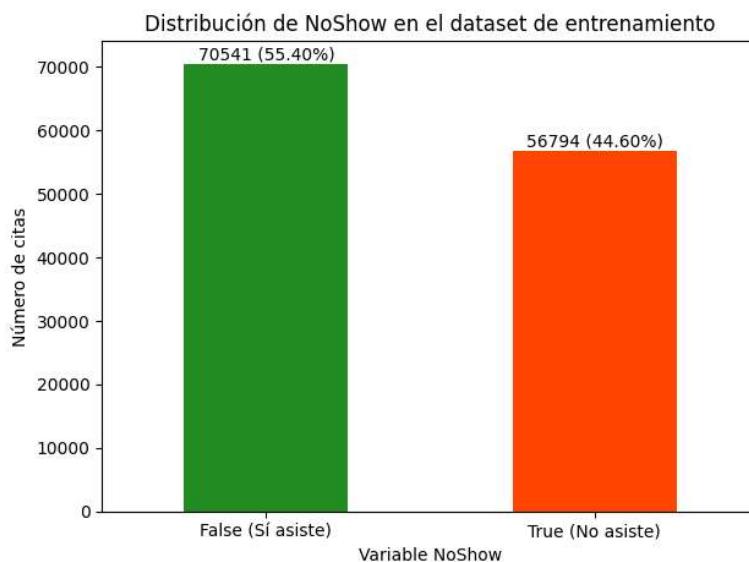
### **SMOTE-ENN**

Implementamos la técnica SMOTE-ENN mediante el uso de Pipeline de scikit-learn. SMOTE (*Synthetic Minority Over-sampling Technique*) genera nuevas muestras sintéticas de la clase minoritaria, mientras que ENN (*Edited Nearest Neighbours*) elimina instancias ruidosas, logrando así un equilibrio de clases en este conjunto de datos como se visualiza en la imagen.

```
[ ] # Se crea una instancia de SMOTE y ENN
smote = SMOTE(sampling_strategy = 'minority',
               random_state = 42)
enn = EditedNearestNeighbours(sampling_strategy = 'not minority',
                               kind_sel = 'all',
                               n_neighbors = 2)

# Se crea una instancia de la clase Pipeline
pipeline = Pipeline([('smote', smote), ('enn', enn)])

# Se aplica el pipeline a los datos
X_train_resampled, y_train_resampled = pipeline.fit_resample(X_train_scaled, y_train_set)
```

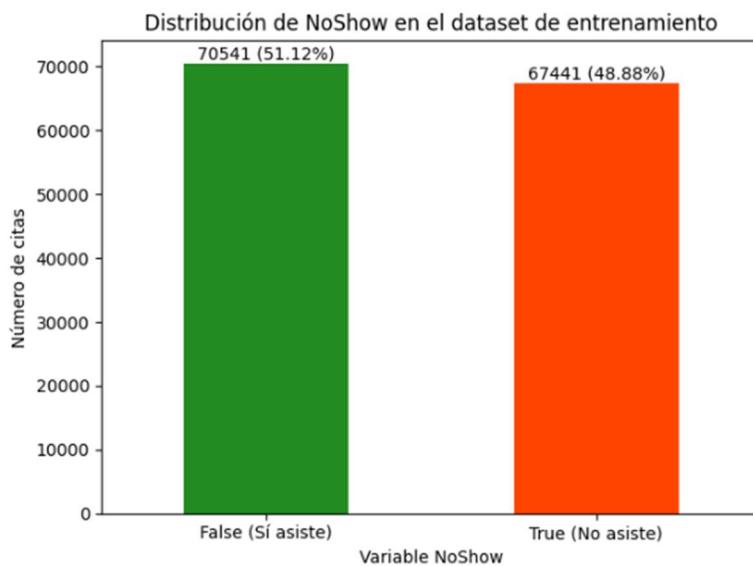


## ADASYN

En otra copia independiente de los datos de entrenamiento, aplicamos ADASYN (*Adaptive Synthetic Sampling Approach*). ADASYN es una técnica que no solo genera nuevas muestras sintéticas para la clase minoritaria, sino que lo hace de manera adaptativa. Esto significa que ADASYN prioriza la generación de nuevas muestras en las áreas del espacio de características donde los datos son más escasos y difíciles de clasificar. Al enfocarse en las casuísticas con menor densidad de datos, se espera que ADASYN mejore la capacidad del modelo para manejar casos difíciles y, por lo tanto, aumente la precisión del modelo en la clasificación de la clase minoritaria.

```
[ ] # Crear una instancia de ADASYN
adasyn = ADASYN(sampling_strategy='minority', random_state=42)

# Aplicar ADASYN a los datos de entrenamiento
X_train_resampled, y_train_resampled = adasyn.fit_resample(X_train_scaled, y_train_set)
```



## Reducción de dimensionalidad

La reducción de dimensionalidad es una técnica fundamental para simplificar el conjunto de datos y mejorar la eficiencia de los modelos de inteligencia artificial sin perder información importante. Para este propósito, se ha implementado el *Análisis de Componentes Principales (PCA)*, una técnica ampliamente utilizada en aprendizaje automático.

Con el PCA, nuestro objetivo era conservar al menos el 95% de la varianza total mientras reducímos la dimensionalidad de nuestro conjunto de datos.

```
[ ] # Aplicar PCA para reducir la dimensionalidad mientras se conserva la mayor cantidad de varianza posible
pca = PCA(n_components = 0.95,
           svd_solver = 'full')

# Ajustar y transformar los datos de entrenamiento
X_train_pca = pca.fit_transform(X_train_resampled)
X_test_pca = pca.transform(X_test_scaled)
```

Sin embargo, después de aplicar el PCA, observamos que, de las 19 variables predictoras originales, solo se seleccionaron 16 componentes principales. A pesar de ser un número menor de variables, estos componentes lograron explicar el 96.66% de la varianza total de los datos, proporcionando una representación efectiva de la información contenida en el conjunto de datos original.

```
Número de componentes: 16
Varianza explicada por cada componente: [0.13707614 0.12212461 0.09235221 0.0689055 0.06253879 0.05901496
0.05718571 0.05134638 0.05082864 0.05010441 0.04323952 0.03925022
0.03778197 0.03715579 0.03256729 0.02512141]
Varianza total explicada: 0.9665935231029232
```

Dada la posibilidad de que la reducción en el número de variables conlleve a una pérdida de información, consideramos que era más prudente mantener todas las variables originales en nuestros modelos. Esta decisión se basó en la observación de que el PCA eliminó solo un pequeño número de variables, y que esta pérdida no justificaba la posible reducción en la calidad de nuestras predicciones, especialmente considerando que ya estábamos conservando una varianza explicada del 95%.

## Entrenamiento de modelos

### Particionamiento de datos

Dado que no se dispone de un conjunto de datos de pruebas, se procede a generar uno mediante el particionamiento de los datos disponibles sin la etiqueta de asistencia ("No-show"). Este proceso se realiza con el propósito de evaluar el rendimiento de los modelos predictivos en datos no vistos durante el entrenamiento. Previamente, se decide crear conjuntos de datos de acuerdo a hipótesis sobre los que posteriormente se aplicará el particionamiento.

#### Hipótesis para la creación de conjuntos de datos:

- A. *Mantenimiento de la totalidad del conjunto de datos:* Bajo la premisa de que la retención de la totalidad de los datos maximiza la información disponible y potencialmente mejora la capacidad predictiva, se mantiene el conjunto de datos original sin ningún tipo de filtrado.
- B. *Pacientes sin condiciones médicas:* Se crea un conjunto de datos que incluye solo a los pacientes del conjunto original que no tienen condiciones médicas registradas. Esto se basa en la hipótesis subyacente de que los pacientes sin condiciones médicas pueden tener una mayor probabilidad de inasistencia. Al generar este dataset se eliminan las variables 'Hypertension', 'Diabetes', 'Alcoholism', 'Handicap', 'Presence\_Health\_Conds'.
- C. *Pacientes de edad entre 5 y 30 años:* Se genera un conjunto de datos que contiene únicamente a los pacientes del conjunto original cuya edad se encuentra entre 5 y 30 años. Esto se basa en la hipótesis subyacente de que este grupo demográfico tiende a faltar más a las citas médicas.
- D. *Pacientes con citas programadas para otro día:* Se forma un conjunto de datos que incluye exclusivamente a los pacientes del conjunto original cuyas citas están programadas para un día distinto al de su solicitud. Esto se sustenta en la hipótesis subyacente de que existe una correlación significativa entre este grupo de pacientes y la inasistencia.

- E. *Pacientes de barrios con centro médico*: Se establece un conjunto de datos que contiene únicamente a los pacientes del conjunto original de barrios donde existe un centro médico. Esto se basa en la hipótesis subyacente de que los pacientes que están en barrios con centros médicos pueden tener una menor sensación de urgencia o responsabilidad al tener acceso fácil a atención médica.

