Universitat Oberta de Catalunya

uoc.edu

Modelo de predicción de mortalidad en la insuficiencia respiratoria aguda: análisis de Registros Electrónicos de Salud.

Eduardo González Constán

Máster en Bioinformática y Bioestadística

Área 2 – Subárea 2: Análisis de datos.

Nombre Consultor/a: Nuria Pérez Álvarez

Nombre Profesor/a responsable de la asignatura: Carles Ventura Royo

Junio 2022

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Modelo de predicción de mortalidad en la insuficiencia respiratoria aguda: análisis de Registros Electrónicos de Salud.*
Nombre del autor:	*Eduardo González Constán*
Nombre del consultor/a:	*Nuria Pérez Álvarez*
Nombre del PRA:	*Carles Ventura Royo*
Fecha de entrega (mm/aaaa):	06/2022
Titulación:	*Máster en Bioestadística y Bioinformática*
Área del Trabajo Final:	*Área 2 – Subárea 2: Análisis de datos.*
Idioma del trabajo:	Español
Número de créditos:	15
Palabras clave	*electronic health record, real world evidence, acute respiratory faillure.*

Resumen del Trabajo:

En los últimos años se ha ido generalizando el uso de datos de salud en formato electrónico (EHR) que se han recogido de forma masiva en sistemas EHR. Aunque inicialmente dichos datos tenían una función puramente administrativa se han ido extendiendo sus objetivos hasta abarcar tanto un uso médico como de investigación. Un ejemplo de EHR lo constituye el conjunto de datos MIMIC III, que recoge datos de salud de más de 55000 pacientes ingresados en cuidados intensivos. Este estudio se ha centrado en obtener un modelo de predicción de mortalidad a 30 días en pacientes ingresados con insuficiencia respiratoria aguda mediante técnicas de machine learning (ML) y comparar su resultado con 2 scores de uso habitual en UCI: OASIS y SAPS II. Para encontrar el mejor modelo de clasificación se han usado tanto algoritmos de ML como redes neuronales profundas. El mejor modelo obtenido en el grupo test se corresponde con la regresión logística, logrando un AUC de 0.71. Este resultado es ligeramente superior al obtenido con los scores SAPS II y OASIS (AUC 0.69 y 0.63, respectivamente). Podemos concluir, por tanto, que el uso de técnicas de ML puede mejorar los scores habituales de predicción de mortalidad.

No obstante, aún existe un rendimiento limitado al aplicar dichas técnicas a cohortes específicas de pacientes, por lo que se hacen necesarios más estudios de este tipo en los que se aplique ML para predecir mortalidad a pacientes que comparten procesos patológicos similares, más que aplicar dichas técnicas de forma general.

Abstract:

In recent years, the use of health data in electronic format (EHR) has become widespread and has been massively collected in EHR systems. Although initially said data had a purely administrative function, its objectives have been extended to cover both medical and research use. An example of an EHR is the MIMIC III dataset, which collects health data from more than 55,000 patients admitted to intensive care. This study has focused on obtaining a 30-day mortality prediction model in hospitalized patients with acute respiratory failure using machine learning techniques (ML) and comparing the results with two scores commonly used in the ICU: OASIS and SAPS II. To find the best classification model, both ML algorithms and deep neural networks have been used. The best model obtained in the test group corresponds to logistic regression, achieving an AUC of 0.71. This result is slightly higher than that obtained with the SAPS II and OASIS scores (AUC 0.69 and 0.63, respectively). We can conclude, therefore, that the use of ML techniques can improve the usual mortality prediction scores. However, there is still limited performance when applying these techniques to specific cohorts of patients, so more studies of this type are needed in which ML is applied to predict mortality in patients who share similar pathological processes, rather than applying these techniques. generally.

ÍNDICE

LISTA DE FIGURAS.

LISTA DE TABLAS.

1. RESUMEN.

En los últimos años se ha ido generalizando el uso de datos de salud en formato electrónico (EHR) que se han recogido de forma masiva en sistemas EHR. Aunque inicialmente dichos datos tenían una función puramente administrativa se han ido extendiendo sus objetivos hasta abarcar tanto un uso médico como de investigación. Un ejemplo de EHR lo constituye el conjunto de datos MIMIC III, que recoge datos de salud de más de 55000 pacientes ingresados en cuidados intensivos. Este estudio se ha centrado en obtener un modelo de predicción de mortalidad a 30 días en pacientes ingresados con insuficiencia respiratoria aguda mediante técnicas de machine learning (ML) y comparar su resultado con 2 scores de uso habitual en UCI: OASIS y SAPS II. Para encontrar el mejor modelo de clasificación se han usado tanto algoritmos de ML como redes neuronales profundas. El mejor modelo obtenido en el grupo test se corresponde con la regresión logística, logrando un AUC de 0.71. Este resultado es ligeramente superior al obtenido con los scores SAPS II y OASIS (AUC 0.69 y 0.63, respectivamente). Podemos concluir, por tanto, que el uso de técnicas de ML puede mejorar los scores habituales de predicción de mortalidad. No obstante, aún existe un rendimiento limitado al aplicar dichas técnicas a cohortes específicas de pacientes, por lo que se hacen necesarios más estudios de este tipo en los que se aplique ML para predecir mortalidad a pacientes que comparten procesos patológicos similares, más que aplicar dichas técnicas de forma general.

2. INTRODUCCIÓN

2.1 Contexto y justificación del trabajo.

Los registros electrónicos de salud (o electronic health records - EHR -) son ya una realidad en la práctica diaria en las instituciones sanitarias. Inicialmente se utilizaron con fines administrativos, pero con el tiempo se ha ampliado su uso sustituyendo a la clásica historia clínica (1). La explotación de los datos recogidos en los sistemas EHR con fines de investigación es un área que ha crecido enormemente desde el año 2013, ya que permite acceder a una gran cantidad de datos de salud impensables con los métodos de investigación tradicionales. Por todo ello el análisis de este tipo de datos presenta un enorme atractivo, a la vez que un gran potencial, ya que permite obtener evidencia científica (2) de una manera más directa y rápida respecto a la investigación tradicional (3,4).

2.2 Descripción general.

Con este trabajo se pretende obtener conocimiento aprovechando algunos de los EHR disponibles para su uso en la red. Para ello en primer lugar se revisará el estado actual de la cuestión acerca de los EHR. Posteriormente se analizará una cohorte de pacientes seleccionada del dataset *Medical Information Mart for Intensive Care* (MIMIC III) (5) con el fin de obtener evidencia científica sobre dichos pacientes. Finalmente se describirán las ventajas y también las limitaciones que presentan este tipo de estudios.

2.3 Objetivos.

Los objetivos del presente trabajo los he dividido en objetivos generales y objetivos específicos, y se detallan en los siguientes dos apartados. La idea fundamental es en primer lugar recoger la evidencia disponible en la actualidad sobre la temática estudiada (EHR e IRA) para posteriormente extraer la evidencia disponible del dataset MIMIC III en relación a la mortalidad a 30 días en pacientes ingresados con IRA.

2.3.1 Objetivos generales.

- Recoger toda la evidencia disponible actualmente sobre los EHR, especialmente en lo referente a la generación de nueva evidencia científica.
- Realizar un estudio de investigación original sobre pacientes ingresados por insuficiencia respiratoria aguda (IRA) en una unidad de cuidados críticos procedentes del dataset MIMIC III, y discutir las ventajas y limitaciones de la evidencia obtenida.
- Investigación de los procesos de manejo de datos que permiten, a partir de un EHR como MIMIC III, obtener los datos requeridos para su análisis posterior.

2.3.2 Objetivos específicos.

1. Elaborar un estado de la cuestión actualizado sobre los EHR.
2. Elaborar un estado de la cuestión actualizado sobre IRA.

3. Acceso completo a las tablas del conjunto de datos MIMIC III y procesamiento posterior.

4. Exploración de los diferentes métodos estadísticos para el análisis de los datos.

5. Elaborar un modelo de predicción de mortalidad a 30 días sobre los pacientes con IRA a partir de los datos de exploración y de laboratorio obtenidos en las primeras 24 horas del ingreso.

6. Determinar la rentabilidad de dicho modelo, y compararlo con la predicción obtenida mediante otros scores de severidad como son: el Oxford Acute Severity of Illness Score (OASIS) (6) y el Simplified Acute Physiologic Score (SAPS II) (7)

2.4 Enfoque y método a seguir.

La base de datos MIMIC III está constituida por 25 tablas relacionadas entre sí. El acceso a dichas tablas puede hacerse de dos maneras: descargarlas en local o trabajar online mediante BigQuery. Para la realización del trabajo fin de máster (TFM) es preferible trabajar online, ya que las tablas están accesibles en Google Cloud Platform mediante BigQuery. La manipulación y procesamiento de las tablas se ha realizado mediante SQL standard con el fin de obtener una tabla final con la cohorte deseada. Finalmente, la tabla obtenida se importa a RStudio para realizar el análisis de los datos y la elaboración del modelo.

2.5 Planificación.

2.5.1 Tareas.

Los objetivos descritos anteriormente se pueden desglosar en las siguientes tareas:

1. Elaboración de un estado de la cuestión sobre los EHR. Duración aproximada de la tarea: 48 horas. Este punto está dividido en las siguientes subtareas:

   - Búsqueda bibliográfica.

   - Lectura de la bibliografía y realización del estado de la cuestión.

2. Elaboración de un breve estado de la cuestión sobre la IRA. Duración aproximada de la tarea: 48 horas. Este apartado está dividido en las siguientes subtareas:

   - Búsqueda bibliográfica.

   - Lectura de la bibliografía y realización del estado de la cuestión.

3. Realización del curso *Human Research. Data or Specimens Only Research* y posterior acreditación que permite acceder al dataset MIMIC III. Duración aproximada de la tarea: 4 semanas.

- Descripción de la base de datos MIMIC III. Duración aproximada de la tarea: 72 horas.
- Obtención de datos a partir de las tablas de MIMIC III mediante BigQuery. Duración aproximada de la tarea: 20 días. Subtareas que la conforman:
- Exploración del entorno Google-Cloud así como familiarización con el uso de BigQuery.
- Exploración de las tablas del dataset MIMIC III.
- Obtención de los datos necesarios, mediante queries, para realizar el análisis estadístico posterior.

4. Exploración y análisis de los datos obtenidos en R. Duración aproximada de la tarea: 24 días. Subtareas que la conforman:
   - Valoración de datos perdidos.
   - Imputación de variables.
   - Exploración del dataset.
5. Implementación del modelo de predicción y comparación respecto a los scores OASIS y SAPS II. Duración aproximada de la tarea: 7 días. Subtareas que la conforman:
   - Desarrollo de modelos de clasificación mediante diferentes algoritmos de machine learning (ML).
   - Desarrollo de modelos de clasificación mediante deep learning.
   - Comparación de los resultados obtenidos respecto a los scores OASIS y SAPS II.
6. Resultados y conclusiones finales del estudio de investigación. Duración aproximada de la tarea: 15 días.
7. Presentación de los resultados. Duración aproximada de la tarea: 5 horas.

2.5.2 Calendario.

En la figura 1 se muestra el calendario seguido para la realización del TFM. Hay que tener en cuenta que el acceso al dataset MIMIC III implica la realización del CITI program course *Human Research. Data or Specimens Only Research* impartido por el *Massachusetts Institute of Technology Affiliates*. Dicho curso es obligatorio para obtener la acreditación que permita acceder al dataset MIMIC III, ya sea en local u online mediante BigQuery. Una vez realizado el curso el acceso no es inmediato ya que hay que remitir el certificado del curso y el proyecto de investigación a Physionet. La acreditación que permite acceder al dataset suele demorarse varias semanas. Hemos

de considerar que la gestión de las diferentes tablas en MIMIC III es un proceso laborioso que ocupará gran parte del desarrollo del trabajo de investigación. La redacción de la memoria se ha ido realizando de forma concomitante a las tareas que aparecen en el calendario.

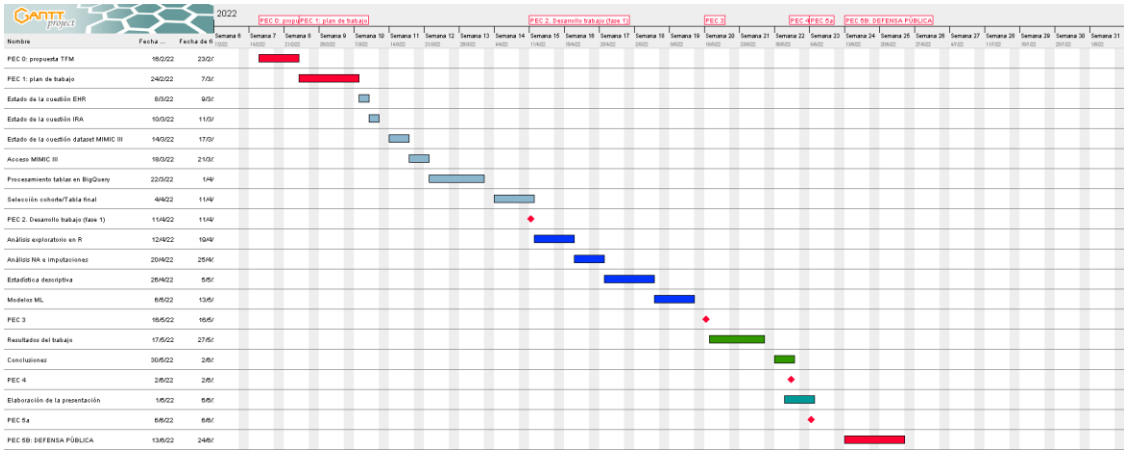


Figura 1. Calendario propuesto para la realización del TFM.

2.5.3 Hitos.

Los hitos están marcados en rojo en el calendario, y coinciden con cada una de las entregas de las PECs. Cada uno de los hitos cierra un conjunto de tareas relacionadas, lo que permite tomar decisiones para tareas sucesivas según los resultados obtenidos.

2.5.4 Análisis de riesgos.

A continuación, señalo los riesgos que pueden alterar los diferentes objetivos específicos:

Objetivos	Riesgos
Estado de la cuestión actualizado EHR	No encontrar información adecuada
Acceso al dataset MIMIC III	Dificultades o retrasos para acceder a la versión completa de MIMIC III
Exploración estadística	Datos de mala calidad o con excesivo NA
Elaborar un modelo de predicción	Retrasos en la elaboración de los modelos debido a su complejidad
Determinar la rentabilidad de dicho modelo	Modelo con una rentabilidad inferior a la esperada

Además, de forma global, puede haber retrasos debido a imprevistos como enfermedad, motivos familiares o de trabajo.

2.6 Resultados esperados.

La realización del TFM implica la obtención de los siguientes elementos o productos:

- Plan de trabajo.
- Memoria del TFM.
- Trabajo de investigación. En función de los resultados obtenidos se puede valorar su publicación en alguna revista científica como artículo original.
- Código SQL utilizado en BigQuery para el procesamiento de las tablas.
- Código en R y Python utilizado para el análisis de los datos e implementación del modelo.
- Certificado del curso *Human Research. Data or Specimens Only Research* impartido por el *Massachusetts Institute of Technology Affiliates*.
- Presentación de los resultados-conclusiones del trabajo de investigación.

3. ESTADO DE LA CUESTIÓN.

3.1 Registros Electrónicos de Salud.

3.1.1 Contexto actual.

El auge del BigData tiene una implicación cada vez mayor en la industria de la salud. Sin embargo, el almacenamiento y el manejo de datos de muy diversa índole en diferentes esquemas y formatos suponen todo un desafío para los sistemas de salud debido a su elevada complejidad (8).

Los *registros electrónicos de salud* (EHR) comprenden información muy heterogénea sobre los pacientes (datos numéricos, categóricos, texto libre, etc) (9). Se trata, por tanto, de registros longitudinales de salud, persistentes en el tiempo, guardados en un sistema electrónico que ofrece acceso seguro a los datos de los pacientes (8). Estos sistemas EHR no están bajo el control de las agencias de medicamentos (por ejemplo, la Food and Drug Administration - FDA - o la Agencia Europea del Medicamento - EMA -), sino que pertenecen a los proveedores de salud, organizaciones e instituciones encargadas del cuidado de la salud (10). Los sistemas EHR, cuando están bien diseñados e implementados, deberían facilitar la toma de decisiones clínicas. Por contra, cuando la implementación no es adecuada y presentan pobres rendimientos el uso rutinario de dichos sistemas decrece (11). Por tanto, parece claro que un sistema EHR eficiente es fundamental para mantener la calidad en los cuidados de salud (8).

El uso de los EHR tiene dos objetivos fundamentales (8):

1. Exclusivamente clínico para la consulta de la historia clínica electrónica, establecer tratamientos, diagnóstico, etc.

2. Uso a nivel de investigación. En estos casos no se busca un único paciente, sino subgrupos o cohortes que satisfagan uno o varios criterios de búsqueda.

De hecho, ambos objetivos de los sistemas EHR están íntimamente relacionados entre sí. La mayoría de las veces se aprovechan los EHR con fines clínicos para realizar estudios observacionales. El incremento del uso de los EHR en los últimos años ha sido paralelo a los esfuerzos en utilizar dichos datos con una finalidad investigadora. Hay que tener en cuenta que hasta un 90% de los hospitales en EEUU usan sistemas EHR (1), con la posibilidad de obtener una masiva y amplia variedad de datos clínicos que sería difícil de conseguir mediante métodos tradicionales. Además, la actual pandemia por SARS-Cov 2 ha acelerado aún más si cabe este tipo de estudios, ya que ha permitido el realizar investigación sobre dicha patología invirtiendo mucho menos tiempo que con los métodos habituales (1). Este incremento en la digitalización de los datos de salud tiene como principal objetivo la mejora en la atención médica a la vez que busca una disminución en los costes de salud (12).

Ya hemos comentado que los EHR contienen una amplia variedad de variables relacionadas con el paciente, como datos demográficos, diagnósticos, medicación, signos vitales, laboratorio, pruebas de imagen, anotaciones médicas, etc. Aunque los

campos recogidos habitualmente son datos estructurados, cada vez es más frecuente el uso de una considerable cantidad de datos no estructurados (13). A continuación, repasaré los tipos de datos que con más frecuencia aparecen en los registros EHR:

a) Datos demográficos. Éstos suelen incluir edad, sexo, raza, población, pero no otros de tipo socioeconómico como el nivel de estudios o ingresos económicos.

b) Datos sobre diagnóstico. Suelen estar codificados en base al *International Classification of Diseases* (ICD) (14), el *International Classification of Primary Care* (ICPC) (15) o el *Diagnostic and Statistical Manual of Mental Disorders* (DSM) (16). La codificación más usada es la ICD, o el DSM en caso de enfermedades mentales.

c) Datos sobre medicación. Este tipo de datos suelen tener una calidad aceptable en caso de usarse los estándares en la nomenclatura, como son el *National Drug Code* (NDCs) (17), el *RxNorm* (18), o el *Anatomical Therapeutic Chemical Classification System* (ATC) (19), además de otros.

d) Datos sobre procedimientos. Esta información hace referencia a cualquier intervención ya sea de tipo diagnóstico o terapéutico, que se realiza sobre el paciente (incluye intervenciones quirúrgicas, técnicas radiológicas, pruebas de laboratorio, etc). También existe una nomenclatura estándar para clasificar los diferentes procedimientos realizados, como por ejemplo la *International Classification of Diseases Clinical Modification* (ICD-CM) (14), el *Current Procedural Terminology* (CPT) (20) o la *Healthcare Comon Procedure Coding System* (HCPCS) (21). No obstante, hay que tener en cuenta que la información recogida en esta categoría puede carecer de los detalles necesarios para determinado tipo de estudios (por ejemplo, en caso de los procedimientos quirúrgicos) por lo que es frecuente tener que extraer la información completa de datos nos estructurados provenientes de anotaciones clínicas. Aun así, si el objetivo es simplemente seleccionar cohortes en base a un procedimiento específico este tipo de clasificaciones sí que puede ser suficiente.

e) Datos de laboratorio. Actualmente la mayoría de los proveedores de registros de salud utilizan sus códigos locales tanto para peticiones como para resultados de laboratorio. Esto puede limitar la interoperabilidad entre sistemas EHR diferentes. Además, existe el problema añadido de que se puedan usar diferentes técnicas para la medición del mismo parámetro entre diferentes laboratorios, lo que podría sesgar los resultados.

f) Datos sobre signos vitales. Aquí incluimos parámetros fisiológicos como la frecuencia cardiaca, tensión arterial, frecuencia respiratoria, temperatura, etc. La codificación más extendida para este tipo de medidas es el LOINC (22), si bien no es ampliamente utilizado por los sistemas EHR.

g) Datos socioeconómicos. Este tipo de variables no se recogen en la mayoría de los sistemas EHR. Aunque se han propuesto algunos estándares para su

codificación, lo habitual es que se usen códigos propios para cada sistema EHR. Por todo ello este tipo de datos pueden considerarse de baja calidad.

h) Datos generados por el paciente. Aquí incluimos una amplia gama de variables: patrones de sueño, registros de electrocardiograma (ECG) procedentes de un Holter, niveles de glucosa en sangre, etc. Además, se trata de datos que muestran una amplia variabilidad entre los sistemas EHR.

i) Datos no estructurados. Los EHR contienen una considerable cantidad de datos no estructurados, fundamentalmente como notas de evolución clínica. Este tipo de datos puede contener información clave no recogida en los datos tabulados estructurados. La complejidad de este tipo de datos junto al hecho de que los algoritmos de procesamiento tienen un rendimiento moderado en ellos, han limitado la extracción de información de texto libre.

3.1.2 Evidencia científica a partir de los EHR.

El objetivo de la investigación clínica es conseguir el mayor grado de evidencia científica. Dicha evidencia está basada en una jerarquía de tipo piramidal, con los ensayos clínicos randomizados (ECR) en la cúspide, seguidos por grandes estudios de cohortes, estudios caso-control, y series de casos aislados así como informes u opiniones de expertos en la base de la pirámide (1).

A pesar del alto nivel de evidencia que ofrecen los ECR, también presentan algunas limitaciones como la dificultad en completarlos, su coste o su validez externa. En este sentido, en el año 2016 el *twenty-first Century Cures Act* (23) requirió a la Food and Drug Administration (FDA) que dictase las guías bajo las cuales las empresas farmacéuticas podrían usar la evidencia proveniente de los datos recogidos en los sistemas EHR (denominados *Real World Data* - RWD -) para apoyar la aprobación de nuevas medicinas (24). Así pues la FDA ha definido los términos *Real World Data* (RWD) y *Real World Evidence* (RWE), y su papel en la investigación actual (2). La siguiente tabla recoge las fortalezas y debilidades de la RWE (25).

***Tabla 1**. Fortalezas y debilidades del RWE*

Fortalezas	Debilidades
Diversidad en los pacientes incluidos	No permite evaluar medicamentos antes de su aprobación
Refleja los aspectos reales del tratamiento estudiado	Riesgo de sesgos debido a la no aleatorización
Refleja la idiosincrasia o variaciones locales en la administración del tratamiento	Capacidad limitada para evaluar determinadas respuestas que se dan fuera del ámbito hospitalario
Grandes muestras de pacientes, muy útiles para farmacovigilancia	Capacidad limitada para evaluar factores socioeconómicos como factores de riesgo
Implementación más rápida y barata que los EC	Capacidad limitada de evaluación en enfermedades psiquiátricas

Fortalezas	Debilidades
Puede ayudar a resolver cuestiones durante situaciones de emergencia	Heterogeneidad de los RWD dependiendo de su procedencia
Se puede valorar un amplio rango de resultados	Abundancia de datos perdidos
Puede apoyar la implementación de métodos estadísticos para investigar relaciones causales	El diagnóstico puede resultar confuso en ocasiones

La FDA define los RWD como aquellos datos relacionados con la salud del paciente que están recogidos en una amplia variedad de fuentes: por ejemplo, los sistemas EHR, pero también registros de facturación médica, dispositivos móviles, etc. La FDA habla de RWE como la evidencia generada acerca del uso y potenciales beneficios o riesgos de productos médicos a partir del análisis del RWD. Para la FDA la RWE se genera a partir de estudios con diseños diferentes, como pueden ser los ensayos clínicos o estudios observacionales (prospectivos o retrospectivos) (2). Otros autores consideran la RWE como la evidencia generada por datos procedentes de sistemas EHR (25,26).

El debate acerca de si la evidencia generada a partir de datos provenientes de EHR puede sustituir a la evidencia obtenida de los ensayos clínicos se inicia como respuesta a la burocratización de los ensayos clínicos randomizados (ECR) en los últimos 25 años con la intención de mejorar la seguridad de los participantes (3,4). Esto ha supuesto un hándicap para la industria farmacéutica, ya que ha implicado un encarecimiento de los costes que conlleva la implementación de un ensayo clínico, a la vez que un aumento en su duración. Este debate ha salido incluso de las revistas médicas especializadas, ya que tiene implicaciones en la aprobación y regulación de medicamentos que afectan a toda la sociedad (27). La siguiente tabla muestra las principales características de los ECR frente a los estudios procedentes del RWD (28).

Tabla 2. *Características de los ECR y de los estudios de RWD*

Parámetro	ECR	RWE
Propósito del estudio	Eficacia	Efectividad
Tipo de estudio	Intervención	Observacional
Diseño	Prospectivo	Prospectivo/Retrospectivo
Población	Homogénea	Heterogénea
Criterios de inclusión	Estricto	Flexible
Tratamiento	Fijo	Variable
Comparador	estándar/placebo	Variable
Randomización	Sí	No
Enmascaramiento	En algunos casos	No
Seguimiento	Según protocolo	Según práctica rutinaria
Atención médica	Investigador	Varios médicos

Parámetro	ECR	RWE
Monitorización del paciente	Según protocolo	variable

Hasta el momento, cuando se ha comparado la evidencia aportada por ECR vs la aportada por análisis provenientes de RWD se ha visto que tan solo un 15% de los ECR publicados en revistas de alto impacto ha podido ser replicado utilizando RWD (29). Estas diferencias no se corrigieron mediante una adecuada selección (inclusión) de los pacientes en los RWD para obtener grupos de características similares. La randomización de los ensayos clínicos juega un papel fundamental, ya que permite que las variables confusoras (ya sean conocidas o no) se repartan de forma balanceada entre los grupos. Este balanceo es muy difícil de conseguir en el caso del RWD, sobre todo cuando las posibles variables confusoras son desconocidas (29). Probablemente necesitemos más estudios donde se compare la coincidencia entre los resultados obtenidos por los ECR frente a los conseguidos por el análisis directo de los RWD, ya que hasta ahora parece que la concordancia entre ambos tipos de estudios puede variar en función de múltiples factores, como la métrica utilizada o la similitud en las indicaciones de la medicación evaluada (30).

De hecho, en muchos casos los resultados obtenidos por estudios provenientes de RWD pueden ser difícilmente comparables con los obtenidos por los ECR, ya que los primeros evalúan fundametalmente la efectividad de un fármaco y los segundos la eficacia. De la misma forma los ECR, debido a la configuración de su diseño, tienden a tener una buena validez interna (adoleciendo de validez externa). En cambio, los estudios que provienen de RWD presentan como ventaja su validez externa pero una menor validez interna (25). Recientemente la EMA también se ha pronunciado al respecto (31), y considera que en algunos casos los ECR podrían ser reemplazados por análisis de RWD, ya que implicaría disponer de información de relevancia a un coste mucho menor.

En los próximos años será necesario desarrollar estudios híbridos que combinen la metodología de los ECR junto a los diseños observacionales obtenidos de los RWD. Aunque actualmente la RWE no puede sustituir a la evidencia obtenida de los ECR (3,24), sí que puede complementarla o dirigir mejor el objetivo de dichos ensayos. En algunos casos de poblaciones especiales (por ejemplo, el caso de los niños o gestantes), o en procesos de baja prevalencia donde no es factible realizar ECR, la RWE puede llegar a ser la única evidencia científica disponible (25,32).

3.1.3 Limitaciones del RWD.

A pesar de los esfuerzos para implementar sistemas EHR, existen algunas barreras o limitaciones importantes, como pueden ser la ausencia de interoperabilidad, los bajos niveles de usabilidad o el alto coste de mantenimiento (8). A continuación, se desarrollarán cada uno de estos apartados:

a) A la integración y combinación de información procedente de diferentes sistemas EHR, sin un esfuerzo añadido por parte del investigador, le denominamos **interoperabilidad**. Los sistemas EHR, por tanto, pueden ser no interoperables, interoperables o totalmente integrados. En el primer caso no se

permite la transferencia de datos entre un sistema EHR y el formulario electrónico de recogida de datos de un estudio (habrá de realizarse de forma manual, lo que implica más errores e ineficiencias). Esta deficiencia está directamente relacionada con una combinación de barreras técnicas y económicas en la adopción de los EHR (33). Los sistemas interoperables permiten en cambio la captura de datos procedentes de sistemas EHR de forma automática hacia los formularios electrónicos de los estudios. Finalmente, los sistemas totalmente integrados permiten usar directamente los datos de los sistemas EHR para realizar estudios, lo que se traduce en menores errores y más eficiencia (10). Durante los últimos años se han desarrollado estandarizaciones que permiten un mejor intercambio de datos entre diferentes sistemas, como pueden ser el ISO13606, HL7-CDA, OpenEHR, CIMI, OMOP o SNOMED CT (8). El conseguir una óptima interoperabilidad entre sistemas EHR supone un importante desafío debido a la complejidad de los datos que se manejan y a la diversidad de los estándares que se usan. Puesto que los EHR buscan ser generalizados a grandes grupos de población, el conseguir que los sistemas EHR sean interoperables es un paso clave para poder obtener registros de calidad (13).

b) El término **usabilidad** hace referencia al uso que el personal sanitario hace de estos sistemas. La implementación de sistemas EHR en hospitales o instituciones sanitarias ha venido acompañada en muchos casos de insatisfacción entre el personal encargado de su manejo, sobre todo cuando estos sistemas están pobremente diseñados e implican una gran complejidad en su funcionamiento (34). Algunos estudios indican que actualmente los médicos pasan más tiempo interactuando con el ordenador que evaluando a los pacientes (35,36). De hecho, se ha cuantificado en una media del 44% el tiempo mirando la pantalla frente a un 24% el tiempo de comunicación con el paciente (37). Aunque muchas veces recae sobre el profesional sanitario la responsabilidad del pobre uso de los sistemas EHR, todo indica que en general los médicos están más satisfechos con este tipo de tecnología cuando los perciben como una herramienta útil en su trabajo, más que como una mejora de la eficiencia (38). En USA los vendedores de sistemas EHR están obligados a seguir los requerimientos de certificación fijados por la *Office of the National Coordinator for Health Information Technology*, que se traduce en un esfuerzo por promover un diseño de los sistemas más centrado en el usuario final. No obstante, existen importantes variaciones en este proceso, lo que supone una dificultad añadida para los usuarios no entrenados (34). Existen métodos para evaluar la usabilidad de los sistemas EHR (34,39). Lamentablemente estos sistemas de evaluación no están implementados de forma generalizada. El implicar a los usuarios en la participación del diseño final de los sistemas EHR podría mejorar la satisfacción de éstos (38).

c) El **coste** del mantenimiento de los sistemas EHR está en relación con su capacidad de almacenaje de información, que se incrementa considerablemente con el tiempo. Para optimizar al máximo la capacidad de almacenaje se hace uso

de sistemas de gestión de bases de datos (DBMSs). Durante décadas las bases de datos relacionales han dominado estos DBMSs (40). Sin embargo, en los últimos años se han desarrollado sistemas de almacén de datos basados en bases de datos NoSQL que incluyen datos en otros formatos además de los tabulares típicos de las bases de datos relacionales (8). Se ha comprobado que el modelo tradicional de bases de datos relacionales no es lo suficientemente flexible ni eficiente para manejar bigdata. Incluso aunque muchos datos médicos tienen características que permiten un adecuado manejo con bases de datos relacionales, existe una amplia variedad de datos que no se ajustan bien a este tipo de bases de datos (datos no estructurados, complejos, irregulares o dinámicos) (8).

