Instituto Tecnológico de Buenos Aires

Escuela de Ingeniería y Tecnología

Predicción de mortalidad específica a dos años y mutación del gen KRAS en metástasis hepática de cáncer colorrectal mediante Radiómica y Aprendizaje Automático

---

Autores:

Dujaut, Iván Maximiliano (Leg. N° 56278)

Esposito, Marco Iván (Leg N° 56419)

Tutora:

Bioing. Ricci Lara, María Agustina

Asesora:

Dra. Aineseder, Martina

Proyecto Final de Carrera

Título de Bioingeniería

Buenos Aires

28 de Abril de 2023

Agradecimientos

Queremos agradecer al Hospital Italiano de Buenos Aires y su Programa de Innovación Abierta por incluirnos en este proyecto. También agradecemos al servicio de Diagnóstico por Imágenes y a nuestra asesora la doctora Martina Aineseder, por proveernos de los estudios necesarios para llevar adelante la investigación y guiarnos en este trabajo. Asimismo, damos nuestras gracias al Departamento de Informática en Salud, al programa de Inteligencia Artificial en Salud del Hospital Italiano de Buenos Aires, y especialmente a nuestra tutora, la bioingeniería María Agustina Ricci Lara. Ella nos enseñó sobre la Inteligencia Artificial, nos acompañó durante el largo desarrollo de este proyecto, y puso tanto entusiasmo como nosotros para concretarlo.

También queremos agradecer a nuestras familias y amigos, que nos han acompañado por muchos años en este camino, desde nuestro ingreso a la universidad hasta ahora. Ellos conocen el tiempo y esfuerzo que nos ha demandado esta carrera, y comparten nuestra alegría de estar en esta etapa final.

Índice

Glosario de abreviaturas

2D: Dos Dimensiones.
3D: Tres Dimensiones.
ABC: *AdaBoost Classifier.*
AUC: Área bajo la curva (*Area Under Curve).*
BS: Brier Score.
DICOM: *Digital Imaging and Communication in Medicin*e.
DL: Aprendizaje Profundo (*Deep Learning)*
DT: Árbol de Decisión (*Decision Tree*).
ECE: Error de Calibración Esperado (*Expected Calibration Error*).
F1: Valor F1.
FA: Fase Arterial.
FSC: Fase Sin Contraste.
FVP: Fase Venosa Portal.
FVT: Fase Venosa Tardía.
GBC: *Gradient Boosting Classifier.*
GLCM: Matriz de Co-ocurrencia de Niveles de Gris (*Gray Level Co-occurrence Matrix*).
GLDM: Matriz de Dependencia de Niveles de Gris (*Gray Level Dependance Matrix*).
GLRLM: Matriz de Largo de Secuencia de Niveles de Gris (*Gray Level Run Length Matrix*).
GLSZM: Matriz de Tamaño de Zona de Nivel de Gris (*Gray Level Size Zone Matrix*).
GNB: *Gaussian Naive Bayes.*
HIBA: Hospital Italiano de Buenos Aires.
IA: Inteligencia Artificial.
IC: Intervalo de Confianza.
KNN: K Vecinos Cercanos (*K Nearest Neighbours*).
LDA: Análisis de Discriminante Lineal (*Linear Discriminant Analysis*)**.**
LdG: Laplaciano del Gaussiano.
LASSO: *Least Absolute Shrinkage and Selection Operator*.
MCC: Coeficiente de Correlación de Matthews (*Matthews Correlation Coefficient*).
MCE: Media de Error de Calibración (*Mean Calibration Error*).
MHCC: Metástasis Hepática de Cáncer Colorrectal.
ML: Aprendizaje Automático (*Machine Learning*).
RL: Regresión Logística.
RF: Bosque Aleatorio (*Random Forest*).
RM: Resonancia Magnética.
ROC: Característica Operativa del Receptor (*Receiver Operating Characteristic*).
ROI: Región De Interés (*Region Of Interest*).
SVM: Máquina de Vectores de Soporte (*Support Vector Machine*).
TC: Tomografía Computada.
UH: Unidad Hounsfield.
VOI: Volumen De Interés (*Volume Of Interest*).
VPN: Valor Predictivo Negativo.
VPP: Valor Predictivo Positivo, precisión.

Resumen

El cáncer colorrectal es la segunda causa de muerte por cáncer en el mundo, y suele estar acompañado por metástasis hepática. Entre posibles tratamientos, la metástasis puede ser eliminada por medio de una cirugía de resección, y se sabe que una mutación en el gen KRAS es indicador de un peor pronóstico. Se han publicado investigaciones prometedoras sobre el uso de Inteligencia Artificial como herramienta de soporte a la toma de decisiones en el diagnóstico y pronóstico de pacientes con este tipo de patologías.

Para este proyecto, se plantearon dos objetivos primarios, y múltiples secundarios. Como uno de los objetivos primarios, se definió entrenar y evaluar la capacidad diagnóstica de un clasificador basado en Radiómica e Inteligencia Artificial, para predecir la mortalidad específica a dos años en pacientes con metástasis hepática de cáncer colorrectal, utilizando imágenes de tomografía computada en fase venosa portal. Para este objetivo, el comienzo del período de dos años fue la cirugía de resección hepática como tratamiento a la enfermedad, y se definió la variable de respuesta **Óbito,** con dos posibles valores: cero o negativo, si el paciente vivía al finalizar el período, y uno o positivo, si la causa de muerte registrada fue la patología planteada. El otro objetivo primario consistió en entrenar y evaluar la capacidad diagnóstica de un clasificador basado en Radiómica e Inteligencia Artificial, empleando la misma modalidad y fase de imagen, pero para predecir el estado de mutación del gen KRAS. Para ello se definió la variable de respuesta **KRAS,** con dos valores posibles: cero o negativo para el estado Wild Type, y uno o positivo para el estado mutado, determinado por estudio genético.

Como objetivos secundarios, se planteó el entrenamiento y evaluación de seis clasificadores basados en Radiómica e Inteligencia Artificial, con imágenes de tomografía computada en las fases sin contraste, venosa tardía**,** y arterial, para la predicción de las variables Óbito y KRAS.

Se realizó un estudio de carácter retrospectivo, utilizando una base de datos de tomografías computadas conformada en el Hospital Italiano de Buenos Aires. Las imágenes fueron segmentadas por el departamento de Diagnóstico por Imágenes de la institución para identificar la lesión hepática de mayor tamaño y luego extraer las características radiómicas del volumen segmentado. La radiómica es un método de extracción de características cuantitativas de las imágenes, utilizadas como biomarcadores en medicina, y para el entrenamiento de algoritmos de Inteligencia Artificial.

Se aplicaron métodos de selección de características y se realizó un ajuste de hiperparámetros para optimizar los modelos clasificadores. Los mismos se evaluaron con métricas como el área bajo la curva de la característica operativa del receptor. El modelo final para cada variable se produjo tomando el promedio de los puntajes de salida de los mejores modelos seleccionados.

Mediante la combinación de filtros, métodos de selección de características y algoritmos de Inteligencia Artificial, se entrenaron más de 1.000 modelos entre todos los objetivos previstos. Para la variable Óbito, se utilizaron 101 casos en entrenamiento (64 vivos, 37 fallecidos) y 35 en evaluación (22 vivos, 13 fallecidos), resultando en un área bajo la curva de 0,875. Para la variable KRAS, se usaron 55 casos en entrenamiento (31 Wild Type, 24 mutados) y 30 en evaluación (17 Wild Type, 13 mutados), resultando en un área bajo la curva de 0,895.

Los modelos entrenados mostraron resultados prometedores en su evaluación y podrían constituir una herramienta no invasiva de soporte a la toma de decisiones en el tratamiento de metástasis hepática de cáncer colorrectal. Existe la posibilidad de desarrollar futuros trabajos para mejorar los resultados alcanzados y validar el uso de los predictores para su uso clínico.

1. Introducción

El cáncer colorrectal (**CCR**) es una enfermedad maligna que afecta el colon y el recto. Es el segundo en prevalencia y mortalidad en todo el mundo y en Argentina (con los primeros siendo el cáncer de mama en prevalencia y pulmón en mortalidad), según las cifras reportadas en 2020 por el *Global Cancer Observatory* de la Organización Mundial de la Salud [1]. Es curable por medio de cirugía y medicación si se lo detecta y trata de forma temprana, pero también tiene elevada recurrencia [2]. Las cifras de mortalidad a nivel global y en Argentina se muestran en la **Figura 1**. Según el último Boletín de Mortalidad por Cáncer en Argentina, el cáncer colorrectal es la segunda causa de muerte por cáncer representando el 12.4% y se estima que el 20% de los pacientes desarrollan metástasis hepática. La identificación temprana de esta enfermedad es crucial para el éxito del tratamiento y la supervivencia del paciente [4].

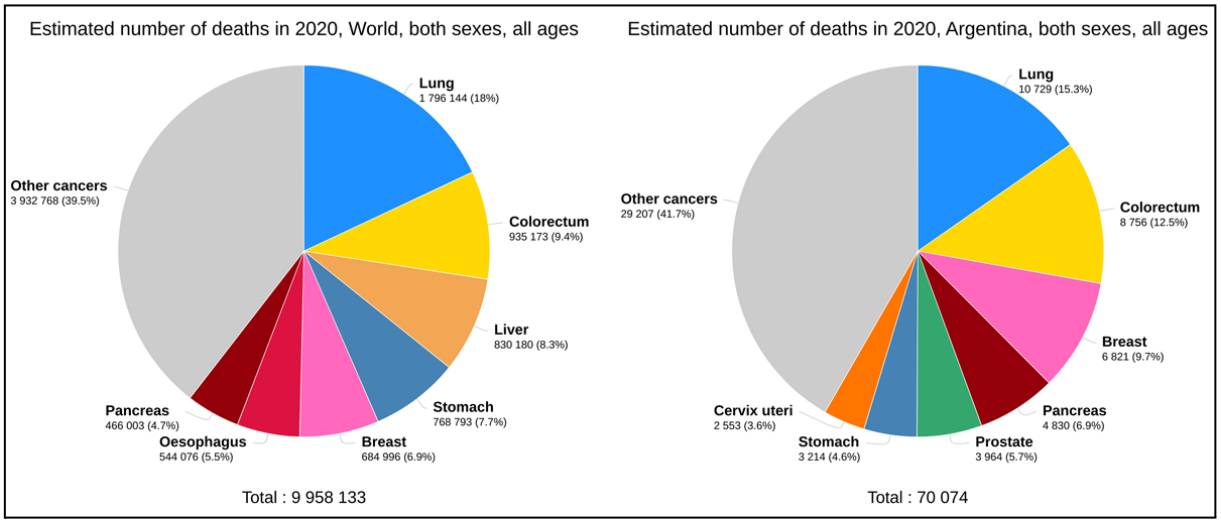

Figura 1. Gráfico circular de muertes estimadas por cáncer en el año 2022, sin diferenciar por sexo o edad. Como referencia, el cáncer pulmonar está en celeste, el CCR en amarillo, y el cáncer de mama en rosa. Izquierda: gráfico nivel global. Derecha: gráfico de Argentina [3].

Los principales sitios de extensión a distancia del CCR son el hígado y el pulmón. Se estima que un 25% de los pacientes con CCR ya han desarrollado metástasis al momento del diagnóstico, y que el 50% la desarrollarán durante la enfermedad. La mayoría de estas son hepáticas [5], y la metástasis hepática de cáncer colorrectal (**MHCC**) representa la principal causa de muerte en pacientes con esta enfermedad en todo el mundo [6].

La tomografía computada (**TC**) es el método de elección para la estadificación inicial de los pacientes con diagnóstico de CCR. La fase venosa portal es la que más información aporta para la valoración hepática debido a la hipovascularidad de estos tumores (aunque hasta en un 10% de los pacientes pueden existir lesiones hipervasculares). La MHCC puede tratarse con fines curativos a través de la de resección hepática que extirpe a los tumores, y en los estudios preoperatorios se suele realizar una tomografía de tórax, abdomen y pelvis con protocolo de trifásica hepática. Según el número y tamaño de las metástasis, se pueden realizar diferentes tipos de cirugía, como se muestra en la **Figura 2**.

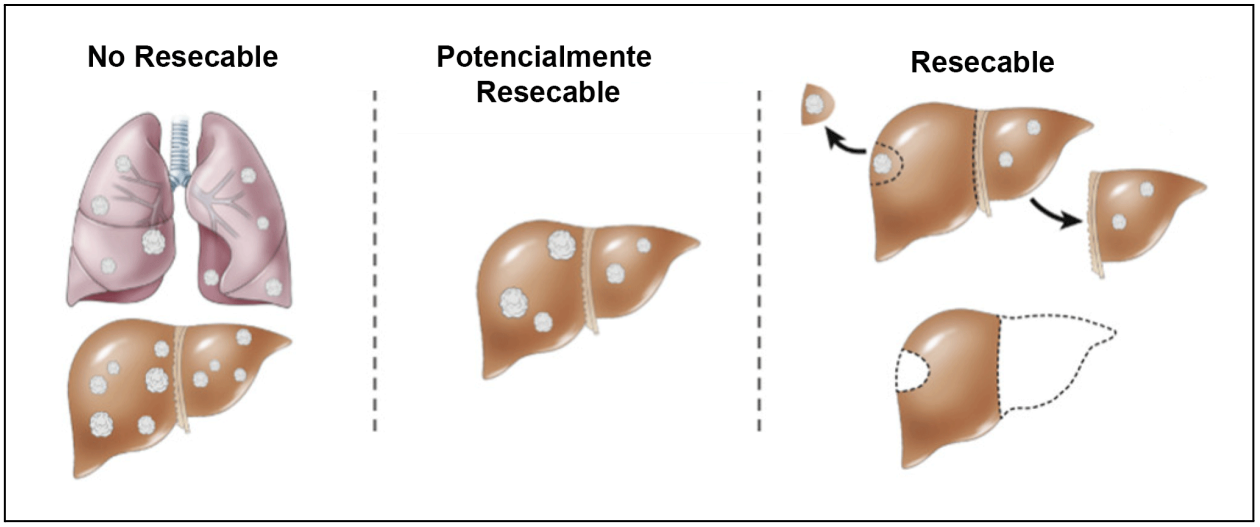

Figura 2. Esquemas de diferentes etapas de MHCC y su elegibilidad para cirugía de resección. Izquierda: representa una metástasis no resecable, extendida y en múltiples órganos, con mala respuesta a los tratamientos. Centro: representa casos intermedios, que pueden evolucionar con tratamientos como quimioterapia. Derecha: representa los casos menos agresivos o que han respondido favorablemente a tratamientos y pueden ser sujetos a cirugía. Figura adaptada de [7].

Según el grado de avance de la enfermedad, el paciente podría no ser elegible para la cirugía de resección, pero podría recibir otros tratamientos para atacar a la metástasis. Por ejemplo, podría recurrirse inicialmente a la quimioterapia, y en caso de reducirse el tamaño de las lesiones, proceder luego con la extirpación. Luego de la cirugía, es posible que haya una recurrencia de la enfermedad que lleve al fallecimiento del paciente.

Por esta razón, es importante caracterizar correctamente las lesiones hepáticas, y sobre todo evaluar la agresividad tumoral, para elegir un tratamiento adecuado para el paciente, que en ocasiones puede estar acompañado de un tratamiento de adyuvancia luego de la cirugía. Uno de los biomarcadores en esta enfermedad es el gen KRAS. Este gen codifica a una proteína del mismo nombre, que participa en una vía que bloquea la proliferación desmedida de las células. En un 30% de los casos de CCR el gen KRAS está mutado, lo que altera el funcionamiento de la proteína y está asociado con tumores más agresivos, y con mayor tasa de recurrencia [6].

Sería de utilidad en el tratamiento de esta enfermedad contar con más herramientas para caracterizar el estado inicial del paciente previo al tratamiento. Se han publicado decenas de trabajos sobre la posibilidad de realizar un análisis cuantitativo de imágenes de TC en CCR y MHCC para que los profesionales de salud puedan obtener más información útil para la toma de decisiones sobre el tratamiento. Entre estas publicaciones, se han encontrado numerosos ejemplos del desarrollo de herramientas de Inteligencia Artificial (**IA**) para el análisis de imágenes y la estimación de pronósticos para pacientes. Los objetivos definidos para el presente proyecto fueron generar una herramienta de IA que permita, mediante el análisis de características en la TC, evaluar el riesgo de mortalidad relacionado con la enfermedad antes de decidir sobre el tratamiento.

2. Marco teórico

2.1 Tomografía Computada

Las imágenes médicas son una herramienta que permite estudiar ciertos aspectos de la función y anatomía interna de un paciente, con fines tanto diagnósticos como terapéuticos. Algunos tipos de imagen médica son la radiografía, la resonancia magnética, y la ecografía; la TC es una de estas técnicas. Las imágenes producidas son secciones consecutivas del plano axial del paciente **Figura 3**, que en conjunto permiten interpretar un volumen. El equipo de adquisición es el tomógrafo: cuenta con un tubo de rayos X, que irradia en forma de cono hacia el paciente. Opuesto al tubo se encuentra un arreglo de detectores, que captan los rayos que atraviesan al paciente. Tanto el tubo como los detectores son rotatorios, y se mueven para tomar la información desde distintos ángulos. En la **Figura 4** se ilustran los componentes y funcionamiento del tomógrafo.

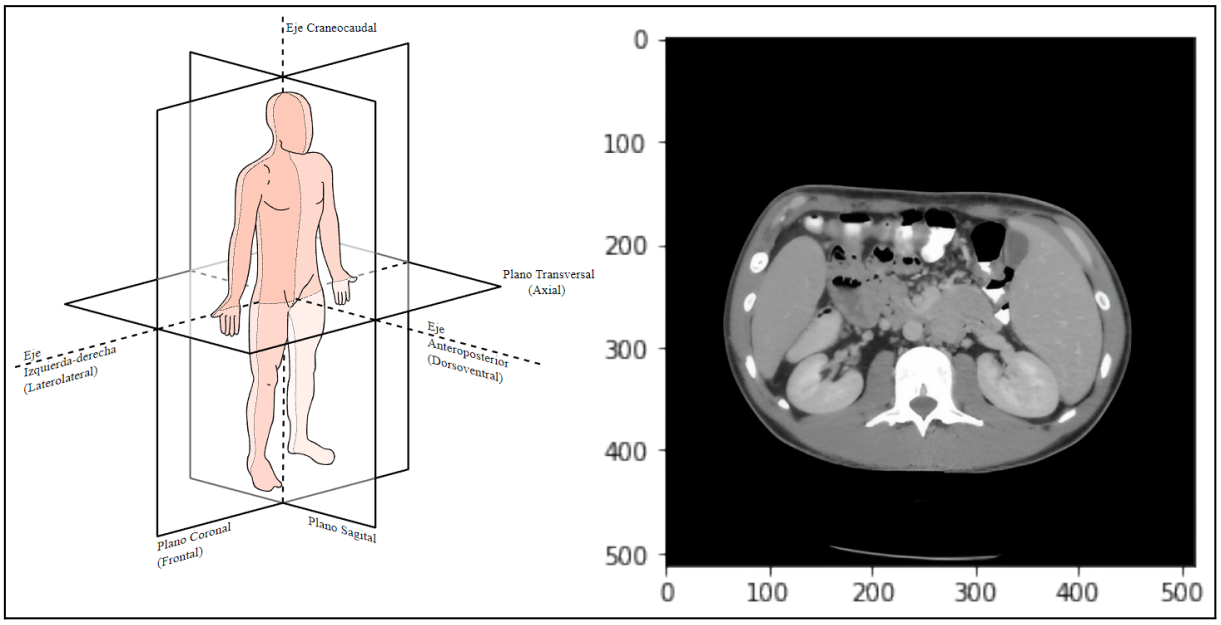

Figura 3. Izquierda: esquema de los ejes y planos anatómicos [8]. El plano axial o transverso es una sección horizontal de la persona. Derecha corte de una TC abdominal, que se hace en el plano axial.

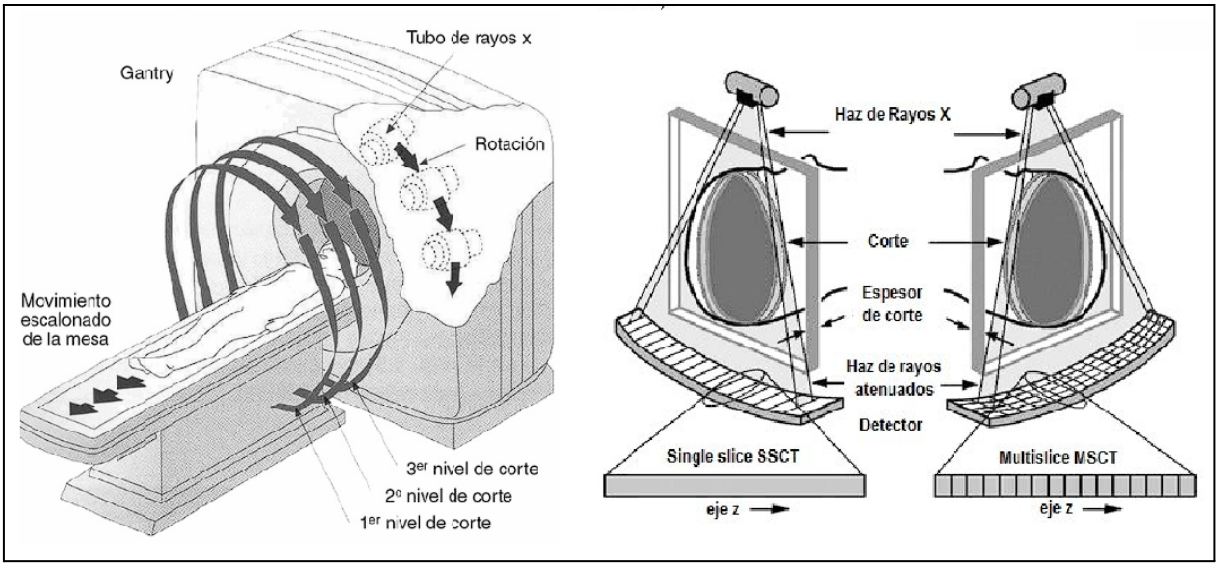

Figura 4. Izquierda: ilustración de un tomógrafo multicorte. Tanto el tubo de rayos X como el arreglo de detectores se ubican directamente opuestos en el gantry (cuerpo del equipo que presenta un orificio central), la estructura rotatoria dentro del tomógrafo. El par tubo-arreglo rota sobre el paciente a medida que se irradia para captar la información desde diferentes ángulos. La mesa en la cual reposa el paciente puede moverse en sincronía con los disparos para capturar las imágenes a lo largo del plano axial. Derecha: Esquema del arreglo de detectores. Los tomógrafos actuales usan varias filas de detectores para capturar múltiples segmentos en el plano axial con un único disparo de rayos X [9].

La intensidad de los rayos X, atenuada al pasar por el paciente, es captada y transducida a una señal eléctrica por los detectores. Todas las señales de un mismo corte, desde diferentes ángulos, se transforman en una matriz que tiene típicamente 512 filas y columnas, a partir de la cual se puede visualizar la TC. Si bien es una imagen en dos dimensiones (**2D**), el corte no es infinitamente fino, sino que representa un volumen en tres dimensiones (**3D**), cuyo espesor está determinado por la cantidad y tamaño de detectores, entre otros factores.

El valor de cada píxel está dado por el coeficiente de atenuación de los tejidos, en la escala de unidades Hounsfield (**UH**). Para una imagen tomada en condiciones normales de presión y temperatura, a cada tejido o sustancia le corresponde un valor diferente, con -1.000 siendo aire, 0 el agua, 1.500 el hueso compacto, y 3.000 metal. Por esto, las TC se almacenan en formato de 12 bits, que permite el uso de 4096 valores distintos. Si bien cada corte puede guardarse en alguno de los formatos comúnmente usados para imágenes digitales, existen formatos específicos para imágenes médicas que, además de almacenar los datos de los píxeles, tienen también información útil para la práctica clínica e investigación.

2.1.1 Los formatos DICOM y NIfTI

DICOM (Digital Imaging and Communication in Medicine) [10] es el estándar internacional para la transmisión, almacenamiento, procesamiento y visualización de imágenes médicas. Su objetivo es la interoperabilidad de las imágenes y su información, integrando a los equipos de diferentes fabricantes usados en las tareas antes mencionadas. Este formato guarda no solo los píxeles, sino también información asociada al estudio en el cual se adquirió la imagen, identificando cada tipo de dato con una etiqueta numérica. Los datos para la identificación del paciente (como el nombre y la edad) o datos técnicos (modalidad del estudio, equipo de

adquisición, ancho de píxel) son algunos ejemplos de los cientos de etiquetas disponibles para almacenar en un archivo DICOM.

Si bien el formato DICOM es el estándar en imágenes médicas, existen alternativas con propósitos más específicos. La iniciativa **NIfTI** (*Neuroimaging Informatics Technology Initiative*) [11] tiene como objetivo acelerar y mejorar la utilidad de herramientas informáticas relacionadas a imágenes médicas, buscando la interoperabilidad entre ellas. Inicialmente, se planteó su uso para la resonancia magnética funcional en neuroimágenes, pero se ha expandido a otras modalidades y especialidades.

La estructura general es similar entre ambos formatos: ambos tienen una cabecera que almacena y organiza distintos tipos de datos. Mientras que en la cabecera de DICOM se puede encontrar cientos de etiquetas con exhaustiva información referida al paciente y al estudio, la información mínima que se encuentra en un archivo NIfTI está exclusivamente referida a la descripción del volumen capturado. Además, NIFTI utiliza un solo archivo para representar a todo un volumen de datos, mientras que en el formato DICOM se tiene un archivo por corte. El segundo es más conveniente para analizar cortes particulares, pero NIfTI ofrece herramientas y ventajas para trabajar con volúmenes y solucionar problemas de espacio y alineamiento en las imágenes; es por esto último que es usado extensamente en procesamiento de imágenes e investigación, mientras que el formato DICOM prevalece en la práctica médica.

2.2 Procesamiento de imágenes digitales

El procesamiento de imágenes digitales se refiere al uso de algoritmos que modifican los píxeles tal que la imagen resultante sea más adecuada para una aplicación específica, en general facilitando su interpretación, sea humana o por un software. El procesamiento se logra mediante algoritmos u operaciones matemáticas conocidas como filtros. Se pueden considerar diferentes tipos de procesamiento: el mejoramiento, referido a la reducción de imperfecciones en la imagen ocurridas en la generación de la misma; la segmentación, que es la división de la imagen en segmentos con características comunes; el análisis de textura, que corresponde a la descripción de una imagen de acuerdo a valores cuantificables; y la codificación, usada para modificar cómo se guardan y representan los datos de los píxeles y optimizar su almacenamiento y transmisión [12] [13]. En esta sección se presentan tres técnicas de mejoramiento: los filtros convolucionales, los filtros morfológicos, y el registro de imágenes.

2.2.1 Filtros convolucionales

Los filtros convolucionales son un tipo de filtro espacial: alteran la imagen en base a la relación espacial entre píxeles cercanos unos a otros. Se definen por una **máscara**, que es una matriz pequeña con coeficientes propios de cada filtro, y se caracterizan por aplicarse mediante las operaciones de convolución o correlación cruzada. Si bien la convolución y correlación cruzada son operaciones diferentes, siempre que la máscara sea una matriz simétrica, el resultado de una u otra será el mismo, y en general se prefiere utilizar la correlación cruzada (el filtro se considera convolucional independientemente de cuál de las dos operaciones se usa) [13]. La correlación cruzada para una matriz $A(x,y)$ de dimensión M x N con una máscara k(a,b) de

dimensión *n* x *m* se define de la siguiente forma:

$$A(x,\ y)\ *\ k\ =\ \sum_{a=-\frac{N-1}{2}}^{\frac{N-1}{2}}\ \sum_{b=-\frac{M-1}{2}}^{\frac{M-1}{2}} k(a,\ b)\ A(x\ +\ a,\ y\ +\ b)$$

(1)

El filtro se aplica a cada píxel de la imagen, como se ejemplifica con la **Figura 5**: para una matriz *A(x,y)*, se aplica el filtro *k* con una máscara de dimensión 3 x 3 sobre el píxel *A(2,2)*. La máscara se superpone a ese píxel, y se hace el producto entre cada par de puntos superpuestos entre la imagen y la máscara. El resultado del filtro para *A(2, 2)* es la sumatoria de esos productos. Para filtrar a *A(x, y)* con el filtro *k*, se repite el proceso con todos los píxeles de la imagen.

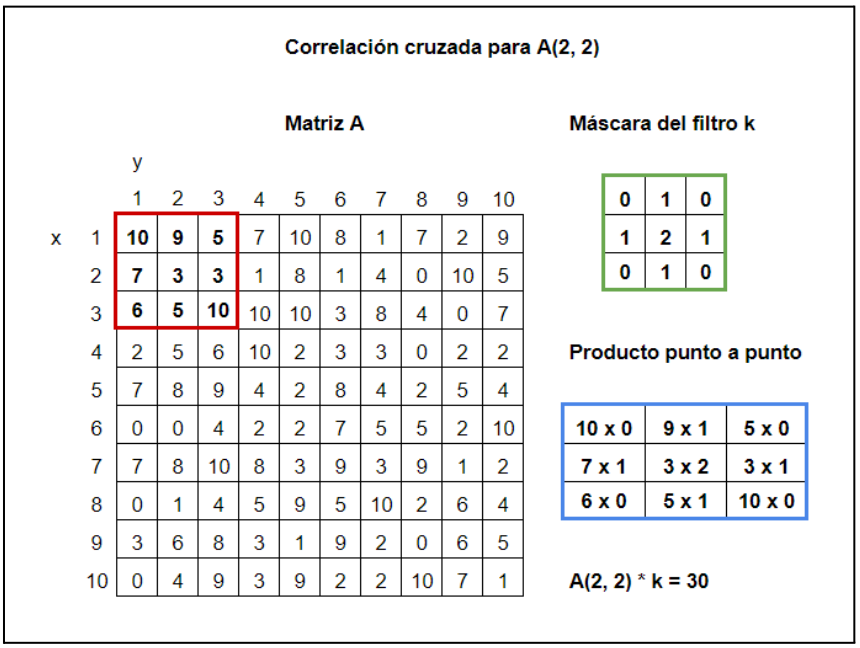

Figura 5. Ejemplo de correlación cruzada. Se toma el punto *(2, 2)* en la matriz *A*, y sus píxeles vecinos (cuadro rojo). Se hace el producto punto a punto (cuadro azul) con la máscara del filtro *k* (cuadro verde). El resultado de la sumatoria, y del filtro para el punto *(2, 2)*, es 30.

Los píxeles en los bordes de las imágenes presentan un problema para la correlación cruzada, ya que en su vecindad no existen todos los píxeles para formar una matriz del tamaño requerido para la operación. Sin resolver esto, los bordes no podrían ser incluidos en el filtro y se perderían en el resultado. La solución es el relleno o ampliación de la imagen con valores artificiales que solo son usados en el filtrado de los píxeles originales de la matriz. Hay diferentes técnicas de relleno, como el uso de ceros, el uso del valor de los píxeles en el extremo opuesto de la imagen, o el espejado del valor de los píxeles de borde.

Según su uso, los filtros convolucionales tienen dos agrupaciones. Los filtros suavizantes usan una máscara que promedia el valor de cada píxel con sus vecinos, logrando una imagen más

homogénea, pero atenuando las variaciones rápidas en intensidad que son características de los bordes de objetos. Los filtros de realce tienen el efecto opuesto, intensificando las diferencias de valores entre píxeles vecinos para destacar contornos, suprimiendo las regiones de alta homogeneidad.

Los filtros de realce pueden beneficiarse del uso previo de un filtro de suavizado para disminuir el ruido en la imagen. El filtro Laplaciano de Gaussiano (LdG) es un filtro de realce que incorpora tal disminución. El nombre viene de la fórmula de la máscara del filtro, que toma el laplaciano de la función gaussiana. La máscara de un filtro Gaussiano, un filtro suavizante, se construye utilizando la función gaussiana bidimensional:

$$G(x, y) = \frac{1}{2\pi\alpha^2} e^{\frac{x^2+y^2}{2\sigma^2}}$$

(2)

Si se aplica el operador laplaciano, la máscara estará dada por la derivada segunda de *G(x, y)* en ambas variables:

$$\frac{\delta^2 G(x,y)}{\delta x^2} + \frac{\delta^2 G(x,y)}{\delta y^2} = -\frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

(3)

El factor *σ* determina el área de efecto del filtro. Si el valor es pequeño, entonces el filtro trabaja sobre vecindades pequeñas, y solo destaca texturas finas. Al aumentar *σ*, aumenta el área afectada: las texturas finas serán menos relevantes debido al suavizado, y serán resaltadas las texturas más gruesas, como se muestra en la **Figura 6**.

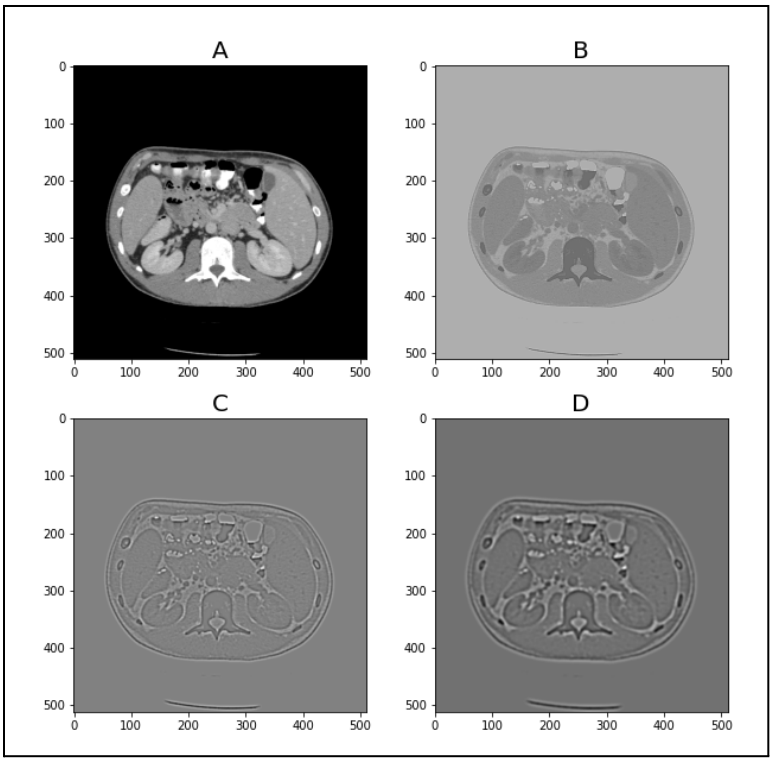

Figura 6. Aplicación de un filtro LdG a una TC abdominal. (A): imagen sin filtrar. (B): filtro LdG con σ = 0.5. (C): filtro LdG σ = 1.5. (D): filtro LdG con σ = 2.5.

2.2.2 Filtros morfológicos

En matemática, la morfología es una teoría y práctica para el estudio y tratamiento de estructuras geométricas. En imágenes, se puede utilizar para reconocer a un continuo de puntos como un objeto y sus características geométricas. Los filtros morfológicos utilizan esta teoría para modificar los objetos en una imagen. Las operaciones de filtrado se hacen entre la imagen y un **elemento estructurante**: una máscara de valores binarios, que definen una figura (como un cuadrado, círculo o cruz), en general simétrica y con su origen de coordenadas en el centro geométrico. El comportamiento del filtro está definido tanto por la forma del elemento estructurante como por el tipo de comparación realizada [13].

Los filtros morfológicos se suelen usar en imágenes binarizadas o segmentadas, en la cual los píxeles de cada objeto de interés tienen un mismo valor, diferente al valor de otros objetos, o el fondo de la imagen. En una imagen binaria, el objeto de interés suele marcarse con píxeles de valor 255, mientras que todo lo demás, considerado fondo, tiene valor 0. La operación consiste en superponer al elemento estructurante con cada uno de los píxeles en la imagen, y comparar los valores de los píxeles en su vecindad con los valores de la máscara. Según el tipo de operación y el resultado de la comparación, se puede modificar el valor del píxel sobre el cual está centrada la máscara. Bajo esta comparación se definen las operaciones morfológicas elementales, la **erosión** y **dilatación**. Como ejemplo, sea una imagen *I(x, y)* binaria de dimensión *N x M*, los píxeles del fondo valen 0, y los de un objeto *A*, valen 1. La dilatación del conjunto *A* con un elemento estructurante *B* se demuestra en la **Figura 7**.

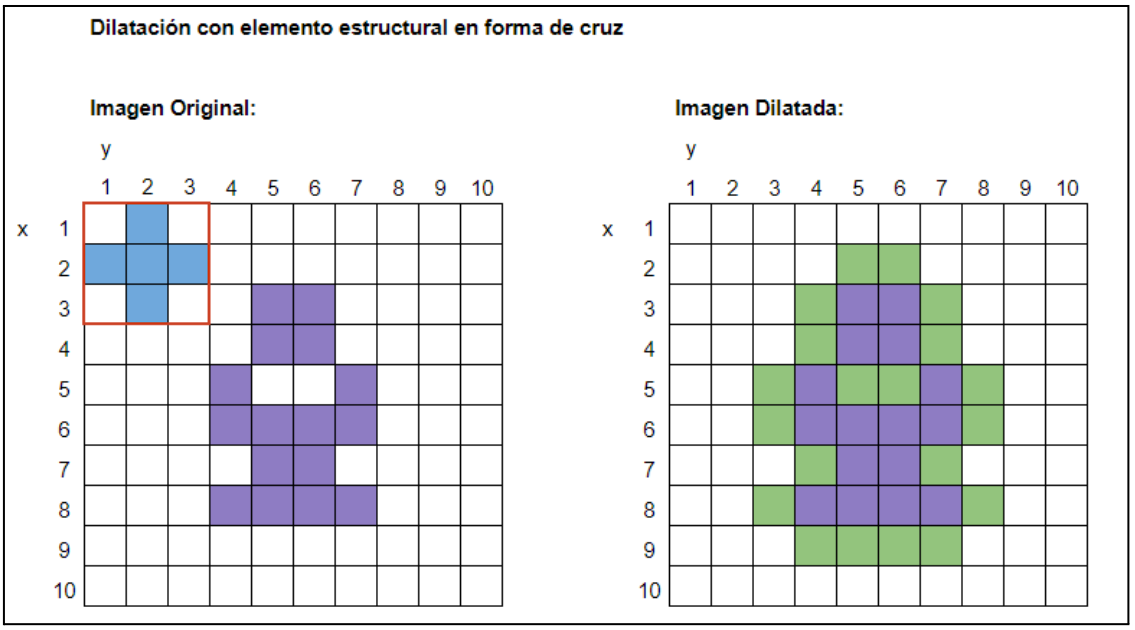

Figura 7. Ejemplo de dilatación de una imagen. Izquierda: imagen original, con sus píxeles en color violeta, y superpuesta en su esquina superior izquierda en color celeste. Derecha: imagen dilatada, con los píxeles originales en color violeta, y los píxeles agregados en color verde.

En cualquier caso donde haya al menos un píxel no nulo con el mismo valor tanto en el conjunto *A* y el conjunto *B*, entonces el píxel sobre el cual está centrada la máscara será considerado parte de *A*, dilatando al conjunto.

Por otro lado, en la erosión solo serán parte de *A* aquellos elementos que coincidan completamente con *B*. Al anular los píxeles que no cumplen con la condición, se degradan los bordes del objeto, como se ve en la **Figura 8**.

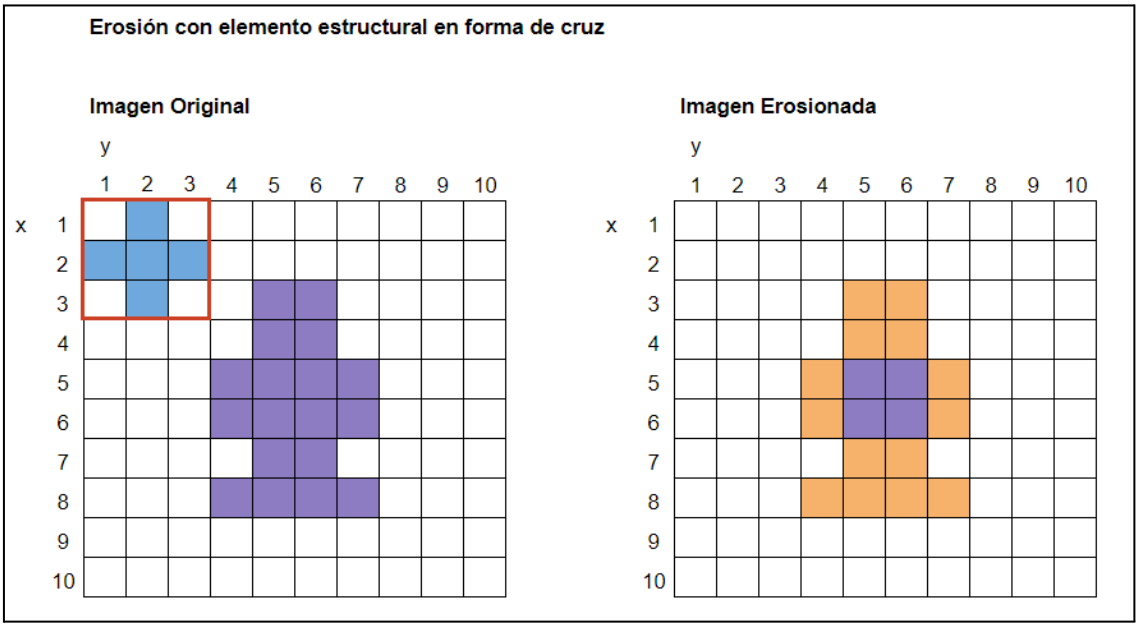

Figura 8. Ejemplo de erosión de una imagen. Izquierda: imagen original, con sus píxeles en color violeta, y superpuesta en su esquina superior izquierda en color celeste. Derecha: imagen erosionada, con los píxeles remanentes en color violeta, y los píxeles eliminados en color verde.

El **cierre** es una operación compuesta de las dos anteriores, siendo la erosión de una figura dilatada. La primera dilatación tiene el efecto de conectar figuras separadas por pocos puntos, rellenar pequeños huecos y suavizar concavidades, mientras que la erosión revierte el efecto de dilatación en los bordes, lo que preserva la forma original de la imagen, como se muestra en la **Figura 9**.

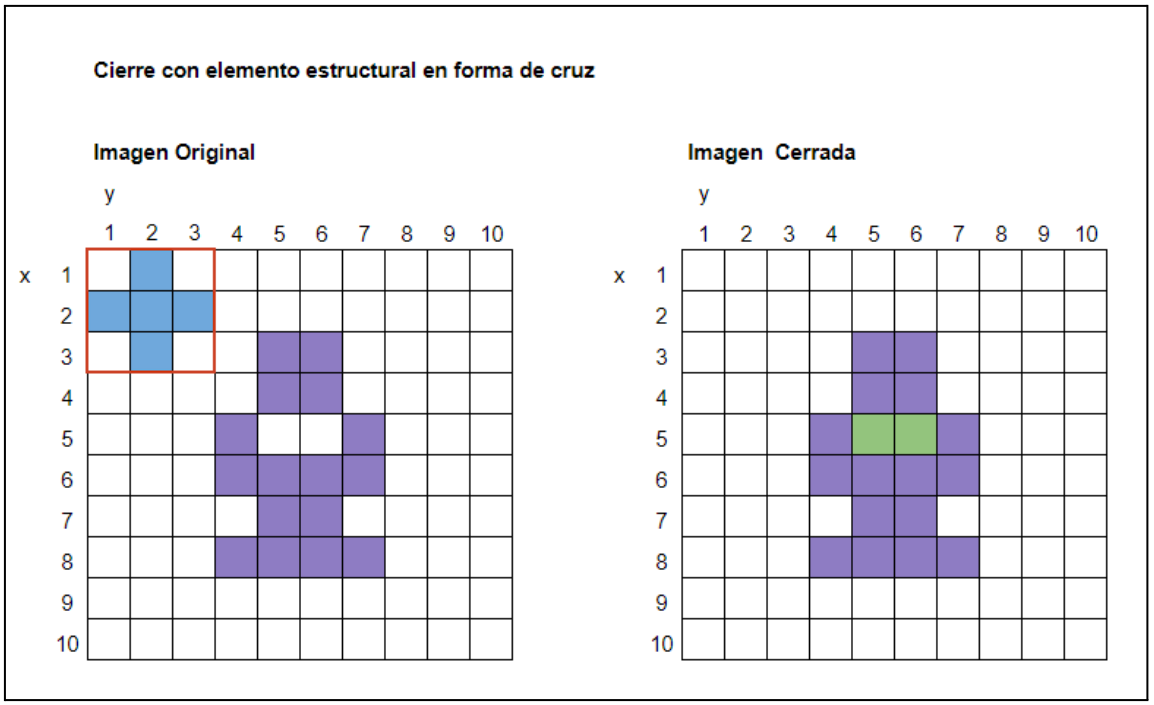

Figura 9. Ejemplo del cierre de una imagen. Izquierda: imagen original, con sus píxeles en color violeta, y superpuesta en su esquina superior izquierda en color celeste. Derecha: imagen cerrada, con los píxeles originales en color violeta, y los píxeles agregados en color verde.

2.2.3 Registro de imágenes

La comparación de muestras en un estudio es más fácil si estas están estandarizadas en cuanto aspecto sea posible, y esto incluye a las imágenes médicas. Idealmente, las imágenes de cualquier estudio se adquieren con los pacientes en la misma posición, pero lo normal es que haya algún grado de inclinación en cualquiera de los tres ejes. La heterogeneidad en la posición del paciente no está limitada a estudios de distintas personas; por ejemplo, en un estudio con contraste, es habitual adquirir la imagen tanto con el efecto del contraste como sin él, y el paciente puede moverse entre capturas. El procesamiento para hallar y realizar la transformación que permita alinear imágenes en un mismo sistema de coordenadas se conoce como registro o registración, y es una solución a este problema [13].

En el registro se tiene una imagen de referencia o fija, con el sistema de coordenadas deseado *(x, y)*, y una o más imágenes origen, con un sistema de coordenadas *(x', y')*. Una transformación *T* puede convertir los píxeles del sistema de origen al deseado y se expresa comúnmente de forma matricial. Hay distintos tipos de transformaciones y su funcionamiento se refleja en las componentes de *T*. Por ejemplo, las transformaciones geométricas operan sobre las coordenadas de los píxeles, transportando los valores de intensidad a una posición

distinta. La transformación bidimensional genérica (calculada según el paquete Nibabel para imágenes NIfTI en Python [16]) se escribe como:

$$\begin{bmatrix} x \\ y \end{bmatrix} = T(x', y') = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

(4)

Donde los coeficientes *a* producen el escalado de la imagen, y los coeficientes *b* realizan una traslación. Otro tipo de coeficientes o matrices producen otros efectos, como la rotación en un ángulo *Θ*:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

(5)

Estas y otras transformaciones pueden alterar la relación y distorsionar la imagen, y hay un compromiso asumido en los aspectos preservados y afectados de la imagen al hacer el registro. Las transformaciones afines son un tipo de transformación lineal donde se preserva el paralelismo entre las líneas del eje de aplicación. La matriz de una transformación afín es de una dimensión mayor a la de la imagen, donde los coeficientes *a* realizan la rotación y escalado, y los coeficientes *t* la traslación. La tercera fila existe para poder incluir los coeficientes de *a* y *t* en una misma matriz:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

(6)

2.3 Radiómica

Las técnicas de procesamiento pueden usarse para facilitar la interpretación de las imágenes, pero el análisis visual no es la única forma de obtener información de ellas. Las características de una imagen pueden representarse numéricamente, por ejemplo, con el valor promedio de los píxeles. La radiómica es un campo de la medicina que tiene el objetivo de extraer estos valores representativos en imágenes médicas, llamadas **características radiómicas**. Estas características se pueden usar en un análisis cuantitativo, aquel hecho con valores objetivamente medibles. Diferente es el análisis cualitativo, que depende de la interpretación y relación de los datos en base a la experiencia y conocimiento del intérprete. Por ejemplo, en el diagnóstico por imágenes, un especialista puede interpretar un cuerpo hallado en una TC y relacionarlo con mediciones realizadas sobre las imágenes o de un estudio de laboratorio para dar su diagnóstico. La consistencia en las mediciones de las características radiómicas les da utilidad en el diagnóstico y la investigación, y pueden ser usadas por los algoritmos de soporte

a la toma de decisiones en medicina.

Las características radiómicas son de gran utilidad cuando se demuestra su relación con una problemática en la medicina, características llamadas **biomarcadores**: una característica que es objetivamente medible y evaluable como indicador de procesos biológicos normales, procesos patológicos, o respuestas farmacológicas a una intervención terapéutica [14]. Aún cuando se pudieran generar biomarcadores, se necesita modelar su relación con una patología o respuesta, para lo cual se requiere una investigación y un volumen de datos suficiente.

Si se logra demostrar que una característica guarda relación con un resultado de interés (como un diagnóstico o la respuesta a un tratamiento), entonces esa característica pasa a ser un biomarcador del tipo de imagen del cual se obtuvo, para el resultado con el cuál fue asociada [13]. Esto abre la puerta a la investigación y desarrollo de métodos cuantitativos usando imágenes radiológicas, y es de particular interés para la oncología. Al describir con mayor precisión las características de un tumor, se tienen más herramientas para entender el estado de un paciente o dar un diagnóstico. En lo que concierne al tratamiento, este podría elegirse con más seguridad y ajustado a las necesidades del paciente, e incluso estimar la respuesta del paciente al mismo.

Los algoritmos de extracción de características radiómicas operan sobre una porción de la totalidad de la imagen de entrada. La segmentación es la tarea de determinar y delimitar una o varias regiones de interés (*Region of Interest*, **ROI**). La ROI debe capturar una sección de la imagen con información relevante para el estudio; en oncología se suelen delimitar tumores o lesiones. En la **Figura 10** se muestra un ejemplo de cómo se visualiza una segmentación.

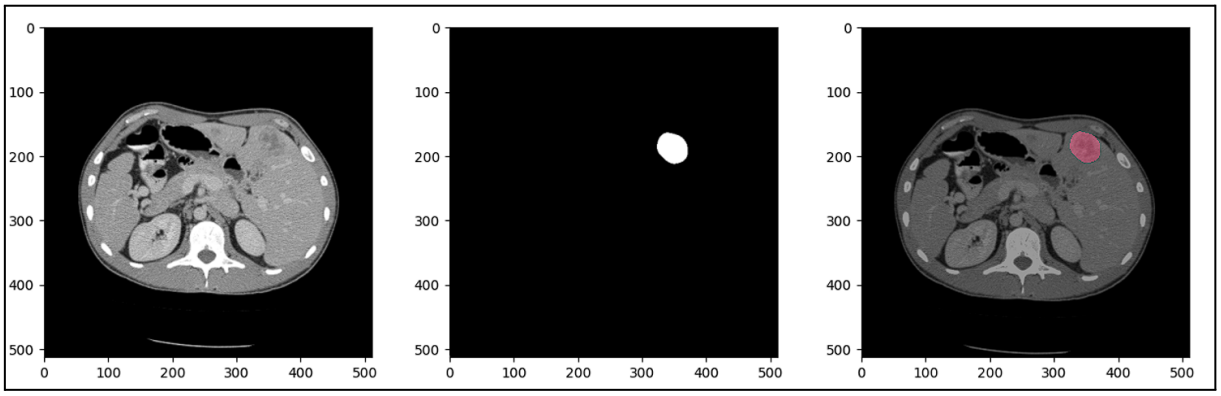

Figura 10. Visualización de una máscara de segmentación de un tumor hepático. Izquierda: corte de TC abdominal. Centro: Máscara de la ROI segmentada en el mismo corte. Derecha: superposición de la máscara con el corte.

Existen aplicaciones para segmentar una imagen, que permiten marcar los píxeles correspondientes a la ROI. Digitalmente, se genera una **máscara**, una matriz de igual dimensión a la imagen segmentada, donde los píxeles que corresponden a una ROI tienen un mismo valor (y distintas ROI en una imagen pueden tener distintos valores para diferenciarlas), mientras que el resto de píxeles están identificadas por otro valor. En caso de haber una única ROI, los valores típicos son cero si los píxeles están fuera de ella, y 1 si pertenecen. Una ROI corresponde a una única imagen, pero la segmentación puede delimitar un volumen de interés (*Volume of Interest,* **VOI**) abarcando varios cortes consecutivos dentro de un estudio.

2.3.1 Extracción de características

Para facilitar la reproducibilidad de la extracción de características, es conveniente que la distancia entre el centro de un píxel a todos sus vecinos sea la misma, en todos los ejes de la imagen; en el análisis del VOI de una TC, se consideran las distancias en 3D. Para lograr un espaciado isotrópico (misma distancia en los tres ejes), se submuestrean o sobremuestrean las imágenes. En el submuestreo, se utilizan menos píxeles para representar a la imagen original, lo que requiere que cada píxel individual represente una parte más grande de la imagen, haciendo que cada píxel en sí sea más grande y aumentando la distancia entre sus centros. En el sobremuestreo, se utilizan más píxeles, con lo cual cada uno representa un espacio menor, y es entonces más pequeño y es menor la distancia entre sus centros. Al submuestrear, se toma la dimensión con mayor distancia entre los centros como referencia, mientras que al sobremuestrear, se hace lo contrario [13]. Otra alteración a las imágenes que se hace antes de la extracción es la aplicación de filtros.

Las imágenes transformadas presentan características con valores distintos a si se las hubiera calculado en la imagen sin filtrar. Es posible que una característica sea poco relevante para un problema cuando se la calcula en la imagen original, pero determinante si se la resalta o expone mediante estos filtros. Como la aplicación de estos filtros suele llevar el valor de los píxeles por fuera de su rango habitual, es común escalar la intensidad de vuelta al rango original luego de hacer la transformación. Algunos filtros comunes son:

- **Filtro de cuadrados:** cada píxel se eleva al cuadrado.
- **Filtro de raíz cuadrada:** se toma la raíz cuadrada del valor absoluto de cada píxel.
- **Filtro logarítmico**: se toma el logaritmo del valor de cada píxel + 1.
- **Filtro exponencial:** se calcula la exponencial del valor absoluto de cada píxel.
- **Filtro laplaciano de gaussiano (LdG)**: un filtro de realce de bordes. Se utiliza el factor σ para establecer el tamaño de la máscara. Si es bajo, se resaltan cambios de intensidad en la cercanía del píxel central (texturas finas), mientras que uno alto resalta cambios más lejanos (texturas gruesas). Los valores típicos son 0,5 (texturas finas), 1,5 y 2,5 (texturas gruesas).

Las características radiómicas se dividen en varios grupos de acuerdo a qué es lo que describen. Las características de primer orden describen la intensidad de los píxeles utilizando los estadísticos de primer orden. Las características de forma analizan las dimensiones de la ROI, en cada corte de un estudio 2D, y en el VOI. Las características de textura o de segundo orden describen la ROI observando la relación espacial entre los píxeles y la variación de intensidad entre píxeles vecinos. En las secciones siguientes se listan características de cada grupo, con una breve descripción de su propósito y método con el cuál son calculadas, siendo esa la razón de su agrupamiento. Cada característica y fórmula presentada corresponden a las implementadas en la librería PyRadiomics [15] para la extracción de características radiómicas en Python y complementadas con las definiciones del manual de *Image Biomarker Standardization Initiative* [17].

2.3.1.1 Características de primer orden

Los estadísticos de primer orden son calculados usando datos unidimensionales de la ROI,

utilizando el histograma de intensidades de la misma. El rango de valores posibles de cada píxel está determinado por la profundidad de color o píxel. Los píxeles en una imagen de 8 bits pueden tomar 256 valores distintos, mientras una de 16, hasta 65.536. Dado un conjunto de datos y el rango de valores que estos pueden tomar, el histograma es un gráfico que muestra cuántas veces se repite cada valor de los posibles en el rango, en ese conjunto de datos, ejemplificado en la **Figura 11**.

En una imagen de TC, el histograma representa la frecuencia con la que aparecen las distintas intensidades de gris. Para armar el histograma, el rango de valores a evaluar se separa en intervalos, llamados *bins*. La agrupación de estos valores se determina con el ancho o la cantidad de *bins*: el ancho indica cuántos valores entran en cada intervalo, mientras que la cantidad indica el número de intervalos. Si se tiene una imagen con un rango de 0 a 255 y se elige un ancho de *bin* de 8, cada *bin* deberá incluir ocho valores, y se necesitarán 32 *bins* para hacer el histograma de la imagen.

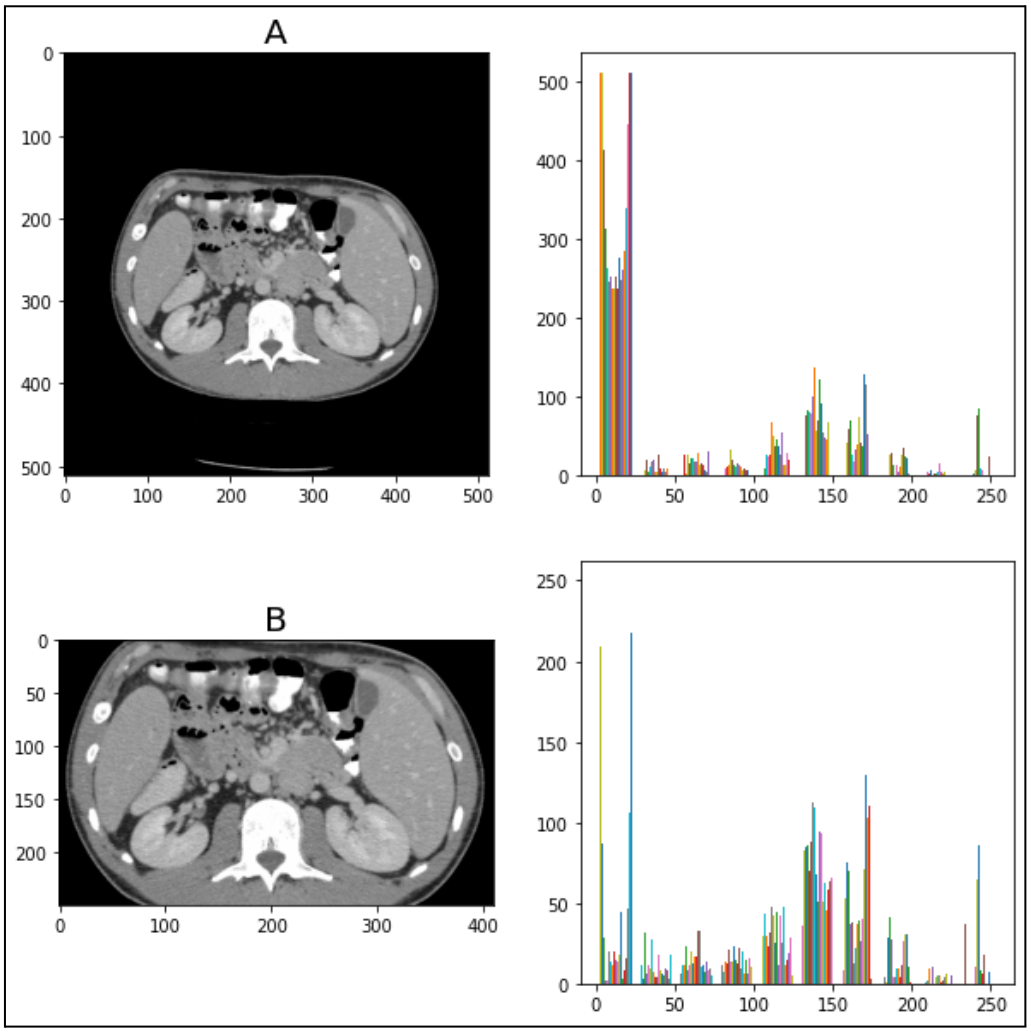

Figura 11. Ejemplos de histogramas en TC. (A): histograma de una TC abdominal, donde se ve una elevada frecuencia (eje vertical) de píxeles de baja intensidad (eje horizontal). (B):histograma de la misma TC, recortada. Al eliminarse píxeles negros, la frecuencia de los valores está más balanceada y es más fácil de apreciar en el gráfico.

