



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Implementación de un prototipo de software para predecir complicaciones en pacientes en unidad de cuidados intensivos pediátricos (UCIP), a través de modelos de aprendizaje automático

Diego Andrés Baquero Tibocho

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá, Colombia

2023

Implementación de un prototipo de software para predecir complicaciones en pacientes en unidad de cuidados intensivos pediátricos (UCIP), a través de modelos de aprendizaje automático

Diego Andrés Baquero Tibocho

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

Magister en Ingeniería de Sistemas y Computación

Director:

Ph.D. Luis Fernando Niño Vásquez

Codirectora:

MD, MsC. Ledys María Izquierdo Borrero

Línea de Investigación:

Sistemas inteligentes

Grupo de Investigación:

Laboratorio de Investigación en Sistemas Inteligentes - LISI

Universidad Nacional de Colombia

Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial

Bogotá, Colombia

2023

Dedicatoria

A mi esposa Stephania, quien es mi motor para construir este sueño que vivimos juntos cada día.

Se necesita confianza para ser humilde, para exhibir las debilidades.

Adam Grant

Declaración de obra original

Yo declaro lo siguiente:

He leído el Acuerdo 035 de 2003 del Consejo Académico de la Universidad Nacional. «Reglamento sobre propiedad intelectual» y la Normatividad Nacional relacionada al respeto de los derechos de autor. Esta disertación representa mi trabajo original, excepto donde he reconocido las ideas, las palabras, o materiales de otros autores.

Cuando se han presentado ideas o palabras de otros autores en esta disertación, he realizado su respectivo reconocimiento aplicando correctamente los esquemas de citas y referencias bibliográficas en el estilo requerido.

He obtenido el permiso del autor o editor para incluir cualquier material con derechos de autor (por ejemplo, tablas, figuras, instrumentos de encuesta o grandes porciones de texto).

Por último, he sometido esta disertación a la herramienta de integridad académica, definida por la universidad.



Diego Andrés Baquero Tibocho

Fecha 28/01/2023

Agradecimientos

Al profesor Luis Fernando Niño, por su constante apoyo y guía en el desarrollo de esta investigación y quien ha demostrado en estos años ser un excelente maestro, y mejor aún, un ser humano excepcional, al que le debo toda mi admiración y respeto.

A la doctora Ledys, quien ha compartido su invaluable conocimiento en el área médica para que este trabajo pudiese salir a la luz y ser expuesto a tantos profesionales que buscan, al igual que nosotros, usar la tecnología como herramienta para mejorar la calidad de vida de las personas.

A cada uno de los integrantes del grupo LISI por compartir sus conocimientos y brindar una sincera retroalimentación a esta investigación. En especial a Lida Méndez con quien compartimos en tantos espacios y trabajamos juntos para abordar el problema desde diferentes caminos.

A mi esposa Stephania, quien siempre ha confiado en mis decisiones y me ha apoyado incondicionalmente por años. La persona que con amor y paciencia ha estado junto a mí para verme cumplir mis sueños, de esos que hablamos cuando fuimos adolescentes.

A mi madre, quien estuvo junto a mí en las traspasadas y madrugadas. La persona que me ha dado sus palabras de aliento para seguir adelante en cada obstáculo de mi vida, quien siempre reza por mí, a la que le debo la vida y el ser humano que soy.

A mi hermano, quien me enseñó que la educación es un valor preciado al que pocos tenemos derecho, y que la mejor forma de agradecer este regalo es dándole a la sociedad un profesional que trabaje por el bienestar de los demás.

Resumen

Implementación de un prototipo de software para predecir complicaciones en pacientes en unidad de cuidados intensivos pediátricos (UCIP), a través de modelos de aprendizaje automático

Los cuidadores de pacientes en estado crítico no siempre tienen las habilidades o la experiencia para tratar este tipo de pacientes (Wheatley, 2006). Además, el deterioro fisiológico se puede detectar a partir de cambios sutiles en los signos vitales dentro de una Unidad de Cuidados Intensivos Pediátricos (UCIP) (Izquierdo, 2021). Esto conlleva dificultades para el personal médico al realizar un pronóstico sobre una futura complicación.

De acuerdo con lo anterior, este estudio tiene como objetivo implementar un prototipo de software capaz de predecir estados fisiológicos a través de los signos vitales, siguiendo como metodología el Proceso de Aprendizaje Automático (*Machine Learning Process*, MLP) sobre el conjunto de datos seleccionado. El prototipo se implementó de forma exitosa y se obtuvieron resultados prometedores en cuanto al uso de técnicas de aprendizaje automático para representar el estado actual y futuro de los pacientes en UCIP. Por lo tanto, se debe seguir trabajando en el esfuerzo de complementar el conjunto de datos e implementar nuevas propuestas del uso de las técnicas de aprendizaje automático, y así lograr una monitorización constante sobre los pacientes.

Palabras clave: Signos vitales, aprendizaje automático, predicción, estado clínico, cuidado intensivo pediátrico.

Abstract

Implementation of a software prototype to predict complications in patients in a pediatric intensive care unit (PICU), through machine learning models

Caregivers of critically ill patients do not always have the skills or experience to treat these types of patients (Wheatley, 2006). Furthermore, physiological deterioration can be detected from subtle changes in vital signs within a Pediatric Intensive Care Unit (PICU) (Izquierdo, 2021). This leads to difficulties for medical personnel when making a prognosis about a future complication.

Accordingly, this study aims to implement a software prototype capable of predicting physiological states through vital signs, following the Machine Learning Process (MLP) methodology on the selected data set. The prototype was successfully implemented, and promising results were obtained regarding the use of machine learning techniques to represent the current and future state of PICU patients. Therefore, work must continue in the effort to complement the data set and implement new methods for the use of machine learning techniques, and thus achieve constant monitoring of patients.

Keywords: Vital signs, machine learning, prediction, clinical status, pediatric intensive care.

Este Trabajo Final de maestría fue calificado en marzo de 2023 por la siguiente
evaluadora:

Elizabeth León Guzmán PhD.
Profesora Facultad de Ingeniería
Universidad Nacional de Colombia

Contenido

	Pág.
Resumen	IX
Lista de figuras	XIII
Lista de tablas	XIV
Lista de Símbolos y abreviaturas	XV
1. Estado del Arte	3
1.1 Monitorización de signos vitales	3
1.2 Signos vitales	4
1.2.1 Frecuencia cardiaca (FC)	4
1.2.2 Frecuencia respiratoria (FR)	5
1.2.3 Presión arterial (TA)	6
1.2.4 Saturación periférica de oxígeno (SpO ₂)	7
1.2.5 Temperatura (ToC)	7
1.3 Aprendizaje automático	8
1.4 Trabajos relacionados	9
1.4.1 <i>Long short-term memory</i> (LSTM)	14
1.4.2 <i>Random Forest</i> (RF)	14
1.4.3 <i>Support Vector Machine</i> (SVM)	15
1.5 Justificación	16
2. Metodología	18
2.1 Muestreo y tipo de instrumento	18
2.2 Diseño de la investigación	19
2.3 Procedimiento	19
3. Preparación y análisis exploratorio de datos	21
3.1 Limpieza de datos	21
3.1.1 Eliminar datos duplicados	21
3.1.2 Modificar unidades de medida	22
3.1.3 Administrar datos nulos	22
3.1.4 Cambiar notación	22
3.1.5 Modificar valores inválidos	23
3.1.6 Modificar valores atípicos	23
3.1.7 Cambiar tipo de variable	23
3.2 Análisis exploratorio de datos	24

3.2.1 Matriz de correlación	26
4. Modelos de aprendizaje automático implementados	29
4.1 Clasificación de si sobrevive o no después de la estancia en UCIP, según los signos vitales	32
4.1.1 Clasificador basado en árboles de decisión (DTC)	32
4.1.2 Clasificador basado en un bosque aleatorio (RFC).....	34
4.2 Clasificación del estado actual del paciente, según los signos vitales	35
4.2.1 Clasificador basado en árboles de decisión (DTC)	40
4.2.2 Clasificador basado en un bosque aleatorio (RFC).....	42
4.2.3 Clasificador basado en regresión logística (LRC)	44
4.2.4 Clasificador en datos con categorización	45
4.3 Predicción de estados futuros, según los signos vitales	46
4.3.1 <i>Long short-term memory</i> (LSTM).....	49
4.3.2 <i>Gated recurrent unit</i> (GRU).....	50
5. Prototipo de software.....	52
5.1 Frontend.....	52
5.2 Backend.....	55
5.3 Model API	57
5.4 Diagrama de secuencia.....	58
5.5 Diagrama de despliegue.....	59
6. Discusión y análisis de resultados	62
7. Conclusiones y recomendaciones.....	65
7.1 Conclusiones.....	65
7.2 Recomendaciones.....	66
7.2.1 Trabajos futuros.....	67
A. Anexo: Matrices de confusión de los modelos especializados	68
B. Anexo: Resultados de predicciones para cada uno de los pacientes.....	73
C. Anexo: Enlace a la aplicación y repositorios del código fuente	80
Bibliografía	81

Lista de figuras

	Pág.
Figura 1-1	10
Figura 2-1	19
Figura 3-1	22
Figura 3-2	25
Figura 3-3	26
Figura 3-4	27
Figura 3-5	28
Figura 4-1	30
Figura 4-2	31
Figura 4-3	33
Figura 4-4	34
Figura 4-5	35
Figura 4-6	37
Figura 4-7	38
Figura 4-8	39
Figura 4-9	46
Figura 4-10	47
Figura 4-11	48
Figura 4-12	48
Figura 4-13	49
Figura 4-14	51
Figura 5-1	52
Figura 5-2	53
Figura 5-3	54
Figura 5-4	54
Figura 5-5	55
Figura 5-6	57
Figura 5-7	59
Figura 5-8	61

Lista de tablas

	Pág.
Tabla 1 Rangos de normalidad de FC, por edad	5
Tabla 2 Rangos de normalidad de FR, por edad	6
Tabla 3 Rangos de normalidad de TAS y TAD, por edad.....	7
Tabla 4 Rangos de normalidad de ToC normales	8
Tabla 5 Resumen de trabajos fuertemente relacionados	11
Tabla 6 Estadísticos básicos.....	24
Tabla 7 Cantidad de datos según el estado, por grupo.....	36
Tabla 8 Exactitud de los modelos DTC, RFC, KNN y MPC; por grupo.....	37
Tabla 9 Resultados de DTC para el grupo 1	41
Tabla 10 Resultados de DTC para el grupo 2	41
Tabla 11 Resultados de DTC para el grupo 3	41
Tabla 12 Resultados de DTC para el grupo 4	42
Tabla 13 Resultados de RFC para el grupo 1	43
Tabla 14 Resultados de DTC para el grupo 2	43
Tabla 15 Resultados de RFC para el grupo 3	43
Tabla 16 Resultados de RFC para el grupo 4	44
Tabla 17 Resultados de LRC para el grupo 1	44
Tabla 18 Resultados de LRC para el grupo 2	44
Tabla 19 Resultados de LRC para el grupo 3	45
Tabla 20 Resultados de LRC para el grupo 4	45
Tabla 21 Resultados de LSTM para los 4 grupos.....	50
Tabla 22 Resultados de GRU para los 4 grupos	51

Lista de Símbolos y abreviaturas

Abreviaturas

Abreviatura	Término
<i>UCIP</i>	Unidad de cuidados intensivos pediátricos
<i>UCI</i>	Unidad de cuidados intensivos
<i>LISI</i>	Laboratorio de Investigación en Sistemas Inteligentes
<i>MLP</i>	Machine Learning Process (Proceso de aprendizaje automático)
<i>HMM</i>	Hidden Markov Models (Modelos Ocultos de Márkov)
<i>ML</i>	Machine Learning (Aprendizaje automático)
<i>AI</i>	Artificial Intelligence (Inteligencia artificial)
<i>FC</i>	Frecuencia cardíaca
<i>FR</i>	Frecuencia respiratoria
<i>SpO2</i>	Saturación periférica de oxígeno
<i>ToC</i>	Temperatura corporal
<i>TA</i>	Presión arterial
<i>TAS</i>	Presión arterial sistólica
<i>TAD</i>	Presión arterial diastólica
<i>TAM</i>	Presión arterial media
<i>CRISP-DM</i>	CRoss-Industry Standard Process for Data Mining
<i>LSTM</i>	Long short-term memory
<i>RF</i>	Random Forest
<i>SVM</i>	Support Vector Machine
<i>ANN</i>	Artificial Neural Network
<i>MIMIC-3</i>	Medical Information Mart for Intensive Care III
<i>VP</i>	Verdaderos positivos
<i>FP</i>	Falsos positivos
<i>VN</i>	Verdaderos negativos
<i>FN</i>	Falsos negativos
<i>DTC</i>	Decision Tree Classifier
<i>RFC</i>	Random Forest Classifier
<i>KNN</i>	K-Nearest-Neighbor
<i>MPC</i>	Multilayer Perceptron Classifier
<i>LRC</i>	Linear Regression Classifier
<i>API</i>	Application Programming Interfaces
<i>ORM</i>	Object-Relational mapping
<i>AWS</i>	Amazon Web Services
<i>GCP</i>	Google Cloud Platform

Introducción

Este estudio es una continuación del trabajo desarrollado en el grupo de investigación Laboratorio de Investigación en Sistemas Inteligentes (LISI), realizado por la doctora Ledys María Izquierdo Borrero, en la maestría en Ingeniería Biomédica en la Universidad Nacional de Colombia. En su trabajo se resalta la importancia de la monitorización continua de los signos vitales en entornos de cuidados intensivos, específicamente, en la unidad de cuidados intensivos pediátricos (UCIP). Particularmente, en su estudio, se presenta una implementación de un modelo predictivo para la detección de anomalías de manera temprana a través de modelos ocultos de Márkov (*Hidden Markov Models*, HMM).

Los cuidadores de pacientes en estado crítico no siempre tienen las habilidades o la experiencia para tratar este tipo de pacientes (Wheatley, 2006). Aunque se cuenta con diferentes tipos de monitores, el proceso de monitorización no es una labor sencilla para algunos integrantes del personal de la salud. Por otro lado, el deterioro fisiológico se puede detectar a partir de cambios sutiles en los signos vitales dentro de UCIP (Izquierdo, 2021). La dificultad en la comprensión de la información representada a través de los monitores de los signos vitales o la imposibilidad de realizar un pronóstico sobre un posible deterioro fisiológico podría desembocar en un estado crítico y hasta en la muerte del paciente (Izquierdo, 2021).

De acuerdo con lo anterior, es necesario mejorar la capacidad de predicción de un estado fisiológico futuro, que optimice el proceso de monitorización de los signos vitales. Este esfuerzo es motivado por la siguiente pregunta de investigación: *¿cómo se puede crear algún software que pueda alertar al personal médico, de forma oportuna, un posible deterioro fisiológico, y así evitar estados críticos en la salud del paciente?*

En el desarrollo de este trabajo se implementan diferentes técnicas de aprendizaje automático sobre datos ya existentes, los cuales fueron recolectados con la previa

autorización del comité de ética en investigación del Hospital Militar Central, acta número 013-164-19.

En este estudio se presenta el desarrollo de un prototipo de software capaz de clasificar los signos vitales de un paciente en los estados fisiológicos: crítico, inestable y estable; en una ventana de tiempo determinada. Para el desarrollo de estos modelos se usó como metodología el proceso de aprendizaje automático. Puesto que, el uso del aprendizaje automático en medicina podría ser una herramienta valiosa para el personal médico al permitir la creación de algoritmos que calculen estados fisiológicos de los pacientes en UCIP.

Este documento está estructurado en siete capítulos, de acuerdo con el desarrollo cronológico de este trabajo, a partir de la metodología propuesta, y está organizado de la siguiente manera:

- Capítulo 1. Estado del arte: Describe estudios realizados previamente y expone la justificación para el desarrollo de esta investigación.
- Capítulo 2. Metodología: Se presenta el muestreo y el tipo de instrumento, el diseño de la investigación y el procedimiento establecido para la ejecución de este estudio.
- Capítulo 3. Preparación y análisis exploratorio de datos: Se analiza el conjunto de datos y se aplican técnicas de ciencia de datos para comprender y preparar los datos para los modelos de aprendizaje automático.
- Capítulo 4. Modelos de aprendizaje automático implementados: Presenta los diferentes modelos de aprendizaje automático implementados para la clasificación y predicción de un estado fisiológico a partir de los signos vitales.
- Capítulo 5. Prototipo de software: Expone los componentes de software desarrollados para la ejecución de los objetivos propuestos de este trabajo.
- Capítulo 6. Discusión y análisis de resultados: Muestra un análisis de los resultados obtenidos en la implementación de la plataforma.
- Capítulo 7. Conclusiones y recomendaciones: Se presentan las conclusiones de este estudio, se exponen algunas recomendaciones generales y se describen algunos trabajos futuros que pueden dar continuidad al presentado en este documento.

1. Estado del Arte

En este capítulo se tratarán temas relacionados con la monitorización de signos vitales. Se hace una revisión de algunos trabajos relacionados en la literatura científica que se han publicado relacionados con esta temática, donde se presentan diferentes enfoques, técnicas y sistemas de monitorización de los signos vitales basados en aprendizaje automático.

1.1 Monitorización de signos vitales

Los monitores de signos vitales son dispositivos que cumplen con la tarea de amplificar, procesar, registrar y representar señales biológicas. Dichas señales tienen una importancia clínica para evaluar el estado actual o la evolución en función del tiempo de un paciente. Además, la monitorización, en este contexto, es el proceso de observación de la representación de bioseñales en el monitor y evaluación de una situación clínica particular, tarea llevada a cabo por el personal médico (Izquierdo, 2021).

La monitorización de pacientes es fundamental para el personal médico en ambientes hospitalarios, especialmente, en unidades de cuidados intensivos (UCI), donde los cuidadores de pacientes en estado crítico requieren saber medir los signos vitales, interpretarlos y tomar decisiones con base en estos (Izquierdo, 2021).

En cuanto a la evolución de los monitores, por un lado, se busca que cada vez sean más precisos en la muestra de bioseñales, con el fin de convertirlos en dispositivos más fiables —en efecto, cada vez son aparatos más seguros y precisos—. Por otro lado, se debe seguir en la búsqueda de otras variables que permitan identificar el estado actual del paciente de forma precisa. Sin embargo, es responsabilidad del personal médico seguir las indicaciones e interpretar los resultados en pro del bienestar del paciente, es decir, el

monitor, por sí solo, no reemplazará a los cuidadores de pacientes en estado crítico (Izquierdo, 2021).

Un monitor de signos vitales recibe los datos del paciente a través de cables por sus módulos amplificadores, los cuales son interfaces de entrada al dispositivo. Posteriormente, las bioseñales son procesadas a través del dispositivo y representada a través de su pantalla, donde puede ilustrar la señal en forma de onda o numéricamente (Izquierdo, 2021).

1.2 Signos vitales

Anteriormente, en este documento se ha hablado sobre los signos vitales, aunque no se ha profundizado en este tema. Los monitores de signos vitales permiten la representación de bioseñales y, a su vez, posibilitan un continuo seguimiento sobre el estado de los pacientes. Entre los signos vitales más comúnmente monitoreados se encuentran la frecuencia cardiaca (FC), la frecuencia respiratoria (FR), la presión arterial (TA), la saturación de oxígeno periférica (SpO_2) y la temperatura corporal (ToC) (Izquierdo, 2021).

Los signos vitales permiten al personal médico conocer el estado del paciente y evaluar si está respondiendo bien a un tratamiento o si su salud se deteriora; además, ayudan a la detección de posibles complicaciones relacionadas con insuficiencias cardiacas o respiratorias e infecciones, entre otros, que si no son tratados a tiempo, podrían producir resultados fatales (Izquierdo, 2021).

1.2.1 Frecuencia cardiaca (FC)

Representa la cantidad de pulsaciones del latido cardiaco en un instante de tiempo, generalmente por minuto. En otras palabras, es una onda en la sangre ocasionada por la contracción del ventrículo izquierdo del corazón, que ocasiona la expansión o contracción de las arterias. Las anomalías en la frecuencia cardiaca son llamadas arritmias y pueden ser bradicardias y taquicardias, cuando el ritmo cardiaco se encuentra por debajo o por encima de los rangos normales, respectivamente (Izquierdo, 2021).

Los rangos de normalidad pueden variar según la edad en pacientes pediátricos. En la siguiente tabla se representan estos rangos de edades y rangos en donde la frecuencia cardiaca se considera normal.