3.1.4 Calidad de los sistemas EHR.

Uno de los principales desafíos sobre este tipo de datos es si se satisfacen los estándares de calidad que permitan obtener conclusiones relevantes de sus estudios. Los EHR son recogidos por los proveedores de servicios de salud, y por tanto no tienen los mismos estándares de calidad que los recogidos en un estudio de investigación. Así pues, el primer paso antes de preguntarnos si determinado EHR pueden contestar adecuadamente a una pregunta de salud es evaluar la calidad de los mismos (9). Para ello se evalúan tres características: conformación, completabilidad y plausibilidad, que describimos a continuación:

a) **Conformación**. Evaluamos si los datos se corresponden o son conformes a las especificaciones y estructura de la base de datos (por ejemplo, introducir la edad como texto en lugar de como número entero).

b) **Completabilidad**. Supone el estudiar la ausencia de datos, y si éstos son los esperables para nuestra base de datos. Aquí habría que diferenciar entre datos que "*deben estar*" ("*must-have*") en nuestro registro frente a datos que "*nos gustaría tener*" ("*nice-to-have*") (13). Evidentemente la ausencia de datos "*must-have*" tendrá un mayor impacto en la calidad de los mismos.

c) **Plausibilidad**. Determina la credibilidad de los datos. A su vez puede ser *atemporal* (por ejemplo, introducir como peso 680 kg), o *temporal* (por ejemplo introducir como tensión arterial siempre el mismo valor en varias tomas sucesivas).

Debido a que los EHR están formados por bigdata, la revisión manual de los datos es técnicamente imposible, por tanto, es necesario el uso de algoritmos para su evaluación. Hasta el momento la evaluación de la calidad de los datos no se ha estandarizado, si bien algunos estudios han examinado de forma aislada alguno de los componentes de calidad revisados previamente (9).

La calidad de los datos suele ser, además, muy variable dependiendo del tipo de datos que consideremos. Por tanto, la calidad de los EHR condicionará el tipo de estudio que podamos hacer con ellos puesto que algunos estudios (por ejemplo epidemiológicos, o

de determinación de factores de riesgo) pueden no ser adecuados para nuestro set de datos.

Probablemente el apartado sobre la calidad sea uno de los puntos débiles más importantes de los EHR. La calidad de los EHR varía mucho entre diferentes instituciones u organizaciones. Incluso dentro de una misma institución el hecho de actualizar o "mejorar" el sistema EHR mediante la inclusión de nuevas variables puede tener el efecto paradójico de empeorar la calidad global de los mismos (13).

3.1.5 Ejemplos de EHR.

Como se ha comentado anteriormente, en los últimos años se ha extendido el uso de sistemas EHR, con la consiguiente aparición de estudios derivados del análisis de dichos datos. La RWE derivada del RWD cubre prácticamente todas las especialidades médicas con una amplia variedad de casos en neurología (41), pediatría (42), infeccioso (1,43), oftalmología (34), medicina interna (44), o anestesiología (45,46), entre otras.

En la mayoría de los casos, los sistemas EHR engloban a varias instituciones o centros sanitarios, lo que permite acceder a una gran cantidad de datos. Sin embargo, destacan algunos sistemas EHR globales en los que está incluida toda la red sanitaria del país. Algunos de estos registros globales derivan de datos recogidos desde hace décadas a nivel nacional por las respectivas agencias de salud (13). Un ejemplo a considerar son los países nórdicos, donde una amplia red de bases de datos de salud sobre la población permite su uso posterior para la realización de investigaciones de tipo observacional o como apoyo para la evidencia obtenida de los ECR (47).

3.1.6 Conclusión.

A lo largo del trabajo se ha revisado el estado de la cuestión de los EHR. Hemos visto como a partir del año 2016, y gracias a la *twenty-first Century Cures Act* (23), los estudios derivados de dichos registros aumentaron de forma exponencial. Actualmente estamos inmersos en el debate acerca del nivel de evidencia que aportan las investigaciones derivadas del RWD, y si dicho nivel de evidencia puede ser suficiente para aplicar políticas regulatorias sobre nuevos fármacos por las agencias nacionales de salud. Aunque este debate nació por iniciativa de las empresas farmacéuticas en un intento de abaratar en tiempo y en coste la actualmente burocratizada vía de los ECR, no creo que por ello debamos minusvalorar el papel que el RWD y la RWE jugarán en el futuro. Probablemente la RWE tendrá un papel cada vez más importante y permitirá aplicar políticas de regulación y aprobación de nuevos medicamentos, así como de ampliación en las indicaciones de los ya existentes de una forma más flexible. Este cambio de paradigma en el concepto de evidencia científica ha de estar asociado a un control exhaustivo en los estándares de calidad de los RWD, algo que actualmente aún estamos lejos de conseguir.

Este trabajo se ha centrado en la obtención de evidencia a partir de EHR de pacientes ingresados en una unidad de cuidados intensivos (UCI) con el diagnóstico de IRA. Hasta la fecha no se ha publicado ningún trabajo que realice este tipo de análisis en pacientes con IRA que no cumplen criterios de síndrome de distrés respiratorio agudo (SDRA). A

lo largo del trabajo comprobaremos el potencial que los EHR presentan de cara a la investigación clínica.

3.2 Conjunto de datos MIMIC III.

La base de datos MIMIC III, en su versión 1.4, es un dataset de acceso público que recoge EHR de más de 40000 pacientes desidentificados que ingresaron en el Beth Israel Deaconess Medical Center entre 2001 y 2012 (5,48). Este conjunto de datos incluye una gran variedad de información, desde datos generales como edad, raza o sexo, hasta datos de laboratorio, además de diagnósticos clínicos y procedimientos realizados. La accesibilidad de esta base de datos a dado lugar a numerosas publicaciones en diferentes especialidades médicas (49–53). Concretamente, desde el año 2014, se han publicado 147 trabajos registrados en pubmed en los que se ha utilizado MIMIC III como fuente de datos. Los estudios publicados se centran fundamentalmente en sepsis (28%), insuficiencia renal aguda (17%) y SDRA (4%). No obstante, a pesar de la proliferación de publicaciones durante los últimos años, hasta muy recientemente no se había evaluado la calidad de los datos recogidos en el dataset MIMIC III (54). En el trabajo publicado por Afshar et al se comprobó que la calidad de los datos estudiados (concretamente determinan la frecuencia cardiaca, frecuencia respiratoria, spO2 y presión arterial) disminuye según avanza la estancia de cada pacientes en la UCI. Un problema que, como ya se ha comentado anteriormente, comparten la gran mayoría de los registros electrónicos de salud.

MIMIC III es una base de datos relacional formada por 26 tablas. Las tablas están enlazadas mediante identificadores que utilizan el sufijo *ID*. Por ejemplo, *subject_id* hace referencia al número de identificación de cada paciente. La variable *hadm_id* se refiere al número de identificación del ingreso hospitalario y la variable *icustay_id* nos indica la identificación del ingreso en cuidados intensivos (UCI). De esta forma un mismo paciente (subject_id) puede tener varios ingresos hospitalarios (hadm_id), y a su vez también puede tener diferentes ingresos en UCI (icustay_id). Entre las 26 tablas encontramos 5 de ellas que funcionan como diccionarios, donde se especifica a qué procedimiento, diagnóstico o exploración corresponde un determinado código. Estas tablas son: D_CPT, D_ICD_DIAGNOSES, D_ICD_PROCEDURES, D_ITEMS y D_LABITEMS. El conjunto de todas las tablas del dataset se muestra en la siguiente tabla:

Tabla 3. *Descripción de las tablas del dataset MIMIC III*

Tabla	Descripción
ADMISSIONS	Hospitalizaciones únicas de cada paciente definidas por HADM_ID
CALLOUT	Información acerca de ingresos y altas en UCI
ICUSTAYS	Ingresos únicos en UCI de cada paciente definidos por ICUSTAY_ID
PATIENTS	Información sobre cada paciente definido por SUBJECT_ID
SERVICES	Servicio médico a cargo del paciente

Tabla	Descripción
TRANSFERS	Traslados de pacientes dentro del hospital
CAREGIVERS	Identificación del sanitario que atendió al paciente
CHARTEVENTS	Registros de pruebas acerca de los pacientes
DATETIMESEVENTS	Fechas de procedimientos realizados
INPUTEVENTS_CV	Registros de monitor en UCI usando el sistema Phillips CareVue
INPUTEVENTS_MV	Registros de monitor en UCI usando el sistema Metavision
NOTEEVENTS	Notas de médicos y enfermeros de muy diverso tipo
OUTPUTEVENTS	Información de resultados mientras el paciente está en UCI
PROCEDUREEVENTS_MV	Información sobre procedimientos realizados en UCI recogidos con el sistema MetaVision
CPTEVENTS	Procedimientos realizados, recogidos mediante el código CPT
DIAGNOSES_ICD	Diagnósticos asignados mediante el código ICD
DRGCODES	Diagnósticos realizados según código DRG
LABEVENTS	Mediciones de laboratorio
MICROBIOLOGYEVENTS	Resultados microbiológicos
PRESCRIPTIONS	Prescripción de medicamentos
PROCEDURES_ICD	Procedimientos realizados codificados según el ICD
D_CPT	Diccionario de los códigos Current Procedural Terminology (CPT)
D_ICD_DIAGNOSES	Diccionario de los códigos de la clasificación internacional de enfermedades (ICD 9)
D_ICD_PROCEDURES	Diccionario de los códigos de los procedimientos según la ICD
D_ITEMS	Diccionario de ITEMID que aparecen en la base de datos
D_LABITEMS	Diccionario de ITEMID relacionados con medidas de laboratorio

Además de las 26 tablas comentadas anteriormente, MIMIC III ofrece unas tablas derivadas que ofrecen resúmenes, agrupan eventos o calculan scores que según el tipo de estudio pueden ser muy interesantes de evaluar.

Aunque MIMIC III es una base de datos pública, para poder acceder a la versión completa es necesario tener una cuenta acreditada en Physionet. La acreditación se consigue mediante la realización del curso *Human Research. Data or Specimens Only Research* impartido por el *Massachussets Institute of Technology Affiliates*. Una vez se

obtiene la acreditación ya se puede acceder a las tablas de la versión completa del MIMIC III.

El acceso a los datos, desde el año 2019, puede realizarse online a través de Google Cloud Platform o de Amazon Web Services (AWS). No obstante, el acceso mediante la descarga local de todas las tablas sigue estando disponible. En la web del proyecto MIMIC se encuentra información detallada (55) para acceder a las tablas bien en local u online.

El conjunto de datos MIMIC III constituye un claro ejemplo de cómo a partir de EHR se puede generar conocimiento en forma de RWE. Desde el año 2014 siguen apareciendo nuevos trabajos que utilizan los datos de dicha plataforma. De hecho, este conjunto de datos se ha hecho muy popular a la hora de elaborar modelos de ML. Hasta 17 trabajos publicados en Pubmed utilizando este conjunto de datos (11.5%) han elaborado predicciones aplicando diferentes algoritmos de ML.

3.3 Insuficiencia Respiratoria Aguda.

Podemos definir la IRA como aquella situación en la que se produce una alteración en el intercambio gaseoso a nivel pulmonar, lo que implica una disminución del oxígeno arterial (hipoxemia) con/sin aumento de la concentración de CO2 en sangre. Por tanto, en la gasometría arterial nos encontraremos una PaO2 <60 mm Hg y/o PaCo2 > 50 mm Hg (para una fracción inspirada de oxígeno del 21%). La hipoxemia puede desencadenar una hipoxia que afectará de forma progresiva a los territorios más sensibles y que afectará al normal funcionamiento de órganos y sistemas (56). Clásicamente se ha considerado como IRA tipo 1 la situación clínica de hipoxemia sin hipercapnia, y como IRA tipo 2 cuando la hipoxemia se acompaña de hipercapnia (57). En las tablas 4 y 5 se resumen las principales causas de ambos tipos de IRA.

Tabla 4. *Insuficiencia respiratoria aguda tipo I*

Causas	Ejemplos
Alteración de la ventilación/perfusión	Neumonía, SDRA, EAP
Alteración de la difusión	Enfermedades del intersticio pulmonar
Aumento del shunt fisiológico	Shunt derecha-izquierda, sdme hepatopulmonar
Disminución de la FiO2	Altitud
Hipoventilación	EPOC, Obstrucción de la VAS

Tabla 5. *Insuficiencia respiratoria aguda tipo II*

Causas	Ejemplos
Disminución del estímulo respiratorio	Sobredosis de opiáceos, efectos de la AG, hipotiroidismo
Debilidad muscular	Miastenia gravis, sdme de Guillain-Barré

Causas	Ejemplos
Distorsión de la caja torácica	Obesidad mórbida, traumatismo torácico
Aumento del espacio muerto	TEP, SDRA, bajo gasto cardiaco
Aumento de la producción de CO2	Hipertermia maligna, sepsis

La IRA supone uno de los principales motivos de ingreso en las unidades de cuidados intensivos. En el dataset MIMIC III hasta un 20% de los ingresos fueron por este motivo (incluyendo también el SDRA).

La sintomatología que presentan estos pacientes dependerá en gran medida de la causa subyacente, aunque en general predominará la dificultad respiratoria (disnea) que obligará a la utilización de músculos respiratorios accesorios. En los cuadros de IRA tipo II podemos encontrarnos también con cuadros de confusión y obnubilación. La auscultación cardio respiratoria es fundamental, ya que nos puede orientar hacia algunos tipos de causas: asma o EPOC, neumonía, derrame pleural, etc. Las pruebas complementarias de diagnóstico inicial pasan por la realización de una gasometría arterial y una radiografía de tórax.

El manejo clínico y tratamiento de estos pacientes se basa en tratar la causa de la IRA. Además, se aplicará desde oxigenoterapia inicialmente hasta soporte ventilatorio en los casos más graves con el fin de mejorar el intercambio de gases a nivel alveolar (58).

El soporte ventilatorio de los pacientes con IRA es un tema amplio y complejo que excede los objetivos de esta revisión. Simplemente comentaré que disponemos de dos estrategias de ventilación: ventilación invasiva (VI) y ventilación no invasiva (VNI) (59). Ambas tienen en común el aplicar presión positiva en la vía aérea de los pacientes con el objetivo de mejorar la ventilación de éstos y facilitar el trabajo respiratorio. La VI requiere intubación orotraqueal, además de sedación en mayor o menor medida, y por tanto es más agresiva que la VNI.

En los pacientes del dataset MIMIC III diagnosticados de IRA hasta un 20% requirieron soporte ventilatorio, la inmensa mayoría mediante VI. Si bien la insuficiencia respiratoria en cuidados críticos es un tema que ya se ha estudiado utilizando pacientes del conjunto de datos MIMIC III, los trabajos se han centrado en el caso particular del SDRA (49,50,60). El interés de este trabajo radica en que recoge aquellos pacientes con IRA que no cumplen criterios de SDRA, un tema que hasta la fecha no ha sido estudiado.

4. METODOLOGÍA.

4.1 Acceso a los datos.

Como se ha comentado con anterioridad para acceder al conjunto de datos MIMIC III v 1.4 ha de realizarse un curso que acredite para manejar este tipo de datos. La base de datos se halla ubicada en el servidor de Physionet (61), y en su web te ofrecen toda la información para poder acceder a ella. El certificado del curso *Human Research. Data or Specimens Only Research* fue obtenido el 12 de febrero de 2022 (Anexo I). Para la obtención del certificado es necesario aprobar los 9 módulos de los que consta con una puntuación media igual o superior al 90%. Finalmente, el acceso a los datos fue concedido con fecha de 15 de marzo de 2022.

El acceso a las tablas del conjunto de datos se ha realizado online a través de Google Cloud Platform (62). Para ello hay que registrarse gratuitamente a través de la cuenta de Google, lo que permite una utilización de la demo durante 90 días sin coste alguno. En la web de Physionet hay que configurar la cuenta para que enlace con Google Cloud. Una vez configurado el enlace es automático, de forma que desde la plataforma Google Cloud y mediante BigQuery se accede a las diferentes tablas fácilmente sin necesidad de descargarlas. BigQuery funciona mediante SQL standard, por lo que el procesamiento de las tablas, aunque laborioso, es relativamente sencillo. Con cada query efectuada se nos informa de la cantidad de información que se procesa en la consulta, y es en base a ese parámetro como se realiza la facturación. Como he comentado antes, al estar en el periodo de demo de 90 días no se produce cargo alguno, independientemente del tamaño de la consulta. El Anexo II recoge el código SQL utilizado para la creación de la tabla definitiva.

Para este trabajo se han utilizado las siguientes tablas, que se han procesado de manera jerárquica hasta llegar a la cohorte final: *ADMISSIONS, PATIENTS, ICUSTAYS, CHARTEVENTS, LABEVENTS, DIAGNOSES_ICD, PROCEDURES_ICD y PRESCRIPTIONS*. Además, se han usado para identificar los diferentes códigos las tablas-diccionario siguientes: *D_ICD_DIAGNOSES, D_ICD_PROCEDURES, D_ITEMS y D_LABITEMS*. En relación con las tablas derivadas, se han utilizado *weight_first_day*, *height_first_day*, *vitals_first_day* y *sapsii* De esta forma la tabla final está conformada por los siguientes tipos de datos:

a) Datos generales: edad, sexo, raza, religión, días de ingreso en UCI, exitus (en caso de producirse).

b) Mediciones físicas: peso, talla, escala de coma de Glasgow (GCS), TAS, TAD, TAM, FC, temperatura, diuresis.

c) Gasometría arterial: FiO2, sO2, pH, ABE, HCO3, PaO2, PaCO2, lactato.

d) Hemograma: Hb, Hto, leucocitos, plaquetas.

e) Bioquímica: glucosa, creatinina, BUN, albúmina, pro-BNP.

f) Respuesta inflamatoria: D-dímero, proteína C reactiva.

g) Fármacos: dopamina, dobutamina, adrenalina, noradrenalina, heparina, dexametasona, prednisolona.

h) Comorbilidad: medición de la comorbilidad de Elixhauser (63) según el algoritmo publicado por Quan (64).

i) Ventilación mecánica (en caso de aplicarse).

j) Puntuación de los siguientes scores de severidad: OASIS y SAPS II.

De todos los ítems medidos se ha calculado la media en las primeras 24 horas del ingreso en UCI. En el caso de los fármacos y de la ventilación mecánica se ha determinado si se administraron en las primeras 24 horas del ingreso en UCI, así como su duración. La determinación de comorbilidad de Elixhauser recoge 30 ítems diferentes y cada uno de los ítems aparece codificado como 0/1 según esté ausente o presente. Los scores OASIS y SAPS II son índices de predicción de la mortalidad en unidades de cuidados intensivos, y como tal se recogen.

4.2 Selección de la cohorte de estudio.

A continuación, se detallan los criterios de inclusión en el estudio:

a) Pacientes adultos entre 18 y 100 años.

b) Diagnóstico en base al International Classification of Disease (ICD) en su versión 9 (65) correspondiente a los siguientes códigos:

- 518.81: acute respiratory failure.
- 518.84: acute and chronic respiratory failure.
- 518.51: acute respiratory failure following trauma and surgery.
- 518.53: acute and chronic respiratory failure following trauma and surgery.

c) Datos completos para todos los pacientes en relación a los siguientes ítems:

- Datos generales: edad, sexo, exitus (en caso de producirse)
- Mediciones físicas: TAS, TAD, pulso, temperatura, diuresis y GCS.
- Gasometría arterial.
- Hemograma.
- Bioquímica: glucosa, creatinina.

Así pues, la cohorte con diagnóstico de IRA que se ha seleccionado no incluiría aquellos pacientes con SDRA (ICD9: 518.82) ni con diagnóstico de parada respiratoria (ICD9: 799.1). La figura 2 muestra el esquema de selección de la cohorte final.

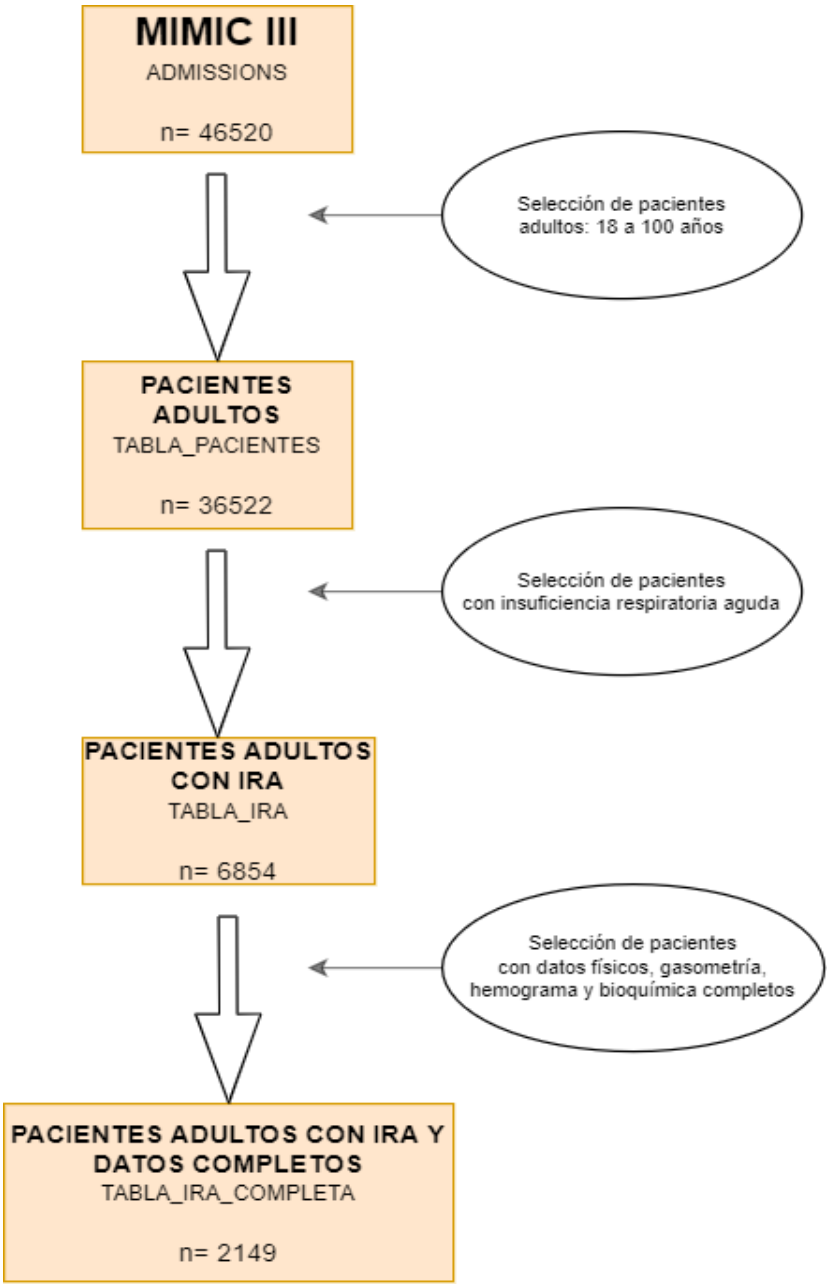


Figura 2. Recorrido del proceso de selección de la cohorte del estudio.

Se ha comentado en el apartado *3.2* la relación existente entre un paciente en particular (*subject_id*), sus ingresos hospitalarios (*hadm_id*), y sus ingresos en UCI (*icustay_id*). La figura 3 resume este tipo de relación 1:m: n. Para este estudio se ha decidido incluir los pacientes en base al *subject_id*, de forma que en aquellos pacientes que presenten varios ingresos en UCI se ha seleccionado únicamente el primero de ellos.

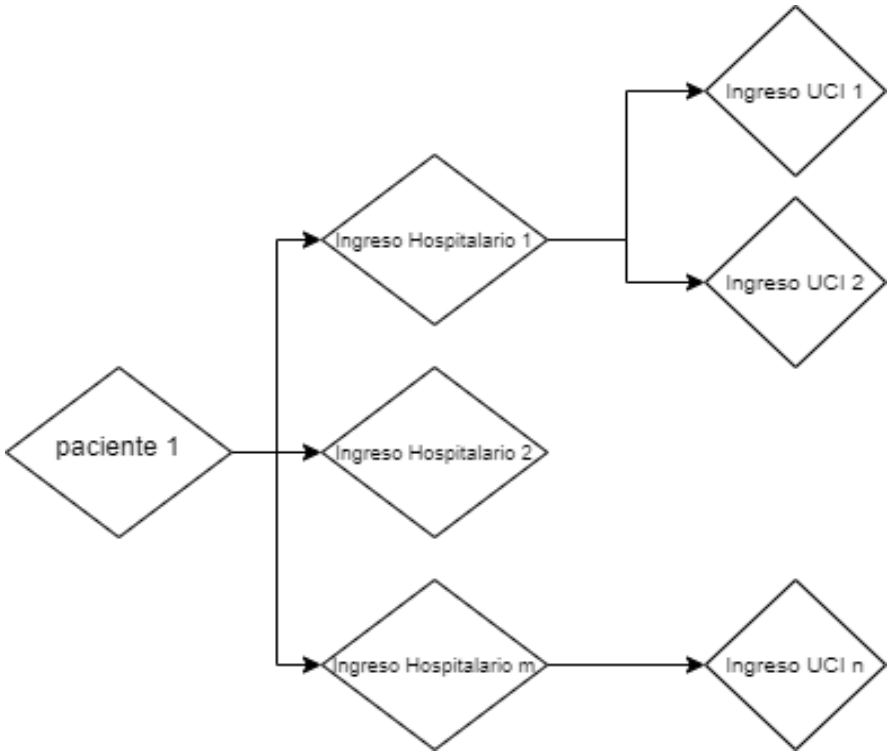


Figura 3. Relación entre pacientes, ingresos hospitalarios e ingresos en UCI.

4.3 Análisis de los datos.

La gestión y procesamiento de las tablas de MIMIC III se ha realizado online mediante BigQuery. El resultado final es una única tabla con 2149 registros que corresponde a la cohorte de pacientes ingresados en UCI con IRA que cumplen los criterios de inclusión.

El análisis estadístico, así como el desarrollo de los diferentes modelos de ML se han realizado en RStudio (versión 2021.09.1 para Windows) en lenguaje R. La implementación de las redes neuronales (secuencial y convolucional) se ha realizado también en RStudio, pero en lenguaje Python mediante la API keras de TensorFlow. El código utilizado en ambos lenguajes está disponible en el Anexo III.

Para comparar datos categóricos se ha utilizado el test Chi^2. Para comparar variables cuantitativas se ha optado por el test de Welch en caso de varianzas diferentes entre los dos grupos, y el test t de Student en caso contrario. La correlación entre datos cuantitativos se ha realizado mediante el test de Pearson.

Para elaborar el modelo de predicción se ha dividido el dataset en un grupo test (30%) y un grupo train (70%), que se ha mantenido idéntico en cada uno de los modelos estudiados. En el desarrollo de modelos de ML se han analizado los siguientes algoritmos: KNN, RL, NB, SVM lineal, SVM radial y RF. Finalmente se ha utilizado aquel que mejor rendimiento ofrecía. Igualmente, dentro de los modelos de deep learning, se ha evaluado una RNP secuencial y una RNP convolucional. Aquella red neuronal que

presentaba un mejor rendimiento se ha comparado con el modelo de ML seleccionado previamente para quedarnos únicamente con un algoritmo de predicción de mortalidad.

El rendimiento de los diferentes modelos de predicción se ha evaluado mediante el cálculo del área bajo la curva (AUC) de la curva ROC correspondiente, además del cálculo de la sensibilidad (S), especificidad (E), valor predictivo positivo y negativo (VPP y VPN).

El TFM se ha realizado en su totalidad mediante RMarkdown, lo que ha permitido realizar informes dinámicos y modificar automáticamente los resultados del trabajo en función de los datos utilizados.

5. RESULTADOS.

5.1 Valoración datos perdidos.

Dado que la presencia de datos perdidos (NA) es frecuente en el conjunto de datos MIMIC III (54) en primer lugar exploramos cómo se distribuyen los NA en la tabla definitiva (*TABLA_IRA_COMPLETA*). La tabla 6 nos muestra la calidad de los datos en los pacientes con IRA, basándonos en la presencia de NA, agregados en base a la variable *exitus* (mortalidad a los 30 días). El valor p obtenido (test X^2) nos muestra aquellas variables con diferencias significativas en la presencia de NA según el grupo *exitus*.

Tabla 6. *Valores nulos de los datos en cada variable.*

Variables	Exitus: n(%)	No exitus: n(%)	p
religion	6 (0.35)	3 (0.68)	0.58097
gender	0 (0)	0 (0)	NaN
ethnicity	0 (0)	0 (0)	NaN
peso	174 (10.17)	51 (11.64)	0.41690
talla	575 (33.61)	180 (41.1)	0.00406
gcs	0 (0)	0 (0)	NaN
tas	0 (0)	0 (0)	NaN
tad	0 (0)	0 (0)	NaN
tam	0 (0)	0 (0)	NaN
fc	0 (0)	0 (0)	NaN
temp	0 (0)	0 (0)	NaN
diuresis	0 (0)	0 (0)	NaN
fio2	0 (0)	0 (0)	NaN
so2	0 (0)	0 (0)	NaN
ph	0 (0)	0 (0)	NaN
abe	0 (0)	0 (0)	NaN
hco3	0 (0)	0 (0)	NaN
po2	0 (0)	0 (0)	NaN
pco2	0 (0)	0 (0)	NaN
hb	0 (0)	0 (0)	NaN
hto	0 (0)	0 (0)	NaN
leucos	0 (0)	0 (0)	NaN
plaquetas	0 (0)	0 (0)	NaN
glucosa	0 (0)	0 (0)	NaN
creat	0 (0)	0 (0)	NaN
bun	110 (6.43)	44 (10.05)	0.01191

Variables	Exitus: n(%)	No exitus: n(%)	p
albu	634 (37.05)	124 (28.31)	0.00078
bnp	1370 (80.07)	360 (82.19)	0.35109
ddimer	1609 (94.04)	407 (92.92)	0.45088
pcr	1626 (95.03)	413 (94.29)	0.61322

Destacan algunas variables como *ddimer* o *pcr* donde la presencia de NA es superior al 90%. La IRA basa su diagnóstico en la gasometría arterial, por tanto, se ha exigido que las variables que la conforman estén presentes en la totalidad de los pacientes. A este criterio exigente hemos añadido también parámetros muy básicos de hemograma y biquímica que deben estar en toda exploración complementaria. Por todo ello las variables que conforman estos grupos no tienen datos ausentes.

Esta tabla nos ofrece la información necesaria para saber sobre qué variables hemos de realizar imputaciones. En este estudio vamos a considerar un umbral del 20% de presencia de NA para imputar. En base a ello, y según nos indica la tabla 6, imputaremos las siguientes columnas: *peso* (10% de NA) y *bun* (7% de NA). La *talla*, *albúmina*, *bnp*, *ddimer* y *pcr* presentan una prevalencia de NA demasiado elevada y el imputar dichas variables supondría asumir un elevado margen de error.