A continuación se listan 19 estadísticos de primer orden, siendo *I(x)* la función que representa

todos los valores de intensidad en la ROI [15].

1. **Máximo**: el máximo de *I(x)* en la ROI.
2. **Mínimo**: el mínimo de *I(x)* en la ROI.
3. **Rango**: el rango de *I(x)* en la ROI, la diferencia entre Máximo y Mínimo.
4. **Mediana**: valor de intensidad que, al ordenar todos los píxeles de la imagen de mayor a menor intensidad, queda en el centro del conjunto de datos.
5. **Percentil 10**: valor de intensidad para el cual el 10% de los píxeles de la ROI tienen un valor de intensidad menor.
6. **Percentil 90**: valor de intensidad para el cual el 90% de los píxeles de la ROI tienen un valor de intensidad menor.
7. **Rango intercuartílico**: rango de intensidad entre el percentil 25 (25% de los píxeles tienen una intensidad menor) y el percentil 75 (75% de los píxeles tienen una intensidad menor).
8. **Media**: promedio de intensidad en la ROI, con N siendo el total de píxeles en ella.

$$Media = \frac{1}{N}\sum_{i=1}^{N} I(i) \tag{7}$$

9. **Varianza**: representa la distribución de la intensidad alrededor de la Media. Se calcula como la media de las diferencias al cuadrado de cada intensidad con la Media.

$$Varianza = \frac{1}{N}\sum_{i=1}^{N} (I(i) - \bar{I})^2 \tag{8}$$

10. **Desviación de la media absoluta (MAD)**: es el promedio de las distancias absolutas de cada valor de intensidad a la media.

$$MAD = \frac{1}{N}\sum_{i=1}^{N} |I(i) - \bar{I}| \tag{9}$$

11. **Desviación de la media absoluta robusta (rMAD)**: es el cálculo de la MAD que se hace considerando sólo los píxeles entre el percentil 10 y percentil 90, lo que le da robustez ante los valores de intensidad aislados en los extremos.

$$rMAD = \frac{1}{N_{10-90}}\sum_{i=1}^{N} |I_{10-90}(i) - \bar{I}_{10-90}| \tag{10}$$

12. **Valor cuadrático medio (RMS)**: Es la raíz cuadrada del promedio de cuadrados de la intensidad.

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} I(i)^2} \tag{11}$$

13. **Asimetría**: representa la asimetría de la distribución de los datos respecto al valor de la media. Un valor negativo indica que hay más frecuencia de valores menores a la media, uno positivo indica que son más frecuentes aquellos valores mayores, y uno cercano a cero indica simetría.

$$Asimetría = \frac{\frac{1}{N}\sum_{i=1}^{N} (I(i) - \bar{I})^2}{\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N} (I(i) - \bar{I})^3}\right)^3} \tag{12}$$

14. **Kurtosis**: mide con qué frecuencia se dan valores que están más alejados de la media. Si la kurtosis es alta, los valores se acumulan en los extremos, mientras que si es baja, se acumulan cerca de la media.

$$Kurtosis = \frac{\frac{1}{N}\sum_{i=1}^{N} (I(i) - \bar{I})^4}{\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N} (I(i) - \bar{I})^2}\right)^2} \tag{13}$$

15. **Uniformidad**: mide la homogeneidad de intensidad, donde un valor elevado indica alta homogeneidad, o un rango acotado en los valores de intensidad. En la fórmula, *p(i)* son los valores del histograma normalizado.

$$Uniformidad = \sum_{i=1}^{N} p(i)^2 \tag{14}$$

16. **Energía**: es una medida de la magnitud de los valores de vóxel en la región.

$$Energía = \sum_{i=1}^{N} I(i)^2 \tag{15}$$

17. **Energía total**: es la energía escalada por el VOI (en milímetros cúbicos) de un vóxel en la región.

$$Energía\,Total\ = V_{vóxel} \sum_{i=1}^{N} I(i)^2 \tag{16}$$

18. **Entropía**: es una medida que especifica la aleatoriedad en los valores. Una entropía alta se corresponde con una frecuencia similar de todos los valores, mientras que una baja se asocia a frecuencias dispares. En la fórmula, *p(i)* son los valores del histograma normalizado.

$$Entropía\ = \ -\sum_{i=1}^{N} p(i)\ log_2(p(i)) \tag{17}$$

2.3.1.2 Características de forma en 2D

Los cálculos necesarios para la extracción de estas características se hacen usando una malla que representa, de manera aproximada, la forma de la ROI en un corte. Para generar esta malla, se utiliza el algoritmo de cuadrados de marcha, junto a la información de qué píxeles conforman la ROI. El algoritmo consiste en recorrer la máscara con un cuadrado que abarca 2x2 píxeles. En cada paso, se verifica si los vértices del cuadrado están dentro (marcando ese vértice como 0) o fuera (marcándolo como 1) de la ROI. Al haber dos valores posibles para cuatro vértices, hay 16 combinaciones posibles para cada cuadrado. Los 16 valores están tabulados e indican la línea que corresponde a ese píxel. Los píxeles quedan compuestos por dos triángulos, que pueden pertenecer ambos a la ROI (si la malla queda alineada a sus bordes) o solo uno (si la malla cruza al píxel) [18]. La **Figura 12** ejemplifica cómo se determinan los bordes de una figura utilizando el algoritmo de cuadrados de marcha.

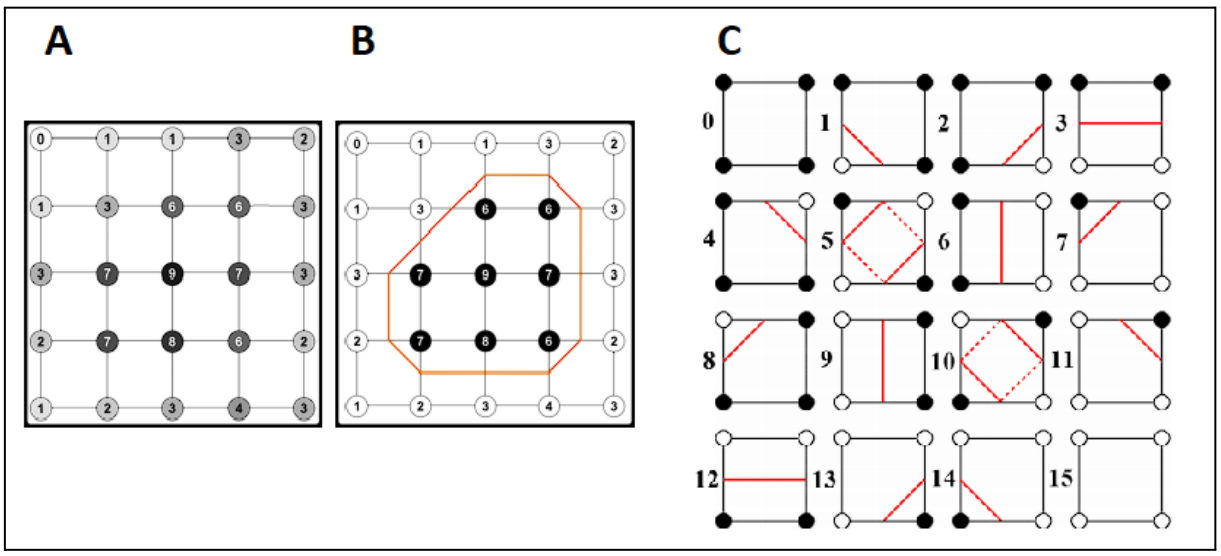

Figura 12. (A): matriz de píxeles con valores entre 0 y 9. (B): resultado de usar cuadrados de marcha, considerando todos los píxeles de valor cinco o más como parte de la región. (C) referencia de las 16 combinaciones posibles de píxeles fuera y dentro de la región y las líneas que representan [18]. Copyright © 2016, IEEE.

Con la máscara de la ROI, las dimensiones en milímetros de píxel, y la definición de la malla perimetral, se tienen cuatro parámetros: N_p, la cantidad de píxeles en la ROI; N_f, la cantidad de líneas que definen la malla; A, la superficie de la malla en milímetros cuadrados; y P, el perímetro de la malla en milímetros. Además, se define un elipsoide que contiene a la ROI, y del cual calculan los ejes ortogonales mayor y menor de la ROI y sus componentes (λ_{mayor} y λ_{menor}) mediante análisis de componentes principales [15], ejemplificado en la **Figura 13**.

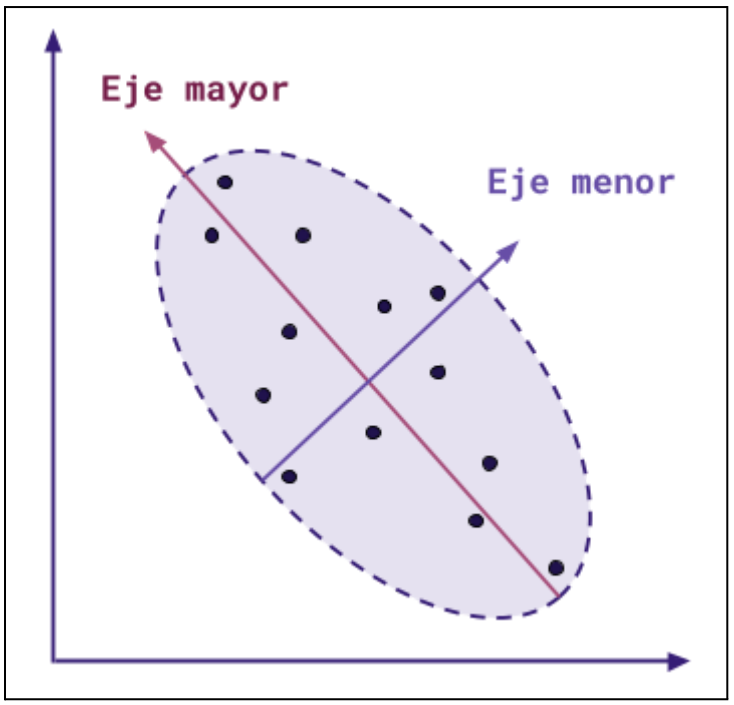

Figura 13. Gráfico del análisis de componentes principales sobre un conjunto de puntos. En el análisis, se calcula el elipsoide que mejor se ajusta a la geometría del conjunto. El eje mayor es la longitud del eje más largo del elipsoide. El eje menor es el eje perpendicular al eje mayor, el más corto en el elipsoide. En tres dimensiones, se agrega el eje mínimo, también perpendicular al eje mayor, y el más corto de los tres.

Las características de forma en 2D son aquellas que describen la geometría de una ROI, y se describen nueve de ellas a continuación.

1. **Superficie de la malla**: es la superficie total de los píxeles que conforman la malla. En este caso A_i es la superficie de un píxel perimetral. Como estos píxeles pueden pertenecer parcialmente a la ROI, su superficie no es uniforme.

$$A = \sum_{i=1}^{N_f} A_i \tag{18}$$

2. **Superficie de pixel**: es la superficie de los píxeles en la ROI. Se consigue de forma aproximada al multiplicar el tamaño de píxel en milímetros cuadrados por la cantidad de píxeles.

3. **Perímetro**: se calcula como la suma de la longitud de todos los segmentos que conforman la malla. Sean *a* y *b* dos vértices que forman una línea en la malla, se calcula la distancia *l* entre ellos.

$$P = \sum_{i=1}^{N_f} \sqrt{(a_i - b_i)^2}$$

(19)

4. **Relación perímetro a superficie**: La división del perímetro por la superficie da una noción de la geometría de la ROI. Si el valor es bajo, entonces la malla se asemeja a una circunferencia.
5. **Esfericidad**: es la razón entre el área circular aproximada de la ROI al perímetro de la misma. Su valor está entre 0 y 1, donde un 1 indica un círculo perfecto.

$$Esfericidad = \frac{2\pi R_{ROI}}{P} = \frac{2\pi}{P}\sqrt{\frac{A}{\pi}} = \frac{\sqrt{2\pi A}}{P}$$

(20)

6. **Máximo diámetro 2D**: es la máxima distancia euclídea entre dos vértices de la malla.
7. **Longitud del eje mayor**: es la longitud del eje mayor en el elipsoide que encapsula a la ROI.

$$Longitud\ de\ Eje\ Mayor = 4\sqrt{\lambda_{mayor}}$$

(21)

8. **Longitud del eje menor**: es el largo del segundo eje de mayor longitud en el elipsoide que encapsula a la ROI.

$$Longitud\ de\ Eje\ Menor = 4\sqrt{\lambda_{menor}}$$

(22)

9. **Elongación**: es la relación entre las mayores componentes principales en la forma de la ROI. Se define de forma inversa a la elongación real por facilidad de cómputo. Su valor se encuentra entre 0 y 1. Si es cero, la elongación es máxima y la región es una línea; si es 1, no hay elongación y la región es un círculo.

$$Elongación = \sqrt{\frac{\lambda_{menor}}{\lambda_{mayor}}}$$

(23)

2.3.1.3 Características de forma en 3D

El análisis en 3D se hace de manera similar al anterior. No se utilizan los valores de intensidad

de la región, sino una malla generada que la contiene. La forma de construir la malla también es similar, con el algoritmo de cubos de marcha: utiliza uno cubo para recorrer la matriz de la VOI, y en cada posición se codifican las aristas del cubo como fuera o dentro (0 o 1), con 256 combinaciones posibles. Muchas de estas combinaciones son redundantes, formando la misma superficie pero en una posición distinta, reduciendo el total de combinaciones efectivas a 14, mostrado en la **Figura 14**.

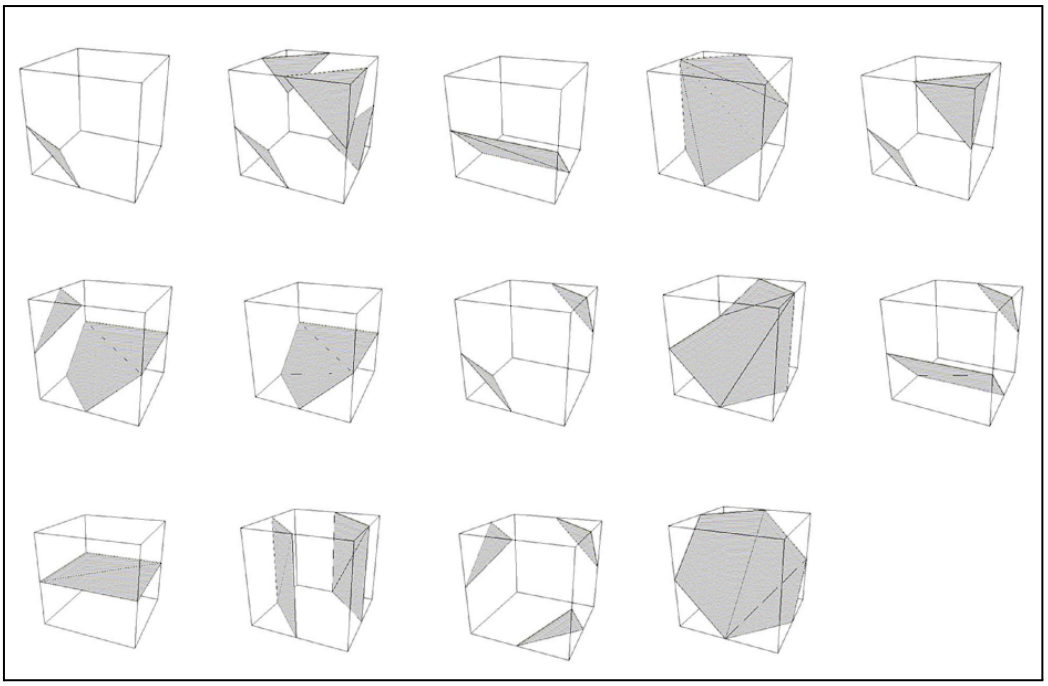

Figura 14. Las 14 superficies generadas por el algoritmo de cubos de marcha [19]. Copyright © 2016, IEEE.

En este tipo de características, se definen la cantidad de vóxeles incluidos en el VOI (N_v), el número de triángulos que definen la malla (N_t), el volumen de la malla en milímetros cúbicos (V), y la superficie de la malla en milímetros cuadrados (A). También se realiza el análisis de componentes principales para extraer esta vez tres ejes ortogonales, en orden de longitud, λ_{mayor}, λ_{menor} y $\lambda_{mínimo}$. A continuación se listan 14 características de forma en 3D, que describen la geometría del VOI.

1. **Volumen de malla**: es la suma de los triángulos que la componen. Por cada triángulos, se calcula el volumen del tetraedro conformado por los tres vértices de la cara (*a, b y c*) y el centro de la malla (*O*).

$$V = \sum_{i=1}^{N_f} \frac{Oa_i \cdot (Ob_i \times Oc_i)}{6} \tag{24}$$

2. **Volumen de voxel**: el volumen aproximado de los vóxeles en el VOI se consigue multiplicando N_v por el tamaño de un voxel.

3. **Superficie**: es la suma de la superficie de cada triángulo que conforma la malla, donde *a*, *b* y *c* definen los lados de un triángulo:

$$A = \sum_{i=1}^{N_f} \frac{1}{2} |a_i b_i \times a_i c_i|$$

(25)

4. **Relación superficie a volumen**: la división de la superficie por el volumen da una noción de qué tan compacto es el VOI. Si el valor es bajo, entonces la malla se asemeja a una esfera.
5. **Esfericidad**: compara la forma del VOI con aquella de una esfera. Su valor está entre 0 y 1, donde 1 indica una esfera perfecta.

$$Esfericidad = \frac{\sqrt[3]{36\,\pi\,V^2}}{A}$$

(26)

6. **Máximo diámetro en 3D**: es la máxima distancia euclídea entre dos vértices de la malla.
7. **Máximo diámetro en 2D (Corte)**: es la máxima distancia euclídea entre dos vértices de la malla en el plano axial.
8. **Máximo diámetro en 2D (Columna)**: es la máxima distancia euclídea entre dos vértices de la malla en el plano coronal.
9. **Máximo diámetro en 2D (Fila)**: es la máxima distancia euclídea entre dos vértices de la malla en el plano sagital.
10. **Longitud del eje mayor**: es la longitud del eje mayor (λ_{mayor}) en el elipsoide que encapsula al VOI. Se define de la misma forma que la Ecuación 21.
11. **Longitud del eje menor**: es el largo del segundo eje de mayor longitud (λ_{menor}) en el elipsoide que encapsula al VOI. Se define de la misma forma que la Ecuación 22.
12. **Longitud del eje mínimo**: es la longitud del eje de menor longitud ($\lambda_{mínimo}$) en el elipsoide que encapsula al VOI.

$$Longitud\ de\ Eje\ Mínimo = 4\sqrt{\lambda_{mínimo}}$$

(27)

13. **Elongación**: es la relación entre los mayores componentes principales en la forma del VOI (λ_{mayor} y λ_{menor}). Se define de forma inversa a la elongación real por facilidad de cómputo. Su valor está entre 0 y 1, donde 0 es la elongación máxima (un VOI bidimensional), y 1 indica que una sección transversal sobre λ_{mayor} y λ_{menor} es circular. Se define de la misma forma que la Ecuación 23.

14. **Aplastamiento**: es la relación entre las componentes $\lambda_{mínimo}$ y λ_{mayor} del VOI. Se define de forma inversa a el aplastamiento real por facilidad de cómputo. Su valor está entre 0 y 1, donde 0 es la elongación máxima (un VOI bidimensional), y 1 indica que una sección transversal sobre $\lambda_{mínimo}$ y λ_{mayor} es circular.

$$Aplastamiento = \sqrt{\frac{\lambda_{mínimo}}{\lambda_{mayor}}} \quad (28)$$

2.3.1.4 Características de textura

Las texturas describen aspectos de una imagen que son difíciles, o incluso imposibles de detectar para el ojo humano. Su cálculo requiere hacer un trabajo adicional sobre las matrices que representan a las imágenes, tal que quede plasmada la información sobre la relación entre los píxeles en una vecindad. Hay diferentes categorías de texturas, diferenciadas por la matriz usada para calcularlas y el aspecto de la imagen en el cual se enfocan. A continuación se muestran algunos grupos de características, y en su explicación, se hace referencia a su cálculo utilizando tanto píxeles como vóxeles.

2.3.1.4.1 Matriz de coocurrencia de niveles de gris (GLCM)

La matriz de coocurrencia de niveles de gris (*Gray Level Co-occurrence Matrix*, GLCM) describe la probabilidad conjunta de pares píxeles, llamados de referencia y vecino, en una imagen o región de ella. Sea una matriz de tamaño *N* x *N,* su GLCM es *P(i, j | δ, θ)*, donde *i* es el índice de las filas, *j* el índice de las columnas, *δ* es la distancia entre el píxel de referencia y su vecino (en cantidad de píxeles), y *θ* es el ángulo en el cual se encuentra ese vecino (0° si se encuentra a la derecha, 45° en la diagonal superior derecha, 90° si está arriba, y 135° en la diagonal superior izquierda). La matriz se hace calculando también en la dirección opuesta de *θ*, resultando simétrica y analizando todas las direcciones [13].

La GLCM es una matriz cuadrada cuyas dimensiones están dadas por el rango de valores posibles en la imagen (su profundidad), sus filas representan a los valores posibles de píxeles de referencia, y sus columnas los valores de píxeles vecinos. Cada celda contiene la frecuencia del par de valores indicado por el índice de fila y columna. A modo de ejemplo, sea matriz la *A*, de dimensión 4 x 4 y un rango de intensidad de 1 a 5, para la cual se calculará la GLCM *P(i, j)* con *δ* = 1 y *θ* = 0:

$$A = \begin{bmatrix} 3 & 2 & 5 & 2 \\ 1 & 2 & 1 & 3 \\ 2 & 3 & 5 & 5 \\ 1 & 2 & 4 & 3 \end{bmatrix} \quad (29)$$

En *A(1,1)* se encuentra el valor 3. El valor que está a su derecha *(θ = 0)* a un píxel de distancia *(δ = 1)* es 2. Por otro lado, si *δ* hubiera sido 2, el vecino tendría valor 5. Si se recorre la matriz pixel a pixel y se mira a su vecino, el par de valores *(3,2)* no se repite. Cuando se hace el

mismo recorrido tomando como vecino al pixel de la izquierda (ya que se considera también al opuesto de *θ* elegido), en la posición *A(2, 1)* vuelve a ocurrir el par *(3, 2)*, y entonces aparece en total dos veces. En *P(3, 2)*, completada más abajo, se coloca entonces un 2, y al ser simétrica, se coloca el mismo valor en *P(2, 3)*.

$$P = \begin{bmatrix} 0 & 3 & 1 & 0 & 0 \\ 3 & 0 & 2 & 1 & 2 \\ 1 & 2 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 & 2 \end{bmatrix}$$

(30)

La mayor información se obtiene al calcular la GLCM para los cuatro ángulos posibles y promediar las matrices resultantes. Para simplificar las expresiones de las características de GLCM, se listan a continuación ciertos valores necesarios en su cálculo [15] [21]:

- *P(i, j)*: GLCM para un determinado *δ* y *θ*.
- *p*(*i*, *j*): GLCM normalizada, presenta la probabilidad de cada par de valores.
- N_g: número de valores discretos de intensidad en la imagen. GLCM es de dimensión N_g x N_g
- $p_x(i)$: la probabilidad marginal de las filas:

$$p_x(i) = \sum_{j=1}^{N_g} p(i, j)$$

(31)

- $p_y(j)$: la probabilidad marginal de las columnas:

$$p_y(j) = \sum_{i=1}^{N_g} p(i, j)$$

(32)

- μ_x: la media de intensidad de p_x, definida como:

$$\mu_x = \sum_{i=1}^{N_g} p_x(i)\, i$$

(33)

- μ_y: la media de intensidad de p_y, definida como:

$$\mu_y = \sum_{j=1}^{N_g} p_y(j)\, j$$

(34)

- σ_x: el desvío estándar de p_x.
- σ_y: el desvío estándar de p_y.

- $p_{x+y}(k)$: donde $k = i+j$ y está entre 2 y 2 N_g:

$$p_{x+y} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \tag{35}$$

- $p_{x-y}(k)$: donde $k = |i-j|$ y está entre 0 y N_g-1:

$$p_{x-y} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \tag{36}$$

- HX: la entropía de p_x:

$$HX = - \sum_{i=1}^{N_g} p_x \log_2(p_x(i)) \tag{37}$$

- HY: la entropía de p_y.

$$HY = - \sum_{j=1}^{N_g} p_y \log_2(p_y(j)) \tag{38}$$

- HXY: la entropía de $p(i, j)$:

$$HXY = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \log_2(p(i, j)) \tag{39}$$

- HXY1: un tipo de entropía usado en el cálculo de texturas [20].

$$HXY\ 1 = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \log_2(p_x(i)\, p_y(j)) \tag{40}$$

- HXY2: un tipo de entropía usado en el cálculo de texturas [20].

$$HXY\ 2 = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_x(i)\, p_y(j) \log_2(p_x(i)\, p_y(j)) \tag{41}$$

En las próximas páginas se dan las fórmulas para 23 características de textura basadas en la GLCM.

1. **Autocorrelación**: en este contexto, define una textura fina o rugosa:

$$Autocorrelación = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j)\, i\, j$$

(42)

2. **Promedio conjunto**: es el nivel de gris promedio de la probabilidad marginal en *x*, μ_x, definido en la Ecuación 35.

$$Promedio\ Conjunto = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j)\, i$$

(43)

3. **Prominencia del grupo** (*Cluster prominence*): mide la asimetría de la GLCM. Si es alto, es asimétrica respecto de la media, mientras que si es bajo, la variación respecto a la media de intensidad es escasa.

$$Prominencia\ del\ grupo = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i + j - \mu_x - \mu_y)^4\, p(i, j)$$

(44)

4. **Sombra del grupo** (*Cluster shade*): mide la asimetría de la GLCM. A mayor valor, mayor asimetría respecto a la media de intensidad.

$$Sombra\ del\ grupo = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i + j - \mu_x - \mu_y)^3\, p(i, j)$$

(45)

5. **Tendencia del grupo** (*Cluster tendency*): mide la tendencia de grupos de vóxeles con valores similares.

$$Tendencia\ del\ grupo = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i + j - \mu_x - \mu_y 2\, p(i, j)$$

(46)

6. **Contraste**: expresa la intensidad de la variación local. Valores altos corresponden a una mayor disparidad de valores entre vóxeles vecinos.

$$Contraste = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - j)^2\, p(i, j)$$

(47)

7. **Correlación**: un valor entre 0 y 1, indicando la ausencia o existencia, respectivamente, de una dependencia lineal en los valores de intensidad respecto a sus vóxeles.

$$Correlación = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)\, i\, j - \mu_x \mu_y}{\sigma_x(i)\, \sigma_y(j)}$$

(48)

8. **Diferencia promedio**: mide la relación de ocurrencia entre pares de intensidad con valores similares y pares de intensidad con valores disímiles.

$$Diferencia\ promedio = \sum_{k=0}^{N_g-1} k\, p_{x-y}(k)$$

(49)

9. **Entropía de diferencia**: es el cálculo de la variabilidad en las diferencias de valores de intensidad.

$$Entropía\ de\ diferencia = \sum_{k=0}^{N_g-1} k\, p_{x-y}(k)\, log_2(p_{x-y}(k))$$

(50)

10. **Varianza de diferencia:** es una medida de heterogeneidad, que da mayor importancia a los pares de intensidad que más se desvían de la media.

$$Varianza\ de\ diferencia = \sum_{k=0}^{N_g-1} (k - \sum_{k=0}^{N_g-1} k\, p_{x-y}(k))^2\, p_{x-y}(k)$$

(51)

11. **Energía conjunta:** es una medida de los patrones homogéneos en la imagen. Si resulta homogénea, habrá una mayor frecuencia de pares repetidos que están cercanos unos de otros.

$$Energía\ conjunta = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,\, j)^2$$

(52)

12. **Entropía conjunta (HXY)**: es una medida de la aleatoriedad en conjuntos de píxeles.

$$HXY = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,\, j)\, log_2(p(i,\, j))$$

(53)

13. **Medida Informativa de Correlación** (*Informative Measure of Correlation,* **IMC**) **1**: estima la correlación entre las distribuciones de probabilidad, actuando también como un estimador de la complejidad de la textura. Si son independientes, no hay correlación y el resultado es cero, mientras que si son completamente dependientes, el resultado será $log_2(N_g)$.

$$IMC\,1 = -\ HXY\ +\ HXY\,1$$

(54)

14. **Medida Informativa de Correlación (IMC) 2**: también mide la correlación de distribuciones, pero está acotado entre [0, 1).

$$IMC\,2 = \sqrt{1 - e^{-2(HXY2 - HXY)}}$$

(55)

15. **Momento inverso de diferencia** (*Inverse Difference Moment*, **IDM**): mide la homogeneidad local. Está ponderado de forma inversa al Contraste, con sus valores disminuyendo de forma exponencial de la diagonal de la GLCM.

$$IDM = \sum_{k=0}^{N_g - 1} \frac{p_{x-y}(k)}{1 + k^2}$$

(56)

16. **Momento inverso de diferencia normalizado** (*Inverse Difference Moment Normalized*, **IDMN**): similar a IDM, se normaliza el cuadrado de la diferencia entre valores vecinos al dividir por el cuadrado de la cantidad discreta de valores de intensidad en la imagen.

$$IDMN = \sum_{k=0}^{N_g - 1} \frac{p_{x-y}(k)}{1 + \left(\frac{k}{N_g}\right)^2}$$

(57)

17. **Diferencia inversa** (*Inverse Difference*)**:** esta medida de homogeneidad incrementa al haber un mayor nivel de valores uniformes.

$$Diferencia\ Inversa = \sum_{k=0}^{N_g - 1} \frac{p_{x-y}(k)}{1 + k}$$

(58)

18. **Diferencia inversa normalizada** (*Inverse Difference Normalized*): en este caso, la normalización se hace sobre la diferencia y no su cuadrado.

$$Diferencia\ Inversa\ Normalizada = \sum_{k=0}^{N_g - 1} \frac{p_{x-y}(k)}{1 + \frac{k}{N_g}}$$

(59)

19. **Varianza inversa:** se define de la siguiente forma.

$$Varianza\ Inversa\ = \sum_{k=0}^{N_g-1} \frac{p_{x-y}(k)}{k^2}$$

(60)

20. **Probabilidad máxima:** es el par de valores vecinos con predominancia máxima y con alta probabilidad de ocurrencia.

$$Probabilidad\ máxima\ = max(p(i,\ j))$$

(61)

21. **Suma de promedio:** estudia la relación de frecuencia de los pares con valores de intensidad bajos y pares de valores de intensidad altos. Al ser la GLCM simétrica, es el **doble del promedio conjunto.**

22. **Suma de entropía**: la suma de diferencias de valores de intensidad en grupos de píxeles.

$$Suma\ de\ entropía\ = \sum_{k=2}^{2N_g} p_{x+y}(k)\ log_2(p_{x+y}(k))$$

(62)

23. **Suma de cuadrados**: mide la varianza de la distribución de pares de intensidad respecto de la media en la GLCM.

$$Suma\ de\ cuadrados\ = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i\ -\ \mu_x)^2 p(i,j)$$

(63)

2.3.1.4.2 Matriz de tamaño de zona de nivel de gris (GLSZM)

La matriz de tamaño de zona de nivel de gris (*Gray Level Size Zone Matrix*, GLSZM) cuantifica las zonas de tamaño de nivel, definidas como un conjunto ininterrumpido de píxeles o vóxeles con el mismo valor de intensidad. En una GLSZM *P(i, j)*, el *(i, j)-ésimo* elemento corresponde a una región con un nivel de gris *i* con *j* elementos conexos. Esta matriz es adireccional, y cada píxel o voxel se considera una sola vez. Sea la siguiente matriz *A(i, j)* y su GLSZM *P(i, j)*:

$$A(i,j) = \begin{bmatrix} 2 & 2 & 5 & 2 & 2 \\ 1 & 2 & 1 & 3 & 3 \\ 1 & 4 & 5 & 5 & 3 \\ 1 & 2 & 4 & 3 & 5 \\ 2 & 4 & 4 & 5 & 5 \end{bmatrix}$$

(64)

A(1,1) tiene el valor 2, y hay otras dos celdas adyacentes del mismo valor. Entonces, en su

GLSZM *P(i, j)* se cuenta una ocurrencia en *P(2, 3)*. *A(1, 2)* ya forma parte de un grupo, y no se tiene en cuenta. *A(1, 3)* vale 5, y no hay celdas adyacentes del mismo valor, por lo que se suma un caso a *P(5, 1)*. En *A(2, 4)*, la celda vale 3, y forma un grupo de 4 celdas conectadas (las adyacentes, y la celda en *A(4,4)* conectada a *A(3,5)*), sumando un caso a *P(3, 4)*. Hay dos ocurrencias para *P(2, 2)*: el grupo formado por *A(1, 4)* y *A(1, 5)*, y el grupo formado por *A(4, 2)* y *A(5, 1)*, entonces *P(2, 2)* vale 2. *P(i, j)* completa es:

$$P(i,j) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(65)

Para simplificar las expresiones de las características de GLSZM, se listan los siguientes valores:

- N_g: es la cantidad de valores de intensidad en la imagen.
- N_t: es la cantidad de zonas de tamaño de nivel en la imagen.
- N_v: es el número de vóxeles en la imagen.
- N_z: es el número de zonas en la ROI, que debe ser menor a N_v.
- *P(i, j)*: es la GLSZM.
- *p*(*i*, *j*): es la forma normalizada de *P(i, j)*.

En las próximas páginas se dan las fórmulas para 16 características de textura basadas en la GLSZM.

1. **Énfasis de área chica** (*Small Area Emphasis,* **SAE**): mide la distribución de zonas pequeñas, donde un valor alto indica mayor presencia de estas zonas, y texturas más finas.

$$SAE = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_t} \frac{P(i,j)}{j^2}}{N_z}$$

(66)

2. **Énfasis de área grande** (*Large Area Emphasis*, **LAE**)**:** mide la distribución de zonas grandes. Un valor alto indica mayor presencia de estas zonas, y texturas más gruesas.

$$LAE = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_t} P(i,j)\, j^2}{N_z}$$

(67)

3. **No uniformidad de nivel de gris** (*Gray Level Non-uniformity*, **GLN**)**:** mide la varianza de la intensidad en la imagen, donde los valores bajos marcan homogeneidad en los grises.

$$GLN = \frac{\sum_{i=1}^{N_g} \left(\sum_{j=1}^{N_t} P(i,j) \right)^2}{N_z}$$

(68)

4. **No uniformidad de nivel de gris normalizada** (*Gray Level Non-uniformity Normalized*, **GLNN**)**:** es la forma normalizada de GLN. Mide la variabilidad de los valores de intensidad en la imagen, marcando mayor similaridad entre ellos si la medida es baja.

$$GLNN = \frac{\sum_{i=1}^{N_g} \left(\sum_{j=1}^{N_t} P(i,j) \right)^2}{N_z^{\,2}}$$

(69)

5. **No uniformidad de zona de tamaño** (*Size Zone Non-uniformity*, **SZN**)**:** es una medida de la variabilidad de las zonas de tamaño de nivel, donde los valores bajos indican homogeneidad entre estas zonas.

$$SZN = \frac{\sum_{i=1}^{N_t} \left(\sum_{j=1}^{N_g} P(i,j) \right)^2}{N_z}$$

(70)

6. **No uniformidad de zona de tamaño normalizada** (*Size Zone Non-uniformity Normalized*, **SZNN**)**:** es la forma normalizada de SZN. Mide la variabilidad en las zonas de tamaño de nivel, indicando homogeneidad en las zonas si el valor es bajo.

$$SZNN = \frac{\sum_{i=1}^{N_t} \left(\sum_{j=1}^{N_g} P(i,j) \right)^2}{N_z^{\,2}}$$

(71)

7. **Porcentaje de zona** (*Zone Percentage*, **ZP**)**:** mide la rugosidad de la textura. Está acotado entre la inversa de N_t y 1, donde los valores cercanos a 1 muestran que la ROI se compone de zonas chicas, siendo más rugosa.

$$ZP = \frac{N_z}{N_v}$$

(72)

8. **Varianza de nivel de gris** (*Gray Level Variance*, **GLV**): es la varianza de intensidad entre las zonas de tamaño de nivel.

$$\mu = \sum_{i=1}^{N_g} \sum_{j=1}^{N_t} p(i,j)\, i$$

$$GLV = \sum_{i=1}^{N_g} \sum_{j=1}^{N_t} p(i,j)(i - \mu)^2$$

(73)

9. **Varianza de zona** (*Zone Variance*, **ZV**): mide la varianza de volumen entre las zonas de tamaño de nivel.

$$\mu = \sum_{i=1}^{N_g} \sum_{j=1}^{N_t} p(i,j)\, j$$

$$ZV = \sum_{i=1}^{N_g} \sum_{j=1}^{N_t} p(i,j)(j - \mu)^2$$

(74)

10. **Entropía de Zona** (*Zone Entropy*, **ZE**): es una medida de aleatoriedad en la distribución de las zonas de tamaño de nivel y los niveles de gris. A mayor valor, los patrones de textura son más heterogéneos.

$$ZE = -\sum_{i=1}^{N_g} \sum_{j=1}^{N_t} p(i,j)\, log_2(p(i,j))$$

(75)

11. **Énfasis de zona de nivel de gris bajo** (*Low Gray Level Zone Emphasis*, **LGLZE**): mide la distribución de las zonas de tamaño de nivel con intensidades bajas. Si el valor es alto, hay mayor prevalencia de zonas de este tipo.

$$LGLZE = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_t} \frac{P(i,j)}{i^2}}{N_z}$$

(76)

12. **Énfasis de zona de nivel de gris alto** (*High Gray Level Zone Emphasis*, **HGLZE**)**:** mide la distribución de las zonas de tamaño de nivel con intensidades altas. Si el valor es alto, hay mayor prevalencia de zonas de este tipo.

$$HGLZE = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_t} P(i,j)\, i^2}{N_z}$$

(77)

13. **Énfasis de zona chica de nivel de gris bajo** (*Small Area Low Gray Level Emphasis*, **SALGLE**)**:** es una medida de la distribución conjunta de zonas de intensidad y tamaño bajo.

$$SALGLE = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_t} \frac{P(i,j)}{i^2 j^2}}{N_z}$$

(78)

14. **Énfasis de zona chica de nivel de gris alto** (*Small Area High Gray Level Emphasis*, **SAHGLE**)**:** es una medida de la distribución conjunta de zonas de intensidad alta pero tamaño bajo.

$$SAHGLE = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_t} \frac{P(i,j)\, i^2}{j^2}}{N_z}$$

(79)

15. **Énfasis de zona grande de nivel de gris bajo** (*Large Area Low Gray Level Emphasis*, **LALGLE**)**:** es una medida de la distribución conjunta de zonas de intensidad baja pero tamaño grande.

$$LALGLE = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_t} \frac{P(i,j)\, j^2}{i^2}}{N_z}$$

(80)

16. **Énfasis de zona grande de nivel de gris alto** (*Large Area High Gray Level Emphasis*, **LAHGLE**)**:** es una medida de la distribución conjunta de zonas de intensidad y tamaño elevado.

$$LAHGLE = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_t} P(i,j)\, i^2\, j^2}{N_z}$$

(81)

2.3.1.4.3 Matriz de largo de franja de nivel de gris (GLRLM)

La matriz de largo de franja de nivel de gris (*Gray Level Run Length Matrix*, GLRLM) es similar a la GLSZM, ya que describe regiones de píxeles con igual valor. En este caso, las regiones son una secuencia de píxeles que, en una dirección, tienen la misma intensidad. Como la GLCM, entonces, la dirección *θ* influye en los valores resultantes.

P(i, j | θ) es la GLRLM en la cual el elemento *(i, j)-ésimo* describe la cantidad de secuencias de píxeles con intensidad *i* y longitud *j* en el ángulo *θ*. Una vez que un píxel forma parte de una secuencia, no vuelve a ser considerado, y las secuencias sólo se forman con píxeles que pertenecen a la misma fila, columna o diagonal. La práctica común es calcular la matriz en todas las direcciones posibles y promediar el resultado. Como ejemplo, se tiene la siguiente matriz *A(i, j)*:

$$A(i,j) = \begin{bmatrix} 2 & 2 & 5 & 2 & 2 \\ 1 & 2 & 1 & 3 & 3 \\ 1 & 4 & 5 & 5 & 3 \\ 1 & 2 & 4 & 3 & 5 \\ 2 & 4 & 4 & 5 & 5 \end{bmatrix}$$

(82)

En *θ = 0*, en la posición *A(1,1)*, hay un píxel de valor 2. A su derecha, en *A(1, 2)* hay otro píxel de valor 2, pero en *A(1, 3)* hay uno de valor 5. Entonces, hay una secuencia de dos elementos con valor 2. Esto vuelve a ocurrir al ver la posición *A(1, 4)*, por lo que *P(2, 2 | 0)*. El análisis de *A(i, j)* en la dirección *θ = 0*, *P(i, j | 0),* es:

$$P(i,j|0) = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 \end{bmatrix}$$

(83)

Para simplificar las expresiones de las características de GLSZM, se listan los siguientes factores:

- N_g: es la cantidad de valores de intensidad en la imagen.
- N_s: es el número de secuencias totales en la imagen.
- N_v: es el número vóxeles en la imagen.
- $N_s(\theta)$: es el número de secuencias en la imagen, a lo largo de la dirección θ.
- $P(i, j \mid \theta)$: es la GLRLM en la dirección θ.
- $p(i, j \mid \theta)$: es la forma normalizada de $P(i, j \mid \theta)$.

Todas las características derivadas de la GLSZM se calculan también para la GLRLM, con la distinción de que se describen las secuencias de vez de las zonas. Para evitar reiterar las fórmulas y descripciones, se listan sólo los nombres de las características de la GLRLM y a qué característica de la GLSZM equivalen. Para su cálculo, se debe tener en cuenta que se debe reemplazar $P(i, j)$ por $P(i, j \mid \theta)$, $p(i, j)$ por $p(i, j \mid \theta)$, N_z por $N_s(\theta)$ y en el límite de sumatorias, N_z por N_s.

1. **Énfasis de secuencia corta** (*Short Run Emphasis*, **SRE**)**:** ver SAE en la Ecuación 66.
2. **Énfasis de secuencia larga** (*Long Run Emphasis*, **LRE**)**:** ver LAE en la Ecuación 67.
3. **No uniformidad de nivel de gris** (*Gray Level Non-uniformity*, **GLN**)**:** ver GLN en la Ecuación 68.
4. **No uniformidad de nivel de gris normalizada** (*Gray Level Non-uniformity Normalized*, **GLNN**): ver GLNN en la Ecuación 69.
5. **No uniformidad de largo de secuencia** (*Run Length Non-uniformity*, **RLN**): ver SZN, en la Ecuación 70.
6. **No uniformidad de largo de secuencia normalizada** (*Run Length Non-uniformity Normalized*, **RLNN**): ver SZNN en la Ecuación 71.
7. **Porcentaje de secuencia** (*Run Percentage*, **RP**)**:** ver PZ en la Ecuación 72.
8. **Varianza de nivel de gris** (*Gray Level Variance*, **GLV**)**:** ver GLV en la Ecuación 73.
9. **Varianza de secuencia** (*Run Sequence*, **RV**): ver ZV en la Ecuación 74.
10. **Entropía de secuencia** (*Run Entropy*, **RE**): ver ZE en la Ecuación 75.
11. **Énfasis de secuencia de nivel de gris bajo** (*Low Gray Level Run Emphasis*, **LGLRE**)**:** ver LGLZE en la Ecuación 76.
12. **Énfasis de secuencia de nivel de gris alto** (*High Gray Level Run Emphasis*, **HGLRE**): ver HGLZE en la Ecuación 77.
13. **Énfasis de secuencia corta de nivel de gris bajo** (*Short Run Low Gray Level Emphasis*, **SRLGLE**)**:** ver SALGLE en la Ecuación 78.
14. **Énfasis de secuencia corta de nivel de gris alto** (*Short Run High Gray Level Emphasis*, **SRHGLE**)**:** ver SAHGLE en la Ecuación 79.
15. **Énfasis de secuencia larga de nivel de gris bajo** (*Long Run Low Gray Level Emphasis*, **LRLGLE**)**:** ver LALGLE en la Ecuación 80.
16. **Énfasis de secuencia larga de nivel de gris alto** (*Long Run High Gray Level Emphasis*, **LRHGLE**)**:** ver LAHGLE en la Ecuación 81.

2.3.1.4.4 Matriz de dependencia con niveles de gris vecinos (GLDM)

La matriz de dependencia con niveles de gris vecinos (*Gray Level Dependence Matrix*, GLDM) describe el nivel de dependencia que existe entre los niveles de gris dentro de la imagen. Se define la matriz *P*(*i*, *j*), donde el (*i*, *j*)-ésimo elemento describe la cantidad de veces que aparece un voxel con nivel de gris *i* con una cantidad *j* de vóxeles dependientes en su vecindad. Un voxel vecino con nivel de gris *j* se considera dependiente con un voxel central a una distancia *δ* si resulta que |i-j| ≦ *α*, donde *α* es un umbral de intensidad elegido. Todos los píxeles de la matriz se ven representados en la GLDM. Como ejemplo, se tiene la matriz *A(i j)*:

$$A(i,j) = \begin{bmatrix} 2 & 2 & 5 & 2 & 2 \\ 1 & 2 & 1 & 3 & 3 \\ 1 & 4 & 5 & 5 & 3 \\ 1 & 2 & 4 & 3 & 5 \\ 2 & 4 & 4 & 5 & 5 \end{bmatrix}$$

(84)

Para *α = 0* y *δ = 1*, en *A(1, 1)*, el píxel central tiene valor 2. Los tres píxeles a distancia 1, *A(1, 2)*, *A(2, 1)*, *A(2, 2)*, tienen valores de 2, 1 y 2, respectivamente. El módulo de la resta del píxel central con cualquiera de los vecinos es menor o igual a 0 en dos casos. Además, se considera al píxel central dependiente consigo mismo, sumando tres píxeles dependientes. Entonces, se suma una ocurrencia en *P(2, 3)*, que ocurre otras dos veces en la matriz *A(i, j)* en *A(1, 2)* y *A(2, 2)*, con lo cual *P(2, 3) = 3*. *P(i, j)* queda de la siguiente forma:

$$P(i,j) = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 4 & 3 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 1 & 2 & 1 \\ 1 & 0 & 3 & 1 \end{bmatrix}$$

(85)

Para simplificar las expresiones de las características de GLDM, se listan los siguientes valores:

- N_g: es la cantidad de valores de intensidad en la imagen.
- N_d: es la cantidad discreta de posibles píxeles dependientes en la imagen.
- N_z: es el número de dependencias en la imagen, coincidente con la cantidad de píxeles en ella.
- *P(i, j):* es la GLDM.
- *p*(*i*, *j*): es la forma normalizada de *P(i, j)*.

Todas las características derivadas de la GLSZM se calculan también para la GLDM, con la distinción de que se describe la dependencia de niveles de gris en vez de las zonas. Para

evitar reiterar las fórmulas y descripciones, se listan sólo los nombres de las características de la GLDM y a qué característica de la GLSZM equivalen. Para su cálculo, se debe tener en cuenta que se debe reemplazar N_s por N_d, con el resto de las notaciones siendo coincidentes.

1. **Énfasis de dependencia chica** (*Small Dependency Emphasis*, **SDE**)**:** ver SAE en la Ecuación 66.
2. **Énfasis de dependencia grande** (*Large Dependency Emphasis*, **LDE**)**:** ver LAE en la Ecuación 67.
3. **No uniformidad de nivel de gris** (*Gray Level Non-uniformity*, **GLN**)**:** ver GLN en la Ecuación 68.
4. **No uniformidad de dependencia** (*Dependency Non-uniformity*, **DN**): ver SZN en la Ecuación 70.
5. **No uniformidad de dependencia normalizada** (*Dependency Non-uniformity Normalized*, **DNN**): ver SZNN en la Ecuación 71.
6. **Varianza de nivel de gris** (*Gray Level Variance*, **GLV**)**:** ver GLV en la Ecuación 73.
7. **Varianza de dependencia** (*Dependency Variance,* **DV**): ver ZV en la Ecuación 74.
8. **Entropía de dependencia** (*Dependency Entropy,* **DE**): ver ZE en la Ecuación 75.
9. **Énfasis de nivel de gris bajo** (*Low Gray Level Emphasis*, **LGLE**)**:** ver LGLZE en la Ecuación 76.
10. **Énfasis de nivel de gris alto** (*High Gray Level Emphasis*, **HGLE**): ver HGLZE en la Ecuación 77.
11. **Énfasis de dependencia chica de nivel de gris bajo** (*Small Dependency Gray Low Gray Level Emphasis*, **SDLGLE**)**:** ver SALGLE en la Ecuación 78.
12. **Énfasis de dependencia chica de nivel de gris alto** (*Small Dependency High Gray Level Emphasis*, **SDHGLE**)**:** ver SAHGLE en la Ecuación 79.
13. **Énfasis de dependencia grande de nivel de gris bajo** (*Large Dependency Low Gray Level Emphasis*, **LDLGLE**)**:** ver LALGLE en la Ecuación 80.
14. **Énfasis de dependencia grande de nivel de gris alto** (*Large Dependency High Gray Level Emphasis*, **LDHGLE**)**:** ver LAHGLE en la Ecuación 81.

2.4 Inteligencia Artificial

2.4.1 Definición de Inteligencia Artificial

La IA es un campo de la ciencia computacional dedicado al desarrollo de sistemas capaces de procesar, analizar y actuar sobre información de manera autónoma [22]. Aunque tradicionalmente se ha asociado la IA con la imitación de la inteligencia humana, hoy en día se reconoce que los sistemas inteligentes pueden ser diseñados para desempeñar tareas específicas sin necesidad de replicar la complejidad de la cognición humana. En la teoría es un área de aplicación amplia, siendo útil para cualquier tarea. En la práctica, la inteligencia es algo difícil de replicar, y estos sistemas se construyen con objetivos concretos. Algunos ejemplos

genéricos de su uso son máquinas para jugar ajedrez, componer música o diagnosticar enfermedades.

Para entender qué es lo que hace la IA, puede ser útil definir qué se entiende por inteligencia. De por sí, es un concepto con más de una definición, pero en lo que concierne a esta rama de la ciencia y su evolución en los años, se han planteado dos paradigmas: *inteligencia racional o humana, y pensamiento o acción* [22].

En el primer paradigma, la inteligencia racional es la puramente matemática, que funciona por un conjunto de reglas para alcanzar una respuesta; la inteligencia humana puede hacer uso de esas reglas, pero es más compleja al poder incorporar una componente psicológica o emocional al proceso de pensamiento, y poder interpretar e interconectar reglas. La IA se ha dedicado mayormente al primer tipo. Es un método ya conocido y adoptado por la ciencia, cuya base en la lógica y matemática hace que los resultados sean reproducibles, mientras que el pensamiento humano aún no es completamente comprendido [22]. Tomando el ejemplo del juego de ajedrez, el enfoque racional es suficiente para ganar: observa las variables en una partida (las posiciones de piezas de cada jugador) y dentro de las reglas del juego, calcula las jugadas posibles. El enfoque humano sería innecesariamente complejo para este problema.

El ejemplo anterior sirve para explicar el segundo paradigma. Por un lado, una corriente dice que como la máquina calculó todas las posibles jugadas, la misma ha pensado y por lo tanto demuestra inteligencia. La otra corriente dice que la máquina debe poder decidir cuál es la mejor opción, y esto es lo que se hace en el juego de ajedrez. Una vez que el programa calcula todas las posibles jugadas (y no sabiendo qué hará el otro jugador), debe elegir aquella que le dará mayores chances de ganar.

La IA de razonamiento matemático y con capacidad de decisión es entonces la más desarrollada y utilizada en la solución de problemas, pero existe una diferencia con nuestra inteligencia que limita la capacidad de las máquinas. A una herramienta de IA se le puede enseñar una tarea, como también se le puede enseñar a adaptarse a cambios en esa tarea, pero está limitada en lo que puede hacer o aprender por cómo fue programada. La máquina o software que fue creada para jugar ajedrez no podrá aprender otro juego a menos que sea preparada para esa nueva tarea.

2.4.2 Tipos de Inteligencia Artificial

Existen diferentes tipos de algoritmos inteligentes que son entrenados y funcionan de formas diversas. Una de las ramas más conocidas de la IA es aquella de los sistemas basados en conocimiento [13]. Estos algoritmos se centran en la utilización de conocimientos expertos y reglas predefinidas para tomar decisiones o realizar tareas. Se construyen a partir de la captura y representación del conocimiento humano en forma de reglas, heurísticas u ontologías, y se utilizan para inferir y razonar sobre situaciones específicas. Hay diferentes tipos de sistemas basados en conocimiento, como por ejemplo, los basados en reglas y los basados en casos.

Los sistemas basados en reglas codifican el conocimiento de expertos en preguntas que, según las respuestas, llevan a diferentes conclusiones. La información de los usuarios es

procesada por un motor de inferencia, que arriba a una respuesta al comparar las entradas con reglas preestablecidas en la base de conocimiento. Estos sistemas son los más sencillos conceptualmente, y los más transparentes, pero su desarrollo es trabajoso si el problema que tratan es complejo.

Un ejemplo podría ser un sistema de diagnóstico médico. Este sistema utiliza reglas y conocimientos médicos expertos para analizar los síntomas y datos del paciente y realizar un diagnóstico. Por ejemplo, si un paciente presenta fiebre, dolor de garganta y tos, el sistema podría inferir que tiene una infección respiratoria basándose en reglas previamente establecidas.

Los sistemas basados en casos funcionan mediante la comparación de un caso de entrada con otros casos ya conocidos y almacenados en una base de datos. Es en base a esta comparación que se arriba a una respuesta, siendo la misma asignada al caso más parecido. Este tipo de sistema es útil cuando las reglas necesarias para llegar a una conclusión no son bien conocidas o codificables. En su forma más simple, requiere de la construcción de una base de datos de casos con características que los identifiquen y una respuesta asociada.

Un sistema de diagnóstico médico también puede ser un ejemplo de un sistema basado en casos. Se podrían registrar síntomas del paciente, como también comorbilidades, datos como la edad, sexo y frecuencia de visitas, y tratamientos realizados. El sistema podría asignar la respuesta del caso más similar a una nueva entrada, pero podría ser más complejo, adaptando la base de datos al contexto de la nueva entrada o interpretando las diferencias con ella. Con cada nuevo caso de trabajo, la base de datos puede crecer si los expertos lo consideran apropiado.

Otra de las ramas de la IA es la del **Aprendizaje Automático** (Machine Learning, **ML**) y su subrama, el **Aprendizaje Profundo** (Deep Learning, **DL**)[13]. Estos sistemas utilizan información de expertos en el tema que son aplicados, pero no la codifican o representan en su funcionamiento. Los algoritmos de ML determinan las reglas o la lógica que llevan a una respuesta por cuenta propia. Para lograrlo, analizan una gran cantidad de casos (cada uno de ellos, representados por datos y con una respuesta asociada), buscan información o correlaciones entre ellos, y producen un modelo que puede generar una respuesta al problema planteado. Con la incorporación de algún tipo de retroalimentación, el algoritmo puede intentar optimizar una métrica de desempeño, que luego se puede utilizar para determinar si el modelo es satisfactorio. El conocimiento de un experto es necesario para construir la base de datos que el algoritmo utiliza al aprender: deben ser casos apropiados a la problemática que se quiere resolver, con características representativas identificas, y un valor de respuesta asociado a cada caso.

Una de las diferencias fundamentales con los algoritmos expertos es el hecho de que en ML los algoritmos generan un modelo que puede dar una respuesta sólo utilizando los datos de entrenamiento, sin necesidad de codificar el conocimiento de un experto. Otra diferencia es que los algoritmos de ML suelen ser menos transparentes en cómo modelan los problemas y arriban a una respuesta, especialmente cuando el problema es complejo.

Al presentar un conjunto de datos con información apropiada a un algoritmo de ML, es posible generar una herramienta para responder a una problemática. Los algoritmos de ML requieren que los datos de entrada sean estructurados, donde cada tipo de dato sigue un mismo formato. Una imagen no puede ser consumida de forma directa por un algoritmo de ML, así que se extraen datos de ella para representarla (como se explicó en la Sección 2.3). Un algoritmo de DL no requiere datos estructurados: es capaz de procesar una entrada como una imagen o un texto para extraer datos. En esta sección no se desarrolla más información sobre el DL, ya que no fue utilizado en el trabajo.

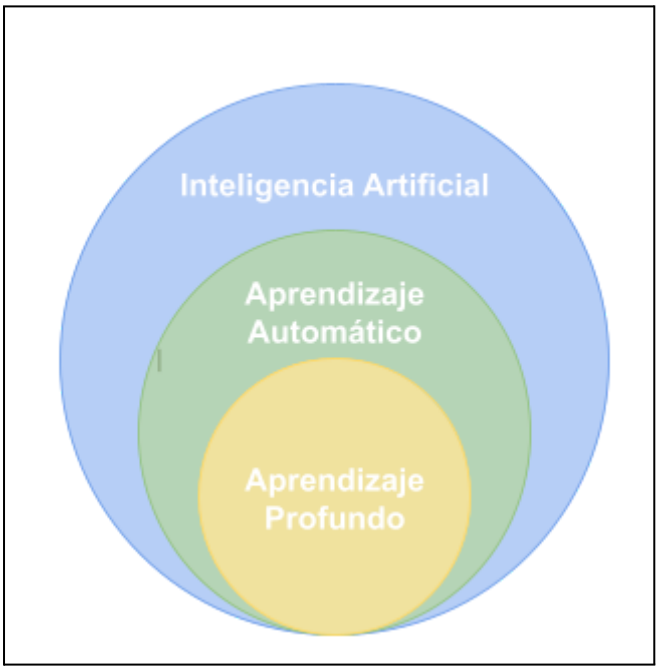

Figura 15. Diagrama simplificado de algunos campos de la IA. El aprendizaje automático es un subcampo de la IA, y el aprendizaje profundo es un tipo particular de aprendizaje automático.

2.4.3 Aprendizaje Automático

El ML evolucionó como un subcampo de la IA que involucra algoritmos de autoaprendizaje que derivan el conocimiento a partir de datos para crear predicciones [25]. En lugar de necesitar al humano para derivar de forma manual las reglas y crear modelos a partir del análisis de grandes cantidades de datos, el aprendizaje automático ofrece una alternativa más eficiente para capturar el conocimiento en datos, mejorar gradualmente el rendimiento de los modelos predictivos y tomar decisiones basadas en esos datos.

Según la definición de Arthur Samuel, "*el Aprendizaje Automático es el campo de estudio que da a los ordenadores la capacidad de aprender sin ser programados de manera explícita*" [23]. Tom Mitchell da una definición más orientada a la ingeniería en donde dice que "*un programa de ordenador aprende de la experiencia E, con respecto a una tarea T y una medida de rendimiento R, si su rendimiento T, medido por R, mejora con la experiencia E*" [24].

Un algoritmo de ML genera un **modelo** en base a **observaciones**. Este modelo funciona como una hipótesis o representación del mundo real y como una herramienta que puede solucionar

problemas. Cualquier situación o tarea que dependa de aprender a relacionar o entender un conjunto de datos para llegar a una conclusión puede beneficiarse de las herramientas de ML [13], [22].

Una observación es un ejemplo o muestra que se le entrega al algoritmo para el aprendizaje. La misma tiene **características,** datos que describen de alguna manera a esa observación. Además, cada observación está vinculada con un resultado, aquello que se desea predecir o detectar**.** Las observaciones que alimentan al algoritmo se conocen comúnmente **variables de entrada, entradas, predictores, variables independientes o variables explicativas,** mientras que el resultado que se quiere explicar, se conocen como **variables de salida, salidas, variables dependientes, o variables de respuesta**. El conjunto de observaciones usado para entrenar o evaluar al modelo es el **conjunto de datos**. Dentro de un conjunto de datos, todas las observaciones independientes deben contar con los mismos tipos y cantidad de características: si se usa el valor de los píxeles, entonces no se puede incluir en un mismo conjunto de datos imágenes con diferente cantidad de ellos. Con estos conceptos introducidos, se pueden explicar los tres tipos de aprendizaje automático: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo [25].