Tabla 1

Rangos de normalidad de FC, por edad

Rango de edad (años)	Rangos normales (latidos/min)
0-1	100-160
1-3	95-150
4-5	80-140
6-12	70-120
13-18	60-100

Nota. Adaptado de *Modelamiento del espacio de signos vitales para detectar el deterioro de los pacientes en una unidad de cuidados intensivos*, de Izquierdo, L. M., Niño, L. F., 2021. *Universidad Nacional de Colombia*

1.2.2 Frecuencia respiratoria (FR)

Está dada por la cantidad de respiraciones en un instante de tiempo, generalmente en un minuto. El proceso de respiración está contemplado por dos fases: inhalación y exhalación. Las alteraciones en la frecuencia respiratoria son llamadas bradipnea, polipnea y taquipnea, cuando esta se encuentra por debajo o por encima de los rangos normales, respectivamente (Izquierdo, 2021).

Los rangos de normalidad pueden variar según la edad. En la tabla 2 se representan estos rangos de edades y rangos en donde la frecuencia respiratoria es considerada normal.

Tabla 2*Rangos de normalidad de FR, por edad*

Rango de edad (años)	Rangos normales (respiraciones/min)
0-1	30-60
1-3	24-40
4-5	22-34
6-12	18-30
13-18	12-16

Nota. Adaptado de *Modelamiento del espacio de signos vitales para detectar el deterioro de los pacientes en una unidad de cuidados intensivos*, de Izquierdo, L. M., Niño, L. F., 2021. Universidad Nacional de Colombia

1.2.3 Presión arterial (TA)

Presión que ejerce la sangre sobre las paredes de las arterias; varía con respecto a los latidos del corazón y se divide en dos: presión arterial sistólica (TAS) y presión arterial diastólica (TAD). La TAS corresponde a la presión máxima sobre las paredes arteriales y ocurre con la contracción de los ventrículos y la TAD es la presión mínima, consecuencia de la relajación de los ventrículos. Además, la presión arterial media (TAM) corresponde a la presión arterial promedio y se puede calcular con la siguiente fórmula (Izquierdo, 2021):

$$TAS = \frac{TAS + 2TAD}{3} \quad (1.1)$$

Al igual que los signos vitales mencionados anteriormente, los rangos de normalidad están dados entre rangos, dependiendo de la edad.

Tabla 3

Rangos de normalidad de TAS y TAD, por edad

Rango de edad (años)	Rangos normales TAS (mmHg)	Rangos normales TAD (mmHg)
0-1	60-90	30-62
1-3	78-112	48-78
4-5	85-114	52-85
6-12	95-135	58-88
De 13 en adelante	100-120	60-80

Nota. Adaptado de *Modelamiento del espacio de signos vitales para detectar el deterioro de los pacientes en una unidad de cuidados intensivos*, de Izquierdo, L. M., Niño, L. F., 2021. Universidad Nacional de Colombia

1.2.4 Saturación periférica de oxígeno (SpO₂)

Cantidad de oxígeno transportado a través de la sangre por la hemoglobina, medida a través de un oxímetro de pulso. El nivel de saturación se da en porcentaje y debe ser superior al 88 % en Bogotá, debido a la altura. Por debajo de este valor, se considera la presencia de hipoxemia (Izquierdo, 2021).

1.2.5 Temperatura (ToC)

Equilibrio de la pérdida y producción de calor en el cuerpo; puede variar en función del sexo, la actividad y el consumo de alimentos, entre otros. Las alteraciones en la temperatura son llamadas hipotermia e hipertermia, cuando esta se encuentra por debajo o por encima de los rangos normales, respectivamente (Izquierdo, 2021). Los rangos de temperatura se clasifican como se muestra en la tabla 4.

Tabla 4*Rangos de normalidad de ToC normales*

Temperatura	Descripción
Inferior a 35 °C	Hipotermia
35°-37 °C	Normal
37.1°-38 °C	Febrícula
38.1°-39 °C	Fiebre moderada
De 39.1° en adelante	Fiebre alta

Nota. Adaptado de *Modelamiento del espacio de signos vitales para detectar el deterioro de los pacientes en una unidad de cuidados intensivos*, de Izquierdo, L. M., Niño, L. F., 2021. Universidad Nacional de Colombia

Los monitores de signos vitales, generalmente, son capaces de medir con bastante precisión las variables mencionadas anteriormente. Esto permite que el personal médico pueda evaluar la evolución o deterioro de un paciente en un instante de tiempo con dicha información (Izquierdo, 2021).

1.3 Aprendizaje automático

El aprendizaje automático, en inglés *Machine Learning* (ML), es una rama de la inteligencia artificial (AI) que se centra en imitar el aprendizaje humano a través del uso de datos, estrategias y algoritmos (muchas veces basados en métodos estadísticos), con el fin de mejorar la precisión de clasificaciones o predicciones en un conjunto de datos diferente a aquel con el que fue entrenado (IBM, 2020).

Según el doctor Mike Tamir, de la Universidad de Berkeley, un sistema basado en aprendizaje automático se desarrolla en tres etapas:

- **Proceso de decisión:** generalmente, los algoritmos de aprendizaje automático reciben una entrada de datos que pueden estar etiquetados o no; a partir de estos, se puede dar una estimación de un patrón en los datos. Típicamente, los modelos de aprendizaje automático tienen como objetivo realizar una clasificación o una predicción basada en los datos de entrada.

- **Evaluación de la función de error:** usada para validar la precisión de clasificar o predecir de un modelo de aprendizaje automático y se evalúa tomando un conjunto de datos diferente a los datos del entrenamiento y comparando la salida del modelo con la esperada.
- **Proceso de optimización:** en esta etapa se realizan ajustes en la implementación del modelo; se varían diferentes tipos de meta-parámetros, incluso la arquitectura del modelo, con el objetivo de mejorar la precisión.

Todo el proceso anterior se debe repetir tantas veces como sea necesario para mejorar el rendimiento del modelo (Tamir, 2020).

1.4 Trabajos relacionados

La monitorización constante se ha convertido en un punto clave para ayudar a identificar factores de riesgo en la salud de un paciente. Debido a lo anterior, se han desarrollado una serie de estudios que buscan detectar oportunamente complicaciones, a través del uso de algoritmos y modelos estadísticos, basados en sistemas inteligentes.

Para el desarrollo de esta investigación, se realizó una búsqueda sistemática de literatura y un proceso de selección de cada una de las investigaciones con base en las preguntas orientadoras. La búsqueda de material bibliográfico se hizo principalmente en SCOPUS e IEEE, entre 2016 y 2021.

Figura 1-1

Proceso de selección de fuentes



Nota. Fuente: Elaboración propia

Se realizó un análisis sobre la lista filtrada con base en la metodología y resultados obtenidos en las investigaciones, donde se destaca el uso de métodos de aprendizaje automático en unidades de cuidados intensivos. Se obtuvieron 10 estudios que se consideraron pertinentes por su afinidad con la implementación del modelo que se desea realizar.

El primero de estos estudios, desarrollado por Ledys María Izquierdo Borrero, en su trabajo Modelamiento del espacio de signos vitales para detectar el deterioro de los pacientes en una unidad de cuidados intensivos, es un trabajo precursor de esta investigación. En su trabajo, la doctora Izquierdo plantea que un deterioro fisiológico temprano puede no ser detectado a tiempo debido a las limitaciones por parte del equipo médico y a la cantidad de información recolectada a través de los monitores (Izquierdo, 2021).

En la investigación mencionada, se utilizó la metodología CRISP-DM (*CRoss-Industry Standard Process for Data Mining*) como proceso de minería de datos. Posteriormente, se usaron cadenas de Markov para identificar los estados clínicos por los que pasa el paciente. Finalmente, se implementó un modelo oculto de Markov (*Hidden Markov Model*, HMM) para la clasificación y predicción de los deterioros de un paciente (Izquierdo, 2021).

A continuación, en la Tabla 5, se presenta de manera resumida el resto de los trabajos relacionados y el resultado de cada uno de ellos. Además, se profundizará sobre algunos de estos.

Tabla 5

Resumen de trabajos fuertemente relacionados

No.	Año	Autores	Documento	Método	Resultados
1	2019	Fagerström, J. Bång, M. Wilhelms, D. Chew, M.S.	LiSep LSTM: A Machine Learning Algorithm for Early Detection of Septic Shock	Uso de LSTM ¹ en la base de datos <i>Medical Information Mart for Intensive Care</i> (59,000 pacientes en UCI ²) para predecir posibles estados asociados al shock séptico.	Exactitud: 0,83
2	2020	Essay, P. Balkan, B. Subbian, V.	Decompensation in critical care: Early prediction of acute heart failure onset	Uso de RF ³ y regresión logística multivariable en la base de datos <i>tele-UCI</i> (cerca de 128,000 pacientes) para predecir descompensación e insuficiencia cardíaca.	Exactitud (RF): 0,95
3	2020	Li, X. Ge, P. Zhu, J. Li, H. Graham, J. Singer, A. Richman, P.S. Duong, T.Q.	Deep learning prediction of likelihood of ICU admission and mortality in COVID-19 patients using clinical variables	Uso de RF sobre los datos de 5766 personas bajo investigación para covid-19, para predecir futuros ingresos a UCI y mortalidad.	Exactitud: 0,78 (ingreso a UCI) Exactitud: 0,84 (mortalidad)

(continúa)

No.	Año	Autores	Documento	Método	Resultados
4	2020	Chiu, Y.-D. Villar, S.S. Brand, J.W. Patteril, M.V. Morrice, D.J. Clayton, J. Mackay, J.H.	Logistic early warning scores to predict death, cardiac arrest or unplanned intensive care unit re-admission after cardiac surgery	Uso de regresión logística en tiempo discreto en 13.631 pacientes dados de alta después de una cirugía cardíaca, para predecir muerte intrahospitalaria, ingreso a UCI e infarto cardíaco, en ventanas de 24 y 6 horas.	Exactitud: 0,78 (24 horas) Exactitud: 0.84 (6 horas)
5	2020	Hever, G. Cohen, L. O'Connor, M.F. Matot, I. Lerner, B. Bitan, Y.	Machine learning applied to multi-sensor information to reduce false alarm rate in the ICU	Uso de RF para la predicción de falsas alarmas, causadas por ausencia de parámetros (signos vitales) en monitores instalados en camas de UCI.	Exactitud: 0,94 – 0,97
6	2020	Yoon, J.H. Jeanselme, V. Dubrawski, A. Hravnak, M. Pinsky, M.R. Clermont, G.	Prediction of hypotension events with physiologic vital sign signatures in the intensive care unit	Permite hacer predicción sobre riesgo de hipotensión minuto a minuto en una ventana entre 15 y 60 minutos, con el uso de RF.	Exactitud: 0,93 (15 min) Exactitud: 0,88 (60 min)
7	2020	Monteiro, F. Meloni, F. Baranauskas, J.A. Macedo, A.A.	Prediction of mortality in Intensive Care Units: a multivariate feature selection	Predicción de mortalidad en pacientes en UCI. El modelo con mejor rendimiento fue SVM ⁴ .	Exactitud: 0,73

(continúa)

No.	Año	Autores	Documento	Método	Resultados
8	2021	Yin, Y.a, Chou, C.b	A Novel Switching State Space Model for Post-ICU Mortality Prediction and Survival Analysis	Modelos de regresión logística para predecir la probabilidad de supervivencia de un paciente en UCI.	Exactitud: 0,8
9	2021	Staziaki, P.V. Wu, D. Rayan, J.C. Santo, I.D.O. Nan, F. Maybury, A. Gangasani, N. Benador, I. Saligrama, V. Scalera, J. Scalera, J. Anderson, S.W.	Machine learning combining CT findings and clinical parameters improves prediction of length of stay and ICU admission in torso trauma	Se hizo predicción de complicaciones en pacientes, para predecir el ingreso a UCI y estadía prolongada en esta con traumatismo de torso. Los modelos usados fueron ANN ⁵ y SVM.	Exactitud (ANN): 0,87 Exactitud (SVM): 0,78

Nota. Fuente: Elaboración propia

¹ LSTM: Long short-term memory

² UCI: Unidad de Cuidados intensivos

³ RF: Random Forest

⁴ SVM: Support Vector Machine

⁵ ANN: Artificial Neural Network

Con el objetivo de presentar al lector de forma clara y precisa los trabajos relacionados que sustentan esta investigación, se hará una breve descripción de los estudios, ordenados según los modelos y algoritmos de clasificación empleados en cada uno de ellos. Esto será una base importante para la investigación que se propone.

1.4.1 Long short-term memory (LSTM)

En 2019, Fagerström, J. y colaboradores presentaron LiSep LSTM, una red neuronal *Long short-term memory* (LSTM), para la identificación temprana de un shock séptico. Las LSTM suelen ser adecuadas para detectar dependencias entre variables en series temporales (Fagerström et al., 2019). El modelo fue entrenado con cerca de 59.000 pacientes de UCI, de la base de datos *Medical Information Mart for Intensive Care*, que contiene signos vitales, datos de laboratorio y entradas de diario, entre otros.

Para calcular el rendimiento de las diferentes técnicas aplicadas, se usó validación cruzada de seis pliegues (*6-fold cross validation*). Es decir, se dividió el conjunto de datos (*dataset*) en varios subconjuntos: el 5 % como datos de validación y el 95 % restante se dividió en 6 partes iguales, 5 para entrenamiento y 1 para prueba (test). Finalmente, la precisión obtenida por el modelo seleccionado fue de 0,8306, con un intervalo de confianza del 95 %.

1.4.2 Random Forest (RF)

En 2020, se desarrollaron varias investigaciones relacionadas con predicción de complicaciones en UCI implementando *Random Forest* (RF), en español bosque aleatorio, para la predicción de mortalidad e ingreso a UCI, entre otros.

Essay y colaboradores trabajaron en el estudio de la viabilidad de poder predecir una insuficiencia cardiaca, poco tiempo después del ingreso en la UCI. Este estudio se realizó a través de 96.350 pacientes ingresados sin factores de riesgo y 26.534 pacientes sin antecedentes de insuficiencia cardiaca, pero con factores de riesgo específicos de insuficiencia cardiaca, entre éstos: enfermedad arterial coronaria, hipertensión e infarto de miocardio (Essay et al., 2019), tomados de tele-UCI, que cuenta con el registro de más de 200 hospitales de Estados Unidos. Los resultados obtenidos tuvieron una precisión de 0,951 en una ventana de tiempo de cerca de 200 minutos, usando *Random Forest* (RF) y una combinación de signos vitales, valores de laboratorio y mediciones fisiológicas discretas (Essay et al., 2020).

Por otro lado, Li et al. (2020) basaron su estudio en la construcción de un modelo que permitiera predecir el ingreso en UCI y mortalidad intrahospitalaria, según los datos de 5.766 personas bajo investigación por Covid-19. Entre los datos recolectados, se presenta

información demográfica, comorbilidades crónicas, signos vitales, síntomas y pruebas de laboratorio al ingreso. En cuanto a los resultados, se obtuvo una precisión de 0,78 y 0,848 para la predicción del ingreso en la UCI y la mortalidad, respectivamente.

Por otro lado, Hover y sus colaboradores mencionaron un problema asociado a la precisión en los monitores de signos vitales en unidades de cuidados intensivos, ya que, según los autores, pueden generar una tasa de falsas alarmas que oscila entre 0,72 y 0,99 (Hover et al., 2020). Por lo tanto, aplicaron ML a la información recolectada por los multisensores encargados de registrar los signos vitales de los pacientes con el objetivo de reducir falsas alarmas en ausencia de uno a tres signos vitales (por ejemplo, la frecuencia cardiaca o la presión arterial), en un conjunto de datos de siete escenarios de alarma clínica. Se comparó la implementación con RF y un algoritmo basado en expertos; RF mostró una clara superioridad, en casi todas las combinaciones de escenarios y números de parámetros ausentes, con una precisión entre 0,94 y 0,97 (Hover et al., 2020).

Finalmente, Yoon et al. (2020) desarrollaron un modelo de aprendizaje automático para predecir el evento de hipotensión inicial en pacientes de una UCI, y diseñaron un sistema de alerta que se ubicaría junto a la cama. Para esta investigación se usó el conjunto de datos *Medical Information Mart for Intensive Care III* (MIMIC-3), de donde se obtuvieron los signos vitales minuto a minuto. El modelo RF mostró una precisión de 0,93 y 0,88 a los 15 y 60 minutos, respectivamente.

1.4.3 Support Vector Machine (SVM)

Monteiro et al. (2020) plantearon el problema de la cantidad de camas con las que debe contar una UCI, teniendo las implicaciones éticas y financieras; es decir, se trata de maximizar la cantidad de camas UCI, minimizando la cantidad de costos asociados. La iniciativa de esta investigación surgió de un reto creado por *PhysioNet/Computing in Cardiology* en 2012, que pretendía estimular el desarrollo de técnicas para predecir la mortalidad en las UCI, basándose en características biométricas y fisiológicas recogidas de los pacientes. El mejor clasificador de Monteiro et al. (2020) fue implementado con SVM, el cual tuvo como resultado 0,73 de precisión (mejor que en estudios anteriores, donde el mayor obtuvo una precisión del 0,55), con un intervalo de confianza del 99 %.

Por otro lado, Staziaki et al. (2020) desarrollaron un modelo capaz de predecir el ingreso a UCI y una duración prolongada de la estadía, después de un traumatismo en el torso (pecho, abdomen o pelvis), mediante datos clínicos o imágenes. La investigación se llevó a cabo sobre 840 pacientes ingresados con este tipo de traumatismo, en un lapso de un año, donde se tuvieron en cuenta la edad, el sexo, los signos vitales, las puntuaciones clínicas, los valores de laboratorio y los datos de imágenes relacionados con la lesión (Staziaki et al, 2020).

Para esta investigación se probaron modelos de redes neuronales artificiales (ANN) y *Support Vector Machine* (SVM). Por un lado, la exactitud (*accuracy*) de las implementaciones de ANN y SVM para el ingreso a UCI fueron de $0,87 \pm 0,03$ y $0,78 \pm 0,02$, respectivamente. Por otro lado, la precisión asociada con la predicción de estadías prolongadas en UCI para ANN y SVM fueron de $0,8 \pm 0,04$ y $0,81 \pm 0,05$, respectivamente.

Además, existen otros estudios que hacen uso de cadenas de Markov, regresiones lineales, regresiones logísticas y árboles de decisión, entre otros, para la predicción de eventos clínicos particulares.

Los estudios mencionados anteriormente denotan la viabilidad de una implementación basada en modelos de aprendizaje automático para la predicción de eventos clínicos con base en los signos vitales de los pacientes internados en una UCI. Además, la búsqueda de literatura permitió conocer algunos de los modelos de aprendizaje automático más usados para lograr los objetivos de esta investigación.

1.5 Justificación

Debido a las limitaciones relacionadas con el equipo médico y los grandes volúmenes de información que se recolectan a diario en una UCI, es pertinente llevar el proceso de monitorización a un estado en el cual se puedan predecir deterioros fisiológicos graves y actuar de forma oportuna para reducir la cantidad de estados críticos y de mortalidad.

El desarrollo del aprendizaje de máquina y su evolución en la última década por la capacidad computacional con la que se cuenta en la actualidad podría permitir hacer una predicción de estados futuros de los pacientes.

De esta manera, la aplicación de la inteligencia artificial ofrecería una propuesta de valor al área de la medicina, en la cual se podría mejorar el estilo de vida de algunos pacientes e incluso podría salvar su vida, si el personal médico cuenta con el conocimiento adecuado y toma decisiones de forma oportuna. Por lo anterior, el desarrollo de esta investigación podría llegar a ser un apoyo importante para el personal médico y pacientes en ambientes de cuidados intensivos.

2. Metodología

En este capítulo se presenta la descripción de la metodología de este estudio de manera conceptual, la cual está basada en el proceso de aprendizaje automático. Además, se menciona la fuente del conjunto de datos, el muestreo, el tipo de instrumento y el diseño de la investigación.

Vale añadir que en el capítulo 3 se presenta la preparación y el análisis exploratorio de los datos. Luego, en el capítulo 4 se muestra la implementación de los modelos de aprendizaje automático utilizados y los resultados obtenidos. Finalmente, en el capítulo 5 se expone el prototipo de software desarrollado para predecir estados fisiológicos de los pacientes en UCIP.

2.1 Muestreo y tipo de instrumento

Para el desarrollo de esta investigación, se ha seleccionado un conjunto de datos existente y disponible en Kaggle (<https://www.kaggle.com/ledysmai5/critically-illpediatric-patients-in-picucsv>). La doctora Ledys María Izquierdo Borrero realizó un estudio de cohorte transversal entre enero de 2018 y enero de 2020, en el cual realizó la recolección y almacenamiento de datos relacionados con los signos vitales de pacientes menores de 18 años que se encontraban hospitalizados en la UCIP del Hospital Militar Central en Bogotá (Colombia). La información recolectada cuenta con 94.678 instancias de 90 pacientes. Además, es importante mencionar que los datos se tomaron a través del tiempo en intervalos de 24 horas, minuto a minuto.