5.2 Imputación de variables.

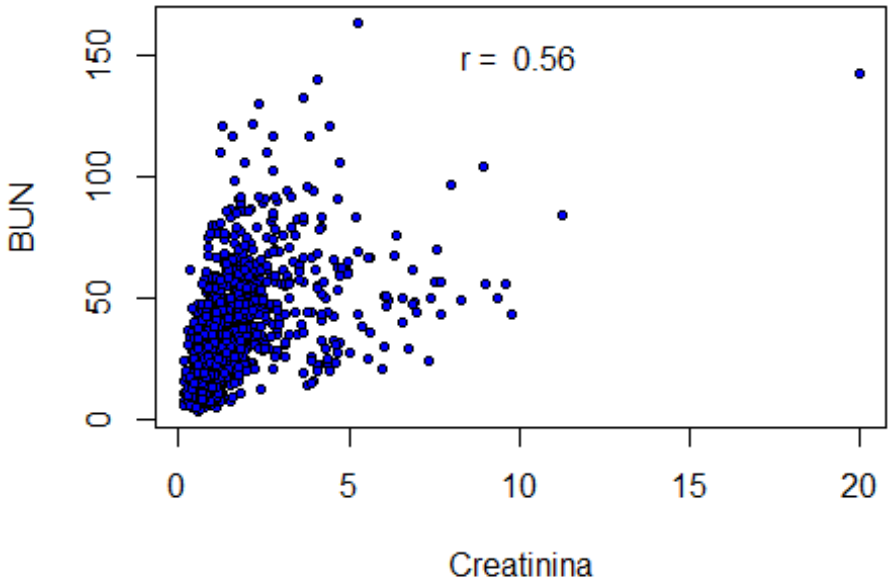
5.2.1 Imputación del peso.

Para la imputación del **peso** se ha analizado, mediante regresión lineal, aquellas variables que permitirían predecir su valor. Sin embargo, la R^2 obtenida es tan solo de `0.18`. Uno de los problemas que nos encontramos con esta variable es que la *talla*, que es una variable muy relacionada con el peso, está ausente en un 35% de las observaciones. Por tanto, la imputación se realiza finalmente sustituyendo el valor faltante por el de la mediana (para la variable *peso* la mediana es `79.8` Kg).

5.2.2 Imputación del BUN.

Por último, vamos a realizar la imputación de la variable **bun**. Esta columna, que recoge el valor de los productos nitrogenados fruto del metabolismo proteico, está relacionada con la función renal (al igual que la creatinina). En la siguiente figura vemos cómo se correlacionan ambas variables.

Figura 4. Relación creatinina-BUN



Además de la creatinina, existen otras variables que también se correlacionan con el valor de *bun*, tal y como se muestra en la siguiente tabla:

Tabla 7. *Coeficiente de correlación respecto a la variable BUN*

	r (Pearson)
edad	0.24
tam	-0.13
diuresis	-0.13
glucosa	0.16
creat	0.56

Así pues, para la imputación de *bun* se utilizará un modelo de regresión lineal que prediga el valor de los productos nitrogenados a partir de las variables antes comentadas.

El modelo de regresión lineal nos ofrece un R^2 de `0.38`, que para este caso en particular lo consideramos aceptable para hacer la imputación correspondiente.

5.3 Exploración del dataset.

El dataset *tabla_ira_completa* está consituido por `2149` pacientes, de los cuales un `44` % son mujeres y un `56` % son varones. La mortalidad global a los 30 días en el conjunto de datos fue del `21` %. En los siguientes epígrafes repasaremos las diferentes variables que conforman este dataset.

5.3.1 Datos generales.

Estas variables proceden tanto de datos administrativos como de la exploración física realizada a los pacientes. Incluyen el sexo, la edad, el peso, el GCS, la tensión arterial, la temperatura, la frecuencia cardiaca y la diuresis en las primeras 24 horas. La diuresis la transformamos para expresarla en ml/kg/hora.

La figura 5 nos muestra la distribución del sexo y su mortalidad a los 30 días. Se observa que no existen diferencias significativas entre hombre y mujeres respecto a la mortalidad.

Figura 5.Sexo de los pacientes y mortalidad asociada.

La tabla 8 nos muestra la media del resto de variables en relación con la mortalidad a 30 días.

Tabla 8. *Datos generales y su relación con la mortalidad.*

Exitus	peso	edad	tam	gcs	temp	diuresis
0	84.749	63.789	78.697	13.330	36.893	1.039
1	78.606	69.022	77.233	13.080	37.060	0.885
p valor	0.00000	0.00000	0.00000	0.16100	0.39700	0.0000000

Apreciamos cómo el grupo de *Exitus* presenta una mayor edad, menor TAM y menor puntuación de GCS, lo que parece lógico ya que estas variables están relacionadas directa o indirectamente con la mortalidad.

5.3.2 Gasometría.

Las variables incluidas en la gasometría arterial son: FiO2, pH, PaO2, PaCO2, HCO3, ABE, SaO2 y lactato. En la tabla 9 resumimos los hallazgos encontrados.

Tabla 9. *Gasometría arterial (media) y su relación con la mortalidad*

Exitus	FiO2	SaO2	pH	PaO2	PaCO2	HCO3	ABE	Lactato
0	58.87	97.04	7.40	139.090	46.290	27.730	4.280	1.92
1	64.38	96.41	7.39	136.520	45.000	26.940	4.310	2.48
p valor	0.00	0.00	0.30	0.469	0.061	0.004	0.865	0.00

Observamos en la tabla que el grupo de exitus requiere una mayor FiO2 y presenta ligeramente peores parámetros de intercambio gaseoso respecto al grupo que sobrevive. Además, se aprecia cómo el lactato, un parámetro que aumenta en la hipoxia tisular, se encuentra incrementado significativamente en el grupo de los pacientes que fallecen.

5.3.3 Hemograma.

En el hemograma incluímos el nivel de hemoglobina (hb), el hematocrito (hto), los leucocitos totales y el recuento de plaquetas. La tabla 10 resume los hallazgos encontrados.

Tabla 10. *Hemograma (media) y su relación con la mortalidad*

Exitus	Hb	Hto	Leucocitos	Plaquetas
0	10.870	32.720	12.370	244.370
1	10.590	31.790	13.800	228.660
p valor	0.005	0.001	0.003	0.018

Destaca cómo los pacientes que fallecen tienen una Hb significativamente inferior, y unos leucocitos significativamente superiores (lo que puede relacionarse con la asociación de sepsis).

5.3.4 Bioquímica y parámetros inflamatorios.

La bioquímica básica está conformada por la glucosa, parámetros de función renal (BUN y creatinina), la albúmina y la proteína pro_BNP. Como parámetros de inflamación hemos considerado el D-Dímero y la proteína C reactiva (PCR). La tabla 11 recoge el resumen de los resultados.

Tabla 11. *Bioquímica e inflamación (media) y su relación con la mortalidad*

Exitus	Glucosa	BUN	Creatinina	Albúmina	D-Dímero	PCR	BNP
0	138.370	27.08	1.790	2.54	4890.640	116.020	6319.860
1	169.520	32.34	1.540	2.41	2683.500	112.440	7805.410
p valor	0.133	0.00	0.939	0.00	0.219	0.512	0.272

En este apartado destaca que el D-Dímero se encuentra incrementado en el grupo que sobrevive. Tanto la pro-bnp como la PCR se encuentran aumentados en el grupo que fallece. No obstante las diferencias no son significativas. En este punto hay que destacar que dichos parámetros están presentes tan solo en un pequeño porcentaje de los pacientes totales, concretamente en `133` y en `110` pacientes para el D-Dímero y la PCR, respectivamente. Otro aspecto a destacar es que la función renal parece estar más conservada en el grupo de *exitus*. Sin embargo este dato puede estar sesgado debido al aumento de requerimientos de terapias de sustitución renal que presentan los pacientes más graves, y que pueden forzar unos parámetros de función renal cerca de la normalidad. Con la proteína pro-BNP ocurre algo similar, ya que sólo se encuentra disponible en `419` pacientes.

5.3.5 Medicación.

A continuación, exploraremos la medicación administrada en las primeras 24 horas del ingreso en UCI a los pacientes con IRA. En este caso nos hemos centrado en las denominadas drogas vasoactivas (dopamina, adrenalina, noradrenalina y dobutamina), pero también se ha recogido en este trabajo la administración de heparina y el uso de corticoides (dexametasona y metil-prednisolona).

La tabla 12 resume la relación de las drogas vasoactivas administradas con la mortalidad de los pacientes con IRA.

Tabla 12. *Relación drogas vasoactivas y mortalidad*

Exitus	Dopa: n(%)	Dobuta:n(%)	Nora:n(%)	Adrena:n(%)
0	54 (3.2)	9 (0.5)	200 (11.7)	12 (0.7)
1	23 (5.2)	6 (1.3)	89 (20)	2 (0.4)
p valor	0.06	0.126	0	0.747

En la tabla destaca que todos los pacientes que han fallecido han necesitado en mayor porcentaje drogas vasoactivas (excepto para la adrenalina). Las diferencias unicamente son significativas para la dobutamina y noradrenalina, aunque a excepción de la noradrenalina, la muestra del resto de drogas es escasa.

Respecto al uso de heparina y corticoides en las primeras 24 horas la tabla 13 resume los hallazgos encontrados.

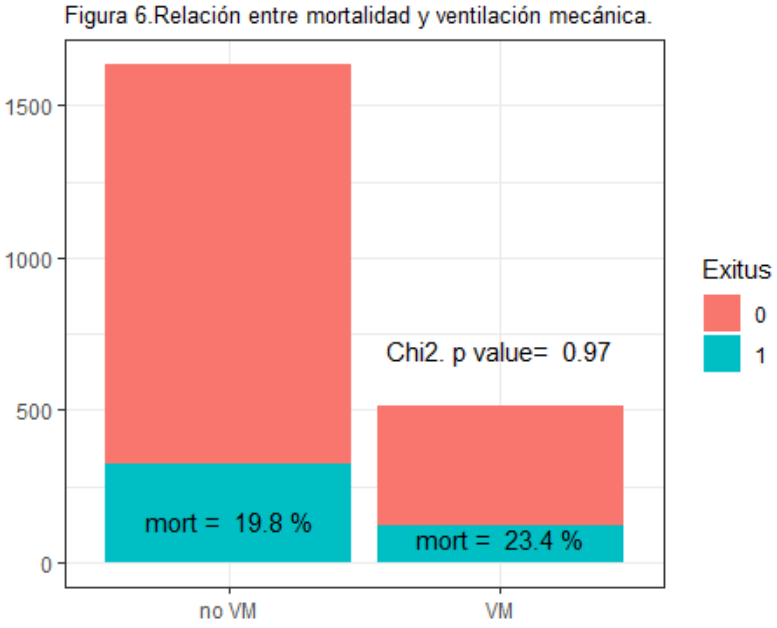
Tabla 13. *Heparina y corticoides, relación con la mortalidad*

Exitus	Heparina: n(%)	Dexametasona:n(%)	Metilprednisolona:n(%)
0	734 (43.1)	55 (3.2)	158 (9.3)
1	163 (36.6)	15 (3.4)	49 (11)
p valor	0.016	0.999	0.309

En este resultado sorprende que hasta un 58.3% de los pacientes ingresados en la UCI con IRA no han recibido heparina (ni a dosis terapeútica ni profiláctica) en las primeras 24 horas, tratándose además de una población de riesgo y con comorbilidad asociada. Una posibilidad a tener en cuenta es que falten datos completos sobre la utilización de heparina en estos pacientes en el conjunto de datos MIMIC III. Otra opción es que la heparina no se administrase de forma precoz y se demorase su administración hasta después de las primeras 24 horas de ingresar en UCI.

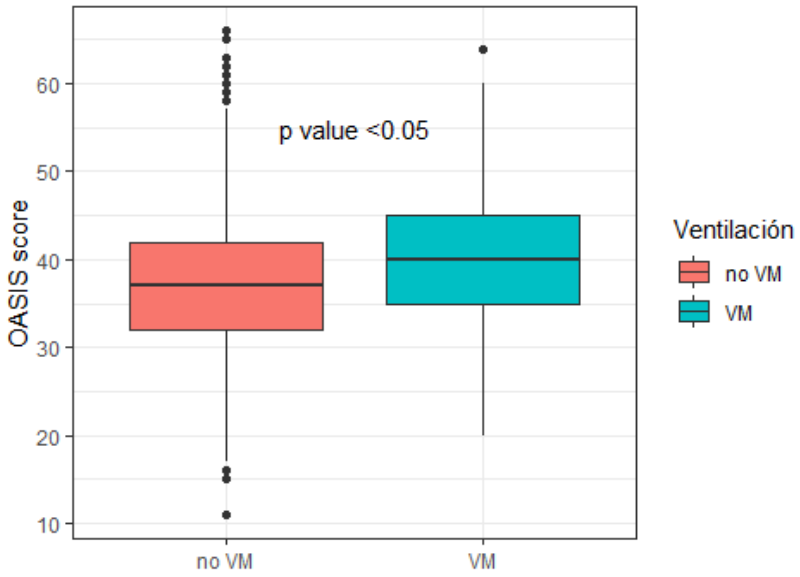
5.3.6 Ventilación mecánica.

En la tabla que estamos estudiando un 24% de pacientes han necesitado VM. Entre los pacientes ventilados, en un 97.1% se trataba de VMI. La figura 6 muestra un gráfico de barras relacionando la mortalidad a 30 días con la administración de VM.



Se aprecia cómo la mortalidad es ligeramente superior, aunque no significativamente, en el grupo de pacientes ventilados comparado con el de no ventilados. Este hecho puede explicarse por la mayor gravedad (medido mediante el score OASIS) que tienen los pacientes que necesitan ventilación, tal y cómo se aprecia en el siguiente gráfico.

Figura 7. Relación OASIS score y VM



5.3.7 Scores de severidad.

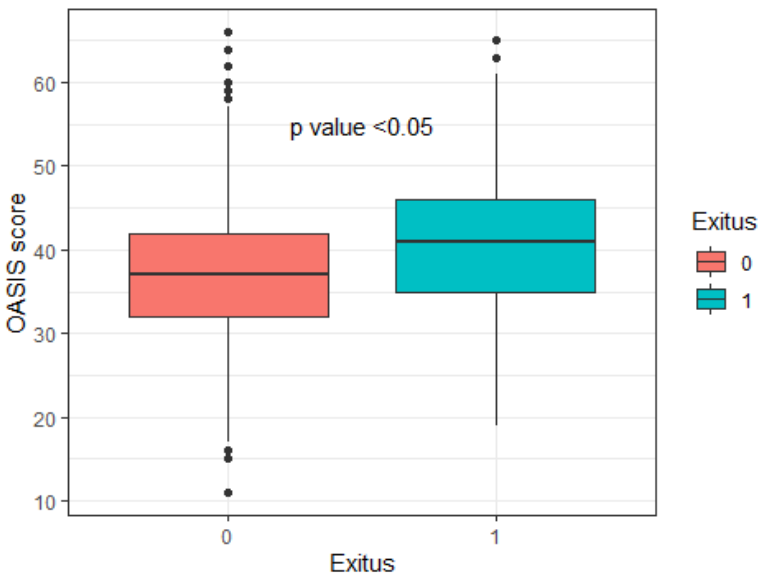
El **OASIS** asigna una puntuación según la gravedad de los pacientes calculada en base a 10 variables:

- Duración del ingreso antes de entrar en UCI.
- Puntuación GCS.
- Frecuencia cardiaca.
- Tensión arterial media.
- Frecuencia respiratoria.
- Temperatura.
- Diuresis.
- Necesidad de ventilación mecánica.
- Cirugía programada o urgente.

Según los pesos asignados a cada uno de estos ítems el rango de puntuación está comprendido entre 0-66, de forma que a mayor puntuación corresponde una mayor gravedad (6). El estudio original extrae los datos del APACHE IV (66) en un intento de simplificar dicho score. Por tanto, al igual que el APACHE, utiliza para su cálculo los peores valores de cada variable en las primeras 24 horas.

La figura 8 muestra la relación del score con la mortalidad en la cohorte estudiada.

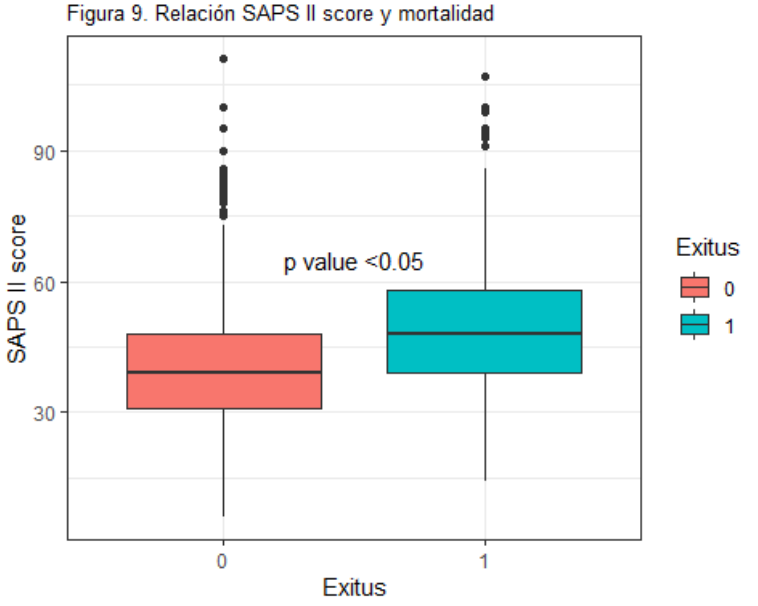
Figura 8. Relación OASIS score y mortalidad



Encontramos en este caso diferencias significativas en la puntuación de OASIS entre el grupo de exitus respecto a los que sobreviven, con una mayor puntuación para los primeros.

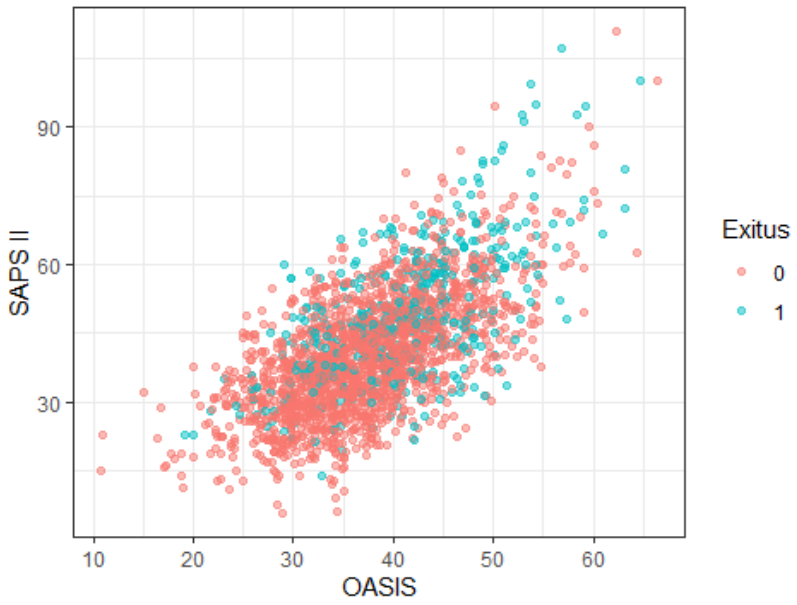
El **SAPS II** se calcula en base a 17 variables (7): 12 variables fisiológicas, edad, tipo de ingreso (cirugía programada, cirugía urgente o ingreso médico) y 3 variables de enfermedad subyacente (síndrome de inmunodeficienca adquirida, enfermedad metastásica y enfermedad hematológica maligna). El rango de puntuación está entre 0-166, que se corresponde con una mortalidad del 0-100%. Comparte con OASIS que el cálculo se realiza con el peor valor de cada variable en las primeras 24 horas. EL gráfico que relaciona SAPS II con la mortalidad encuentra diferencias significativas entre aquellos pacientes que fallecen y los que sobreviven a los 30 días (figura 9).

Figura 9. Relación SAPS II score y mortalidad



La correlación entre ambos scores (figura 10) es de `0.65`, lo que nos indica que su cálculo se basa en la medición de ítems similares.

Figura 10. Correlación SAPS II ~ OASIS



5.3.8 Comorbilidad.

La comorbilidad se ha recogido tal y como describe Elixhauser en su estudio (63), lo que implica la tabulación de 30 variables binarias obtenidas a partir del ICD-9. Este proceso tiene la ventaja de poder obtener la información a través de un algoritmo que procese datos puramente administrativos. En este estudio usaremos para implementar un modelo predictivo aquellos ítems de comorbilidad que demuestran una asociación significativa con la mortalidad (tabla 14).

Tabla 14. *Items de comorbilidad significativos.*

item	p_valor
cardiac_arrhythmias	0.0002459
hypertension	0.0016366
diabetes_uncomplicated	0.0164299
liver_disease	0.0121210
lymphoma	0.0409985
metastatic_cancer	0.0000000
solid_tumor	0.0064434
coagulopathy	0.0003720
obesity	0.0003089

5.4 Modelo de predicción de mortalidad a 30 días.

5.4.1 Selección de predictores.

En el modelo de predicción incluiré aquellas variables que han resultado significativas en el análisis univariante realizado en los apartados previos, y que por tanto, presenta mayor capacidad de discriminación entre aquellos que fallecen a los 30 días y los que sobreviven. Aquellas columnas definidas como factor se recodifican como one-hot.

c) Datos generales: edad, peso.

d) Mediciones fisiológicas: TAM, diuresis.

e) Gasometría arterial: FiO2, sO2, PaCO2, HCO3, lactato.

f) Hemograma: hemoglobina, leucocitos, plaquetas.

g) Bioquímica: BUN.

h) Fármacos: dopamina, noradrenalina, heparina.

i) Comorbilidad: arritmias cardiacas, hipertension arterial, diabetes no complicada, enfermedad hepática, linfoma, cáncer metastásico, tumor sólido, coagulopatía, obesidad.

Así pues, el dataset contiene un total de 25 predictores y `2149 registros.

5.4.2 División del dataset en train y test.

Para entrenar y posteriormente probar el modelo que desarrollemos dividiremos el conjunto de datos en un grupo de entrenamiento (70%) y un grupo test (30%). Esta división se mantendrá para cada uno de los modelos analizados. El grupo test unicamente se utilizará para calcular el rendimiento final del modelo en una población diferente a la del entrenamiento.

El grupo train y test están formados por 1505 y 644 registros, respectivamente. El número de predictores es de 25.

5.4.3 Transformación de los datos.

El conjunto de datos está formado por variables de muy diversa índole, con unidades de medida muy diferentes. Se hace necesario el hacer algunas transformaciones previo a la aplicación de los diferentes algoritmos de ML. Aquellas columnas numéricas serán estandarizadas en un rango 0-1 mediante el método max-min. La estandarización se realiza tanto para el grupo train como para el grupo test según los parámetros del grupo train.

5.4.4 Implementación de modelos de Machine Learning.

En primer lugar se realiza un barrido en el grupo train de los siguientes algoritmos de ML: *KNN, Naive-Bayes, regresión logística (glm), SVM con kernel lineal, SVM con kernel*

Gaussiano y Random Forest. Este análisis preliminar se realiza mediante el paquete *Caret* de *R* con la configuración por defecto de los hiperparámetros. En la tabla 15 se resume el resultado del AUC obtenido para cada uno de los algoritmos.

Tabla 15. *Rentabilidad de diferentes algoritmos de ML.*

Modelo	ROC
knn	0.591
glm	0.692
nb	0.679
svmLinear	0.627
svmRadial	0.679
rf	0.700

Observamos que los dos mejores algoritmos nos los ofrece la regresión logística y el random forest. A continuacion se intentará optimizar cada uno de estos dos algoritmos mediante la configuración de los hiperparámetros. El cálculo se realiza de nuevo mediante el paquete *Caret* de *R*, a través de una validación cruzada de 5 folds en el grupo train.

- **Regresión Logística.**

La regresión logística la realizamos mediante el método stepwise, eligiendo las variables del modelo final en base a la optimización del AIC.

Este algoritmo nos ofrece un AUC en el grupo train tras validación cruzada de `0.703`. Las 15 variables del modelo final son: `edad, peso, diuresis, fio2, so2, lactato, bun, nora1, heparina1, cardiac_arrhythmias, hypertension, diabetes_uncomplicated, liver_disease, metastatic_cancer, solid_tumor`. Las variables que mayor importancia tienen en el modelo, en base a sus coeficientes de regresión, son: `diuresis, so2 y lactato`.

- **Random forest.**

En este algoritmo configuramos el hiperparámetro *mtry* en un rango de 2-26, con un número de árboles para **bagging** de 500. El AUC que obtenemos en el grupo train tras validación cruzada de `0.698`. Las variables que mayor importancia tienen en el modelo, en base al índice Gini, son: `edad, plaquetas y lactato`.

Podemos también optimizar el RF mediante **boosting** con el método *xgbTree*. Para ello configuramos que la función *train* nos calcule el AUC en base al mejor valor obtenido de 100 combinaciones de los diferentes hiperparámetros (*tuneLength=100*). Este algoritmo nos ofrece un AUC en el grupo train tras validación cruzada de `0.687`.

5.4.5 Implementación de red neural profunda.

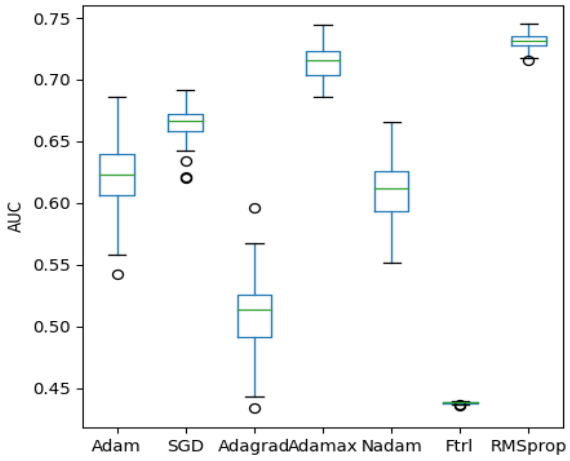
Las redes neuronales presentan un buen comportamiento ante problemas de clasificación o regresión complejos cuando disponemos de una gran cantidad de datos

(superior a 5000). Aunque este no es el caso, no obstante, desarrollaremos para este estudio dos redes neuronales para comprobar su comportamiento en este tipo de datos y poder comparar su rendimiento respecto al resto de técnicas de ML. Para ello nos centraremos en el desarrollo de una Red Neuronal Profunda (RNP) de tipo secuencial, y una RNP Convolucional.

La **RNP Secuencial** se ha construido con aquellos predictores que son significativos en el análisis univariante. A su vez el grupo de entrenamiento se ha subdividido en un subgrupo de validación que corresponde a un 20% de los registros del grupo train. Se han construido diferentes arquitecturas de RNP Secuencial, según los siguientes hiperparámetros: número de capas densas (de 1 a 4), número de nodos por capa (125-200-300), tipo de optimizador (*Adam, Adagrad, Adamax, Nadam, SGD, Ftrl y RMSprop*). Las gráficas en las que se muestra el rendimiento obtenido por los diferentes parámetros se han elaborado con los datos obtenidos de 100 repeticiones efectuadas para cada valor del hiperparámetro. La función de pérdida utilizada para los cálculos es *binary-crossentropy*. El rendimiento de la red se ha medido mediante la métrica *AUC* obtenida de la librería *sklearn* en lugar de la que ofrece *TensorFlow* ya que esta última es una aproximación a la primera que se calcula durante el proceso de entrenamiento y que en este trabajo hemos comprobado que daba valores excesivamente altos. Dado el disbalance que existe entre el porcentaje de pacientes que sobreviven (80%) y los que fallecen (20%) se ha realizado un ajuste en la corrección de los pesos durante el proceso de ajuste de la red mediante la función *class_weight* en una relación 1:4.

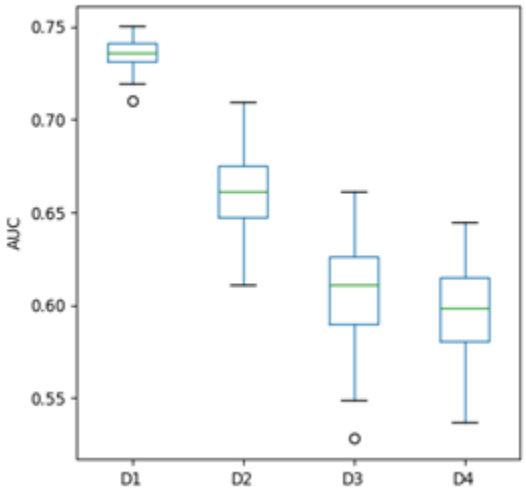
A continuación, vemos el rendimiento de la red en el grupo de validación en función del optimizador utilizado (figura 11).

Figura 11. Comparación de los diferentes optimizadores.



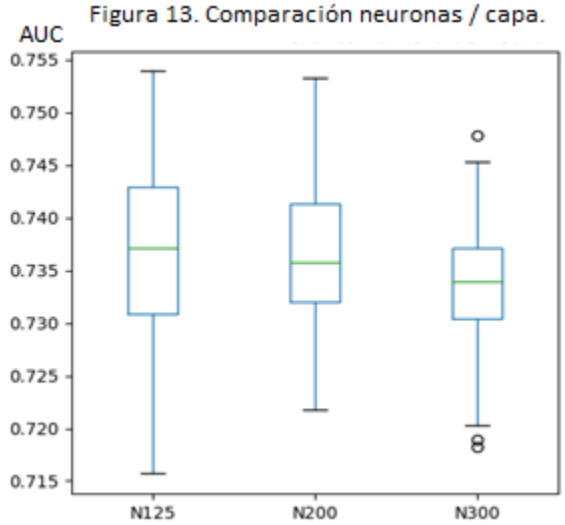
La figura 12 muestra el rendimiento de la red según el número de capas profundas utilizado.

Figura 12. Comparación número capas densas.



Finalmente, en la figura 13 tenemos el rendimiento de la red en el grupo de validación según el número de neuronas por capa.

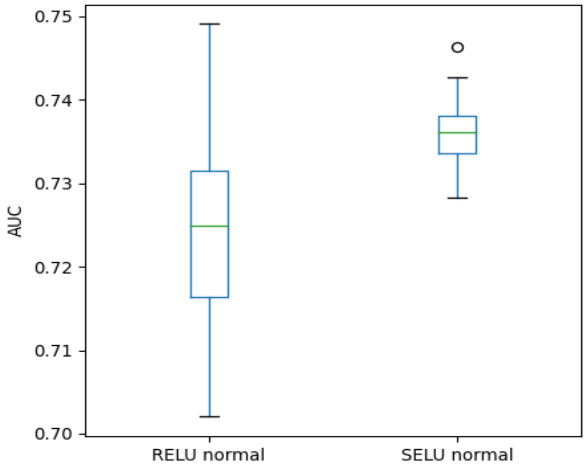
Figura 13. Comparación neuronas / capa.



Según estos resultados el mejor rendimiento se obtiene al utilizar 1 capa densa de 125 neuronas, utilizando el optimizador *RMSprop*.