Aprendizaje por refuerzo: este tipo de aprendizaje genera modelos (en estos casos llamados agentes) para resolver problemas donde la respuesta asociada a una entrada no es única o requiere de una interacción para validarla, como muestra el esquema de la **Figura 16**. El agente recibe una retroalimentación según la respuesta producida, y se ajusta para mejorar su capacidad de dar una buena respuesta. El algoritmo que juega ajedrez puede pensarse como un caso de aprendizaje por refuerzo: si las jugadas elegidas por el agente llevan a una victoria, entonces la retroalimentación es positiva, mientras que si llevan a una derrota, es negativa. La retroalimentación podría ser más compleja, considerando no sólo la victoria, sino la cantidad de piezas perdidas o la cantidad de movimientos necesarios para ganar.

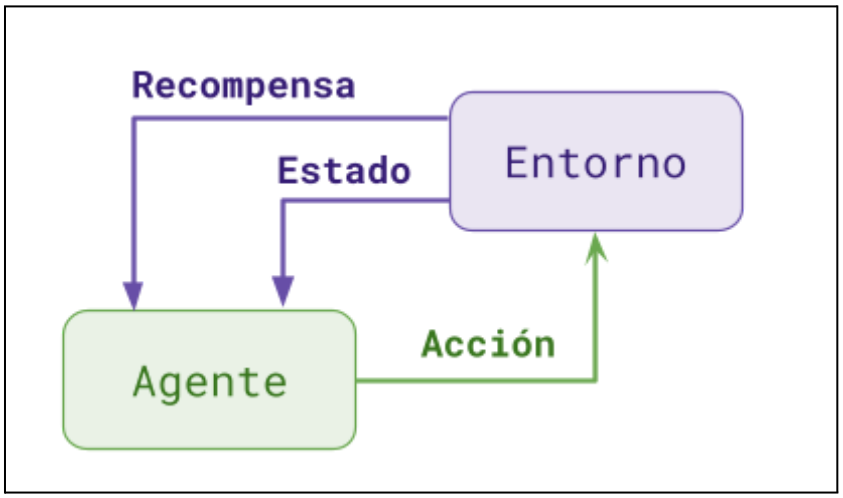

Figura 16. Esquema de aprendizaje por refuerzo. El agente ejecuta una acción sobre el entorno, como mover una pieza de ajedrez en una partida. El agente recibe una retroalimentación, por ejemplo, el conocimiento de haber robado una pieza del oponente; además, se actualiza el estado del agente, que puede ser el estado del tablero tras el movimiento del oponente.

Aprendizaje no supervisado: en el aprendizaje sin supervisión, los datos se tratan sin conocer el valor o la naturaleza de la variable de respuesta. Este tipo de aprendizaje se utiliza para estudiar la estructura de los datos de entrada y extraer información de ellos, encontrando factores comunes que los describan, como se muestra en la **Figura 17**. Una técnica para hacer esto es el agrupamiento, que consiste en organizar grandes cantidades de información en subgrupos significativos sin tener conocimiento previo de las características en común de los datos de entrada. Cada subgrupo generado define ciertas características de similitud, y los datos que pertenecen a ese grupo comparten esas características; el concepto de agrupamiento se muestra en la **Figura 18**, con un ejemplo de tres subgrupos basados en dos características, x_1 y x_2.

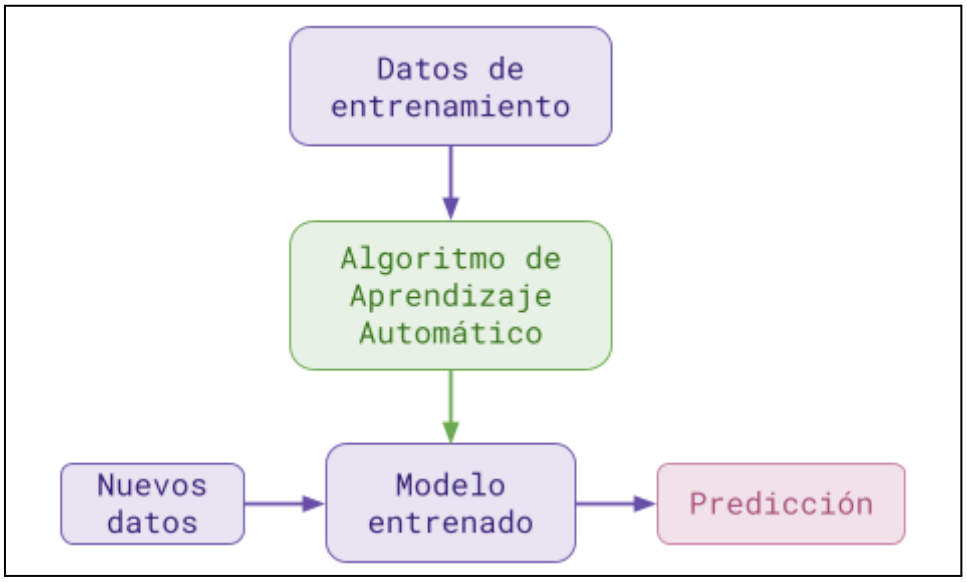

Figura 17. Esquema de aprendizaje no supervisado del tipo agrupamiento . El algoritmo usa los datos de entrenamiento para encontrar patrones e información que permita diferenciar o agrupar a las muestras. Al recibir un nuevo dato, puede comparar sus características con el resto para predecir su pertenencia a un grupo.

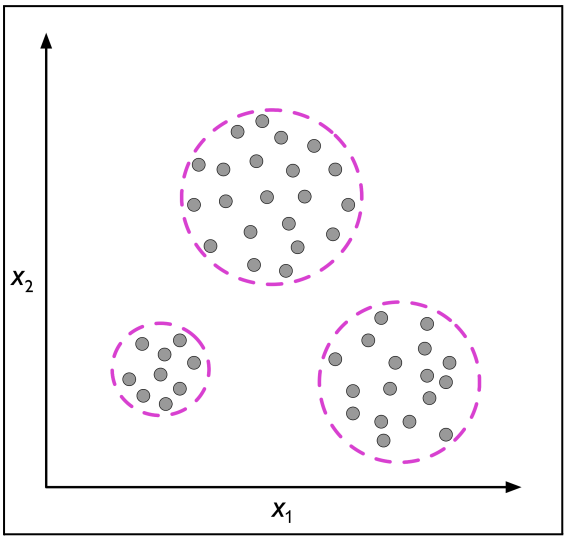

Figura 18. Ejemplo de agrupamiento. Los datos son evaluados según las características x_1 y x_2. Se encuentra que es posible agruparlos en tres subgrupos diferentes, de acuerdo a la similitud de valores en estas características [25].

Aprendizaje supervisado: este tipo de aprendizaje se caracteriza por conocer no solo el valor de las variables explicativas, sino particularmente el valor de la variable de respuesta que se busca predecir usados en el entrenamiento, denominado **etiqueta**. Una vez generado el modelo, se puede utilizar para evaluar una nueva entrada e inferir su valor de salida asociado, como se muestra en la **Figura 19**. Una tarea de aprendizaje supervisado con una variable de respuesta discreta se conoce como **tarea de clasificación**, mientras que si es continua, se habla de una tarea de **regresión**.

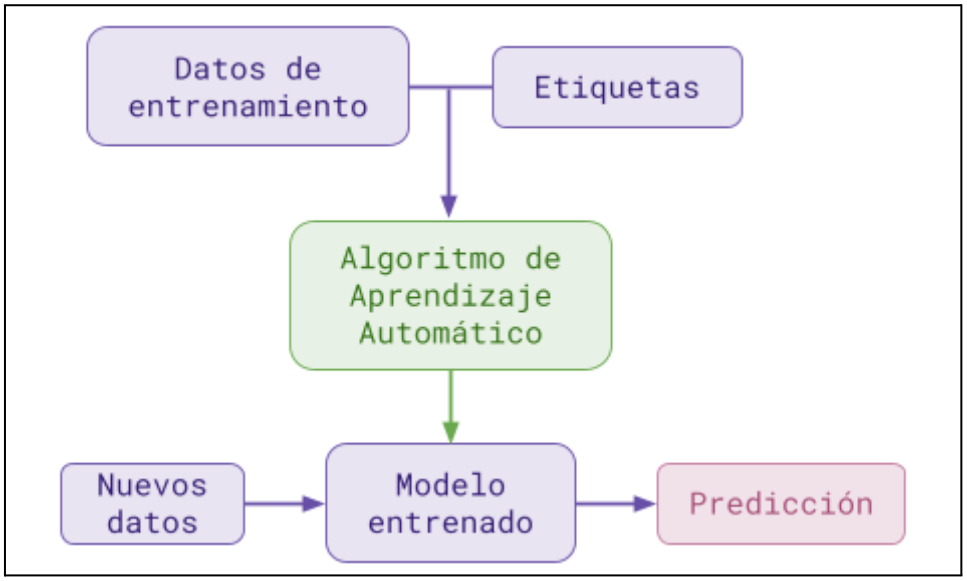

Figura 19. Esquema de aprendizaje supervisado. Los datos de entrenamiento, que deben tener las etiquetas de clase, se utilizan para entrenar al algoritmo que produce un modelo para la tarea de clasificación o regresión. El modelo recibe nuevos datos, sin etiquetar, y responde con una predicción.

En una tarea de clasificación, las etiquetas identifican a las entradas como miembros de una **clase.** La cantidad de clases puede ser igual a dos (clasificación binaria) o superior a dicho valor (clasificación multiclase). En la clasificación binaria, es común que haya una clase que identifica algo que se está buscando, como la presencia de un patógeno, y se la denomina **clase positiva**, mientras que la ausencia de lo que se busca se denomina **clase negativa**. El modelo solo da una respuesta correcta si asigna la clase a la que realmente pertenece una entrada. La **Figura 20** ilustra el concepto de una tarea de clasificación binaria con 30 observaciones de entrenamiento. Una parte pertenece a la clase negativa, y la otra a la clase positiva. Cada observación tiene dos características (x_1 y x_2) asociadas. Un algoritmo de aprendizaje automático supervisado podría establecer una regla para decidir qué combinaciones de valores corresponden a cada clase (representado con una línea punteada):

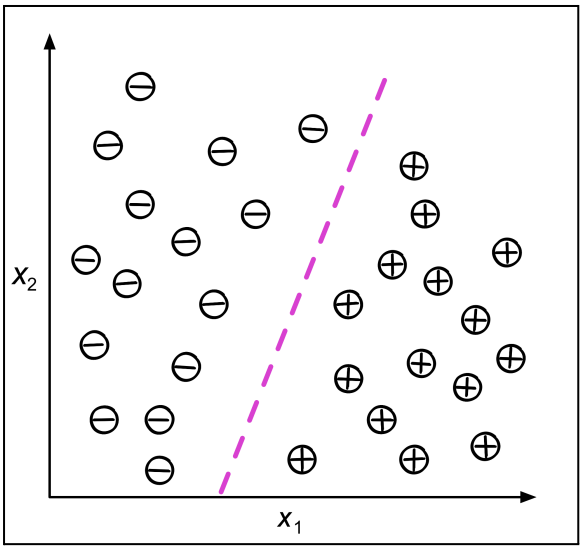

Figura 20. Ejemplo de clasificación binaria. Un algoritmo ha separado el espacio de características x_1 y x_2 con una recta. Nuevos datos en el espacio a la izquierda de la recta pertenecen a la clase negativa, y nuevas muestras a su derecha pertenecen a la clase positiva [25].

En una tarea de regresión, en cambio, la variable de respuesta es un dato continuo. En este caso, la salida del modelo puede no ser exactamente el valor que realmente corresponde a una entrada, pero su error puede ser pequeño y aceptable o grande e inaceptable. La **Figura 21** ilustra el concepto de la regresión lineal. Dada una variable explicativa *x* y una variable de respuesta *y*, se busca la ecuación de la recta que mejor se ajuste a las observaciones (aquella que minimice la distancia de la recta a cada punto). La ecuación de esta recta se puede utilizar con una nueva observación para predecir su valor.

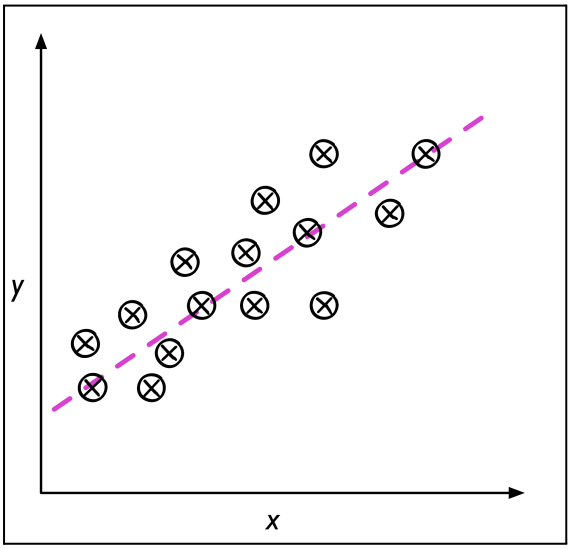

Figura 21. Ejemplo de un algoritmo de regresión. La línea discontinua representa la recta que se ajusta al conjunto de datos. Para una nueva muestra, su valor en *x* y la ecuación de la recta se puede utilizar para conseguir un valor aproximado de *y* [25].

De aquí en más, se desarrollará específicamente el funcionamiento de los algoritmos de aprendizaje supervisado en tareas de clasificación, ya que fue el utilizado en este trabajo. Para

explicar el concepto de la predicción de un modelo, se propone el ejemplo de un clasificador de TC abdominal cuya tarea es diferenciar entre la presencia o ausencia de tumores hepáticos.

En el ejemplo propuesto, una imagen de TC puede ser una observación; las características pueden ser cada uno de los píxeles que representan a la imagen u otros valores calculados que hablen de ella (como las características radiómicas). Además, para cada imagen usada en el entrenamiento, se conoce su clase: la ausencia o presencia de un tumor (clase negativa y positiva, respectivamente). El conjunto de datos utilizado en el entrenamiento queda conformado por la totalidad de pacientes con TC hepática a nuestra disposición, de los cuales sabemos si tienen o no un tumor.

El objetivo de un algoritmo de ML es ajustar un modelo que describa la relación entre las características y las clases. El modelo será una función que transforma los valores de las características para dar un puntaje asociado a la pertenencia a una o varias clases. Dado un modelo *f(x),* una observación *x*, y su clase *y*, el modelo toma las características de *x* y **estima** el valor de la clase:

$$\hat{y} = y + \epsilon = f(x) \tag{86}$$

Donde *ŷ* es la estimación y *ε* el error con el valor real. El error se origina en que el modelo se ajusta de la mejor forma posible a los datos de los cuales ha aprendido, y depende de que esos datos representen la mayoría de los casos posibles en un problema. Como regla general, el modelo no tendrá un buen desempeño si se tienen menos observaciones que características que las describan.

El estimado *ŷ* es el puntaje asignado a la observación, y en un modelo de clasificación binaria, su cercanía a cero indica que el modelo asocia a las características de la observación con la clase negativa, mientras que su cercanía a uno indica una asociación con la clase positiva. Para pasar de este puntaje al valor discreto de clasificación (uno si pertenece, cero si no), se establece un **umbral de decisión**, tal que puntajes mayores o iguales a él se consideran de la clase positiva, y aquellos menores, de la clase negativa; la forma de establecer el umbral se desarrolla en las Secciones 2.4.4.4 y 4.6.1. En un modelo multiclase, hay un puntaje para cada una de las clases, y cada puntaje que supere el umbral indica la pertenencia a una de las clases. En caso de que sean clases exclusivas (cada caso solo puede pertenecer a una de las clases), será el mayor puntaje el que se tenga en consideración. Uno de los objetivos al entrenar un modelo es que este aprenda la distribución de probabilidad condicional *P(*y|*x)* de cada clase *y* dadas las características *x*, conocida como probabilidad a posteriori en la regla de Bayes [33]. De esta forma, podría asignar puntajes a cada observación y comunicar al usuario la certeza con la cual cada observación se asocia a una clase. Si bien los puntajes de salida son comúnmente conocidos como la probabilidad de que una observación pertenezca a una clase, esto solo es así cuando el modelo está calibrado. Si está calibrado, entonces los puntajes de salida del modelo realmente se asemejan a la probabilidad a posteriori; este tema se detalla más en la Sección 2.4.4.4.

2.4.4 Desarrollo de un modelo de Aprendizaje Automático

El proceso de entrenamiento en ML puede plantearse en una secuencia de pasos, como se muestra en la **Figura 22**. En cada uno de los pasos se toman decisiones que darán forma al entrenamiento, y en cada uno de ellos se puede experimentar con diferentes opciones para cambiar los resultados. Esta serie de pasos asume que ya se posee una base de datos con la cual trabajar [13].

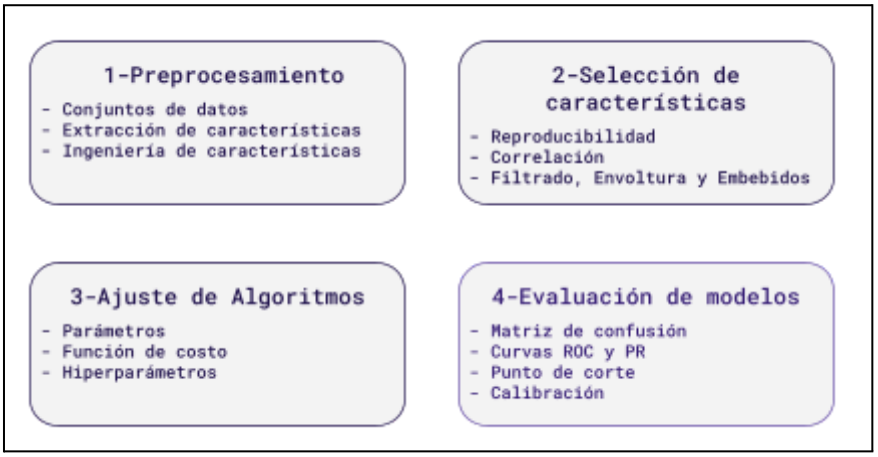

Figura 22. Proceso de desarrollo de un modelo de ML, separado en cuatro pasos secuenciales: Preprocesamiento, Selección de características, Ajuste de algoritmos, y Evaluación de modelos. En cada paso se mencionan algunos temas relevantes al mismo.

2.4.4.1 Preprocesamiento

Esta sección comprende todas las tareas que se realizan sobre los datos con el fin de acondicionarlos al modelo que se quiere entrenar. Los algoritmos de DL tienen cierta capacidad de extraer características de los datos de entrada por cuenta propia. Los algoritmos de ML, por su parte, no tienen esta capacidad y requieren que las características sean introducidas al modelo directamente. Por esta razón, es fundamental no solo asegurar la integridad de los datos a usar, sino también encontrar la forma de presentarlos de la forma más útil para el algoritmo y el problema a resolver, tarea que se llama **extracción de características** [13]**.**

Entre los primeros pasos se puede contar la verificación y exploración de los datos disponibles para el desarrollo del modelo. Es recomendable corroborar que todos las muestras de la base de datos sean de un mismo formato y que estén completas, tanto en sus datos como en sus etiquetas de clase. Si se quiere crear un modelo que clasifique imágenes de gatos y perros, se debería comprobar que todas las entradas en la base de datos sean imágenes de un formato que uno pueda utilizar, y que se tenga la etiqueta de cada una. También es útil visualizar los datos en cierta medida, ya que podrían encontrarse muestras con valores atípicos o que no coincidan con lo que uno busca en la base de datos (podría ser una imagen de muy mala calidad, o que en vez de perros o gatos, tuviera autos). Según cada problema hallado en la base de datos, las entradas pueden corregirse, eliminarse o quedar sin modificar.

Una vez depurada la base de datos, se procede a dividirla en tres subconjuntos: el conjunto de entrenamiento, el conjunto de validación, y el conjunto de evaluación [25]. Es fundamental

evitar la **fuga de datos** (*data leakage*): la presencia de una muestra en más de uno de los conjuntos de datos. Una muestra solo puede pertenecer a uno de los conjuntos [13].

El **conjunto de entrenamiento** es utilizado para ajustar el modelo y aprender los patrones presentes en los datos, y se busca que sea de tamaño suficiente y representativo. En términos de tamaño, debe ser lo suficientemente grande para garantizar que el modelo tenga ejemplos para aprender de los patrones presentes en los datos. Sin embargo, también es importante tener en cuenta que el conjunto puede ser excesivamente grande, llevando a que el modelo no logre aprender o identificar los patrones de una cantidad suficiente de datos. En términos de representatividad, debe ser lo suficientemente diverso para incluir una amplia variedad de ejemplos y patrones. De no ser así, entonces el modelo puede tener dificultades para generalizar a nuevos datos cuyas características no se vean representadas en el conjunto de entrenamiento. Uno de los pasos para lograr esto es asegurar que el conjunto de entrenamiento se seleccione de forma aleatoria y estratificada. Esto quiere decir que la separación respetará la proporción de una variable de interés. Por ejemplo, si se estratifica según la clase de la variable de respuesta, y tres de cada diez casos son positivos, entonces tanto en el conjunto de entrenamiento como en el de evaluación se respetará esa proporción.

El **conjunto de validación** se utiliza para evaluar el rendimiento del modelo durante el entrenamiento y ajustar los parámetros del algoritmo para mejorarlo. Es en base a los resultados del conjunto de validación que se elige al mejor modelo entre todos los entrenados. También se quiere que este conjunto sea grande y representativo, para poder evaluar el rendimiento del algoritmo en muestras diversas.

El **conjunto de evaluación** se utiliza para evaluar el rendimiento del modelo seleccionado, y estos son los resultados que informan sobre cómo sería el desempeño del modelo en la población general. Este conjunto no se usa para ajustar los aspectos configurables del algoritmo, y se busca también que sea suficientemente grande y representativo.

Se debe evitar la fuga de datos en todos los conjuntos, ya que si el modelo ya ha visto una muestra en alguno de ellos, ya conoce la respuesta correcta en caso de encontrarla en otro conjunto. Si esto ocurre, la inferencia sobre esa muestra no estaría representando la verdadera capacidad del algoritmo.

En el ejemplo del estudio de prevalencia de MHCC, ocurre al asignar a un paciente sin la enfermedad al grupo de pacientes que sí la tiene, o viceversa. En caso de que la información se obtuviera por entrevistas, podrían darse otros sesgos si el entrevistador induce la respuesta o el paciente oculta o no recuerda la verdad.

El tamaño de cada conjunto debe considerarse en cada caso particular. Cada conjunto debe ser tan grande como para tener suficientes muestras de todas las clases del problema. El conjunto de entrenamiento siempre es el más grande de los tres, y los conjuntos de validación y evaluación suelen tener un tamaño similar. Independientemente del conjunto al cual pertenezcan las muestras, deben ser procesadas para obtener características relevantes al problema que se busca resolver. En una imagen, la intensidad de cada uno de sus píxeles puede tomarse como una característica.

Es posible que la información cruda de una observación no sea útil como característica para un problema de ML. Esto puede deberse a que esa información no describe apropiadamente el problema, y/o porque es demasiada información como para ser procesada y entendida por el algoritmo. Un corte de TC típico se visualiza mediante una matriz de 512 filas y 512 columnas. Es decir, está compuesta por 262.144 píxeles. A priori no se puede decir que estos píxeles no describen el problema, pero anteriormente se mencionó que no se deberían tener menos observaciones que características. Conformar un conjunto de datos de al menos 262.144 imágenes requeriría un gran esfuerzo, y posteriormente, una gran capacidad de cómputo para que el algoritmo procese los datos.

La extracción de características es el proceso donde se obtienen características significativas a partir de los datos de entrada [13]. Es un paso fundamental para utilizar ML con imágenes o señales, por el ejemplo dado anteriormente. La extracción puede realizarse de varias formas, ya sea de forma directa haciendo mediciones sobre la imagen (por ejemplo, el cálculo de estadísticos como la media, mínimo o máximo de intensidad, entre otros), o de forma automatizada utilizando redes neuronales. Una alternativa semi-automática es la extracción de características radiómicas, explicada en la Sección 2.3.

2.4.4.2 Selección de características

Es posible que la extracción de características resulte en un elevado número de características singulares, que puede ser excesivo por diferentes razones. Por un lado, se podrían tener más características que datos. Por otro lado, por lo general cada característica agregada al modelo lo hace más complejo, y menos interpretable. Además, algunas de las características seleccionadas pueden estar altamente relacionadas entre sí, lo que puede hacerlas redundantes hasta cierto punto. Eliminar características de poca importancia puede agilizar el entrenamiento y mejorar los resultados del modelo generado [13].

A priori, no se conoce qué características son útiles para un modelo y qué características no. Tener más características puede llevar a un mejor ajuste, pero esto incrementa la complejidad del modelo, y tener muy pocas puede llevar a un mal desempeño. Hay tres criterios que se pueden usar para eliminar las características de poco valor.

Primero está la reproducibilidad, que exige que las características tengan una varianza baja para un mismo tipo de dato. El tipo de dato no refiere solo a, por ejemplo, imágenes, sino imágenes de un tejido específico sano y adquiridas de la misma manera. Si la varianza de una característica es alta, la diferencia de su valor caso a caso compromete a la generalización del modelo, incluso cuando su uso en el entrenamiento resulta en un buen ajuste.

Segunda está la correlación, que propone eliminar características que guarden suficiente similitud. Esto se hace calculando la correlación entre cada par de características, y estableciendo un umbral de correlación a partir del cual se considera que son muy similares. Si se encuentra un par de características altamente correlacionadas, se elimina una por considerarla redundante en el modelo. La correlación entre las características se puede visualizar en gráficos como el de la **Figura 23**.

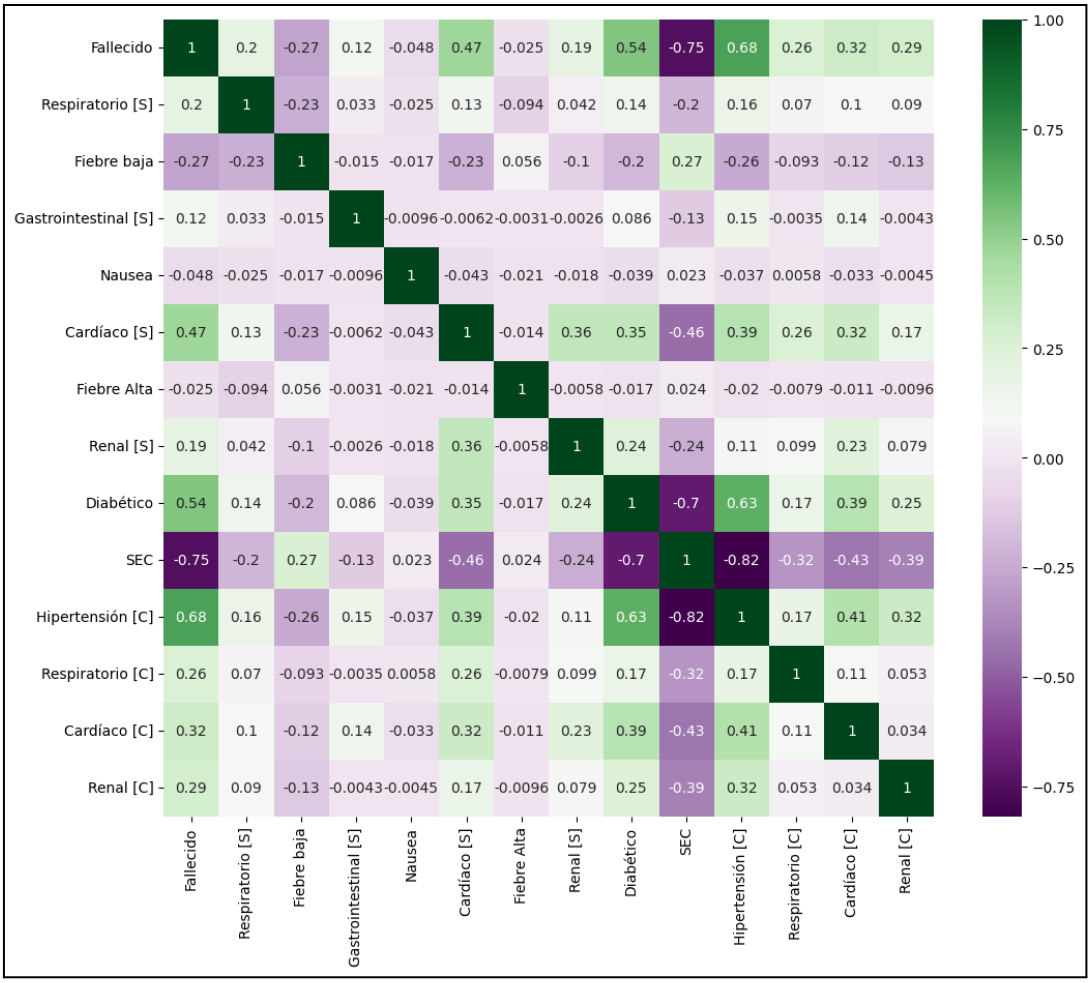

Figura 23. Matriz de correlación generada con datos de comorbilidades para el Covid-19 [26]. La variable de respuesta es Fallecido, y el umbral de correlación en el ejemplo es 0,5. Se encuentran correlacionadas Diabético, Sin Otras Enfermedades Crónicas (SEC), e Hipertensión Crónica. Se podría conservar SEC (alta correlación negativa con la respuesta) o Hipertensión Crónica (alta correlación positiva con la respuesta). La diagonal siempre vale uno, correspondiendo a la correlación de una característica consigo misma.

Tercero, está la selección de características en función de su utilidad para el modelo. Existen métodos para probar la importancia que una característica tendrá en el desempeño de un modelo para resolver una cierta tarea, y se puede seleccionar un número reducido de ellas que solo incluya a las mejores [13]. Existen tres categorías de métodos de selección, mostradas en la **Figura 24**.

Los métodos de **filtrado** aplican una regla de eliminación sobre una relación establecida entre las características y la variable de respuesta. Estos métodos no dependen del algoritmo de ML usado y por eso pueden utilizarse antes del ajuste.

Los métodos de **envoltura** utilizan diferentes combinaciones de características para entrenar un mismo algoritmo y, en base al rendimiento en el conjunto de validación, se elige la mejor combinación.

Los métodos **embebidos** corresponden a algoritmos de ML que incorporan la selección de características dentro de su entrenamiento. Estas estrategias pueden usarse tanto como el

modelo final, o como método para seleccionar pocas características para luego alimentar a otro algoritmo.

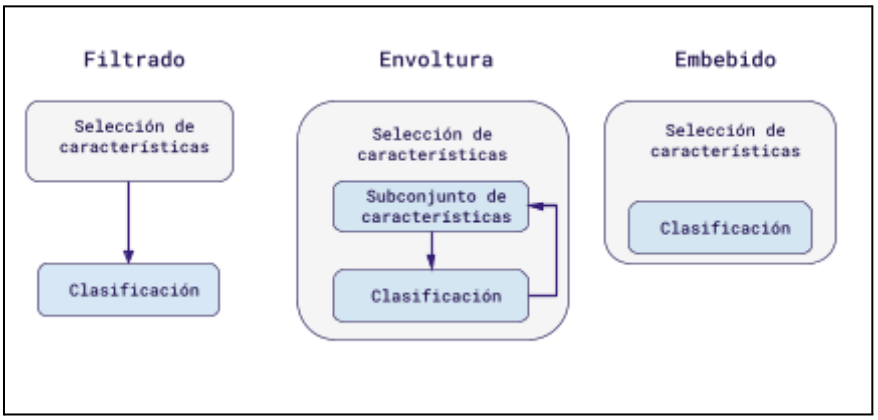

Figura 24. Esquema de los métodos de selección de característica por su funcionamiento: métodos de filtrado, métodos de envoltura, y métodos embebidos.

2.4.4.3 Ajuste de algoritmos

Cuando se entrena un algoritmo de ML, se ajustan los **parámetros** internos que modelizan la relación entre las variables explicativas y la variable de respuesta. Si el modelo consiste en una línea que separa un espacio bidimensional, los parámetros del modelo serán los coeficientes que la definen. El entrenamiento consiste en probar diferentes combinaciones de parámetros de forma iterativa, con algún criterio que le permita entender si una iteración es mejor a la anterior, conocido como **función de costo** [13]. Sea $f(\theta, x_i)$ la función que describe un modelo con parámetros θ, características x de una muestra i, y J la función de costo entre la predicción del modelo y la etiqueta y de una muestra i. Una función de costo J en función de los parámetros θ y condicionada por el conjunto de muestras X y sus correspondientes etiquetas Y, se define en términos generales de la siguiente forma:

$$J(\theta \mid X, Y) = \sum_{i=1}^{n} J(y_i, f(\theta, x_i)) \tag{87}$$

En cada iteración (y en la menor cantidad de ellas), se intenta reducir el error en la mayor medida posible, en otras palabras, con el menor costo posible. Una función de optimización, u **optimizador**, ajusta los parámetros con el objetivo de minimizar la función de costo. Un optimizador común es el algoritmo de gradiente descendiente, que busca la dirección de máximo descenso de la función de costo.

Uno de los objetivos del entrenamiento es producir un modelo que sea generalizable: que sea capaz de asociar a los datos de entrada con su etiqueta real no sólo en los casos de entrenamiento, sino en todos los casos. El ajuste del modelo a los datos de entrenamiento puede resultar en uno de tres casos, como muestra la **Figura 25**, y típicamente se presenta con la analogía de un estudiante preparándose para un exámen [13].

Un modelo **subajustado** es aquel que no ha logrado modelar correctamente las relaciones entre variables en el conjunto de entrenamiento, y también fallará con nuevos datos. En este caso, el estudiante no se ha preparado lo suficiente para el exámen, y no puede resolver los ejercicios o preguntas llegado el caso.

Un modelo **sobreajustado** es aquel que ha modificado sus parámetros para que representen a la perfección la relación entre las variables explicativas y la variable de respuesta en el conjunto de entrenamiento. Un modelo de este tipo está tan ajustado a los datos de entrenamiento, que probablemente tenga problemas para clasificar nuevos datos. En este caso, el estudiante ha memorizado todos los ejercicios o preguntas, pero no conoce realmente el tema, y falla cuando el examen presenta ejercicios que no ha memorizado.

Un modelo con un **buen ajuste** es aquel que ha modelado apropiadamente la relación entre las variables explicativas y respuesta. Es posible que este modelo no tenga las mejores métricas de desempeño en el entrenamiento, pero si tenga suficiente capacidad para inferir sobre nuevos datos. En este caso, el estudiante se ha preparado debidamente en el tema, y aunque tal vez no consiga un puntaje perfecto, logra aprobar el examen.

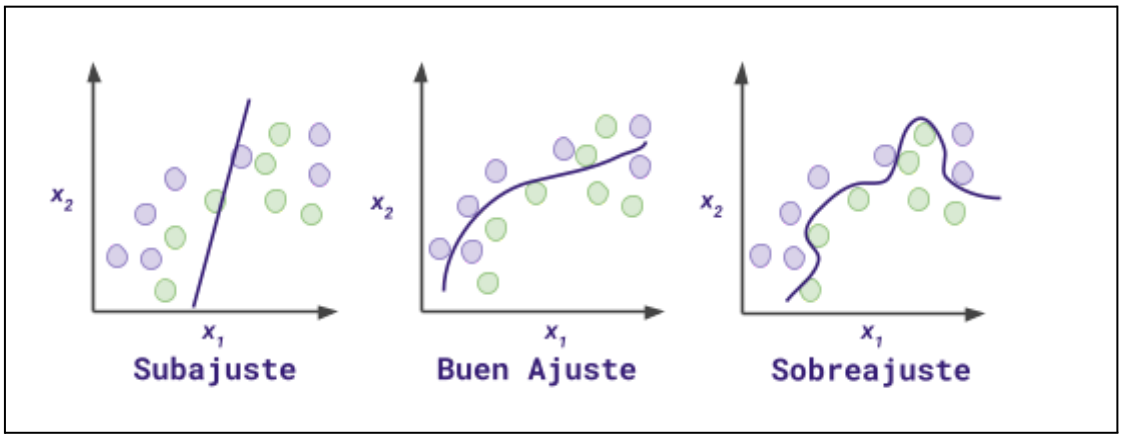

Figura 25. Gráficos de tres modelos de clasificación ajustados a un conjunto de puntos definidos por dos características x_1 y x_2, y pertenecientes a dos clases identificadas por los colores verde y violeta. Izquierda: modelo subajustado. Centro: modelo ajustado. Derecha: modelo sobreajustado.

Hay dos errores a los que se atribuye un ajuste inadecuado: el sesgo y la varianza. El primero corresponde a un modelo que se ha ajustado y produce resultados consistentes pero erróneos. El segundo corresponde a un modelo cuyos resultados ante nuevos datos son muy variantes y poco consistentes [13]. En clasificación, el sesgo está relacionado inversamente con la exactitud (la proporción de aciertos sobre el total de casos), y la varianza está inversamente relacionada con la precisión (la proporción de casos realmente positivos correctamente clasificados sobre el total de casos clasificados como positivos. El efecto de un sesgo o varianza elevada se muestra en la **Figura 26**.

Una forma de controlar el sesgo es permitir al algoritmo más libertad en el ajuste de parámetros, lo que implica utilizar un modelo de mayor complejidad. Esto trae un problema acoplado, en que el algoritmo tendrá mayores capacidades para encontrar los parámetros que se ajustan a la perfección a los datos de entrenamiento, aumentando la varianza; estos

aspectos se deben balancear durante el entrenamiento.

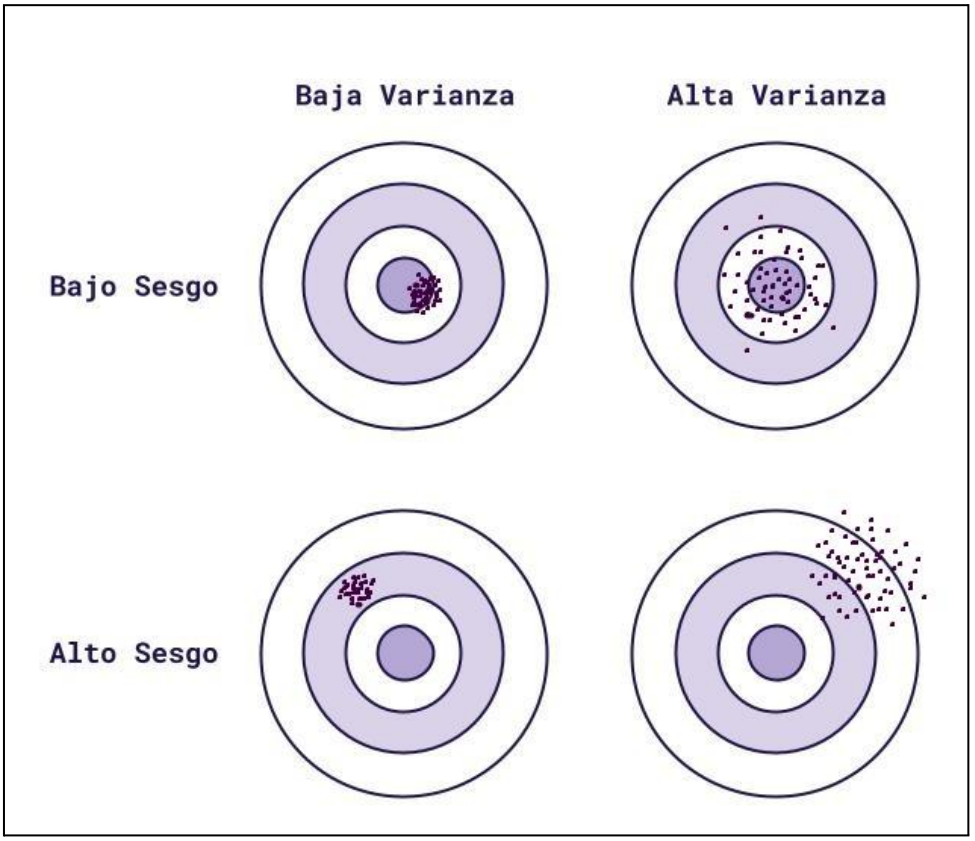

Figura 26. Diagrama representativo de valores bajos y altos de sesgo y varianza y su relación. El caso ideal es aquel de bajo sesgo y baja varianza, en el cual la gran mayoría de los casos se ubican en el centro de la diana.

La varianza se puede combatir sin reducir la complejidad del modelo al agregar más datos. Ya que el algoritmo debe ajustar los parámetros a todos los casos de la mejor forma posible, usar más datos reduce la posibilidad de sobreajuste, pero conseguir más datos puede ser un problema, sobre todo en el ámbito médico. Otra solución es el uso de la **regularización**, que consiste en penalizar la función de costo y prevenir un aumento desmedido de los parámetros, limitando la capacidad del algoritmo de exagerar la importancia de las características [13]. Sea R una función de regularización de los parámetros θ y una constante positiva λ, la función de regularización se suma en la función de costo:

$$J(\theta \mid X,\ Y) \ = \sum_{i=1}^{n} J(y_i,\ f(\theta,\ x_i)) \ + \ \lambda\, R(\theta) \tag{88}$$

De esta forma, si los parámetros son muy elevados, también será alto el costo de la iteración, y el optimizador limitará el incremento de los mismos. La función de regularización puede tomar diferentes formas, como la regularización L1 (que se calcula como la media del valor absoluto de los parámetros) y la regularización L2 (que se calcula como la media del cuadrado de los coeficientes) [27]. El valor de λ determina la fuerza con la cual se hace la regularización, y es un aspecto configurable del algoritmo que impacta en la forma que se hace el entrenamiento del modelo y su resultado final. Este tipo de variables se conocen como **hiperparámetros**;

cada algoritmo tiene hiperparámetros que se pueden modificar antes de iniciar un entrenamiento, como también pueden tenerlos las funciones de costo, optimización y regularización. Diferentes combinaciones de hiperparámetros llevan a diferentes parámetros ajustados en el entrenamiento, que a su vez llevan a mejores o peores resultados. La búsqueda de una buena combinación de ellos se conoce como **ajuste de hiperparámetros**.

Para decidir si un conjunto de hiperparámetros es bueno, se prueba su desempeño en el conjunto de validación, y se comparan los resultados al usar diferentes combinaciones. En casos donde conseguir una cantidad numerosa de muestras para un estudio es difícil (como suelen ser los proyectos en medicina), contar con suficientes imágenes para conformar tres conjuntos de datos (entrenamiento, validación y evaluación) es también complicado.

Una forma de paliar este problema es el uso de la validación cruzada en lugar de un conjunto de validación [13]. Esta técnica consiste en realizar múltiples ajustes en los cuales se cambia qué muestras se usan para el entrenamiento y qué muestras para la validación. Un tipo de validación cruzada es la *K-Fold*, que consiste en subdividir el conjunto de entrenamiento en *k* o pliegues (*folds*) de igual tamaño. Todos menos uno de los pliegues se usan para entrenar al modelo, mientras que la restante se utiliza como conjunto de validación. Una vez hecho esto, se vuelve a entrenar el modelo con los mismos hiperparámetros, pero se cambia el pliegue que queda afuera. Esto se repite hasta haber entrenado al modelo *k* veces. Cada pliegue produce un modelo ligeramente diferente a los otros, y a partir de los resultados de todas los pliegues, se pueden calcular métricas representativas de lo que actúa como conjunto de validación. Es habitual utilizar 5 o 10 pliegues, y el caso particular de utilizar tantos pliegues como datos de entrenamiento se conoce como *leave one out*.

Al componerse de varios conjuntos de datos pequeños, la validación cruzada tiende a entrenar modelos con varianza más elevada. Además, esta técnica requiere que el modelo se entrene varias veces, con lo cual tiene un costo computacional elevado. A pesar de estas limitaciones, es una herramienta que se suele utilizar cuando el tamaño del conjunto de datos no permite hacer una buena división en tres subconjuntos. En estos casos, se divide solo en entrenamiento y evaluación, y se usa la validación cruzada.

El conjunto de validación usado en el entrenamiento de los algoritmos puede confundirse con otro concepto del mismo nombre utilizado en investigación. En el contexto de un estudio, el mismo puede tener **validez interna** (las conclusiones del estudio son válidas para las muestras utilizadas y bajo las condiciones del mismo) y **validez externa** (los resultados son, en cierto grado, extrapolables a la población general).

Las razones por las cuales un estudio puede carecer de validez se encuentran en el tamaño de la muestra utilizada (que fue comentado en esta sección) y en el diseño del estudio. Los errores en el diseño pueden inducir sesgos que afectan a la validez y son trasladados al modelo. Algunos de los sesgos que se pueden producir (de selección, de confusión y de clasificación) se detallan a continuación.

El **sesgo de selección** ocurre cuando los datos utilizados para un estudio no representan adecuadamente a la población general o el problema que se desea resolver. Esto puede

suceder si la muestra de datos se recopila - incluyendo sólo ciertos tipos de casos, o porque la población disponible ya incluía este sesgo. Un ejemplo sería la realización de un estudio sobre la prevalencia de la MHCC en el cual solo se incluyen muestras de mujeres.

El **sesgo de confusión** ocurre cuando hay variables o factores externos que pueden influir en la relación entre las variables de entrada y salida del estudio, y que no han sido consideradas en el mismo. Estos factores pueden generar una correlación espuria y afectar negativamente a los resultados. Los confusores afectan particularmente a la validez interna. El sesgo de confusión impide explicar una correlación entre la variable explicativa y la de respuesta, pero esto no suele ser un problema para los modelos de IA que solo se proponen la predicción o clasificación.

Un ejemplo común es la realización de un estudio de la relación del consumo de café y el riesgo de enfermedades cardiovasculares. Al hacer el estudio, se encuentra que los bebedores de café, con mayor frecuencia, tienen enfermedades cardiovasculares, con lo cual se deduce que es un factor de riesgo para ellas. Pero estas enfermedades pueden ser causadas por otros motivos, como el nivel de actividad de la persona, su dieta, o su consumo de cigarrillos. Al no controlar estas variables en el estudio, actúan como confusores.

Finalmente, el **sesgo de clasificación,** y otros como el entrevistador, el de recuerdo o el de vigilancia ocurren durante la recolección de información para el estudio. Incurrir en ellos lleva a información errónea: datos que corresponden al estado de un paciente, son imprecisos o erróneos. En particular, el sesgo de mala clasificación se refiere a asignar un paciente a un grupo al cual no corresponde.

2.4.4.4 Evaluación de modelos

La salida cruda de un algoritmo de inteligencia artificial en un problema de clasificación es el puntaje asociado a que una entrada pertenezca a una de las clases de salida. Estos puntajes se conocen como *soft scores*, y tienen un valor continuo entre cero y uno. En un problema de clasificación binaria, se considera una clase positiva y una negativa, y el puntaje normalmente se refiere a la pertenencia a la clase positiva. En particular, para las pruebas diagnósticas, se considera positiva la clase que representa el hallazgo, como encontrar un antígeno o la presencia de un tumor.

Los modelos de ML son evaluados utilizando los puntajes de salida, tanto en los conjuntos de entrenamiento, validación y evaluación. Existen métricas que evalúan el desempeño diagnóstico y permiten comparar al modelo con otros métodos de clasificación, sean otros modelos, otros estudios diagnósticos, o la evaluación de profesionales especialistas en el tema. Hay métricas que se calculan con los puntajes, y métricas que se calculan con los resultados de la clasificación. Para poder clasificar cada caso, se debe establecer un umbral de decisión, tal que las muestras con puntaje igual o superior al umbral se consideren positivas, y de lo contrario se consideren negativas. La selección del umbral está ligada al desempeño del modelo mediante el cálculo de estas métricas, ya que se elegirá aquel que maximice alguna métrica de interés. Para un umbral dado (por defecto es 0.5), la clasificación de una muestra puede resultar en cuatro casos: **verdadero positivo (VP)**, la muestra se clasificó como positiva

y ese era su valor verdadero; **verdadero negativo (VN)**, la muestra se clasificó como negativa y ese era su valor verdadero; **falso positivo (FP)**, la muestra se clasificó como positiva, pero su valor real era negativo; y **falso negativo (FN)**, la muestra se clasificó como negativa, pero su valor real era positivo [13]. La **Figura 27** muestra un ejemplo de matriz de confusión.

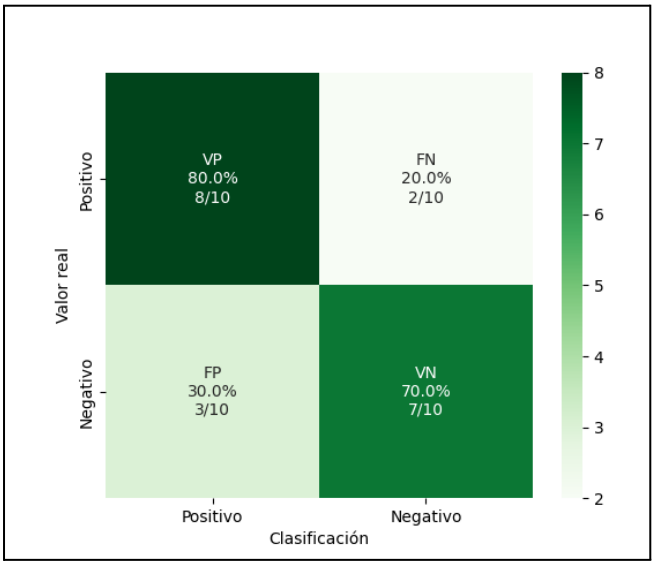

Figura 27. Ejemplo de una matriz de confusión para una clasificación genérica. Se clasificaron 20 casos, 10 positivos y 10 negativos. Cada cuadrante muestra el tipo de clasificación (VP, FN, FP y VN). Además, muestra la proporción y porcentaje de aciertos o errores respecto del valor real de la clase.

Como se dijo anteriormente, estas métricas requieren de la selección de un umbral para poder ser calculadas, y se lo puede elegir para favorecer a una métrica por sobre el resto, o utilizar métodos de cálculo para elegir aquel que resulta en el desempeño óptimo. La curva de la característica operativa del receptor (**ROC**), por otro lado, permite visualizar el desempeño del clasificador para múltiples valores umbrales en un único gráfico. La curva ROC se construye calculando la sensibilidad (**Sen**) y especificidad (**Esp**) para todos los posibles umbrales de clasificación que resultan de los puntajes de salida del modelo en el conjunto de datos en que se está calculando. Para estas métricas, se usará el ejemplo de un clasificador de TC abdominal para la detección de tumores hepáticos.

La **sensibilidad**, también conocida como *recall* y tasa de verdaderos positivos, es la probabilidad condicional de que el resultado de la clasificación sea positivo dado que el caso en cuestión pertenece a la clase positiva. Si el clasificador del ejemplo tiene una sensibilidad de 0,8, entonces hay un 80% de probabilidades de que el resultado sea positivo (se detectan tumores), dado que exista una lesión maligna en la tomografía.

La **especificidad,** también conocida como tasa de verdaderos negativos, es la probabilidad condicional de que el resultado de la clasificación sea negativo dado que el caso en cuestión pertenece a la clase negativa. Siguiendo el ejemplo anterior, si el clasificador tiene una especificidad de 0,8, entonces hay un 80% de probabilidades de que el resultado sea negativo (no se detectan tumores), dado que no exista una lesión maligna en la tomografía.

La curva se grafica como la sensibilidad en función del complemento de la especificidad,

ejemplificado en la **Figura 28**, y sus valores se obtienen a partir de los VP, VN, FP y FN. Las ecuaciones de estas y otras métricas se describen en la Sección 4.6.

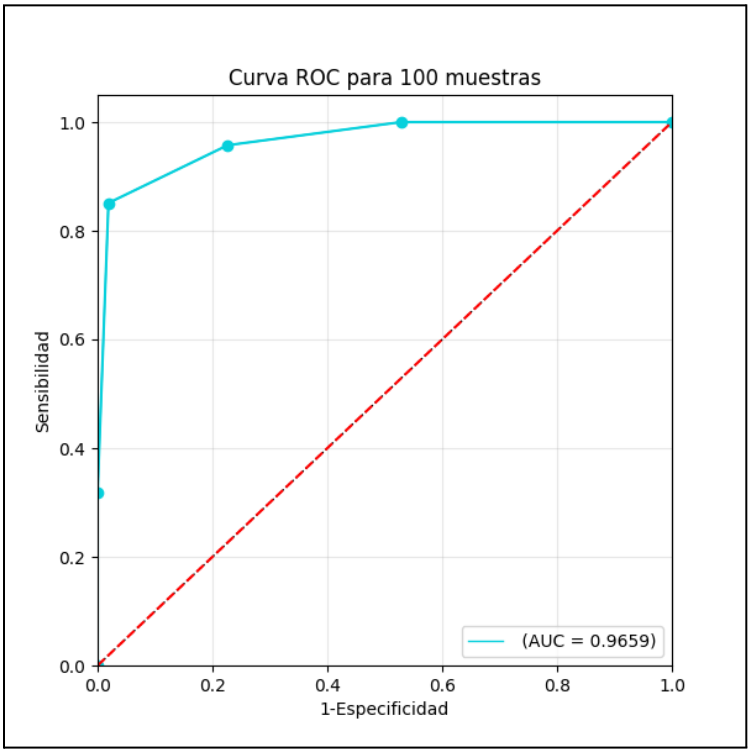

Figura 28. Ejemplo de curva ROC para un clasificador binario a partir de 100 muestras, con la curva en celeste y la línea de no discriminación en rojo punteado. Los puntos celestes corresponden a pares de sensibilidad y especificidad para un umbral (o varios umbrales con el mismo resultado).

Los valores de la curva que más se acerquen a la esquina superior izquierda corresponden a umbrales con valores altos de ambas métricas; al acercarse más a la esquina inferior izquierda, se favorece a la especificidad contra la sensibilidad, mientras que la esquina superior derecha denota lo contrario. El área bajo la curva (**AUC**) es una métrica que habla de la capacidad del clasificador: el valor de 1 es un clasificador perfecto, mientras que un valor de 0.5 representa a un clasificador aleatorio. La línea de no discriminación es una línea recta punteada que representa los puntos donde la clasificación es aleatoria [28].

Un clasificador con un AUC ROC inferior a 0.5 está clasificando casos incorrectamente, pero podría ser consistente en ello: si el clasificador casi siempre determina que los casos positivos son negativos y viceversa, invertir la salida del clasificador puede corregir el problema y resultar en un buen clasificador.

En la curva ROC se puede observar que, dados los posibles umbrales de decisión de una prueba, se pueden obtener valores que favorezcan a la sensibilidad, a la especificidad, o que las métricas sean similares. En general, es posible elegir un umbral que maximice a una métrica a costa de otra: si se establece un umbral de 0,1, es posible que la mayoría de las clasificaciones sea positiva, hayan muy pocos FN, y muchos VP, lo que resulta en una sensibilidad alta. Esto también hace que haya muchos FP, lo que resulta en una especificidad baja. Un umbral muy alto, en general, lleva al resultado opuesto. Qué métrica favorecer

depende del objetivo de la prueba.

Como ejemplo, en las pruebas de cribado (o *screening*, por su denominación en inglés), se busca detectar pacientes de riesgo para una condición médica, idealmente de forma temprana, para su posterior confirmación y tratamiento. En este tipo de pruebas, se prefiere tener el menor número de FN posibles, con lo cual deben tener alta sensibilidad. De esta manera, se reduce la posibilidad de que no sean detectados pacientes que podrían tener o desarrollar una enfermedad y que requieren de un estudio confirmatorio. Tal estudio es una prueba diagnóstica, donde el objetivo es confirmar una condición o patología en pacientes con indicios o factores de riesgo. En estos casos, se quiere minimizar la cantidad de FP, ya que este resultado indica el inicio de un tratamiento que no sería necesario. Sin embargo, depende de cada prueba diagnóstica en qué nivel se puede comprometer a la sensibilidad para favorecer a la especificidad, ya que el costo de un FN (no iniciar un tratamiento cuando fuera necesario) en el diagnóstico podría ser igual o mayor a un FP (iniciarlo cuando no fuera necesario).

Otra métrica de evaluación es la curva de *Precision-Recall* (**PR**), que representa a la precisión (**VPP**) en función de la sensibilidad, ejemplificada en la **Figura 29**. Es similar a la curva ROC, pero se enfoca en la clasificación de la clase positiva, y es usada cuando esa clase está poco representada en el conjunto de datos. En estos casos, la curva ROC puede mostrar un buen resultado en base al desempeño en la clase negativa, que podría opacar el desempeño en la clase positiva. En la curva PR, la línea de no discriminación es recta y corresponde a la proporción de positivos en el conjunto de datos [29].

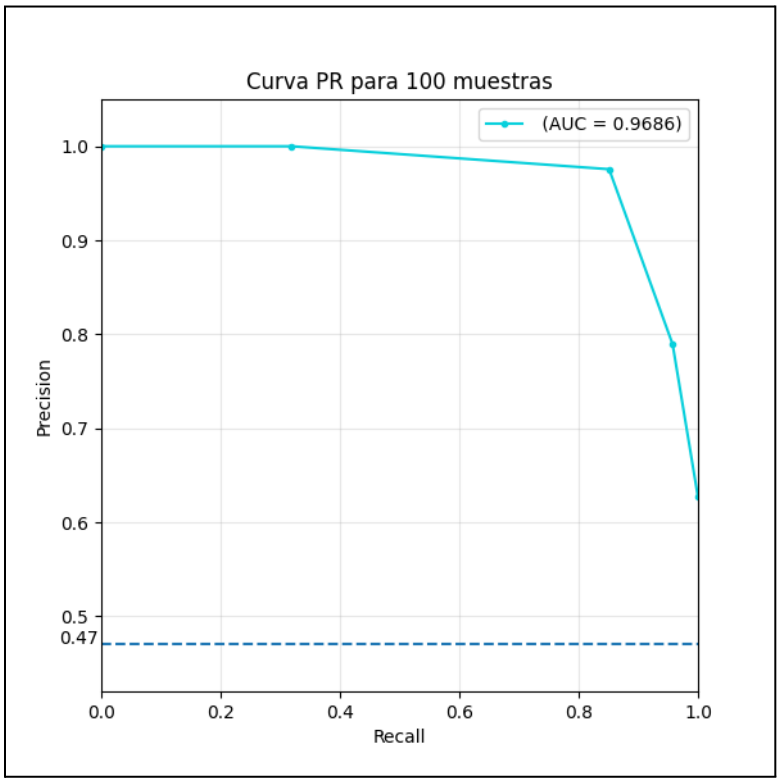

Figura 29. Ejemplo de curva PR para un clasificador binario a partir de 100 muestras, con la curva en celeste y la línea de no discriminación en azul punteado. Los puntos celestes corresponden a pares de precisión y recall para un umbral (o varios umbrales con el mismo resultado).

Si bien la salida final de un clasificador para una entrada suele verse como una predicción o etiqueta que lo asigna a una clase, el puntaje que el modelo le asigna a la entrada de pertenecer a esa clase también puede ser un resultado útil. Es posible que esto presente un problema, ya que los algoritmos pueden producir puntajes que sobreestiman la posibilidad de que una muestra pertenezca a la clase de interés. Un modelo **calibrado** es aquel que asigna puntajes que son realmente probabilidades a posteriori, es decir, la probabilidad condicional *P(y|x)* de cada clase *y* dadas las características *x*. Si un clasificador de TC abdominal con MHCC da un puntaje de 0.7 para un caso, se espera que en un conjunto de 100 TC con determinadas características, 70 de ellas sean positivas. Si el modelo está calibrado, entonces sus probabilidades también pueden usarse como herramienta de interpretación [30]. Esto es buscado en aplicaciones médicas, donde los modelos funcionan como un sistema de soporte a la toma de decisiones. En estos casos, es importante mostrar la confianza del modelo en cada clasificación, por medio de las probabilidades.