De acuerdo con lo mencionado anteriormente, esta investigación experimental contó con un muestreo probabilístico y por conveniencia. Además, el tipo de instrumento será un análisis secundario, debido a que esta investigación parte de un conjunto de datos existente.

2.2 Diseño de la investigación

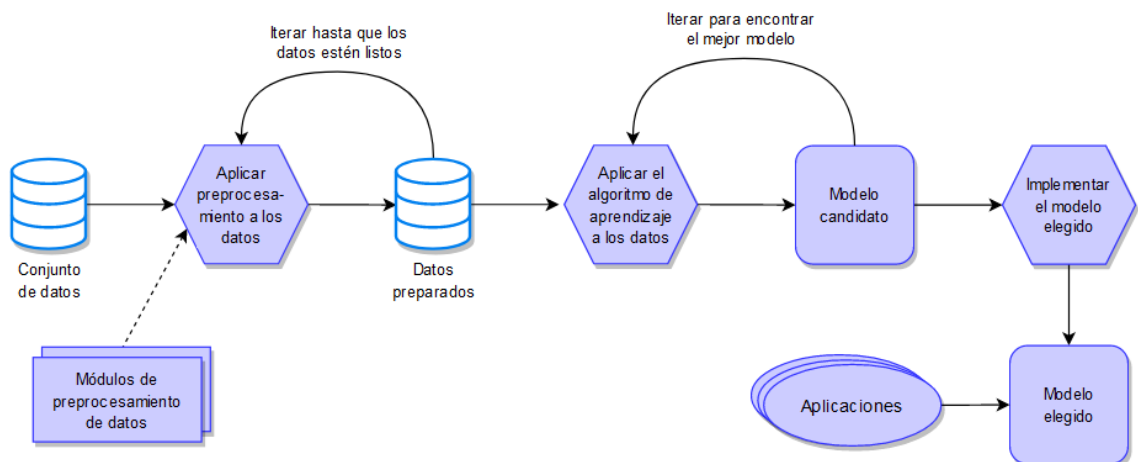
Con base en la naturaleza de los datos y los objetivos planteados en este estudio, se puede clasificar como una investigación de tipo cuantitativo y correlacional, donde se busca asociar variables, cuantificar su grado de correlación y realizar predicciones.

2.3 Procedimiento

Para el desarrollo de este estudio, se utilizará como guía el proceso de aprendizaje automático (*Machine Learning Process*, en inglés), debido a que es usado ampliamente como referencia para llevar a cabo proyectos relacionados con aprendizaje automático e inteligencia artificial.

Figura 2-1

Proceso de aprendizaje automático



Nota. Adaptado de Machine Learning Process. (s. f.). [Ilustración]. AnalystPrep.

(https://analystprep.com/study-notes/wp-content/uploads/2020/04/MachineLearningApproaches_img1.jpg)

El proceso de usar aprendizaje automático, por lo general, conlleva pasos adicionales al entrenamiento del modelo. Aunque en la bibliografía varían los pasos que se deben seguir para realizar una implementación de aprendizaje automático adecuado, Codecademy (2021) los resume en los siguientes:

1. Recolección de datos: se plantea una metodología para la recolección de datos o se realiza la selección de conjuntos de datos existentes que puedan servir para el desarrollo del proyecto.
2. Comprensión de los datos: cuando ya se han seleccionado o recolectado las fuentes de información, es necesario comprender los datos para elegir algunos modelos de aprendizaje de máquina candidatos. Generalmente, se realiza un proceso de estadística descriptiva y se representan los datos para comprender sus comportamientos.
3. Limpieza de datos: se les hace un tratamiento a los valores faltantes o atípicos en el conjunto de datos seleccionado.
4. Ingeniería de características: el investigador selecciona entre las variables del conjunto de datos las más relevantes y realiza las transformaciones que considere adecuadas para la preparación de los datos que entrarán al modelo.
5. Selección de modelos candidatos: dependiendo de la naturaleza del problema y de los datos, el investigador debe seleccionar uno o varios modelos de aprendizaje automático adecuados para lograr el objetivo propuesto.
6. Ajuste y evaluación: la evaluación de la implementación se debe realizar en el proceso del entrenamiento de un modelo basado en el conjunto de datos. A menudo se realiza a través de las matrices de confusión; sin embargo, es común encontrar que el modelo requiera un ajuste debido a los resultados obtenidos.
7. Selección y uso del modelo: posteriormente al logro de la precisión que el investigador considere adecuada, se selecciona un modelo (si se ha implementado más de uno) y se hace el despliegue de este, con el objetivo de usarlo para resolver el problema que motivó el proceso en cuestión.

3.Preparación y análisis exploratorio de datos

En este capítulo se describe el proceso desarrollado para la preparación de los datos previo a los procesos de análisis. Además, se expone un primer proceso de análisis realizado que fue el análisis exploratorio de datos. En el siguiente capítulo se presentará el proceso analítico realizado basado en aprendizaje automático.

3.1 Limpieza de datos

La limpieza de datos es el proceso de corregir o eliminar datos incorrectos, corruptos, con formato incorrecto, duplicados o incompletos dentro de un conjunto de datos.

Para el desarrollo de la implementación se realizó una limpieza exhaustiva de datos sin perder datos: (1) Eliminar datos duplicados, (2) Modificar unidades de medida, (3) Administrar datos nulos, (4) Cambiar notación, (5) Modificar valores atípicos, (6) Modificar valores incorrectos, y (7) Cambiar tipo de variable.

3.1.1 Eliminar datos duplicados

En el conjunto de datos seleccionado se observaron 803 datos duplicados. Por lo tanto, se deben eliminar, ya que esto puede ocasionar sobreajuste u otro tipo de sesgo en los modelos entrenados. Pandas provee una función para eliminar estos datos duplicados, como se ilustra en la Figura 3.1.

Figura 3-1

Eliminación de datos duplicados

```
[▶] print("Size", ucip.shape)
     ucip = ucip.drop_duplicates()
     print("New size", ucip.shape) # Print new size (no duplicates values)
```

```
↳ Size (94678, 15)
   New size (93875, 15)
```

Nota. Fuente: Elaboración propia

Además, se observó que seis pacientes del conjunto de datos tenían los mismos datos, por lo tanto, se decidieron eliminar.

3.1.2 Modificar unidades de medida

Al revisar el conjunto de datos, se observó que la edad estaba en diferentes unidades de medida (meses y años). Por lo tanto, se modificó el conjunto de datos a través de una función y así utilizar una sola unidad de medida (meses).

3.1.3 Administrar datos nulos

Se encontró que la altura tenía 7404 valores nulos. Por lo tanto, se escribió una función para calcular la estatura promedio entre los pacientes de una edad similar en un rango de 6 meses en relación con la edad del paciente.

3.1.4 Cambiar notación

Para facilitar el uso de los datos, se decidió cambiar la notación de las variables *Condition*, *Outcome* y *Gender*.

La variable *Outcome* hace referencia a si el paciente sobrevivió o no, después de su estancia en la UCIP. El valor *survived* fue reemplazado por 1 y *dead* fue reemplazado por 0.

La variable *Gender*, como su nombre lo indica, hace referencia al sexo del paciente, donde se clasifica con dos variables: masculino y femenino. Los valores fueron reemplazados de *female* y *male* a 0 y 1, respectivamente.

Finalmente, la variable *Condition* describe la variable del estado clínico actual del paciente: crítico, inestable y estable. El valor “Crítico” fue reemplazado por 0, “Inestable” por 1 y “Estable” por 2.

Es necesario mencionar que esta última variable no se encontraba en el conjunto de datos inicial, por lo tanto, se contactó a la autora para solicitar esta variable y fue incluida en el conjunto de datos.

3.1.5 Modificar valores inválidos

En algunos registros se encontraron valores inválidos para la variable SpO2. Dado que la variable es continua, se realizó una interpolación lineal entre los valores que estaban antes y después del valor incorrecto, para calcular su posible valor.

3.1.6 Modificar valores atípicos

Se observó que algunos valores relacionados con la frecuencia cardíaca tenían un valor de 400 (valor atípico). Así que se comprobó mediante el cálculo del posible valor que se trataba de un error tipográfico. El valor calculado fue 40 y se hizo la modificación correspondiente.

3.1.7 Cambiar tipo de variable

Al realizar el análisis exploratorio se observó que los valores de la variable SpO2 eran cadenas de texto, por lo tanto, se procedió a cambiar los valores a un tipo de dato de números enteros.

3.2 Análisis exploratorio de datos

Tras completar el proceso de limpieza de datos, se procedió a obtener información sobre algunas de las variables numéricas del conjunto de datos. Por lo tanto, se calcularon la media, desviación estándar, mínimo, máximo y percentiles (25%, 50% y 75%); de las variables: edad, peso, estatura, frecuencia cardíaca, frecuencia respiratoria, presión arterial sistólica, presión arterial diastólica y presión arterial media, para los 94.241 registros.

Tabla 6

Estadísticos básicos

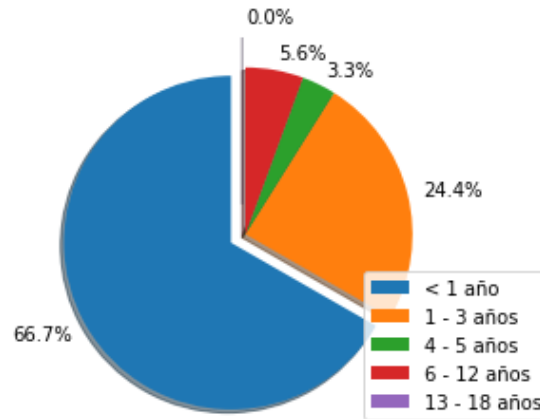
	Edad (meses)	Peso (kg)	Estatura (cm)	FC	SpO2	FR	TAS	TAD	TAM
Media	10,54	15,20	85,38	131,44	91,84	30,21	98,10	59,84	72,61
Desviación estándar	17,10	15,81	35,17	29,40	10,56	13,81	20,16	16,40	16,78
Mínimo	2	4	45	0	0	0	0	0	0
25%	3	5	55	118	90	23	87	49	61
50%	4	8	69,4	138	95	30	100	60	73
75%	11	13	100	151	97	34	110	72	85
Máximo	84	65	170	205	100	136	163	112	124

Nota. Fuente: Elaboración propia

Con base en la Tabla 6, se observó que la media era de 10,54 meses. Por lo tanto, se decidió analizar los valores relacionados con esta variable y el resultado fue que aproximadamente dos terceras partes del conjunto de datos correspondía a pacientes menores de un año. Además, es de anotar que el conjunto de datos no cuenta con signos vitales de pacientes en el rango de edades entre 13 y 18 años (ver Figura 3-2).

Figura 3-2

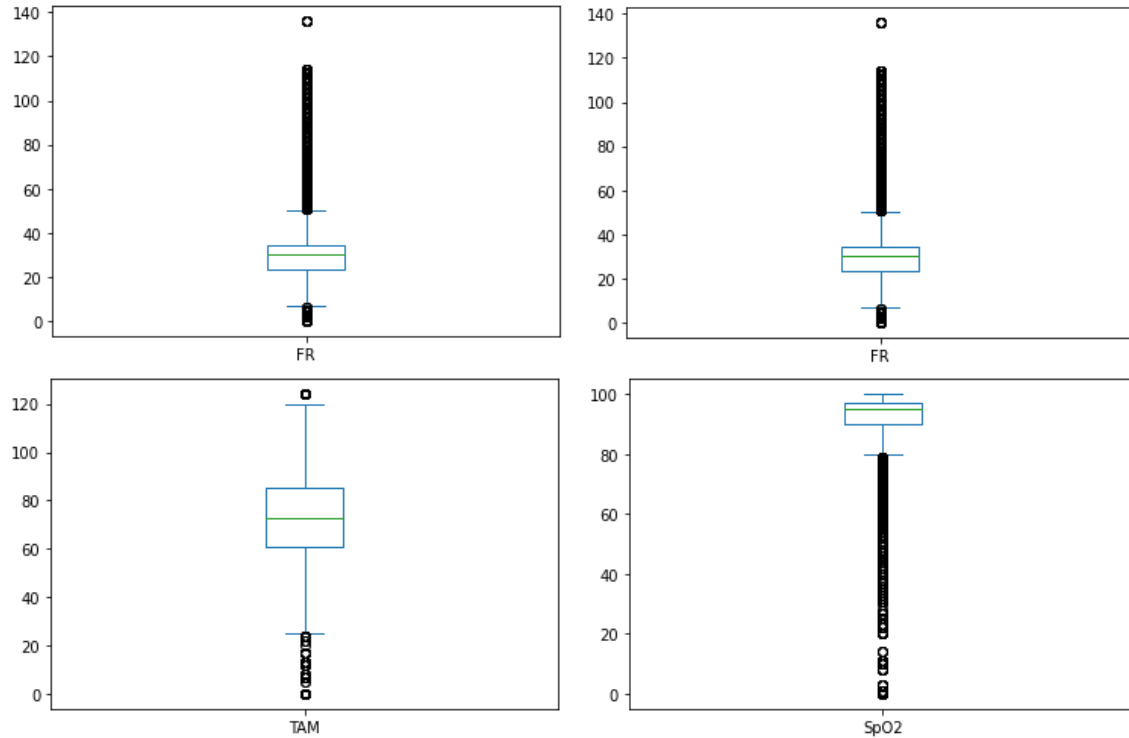
Distribución de edades (en años)



Nota. Fuente: Elaboración propia

Es necesario mencionar que se usó la variable en años para facilitar la visualización de la información y se separó en los grupos definidos en la literatura.

Asimismo, se analizaron los signos vitales para comprobar si los datos se encontraban en un rango de valores válidos, haciendo un contraste con la bibliografía seleccionada: La frecuencia cardíaca para niños menores de 1 año está entre 100 y 160. Esto está bien descrito dentro de la media calculada de 131 con una desviación estándar de 30. La frecuencia respiratoria para niños menores de un año debe oscilar entre 30 y 60. Se encontró que la media aproximada es 30 con una desviación de 14. Sin embargo, la media se encuentra en el límite inferior de los parámetros normales. La saturación periférica de oxígeno debe estar entre 95 y 100. Sin embargo, la media es 92, con una desviación estándar de 10.

Figura 3-3*Diagramas de cajas de los signos vitales**Nota.* Fuente: Elaboración propia

3.2.1 Matriz de correlación

Este tipo de matrices de correlación lineal se utilizan para medir la relación lineal entre diferentes variables numéricas, donde se relacionan cada par de variables del conjunto de datos a través de su coeficiente de correlación. Para este análisis se calcularon los valores de la correlación de Pearson entre cada par de variables.

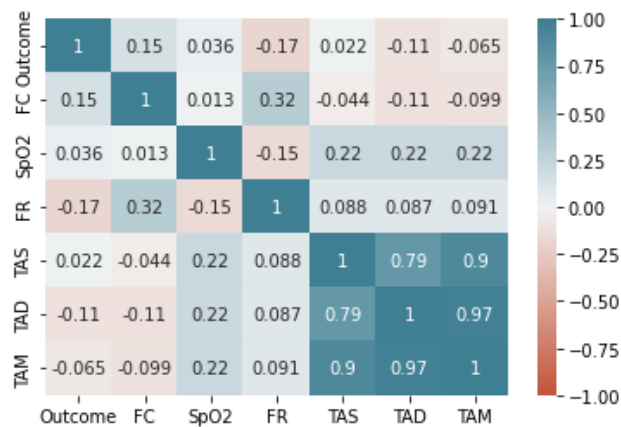
Para los coeficientes de correlación, un valor de 1 indica una correlación perfectamente positiva, lo que significa que son directamente proporcionales. Un valor de -1 indica una correlación perfectamente negativa, que implica que son inversamente proporcionales. Finalmente, un valor de cero indica que no hay relación lineal entre las dos variables (Platzi, 2018).

Se debe tener en cuenta este grado de correlación entre las variables para: (a) entender mejor el comportamiento de los datos y seleccionar las variables más relevantes, (b) eliminar variables redundantes en el conjunto de datos, e (c) identificar multicolinealidad entre más de un par de variables.

Con el objetivo de encontrar la relación lineal que existía entre los signos vitales, se graficó la correspondiente matriz de correlación como se muestra en la Figura 3-4.

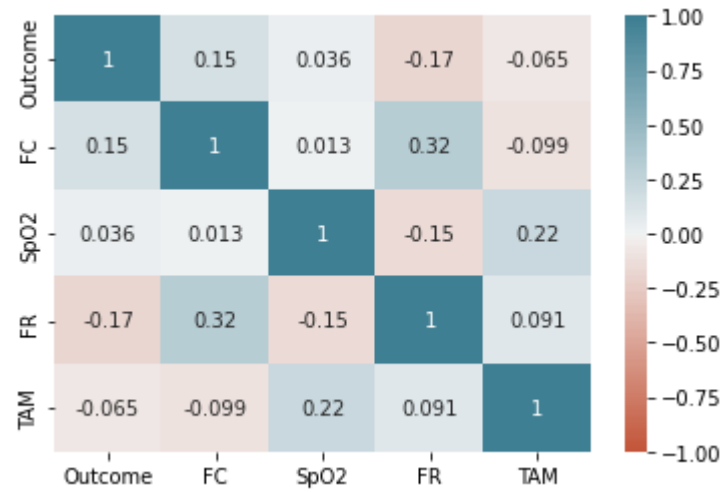
Figura 3-4

Matriz de correlación



Nota. Fuente: Elaboración propia

Se observó un valor alto en los coeficientes de correlación entre las variables TAM, TAS y TAD. Debido a que la presión arterial media se calcula a través de una función lineal de la presión arterial diastólica y la presión arterial sistólica, se nota un alto nivel de correlación entre estas tres variables. Por lo cual, se descartaron las variables TAS y TAD, puesto que TAM es capaz de describir el comportamiento de estas variables, como se muestra en la Figura 3-5. Además, se preservaron las demás variables que tenían un coeficiente de correlación bajo.

Figura 3-5*Matriz de correlación modificada*

Nota. Fuente: Elaboración propia

4. Modelos de aprendizaje automático implementados

En el desarrollo de esta investigación se implementaron diferentes modelos de aprendizaje automático y con diferentes fines, cada uno de estos seleccionados con base en la literatura estudiada.

Este capítulo se dividirá en tres secciones de acuerdo con el objetivo de cada uno de los modelos de aprendizaje utilizados. Inicialmente, se implementaron dos modelos de clasificación con el objetivo de encontrar si un paciente iba a sobrevivir o no según los signos vitales. Posteriormente, se entrenaron algunos modelos con el fin de clasificar entre los diferentes estados del paciente (crítico, inestable o estable). Finalmente, se hizo uso de dos modelos de redes neuronales profundas para predecir los estados del paciente en una ventana de tiempo.

Las métricas utilizadas para evaluar el rendimiento de los modelos implementados en este capítulo son: Exactitud (*accuracy*) y Exhaustividad (*recall*). Estos valores se calculan a través de fórmulas basadas en:

- Verdaderos Positivos (VP): datos que el modelo clasifica o predice correctamente en la clase positiva.
- Falsos Positivos (FP): datos que el modelo clasifica o predice incorrectamente en la clase positiva.
- Verdaderos Negativos (VN): datos que el modelo clasifica o predice correctamente en la clase negativa.
- Falsos Negativos (FN): datos que el modelo clasifica o predice incorrectamente en la clase negativa.

Estos valores suelen usarse para producir una matriz de confusión, donde se contrastan los valores reales de la clase con los valores predichos por el modelo. Es decir, la matriz de confusión se construye con los números de casos que son correcta o incorrectamente clasificados por el modelo, como se muestra en la Figura 4-1.

Figura 4-1

Matriz de confusión

		Real	
		Falso	Verdadero
Predicción	Falso	VN	FN
	Verdadero	FP	VP

Nota. Adaptado de Matrices de confusión [Ilustración]. Microsoft. (<https://learn.microsoft.com/es-es/training/modules/machine-learning-confusion-matrix/2-confusion-matrices>).

La matriz de confusión de la Figura 4-1 muestra en color verde los valores que fueron correctamente clasificados y en color rojo los que no fueron correctamente clasificados para las clases Verdadero y Falso. Las matrices de confusión se pueden extender a tantas clases como se desee y comparar los valores que fueron correctamente clasificados a través de la diagonal de la matriz, como se muestra en la Figura 4-2.

