



**UNIVERSIDAD
DE ANTIOQUIA**

**HERRAMIENTA PARA EVALUACIÓN DEL
DOLOR EN NIÑOS BASADA EN ESCALAS DE
DOLOR EXISTENTES Y VARIABLES
FISIOLÓGICAS.**

Autor

Andrea Carolina Giraldo Bedoya

Universidad de Antioquia

Facultad de Ingeniería, Programa de Bioingeniería

Medellín, Colombia

2020



Herramienta para evaluación del dolor en niños basada en escalas de dolor existentes y variables fisiológicas.

Andrea Carolina Giraldo Bedoya

Tesis como requisito parcial para optar por el título de:

Bioingeniero

Asesora:

MSc. Jenny Kateryne Aristizábal Nieto

Bioingeniero

Línea de Investigación: Aplicaciones móviles e inteligencia artificial.

Grupo de Investigación: GRIMPA

Universidad de Antioquia

Facultad de Ingeniería, Programa de Bioingeniería

Medellín, Colombia

2020

AGRADECIMIENTOS

A mis padres por todo su apoyo durante todo el proceso de formación que culmina en este proyecto. A mi hermano por todo el amor que me ha brindado.

Agradezco a la profesora Jenny Aristizábal por guiarme no solo durante este proyecto, si no durante la mayor parte de la carrera, es quizás de la persona que más aprendí durante todo mi proceso formativo.

Finalmente, a todos los participantes en el proyecto que aportaron sus ideas y realimentaciones para llevar a cabo los objetivos.

RESUMEN

En el marco del proyecto *“Desempeño y diseño de un modelo de dolor postquirúrgico en niños menores de 36 meses para la clasificación de dolor en comparación con la escala tradicional pediátrica FLACC”*, desarrollado por el Hospital San Vicente Fundación en conjunto con la Universidad de Antioquia, se hace necesario el desarrollo de una aplicación que facilite la recolección de datos del paciente, además con el objetivo de usar los datos recolectados para beneficio de los sujetos del estudio, se realizó un análisis de los datos, con el fin de entrenar un modelo de inteligencia artificial(máquinas de soporte vectorial) para clasificar con base en las variables fisiológicas los pacientes con dolor de los pacientes sin dolor.

La aplicación móvil FLACC se desarrolló con el software Android Studio, esta tuvo buena acogida de los usuarios que la evaluaron en términos de funcionalidad, usabilidad y diseño, sugiriendo que la información fuera sincronizada en la nube para que todos tuvieran acceso sin importar desde cual dispositivo se registraran los datos.

Por otro lado, con base en el análisis de los datos y las máquinas de soporte vectorial entrenadas, los resultados demuestran que con los datos recolectados no es posible diferenciar entre pacientes con dolor y sin dolor a partir de las variables fisiológicas.

CONTENIDO

Índice de Tablas	7
Índice de Figuras	8
1. Introducción	10
2. Objetivos	12
2.1. Objetivo general	12
2.2. Objetivos específicos	12
3. Marco Teórico.....	13
3.1. Evaluación del Dolor	13
3.1.1. Escala de dolor FLACC.....	13
3.1.2. Variables fisiológicas	14
3.2. Bases de datos.....	15
3.2.1. SQLite.....	15
3.3. Aplicación Android.....	16
3.3.1. Entorno de desarrollo: Android Studio.....	16
3.3.2. Flujo para la creación de una aplicación	18
3.4. Inteligencia Artificial.....	20
3.4.1. Preprocesamientos de los datos.....	21
3.4.2. Máquinas de soporte vectorial.....	23
3.4.3. Evaluación del modelo	27
3.4.4. Scikit-learn.....	30
4. Metodología	31
4.1. Aplicación Android.....	32
4.1.1. Levantamiento de requerimientos.....	32
4.1.2. Diseño del mockup	32
4.1.3. Desarrollo del Front-end y Back-end.....	32
4.1.4. Validación de la aplicación.....	33
4.2. Modelo de Inteligencia Artificial.....	34
4.2.1. Preprocesamiento de datos	34