Los diferentes algoritmos, por su mismo funcionamiento, son más o menos propensos a estar descalibrados. Por ejemplo, la función de costo de la Regresión Logística o el promediado de múltiples respuestas del Bosque Aleatorio tienden a producir puntajes lejanos de cero o uno (es decir, más calibrados), mientras que Gaussian Naive Bayes tiende a hacer lo contrario [30]. Estos algoritmos serán detallados en la Sección 4.4.2.

La calibración de un modelo se puede visualizar en un gráfico de fiabilidad (*reliability plot*), como se muestra en la **Figura 30**. En este diagrama se contrasta el puntaje de salida (en este contexto, llamado confianza) y la **fracción positiva** (la proporción de casos positivos del total) [31]. En un modelo calibrado, habrá poca o ninguna diferencia entre ambos valores, y en caso de existir una diferencia significativa, se puede determinar si el modelo asigna puntajes demasiado altos (con sobre confianza) o demasiado bajos (con poca confianza).

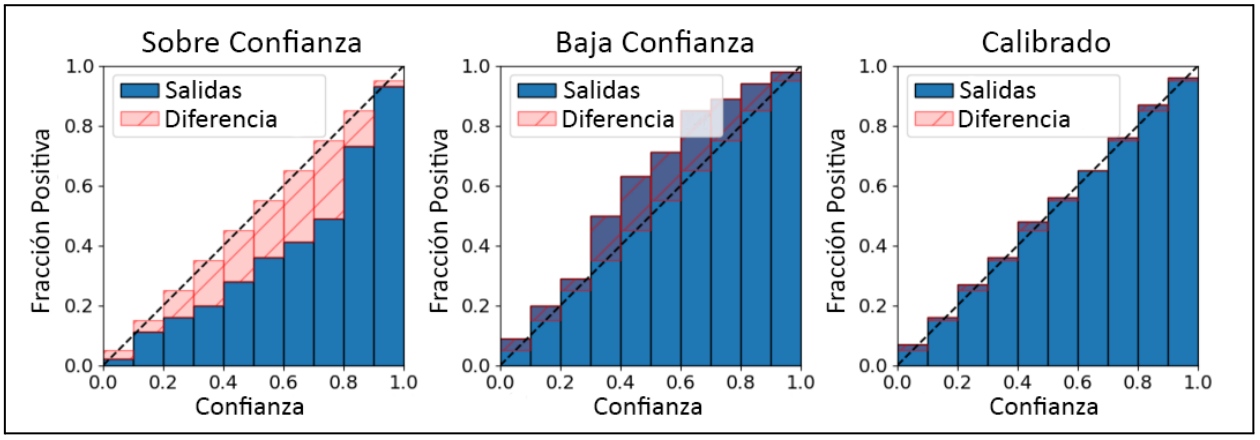

Figura 30. Ejemplo de gráficos de fiabilidad. El eje vertical corresponde a la fracción positiva, y el eje horizontal a la confianza. En color azul se muestran las salidas del modelo, mientras que en color rosa se muestra la diferencia con un modelo calibrado. El gráfico de la izquierda muestra un modelo cuya confianza es mayor a la fracción positiva. El gráfico central muestra un modelo cuya fracción positiva es mayor que su confianza. El gráfico de la derecha muestra un modelo con confianza y fracción positiva similares, y está calibrado. Los bins cuyos extremos están sobre la línea discontinua tienen coincidencia en su exactitud y confianza. Figura adaptada de [32].

2.5 Análisis estadístico

Un proyecto de investigación plantea una hipótesis, una declaración que se quiere validar. Se realiza en un universo reducido, con variables controladas y muestras limitadas, y su objetivo es inferir la validez de la hipótesis en el entorno reducido y también su validez al extrapolar los resultados al universo en general. En esta sección se definen algunos conceptos del análisis estadístico necesarios para validar o refutar las hipótesis de un proyecto de investigación.

En una investigación, la **población blanco** es un conjunto de datos o medidas que cumplen con ciertas características buscadas en el estudio, como la totalidad de pacientes con MHCC. La **población accesible** es aquella disponible para el estudio, por ejemplo, los pacientes cuya información es accesible y utilizable por los investigadores. De la población accesible se toma un subconjunto, llamado una **muestra**. La muestra tiene características cuyo comportamiento se quiere estudiar (en este contexto la variable de respuesta) y asociar con o explicar a través de otra u otras características de la misma (en este contexto las variables explicativas) [33].

Como se explicó en secciones anteriores, el resultado de modelar la relación de variables es un espacio de probabilidad, y solo se puede asegurar que el modelo conozca el espacio para las muestras de la investigación. Para extrapolar los resultados a la población blanco, se **infiere** que el comportamiento es similar en el resto de la población. El Teorema Central del Límite dice que si una población es suficientemente grande, la distribución de probabilidad de sus variables se asemeja entonces a la distribución normal [34]. Como se muestra en la **Figura 31**, esto implica que, conocidas la media μ y el desvío estándar σ de la población, la distribución de probabilidad está centrada alrededor de μ, y una muestra aleatoria de la población tendrá altas probabilidades de que su valor esté entre el rango de $\mu - \sigma$ y $\mu + \sigma$.

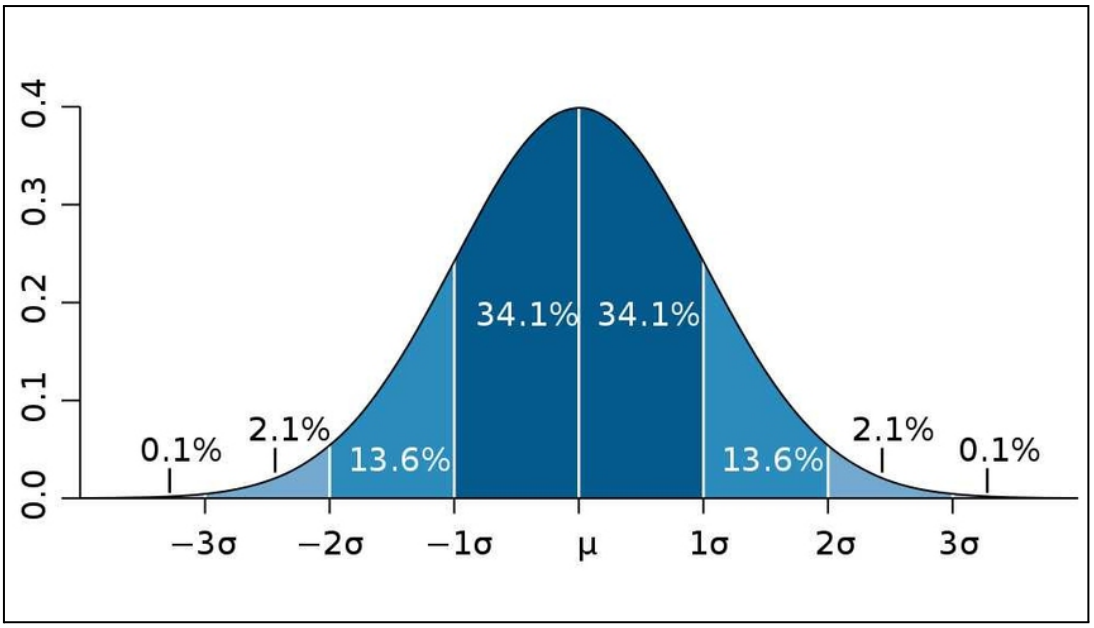

Figura 31: Gráfico de la distribución normal, con el rango de valores de una variable en el eje horizontal, y la probabilidad de que la variable tenga ese valor en el vertical. La distribución de probabilidades está centrada alrededor de la media μ, y se ilustra la proporción de casos en distintos rangos de valores de σ. Por ejemplo, el 34.1% de los casos estará entre $\mu - \sigma$ y μ, mientras que el 68.2% estará entre $\mu - \sigma$ y $\mu + \sigma$ [35].

Cuando una variable en una población tiene distribución normal, se la describe utilizando la media y el desvío estándar. Si no es así, la distribución de las variables no es simétrica respecto a la media, y se las describe mejor utilizando la mediana (en lugar de la media), y el rango intercuartílico (en lugar del desvío estándar). La normalidad de una distribución puede estimarse al estudiar su simetría en un gráfico, o utilizando métodos más robustos, como la prueba de Shapiro-Wilk o la prueba de Kolmogorov [33].

Una forma de inferir sobre la población es mediante la **prueba de hipótesis.** En ella, se plantea una **hipótesis nula**, que niega la existencia de un efecto mediante una relación entre variables explicativa y respuesta. La forma de comparar las variables depende de la pregunta de investigación: por ejemplo, se podría plantear que la supervivencia de pacientes con MHCC y entropía baja es igual que en pacientes con entropía alta. Además, se plantea una **hipótesis alternativa**, que propone que si existe el efecto: siguiendo el ejemplo anterior, se podría plantear que la supervivencia en pacientes con entropía baja es menor a la supervivencia en pacientes con entropía alta. En la prueba de hipótesis, se busca rechazar la hipótesis nula, y por consecuencia aceptar la alternativa. Para esto, se calcula la probabilidad de que sea verdadera la hipótesis nula, el **valor p**, y se establece un límite a partir del cual se considera que es muy poco probable que sea cierta, el nivel de significancia. Si el valor p es menor al límite, se considera al resultado estadísticamente significativo y se rechaza la hipótesis nula.

Hay dos tipos de errores que se pueden cometer en estas pruebas: el **tipo I o α** (rechazar una hipótesis nula verdadera), **y el tipo II o β** (fallar en rechazar una hipótesis nula falsa). Al plantear una prueba de este tipo, se definen inicialmente la hipótesis nula, alternativa, y el valor de α que se acepta. Lo usual es que el valor de α sea 1% o 5%, y es el porcentaje de muestras en una población que se acepta tendrá un comportamiento diferente. Típicamente, esto se expresa como 1 - α, llamado **intervalo de confianza (IC)**. Para α = 5%, se tiene un IC 95%, y es la proporción de muestras que se comportará como lo establece la hipótesis alternativa.

Hay diferentes formas de construir los IC, que se ajustan a diferentes características de las variables, diseño de investigación y distribuciones de probabilidad. Uno de estos métodos es el ***bootstrapping*** [36] [34]. Este método se usa para para poder estimar el valor de métricas de desempeño en la población general. Consiste en tomar muestras con reposición de la población del estudio repetidas veces, y calcular las métricas de desempeño en todas ellas. De esta forma se puede inferir cuál sería el desempeño real del modelo, calculando los IC y valor p. Si se tiene a la población del estudio $X = \{x_1, x_2, \ldots, x_n\}$, con n datos diferentes, una muestra de *bootstrap* del mismo tamaño $Y = \{y_1, y_2, \ldots, y_n\}$ es tomada a partir de X. Al ser tomada con reposición, todos los datos en X tienen la misma probabilidad de ser elegidos para integrar Y, incluso si ya fueron seleccionados. De esta forma, pueden haber datos repetidos, y datos que no se usan. Se muestrea de esta forma un gran número de veces (al menos 1.000), y se calcula el valor de una métrica en cada iteración del *bootstrap*. Los valores de la métrica se ordenan, se calcula su media y se buscan los percentilos α / 2 y 1 - α / 2, que constituyen el IC del *bootstrap*. Por ejemplo, para un α = 0.05, se buscan los percentilos de 0.025 y 0.975, o en porcentajes, 2.5% y 97.5%, que dan el IC 95%.

3. Estado del arte

En la última década se han desarrollado numerosas investigaciones sobre la clasificación de tumores y metástasis utilizando imágenes médicas, utilizando características radiómicas como variables de análisis. También se han encontrado estudios sobre la aplicación de diferentes técnicas de IA, particularmente ML, para estas tareas de clasificación. En esta sección se presenta una serie de trabajos publicados sobre estos temas, que están vinculados a este proyecto final, sea por la temática, metodología u objetivo. Estos trabajos dan cuenta de la factibilidad de emprender una investigación de este tipo, y constituyen una base de conocimiento de las prácticas efectivas para la conformación de conjuntos de datos y planificación del flujo de entrenamiento y evaluación.

3.1 Análisis estadístico y de supervivencia

En este segmento se resumen los resultados de publicaciones que utilizaron características radiómicas de imágenes de TC para estudiar la MHCC, pero donde la metodología del trabajo difiere del uso de ML para la clasificación binaria. Incluso si los resultados de estos estudios no son directamente comparables con los de este proyecto final, han provisto evidencia de la viabilidad del análisis usando características radiómicas.

En los estudios presentados a continuación, la relación entre variables se estudia por diferentes métodos estadísticos según la naturaleza de la variable de respuesta. Por ejemplo, para hallar la conexión de una variable explicativa con una variable de respuesta ordinal (aquella cuyos valores están en una escala ordenada, como puede ser el estado de avance de un tumor), uno de los métodos comunes es la regresión lineal, en la cual se modela la relación entre variables con la ecuación de una recta. Para el caso de una variable de respuesta binaria, se prefiere la regresión logística, donde el resultado de aplicar la transformación logística es la probabilidad de que un caso pertenezca a la clase positiva [33]. El otro caso a introducir es el estudio del tiempo hasta un evento, como sucede en el análisis de supervivencia.

El análisis de supervivencia estudia la correlación de variables con el tiempo entre dos eventos. En general, el evento inicial es el diagnóstico de una enfermedad o el comienzo de un tratamiento, y el final es el fallecimiento del sujeto. Hay diferentes definiciones para la supervivencia: cuando la causa específica de muerte no es relevante, se habla de **supervivencia en general** (*Overall Survival*); cuando solo incluye causas asociadas a una enfermedad que se está estudiando, se habla de **supervivencia específica** (*Disease Specific Survival*); y cuando es de interés el tiempo hasta que se ve una reincidencia de la enfermedad, se habla de **supervivencia sin enfermedad** (*Disease Free Survival*) [37] [38]**.** Diferente es el análisis de mortalidad, donde si bien se establece un rango de tiempo posible para que ocurra el evento final, no es importante el tiempo que pasa hasta que este ocurra. Puede compartir definiciones con los tipos de supervivencia. Por ejemplo, la **mortalidad específica por la enfermedad** (*Disease Specific Mortality*) se refiere al fallecimiento causado por la enfermedad estudiada. Mientras que el análisis de supervivencia da la probabilidad de que ocurra un evento en un tiempo determinado, el análisis de mortalidad solo da la probabilidad de que ocurra el

evento. La relación entre las variables de entrada (como pueden ser diferentes tratamientos) con la supervivencia en general se consigue típicamente con una prueba de rango logarítmico (*Log Rank Test*, para el caso univariado) o la regresión de Cox (en un caso multivariado). En cualquiera de estos casos, la salida del análisis de supervivencia es la probabilidad de que el paciente con las características de las variables de entrada tenga una supervivencia en general mayor a un tiempo determinado [33].

En todos los trabajos que se presentan a continuación, las imágenes provienen de TC abdominal con contraste en FVP, y fueron tratadas con un filtro LdG (σ = 0,5, 1,5, y 2,5), a menos que se especifique otro filtrado o tipo de adquisición. Se utilizaron distintos softwares para el análisis estadístico (MedCalc[1], SPSS[2]), el cálculo de texturas (MATLAB[3], PyRadiomics) y segmentación de lesiones (Scout Liver[4], 3D Slicer)

En 2005 se presentó uno de los primeros estudios de cáncer colorrectal con características de imágenes. Miles et al. [39] realizaron un análisis de supervivencia en pacientes sujetos a resección de cáncer colorrectal. Se estudió la supervivencia en general a dos años, siendo el evento final el fallecimiento debido a metástasis hepática, y se usó una imagen de TC de cada uno de 48 pacientes. En cada caso, la ROI incluyó a la parénquima hepática, que se segmentó manualmente excluyendo vasos, huesos y tejido graso del hígado, y se calculó la media de intensidad y la uniformidad; las variables se normalizaron según los valores del filtro más grueso (σ = 2,5). Se encontró que la uniformidad calculada con filtros 1,5 y 2,0 estaba directamente correlacionada con el fallecimiento de los pacientes.

Las micro metástasis hepáticas (aquellas de tamaño muy pequeño y no detectadas por los métodos habituales) empeoran el pronóstico del paciente, ya que no son tratadas en cirugías de resección. Por este motivo, en 2014, Sheng-Xiang et al. [40] estudiaron la posibilidad de detectarlas con análisis de texturas en pacientes con MHCC. Se compararon pacientes sin hallazgo de metástasis, pacientes con hallazgos en su primera consulta, y pacientes con hallazgos dentro de los 18 meses de haber iniciado el seguimiento. Se usaron imágenes de TC abdominal de pacientes. A partir de una ROI de la parénquima segmentada manualmente (excluyendo vasos, huesos y lesiones visibles), se calculó la media, entropía y uniformidad. Mediante pruebas T de Student o U de Mann-Whitney, se encontró que la entropía y uniformidad, con σ = 1,5 y 2,5, estaban relacionadas a la aparición de metástasis con una AUC ROC entre 0,730 y 0,780. En particular, se halló que la entropía alta y uniformidad baja estaba asociada al órgano con metástasis, lo opuesto a lo hallado en estudios similares previos [41].

Un año después, Sheng-Xiang et al. [42] presentaron otra investigación en análisis de características, donde estudiaron la respuesta de los tumores de pacientes con MHCC a la quimioterapia, y lo compararon con el criterio RECIST 1.1 [43], comúnmente usado para determinar la respuesta a un tratamiento en base al tamaño de las lesiones. Se incluyeron TC anteriores y posteriores a la quimioterapia de 21 pacientes, en las que se segmentó el VOI de

[1] MedCalc for Windows, version 9.2.0.0 (MedCalc Software bv, Ostend, Belgium; https://www.medcalc.org).

[2] Statistical Package for the Social Sciences, version 22.0. (Released 2013. IBM SPSS Statistics for Windows,. Armonk, NY; https://www.ibm.com/ar-es/products/spss-statistics).

[3] MATLAB (MathWorks Inc, Natick, MA, USA; TexRAD Ltd, Somerset, UK; https://la.mathworks.com/).

[4] Scout Liver (Pathfinder Technologies Inc., Tennessee).

todas las metástasis visibles, usando finalmente sólo la de mayores dimensiones. Se calculó la media, la entropía y la uniformidad, pero las variables usadas en el análisis fueron la diferencia entre las texturas entre la imagen posterior y anterior al tratamiento. Se realizó un análisis univariado y multivariado, eligiendo variables no correlacionadas para el segundo. En el análisis univariado, la entropía y uniformidad de imágenes no filtradas fueron los mejores predictores de la respuesta, y los resultados fueron mejores al usar ambas variables en el análisis multivariado. Con estos resultados, los autores concluyeron que el tejido homogéneo (alta uniformidad, baja entropía) se asociaba a un peor pronóstico. Otras investigaciones sobre la respuesta a tratamientos específicos llegaron a resultados concordantes [44] [45] [46].

También en 2015, Lubner et al. [47] presentaron un análisis de supervivencia basado en texturas, donde estudiaron su asociación con diferentes características clínicas, como el estado de mutación del gen KRAS. Usaron imágenes de TC con contraste en FVP de 77 pacientes con MHCC, todas tomadas antes de que se iniciaran tratamientos con quimioterapia. Para cada paciente se eligió el corte con la lesión de mayor diámetro, y se la segmentó de forma manual. Además, se probó analizar todo el VOI en 20 pacientes. Se calcularon todas las características de primer orden, y se encontró una correlación inversa entre el estado del gen KRAS y la asimetría (σ = 1,5) y kurtosis (σ = 2,5). El grado de tumor se encontró negativamente relacionado a la entropía, la media de píxeles positivos, y el desvío estándar en todos los filtros. La entropía también mostró relación negativa con la supervivencia. Estos resultados sumaron evidencia de que los tumores más homogéneos serían potencialmente más agresivos. Los hallazgos sobre la kurtosis y asimetría en el tumor primario y su tendencia a una relación con la mutación del gen KRAS aportaron información sobre un aspecto de la enfermedad que poco se había analizado en estudios de este tipo.

Simpson et al. [48] presentaron en 2017 otro trabajo sobre el uso de texturas para la detección de micrometástasis. Usaron TC tomadas dentro de las 6 semanas previas a una cirugía de resección de 198 pacientes. Se segmentaron dos VOI: uno fue el tumor de mayor tamaño, y otro que se consiguió sustrayendo la imagen posterior a la cirugía de la imagen anterior a la cirugía. La utilidad de este VOI fue analizar lo que sería el hígado remanente, pero en la imagen previa a la cirugía. Se usaron datos clínicos y demográficos, y cinco características de GLCM: contraste, correlación, energía, entropía y homogeneidad, y se descartaron variables para el análisis multivariado usando la correlación de Spearman. Respecto a la supervivencia en general, se encontró que la homogeneidad, contraste, energía y entropía del hígado remanente, y la correlación, homogeneidad y contraste del tumor están relacionadas de forma univariada. En la supervivencia sin recurrencia, las texturas relevantes fueron energía y entropía del hígado remanente, y la correlación y contraste del tumor. En este caso, también se encontró que el tumor homogéneo estaba asociado a mayor riesgo, sea en reincidencia o supervivencia, y la cohorte de este estudio fue la más numerosa entre los trabajos relevados.

Los trabajos en esta sección permitieron identificar algunas prácticas comunes en los experimentos:

- Se utilizaron con mayor frecuencia TC con contraste tomadas en FVP
- Los filtros LdG se usan habitualmente con σ = 0,5, 1,5, y 2,5

- Las características que indican homogeneidad en el tejido se encontraron frecuentemente asociadas asociadas a mal pronóstico.

3.2 Trabajos con Aprendizaje Automático

En esta sección se presentan dos trabajos que utilizaron ML en el tratamiento de MHCC y cuya metodología es comparable a la de este proyecto final.

Publicado en 2019, el trabajo de Liang et al. [51] (realizado con imágenes de TC y RM) tuvo como objetivo predecir la aparición de metástasis hepática metacrónica (la primera metástasis hallada luego del diagnóstico del cáncer primario) en pacientes con cáncer colorrectal.

Se conformó un conjunto de datos de pacientes con TC de abdomen y pecho y RM de recto, todas con contraste, realizadas antes de cualquier tratamiento para cáncer colorrectal. Se buscaron pacientes que no tuvieran evidencia de metástasis o tumores anteriores al cáncer colorrectal. Los 108 pacientes incluidos en el estudio se separaron en negativos (no desarrollaron metástasis, n=54) y positivos (sí desarrollaron metástasis, n=54).

Las imágenes fueron segmentadas manualmente por radiólogos expertos, que delinearon VOI en la FVP de las TC y RM. Se extrajeron características radiómicas para cada paciente, en ambas modalidades de imagen médica. Las imágenes se trataron con cinco filtros distintos (exponencial, cuadrado, raíz cuadrada, logarítmico y *wavelet*). Se incluyeron características de primer orden, de forma, tamaño, de GLCM, GLRM y GLSZM. En total se extrajeron 2058 características, que fueron normalizadas, y se redujo su número con el método LASSO (explicado en la Sección 4.4.1.2).

Se conformaron los siguientes conjuntos de datos: con las características de RM (C_{RM}), con las de TC (C_{TC}), y la combinación de ambos ($C_{combinado}$). Con todos estos conjuntos se hizo una selección de características. Además, se formó un conjunto como la combinación de C_{RM} y C_{TC} después de hacer la selección de características, llamado $C_{óptimo}$.

Se evaluaron los modelos mediante AUC ROC, exactitud, sensibilidad, y especificidad. Los modelos fueron comparados con la prueba de DeLong, usada para probar que la diferencia entre sus AUC ROC es estadísticamente significativa, y se realizó una validación cruzada quíntuple de 100 rondas. El modelado y el análisis de resultados se realizaron en Python con las librerías Scikit-learn y Matplotlib.

Para cada conjunto de imágenes, se seleccionaron cinco características para C_{RM}, ocho para C_{TC} (por lo tanto, 13 para $C_{óptimo}$) y 22 para $C_{combinado}$. La mayoría de las características seleccionadas corresponden a imágenes filtradas con wavelets.

El mejor modelo fue con el algoritmo de Regresión Logística para $C_{óptimo}$, alcanzando un AUC ROC de 0,870, exactitud de 0,800, sensibilidad de 0,830 y especificidad de 0,760. La comparación de cada modelo en los cuatro conjuntos de datos mostró que $C_{óptimo}$ siempre tuvo los mejores resultados.

Este concluyó que el uso de RM y TC en conjunto da mejores resultados que por separado, implicando que las texturas en cada una describen aspectos distintos. Entre las limitaciones de la investigación, se encontraron similitudes con otros estudios sobre MHCC, como la cantidad de pacientes, el uso de imágenes de TC solo en FVP, y el hecho de ser un estudio completamente retrospectivo. Desde el lado de la IA, se probaron pocos algoritmos de selección de características (un sólo método), de ML (dos algoritmos), y no se realizó un ajuste de hiperparámetros.

En 2020, Dercle et al. [50] desarrollaron un modelo integrando radiómica, DL y ML para predecir la respuesta al tratamiento de MHCC. Incluyeron imágenes de TC en dos etapas del tratamiento: una al comienzo del mismo, y otra a las 8 semanas de haber comenzado. La respuesta al tratamiento se determinó en base a la supervivencia de cada paciente, considerándolo sensible si su supervivencia en general superó la media de 17,7 meses de la cohorte, y resistente en caso contrario. También desarrollaron modelos con el objetivo de predecir el estado de mutación del gen KRAS.

Analizaron las imágenes de un ensayo clínico multicéntrico realizado anteriormente, en el que se usaron diferentes tratamientos y se adquirieron imágenes en alta y baja calidad de 661 pacientes. Utilizando esas imágenes, conformaron un conjunto de entrenamiento de 78 pacientes, y un conjunto de validación de 51 pacientes, con imágenes de alta calidad y el mismo tratamiento. Los pacientes se etiquetaron de acuerdo a la respuesta al tratamiento como negativos si fueron sensibles al mismo, o positivos si fueron resistentes al mismo; se consideró a un paciente resistente si su supervivencia en general fue menor a los 17,7 meses. El resto de los pacientes del ensayo se utilizaron para validar el modelo.

Las imágenes fueron segmentadas por un radiólogo experto, generando un VOI para cada lesión hallada. En cada estudio, se extrajeron y promediaron las características de todos los VOI, representativas de todos los tumores en el hígado del paciente.

Se extrajeron dos tipos de características: radiómicas y generadas con un algoritmo de DL. Las características extraídas de las imágenes anteriores y posteriores al tratamiento fueron restadas para producir las características finales que cuantificaron el efecto del tratamiento. A su vez, se redujo la cantidad de características usando un algoritmo de Bosque Aleatorio, identificando las características más reproducibles, informativas y menos redundantes. Se eligieron cuatro características del modelo de ML con mejor AUC ROC.

Utilizaron dos estrategias para construir la firma radiómica, llamadas “gruesa” y “fina”. La estrategia gruesa consistió en seleccionar características en base a tres procesos: un análisis de reproducibilidad (las características extraídas en imágenes levemente modificadas deben ser similares), uno de redundancia (se descartan características con alta correlación entre sí) y la puntuación de las características en diferentes análisis univariados. El análisis fino consistió en la búsqueda de las características óptimas mediante la prueba de sus combinaciones (hasta 20 características), utilizando aquellas filtradas por el análisis grueso. Estas cuatro características se usaron para componer una firma radiómica de las imágenes.

Su mejor modelo predictivo utilizó un Bosque Aleatorio (explicado en la Sección 4.4.2.4) y

cuatro características radiómicas. En el conjunto de entrenamiento, consiguieron un AUC ROC de 0,830 (IC: 0,750 - 0,950), sensibilidad de 0,770 y especificidad de 0,850. En el conjunto de validación, obtuvieron un AUC ROC de 0,800 (IC: 0,690 - 0,940), sensibilidad de 0,800 y especificidad de 0,780. El algoritmo se probó en otros conjuntos de datos (aquellos con imágenes de peor calidad, y con tratamientos diferentes), en los cuales se observaron resultados inferiores.

La señal radiómica se utilizó para entrenar diferentes modelos para la predicción del estado de mutación del gen KRAS en 164 pacientes. El mejor clasificador utilizó el algoritmo Naive Bayes (similar a aquel explicado en la Sección 4.4.2.6), que al evaluarlo en un conjunto de 46 pacientes, alcanzó un AUC ROC de 0,630 (IC: 0,580-0,670).

Este estudio concluyó que las características radiómicas funcionaron como marcadores de los cambios biológicos por la enfermedad y el tratamiento, que describen la heterogeneidad del tejido metastásico y la interacción entre el tumor y la parénquima.

Si bien en esta sección se presentó un número menor de trabajos, se pueden rescatar los siguientes puntos:

- Estos estudios también utilizaron imágenes adquiridas en FVP
- Ambos trabajos iniciaron con una gran cantidad de características y seleccionaron, en el mayor caso, 22 de ellas.
- Los algoritmos usados en el procesamiento, selección de características y entrenamiento fueron diferentes, y en ambos casos se lograron buenos resultados.

4. Métodos

En esta sección se detallan los métodos utilizados para realizar este proyecto: los criterios de selección de pacientes, la adquisición y procesamiento de las imágenes, la extracción de características radiómicas, los métodos de entrenamiento de algoritmos, la forma de evaluarlos, el análisis estadístico realizado y la selección de los mejores modelos. Los pasos seguidos a lo largo del trabajo se esquematizan en la **Figura 32.** Todos los procedimientos se realizaron en Python V 3.7.9 [52], con la excepción de la corrección de imágenes y el *bootstrap* (que se hizo en Google Colaboratory [53], que al momento de realizar el trabajo utilizaba Python V 3.8.0) y la segmentación de los VOI. El desarrollo se realizó en Python por ser el que utiliza el Hospital Italiano de Buenos Aires (HIBA) en todos sus proyectos de IA. El código se ejecutó mayoritariamente en Windows 10, con la excepción de lo ejecutado en Google Colaboratory, que funciona en Ubuntu mediante máquinas virtuales. No se detectaron requerimientos mínimos, pero el acceso a más de 4 *gigabytes* de memoria de acceso aleatorio es recomendada para el uso del programa 3D Slicer [56] y la extracción de características con PyRadiomics.

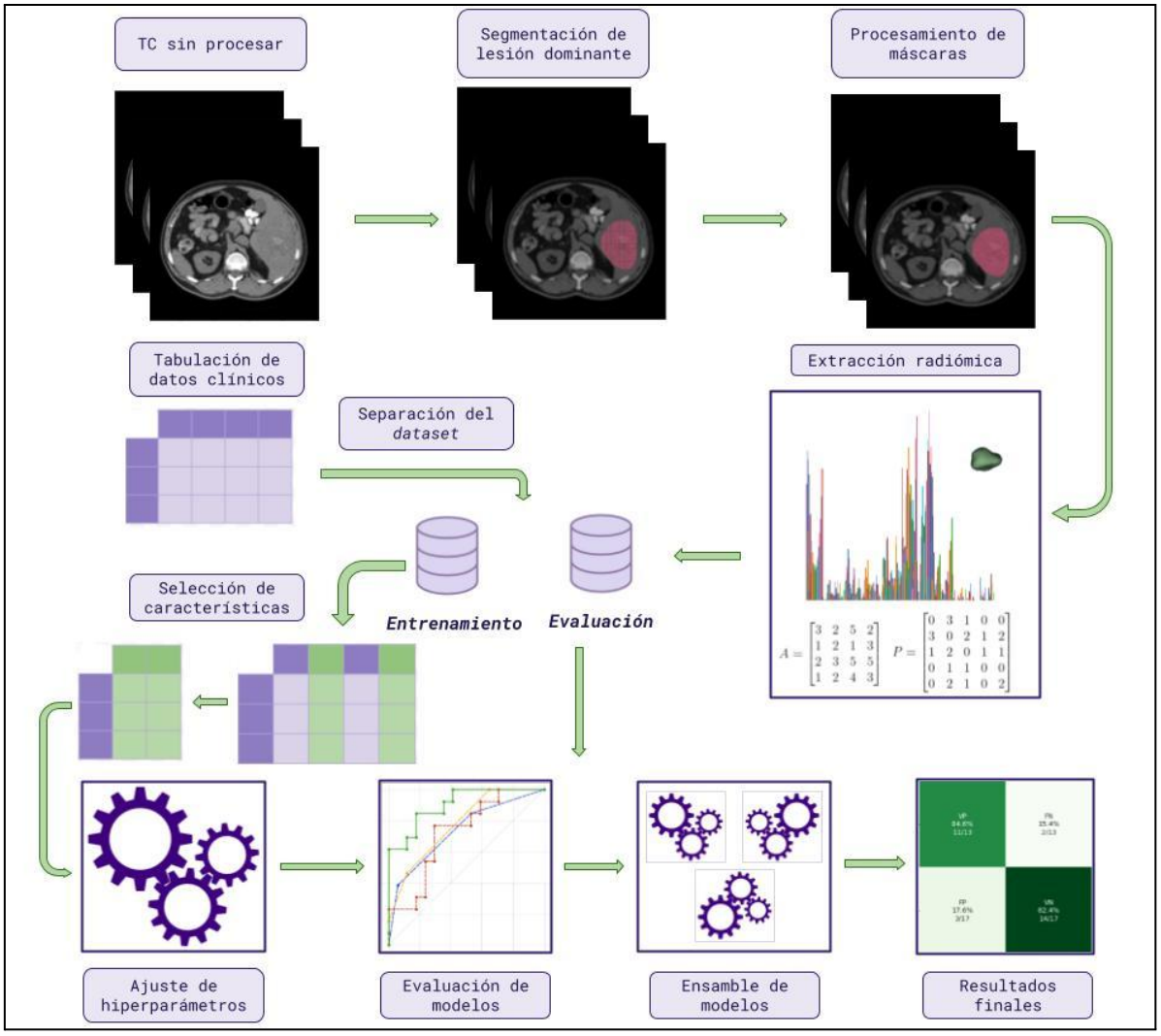

Figura 32. Esquema de los pasos seguidos para la realización de los experimentos. Abreviaturas: tomografía computada (TC).

4.1 Diseño del estudio y población

El proyecto de investigación fue concebido y desarrollado en el HIBA, junto al Servicio de Diagnóstico por Imágenes y el Departamento de Informática en Salud. El HIBA es una organización HIMSS nivel 7+, con desarrollos propio de sistema de información de salud [54], historia clínica electrónica web, un portal personal de salud, un servidor de terminología referenciado a SNOMED CT, y un sistema PACS integrado [55]. Es un líder regional en el desarrollo de sistemas informáticos orientados a los procesos clínicos y asistenciales, y dentro del Área de Investigación e Innovación Tecnológica del Departamento de Informática en Salud funciona el programa de Inteligencia Artificial en Salud (pIASHIBA), dedicado al desarrollo de herramienta de IA aplicadas en el ámbito médico.

Para este proyecto, se plantearon dos objetivos primarios, y múltiples objetivos secundarios. Los objetivos primarios son idénticos en su metodología, y difieren únicamente en la variable de respuesta estudiada. Los objetivos secundarios difieren entre sí tanto por la variable de respuesta estudiada, y difieren de los primarios por el tipo de imagen utilizada.

Como uno de estos objetivos primarios, se definió entrenar y evaluar la capacidad diagnóstica de un clasificador basado en radiómica y ML, para predecir la mortalidad específica a dos años en pacientes con MHCC, utilizando imágenes de TC en FVP. En relación a este objetivo, el comienzo del período de dos años fue la cirugía de resección hepática como tratamiento a la enfermedad, y se definió la variable de respuesta **Óbito,** con dos posibles valores: cero o negativo, si el paciente estaba vivo al finalizar el período, y uno o positivo, si la causa de muerte registrada fue por MHCC.

El otro objetivo primario también consistió en entrenar y evaluar la capacidad diagnóstica de un clasificador basado en radiómica y ML, empleando la misma modalidad y fase de imagen, pero en este caso para predecir el estado de mutación del gen KRAS. Con este propósito se definió la variable de respuesta **KRAS,** con dos valores posibles: cero o negativo para el estado Wild Type, y uno o positivo para el estado mutado, determinado por estudio genético.

Como objetivos secundarios, se planteó el entrenamiento y evaluación de seis clasificadores basados en radiómica y ML utilizando imágenes de TC en fase sin contraste (**FSC**), fase venosa tardía (**FVT**)**,** y fase arterial (**FA**), para la predicción de las variables respuesta Óbito y KRAS. Estos objetivos se definieron como secundarios ya que, como se ha visto en trabajos relacionados, la FVP es considerada la más informativa para esta patología, pero se cree que el uso de múltiples fases podría ser complementario [49].

El estudio fue de carácter **retrospectivo** y **observacional**, utilizando datos ya disponibles en la base de datos del HIBA. Al momento de empadronarse en el hospital, los pacientes firman su consentimiento al uso de sus datos anonimizados para proyectos de investigación. Este proyecto fue presentado y aprobado por el Comité de Ética de la institución bajo el protocolo número 5084. En los términos de un estudio observacional, el desarrollo de un modelo de IA como el de este proyecto tiene características tanto descriptivas como analíticas.

Un estudio **descriptivo** se dedica a recopilar información y características de una población de interés, y el armado de la base de datos corresponde a esta categoría. Un estudio **analítico** tiene el propósito de encontrar o estudiar una relación entre variables. En su objetivo, y en los de este trabajo, los algoritmos de ML y otros de IA modelan la relación entre las variables de entrada y salida, con lo cual su desarrollo cae en esta categoría [34]. Cabe aclarar que si bien es posible modelar la relación, no necesariamente se podrá saber cuál; esto dependerá de qué tan interpretable sea el modelo.

En un estudio de **cohortes**, se estudia la relación entre variables explicativas y de respuesta en un grupo de individuos que poseen dichas variables explicativas a lo largo del tiempo. El estudio de la evolución permite dar mejor evidencia del efecto en las variables de respuesta [34]. Dado que la variable Óbito requiere del seguimiento del paciente, puede parecer que este fue un estudio de cohortes, pero no es así, ya que se usó solo la primera TC de cada paciente.

En un estudio de **casos y controles** se seleccionan individuos que poseen la variable de respuesta (casos) e individuos que no (controles). Se los compara para buscar posibles factores de riesgo y su variación entre los dos grupos [34]. Este tipo de estudios se prefiere para situaciones de baja frecuencia o donde la enfermedad se desarrolle por un período prolongado, pero su propósito no se alinea con el del trabajo.

Se considera que este trabajo fue un estudio analítico de **corte transversal**, en el cual la información de las variables explicativas y de respuesta se obtiene en un mismo momento, y los casos se eligen mediante criterios de inclusión y exclusión. Estos estudios pueden ser descriptivos, y en ese caso se utilizan para caracterizar la prevalencia, por ejemplo, de una enfermedad. En el caso analítico, busca estimar la posible asociación entre las variables explicativas y de respuesta, que en este proyecto está representada por los modelos [34].

Se realizó una búsqueda en la base de datos institucional de pacientes, desde el año 2012 al 2019. Se tabuló la información del diagnóstico, tratamiento y seguimiento de cada paciente, sin sus datos personales. Luego, se consideraron los siguientes criterios de inclusión para incorporar los casos al estudio:

1. Paciente mayor de 18 años.
2. Paciente con diagnóstico de CCR primario y metástasis hepática.
3. Paciente con al menos una cirugía de resección hepática (metastasectomía, segmentectomía o segmentectomía atípica).
4. Paciente con informe histopatológico de la metástasis hepática, previo al tratamiento.
5. Paciente con al menos un estudio de TC abdominal realizado en y en poder del HIBA, previo a la cirugía de resección y con contraste endovenoso, hecho en FVP o con protocolo trifásico hepático.
6. Paciente con seguimiento postquirúrgico de al menos dos años.

Además, se consideraron los siguientes criterios de exclusión:

1. Paciente con cirugías hepáticas previas a realizarse la TC.
2. Paciente que había sido tratado con quimioterapia antes de realizarse la TC.
3. Paciente cuyo estudio de TC poseía artefactos de movimiento, o estaba incompleto.

Se tabuló la información de aquellos pacientes que cumplieron los criterios de inclusión y superaron los de exclusión. Luego, se conservaron aquellos casos en los que el paciente aún estaba vivo pasados dos años de la cirugía (en Óbito, los casos negativos) y de aquellos en el que el paciente había fallecido por la enfermedad (en Óbito, los casos positivos). Los pacientes que habían fallecido por otra causa fueron descartados.

4.2 Adquisición, segmentación y preprocesamiento de imágenes

Debido a que el estudio es retrospectivo, no se realizaron TC específicamente para el mismo, y las características de la adquisición no son uniformes, teniendo casos hechos en protocolo trifásico hepático y en protocolo de FVP. En general, el hígado se estudia con el protocolo trifásico hepático, donde las diferentes fases son útiles en la identificación de diferentes patologías. Cuando una TC se hace como parte del seguimiento de un caso ya conocido, se adquiere solo en la fase relevante a ese caso para evitar irradiar excesivamente al paciente.

El protocolo trifásico hepático se realiza utilizando 1,5 ml/kg de contraste iodado (iobitridol) y 50 ml de solución fisiológica posterior al contraste, con un flujo de 3.5 ml/seg. Consta de 4 fases de adquisición que se coordinan usando un bolo de contraste (FSC, FA, FVP y FVT). La FA

hepática se dispara a los 8 segundos luego de que el bolo de contraste llega a 180 UH en la aorta abdominal; la FVP se toma a los 60 segundos después de la inyección del contraste; y la FVT se adquiere a los 180 segundos después de la inyección. La captura del protocolo en FVP se hace de la misma forma que el protocolo trifásico, pero solo se captura en el momento indicado para la FVP.

Las imágenes se capturaron desde el domo hepático hasta las crestas ilíacas, con la excepción de la FVP, que incluyó hasta la sínfisis púbica. Las imágenes se adquirieron en tomografía multicorte Toshiba Aquilion 64 y Toshiba Aquilion One 320. Los parámetros de la adquisición fueron: 120 kVp, 200 a 400 mA (según las dimensiones del paciente), y espesor de corte de 1mm.

Un grupo de radiólogos y oncólogos expertos con al menos 4 años de experiencia en el hospital recolectaron los datos y segmentaron las imágenes. Los nombres de los pacientes fueron reemplazados por identificadores alfanuméricos para anonimizar los estudios. La segmentación se llevó a cabo con el software gratuito 3D Slicer, que permite delimitar las áreas de interés en una imagen y compilarlas en un VOI. El resultado se exporta como una máscara binaria del VOI, donde los píxeles pueden valer 0 (no pertenece a la región) o 1 (pertenece a la región). Los volúmenes y máscaras fueron guardados en formato NIfTI, eliminando así datos identificatorios de los pacientes presentes en el DICOM original.

Un ejemplo del proceso de segmentación se muestra en la **Figura 33**. Para cada paciente, se segmentó solo la lesión hepática dominante, entendida como aquella de mayor tamaño. Durante la segmentación, las imágenes fueron visualizadas con una ventana de UH para tejido de hígado (Centro 40, Ancho 400). Los expertos realizaron la segmentación volumétrica de la lesión hepática dominante en la TC en FVP; en aquellos casos donde los contornos de la lesión no fueron suficientemente claros para segmentar en dicha fase, se realizó la segmentación en otra de las fases disponibles. En cualquiera de los casos, solo se segmentó en una fase, ya que es un proceso que requiere mucha atención y tiempo de los especialistas.

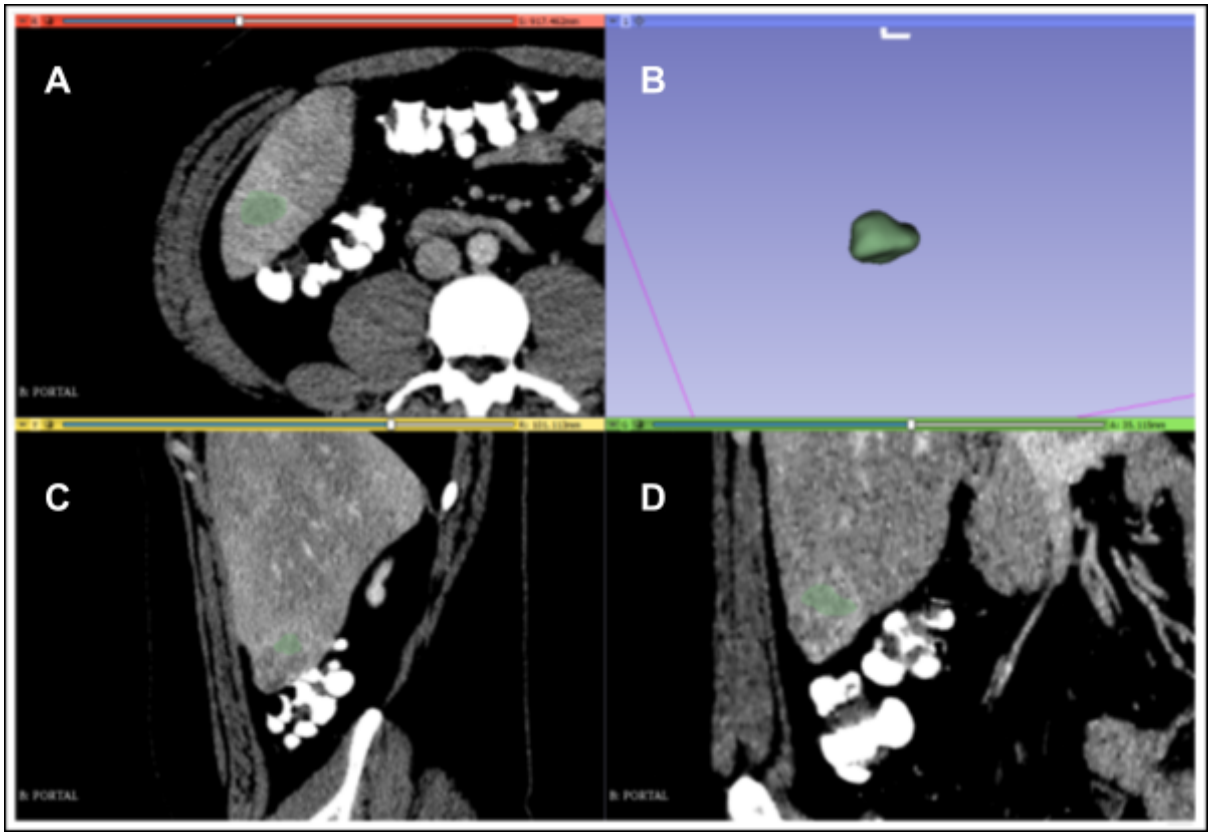

Figura 33. Ejemplo del resultado de segmentación con 3D Slicer. En la captura de pantalla del programa, se tienen vistas de los planos axial (A), sagital (C) y coronal (D) de una TC con contraste en FVP. El VOI (B) se visualiza en verde y su silueta aparece superpuesta sobre las otras vistas.

Como las imágenes en distintas fases se adquieren en diferentes momentos, la máscara segmentada en una fase podría no coincidir con el VOI esperado en las fases restantes. Para tener una máscara en todas las fases sin tener que segmentarlas individualmente, se recurrió a la librería Nibabel [57]. Este paquete da soporte para las operaciones con imágenes en el formato NIfTI, y en particular se puede usar para el registro de imágenes con la transformación afín. Para obtener la matriz afín que registra la máscara de, por ejemplo, una imagen en FVP a una imagen en FVT, se siguieron estos pasos, esquematizados en la **Figura 34**:

1. Se tomó la matriz afín del VOI en FVP (Afín FVP). Esta matriz permite hacer una transformación desde el espacio del VOI en FVP hacia el Espacio de Referencia.
2. Se invirtió la matriz afín del VOI en FVT (Afín FVT^{-1}). Esta matriz permite hacer una transformación desde el Espacio de Referencia al espacio del VOI en FVT.
3. Se hizo el producto punto entre la matriz Afín FVT^{-1} y la matriz Afín FVP. El resultado permite hacer una transformación desde el espacio del VOI en FVP al espacio del VOI en FVT.
4. Se usó la función *apply_affine* en la Máscara, que aplica la matriz de transformación calculada en el paso (3) para obtener la máscara registrada al VOI de la FVP.

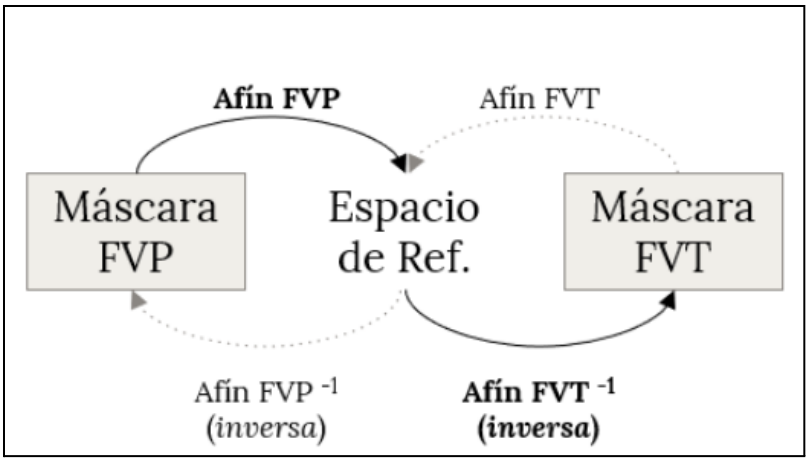

Figura 34. Esquema del proceso de registración de máscaras.

Luego de registrar las máscaras, se encontró que algunas de ellas mostraban discontinuidades que formaban un “entramado”. Para corregir este problema y conseguir máscaras continuas, se utilizó la operación morfológica de cierre, que primero dilata la imagen para rellenar los espacios discontinuos, y luego la erosiona para recortar los bordes ampliados generados por la dilatación. Un caso de ejemplo y su solución se muestra en la **Figura 35**. Para esto, se utilizó el paquete OpenCV [58] y su función *MORPH_CLOSE*.

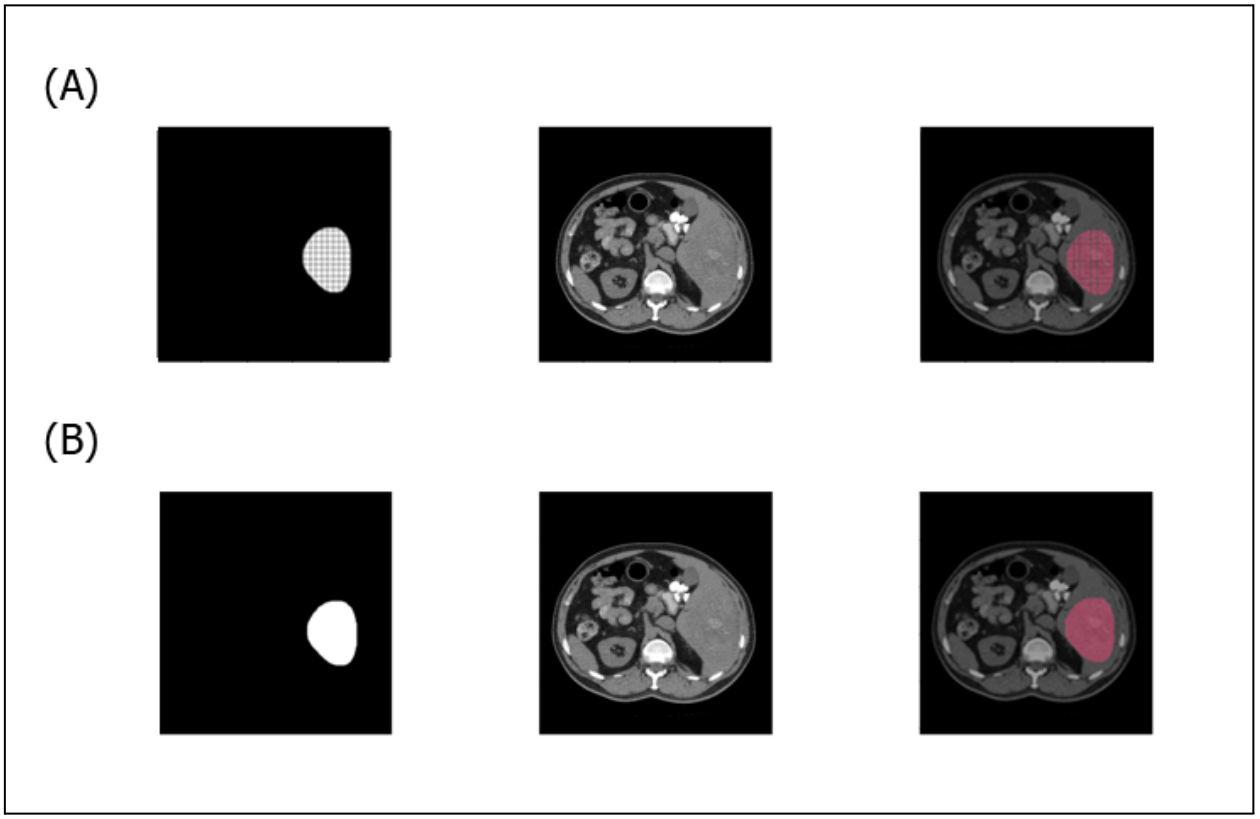

Figura 35. Ejemplo de una máscara con entramado. Izquierda: el gráfico de la máscara entramada. Las líneas negras corresponden a píxeles nulos, que erróneamente no están dentro de la máscara. Centro: el corte de TC al que corresponde la máscara. Derecha: la superposición entre ambas. En la fila (A) se muestra el ejemplo de la máscara con entramado, y en la fila (B) se muestra la máscara arreglada con la operación de cierre.

4.3 Extracción de características y armado de conjuntos de datos

La extracción de características radiómicas se realizó con el paquete PyRadiomics V 3.0.1 [15]. La librería permite configurar la extracción mediante tres tipos de factores: si se requiere algún preprocesamiento de las imágenes, por ejemplo, si se realizará normalización o interpolación (parámetro *setting*); si se aplicarán filtros a las imágenes, cuáles y la configuración de estos (parámetro *imageType*); y las características que se quieren extraer, donde se pueden elegir grupos, como también listar características individuales (parámetro *featureClass*). Estos parámetros se pueden guardar en un archivo de texto en formato YAML[5], que PyRadiomics puede leer e interpretar.

Los resultados de la extracción se compilan en una planilla y se guardan en un archivo separado por comas. En el mismo, cada fila corresponde a un VOI, y cada columna, al valor de una característica para ese VOI. Además, incluye columnas con valores descriptivos de la malla y la imagen (como sus tamaños), versiones de paquetes de Python implicados en la extracción, y otros datos que uno tenga y quiera incluir (como la etiqueta de clasificación de cada VOI). Nuestra extracción se configuró de la siguiente forma:

- *setting*: no se realizó ningún tipo de preprocesamiento ofrecido por PyRadiomics. Se utilizó un ancho de *bin* de 25 unidades para las características de primer orden. Se definió el valor de los píxeles de las máscaras igual a 1,0.
- *imageType*: se hizo la extracción de características en las imágenes originales (sin filtrar), y con filtros LdG (bajo tres configuraciones diferentes, con σ = 0,5, 1,5 y 2,5), Cuadrado, Raíz Cuadrada, Logarítmico y Exponencial.
- featureClass: se extrajeron todas las características de forma (en 2D y 3D), de primer orden, y de textura GLCM, GLRLM, GLSZM y GLDM.

En la **Figura 36** se muestra un ejemplo de archivo YAML que configura la extracción tal y como se describió anteriormente. Debido a que la extracción de características radiómicas es un proceso computacionalmente exigente y largo, se realizó por etapas (una para cada filtro), con un archivo para cada una de ellas. También se hizo por separado la extracción de características de estudios de distintas fases.

[5] YAML (*YAML Ain't Markup Language*), yaml.org

```yaml
setting:
  binWidth: 25
  label: 1.0
  interpolator: 'sitkBSpline'
  resampledPixelSpacing:
  weightingNorm:

imageType:
  Original: {}
  LoG:
    sigma: [0.5, 1.5, 2.5]
  Square: {}
  SquareRoot: {}
  Logarithm: {}
  Exponential: {}

featureClass:
  shape:
  firstorder:
  glcm:
  glrlm:
  glszm:
  gldm:

```

Figura 36. Ejemplo de un archivo de configuración de extracción en PyRadiomics. En *settings*, los factores declarados pero no configurados anulan los procesos de normalización y muestreo. En *imageType*, se usó la configuración por defecto de todos los filtros excepto el LdG. En *featureClass*, a menos que se especifique cada característica individualmente, se extraen todas de todos los grupos incluidos. Los filtros se escriben con su denominación en inglés (*LoG* para LdG, *Square* para Cuadrado, *Square Root* para Raíz Cuadrada, *Logarithm* para Logarítmico, y *Exponential* para Exponencial).

Las planillas obtenidas de la extracción de características radiómicas se utilizaron como conjuntos de datos para alimentar a los algoritmos de ML. Cada una de ellas contiene decenas de variables o cientos de variables, y se utilizaron de forma separada en los entrenamientos. Se utilizaron los siguientes conjuntos:

- **Original**: características extraídas sin utilizar filtros.
- **LdG:** características extraídas con las tres configuraciones de filtro LdG.
- **C+RC:** características extraídas con los filtros cuadrado y raíz cuadrada.
- **E+L:** características extraídas con los filtros exponencial y logarítmico.
- **Original+LdG:** combinación entre Original y LdG
- **Original+LdG+E+L:** combinación entre Original, LdG y E+L.

Debido a la cantidad de casos disponibles en la base de datos, y para evitar tener conjuntos con un número muy reducido de observaciones, se decidió utilizar sólo los conjuntos de entrenamiento y evaluación. La falta del conjunto de validación se aborda en la Sección 4.4.2. La separación de los datos en conjuntos de entrenamiento y evaluación se realizó teniendo en cuenta la cantidad de pacientes y las fases disponibles para cada uno, y la metodología fue diferente para la variables respuesta Óbito y KRAS. Para hacer la separación, se usó el paquete Scikit-Learn [59] y su función *train_test_split*, que separa una lista de datos en dos

conjuntos de forma aleatoria y de forma estratificada. La adición de estas restricciones hace que no siempre se pueda alcanzar la división con las características exactas deseadas, pero sí un buen aproximado.

En caso de Óbito, se decidió estratificar por la clase y también por la cantidad de fases disponibles para cada estudio. Para cada paciente, se contó si se tenía una, dos, tres o las cuatro fases disponibles. Se usó un tamaño del **75%** y **25%** para los conjuntos de entrenamiento y evaluación. Con la separación realizada a nivel paciente, se garantizó que si un paciente fue seleccionado para el conjunto de entrenamiento, los VOI en todas las fases disponibles también fueron seleccionados para ese mismo conjunto y no ocurrió que, por ejemplo, el VOI de un *paciente A* en FVP se colocó esté en el conjunto de entrenamiento, y el VOI del mismo *paciente A* en FA se colocó en el conjunto de evaluación.

En el caso de KRAS, se encontró un número bajo de estudios con el estado de mutación confirmado para las FA, FVT y FSC, por lo que solo se utilizaron las imágenes en FVP y no se avanzó en los objetivos secundarios asociados a esta variable. Por esta razón, se estratificó sólo usando el valor de la variable KRAS, y se buscó que un **65%** de los casos fueran de entrenamiento, con el **35%** restante de evaluación.

Todos los archivos de características y resultados que se guardaron como valores separados por coma fueron manipulados con el paquete Pandas [60], que ofrece herramientas para la lectura, modificación y guardado de este tipo de archivos. Los datos se cargan en una estructura llamada *dataframe*, a través del cual se pueden visualizar y manipular.