Figura 4-2

Matriz de confusión para clasificación en tres clases

		Real		
		Estable	Inestable	Crítico
Predicción	Estable	Verdadero Estable	Falso Estable	Falso Estable
	Inestable	Falso Inestable	Verdadero Inestable	Falso Inestable
	Crítico	Falso Crítico	Falso Crítico	Verdadero Crítico

Nota. Adaptado de Matrices de confusión [Ilustración]. Microsoft. (<https://learn.microsoft.com/es-es/training/modules/machine-learning-confusion-matrix/2-confusion-matrices>).

Las matrices de confusión permiten comprender fácilmente los datos que fueron clasificados correcta o incorrectamente. Además, como se mencionó anteriormente, las métricas para medir el rendimiento de los modelos están dadas por las siguientes fórmulas:

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN} \tag{4.1}$$

$$Exhaustividad = \frac{VP}{VP + FN} \tag{4.2}$$

Mientras la exactitud mide el porcentaje de casos que los modelos han acertado, la exhaustividad indica la cantidad de positivos reales que es capaz de identificar y marcar

como positivo verdadero, es decir, es un valor que indica qué tan bueno es el modelo para clasificar una clase en particular. La razón por la cual se decidió usar ambas métricas es porque cuando el conjunto de datos está desbalanceado, puede obtener una exactitud alta, pero las clases con pocos datos pueden pasar desapercibidas para el modelo y son clasificadas incorrectamente.

4.1 Clasificación de si sobrevive o no después de la estancia en UCIP, según los signos vitales

En este estudio se ha desarrollado un modelo de clasificación basado en diferentes técnicas de aprendizaje automático para predecir el *Outcome* de pacientes en unidades de cuidado intensivo pediátrica (UCIP). El objetivo del modelo era utilizar los signos vitales de los pacientes, como la frecuencia cardíaca, frecuencia respiratoria, presión arterial media y la saturación periférica de oxígeno; para determinar si el paciente sobreviviría o no a su estancia en la UCIP.

Se implementaron dos modelos de clasificación: clasificador basado en árboles de decisión (*Decision Tree Classifier*, DTC) y un clasificador basado en un bosque aleatorio (*Random Forest Classifier*, RFC). Cada uno de estos modelos se entrenó con el 70% de registros del conjunto de datos y se evaluó en un subconjunto de datos correspondiente al 30% restante.

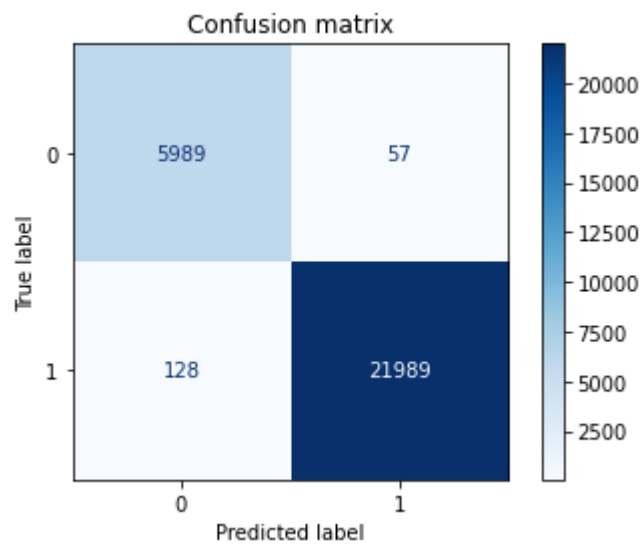
4.1.1 Clasificador basado en árboles de decisión (DTC)

Para la implementación de este modelo se usó *sklearn*, donde se entrenó un clasificador basado en árboles de decisión con una profundidad máxima de 20. Sin embargo, este valor de profundidad fue modificado varias veces hasta encontrar un punto medio entre la tasa de acierto y la capacidad de procesamiento, es decir, si se duplica la profundidad máxima del árbol, la tasa de acierto del modelo solo crecía unas pocas milésimas. Además, es importante mencionar que ampliar la profundidad del árbol puede generar un sobreajuste, es decir, que el árbol de decisión en lugar de encontrar patrones en los datos de entrenamiento se ajusta a estos y al momento de probar con los datos de prueba, su porcentaje de acierto disminuye.

Los resultados mostraron que el modelo era capaz de clasificar con gran precisión el *Outcome* (si sobrevivía o no) con una tasa de acierto del 99,29%. En la Figura 4-3 se muestra una matriz de confusión de los resultados obtenidos, donde 0 es que el paciente fallece y 1 que sobrevive.

Figura 4-3

Matriz de confusión DTC de Outcome



Nota. Fuente: Elaboración propia

En cuanto a la exhaustividad, para la clase 0 se obtuvo un valor de 99,34% y para la clase 1 se obtuvo un valor de 99,42%, como se muestra en las ecuaciones 4.3 y 4.4. Cabe resaltar que se incluyeron dichas ecuaciones para facilitar la comprensión del lector, pero en las secciones posteriores se omitirán y se mencionarán los resultados junto a la matriz de confusión generada.

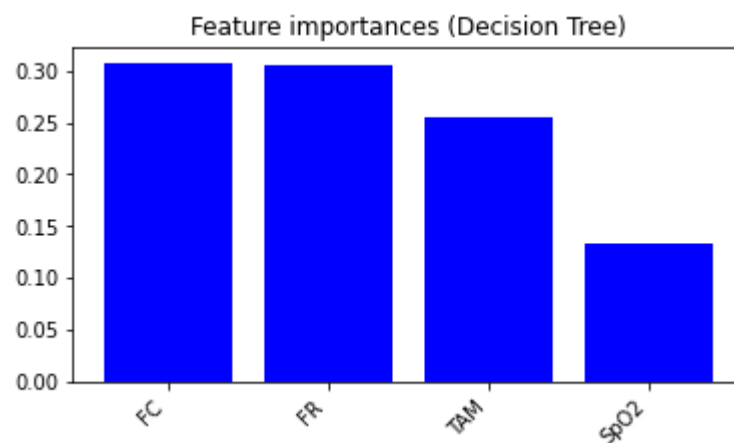
$$Ex_0 = \frac{5989}{5989 + 57} \cong 0,9906 \tag{4.3}$$

$$Ex_1 = \frac{21989}{21989 + 128} \cong 0,9942 \tag{4.4}$$

Además de lo mencionado anteriormente, los árboles de clasificación tienen la capacidad de proporcionar información sobre la importancia que tiene cada una de las características para lograr el resultado. Por consiguiente, se calculó el peso de cada signo vital, como se muestra en la Figura 4-4.

Figura 4-4

Importancia de las características



Nota. Fuente: Elaboración propia

La frecuencia cardíaca y la frecuencia respiratoria son las características con más importancia en el árbol de decisión con un 30,74% y 30,60, respectivamente; seguidos de la presión arterial media con 25,41% y, por último, la saturación periférica de oxígeno con 13,26%

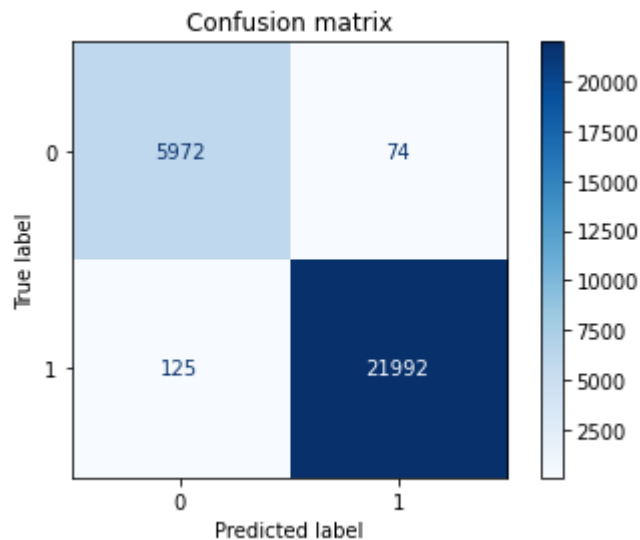
4.1.2 Clasificador basado en un bosque aleatorio (RFC)

En la implementación de esta técnica de aprendizaje automático también se utilizó *sklearn*. Se creó un RFC con 20 estimadores, es decir, con 20 árboles de decisión. Sin embargo, esta cantidad de estimadores fue modificada varias veces donde tomó valores de 100, 80, 50, 30, 20 y 10; pero con 20 árboles era suficiente para obtener resultados satisfactorios.

El modelo RFC es capaz de clasificar el *Outcome* con una exactitud del 99,29%. En cuanto a la exhaustividad se obtuvo 98,78% para la clase 0 y 99,43% para la clase 1. En la Figura 4-5 se muestra la matriz de confusión de los resultados obtenidos con esta técnica.

Figura 4-5

Matriz de confusión RFC de Outcome



Nota. Fuente: Elaboración propia

Los resultados obtenidos fueron muy similares a los resultados del DTC, esto es debido a que conceptualmente un RFC es un conjunto de árboles de decisión y los resultados de cada uno de estos, es promediado para obtener el resultado final del modelo.

4.2 Clasificación del estado actual del paciente, según los signos vitales

Los modelos vistos anteriormente pueden arrojar resultados significativamente positivos y ser una fuente de información valiosa para el equipo médico. Sin embargo, se decidió implementar modelos que sean capaces de clasificar el estado actual del paciente, según sus signos vitales. Esto permite al equipo médico evaluar si un paciente está presentando

un deterioro clínico y de esta manera actuar de forma oportuna y tener la posibilidad de revertirlo.

Como se mencionó anteriormente, los signos vitales pueden variar según algunos rangos de edades, por lo tanto, basados en la literatura expuesta, se decidió separar el conjunto de datos en 4 grupos: menores de un año (grupo 1), entre un año y tres años (grupo 2), entre cuatro y cinco años (grupo 3), y entre seis y doce años (grupo 4).

Inicialmente, se implementaron varias técnicas de aprendizaje automático para calcular el estado con base en los signos vitales, similar a como se implementaron los modelos para el *Outcome* en la sección anterior. Sin embargo, los resultados no fueron los esperados debido a que el conjunto de datos tiene las clases extremadamente desbalanceadas. En la Tabla 7, se puede observar la cantidad de registros por estado para cada uno de los grupos, donde el estado inestable tiene aproximadamente 64,5 veces la cantidad de datos del estado crítico y aproximadamente 8,5 veces el número de registros del estado estable.

Tabla 7

Cantidad de datos según el estado, por grupo

	Crítico	Inestable	Estable
Grupo 1	1073	63553	7844
Grupo 2	80	9036	1286
Grupo 3	19	2595	9
Grupo 4	57	4035	181
Total	1229	79219	9320

Nota. Fuente: Elaboración propia

Los modelos seleccionados para realizar esta clasificación fueron: un clasificador basado en árboles de decisión (DTC), un clasificador de bosque aleatorio (RFC), un clasificador basado en el algoritmo de K vecinos más próximos (KNN) y un perceptrón multicapa (MPC).

Para el entrenamiento de estos modelos, se partió el conjunto de datos en dos: 70% de los datos para entrenar el modelo, y el 30% restante para probar y validar los resultados obtenidos para cada una de las técnicas. Además, se tomaron como entrada los signos

vitales (FC, FR, TAM y SpO2) y como salida el estado del paciente en este instante de tiempo.

Los resultados obtenidos tuvieron una exactitud muy alta, sin embargo, la exhaustividad estuvo por debajo de lo esperado. En la Tabla 8 se puede observar la exactitud para cada uno de los modelos y grupos.

Tabla 8

Exactitud de los modelos DTC, RFC, KNN y MPC; por grupo

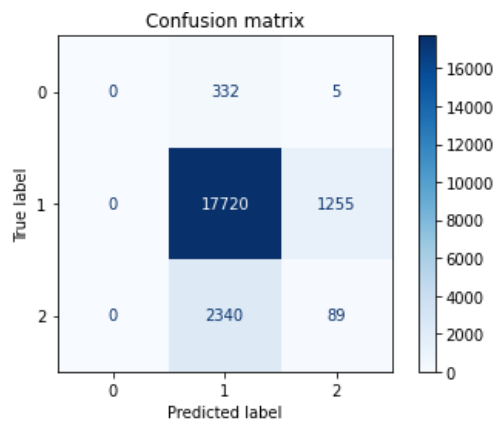
	DTC (%)	RFC (%)	KNN (%)	MPC (%)
Grupo 1	81,91	86,07	86,57	87,28
Grupo 2	62,03	80,58	86,13	86,35
Grupo 3	81,03	84,81	82,81	97,55
Grupo 4	89,71	88,97	86,76	80,88

Nota. Fuente: Elaboración propia

Aunque el porcentaje de exactitud es alto para todas las técnicas en prácticamente todos los grupos (ver Tabla 8), la realidad es muy diferente. La Figura 4-6 muestra la matriz de confusión del DTC para las clases 0 (crítico), 1 (inestable) y 2 (estable); con los datos de prueba del grupo.

Figura 4-6

Matriz de confusión DTC de clasificación del estado



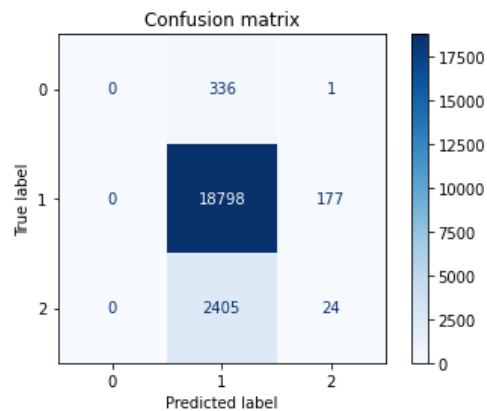
Nota. Fuente: Elaboración propia

Si se observa fijamente la Figura 4-6, se puede apreciar que el modelo debía clasificar 337 datos con la etiqueta 0 (estado crítico); sin embargo, el modelo clasificó 332 registros como 1 (inestable) y 5 datos como 2 (estable). Es decir, la exhaustividad fue de cero, puesto que no clasificó bien ninguno de estos registros. Luego, si se observa la clase 2, se tienen 2420 registros que se debieron clasificar con esa etiqueta, pero el modelo solo logró clasificar correctamente 89 registros y falló en 2340, obteniendo una exhaustividad del 3,66%.

Ahora bien, en la implementación de la técnica basada en el algoritmo de K vecinos más próximos, también obtuvo resultados deficientes. Como se puede observar en la Figura 4-7, el modelo clasificó 336 registros con la etiqueta 1 y un registro con la etiqueta 2 para la clase 0, con una exhaustividad del 0%. Para la clase 2, el modelo clasificó correctamente 24 registros, obteniendo una exhaustividad del 0,99%. Aunque la clase 1 tuvo una mejora en su exhaustividad el modelo muestra un rendimiento muy por debajo del promedio.

Figura 4-7

Matriz de confusión KNN de clasificación del estado



Nota. Fuente: Elaboración propia

Con base en los resultados obtenidos por las técnicas mencionadas, se observó un gran sesgo de los modelos de clasificar los signos vitales de los pacientes en el estado inestable. Por consiguiente, se decidió dividir el modelo en dos y balancear los pesos de cada una de las clases al momento de realizar el entrenamiento a través de *sklearn*.

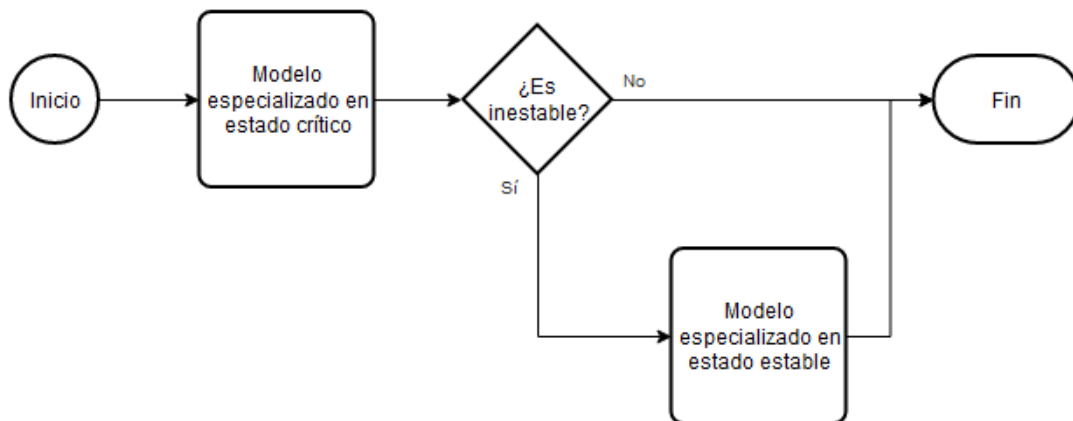
Por un lado, se crearon dos modelos, uno con el objetivo de clasificar entre los estados crítico e inestable y otro con el fin de clasificar entre los estados inestable y estable. De esta manera, cada uno de estos modelos solo tendría que clasificar entre dos clases 0 y 1 o 1 y 2.

Los modelos creados se llamaron *Especializado en estado crítico* y *Especializado en estado estable*. Además, en el entrenamiento se eliminaron los datos que no pertenecían a las clases que tenían que clasificar. Es decir, el conjunto de entrenamiento se dividió en dos: los registros que estaban etiquetados con 0 y 1 (para el especializado en estado crítico), y los registros que estaban etiquetados con 1 y 2 (para el especializado en estado estable).

En la Figura 4-8 se puede observar el flujo de la información y cómo se llega al resultado final. Inicialmente, los signos vitales ingresan al modelo especializado en estado crítico. Si el resultado de este modelo es 0 y devuelve esta etiqueta. Si el resultado es 1, los signos vitales se ingresan al modelo especializado en estado estable. Finalmente, el resultado devuelve la etiqueta resultante de este último modelo.

Figura 4-8

Flujo de los datos en los modelos de clasificación



Nota. Fuente: Elaboración propia

Por otro lado, con el fin de balancear el conjunto de datos se decidió incluir un peso a cada una de las clases en las diferentes técnicas. Para esto, *sklearn* cuenta con el parámetro *class_weight* en algunos de sus clasificadores y que a través de un diccionario se puede definir la relevancia de cada clase en el entrenamiento.

Anteriormente se observó que para todos los grupos el número de registros con estado inestable era superior a los estados crítico y estable. Por lo tanto, con el objetivo de balancear el conjunto de datos se un valor d , que está dada por las siguientes fórmulas:

$$d_0 = \frac{\# \text{registros de inestables}}{\# \text{registros de críticos}} \quad (4.5)$$

$$d_2 = \frac{\# \text{registros de inestables}}{\# \text{registros de estables}} \quad (4.6)$$

Las fórmulas 4.5 y 4.6 definen la proporción que existe entre el número de registros de la clase inestable con las clases crítico y estable, respectivamente. Por ejemplo, si en un conjunto de datos de entrenamiento tenemos 100 registros para la clase inestable y 5 registros para la clase estable, d_2 es 20 y los posibles pesos de cada clase en el modelo serán 1 para la clase inestable y 20 para la clase estable.

4.2.1 Clasificador basado en árboles de decisión (DTC)

En la implementación de la técnica basada en árboles de decisión, se entrenaron dos modelos (especializado en estado crítico y especializado en estado estable) para cada uno de los grupos.

Los parámetros usados en cada uno de estos métodos de clasificación fueron: profundidad máxima de 100 (*max_depth*) y un mínimo número de muestras en cada partición (*min_samples_split*) de 256. Es decir, cada nodo tendrá una cantidad de nodos hijos de 256 hasta que el árbol tenga una profundidad de 100.

En la Tabla 9, se pueden apreciar los resultados obtenidos para los dos árboles de decisión implementados para el grupo 1.

Tabla 9

Resultados de DTC para el grupo 1

Grupo 1	Especializado en crítico		Especializado en estable	
	Clase 0	Clase 1	Clase 1	Clase 2
Exhaustividad	0,77	0,73	0,71	0,71
Exactitud	0,73		0,71	

Nota. Fuente: Elaboración propia

Los resultados obtenidos para el grupo 2 en los dos árboles de decisión implementados se presentan en la Tabla 10.

Tabla 10

Resultados de DTC para el grupo 2

Grupo 2	Especializado en crítico		Especializado en estable	
	Clase 0	Clase 1	Clase 1	Clase 2
Exhaustividad	0,67	0,86	0,74	0,76
Exactitud	0,86		0,74	

Nota. Fuente: Elaboración propia

La Tabla 11 presenta los resultados obtenidos por los árboles de decisión para el grupo 3. Los valores fueron muy buenos en comparación con los resultados obtenidos en los grupos 1 y 2.

Tabla 11

Resultados de DTC para el grupo 3

Grupo 3	Especializado en crítico		Especializado en estable	
	Clase 0	Clase 1	Clase 1	Clase 2
Exhaustividad	1,00	0,95	0,95	1,00
Exactitud	0,95		0,95	

Nota. Fuente: Elaboración propia

La Tabla 12 muestra los resultados para el grupo 4 en esta técnica implementada, donde se observa un decremento de la exhaustividad de la clase 1 y de su exactitud en el árbol especializado en detectar estados estables.