Por defecto la inicialización de los pesos de cada nodo se realiza de forma aleatoria según una distribución uniforme. Para intentar mejorar el rendimiento se puede configurar que dicha inicialización siga una distribución N(0,1), a la vez que se cambia la función de activación de ReLu a SeLu. Para una muestra de 100 modelos de RNP secuencial el resultado obtenido en el subgrupo de validación es el siguiente:

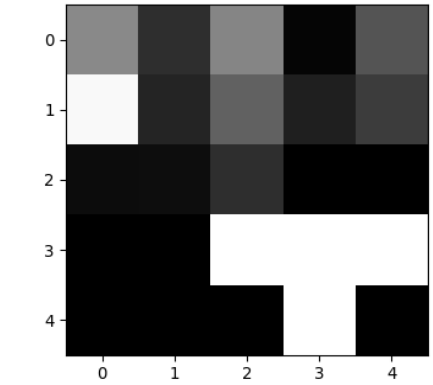
Figura 14. Inicialización normal de pesos.



La **RNP Convolucional** (CNN) se ha construido con las mismas variables que la RNP Secuencial. Utilizando la misma división del grupo train en *entrenamiento* y *validación*. Se han diseñado diferentes arquitecturas de RNP Convolucional, según los siguientes hiperparametros: número de capas convolucionales (de 1 a 4), número de kernels (125-200-300), tipo de optimizador (*Adam, Adagrad, Adamax, Nadam, SGD, Ftrl y RMSprop*). Las gráficas en las que se muestra el rendimiento obtenido por los diferentes parámetros se han elaborado con los datos obtenidos de 100 repeticiones efectuadas para cada valor del hiperparámetro. La función de pérdida utilizada para los cálculos es *binary-crossentropy*. El rendimiento de la red se ha medido mediante la métrica *AUC* obtenida de la librería *sklearn*.

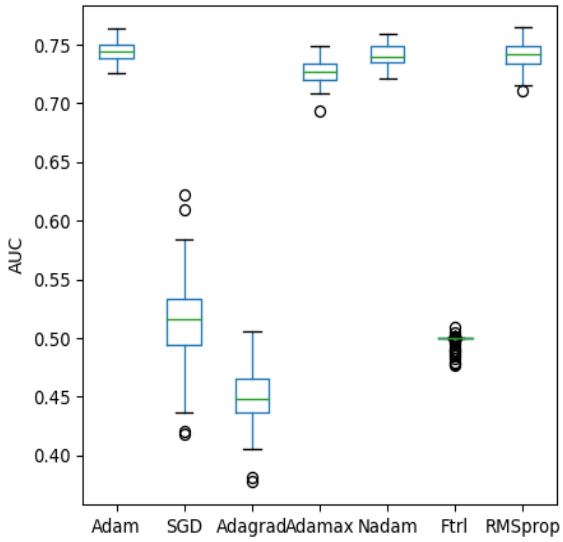
Puesto que las RNP Convolucionales están diseñadas para leer imágenes, la entrada en la red se ha de realizar en forma de matriz. En este estudio redimensionamos el vector X de 25 predictores en una matriz de 5 x 5. En la siguiente figura se muestra como ejemplo la imagen 5 x 5 obtenida de un individuo con la característica *exitus=0*.

Figura 15. Mapa de bits de un pacientes exitus=0



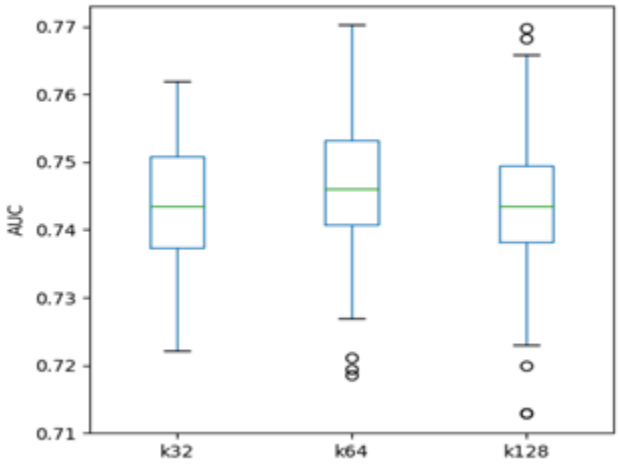
La figura 16 nos muestra el rendimiento de la red en el grupo de validación en función del optimizador utilizado.

Figura 16. Comparación de los diferentes optimizadores.



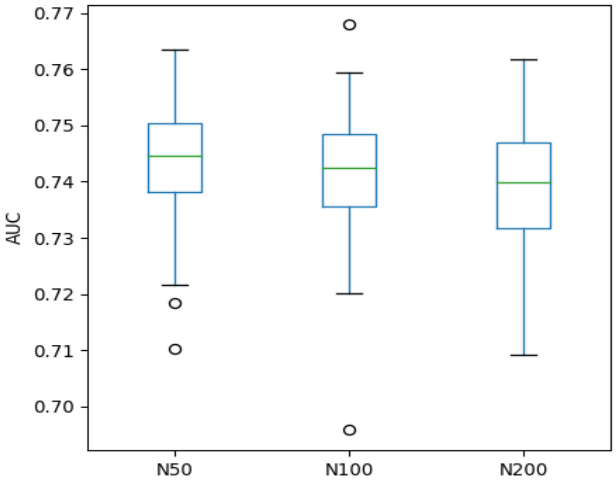
Seguidamente muestro (figura 17) el rendimiento de la red en función del número de kernels.

Figura 17. Comparación número de kernels.



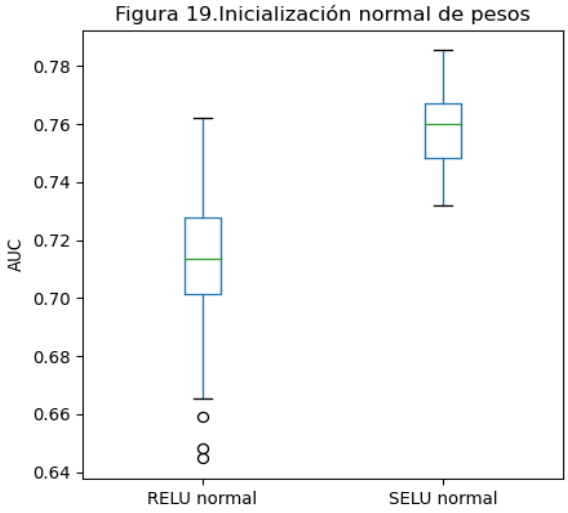
La figura 18 compara el rendimiento de la red en función del número de neuronas en la capa densa.

Figura 18. Comparación diferentes número de neuronas.



En base a estos resultado comprobamos que el mejor rendimiento lo conseguimos con 64 kernels usando un optimizador *Nadam* y con 50 neuronas en la capa densa. Al igual que hicimos en el modelo de red secuencial, podemos variar la inicialización de pesos a modo N(0,1) modificando además la función de activación de ReLu a SeLu, intentando así optimizar el rendimiento del modelo. Para una muestra de 100 modelos CNN con las características comentadas el resultado obtenido en el subgrupo de validación es el siguiente (figura 19):

Figura 19.Inicialización normal de pesos

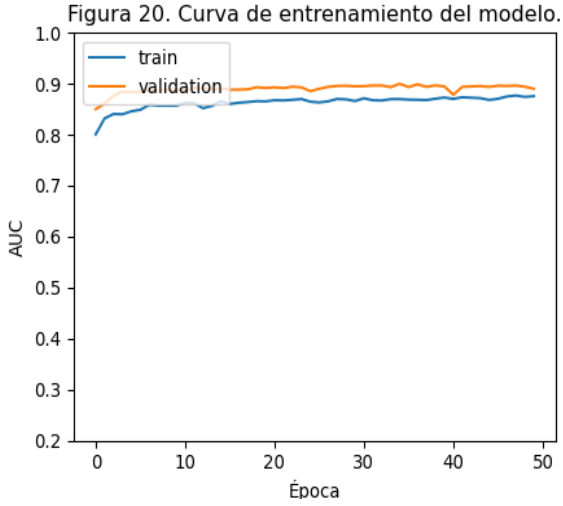


Observamos que el mejor rendimiento lo obtenemos para una red convolucional optimizada mediante inicialización de pesos según distribución N(0,1) y función de activación SeLu.

Por tanto, en base a los resultados obtenidos en el grupo de validación, el AUC obtenido con el mejor modelo de RNP secuencial densa es `0.74`, y el AUC obtenido con el mejor

modelo de CNN es `0.76`. Así pues, nos quedaremos con el modelo implementado por la red CNN para realizar la comparación con el resto de modelos de ML. La curva de entrenamiento para 50 épocas del modelo CNN se muestra en la siguiente figura.

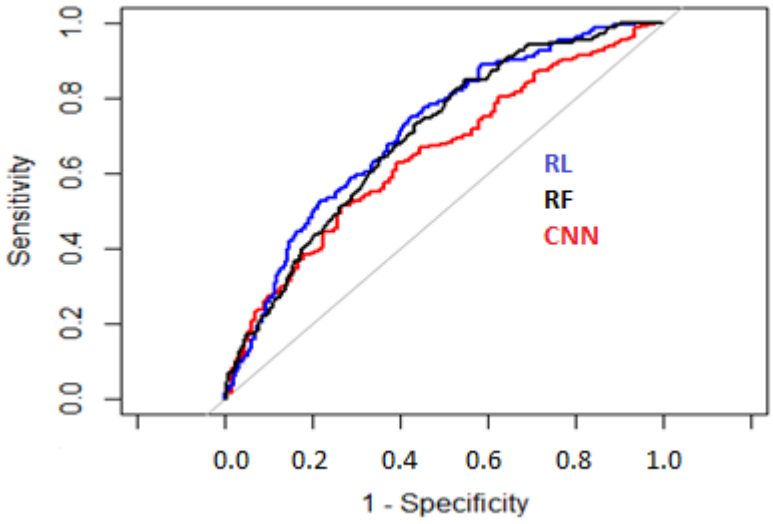
Figura 20. Curva de entrenamiento del modelo.



5.4.6 Validación en el grupo test.

Una vez que hemos estudiado los diferentes modelos de ML y deep learning estamos en condiciones de aplicarlos al grupo test y comparar los resultados. En la figura 21 se muestran las curvas ROC de los modelos estudiados: regresión logística (RL), random forest (RF) y CNN.

Figura 21. Rendimiento de los modelos en el grupo test.

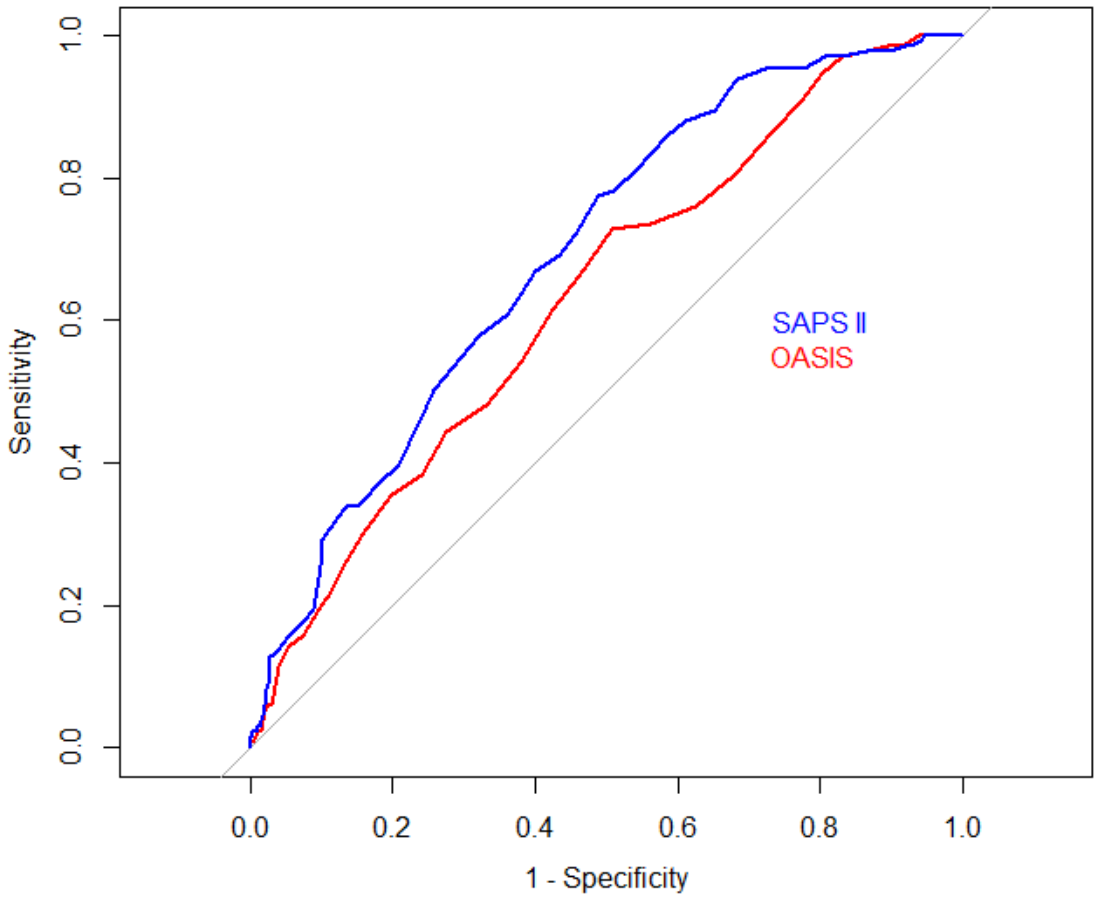


El AUC en el grupo test de los modelos Regresión logística, Random Forest y CNN es `0.71`, `0.7` y `0.65`, respectivamente.

5.4.7 Comparación de resultados con los scores.

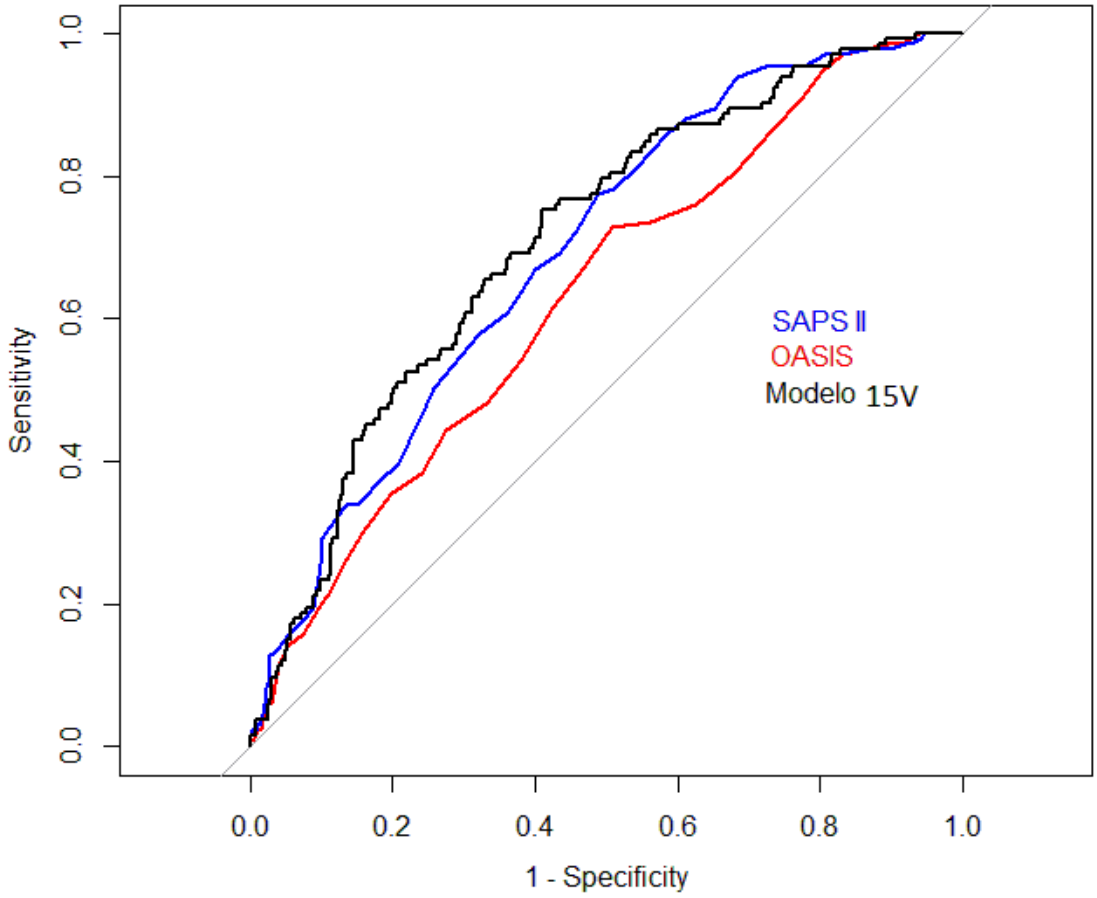
La siguiente figura muestra el rendimiento de los scores OASIS y SAPS II en el grupo test de la cohorte de pacientes con insuficiencia respiratoria estudiada. Destaca que el score SAPS II presenta un mejor rendimiento en el grupo test respecto al score OASIS para predecir la mortalidad a 30 días (AUC 0.69 vs 0.63).

Figura 22. Comparación rendimiento SAPS II - OASIS.



La figura 23 muestra el rendimiento del mejor modelo obtenido en este estudio (regresión logística), comparado con los scores previos. Observamos cómo el modelo de regresión obtenido a partir de los 15 predictores estudiados (modelo 15V) clasifica ligeramente mejor que los scores OASIS y SAPS II.

Figura 23. Comparación rendimiento SAPS II - OASIS - Modelo 15V



La siguiente tabla resume las métricas obtenidas en el grupo test de los scores de referencia (OASIS y SAPS II) y del mejor modelo de clasificación que hemos obtenido (modelo 15V).

Tabla 16. *Métricas de los modelos.*

PREDICCION	AUC	Kappa	SENS	ESP	VPP	VPN
OASIS	0.63	0.14	0.73	0.49	0.27	0.87
SAPS II	0.69	0.18	0.77	0.51	0.29	0.90
Modelo 15V	0.71	0.22	0.74	0.58	0.32	0.90

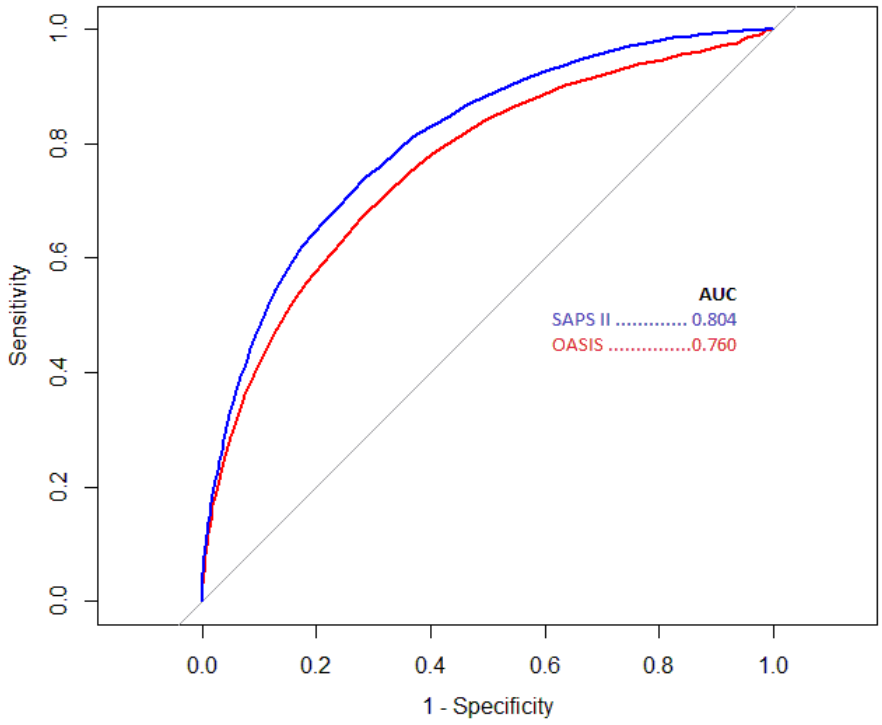
6. DISCUSIÓN.

En este estudio se ha intentado aproximar un modelo de clasificación centrado en una cohorte muy específica: pacientes ingresados en UCI con el diagnóstico de IRA sin SDRA. Durante todo el proceso se ha hecho evidente la dificultad que presentaba la cohorte seleccionada, ya que no hemos encontrado predictores que permitan discriminar aquellos pacientes que fallecen a los 30 días frente a los que no. Variables que podrían indicarnos gravedad de la IRA, como puede ser los valores gasométricos, no se mostraban claramente como discriminadores de mortalidad. Lo mismo ha ocurrido con otros predictores, como necesidad de ventilación mecánica o necesidad de drogas vasoactivas. Aunque muchas de estas variables resultaban significativas en el análisis univariante, las diferencias encontradas tenían poca relevancia clínica.

Las técnicas de ML que se han mostrado más efectivas han sido la regresión logística y el random forest. Estos hallazgos coinciden con lo encontrado en otros trabajos sobre scores pronóstico (49,68,69). Aunque el rendimiento final obtenido en el grupo test supera a los scores SAPS II y OASIS, carece de la necesaria capacidad de discriminación como para poder ser usado en la práctica clínica diaria con fines pronósticos.

Con respecto a los dos índices de predicción de mortalidad más importantes, SAPS II y OASIS (6,7), se hace necesario destacar el pobre rendimiento que ofrecen ambos índices en la cohorte seleccionada (AUC de 0.69 y 0.63 para SAPS II y OASIS, respectivamente). Estos resultados contrastan con los obtenidos en pacientes ingresados en UCI en el trabajo original (AUC de 0.89 y 0.88 para SAPS II y OASIS, respectivamente) para predecir mortalidad (6,67). Cuando hemos aplicado los scores SAPS II y OASIS a toda la población de pacientes MIMIC III adultos (36522 pacientes) el rendimiento de los índices se aproxima mucho más a lo registrado en los trabajos originales (AUC de 0.80 y 0.76 para SAPS II y OASIS, respectivamente), tal y como se aprecia en la figura 24. Este hecho nos indica que los scores generales pueden presentar rendimientos significativamente inferiores cuando se aplican a cohortes específicas, tal y como ocurre en este trabajo. Hallazgos similares ya se habían descrito en el caso de IRA secundaria a SDRA (49). Probablemente una cohorte determinada está integrada por un tipo de pacientes más homogéneo que cuando se selecciona a todos los ingresos en UCI. Por tanto, el aplicar índices de pronóstico general a una subpoblación específica no permite evaluar correctamente las características de dicha cohorte.

Figura 24. Curva ROC de scores OASIS y SAPS II en el total de pacientes MIMIC III



Por otra parte, los dos modelos de deep learning utilizados (red neuronal secuencial densa y red convolucional) se han mostrado muy inferiores a las técnicas de ML implementadas. Los modelos de aprendizaje profundo precisan de datos masivos para aprender a encontrar patrones, algo de lo que no disponíamos en este estudio. Si bien se ha intentado realizar transferencia de aprendizaje a partir de un modelo de red neuronal entrenado inicialmente con la totalidad de los ingresos MIMIC III, los resultados de dicha transferencia fueron realmente pobres y dicho modelo fue desechado.

En la actualidad este es el único estudio en el que se evalúa la mortalidad a los 30 días mediante modelos de ML en pacientes con IRA sin SDRA. Previamente se había evaluado mediante RF a pacientes con IRA secundaria a SDRA (49), obteniendo rendimientos mucho mejores (AUC: 0.883). Estas diferencias en el rendimiento de los algoritmos de ML entre pacientes con IRA sin SDRA y pacientes con IRA con SDRA nos lleva a pensar que el primer grupo presenta una serie de características muy diferentes al segundo grupo que hace que el modelar una respuesta basada en la mortalidad presente grandes dificultades.

7. CONCLUSIONES.

7.1 Conclusiones.

En este trabajo se ha desarrollado un modelo de predicción de mortalidad en pacientes con IRA que mejora ligeramente dos scores clásicos globales como son el OASIS y el SAPS II. En general todos estos modelos adolecen de una falta de especificidad, con VPP demasiado bajos como para considerarlos de utilidad clínica.

Además, los scores clásicos SAPS II y OASIS han demostrado igualmente un rendimiento muy pobre en la cohorte estudiada, a diferencia del rendimiento que presentan en el global de pacientes ingresados en UCI. Ante esto nos debemos cuestionar la utilidad pronóstica de este tipo de scores globales cuando se aplican a una cohorte determinada.

Por tanto, los EHR han demostrado ser una fuente de datos clínicos que permiten generar evidencia clínica y, por las características de las variables recogidas en los sistemas EHR , se prestan a ser evaluados mediante técnicas de ML.

7.2 Líneas de futuro.

A lo largo del trabajo se han demostrado las limitaciones que tienen los scores globales de pronóstico al aplicarlos sobre cohortes de pacientes con unas características específicas. Sería interesante desarrollar futuros scores de predicción centrados en cohortes homogéneas. En esta línea, el uso de datos provenientes de EHR supone un gran avance, ya que nos permite acceder a gran cantidad de datos clínicos y permite el desarrollo de scores o modelos de predicción de forma más rápida que los medios clásicos.

A su vez el uso de técnicas de deep learning supone un reto en aquellas cohortes en las que las técnicas de ML no logran un adecuado rendimiento. No obstante, es importante el disponer de datos clínicos masivos con los que poder entrenar dichas redes, algo que se ha echado en falta en el presente estudio.

7.3 Seguimiento de la planificación.

La planificación efectuada al inicio del trabajo (apartado 2.5) se ha cumplido siguiendo los plazos establecidos. Las dificultades que se han presentado en las diferentes tareas se han subsanado sin afectar a las fechas de entrega. La mayor parte del tiempo invertido en este trabajo lo ha constituido el estudio y gestión de las tablas mediante BigQuery. La elaboración de los modelos ha consumido el resto del tiempo del trabajo, ya que al no conseguir un modelo con un rendimiento óptimo se ha intentado optimizar al máximo cada modelo implementado.

8. GLOSARIO.

ABE: Exceso de base.

AG: Anestesia General.

APACHE: Acute Physiology and Chronic Health Evaluation

ATC: Anatomical Therapeutic Chemical **Classification** System.

AUC: Area Under Curve.

AWS: Amazon Web Services.

BUN: Blood Urea Nitrogen.

CO2: Dióxido de carbono.

CNN: Convolutional Neural Net.

CPT: Current Procedural Terminology.

DBMSs: Sistemas de gestión de bases de datos

DSM: Diagnostic and Statistical Manual of Mental Disorders.

E: Especificidad

EAP: Edema Agudo de Pulmón.

ECG: Electrocardiograma.

ECR: Ensayos Clínicos Randomizados.

EHR: Registros Electrónicos de Salud.

EMA: European Medicament Agency.

EPOC: Enfermedad Pulmonar Obstructiva Crónica.

FC: Frecuencia Cardiaca.

FDA: Food and Drug Administration.

FiO2: Fracción inspirada de Oxígeno.

GCS: Escala de Coma de Glasgow.

HB: Hemoglobina.

HCPCS: Healthcare Comon Procedure Coding System.

HCO3: Bicarbonato.

HTO: Hematocrito.

ICD: International Classification of Diseases.

ICD-CM: International Classification of Diseases Clinical Modification.

ICPC: International Classification of Primary Care

IRA: Insuficiencia Respiratoria Aguda.

KNN: k-Nearest Neighbors.

ML: Machine Learning.

MIMIC III: Medical Information Mart for Intensive Care.

NA: datos perdidos.

NB: Naive-Bayes

NDCc: National Drug Code

OASIS: Oxford Acute Security of Illness Score.

PaCO2: Presión arterial de CO2.

PaO2: Presión arterial de Oxígeno.

PCR: Proteína C Reactiva.

RF: Random Forest.

RL: Regresión Logística.

RNP: Red Neuronal Profunda.

RWD: Real World Data.

RWE: Real World Evidence.

S: Sensibilidad.

SAPS II: Simplified Acute Physiologic Score.

SDRA: Síndrome de Distrés Respiratorio Agudo.

spO2: Saturación parcial arterial de oxígeno.

SQL: Structured Query Language.

SVM: Support Vector Machine

TAD: Tensión arterial diastólica.

TAM: Tensión arterial media.

TAS: Tensión arterial sistólica.

TEP: Tromboembolismo pulmonar.

TFM: Trabajo Fin de Máster.

UCI: Unidad de Cuidados Intensivos.

VAS: Vía Aérea Superior.

VI: Ventilación mecánica Invasiva.

VM: Ventilación Mecánica.

VNI: Ventilación mecánica No Invasiva.

VPN: Valor Predictivo Negativo.

VPP: Valor Predictivo Positivo.

9. ANEXOS.

9.1 Anexo I: certificado del curso 'Data or Specimens Only Research.' Collaborative Institutional Training Initiative (CITI program).

COLLABORATIVE INSTITUTIONAL TRAINING INITIATIVE (CITI PROGRAM)

COMPLETION REPORT - PART 1 OF 2
COURSEWORK REQUIREMENTS*

* NOTE: Scores on this Requirements Report reflect quiz completions at the time all requirements for the course were met. See list below for details. See separate Transcript Report for more recent quiz scores, including those on optional (supplemental) course elements.

- **Name:** Eduardo González Constán (ID: 10804644)
- **Institution Affiliation:** Massachusetts Institute of Technology Affiliates (ID: 1912)
- **Institution Email:** egonzalezco@uoc.edu
- **Institution Unit:** Biostatistics

- **Curriculum Group:** Human Research
- **Course Learner Group:** Data or Specimens Only Research
- **Stage:** Stage 1 - Basic Course

- **Record ID:** 47341585
- **Completion Date:** 12-Feb-2022
- **Expiration Date:** 11-Feb-2025
- **Minimum Passing:** 90
- **Reported Score*:** 94

REQUIRED AND ELECTIVE MODULES ONLY	DATE COMPLETED	SCORE
Belmont Report and Its Principles (ID: 1127)	11-Feb-2022	3/3 (100%)
History and Ethics of Human Subjects Research (ID: 498)	11-Feb-2022	5/5 (100%)
Basic Institutional Review Board (IRB) Regulations and Review Process (ID: 2)	11-Feb-2022	5/5 (100%)
Records-Based Research (ID: 5)	12-Feb-2022	3/3 (100%)
Genetic Research in Human Populations (ID: 6)	12-Feb-2022	4/5 (80%)
Populations in Research Requiring Additional Considerations and/or Protections (ID: 16680)	12-Feb-2022	5/5 (100%)
Research and HIPAA Privacy Protections (ID: 14)	12-Feb-2022	5/5 (100%)
Conflicts of Interest in Human Subjects Research (ID: 17464)	12-Feb-2022	4/5 (80%)
Massachusetts Institute of Technology (ID: 1290)	12-Feb-2022	No Quiz

For this Report to be valid, the learner identified above must have had a valid affiliation with the CITI Program subscribing institution identified above or have been a paid Independent Learner.