Procedimiento de particionamiento de datos:

Una vez formuladas las hipótesis y generado los conjuntos de datos correspondientes, se procede al particionamiento de los datos para entrenamiento y prueba. Este proceso se lleva a cabo con el objetivo de evaluar y validar la eficacia de los modelos de aprendizaje automático en datos no vistos durante el entrenamiento. El procedimiento se detalla a continuación:

1. *Cálculo de variables relevantes*. Para cada hipótesis, se calcula el número de pacientes en el conjunto de datos (NUMBER\_PATIENTS), el número de citas correspondientes a no asistencia (NUMBER\_NO\_SHOWS), y el número total de muestras en el conjunto de datos (NUMBER\_SAMPLES).
2. *Determinación del tamaño del conjunto de prueba*. Se determina el tamaño del conjunto de prueba multiplicando el número total de muestras por un porcentaje predefinido, usualmente el 20%. Esto garantiza una división adecuada entre los conjuntos de entrenamiento y prueba.

```
[ ] NUMBER_Apps_inTest = int(NUMBER_SAMPLES * 0.20)
print(f"Para alcanzar el 20% del muestreo en nuestro set de pruebas\n \
necesitamos {NUMBER_Apps_inTest} citas de pacientes distintos, es decir,\n \
muestras de un {NUMBER_Apps_inTest / NUMBER_PATIENTS:.2%} de los pacientes.")
```

→ Para alcanzar el 20% del muestreo en nuestro set de pruebas necesitamos 22097 citas de pacientes distintos, es decir, muestras de un 35.48% de los pacientes.

3. *Creación de subconjuntos de datos*. Se crea un subconjunto de datos que contiene únicamente la última cita de cada paciente, ordenado según la fecha de la cita, para cada hipótesis generada. Esto asegura que cada paciente esté representado por una única muestra en el conjunto de prueba.
4. *Determinación de la proporción de no asistencia*. Se calcula la cantidad de pacientes que no asistieron y que asistieron necesarios para mantener la misma proporción en el conjunto de prueba como en el conjunto de datos completo, para cada hipótesis. Esto asegura que la distribución de asistencia y no asistencia se mantenga en el conjunto de prueba.

```
# Cálculo del número de citas NoShow = True que se requieren para mantener la estratificación de Clases
NUMBER_NoShows_inTest = int(NUMBER_Apps_inTest * (NUMBER_NO_SHOWS / NUMBER_SAMPLES))
NUMBER_Shows_inTest = int(NUMBER_Apps_inTest - NUMBER_NoShows_inTest)
print(f"Se requieren {NUMBER_NoShows_inTest} pacientes con 'NoShow' = True en el Set de Prueba, y\n \
{NUMBER_Shows_inTest} pacientes con 'NoShow' = False.\n")
```

5. *Selección de citas para el conjunto de prueba*. Se seleccionan las últimas citas de pacientes que no asistieron y asistieron según el número calculado anteriormente, para cada hipótesis. Estas citas se utilizarán para formar el conjunto de datos de prueba final.

```

# Selección de las últimas 4463 muestras del sub-dataset 'last_app_byPatient' con 'NoShow' = True
test_set_NoShows = last_app_byPatient[last_app_byPatient['NoShow'] == True][- (NUMBER_NoShows_inTest+1):-1]

# Selección de las últimas 17642 muestras del sub-dataset 'last_app_byPatient' con 'NoShow' = False
test_set_Shows = last_app_byPatient[last_app_byPatient['NoShow'] == False][- (NUMBER_Shows_inTest+1):-1]

```

6. *Filtrado del conjunto de datos original:* Después de haber seleccionado las citas necesarias para el conjunto de prueba, se utilizan esas citas para filtrar el conjunto de datos original y crear tanto el conjunto de prueba como el conjunto de entrenamiento definitivos, para cada hipótesis. Esto garantiza que cada conjunto de prueba final contenga exactamente las citas necesarias, mientras que cada conjunto de entrenamiento final contenga todas las demás citas que no están en el conjunto de prueba.

## Entrenamiento

Se procede a entrenar para cada una de las distintas bases de datos 3 modelos de machine learning: regresión logística, árbol de decisión, en ambos casos utilizando la librería Sklearn de Python y redes neuronales utilizando la librería Pytorch entrenadas utilizando GPU. Se desestima de momento la utilización de otros modelos de machine learning debido a los tiempos de entrenamiento en el hardware que disponemos. A continuación, se detalla el entrenamiento de cada modelo:

### Regresión logística

- Se procede a realizar una búsqueda de mejores hiperparámetros utilizando búsqueda en Cuadrícula (Grid Search): Se utiliza GridSearchCV para explorar un espacio de hiperparámetros:

```

# Se define el rango de hiperparámetros para la búsqueda en cuadrícula
param_grid = {
    'C': [0.01, 0.1, 1, 10, 100],
    'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'],
    'max_iter': [100, 200, 300, 500, 1000]
}

```

- Posteriormente se realiza validación Cruzada (K-Fold Cross-Validation): Se utiliza un KFold con 5 particiones (n\_splits=5) y se barajan los datos (shuffle=True) para asegurar una evaluación robusta del modelo. Durante cada iteración del K-Fold, se ajusta el modelo y se evalúa el rendimiento en un conjunto de validación. El modelo con la mejor puntuación de validación se selecciona como el mejor, se guarda y se utiliza en el conjunto de test.

### Árbol de decisión

- Al igual que en la regresión logística se procede a realizar una búsqueda de mejores hiperparámetros utilizando búsqueda en Cuadrícula (Grid Search): Se utiliza GridSearchCV para explorar un espacio de hiperparámetros:

```

# Se define el rango de hiperparámetros para la búsqueda en cuadricula
param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [10, 15, 20],
    'min_samples_split': [2, 3, 5],
    'min_samples_leaf': [2, 3, 4]
}

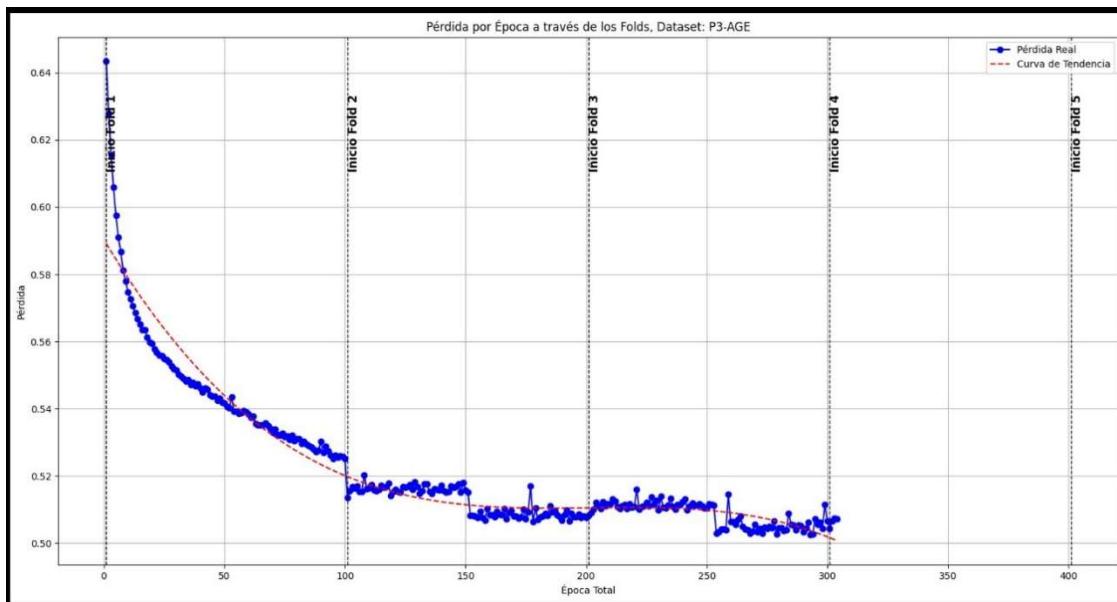
```

- Se repite el mismo proceso de validación Cruzada (K-Fold Cross-Validation): Se utiliza un KFold con 5 particiones (`n_splits=5`), se barajan los datos (`shuffle=True`) y se evalúa el rendimiento en un conjunto de validación. El modelo con la mejor puntuación se selecciona, se guarda y se utiliza en el conjunto de test.