Tabla 12

Resultados de DTC para el grupo 4

Grupo 4	Especializado en crítico		Especializado en estable	
	Clase 0	Clase 1	Clase 1	Clase 2
Exhaustividad	0,94	0,88	0,63	0,83
Exactitud	0,88		0,64	

Nota. Fuente: Elaboración propia

Los pesos se modificaron para cada uno de los grupos según las fórmulas (4.5) y (4.6). Los valores correspondientes daban un peso a las clases minoritarias, para evitar el sesgo hacia la clase 1. Por ejemplo, los pesos para el grupo 1 fueron aproximadamente 60 para la clase 0, 1 para la clase 1 y 8 para la clase 2. Es decir, la clase 1 contaba con aproximadamente 60 veces la cantidad de registros que la clase 0 y cerca de 8 veces la cantidad de registros de la clase 2.

4.2.2 Clasificador basado en un bosque aleatorio (RFC)

De manera similar al modelo anterior, se entrenaron dos modelos (uno para estado crítico y otro para estado estable) utilizando la técnica de bosque aleatorio.

En estos métodos de clasificación se utilizaron 100 estimadores, es decir, un máximo de 100 árboles y una cantidad mínima de muestras para dividir en cada nodo de 256. Además, no se especificó una profundidad máxima para cada uno de los árboles.

En este caso, se realizó también un ajuste de los pesos para cada una de las clases basado en las ecuaciones (3.5) y (3.6).

En la Tabla 13, se pueden apreciar los resultados obtenidos para el grupo 1.

Tabla 13

Resultados de RFC para el grupo 1.

Grupo 1	Especializado en crítico		Especializado en estable	
	Clase 0	Clase 1	Clase 1	Clase 2
Exhaustividad	0,71	0,79	0,72	0,74
Exactitud	0,79		0,72	

Nota. Fuente: Elaboración propia

Los resultados del grupo 2 se presentan en la Tabla 14.

Tabla 14

Resultados de DTC para el grupo 2

Grupo 2	Especializado en crítico		Especializado en estable	
	Clase 0	Clase 1	Clase 1	Clase 2
Exhaustividad	0,67	0,90	0,74	0,76
Exactitud	0,90		0,75	

Nota. Fuente: Elaboración propia

En la Tabla 15 se observan los resultados obtenidos para el grupo 3.

Tabla 15

Resultados de RFC para el grupo 3

Grupo 3	Especializado en crítico		Especializado en estable	
	Clase 0	Clase 1	Clase 1	Clase 2
Exhaustividad	1,00	0,97	0,91	0,67
Exactitud	0,97		0,91	

Nota. Fuente: Elaboración propia

La Tabla 16 muestra los resultados del grupo 4.

Tabla 16

Resultados de RFC para el grupo 4

Grupo 4	Especializado en crítico		Especializado en estable	
	Clase 0	Clase 1	Clase 1	Clase 2
Exhaustividad	0,82	0,90	0,70	0,83
Exactitud	0,90		0,71	

Nota. Fuente: Elaboración propia

4.2.3 Clasificador basado en regresión logística (LRC)

Se implementaron los dos modelos mencionados anteriormente usando la técnica de un clasificador basado en regresión logística y, de manera similar a las técnicas de DTC y RFC, se modificaron los pesos con las fórmulas descritas (3.5) y (3.6). Para la implementación de esta técnica se utilizó un número máximo de 100 iteraciones.

En la Tabla 17 se observan los resultados obtenidos para el grupo 2.

Tabla 17

Resultados de LRC para el grupo 1

Grupo 1	Especializado en crítico		Especializado en estable	
	Clase 0	Clase 1	Clase 1	Clase 2
Exhaustividad	0,50	0,69	0,64	0,62
Exactitud	0,69		0,64	

Nota. Fuente: Elaboración propia

La Tabla 18 muestra los resultados del grupo 2.

Tabla 18

Resultados de LRC para el grupo 2

Grupo 2	Especializado en crítico		Especializado en estable	
	Clase 0	Clase 1	Clase 1	Clase 2
Exhaustividad	0,50	0,77	0,49	0,65
Exactitud	0,77		0,51	

Nota. Fuente: Elaboración propia

Los resultados del grupo 3 se presentan en la Tabla 19.

Tabla 19

Resultados de LRC para el grupo 3

Grupo 3	Especializado en crítico		Especializado en estable	
	Clase 0	Clase 1	Clase 1	Clase 2
Exhaustividad	0,83	0,97	0,79	0,67
Exactitud	0,97		0,79	

Nota. Fuente: Elaboración propia

Por último, los resultados obtenidos para el grupo 4, se pueden observar en la Tabla 20.

Tabla 20

Resultados de LRC para el grupo 4

Grupo 4	Especializado en crítico		Especializado en estable	
	Clase 0	Clase 1	Clase 1	Clase 2
Exhaustividad	0,53	0,67	0,49	0,78
Exactitud	0,67		0,50	

Nota. Fuente: Elaboración propia

4.2.4 Clasificador en datos con categorización

Posterior al inicio de esta investigación, el equipo médico complementó el conjunto de datos con los signos vitales categorizados con base en su experiencia y el estado del paciente, los cuales se clasificaron como:

- Muy bajo
- Bajo
- Normal
- Alto
- Muy alto

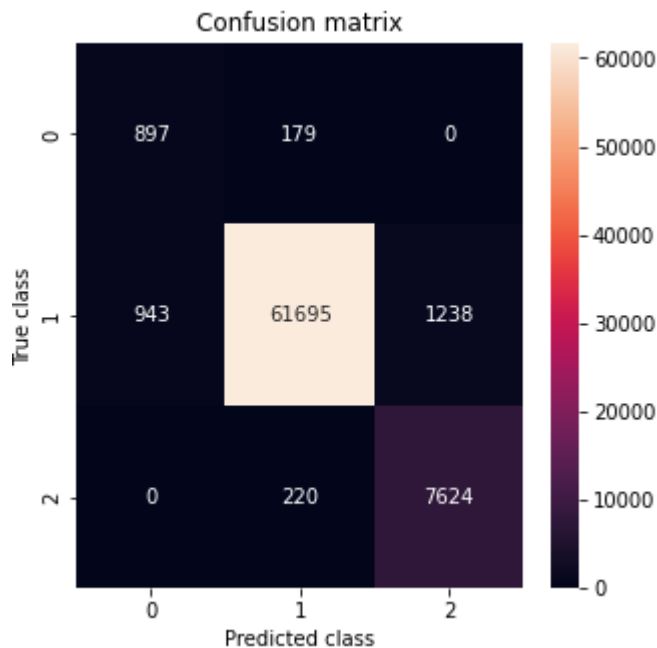
Por lo tanto, se escribió un algoritmo que permite analizar este conjunto de parámetros categorizados y calcular el estado actual del paciente. Dicho algoritmo consiste en asignar un valor a cada categoría para los signos vitales y, con base en esto, se calculaba el

resultado. Básicamente, (a) si todos los signos vitales fueron clasificados como normales, el estado es estable; (b) si todos los signos vitales fueron clasificados como anormales, el estado es crítico; y (c) en caso contrario, el estado es inestable.

En la Figura 4-9 se puede observar la matriz de confusión calculada con este algoritmo para todo el conjunto de datos. La exactitud fue de 96% y la exhaustividad para las clases crítico, inestable y estable fueron 83%, 97% y 97%, respectivamente.

Figura 4-9

Matriz de confusión del algoritmo de clasificación



Nota. Fuente: Elaboración propia

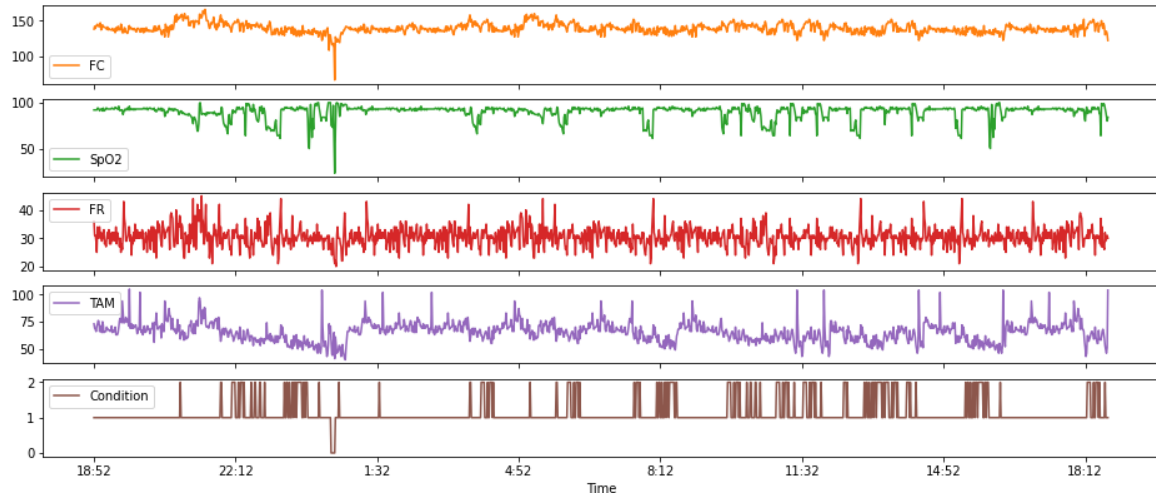
4.3 Predicción de estados futuros, según los signos vitales

Para la predicción de los futuros estados basado en los signos vitales, a diferencia de la sección anterior, los datos se toman como una serie de tiempo. Es decir, cada uno de los registros de un paciente fueron registrados en orden cronológico y esto se puede utilizar para analizar y encontrar patrones en el conjunto de datos.

En este caso, los signos vitales entran a la red neuronal en conjuntos de 5 datos.

Figura 4-10

Señales de los signos vitales y el estado a través del tiempo.



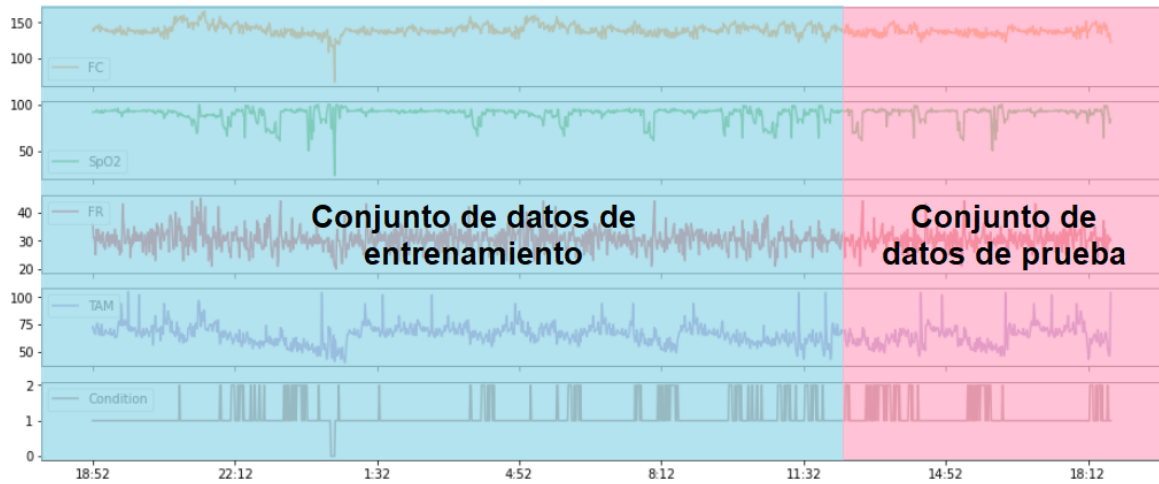
Nota. Fuente: Elaboración propia

Para el entrenamiento de los modelos, se tomó el 70% de los registros de cada uno de los pacientes para entrenamiento y el 30% restante como datos de prueba. Es decir, el conjunto de los datos de prueba se creó dividiendo en los grupos ya definidos. Posteriormente, se concatenaron los registros del primer 70% de cada paciente para el conjunto de entrenamiento. Finalmente, los registros restantes de cada paciente sirvieron para evaluar la eficiencia de los modelos para predecir los estados de dichos registros, de forma individual para cada paciente.

En la Figura 4-11 se muestra la división de los subconjuntos de entrenamiento y prueba de un paciente. El conjunto de datos de entrenamiento se unió a todos los conjuntos de entrenamiento de los demás pacientes del mismo grupo, y se realizó el mismo proceso con los conjuntos de datos de prueba.

Figura 4-11

Conjuntos de datos de entrenamiento y prueba de un paciente



Nota. Fuente: Elaboración propia

Por un lado, para el entrenamiento se crearon subconjuntos del conjunto de pruebas de 5 registros. Esto con el objetivo de que las redes neuronales implementadas pudiesen calcular el siguiente valor, y se repite el proceso para los demás datos (ver Figura 4-12).

Figura 4-12

Predicción de los datos según los antecesores



Nota. Fuente: Elaboración propia

Por otro lado, como los datos están desbalanceados se escribió una función para hacer un remuestreo. Dicha función consiste en aumentar los datos de las clases minoritarias dependiendo de los datos ya existentes en el conjunto de datos.

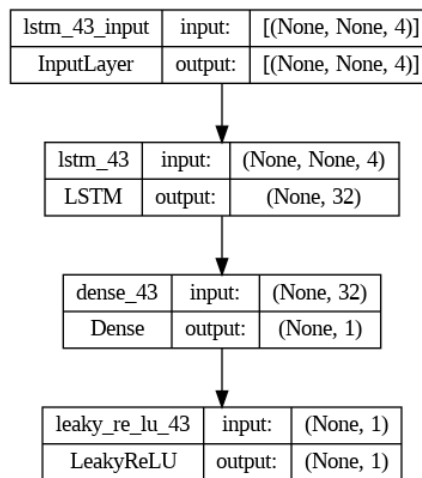
4.3.1 Long short-term memory (LSTM)

Una LSTM es una red neuronal artificial compuesta por varias unidades (celdas) LSTM, cada una de estas es capaz de aprender a identificar información relevante en una entrada, almacenarla en su memoria a largo plazo, retenerla durante el tiempo necesario y extraerla cuando sea requerida. Por lo tanto, este tipo de redes suele tener éxito en tareas como el reconocimiento de patrones a largo plazo en series temporales, textos extensos y grabaciones de audio, entre otros (Géron, 2017).

Para la implementación se creó una red neuronal con cuatro capas: (a) una capa de entrada, (b) una capa LSTM con 32 unidades, (c) una capa densa con una función de activación lineal, y (d) una capa LeakyReLU. En la Figura 4-13 se muestra el grafo correspondiente a la red neuronal construida.

Figura 4-13

Grafo de la red neuronal basada en LSTM



Nota. Fuente: Elaboración propia

El entrenamiento se hizo por 30 épocas para cada uno de los grupos, con un optimizador *adam* y una función de pérdida basada en error cuadrático medio. Posteriormente, se calculó la exactitud del modelo comparado con el conjunto de datos de prueba.

Tabla 21

Resultados de LSTM para los 4 grupos

LSTM	Exactitud
Grupo 1	77,18%
Grupo 2	72,21%
Grupo 3	82,42%
Grupo 4	96,02%

Nota. Fuente: Elaboración propia

Para conocer más detalles sobre los resultados, ver el Anexo B.

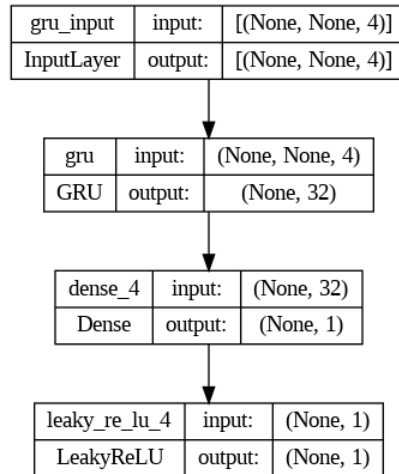
4.3.2 Gated recurrent unit (GRU)

Una GRU es una red neuronal recurrente similar a la LSTM. Cada celda es una versión simplificada de una celda LSTM que funciona tan bien en el reconocimiento de patrones en series de tiempo. Los cambios principales son: (a) la combinación de ambos vectores de estado en uno solo, (b) un único controlador que administra las puertas de olvido y entrada, (c) no hay puerta de salida (Géron, 2017).

De manera similar a la implementación de la LSTM, para esta implementación se construyó una red neuronal compuesta por cuatro capas: (a) una capa de entrada, (b) una capa GRU con 32 unidades, (c) una capa densa con una función de activación lineal y (d) una capa LeakyReLU. La Figura 4-14 ilustra el diagrama de la red neuronal diseñada. Además, para el entrenamiento se usó un optimizador de *adam* y una función de pérdida de error cuadrático medio.

Figura 4-14

Grafo de la red neuronal basada en GRU



Nota. Fuente: Elaboración propia

La red neuronal se entrenó con 30 épocas y se creó una red para cada uno de los grupos del conjunto de datos.

Luego, se calculó la exactitud del modelo comparado con el conjunto de datos de prueba y los resultados obtenidos se muestran en la Tabla 22.

Tabla 22

Resultados de GRU para los 4 grupos

GRU	Exactitud
Grupo 1	72,90%
Grupo 2	65,05%
Grupo 3	86,21%
Grupo 4	95,88%

Nota. Fuente: Elaboración propia

Para ver más detalle sobre los resultados, ver el Anexo B.

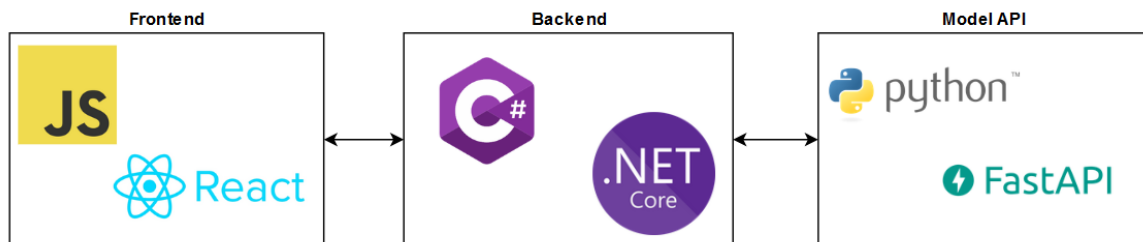
5. Prototipo de software

Para la implementación de esta aplicación del aprendizaje automático a la medicina y su interacción con el personal médico, se diseñó una plataforma que se divide en tres aplicaciones llamadas: *Frontend*, *Backend*, y *Model API*. Las aplicaciones interoperan entre sí para recibir las entradas del usuario, leer los datos, y generar las clasificaciones y predicciones, respectivamente.

En la Figura 5-1 se observan las tres aplicaciones y los canales de comunicación entre ellas.

Figura 5-1

Aplicaciones que componen el sistema



Nota. Fuente: Elaboración propia

5.1 Frontend

La primera de estas aplicaciones llamada *Frontend* fue desarrollada usando JavaScript y la librería React. Este proyecto permite al usuario, en este caso el personal médico, subir un archivo de Excel y seleccionar el rango de edad del paciente. Una vez procesada la información, esta aplicación muestra los resultados de clasificación y predicción a través de gráficos de *gauge* y una tabla, facilitando así la comprensión de los resultados del

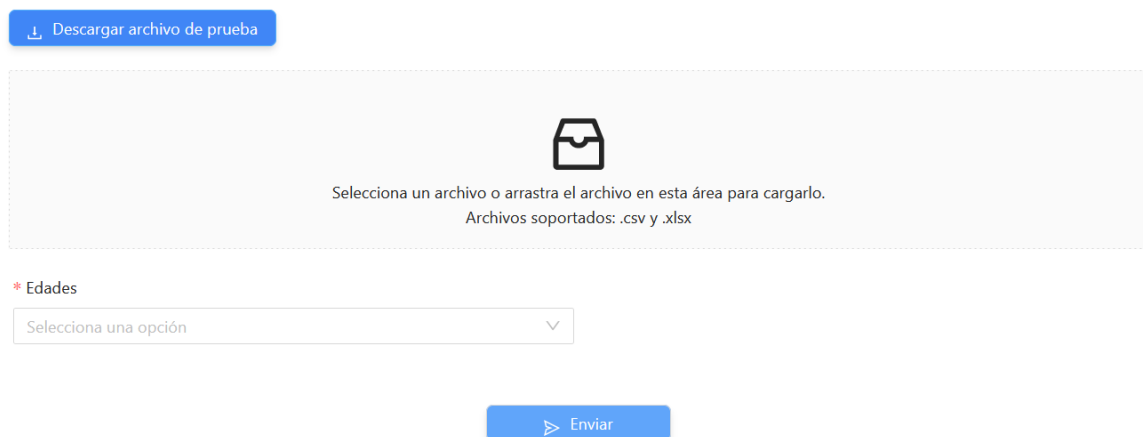
análisis. En resumen, esta aplicación tiene como objetivo la interacción del usuario con la plataforma.