4.4 Selección de características y entrenamiento de modelos

El flujo de entrenamiento siguió tres pasos: la normalización de las características radiómicas, la selección de características, y el ajuste de hiperparámetros. Este flujo se realizó de la misma forma, independientemente de la variable de respuesta analizada o el conjunto de datos seleccionado, y fue implementado en todos los casos. El entrenamiento de una herramienta de IA es un proceso experimental, donde diferentes algoritmos, con diferentes hiperparámetros y diferentes estrategias de procesamiento, pueden tener resultados variados. La dificultad reside en balancear el uso de tiempo y recursos con la cantidad de experimentos a realizar, para lo cual es importante elegir con criterio qué combinaciones probar. A menos que se aclare lo contrario, los métodos de selección de características y los algoritmos fueron implementados mediante el paquete Scikit-Learn. Los experimentos realizados se diferenciaron en los siguientes puntos:

- La variable de respuesta, Óbito o KRAS.
- La fase de imagen usada, FVP, FA, FVT o FSC.
- El conjunto de datos usado, como Original o FdG.
- El método de selección de características, por ejemplo Selección hacia adelante.
- La cantidad de características a utilizar por el clasificador, 7, 10, 20 o 30.
- El algoritmo de ML ajustado, como Árbol de Decisión.
- El método de ajuste de hiperparámetros, GridSearchCV o RandomizedSearchCV.

Todos los experimentos se identificaron por la fecha de realización y un índice incremental, indicando el número de experimentos realizados. Se documentaron todas las variables configurables anteriormente listadas, como también el resultado de cada etapa del proceso: las características seleccionadas, los hiperparámetros óptimos, y las métricas de desempeño de la clasificación.

4.4.1 Selección de Características

La normalización de los datos se considera un paso importante al utilizar características cuyos valores difieren en órdenes de magnitud para agilizar los cálculos realizados por los algoritmos de IA. Para hacerlo, se usó el método *StandardScaler* de Scikit-Learn, que obtiene la media y el desvío estándar de cada característica de un conjunto de datos, y genera la transformación que las estandariza:

$$z = \frac{x - \mu}{\sigma} \tag{89}$$

Para cada conjunto de datos usado, la transformación se generó en el conjunto de entrenamiento, y se aplicó tanto a este como al conjunto de evaluación. Luego, se utilizó el conjunto de entrenamiento para hacer la selección de características. Los métodos utilizados se detallan a continuación.

4.4.1.1 Correlación de Pearson

La **Correlación de Pearson** es una medida estadística de la relación lineal entre dos variables. En el contexto de la selección de características, es un método de filtrado que se utiliza para medir la relación entre una característica y la variable de respuesta del modelo. El coeficiente se calcula de la siguiente forma:

$$P = \frac{cov(x, y)}{\sigma_x \sigma_y} \tag{90}$$

Donde *x* es el valor de una característica para todas las muestras, *y* es el valor de la variable de respuesta , *σ* es la media de la respectiva variable, y *cov(x, y)* es la covarianza entre *x* e *y* [33].

El coeficiente de correlación de Pearson oscila entre -1 y 1, donde -1 indica una relación inversa perfecta, 0 indica que no hay relación lineal y 1 indica una relación directa perfecta. Por lo tanto, la correlación de Pearson se puede utilizar para medir la fuerza y la dirección de la relación lineal entre dos variables.

En la selección de características, las características con una alta correlación (tanto positiva como negativa) con la variable de respuesta se consideran importantes y se seleccionan para su inclusión en el modelo. Esto se diferencia de la selección de características por alta correlación explicado en la Sección 2.4.4.2, donde las características con alta correlación entre sí se eliminan, en vez de ser seleccionadas, debido a su redundancia.

La correlación de Pearson sólo mide la relación lineal entre dos variables. Si hay una relación

no lineal entre una característica y la variable de respuesta, entonces la correlación de Pearson puede no ser efectiva para seleccionar esa característica. En tales casos, se pueden utilizar otros métodos de selección de características, como la correlación de rango de Spearman o la ganancia de información mutua.

Fue implementado utilizando el método *corr* del paquete Pandas (calcula la correlación para todas las columnas del *dataframe* con una variable), y se conservaron todas aquellas cuya correlación con la variable de respuesta fuera mayor a 0,5.

4.4.1.2 LASSO

LASSO (*Least Absolute Shrinkage and Selection Operator*) es un método de selección de características embebido utilizado en ML y estadística. Este método ajusta un modelo de regresión lineal con un término de regularización, penalizando a la función de costo en función del valor de los parámetros. La fuerza de la penalización está controlada por un hiperparámetro de regularización (generalmente llamado lambda), que determina el grado de reducción en el tamaño de los coeficientes [25]. La función de costo a minimizar es la siguiente:

$$\sum_{i=1}^{n} (y_i - \sum_{j} x_{i,j}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{91}$$

Donde y_i son los valores de la salida, $x_{i,j}$ los valores de cada característica para cada entrada, β_j son los coeficientes de cada característica, y λ es la fuerza de regularización. La segunda sumatoria es el término de regularización, conocido como norma L1, una medida de la magnitud de un vector en términos de la suma de los valores absolutos de sus componentes.

El resultado de la aplicación del método es un conjunto de coeficientes que se acercan a cero, lo que indica que las características correspondientes no son importantes para la predicción y, por lo tanto, pueden ser eliminadas del modelo. Las características seleccionadas pueden luego ser utilizadas con otro algoritmo.

Se implementó con el método *SelectFromModel*, que modifica un algoritmo de ML para registrar el número deseado de las características más relevantes. Las características seleccionadas fueron aquellas cuya importancia fuera mayor o igual a la media de todas las importancias. Se utilizó una regresión con λ = 1,0 y un máximo de 5.000 iteraciones.

4.4.1.3 Extra Trees Classifier

Extra Trees Classifier (**ETC**) es un método de selección de características embebido basado en los árboles de decisión, y utiliza múltiples de ellos para evaluar la importancia de cada característica en el conjunto de datos. Su nombre es la abreviación de *Extremely Randomized Trees*, y es similar en su funcionamiento al algoritmo de bosques aleatorios, con la adición de algunos componentes aleatorios [25]. El algoritmo de árbol de decisión será explicado en la Sección 4.4.2.3, y el bosque aleatorio será explicado en la Sección 4.4.2.4.

Consiste en crear varios árboles de decisión con diferentes subconjuntos de datos de

entrenamiento y características, ambos seleccionados de forma aleatoria. Los datos de entrenamiento son tomados sin reposición, a diferencia del algoritmo de bosques aleatorios, y de esta forma cada árbol tiene un conjunto de datos único.

Cada árbol se entrena y se utiliza para evaluar la importancia de cada característica. Luego, se calcula la importancia de cada característica como el promedio de la importancia evaluada en cada árbol. Finalmente, se seleccionan las características más importantes en función de un umbral predefinido.

Debido a que el método utiliza varios árboles de decisión, es más resistente al sobreajuste que otros métodos de selección de características basados en un solo modelo.

Se implementó con la función *ExtraTreesClassifier*, con los siguientes hiperparámetros: *n_estimators*=300, *criterion*="entropy", *max_features*='sqrt', *class_weight*='balanced', *warm_start*=True.

4.4.1.4 Selección hacia adelante

El método de selección hacia adelante (*Forward Selection*, **FS**) es uno de los métodos de envoltura. Para un algoritmo elegido, se comienza con un modelo que no incluye características, y se agrega una por una en un proceso iterativo. En cada iteración, se selecciona la característica que mejora el rendimiento del modelo, según un criterio de evaluación específico, y se la agrega a la próxima iteración. El proceso se repite hasta que se alcanza un punto de detención predefinido, como un número máximo de características o una mejora de rendimiento insuficiente [25]. La **Figura 37** ejemplifica el método secuencial.

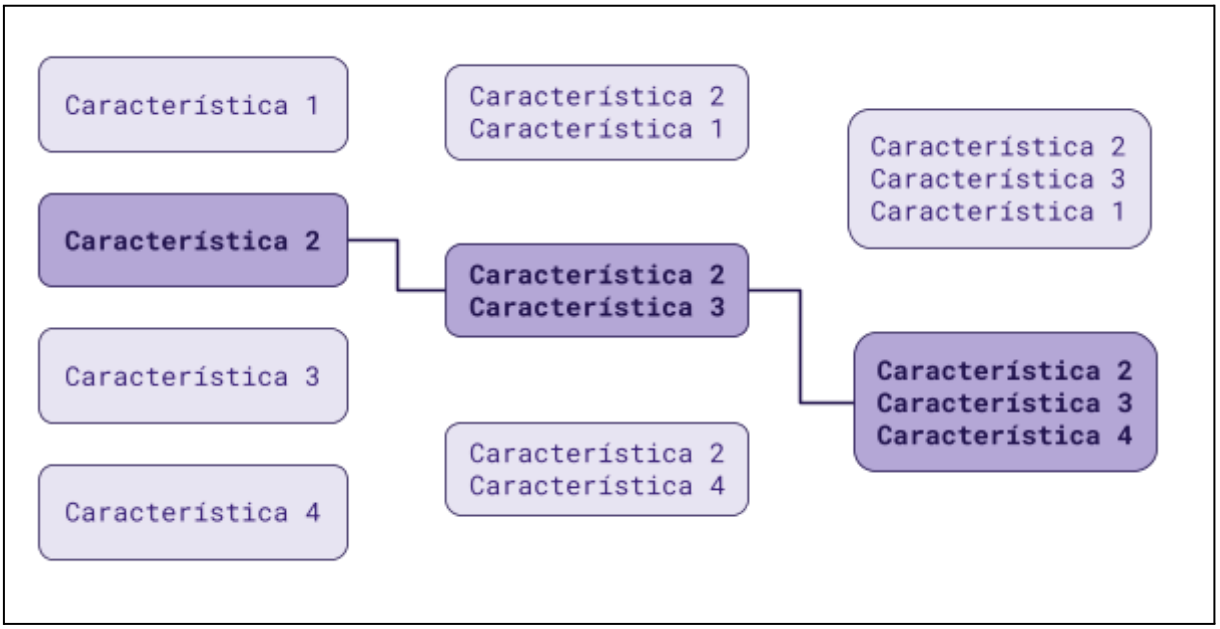

Figura 37. Diagrama del método FS con cuatro posibles características y un máximo de tres a seleccionar. En la primera columna, el modelo se entrena con solo una característica a la vez, y obtiene el mejor resultado con la característica dos. En la segunda columna, entrena con los tres posibles pares, seleccionando el par dos y tres. En la tercera columna, entrena con las dos posibles combinaciones de tres valores, siendo la combinación de dos, tres y cuatro la de mejor resultado.

Este método tiene varias ventajas: es relativamente simple y fácil de implementar, y se puede utilizar con una variedad de criterios de evaluación, como la precisión, la sensibilidad, la especificidad o el error cuadrático medio, según la aplicación específica.

Una de sus limitaciones es que puede ser computacionalmente costoso para grandes conjuntos de datos. Además, si las características son altamente correlacionadas o redundantes, el método puede seleccionar características similares y no necesariamente las más informativas.

Se implementó con la función *SFS* (*Sequential Feature Selection*) del paquete Mlxtend [62]. La misma requiere un algoritmo de ML (seleccionado según el experimento), la cantidad de características deseadas, y la métrica con la cual decidir el desempeño de cada clasificador probado, para lo que se usó el valor F1 (una medida de exactitud compuesta, detallada en la Sección 4.6.2).

4.4.1.5 Eliminación hacia atrás

El algoritmo de eliminación hacia atrás (*Backward Elimination*, **BE**) es otro método de selección de características de envoltura. Implica entrenar un algoritmo de elección con todas las características, y eliminar iterativamente aquellas menos importantes [25].

El proceso comienza con el entrenamiento de un modelo utilizando todas las características disponibles y evaluando su rendimiento. A continuación, se selecciona la característica menos importante y se elimina del conjunto. Luego, se vuelve a entrenar el modelo con las características restantes y se evalúa su rendimiento nuevamente. Si la métrica de desempeño del modelo no disminuye significativamente después de eliminar una característica, esta se considera menos importante y se elimina del conjunto. Este proceso se repite iterativamente hasta que se alcanza una precisión satisfactoria o no quedan características para eliminar, como se muestra en la **Figura 38**.

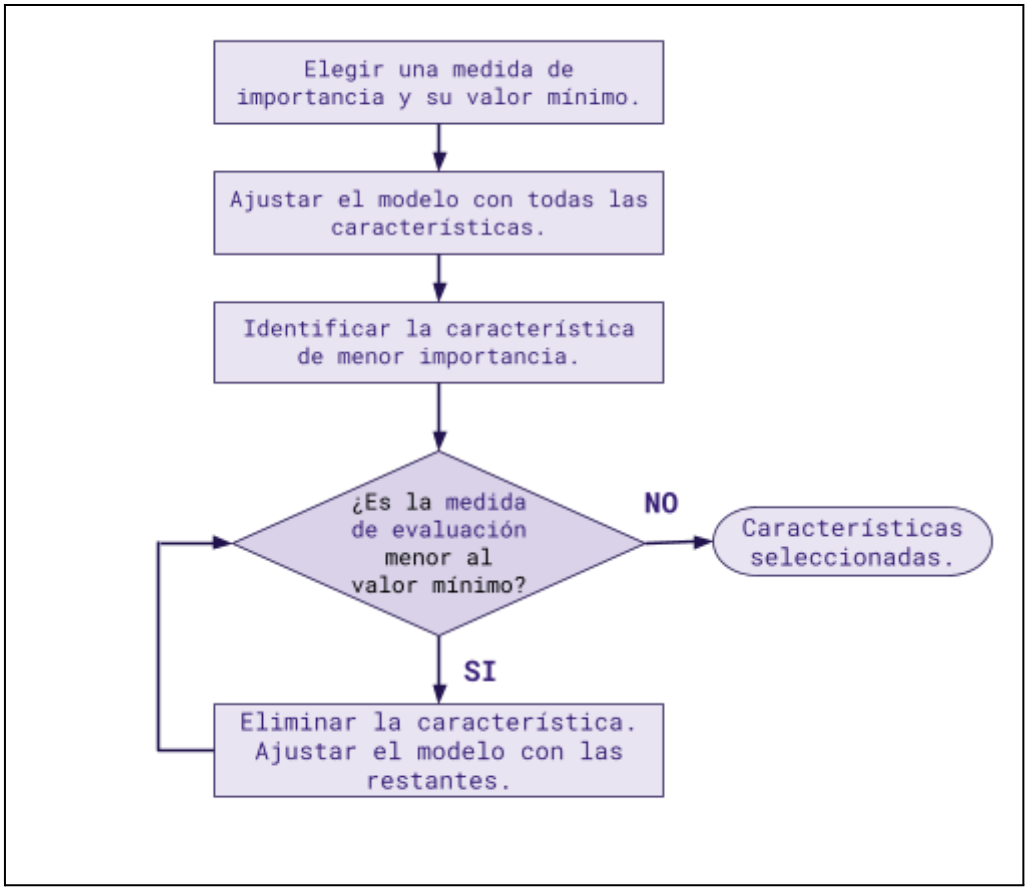

Figura 38. Diagrama de flujo del método de eliminación hacia atrás.

Este método tiene algunas ventajas sobre otros enfoques de selección de características, ya que puede manejar conjuntos de datos grandes y reducir el riesgo de sobreajuste. Además, es relativamente fácil de implementar y se puede utilizar con una variedad de algoritmos de aprendizaje automático. Sin embargo, una desventaja potencial de este método es que puede ser computacionalmente costoso si hay muchas características en el conjunto original. Además, es posible que algunas características importantes se eliminen demasiado pronto, lo que puede afectar el desempeño del modelo final.

Se implementó con la función *RFE* (*Recursive Feature Elimination*), que requiere un algoritmo de ML (seleccionado según el experimento) y la cantidad de características deseadas.

4.4.1.6 Análisis de Componentes Principales

En el contexto del ML, el método de Análisis de Componentes Principales (*Principal Component Analysis*, **PCA**) es un enfoque comúnmente utilizado para la selección de características en conjuntos de datos de alta dimensionalidad. El PCA es en rigor una técnica de reducción de dimensionalidad que busca reducir el número de características de un conjunto de datos, mientras se conserva la mayor cantidad posible de información [25].

El PCA se basa en la transformación lineal de los datos originales en un nuevo espacio de características, que se define mediante la combinación lineal de las características originales. La transformación se realiza de tal manera que la primera componente principal tiene la mayor

varianza posible, la segunda componente principal tiene la segunda mayor varianza, y así sucesivamente. La cantidad de componentes generadas depende del criterio usado en el algoritmo, pero cumple con dos condiciones. Una es que, como método de reducción de dimensionalidad, es menor a la cantidad original. La otra es que la sumatoria de la varianza explicada por cada componente resulta en la varianza total. Cada componente principal es una combinación lineal de las características originales, y se puede interpretar como una dirección en el espacio de características.

Una vez que se han generado todas las componentes principales, se las puede utilizar como las nuevas características. Se puede preestablecer que el algoritmo genere tantas componentes como características se quiere, o no establecer un límite y luego elegir tantas como se desee.

La **Figura 39** muestra un ejemplo de cómo se reduce la dimensionalidad de los datos con PCA; en la misma se visualizan los datos expresados a través de la primera y segunda componente principal. En el ejemplo se utiliza la base de datos *Iris* [61], que incluye las especies de flores *Iris setosa, Iris virginica* y *Iris versicolor*. Cada flor se describió con cuatro características: largo de sépalo (*sepal length*), ancho de sépalo (*sepal width)*, largo de pétalo (*petal length*) y ancho de pétalo (*petal width*). Esta base de datos se usará en otros ejemplos del trabajo.

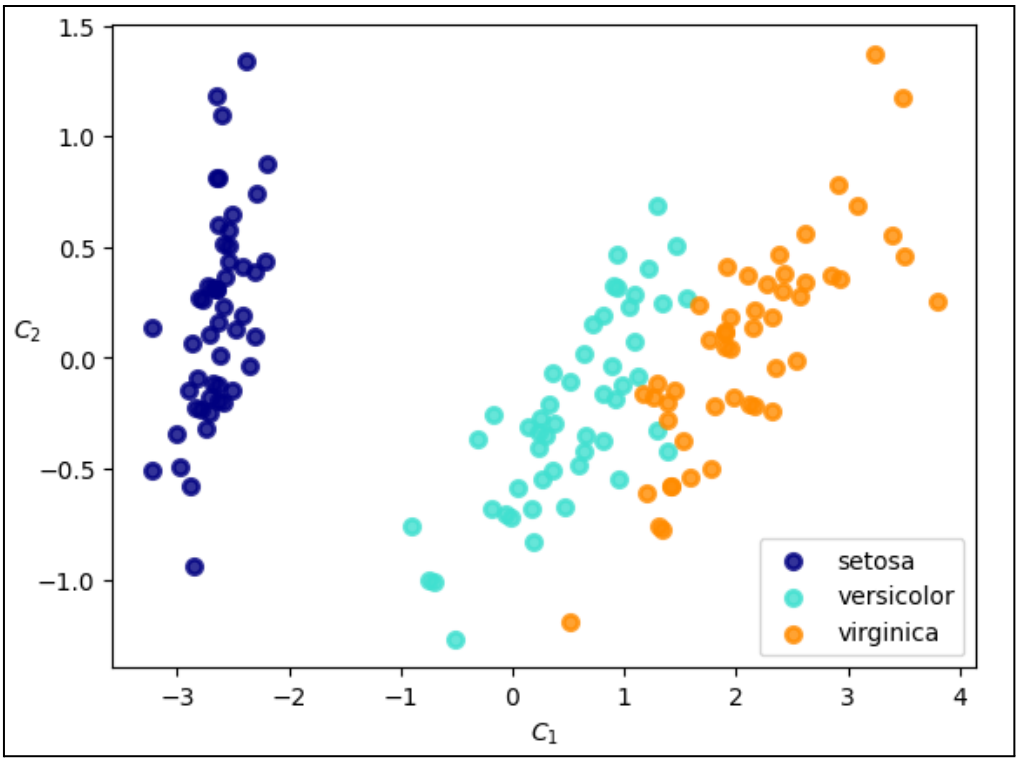

Figura 39. Ejemplo de reducción de dimensionalidad con el algoritmo PCA utilizando la base de datos *Iris*. A partir de los datos y las cuatro características originales, se generaron las componentes C_1 y C_2. Al graficar los datos en el espacio de estas características, se puede ver que en general, datos de una misma especie quedan agrupados.

La selección de características basada en PCA puede ser útil en situaciones en las que el conjunto de datos contiene características altamente correlacionadas o redundantes. En estos casos, este método puede ayudar a identificar las características que están altamente

correlacionadas y seleccionar solo las más informativas. Sin embargo, es importante tener en cuenta que la interpretación de las características seleccionadas puede ser difícil, ya que se expresan en términos de las componentes principales, que son combinaciones lineales de las características originales. Además, puede no ser adecuada para conjuntos de datos que contienen características no lineales o interacciones complejas entre características.

Se implementó con la función *PCA*, ajustando un clasificador PCA del cual luego se extraen las características de mayor importancia. Se configuró con la cantidad de características deseadas, *tol* = 0.95, y *whiten* = True.

4.4.1.7 LinearSVC

LinearSVC es una método de selección de características embebido que se realiza mediante la minimización de una función de costo en función del número de características seleccionadas y la importancia de cada una de ellas.

El algoritmo comienza por entrenar un modelo de Máquina de Vectores de Soporte (explicado en la Sección 4.4.2.5) lineal en los datos de entrenamiento y luego utiliza los coeficientes de los vectores de soporte obtenidos en el proceso para determinar la importancia de cada característica. Las características con coeficientes más grandes se consideran más importantes para la clasificación, mientras que las características con coeficientes más pequeños se consideran menos importantes.

La selección de características utilizando LinearSVC puede ser una técnica eficaz para reducir el número de características en conjuntos de datos de gran tamaño, lo que a su vez puede mejorar la precisión de los modelos de aprendizaje automático. Sin embargo, es importante tener en cuenta que esta técnica de selección de características sólo es adecuada para problemas de clasificación binaria y requiere la definición de un hiperparámetro de regularización C que controla el balance entre la precisión del modelo y el número de características seleccionadas, como fue mencionado en el concepto de función de costo en la Sección 2.4.4.3. Se implementó usando la función *LinearSVC*, con *penalty* = l2, *C* = 1.0 y *class_weight* = balanced.

4.4.2 Entrenamiento de modelos

Una vez finalizada la selección de características por los métodos mencionados anteriormente, se procedió al entrenamiento de algoritmos y el ajuste de hiperparámetros. El ajuste se llevó a cabo con dos métodos implementados por el paquete Scikit-Learn, cuya función es dar una herramienta para hacer entrenamientos iterativos sobre un mismo clasificador, pero con diferentes hiperparámetros en cada caso, eligiendo los hiperparámetros generen el mejor clasificador en los términos de una métrica definida por el usuario. Los dos métodos y sus diferencias son las siguientes:

- *GridSearchCV* requiere al usuario explicitar todos los hiperparámetros que se quieren probar, y todos los valores para los mismos que deben ser probados. Por ejemplo, si se tiene un algoritmo que utiliza una función de costo con regularización, se puede incluir el

hiperparámetro de regularización C con los valores 1 y 10. Entonces, se entrenan dos clasificadores de ese algoritmo, probando ambos valores de C.

- *RandomizedSearchCV* también requiere que se definan los hiperparámetros que se quieren probar, pero sus valores se eligen aleatoriamente de un rango definido por el usuario, probando tantas veces como se configure la variable *n_iter*. Siguiendo el ejemplo anterior, si se define para C el rango (1, 10) y se piden cinco iteraciones, entonces se realizan cinco entrenamientos, y en cada uno, el hiperparámetro toma aleatoriamente un valor entre 1 hasta 10.

Ya que se decidió no separar casos para un conjunto de validación (debido al bajo número disponible), se realizó una validación cruzada en cada uno de los entrenamientos. Tanto para GridSearchCV como RandomizedSearchCV, el ajuste se hizo con validación cruzada de 10 pliegues, priorizando obtener el mejor valor de AUC ROC, y secundariamente el mejor valor de AUC PR. Todos los algoritmos de clasificación fueron implementados con la librería Scikit-Learn, con la excepción del algoritmo XGBoost, implementado con el paquete de su mismo nombre [63]. En estas implementaciones, funcionan como algoritmos de aprendizaje supervisado para clasificación binaria o multiclase, pero algunos de ellos pueden ser usados en otro tipo de aprendizaje o tarea con una implementación diferente. A continuación se describen estos algoritmos, mientras que los hiperparámetros probados y sus valores están registrados en el Anexo B.

4.4.2.1 K Vecinos Cercanos

El clasificador de k-Vecinos Cercanos (*k-Nearest Neighbours*, **KNN**) es un algoritmo basado en el concepto de vecindad para clasificar patrones en datos. Es un método no paramétrico, lo que significa que no asume ningún tipo de distribución a priori sobre los datos [25]. Esto lo hace especialmente útil para trabajar con conjuntos de datos desconocidos o de alta dimensionalidad de manera efectiva.

El algoritmo KNN no genera un modelo propiamente dicho, sino que guarda instancias de los datos de entrenamiento y realiza comparaciones entre los datos de nuevos casos a clasificar y los datos almacenados. Las características definen un espacio de características en el cual se ubican las instancias y en el cual es posible calcular la distancia entre datos. Para clasificar un nuevo dato, se lo dispone en el espacio de características y se consideran los valores de un número *k* de otros elementos en el espacio, denominados vecinos. En la forma más simple del algoritmo, la clase elegida para el nuevo dato será la clase más frecuente entre los *k* vecinos más cercanos. La clasificación por KNN y el efecto de la cantidad de vecinos se muestra en la **Figura 40**.

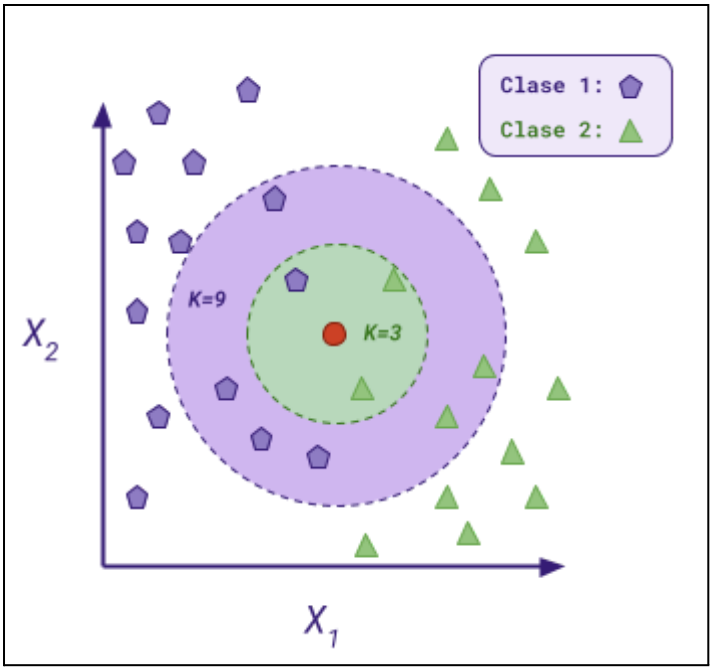

Figura 40. Gráfico de un clasificador KNN binario. Los datos están en un espacio de dos características, X_1 y X_2, y se separan en la clase 1 (pentágonos violetas) y clase 2 (triángulos verdes). Se quiere clasificar un nuevo elemento en el centro (círculo rojo): si se consideran los 3 vecinos más cercanos, la clase más frecuente es la 2, pero si se consideran 9, la clase más frecuente es la 1.

Uno de los principales desafíos del clasificador KNN es la elección adecuada del valor de *k* (n_neighbors). Un valor de “k” demasiado pequeño puede causar sobreajuste en los datos, mientras que un valor demasiado grande puede causar subajuste. Por lo tanto, es importante seleccionar un valor adecuado mediante técnicas como la validación cruzada.

La importancia de los vecinos en general se define con parámetros (weights). Por un lado, *uniform* considera a todos los vecinos por igual, sin importar cuáles de los seleccionados están más cerca (en caso de un empate, se elige la clase a la cual pertenece el dato de entrenamiento que primero se introdujo al algoritmo). Por otro lado, *distance* pondera el voto de cada vecino de acuerdo a la distancia con la nueva muestra. En caso de usar *distance*, metric permite elegir la forma de calcular la distancia, con opciones como *euclidean*, *manhattan*, *chebyshev*, y *minkowski*.

La forma en la cual se resuelve el cálculo de distancias se determina con algorithm, con tres opciones posibles: *brute*, *kd_tree* y *ball_tree*. Por un lado, la opción *brute* calcula las distancias a todos los vecinos, con lo cual es lento con conjuntos grandes. Por otro lado, las opciones *kd_tree* y *ball_tree* construyen mapas interconectados entre los puntos, conociendo la distancia relativa entre ellos: si se tienen tres datos, A, B (ya conocidos por el clasificador), y C (uno nuevo). Se sabe que los datos A y B son cercanos, y al calcular la distancia entre A y C, se encuentra que es grande. Entonces, se presume que la distancia entre C y B será grande, y no se calcula. Estos algoritmos son más rápidos en la inferencia con conjuntos grandes, pero son costosos de entrenar.

4.4.2.2 Regresión Logística

La Regresión Logística (*Logistic Regression*, **LR**) es un algoritmo de aprendizaje supervisado que se utiliza para predecir la probabilidad de que un evento binario ocurra, es decir, si algo es verdadero o falso, sí o no, 0 o 1. Un ejemplo concreto puede ser el hecho de que un paciente tenga o no tenga una enfermedad, o si un medicamento funcionará o no para tratar una afección médica específica.

La Regresión Logística se basa en un modelo matemático que utiliza la técnica de máxima verosimilitud para estimar los parámetros que definen la función logística [25]. Estos parámetros se ajustan iterativamente para minimizar la discrepancia entre las predicciones del modelo y las etiquetas de clase observadas en los datos de entrenamiento. Cada parámetro está asociado a una característica, que son variables independientes, mientras que la clase de la muestra es la variable dependiente.

Su implementación es un modelo lineal, contrario a lo indicado por el nombre. La función logística, una función sigmoidea, toma el resultado del modelo lineal para ajustar su valor a un rango entre cero y uno, que representa la probabilidad de que ocurra el evento buscado. La función logística se define como se muestra en la **Ecuación 92** y la **Figura 41**.

$$P(x) = \frac{1}{1 + e^{-x}} \tag{92}$$

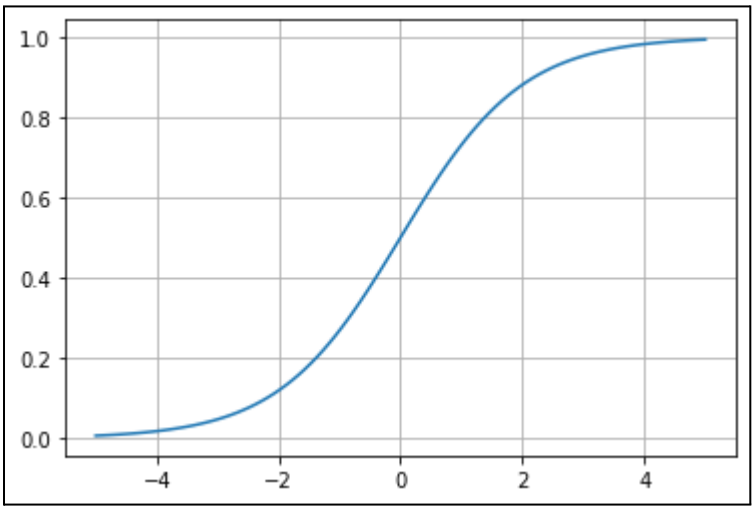

Figura 41. Gráfico de la función logística, acotada entre -5 y 5.

El algoritmo utiliza una función de costo para optimizar el modelado de la función logística, con un factor de regularización. El hiperparámetro penalty determina la función de regularización (como *L1*, *L2*), C determina la fuerza de regularización, y solver el algoritmo de optimización; si fit_intercept es verdadero, el modelo lineal tendrá un término independiente.

La importancia de cada clase se puede incluir en el entrenamiento con class_weight, que multiplica al factor C por un valor predeterminado para cada clase. Esto permite penalizar con

mayor fuerza a una clase por sobre otras: puede no usarse (*None*), explicitar los valores para cada clase, o calcular el factor en base a la proporción de cada clase en el conjunto de datos (*balanced*).

El algoritmo concluye cuando se haya realizado la cantidad de iteraciones establecida en max_iter, o cuando la función de costo tenga un valor menor al valor de tol. Finalmente, warm_start permite usar la solución de una iteración previa para inicializar la siguiente. Este hiperparámetro sólo se puede utilizar en algoritmos que iteran para variar parámetros internos, o en cualquier algoritmo durante el ajuste de hiperparámetros.

4.4.2.3 Árbol de Decisión

Los Árboles de Decisión (*Decision Tree*, DT) pueden referirse a un diagrama de flujo para la toma de decisiones o un motor de reglas, o el algoritmo que los genera. Los DT son algoritmos versátiles que pueden realizar tanto tareas de clasificación como de regresión, y son visualmente interpretables, ya que se puede graficar el diagrama de flujo que representan. Los DT son también una parte fundamental de otros algoritmos, como los Bosques Aleatorios (detallados en la Sección 4.4.2.4) y otros algoritmos de ensamble [13].

En su forma más simple, un DT es un diagrama en cuya raíz está el conjunto de datos completo. Cada nodo representa una característica o atributo, cada rama representa una decisión basada en ese atributo, y cada hoja representa una etiqueta de clase o un valor de regresión [25]. El proceso de construir un árbol de decisión implica dividir repetidamente los datos en subconjuntos basados en la mejor característica para clasificar o predecir la variable objetivo.

El entrenamiento de un DT implica encontrar la mejor manera de dividir los datos en cada nodo para minimizar la impureza de la etiqueta de clase (o la varianza, en el caso de la regresión). La impureza es una medida de la homogeneidad del nodo respecto a las muestras que hay en el mismo. Si todas las muestras en un nodo tienen la misma etiqueta, entonces el nodo es puro y se considera una hoja. Existen métricas de impureza, como el índice de Gini [25], que permiten evaluar la calidad de la separación lograda en cada nodo.

La visualización puede ser muy útil para comprender cómo se está haciendo la clasificación. En la **Figura 42** se muestra como ejemplo el diagrama de flujo de un DT construido sobre la base de datos Iris.

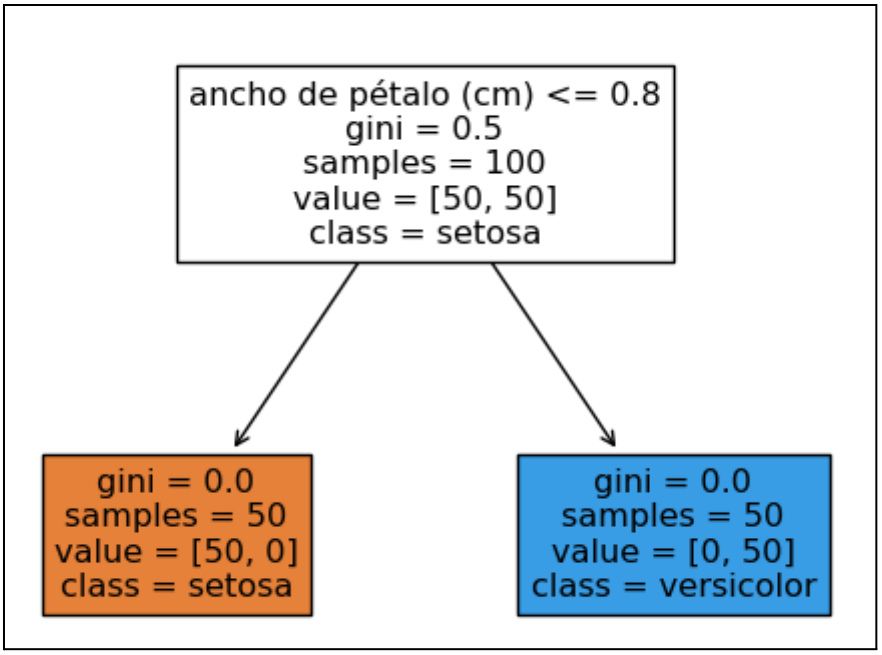

Figura 42. Ejemplo de DT binario utilizando la base de datos Iris con muestras de solo dos de sus clases (setosa y versicolor). En el nodo raíz (profundidad cero), se pregunta si el ancho del pétalo es menor o igual a 0,8 cm. Si lo es, se baja al nodo hijo izquierdo (profundidad 1), que es un nodo puro, una hoja, de la clase setosa. Caso contrario, se baja al nodo hijo derecho (profundidad 1), que también es una hoja, de la clase versicolor.

También se puede tener una visualización en cuanto a los límites de decisión de dicho árbol en el conjunto de datos, como se muestra en la **Figura 43.**

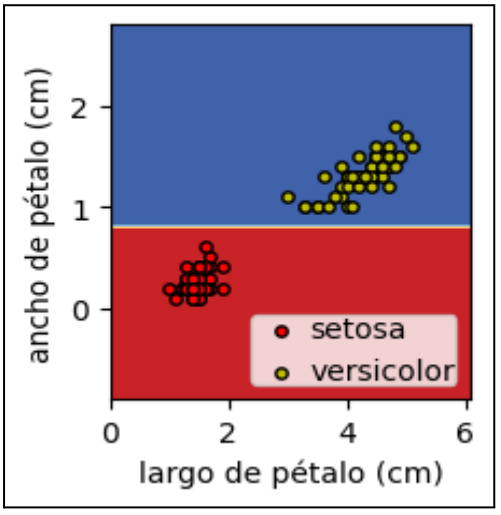

Figura 43. Límites de decisión del DT binario utilizando la base de datos *Iris* con muestras de solo dos de sus clases (setosa y versicolor) y dos características, ancho de pétalo y largo de pétalo. Se observa una línea horizontal continua, la cual representa el límite de decisión del nodo raíz (profundidad cero), en 0,8 cm de ancho de pétalo. Como los nodos en profundidad uno son hojas, ya no puede dividirse más.

Los DT suelen denominarse como “modelo no paramétrico”, porque su número de parámetros no está determinado antes del entrenamiento, así que la estructura del modelo es libre para ajustarse a los datos, con tendencia al sobreajuste. Por el contrario, un “modelo paramétrico”, como un modelo lineal, tiene un número de parámetros predeterminado y su grado de libertad

está limitado, reduciendo el riesgo de sobreajuste (pero aumentando el de subajuste).

Esto puede controlarse con min_samples_split (mínimo de muestras necesarias para una división) y min_samples_leaf (mínimo de muestras necesario para una hoja). Sus valores por defecto, 2 y 1 respectivamente, tienden a crear árboles sobreajustados, y valores más altos llevan a un DT más robusto. La cantidad máxima de características evaluadas al considerar una separación (max_features) y la profundidad máxima de un árbol (max_depth) también permiten controlar el ajuste del DT. Finalmente, la importancia de cada clase se puede incluir en la construcción del árbol con class_weight, realizando una ponderación como en el caso del algoritmo LR.

Una de las cualidades de los árboles de decisión es que requieren poca preparación de los datos, sin necesitar escalado de características ni centrado. Por otro lado, su principal problema es la sensibilidad a pequeñas variaciones en los datos de entrenamiento.

4.4.2.4 Bosques Aleatorios

Los Bosques Aleatorios (*Random Forest,* RF) son un algoritmo de aprendizaje supervisado que se utiliza para la clasificación y la regresión en problemas de alta dimensionalidad. Se basa en la construcción de múltiples árboles de decisión en paralelo. Cada árbol de decisión se entrena con una muestra aleatoria de los datos de entrenamiento tomada con reposición y utiliza un subconjunto aleatorio de las características de entrada para hacer el ajuste. La combinación de los resultados de todos los árboles de decisión resulta en una predicción final [25]. El uso de múltiples modelos para arribar a un único resultado se conoce como ensamble. En particular, si se consideran todos los resultados de modelos entrenados en paralelo, como puede ser promediando sus puntajes de salida, o tomando la moda de sus clasificaciones, se habla de una técnica de *bagging* [13].

La varianza y el sesgo del modelo se reducen de dos formas. La varianza se reduce al construir múltiples árboles de decisión en paralelo, mientras que el sesgo se reduce al limitar la profundidad de los árboles de decisión y al seleccionar aleatoriamente un subconjunto de características de entrada para cada árbol de decisión. La **Figura 44** muestra un esquema del algoritmo RF.

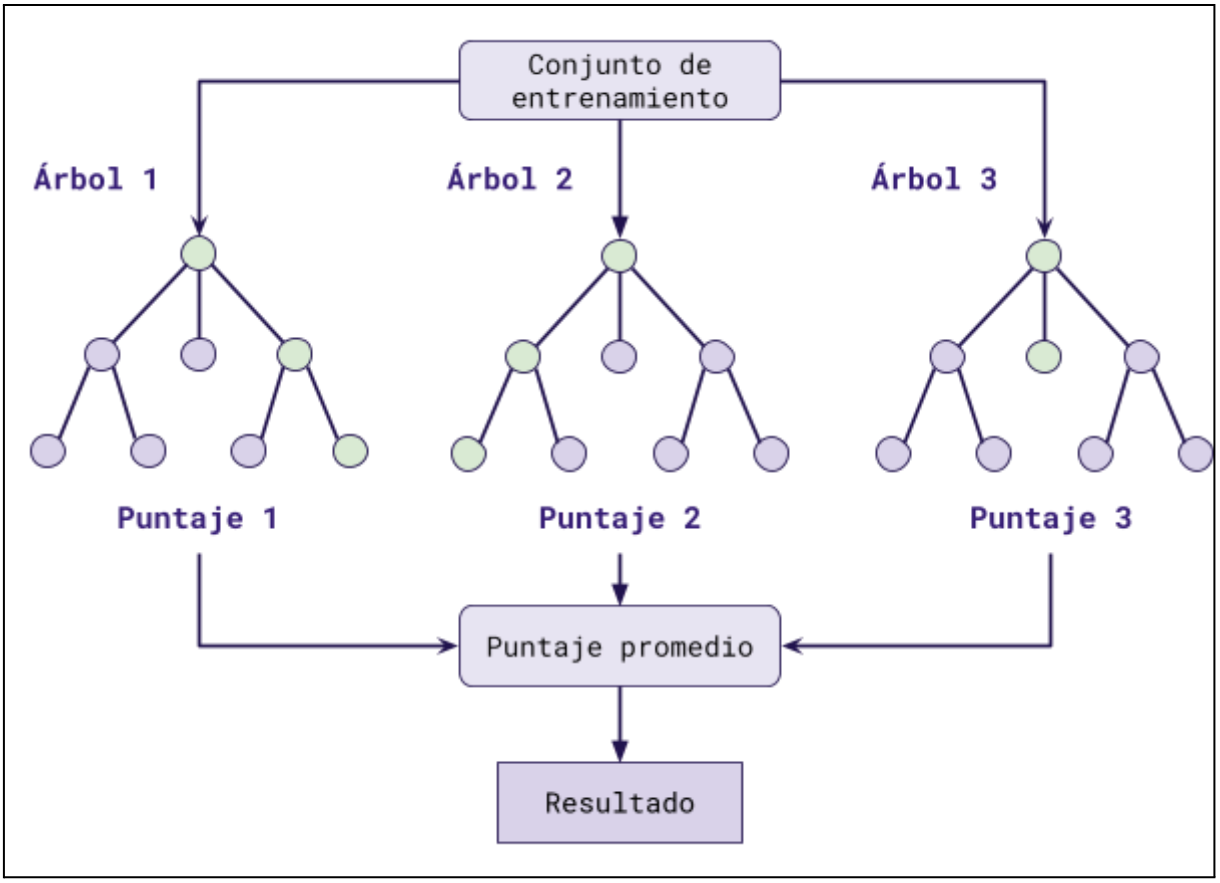

Figura 44. Diagrama de un RF con tres árboles. Cada árbol toma una muestra con reposición del conjunto de entrenamiento, y una muestra aleatoria de las características. Para una nueva muestra del conjunto de datos, cada árbol la evalúa usando las características asociadas a él (nodos verdes) y responde con un puntaje de clasificación. Los puntajes se promedian y el resultado se usa para clasificar.

Una de las ventajas de RF es que puede manejar datos faltantes y variables categóricas sin la necesidad de imputar valores o convertir variables categóricas en numéricas. Además, es robusto frente al ruido y a las variables irrelevantes, y puede manejar relaciones no lineales entre las características de entrada y la variable de salida.

Sin embargo, también hay algunas limitaciones. Por ejemplo, puede ser computacionalmente costoso y requerir mucho tiempo para entrenar y ajustar. Además, puede ser difícil de interpretar debido a la complejidad del modelo y la dificultad de determinar el criterio de cada árbol en el bosque.

RF comparte muchos hiperparámetros con DT (criterion, min_samples_split, min_samples_leaf, max_features, min_impurity_decrease, class_weight). Otros hiperparámetros configurables son la cantidad de árboles en el bosque (n_estimators) y el uso de warm_start, cuyo funcionamiento se explicó en LR.

4.4.2.5 Máquina de Vectores de Soporte

El algoritmo de Máquina de Vectores de Soporte (*Support Vector Machine*, **SVM**) se basa en encontrar un hiperplano que separe los datos de entrenamiento en regiones distintas, lo más

separados posible. En el caso de la clasificación binaria, el hiperplano separa las dos clases de puntos en el espacio de características. En el caso de la regresión, el hiperplano se ajusta a los datos de entrenamiento y se utiliza para hacer predicciones [25].

En la clasificación binaria, el objetivo del SVM es encontrar el hiperplano que maximiza la distancia entre las dos clases, conocida como margen. Los puntos de entrenamiento más cercanos al hiperplano se denominan vectores de soporte, y la distancia hasta el hiperplano determina el margen funcional. Este margen se utiliza para determinar la separabilidad de las clases, y el algoritmo busca maximizarlo para lograr una clasificación más precisa y generalizable. La **Figura 45** muestra un ejemplo de separación por el algoritmo SVM.

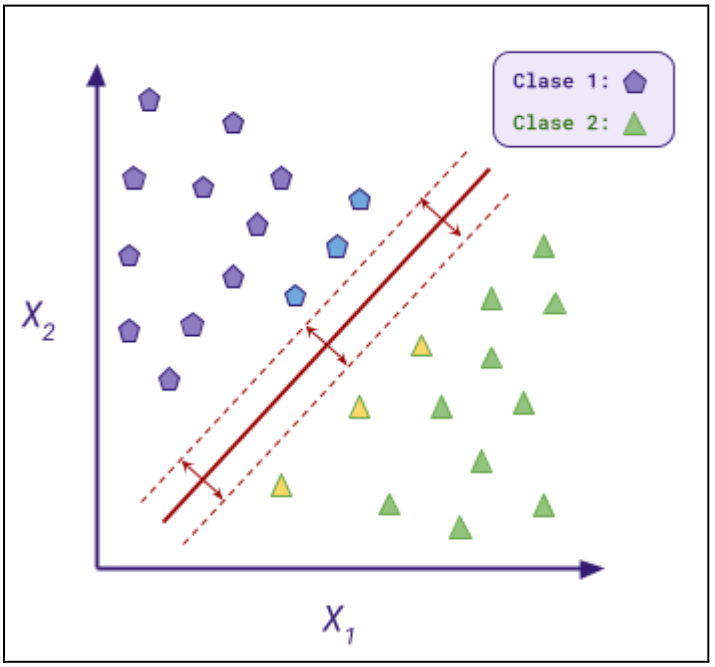

Figura 45. Gráfico de un hiperplano generado con SVM para un problema de clasificación binaria con dos características, X_1 y X_2. La recta roja es el hiperplano que separa las dos clases. Las líneas punteadas establecen el margen calculado utilizando los puntos más cercanos, los vectores de soporte, que se destacan con otro color [64].

En el caso de que los datos de entrenamiento no sean linealmente separables, el SVM utiliza un truco matemático llamado *kernel*. Consiste en mapear los datos a un espacio de características de mayor dimensión donde es más probable que sean linealmente separables, como se muestra en la **Figura 46**.

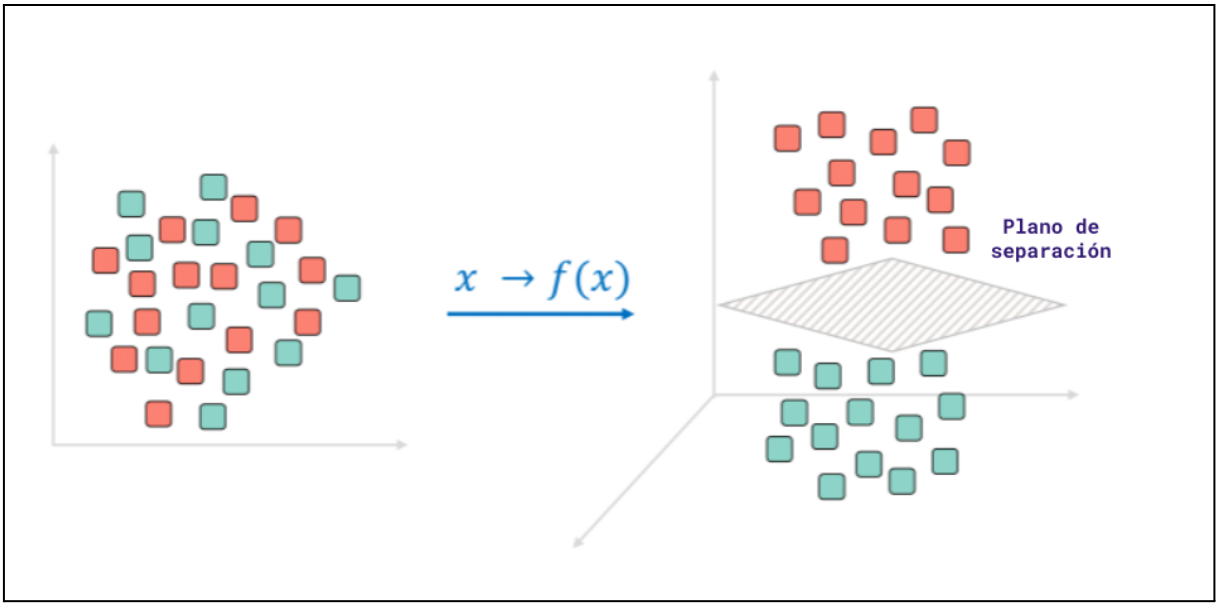

Figura 46. Visualización del efecto del *kernel*, con datos de dos clases, una color rojo y otra color verde. Izquierda: distribución de las muestras en un espacio de dos características, donde no es posible encontrar un buen hiperplano que las separe. Derecha: aplicación de una función *kernel* que lleva a las muestras a un espacio de mayor dimensión, donde se puede encontrar un plano que las separe. Figura adaptada de [65].

Además, el SVM es resistente al sobreajuste debido a la técnica de margen suave, que permite la presencia de puntos de entrenamiento en el margen o incluso en el lado incorrecto del hiperplano de separación. También puede manejar problemas de alta dimensionalidad con un número de características mucho mayor que el número de puntos de entrenamiento.

Sin embargo, el SVM puede ser sensible a la elección del kernel y a los parámetros de ajuste, lo que puede llevar a un rendimiento subóptimo si no se ajustan correctamente. Además, puede ser computacionalmente costoso en problemas de gran escala con una gran cantidad de puntos de entrenamiento.

La implementación de SVM ofrece varias opciones de kernel: *rbf, poly* y *sigmoid*, que cuentan con regularización exponencial de fuerza gamma, y *linear* y *precomputed* tienen regularización L1 de fuerza C. El kernel *poly* es una función polinomial de grado degree. La importancia de cada clase puede incluirse como en otros algoritmos junto a la regularización de C mediante class_weight, y al ser un algoritmo iterativo, se puede definir el criterio de frenado con tol o la cantidad máxima de iteraciones con max_iter, como en casos anteriores.

4.4.2.6 Gaussian Naive Bayes

El algoritmo de Gaussian Naive Bayes (GNB) es un método de clasificación de aprendizaje supervisado que se basa en el teorema de Bayes y asume que la distribución de probabilidad de las características es Gaussiana [13]. Los métodos ingenuos (*naive*) asumen que las características son independientes, y por lo tanto las probabilidades asociadas a una característica no condicionan a las probabilidades de otra.

En el proceso de entrenamiento, se calcula la probabilidad a priori de cada clase (la proporción de muestras de esa clase para el total), así como la probabilidad condicional de cada

característica dada cada clase. La clasificación de cada observación depende de su combinación única de características, utilizando las probabilidades antes mencionadas [25]. Para cada característica, el algoritmo conoce la distribución normal de los datos de entrenamiento según la clase a la cual pertenecen, como se muestra en la **Figura 47**.

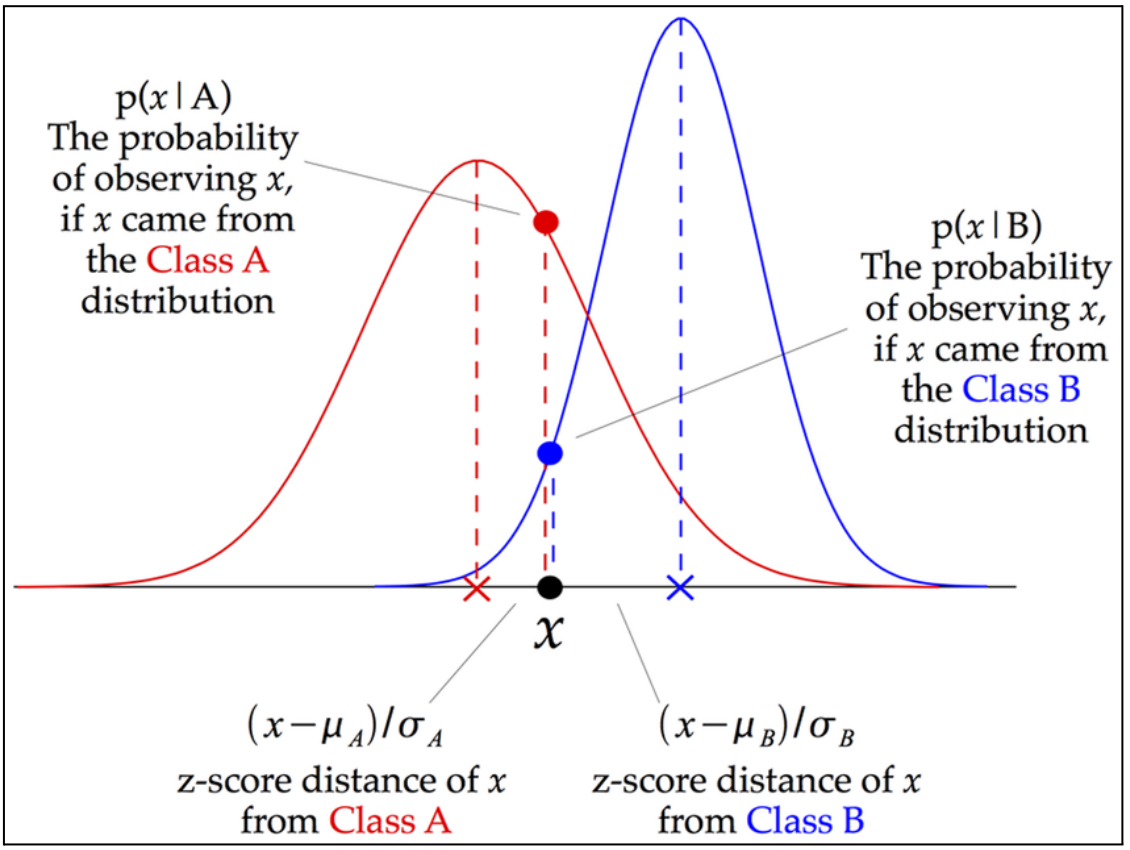

Figura 47. Gráfico para un clasificador binario GNB (*Class A*, roja, *Class B*, azul) y una muestra a clasificar (*x*, negro). Según la distribución de cada clase, se asigna un puntaje a *x* para pertenecer a la clase *A* o *B*, y se elige la clase más probable [66].

Durante la fase de clasificación, el algoritmo utiliza las probabilidades calculadas previamente para determinar la clase más probable para una observación dada. Es decir, para una nueva observación, el algoritmo calcula la probabilidad de que pertenezca a cada clase, y asigna la clase con la probabilidad más alta a la nueva observación.

Una de las ventajas del algoritmo es su simplicidad y velocidad de entrenamiento y clasificación, lo que lo hace adecuado para grandes conjuntos de datos. Sin embargo, una de las limitaciones es que asume la independencia entre características, lo que puede no ser cierto en algunos casos.

Al momento de realizar el entrenamiento, la implementación de GNB no contaba con hiperparámetros ajustables.

4.4.2.7 Análisis de Discriminante Lineal

El Análisis de Discriminante Lineal (*Linear Discriminant Analysis*, LDA) es un algoritmo de aprendizaje supervisado utilizado para la clasificación de datos en dos o más clases. Se basa en la idea de proyectar los datos de entrada en un espacio de menor dimensión mientras se

maximiza la separación entre las clases [25].

El objetivo principal del LDA es encontrar una combinación lineal de las variables de entrada que maximice la separación entre las clases y minimice la variabilidad dentro de las clases. En otras palabras, busca encontrar la dirección en la que los datos están mejor separados por clase. En principio funciona de forma similar al algoritmo PCA explicado en la Sección 4.4.1.6, ya que ambos utilizan un criterio para reducir la cantidad de características. Sin embargo, PCA solo utiliza los valores de las características de las muestras, mientras que el criterio de LDA contempla la clase asociada a cada observación.

En el proceso de entrenamiento, el algoritmo busca la transformación que lleve a los datos de entrada a un espacio que cumpla el objetivo de separación. El método usado para resolver el problema se elige con solver, que ofrecen diferentes métodos de regularización: shrinkage, que actúa como el hiperparámetro de regularización en modelos anteriores, y tol, que establece un valor mínimo para que una nueva característica en el nuevo espacio se considere significativa (de lo contrario se elimina).

4.4.2.8 Extreme Gradient Boosting

Extreme Gradient Boosting (XGB) es un algoritmo que utiliza una técnica llamada *boosting* para mejorar la precisión de los modelos. El boosting es otro tipo de ensamble que se realiza a partir de una secuencia de modelos débiles, construyendo sobre los errores de un modelo para mejorar el próximo. El resultado final es un modelo más fuerte y preciso, y el proceso se ilustra en la **Figura 48.**

Los modelos débiles en XGB son árboles de decisión, construidos de manera secuencial, con cada árbol tratando de corregir los errores del árbol anterior. Este proceso se basa en una función de costo que mide la diferencia entre las predicciones del modelo y los valores reales de los datos de entrenamiento. Utiliza una función de costo llamada *Gradient Boosting*, que calcula el gradiente de la función de costo para minimizar el error en cada iteración [25].

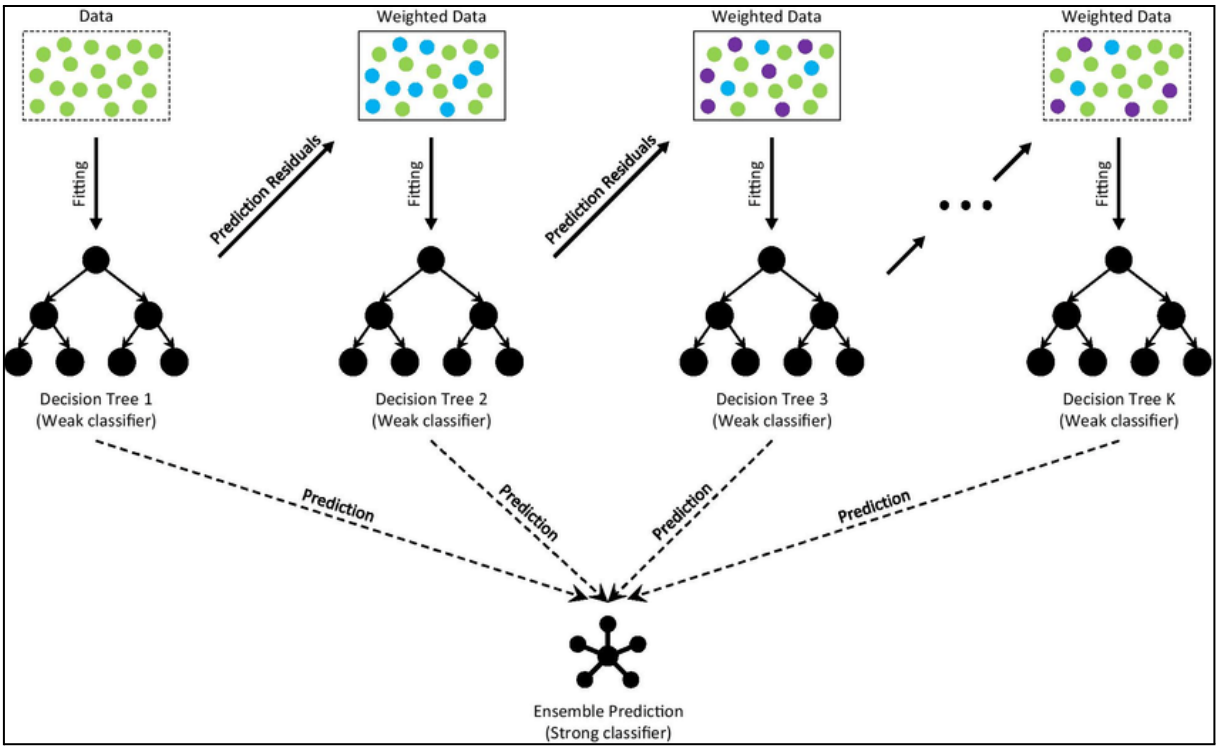

Figura 48. Gráfico de la técnica de *boosting* utilizando un ensamble de árboles de decisión. Se comienza con el ajuste de los datos de entrenamiento a un árbol de decisión. Según los resultados del entrenamiento, se ponderan los datos antes de hacer un nuevo ajuste, idealmente para mejorar el modelo, con el árbol final *K* siendo el mejor de todos. El puntaje de salida de todos los árboles se pondera en un ensamble para hacer la predicción [67].