#### Red neuronal

- **Definición:** Se define una clase `NeuralNet` que hereda de `nn.Module`. La red consta de:
  - **Capas lineales (nn.Linear):** Estas capas son responsables de las transformaciones lineales de los datos de entrada, aprendiendo pesos y sesgos que se ajustan durante el entrenamiento.
  - **Normalización por lotes (nn.BatchNorm1d):** Reduce el desplazamiento interno de covariables, haciendo el entrenamiento más eficiente.
  - **Funciones de activación ReLU (nn.ReLU):** Introducen no linealidades en el modelo, permitiendo que la red neuronal aprenda relaciones complejas en los datos.
  - **Función de activación Sigmoid (nn.Sigmoid):** Se utiliza en la capa de salida para transformar las salidas en probabilidades, es útil ya que es un problema de clasificación binaria.
  - **Capa de Dropout (nn.Dropout):** desactiva aleatoriamente algunas neuronas durante el entrenamiento, ayuda a prevenir el sobreajuste.
- **Validación:** Se utiliza StratifiedKFold con 5 particiones (`n_splits=5`) y se barajan los datos (`shuffle=True`) para asegurar una evaluación robusta del modelo y se divide el conjunto de datos en pliegues estratificados para mantener la proporción de clases en cada pliegue. Para cada pliegue, se crean tensores de entrenamiento y validación. Se utilizan DataLoader para cargar los datos en mini-lotes (`batch_size=64`), lo que facilita el entrenamiento por lotes. En cada época, se entrena el modelo, se calcula la pérdida de entrenamiento y validación, y se ajusta el optimizador.
- **Entrenamiento:** Se establecen los parámetros de entrenamiento como:
  - Número de épocas (`num_epochs=100`)
  - Criterio de pérdida (`nn.BCELoss`), optimizador (`optim.Adam`) con `lr=1e-4, weight_decay=1e-5` util en problemas de clasificación binaria.

- Scheduler para ajustar la tasa de aprendizaje ReduceLROnPlateau con factor=0.5, patience=3. Lo que permite ajustar dinámicamente la tasa de aprendizaje, mejorando la convergencia del modelo.
- Se implementa el Early Stopping con patience=3 para detener el entrenamiento si no hay mejora en la pérdida de validación, evitando así el sobreajuste.



Se calculan las pérdidas de entrenamiento y validación en cada época, con los parámetros mencionados, como se puede observar en la imagen existe una curva descendente consistente en las perdidas hasta el 4to fold, donde Early Stopping detiene el entrenamiento para así evitar el sobreajuste.

Una vez completado el entrenamiento en todos los pliegues, se guarda el mejor modelo encontrado para ser utilizado con los datos de test.

## Evaluación de modelos

Una vez entrenados y guardados los mejores modelos se procede graficar los resultados y probar con los datos de test, se compararán y evaluarán las capacidades predictivas de cada modelo utilizando métricas como accuracy, precisión, recall, F1-score y curva ROC. Para la regresión logística, árbol de decisión y red neuronal respectivamente. Obteniendo los siguientes resultados para cada dataset:

Dataset	Model	Test Accuracy	Test Precision	Test Recall	Test F1 Score	Test AUC
P1-ALL-ADASYN	Logistic Regression	64,57%	30,70%	60,03%	40,62%	0,66
	Decision Tree	65,04%	28,69%	49,25%	36,26%	0,64
	Neural Network	58,78%	30,11%	78,82%	43,57%	0,72
P1-ALL	Logistic Regression	67,72%	32,05%	53,44%	40,07%	0,66
	Decision Tree	71,05%	31,14%	35,84%	33,33%	0,63
	Neural Network	66,55%	32,37%	60,28%	42,12%	0,72
P2-NOCONDITIONS	Logistic Regression	67,45%	32,74%	52,71%	40,40%	0,66
	Decision Tree	72,47%	34,09%	33,84%	33,97%	0,63
	Neural Network	68,98%	34,85%	55,53%	42,82%	0,72
P3-AGE	Logistic Regression	68,93%	39,76%	52,17%	45,13%	0,69
	Decision Tree	69,15%	38,10%	41,57%	39,76%	0,62
	Neural Network	68,67%	39,44%	52,23%	44,95%	0,72
P4-TIME	Logistic Regression	67,37%	38,86%	25,16%	30,54%	0,59
	Decision Tree	65,12%	31,77%	19,45%	24,13%	0,52
	Neural Network	69,95%	42,29%	14,70%	21,82%	0,58
P5-HEALTHCENTRE	Logistic Regression	69,19%	33,12%	50,64%	40,04%	0,67
	Decision Tree	71,69%	33,73%	40,81%	36,93%	0,64
	Neural Network	68,08%	34,01%	60,75%	43,61%	0,72

Se adjuntan los anexos con las respectivas matrices de confusión, y curvas ROC para cada entrenamiento y test de cada uno de los modelos analizados:

- [Anexo 1 resultados P1-All-ADASYN](#)
- [Anexo 2 resultados P1-All](#)
- [Anexo 3 resultados P2 -NOCONDITIONS](#)
- [Anexo 4 resultados P3-AGE](#)
- [Anexo 5 resultados P4-TIME](#)
- [Anexo 6 resultados P5-Healthcentre](#)

#### Análisis de resultados

Se observa que todos los modelos logran un AUC > 0.5 por lo que son superiores al azar en la predicción, en particular, llama la atención que los modelos de redes neuronales alcanzan consistentemente un valor de AUC de 0.72 para todos los dataset a excepción de P4 time.

Actualmente, como se observa en la imagen la optimización está enfocada en maximizar la accuracy. Sin embargo, dependiendo de los objetivos específicos del proyecto, se podría cambiar el enfoque del grid search para mejorar otras métricas como el recall o la precisión, por ejemplo:

- **Optimizando la Precisión:** Minimizando así el número de pacientes que predecimos no asistirán, pero que finalmente sí lo hacen, lo que genera overbooking en la Fase 2 del proyecto. Es decir, minimizamos el overbooking real generado.
- **Optimizando el Recall:** Minimiza el número de pacientes que predecimos que asistirán, pero que finalmente no lo hacen, generando huecos libres en las agendas médicas. Esto ayuda a minimizar las pérdidas por inactividad en la Fase 2 del proyecto.

## Fase 2. Implementación de un sistema de overbooking

### Estudio de Bibliografía relacionada

La extensa bibliografía desarrollada en relación con los Sistemas “System Appointment Scheduling” (SAS) es un buen indicador de dificultad para resolver el problema, como también de la importancia que tiene mejorar el servicio en atención médica y disminuir costos asociados.

En “A Review of Optimization Studies for System Appointment Scheduling” (11) se definen las características principales de un SAS, destacando que, en los servicios ambulatorios, una cuestión central para la programación de las citas es cómo asignar los espacios de tiempo disponibles para los pacientes, reduciendo las demoras de los pacientes y la disponibilidad de los médicos o el tiempo adicional.

Son muchos los factores que complican la resolución real del problema, empezando por la selección del marco en el que se toman las decisiones para optimizar las citas, según el cual tomaremos 3 tipos de decisiones:

- Estratégicas: decisiones a largo plazo para determinar cómo se va a permitir el acceso de los pacientes a los servicios médicos. Si van a entrar todos bajo cita previa, o se van a permitir pacientes el mismo día, o algo híbrido.
- Tácticas: decisiones más a medio plazo para determinar cómo se utilizan los servicios, si se permite la elección de médico, si se otorga preferencia a determinados grupos, etc.
- Operativas: son las decisiones a corto plazo que hay que tomar a la hora de asignar las citas. Si éstas se toman con enfoques basados en reglas bien definidas (son las usadas en los métodos tradicionales), o con enfoques más basados en la optimización de algún parámetro, como puede ser nuestro caso para la optimización del coste.

Así pues, la dificultad en la resolución del problema radica en la multitud de decisiones que se deben realizar para definir el problema de optimización estocástica, así como en el grado de incertidumbre que tienen los parámetros y variables que finalmente se deciden tener en cuenta.

A continuación, se resumen los parámetros que aportan mayor incertidumbre y variabilidad a los Sistemas SAS:

- **Incertidumbre en las predicciones de asistencia.** Los modelos de predicción están lejos de ser perfectos. Aunque los resultados son mejores que una mera asignación de probabilidad media de asistencia o una predicción al azar, las métricas de precisión y sensibilidad siguen proporcionando un grado de incertidumbre bastante elevado, y que será automáticamente trasladado al modelo de optimización.
- **Variedad e incertidumbre en los factores ambientales.** El número de factores ambientales que se pueden aplicar al sistema son muy numerosos, y seguramente diferentes para cada centro hospitalario o de salud, pues

dependen de las decisiones estratégicas y tácticas mencionadas más arriba. Adicional, muchos de ellos añaden unos valores de incertidumbre que ni siquiera están estudiados por modelos predictivos. Así pues, podemos distinguir entre factores más certeros (número de servicios entregados, número de doctores, número de citas por sesión, prioridad en la atención) y más inciertos (puntualidades, tiempos de atención, nivel de interrupción de los doctores, llegadas espontáneas).

- **Variedad en las reglas de cita y en reglas de secuencia.** Las reglas de cita definen las restricciones que imponemos al modelo para limitar las asignaciones (número de pacientes que se puede asignar a un único slot, duración de los slots, etc.), y las reglas de secuencia definen el orden con los que los pacientes son asignados a los slots en función de una determinada clasificación (prioritarios, primerizos, recurrentes, etc.). Forman parte de las decisiones operativas mencionadas por Tiantian et al. (11). Sólo Cayirli, Veral y Rosen (12) ya realizaron un estudio probando 6 diferentes reglas de secuencia con 7 reglas de cita, lo que hace un total de 42 Sistemas de Asignación (AS, del inglés) diferentes. Para hacernos otra idea de la magnitud y complejidad del problema, Ho y Lau (13) comparaban 9 diferentes sistemas de citas en 27 diferentes escenarios clínicos caracterizados por 3 factores ambientales (probabilidad de ausencias, variación de tiempos de atención y número de pacientes por sesión).
- **Definición del Coste.** Para minimizar la Función de Coste que permite realizar la mejor programación de citas médicas se tiene que definir primero cuál es dicho coste y cómo calcularlo, lo cual no es banal. Y como se indica a continuación, las posibilidades son prácticamente infinitas.