Cuando el usuario ingresa a la aplicación podrá (ver Figura 5-2):

1. Descargar un documento de Excel de prueba.
2. Cargar un documento de Excel con los signos vitales.
3. Seleccionar el rango de edad del paciente, el cual define el grupo al que pertenece.
4. Enviar el documento de Excel y el rango de edades.

Figura 5-2

Vista inicial de la aplicación



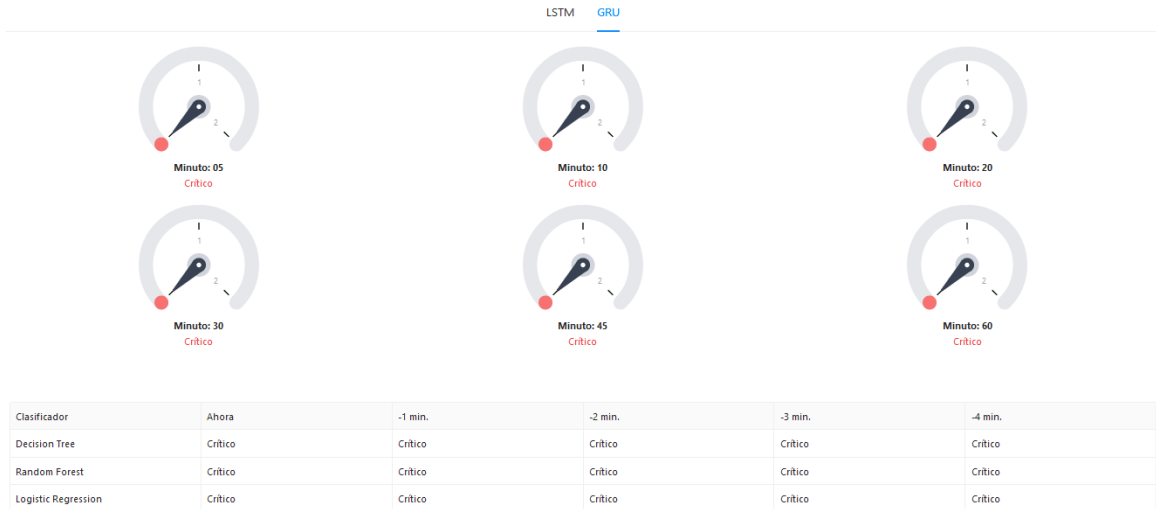
The screenshot shows the initial interface of the application. At the top left, there is a blue button with a download icon and the text "Descargar archivo de prueba". Below this is a large, light gray dashed box representing a file upload area. In the center of this box is a folder icon, and below it, the text "Selecciona un archivo o arrastra el archivo en esta área para cargarlo." and "Archivos soportados: .csv y .xlsx". Below the upload area is a dropdown menu labeled "* Edades" with the placeholder text "Selecciona una opción" and a downward arrow. At the bottom center, there is a blue button with a right-pointing arrow and the text "Enviar".

Nota. Fuente: Elaboración propia

Luego de enviar la información, al cabo de unos segundos, se pueden observar los resultados obtenidos (ver Figura 5-3 y Figura 5-4). En la parte superior se pueden visualizar dos pestañas: LSTM y GRU, las cuales muestran a través gráficos de gauge las predicciones para los minutos 5, 10, 20, 30, 45 y 60; para cada uno de estos modelos. En la parte inferior, el usuario podrá visualizar el resultado de clasificación de los modelos DTC, RFC y LRC; para los últimos 5 registros del conjunto de datos.

Figura 5-3

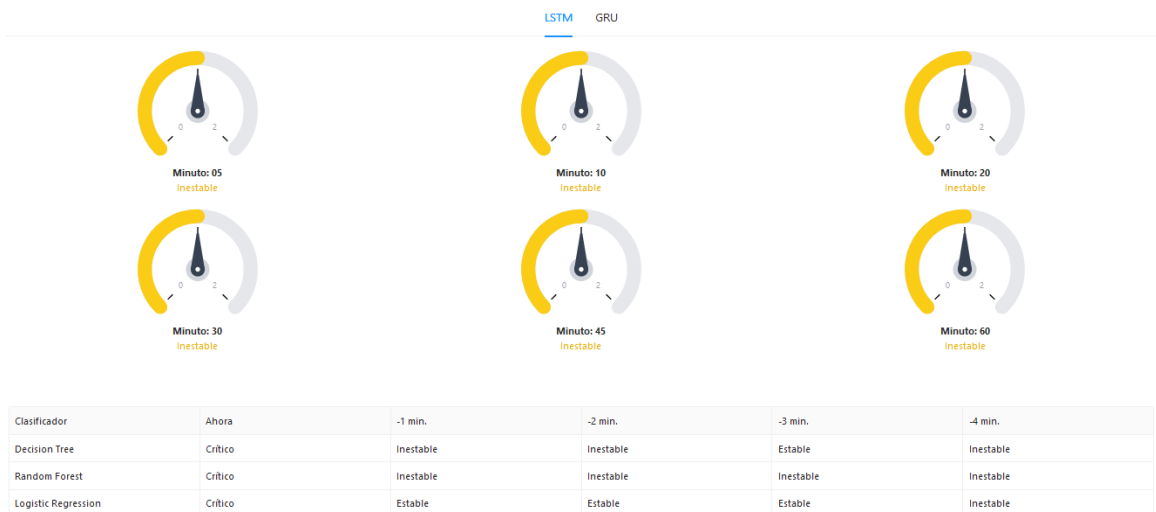
Vista con los resultados obtenidos (GRU)



Nota. Fuente: Elaboración propia

Figura 5-4

Vista con los resultados obtenidos (LSTM)



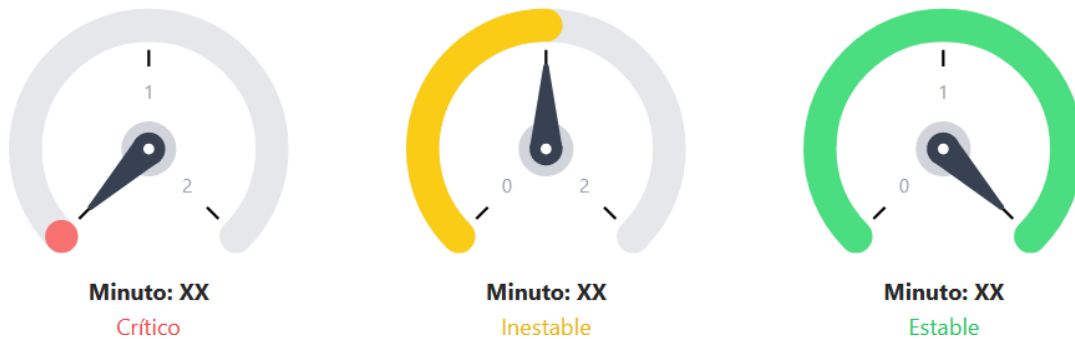
Nota. Fuente: Elaboración propia

Como se mencionó anteriormente, los modelos basados en LSTM y GRU hacen una predicción de un máximo de 60 minutos, para los datos cargados por el usuario. Las predicciones son representadas en un gráfico de *gauge*, que presenta el estado predicho.

Las posibles representaciones son Crítico (rojo), Inestables (amarillo) y Estable (verde), como se muestra en la Figura 5-5.

Figura 5-5

Representaciones de estados en gráficos gauge



Nota. Fuente: Elaboración propia

Dicho lo anterior, esta aplicación se encarga de mostrar al personal médico representaciones visuales de los resultados de los modelos implementados para facilitar la comprensión de los signos vitales y apoyar al equipo médico en la toma de decisiones.

5.2 Backend

La segunda aplicación de esta plataforma se llama *Backend*, y tiene como función principal leer los datos del documento en Excel y transformarlos para que sirvan como entrada a los diferentes modelos expuestos en *Model API*. Esta aplicación fue desarrollada usando el lenguaje C# y haciendo uso del marco de trabajo .NET 6.0.

El archivo debe contar con las columnas: FC, SPO2, FR, TAM y ESTADOS. Las 4 primeras hacen referencia a los signos vitales tratados en esta investigación y son fundamentales para la clasificación y predicción de los modelos, mientras que la última columna sólo es usada por los modelos de predicción para calcular los siguientes estados en los próximos 60 minutos. Es necesario mencionar que esta ventana de tiempo fue sugerida por el equipo médico que acompañó el desarrollo de esta investigación, puesto que, se considera que

rangos de tiempo muy extensos en una UCIP no son útiles por el seguimiento constante a los pacientes.

Inicialmente, esta aplicación recibe el grupo al que corresponde el paciente y el documento en Excel. Luego, a través de una librería llamada *EPPlus* (v4.5.2.1) se lee la información en el archivo adjunto y se mapea en una lista de objetos de una clase. Una vez mapeados los datos, son enviados junto con el grupo al *Model API*, a través de una petición REST. Finalmente, los resultados son mapeados en otra clase para ser devueltos a la aplicación *Frontend* para la visualización de los resultados.

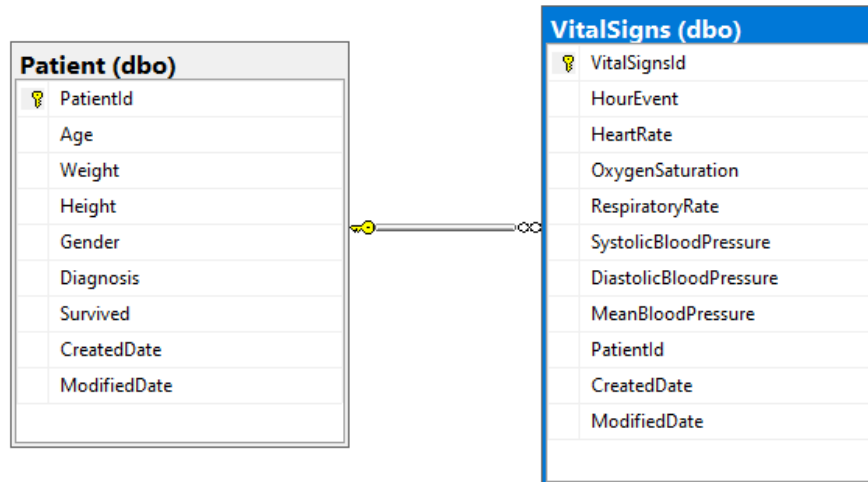
Con el fin de facilitar la recolección de datos en futuras versiones, se han implementado clases de modelo que a través del ORM llamado *Entity Framework* se agreguen a una base de datos los registros que se suban a través de la plataforma. Es decir, la información que se cargará a través de los archivos de Excel se almacenará para un futuro preprocesamiento y entrenamiento de los modelos existentes o para el uso de posibles nuevas técnicas de aprendizaje automático que se implementen.

Para el desarrollo de esta funcionalidad, se implementó una base de datos sencilla, que cuenta con solo dos tablas: *Patient* y *VitalSigns* (ver Figura 5-6). Estas dos tablas cuentan con una relación *1-N*. Es decir, un paciente puede tener *n* registros asociados a sus signos vitales, pero cada uno de estos registros sólo puede estar asociado a un paciente.

Ambas tablas cuentan con dos columnas llamadas *CreatedDate* y *ModifiedDate*, con el fin de realizar auditoría sobre los registros. Además, *PatientId* y *VitalSignsId* son identificadores alfanuméricos autogenerados.

Figura 5-6

Entidades creadas en la base de datos



Nota. Fuente: Elaboración propia

5.3 Model API

Esta aplicación fue desarrollada en Python y haciendo uso del marco de trabajo web FastAPI. Esta API cuenta con todos los modelos entrenados en el desarrollo de esta investigación y se encarga de cargar solamente los necesarios para realizar la clasificación y predicción correspondiente.

Esta aplicación expone un servicio llamado *predict*. El cual se encarga de tomar la información que es enviada desde la aplicación *Backend* y enviarla a los diferentes modelos entrenados (1) DTC, (2) RFC, (3) LRC, (4) LSTM y (5) GRU. Es necesario recordar que para los modelos de clasificación, primero se llama al modelo especializado en detectar estados críticos, y si el resultado es “inestable”, se llama al modelo especializado en detectar estados estables. En total, esta aplicación cuenta con 32 modelos entrenados para calcular el resultado de la clasificación y la predicción de los datos.

Finalmente, luego de que los resultados son calculados, el *Model API* envía una respuesta al *Backend* con esta información. Por su parte, el *Backend* se encargará de enviar la respuesta al *Frontend* que, a su vez, presentará la información al usuario.

5.4 Diagrama de secuencia

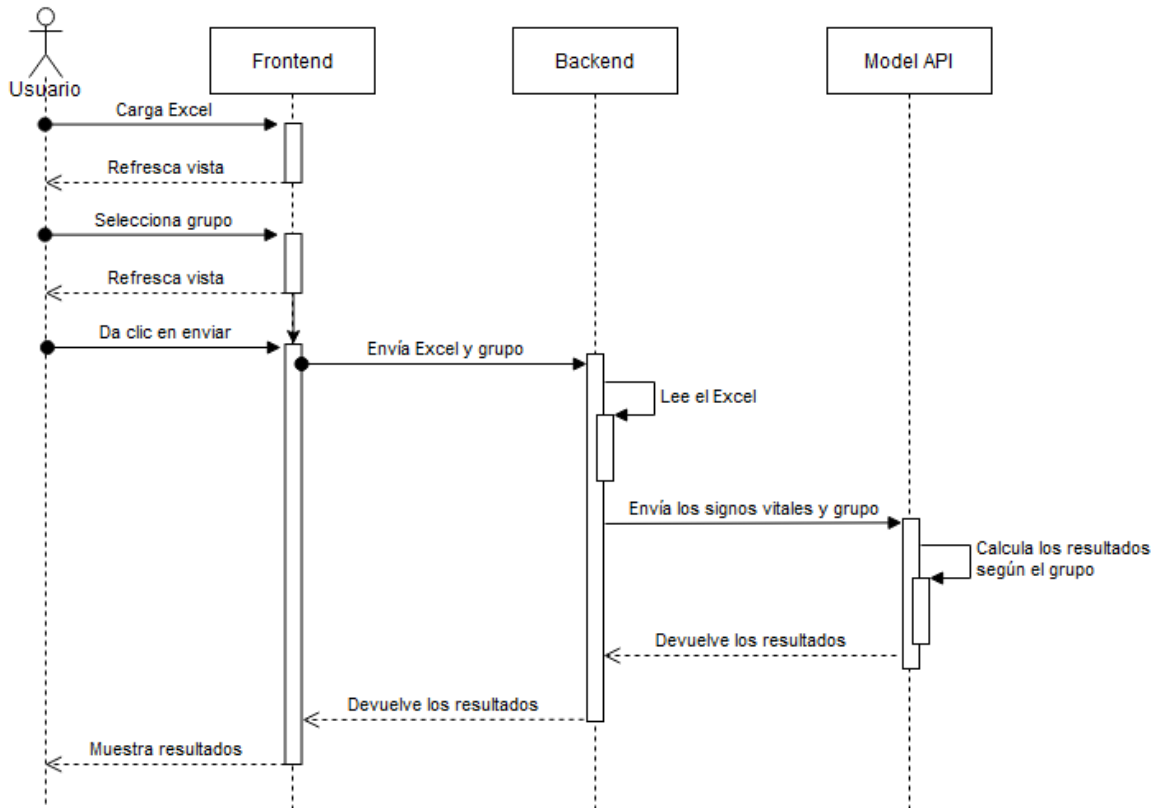
Con el fin de ilustrar el funcionamiento de la plataforma, se ha implementado un diagrama de secuencia, el cual muestra la forma en la que interactúan las diferentes aplicaciones y usuarios del sistema en un período de tiempo.

Como se muestra en la Figura 5-7, el inicializador del proceso es el usuario, quien, desde la aplicación llamada *Frontend*, es el encargado de cargar el documento en Excel y seleccionar el rango de edad al cual pertenece el paciente. Posteriormente, el sistema realiza los siguientes pasos:

1. Envía el documento de Excel y el grupo al *Backend*
2. *Backend* lee el archivo de Excel y lo mapea en objetos de clase
3. Se envían los datos desde *Backend* a *Model API*
4. *Model API* selecciona los modelos entrenados a usar a partir del grupo
5. Se obtienen los diferentes resultados de los modelos y se mapean en un diccionario
6. Se envía el resultado de *Model API* al *Backend*
7. *Backend* recibe el resultado y lo reenvía al *Frontend*
8. *Frontend* obtiene la respuesta y recarga la vista, presentando al usuario los gráficos y la tabla correspondiente

Figura 5-7

Diagrama de secuencia para obtener los resultados



Nota. Fuente: Elaboración propia

5.5 Diagrama de despliegue

Para el despliegue de las aplicaciones se usó la plataforma de nube pública de Microsoft, llamada Azure. Esta permite una gran variedad de servicios computacionales, bases de datos y otros tipos de almacenamiento, análisis de datos, uso de inteligencia artificial, entre otros.

Además, se utilizó un grupo de recursos que cuenta con tres App Service, el cual es un servicio de Azure que permite crear, desplegar y escalar aplicaciones web, móviles y las

API. Este servicio se escogió porque el despliegue de aplicaciones web y API es un proceso sencillo, como es el caso de esta plataforma.

También, es necesario mencionar que se utilizó Azure como plataforma en la nube debido a la experiencia del autor de esta investigación. Es decir, el despliegue se realizó en Azure por preferencia propia, pero estas aplicaciones se pudieron haber desplegado en otros servicios de nube, como AWS o GCP, entre otros.

Por un lado, para las aplicaciones *Frontend* y *Backend*, se creó un App Service para cada una, que cuenta con Windows Server y la versión de .NET 6.0. Sin embargo, la aplicación *Frontend* no hace uso de este *framework*, ya que a través de un *script* y de forma automatizada obtiene los paquetes de Node requeridos para el funcionamiento.

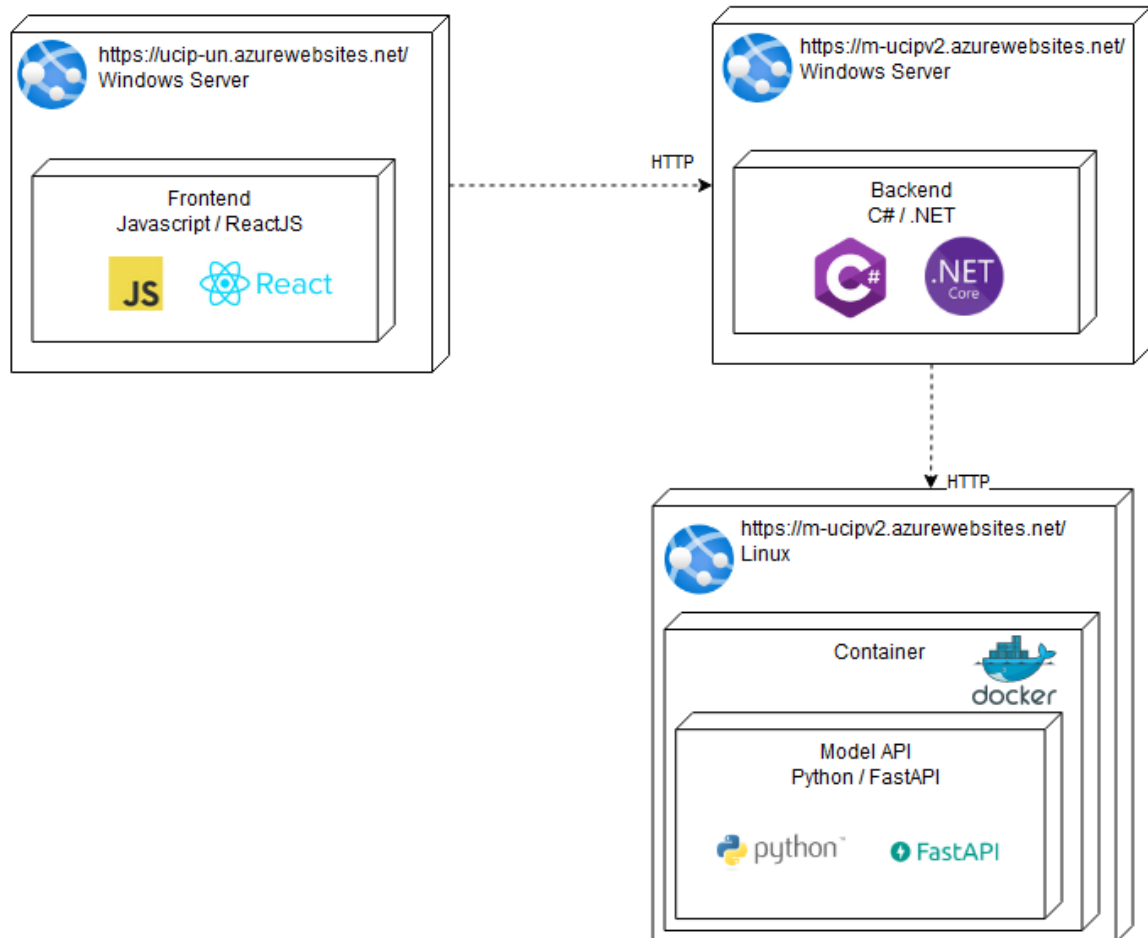
Por otro lado, *Model API* fue publicado en un servidor Linux, esto debido a que Azure cuenta con versiones diferentes de Python y de los paquetes requeridos para cargar los modelos y que son expuestos a través de FastAPI. Por lo tanto, para conservar los requerimientos de las librerías se decidió usar *Docker* para esta aplicación.