Otra característica importante de este algoritmo es su capacidad para manejar datos de alta dimensión y características escasas. Esto se logra mediante el uso de una técnica llamada factorización de matriz regularizada, que reduce la dimensionalidad de las características y mejora la eficiencia computacional.

XGB tiene dos algoritmos basados en árboles que se pueden elegir con booster: *gbtree* y *dart* (el segundo puede eliminar árboles que considera sobreajustados como método de regularización). Si booster está basado en árboles, comparte hiperparámetros con los algoritmos antes mencionados, como max_depth.

Otros hiperparámetros son subsample, que establece el porcentaje de los datos de entrenamiento que se muestrea para realizar cada interacción; lambda, el coeficiente de regularización L2 que se puede aplicar en el entrenamiento; y max_delta_step, que establece un valor máximo que pueden alcanzar los parámetros de cada árbol.

Por su similitud con RF, XGB es capaz de manejar datos faltantes en los conjuntos de datos, lo que lo hace ideal para problemas en donde los datos pueden ser incompletos o faltantes. Además, es eficiente computacionalmente, tiene una excelente capacidad para manejar grandes cantidades de datos y una facilidad de uso para la optimización de hiperparámetros.

4.4.2.9 Gradient Boosting Classifier

El Gradient Boosting Classifier (GBC) es otro algoritmo de la familia de los métodos de

ensamble por *boosting*. Conceptualmente, funciona igual que el algoritmo XGB, pero GBC proviene de una implementación diferente [25]. Además, tiene una diferencia funcional en que XGB utiliza regularización L2, mientras que GBC no.

Cuenta con las mismas ventajas, como la capacidad de manejar conjuntos de datos grandes y complejos, variables categóricas y numéricas, y de detectar y manejar valores atípicos y datos faltantes. Su entrenamiento también es computacionalmente intensivo y requiere de ajustes de hiperparámetros adecuados para lograr el mejor rendimiento.

Ya que su implementación utiliza el algoritmo DT, GCB comparte varios hiperparámetros con ellos y con RF (n_estimators, criterion, min_samples_split, min_samples_leaf, max_features, max_depth). Además, se puede elegir la función de costo con loss, con la posibilidad de usar *deviance* y *exponential*. Finalmente, se puede ajustar la contribución de cada clasificador débil mediante learning_rate, un número racional positivo que multiplica el aporte de cada uno de estos clasificadores. Si es un número muy pequeño, los ajustes en cada iteración son más conservadores, y requieren de más árboles para llegar a un buen resultado, pero lleva a un entrenamiento más estable, y es la práctica preferida.

4.4.2.10 AdaBoost Classifier

El AdaBoost Classifier (ABC) es un algoritmo de aprendizaje supervisado que se utiliza para la clasificación de datos en dos o más clases. Como XGB y GBC, es un algoritmo de ensamble por *boosting*. La lógica de su funcionamiento es similar a los algoritmos anteriores, pero permite trabajar con diferentes algoritmos de clasificación, y optimiza el modelo con una función de costo exponencial, en vez del uso de gradientes [25].

Aunque permite trabajar con diferentes algoritmos, el modelo clásico de ADB se basa en la construcción de árboles de decisión con un único nodo, lo que genera un clasificador débil. En el proceso de entrenamiento, el algoritmo asigna parámetros a cada observación, dando mayor peso a las observaciones que fueron mal clasificadas en el modelo anterior y menor peso a las observaciones que fueron correctamente clasificadas. Luego, se ajusta un modelo de clasificación débil en los datos ponderados y se repite el proceso para construir un conjunto de modelos.

Una vez que se construyen los modelos, se combinan para formar un modelo final utilizando una votación ponderada de los modelos individuales. En general, los modelos más débiles se combinan para formar un modelo más fuerte que puede clasificar con precisión nuevos datos de entrada.

Ya que ABC permite seleccionar el algoritmo que se usará en el entrenamiento con estimator (por defecto es un DT), no hay una forma directa de elegir los hiperparámetros de este algoritmo subyacente. Es posible ajustar la cantidad máxima de clasificadores entrenados por ABC con n_estimators, y el valor de learning_rate.

4.5 Selección y ensamble de modelos

Una vez finalizado el ajuste de hiperparámetros, se utilizaron los modelos para inferir sobre el conjunto de evaluación. Los puntajes de salida se utilizaron para calcular las métricas de

desempeño de cada experimento, que se usaron posteriormente para seleccionar los mejores modelos. El primer paso en el cálculo de las métricas fue la obtención de la curva ROC. Se usó la función *roc_curve* de Scikit-Learn, que recibe como entrada el puntaje de cada muestra asociado a la pertenencia a la clase positiva, y el valor de la etiqueta de cada muestra. La función *auc_roc_score* funciona de la misma forma, y computa el AUC ROC. El mismo procedimiento se siguió para hacer el gráfico de la curva PR y conseguir el AUC PR, usando las funciones *precision_recall_curve* y *auc*.

Los valores de AUC ROC se utilizaron para elegir aquellos modelos con mejor desempeño, tomando como criterio de selección un AUC ROC de al menos 0,8 y menor a 1 en el conjunto de entrenamiento, y de al menos 0,7 en el conjunto de evaluación. Se evitó usar los modelos con AUC ROC de 1 por la posibilidad de estar sobreajustados a los datos de entrenamiento, y se tomó un piso menor para el conjunto de evaluación ya que, al ser datos que el clasificador no ha usado para entrenar, sus resultados tienden a ser más bajos.

Aquellos experimentos que superaron el criterio de selección se utilizaron en el ensamble de modelos, un método que tiene en cuenta la salida de más de un modelo antes de hacer la clasificación. Está basado en el hecho de que modelos diferentes, que pueden haber sido entrenados con diferentes configuraciones (características, algoritmos e hiperparámetros), pueden tener un desempeño distinto al evaluar un mismo caso. Al tener en cuenta la respuesta de diferentes modelos, es posible obtener una clasificación que sea más robusta, con el concepto de que uno o varios modelos que en un caso funcionan bien, mejoran el resultado de un modelo que en ese mismo caso no. Esta lógica es aplicada por los algoritmos de ensamble antes explicados.

Se realizó un ensamble del tipo *bagging*. El puntaje de salida del ensamble para un dato se calculó como el promedio de los puntajes de salida emitidos por cada modelo integrante del ensamble para ese dato. A partir de este puntaje se calcularon las métricas de desempeño del ensamble, y también se realizó la búsqueda del umbral óptimo.

Entre los modelos que cumplieron con los criterios de selección basados en el valor de AUC ROC para los conjuntos de entrenamiento y evaluación, para cada fase y variable se eligió la menor cantidad de modelos que produjera el mejor ensamble, o un único modelo si no fue posible lograr un ensamble con mejores métricas que los modelos individuales.

Finalmente, se buscó el umbral óptimo de clasificación tanto para los modelos individuales como para los ensambles. Este proceso también fue experimental, ya que se probó calcular utilizando diferentes índices que se detallan en la Sección 4.6.1. Con el umbral encontrado por cada uno de estos métodos, se obtuvieron las predicciones (pertenencia a una clase o a la otra) según el puntaje de salida del modelo estuviese por debajo o por encima del valor umbral computado. Para cada experimento, la búsqueda del umbral siempre se hizo sobre los datos de entrenamiento, y se usó el mismo umbral hallado en los datos de evaluación.

4.6 Evaluación de modelos

Una vez clasificadas las muestras con cada modelo (sea uno individual o el ensamble de varios), se calculó la matriz de confusión, y se repitieron los gráficos de las curvas ROC y PR. Además, se calcularon siete métricas de discriminación y tres de calibración que están detalladas en esta sección. Se hizo un *bootstrap* de 1.000 iteraciones para estimar los IC del 95% de las métricas de cada modelo. Se utilizaron los puntajes de salida de cada modelo y se muestreó con reposición, garantizando la existencia de muestras de ambas clases. Todas las funciones mencionadas en esta sección provienen de Scikit-Learn, y los gráficos se realizaron con el paquete Matplotlib [68].

4.6.1 Métodos de selección de umbral

Como fue comentado anteriormente, para asignar una clase a una muestra se observa el puntaje emitido para ella por el modelo. Por defecto, si es mayor o igual a 0,5, en clasificación binaria se considera que la muestra es de la clase positiva, de lo contrario se considera que es negativa. De acuerdo a los puntajes dados por el clasificador durante el entrenamiento, es posible que decidir con ese umbral no lleve a los mejores resultados. La curva ROC nos permite ver el desempeño de diferentes umbrales, y es una opción elegir uno a partir de ese gráfico, pero también existen métodos para calcular un umbral de resultados óptimos.

Una forma de obtener el umbral óptimo es mediante el cálculo de la media geométrica entre la sensibilidad y el complemento de la especificidad como función de los umbrales de clasificación. Se elige el umbral que resulte en la mayor media geométrica [69].

$$Media\ Geométrica\ = \sqrt{Sen(1\ -\ Esp)} \tag{93}$$

Otro método de selección es el cálculo del índice de Youden (*J*). Se propone que la combinación de sensibilidad y especificidad que mayor índice de Youden produzca será también la que maximice ambas métricas [70].

$$J\ = Sen\ -\ Esp\ -\ 1 \tag{94}$$

Por otro lado, el criterio del punto más cercano (*ER*) busca el umbral que produzca el punto más cercano a la esquina superior izquierda en el gráfico de la curva ROC, que podría favorecer a una métrica más que a la otra [70].

$$ER\ = \sqrt{(1\ -\ Sen)^2\ +\ (1\ -\ Esp)^2} \tag{95}$$

Una alternativa más reciente es el índice de unión (*IU*) incorpora el valor de AUC ROC para encontrar la combinación que maximiza tanto sensibilidad como especificidad [70].

$$IU\ = |\ Sen\ -\ AUC|\ +\ |Esp\ -\ AUC| \tag{96}$$

Finalmente, el índice CZ propone que el mejor umbral es aquel que maximice el producto entre la sensibilidad y especificidad [71].

$$CZ = Sen * Esp \tag{97}$$

4.6.2 Métricas de discriminación

En esta sección se listan algunas métricas comúnmente calculadas en base a los aciertos y errores en la predicción, y usadas en este proyecto. Si bien todas tienen valores de cero a uno, donde cero es el peor resultado y uno el mejor, lo normal es expresarlas como porcentaje [13]. Para algunas de estas métricas, se usará el ejemplo de un clasificador de TC abdominal para la detección de tumores hepáticos.

1. **Exactitud** (*Accuracy*, **Acc**): es la proporción de aciertos sobre el total de casos. Es la métrica más intuitiva, pero no es sensible a diferencias en la prevalencia de las clases, y no distingue entre un clasificador que favorezca la correcta clasificación de positivos, negativos o ambos.

$$Acc = \frac{VP + VN}{VP + VN + FP + FN} \tag{98}$$

2. **Sensibilidad**: también conocida como *recall* y tasa de verdaderos positivos, es la probabilidad condicional de que el resultado de la clasificación sea positivo dado que el caso en cuestión pertenece a la clase positiva. Si el clasificador del ejemplo tiene una sensibilidad de 0,8, entonces hay un 80% de probabilidades de que el resultado sea positivo (se detectan tumores), dado que exista una lesión maligna en la tomografía.

$$Sen = \frac{VP}{VP + FN} \tag{99}$$

3. **Especificidad**: también conocida como tasa de verdaderos negativos, es la probabilidad condicional de que el resultado de la clasificación sea negativo dado que el caso en cuestión pertenece a la clase negativa. Siguiendo el ejemplo anterior, si el clasificador tiene una especificidad de 0,8, entonces hay un 80% de probabilidades de que el resultado sea negativo (no se detectan tumores), dado que no exista una lesión maligna en la tomografía.

$$Esp = \frac{VN}{FP + VN} \tag{100}$$

   En este proyecto, la importancia de la sensibilidad y especificidad depende de la variable de respuesta.

Para Óbito, ambas son importantes, con prioridad de la sensibilidad. Un FP sugiere que el paciente no tiene un buen pronóstico tras la cirugía de resección y debería seguir otro tratamiento, cuando realmente el pronóstico es positivo. Este caso implica un retraso en la cirugía y el mejoramiento del paciente. Un FN sugiere que el pronóstico es bueno cuando realmente no lo es, y no se debería avanzar con la cirugía. En la mayoría de los casos, la consecuencia de un FN es más grave que la de un FP, con lo cual la sensibilidad tendría más peso.

Para KRAS, la sensibilidad tiene mayor relevancia, pero los FN no tienen la misma gravedad. El estado mutado del gen actúa como un factor de riesgo o un agravante del pronóstico del paciente, con lo cual su clasificación se asemeja a una prueba de cribado. Si el modelo clasifica un caso como mutado, entonces se puede confirmar mediante un estudio de laboratorio. Si bien un FN es problemático, ya que resta pruebas de la agresividad de la enfermedad, y así afecta negativamente a la decisión de tratamiento, no es tan grave como en el caso de Óbito.

4. **Valor Predictivo Positivo** (**VPP**): también conocido como precisión, es la proporción de casos realmente positivos correctamente clasificados sobre el total de casos clasificados como positivos por el modelo, independientemente de que realmente sean positivos o no. En el ejemplo, un VPP de 0,8 indica que hay una probabilidad de 80% de que un caso identificado como positivo por el modelo, realmente tenga una lesión maligna.

$$VPP = \frac{VP}{VP + FP}$$

(101)

5. **Valor Predictivo Negativo** (**VPN**): es la proporción de casos realmente negativos correctamente clasificados sobre el total de casos clasificados como negativos por el modelo, independientemente de que realmente sean negativos o no. En el ejemplo, un VPN de 0,8 indica que hay una probabilidad de 80% de que un caso identificado como negativo por el modelo realmente esté libre de lesiones malignas.

$$VPN = \frac{VN}{VN + FN}$$

(102)

Los valores predictivos dependen de la prevalencia de las clases en el conjunto de datos, y el valor predictivo de la clase más prevalente tiende a ser mayor.

6. **Valor F1** (**F1**): es una medida de exactitud compuesta, calculada como la media armónica entre la sensibilidad y el valor predictivo positivo.

$$F1 = \frac{2\,Sen\,VPP}{Sen + VPP}$$

(103)

7. **Coeficiente de correlación de Matthews** (MCC): también conocido como coeficiente phi, es una medida de correlación entre las clases, y de la capacidad de un modelo a

ajustarse a los datos. Un valor de 0 indica ausencia de correlación y una clasificación aleatoria, un valor de 1 indica una correlación perfecta, y un valor de -1 indica una correlación inversa. Al ser un caso específico de la correlación de Pearson aplicado a la matriz de confusión, la interpretación de sus valores es idéntica.

$$MCC = \frac{VP\ VN - FP\ FN}{\sqrt{(VP + FP)\ (VP + FN)\ (VN + FP)\ (VN + FN)}}$$

(104)

Todas las métricas de clasificación se calcularon según las fórmulas antes descritas, con la excepción de Acc y F1, que se calcularon con las funciones *accuracy_score* y *f1_score*.

4.6.3 Métricas de calibración

A continuación se listan y definen las métricas calculadas para evaluar la calibración de los modelos. Estas funciones definen errores, y por lo tanto, un valor cercano a cero indica un error chico, y un modelo calibrado.

1. **Error Esperado de Calibración** (*Expected Calibration Error,* **ECE**): esta métrica calcula el error entre la confianza del algoritmo (puntaje de salida) y su fracción positiva [31]. Para calcularla, se divide el intervalo de la confianza entre cero y uno en M_i *bins* del mismo tamaño:

$$bin = (\frac{i-1}{M},\ \frac{i}{M}]$$

(105)

B_i son las muestras cuyos puntajes caen en el *i*-ésimo *bin*, y A_i es la fracción positiva verdadera en ese *bin*. C_i es el puntaje de salida promedio en el *bin*:

$$C_i = \frac{1}{|B_i|} \sum_{j \in B_i} \widehat{p}_j$$

(106)

Finalmente, el ECE se calcula como:

$$ECE = \sum_{i=1}^{M} \frac{|B_i|}{N} |A_i - C_i|$$

(107)

2. **Error Máximo de Calibración** (*Maximum Calibration Error,* **MCE**)**:** esta métrica similar es la máxima diferencia absoluta entre la proporción de casos de la fracción positiva y el puntaje de salida promedio de cada *bin*:

$$MCE = max_{i \in \{1, \ldots, M\}} (A_i - C_i)$$

(108)

3. **Brier Score (BS):** este puntaje se puede usar en tareas de clasificación de eventos mutuamente excluyentes, como en un clasificador binario. En el caso binario, se puede calcular como el error cuadrático medio entre la predicción y la etiqueta en *N* casos [72]:

$$BS = \frac{1}{N}\sum_{j=i}^{N}(\widehat{y_j} - y_j)^2 \quad (109)$$

Para las métricas de calibración, se usó la función *calibration_curve* con diez bins de ancho 0.1, para los cuales se consigue la fracción de positivos y probabilidad esperada de cada bin. Con ellas, se calcularon los ECE y MCE, mientras que el BS se calculó con la función *brier_score_loss*.

5. Resultados

En esta sección se muestran los resultados de todos los procedimientos del proyecto, desde el armado de la base de datos hasta las métricas de los modelos finales. La descripción completa del estudio se presenta en el Anexo A. Los mejores modelos en esta sección son ensambles, y la información de los modelos que los componen se encuentra en el Anexo C. Los resultados para los objetivos secundarios se presentan en el Anexo D.

5.1 Resultados Óbito FVP

5.1.1 Descripción de la población

Se obtuvieron datos de 136 pacientes, con una (86 negativos, 51 positivos; 84 hombres, 54 mujeres). La mediana de la edad fue de 62,50 años, con un rango intercuartílico de 54 a 70 años. En 53 de los pacientes no se conocía el estado de mutación del gen KRAS, mientras que para 48 fue Wild Type, y 37 fue Mutado. La FVP fue aquella con más estudios disponibles, sumando 136; Se encontraron 74 casos en FSC, 61 con FVT y 41 con FA. En la **Tabla 1** se resumen estos datos desglosados por clase.

	Total	Vive - 0	Fallecido - 1
Total	**136**	**86**	**50**
Sexo			
Hombre	83 (61,03%)	48 (55,81%)	35 (70,00%)
Mujer	53 (38,97%)	38 (44,19%)	15 (30,00%)
Edad			
Mediana	62,50 (54,00-70,00)	60 (52,25-67,50)	65,5 (59,50-74,00)
KRAS			
Desconocido	51 (37,50%)	39 (45,35%)	12 (24,00%)
Mutado	37 (27,21%)	21 (24,42%)	16 (32,00%)
Wild Type	48 (35,29%)	26 (30,23%)	22 (44,00%)
Fases			
FA	41 (30,14%)	24 (27,90%)	17 (34,00%)
FVP	136 (100,00%)	86 (100,00%)	50 (100,00%)
FSC	74 (54,41%)	42 (48,84%)	32 (64,00%)
FVT	61 (44,85%)	33 (38,37%)	28 (56,00%)

Tabla 1. Frecuencia y distribución de los datos de los pacientes respecto a la variable Óbito. Los porcentajes corresponden a las proporciones del total de cada columna.

En la **Figura 49** se muestra el histograma de la variable Edad para la población de Óbito FVP, realizado con rangos de 5 años, y se encontró que su distribución es no normal.

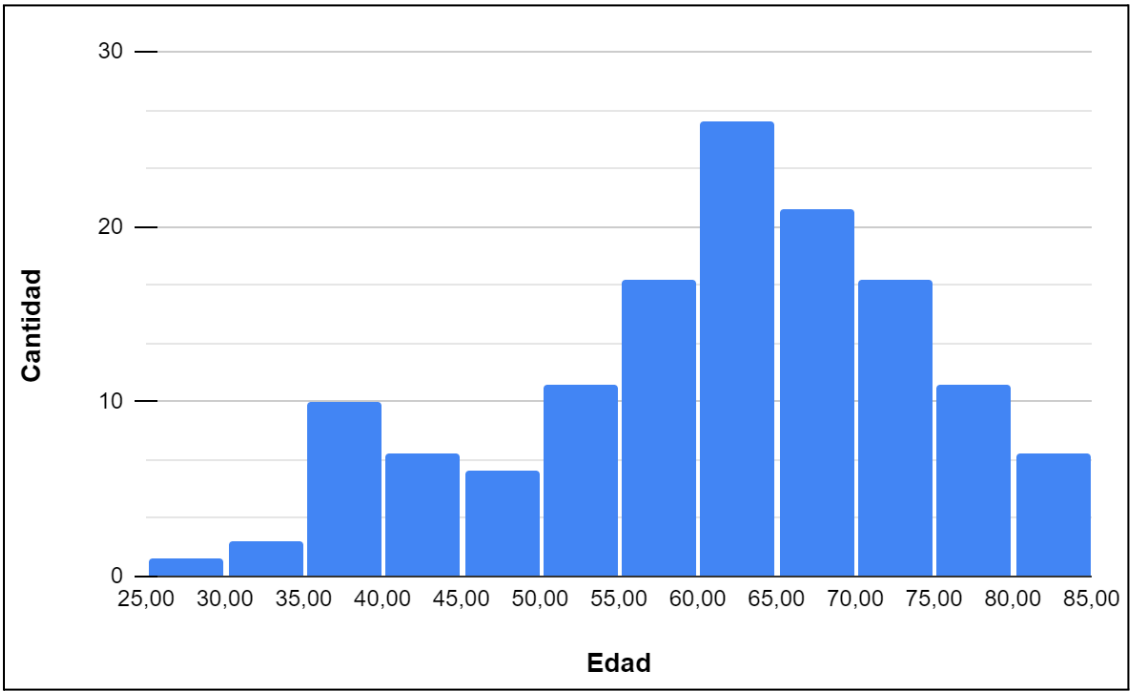

Figura 49. Histograma de edad de los pacientes (rangos de 5 años) en el estudio de Óbito.

En la **Figura 50** se muestra la cantidad de pacientes en los conjuntos de entrenamiento y evaluación, según la variable Óbito, para los pacientes con estudios en FVP. El conjunto de entrenamiento se formó con 101 casos (38 negativos, 63 positivos), mientras que el conjunto de evaluación se formó con 35 (22 negativos, 13 positivos). Esta separación sigue la proporción establecida de 75% de los casos destinados a entrenamiento, y 25% a evaluación.

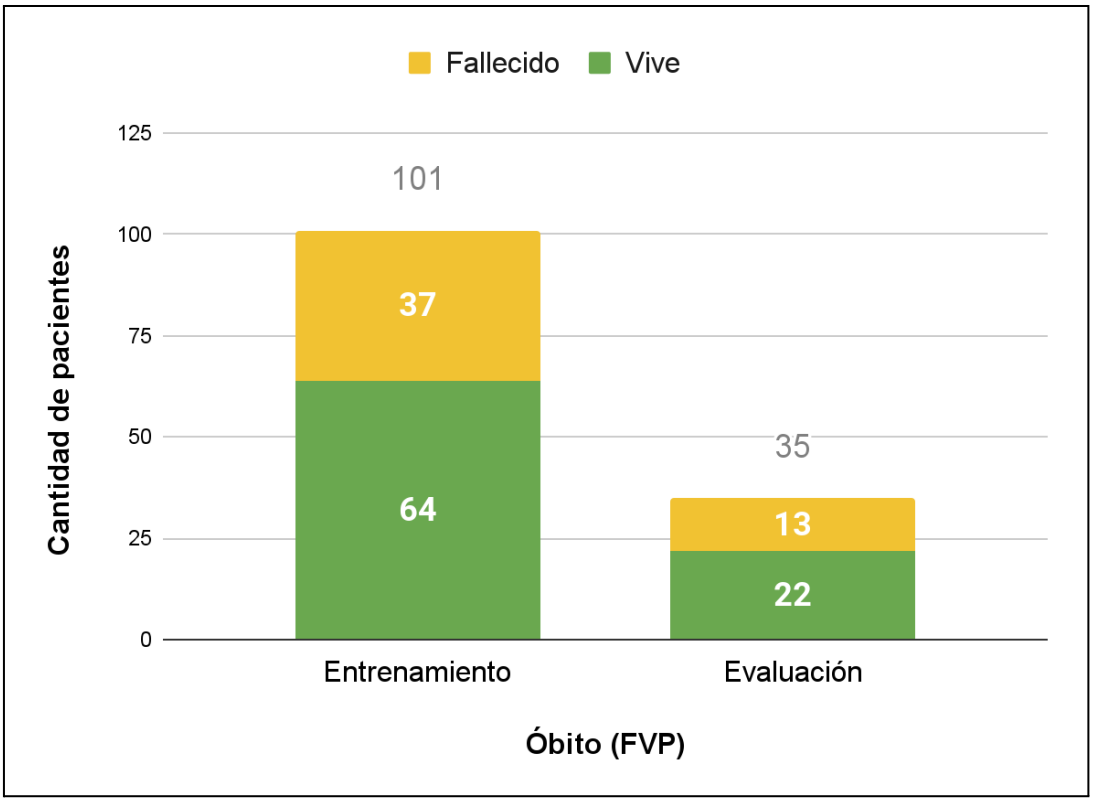

Figura 50. Distribución de Óbito en los conjuntos de entrenamiento y evaluación para pacientes con estudios en FVP.

5.1.2 Evaluación de desempeño

En total, se extrajeron 730 características individuales: 100 en el conjunto Original, 258 en el conjunto LdG, 186 en el conjunto C+RC, y 186 en el conjunto E+L. La cantidad de características en los conjuntos combinados (Original + LdG y Original + LdG + E + EL) acumularon 358 y 544 características, respectivamente. Con ellas, se realizaron se entrenaron 651 modelos en FVP para la variable Óbito, en base a la combinación de los siguientes:

- Una fase de TC: FVP, FA, FVT o FSC
- Un conjunto de datos: Original, LdG**,** C+RC**,** E+L, Original+LdG u Original+LdG+E+L
- Un método de selección de características: Correlación de Pearson, FS, BE, LASSO, ETC, LinearSCV o PCA.
- Una cantidad de características a seleccionar: 7 (solo para KRAS), 10, 20, o 30
- Un algoritmo de ML: KNN, DT, LR, SVM, RF, XGB, LDA, ABC, GNB, o GBC

Entre ellos, se eligieron como mejores los modelos FVP 1 y FVP 2, detallados en la **Tabla 2**. Estos modelos fueron seleccionados por sus valores de AUC ROC en entrenamiento (0,873 para el modelo FVP 1, y 0,876 para FVP 2) y evaluación (0,755 y 0,806, respectivamente). Se reportan las tres características que fueron encontradas de mayor importancia (según la definición de cada algoritmo) para cada modelo.

	Modelo	
	FVP 1	**FVP 2**
Extracción	Original + LdG	LdG
Selección	Eliminación hacia atrás	*Extra Trees Classifier*
Clasificador	Análisis de discriminante lineal	Árbol de decisión
Número de Características	20	10
Características de mayor importancia	original_glcm_SumSquares	original_shape_Sphericity
	original_glrlm_GrayLevelNonUniformity	log-sigma-0-5-mm-3D_glcm_Correlation
	original_glrlm_HighGrayLevelRunEmphasis	log-sigma-2-5-mm-3D_firstorder_90Percentile
AUC-ROC - E	0,873	0,876
AUC-PR - E	0,798	0,808
AUC-ROC - P	0,755	0,806
AUC-PR - P	0,618	0,751

Tabla 2. Descripción de los mejores modelos en FVP. Abreviaturas: área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), AUC ROC en entrenamiento (AUC ROC - E), AUC PR en entrenamiento (AUC PR - E), AUC ROC en evaluación (AUC ROC - P), AUC PR en evaluación (AUC PR - P), Laplaciano de Gaussiano (LdG).

En la **Tabla 3**, se detallan las métricas de discriminación para el ensamble de modelos en FVP, llamado FVP Ensamble. Se presentan tanto las métricas en entrenamiento como evaluación, calculadas con un umbral de 0,510 obtenido con el método de la Media Geométrica. Los resultados en entrenamiento superaron el 0,8, con la excepción del MCC. Aún en evaluación los números fueron altos, con la excepción del VPP, Valor F1 y MCC. Comparando evaluación con entrenamiento, se encontró un incremento en los errores de calibración.

	Óbito FVP Ensamble					
	Entrenamiento			**Evaluación**		
Métrica	**Original**	**Bootstrap**	**IC**	**Original**	**Bootstrap**	**IC**
AUC ROC	0,929	0,930	0,870-0,975	0,878	0,875	0,739-0,979
AUC PR	0,890	0,890	0,870-0,975	0,826	0,827	0,739-0,979
Sensibilidad	0,838	0,809	0,677-0,929	0,846	0,851	0,647-1,000
Especificidad	0,938	0,936	0,866-0,985	0,818	0,809	0,630-0,955
VPP	0,882	0,881	0,763-0,972	0,733	0,735	0,500-0,938
VPN	0,896	0,894	0,810-0,959	0,900	0,897	0,737-1,000
Exactitud	0,891	0,889	0,822-0,941	0,829	0,825	0,686-0,943
Valor F1	0,845	0,842	0,741-0,925	0,786	0,783	0,593-0,933
MCC	0,763	0,760	0,619-0,880	0,649	0,645	0,386-0,885
ECE	0,157	0,171	0,124-0,221	0,176	0,210	0,127-0,295
MCE	0,336	0,407	0,263-0,573	0,483	0,560	0,472-0,727
BS	0,111	0,111	0,085-0,140	0,149	0,151	0,102-0,206

Tabla 3. Métricas del modelo FVP Ensamble, separadas por los conjuntos de entrenamiento y evaluación. Los valores en la columna *bootstrap* corresponden a la media de cada métrica en las 1.000 iteraciones. Abreviaturas: intervalo de confianza (IC), área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), valor predictivo positivo (VPP), valor predictivo negativo (VPN), coeficiente de correlación de Matthews (MCC), error de calibración esperado (ECE), media del error de calibración (MCE), *Brier score* (BS).

El motivo de la brecha entre la sensibilidad y el VPP se puede ver en la **Figura 51,** que muestra las matrices de confusión de FVP Ensamble para los conjuntos de entrenamiento y evaluación. Dado que el porcentaje de verdaderos positivos cambió poco entre los conjuntos, el incremento de falsos positivos en evaluación llevó a un peor VPP.

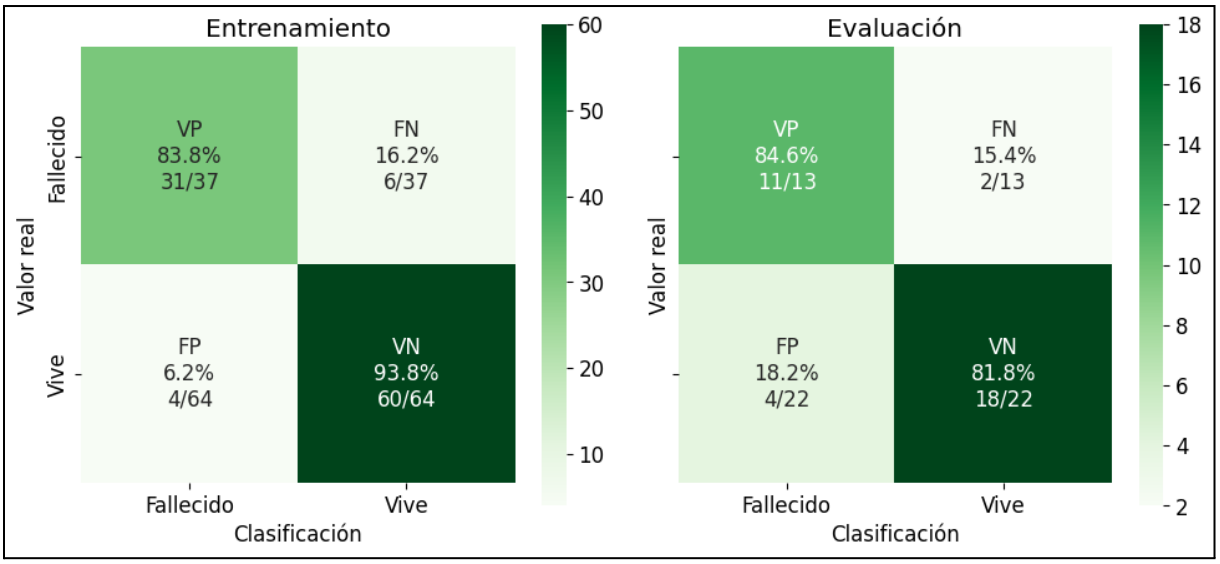

Figura 51. Matrices de confusión para FVP Ensamble. El umbral de decisión para los modelos fue 0,510. Abreviaturas: verdadero positivo (VP), falso negativo (FN), falso positivo (FP), verdadero negativo (VN).

La **Figura 52** muestra las curvas ROC de los modelos FVP 1, FVP 2, y FVP Ensamble, en entrenamiento y evaluación. En los tres modelos se observó un AUC ROC menor para la evaluación, pero también un AUC ROC mayor de FPV Ensamble respecto a los modelos individuales.

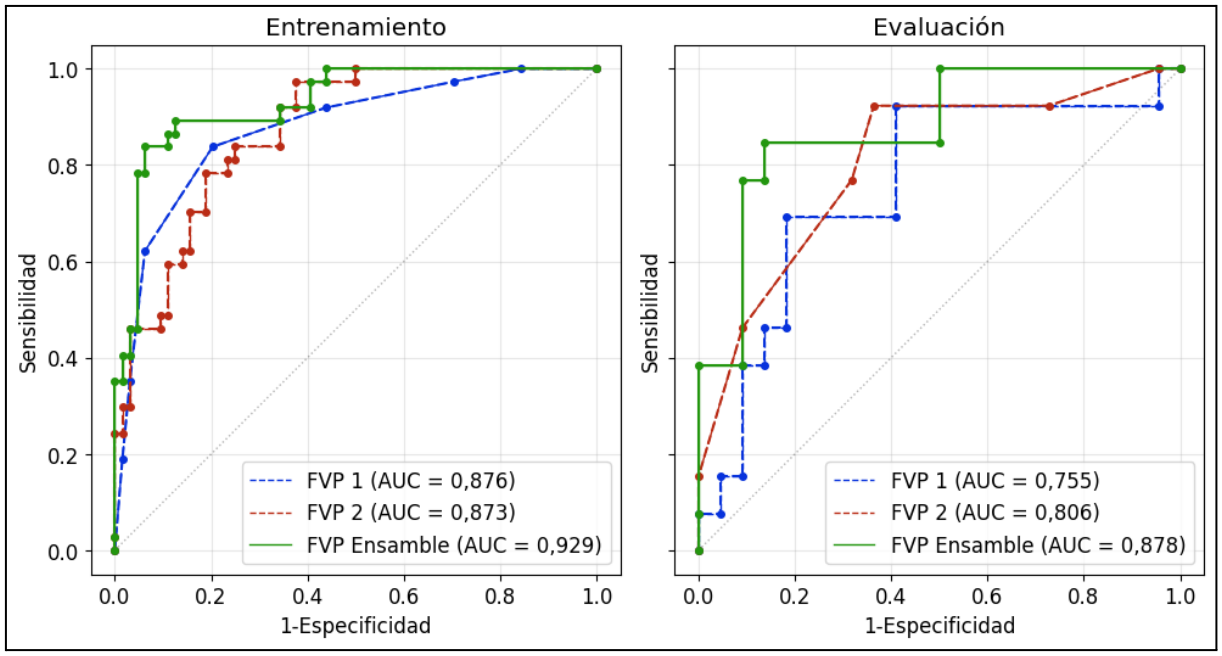

Figura 52. Curvas ROC los modelos de Óbito en FVP. Los modelos individuales están trazados con líneas discontinuas, mientras que el ensamble es una línea sólida. Los posibles umbrales están marcados con puntos de mayor grosor. La línea diagonal gris marca el umbral de no discriminación. Abreviaturas: área bajo la curva (AUC).

Se encontraron resultados similares en las curvas PR, como se muestra en la **Figura 53**. El AUC PR de todos los modelos en evaluación fue menor que en entrenamiento, pero en ambos conjuntos el AUC PR de FVP Ensamble superó al de los modelos individuales. Hubo una mayor diferencia entre el AUC PR que el AUC ROC entre entrenamiento y evaluación.

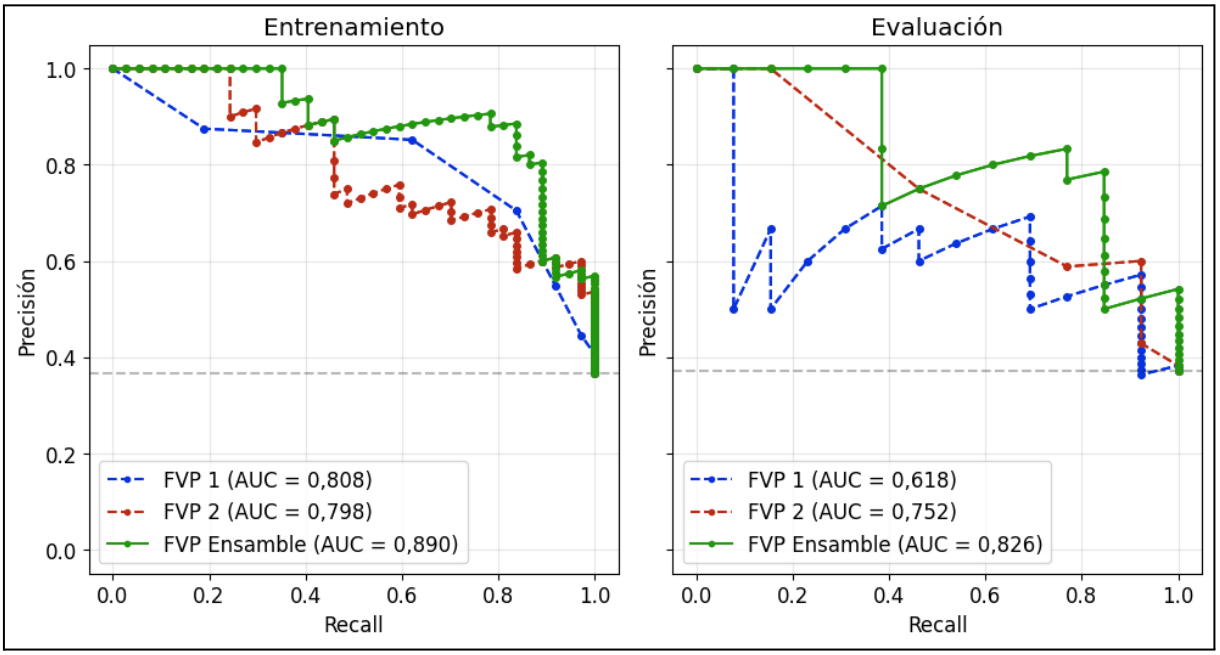

Figura 53. Curvas PR para los modelos de Óbito en FVP. Los modelos individuales están trazados con líneas discontinuas, mientras que el ensamble es una línea sólida. Los posibles puntos de corte están marcados con puntos de mayor grosor. La línea recta gris marca el umbral de no discriminación. Abreviaturas: área bajo la curva (AUC).

5.2 Resultados KRAS FVP

5.2.1 Descripción de la población

Se obtuvieron datos de 85 pacientes (48 negativos, 37 positivos; 51 hombres, 34 mujeres). Para esta población, la mediana de la edad fue de 61 años, con un rango intercuartílico de 54 a 67 años. En la **Tabla 4** se resumen los datos de la variable KRAS, desglosados por clase.

	Total	Wild Type - 0	Mutado - 1
Total	**85**	**48**	**37**
Sexo			
Hombre	51 (60,00%)	32 (66,67%)	19 (51,35%)
Mujer	34 (40,00%)	16 (33,33%)	18 (48,65%)
Edad			
Mediana	61 (54-67)	59,5 (46,75-65,50)	62 (58-69)
Obito			
Vive	47 (55,29%)	26 (54,17%)	21 (56,76%)
Fallecido	38 (44,71%)	22 (45,83%)	16 (43,24%)
Fases			
FA	26 (30,59%)	15 (31,25%)	11 (29,73%)
FVP	85 (100,00%)	48 (100,00%)	37 (100,00%)
FSC	48 (56,47%)	28 (58,33%)	20 (54,05%)
FVT	39 (45,88%)	22 (45,83%)	17 (45,95%)

Tabla 4. Frecuencia y distribución de los datos de los pacientes respecto a la variable KRAS. Los porcentajes corresponden a las proporciones del total de cada columna.

En la **Figura 54** se muestra el histograma de la variable Edad para la población KRAS, realizado con rangos de 5 años. Su distribución es no normal, y se mantiene similar al histograma para la población en Óbito, con frecuencias más bajas.

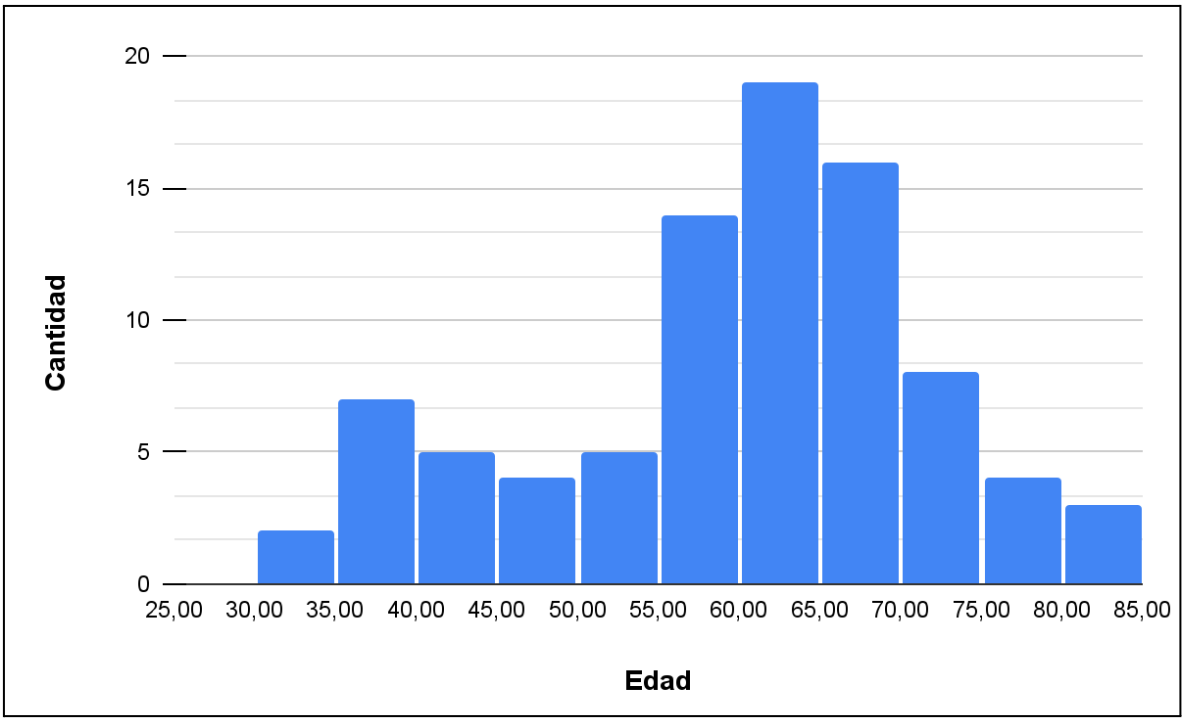

Figura 54. Histograma de edad de los pacientes (rangos de 5 años) para la variable KRAS.

En la **Figura 55** se muestran los casos de la variable KRAS según la variable Óbito. El valor de Óbito para 47 pacientes fue negativo (Vive), mientras que hay 38 casos fueron positivos (Fallecido). Similar al gráfico en la sección anterior, la cantidad de casos positivos (Mutado) y

negativos (Wild Type) para la variable KRAS es parecida, con una mayoría de casos negativos.

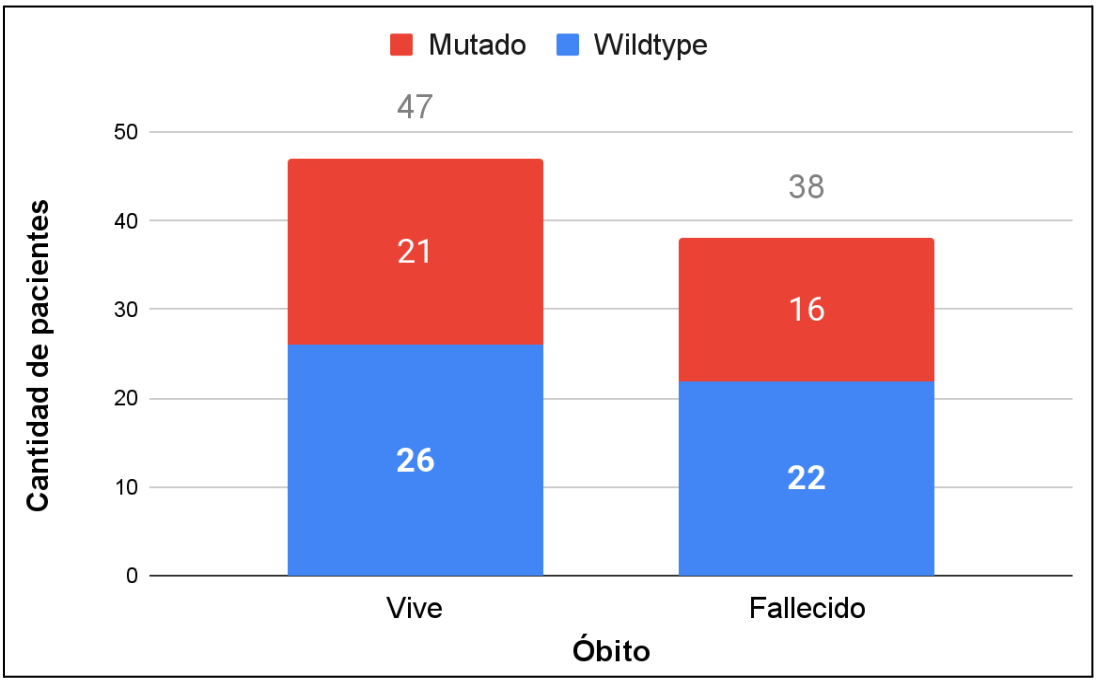

Figura 55. Cantidad de imágenes de la variable KRAS respecto a la variable Óbito.

La partición en conjuntos de entrenamiento y evaluación se muestra en la **Figura 56**. El conjunto de entrenamiento se formó con 55 casos (24 positivos, 31 negativos), mientras que el de evaluación se formó con 30 casos (13 positivos, 17 negativos). Esta separación sigue la proporción establecida de 65% de los casos destinados a entrenamiento, y 35% a evaluación.

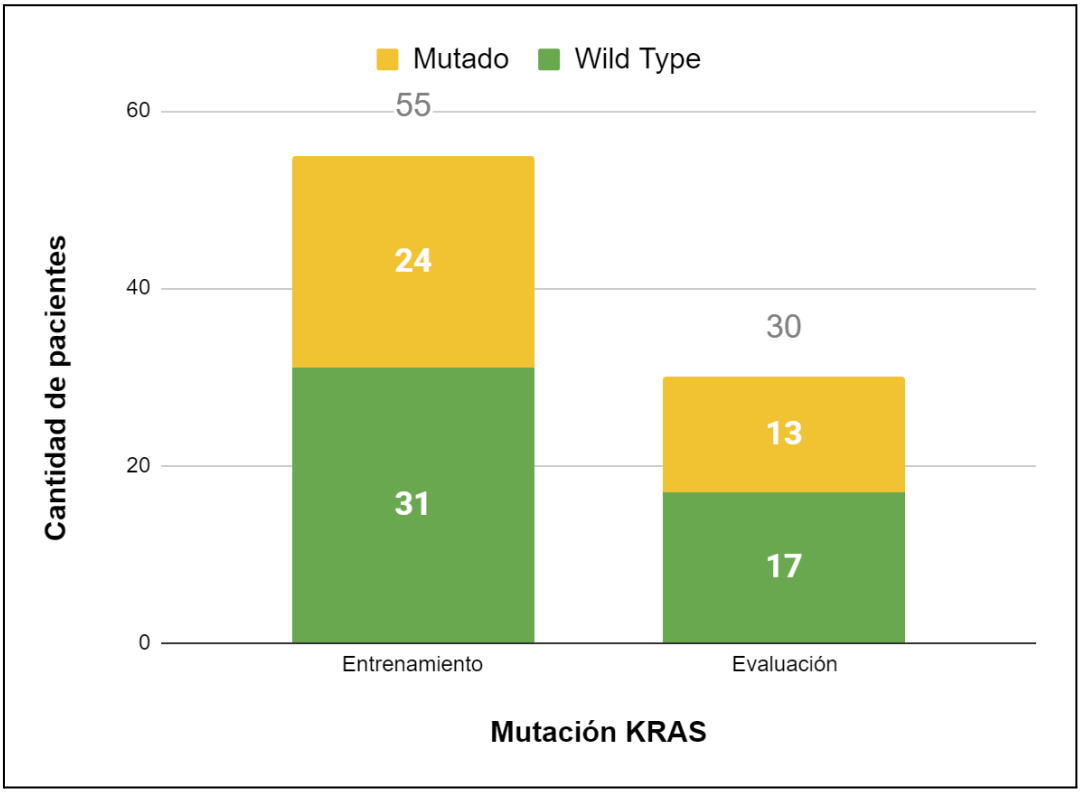

Figura 56. Distribución de la variable KRAS en los conjuntos de entrenamiento y evaluación.

5.2.2 Evaluación de desempeño

Para la variable KRAS, se usó solo la FVP, se definió la posibilidad de seleccionar hasta 7 características, y se evitaron las combinaciones de peor desempeño en los experimentos de Óbito. Utilizando los conjuntos Original, LdG y Original+LdG, se entrenaron 606 modelos. Los mejores entre ellos, KRAS 1, KRAS 2 y KRAS 3, se muestran en la **Tabla 5**, seleccionados por sus valores de AUC ROC en entrenamiento (0,818, 0,995 y 0,834, respectivamente) y evaluación (0,735, 0,724, y 0,776, respectivamente). Se reportan las tres características que fueron encontradas de mayor importancia (según la definición de cada algoritmo) para cada modelo.

	Modelo		
	KRAS 1	**KRAS 2**	**KRAS 3**
Extracción	Original + LdG	Original + LdG	Original
Selección	Análisis de Componentes Principales	Eliminación Hacia Atrás	Selección Hacia Adelante
Clasificador	Árbol de Decisión	Análisis de Discriminante Lineal	Árbol de Decisión
Número de características	30	20	10
Características de mayor importancia	log-sigma-1-5-mm-3D_firstorder_Range	original_glrlm_GrayLevelVariance	original_shape_Flatness
	log-sigma-0-5-mm-3D_glcm_DifferenceEntropy	log-sigma-0-5-mm-3D_glszm_GrayLevelNonUniformity	original_glcm_ClusterTendency
	log-sigma-2-5-mm-3D_glcm_DifferenceEntropy	log-sigma-0-5-mm-3D_glszm_GrayLevelVariance	original_glcm_Contrast
AUC ROC - E	0,818	0,995	0,834
AUC PR - E	0,822	0,995	0,842
AUC ROC - P	0,735	0,724	0,776
AUC PR - P	0,747	0,671	0,771

Tabla 5. Descripción de los modelos que producen el mejor ensamble en KRAS. Abreviaturas: área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), AUC ROC en entrenamiento (AUC ROC - E), AUC PR en entrenamiento (AUC PR - E), AUC ROC en evaluación (AUC ROC - P), AUC PR en evaluación (AUC PR - P), Laplaciano de Gaussiano (LdG).

En la **Tabla 6** se detallan las métricas de discriminación para el ensamble de modelos de la variable KRAS, llamado KRAS Ensamble. Se presentan tanto las métricas en entrenamiento como las de evaluación, calculadas con un umbral de 0,489 obtenido con el método de la Media Geométrica. Se consiguieron resultados excelentes en entrenamiento, con el más bajo siendo el MCC en 0,886. En la evaluación, se encontró un caso similar a Óbito FVP: la mayor disminución en las métricas está en el VPP, el Valor F1 y el MCC. Comparando evaluación con entrenamiento, hubo un incremento en los errores de calibración.

	KRAS Ensamble					
	Entrenamiento			Evaluación		
Métrica	**Original**	**Bootstrap**	**IC**	**Original**	**Bootstrap**	**IC**
AUC ROC	0,996	0,996	0,984-1,000	0,905	0,895	0,762-0,983
AUC PR	0,995	0,994	0,984-1,000	0,896	0,873	0,762-0,983
Sensibilidad	0,917	0,911	0,783-1,000	0,846	0,834	0,600-1,000
Especificidad	0,968	0,968	0,897-1,000	0,824	0,823	0,632-1,000
VPP	0,957	0,955	0,850-1,000	0,786	0,768	0,533-1,000
VPN	0,938	0,937	0,849-1,000	0,875	0,874	0,688-1,000
Exactitud	0,946	0,944	0,873-1,000	0,833	0,827	0,700-0,967
Valor F1	0,936	0,931	0,849-1,000	0,815	0,793	0,571-0,952
MCC	0,889	0,886	0,755-1,000	0,665	0,649	0,346-0,929
ECE	0,218	0,224	0,190-0,259	0,176	0,222	0,145-0,311
MCE	0,536	0,543	0,357-0,623	0,602	0,520	0,273-0,612
BS	0,068	0,070	0,052-0,092	0,145	0,147	0,108-0,187

Tabla 6. Métricas del modelo KRAS Ensamble, separadas por los conjuntos de entrenamiento y evaluación. Los valores en la columna *bootstrap* corresponden a la media de cada métrica en las 1.000 iteraciones. Abreviaturas: intervalo de confianza (IC), área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), valor predictivo positivo (VPP), valor predictivo negativo (VPN), coeficiente de correlación de Matthews (MCC), error de calibración esperado (ECE), media del error de calibración (MCE), *Brier score* (BS).

En las matrices de confusión para KRAS Ensamble, mostradas en la **Figura 57**, se observa un incremento generalizado de los errores en la etapa de evaluación. Sin embargo, la caída notoria en las métricas de evaluación se observó en el VPP. Esto se debe a que el incremento en los falsos positivos es considerablemente mayor al incremento de falsos negativos.

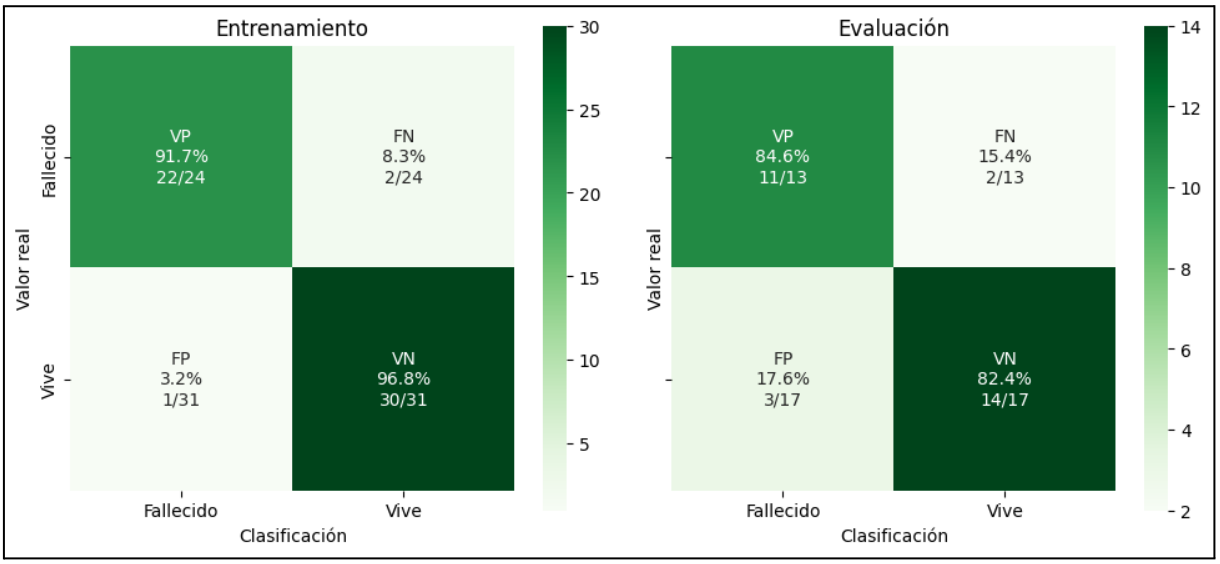

Figura 57. Matrices de confusión del modelo KRAS Ensamble. El umbral de decisión para los modelos fue 0,489. Abreviaturas: verdadero positivo (VP), falso negativo (FN), falso positivo (FP), verdadero negativo (VN).

En las curvas ROC de los modelos KRAS 1, KRAS 2, KRAS 3 y KRAS Ensamble se encontró un resultado llamativo. Tanto KRAS 2 y KRAS Ensamble tienen la misma AUC ROC en entrenamiento, de 0,966, como se muestra en la **Figura 58**. A priori esto indicaría la

redundancia de KRAS Ensamble, pero el AUC ROC en evaluación de KRAS 2 es de 0,724, mientras que el de KRAS Ensamble es de 0,905, mayor a los tres casos individuales.