Para empezar, según Tiantian et al. (11) la optimización varía en función del objetivo que se pretenda conseguir: beneficio social, rentabilidad económica de la clínica, máxima amortización de recursos, etc. Así pues, Chew (14) y Kaandorp and Koole (15) dividen la función en 3 tipologías de coste distintas: el tiempo de espera del paciente en consulta, el tiempo de inactividad de toda la infraestructura médica por falta de paciente, y el tiempo extra dedicado por la infraestructura médica a atender los pacientes después de una jornada laboral normal. Ho y Lau (13) realizan de forma similar, pero sin tener en cuenta el tiempo extra de los médicos. Y Harris and Samorani (16) utilizan los mismos conceptos, pero sin tener en cuenta el tiempo de inactividad de la infraestructura médica. Con otra perspectiva, Almактоом (17) calcula el coste respecto a los costos de realizar una consulta, los de operación de la clínica y los de overbooking, referenciándolos al número de No-Shows y de Overbookings. Sin embargo, en Valenzuela-Núñez et al. (18) no consideran minimizar un coste, sino maximizar la utilidad que genera el centro sanitario por cada cita atendida, menos una penalidad por sobrecarga de overbooking, directamente proporcional a los casos de overbooking registrados. Y Lawley and Muthuraman (19) usaron un modelo estocástico para minimizar el número de pacientes desbordados de una cita a la siguiente.

### **Elección de las hipótesis para la resolución del problema y fórmula para el cálculo del Coste del Sistema a minimizar**

Para definir el Problema de Optimización Estocástico se tiene que decidir sobre:

- La Función Objetivo: lo que se quiere optimizar, relacionado con los marcos de optimización mencionados por Tiantian et al. (11).
- Las Variables de Decisión: aquello que se pretende ajustar en el modelo. En nuestro caso, la distribución de pacientes en los slots que dura una consulta médica.
- Las Restricciones: limitaciones con las que se restringen las posibles soluciones. Estas restricciones pueden ser el número de pacientes esperando, la capacidad máxima de pacientes que se puede atender en un día, o el horario máximo de atención, por ejemplo. Y formarán parte de las hipótesis a fijar en este apartado.
- Las Variables Aleatorias: los elementos que están sujetos a variabilidad y que se modulan mediante distribuciones de probabilidad.

En nuestro caso, en aras de simplificar la resolución del problema, minimizamos las variables del modelo como sigue:

○ **Factores ambientales:**

- Predicciones de Asistencia. Extraídas de la Fase 1.
- Tiempo de Atención por Paciente = 20 min. Igual para todos los pacientes sin excepción.
- Horario de Atención normal. De 9:00 a 13:00 y de 15:00 a 19:00 (8 horas, separadas en 2 sesiones de 4 horas).
- Número de slots por sesión = 24 (consideradas las consultas de veinte minutos en una jornada normal de 8 horas).
- El resto de los factores los anulamos directamente: presuponemos que no hay impuntualidad, que no hay llegada espontánea de pacientes, un único servicio integrado, etc.

○ **Reglas de citas:**

- Utilizamos distintas reglas de citas en función del escenario planteado para el cálculo del Coste del Sistema:
  - Escenario 1 – Tradicional sin Overbooking. Sólo se permite un paciente por slot, sin posibilidad de espera por parte de los pacientes ni tiempos de atención extra, pero sí mucho coste por inactividad de los servicios médicos. Esto equivale a una asignación de 24 pacientes (24 slots) por servicio médico y día.
  - Escenario 2 – Tradicional con Overbooking. Considerando la probabilidad promedio de No Show del 20,19% en nuestro dataset, se decide programar más pacientes a la consulta

médica, a razón de 1,2019 pacientes extra, lo que equivale a una asignación de 30 pacientes por servicio médico y día, distribuidos entre 24 slots, lo que genera posible overbooking. La regla de asignación será 1 slot con 2 pacientes, 3 slots con un paciente, y así sucesivamente hasta completar la asignación de los 30 pacientes.

- Escenario 3 – Modelo ML para Optimización de Overbooking: en este caso, la Regla de Cita, es decir, el número de pacientes a solapar en cada slot, la determinará directamente el modelo optimizador entrenado. Serán los parámetros aprendidos que minimicen la función de coste.

- **Reglas de secuencia:**

- Utilizaremos una secuencia FCFA (First Call First Appointment), también conocida directamente como Regla de No Secuencia, pues simplemente se asignará al paciente en el mejor slot disponible calculado mediante el modelo entrenado de overbooking (escenario 3), o según la Regla de Cita definida en el punto anterior para los escenarios 1 y 2.

- **Función de coste:**

- Utilizaremos una fórmula parecida a la usada por Chew (14), dividiendo y calculando la función de coste en 3 términos independientes:
  - $T_w$  = Tiempo de espera del paciente (Wait Cost). Calculado como unidad temporal que debe esperar un paciente a ser atendido por encontrarse el slot sobrecargado. Si tiene no tiene que esperar será igual a cero, si tiene que esperar un slot será igual a 1, y así sucesivamente. (unidad temporal = 20 min = 1 slot).
  - $T_o$  = Tiempo extra a realizar por el doctor (Overtime Cost). Calculado como unidad temporal que tiene que trabajar fuera de la sesión o jornada laboral normal de 8 horas. Si no tiene que trabajar ningún slot será igual a cero, si tiene que trabajar un slot será igual a 1, y así sucesivamente. (unidad temporal = 20 min = 1 slot).
  - $T_i$  = Tiempo de inactividad en la clínica (Idle Cost). Calculado como unidad temporal sin estar atendiendo a ningún paciente. Si el doctor está siempre ocupado será igual a cero, si queda un slot vacío será igual a 1, y así sucesivamente. (unidad temporal = 20 min = 1 slot).
- A cada término independiente le daremos pesos distintos en función de cuál sea el coste que más queremos penalizar, en el entendido de que hay valores muy subjetivos en el cálculo de dichos pesos, como

pueden ser el coste que tiene para un paciente la hora de espera, o para un doctor la hora extra de trabajo extra.

### Cálculo del Coste del Sistema en métodos tradicionales de asignación de citas médicas

Desarrollo pendiente, versión preliminar programada para la entrega 3 de desarrollo del proyecto

### **Fase 3. Creación de un Asistente Virtual basado en el Procesamiento de Lenguaje Natural (NLP) que permita el contacto entre Centros de Salud y Pacientes para la gestión de citas y recordatorios.**

En la creación de un chatbot para la asignación, cambio y / o cancelación de citas médicas, ya hemos mencionado sus ventajas, entre otras la minimización de errores humanos, la facilidad de obtención de citas en cualquier momento, etc.

También hemos mencionado la necesidad de tener en cuenta las interfases para el acceso e interacción con el software médico.

En los siguientes párrafos se registrarán los siguientes ítems: Desarrollo del Chatbot, desarrollo de medidas de seguridad, integración del Chatbot con el software médico, y finalmente la integración del Chatbot con un sitio Web.

#### **Desarrollo del Chatbot**

1. **Análisis de Requisitos:** Entender a profundidad las necesidades del sistema y definir casos de uso.
2. **Diseño del Chatbot:** Crear diagramas de flujo de conversación y estructuras de datos.
3. **Desarrollo del Backend:** Configurar servidores, bases de datos y APIs.
4. **Desarrollo del Frontend:** Crear interfaces de usuario en aplicaciones web y móviles.
5. **Integración del Chatbot:** Configurar plataformas como Dialogflow o Microsoft Bot Framework.
6. **Pruebas y Validación:** Realizar pruebas unitarias, de integración y de usuario final.
7. **Despliegue:** Implementar el sistema en producción y realizar monitoreo continuo.

#### **Medidas de seguridad.**

De otro lado como también hemos mencionado se deben tener en cuenta medidas de seguridad que se definirían teniendo en cuenta:

1. **Autenticación y Autorización:** Uso de tokens JWT, OAuth.
2. **Cifrado de Datos:** SSL/TLS para la transmisión de datos y cifrado en reposo para bases de datos.
3. **Seguridad en APIs:** Protección contra ataques CSRF, XSS, SQL Injection.

4. **Auditoría y Monitoreo:** Registro de todas las actividades del sistema para auditorías y detección de anomalías.

5. **Cumplimiento de Normativas:** Asegurar que el sistema cumple con normativas como GDPR, HIPAA.

### **Integración del Chatbot con software asistencial**

Y en cuanto a la integración del Chatbot con el software asistencial tendríamos:

1. **Mapeo de Procesos:** Identificar cómo el chatbot interactuará con el software existente.

2. **Desarrollo de APIs:** Crear endpoints para que el chatbot pueda realizar operaciones de lectura/escritura.