4.2.2.	Revisión de modelos de inteligencia artificial.....	34
4.2.3.	Entrenamiento del modelo.....	34
4.2.4.	Validación del modelo.....	34
4.3.	Implementar modelo de inteligencia artificial en la aplicación.....	34
5.	Resultados	35
5.1.	Aplicación Android	35
5.1.1.	Diseño del mockup.....	36
5.1.2.	Desarrollo del Front-End y el Back-End	37
5.1.3.	Validación de la aplicación.....	42
5.2.	Modelo de Inteligencia Artificial	43
5.2.1.	Preprocesamiento de los datos	43
5.2.2.	Revisión, entrenamiento y validación del modelo.....	47
6.	Análisis de Resultados	51
6.1.	Aplicación Android.....	51
6.1.1.	Diseño del mockup.....	51
6.1.2.	Desarrollo de FrontEnd y BackEnd.....	51
6.1.3.	Validación de la aplicación.....	52
6.2.	Modelo de Inteligencia Artificial.....	53
6.2.1.	Preprocesamiento de los datos	53
6.2.2.	Revisión, entrenamiento y validación del modelo.....	54
7.	Conclusiones	55
8.	Bibliografía	57

ÍNDICE DE TABLAS

Tabla 1. Escala de valoración del dolor FLACC.....	14
Tabla 2. Signos vitales normales para niños de 0-3 años .	15
Tabla 3. Estructura de la encuesta realizada.....	33
Tabla 4. Cantidad de sujetos clasificados según la FLACC.	43
Tabla 5. Media y desviación estándar de las variables normalizadas.	44
Tabla 6. Porcentaje de alteración en las variables fisiológicas para cada clase de dolor.	46
Tabla 7. Cantidad de variables alteradas por cada clase de dolor.....	47
Tabla 8. Clases binarias definidas.	47
Tabla 9. Cantidad de datos usados para el entrenamiento.	48
Tabla 10. Parámetros usados en la máquina de soporte vectorial.....	48

ÍNDICE DE FIGURAS

Figura 1. Interfaz de Android Studio	17
Figura 2. Elementos del editor de diseño visual. 1)Paleta, 2) Component tree, 3) Toolbar, 4) Design Editor, 5) Attributes.....	17
Figura 3. Fast emulator proporcionado por Android Studio.....	18
Figura 4. Flujo para la creación de una aplicación	20
Figura 5. Tipos y técnicas de aprendizaje de máquina.....	21
Figura 6. División del espacio en dos por medio de un vector de pesos que definen un hiperplano	24
Figura 7. Representación de como divide el espacio un hiperplano con diferentes kernel. ...	25
Figura 8. Margen de separación.	26
Figura 9. Efecto de la constante C en el máximo margen de separación.....	27
Figura 10. Matriz de confusión.....	28
Figura 11. Curva ROC, con AUC=1.....	29
Figura 12. Curva ROC para un AUC=0.5	29
Figura 13. Curva ROC para una AUC=0.....	30
Figura 14. Metodología implementada en el proyecto.....	31
Figura 15. Definición de requerimientos	35
Figura 16. Mockup diseñado.	36
Figura 17. Diseño de la base de datos implementada.	37
Figura 18. Pantalla principal de la aplicación.....	38
Figura 19. Ingreso datos del paciente.	38
Figura 20. Ingreso de la FLACC	39
Figura 21. Ingreso de variables fisiológicas	39
Figura 22. Consulta de pacientes.....	40
Figura 23. a) Consulta por historia clínica. b) Consulta por consecutivo.....	40
Figura 24. a) Listado de pacientes. b) Datos del paciente	41
Figura 25. Nivel de satisfacción promedio por usuario.....	42
Figura 26. Promedio de satisfacción por parámetro evaluado.....	42
Figura 27. Porcentajes de las clases de dolor.	43
Figura 28. Matriz de correlación	44
Figura 29. Distribución de los datos para la clase de no dolor(morado) vs dolor leve(verde). a) Presión arterial sistólica vs frecuencia cardiaca. b) Frecuencia cardiaca vs Saturación de oxígeno. c) Presión arterial diastólica vs Saturación de oxígeno.	45