Docker es una plataforma de contenedores basada en la virtualización, y que facilita el empaquetamiento y distribución de las aplicaciones, donde se aíslan este tipo de aplicaciones de su entorno de ejecución. Es decir, se crea un paquete con la versión de Python, y de las librerías requeridas, y va a ejecutarse de manera similar en cualquier sistema operativo de la máquina que contiene *Docker*.

En la Figura 5-8 se pueden observar los *App Services* con sus respectivos enlaces y las aplicaciones que son ejecutadas dentro de cada una de estas.

Figura 5-8

Diagrama de despliegue de las aplicaciones



Nota. Fuente: Elaboración propia

6. Discusión y análisis de resultados

En el desarrollo del prototipo propuesto se obtuvieron buenos resultados en cuanto a la implementación de los modelos de clasificación y predicción. Estos modelos pueden proporcionar información valiosa a los médicos y enfermeros que trabajan en las UCIP para tomar decisiones informadas sobre el tratamiento de los pacientes.

Se realizó un estudio del conjunto de datos existente con el objetivo de comprender el comportamiento de las diferentes variables relacionadas con los signos vitales y la correlación lineal existente entre estas, dando como resultado la eliminación de las características TAS y TAD por su alto grado de correlación lineal con la variable TAM. Asimismo, se evaluó la distribución del conjunto de datos en distintos grupos, según la edad, de acuerdo con los estudios existentes y esta información fue crucial para la creación de varios modelos de aprendizaje automático.

La fase de preprocesamiento permitió aplicar técnicas de minería de datos para realizar una limpieza exhaustiva de la base de datos, algunos de los pasos realizados fueron: (1) Eliminar datos duplicados, (2) Modificar unidades de medida, (3) Administrar datos nulos, (4) Cambiar notación, (5) Modificar valores atípicos, (6) Modificar valores incorrectos, y (7) Cambiar tipo de variable. Esta fase se validó con el equipo médico colaborador del grupo de investigación LISI. Puesto que fue necesario corroborar si los posibles valores atípicos e incorrectos detectados, en realidad lo eran o si se trataba de una anomalía causada por el estado de salud del paciente.

Se implementaron técnicas de aprendizaje automático para clasificar, a través de los signos vitales, si un paciente sobrevive o no a su estancia en UCIP. Para esto, se creó un clasificador de árboles de decisión y uno basado en bosque aleatorio. El primero de estos dos clasificadores obtuvo una tasa de acierto del 99,34%, en cuanto a la exhaustividad

para las clases sobrevive y no sobrevive fueron de 99,42% y 99,06%, respectivamente. El segundo de estos algoritmos de clasificación logró una exactitud de 99,29% y una exhaustividad para las clases sobrevive o no sobrevive de 99,43% y 98,78%, respectivamente. Aunque los modelos presentan buenos resultados, dicha información podría ser insuficiente para el personal médico, ya que estos clasifican los signos vitales entre si el paciente sobrevive o no con estos signos vitales.

La limitación de los modelos anteriores permitió la creación de modelos capaces de clasificar entre los estados crítico, inestable o estable; y no sólo sobre el posible desbalance del paciente a su estancia en UCIP. Para dichas implementaciones se usaron los modelos basados en árboles de decisión, bosque aleatorio, k-vecinos más cercanos y un perceptrón multicapa. Sin embargo, los resultados no fueron favorables debido al desbalanceo existente entre las clases, puesto que el estado inestable contaba con una cantidad de registros mucho mayor que para las clases crítico y estable.

El desbalanceo de las clases permitió la creación de modelos de clasificación especializados sobre los estados del paciente usando las técnicas de aprendizaje automático DTC, RFC y LRC. Estas implementaciones contaron con la posibilidad de modificar el peso de las clases en el entrenamiento y permitiendo que los modelos de clasificación no tuviesen un sesgo sobre la clase mayoritaria. Los resultados obtenidos fueron aceptables y se observa que crear estos modelos especializados puede llegar a ser una buena alternativa para enfrentar el problema relacionado con el desbalanceo de las clases.

La categorización de los signos vitales por parte del personal médico permitió escribir un algoritmo que calcula el estado clínico del paciente. Los resultados obtenidos fueron muy buenos, puesto que la categorización de los signos vitales con base en la experiencia en UCIP del personal médico y las particularidades del paciente no están relacionadas directamente con los rangos *normales* de los signos vitales descritos en la literatura.

Finalmente, la implementación de los modelos predictivos basados en LSTM y GRU arrojaron resultados aceptables en cuanto a la predicción de estados vitales en la ventana de tiempo definida de una hora. Es necesario mencionar que este rango de tiempo de

predicción fue sugerido por el equipo médico colaborador del LISI que acompañó esta investigación, dado que en las unidades de cuidados intensivos el seguimiento al paciente es constante y predicciones en ventanas de tiempo muy grandes no son relevantes para el personal médico. Se realizó un análisis del comportamiento de la predicción para cada uno de los pacientes y la exactitud fue alta, pero el desbalanceo de las clases sigue siendo un reto y problema de estudio, puesto que algunos pacientes obtuvieron una exhaustividad alta y otros muy baja. Esto se debe a que las clases minoritarias en algunos de los pacientes contaban con solo unos cuantos registros de dichas clases.

En cuanto al desarrollo de la plataforma de software, se realizó un prototipo de software a partir de la interoperabilidad de tres aplicaciones de forma exitosa. Cada una de las tres aplicaciones tiene su responsabilidad demarcada en la plataforma implementada: interfaz de usuario, leer documento en Excel y evaluar los datos a través de los modelos implementados. La interfaz gráfica cuenta con una gran usabilidad y de fácil manejo y comprensión para el usuario final.

Dicho lo anterior, el prototipo de software permite aportar información valiosa al personal médico representando a través de una tabla y gráficos de *gauge* el estado clínico actual y futuro, respectivamente. Sin embargo, es necesario mencionar que este prototipo podría ser una herramienta de apoyo al personal médico, pero aún cuenta con un amplio camino de mejora. Además, el equipo médico cuenta con un panorama más amplio y con muchas más variables que no fueron tenidas en esta investigación.

7. Conclusiones y recomendaciones

7.1 Conclusiones

La implementación del aprendizaje automático en el área de la medicina es una valiosa herramienta de apoyo. Por lo tanto, el desarrollo del prototipo de software propuesto en esta investigación es de gran utilidad para la predicción de estados fisiológicos de los pacientes en una unidad de cuidados intensivos pediátrica, a través de los signos vitales y con el uso de diferentes técnicas de aprendizaje automático. Sin embargo, aún queda un amplio camino por recorrer para brindar al personal médico una solución cada vez más oportuna, confiable y portable; con el fin de brindar información importante de forma práctica e inmediata al personal médico.

La implementación de la metodología basada en el proceso de aprendizaje automático permitió definir muy bien las tareas y objetivos que se desarrollaron a lo largo de este proyecto. Es necesario mencionar, que todos los pasos del proceso mencionado son indispensables y se deben realizar de forma iterativa hasta lograr niveles de precisión cada vez más altos.

Por un lado, en cuanto a los modelos para clasificar si un paciente sobrevivía o no a su estancia en UCIP, con base en los signos vitales, las técnicas DTC y RFC obtuvieron una tasa de acierto alta y muy similar entre ellas. Por otro lado, los modelos de aprendizaje automático implementados permitieron clasificar los signos vitales en los estados fisiológicos: crítico, inestable y estable; sin embargo, es necesario hacer alusión a que se crearon modelos especializados para manejar el desbalance existente entre las diferentes clases, donde DTC y RFC obtuvieron resultados similares y una mayor tasa de acierto en comparación con el LRC. Además, los modelos LSTM y GRU fueron capaces de predecir

los estados fisiológicos mencionados en una ventana de tiempo determinada de 60 minutos, con resultados similares.

El uso de las tecnologías de la información y la comunicación, y principalmente el uso del aprendizaje automático y la inteligencia artificial tienen la capacidad de potencializar el monitoreo y seguimiento constante de los pacientes. La monitorización constante podría aportar información valiosa sobre una persona, como: el estado fisiológico, estado emocional, hábitos, y demás información que crea un gran número de campos de acción de la inteligencia artificial; con el objetivo de mejorar la calidad de vida de las personas.

7.2 Recomendaciones

Es necesaria la continua recolección de los datos desde diferentes fuentes. Esto permitirá a los modelos ser cada vez más precisos en las predicciones que se realicen. Sin embargo, es necesario automatizar el preprocesamiento de la información, ya que el aumento de la información producirá retos en cuanto al manejo de los grandes volúmenes de información.

Esta investigación planteó la creación de modelos especializados y ajuste a los pesos de las clases en el proceso de clasificación para abordar el problema del desbalanceo en el conjunto de datos. Sin embargo, hay diferentes alternativas para tratar este problema, como: remuestreo, creación de muestras sintéticas y métodos de penalización, entre otros.

Puesto que los pacientes en UCIP cuentan con signos vitales, en su mayoría, fuera de los rangos *normales* descritos por la literatura, cada paciente tendrá una serie de características propias que no se podrían generalizar a partir de pacientes con la misma edad. Por lo tanto, se propone como un trabajo futuro crear una red neuronal de predicción que cuente con dos modelos: uno pre-entrenado y otro que se entrene a partir de los datos del paciente. Es decir, que su entrenamiento se realice en un período de tiempo corto con los datos del paciente, con el objetivo de que el modelo sea capaz de encontrar patrones particulares para ese paciente.

7.2.1 Trabajos futuros

Los estados fisiológicos de un paciente en una unidad de cuidados intensivos pueden llegar a ser un reto enorme de predicción con precisión debido a que cada individuo cuenta con particularidades que no se pueden reflejar claramente en los signos vitales. Por lo tanto, es importante seguir realizando implementaciones de aprendizaje automático con el fin de clasificar y predecir el estado fisiológico de los pacientes.

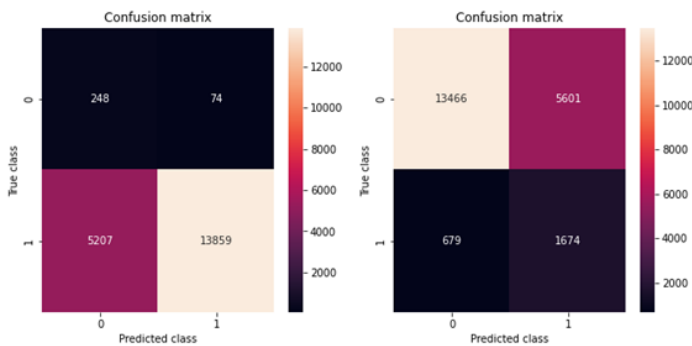
Es importante la automatización de la recolección y correcto almacenamiento de los datos en tiempo real de los signos vitales del paciente. Esto permitiría que la cantidad de registros aumente considerablemente y evita errores humanos relacionados con la transcripción de los datos. La creación de esta base de datos podría permitir tener registros de diferentes fuentes y aumentar las muestras por grupo y por estado, evitando así los sesgos presentados en esta investigación, donde en su mayoría el estado era “inestable” y cerca de dos terceras de los pacientes tenían edades inferiores a un año. Así mismo, a través de diferentes algoritmos, realizar un preprocesamiento y clasificación de la información en otra base de datos en limpio. Este conjunto de datos será el insumo para entrenar los diferentes modelos de clasificación y predicción que se implementen en el futuro.

A. Anexo: Matrices de confusión de los modelos especializados

A continuación, se presentan las matrices de confusión de los modelos especializados implementados usando las técnicas: DTC, RFC y LRC.

DTC

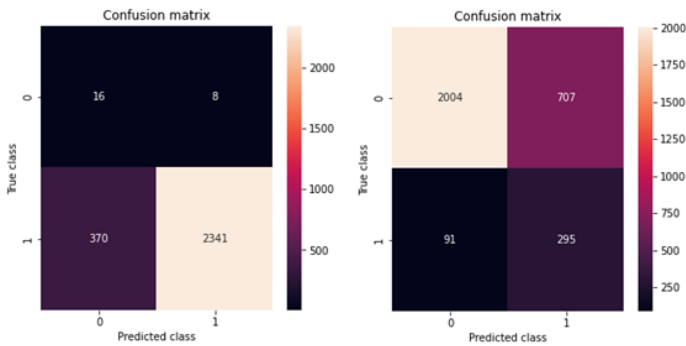
Grupo 1



Especializado en crítico

Especializado en estable

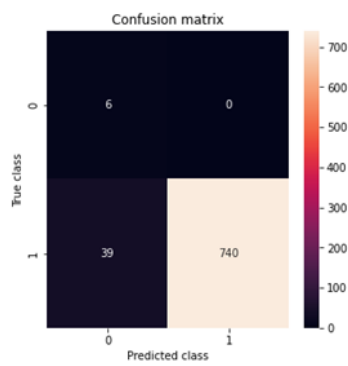
Grupo 2



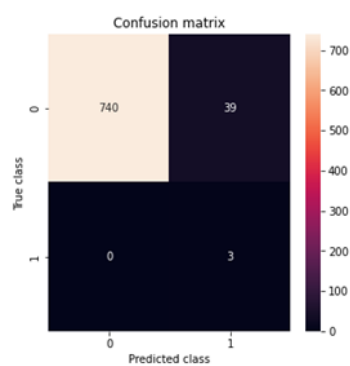
Especializado en crítico

Especializado en estable

Grupo 3

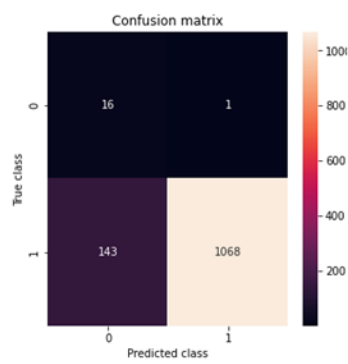


Especializado en crítico

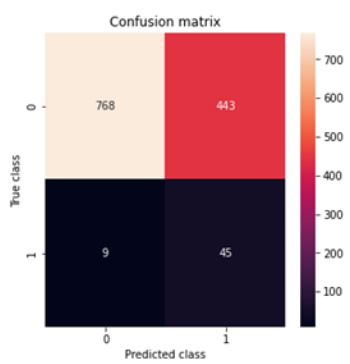


Especializado en estable

Grupo 4



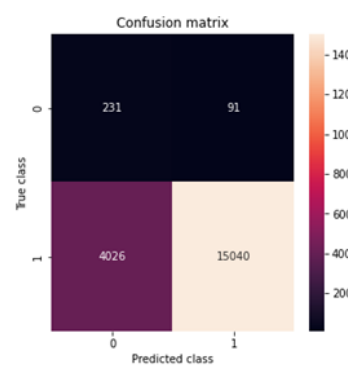
Especializado en crítico



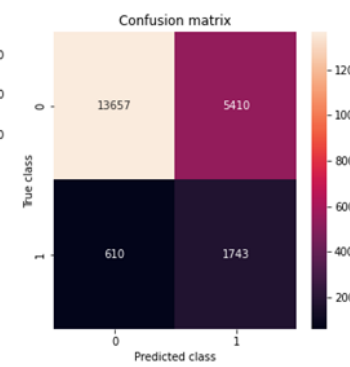
Especializado en estable

RFC

Grupo 1

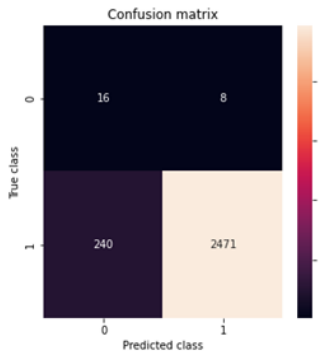


Especializado en crítico

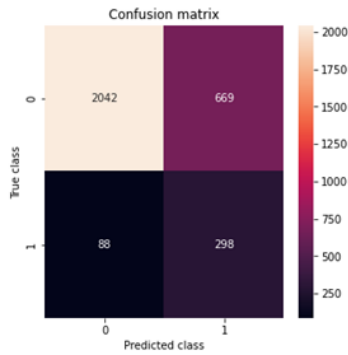


Especializado en estable

Grupo 2

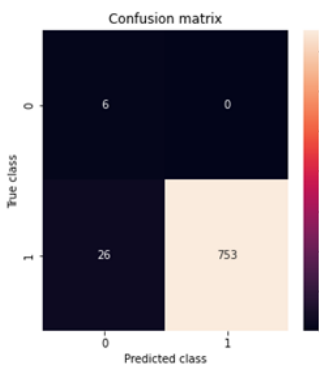


Especializado en crítico

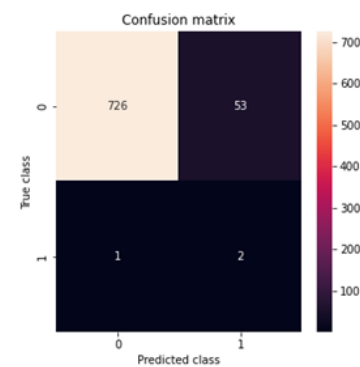


Especializado en estable

Grupo 3

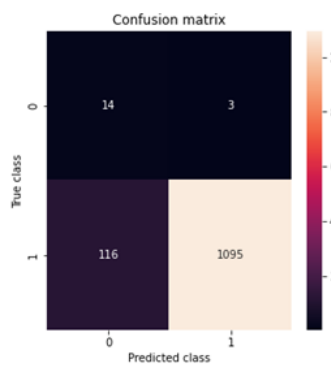


Especializado en crítico

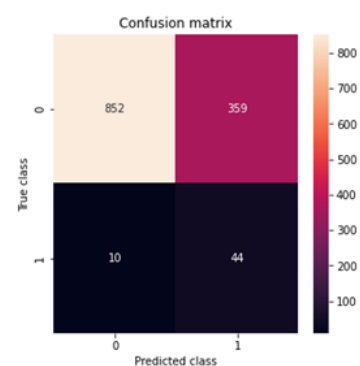


Especializado en estable

Grupo 4



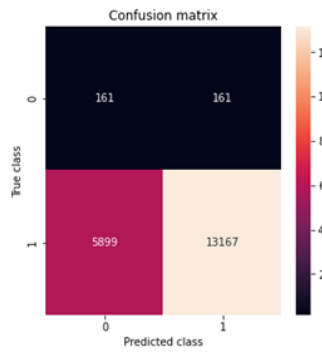
Especializado en crítico



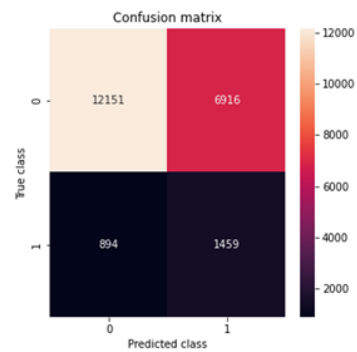
Especializado en estable

LRC

Grupo 1

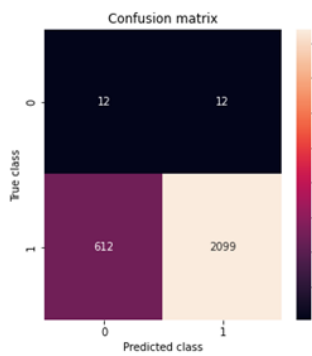


Especializado en crítico

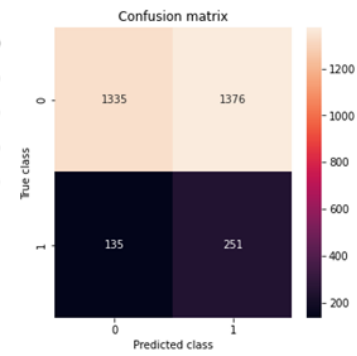


Especializado en estable

Grupo 2

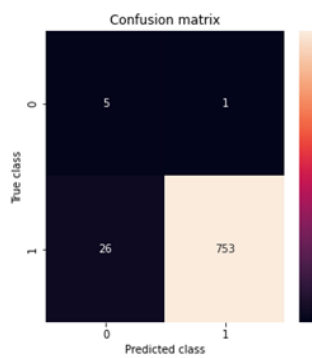


Especializado en crítico

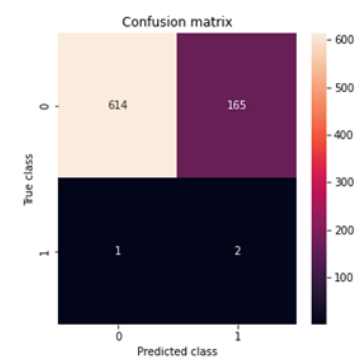


Especializado en estable

Grupo 3

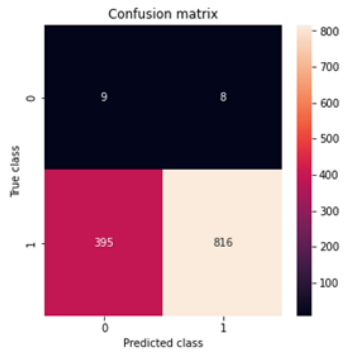


Especializado en crítico

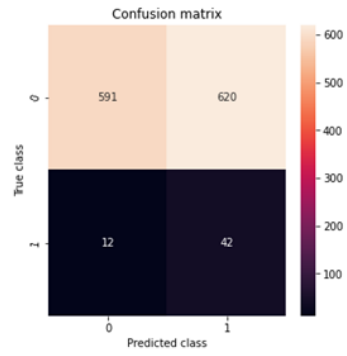


Especializado en estable

Grupo 4



Especializado en crítico



Especializado en estable

B. Anexo: Resultados de predicciones para cada uno de los pacientes

En este anexo se presentan predicciones de ejemplo para los grupos definidos usando LSTM y GRU. Es importante mencionar que incluir todos los registros podría extender demasiado este documento, por lo tanto, se agregó un máximo de 5 registros por paciente, tomando los primeros registros.