3. **Pruebas de Integración:** Asegurarse de que las interacciones del chatbot con el software asistencial funcionan correctamente.

4. **Monitoreo Continuo:** Establecer sistemas de monitoreo para detectar y resolver problemas rápidamente.

### **Integración del Chatbot con un sitio Web**

Una fase muy importante, que debemos mencionar es la integración del Chatbot con un sitio Web.

1. **Desarrollo del Widget del Chatbot:** Crear un widget embebible para el sitio web.

2. **Integración en el Sitio Web:** Incluir el widget en las páginas relevantes del sitio.

3. **Pruebas de Funcionamiento:** Realizar pruebas para asegurar que el chatbot funciona correctamente dentro del sitio web.

4. **Monitoreo y Mejora Continua:** Monitorear el uso del chatbot en el sitio web y hacer mejoras según sea necesario.

Todo lo anterior se está definiendo para realizar los respectivos cronogramas.

## **Fase 4. Proceso de refinamiento del sistema**

Desarrollo pendiente, versión preliminar programada para la entrega 3 de desarrollo del proyecto

### **Conclusiones**

Desarrollo pendiente, versión preliminar programada para la entrega 3 de desarrollo del proyecto

# Bibliografía

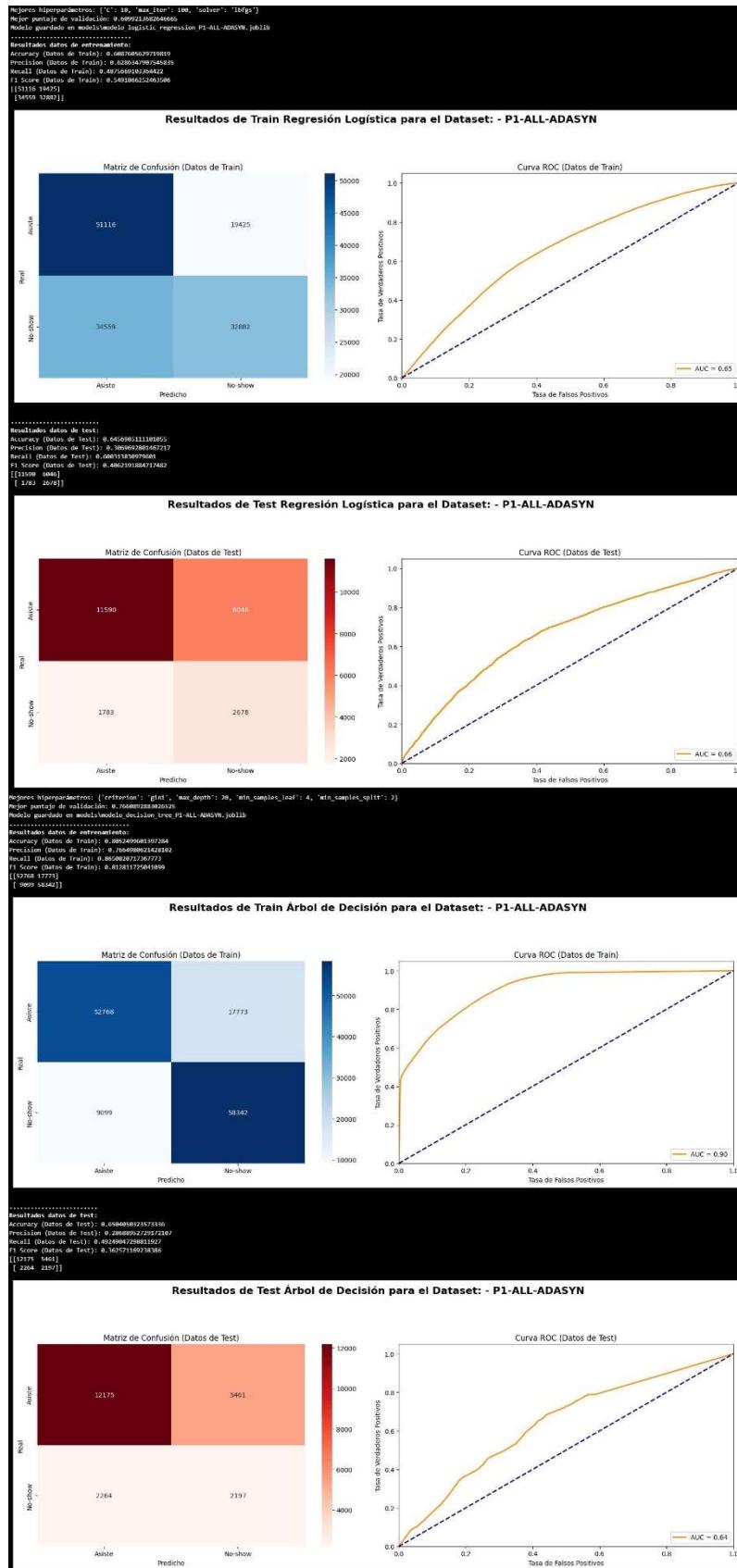
1. **Ministerio de Sanidad, Consumo y Bienestar Social, Gobierno de España.** *Informe Anual del Sistema Nacional de Salud.* 2022.
2. **Ministerio de Salud, Gobierno de Chile.** *Lista de Espera No Ges y Garantías de Oportunidad GES retrasadas. Glosa 06. IV Trimestre.* 2022.
3. **Danke, Karen, y otros.** *Estudio de brechas de médicos y odontólogos generales y especialistas en el sector público de salud. Período 2020-2030.* 2020.
4. **Rico, Juan Pablo.** Inasistencia horas médicas: La oportunidad para implementar un modelo de atención digital. [En línea] 2023.  
<https://tierramarillano.cl/2023/06/27/inasistencia-horas-medicas-la-oportunidad-para-implementar-un-modelo-de-atencion-digital/>.
5. **Cruz, Martín.** Cómo el uso de tecnología ha permitido disminuir el “no show” de pacientes a sus citas médicas. [En línea] 2024. <https://tekiosmag.com/2024/01/26/como-el-uso-de-tecnologia-ha-permitido-disminuir-el-no-show-de-pacientes-a-sus-citas-medicas/>.
6. **Centro Investigaciones Sociológicas (CIS).** *Estudio nº3426. Barómetro Sanitario 2023 (tercera oleada).* 2023.
7. **JoniHoppen.** Kaagle. [En línea]  
<https://www.kaggle.com/datasets/joniarroba/noshowappointments>.
8. **Alanwillms.** GitHub : geoinfo. [En línea] <https://github.com/alanwillms/geoinfo>.
9. **Vitória, Prefeitura de.** CIDADÃO: SERVIÇOS PARA A PESSOA IDOSA. *Prefeitura de Vitória.* [En línea] [https://www.vitoria.es.gov.br/cidadao/servicos-para-a-pessoa-idosa#a\\_listaunidadesdesaude](https://www.vitoria.es.gov.br/cidadao/servicos-para-a-pessoa-idosa#a_listaunidadesdesaude).
10. **Vitória Weather In May 2016.** *Weather and Climate.* [En línea] Mayo de 2016. <https://weatherandclimate.com/brazil/espirito-santo/vitoria/may-2016>.
11. **A Review of Optimization Studies for System.** Tiantian Niu, Bingyin Lei, Li Guo, Shu Fang, Qihang Li, Bingrui Gao, Li Yang and Kaiye Gao. 16, s.l. : Axioms, 2023, Vol. 13.
12. **Designing Appointment Scheduling Systems for Ambulatory Care Services.** Cayirli, Tugba, Veral, Emre A. y Rosen, Harry. s.l. : Health Care Manage Sci, 2005, Vol. 9.
13. **Minimizing Total Cost in Scheduling Outpatient Appointments.** Chrwan-Jyh, Ho y Hon-Shiang, Lau. 12, s.l. : Management Science, 1992, Vol. 38.
14. **Outpatient Appointment Scheduling with.** Chew, Song Foh. s.l. : Hindawi Publishing Corporation, 2011, Vol. 2011.
15. **Optimal outpatient appointment scheduling.** Koole, Guido C. Kaandorp and Ger. 10, s.l. : VU University Amsterdam, 2007, Vol. Health Care Management Science.
16. **On Selecting a Probabilistic Classifier for Appointment No-show Prediction.** Samorani, Shannon L. Harris and Michele.
17. **Health care overbooking cost minimization model.** Almaktoom, Abdulaziz T. s.l. : Heliyon, 2023, Vol. 9.
18. **Smart Medical Appointment Scheduling: Optimization, Machine Learning, and Overbooking to Enhance Resource Utilization.** Valenzuela-Núñez, Catalina,