Verify at: www.citiprogram.org/verify/?k1225845e-143e-4a96-a089-7e660873b2ff-47341585

Collaborative Institutional Training Initiative (CITI Program)
Email: support@citiprogram.org
Phone: 888-529-5929
Web: https://www.citiprogram.org

9.2 Anexo II: código SQL

9.2.1 Anexo II.1: tablas madre.

Código BigQuery en SQL standard para crear las tablas madre iniciales.

```
--- GENERACION DE TABLAS -----------------------------

-- Unifico tablas: admisions + patients--> t_admissions:
create or replace table `angelic-button-331918.dataset.t_admissions` AS
select a.subject_id ,a.hadm_id ,a.religion ,a.ethnicity ,a.deathtime,a.ho
spital_expire_flag ,
p.gender ,p.dob
from `physionet-data.mimiciii_clinical.admissions` a,`physionet-data.mimi
ciii_clinical.patients` p
where a.subject_id =p.subject_id ;

-- Unifico tablas: t_admissions + temp_icustays-->t1_admissions:
create or replace table `angelic-button-331918.dataset.t1_admissions` AS
select i.intime,i.outtime,i.los, i.ICUSTAY_ID,a.*
from `angelic-button-331918.dataset.t_admissions` a left join `physionet-
data.mimiciii_clinical.icustays` i
using (hadm_id);

-- Calculo edad de los pacientes y tras cuántos días fallecen en la tabla
t1_admissions-->t2_pacientes:
create or replace table `angelic-button-331918.dataset.t2_pacientes` AS
select subject_id ,hadm_id ,icustay_id,los as duracion_ingreso,religion,e
thnicity ,gender,
date_diff (intime,dob,day)/365 as edad,hospital_expire_flag as exitus, da
te_diff (deathtime,intime,day) as fallece
from `angelic-button-331918.dataset.t1_admissions`;

-- Selecciono unicamente los pacientes entre 18-100 años -->t3_pacientes:
create or replace table `angelic-button-331918.dataset.t3_pacientes` AS
select *  from `angelic-button-331918.dataset.t2_pacientes`  where edad>=
18 and edad<100;

-- Añado índice elixhauser a la tabla t3 de pacientes: t3_pacientes + t_c
omorbilidad --> t4_pacientes:
create or replace table `angelic-button-331918.dataset.t4_pacientes` AS
SELECT  t3.*, c.elixhauser_SID29,c.elixhauser_SID30,c.elixhauser_vanwalra
ven
FROM `angelic-button-331918.dataset.t3_pacientes` t3
left join `angelic-button-331918.dataset.t_comorbilidad` c
using(hadm_id);

-- Unifico tablas temp_procedures_icd + temp_d_icd_procedures --> t1_proc
edimientos:
```

```
create or replace table `angelic-button-331918.dataset.t1_procedimientos`
AS
select d.* , p.subject_id ,p.hadm_id,p.seq_num
from `physionet-data.mimiciii_clinical.procedures_icd` p
left join `physionet-data.mimiciii_clinical.d_icd_procedures` d
using (icd9_code)

-- Unifico tablas temp_diagnoses_icd + temp_d_icd_diagnoses --> t1_diagno
sticos:
create or replace table `angelic-button-331918.dataset.t1_diagnosticos` A
S
select d_icd.*,icd.subject_id ,icd.hadm_id ,icd.seq_num
from `physionet-data.mimiciii_clinical.diagnoses_icd` icd
left join `physionet-data.mimiciii_clinical.d_icd_diagnoses` d_icd
using(icd9_code);

-- Unifico tablas t1_diagnosticos + t1_procedimientos --> t2_dxproc:
create or replace table `angelic-button-331918.dataset.t2_dxproc` AS
select td.subject_id ,td.hadm_id ,td.icd9_code as icd9_dx ,td.short_title
as short_title_dx ,td.long_title as long_title_dx,tp.icd9_code as icd9_pr
oc ,
tp.short_title as short_title_proc ,tp.long_title as long_title_proc
from `angelic-button-331918.dataset.t1_diagnosticos`  td left join `angel
ic-button-331918.dataset.t1_procedimientos` tp
using (subject_id)

-- Añado variable intime (fecha ingreso en uci) a la tabla temp_charteven
ts --> t1_chartevents:
create or replace table `angelic-button-331918.dataset.t1_chartevents` AS
select t1.intime ,tc.*
from `physionet-data.mimiciii_clinical.chartevents` tc
left join `angelic-button-331918.dataset.t1_admissions` t1
using(hadm_id);

-- Calculo momento de la medición en la tabla t1_chartevents --> t1_1_cha
rtevents:
create or replace table `angelic-button-331918.dataset.t1_1_chartevents`
AS
select subject_id,hadm_id ,icustay_id,itemid,valuenum ,valueuom, ERROR as
errores,
date_diff(charttime,intime,hour)as momento
from `angelic-button-331918.dataset.t1_chartevents` t1;

-- Selecciono las mediciones que no presentan error en la medición --> t2
_chartevents:
create or replace table `angelic-button-331918.dataset.t2_chartevents` AS
SELECT subject_id,hadm_id ,icustay_id,itemid,valuenum ,valueuom,momento
FROM `angelic-button-331918.dataset.t1_1_chartevents`
where errores is null or errores=0;
```

```
-- Añado variable intime (fecha ingreso en uci) e icustay_id a la tabla l
abevents --> t1_labevents:
create or replace table `angelic-button-331918.dataset.t1_labevents` AS
SELECT lb.HADM_ID,lb.SUBJECT_ID,t1.icustay_id,lb.ITEMID,lb.VALUENUM,lb.VA
LUEUOM,lb.CHARTTIME,t1.intime
FROM `physionet-data.mimiciii_clinical.labevents` lb left join `angelic-b
utton-331918.dataset.t1_admissions` t1
using(hadm_id)

-- Calculo momento de la medición en la tabla t1_labevents --> t2_labeven
ts:
create or replace table `angelic-button-331918.dataset.t2_labevents` AS
select subject_id,hadm_id ,icustay_id,itemid,valuenum ,valueuom,
date_diff(charttime,intime,hour)as momento
from `angelic-button-331918.dataset.t1_labevents`;

----- CÓDIGO BIGQUERY PARA CREAR TABLA VENTILACION MECANICA ----------

-- Creo la tabla con aquellos pacientes que han llevado VM (VMI y VMNI) -
-> t0_vm:
create or replace table `angelic-button-331918.dataset.t0_vm` AS
select t1.ICUSTAY_ID,t1.STARTTIME,t1.ENDTIME,t1.itemid,t2.LABEL
FROM
(SELECT distinct ICUSTAY_ID, STARTTIME,ENDTIME,itemid FROM `physionet-dat
a.mimiciii_clinical.procedureevents_mv`
where itemid in (225792,225794)) as t1
inner join
(SELECT LABEL, itemid FROM `physionet-data.mimiciii_clinical.d_items`
where itemid in (225792,225794)) as t2
using(itemid)
order by t1.ICUSTAY_ID

-- Creo la tabla con aquellos pacientes que han iniciado VM <24 horas des
de el ingreso en uci y duración >1h --> t_vm:
create table `angelic-button-331918.dataset.t_vm` (`subject_id` integer,`
hadm_id` integer, `icustay_id` integer,
`vm` string,`duracion_vm` float64, `inicio_vm` float64);
insert into `angelic-button-331918.dataset.t_vm`
with tabla as
(select subject_id,hadm_id,icustay_id,itemid,label,date_diff (endtime,sta
rttime,hour) as duracion_vm,
date_diff(STARTTIME, intime, hour) as inicio_vm
from `angelic-button-331918.dataset.t1_admissions` inner join `angelic-bu
tton-331918.dataset.t0_vm`
using (icustay_id))
select subject_id, hadm_id,icustay_id,label as vm, duracion_vm,inicio_vm
from tabla
where inicio_vm<=24 and duracion_vm>=60
```

9.2.2 Anexo II.2: tablas con media de variables.

Código en BigQuery para crear las tablas con la media de las variables en las primeras 24 horas de ingreso en UCI.

```
---- CÓDIGO BIGQUERY PARA CREAR VARIABLES: MEDIA DE LAS PRIMERAS 24 HORAS EN UCI ---

-- Query para crear --> tabla t_peso:
create or replace table `angelic-button-331918.dataset.t_peso` as
select icustay_id, weight as peso
FROM `physionet-data.mimiciii_derived.weight_first_day`
where weight is not null;

-- Query para crear --> tabla t_talla:
create or replace table `angelic-button-331918.dataset.t_talla` as
select icustay_id, Height as talla
FROM `physionet-data.mimiciii_derived.heightfirstday`
where Height is not null;

-- Query para crear --> tabla t_tas:
create or replace table `angelic-button-331918.dataset.t_tas` as
select subject_id,hadm_id, icustay_id, sysbp_mean as tas
FROM `physionet-data.mimiciii_derived.vitals_first_day`
where sysbp_mean is not null;

-- Query para crear --> tabla t_tam:
create or replace table `angelic-button-331918.dataset.t_tam` as
select subject_id,hadm_id, icustay_id, meanbp_mean as tam
FROM `physionet-data.mimiciii_derived.vitals_first_day`
where meanbp_mean  is not null;

-- Query para crear --> tabla t_tad:
create or replace table `angelic-button-331918.dataset.t_tad` as
select subject_id,hadm_id, icustay_id, diasbp_mean as tad
FROM `physionet-data.mimiciii_derived.vitals_first_day`
where diasbp_mean  is not null;

-- Query para crear --> tabla t_fc:
create table `angelic-button-331918.dataset.t_fc` (`subject_id` integer,`
hadm_id` integer, `icustay_id` integer,`fc` float64);
insert into `angelic-button-331918.dataset.t_fc`
select subject_id, hadm_id,icustay_id,avg(valuenum) as fc
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (211,220045) and momento<24
group by hadm_id,subject_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_temp:
create table `angelic-button-331918.dataset.t_temp` (`subject_id` integer
```

```
,`hadm_id` integer, `icustay_id` integer, `temp` float64);
insert into `angelic-button-331918.dataset.t_temp`
select subject_id, hadm_id, icustay_id,avg((valuenum-32)*0.5555555) as te
mperatura
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (678,3654,223761) and momento<24
group by hadm_id,subject_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_diuresis:
create or replace table `angelic-button-331918.dataset.t_diuresis` as
select subject_id,hadm_id,icustay_id, urineoutput as diuresis
FROM `physionet-data.mimiciii_derived.urine_output_first_day`
where urineoutput is not null;

-- Query para crear --> tabla t_so2:
create table `angelic-button-331918.dataset.t_so2` (`subject_id` integer,
`hadm_id` integer, `icustay_id` integer, `so2` float64);
insert into `angelic-button-331918.dataset.t_so2`
select subject_id, hadm_id,icustay_id,avg(valuenum) as so2
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (646,3288,220277,8498,834) and momento<24
group by hadm_id,subject_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_hb:
create table `angelic-button-331918.dataset.t_hb` (`subject_id` integer,`
hadm_id` integer, `icustay_id` integer, `hb` float64);
insert into `angelic-button-331918.dataset.t_hb`
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as hb
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (220228,814) and momento<24
union all
select subject_id, hadm_id,icustay_id,valuenum as hb
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (50811,51222) and momento<24)
select subject_id,hadm_id,icustay_id,avg(hb) as hb from t2
where hb is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_hto:
create table `angelic-button-331918.dataset.t_hto` (`subject_id` integer,
`hadm_id` integer, `icustay_id` integer,`hto` float64);
insert into `angelic-button-331918.dataset.t_hto`
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as hto
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (220545,226540,813) and momento<24
```

```
union all
select subject_id, hadm_id,icustay_id,valuenum as hto
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (50810,51221) and momento<24)
select subject_id,hadm_id,icustay_id,avg(hto) as hto from t2
where hto is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_leucos:
create table `angelic-button-331918.dataset.t_leucos` (`subject_id` integ
er,`hadm_id` integer, `icustay_id` integer, `leucos` float64);
insert into `angelic-button-331918.dataset.t_leucos`
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as leucos
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (220546,4200,1127,861,1542) and momento<24
union all
select subject_id, hadm_id,icustay_id,valuenum as leucos
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (51300) and momento<24)
select subject_id,hadm_id,icustay_id,avg(leucos) as leucos from t2
where leucos is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_glucosa:
create table `angelic-button-331918.dataset.t_glucosa` (`subject_id` inte
ger,`hadm_id` integer, `icustay_id` integer,`glucosa` float64);
insert into `angelic-button-331918.dataset.t_glucosa`
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as glucosa
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (228388,220621,226537,811,3744,1529) and momento<24
union all
select subject_id, hadm_id,icustay_id,valuenum as glucosa
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (50809,50931) and momento<24)
select subject_id,hadm_id,icustay_id,avg(glucosa) as glucosa from t2
where glucosa is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_creat:
create table `angelic-button-331918.dataset.t_creat` (`subject_id` intege
r,`hadm_id` integer, `icustay_id` integer,`creat` float64);
insert into `angelic-button-331918.dataset.t_creat`
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as creat
from `angelic-button-331918.dataset.t2_chartevents`
```

```
where itemid in (220615,791,3750,1525) and momento<24
union all
select subject_id, hadm_id,icustay_id,valuenum as creat
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (50912,51081) and momento<24)
select subject_id,hadm_id,icustay_id,avg(creat) as creat from t2
where creat is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_bun:
create table `angelic-button-331918.dataset.t_bun` (`subject_id` integer,
`hadm_id` integer, `icustay_id` integer, `bun` float64);
insert into `angelic-button-331918.dataset.t_bun`
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as bun
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (1162,5876,225624) and momento<24)
select subject_id,hadm_id,icustay_id,avg(bun) as bun from t2
where bun is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_bnp:
create or replace table `angelic-button-331918.dataset.t_bnp` as
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as bnp
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (7294,227446,225622) and momento<24
union all
select subject_id, hadm_id,icustay_id,valuenum as bnp
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (50963) and momento<24)
select subject_id,hadm_id,icustay_id,avg(bnp) as bnp from t2
where bnp is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_plaquetas:
create table `angelic-button-331918.dataset.t_plaquetas` (`subject_id` in
teger,`hadm_id` integer, `icustay_id` integer, `plaquetas` float64);
insert into `angelic-button-331918.dataset.t_plaquetas`
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as plaquetas
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (6256,227457,828) and momento<24
union all
select subject_id, hadm_id,icustay_id,valuenum as plaquetas
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (51265) and momento<24)
```

```
select subject_id,hadm_id,icustay_id,avg(plaquetas) as plaquetas from t2
where plaquetas is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_ph:
create table `angelic-button-331918.dataset.t_ph` (`subject_id` integer,`
hadm_id` integer, `icustay_id` integer, `ph` float64);
insert into `angelic-button-331918.dataset.t_ph`
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as ph
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (1673,1126,4753,780,223830) and momento<24
union all
select subject_id, hadm_id,icustay_id,valuenum as ph
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (50820) and momento<24)
select subject_id,hadm_id,icustay_id,avg(ph) as ph from t2
where ph is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_abe:
create or replace table `angelic-button-331918.dataset.t_abe` as
select subject_id,hadm_id,icustay_id, BASEEXCESS as abe
FROM `physionet-data.mimiciii_derived.bloodgasfirstdayarterial`
where BASEEXCESS is not null;

-- Query para crear --> tabla t_hco3:
create table `angelic-button-331918.dataset.t_hco3` (`subject_id` integer
,`hadm_id` integer, `icustay_id` integer, `hco3` float64);
insert into `angelic-button-331918.dataset.t_hco3`
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as hco3
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (227443,812) and momento<24
union all
select subject_id, hadm_id,icustay_id,valuenum as hco3
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (50882) and momento<24)
select subject_id,hadm_id,icustay_id,avg(hco3) as hco3 from t2
where hco3 is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_po2:
create table `angelic-button-331918.dataset.t_po2` (`subject_id` integer,
`hadm_id` integer, `icustay_id` integer, `po2` float64);
insert into `angelic-button-331918.dataset.t_po2`
with t2 as
```

```
(select subject_id, hadm_id,icustay_id,valuenum as po2
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (779,3837,3785) and momento<24
union all
select subject_id, hadm_id,icustay_id,valuenum as po2
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (50821) and momento<24)
select subject_id,hadm_id,icustay_id,avg(po2) as po2 from t2
where po2 is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_albu:
create table `angelic-button-331918.dataset.t_albu` (`subject_id` integer
,`hadm_id` integer, `icustay_id` integer, `albu` float64);
insert into `angelic-button-331918.dataset.t_albu`
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as albu
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (2358,772,1521,3727) and momento<24
union all
select subject_id, hadm_id,icustay_id,valuenum as albu
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (50862) and momento<24)
select subject_id,hadm_id,icustay_id,avg(albu) as albu from t2
where albu is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_pco2:
create table `angelic-button-331918.dataset.t_pco2` (`subject_id` integer
,`hadm_id` integer, `icustay_id` integer, `pco2` float64);
insert into `angelic-button-331918.dataset.t_pco2`
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as pco2
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (777,778,3835,3784) and momento<24
union all
select subject_id, hadm_id,icustay_id,valuenum as pco2
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (50818) and momento<24)
select subject_id,hadm_id,icustay_id,avg(pco2) as pco2 from t2
where pco2 is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_lactato:
create table `angelic-button-331918.dataset.t_lactato` (`subject_id` inte
ger,`hadm_id` integer, `icustay_id` integer, `lactato` float64);
insert into `angelic-button-331918.dataset.t_lactato`
```

```sql
select subject_id, hadm_id,icustay_id,avg(valuenum) as lactato
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (50813) and momento<24
group by hadm_id,subject_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_ddimer:
create table `angelic-button-331918.dataset.t_ddimer` (`subject_id` integ
er,`hadm_id` integer, `icustay_id` integer, `ddimer` float64);
insert into `angelic-button-331918.dataset.t_ddimer`
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as ddimer
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (225636,793,1526) and momento<24
union all
select subject_id, hadm_id,icustay_id,valuenum as ddimer
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (50915,51196) and momento<24)
select subject_id,hadm_id,icustay_id,avg(ddimer) as ddimer from t2
where ddimer is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_pcr:
create table `angelic-button-331918.dataset.t_pcr` (`subject_id` integer,
`hadm_id` integer, `icustay_id` integer, `pcr` float64);
insert into `angelic-button-331918.dataset.t_pcr`
with t2 as
(select subject_id, hadm_id,icustay_id,valuenum as pcr
from `angelic-button-331918.dataset.t2_chartevents`
where itemid in (227444) and momento<24
union all
select subject_id, hadm_id,icustay_id,valuenum as pcr
from `angelic-button-331918.dataset.t2_labevents`
where itemid in (50889) and momento<24)
select subject_id,hadm_id,icustay_id,avg(pcr) as pcr from t2
where pcr is not null
group by subject_id,hadm_id,icustay_id
order by hadm_id;

-- Query para crear --> tabla t_gcs:
create or replace table `angelic-button-331918.dataset.t_gcs` as
select subject_id,hadm_id, icustay_id, minGCS as gcs
FROM `physionet-data.mimiciii_derived.gcs_first_day`
where minGCS is not null;

-- Query para crear --> tabla t_fio2:
create table `angelic-button-331918.dataset.t_fio2` (`subject_id` integer
,`hadm_id` integer, `icustay_id` integer, `fio2` float64);
insert into `angelic-button-331918.dataset.t_fio2`
```

```
with t2 as
(select subject_id,hadm_id,icustay_id,valuenum as fio2 from `angelic-butt
on-331918.dataset.t2_chartevents`
where itemid=3420 and valuenum is not null and momento<24
union all
select subject_id,hadm_id,icustay_id,valuenum as fio2 from `angelic-butto
n-331918.dataset.t2_chartevents`
where itemid=223835 and valuenum is not null and valuenum between 21 and
100 and momento<24
union all
select subject_id,hadm_id,icustay_id,valuenum*100 as fio2 from `angelic-b
utton-331918.dataset.t2_chartevents`
where itemid=223835 and valuenum is not null and valuenum between 0.21 an
d 1 and momento<24
union all
select subject_id,hadm_id,icustay_id,valuenum as fio2 from `angelic-butto
n-331918.dataset.t2_chartevents`
where itemid=3422 and momento<24 and valuenum is not null
union all
select subject_id,hadm_id,icustay_id,valuenum as fio2 from `angelic-butto
n-331918.dataset.t2_labevents`
where itemid=50816 and valuenum is not null and valuenum between 21 and 1
00 and momento<24
union all
select subject_id,hadm_id,icustay_id,valuenum*100 as fio2 from `angelic-b
utton-331918.dataset.t2_labevents`
where itemid=50816 and valuenum is not null and valuenum between 0.21 and
1 and momento<24)

select subject_id,hadm_id,icustay_id,avg (fio2) as fio2 from t2
group by subject_id, hadm_id, icustay_id;
```

9.2.3 Anexo II.3: tablas de drogas.

Creación de las tablas con la medicación administrada en las primeras 24 horas.

```
------------ TABLAS DE DROGAS -------------------------

-- Query para crear --> tabla t_dopa:
create table `angelic-button-331918.dataset.t_dopa` (`subject_id` integer
,`hadm_id` integer, `icustay_id` integer,
`duracion_droga` float64,`inicio_droga` float64, `dopa` string);
insert into `angelic-button-331918.dataset.t_dopa`
with drogas as
(select subject_id,hadm_id,icustay_id,date_diff(ENDDATE, STARTDATE, hour)
as duracion_droga,
drug,gsn,ndc,date_diff(STARTDATE,intime,hour) as inicio_droga from
(select p.SUBJECT_ID,p.HADM_ID,p.ICUSTAY_ID,p.STARTDATE,p.ENDDATE,p.DRUG,
p.GSN,p.NDC, ic.INTIME
from `physionet-data.mimiciii_clinical.prescriptions` p left join `physio
net-data.mimiciii_clinical.icustays` ic
using (icustay_id)))
select subject_id,hadm_id,icustay_id,duracion_droga, inicio_droga,left(up
per(drug),4) as dopa from drogas
where ndc in (74582010,409910420,74780922,517180525,74910420,338100702,40
9780922,338101102)
and  inicio_droga<=24 and duracion_droga>=3;

-- Query para crear --> tabla t_nora:
create table `angelic-button-331918.dataset.t_nora` (`subject_id` integer
,`hadm_id` integer, `icustay_id` integer,
`duracion_droga` float64,`inicio_droga` float64, `nora` string);
insert into `angelic-button-331918.dataset.t_nora`
with drogas as
(select subject_id,hadm_id,icustay_id,date_diff(ENDDATE, STARTDATE, hour)
as duracion_droga,
drug,gsn,ndc,date_diff(STARTDATE,intime,hour) as inicio_droga from
(select p.SUBJECT_ID,p.HADM_ID,p.ICUSTAY_ID,p.STARTDATE,p.ENDDATE,p.DRUG,
p.GSN,p.NDC, ic.INTIME
from `physionet-data.mimiciii_clinical.prescriptions` p left join `physio
net-data.mimiciii_clinical.icustays` ic
using (icustay_id)))
select subject_id,hadm_id,icustay_id,duracion_droga, inicio_droga,left(up
per(drug),4) as nora from drogas
where ndc in (74144304,74704101,247120004,409144304,703115303,781893285,6
1553015311)
and  inicio_droga<=24 and duracion_droga>=3;

-- Query para crear --> tabla t_adre:
create table `angelic-button-331918.dataset.t_adre` (`subject_id` integer
,`hadm_id` integer, `icustay_id` integer,
```

```
`duracion_droga` float64,`inicio_droga` float64, `adre` string);
insert into `angelic-button-331918.dataset.t_adre`
with drogas as
(select subject_id,hadm_id,icustay_id,date_diff(ENDDATE, STARTDATE, hour)
as duracion_droga,
drug,gsn,ndc,date_diff(STARTDATE,intime,hour) as inicio_droga from
(select p.SUBJECT_ID,p.HADM_ID,p.ICUSTAY_ID,p.STARTDATE,p.ENDDATE,p.DRUG,
p.GSN,p.NDC, ic.INTIME
from `physionet-data.mimiciii_clinical.prescriptions` p left join `physio
net-data.mimiciii_clinical.icustays` ic
using (icustay_id)))
select subject_id,hadm_id,icustay_id,duracion_droga, inicio_droga,left(up
per(drug),4) as adre from drogas
where ndc in (74492134,74724101,409492134,409724101,42023012225)
and  inicio_droga<=24 and duracion_droga>=3;

-- Query para crear --> tabla t_dobuta:
create table `angelic-button-331918.dataset.t_dobuta` (`subject_id` integ
er,`hadm_id` integer, `icustay_id` integer,
`duracion_droga` float64,`inicio_droga` float64, `dobuta` string);
insert into `angelic-button-331918.dataset.t_dobuta`
with drogas as
(select subject_id,hadm_id,icustay_id,date_diff(ENDDATE, STARTDATE, hour)
as duracion_droga,
drug,gsn,ndc,date_diff(STARTDATE,intime,hour) as inicio_droga from
(select p.SUBJECT_ID,p.HADM_ID,p.ICUSTAY_ID,p.STARTDATE,p.ENDDATE,p.DRUG,
p.GSN,p.NDC, ic.INTIME
from `physionet-data.mimiciii_clinical.prescriptions` p left join `physio
net-data.mimiciii_clinical.icustays` ic
using (icustay_id)))
select subject_id,hadm_id,icustay_id,duracion_droga, inicio_droga,left(up
per(drug),4) as dobuta from drogas
where (ndc in (2717510,74234632,338107302,409234632,55390056090) or gsn i
n ('021502'))
and  inicio_droga<=24 and duracion_droga>=3;

-- Query para crear --> tabla t_heparina:
create table `angelic-button-331918.dataset.t_heparina` (`subject_id` int
eger,`hadm_id` integer, `icustay_id` integer,
`duracion_droga` float64,`inicio_droga` float64, `heparina` string);
insert into `angelic-button-331918.dataset.t_heparina`
with drogas as
(select subject_id,hadm_id,icustay_id,date_diff(ENDDATE, STARTDATE, hour)
as duracion_droga,
drug,gsn,ndc,date_diff(STARTDATE,intime,hour) as inicio_droga from
(select p.SUBJECT_ID,p.HADM_ID,p.ICUSTAY_ID,p.STARTDATE,p.ENDDATE,p.DRUG,
p.GSN,p.NDC, ic.INTIME
from `physionet-data.mimiciii_clinical.prescriptions` p left join `physio
net-data.mimiciii_clinical.icustays` ic
using (icustay_id)))
```

```
select subject_id,hadm_id,icustay_id,duracion_droga, inicio_droga,left(up
per(drug),4) as heparina from drogas
where (ndc in (74115112,74115170,409115170,409779362,641040025,641041425,
641243645,641244045,8290036005,17191003500,
63323026201,63323054011,63323054031,63323054201,64253022233,64253033335)
or gsn in ('006522'))
and  inicio_droga<=24 and duracion_droga>=3;

-- Query para crear --> tabla t_dexa:
create table `angelic-button-331918.dataset.t_dexa` (`subject_id` integer
,`hadm_id` integer, `icustay_id` integer,
`duracion_droga` float64,`inicio_droga` float64, `dexa` string);
insert into `angelic-button-331918.dataset.t_dexa`
with drogas as
(select subject_id,hadm_id,icustay_id,date_diff(ENDDATE, STARTDATE, hour)
as duracion_droga,
drug,gsn,ndc,date_diff(STARTDATE,intime,hour) as inicio_droga from
(select p.SUBJECT_ID,p.HADM_ID,p.ICUSTAY_ID,p.STARTDATE,p.ENDDATE,p.DRUG,
p.GSN,p.NDC, ic.INTIME
from `physionet-data.mimiciii_clinical.prescriptions` p left join `physio
net-data.mimiciii_clinical.icustays` ic
using (icustay_id)))
select subject_id,hadm_id,icustay_id,duracion_droga, inicio_droga,left(up
per(drug),3) as dexa from drogas
where (ndc in (54418425,54817625,54817925,517490125,517490525,641036725,6
41227741,703352403,63323016501,
63323016505,63323051610) or gsn in ('006776'))
and  inicio_droga<=24 and duracion_droga>=3;

-- Query para crear --> tabla t_predni:
create table `angelic-button-331918.dataset.t_predni` (`subject_id` integ
er,`hadm_id` integer, `icustay_id` integer,
`duracion_droga` float64,`inicio_droga` float64, `predni` string);
insert into `angelic-button-331918.dataset.t_predni`
with drogas as
(select subject_id,hadm_id,icustay_id,date_diff(ENDDATE, STARTDATE, hour)
as duracion_droga,
drug,gsn,ndc,date_diff(STARTDATE,intime,hour) as inicio_droga from
(select p.SUBJECT_ID,p.HADM_ID,p.ICUSTAY_ID,p.STARTDATE,p.ENDDATE,p.DRUG,
p.GSN,p.NDC, ic.INTIME
from `physionet-data.mimiciii_clinical.prescriptions` p left join `physio
net-data.mimiciii_clinical.icustays` ic
using (icustay_id)))
select subject_id,hadm_id,icustay_id,duracion_droga, inicio_droga,left(up
per(drug),4) as predni from drogas
where (ndc in (9003928,9019016,63323025803,9011312,9011319,55390020910,90
76502,9338901,63323026530,63323026530,
55390020910,9011319,9338901,9019016,55390020910,9076502))
and  inicio_droga<=24 and duracion_droga>=3;
```