Figura 30. Distribución de los datos para la clase de no dolor vs dolor moderado. a) Presión arterial sistólica vs frecuencia cardiaca. b) Frecuencia cardiaca vs Saturación de oxígeno. c) Presión arterial diastólica vs Saturación de oxígeno	45
Figura 31. Distribución de los datos para la clase de no dolor vs dolor. a) Presión arterial sistólica vs frecuencia cardiaca. b) Frecuencia cardiaca vs Saturación de oxígeno. c) Presión arterial diastólica vs Saturación de oxígeno.....	46
Figura 32. Matriz de confusión primera clasificación binaria.....	49
Figura 33. Matriz de confusión segunda clasificación binaria con Kernel RBR y sigmoid. .	49
Figura 34. Matriz de confusión segunda clasificación binaria con Kernel polinomial de grado 3.....	50
Figura 35. Matriz de confusión para la tercera clasificación.....	50

1. INTRODUCCIÓN

El dolor se define como una experiencia sensorial desagradable y es un síntoma de muchas patologías, por esto, la evaluación del dolor es un paso necesario en el diagnóstico. Sin embargo, esta evaluación se dificulta en los niños no comunicativos, debido a que estos niños no tienen el desarrollo mental ni el lenguaje para describir su dolor, por esta razón, se hace necesaria una evaluación adecuada para el dolor, con el fin de poder diagnosticar enfermedades en esta población y orientar las intervenciones analgésicas; teniendo en cuenta que el dolor no tratado puede traer graves repercusiones a largo plazo tanto física como mentalmente [1].

Con el propósito de evaluar el dolor en estos pacientes, se han creado diversas escalas de dolor de uso común a nivel mundial, como la FLACC, CHEOPS, CHIPPS [2] entre otras. El problema con el uso de estas escalas de evaluación, es que ninguna ha incluido a población colombiana en sus estudios para la definición de las variables, por esto es importante reestructurar estas escalas de una manera que sea acorde a la población del Hospital Fundación San Vicente, y establecer una estrategia de implementación que sea de fácil acceso al personal médico además de ofrecer la posibilidad de tener una base de datos propia del Hospital, con datos de la población promedio que atienden, esto para que en un futuro la intensidad de dolor que se obtiene después de la evaluación sea más acertada con la sensación del paciente y así brindar unos cuidados más asertivos. Para lograr esto, se propone el diseño de una aplicación móvil que permita el registro de datos, almacenamiento de estos, y análisis.

Además se propone analizar los datos recolectados en un período de un año intentando relacionar las variables fisiológicas registradas con el valor de la FLACC para cada sujeto, usando un modelo de inteligencia artificial que clasifique entre dolor y no dolor, devolviendo así una predicción del dolor que puede estar sintiendo el paciente, esto basado en la premisa que los signos vitales en particular la frecuencia cardiaca, la saturación de oxígeno y la presión arterial aumenta cuando el sujeto siente dolor.

En este proyecto se llevó a cabo el desarrollo y evaluación de una aplicación móvil haciendo uso del software Android Studio, que permite la recolección de datos asociados al proyecto *“Desempeño y diseño de un modelo de dolor postquirúrgico en niños menores de 24 meses para la clasificación de dolor en comparación con la escala tradicional pediátrica FLACC”*, que se está llevando a cabo en el Hospital San Vicente Fundación, entre estos datos se encuentra el valor de la escala FLACC y las variables fisiológicas de cada sujeto, luego con estos datos recolectados se entrenaron varias máquinas de soporte vectorial para evaluar si algún modelo podía clasificar entre pacientes sin dolor y pacientes con dolor a partir de las variables fisiológicas, finalmente se evaluaron los modelos y si su rendimiento justificaba la implementación de estos modelos en la aplicación.