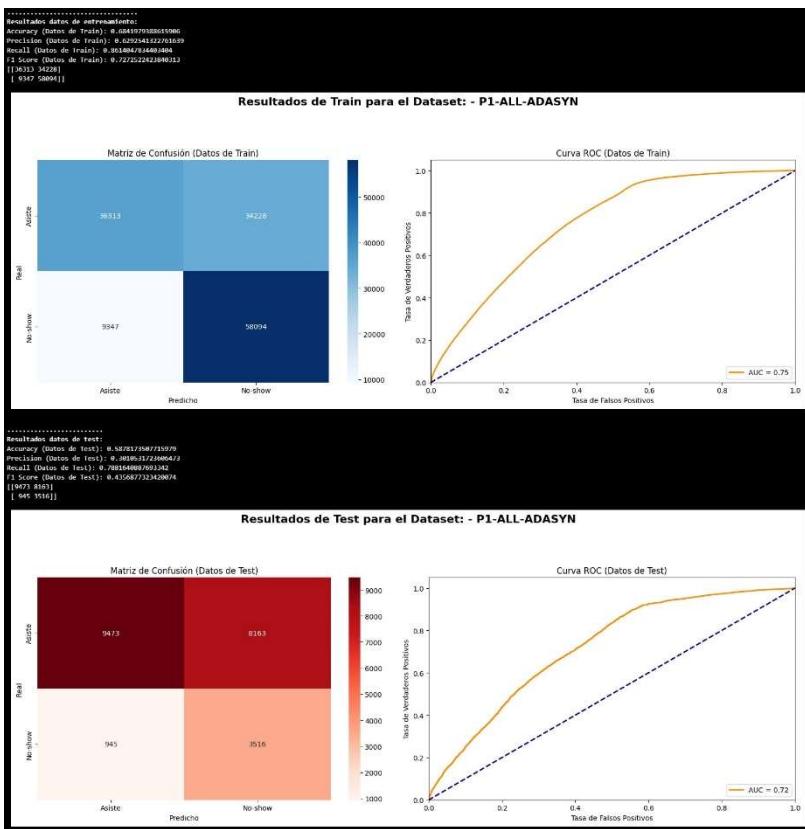
**Latorre-Núñez, Guillermo y Troncoso Espinosa, Fredy.** s.l. : IEEE Acces, 2024,  
Vol. 12.

**19. A stochastic overbooking model for outpatient clinical scheduling with no-shows.**  
Muthuraman, Kumar y Lawley, Mark. s.l. : IIE Transactions, 2008, Vol. 40.