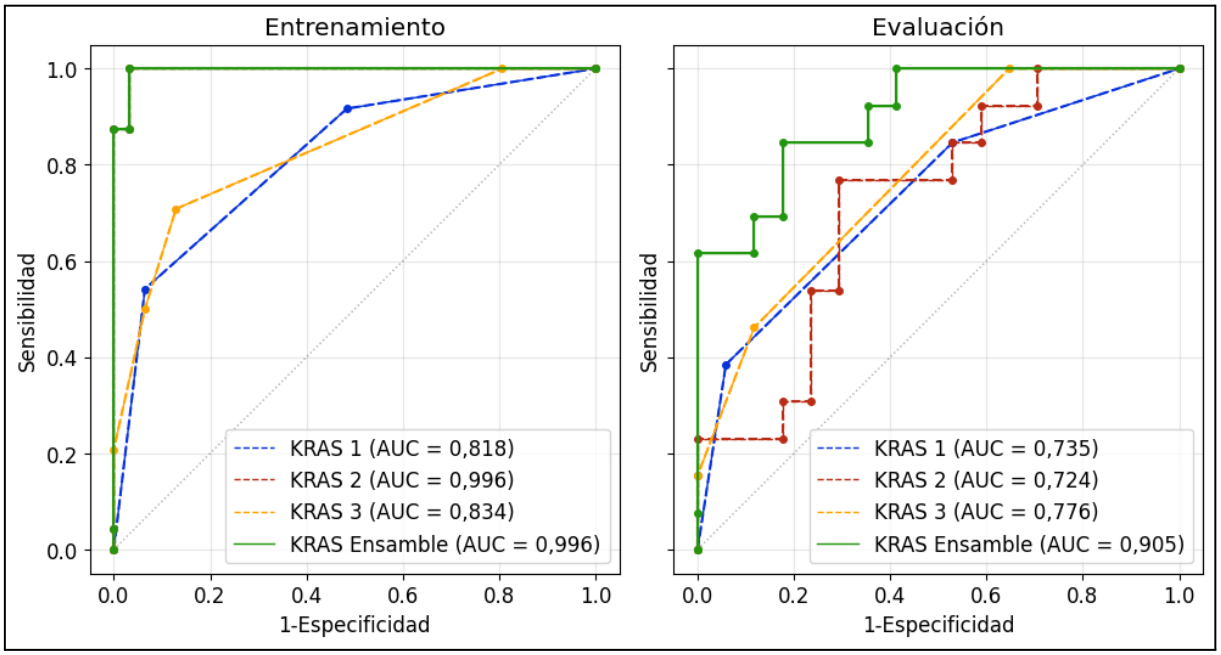

Figura 58. Curvas ROC para el ensamble de modelos de KRAS. Los modelos individuales están trazados con líneas discontinuas, mientras que el ensamble con una línea sólida. Los posibles umbrales están marcados con puntos de mayor grosor. La línea diagonal gris marca el umbral de no discriminación. KRAS 2 queda debajo del modelo KRAS Ensamble en entrenamiento por tener las curvas el mismo trazado. Abreviaturas: área bajo la curva (AUC).

En lo que respecta a las curvas PR, se observó el mismo fenómeno que en las curvas ROC. En la **Figura 59** se muestra que en todos los modelos el AUC PR fue menor para la evaluación, pero también se observó un AUC PR mayor de KRAS Ensamble respecto a los modelos individuales. Fue mayor la diferencia en el AUC PR que el AUC ROC entre los conjuntos de entrenamiento y evaluación.

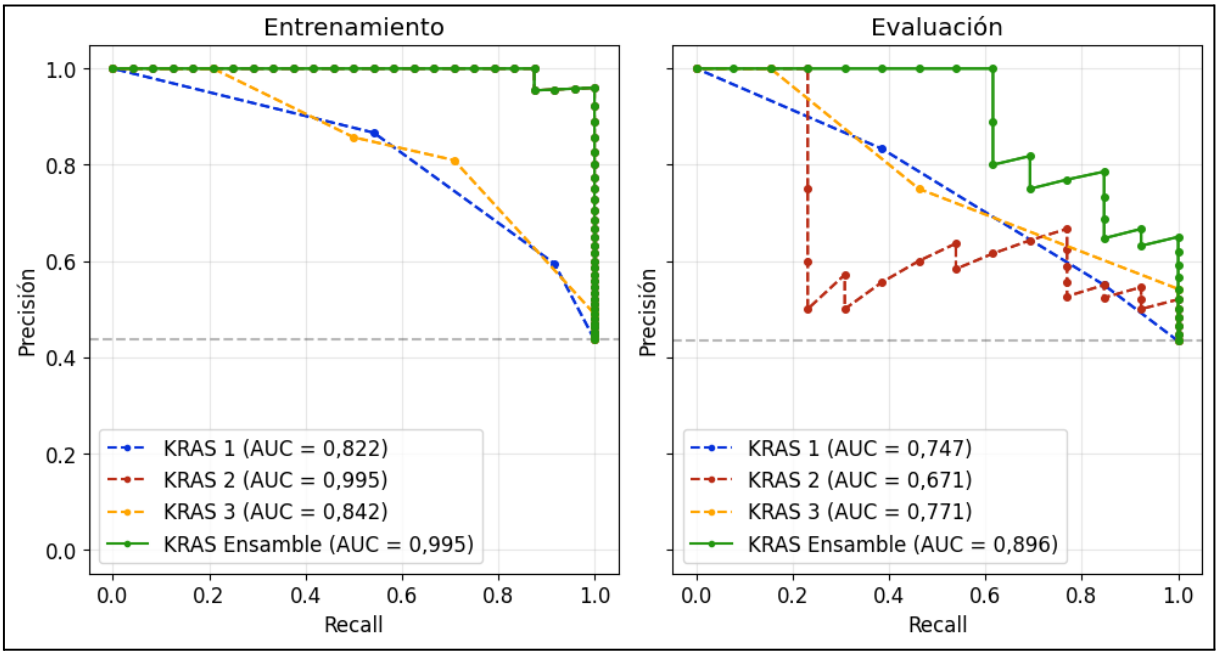

Figura 59. Curvas PR para el ensamble de modelos de KRAS. Los modelos individuales se trazaron con líneas discontinuas, mientras que el ensamble con una línea sólida. Los posibles puntos de corte se marcaron con puntos de mayor grosor. La recta gris marca el umbral de no discriminación. El modelo KRAS 2 queda debajo del modelo KRAS Ensamble en entrenamiento por tener las curvas el mismo trazado. Abreviaturas: área bajo la curva (AUC).

6. Discusión

En este trabajo se presentaron los experimentos llevados a cabo para desarrollar modelos de ML basados en radiómica para evaluar el pronóstico de MHCC. Debido a la prevalencia y mortalidad de esta enfermedad, tienen valor las herramientas que puedan aportar información sobre el estado de un paciente, hecho que está respaldado por los varios trabajos de IA realizados sobre este mismo tema. Este proyecto sería el primero en estudiar particularmente el desempeño de modelos para predecir la mortalidad específica a dos años en pacientes con MHCC luego de una cirugía de resección hepática. Respecto a la predicción del estado de mutación del gen KRAS, solo se tiene conocimiento del trabajo realizado por Dercle et al. [50].

Las métricas de discriminación para el conjunto de evaluación en los objetivos primarios alcanzaron buenos resultados. Fueron clasificados correctamente el 81,8% de los casos negativos para la variable Óbito (18 de 22 casos), y 84,6% de los casos positivos (11 de 13). Para la variable KRAS, se clasificaron correctamente el 82,4% de los casos negativos (14 de 17), y 84,6% de los casos positivos (11 de 13 casos). Además, estos resultados varían en no más de un punto porcentual al realizarse el *bootstrap*. En ambos objetivos, el punto débil se encontró en las mismas métricas, el VPP, el F1 y el MCC.

Por un lado, el VPP (o precisión) es la métrica que establece la probabilidad de que, dado un estudio con resultado positivo, el paciente efectivamente pertenezca a la clase positiva. En el caso de este trabajo, un VPP alto indicaría que, si un estudio fuera positivo, entonces el paciente probablemente fallecería en los dos años posteriores a la cirugía de tratamiento. En

los casos donde la clase positiva es de prevalencia baja, se espera que el VPP también sea bajo. Por otro lado, el valor F1 es una métrica que es alta cuando es alta tanto la sensibilidad como el VPP, y es esperable entonces que aunque se hayan logrado buenos valores de sensibilidad, el valor de F1 sea menor junto al VPP; ambas métricas tuvieron un valor menor al 0,800, pero superior a 0,700.

Por otro lado, el MCC fue de 0,649 para la variable Óbito, y 0,665 para la variable KRAS. Si bien a primera vista es un valor más bajo que otras métricas, al ser un coeficiente de correlación, su interpretación sugiere una correlación positiva entre moderada a fuerte entre la predicción y los valores reales.

Estos resultados, por sí solos, indican que los modelos entrenados podrían ser utilizados en la tarea de estimar la mortalidad específica a dos años en pacientes sometidos a cirugía de resección por MHCC como también estimar el estado de mutación del gen KRAS. Esta capacidad brindaría más información a los médicos a la hora de decidir el tipo de tratamiento más conveniente para un paciente, pero no antes de que el modelo sea validado con nuevos y más numerosos conjuntos de datos.

Existen diferencias fundamentales con otros trabajos de radiómica: la mayoría de ellos trabajaron en modelos de regresión en análisis de supervivencia sin el uso de IA (emplearon métodos estadísticos clásicos), mientras que en este proyecto se trabajó en modelos de clasificación binaria con ML. A pesar de estas diferencias, gracias a que algunos de esos trabajos han presentado métricas de discriminación, es posible una comparación.

Ganeshan et al. [41] alcanzaron un AUC ROC de 0,609 para la clasificación de mortalidad de pacientes de cirugía de resección de CCR, en imágenes tomadas posteriormente a la cirugía. Dercle et al. [50] alcanzaron un AUC ROC de 0,800 (IC 0,690-0,940), sensibilidad de 0,800 y especificidad de 0,780 en su conjunto de validación, utilizando una señal compuesta de características radiómicas para detectar la respuesta positiva al tratamiento de quimioterapia para CCR Utilizando esta señal alcanzaron un modelo predictivo del estado de mutación KRAS con un AUC ROC de 0,630 (IC: 0,580-0,670). Finalmente, Liang et al. [51] utilizaron características radiómicas de imágenes de TC y RM para predecir la aparición de MHCC mediante ML, obteniendo un AUC ROC de 0,870 (IC 0,730 - 0,880), sensibilidad de 0,830 y especificidad de 0,760. Los resultados obtenidos en el presente trabajo para la variable Óbito son entonces ligeramente superiores a aquellos encontrados en la bibliografía, y mejores en comparación al único trabajo en la predicción de la variable KRAS encontrado. Sin embargo, hay otras comparaciones que hacer en términos de metodología, conformación de bases de datos, y las conclusiones alcanzadas.

Como punto de partida, el número de pacientes en la base de datos construida para este proyecto es comparable a aquel utilizado en trabajos similares. Se contó con 136 pacientes para el estudio de Óbito en FVP, cuando en pocos trabajos se consiguieron conjuntos que superaran los 100 pacientes. En Dercle et al. [50] se trabajó con 129, en Simpson et al. [48] con 198, y en Liang et al. [51] con 108 casos. El número de pacientes para el estudio de KRAS fue menor, con 85 casos, pero también comparable con otros trabajos.

Los estudios de Óbito en FSC, FVT y FA contaron con menos casos (74, 61 y 41 casos totales), y si bien la diferencia con la cantidad de casos en fase portal no es muy grande, esto tiene un efecto en la capacidad de aprendizaje y generalización de los modelos. Como el foco de la investigación de MHCC y CCR ha estado sobre la FVP, tampoco existe suficiente material publicado con el cual comparar, y cualquier resultado o métrica obtenido para estas imágenes debería ser analizado con especial precaución.

Si bien el número de casos está en el rango de los otros trabajos relevados, puede ser bajo para un desarrollo de IA. Asumiendo que se tienen muestras de calidad y representativas de la población real, cuantas más muestras tiene el algoritmo, más oportunidades tiene para aprender, y se pueden utilizar más características en simultáneo sin incrementar el riesgo de un sobreajuste. Al tratarse de un proyecto de radiómica, con la cual se pueden extraer cientos de características para una sola muestra, el tamaño de la base de datos puede considerarse como una limitación.

Otro problema a analizar es el desbalance de las clases en la base de datos. En este trabajo, la proporción entre las clases positiva y negativa a lo largo de las fases y variables de respuesta no fue equitativa. La fase mejor balanceada fue la FVT, con un 46% de casos positivos y 54% de casos negativos, mientras que la menos balanceada fue la FVP para Óbito, con 37% de casos positivos y 63% negativos. El desbalance puede ser problemático en el entrenamiento de IA, ya que el algoritmo aprenderá con más ejemplos de una clase que de otra. Esto está vinculado a la cantidad de datos, ya que con un número suficientemente grande para ambas clases, el algoritmo tiene amplia cantidad de ejemplos de ambas clases, y el desbalance deja de ser un problema. Aún así, se lograron modelos con buenos resultados en la clasificación de ambas clases, lo que indica que las estrategias empleadas a lo largo del proceso superaron esta problemática.

Respecto a la extracción de características, se encontró que en los experimentos realizados para todas las fases y variables, los mejores modelos siempre se consiguieron al entrenar con características originadas de la extracción Original+LdG, con solo FVP 2 usando características exclusivamente obtenidas de LdG, y KRAS 3 usando sólo las características de Original. Este resultado coincide con la preferencia por el filtro LdG para la extracción de características en imágenes, expresado en varios trabajos, como Lubner et al. [47] y Simpson et al. [48]. Esto no necesariamente quiere decir que no haya utilidad en el uso de los filtros exponencial, logarítmico, cuadrado o raíz cuadrada. Los modelos en los cuales se hizo la selección de características y entrenamiento utilizando sólo los conjuntos E+L y C+RC no produjeron los mejores resultados, pero es posible que algunas características dentro de estos conjuntos, combinadas con otras características en los conjuntos Original, LdG u Original+LdG puedan generar modelos incluso mejores. Esto requeriría combinar todos los conjuntos de características en la selección, que requeriría de mayores recursos computacionales. Una opción diferente, y que podría probarse en un trabajo futuro, es la selección de las mejores características en cada uno de los conjuntos y su posterior combinación en un nuevo grupo de características para hacer la selección final, similar al proceso empleado por Liang et al. [51].

Respecto a los métodos de selección de características, sería útil encontrar aquellos que han

llevado a los mejores resultados para su uso en futuros trabajos. Dos métodos que no parecen prometedores son la Correlación de Pearson y LinearSVC, ya que ninguno de los experimentos en los que fueron usados tuvieron buen desempeño. Para los demás, no sería apropiado simplemente considerar como mejor al método de selección más frecuente entre los mejores modelos, ya que se tienen pocos ejemplos, pero hay otros factores que se pueden analizar.

Uno de estos factores es la interpretabilidad del clasificador. En particular, el PCA añade un nivel de complejidad en este aspecto, ya que genera nuevas características a partir de las originales. Aun cuando es posible identificar a las características que tuvieron mayor peso en la transformación, es un componente más que debe ser interpretado para comprender el funcionamiento del clasificador y las características de la imagen que llevan a la decisión del mismo, un aspecto que es fundamental en la aceptación de herramientas de soporte para la toma de decisiones en el ámbito médico. Esta propiedad, junto al hecho de que PCA fue utilizado por sólo uno de los mejores modelos, hace que este método sea poco deseable para futuras pruebas.

Por otro lado, está la naturaleza de los métodos de selección: FS y BE son métodos de envoltura, y las características que estos eligen suelen ser útiles específicamente para el algoritmo con el cual fueron usados, mientras que las características obtenidas mediante ETC y LASSO, ambos métodos embebidos, tienden a funcionar mejor con los algoritmos específicos que estos métodos usan (RF y RL, respectivamente). Tanto los modelos que usaron los métodos de envoltura (la mayoría de ellos) como aquellos que usaron los métodos embebidos utilizaron uno de dos algoritmos de ML: DT y LDA. Debido a esto, en un trabajo futuro podría ser relevante utilizar sólo los métodos de envoltura y ajustar los hiperparámetros de la selección para elegir características que sean incluso de mayor utilidad para el entrenamiento del clasificador.

Varios de los estudios presentados en la Sección 3. reportaron una correlación positiva entre las características de entropía y contraste con la supervivencia en general, mientras que la uniformidad tuvo una correlación negativa. En los mejores modelos obtenidos en este proyecto, estas características no fueron utilizadas, pero se hizo uso de otras que describen aspectos similares de las imágenes. En la extracción Original+LdG se incluyeron 358 características, de las cuales solo 90 fueron usadas en los mejores modelos. De ellas, 12 fueron usadas por al menos dos modelos:

- *log-sigma-0-5-mm-3D_gldm_DependenceEntropy*
- *log-sigma-1-5-mm-3D_glcm_MaximumProbability*
- *original_shape_Sphericity*
- *original_shape_Flatness*
- *log-sigma-2-5-mm-3D_glcm_Imc1*
- *original_glszm_LargeAreaLowGrayLevelEmphasis*
- *log-sigma-1-5-mm-3D_glszm_SmallAreaLowGrayLevelEmphasis*
- *log-sigma-1-5-mm-3D_glszm_LargeAreaLowGrayLevelEmphasis*
- *log-sigma-0-5-mm-3D_glszm_LargeAreaEmphasis*
- *log-sigma-0-5-mm-3D_glszm_ZoneVariance*
- *log-sigma-2-5-mm-3D_glszm_ZoneVariance*

- *log-sigma-1-5-mm-3D_firstorder_InterquartileRange*

Entre estas características, sólo *original_shape_Flatness* fue usada en cuatro oportunidades, en los modelos FVP 2, FSC, KRAS 1 y KRAS 2. Además, solo *original_shape_Sphericity* y *original_shape_Flatness* aparecen dos veces entre las tres características de mayor importancia en un modelo, lo cual podría indicar que la forma de la metástasis puede estar relacionada a ambas variables de respuesta. De las 90 características, 64 provienen del conjunto LdG. Se utilizaron 15 características de primer orden, cuatro de forma, 27 de GLCM, 12 de GLDM, 21 de GLSZM, 11 de GLRLM. Estos números no dan apoyo para encontrar un tipo de característica de mayor importancia, ya que si bien solo se usaron cuatro características de forma, *original_shape_Sphericity* y *original_shape_Flatness* están entre las de mayor importancia en sus modelos. La heterogeneidad en las características seleccionadas puede originarse en los múltiples métodos de filtrado, selección de características y algoritmos de clasificación. Otro aspecto que podría ser relevante es la configuración de la extracción de características. Si bien se utilizaron diferentes filtros, y con diferentes valores en el caso del LdG, no se utilizaron otras técnicas como la normalización de la imagen antes de la extracción, o el remuestreo e interpolado de la imagen o la máscara.

Los aspectos antes mencionados dan cuenta de todas las variables involucradas en este trabajo, y de otros posibles experimentos que se podrían haber realizado. En cada uno de los pasos del desarrollo se encuentra la posibilidad de ajustar una variable y cambiar las condiciones de trabajo en pasos posteriores o los resultados finales. No sería posible explorar todo el universo de combinaciones posibles de estas variables por la necesidad de recursos y tiempo, y la dificultad en este tipo de trabajos está en encontrar a qué aspectos del entrenamiento dedicar estos recursos y tiempo. Sin embargo, los resultados alcanzados dan una base sobre la cual se podrían fijar ciertas variables y ajustar otras (procesamiento de imágenes para la extracción de características, mejores combinaciones de características, exploración profunda de hiperparámetros con un único algoritmo) para refinar el desempeño de los modelos generados. Además, hay otras limitaciones que impulsan a continuar trabajando.

Debido a que no todos los pacientes tenían un estudio confirmatorio del estado de mutación del gen KRAS, los conjuntos para cada variable no fueron coincidentes respecto a qué casos se usaron para entrenar y evaluar los algoritmos, y esto impidió hacer comparaciones profundas entre los modelos generados, sus resultados e interpretación.

Los resultados del trabajo, si bien son prometedores, requieren su validación en un estudio prospectivo, e incluso multicéntrico de ser posible, ya que es propio de los estudios retrospectivos y confinados a una población reducida contener algún tipo de sesgo que compromete la capacidad de generalización de los modelos. Si bien los criterios de inclusión y exclusión son específicos a la patología estudiada, al ser un estudio limitado a la población de un único hospital,existe un sesgo de selección que solo puede resolverse mediante una posterior validación externa.

No se considera que haya un significativo sesgo de clasificación para la variable Óbito, ya que la causa de muerte para cada paciente estaba debidamente informada en su historia clínica electrónica. El valor de KRAS, por otro lado, se obtuvo de informes del servicio de oncología. Si

bien el estudio de laboratorio que confirma la mutación es de muy alta capacidad diagnóstica, al no utilizar los mecanismos de la historia clínica electrónica del hospital, hay más posibilidades de incurrir en un error humano en la carga de datos.

El sesgo más notorio en este proyecto es el de confusión, específicamente para la variable Óbito. En este objetivo, se busca hacer una clasificación de un resultado que ocurrirá más de dos años después de tomar la imagen que es el dato principal para el modelo. Aunque en este proyecto se hayan conseguido buenos resultados en este objetivo, no se ignora que la supervivencia del paciente y la reincidencia de la enfermedad casi con seguridad están afectados por variables que no están contempladas ni controladas en el mismo. Es por este sesgo que es fundamental validar los modelos en otras instituciones, en mayores poblaciones, y en un estudio prospectivo con un seguimiento más controlado, para garantizar que este problema de diseño no tiene un efecto significativamente negativo en el desempeño de los modelos.

Una mejora que podría incluirse en próximos estudios es la incorporación de variables clínicas. Aunque en este proyecto se haya trabajado con ML en radiómica, nada impide incluir ese tipo de información ya sea en el entrenamiento del modelo o en otra herramienta que tome como entrada tanto los datos clínicos como la salida del clasificador. Las variables clínicas han sido utilizadas por otros trabajos descritos en la Sección 3. Esto no solo podría mejorar el desempeño de los modelos, sino que al incluir biomarcadores conocidos por los expertos en salud se puede mejorar la interpretabilidad de la herramienta. Es difícil explicar que un modelo considera un caso positivo en base a una característica radiómica de textura compleja, pero no tanto si lo hace también porque el gen KRAS ha mutado.

Algo que otros trabajos consideraron fue la separación de los pacientes según el tratamiento que cada uno recibió. Si los tratamientos tienen efectos notables y diferenciables desde el punto de vista radiómico, entonces el desempeño de los modelos también depende del tratamiento utilizado. Como fue mencionado, si bien el conjunto de datos con el cual se trabajó es grande en comparación al reportado en otras publicaciones, no es lo suficientemente grande en el contexto de la IA (como para, por ejemplo, aplicar técnicas de DL), y fragmentarlo aún más llevaría a problemas como el sobreajuste. Una base de datos más grande permitiría hacer esta consideración, como también tener resultados más confiables para la FA, la FVT y la FSC.

Finalmente, este trabajo estuvo restringido al uso de imágenes de TC y segmentación de un único VOI del hígado (la lesión hepática dominante). Otros trabajos han utilizado imágenes de otros órganos como el colon y el recto. Liang et al. [51] encontraron que utilizar múltiples modalidades de imagen produjo mejores resultados en comparación a modelos que utilizaron una única modalidad. Finalmente, el uso de múltiples lesiones por paciente podría ofrecer más información. Cabe destacar que cada mejora propuesta para un trabajo futuro llevaría a la complejización del modelo, dificultando su interpretabilidad y su usabilidad, especialmente si se requiere de múltiples estudios adicionales a los que normalmente se sometería el paciente.

7. Conclusiones

En este proyecto se presentó el desarrollo y evaluación de dos modelos basados en ML y radiómica para predecir resultados en pacientes con MHCC. Puntualmente, un modelo para la predicción de mortalidad específica a dos años luego de realizarse una cirugía de resección hepática y un modelo para predecir el estado de mutación del gen KRAS. Ambos modelos utilizaron imágenes de TC abdominal en FVP tomadas como parte del control del paciente y antes de realizarse la cirugía de tratamiento. Se entrenaron y evaluaron también modelos para la predicción de mortalidad específica utilizando imágenes en FA, FVT y FSC.

La IA puede constituir una herramienta para los profesionales de salud en el tratamiento de la MHCC, como ya lo hace en otras aplicaciones en medicina. La prevalencia y mortalidad de esta y otras enfermedades vinculadas ha llevado a la realización de múltiples desarrollos en radiómica e IA abordando la problemática desde diferentes puntos de vista. En particular, los modelos desarrollados tienen un objetivo claro en brindar información para asistir en la toma de decisiones sobre el tratamiento de los pacientes, utilizando imágenes médicas que no requieren de la realización de un nuevo estudio para estas personas.

El desempeño alcanzado por los modelos entrenados en los objetivos primarios fue prometedor en los conjuntos de evaluación. Esto impulsa a continuar el trabajo en el desarrollo de modelos más refinados y en validaciones clínicas que permitan confirmar la capacidad de generalización de estas herramientas y su posible implementación en el flujo clínico.

8. Referencias bibliográficas

[1] Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021, Febrero 4). Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA: A Cancer Journal for Clinicians*, *71*(3), 209–249. https://doi.org/10.3322/caac.21660.

[2] Hossain, M. S., Karuniawati, H., Jairoun, A. A., Urbi, Z., Ooi, D. J., John, A., Lim, Y. C., Kibria, K. M. K., Mohiuddin, A. M., Ming, L. C., Goh, K. W., & Hadi, M. A. (2022, March 29). Colorectal Cancer: A Review of Carcinogenesis, Global Epidemiology, Current Challenges, Risk Factors, Preventive and Treatment Strategies. *Cancers*, *14*(7), 1732. https://doi.org/10.3390/cancers14071732.

[3] Agency for Research on Cancer (IARC), T. I. (n.d.). *Global Cancer Observatory*. Global Cancer Observatory. https://gco.iarc.fr/.

[4] Estadísticas - Mortalidad. (2019, Febrero 17). Argentina.gob.ar. https://www.argentina.gob.ar/salud/instituto-nacional-del-cancer/estadisticas/mortalidad.

[5] Ferlay, J., Soerjomataram, I., Dikshit, R., Eser, S., Mathers, C., Rebelo, M., Parkin, D. M., Forman, D., & Bray, F. (2014, Octubre 9). Cancer incidence and mortality worldwide: Sources, methods and major patterns in GLOBOCAN 2012. *International Journal of Cancer*, *136*(5), E359–E386. https://doi.org/10.1002/ijc.29210.

[6] Martin, J., Petrillo, A., Smyth, E. C., Shaida, N., Khwaja, S., Cheow, H., Duckworth, A.,

Heister, P., Praseedom, R., Jah, A., Balakrishnan, A., Harper, S., Liau, S., Kosmoliaptsis, V., & Huguet, E. (2020, Octubre 24). Colorectal liver metastases: Current management and future perspectives. *World Journal of Clinical Oncology*, *11*(10), 761–808. https://doi.org/10.5306/wjco.v11.i10.761.

[7] Zaharia, C., Veen, T., Lea, D., Kanani, A., Alexeeva, M., & Søreide, K. (2022, Diciembre 28). Histopathological Growth Pattern in Colorectal Liver Metastasis and The Tumor Immune Microenvironment. MDPI. https://doi.org/10.3390/cancers15010181.

[8] *Plano anatómico - Wikipedia, la enciclopedia libre*. (n.d.). Plano Anatómico - Wikipedia, La Enciclopedia Libre. https://es.wikipedia.org/wiki/Plano_anat%C3%B3mico.

[9] Quintero, Juana & Martín-Landrove, Miguel. (2011). Determinación in vitro del Índice de Conformidad para la Evaluación de Planes de Tratamiento 3D en Radiocirugía Estereotáctica Intracraneal. https://doi.org/10.13140/RG.2.2.31848.39686.

[10] Bidgood, W. D., Horii, S. C., Prior, F. W., & Van Syckle, D. E. (1997, Mayo 1). Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging. *Journal of the American Medical Informatics Association*, *4*(3), 199–212. https://doi.org/10.1136/jamia.1997.0040199.

[11] *NIfTI: Neuroimaging Informatics Technology Initiative.* https://nifti.nimh.nih.gov/.

[12]: Acharya, T. & Ray, A. (2005). *Image Processing: Principles and Applications.* Wiley-Interscience.

[13] Mosquera, C., Ricci Lara, M. & Díaz, F. (2021). *Inteligencia Artificial en Imágenes Médicas. De la teoría a la aplicación*. Magister.

[14] Zwanenburg, A., Vallières, M., Abdalah, M. A., Aerts, H. J. W. L., Andrearczyk, V., Apte, A., Ashrafinia, S., Bakas, S., Beukinga, R. J., Boellaard, R., Bogowicz, M., Boldrini, L., Buvat, I., Cook, G. J. R., Davatzikos, C., Depeursinge, A., Desseroit, M. C., Dinapoli, N., Dinh, C. V., . . . Löck, S. (2020, Mayo). The Image Biomarker Standardization Initiative: Standardized Quantitative Radiomics for High-Throughput Image-based Phenotyping. *Radiology*, *295*(2), 328–338. https://doi.org/10.1148/radiol.2020191145.

[15] van Griethuysen, J. J. M., Fedorov, A., Parmar, C., Hosny, A., Aucoin, N., Narayan, V., Beets-Tan, R. G. H., Fillion-Robin, J. C., Pieper, S., Aerts, H. J. W. L. (2017). Computational Radiomics System to Decode the Radiographic Phenotype. Cancer Research, 77(21), e104–e107. V 3.0.1. https://doi.org/10.1158/0008-5472.CAN-17-0339.

[16] *Neuroimaging in Python — NiBabel 5.1.0 documentation*. (n.d.). Neuroimaging in Python — NiBabel 5.1.0 Documentation. https://nipy.org/nibabel/coordinate_systems.html.

[17] Zwanenburg, A., Vallières, M., Abdalah, M. A., Aerts, H. J. W. L., Andrearczyk, V., Apte, A., Ashrafinia, S., Bakas, S., Beukinga, R. J., Boellaard, R., Bogowicz, M., Boldrini, L., Buvat, I., Cook, G. J. R., Davatzikos, C., Depeursinge, A., Desseroit, M. C., Dinapoli, N., Dinh, C. V., . . . Löck, S. (2020, Mayo). The Image Biomarker Standardization Initiative: Standardized Quantitative Radiomics for High-Throughput Image-based Phenotyping. *Radiology*, *295*(2), 328–338. https://doi.org/10.1148/radiol.2020191145.

[18] de Queiroz Neto, J. F., Santos, E. M. D., & Vidal, C. A. (2016, Octubre). MSKDE - Using Marching Squares to Quickly Make High Quality Crime Hotspot Maps. *2016 29th SIBGRAPI*

Conference on Graphics, Patterns and Images (SIBGRAPI). https://doi.org/10.1109/sibgrapi.2016.049.

[19] Maple, C. (2003, Julio). Geometric design and space planning using the marching squares and marching cube algorithms. *2003 International Conference on Geometric Modeling and Graphics, 2003. Proceedings*. https://doi.org/10.1109/gmag.2003.1219671 Copyright © 2003, IEEE.

[20] Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973, Noviembre). Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-3*(6), 610–621. https://doi.org/10.1109/tsmc.1973.4309314.

[21] Löfstedt, T., Brynolfsson, P., Asklund, T., Nyholm, T., & Garpebring, A. (2019, Febrero 22). Gray-level invariant Haralick texture features. *PLOS ONE*, *14*(2), e0212110. https://doi.org/10.1371/journal.pone.0212110.

[22] Russel & Norvig (2003). *Artificial Intelligence, A Modern Approach.* Pearson Education, Inc.

[23] Samuel, A.L. (1967). Some Studies in Machine Learning Using the Game of Checkers. *IBM J. Res. Dev., 44*, 206-227.

[24] Mitchell. (1997, Marzo 1). *Machine Learning*. McGraw Hill.

[25] Raschka, S. & Mirjalili, V. (2019). *Aprendizaje Automático y aprendizaje profundo con Python, scikit-learning y TensorFlow*. Marcombo.

[26] Epidemiological Data from the nCoV-2019 Outbreak: Early Descriptions from Publicly Available Data. (2020, Enero 23). Virological. https://virological.org/t/epidemiological-data-from-the-ncov-2019-outbreak-early-descriptions-from-publicly-available-data/337

[27] Melkumova, L., & Shatskikh, S. (2017). Comparing Ridge and LASSO estimators for data analysis. Procedia Engineering, 201, 746–755. https://doi.org/10.1016/j.proeng.2017.09.615.

[28] Cerda & Cifuentes (2012). Uso de curvas ROC en investigación clínica. Aspectos teórico-prácticos. *Revista chilena de infectología.* Volumen 29 (Número 2) http://dx.doi.org/10.4067/S0716-10182012000200003.

[29] Saito, T., & Rehmsmeier, M. (2015, Marzo 4). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. PLOS ONE, 10(3), e0118432. https://doi.org/10.1371/journal.pone.0118432.

[30] Mukhoti, J. et al. (2020). Calibrating Deep Neural Networks using Focal Loss. *arXiv*, 2020 https://doi.org/10.48550/arXiv.2002.09437.

[31] Pakdaman Naeini, M., Cooper, G., & Hauskrecht, M. (2015, February 21). Obtaining Well Calibrated Probabilities Using Bayesian Binning. Proceedings of the AAAI Conference on Artificial Intelligence, 29(1). https://doi.org/10.1609/aaai.v29i1.9602.

[32] Gawlikowski, J. (2021). A Survey of Uncertainty in Deep Neural Networks. *arXiv*. https://doi.org/10.48550/arXiv.2107.03342.

[33] Rosner, Bernard. *Fundamentals of Biostatistics*. Brooks/Cole, 2010.

[34] Newman, T. B., Hulley, S. B., Cummings, S. R., Browner, W. S., & Grady, D. G. (2013, July 2). Designing Clinical Research.

[35] Chandler, MIT News Office, D. L. (2012, Febrero 9). Explained: Sigma. MIT News | Massachusetts Institute of Technology. https://news.mit.edu/2012/explained-sigma-0209.

[36] Artificial Intelligence in Medical Imaging. (2019, February 7). In E. R. Ranschaert, S. Morozov, & P. R. Algra (Eds.), Opportunities, Applications and Risks. Springer. https://doi.org/10.1007/978-3-319-94878-2.

[37] Rebasa, P. (2005, Octubre). Conceptos básicos del análisis de supervivencia. *Cirugía Española*, 78(4), 222–230. https://doi.org/10.1016/s0009-739x(05)70923-4.

[38] "Diccionario De Cáncer Del NCI." *Instituto Nacional Del Cáncer*, www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer.

[39] Miles, Kenneth A., et al. "Colorectal Cancer: Texture Analysis of Portal Phase Hepatic CT Images as a Potential Marker of Survival." *Radiology*, vol. 250, no. 2, Radiological Society of North America (RSNA), Feb. 2009, pp. 444–52. *Crossref*, https://doi.org/10.1148/radiol.2502071879.

[40] Rao, S. X., Lambregts, D. M., Schnerr, R. S., van Ommen, W., van Nijnatten, T. J., Martens, M. H., Heijnen, L. A., Backes, W. H., Verhoef, C., Zeng, M. S., Beets, G. L., & Beets-Tan, R. G. (2014, Diciembre). Whole-liver CT texture analysis in colorectal cancer: Does the presence of liver metastases affect the texture of the remaining liver? United European Gastroenterology Journal, 2(6), 530–538. https://doi.org/10.1177/2050640614552463.

[41] Ganeshan, B., Miles, K. A., Young, R. C., & Chatwin, C. R. (2007, December). Hepatic Enhancement in Colorectal Cancer. Academic Radiology, 14(12), 1520–1530. https://doi.org/10.1016/j.acra.2007.06.028.

[42] Rao, S., Lambregts, D. M., Schnerr, R. S., Beckers, R. C., Maas, M., Albarello, F., Riedl, R. G., Dejong, C. H., Martens, M. H., Heijnen, L. A., Backes, W. H., Beets, G. L., Zeng, M., & Beets-Tan, R. G. (2016, Abril). CT texture analysis in colorectal liver metastases: A better way than size and volume measurements to assess response to chemotherapy? United European Gastroenterology Journal, 4(2), 257–263. https://doi.org/10.1177/2050640615601603.

[43] Cervera Deval, J. (2014, Mayo). RECIST y el radiólogo. Radiología, 56(3), 193–205. https://doi.org/10.1016/j.rx.2012.03.010.

[44] Dohan, A., Gallix, B., Guiu, B., Le Malicot, K., Reinhold, C., Soyer, P., Bennouna, J., Ghiringhelli, F., Barbier, E., Boige, V., Taieb, J., Bouché, O., François, E., Phelip, J. M., Borel, C., Faroux, R., Seitz, J. F., Jacquot, S., Ben Abdelghani, M., . . . Hoeffel, C. (2019, May 17). Early evaluation using a radiomic signature of unresectable hepatic metastases to predict outcome in patients with colorectal cancer treated with FOLFIRI and bevacizumab. Gut, 69(3), 531–539. https://doi.org/10.1136/gutjnl-2018-316407.

[45] Ravanelli, M., Agazzi, G. M., Tononcelli, E., Roca, E., Cabassa, P., Baiocchi, G., Berruti, A., Maroldi, R., & Farina, D. (2019, Junio 6). Texture features of colorectal liver metastases on pretreatment contrast-enhanced CT may predict response and prognosis in patients treated with bevacizumab-containing chemotherapy: a pilot study including comparison with standard chemotherapy. La Radiologia Medica, 124(9), 877–886. https://doi.org/10.1007/s11547-019-01046-4.

[46] Bang, J. I., Ha, S., Kang, S. B., Lee, K. W., Lee, H. S., Kim, J. S., Oh, H. K., Lee, H. Y., & Kim, S. E. (2015, Septiembre 4). Prediction of neoadjuvant radiation chemotherapy response

and survival using pretreatment [18F]FDG PET/CT scans in locally advanced rectal cancer. European Journal of Nuclear Medicine and Molecular Imaging, 43(3), 422–431. https://doi.org/10.1007/s00259-015-3180-9.

[47] Lubner, M. G., Stabo, N., Lubner, S. J., del Rio, A. M., Song, C., Halberg, R. B., & Pickhardt, P. J. (2015, Mayo 13). CT textural analysis of hepatic metastatic colorectal cancer: pre-treatment tumor heterogeneity correlates with pathology and clinical outcomes. Abdominal Imaging, 40(7), 2331–2337. https://doi.org/10.1007/s00261-015-0438-4.

[48] Simpson, A. L., Doussot, A., Creasy, J. M., Adams, L. B., Allen, P. J., DeMatteo, R. P., Gönen, M., Kemeny, N. E., Kingham, T. P., Shia, J., Jarnagin, W. R., Do, R. K. G., & D'Angelica, M. I. (2017, Mayo 30). Computed Tomography Image Texture: A Noninvasive Prognostic Marker of Hepatic Recurrence After Hepatectomy for Metastatic Colorectal Cancer. Annals of Surgical Oncology, 24(9), 2482–2490. https://doi.org/10.1245/s10434-017-5896-1

[49] Badic, B., Desseroit, M. C., Hatt, M., & Visvikis, D. (2019, April). Potential Complementary Value of Noncontrast and Contrast Enhanced CT Radiomics in Colorectal Cancers. Academic Radiology, 26(4), 469–479. https://doi.org/10.1016/j.acra.2018.06.004

[50] Dercle, L., Lu, L., Schwartz, L. H., Qian, M., Tejpar, S., Eggleton, P., Zhao, B., & Piessevaux, H. (2020, Febrero 4). Radiomics Response Signature for Identification of Metastatic Colorectal Cancer Sensitive to Therapies Targeting EGFR Pathway. JNCI: Journal of the National Cancer Institute, 112(9), 902–912. https://doi.org/10.1093/jnci/djaa017.

[51] Liang, M., Cai, Z., Zhang, H., Huang, C., Meng, Y., Zhao, L., Li, D., Ma, X., & Zhao, X. (2019, Noviembre). Machine Learning-based Analysis of Rectal Cancer MRI Radiomics for Prediction of Metachronous Liver Metastasis. Academic Radiology, 26(11), 1495–1504. https://doi.org/10.1016/j.acra.2018.12.019.

[52] Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.

[53] Bisong, E. (2019). Google Colaboratory. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, 59–64. https://doi.org/10.1007/978-1-4842-4470-8_7.

[54] Plazzotta, F., & Luna, D. (2015). Health information systems: integrating clinical data in different scenarios and users. Revista peruana de medicina experimental y salud pública, 32(2), 343-351.

[55] Soriano E, Plazzotta F, Campos F, Kaminker D, Cancio A, Aguilera Díaz J, Luna D, Seehaus A, Carcía Mónaco R, de Quirós FG. Integration of healthcare information: from enterprise PACS to patient centered multimedia health record. Stud Health Technol Inform. 2010;160(Pt 1):126-30. PMID: 20841663.

[56] Kikinis, Ron, et al. "3D Slicer: A Platform for Subject-Specific Image Analysis, Visualization, and Clinical Support." *Intraoperative Imaging and Image-Guided Therapy*, Springer New York, Nov. 2013, pp. 277–89. *Crossref*, https://doi.org/10.1007/978-1-4614-7657-3_19.

[57] Brett, M. et al. nipy/nibabel: 3.1.1. 2020, Zenodo. https://doi.org/10.5281/zenodo.3924343.

[58] Bradski G. The OpenCV Library. Dr Dobbs Journal of Software Tools, 2000.

[59] Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., Gramfort, A., Thirion, B., & Varoquaux, G. (2014). Machine learning for neuroimaging with scikit-learn.

Frontiers in Neuroinformatics, 8. https://doi.org/10.3389/fninf.2014.00014.

[60] Reback, J. (2021). Pandas-Dev/Pandas, Pandas 1.3.1, Zenodo [code], https://doi.org/10.5281/zenodo.4524629.

[61] Fisher, R. A. (1936, Septiembre). The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7(2), 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

[62] Raschka, S. (2018, Abril 22). MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. Journal of Open Source Software, 3(24), 638. https://doi.org/10.21105/joss.00638.

[63] Chen, T., & Guestrin, C. (2016, Agosto 13). XGBoost. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/2939672.2939785.

[64] Manjrekar, O. N., & Dudukovic, M. P. (2019, Mayo). Identification of flow regime in a bubble column reactor with a combination of optical probe data and machine learning technique. *Chemical Engineering Science: X*, *2*, 100023. https://doi.org/10.1016/j.cesx.2019.100023.

[65] Sheykhmousa, M., Mahdianpari, M., Ghanbari, H., Mohammadimanesh, F., Ghamisi, P., & Homayouni, S. (2020). Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 6308–6325. https://doi.org/10.1109/jstars.2020.3026724.

[66] Raizada, R. D. S., & Lee, Y. S. (2013, Julio 26). Smoothness without Smoothing: Why Gaussian Naive Bayes Is Not Naive for Multi-Subject Searchlight Studies. *PLoS ONE*, *8*(7), e69566. https://doi.org/10.1371/journal.pone.0069566.

[67] Deng, H., Zhou, Y., Wang, L., & Zhang, C. (2021, Diciembre). Ensemble learning for the early prediction of neonatal jaundice with genetic features. *BMC Medical Informatics and Decision Making*, *21*(1). https://doi.org/10.1186/s12911-021-01701-9.

[68] Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/mcse.2007.55

[69] Berrar, D. (2019). *Performance Measures for Binary Classification. In Encyclopedia of Bioinformatics and Computational Biology (pp. 546–560)*. Elsevier. https://doi.org/10.1016/b978-0-12-809633-8.20351-8.

[70] Unal, I. (2017). Defining an Optimal Cut-Point Value in ROC Analysis: An Alternative Approach. Computational and Mathematical Methods in Medicine, 2017, 1–14. https://doi.org/10.1155/2017/3762651.

[71] Liu, X. (2012, Febrero 3). Classification accuracy and cut point selection. *Statistics in Medicine*, *31*(23), 2676–2686. https://doi.org/10.1002/sim.4509.

[72] Brier, G. (1950) Verification of forecasts expressed in terms of probability. *Monthly weather review*, p 1-3. https://doi.org/10.1175/1520-0493(1950)078.

Anexos

A. Población del estudio

En total, se recolectaron 138 pacientes para el estudio. La cantidad de pacientes con TC en cada una de las fases, y con variable Óbito o KRAS confirmada se muestra en la **Figura A1**. La cantidad de casos hallados en cada fase fue uno de los motivadores para ajustar los objetivos primarios y secundarios, ya que evaluar un modelo utilizando un número bajo de imágenes no permite hablar de la capacidad de generalización del mismo. Los objetivos secundarios se exploraron sólo para la variable Óbito debido a que el número de casos en fases diferentes a la FVP para la variable KRAS se encontró muy bajo siquiera para realizar entrenamientos.

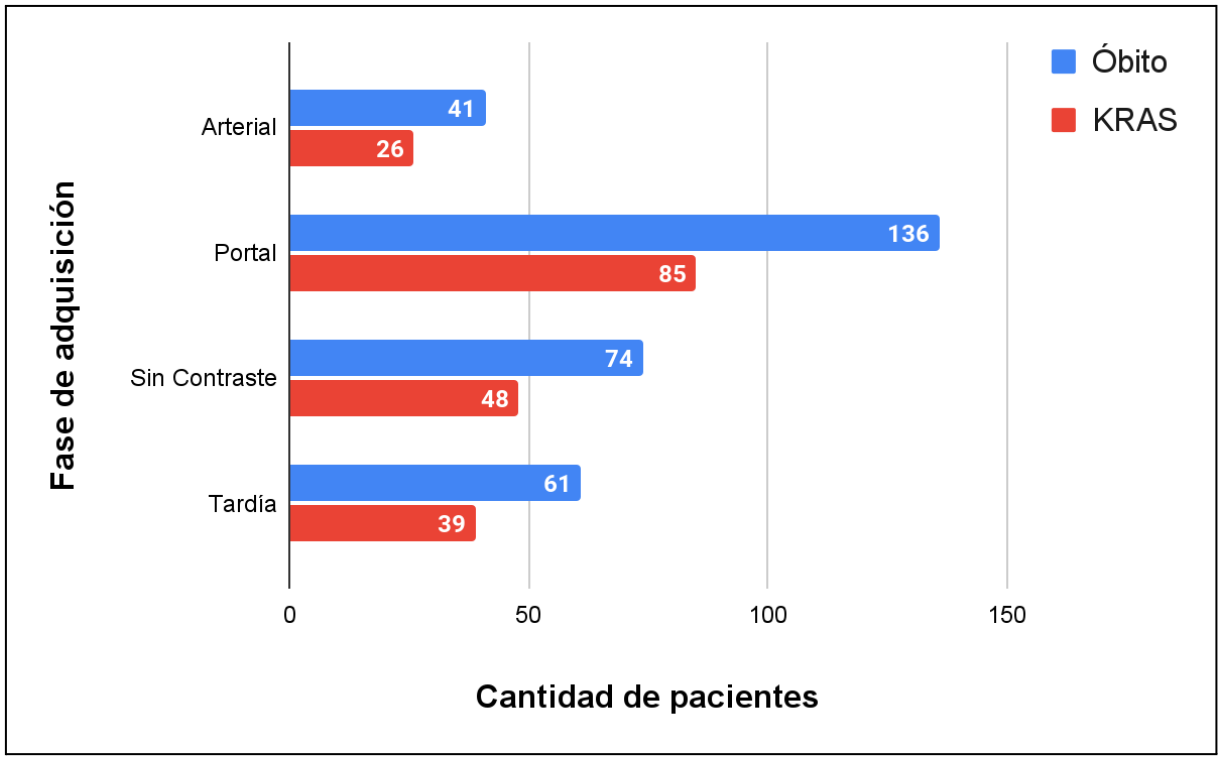

Figura A1. Cantidad de imágenes en cada fase, según la variable de respuesta.

La **Figura A2** muestra la cantidad de estudios de TC disponibles por paciente (sin tener en cuenta en qué fase se adquirieron) y separados por la variable Óbito. La mayoría de pacientes tenía estudios en una única fase (55), seguidos de estudios en cuatro fases (31), tres fases (29), y dos (23). Estos números influyeron en cómo realizar el ensamble de los modelos, dando prioridad al ensamble de modelos dentro de una misma fase, contrario a considerar la combinación de los resultados de múltiples de ellas. Uno de los problemas para realizar este tipo de análisis fue que, por ejemplo, de los 23 pacientes con estudios en dos fases, no todos tenían estudios en el mismo par de fases, reduciendo aún más la cantidad de estudios para comparar.

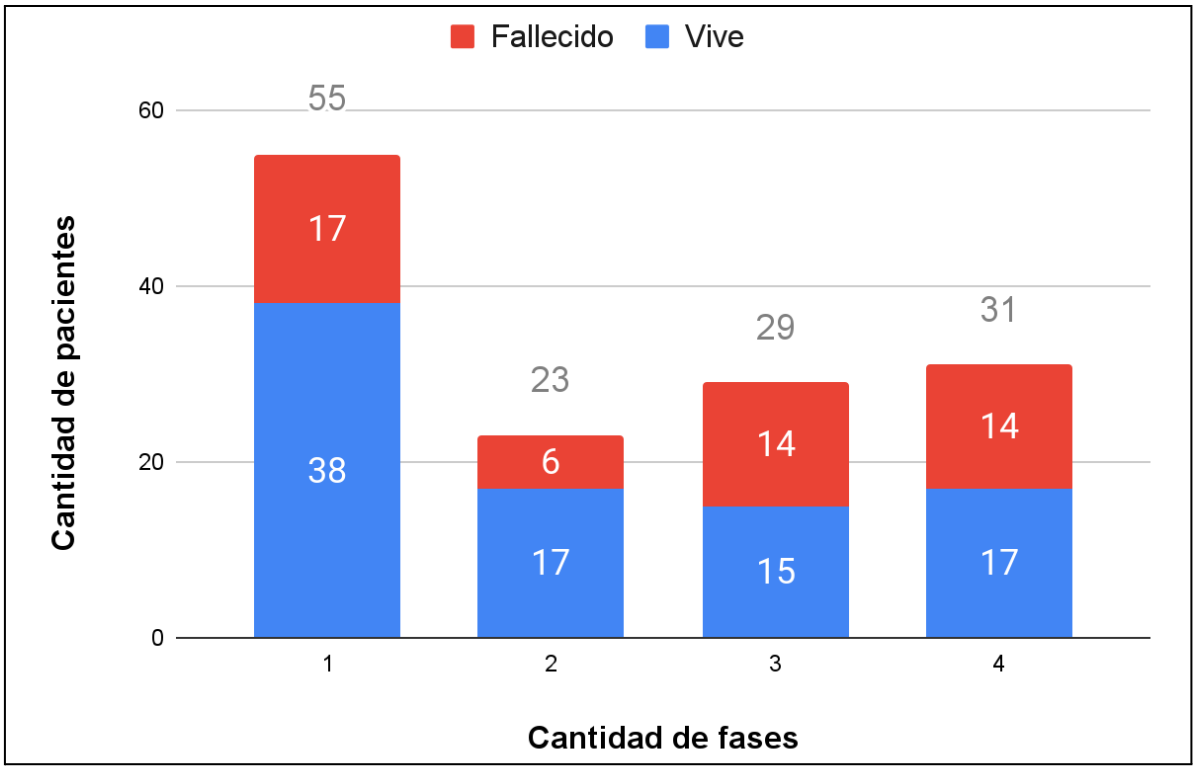

Figura A2. Cantidad de fases por paciente, respecto a la variable Óbito.

Se obtuvieron datos de 138 pacientes (87 negativos, 51 positivos; 84 hombres, 54 mujeres). Para la población en general, la mediana de la edad fue de 62,50 años, con un rango intercuartílico de 54 a 70 años. En 53 de los pacientes no se conocía el estado de mutación del gen KRAS, mientras que para 48 fue Wild Type, y 37 fue Mutado. La FVP fue aquella con más estudios disponibles, sumando 136; Se encontraron 74 casos en FSC, 61 con FVT y 41 con FA. En la **Tabla A1** se resumen los datos de la fase Óbito, desglosados por clase.

	Total	**Vive - 0**	**Fallecido - 1**
Total	**138**	**87**	**51**
Sexo			
Hombre	84 (60,87%)	49 (56,32%)	35 (68,63%)
Mujer	54 (39,13%)	38 (43,68%)	16 (31,37%)
Edad			
Mediana	62,50 (54,00-70,00)	60 (52,50-67,00)	66 (59,50-74,00)
KRAS			
Desconocido	53 (38,41%)	40 (45,98%)	13 (25,49%)
Mutado	37 (26,81%)	21 (24,14%)	16 (31,37%)
Wild Type	48 (34,78%)	26 (29,89%)	22 (43,14%)
Fases			
FA	41 (29,71%)	24 (27,59%)	17 (33,33%)
FVP	136 (98,55%)	86 (98,85%)	50 (98,04%)
FSC	74 (53,62%)	42 (48,28%)	32 (62,75%)
FVT	61 (44,20%)	33 (37,93%)	28 (54,90%)

Tabla A1. Frecuencia y distribución de los datos de los pacientes respecto a la variable Óbito. Los porcentajes corresponden a las proporciones del total de cada columna.

En la **Figura A3** se muestra el histograma de la variable Edad para la población Óbito, realizado con rangos de 5 años, y se encontró que su distribución es no normal.

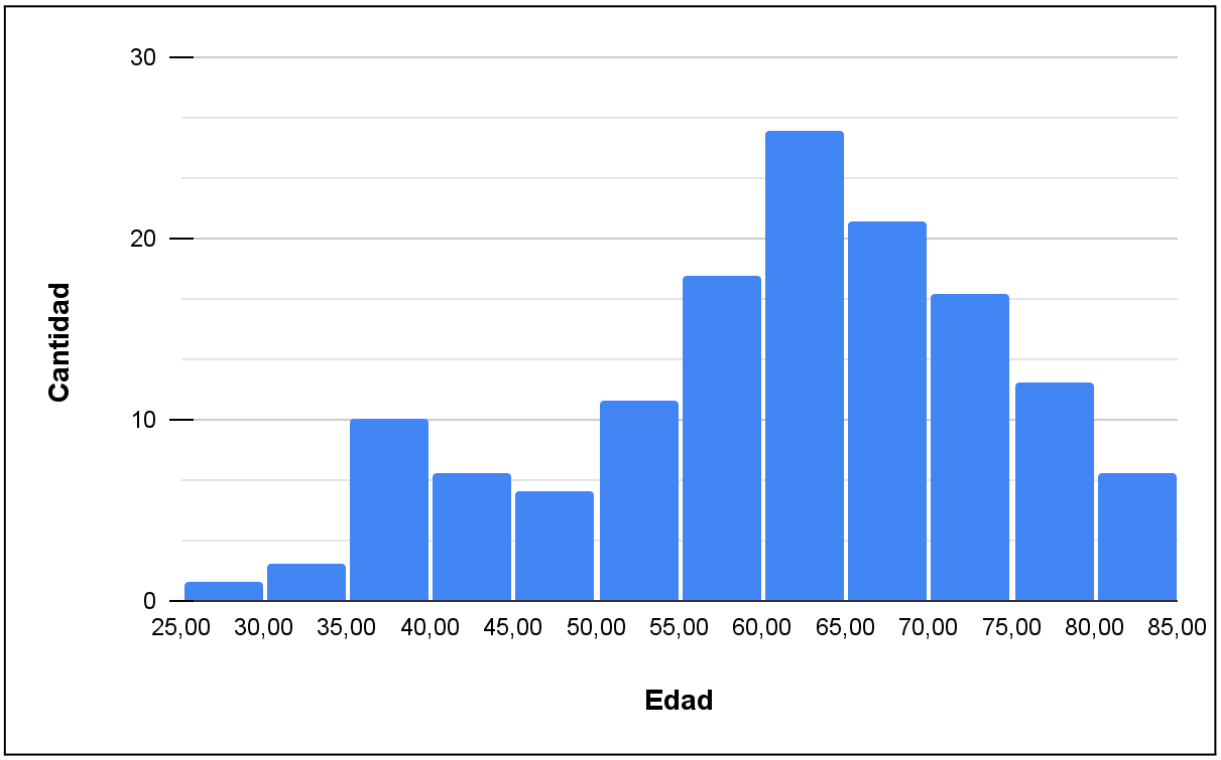

Figura A3. Histograma de edad de los pacientes (rangos de 5 años) en el estudio de Óbito.

En la **Figura A4** se muestra la cantidad de casos de la variable Óbito según la variable KRAS. En la mayoría de los casos se desconoció el estado de mutación, y se encontró una diferencia de 11 pacientes entre los estados Mutado y *Wild Type*. Si bien hay más casos mutados, tanto para los pacientes con estado Mutado y *Wild Type* el valor de la variable Óbito es parejo, con una mayoría de casos con el paciente vivo al finalizar el periodo de dos años.

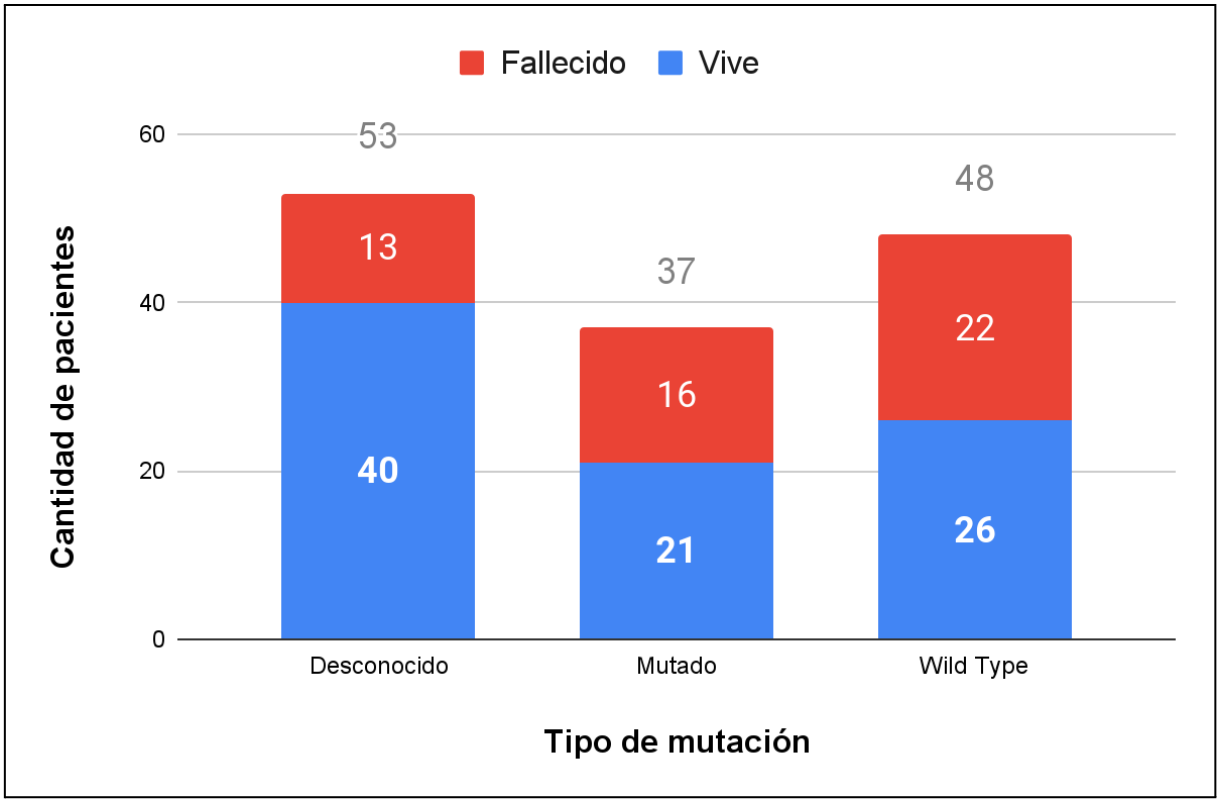

Figura A4. Cantidad de pacientes de la variable Óbito según la variable KRAS.

B. Ajuste de hiperparámetros

En las **Tablas B1** y **B2** se resumen los hiperparámetros probados con cada clasificador , y los rangos de valores probados. No se incluyeron aquellos que no fueron modificados, o los posibles valores que no fueron usados. Para describir un rango continuo de valores, se usó la notación (*x, y, n*), donde *x* fue el valor inicial, *y* el valor final (ambos valores incluidos), y *n* el valor del incremento entre cada elemento.

		Algoritmo de Ajuste		
Clasificador	Hiperparámetro	GridSearchCV	RandomizedSearchCV	Iteraciones
KNN	n_neighbors	(2, 7, 1)	(1, 4, 1)	1000
	weights	uniform, distance		
	algorithm	auto, ball_tree, kd_tree, brute		
	leaf_size	(10, 40, 5)	(5, 50, 1)	
	metric	euclidean, manhattan, chebyshev, minkowski		
DT	criterion	gini, entropy		500
	splitter	best, random	best	
	min_samples_split	(2, 20, 1)	(2, 40, 1)	
	min_samples_leaf	(1, 10, 1)	(1, 20, 1)	
	max_features	auto, sqrt, log2, None		
	min_impurity_decrease	(0, 10, 0,25)	0	
	class_weight	balanced, None		
RL	penalty	l1, l2, elasticnet, None	l2	2000
	tol	1x10-4, 1x10-5, 1x10-6	1x10-4	
	C	escala logarítmica de 100 valores, entre $1x10^{-5}$ y 500	(1, 300, 1)	
	fit_intercept	True, False		
	class_weight	balanced, None		
	max_iter	20.000	10.000	
	solver	newton-ng, lbfgs, liblinear, sag, saga		
	warm_start	True, False	False	
LDA	solver	svd, lsqr, eigen		500
	shrinkage	None, auto	None	
	tol	1x10-4, 1x10-5, 1x10-6	V	
ABC	n_estimators	(10, 200, 5)		500
	learning_rate	(1, 10, 0.5)		

Tabla B1. Configuración del ajuste de hiperparámetros para los clasificadores KNN, DT, RL, LDA y ABC. En los casos donde se usaron los mismos hiperparámetros en GridSearchCV y RandomizedSearchCV, se presentó una sola columna. La columna de Iteraciones corresponde a la cantidad de iteraciones realizadas para ese clasificador con RandomizedSearchCV. Abreviaturas: K vecinos cercanos (KNN), árbol de decisión (DT), regresión logística (RL), análisis de discriminante lineal (LDA), *AdaBoost Classifier* (ABC).