9.2.4 Anexo II.4: tabla de comorbilidad de Elixhauser.

Código Bigquery para la creación de la tabla de comorbilidad de Elixhauser.

```
----- CODIGO EN BIGQUERY PARA CALCULAR LA COMORBILIDAD SEGÚN ELIXHAUSER -


create or replace table `angelic-button-331918.dataset.elixhauser_quan` A
S

with eliflg as
(
select hadm_id, seq_num, icd9_code
, CASE
  when icd9_code in ('39891','40201','40211','40291','40401','40403','404
11','40413','40491','40493') then 1
  when SUBSTR(icd9_code, 1, 4) in ('4254','4255','4257','4258','4259') th
en 1
  when SUBSTR(icd9_code, 1, 3) in ('428') then 1
  else 0 end as chf       /* Congestive heart failure */

, CASE
  when icd9_code in ('42613','42610','42612','99601','99604') then 1
  when SUBSTR(icd9_code, 1, 4) in ('4260','4267','4269','4270','4271','42
72','4273','4274','4276','4278','4279','7850','V450','V533') then 1
  else 0 end as arrhy

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('0932','7463','7464','7465','7466','V4
22','V433') then 1
  when SUBSTR(icd9_code, 1, 3) in ('394','395','396','397','424') then 1
  else 0 end as valve     /* Valvular disease */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('4150','4151','4170','4178','4179') th
en 1
  when SUBSTR(icd9_code, 1, 3) in ('416') then 1
  else 0 end as pulmcirc  /* Pulmonary circulation disorder */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('0930','4373','4431','4432','4438','44
39','4471','5571','5579','V434') then 1
  when SUBSTR(icd9_code, 1, 3) in ('440','441') then 1
  else 0 end as perivasc  /* Peripheral vascular disorder */

, CASE
  when SUBSTR(icd9_code, 1, 3) in ('401') then 1
  else 0 end as htn       /* Hypertension, uncomplicated */
```

```
, CASE
  when SUBSTR(icd9_code, 1, 3) in ('402','403','404','405') then 1
  else 0 end as htncx     /* Hypertension, complicated */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('3341','3440','3441','3442','3443','34
44','3445','3446','3449') then 1
  when SUBSTR(icd9_code, 1, 3) in ('342','343') then 1
  else 0 end as para      /* Paralysis */

, CASE
  when icd9_code in ('33392') then 1
  when SUBSTR(icd9_code, 1, 4) in ('3319','3320','3321','3334','3335','33
62','3481','3483','7803','7843') then 1
  when SUBSTR(icd9_code, 1, 3) in ('334','335','340','341','345') then 1
  else 0 end as neuro     /* Other neurological */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('4168','4169','5064','5081','5088') th
en 1
  when SUBSTR(icd9_code, 1, 3) in ('490','491','492','493','494','495','4
96','500','501','502','503','504','505') then 1
  else 0 end as chrnlung  /* Chronic pulmonary disease */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('2500','2501','2502','2503') then 1
  else 0 end as dm        /* Diabetes w/o chronic complications*/

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('2504','2505','2506','2507','2508','25
09') then 1
  else 0 end as dmcx      /* Diabetes w/ chronic complications */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('2409','2461','2468') then 1
  when SUBSTR(icd9_code, 1, 3) in ('243','244') then 1
  else 0 end as hypothy   /* Hypothyroidism */

, CASE
  when icd9_code in ('40301','40311','40391','40402','40403','40412','404
13','40492','40493') then 1
  when SUBSTR(icd9_code, 1, 4) in ('5880','V420','V451') then 1
  when SUBSTR(icd9_code, 1, 3) in ('585','586','V56') then 1
  else 0 end as renlfail  /* Renal failure */

, CASE
  when icd9_code in ('07022','07023','07032','07033','07044','07054') the
n 1
  when SUBSTR(icd9_code, 1, 4) in ('0706','0709','4560','4561','4562','57
```

```
22','5723','5724','5728','5733','5734','5738','5739','V427') then 1
  when SUBSTR(icd9_code, 1, 3) in ('570','571') then 1
  else 0 end as liver     /* Liver disease */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('5317','5319','5327','5329','5337','53
39','5347','5349') then 1
  else 0 end as ulcer     /* Chronic Peptic ulcer disease (includes bleed
ing only if obstruction is also present) */

, CASE
  when SUBSTR(icd9_code, 1, 3) in ('042','043','044') then 1
  else 0 end as aids      /* HIV and AIDS */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('2030','2386') then 1
  when SUBSTR(icd9_code, 1, 3) in ('200','201','202') then 1
  else 0 end as lymph     /* Lymphoma */

, CASE
  when SUBSTR(icd9_code, 1, 3) in ('196','197','198','199') then 1
  else 0 end as mets      /* Metastatic cancer */

, CASE
  when SUBSTR(icd9_code, 1, 3) in
  (
     '140','141','142','143','144','145','146','147','148','149','150','1
51','152'
    ,'153','154','155','156','157','158','159','160','161','162','163','1
64','165'
    ,'166','167','168','169','170','171','172','174','175','176','177','1
78','179'
    ,'180','181','182','183','184','185','186','187','188','189','190','1
91','192'
    ,'193','194','195'
  ) then 1
  else 0 end as tumor     /* Solid tumor without metastasis */

, CASE
  when icd9_code in ('72889','72930') then 1
  when SUBSTR(icd9_code, 1, 4) in ('7010','7100','7101','7102','7103','71
04','7108','7109','7112','7193','7285') then 1
  when SUBSTR(icd9_code, 1, 3) in ('446','714','720','725') then 1
  else 0 end as arth              /* Rheumatoid arthritis/collagen vascul
ar diseases */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('2871','2873','2874','2875') then 1
  when SUBSTR(icd9_code, 1, 3) in ('286') then 1
```

```
  else 0 end as coag      /* Coagulation deficiency */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('2780') then 1
  else 0 end as obese     /* Obesity      */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('7832','7994') then 1
  when SUBSTR(icd9_code, 1, 3) in ('260','261','262','263') then 1
  else 0 end as wghtloss  /* Weight loss */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('2536') then 1
  when SUBSTR(icd9_code, 1, 3) in ('276') then 1
  else 0 end as lytes     /* Fluid and electrolyte disorders */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('2800') then 1
  else 0 end as bldloss   /* Blood loss anemia */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('2801','2808','2809') then 1
  when SUBSTR(icd9_code, 1, 3) in ('281') then 1
  else 0 end as anemdef  /* Deficiency anemias */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('2652','2911','2912','2913','2915','29
18','2919','3030','3039','3050','3575','4255','5353','5710','5711','5712'
,'5713','V113') then 1
  when SUBSTR(icd9_code, 1, 3) in ('980') then 1
  else 0 end as alcohol /* Alcohol abuse */

, CASE
  when icd9_code in ('V6542') then 1
  when SUBSTR(icd9_code, 1, 4) in ('3052','3053','3054','3055','3056','30
57','3058','3059') then 1
  when SUBSTR(icd9_code, 1, 3) in ('292','304') then 1
  else 0 end as drug /* Drug abuse */

, CASE
  when icd9_code in ('29604','29614','29644','29654') then 1
  when SUBSTR(icd9_code, 1, 4) in ('2938') then 1
  when SUBSTR(icd9_code, 1, 3) in ('295','297','298') then 1
  else 0 end as psych /* Psychoses */

, CASE
  when SUBSTR(icd9_code, 1, 4) in ('2962','2963','2965','3004') then 1
  when SUBSTR(icd9_code, 1, 3) in ('309','311') then 1
  else 0 end as depress  /* Depression */
```

```
from `physionet-data.mimiciii_clinical.diagnoses_icd` icd
where seq_num != 1 -- we do not include the primary icd-9 code
)

, eligrp as
(
  select hadm_id
  , max(chf) as chf
  , max(arrhy) as arrhy
  , max(valve) as valve
  , max(pulmcirc) as pulmcirc
  , max(perivasc) as perivasc
  , max(htn) as htn
  , max(htncx) as htncx
  , max(para) as para
  , max(neuro) as neuro
  , max(chrnlung) as chrnlung
  , max(dm) as dm
  , max(dmcx) as dmcx
  , max(hypothy) as hypothy
  , max(renlfail) as renlfail
  , max(liver) as liver
  , max(ulcer) as ulcer
  , max(aids) as aids
  , max(lymph) as lymph
  , max(mets) as mets
  , max(tumor) as tumor
  , max(arth) as arth
  , max(coag) as coag
  , max(obese) as obese
  , max(wghtloss) as wghtloss
  , max(lytes) as lytes
  , max(bldloss) as bldloss
  , max(anemdef) as anemdef
  , max(alcohol) as alcohol
  , max(drug) as drug
  , max(psych) as psych
  , max(depress) as depress
from eliflg
group by hadm_id
)

select adm.hadm_id
, chf as congestive_heart_failure
, arrhy as cardiac_arrhythmias
, valve as valvular_disease
, pulmcirc as pulmonary_circulation
, perivasc as peripheral_vascular
-- we combine "htn" and "htncx" into "HYPERTENSION"
, case
```

```
    when htn = 1 then 1
    when htncx = 1 then 1
  else 0 end as hypertension
, para as paralysis
, neuro as other_neurological
, chrnlung as chronic_pulmonary
-- only the more severe comorbidity (complicated diabetes) is kept
, case
    when dmcx = 1 then 0
    when dm = 1 then 1
  else 0 end as diabetes_uncomplicated
, dmcx as diabetes_complicated
, hypothy as hypothyroidism
, renlfail as renal_failure
, liver as liver_disease
, ulcer as peptic_ulcer
, aids as aids
, lymph as lymphoma
, mets as metastatic_cancer
-- only the more severe comorbidity (metastatic cancer) is kept
, case
    when mets = 1 then 0
    when tumor = 1 then 1
  else 0 end as solid_tumor
, arth as rheumatoid_arthritis
, coag as coagulopathy
, obese as obesity
, wghtloss as weight_loss
, lytes as fluid_electrolyte
, bldloss as blood_loss_anemia
, anemdef as deficiency_anemias
, alcohol as alcohol_abuse
, drug as drug_abuse
, psych as psychoses
, depress as depression

FROM `physionet-data.mimiciii_clinical.admissions` adm
left join eligrp eli
  on adm.hadm_id = eli.hadm_id
order by adm.hadm_id;
```

9.2.5 Anexo II.5: scores.

Código para calcular los scores SAPS II y OASIS.

```
------------ SCORE DE SEVERIDAD SAPSII -------------------------

create or replace table `angelic-button-331918.dataset.t_sapsii` AS
SELECT subject_id, hadm_id,icustay_id, sapsii,sapsii_prob FROM `physionet
-data.mimiciii_derived.sapsii`;


------------ SCORE DE SEVERIDAD OASIS -------------------------

-- Creo el score de severidad OASIS ---> t_oasis

create or replace table `angelic-button-331918.dataset.t_oasis` as

select subject_id,hadm_id,icustay_id,oasis,oasis_PROB from
(
with surgflag as
(
  select ie.icustay_id
    , max(case
        when lower(curr_service) like '%surg%' then 1
        when curr_service = 'ORTHO' then 1
    else 0 end) as surgical
  FROM `physionet-data.mimiciii_clinical.icustays` ie
  left join `physionet-data.mimiciii_clinical.services` se
    on ie.hadm_id = se.hadm_id
    and se.transfertime < DATETIME_ADD(ie.intime, INTERVAL '1' DAY)
  group by ie.icustay_id
)
, cohort as
(
select ie.subject_id, ie.hadm_id, ie.icustay_id
      , ie.intime
      , ie.outtime
      , adm.deathtime
      , DATETIME_DIFF(ie.intime, adm.admittime, MINUTE) as preiculos
      , DATETIME_DIFF(ie.intime, pat.dob, YEAR) as age
      , gcs.mingcs
      , vital.heartrate_max
      , vital.heartrate_min
      , vital.meanbp_max
      , vital.meanbp_min
      , vital.resprate_max
      , vital.resprate_min
      , vital.tempc_max
      , vital.tempc_min
```

```
      , vent.vent as mechvent
      , uo.urineoutput

      , case
          when adm.ADMISSION_TYPE = 'ELECTIVE' and sf.surgical = 1
            then 1
          when adm.ADMISSION_TYPE is null or sf.surgical is null
            then null
          else 0
        end as electivesurgery

      -- age group
      , case
        when DATETIME_DIFF(ie.intime, pat.dob, YEAR) <= 1 then 'neonate'
        when DATETIME_DIFF(ie.intime, pat.dob, YEAR) <= 15 then 'middle'
        else 'adult' end as icustay_age_group

      -- mortality flags
      , case
          when adm.deathtime between ie.intime and ie.outtime
            then 1
          when adm.deathtime <= ie.intime -- sometimes there are typograp
hical errors in the death date
            then 1
          when adm.dischtime <= ie.outtime and adm.discharge_location = '
DEAD/EXPIRED'
            then 1
          else 0 end
        as icustay_expire_flag
      , adm.hospital_expire_flag
FROM `physionet-data.mimiciii_clinical.icustays` ie
inner join `physionet-data.mimiciii_clinical.admissions` adm
  on ie.hadm_id = adm.hadm_id
inner join `physionet-data.mimiciii_clinical.patients` pat
  on ie.subject_id = pat.subject_id
left join surgflag sf
  on ie.icustay_id = sf.icustay_id
-- join to custom tables to get more data....
left join `physionet-data.mimiciii_derived.gcs_first_day` gcs
  on ie.icustay_id = gcs.icustay_id
left join `physionet-data.mimiciii_derived.vitals_first_day` vital
  on ie.icustay_id = vital.icustay_id
left join `physionet-data.mimiciii_derived.urine_output_first_day` uo
  on ie.icustay_id = uo.icustay_id
left join `physionet-data.mimiciii_derived.ventilation_first_day` vent
  on ie.icustay_id = vent.icustay_id
)
, scorecomp as
(
select co.subject_id, co.hadm_id, co.icustay_id
```

```
, co.icustay_age_group
, co.icustay_expire_flag
, co.hospital_expire_flag

-- Below code calculates the component scores needed for oasis
, case when preiculos is null then null
     when preiculos < 10.2 then 5
     when preiculos < 297 then 3
     when preiculos < 1440 then 0
     when preiculos < 18708 then 1
     else 2 end as preiculos_score
,  case when age is null then null
      when age < 24 then 0
      when age <= 53 then 3
      when age <= 77 then 6
      when age <= 89 then 9
      when age >= 90 then 7
      else 0 end as age_score
,  case when mingcs is null then null
      when mingcs <= 7 then 10
      when mingcs < 14 then 4
      when mingcs = 14 then 3
      else 0 end as gcs_score
,  case when heartrate_max is null then null
      when heartrate_max > 125 then 6
      when heartrate_min < 33 then 4
      when heartrate_max >= 107 and heartrate_max <= 125 then 3
      when heartrate_max >= 89 and heartrate_max <= 106 then 1
      else 0 end as heartrate_score
,  case when meanbp_min is null then null
      when meanbp_min < 20.65 then 4
      when meanbp_min < 51 then 3
      when meanbp_max > 143.44 then 3
      when meanbp_min >= 51 and meanbp_min < 61.33 then 2
      else 0 end as meanbp_score
,  case when resprate_min is null then null
      when resprate_min <   6 then 10
      when resprate_max >  44 then  9
      when resprate_max >  30 then  6
      when resprate_max >  22 then  1
      when resprate_min <  13 then 1 else 0
      end as resprate_score
,  case when tempc_max is null then null
      when tempc_max > 39.88 then 6
      when tempc_min >= 33.22 and tempc_min <= 35.93 then 4
      when tempc_max >= 33.22 and tempc_max <= 35.93 then 4
      when tempc_min < 33.22 then 3
      when tempc_min > 35.93 and tempc_min <= 36.39 then 2
      when tempc_max >= 36.89 and tempc_max <= 39.88 then 2
      else 0 end as temp_score
```

```
,  case when UrineOutput is null then null
      when UrineOutput < 671.09 then 10
      when UrineOutput > 6896.80 then 8
      when UrineOutput >= 671.09
       and UrineOutput <= 1426.99 then 5
      when UrineOutput >= 1427.00
       and UrineOutput <= 2544.14 then 1
      else 0 end as urineoutput_score
,  case when mechvent is null then null
      when mechvent = 1 then 9
      else 0 end as mechvent_score
,  case when electivesurgery is null then null
      when electivesurgery = 1 then 0
      else 6 end as electivesurgery_score

, preiculos
, age
, mingcs as gcs
,  case when heartrate_max is null then null
      when heartrate_max > 125 then heartrate_max
      when heartrate_min < 33 then heartrate_min
      when heartrate_max >= 107 and heartrate_max <= 125 then heartrate_m
ax
      when heartrate_max >= 89 and heartrate_max <= 106 then heartrate_ma
x
      else (heartrate_min+heartrate_max)/2 end as heartrate
,  case when meanbp_min is null then null
      when meanbp_min < 20.65 then meanbp_min
      when meanbp_min < 51 then meanbp_min
      when meanbp_max > 143.44 then meanbp_max
      when meanbp_min >= 51 and meanbp_min < 61.33 then meanbp_min
      else (meanbp_min+meanbp_max)/2 end as meanbp
,  case when resprate_min is null then null
      when resprate_min <   6 then resprate_min
      when resprate_max >  44 then resprate_max
      when resprate_max >  30 then resprate_max
      when resprate_max >  22 then resprate_max
      when resprate_min <  13 then resprate_min
      else (resprate_min+resprate_max)/2 end as resprate
,  case when tempc_max is null then null
      when tempc_max > 39.88 then tempc_max
      when tempc_min >= 33.22 and tempc_min <= 35.93 then tempc_min
      when tempc_max >= 33.22 and tempc_max <= 35.93 then tempc_max
      when tempc_min < 33.22 then tempc_min
      when tempc_min > 35.93 and tempc_min <= 36.39 then tempc_min
      when tempc_max >= 36.89 and tempc_max <= 39.88 then tempc_max
      else (tempc_min+tempc_max)/2 end as temp
,  UrineOutput
,  mechvent
,  electivesurgery
```

```
from cohort co
)
, score as
(
select s.*
    , coalesce(age_score,0)
    + coalesce(preiculos_score,0)
    + coalesce(gcs_score,0)
    + coalesce(heartrate_score,0)
    + coalesce(meanbp_score,0)
    + coalesce(resprate_score,0)
    + coalesce(temp_score,0)
    + coalesce(urineoutput_score,0)
    + coalesce(mechvent_score,0)
    + coalesce(electivesurgery_score,0)
    as oasis
from scorecomp s
)
select
  subject_id, hadm_id, icustay_id
  , icustay_age_group
  , hospital_expire_flag
  , icustay_expire_flag
  , oasis
  , 1 / (1 + exp(- (-6.1746 + 0.1275*(oasis) ))) as oasis_PROB
  , age, age_score
  , preiculos, preiculos_score
  , gcs, gcs_score
  , heartrate, heartrate_score
  , meanbp, meanbp_score
  , resprate, resprate_score
  , temp, temp_score
  , urineoutput, urineoutput_score
  , mechvent, mechvent_score
  , electivesurgery, electivesurgery_score
from score
order by icustay_id

)
```

9.2.6 Anexo II.6: tabla final.

Código BigQuery para la creación de la tabla final sobre la que se hará el análisis estadístico.

```
-------- CONSTRUCCION TABLA FINAL -------------------------
-- Creo la tabla de todos los pacientes con las variables seleccionadas:
-->TABLA_PACIENTES
create or replace table `angelic-button-331918.dataset.TABLA_PACIENTES` a
s
select *
from `angelic-button-331918.dataset.t4_pacientes`
left join (select icustay_id, peso from `angelic-button-331918.dataset.t_
peso` ) using (icustay_id)
left join (select icustay_id, talla from `angelic-button-331918.dataset.t
_talla`) using (icustay_id)
left join (select icustay_id, gcs from `angelic-button-331918.dataset.t_g
cs`) using (icustay_id)
left join (select icustay_id, tas from `angelic-button-331918.dataset.t_t
as`) using (icustay_id)
left join (select icustay_id, tad from `angelic-button-331918.dataset.t_t
ad`) using (icustay_id)
left join (select icustay_id, tam from `angelic-button-331918.dataset.t_t
am`) using (icustay_id)
left join (select icustay_id, fc from `angelic-button-331918.dataset.t_fc
`    where fc is not null) using (icustay_id)
left join (select icustay_id, temp from `angelic-button-331918.dataset.t_
temp`  where temp is not null) using (icustay_id)
left join (select icustay_id, diuresis from `angelic-button-331918.datase
t.t_diuresis`) using (icustay_id)
left join (select icustay_id, fio2 from `angelic-button-331918.dataset.t_
fio2` ) using (icustay_id)
left join (select icustay_id, so2 from `angelic-button-331918.dataset.t_s
o2`   ) using (icustay_id)
left join (select icustay_id, ph from `angelic-button-331918.dataset.t_ph
`     ) using (icustay_id)
left join (select icustay_id, abe from `angelic-button-331918.dataset.t_a
be`   ) using (icustay_id)
left join (select icustay_id, hco3 from `angelic-button-331918.dataset.t_
hco3` ) using (icustay_id)
left join (select icustay_id, po2 from `angelic-button-331918.dataset.t_p
o2`   ) using (icustay_id)
left join (select icustay_id, pco2 from `angelic-button-331918.dataset.t_
pco2` ) using (icustay_id)
left join (select icustay_id, hb from `angelic-button-331918.dataset.t_hb
`     ) using (icustay_id)
left join (select icustay_id, hto from `angelic-button-331918.dataset.t_h
to`   ) using (icustay_id)
left join (select icustay_id, leucos from `angelic-button-331918.dataset.
```

```
t_leucos` ) using (icustay_id)
left join (select icustay_id, plaquetas from `angelic-button-331918.datas
et.t_plaquetas` ) using (icustay_id)
left join (select icustay_id, glucosa from `angelic-button-331918.dataset
.t_glucosa` ) using (icustay_id)
left join (select icustay_id, creat from `angelic-button-331918.dataset.t
_creat`     ) using (icustay_id)
left join (select icustay_id, bun from `angelic-button-331918.dataset.t_b
un`         ) using (icustay_id)
left join (select icustay_id, albu from `angelic-button-331918.dataset.t_
albu`       ) using (icustay_id)
left join (select icustay_id, bnp from `angelic-button-331918.dataset.t_b
np`         ) using (icustay_id)
left join (select icustay_id, ddimer from `angelic-button-331918.dataset.
t_ddimer` ) using (icustay_id)
left join (select icustay_id, pcr from `angelic-button-331918.dataset.t_p
cr` ) using (icustay_id)
left join (select icustay_id, lactato from `angelic-button-331918.dataset
.t_lactato` ) using (icustay_id)
left join (select icustay_id, dopa, inicio_droga as inicio_dopa, duracion
_droga as duracion_dopa from `angelic-button-331918.dataset.t_dopa` ) usi
ng (icustay_id)
left join (select icustay_id, nora,inicio_droga as inicio_nora,duracion_d
roga as duracion_nora from `angelic-button-331918.dataset.t_nora` ) using
(icustay_id)
left join (select icustay_id, adre,inicio_droga as inicio_adre,duracion_d
roga as duracion_adre from `angelic-button-331918.dataset.t_adre` ) using
(icustay_id)
left join (select icustay_id, dobuta,inicio_droga as inicio_dobuta,duraci
on_droga as duracion_dobuta from `angelic-button-331918.dataset.t_dobuta`
) using (icustay_id)
left join (select icustay_id, heparina,inicio_droga as inicio_heparina,du
racion_droga as duracion_heparina from `angelic-button-331918.dataset.t_h
eparina` ) using (icustay_id)
left join (select icustay_id, dexa,inicio_droga as inicio_dexa,duracion_d
roga as duracion_dexa from `angelic-button-331918.dataset.t_dexa` ) using
(icustay_id)
left join (select icustay_id, predni,inicio_droga as inicio_predni,duraci
on_droga as duracion_predni from `angelic-button-331918.dataset.t_predni`
) using (icustay_id)
left join (select icustay_id, vm,inicio_vm, duracion_vm from `angelic-but
ton-331918.dataset.t_vm` ) using (icustay_id)
left join `angelic-button-331918.dataset.elixhauser_quan` using (hadm_id)
left join (select icustay_id,oasis,oasis_prob from`angelic-button-331918.
dataset.t_oasis`) using (icustay_id)
left join (select icustay_id,sapsii,sapsii_prob from`angelic-button-33191
8.dataset.t_sapsii`) using (icustay_id)

------ SELECCIONO COHORTE FINAL EN TABLA_PACIENTES -------------
```

```
-- SELECCIONO COHORTE DE SUJETOS CON ICD9 CORRESPONDIENTES A INSUFICIENCI
A RESPIRATORIA AGUDA --> t_cohorte
create table `angelic-button-331918.dataset.t_cohorte` (`hadm_id` integer
,`subject_id` integer, `icd9_dx` string,
`short_dx` string,`long_dx` string);
insert into `angelic-button-331918.dataset.t_cohorte`
SELECT distinct(hadm_id),subject_id,icd9_dx,short_title_dx as short_dx,lo
ng_title_dx as long_dx
FROM `angelic-button-331918.dataset.t2_dxproc`
where icd9_dx in ('51881','51884','51851','51853');

-- Query para crear tabla de cohorte con dx de insuficienca respiratoria
--> TABLA_IRA
CREATE or replace table  `angelic-button-331918.dataset.TABLA_IRA` AS
SELECT tabla.*,t.icd9_dx,t.long_dx
FROM `angelic-button-331918.dataset.t_cohorte` t
INNER JOIN `angelic-button-331918.dataset.TABLA_PACIENTES` tabla USING (h
adm_id);

-- Query para calcular mortalidad a 30 días (exitus30) en TABLA_IRA:
CREATE OR REPLACE TABLE `angelic-button-331918.dataset.TABLA_IRA` AS
SELECT * ,
    case when fallece<=30 then 1 else 0
    end as exitus30
FROM `angelic-button-331918.dataset.TABLA_IRA`;


-- Query para crear tabla definitiva--> TABLA_IRA_COMPLETA
CREATE or REPLACE table  `angelic-button-331918.dataset.TABLA_IRA_COMPLET
A` AS
with tabla_ira as
(SELECT *
FROM `angelic-button-331918.dataset.TABLA_IRA`
where tas is not null and
tad is not null and
gcs is not null and
fc is not null and
temp is not null and
diuresis is not null and
fio2 is not null and
po2 is not null and
pco2 is not null and
so2 is not null and
hco3 is not NULL and
abe is not null and
hb is not null and
hto is not null and
leucos is not null and
plaquetas is not null and
glucosa is not null and
```

```
creat is not null and
lactato is not null)

SELECT * except (rn)
FROM (select *,row_number() over (partition by subject_id) rn
    from tabla_ira )
where rn=1
order by subject_id;
```

9.2.7 Anexo II.7: calidad de tablas.

Código de BigQuery para crear la tablas con la que explorar la calidad de la tabla final.

```
-- CALIDAD DE LA TABLA: Calculo valores nulos en la tabla 'TABLA_IRA_COMP
LETA':

create or replace table `angelic-button-331918.dataset.TABLA_CALIDAD_IRA_
COMPLETA` AS

select * from
(select exitus30,count(*) as n from `angelic-button-331918.dataset.TABLA_
IRA_COMPLETA`
group by exitus30)
left join
(select exitus30,count(subject_id) as religion_null FROM `angelic-button-
331918.dataset.TABLA_IRA_COMPLETA`
where religion is null group by exitus30) using (exitus30)
left join
(select exitus30,count(subject_id) as gender_null FROM `angelic-button-33
1918.dataset.TABLA_IRA_COMPLETA`
where gender is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as ethnicity_null FROM `angelic-button
-331918.dataset.TABLA_IRA_COMPLETA`
where ethnicity is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as peso_null FROM `angelic-button-3319
18.dataset.TABLA_IRA_COMPLETA`
where peso is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as talla_null FROM `angelic-button-331
918.dataset.TABLA_IRA_COMPLETA`
where talla is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as gcs_null FROM `angelic-button-33191
8.dataset.TABLA_IRA_COMPLETA`
where gcs is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as tas_null FROM `angelic-button-33191
8.dataset.TABLA_IRA_COMPLETA`
where tas is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as tad_null FROM `angelic-button-33191
8.dataset.TABLA_IRA_COMPLETA`
where tad is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as tam_null FROM `angelic-button-33191
8.dataset.TABLA_IRA_COMPLETA`
```

```
where tam is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as fc_null FROM `angelic-button-331918
.dataset.TABLA_IRA_COMPLETA`
where fc is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as temp_null FROM `angelic-button-3319
18.dataset.TABLA_IRA_COMPLETA`
where temp is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as diuresis_null FROM `angelic-button-
331918.dataset.TABLA_IRA_COMPLETA`
where diuresis is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as fio2_null FROM `angelic-button-3319
18.dataset.TABLA_IRA_COMPLETA`
where fio2 is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as so2_null FROM `angelic-button-33191
8.dataset.TABLA_IRA_COMPLETA`
where so2 is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as ph_null FROM `angelic-button-331918
.dataset.TABLA_IRA_COMPLETA`
where ph is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as abe_null FROM `angelic-button-33191
8.dataset.TABLA_IRA_COMPLETA`
where abe is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as hco3_null FROM `angelic-button-3319
18.dataset.TABLA_IRA_COMPLETA`
where hco3 is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as po2_null FROM `angelic-button-33191
8.dataset.TABLA_IRA_COMPLETA`
where po2 is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as pco2_null FROM `angelic-button-3319
18.dataset.TABLA_IRA_COMPLETA`
where pco2 is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as hb_null FROM `angelic-button-331918
.dataset.TABLA_IRA_COMPLETA`
where hb is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as hto_null FROM `angelic-button-33191
8.dataset.TABLA_IRA_COMPLETA`
where hto is null group by exitus30) using(exitus30)
left join
```

```
(select exitus30,count(subject_id) as leucos_null FROM `angelic-button-33
1918.dataset.TABLA_IRA_COMPLETA`
where leucos is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as plaquetas_null FROM `angelic-button
-331918.dataset.TABLA_IRA_COMPLETA`
where plaquetas is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as glucosa_null FROM `angelic-button-3
31918.dataset.TABLA_IRA_COMPLETA`
where glucosa is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as creat_null FROM `angelic-button-331
918.dataset.TABLA_IRA_COMPLETA`
where creat is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as bun_null FROM `angelic-button-33191
8.dataset.TABLA_IRA_COMPLETA`
where bun is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as albu_null FROM `angelic-button-3319
18.dataset.TABLA_IRA_COMPLETA`
where albu is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as bnp_null FROM `angelic-button-33191
8.dataset.TABLA_IRA_COMPLETA`
where bnp is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as ddimer_null FROM `angelic-button-33
1918.dataset.TABLA_IRA_COMPLETA`
where ddimer is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as pcr_null FROM `angelic-button-33191
8.dataset.TABLA_IRA_COMPLETA`
where pcr is null group by exitus30) using(exitus30)
left join
(select exitus30,count(subject_id) as lactato_null FROM `angelic-button-3
31918.dataset.TABLA_IRA_COMPLETA`
where lactato is null group by exitus30) using(exitus30);
```

9.3 Anexo III: código en R y Python.

Cargo librerías en R y Python.

```
setwd('C:/Users/Usuario/Documents/UOC_MASTER_BIOESTADISTICA/TFM/MEMORIA')
library(reticulate)
library(knitr)
knitr.table.format = "latex"
library(tidyverse)
library(factoextra)
library(caret)
library(pROC)
library(randomForest)
library(xgboost)
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import roc_auc_score
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.utils import set_random_seed
from keras.models import Model
import warnings
warnings.filterwarnings('ignore')
```

Datos perdidos del dataset.

```
## Calidad de la TABLA_IRA_COMPLETAsegún los casos nulos por variable##

calidad=read.csv("tabla_calidad_ira_completa.csv")
calidad[is.na(calidad)]=0
variables=calidad[,3:32]
variables=data.frame(t(variables))
colnames(variables)=c('Exitus','No exitus')
exitus_1=calidad[1,2]
exitus_0=calidad[2,2]
porc_exitus_1=round(100*variables[,1]/exitus_1,2)
porc_exitus_0=round(100*variables[,2]/exitus_0,2)



p=vector('double',30)
datos_exitus_1=vector('character',30)
datos_exitus_0=vector('character',30)
for (i in 1:30){
tabla=rbind(variables[i,],c(exitus_1,exitus_0)-variables[i,])
chi2=chisq.test(tabla)
p[i]=round(chi2$p.value,5)
```

```
datos_exitus_1[i]=paste(variables[i,1],'(',porc_exitus_1[i],')')
datos_exitus_0[i]=paste(variables[i,2],'(',porc_exitus_0[i],')')
}
tabla_calidad=data.frame(variables=substr(rownames(variables),1,nchar(row
names(variables))-5),
                         datos_exitus_1,datos_exitus_0,p)
colnames(tabla_calidad)=c('Variables','Exitus: n(%)','No exitus: n(%)', '
p')
tabla_calidad %>% kable(caption='Tabla 6. Valores nulos de los datos en c
ada variable.')
```