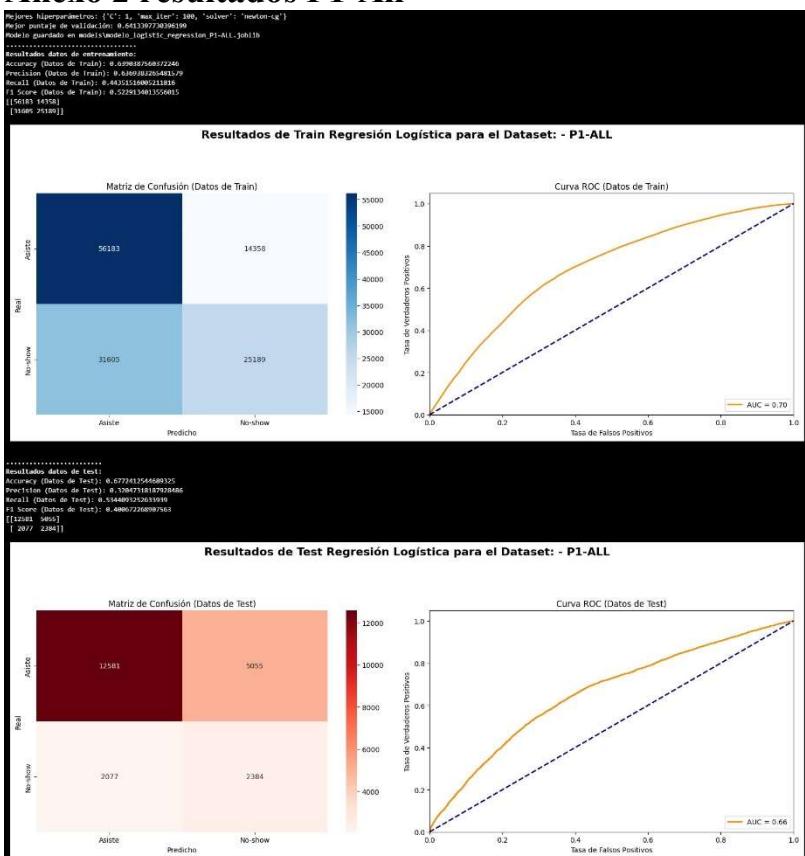
## **Anexos**

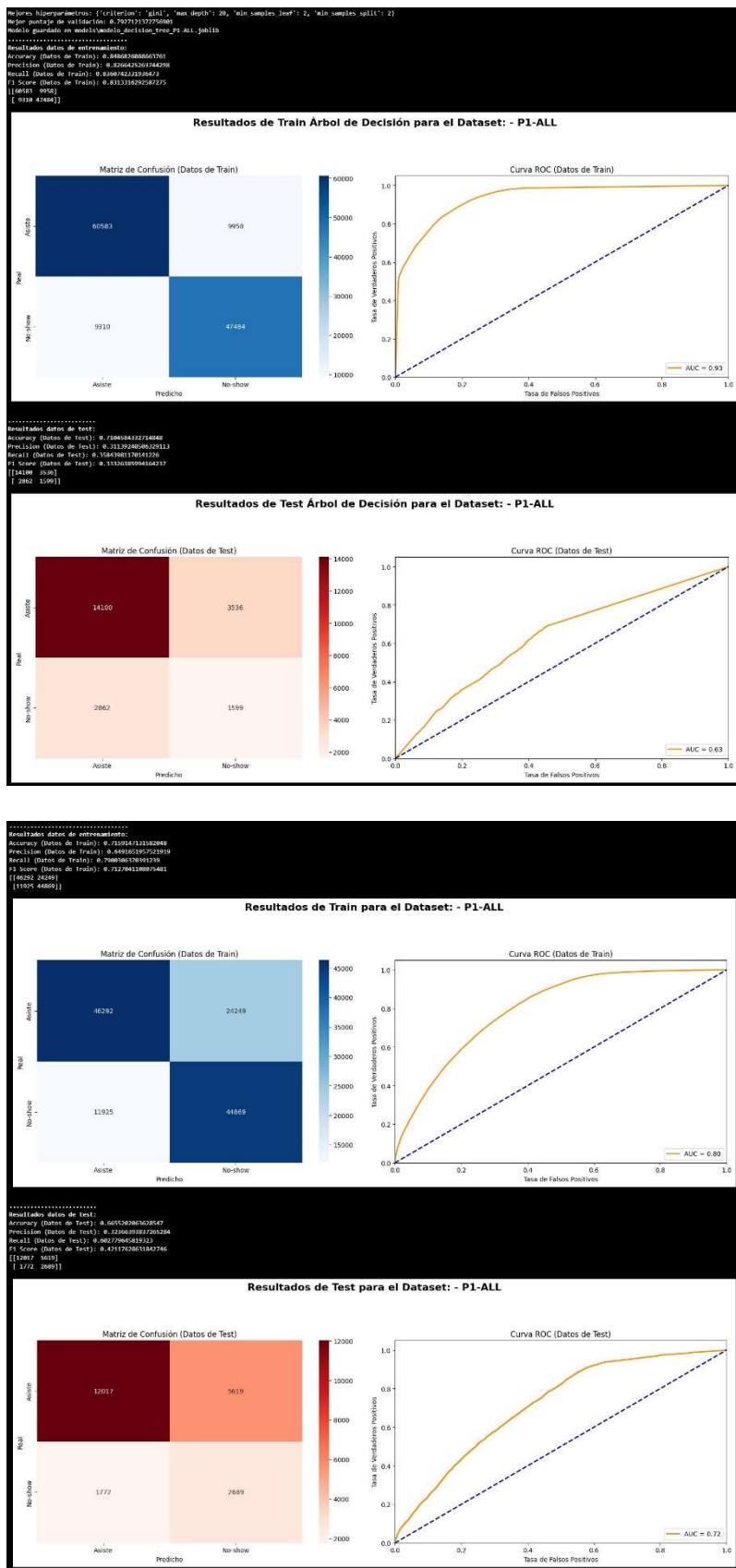
## Anexo 1 resultados P1-All-ADASYN



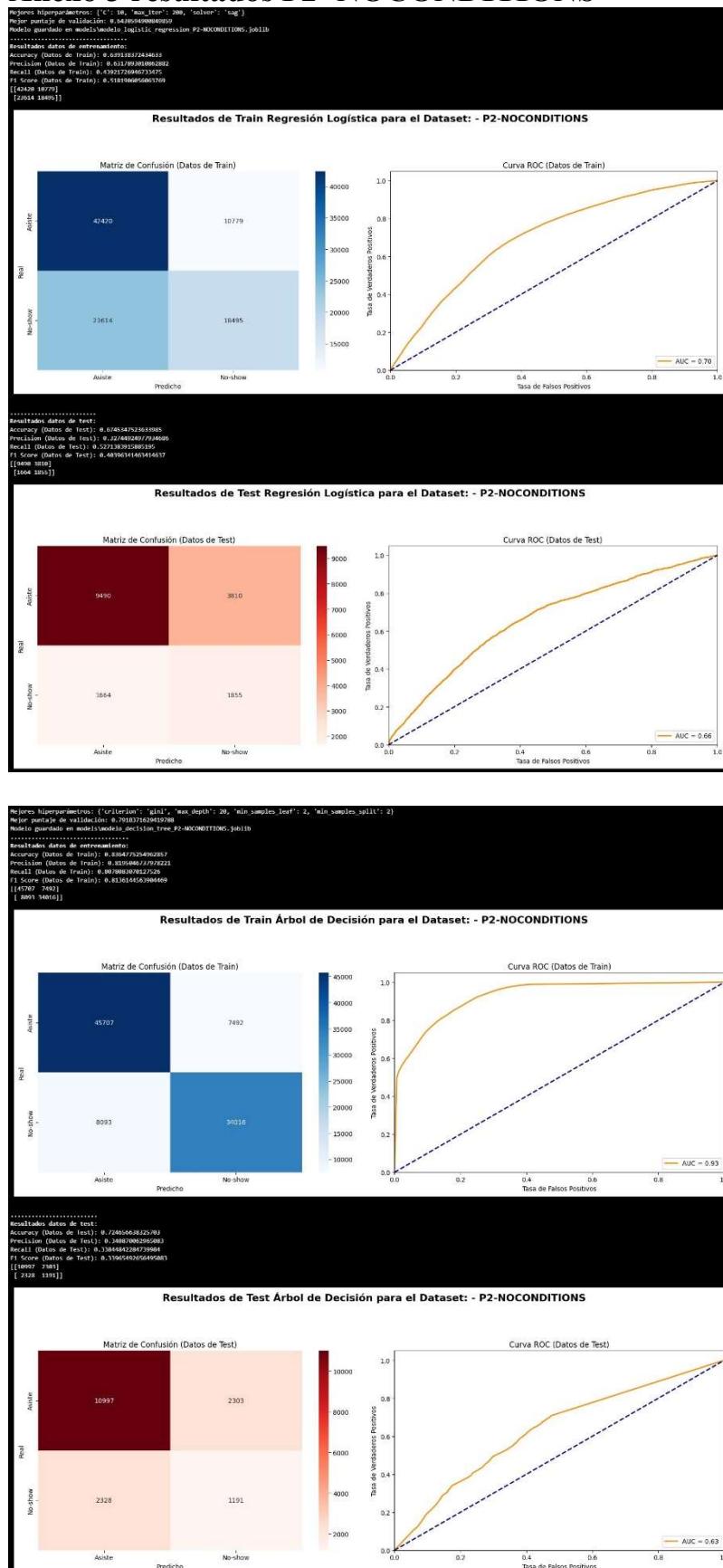


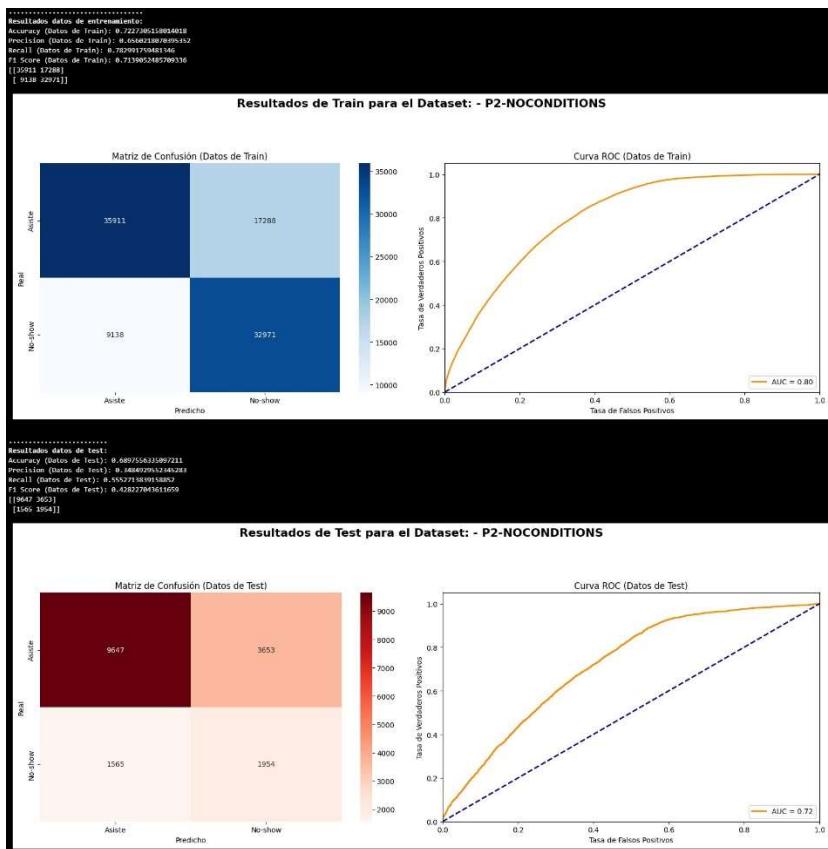
## Anexo 2 resultados P1-All



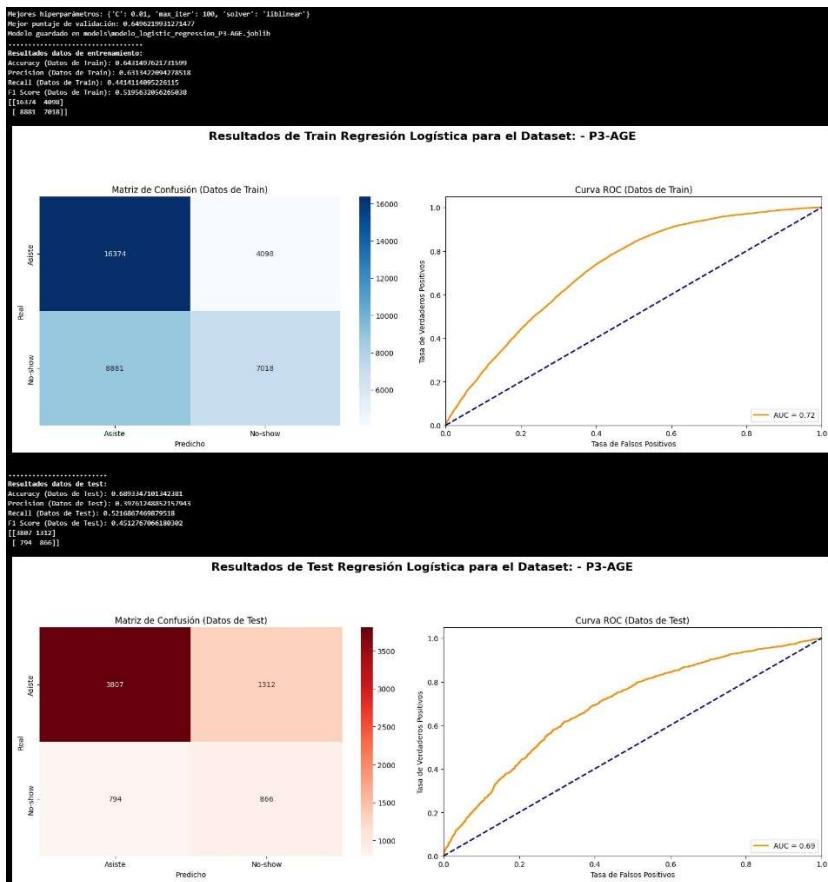