		Algoritmo de Ajuste		
Clasificador	Hiperparámetro	GridSearchCV	RandomizedSearchCV	Iteraciones
RF	n_estimators	(100, 1000, 9)	(100, 850, 150)	100
	criterion	gini, entropy		
	min_samples_split	(2, 20, 1)	(2, 20, 1)	
	min_samples_leaf	(1, 10, 1)	(2, 20, 1)	
	max_features	auto, sqrt, log2, 5, 10, 15, 20	auto, sqrt, log2	
	min_impurity_decrease	(0, 10, 0.25)	0	
	class_weight	balanced, None	balanced, balanced_subsample, None	
	warm_start	True, False	False	
SVM	C	escala logarítmica de 100 valores, entre $1x10^{-5}$ y 500	(1, 300, 1)	1000
	kernel	linear, poly, rbf, sigmoid, precomputed	linear, poly, rbf, sigmoid	
	degree	(1, 20, 1)	(3, 20, 1)	
	shrinking	True, False	False	
	gamma	scale, auto		
	class_weight	balanced, None		
	tol	1x10-3, 1x10-4, 1x10-5, 1x10-6	1x10-3	
XGB	booster	gbtree, gblinear, dart		500
	max_depth	(5, 20, 5)	(5, 40, 1)	
	subsample	(0,5, 1, 0,5)	(0,1, 1, 0,1)	
	lambda	(1, 3, 1)	(1, 5, 1)	
	max_delta_step	1	(1, 10, 1)	
GBC	loss		deviance, exponential	500
	learning_rate		(0.1, 1, 0.1)	
	n_estimators		(10, 300, 10)	
	criterion		friedman_mse, mse	
	min_samples_split		(2, 10, 1)	
	min_samples_leaf		(1, 10, 1)	
	max_depth		(1, 20, 1)	
	max_features		sqrt, log2, None	

Tabla B2. Configuración del ajuste de hiperparámetros para los clasificadores RF, SVM, XGB y GBC. En los casos donde se usaron los mismos hiperparámetros en GridSearchCV y RandomizedSearchCV, se presentó una sola columna. La columna de Iteraciones corresponde a la cantidad de iteraciones realizadas para ese clasificador con RandomizedSearchCV. Abreviaturas: bosque aleatorio (RF), máquina de vectores de soporte (SVM), *Extreme Gradient Boosting* (XGB), *Gradient Boosting Classifier* (GBC).

C. Métricas de modelos en objetivos primarios

En este anexo se muestran las métricas de discriminación y calibración de los modelos individuales que integran los modelos FVP Ensamble y KRAS Ensamble, junto a sus matrices de confusión.

C.1 Métricas: Óbito FVP individual

Las **Tablas C1** y **C2** informan sobre las métricas en entrenamiento y evaluación para los modelos FVP 1 y FVP 2, respectivamente. Se destaca que las métricas de ambos modelos son inferiores a las de FVP Ensamble, y en particular, FVP 1 cuenta con una sensibilidad baja, FVP 2 cuenta con una especificidad baja, pero FVP Ensamble obtuvo buenos resultados en ambas métricas, destacando la utilidad del ensamble.

	FVP 1					
	Entrenamiento			**Evaluación**		
Métrica	**Original**	**Bootstrap**	**IC**	**Original**	**Bootstrap**	**IC**
AUC ROC	0,873	0,879	0,808-0,936	0,755	0,756	0,560-0,920
AUC PR	0,798	0,806	0,808-0,936	0,618	0,624	0,560-0,920
Sensibilidad	0,595	0,595	0,444-0,75	0,692	0,690	0,429-0,933
Especificidad	0,859	0,871	0,780-0,946	0,773	0,761	0,571-0,920
VPP	0,710	0,732	0,563-0,879	0,643	0,635	0,385-0,867
VPN	0,786	0,784	0,685-0,877	0,810	0,803	0,619-0,957
Exactitud	0,762	0,768	0,683-0,842	0,743	0,734	0,600-0,886
Valor F1	0,647	0,653	0,516-0,765	0,667	0,653	0,424-0,846
MCC	0,474	0,490	0,301-0,650	0,459	0,443	0,143-0,746
ECE	0,086	0,114	0,068-0,168	0,202	0,233	0,132-0,354
MCE	0,224	0,439	0,206-0,801	0,583	0,606	0,339-0,848
BS	0,142	0,140	0,102-0,178	0,204	0,206	0,112-0,312

Tabla C1. Métricas del modelo FVP 1, separadas por los conjuntos de entrenamiento y evaluación. Los valores en la columna *bootstrap* corresponden a la media de cada métrica en las 1.000 iteraciones. Abreviaturas: intervalo de confianza (IC), área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), valor predictivo positivo (VPP), valor predictivo negativo (VPN), coeficiente de correlación de Matthews (MCC), error de calibración esperado (ECE), media del error de calibración (MCE), *Brier score* (BS).

	FVP 2					
	Entrenamiento			Evaluación		
Métrica	**Original**	**Bootstrap**	**IC**	**Original**	**Bootstrap**	**IC**
AUC ROC	0,876	0,873	0,790-0,939	0,806	0,795	0,618-0,932
AUC PR	0,808	0,809	0,790-0,939	0,752	0,751	0,618-0,932
Sensibilidad	0,838	0,834	0,710-0,943	0,769	0,766	0,533-1,000
Especificidad	0,797	0,794	0,711-0,943	0,682	0,662	0,458-0,850
VPP	0,705	0,704	0,571-0,838	0,588	0,592	0,353-0,824
VPN	0,895	0,890	0,804-0,964	0,833	0,816	0,611-1,000
Exactitud	0,812	0,809	0,723-0,881	0,714	0,703	0,543-0,857
Valor F1	0,765	0,761	0,649-0,857	0,667	0,661	0,444-0,846
MCC	0,618	0,611	0,440-0,757	0,436	0,418	0,089-0,715
ECE	0,072	0,088	0,041-0,148	0,159	0,174	0,073-0,292
MCE	0,135	0,209	0,085-0,396	0,410	0,492	0,167-0,743
BS	0,134	0,135	0,096-0,178	0,182	0,184	0,109-0,266

Tabla C2. Métricas del modelo FVP 2, separadas por los conjuntos de entrenamiento y evaluación. Los valores en la columna *bootstrap* corresponden a la media de cada métrica en las 1.000 iteraciones. Abreviaturas: intervalo de confianza (IC), área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), valor predictivo positivo (VPP), valor predictivo negativo (VPN), coeficiente de correlación de Matthews (MCC), error de calibración esperado (ECE), media del error de calibración (MCE), *Brier score* (BS).

Las **Figuras C1** y **C2** muestran las matrices de confusión de FVP 1, FVP 2 y FVP Ensamble en entrenamiento y evaluación, respectivamente. En estos gráficos se puede ver rápidamente la fortaleza de cada modelo al mirar la cantidad de VP (alta en FVP 2) y VN (alta en FVP 1), y la mejora en la clasificación producto del ensamble.

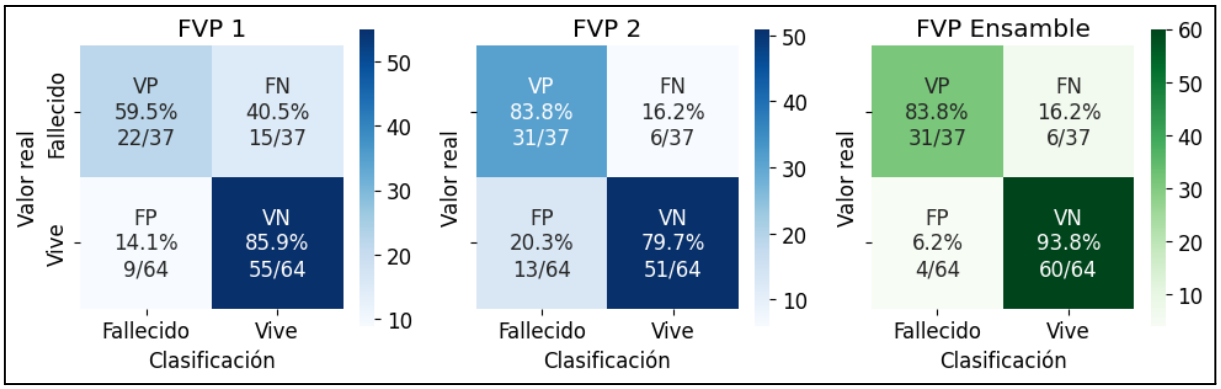

Figura C1. Matrices de confusión para los tres modelos de FVP en entrenamiento. El umbral de decisión fue 0,510. Abreviaturas: verdadero positivo (VP), falso negativo (FN), falso positivo (FP), verdadero negativo (VN).

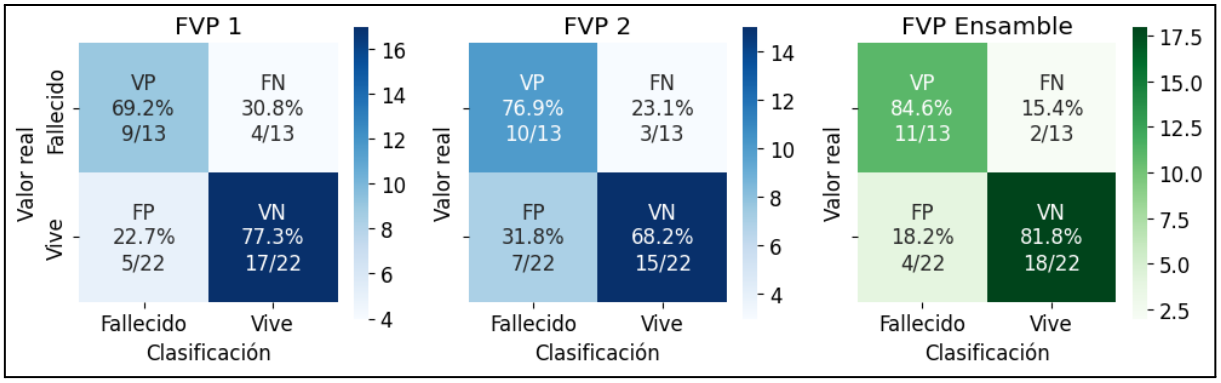

Figura C2. Matrices de confusión para los tres modelos de FVP en evaluación. El umbral de decisión para los modelos fue 0,510. Abreviaturas: verdadero positivo (VP), falso negativo (FN), falso positivo (FP), verdadero negativo (VN).

C.2 Métricas: KRAS FVP individual

Las **Tablas C3**, **C4** y **C5** informan sobre las métricas en entrenamiento y evaluación para los modelos KRAS 1, KRAS 2 y KRAS 3, respectivamente. Se destaca que las métricas de los tres modelos fueron inferiores a las de KRAS Ensamble. KRAS 1 y KRAS 3 tuvieron baja sensibilidad, F1 y MCC, pero gran especificidad y VPP, mientras KRAS 2 se destacó en su sensibilidad y VPN.

	KRAS 1					
	Entrenamiento			Evaluación		
Métrica	**Original**	**Bootstrap**	**IC**	**Original**	**Bootstrap**	**IC**
AUC ROC	0,818	0,821	0,714-0,917	0,735	0,713	0,526-0,864
AUC PR	0,823	0,822	0,714-0,917	0,747	0,703	0,526-0,864
Sensibilidad	0,542	0,565	0,364-0,765	0,385	0,327	0,078-0,600
Especificidad	0,936	0,934	0,833-1,000	0,941	0,938	0,813-1,000
VPP	0,867	0,864	0,668-1,000	0,833	0,833	0,800-1,000
VPN	0,725	0,744	0,605-0,875	0,667	0,664	0,478-0,846
Exactitud	0,764	0,777	0,673-0,891	0,700	0,686	0,500-0,833
Valor F1	0,667	0,678	0,485-0,833	0,526	0,447	0,154-0,800
MCC	0,531	0,550	0,338-0,755	0,404	0,346	0,000-0,676
ECE	0,042	0,086	0,023-0,165	0,053	0,120	0,033-0,235
MCE	0,063	0,146	0,037-0,305	0,061	0,234	0,083-0,560
BS	0,163	0,160	0,109-0,216	0,198	0,209	0,138-0,290

Tabla C3. Métricas del modelo KRAS 1, separadas por los conjuntos de entrenamiento y evaluación. Los valores en la columna *bootstrap* corresponden a la media de cada métrica en las 1.000 iteraciones. Abreviaturas: intervalo de confianza (IC), área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), valor predictivo positivo (VPP), valor predictivo negativo (VPN), coeficiente de correlación de Matthews (MCC), error de calibración esperado (ECE), media del error de calibración (MCE), *Brier score* (BS).

	KRAS 2					
	Entrenamiento			Evaluación		
Métrica	**Original**	**Bootstrap**	**IC**	**Original**	**Bootstrap**	**IC**
AUC ROC	0,996	0,996	0,983-1,000	0,724	0,720	0,525-0,894
AUC PR	0,995	0,994	0,983-1,000	0,671	0,659	0,525-0,894
Sensibilidad	1,000	1,000	1,000-1,000	0,769	0,751	0,471-1,000
Especificidad	0,968	0,970	0,895-1,000	0,647	0,650	0,412-0,875
VPP	0,960	0,960	0,970-1,000	0,625	0,601	0,333-0,833
VPN	1,000	1,000	1,000-1,000	0,786	0,787	0,556-1,000
Exactitud	0,982	0,982	0,946-1,000	0,700	0,692	0,533-0,833
Valor F1	0,980	0,979	0,930-1,000	0,690	0,660	0,417-0,840
MCC	0,964	0,964	0,892-1,000	0,414	0,394	0,067-0,707
ECE	0,052	0,054	0,020-0,100	0,288	0,309	0,185-0,454
MCE	0,430	0,392	0,207-0,430	0,825	0,751	0,412-0,825
BS	0,027	0,027	0,004-0,073	0,274	0,288	0,159-0,431

Tabla C4. Métricas del modelo KRAS 2, separadas por los conjuntos de entrenamiento y evaluación. Los valores en la columna *bootstrap* corresponden a la media de cada métrica en las 1.000 iteraciones. Abreviaturas: intervalo de confianza (IC), área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), valor predictivo positivo (VPP), valor predictivo negativo (VPN), coeficiente de correlación de Matthews (MCC), error de calibración esperado (ECE), media del error de calibración (MCE), *Brier score* (BS).

	KRAS 3					
	Entrenamiento			Evaluación		
Métrica	**Original**	**Bootstrap**	**IC**	**Original**	**Bootstrap**	**IC**
AUC ROC	0,834	0,828	0,730-0,917	0,776	0,789	0,630-0,913
AUC PR	0,842	0,830	0,730-0,917	0,771	0,776	0,630-0,913
Sensibilidad	0,708	0,697	0,500-0,882	0,462	0,499	0,231-0,778
Especificidad	0,871	0,870	0,741-0,970	0,882	0,885	0,714-1,000
VPP	0,810	0,801	0,600-0,950	0,750	0,755	0,429-1,000
VPN	0,794	0,794	0,657-0,925	0,682	0,715	0,522-0,895
Exactitud	0,800	0,796	0,691-0,891	0,700	0,726	0,567-0,867
Valor F1	0,756	0,741	0,579-0,873	0,571	0,588	0,316-0,818
MCC	0,591	0,580	0,361-0,784	0,385	0,422	0,073-0,736
ECE	0,039	0,075	0,020-0,154	0,103	0,107	0,019-0,225
MCE	0,051	0,194	0,050-0,430	0,152	0,232	0,041-0,569
BS	0,152	0,154	0,103-0,209	0,190	0,179	0,107-0,257

Tabla C5. Métricas del modelo KRAS 3, separadas por los conjuntos de entrenamiento y evaluación. Los valores en la columna *bootstrap* corresponden a la media de cada métrica en las 1.000 iteraciones. Abreviaturas: intervalo de confianza (IC), área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), valor predictivo positivo (VPP), valor predictivo negativo (VPN), coeficiente de correlación de Matthews (MCC), error de calibración esperado (ECE), media del error de calibración (MCE), *Brier score* (BS).

Las **Figuras C3** y **C4** muestran las matrices de confusión de KRAS 1, KRAS 2, KRAS 3 y KRAS Ensamble en entrenamiento y evaluación, respectivamente. En estos gráficos se

observa que KRAS 1 se desempeñó bien solo en los casos negativos, KRAS 2 sobreajustó en entrenamiento, y KRAS 3 fue balanceado. En evaluación, el desempeño de KRAS 2 y KRAS 3 cayó, pero su uso en el ensamble llevó a un modelo que superó todos sus componentes.

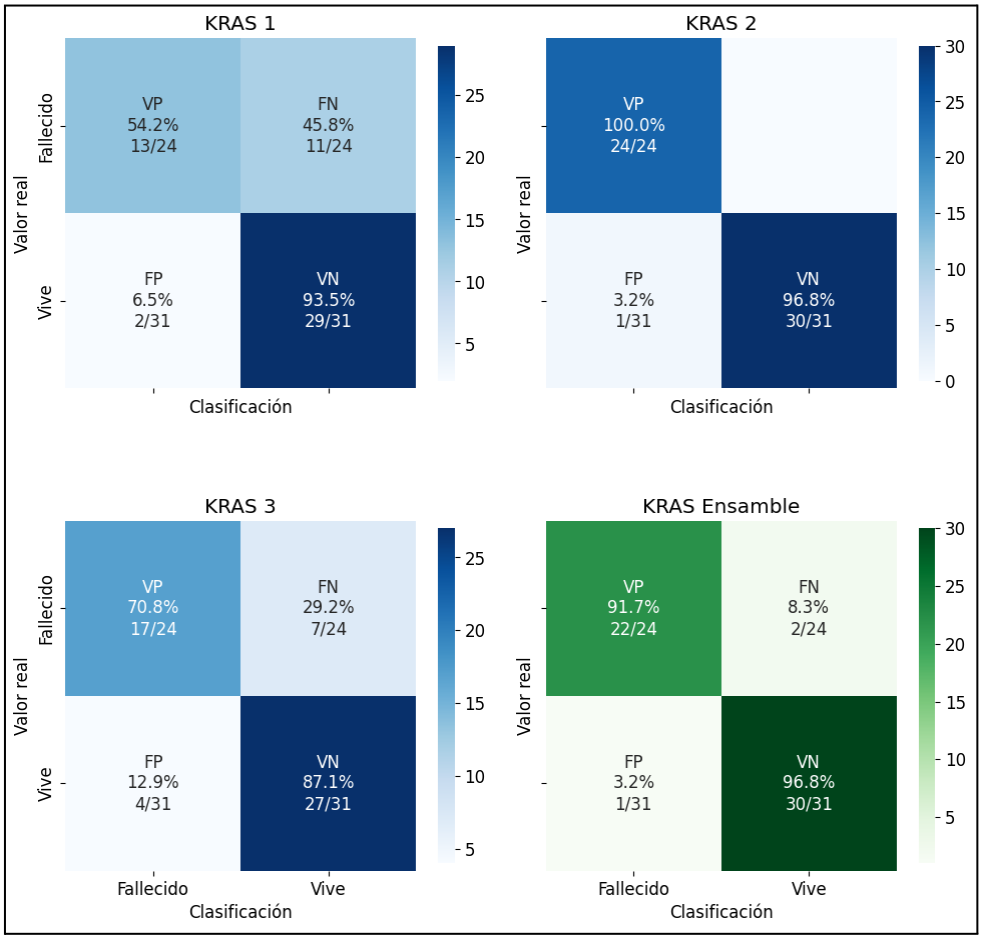

Figura C3. Matrices de confusión para los cuatro modelos KRAS en entrenamiento. El umbral de decisión para los modelos fue 0,489. Abreviaturas: verdadero positivo (VP), falso negativo (FN), falso positivo (FP), verdadero negativo (VN).

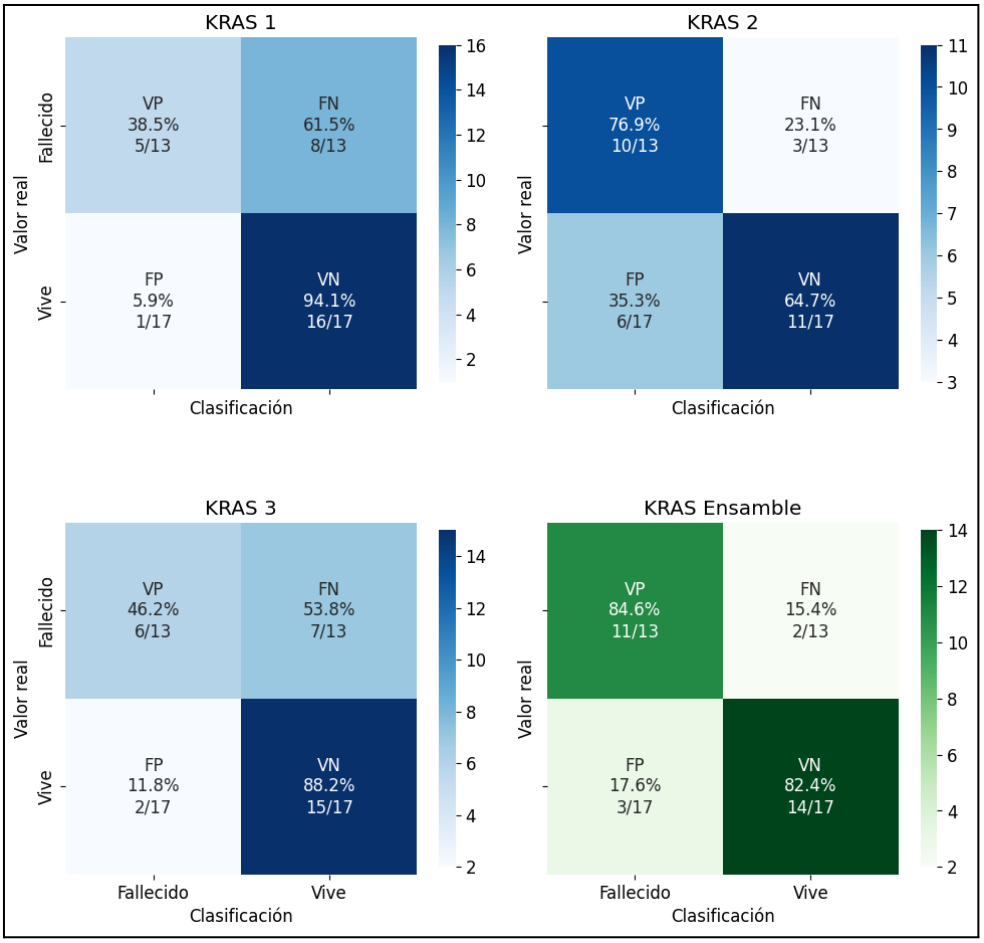

Figura C4. Matrices de confusión para los cuatro modelos KRAS en evaluación. El umbral de decisión para los modelos fue 0,489. Abreviaturas: verdadero positivo (VP), falso negativo (FN), falso positivo (FP), verdadero negativo (VN).

D. Resultados de objetivos secundarios

En este anexo se presenta la información de los modelos seleccionados para la FA, la FVT y la FSC. Incluye la descripción de los modelos, sus métricas de discriminación y calibración, matrices de confusión, y sus curvas ROC y PR.

D.1 Métricas: Óbito FA

Se entrenaron 486 modelos en FA, de los cuales se seleccionó el mejor, llamado FA, por su valor de AUC ROC de 0,961 en entrenamiento y 0,880 en evaluación. Muchos casos fueron descartados por AUC ROC de entrenamiento de 1,0. El modelo FA está detallado en la **Tabla D1**, en la cual se reportaron las tres características que fueron encontradas de mayor importancia (según la definición de cada algoritmo) para cada modelo.

	Modelo
	FA
Extracción	LdG
Selección	Eliminación Hacia Atrás
Clasificador	Análisis de Discriminante Lineal
Número de características	10
Características de mayor importancia	log-sigma-0-5-mm-3D_glcm_ClusterProminence
	log-sigma-0-5-mm-3D_glcm_ClusterShade
	log-sigma-0-5-mm-3D_glrlm_GrayLevelVariance
AUC ROC - E	0,961
AUC PR - E	0,939
AUC ROC - P	0,880
AUC PR - P	0,920

Tabla D1. Descripción de los mejores modelos para la variable Óbito en FA. Abreviaturas: área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), AUC ROC en entrenamiento (AUC ROC - E), AUC PR en entrenamiento (AUC PR - E), AUC ROC en evaluación (AUC ROC - P), AUC PR en evaluación (AUC PR - P), Laplaciano de Gaussiano (LdG).

Las métricas de este modelo se presentan en la **Tabla 16**, tanto en entrenamiento como evaluación, calculadas con un umbral de 0,305 obtenido con el método de la Media Geométrica. Se encontraron métricas elevadas en el entrenamiento, con los casos particulares de sensibilidad y VPN en 1,000. Estos no se repitieron en evaluación, sino que cayeron considerablemente, mientras que incrementaron notablemente la especificidad y el VPN. Comparando evaluación con entrenamiento, se vió un incremento en los errores de calibración.

	FA					
	Entrenamiento			Evaluación		
Métrica	**Original**	**Bootstrap**	**IC**	**Original**	**Bootstrap**	**IC**
AUC ROC	0,961	0,958	0,882-1,000	0,880	0,899	0,625-1,000
AUC PR	0,939	0,936	0,882-1,000	0,920	0,933	0,625-1,000
Sensibilidad	1,000	1,000	1,000-1,000	0,600	0,608	0,167-1,000
Especificidad	0,789	0,834	0,667-1,000	1,000	1,000	1,000-1,000
VPP	0,800	0,796	0,588-1,000	1,000	0,984	1,000-1,000
VPN	0,800	1,000	1,000-1,000	1,000	0,673	0,286-1,000
Exactitud	0,903	0,900	0,774-1,000	0,800	0,782	0,500-1,000
Valor F1	0,900	0,885	0,741-1,000	0,750	0,731	0,286-1,000
MCC	0,821	0,816	0,630-1,000	0,655	0,628	0,272-1,000
ECE	0,119	0,136	0,060-0,221	0,281	0,315	0,102-0,558
MCE	0,782	0,738	0,469-0,782	0,775	0,699	0,335-0,775
BS	0,078	0,079	0,029-0,139	0,208	0,234	0,048-0,450

Tabla D2. Métricas del mejor modelo para la FA, separadas en conjuntos de entrenamiento y evaluación. Los valores en la columna *bootstrap* corresponden a la media de cada métrica en las 1.000 iteraciones. Abreviaturas: intervalo de confianza (IC), área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), valor predictivo positivo (VPP), valor predictivo negativo (VPN), coeficiente de correlación de Matthews (MCC), error de calibración esperado (ECE), media del error de calibración (MCE), *Brier score* (BS).

El número de casos usados en entrenamiento y evaluación pudo ser el causante de estas métricas contradictorias. En la **Figura D1** se muestran las matrices de confusión del modelo seleccionado en FA, para los conjuntos de entrenamiento y evaluación. La inversión de los falsos positivos con falsos negativos respalda que los casos utilizados para el entrenamiento y evaluación no son suficientes.

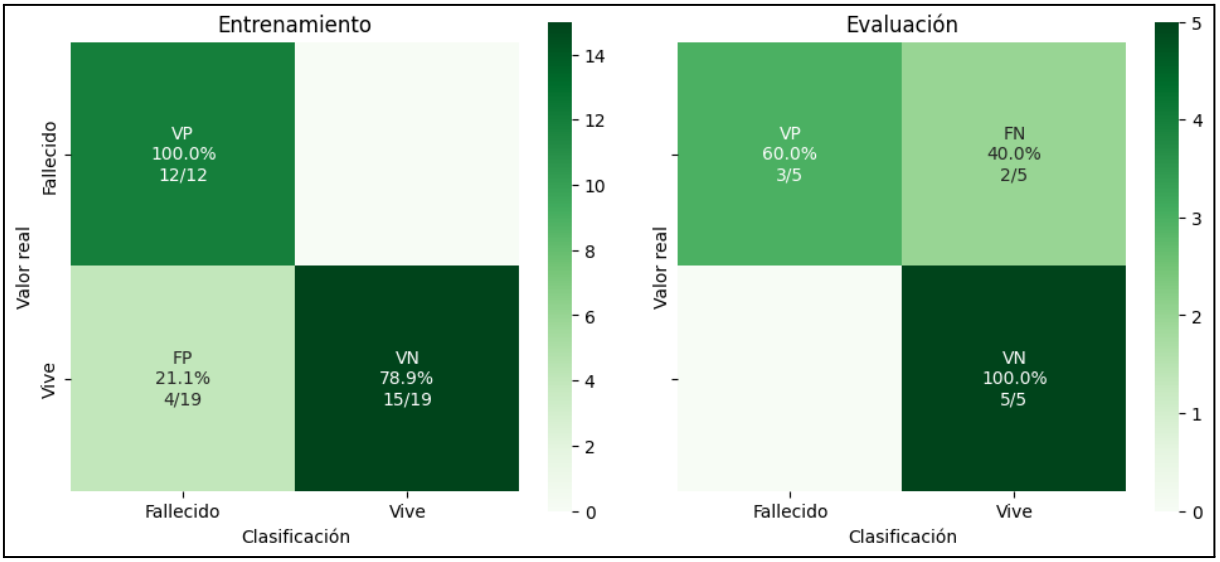

Figura 68. Matrices de confusión para el mejor modelo en FA. El umbral de decisión fue 0,305. Abreviaturas: verdadero positivo (VP), falso negativo (FN), falso positivo (FP), verdadero negativo (VN).

Las **Figuras D2** y **D3** muestran las curvas ROC y PR del modelo FA. En ambos casos se ve solo una caída leve del AUC ROC y AUC PR entre entrenamiento y evaluación. Sin embargo, habiendo visto las métricas de discriminación y calibración, se entiende que estos resultados no son generalizables.

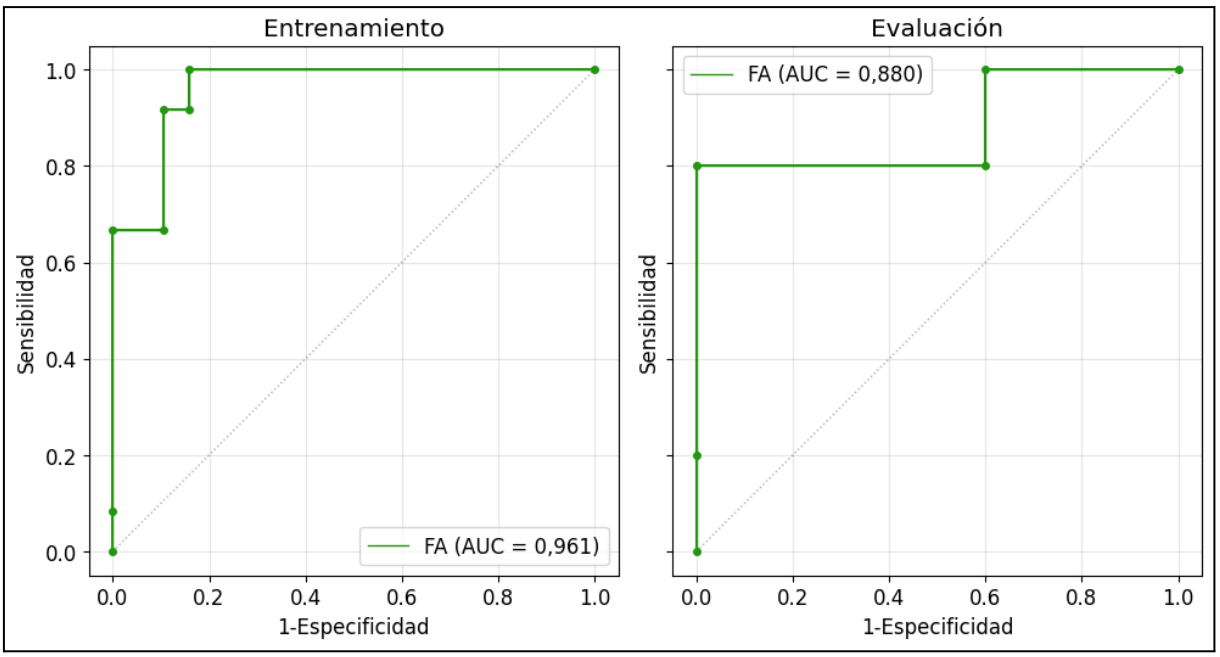

Figura D2. Curvas ROC para el mejor modelo en FA. Los posibles puntos de corte están marcados con puntos de mayor grosor. La línea diagonal gris marca el umbral de no discriminación. Abreviaturas: área bajo la curva (AUC).

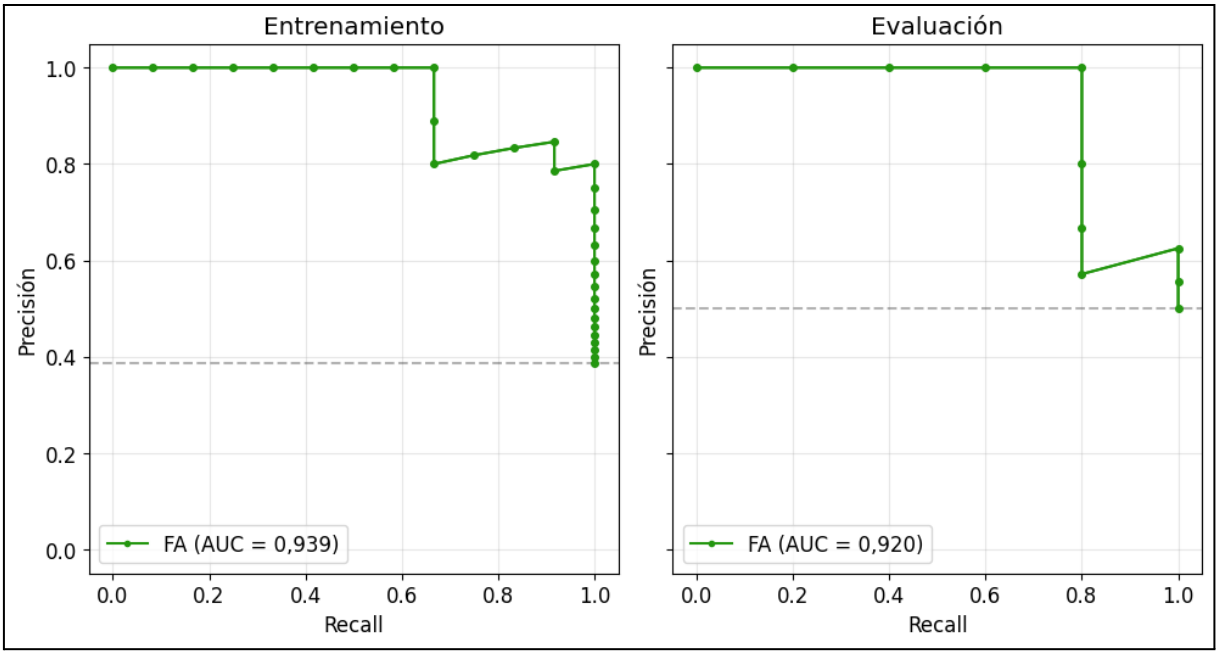

Figura D3. Curvas PR para el mejor modelo en FA. Los posibles puntos de corte están marcados con puntos de mayor grosor. La línea recta gris marca el umbral de no discriminación. Abreviaturas: área bajo la curva (AUC).

D.2 Métricas: Óbito FVT

Se entrenaron 488 modelos en FVT, del cual se seleccionó el mejor, llamado FVT, por su valor de AUC ROC de 0,860 en entrenamiento y 0,903 en evaluación. Los detalles de este modelo se encuentran en la **Tabla D3**. Se reportaron las tres características que fueron encontradas de mayor importancia (según la definición de cada algoritmo) para cada modelo.

	Modelo
	FVT
Extracción	Original+LdG
Selección	Selección Hacia Adelante
Clasificador	Análisis de Discriminante Lineal
Número de características	10
Características de mayor importancia	original_glszm_GrayLevelNonUniformity
	original_glszm_LargeAreaLowGrayLevelEmphasis
	original_gldm_DependenceNonUniformity
AUC ROC - E	0,860
AUC PR - E	0,797
AUC ROC - P	0,903
AUC PR - P	0,928

Tabla D3. Descripción del mejor modelo en FVT. Abreviaturas: área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), AUC ROC en entrenamiento (AUC ROC - E), AUC PR en entrenamiento (AUC PR - E), AUC ROC en evaluación (AUC ROC - P), AUC PR en evaluación (AUC PR - P), Laplaciano de Gaussiano (LdG).

Las métricas de este modelo se presentan en la **Tabla D4**. Se presentan tanto las métricas en entrenamiento como evaluación, calculadas con un umbral de 0,489 obtenido con el método de

la Media Geométrica. A pesar de encontrar un incremento en AUC ROC y AUC PR en evaluación, disminuyeron las restantes métricas de discriminación. Comparando evaluación con entrenamiento, hubo un incremento en los errores de calibración.

	FVT					
	Entrenamiento			Evaluación		
Métrica	**Original**	**Bootstrap**	**IC**	**Original**	**Bootstrap**	**IC**
AUC ROC	0,860	0,885	0,773-0,973	0,903	0,908	0,714-1,000
AUC PR	0,797	0,833	0,773-0,973	0,928	0,933	0,714-1,000
Sensibilidad	0,950	0,949	0,833-1,000	0,875	0,876	0,600-1,000
Especificidad	0,750	0,785	0,607-0,947	0,667	0,619	0,250-1,000
VPP	0,760	0,793	0,621-0,941	0,700	0,700	0,400-1,000
VPN	0,760	0,947	0,824-1,000	0,700	0,831	0,500-1,000
Exactitud	0,841	0,861	0,750-0,955	0,765	0,750	0,529-0,941
Valor F1	0,844	0,861	0,732-0,957	0,778	0,768	0,500-0,947
MCC	0,704	0,736	0,530-0,913	0,549	0,510	0,070-0,887
ECE	0,166	0,192	0,123-0,277	0,184	0,223	0,101-0,362
MCE	0,367	0,523	0,355-0,876	0,497	0,498	0,229-0,706
BS	0,156	0,148	0,104-0,198	0,158	0,160	0,075-0,269

Tabla D4. Métricas modelo en FVT, separadas en conjuntos de entrenamiento y evaluación. Los valores en la columna *bootstrap* corresponden a la media de cada métrica en las 1.000 iteraciones. Abreviaturas: intervalo de confianza (IC), área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), valor predictivo positivo (VPP), valor predictivo negativo (VPN),, coeficiente de correlación de Matthews (MCC), error de calibración esperado (ECE), media del error de calibración (MCE), *Brier score* (BS).

La **Figura D4** muestra las matrices de confusión del modelo seleccionado en FVT, para los conjuntos de entrenamiento y evaluación. Al comparar el desempeño en ambos conjuntos, se ve una reducción ligera en la tasa de aciertos tanto para la clase negativa como positiva, a pesar de haber incrementado en AUC ROC y AUC PR.

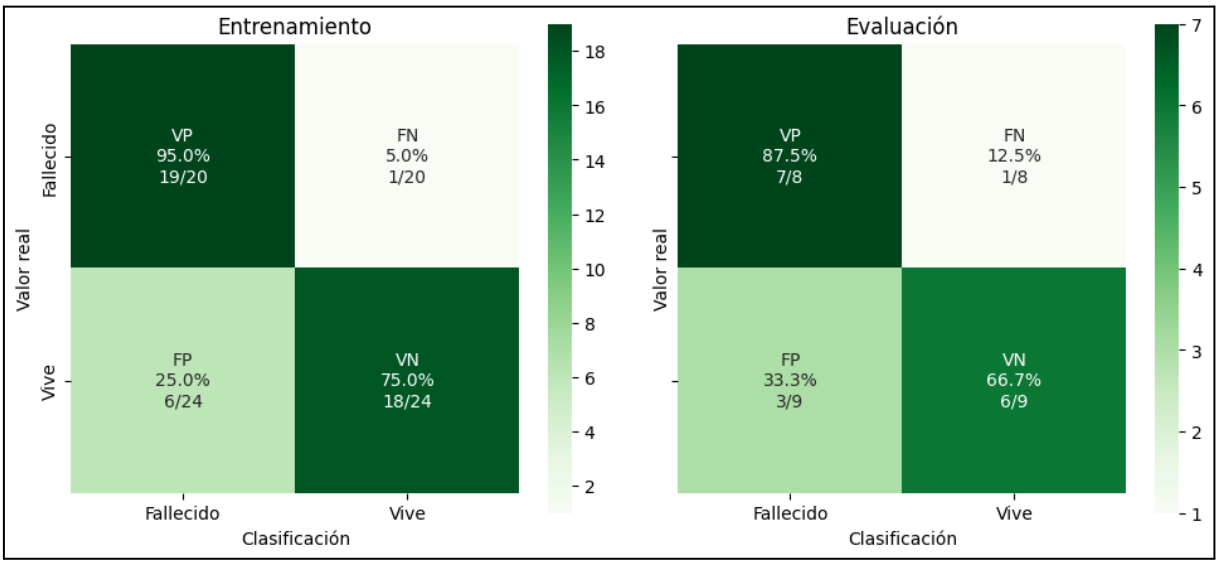

Figura D4 . Matrices de confusión para el mejor modelo en FVT. El umbral de decisión para los modelos fue 0,489. Abreviaturas: verdadero positivo (VP), falso negativo (FN), falso positivo (FP), verdadero negativo (VN).

Las **Figuras D5** y **D5** muestran las curvas ROC y PR del modelo FVT. Como se presentó en la tabla anterior, se ve un incremento del AUC entre entrenamiento y evaluación. Sin embargo, las métricas de discriminación sugieren que este incremento no se ve acompañado de un incremento en el desempeño.

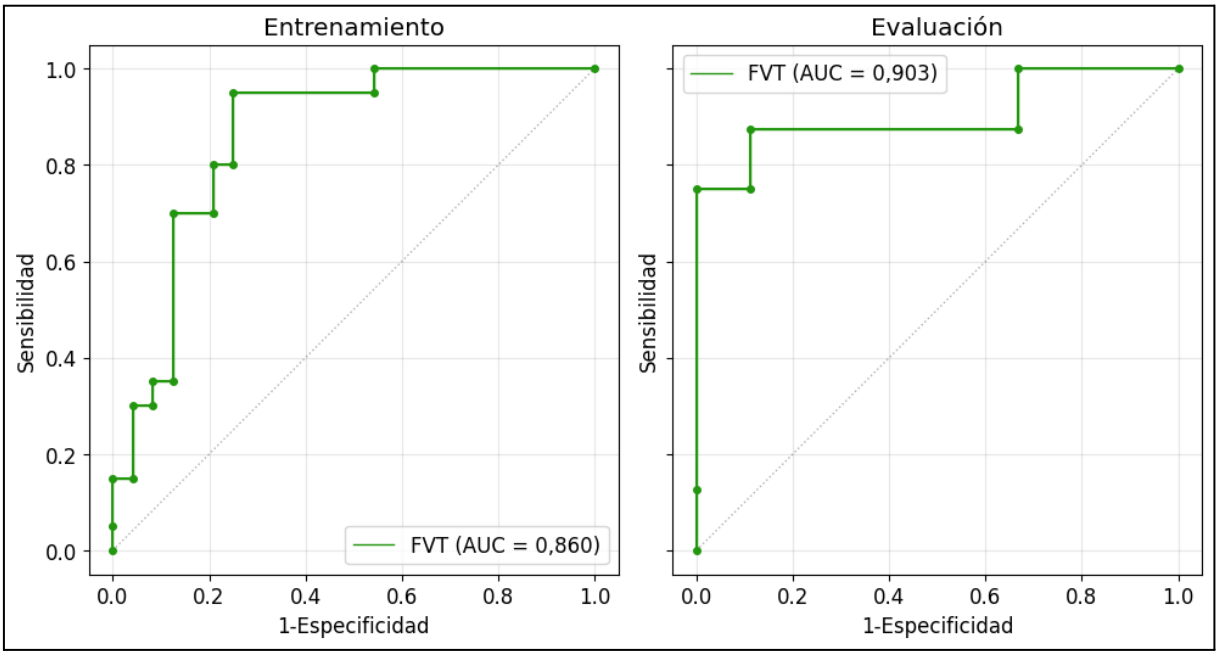

Figura D5. Curvas ROC para el mejor modelo en FVT. Los posibles puntos de corte están marcados con puntos de mayor grosor. La línea diagonal gris marca el umbral de no discriminación. Abreviaturas: área bajo la curva (AUC).

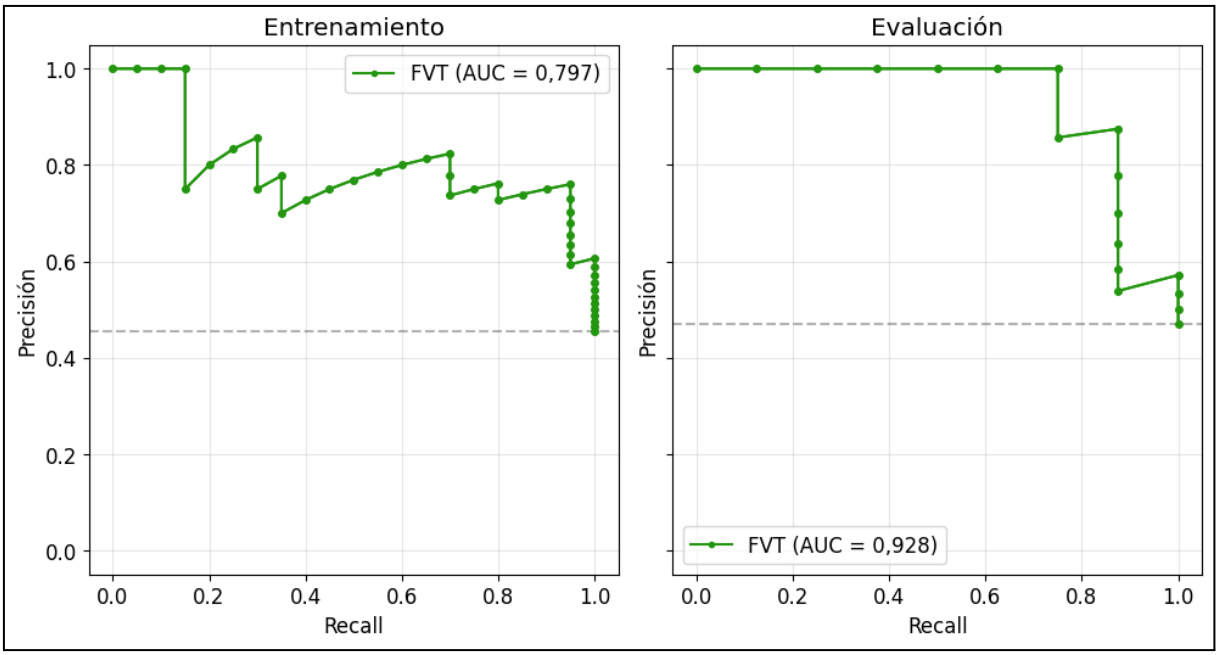

Figura D5. Curvas PR para el mejor modelo en FVT. Los posibles puntos de corte están marcados con puntos de mayor grosor. La línea recta gris marca el umbral de no discriminación. Abreviaturas: área bajo la curva (AUC).

D.3 Métricas: Óbito FSC

Se entrenaron 489 modelos en FSC, de los cuales se seleccionó el mejor, llamado FSC, por su valor de AUC ROC de 0,904 en entrenamiento y 0,706 en evaluación, detallado en la **Tabla D5**. Se reportaron las tres características que fueron encontradas de mayor importancia (según la definición de cada algoritmo) para cada modelo.

	Modelo
	FSC
Extracción	Original+LdG
Selección	LASSO
Clasificador	Árbol de Decisión
Número de características	10
Características de mayor importancia	original_shape_Flatness
	original_shape_Maximum2DDiameterRow
	original_shape_Sphericity
AUC ROC - E	0,904
AUC PR - E	0,900
AUC ROC - P	0,706
AUC PR - P	0,702

Tabla D5. Descripción del mejor modelo en FSC. Abreviaturas: área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), AUC ROC en entrenamiento (AUC ROC - E), AUC PR en entrenamiento (AUC PR - E), AUC ROC en evaluación (AUC ROC - P), AUC PR en evaluación (AUC PR - P), Laplaciano de Gaussiano (LdG), *Least Absolute Shrinkage and Selection Operator* (LASSO).

Las métricas de este modelo se presentan en la **Tabla D6**. Se presentan tanto las métricas en entrenamiento como evaluación, calculadas con un umbral de 0,500 obtenido con el método de la Media Geométrica. Con la excepción de la sensibilidad y el MCC, se obtuvieron buenos resultados en entrenamiento, pero se observó una caída generalizada de todas las métricas en evaluación. Se encontraron múltiples métricas con gran variación en el *bootstrap*. Todas las métricas de discriminación fueron menores a 0,700 en evaluación, y se vió un incremento de los errores de calibración.

	FSC					
	Entrenamiento			Evaluación		
Métrica	**Original**	**Bootstrap**	**IC**	**Original**	**Bootstrap**	**IC**
AUC ROC	0,904	0,904	0,825-0,968	0,706	0,705	0,431-0,917
AUC PR	0,900	0,902	0,825-0,968	0,702	0,714	0,431-0,917
Sensibilidad	0,875	0,627	0,423-0,826	0,875	0,497	0,143-0,857
Especificidad	0,781	0,969	0,897-1,000	0,500	0,667	0,333-1,000
VPP	0,750	0,940	0,800-1,000	0,583	0,572	0,167-1,000
VPN	0,750	0,769	0,641-0,897	0,583	0,597	0,273-0,900
Exactitud	0,821	0,819	0,714-0,911	0,667	0,586	0,333-0,833
Valor F1	0,808	0,747	0,579-0,884	0,700	0,515	0,154-0,783
MCC	0,650	0,649	0,464-0,821	0,395	0,166	0,000-0,632
ECE	0,000	0,060	0,016-0,121	0,158	0,237	0,084-0,429
MCE	0,000	0,149	0,039-0,333	0,363	0,446	0,167-0,791
BS	0,115	0,116	0,068-0,173	0,257	0,256	0,114-0,422

Tabla D6. Métricas del mejor modelo en FSC, en conjuntos de entrenamiento y evaluación. Los valores en la columna *bootstrap* corresponden a la media de cada métrica en las 1.000 iteraciones. Abreviaturas: intervalo de confianza (IC), área bajo la curva de característica operativa del receptor (AUC ROC), área bajo la curva *precision recall* (AUC PR), valor predictivo positivo (VPP), valor predictivo negativo (VPN), coeficiente de correlación de Matthews (MCC), error de calibración esperado (ECE), media del error de calibración (MCE), *Brier score* (BS).

Se encontró una elevada tasa de falsos positivos en entrenamiento, como se muestra en la **Figura D7.** En evaluación, este error se vió incrementado, mostrando sobreconfianza en la clasificación positiva, llevando a la baja de especificidad y los valores predictivos.

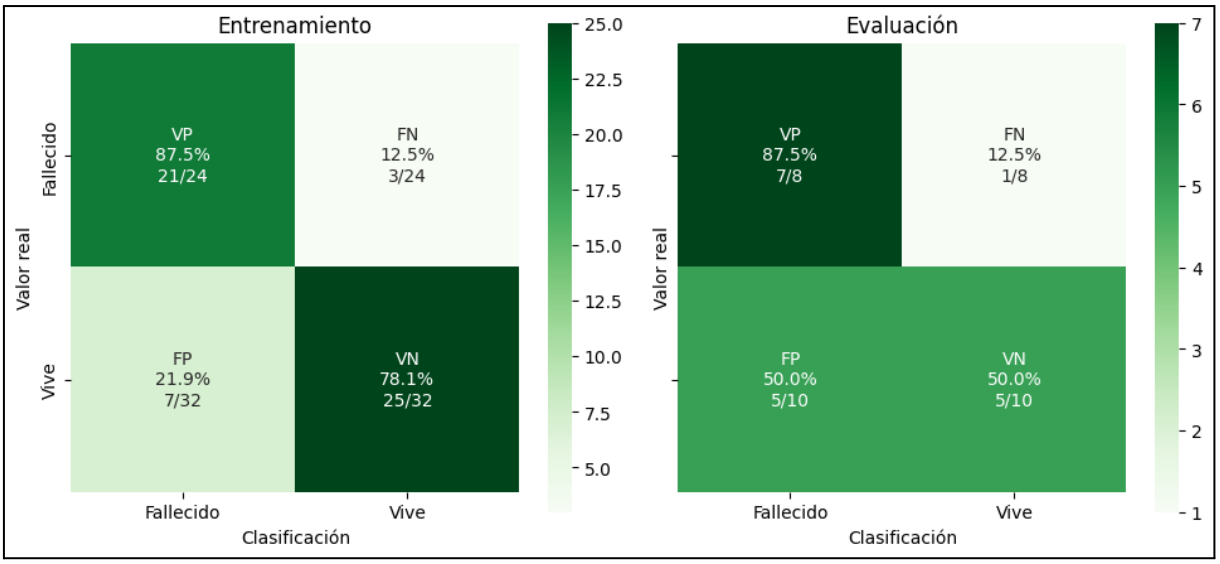

Figura D7. Matriz de confusión para el mejor modelo de Óbito en FSC. El umbral de decisión para los modelos fue 0,500. Abreviaturas: verdadero positivo (VP), falso negativo (FN), falso positivo (FP), verdadero negativo (VN).

Las **Figuras D8** y **D9** muestran las curvas ROC y PR del modelo FSC. En ambas se ve la caída más grande del AUC ROC y AUC PR entre conjuntos de todas las fases.

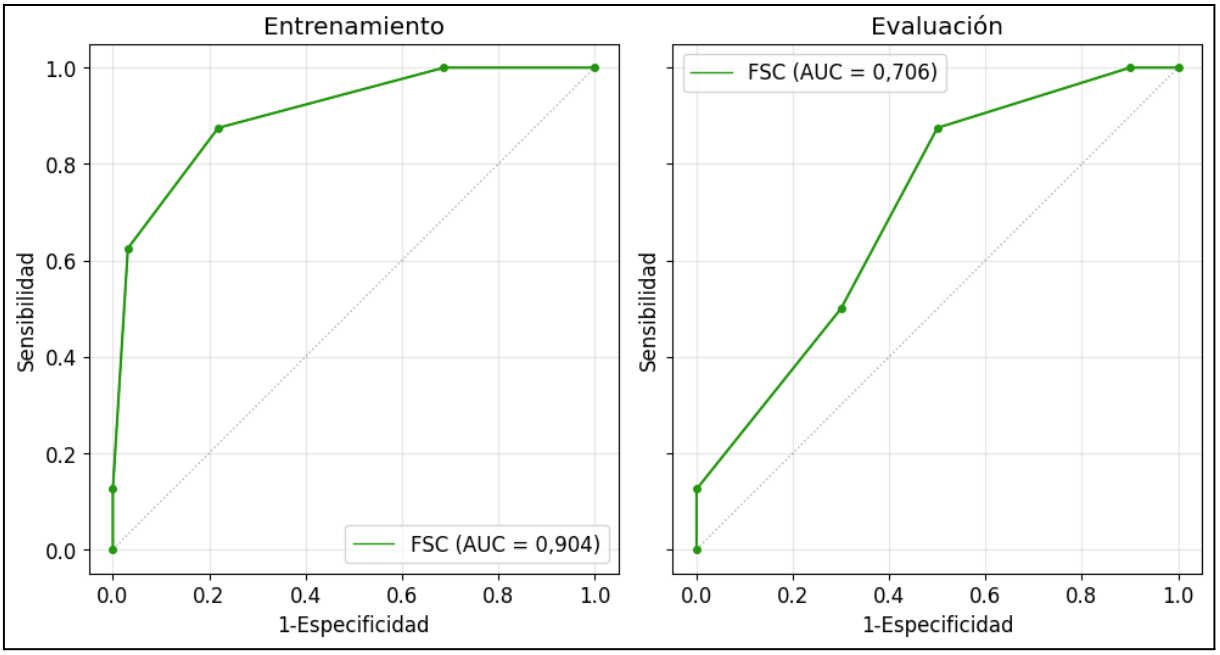

Figura D8. Curvas ROC para el mejor modelo de Óbito en FSC. Los posibles puntos de corte están marcados con puntos de mayor grosor. La línea diagonal gris el umbral de no discriminación. Abreviaturas: área bajo la curva (AUC).

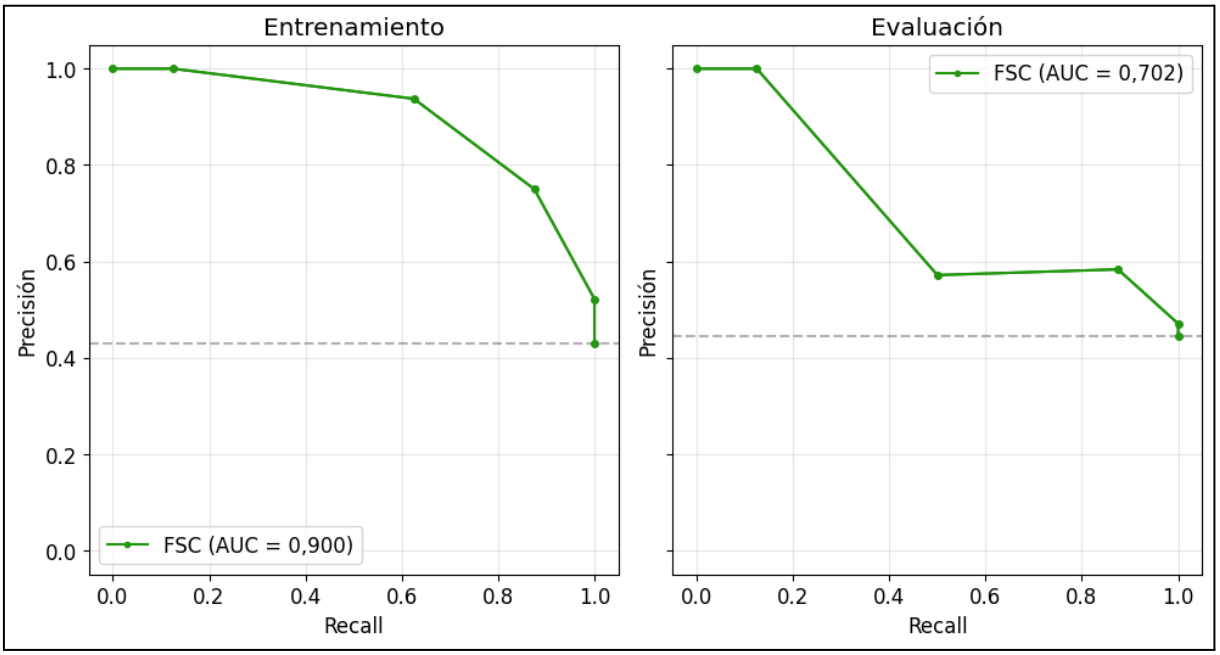

Figura D9. Curvas PR para el mejor modelo de Óbito en FSC. Los posibles puntos de corte se marcaron con puntos de mayor grosor. La línea punteada marca el umbral de no discriminación. Abreviaturas: área bajo la curva (AUC).