LSTM

Grupo 1

Paciente	Minuto	Predicción	Real	Acertó
1	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
2	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
3	5	Estable	Estable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Estable	Inestable	NO
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
5	5	Estable	Inestable	NO
	10	Inestable	Estable	NO
	20	Inestable	Estable	NO

	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
6	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI

Grupo 2

Paciente	Minuto	Predicción	Real	Acertó
4	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Estable	Inestable	NO
	60	Inestable	Inestable	SI
9	5	Estable	Estable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Estable	NO
16	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Estable	Inestable	NO
	60	Inestable	Inestable	SI
21	5	Estable	Estable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Estable	NO
28	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Estable	NO
	30	Inestable	Inestable	SI

	45	Estable	Inestable	NO
	60	Inestable	Estable	NO

Grupo 3

Paciente	Minuto	Predicción	Real	Acertó
8	5	Inestable	Inestable	SI
	10	Estable	Inestable	NO
	20	Inestable	Inestable	SI
	30	Estable	Inestable	NO
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
44	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
75	5	Inestable	Estable	NO
	10	Estable	Inestable	NO
	20	Inestable	Inestable	SI
	30	Estable	Inestable	NO
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI

Grupo 4

Paciente	Minuto	Predicción	Real	Acertó
20	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
32	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
37	5	Inestable	Inestable	SI

	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
63	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
87	60	Inestable	Inestable	SI
	5	Inestable	Estable	NO
	10	Inestable	Estable	NO
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI

GRU

Grupo 1

Paciente	Minuto	Predicción	Real	Acertó
1	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
2	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
3	5	Estable	Estable	SI
	10	Inestable	Inestable	SI
	20	Estable	Inestable	NO
	30	Estable	Inestable	NO
	45	Inestable	Inestable	SI

	60	Estable	Inestable	NO
5	5	Inestable	Inestable	SI
	10	Inestable	Estable	NO
	20	Inestable	Estable	NO
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
6	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI

Grupo 2

Paciente	Minuto	Predicción	Real	Acertó
4	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
9	5	Crítico	Inestable	NO
	10	Crítico	Inestable	NO
	20	Inestable	Inestable	SI
	30	Estable	Inestable	NO
	45	Estable	Inestable	NO
	60	Inestable	Inestable	SI
16	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
21	5	Crítico	Inestable	NO
	10	Estable	Inestable	NO
	20	Estable	Inestable	NO
	30	Estable	Inestable	NO
	45	Estable	Inestable	NO

	60	Inestable	Inestable	SI
28	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI

Grupo 3

Paciente	Minuto	Predicción	Real	Acertó
8	5	Crítico	Inestable	NO
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
44	5	Crítico	Inestable	NO
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
75	5	Crítico	Estable	NO
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI

Grupo 4

Paciente	Minuto	Predicción	Real	Acertó
20	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
32	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI

	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
37	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
63	5	Inestable	Inestable	SI
	10	Inestable	Inestable	SI
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI
87	5	Inestable	Estable	NO
	10	Inestable	Estable	NO
	20	Inestable	Inestable	SI
	30	Inestable	Inestable	SI
	45	Inestable	Inestable	SI
	60	Inestable	Inestable	SI

C. Anexo: Enlace a la aplicación y repositorios del código fuente

En el siguiente enlace podrá encontrar la aplicación desplegada. Ahí podrá descargar un archivo de prueba con datos, para cargar el archivo en Excel y hacer la predicción de un paciente inestable.

- Enlace de la aplicación: <https://ucip-un.azurewebsites.net/predictions>

A continuación, podrá encontrar el código fuente desarrollado a lo largo de este trabajo.

- Frontend: <https://github.com/diegobaqt/ucip-frontend>
- Backend: <https://github.com/diegobaqt/ucip-backend>
- Model API: <https://github.com/diegobaqt/ucip-model-api>
- Python Notebook: <https://github.com/diegobaqt/ucip-notebook>

Bibliografía

Abromavičius, V. y Serackis, A. (2019). Sepsis Prediction Model Based on Vital Signs Related Features.

Alghatani, K., Ammar, N., Rezgui, A. y Shaban-Nejad, A. (2021). Predicting intensive care unit length of stay and mortality using patient vital signs: Machine learning model development and validation. *JMIR Medical Informatics*, 9.

Amer, A.Y.A., Vranken, J., Wouters, F., Mesotten, D., Vandervoort, P., Storms, V., Luca, S., Vanrumste, B. y Aerts, J.-M. (2019). Feature engineering for ICU mortality prediction based on hourly to bi-hourly measurements. *Applied Sciences (Switzerland)*, 9.

AnalystPrep (s. f.). *Machine Learning Process*. [Ilustración].

https://analystprep.com/study-notes/wp-content/uploads/2020/04/MachineLearningApproaches_img1.jpg

Atashi, A., Ahmadian, L., Rahmatinezhad, Z., Miri, M., Nazeri, N. y Eslami, S. (2018). Development of a national core dataset for the Iranian ICU patients outcome prediction: A comprehensive approach. *Journal of Innovation in Health Informatics*, 25, 71-76.

Awad, A., Bader-El-Den, M., McNicholas, J. y Briggs, J. (2017). Early hospital mortality prediction of intensive care unit patients using an ensemble learning approach. *International Journal of Medical Informatics*, 108, 185-195.

Barnes, R., Clarke, D., Farina, Z., Sartorius, B., Brysiewicz, P., Laing, G., Bruce, J. y Kong, V. (2018). Vital sign based shock scores are poor at triaging South African trauma patients. *American Journal of Surgery*, 216, 235-239.

- Bedoya, A.D., Clement, M.E., Phelan, M., Steorts, R.C., O'Brien, C. y Goldstein, B.A. (2019). Minimal Impact of Implemented Early Warning Score and Best Practice Alert for Patient Deterioration. *Critical Care Medicine*, 47, 49-55.
- Bellew, S.D., Cabrera, D., Lohse, C.M. y Bellolio, M.F. (2017). Predicting Early Rapid Response Team Activation in Patients Admitted From the Emergency Department: The PeRRT Score. *Academic Emergency Medicine*, 24, 216-225.
- Brilej, D., Stropnik, D., Lefering, R. y Komadina, R. (2017). Algorithm for activation of coagulation support treatment in multiple injured patients—cohort study. *European Journal of Trauma and Emergency Surgery*, 43, 423-430.
- Brown, A., Ballal, A. y Al-Haddad, M. (2019). Recognition of the critically ill patient and escalation of therapy. *Anaesthesia and Intensive Care Medicine*, 20, 1-5.
- Carlin, C.S., Ho, L.V., Ledbetter, D.R., Aczon, M.D. y Wetzel, R.C. (2018). Predicting individual physiologically acceptable states at discharge from a pediatric intensive care unit. *Journal of the American Medical Informatics Association*, 25, 1600-1607.
- Chen, T., Xu, J., Ying, H., Chen, X., Feng, R., Fang, X., Gao, H. y Wu, J. (2019). Prediction of Extubation Failure for Intensive Care Unit Patients Using Light Gradient Boosting Machine. *IEEE Access*, 7, 150960-150968.
- De Pasquale, M., Moss, T.J., Cerutti, S., Calland, J.F., Lake, D.E., Moorman, J.R. y Ferrario, M. (2017). Hemorrhage prediction models in surgical intensive care: Bedside monitoring data adds information to lab values. *IEEE Journal of Biomedical and Health Informatics*, 21, 1703-1710.
- Ding, Y., Ma, X. y Wang, Y. (2018). Health status monitoring for ICU patients based on locally weighted principal component analysis. *Computer Methods and Programs in Biomedicine*, 156, 61-71.
- Fagerström, J., Bång, M., Wilhelms, D. y Chew, M.S. (2019). LiSep LSTM: A Machine Learning Algorithm for Early Detection of Septic Shock. *Scientific Reports*, 9.
- Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Techniques and Tools to Build Learning Machines*. Van Duuren Media.

- Ge, W., Huh, J.-W., Park, Y.R., Lee, J.-H., Kim, Y.-H. y A. Turchin (2018). An Interpretable ICU Mortality Prediction Model Based on Logistic Regression and Recurrent Neural Networks with LSTM units. AMIA Annual Symposium proceedings. AMIA Symposium, 2018, 460-469.
- Haider, A., Con, J., Prabhakaran, K., Anderson, P., Policastro, A., Feeney, J. y Latifi, R. (2019). Developing a simple clinical score for predicting mortality and need for ICU in trauma patients. *American Surgeon*, 85, 733-737.
- Hever, G. , Cohen, L., O'Connor, M.F., Matot, I., Lerner, B. y Bitan, Y. (2020). Machine learning applied to multi-sensor information to reduce false alarm rate in the ICU. *Journal of Clinical Monitoring and Computing*, 34, 339-352.
- Hever, G., Cohen, L., O'Connor, M. F., Matot, I., Lerner, B., & Bitan, Y. (2019). Machine learning applied to multi-sensor information to reduce false alarm rate in the ICU. *Journal of Clinical Monitoring and Computing*, 34(2), 339–352.
<https://doi.org/10.1007/s10877-019-00307-x>
- Hirano, Y., Kondo, Y., Hifumi, T., Yokobori, S., Kanda, J., Shimazaki, J., Hayashida, K., Moriya, T., Yagi, M., Takauji, S., Tanaka, H. y Yaguchi, A. (2021). Machine learning-based mortality prediction model for heat-related illness. *Scientific Reports*, 11.
- IBM Cloud Education. (2020, 15 julio). *Machine Learning*. IBM.
<https://www.ibm.com/cloud/learn/machine-learning>
- Izquierdo, L. M., Nino, L. F., & Prieto Rojas, J. (2020). Modeling the vital sign space to detect the deterioration of patients in a pediatric intensive care unit. 16th International Symposium on Medical Information Processing and Analysis. Published.
<https://doi.org/10.1117/12.2579629>
- Izquierdo, L. M., Nino, L. F. (2021). Modelamiento del espacio de signos vitales para detectar el deterioro de los pacientes en una unidad de cuidados intensivos. Universidad Nacional de Colombia.

- Karimi Moridani, M. y Haghghi Bardineh, Y. (2018). Presenting an efficient approach based on novel mapping for mortality prediction in intensive care unit cardiovascular patients. *MethodsX*, 5, 1291-1298.
- Kefi, Z., Aloui, K. y Naceur, M.S. (2019). The early prediction of neonates mortality in Intensive Care Unit. (pp. 304-306).
- Koyner, J.L., Carey, K.A., Edelson, D.P. y Churpek, M.M. (2018). The development of a machine learning inpatient acute kidney injury prediction model. *Critical Care Medicine*, 46, 1070-1077.
- Li, X., Ge, P., Zhu, J., Li, H., Graham, J., Singer, A., Richman, P.S. y Duong, T.Q. (2020). Deep learning prediction of likelihood of ICU admission and mortality in COVID-19 patients using clinical variables. *PeerJ*, 8.
- Malycha, J., Farajidavar, N., Pimentel, M.A.F., Redfern, O., Clifton, D.A., Tarassenko, L., Meredith, P., Prytherch, D., Ludbrook, G., Young, J.D. y Watkinson, P.J. (2019). The effect of fractional inspired oxygen concentration on early warning score performance: A database analysis. *Resuscitation*, 139, 192-199.
- Marafino, B.J., Park, M., Davies, J.M., Thombley, R., Luft, H.S., Sing, D.C., Kazi, D.S. Dejong, C., Boscardin, W.J., Dean, M.L. y Dudley, R.A. (2018). Validation of Prediction Models for Critical Care Outcomes Using Natural Language Processing of Electronic Health Record Data. *JAMA Network Open*, 1.
- Masud, M.M., Cheratta, M. y Harahsheh, A.R.A. (2018). Survival prediction of ICU patients using knowledge intensive data grouping and selection.
- Matrices de confusión - Training. (s. f.). Microsoft Learn. <https://learn.microsoft.com/es-es/training/modules/machine-learning-confusion-matrix/2-confusion-matrices>
- Mayampurath, A., Jani, P., Dai, Y., Gibbons, R., Edelson, D. y Churpek, M.M. (2020). A vital sign-based model to predict clinical deterioration in hospitalized children. *Pediatric Critical Care Medicine*, 820-826.
- Melinosky, C., Yang, S., Hu, P., Li, H., Miller, C.H.T., Khan, I., Mackenzie, C., Chang, W.-T., Parikh, G., Stein, D., Stein, D. y Badjatia, N. (2018). Continuous vital sign analysis

to predict secondary neurological decline after traumatic brain injury. *Frontiers in Neurology*, 9.

Messinger, A.I., Bui, N., Wagner, B.D., Szeffler, S.J., Vu, T. y Deterding, R.R. (2019). Novel pediatric-automated respiratory score using physiologic data and machine learning in asthma. *Pediatric Pulmonology*, 54, 1149-1155.

Monteiro, F., Meloni, F., Baranauskas, J.A. y Macedo, A.A. (2020). Prediction of mortality in Intensive Care Units: a multivariate feature selection. *Journal of Biomedical Informatics*, 107.

Nielsen, A.B., Thorsen-Meyer, H.-C., Belling, K., Nielsen, A.P., Thomas, C.E., Chmura, P.J., Lademann, M., Moseley, P.L., Heimann, M., Dybdahl, L., Perner, A. y Brunak, S. (2019). Survival prediction in intensive-care units based on aggregation of long-term disease history and acute physiology: a retrospective study of the Danish National Patient Registry and electronic patient records. *The Lancet Digital Health*, 1, e78e89.

Parreco, J., Soe-Lin, H., Parks, J.J., Byerly, S., Chatoor, M., Buicko, J.L., Namias, N. y Rattan, R. (2019). Comparing machine learning algorithms for predicting acute kidney injury. *American Surgeon*, 85, 725-729.

Platzi: Cursos online profesionales de tecnología. (2018). Platzi.

https://platzi.com/tutoriales/1269-probabilidad-estadistica/2308-coeficiente-de-correlacion-que-es-y-para-que-sirve/?utm_source=google

Pollard, T.J., Johnson, A.E.W., Raffa, J.D., Celi, L.A., Mark, R.G. y Badawi, O. (2018). The eICU collaborative research database, a freely available multi-center database for critical care research. *Scientific Data*, 5.

R.G. Barry, T.T. Wolbert, F.B. Mozaffari, P.D. Ray, E.C. Thompson, T.W. Gress, & D.A. Denning (2019). Comparison of geriatric trauma outcomes when admitted to a medical or surgical service after a fall. *Journal of Surgical Research*, 233, 391-396.

- Radhachandran, A., Garikipati, A., Zelin, N.S., Pellegrini, E., Ghandian, S., Calvert, J., Hoffman, J., Mao, Q. y Das, R. (2021). Prediction of short-term mortality in acute heart failure patients using minimal electronic health record data. *BioData Mining*, 14.
- Rubin, J., Potes, C., Xu-Wilson, M., Dong, J., Rahman, A., Nguyen, H. y Moromisato, D. (2018). An ensemble boosting model for predicting transfer to the pediatric intensive care unit. *International Journal of Medical Informatics*, 112, 15-20.
- Salahuddin, N., Shafquat, A., Marashly, Q., Zaza, K.J., Sharshir, M., Khurshid, M., Ali, Z., Malgapo, M., Jamil, M.G., Shoukri, M., Hijazi, M. y Al-Ghamdi, B. (2018). Increases in Heart Rate Variability Signal Improved Outcomes in Rapid Response Team Consultations: A Cohort Study. *Cardiology Research and Practice*, 2018.
- Satchidanand, N., Servoss, T.J., Singh, R., Bosinski, A.M., Tirpak, P., Horton, L.L. y Naughton, B.J. (2018). Development of a Risk Tool to Support Discussions of Care for Older Adults Admitted to the ICU With Pneumonia. *American Journal of Hospice and Palliative Medicine*, 35, 1201-1206.
- Shamout, F.E., Zhu, T., Sharma, P., Watkinson, P.J. y Clifton, D.A. (2020). Deep Interpretable Early Warning System for the Detection of Clinical Deterioration. *IEEE Journal of Biomedical and Health Informatics*, 24, 437-446.
- Tamir, M. (2021, 29 enero). *What Is Machine Learning?* UCB-UMT. <https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/>
- Timm, F.P., Zaremba, S., Grabitz, S.D., Farhan, H.N., Zaremba, S., Siliski, E., Shin, C.H., Muse, S., Friedrich, S., Mojica, J.E., Ramachandran, S.-K. y Eikermann, M. (2018). Effects of opioids given to facilitate mechanical ventilation on sleep apnea after extubation in the intensive care unit. *Sleep*, 41.
- Tsur, E., Last, M., Garcia, V.F., Udassin, R., Klein, M. y Brotfain, E. (2019). Hypotensive episode prediction in icus via observation window splitting. *LNAI*, Vol. 11053.
- Viegas, R., Salgado, C.M., Curto, S., Carvalho, J.P., Vieira, S.M. y Finkelstein, S.N. (2017). Daily prediction of ICU readmissions using feature engineering and ensemble fuzzy modeling. *Expert Systems with Applications*, 79, 244-253.

Wang, Y., Wei, Y., Yang, H., Li, J., Zhou, Y. y Wu, Q. (2020). Utilizing imbalanced electronic health records to predict acute kidney injury by ensemble learning and time series model. *BMC Medical Informatics and Decision Making*, 20.

Wellner, B., Grand, J., Canzone, E., Coarr, M., Brady, P.W., Simmons, J., Kirkendall, E., Dean, N. Kleinman, M. y Sylvester, P. (2017). Predicting unplanned transfers to the intensive care unit: A machine learning approach leveraging diverse clinical elements. *JMIR Medical Informatics*, 5.

Yoon, J.H., Jeanselme, V., Dubrawski, A., Hravnak, M., Pinsky, M.R. y Clermont, G. (2020). Prediction of hypotension events with physiologic vital sign signatures in the intensive care unit. *Critical Care*, 24.

Zachariasse J.M., Van Der Lee, D., Seiger, N., De Vos-Kerkhof, E., Oostenbrink, R. y Moll, H.A. (2017). The role of nurses' clinical impression in the first assessment of children at the emergency department. *Archives of Disease in Childhood*, 102, 1052-1056.

Zimmerman, L.P., Reyfman, P.A., Smith, A.D.R., Zeng, Z., Kho, A., Sanchez-Pinto, L.N. y Luo, Y. (2019). Early prediction of acute kidney injury following ICU admission using a multivariate panel of physiological measurements. *BMC Medical Informatics and Decision Making*, 19.