Cargo dataset e imputación de variables.

```
# Cargo dataset de IRA:
ira=read.csv('tabla_ira_completa.csv')
n=dim(ira)[1]
ira$gender=factor(ira$gender)
ira$exitus30=factor(ira$exitus30)
ira$dopa=factor(ira$dopa, levels=c('','DOPA'),labels=c(0,1))
ira$dobuta=factor(ira$dobuta, levels=c('','DOBU'),labels=c(0,1))
ira$adre=factor(ira$adre, levels=c('','EPIN'),labels=c(0,1))
ira$nora=factor(ira$nora, levels=c('','NORE'),labels=c(0,1))
ira$heparina=factor(ira$heparina, levels=c('','HEPA'),labels=c(0,1))
ira$dexa=factor(ira$dexa, levels=c('','DEX'),labels=c(0,1))
ira$predni=factor(ira$predni, levels=c('','METH'),labels=c(0,1))
ira$vm=factor(ira$vm,levels =
                c('','Non-invasive Ventilation','Invasive Ventilation'))
levels(ira$vm)=c('no VM','vmni','vmi')
ira=ira %>% mutate(bun_creat=bun/creat)
attach(ira)

# Imputación del peso:
pred_peso=na.omit(ira[,c(8,14, 16:34)])
mod_peso=lm(peso~.,data=pred_peso) # Regresion lineal peso~variables
resumen=summary(mod_peso)
r2=resumen$r.squared
mediana_peso=median(na.omit(peso))
ira$peso[is.na(ira$peso)]=median(na.omit(peso)) # imputación mediante la
mediana

# Correlacion de BUn
pred_bun=na.omit(ira[,c(8,14, 16:36)])
correlaciones=round(cor(pred_bun),2)
corr=data.frame(correlaciones[23,c(1,6,9,21,22)])
colnames(corr)='r (Pearson)'
plot(bun~creat,type='p', pch=21,bg='blue',cex=0.7,
     xlab='Creatinina', ylab='BUN',
     main='Figura 4. Relación creatinina-BUN')
text(10,150, paste('r = ',round(corr[5,1],2)))
```

```
corr %>% kable(caption='Tabla 7. Coeficiente de correlación respecto a la
variable BUN')

# Modelo predictor de bun
modelo_bun=lm(bun~creat+hb+diuresis+abe+edad+tam,data=pred_bun)
resumen_modelo_bun=summary(modelo_bun)
# Coeficientes
interc=resumen_modelo_bun$coefficients[1,1]
cre=resumen_modelo_bun$coefficients[2,1]
hemog=resumen_modelo_bun$coefficients[3,1]
excbase=resumen_modelo_bun$coefficients[4,1]
años=resumen_modelo_bun$coefficients[5,1]
ta=resumen_modelo_bun$coefficients[6,1]
prod_nit=interc+cre*creat+hemog*hb+excbase*abe+años*edad+ta*tam
# Imputación:
ira=ira %>% mutate (bun=ifelse(is.na(bun),prod_nit,bun))
```

Exploración del dataset.

```
# porcentaje de mujeres y hombres:
gender_f=round(100*prop.table(table(gender)),0)[1]
gender_m=round(100*prop.table(table(gender)),0)[2]
# porcentaje de exitus a los 30 días:
mortalidad=round(100*prop.table(table(exitus30)),0)[2]

# Sexo:
tabla_sexo_mort=addmargins(table(exitus30,gender))
mort_F=round(100*tabla_sexo_mort[2,1]/tabla_sexo_mort[3,1],1)
mort_M=round(100*tabla_sexo_mort[2,2]/tabla_sexo_mort[3,2],1)
chi2_gender=chisq.test(table(exitus30,gender))
pvalue=round(chi2_gender$p.value,2)

g1=ggplot(data=ira,aes(x=gender,fill=exitus30))+
  geom_bar()+
  guides(fill = guide_legend(title = "Exitus"))+
  labs(title='Figura 5.Sexo de los pacientes y mortalidad asociada.',
       x='', y='')+
  annotate('text',x=c('F','M'), y=c(140,140),
           label=c(paste('mort = ',mort_F,'%'),paste('mort = ',mort_M,'%')
))+
  annotate('text',x='F', y=1000,
            label=paste('Chi2. p value= ',pvalue))+
  theme_bw()+
  theme(plot.title = element_text(size=10))

g1

# peso-tam-fc-gcs-temp-diuresis

ira=ira %>% mutate(diuresis=diuresis/(peso*24))
dg=aggregate(cbind(peso,edad,tam,gcs,temp,diuresis)~exitus30,data=ira,FUN
```

```
='mean')

t_peso=round(t.test (ira$peso~ira$exitus30, alternative ='two.sided',
                     mu = 0, paired = F, var.equal = F, conf.level = 1-0.
05)$p.value,3)
t_edad=round(t.test (ira$edad~ira$exitus30, alternative = 'two.sided',
                     mu = 0, paired = F, var.equal = F, conf.level = 1-0.
05)$p.value,3)
t_tam=round(t.test (ira$tam~ira$exitus30, alternative = 'two.sided',
                    mu = 0, paired = F, var.equal = T, conf.level = 1-0.0
5)$p.value,3)
t_gcs=round(t.test (ira$gcs~ira$exitus30, alternative = 'two.sided',
                    mu = 0, paired = F, var.equal = F, conf.level = 1-0.0
5)$p.value,3)
t_temp=round(t.test (ira$temp~ira$exitus30, alternative = 'two.sided',
                mu = 0, paired = F, var.equal = F, conf.level = 1-0.05)$p.
value,3)
t_diuresis=round(t.test (ira$diuresis~ira$exitus30, alternative ='two.sid
ed',
                     mu = 0, paired = F, var.equal = F, conf.level = 1-0.
05)$p.value,3)
pvalor=data.frame(exitus30='p valor', peso=t_peso,edad=t_edad,
                  tam=t_edad,gcs=t_gcs,temp=t_temp,diuresis=t_diuresis)
tabla_dg=data.frame(rbind(dg,pvalor));colnames(tabla_dg)[1]='Exitus'

tabla_dg %>% kable(caption='Tabla 8.Datos generales y su relación con la
mortalidad.')

# Comparación valores Gasometría

ira=ira %>% mutate(pafi=po2/(fio2/100))

dgaso=aggregate(cbind(fio2,so2,ph,po2,pco2,hco3,abe,lactato)~exitus30,dat
a=ira,FUN='mean')

t_fio2=round(t.test (fio2~exitus30, alternative = 'two.sided',
                     mu = 0, paired = F, var.equal = T, conf.level = 1-0.
05)$p.value,3)
t_so2=round(t.test (so2~exitus30, alternative = 'two.sided',
                     mu = 0, paired = F, var.equal = F, conf.level = 1-0.
05)$p.value,3)
t_ph=round(t.test (ph~exitus30, alternative = 'two.sided',
                    mu = 0, paired = F, var.equal = T, conf.level = 1-0.0
5)$p.value,3)
t_po2=round(t.test (po2~exitus30, alternative = 'two.sided',
                    mu = 0, paired = F, var.equal = F, conf.level = 1-0.0
5)$p.value,3)
t_pco2=round(t.test (pco2~exitus30, alternative = 'two.sided',
                     mu = 0, paired = F, var.equal = F, conf.level = 1-0.
05)$p.value,3)
```

```
t_hco3=round(t.test (hco3~exitus30, alternative = 'two.sided',
                     mu = 0, paired = F, var.equal = F, conf.level = 1-0.
05)$p.value,3)
t_abe=round(t.test (abe~exitus30, alternative = 'two.sided',
                     mu = 0, paired = F, var.equal = F, conf.level = 1-0.
05)$p.value,3)
t_lactato=round(t.test (lactato~exitus30, alternative = 'two.sided',
                     mu = 0, paired = F, var.equal = F, conf.level = 1-0.
05)$p.value,3)

pvalor_gaso=data.frame(exitus30='p valor', fio2=t_fio2,so2=t_so2,
                   ph=t_ph,po2=t_po2,pco2=t_pco2,hco3=t_hco3,abe=t_abe,
                   lactato=t_lactato)

redondear=function(x){round(x,2)}
dgaso_valores=apply(dgaso[,-1],2,redondear)# redondeo decimales
dgaso[1,-1]=dgaso_valores[1,]
dgaso[2,-1]=dgaso_valores[2,]

tabla_gaso=data.frame(rbind(dgaso,pvalor_gaso))

colnames(tabla_gaso)=c('Exitus','FiO2','SaO2','pH','PaO2','PaCO2','HCO3',
'ABE', 'Lactato')
tabla_gaso %>% kable(caption='Tabla 9. Gasometría arterial (media) y su r
elación con la mortalidad')

# Exploración hemograma.
dhemog=aggregate(cbind(hb,hto,leucos,plaquetas)~exitus30,data=ira,FUN='me
an')

t_hb=round(t.test (hb~exitus30, alternative = 'two.sided',
                     mu = 0, paired = F, var.equal = T, conf.level = 1-0.
05)$p.value,3)
t_hto=round(t.test (hto~exitus30, alternative = 'two.sided',
                    mu = 0, paired = F, var.equal = F, conf.level = 1-0.0
5)$p.value,3)
t_leucos=round(t.test (leucos~exitus30, alternative = 'two.sided',
                   mu = 0, paired = F, var.equal = T, conf.level = 1-0.05
)$p.value,3)
t_plaquetas=round(t.test (plaquetas~exitus30, alternative ='two.sided',
                    mu = 0, paired = F, var.equal = T, conf.level = 1-0.0
5)$p.value,3)

pvalor_hemog=data.frame(exitus30='p valor', hb=t_hb,hto=t_hto,
                      leucos=t_leucos,plaquetas=t_plaquetas)

redondear=function(x){round(x,2)}
dhemog_valores=apply(dhemog[,-1],2,redondear)# redondeo decimales
dhemog[1,-1]=dhemog_valores[1,]
dhemog[2,-1]=dhemog_valores[2,]
```

```
tabla_hemog=data.frame(rbind(dhemog,pvalor_hemog))

colnames(tabla_hemog)=c('Exitus','Hb','Hto','Leucocitos','Plaquetas')
tabla_hemog %>% kable(caption='Tabla 10. Hemograma (media) y su relación
con la mortalidad')

# Bioquímica y parámetros inflamatorios:
n_ddimer=n-sum(is.na(ddimer))
n_pcr=n-sum(is.na(pcr))
n_albu=n-sum(is.na(albu))
n_bnp=n-sum(is.na(bnp))

dbq=aggregate(cbind(glucosa,bun,creat,albu,ddimer,pcr)~exitus30,data=ira,
FUN='mean')
agreg_bnp=aggregate(bnp~exitus30,data=ira,FUN='mean')[,2]
dbq=cbind(dbq,bnp=agreg_bnp)

t_glucosa=round(t.test (glucosa~exitus30, alternative = 'two.sided',
                    mu = 0, paired = F, var.equal = F, conf.level = 1-0.05
)$p.value,3)
t_bun=round(t.test (bun~exitus30, alternative = 'two.sided',
                     mu = 0, paired = F, var.equal = F, conf.level = 1-0.0
5)$p.value,3)
t_creat=round(t.test (creat~exitus30, alternative = 'two.sided',
                       mu = 0, paired = F, var.equal = F, conf.level = 1-
0.05)$p.value,3)
t_albu=round(t.test (albu~exitus30, alternative ='two.sided',
                     mu = 0, paired = F, var.equal = T, conf.level = 1-0.
05)$p.value,3)
t_ddimer=round(t.test (ddimer~exitus30, alternative ='two.sided',
                     mu = 0, paired = F, var.equal = T, conf.level = 1-0.
05)$p.value,3)
t_pcr=round(t.test (pcr~exitus30, alternative ='two.sided',
                     mu = 0, paired = F, var.equal = T, conf.level = 1-0.
05)$p.value,3)
t_bnp=round(t.test (bnp~exitus30, alternative ='two.sided',
                    mu = 0, paired = F, var.equal = T, conf.level = 1-0.0
5)$p.value,3)
pvalor_bq=data.frame(exitus30='p valor', glucosa=t_glucosa,bun=t_bun,
                         creat=t_creat,albu=t_albu,ddimer=t_ddimer,pcr=t_p
cr,
                     bnp=t_bnp)

redondear=function(x){round(x,2)}
dbq_valores=apply(dbq[,-1],2,redondear)# redondeo decimales
dbq[1,-1]=dbq_valores[1,]
dbq[2,-1]=dbq_valores[2,]

tabla_bq=data.frame(rbind(dbq,pvalor_bq))
```

```
colnames(tabla_bq)=c('Exitus','Glucosa','BUN','Creatinina',
                     'Albúmina','D-Dímero','PCR','BNP')

tabla_bq %>% kable(caption='Tabla 11. Bioquímica e inflamación (media) y
su relación con la mortalidad')

# dopa
t_dopa=addmargins((table(exitus30,dopa)))
exitusNo_dopa=paste(t_dopa[1,2],'(',round(100*t_dopa[1,2]/t_dopa[1,3],1),
')')
exitusSi_dopa=paste(t_dopa[2,2],'(',round(100*t_dopa[2,2]/t_dopa[2,3],1),
')')
p_value_dopa=round(chisq.test(table(exitus30,dopa))$p.value,3)
# dobuta
t_dobuta=addmargins((table(exitus30,dobuta)))
exitusNo_dobuta=paste(t_dobuta[1,2],'(',round(100*t_dobuta[1,2]/t_dobuta[
1,3],1),')')
exitusSi_dobuta=paste(t_dobuta[2,2],'(',round(100*t_dobuta[2,2]/t_dobuta[
2,3],1),')')
p_value_dobuta=round(chisq.test(table(exitus30,dobuta))$p.value,3)
# nora
t_nora=addmargins((table(exitus30,nora)))
exitusNo_nora=paste(t_nora[1,2],'(',round(100*t_nora[1,2]/t_nora[1,3],1),
')')
exitusSi_nora=paste(t_nora[2,2],'(',round(100*t_nora[2,2]/t_nora[2,3],1),
')')
p_value_nora=round(chisq.test(table(exitus30,nora))$p.value,3)
# adrenalina
t_adre=addmargins((table(exitus30,adre)))
exitusNo_adre=paste(t_adre[1,2],'(',round(100*t_adre[1,2]/t_adre[1,3],1),
')')
exitusSi_adre=paste(t_adre[2,2],'(',round(100*t_adre[2,2]/t_adre[2,3],1),
')')
p_value_adre=round(fisher.test(table(exitus30,adre))$p.value,3)

# tabla resumen.
Exitus=c(0,1,'p valor')
Dopa=c(exitusNo_dopa,exitusSi_dopa,p_value_dopa)
Dobuta=c(exitusNo_dobuta,exitusSi_dobuta,p_value_dobuta)
Nora=c(exitusNo_nora,exitusSi_nora,p_value_nora)
Adre=c(exitusNo_adre,exitusSi_adre,p_value_adre)

tabla_drogas=data.frame(Exitus,Dopa,Dobuta,Nora,Adre)
colnames(tabla_drogas)=c('Exitus','Dopa: n(%)', 'Dobuta:n(%)','Nora:n(%)'
, 'Adrena:n(%)')

tabla_drogas %>% kable(caption='Tabla 12. Relación drogas vasoactivas y m
ortalidad')
```

```
# heparina, dexa y predni

# heparina
t_heparina=addmargins((table(exitus30,heparina)))
exitusNo_heparina=paste(t_heparina[1,2],'(',round(100*t_heparina[1,2]/t_h
eparina[1,3],1),')')
exitusSi_heparina=paste(t_heparina[2,2],'(',round(100*t_heparina[2,2]/t_h
eparina[2,3],1),')')
p_value_heparina=round(chisq.test(table(exitus30,heparina))$p.value,3)
# dexa
t_dexa=addmargins((table(exitus30,dexa)))
exitusNo_dexa=paste(t_dexa[1,2],'(',round(100*t_dexa[1,2]/t_dexa[1,3],1),
')')
exitusSi_dexa=paste(t_dexa[2,2],'(',round(100*t_dexa[2,2]/t_dexa[2,3],1),
')')
p_value_dexa=round(chisq.test(table(exitus30,dexa))$p.value,3)
# predni
t_predni=addmargins((table(exitus30,predni)))
exitusNo_predni=paste(t_predni[1,2],'(',round(100*t_predni[1,2]/t_predni[
1,3],1),')')
exitusSi_predni=paste(t_predni[2,2],'(',round(100*t_predni[2,2]/t_predni[
2,3],1),')')
p_value_predni=round(chisq.test(table(exitus30,predni))$p.value,3)
# tabla resumen.
Exitus=c(0,1,'p valor')
hepa=c(exitusNo_heparina,exitusSi_heparina,p_value_heparina)
dexamet=c(exitusNo_dexa,exitusSi_dexa,p_value_dexa)
prednisona=c(exitusNo_predni,exitusSi_predni,p_value_predni)

tabla_medicacion=data.frame(Exitus,hepa,dexamet,prednisona)
colnames(tabla_medicacion)=c('Exitus','Heparina: n(%)', 'Dexametasona:n(%
)','Metilprednisolona:n(%)')
tabla_medicacion %>% kable(caption='Tabla 13. Heparina y corticoides, rel
ación con la mortalidad')

# Exploración ventilación mecánica:

# unifico vmni+vmi en una única categoría
ira=ira %>% mutate(ventil=ifelse(vm=='no VM','no VM','VM'))
attach(ira)
tabla_ventil_mort=addmargins(table(exitus30,ventil))
mort_no=round(100*tabla_ventil_mort[2,1]/tabla_ventil_mort[3,1],1)
mort_VM=round(100*tabla_ventil_mort[2,2]/tabla_ventil_mort[3,2],1)
chi2_ventil=chisq.test(table(exitus30,ventil))
pvalue=round(chi2_gender$p.value,2)
g2=ggplot(data=ira,aes(x=ventil,fill=exitus30))+
  geom_bar()+
  guides(fill = guide_legend(title = "Exitus"))+
  labs(title='Figura 6.Relación entre mortalidad y ventilación mecánica.'
,
```

```
       x='', y='')+
  annotate('text',x=c('no VM','VM'), y=c(140,80),
           label=c(paste('mort = ',mort_no,'%'),paste('mort = ',mort_VM,'
%')))+
  annotate('text',x='VM', y=700,
           label=paste('Chi2. p value= ',pvalue))+
  theme_bw()+
  theme(plot.title = element_text(size=10))

# porcentaje de ventilados:
porc_vm=round(100*addmargins(table(ventil))[2]/addmargins(table(ventil))[
3],1)

# porcentaje de ventilados con vmi y vmni (dentro del grupo de ventilados
)
tabla_vm=ira %>% filter(vm!='no VM') %>% group_by(vm) %>% count()
porc_vmni=tabla_vm[1,2]/(tabla_vm[1,2]+tabla_vm[2,2])
porc_vmi=round(100*tabla_vm[2,2]/(tabla_vm[1,2]+tabla_vm[2,2]),1)

g2

# Oasis-ventilacion

p_ventil=t.test (oasis~ventil, alternative = 'two.sided',
                 mu = 0, paired = F, var.equal = T, conf.level = 1-0.05)$
p.value

ggplot (data=ira, aes(x=ventil,y=oasis,fill=ventil))+
  geom_boxplot()+
  labs(title='Figura 7. Relación OASIS score y VM',
       x='',y='OASIS score')+
  theme_bw()+
  theme(plot.title = element_text(size=10))+
  guides(fill = guide_legend(title = "Ventilación"))+
  annotate('text',x=1.5, y=55,
           label=paste('p value',ifelse(p_ventil<0.05,'<0.05','>0.05')))
```

Scores OASIS y SAPS II.

```
# Oasis-mortalidad

p_oasis=t.test (oasis~exitus30, alternative = 'two.sided',
                mu = 0, paired = F, var.equal = T, conf.level = 1-0.05)$p
.value

ggplot (data=ira, aes(x=exitus30,y=oasis,fill=exitus30))+
  geom_boxplot()+
  labs(title='Figura 8. Relación OASIS score y mortalidad',
       x='Exitus',y='OASIS score')+
  theme_bw()+
```

```
  theme(plot.title = element_text(size=10))+
  guides(fill = guide_legend(title = "Exitus"))+
  annotate('text',x=1.5, y=55,
           label=paste('p value',ifelse(p_oasis<0.05,'<0.05','>0.05')))

# sapsii-mortalidad

p_saps=t.test (sapsii~exitus30, alternative = 'two.sided',
               mu = 0, paired = F, var.equal = T, conf.level = 1-0.05)$p
.value

ggplot (data=ira, aes(x=exitus30,y=sapsii,fill=exitus30))+
  geom_boxplot()+
  labs(title='Figura 9. Relación SAPS II score y mortalidad',
       x='Exitus',y='SAPS II score')+
  theme_bw()+
  theme(plot.title = element_text(size=10))+
  guides(fill = guide_legend(title = "Exitus"))+
  annotate('text',x=1.5, y=65,
           label=paste('p value',ifelse(p_oasis<0.05,'<0.05','>0.05')))

saps_oasis_cor=round(cor(ira$oasis,ira$sapsii),2)

ggplot(data=ira,aes(x=oasis,y=sapsii,color=exitus30))+
  geom_jitter(alpha=0.5)+
  labs(title='Figura 10. Correlación SAPS II ~ OASIS',
       x='OASIS',y='SAPS II')+
  theme_bw()+
  theme(plot.title = element_text(size=10))+
  guides(color = guide_legend(title = "Exitus"))

# Rendimiento de OASIS y SAPS II en el total de pacientes MIMIC III

pacientes=read.csv("~/UOC_MASTER_BIOESTADISTICA/TFM/mimic/pacientes_oasis
_saps.csv")
pacientes$exitus=factor(pacientes$exitus)
obj.roc_oasis=roc(pacientes$exitus,pacientes$oasis_prob,levels=c('0','1')
)
obj.roc_sapsii=roc(pacientes$exitus,pacientes$sapsii_prob,levels=c('0','1
'))
plot(obj.roc_oasis,legacy.axes=T,col='red',print.auc=F)
plot(obj.roc_sapsii,legacy.axes=T,col='blue',print.auc=F, add=T)
```

Comorbilidad.

```
## items de comorbilidad relacionados con mortalidad.
comorbilidad=ira[,c(66:95)]

df_comorbilidad=data.frame(comorbilidad,exitus=ira$exitus30)
df_comorbilidad$exitus=ifelse(df_comorbilidad$exitus=='1','muere','vive')
```

```
pvalue=vector('double',30)
for (i in 1:30){
pvalue[i]=fisher.test(df_comorbilidad[,i],
                      df_comorbilidad$exitus)$p.value
}
resumen=data.frame(item=colnames(comorbilidad)[-31],p_valor=pvalue)
comorb_sign=resumen %>% filter(p_valor<=0.05) %>% head(9)
comorb_sign %>% kable(caption='Tabla 14. Items de comorbilidad significat
ivos.')
df_comorb_sign=df_comorbilidad[,comorb_sign$item]
df_comorb_sign$exitus=df_comorbilidad$exitus
```

Modelos de predicción mortalidad a 30 días:

```
datos=ira[,c('edad','peso','tam',
             'diuresis','fio2','so2','pco2','hco3','lactato',
             'hb','leucos','plaquetas','bun','dopa','nora',
             'heparina','exitus30')]
colnames(datos)[17]='exitus'
# añado comorbilidd :
df=data.frame(datos,df_comorb_sign[,-10]) # añado items comorbilidad sign
ificativos
n_predictores=dim(df)[2]-1

# Dumifico variables:
df=data.frame(model.matrix(exitus~.,data=df))[,-1]
df$exitus=datos$exitus

# train y test split (70%-30%).
set.seed(1234)
row_train=createDataPartition(df$exitus,p=0.7,list=F)

train=df[row_train,]
test=df[-row_train,]
train$exitus=factor(ifelse(train$exitus=='1','muere','vive'))
test$exitus=factor(ifelse(test$exitus=='1','muere','vive'))

dim_train=dim(train)
dim_test=dim(test)

metodo=c('knn','glm','nb','svmLinear','svmRadial','rf')
n_metodos=length(metodo)
Roc=vector('double',n_metodos)
control_ajuste=trainControl(classProbs = T,summaryFunction = twoClassSumm
ary)
for (i in 1:n_metodos){
  set.seed(1234)
  modeloML=train(exitus~.,data=train,
                 method=metodo[i],
                 trControl=control_ajuste,
                 preProcess=c('range'),
```

```
                  metric='ROC')
  Roc[i]=round(max(modeloML$results[,'ROC']),3)
}

resumen_modelos=data.frame(Modelo=metodo,ROC=Roc)

resumen_modelos %>% kable(caption='Tabla 15. Rentabilidad de diferentes a
lgoritmos de ML.')

# --- Optimización modelo Regresión Logística---
control_ajuste=trainControl(classProbs = T,
                            summaryFunction = twoClassSummary,
                            method='cv',n=5)
set.seed(1234)
modeloRL=train(exitus~.
               ,data=train,
               method='glmStepAIC',
               trControl=control_ajuste,
               preProcess=c('range'),
               metric='ROC',
               verbosity=0)

auc_train_rl=round(max(modeloRL$results$ROC),3)

# --- Optimización modelo Random Forest ---

hiperparametros=expand.grid(mtry=c(2:26))
set.seed(1234)
modeloRF=train(
  exitus ~., data = train, method = "rf",
  trControl = control_ajuste,
  tuneGrid=hiperparametros,
  metric='ROC',
  preProcess=c('range'),
  ntrees=500,
  verbosity=0)

auc_train_rf=round(max(modeloRF$results[,'ROC']),3)

# --- Optimización modelo RF mediante boosting (xgbTree) ----
set.seed(1234)
modeloRF_boost=train(
  exitus ~., data = train, method = "xgbTree",
  trControl = control_ajuste,
  tuneLengh=100,
  metric='ROC',
  preProcess=c('range'),
  verbosity=0)
auc_train_boost=round(max(modeloRF_boost$results[,'ROC']),3)
```