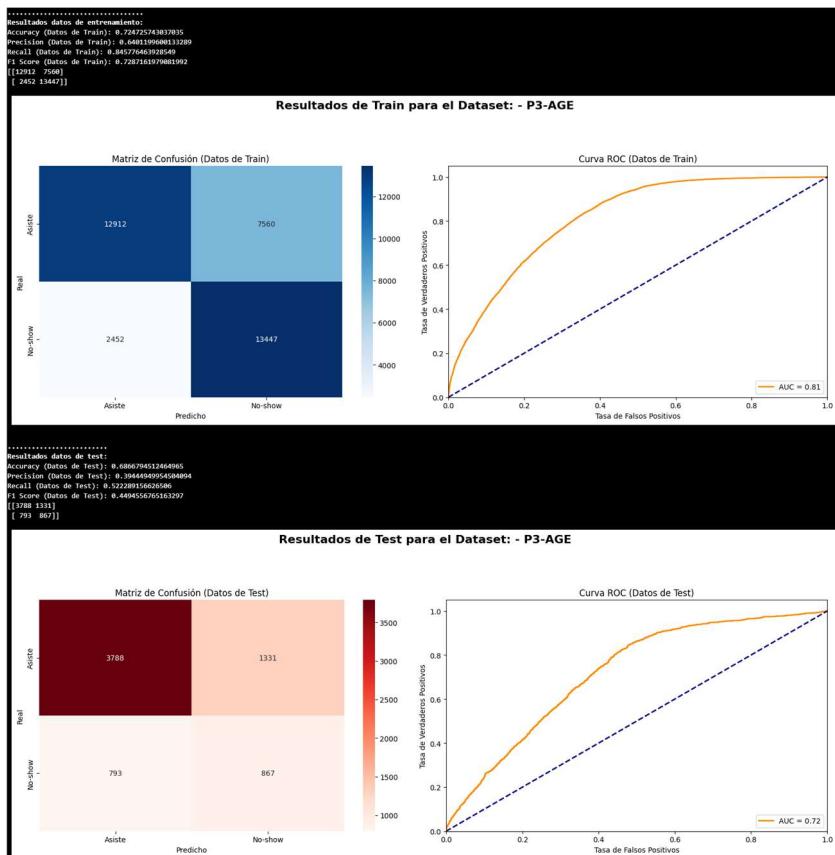
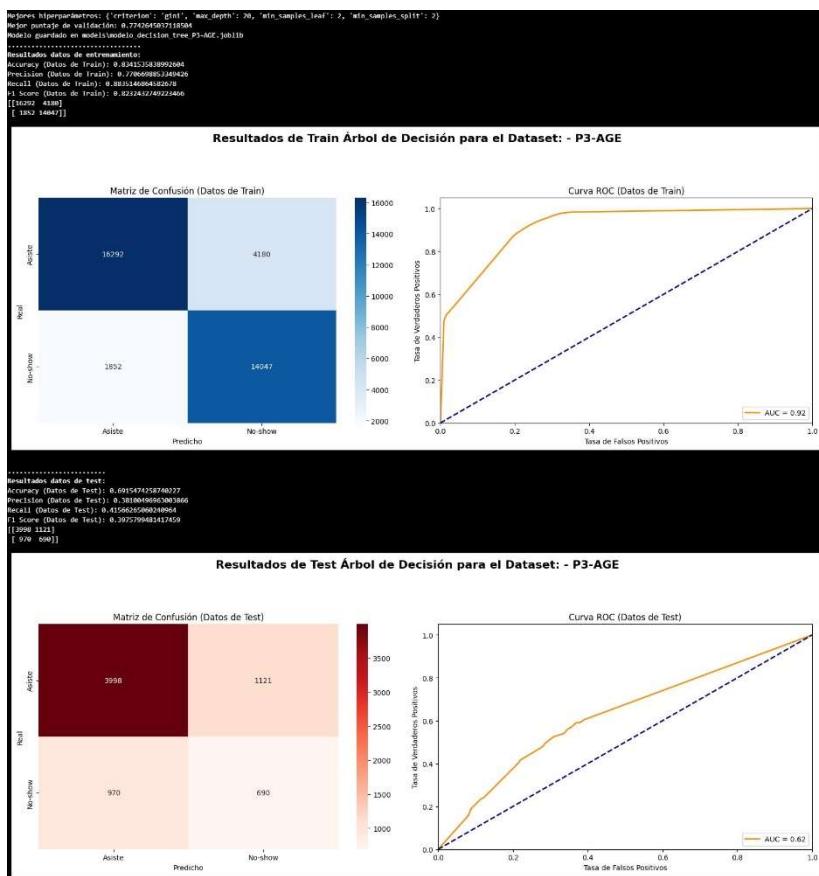
## Anexo 3 resultados P2 -NOCONDITIONS



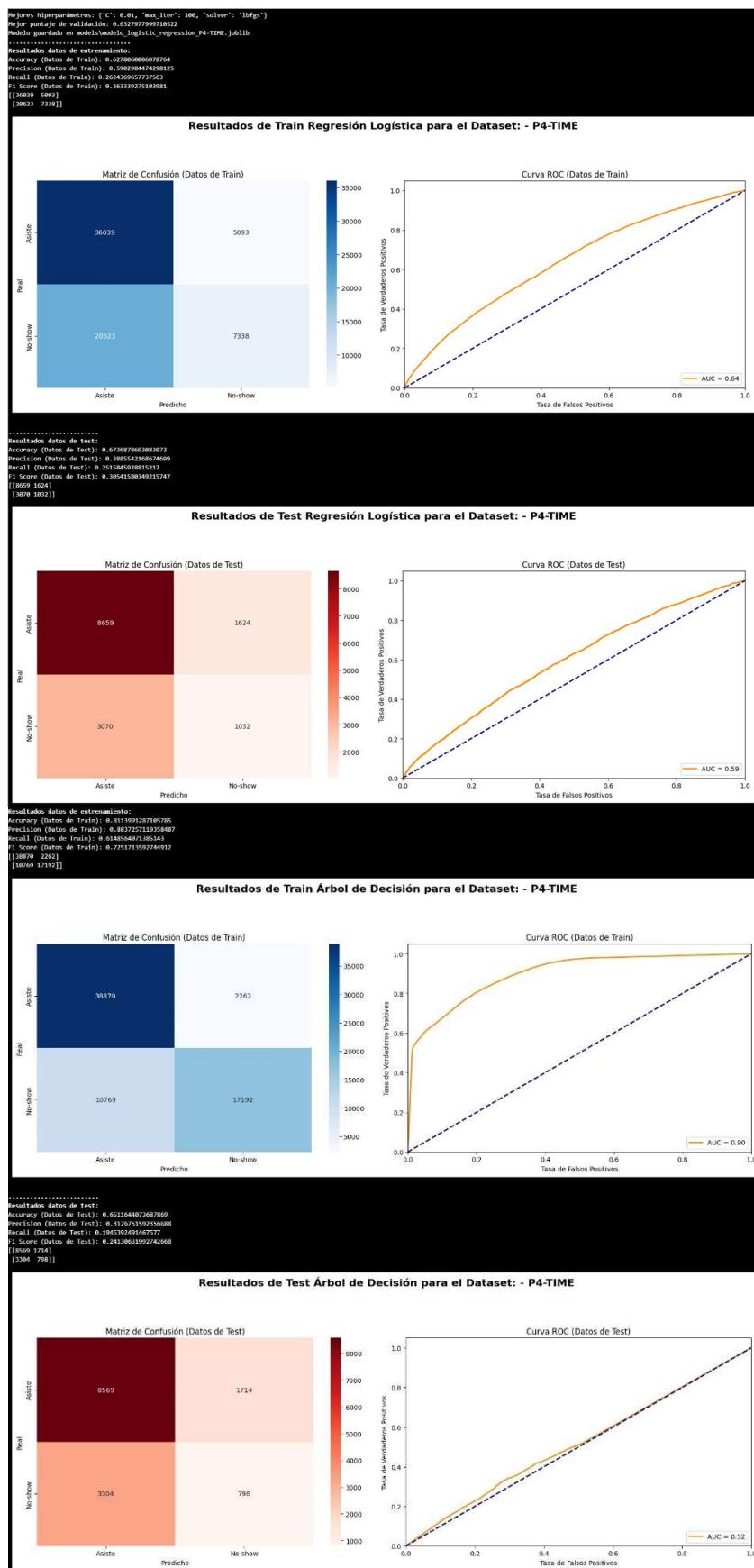


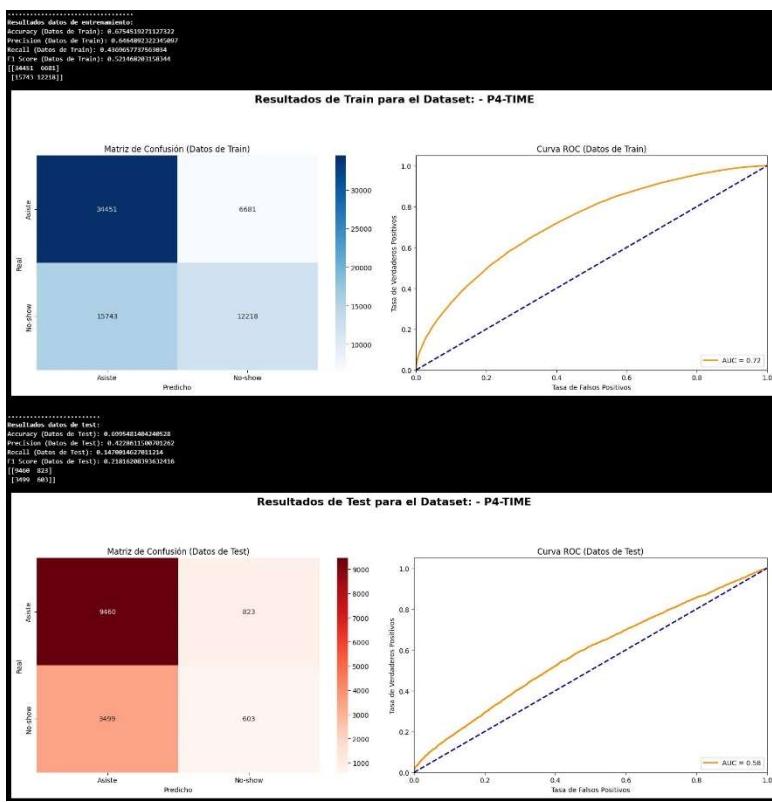
## Anexo 4 resultados P3-AGE





## **Anexo 5 resultados P4-TIME**





## Anexo 6 resultados P5-Healthcentre