Red Neuronal profunda.

```
train=r.train
test=r.test
variables=train.columns
# Separo train y test en: X e y
X_train=train.iloc[0:,0:-1]
X_test=test.iloc[0:,0:-1]
y_train=train.iloc[0:,-1]
y_test=test.iloc[0:,-1]
y_train=pd.DataFrame(y_train)
y_test=pd.DataFrame(y_test)
n_predictores=X_train.shape[1]

# Estandarizo train y test según formato max-min:
minmax=MinMaxScaler()
X_train=pd.DataFrame(minmax.fit_transform(X_train))
X_test=pd.DataFrame(minmax.transform(X_test))
X_train.columns=variables[:-1]
X_test.columns=variables[:-1]

# Estandarizo train y test según formato standardScale:
norm=StandardScaler()
X_train_norm=pd.DataFrame(norm.fit_transform(X_train))
X_test_norm=pd.DataFrame(norm.transform(X_test))
X_train_norm.columns=variables[:-1]
X_test_norm.columns=variables[:-1]

# codifico one-hot exitus:
preproc= OneHotEncoder(handle_unknown='ignore')
y_train=pd.DataFrame(preproc.fit_transform(y_train[['exitus']]).toarray()
)
y_test=pd.DataFrame(preproc.fit_transform(y_test[['exitus']]).toarray())
y_train.columns=['vive','muere']
y_test.columns=['vive','muere']

# grupo de validación del 20% del train total.
X_tr,X_val=X_train[:1200],X_train[1200:]
y_tr,y_val=y_train[:1200],y_train[1200:]

pesos={0:1,1:4}
# Red neuronal Secuencial:
def modeloNN(capas=1,neuronas=300,optimizador='Adam'):
  modeloNN=keras.models.Sequential()
  modeloNN.add(keras.layers.InputLayer(input_shape=[n_predictores])) # Ca
pa de entrada
  for capa in range(capas):
    modeloNN.add(keras.layers.Dense(neuronas,activation='relu')) # Capa d
ensa de 125 nodos
  modeloNN.add(keras.layers.Dense(2,activation='sigmoid')) # Capa de sali
```

```
da de 2 nodos
  modeloNN.compile(loss='binary_crossentropy',metrics='AUC',
  optimizer=optimizador)
  # Ajusto el modelo:
  epocas=50;lote=32
  historial=modeloNN.fit(X_tr,y_tr,validation_data=(X_val,y_val),
  epochs=epocas,batch_size=lote,class_weight=pesos,verbose=0)
  # Evalúo el modelo mediante sklearn-auc:
  prob_val=modeloNN.predict(X_val)
  prob_test=modeloNN.predict(X_test)
  auc=roc_auc_score(y_val,prob_val)
  return auc,historial,prob_test # devuelve AUC en el grupo de validación
, historial de aprendizaje y probabilidades en el grupo test.

rep=100 # número de repeticiones de los modelos de red neural

# Optimización Red Neuronal Secuencial: OPTIMIZADORES.
rend_Adam=[]
for _ in range(rep):
  modelo=modeloNN(capas=3,neuronas=125,optimizador='Adam')
  rend_Adam.append(modelo[0])

rend_SGD=[]
for _ in range(rep):
  modelo=modeloNN(capas=3,neuronas=125,optimizador='sgd')
  rend_SGD.append(modelo[0])

rend_Adagrad=[]
for _ in range(rep):
  modelo=modeloNN(capas=3,neuronas=125,optimizador='Adagrad')
  rend_Adagrad.append(modelo[0])


rend_Adamax=[]
for _ in range(rep):
  modelo=modeloNN(capas=3,neuronas=125,optimizador='Adamax')
  rend_Adamax.append(modelo[0])


rend_Nadam=[]
for _ in range(rep):
  modelo=modeloNN(capas=3,neuronas=125,optimizador='Nadam')
  rend_Nadam.append(modelo[0])


rend_Ftrl=[]
for _ in range(rep):
  modelo=modeloNN(capas=1,neuronas=125,optimizador='Ftrl')
  rend_Ftrl.append(modelo[0])
```

```
rend_RMSprop=[]
for _ in range(rep):
  modelo=modeloNN(capas=1,neuronas=125,optimizador='RMSprop')
  rend_RMSprop.append(modelo[0])


opt=pd.DataFrame({'Adam':rend_Adam,'SGD':rend_SGD,'Adagrad':rend_Adagrad,
'Adamax':rend_Adamax,'Nadam':rend_Nadam, 'Ftrl':rend_Ftrl,'RMSprop':rend_
RMSprop})
opt.plot(kind='box',figsize=(5,5))
plt.title ('Figura 11. Comparación de los diferentes optimizadores.')
plt.ylabel('AUC')
plt.show()

# Optimización Red Neuronal Secuencial: CAPAS OCULTAS.

rend_D1=[]
for _ in range(rep):
  modelo=modeloNN(capas=1,neuronas=125,optimizador='Adam')
  rend_D1.append(modelo[0])

rend_D2=[]
for _ in range(rep):
  modelo=modeloNN(capas=2,neuronas=125,optimizador='Adam')
  rend_D2.append(modelo[0])

rend_D3=[]
for _ in range(rep):
  modelo=modeloNN(capas=3,neuronas=125,optimizador='Adam')
  rend_D3.append(modelo[0])

rend_D4=[]
for _ in range(rep):
  modelo=modeloNN(capas=4,neuronas=125,optimizador='Adam')
  rend_D4.append(modelo[0])



opt=pd.DataFrame({'D1':rend_D1,'D2':rend_D2,'D3':rend_D3,'D4':rend_D4})
opt.plot(kind='box',figsize=(5,5))
plt.title ('Figura 12. Comparación de diferente número de capas densas.')
plt.ylabel('AUC')
plt.show()

# Optimización Red Neuronal Secuencial: NEURONAS POR CAPA.
rend_N125=[]
for _ in range(rep):
```

```
  modelo=modeloNN(capas=1,neuronas=125,optimizador='Adam')
  rend_N125.append(modelo[0])

rend_N200=[]
for _ in range(rep):
  modelo=modeloNN(capas=1,neuronas=200,optimizador='Adam')
  rend_N200.append(modelo[0])

rend_N300=[]
for _ in range(rep):
  modelo=modeloNN(capas=1,neuronas=300,optimizador='Adam')
  rend_N300.append(modelo[0])

opt=pd.DataFrame({'N125':rend_N125,'N200':rend_N200,'N300':rend_N300})
opt.plot(kind='box',figsize=(5,5))
plt.title ('Figura 13. Comparación entre diferente número de neuronas por
capa.')
plt.ylabel('AUC')
plt.show()

# Red neuronal Secuencial:
def modeloNN_1(activacion='relu',kernel_ini='he_normal'):
  modeloNN=keras.models.Sequential()
  modeloNN.add(keras.layers.InputLayer(input_shape=[n_predictores])) # Ca
pa de entrada
  for capa in range(1):
    modeloNN.add(keras.layers.Dense(200,kernel_initializer=kernel_ini,act
ivation=activacion)) # Capa densa de 125 nodos
  modeloNN.add(keras.layers.Dense(2,activation='sigmoid')) # Capa de sali
da de 2 nodos
  modeloNN.compile(loss='binary_crossentropy',metrics='AUC',
  optimizer='RMSprop')
  # Ajusto el modelo:
  epocas=50;lote=32
  historial=modeloNN.fit(X_tr,y_tr,validation_data=(X_val,y_val),
  epochs=epocas,batch_size=lote,class_weight=pesos,verbose=0)
  # Evalúo el modelo mediante sklearn-auc:
  prob_train=modeloNN.predict(X_val)
  prob_test=modeloNN.predict(X_test)
  auc=roc_auc_score(y_val,prob_train)
  return auc,historial,prob_test # devuelve AUC en el grupo de validación
, historial de aprendizaje y probabilidades en el grupo test.

iniz=[]
for _ in range(rep):
  modelo=modeloNN_1(activacion='relu',kernel_ini='he_normal')
  iniz.append(modelo[0])

selu=[]
for _ in range(rep):
```

```
  modelo=modeloNN_1(activacion='selu',kernel_ini='lecun_normal')
  selu.append(modelo[0])


opt_normal=pd.DataFrame({'RELU normal':iniz,'SELU normal':selu})
opt_normal.plot(kind='box',figsize=(5,5))
plt.title ('Figura 14. Inicialización normal de pesos.')
plt.ylabel('AUC')
plt.show()

auc_val_rnp=round(np.mean(selu),2)

# Redimensiono en matriz de 5x5
X_tr_m=np.asarray(X_tr).reshape(len(X_tr),5,5,1)
X_val_m=np.asarray(X_val).reshape(len(X_val),5,5,1)
X_test_m=np.asarray(X_test).reshape(len(X_test),5,5,1)

# Ejemplo de individuo para  y=0 e y=1:
fig,base=plt.subplots()
base.imshow(X_tr_m[0],cmap='gray') # y=0, FIGURA 14
plt.title('Figura 15. Mapa de bits de un pacientes exitus=0')
plt.show()

#Diseño CNN

def model_cnn(n_kernels=32,neuronas=50,optimizador='Adam'):
  model=keras.models.Sequential()
  model.add(keras.layers.Conv2D(n_kernels,kernel_size=(2,2),activation='r
elu',strides=1,input_shape=(5,5,1)))
  model.add(keras.layers.MaxPool2D(pool_size=(2,2)))
  model.add(keras.layers.Flatten())
  model.add(keras.layers.Dense(neuronas,activation='relu'))
  model.add(keras.layers.Dense(2,activation='softmax'))
  model.compile(loss='mean_squared_error',optimizer=optimizador,metrics='
AUC')
  # Ajusto el modelo:
  epocas=50;lote=32
  historial=model.fit(X_tr_m,y_tr,validation_data=(X_val_m,y_val),
  epochs=epocas,batch_size=lote,verbose=0)
  # Evalúo el modelo mediante sklearn-auc:
  prob_train=model.predict(X_val_m)
  prob_test=model.predict(X_test_m)
  auc=roc_auc_score(y_val,prob_train)
  return auc,historial,prob_test # devuelve AUC en el grupo de validación
, historial de aprendizaje y probabilidades en el grupo test.

# Optimización Red Neuronal Secuencial: OPTIMIZADORES.
rend_Adam=[]
for _ in range(rep):
  modelo=model_cnn(n_kernels=32,neuronas=50,optimizador='Adam')
```

```
  rend_Adam.append(modelo[0])

rend_SGD=[]
for _ in range(rep):
  modelo=model_cnn(n_kernels=32,neuronas=50,optimizador='sgd')
  rend_SGD.append(modelo[0])

rend_Adagrad=[]
for _ in range(rep):
  modelo=model_cnn(n_kernels=32,neuronas=50,optimizador='Adagrad')
  rend_Adagrad.append(modelo[0])


rend_Adamax=[]
for _ in range(rep):
  modelo=model_cnn(n_kernels=32,neuronas=50,optimizador='Adamax')
  rend_Adamax.append(modelo[0])


rend_Nadam=[]
for _ in range(rep):
  modelo=model_cnn(n_kernels=32,neuronas=50,optimizador='Nadam')
  rend_Nadam.append(modelo[0])


rend_Ftrl=[]
for _ in range(rep):
  modelo=model_cnn(n_kernels=32,neuronas=50,optimizador='Ftrl')
  rend_Ftrl.append(modelo[0])


rend_RMSprop=[]
for _ in range(rep):
  modelo=model_cnn(n_kernels=32,neuronas=50,optimizador='RMSprop')
  rend_RMSprop.append(modelo[0])

opt=pd.DataFrame({'Adam':rend_Adam,'SGD':rend_SGD,'Adagrad':rend_Adagrad,
'Adamax':rend_Adamax,'Nadam':rend_Nadam, 'Ftrl':rend_Ftrl,'RMSprop':rend_
RMSprop})
opt.plot(kind='box',figsize=(5,5))
plt.title ('Figura 16. Comparación de los diferentes optimizadores.')
plt.ylabel('AUC')
plt.show()

# Optimización Red Neuronal Secuencial: número de kernels.
rend_k32=[]
for _ in range(rep):
  modelo=model_cnn(n_kernels=32,neuronas=50,optimizador='Adam')
  rend_k32.append(modelo[0])
```

```
rend_k64=[]
for _ in range(rep):
  modelo=model_cnn(n_kernels=32,neuronas=50,optimizador='Adam')
  rend_k64.append(modelo[0])

rend_k128=[]
for _ in range(rep):
  modelo=model_cnn(n_kernels=32,neuronas=50,optimizador='Adam')
  rend_k128.append(modelo[0])

opt=pd.DataFrame({'k32':rend_k32,'k64':rend_k64,'k128':rend_k128})
opt.plot(kind='box',figsize=(5,5))
plt.title ('Figura 17. Comparación de los diferentes número de kernels.')
plt.ylabel('AUC')
plt.show()

# Optimización Red Neuronal Secuencial: número de neuronas capa densa.
rend_N50=[]
for _ in range(rep):
  modelo=model_cnn(n_kernels=32,neuronas=50,optimizador='Adam')
  rend_N50.append(modelo[0])

rend_N100=[]
for _ in range(rep):
  modelo=model_cnn(n_kernels=32,neuronas=100,optimizador='Adam')
  rend_N100.append(modelo[0])

rend_N200=[]
for _ in range(rep):
  modelo=model_cnn(n_kernels=32,neuronas=200,optimizador='Adam')
  rend_N200.append(modelo[0])

opt=pd.DataFrame({'N50':rend_N50,'N100':rend_N100,'N200':rend_N200})
opt.plot(kind='box',figsize=(5,5))
plt.title ('Figura 18. Comparación  diferentes número de neuronas.')
plt.ylabel('AUC')
plt.show()

# Cambio inicialización de pesos a distribución Normal con activación rel
u/selu

# grupo de validación del 20% del train total.
X_tr_norm,X_val_norm=X_train_norm[:1200],X_train_norm[1200:]
# Redimensiono en matriz de 5x5
X_tr_norm_m=np.asarray(X_tr_norm).reshape(len(X_tr_norm),5,5,1)
X_val_norm_m=np.asarray(X_val_norm).reshape(len(X_val_norm),5,5,1)
X_test_norm_m=np.asarray(X_test_norm).reshape(len(X_test_norm),5,5,1)

#Diseño CNN:
```

```
def model_cnn_1(activacion='selu',kernel_ini='lecun_normal',Xtrain=X_tr_n
orm_m,Xval=X_val_norm_m,Xtest=X_test_norm_m):
  model=keras.models.Sequential()
  model.add(keras.layers.Conv2D(32,kernel_size=(2,2),activation=activacio
n,kernel_initializer=kernel_ini,strides=1,input_shape=(5,5,1)))
  model.add(keras.layers.MaxPool2D(pool_size=(2,2)))
  model.add(keras.layers.Flatten())
  model.add(keras.layers.Dense(100,activation=activacion,kernel_initializ
er=kernel_ini))
  model.add(keras.layers.Dense(2,activation='softmax'))
  model.compile(loss='mean_squared_error',optimizer='Adam',metrics='AUC')
  # Ajusto el modelo:
  epocas=50;lote=32
  historial=model.fit(Xtrain,y_tr,validation_data=(Xval,y_val),
  epochs=epocas,batch_size=lote,verbose=0)
  # Evalúo el modelo mediante sklearn-auc:
  prob_train=model.predict(Xval)
  prob_test=model.predict(Xtest)
  auc=roc_auc_score(y_val,prob_train)
  return auc,historial,prob_test # devuelve AUC en el grupo de validación
, historial de aprendizaje y probabilidades en el grupo test.

iniz_cnn=[]
for _ in range(rep):
  modelo=model_cnn_1(activacion='relu',kernel_ini='he_normal',Xtrain=X_tr
_m,Xval=X_val_m,Xtest=X_test_m)
  iniz_cnn.append(modelo[0])

selu_no_norm_cnn=[] # introduzco la entrada sin normalizar a media=0 y ds
=1
for _ in range(rep):
  modelo=model_cnn_1(activacion='selu',kernel_ini='lecun_normal',Xtrain=X
_tr_m,Xval=X_val_m,Xtest=X_test_m)
  selu_no_norm_cnn.append(modelo[0])


opt_cnn=pd.DataFrame({'RELU normal':iniz_cnn,'SELU normal':selu_no_norm_c
nn})
opt_cnn.plot(kind='box',figsize=(5,5))
plt.title ('Figura 19.Inicialización normal de pesos')
plt.ylabel('AUC')
plt.show()

auc_val_cnn=round(np.mean(selu_no_norm_cnn),2)

set_random_seed(1234)
result_modelo=model_cnn_1(activacion='selu',kernel_ini='lecun_normal',Xtr
ain=X_tr_m,Xval=X_val_m,Xtest=X_test_m)
historial_cnn=result_modelo[1]
```

```
prob_test=result_modelo[2];prob_test=prob_test[:,1]

# Curva del rendimiento (AUC):
fig,base=plt.subplots()
base.plot(historial_cnn.history['auc'])
base.plot(historial_cnn.history['val_auc'])
base.set_ylim(0.2,1)
base.set_title('Figura 20. Curva de entrenamiento del modelo.')
base.set_ylabel('AUC')
base.set_xlabel('Época')
base.legend(['train','validation'],loc='upper left')
plt.show()
```

Validación en el grupo test.

```
# Metricas en el grupo test del modelo CNN:
prob_test=c(py$prob_test)
y_test=py$y_test['muere']
colnames(y_test)='exitus'
y_test$exitus=factor(ifelse(y_test$exitus==1,'muere','vive'))
obj.roc_cnn=roc(y_test$exitus,prob_test,levels=c('vive','muere'))
asignacion_cnn=factor(ifelse(prob_test<0.789,'vive','muere'))
cm_cnn=confusionMatrix(asignacion_cnn,y_test$exitus,positive='muere')
auc_cnn=round(auc(obj.roc_cnn),2)

# Metricas en el grupo test del modelo RL:
pred_RL=predict(modeloRL,test, type='prob')[,1]
obj.roc_RL=roc(test$exitus,pred_RL,levels=c('vive','muere'))
asignacion_RL=factor(ifelse(pred_RL<0.194,'vive','muere'))
RL_cm=confusionMatrix(asignacion_RL,test$exitus,positive='muere')
auc_RL=round(auc(obj.roc_RL),2)

# Metricas en el grupo test del modelo RF:
pred_RF=predict(modeloRF,test, type='prob')[,1]
obj.roc_RF=roc(test$exitus,pred_RF,levels=c('vive','muere'))
asignacion_RF=factor(ifelse(pred_RF<0.225,'vive','muere'))
RF_cm=confusionMatrix(asignacion_RF,test$exitus,positive='muere')
auc_RF=round(auc(obj.roc_RF),2)

plot(obj.roc_cnn,legacy.axes=T,col='red',main='Figura 21. Rendimiento de
los modelos en el grupo test')
plot(obj.roc_RL,legacy.axes=T,col='blue', add=T)
plot(obj.roc_RF,legacy.axes=T,col='black',add=T)
text(c('RL','RF','CNN'), x=c(0.2,0.2,0.2),y=c(0.6,0.55,0.5), col=c('blue'
,'black','red'))
```

Comparación del mejor modelo con los scores SAPS II y OASIS.

```
### Regresion logistica SAPS II-MORTALIDAD 30 DÍAS:

df_saps=data.frame(saps=ira$sapsii,exitus=ira$exitus30)
```

```
df_saps$exitus=factor(ifelse(df_saps$exitus=='1','muere','vive'))
train_saps=df_saps[row_train,]
test_saps=df_saps[-row_train,]

model_saps=glm(exitus~saps,family=binomial,data=train_saps)
prob_saps=predict(model_saps,data.frame(saps=test_saps$saps),type='respon
se') # prob en test
obj.roc_saps=roc(test_saps$exitus,prob_saps,levels=c('vive','muere'))

asignacion_saps=factor(ifelse(prob_saps>0.827,'vive','muere'))
saps_cm=confusionMatrix(asignacion_saps,test_saps$exitus,positive='muere'
)

### Regresion Logistica OASIS-MORTALIDAD 30 DÍAS:
df_oasis=data.frame(oasis=ira$oasis,exitus=ira$exitus30)
df_oasis$exitus=factor(ifelse(df_oasis$exitus=='1','muere','vive'))
train_oasis=df_oasis[row_train,]
test_oasis=df_oasis[-row_train,]

model_oasis=glm(exitus~oasis,family=binomial,data=train_oasis)
prob_oasis=predict(model_oasis,data.frame(oasis=test_oasis$oasis),
                   type='response') # prob en test
obj.roc_oasis=roc(test_oasis$exitus,prob_oasis,levels=c('vive','muere'))

asignacion_oasis=factor(ifelse(prob_oasis>0.820,'vive','muere'))
oasis_cm=confusionMatrix(asignacion_oasis,test_oasis$exitus,positive='mue
re')


# --- Comparacion scores -----------------
plot(obj.roc_oasis,print.auc=F,legacy.axes=T,col='red', main='Figura 22.
Comparación rendimiento SAPS II-OASIS.')
plot(obj.roc_saps,print.auc=F,legacy.axes=T,col='blue',add=T)
text(c('SAPS II','OASIS'), x=c(0.2,0.2,0.2),y=c(0.6,0.55,0.5), col=c('blu
e','red'))

plot(obj.roc_oasis,print.auc=F,legacy.axes=T,col='red', main='Figura 22.
Comparación rendimiento SAPS II-OASIS-Modelo 25v')
plot(obj.roc_saps,print.auc=F,legacy.axes=T,col='blue',add=T)
plot(obj.roc_RL,print.auc=F,legacy.axes=T,col='black',add=T)
text(c('SAPS II','OASIS', 'Modelo 25V'), x=c(1.3,1.3,1.3),y=c(0.8,0.7,0.6
), col=c('blue','red','black'))

scores=c('OASIS', 'SAPS II', 'Modelo 25V')
auc=round(c(auc(obj.roc_oasis),auc(obj.roc_saps),auc(obj.roc_RL)),2)
k=round(c(oasis_cm$overall[2],saps_cm$overall[2],RL_cm$overall[2]),2)
s=round(c(oasis_cm$byClass[1],saps_cm$byClass[1],RL_cm$byClass[1]),2)
e=round(c(oasis_cm$byClass[2],saps_cm$byClass[2],RL_cm$byClass[2]),2)
vpp=round(c(oasis_cm$byClass[3],saps_cm$byClass[3],RL_cm$byClass[3]),2)
vpn=round(c(oasis_cm$byClass[4],saps_cm$byClass[4],RL_cm$byClass[4]),2)
```

```
tabla_final=data.frame(PREDICCION=scores,
                       AUC=auc,
                       Kappa=k,
                       SENS=s,
                       ESP=e,
                       VPP=vpp,
                       VPN=vpn)

tabla_final %>% kable(caption='Tabla 16. Métricas de los modelos.')
```

10. BIBLIOGRAFÍA:

1. Jhaveri R, John J, Rosenman M. Electronic health record network research in infectious diseases. Vol. 43, Clinical Therapeutics. Elsevier Inc.; 2021. p. 1668–81.

2. U.S. Food & Drug Administration (FDA). Real-world evidence [Internet]. Framework for FDA's Real World Evidence Program [Actualizado diciembre de 2018; consultado 12 de marzo de 2022]. Available from: https://www.fda.gov/media/120060/download

3. Collins R, Bowman L, Landray M, Peto R. The magic of randomization versus the myth of real-world evidence. New England Journal of Medicine. 2020;382.

4. Franklin JM, Schneeweiss S. When and how can real world data analyses substitute for randomized controlled trials? Clinical Pharmacology & Therapeutics. 2017;102(6):924–33.

5. Johnson A MR Pollard T. MIMIC-III clinical database (version 1.4) [Internet]. PhysioNet. [Consultado 10 de febrero de 2022]. Available from: https://doi.org/10.13026/C2XW26

6. Johnson AE, Kramer AA, Clifford GD. A new severity of illness scale using a subset of acute physiology and chronic health evaluation data elements shows comparable predictive accuracy. Critical care medicine. 2013;41(7):1711–8.

7. Le Gall J-R, Lemeshow S, Saulnier F. A new simplified acute physiology score (SAPS II) based on a european/north american multicenter study. Jama. 1993;270(24):2957–63.

8. Standardized electronic health record data modeling and persistence: A comparative review. Vol. 114, Journal of Biomedical Informatics. Academic Press Inc.; 2021.

9. Wang H, Belitskaya-Levy I, Wu F, Lee JS, Shih MC, Tsao PS, et al. A statistical quality assessment method for longitudinal observations in electronic health record data with an application to the VA million veteran program. BMC Medical Informatics and Decision Making. 2021;21.

10. Fda, Cder, Grandinetti, A C. Use of electronic health record data in clinical investigations guidance for industry [Internet]. U.S. Food & Drug Administration. Guidance Document. [Actualizado julio de 2018; consultado 3 de marzo de 2022]. Available from: https://www.fda.gov/regulatory-information/search-fda-guidance-documents/use-electronic-health-record-data-clinical-investigations-guidance-industry

11. Cresswell KM, Sheikh A. Inpatient clinical information systems. Key Advances in Clinical Informatics: Transforming Health Care through Health Information Technology. 2017.

12. Rajkomar A, Oren E, Chen K, Dai AM, Hajaj N, Hardt M, et al. Scalable and accurate deep learning with electronic health records. npj Digital Medicine. 2018 Dec;1.

13. Richard E. Gliklich, M.D., Michelle B. Leavy, M.P.H., Nancy A. Dreyer, Ph.D. Tools and technologies for registry interoperability, 2nd addendum of registries for evaluating patient outcomes, a user's guide, 3rd ed. Washington: AHRQ Publication No. 19(20)-EHC017-EF ;2021. 106 p.

14. Harrison JE, Weber S, Jakob R, Chute CG. ICD-11: An international classification of diseases for the twenty-first century. Vol. 21, BMC Medical Informatics and Decision Making. 2021.

15. Bentsen BG. International classification of primary care. Scandinavian Journal of Primary Health Care. 1986;4.

16. Diagnostic and statistical manual of mental disorders (DSM–5) [Internet]. American Psychiatric Association. [Consultado 15 de marzo 2022]. Available from: https://www.psychiatry.org/psychiatrists/practice/dsm

17. National drug code directory [Internet]. U.S. Food & Drug Administration. [Actualizado 10 de mayo de 2022. Consultado 6 de mayo de 2022]. Available from: https://www.fda.gov/drugs/drug-approvals-and-databases/national-drug-code-directory

18. C L. The unified medical language system (UMLS) of the national library of medicine. J Am Med Rec Assoc. 1990 May;61:40–2.

19. (WHO) WHO. The anatomical therapeutic chemical classification system with defined daily doses (ATC/DDD).

20. Current procedural terminology (CPT) [Internet]. American Medical Association [Consultado 20 marzo de 2022]. Available from: https://www.ama-assn.org/amaone/cpt-current-procedural-terminology

21. HCPCS- general Information [Internet]. Centers for Medicare and Medicaid Services [Consultado 15 marzo de 2022]. Available from: https://www.cms.gov/Medicare/Coding/MedHCPCSGenInfo

22. McDonald CJ, Huff SM, Suico JG, Hill G, Leavelle D, Aller R, et al. LOINC, a universal standard for identifying laboratory observations: A 5-year update. Clinical chemistry. 2003;49(4):624–33.

23. Edwards BN. The 21st century cures act: A patient's miracle or demise? Journal of the National Association of Administrative Law Judiciary. 2020;40(2):79–109.

24. Ramagopalan SV, Simpson A, Sammon C. Can real-world data really replace randomised clinical trials? Vol. 18, BMC Medicine. BioMed Central; 2020.

25. McNair D, Lumpkin M, Kern S, Hartman D. Use of RWE to inform regulatory, public health policy, and intervention priorities for the developing world. Clinical Pharmacology and Therapeutics. 2022 Jan;111:44–51.

26. Franklin JM, Liaw KL, Iyasu S, Critchlow CW, Dreyer NA. Real-world evidence to support regulatory decision making: New or expanded medical product indications. Vol. 30, Pharmacoepidemiology and Drug Safety. John Wiley; Sons Ltd; 2021. p. 685–93.

27. Panner M. Will real-world evidence replace clinical trials? [Internet]. Forbes; 2021 [Actualizado 11 agosto de 2021; Consultado el 3 de marzo de 2022]. Available from: https://www.forbes.com/sites/forbestechcouncil/2021/08/11/will-real-world-evidence-replace-clinical-trials/?sh=722cfe2853d2

28. Nazha B, Yang JC, Owonikoko TK. Benefits and limitations of real-world evidence: Lessons from EGFR mutation-positive non-small-cell lung cancer. Future Oncology. 2021;17(8):965–77.

29. Bartlett VL, Dhruva SS, Shah ND, Ryan P, Ross JS. Feasibility of using real-world data to replicate clinical trial evidence. JAMA Network Open. 2019;2.

30. Franklin JM, Patorno E, Desai RJ, Glynn RJ, Martin D, Quinto K, et al. Emulating randomized clinical trials with nonrandomized real-world evidence studies: First results from the RCT DUPLICATE initiative. Circulation. 2021;143(10):1002–13.

31. Eichler H-G, Koenig F, Arlett P, Enzmann H, Humphreys A, Pétavy F, et al. Are novel, nonrandomized analytic methods fit for decision making? The need for prospective, controlled, and transparent validation. Clinical Pharmacology & Therapeutics. 2020;107(4):773–9.

32. Okada M. Big data and real-world data-based medicine in the management of hypertension. Vol. 44, Hypertension Research. 2021.

33. Gliklich RE, Dreyer NA, Leavy MB. Interfacing registries with electronic health records. Vol. 2, Registries for Evaluating Patient Outcomes: A User's Guide. 2014.

34. Logeswaran A, Chong YJ, Edmunds MR. The electronic health record in ophthalmology: Usability evaluation tools for health care professionals. Vol. 10, Ophthalmology and Therapy. 2021.

35. Tai-Seale M, Olson CW, Li J, Chan AS, Morikawa C, Durbin M, et al. Electronic health record logs indicate that physicians split time evenly between seeing patients and desktop medicine. Health Affairs. 2017;36.

36. Sinsky C, Colligan L, Li L, Prgomet M, Reynolds S, Goeders L, et al. Allocation of physician time in ambulatory practice: A time and motion study in 4 specialties. Annals of Internal Medicine. 2016;165.

37. Arndt BG, Beasley JW, Watkinson MD, Temte JL, Tuan WJ, Sinsky CA, et al. Tethered to the EHR: Primary care physician workload assessment using EHR event log data and time-motion observations. Annals of Family Medicine. 2017;15.

38. Downing NL, Bates DW, Longhurst CA. Physician burnout in the electronic health record era: Are we ignoring the real cause? Vol. 169, Annals of Internal Medicine. American College of Physicians; 2018. p. 50–1.

39. Wronikowska MW, Malycha J, Morgan LJ, Westgate V, Petrinic T, Young JD, et al. Systematic review of applied usability metrics within usability evaluation methods for hospital electronic healthcare record systems. Journal of Evaluation in Clinical Practice. 2021 Dec;27:1403–16.

40. Elmasri R, Navathe SB. Fundamentals of database systems sixth edition. Database Systems. 2016.

41. Kogan E, Twyman K, Heap J, Milentijevic D, Lin JH, Alberts M. Assessing stroke severity using electronic health record data: A machine learning approach. BMC Medical Informatics and Decision Making. 2020;20.

42. Lasky T, Carleton B, Horton DB, Kelly LE, Bennett D, Czaja AS, et al. Real-world evidence to assess medication safety or effectiveness in children: Systematic review. Vol. 7, Drugs - Real World Outcomes. 2020.

43. Binkheder S, Asiri MA, Altowayan KW, Alshehri TM, Alzarie MF, Aldekhyyel RN, et al. Real-world evidence of covid-19 patients' data quality in the electronic health records. Healthcare (Switzerland). 2021;9.

44. Pettus JH, Zhou FL, Shepherd L, Mercaldi K, Preblick R, Hunt PR, et al. Differences between patients with type 1 diabetes with optimal and suboptimal glycaemic control: A real-world study of more than 30 000 patients in a US electronic health record database. Diabetes, Obesity and Metabolism. 2020 Apr;22:622–30.

45. Simpao AF, Ahumada L, Rehman M. Big data and visual analytics in anaesthesia and health care. British journal of anaesthesia. 2015;115(3):350–6.

46. Lundberg SM, Nair B, Vavilala MS, Horibe M, Eisses MJ, Adams T, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. Nature biomedical engineering. 2018;2(10):749–60.

47. Schmidt M, Schmidt SAJ, Adelborg K, Sundbøll J, Laugesen K, Ehrenstein V, et al. The danish health care system and epidemiological research: From health care contacts to database records. Clinical epidemiology. 2019;11:563.

48. Johnson AE, Pollard TJ, Shen L, Lehman LH, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible critical care database. Scientific data. 2016;3(1):1–9.

49. Huang B, Liang D, Zou R, Yu X, Dan G, Huang H, et al. Mortality prediction for patients with acute respiratory distress syndrome based on machine learning: A population-based study. Annals of Translational Medicine. 2021;9(9).

50. Siu BMK, Kwak GH, Ling L, Hui P. Predicting the need for intubation in the first 24 h after critical care admission using machine learning approaches. Scientific reports. 2020;10(1):1–8.

51. Scherpf M, Gräßer F, Malberg H, Zaunseder S. Predicting sepsis with a recurrent neural network using the MIMIC III database. Computers in biology and medicine. 2019;113:103395.

52. Cheng B, Li D, Gong Y, Ying B, Wang B. Serum anion gap predicts all-cause mortality in critically ill patients with acute kidney injury: Analysis of the MIMIC-III database. Disease markers. 2020;2020.

53. Yu Y, Wang J, Wang Q, Wang J, Min J, Wang S, et al. Admission oxygen saturation and all-cause in-hospital mortality in acute myocardial infarction patients: Data from the MIMIC-III database. Annals of Translational Medicine. 2020;8(21).

54. Afshar AS, Li Y, Chen Z, Chen Y, Lee JH, Irani D, et al. An exploratory data quality analysis of time series physiologic signals using a large-scale intensive care unit database. JAMIA open. 2021;4(3):ooab057.

55. MIMIC-III clinical database (version 1.4). [Internet]. Getting started. [Consultado el 10 de febrero de 2022]. Available from: https://mimic.mit.edu/docs/gettingstarted/

56. Roussos C, Koutsoukou A. Respiratory failure. European Respiratory Journal. 2003;22(47 suppl):3s–14s.

57. Lumb AB, Thomas CR. Nunn's applied respiratory physiology eBook. Elsevier Health Sciences; 2020.

58. Pisani L, Corcione N, Nava S. Management of acute hypercapnic respiratory failure. Current opinion in critical care. 2016;22(1):45–52.

59. Nava S, Hill N. Non-invasive ventilation in acute respiratory failure. The Lancet. 2009;374(9685):250–9.

60. Le S, Pellegrini E, Green-Saxena A, Summers C, Hoffman J, Calvert J, et al. Supervised machine learning for the early prediction of acute respiratory distress syndrome (ARDS). Journal of Critical Care. 2020;60:96–102.

61. MIMIC-III clinical database (version 1.4) [Internet]. Physionet [Consultado el 10 de febrero de 2022]. Available from: https://physionet.org/content/mimiciii/1.4/

62. Google Cloud Platform [Internet] [Consultado el 8 de febrero de 2022]. Available from: https://cloud.google.com/gcp/?hl=es&utm_source=bing&utm_medium=cpc&utm_campaign=emea-es-all-es-bkws-all-all-trial-e-gcp-1011340&utm_content=text-ad-none-any-DEV_c-CRE_-ADGP_Hybrid%20%7C%20BKWS%20-%20EXA%20%7C%20Txt%20~%20GCP%20~%20General%23v1-KWID_43700061311461129-kwd-77172170806122%3Aloc-170-userloc_3173&utm_term=KW_google%20cloud%20platform-NET_s-PLAC_&gclid=f6dc734e08ba1662402f1152f4d5f326&gclsrc=3p.ds

63. Elixhauser A, Steiner C, Harris DR, Coffey RM. Comorbidity measures for use with administrative data. Medical care. 1998;8–27.

64. Quan H, Sundararajan V, Halfon P, Fong A, Burnand B, Luthi J-C, et al. Coding algorithms for defining comorbidities in ICD-9-CM and ICD-10 administrative data. Medical care. 2005;1130–9.

65. ICD9Data.com [Internet] [Consultado el 16 de marzo de 2022]. Available from: http://www.icd9data.com/2014/Volume1/default.htm

66. Zimmerman JE, Kramer AA, McNair DS, Malila FM. Acute physiology and chronic health evaluation (APACHE) IV: Hospital mortality assessment for today's critically ill patients. Critical care medicine. 2006;34(5):1297–310.

67. Godinjak A, Iglica A, Rama A, Tančica I, Jusufović S, Ajanović A, et al. Predictive value of SAPS II and APACHE II scoring systems for patient outcome in a medical intensive care unit. Acta medica academica. 2016;45(2).

68. Lukannek C, Shaefi S, Platzbecker K, Raub D, Santer P, Nabel S, et al. The development and validation of the score for the prediction of postoperative respiratory complications (SPORC-2) to predict the requirement for early postoperative tracheal re-intubation: A hospital registry study. Anaesthesia. 2019;74(9):1165–74.

69. El-Manzalawy Y, Abbas M, Hoaglund I, Cerna AU, Morland TB, Haggerty CM, et al. OASIS+: Leveraging machine learning to improve the prognostic accuracy of OASIS severity score for predicting in-hospital mortality. BMC medical informatics and decision making. 2021;21(1):1–3.