En la sección del 3 del presente informe se presentan los conceptos necesarios para entender lo realizado en el proyecto. En la sección 4 se encuentra la metodología desarrollada para cumplir los objetivos, en la sección 5 se presentan los resultados obtenidos y en la sección 6 se analizan estos resultados, finalmente, en la sección 7 se presentan las conclusiones respecto a los objetivos.

2. OBJETIVOS

2.1. Objetivo general

Desarrollar un aplicativo que sirva como herramienta al personal médico para evaluar el dolor en niños con base en escalas de dolor existentes (FLACC) y variables fisiológicas, y que permita obtener un valor de la intensidad de dolor postoperatorio, en la población infantil del Hospital Universitario San Vicente Fundación.

2.2. Objetivos específicos

- Establecer y relacionar variables como: frecuencia cardíaca, saturación de oxígeno, presión arterial, con el valor de la escala FLACC, mediante un modelo de inteligencia artificial (IA) que procese los datos obtenidos.
- Desarrollar una aplicación en Android que implemente la escala de evaluación FLACC y el modelo IA para realizar una predicción del dolor y que facilite el registro y procesamiento de los datos.
- Realizar pruebas de validación de la aplicación y del modelo obtenido.

3. MARCO TEÓRICO

3.1. Evaluación del Dolor

La Asociación Internacional para el Estudio del Dolor (IASP) define el dolor como cualquier experiencia emocional y sensorial molesta que se puede asociar a un daño en los tejidos real o potencial. Este actúa como alarma que permite reconocer un mal funcionamiento en el organismo. Sin embargo, los niños pequeños no están familiarizados con estas sensaciones ya que no pueden expresar verbalmente lo que sienten y es probable que no hayan tenido lesiones pasadas. El dolor se clasifica según su duración (agudo o crónico), su patogénesis (nociceptivo, neuropático o psicogénico), su intensidad, su causa y su localización [3].

Factores que influyen sobre el dolor:

La percepción del dolor varía en cada persona ya que no depende solamente de un proceso biológico, sino de factores genéticos, cognoscitivos, psicológicos y socioculturales [4] los cuales influyen en el manejo que cada paciente requiere. Diversos elementos afectan la reacción ante el dolor de cada niño de manera diferente, debido a esto se hace necesario tener en cuenta a la hora de evaluar el dolor en pediatría, especialmente cuando los pacientes no se pueden comunicar verbalmente, los siguientes factores [5]:

- **Cognitivos:** Creencias, actitudes espirituales y culturales.
- **Afectivos:** Emociones.
- **Conductuales:** Cambio del comportamiento.
- **Fisiológicos:** Frecuencia cardíaca, presión sanguínea, frecuencia respiratoria.

3.1.1. Escala de dolor FLACC

La medición ideal del dolor debe ser sensible, sin sesgos, válida, simple, exacta, fiable y barata, se debe buscar el mayor número de estas cualidades. Dependiendo de la metodología utilizada y debilidades de aplicarlas, se pueden disponer de otros tipos de métodos. En la etapa preverbal de los niños (menores de 3 años) para valorar el dolor se usan escalas que evalúan variables conductuales como: la actividad motora, el llanto, la expresión facial; también variables fisiológicas como: la presión arterial, frecuencia cardíaca, saturación de oxígeno [2].

Hay diversas herramientas que pueden ser usadas para evaluar el dolor postoperatorio agudo en los niños, valorar las expresiones faciales, su agitación, nerviosismo, expresiones verbales que manifieste el niño, como por ejemplo el llanto. Con estos datos se han establecido escalas

como la FLACC (The faces legs consolability scale) diseñada para evaluar el dolor postoperatorio en niños aplicable a niños hasta 4 años o a niños mayores que no colaboran [2], cada una de las letras significa un criterio a evaluar F: face(cara), L: legs(piernas), A:Activity(actividad), C:consolability(consolabilidad), C:Cry(llanto) donde a cada uno se le puede otorgar un puntaje de 0 a 2 (Tabla 1. Escala de valoración del dolor FLACC.). Para usar la FLACC el personal clínico observa el paciente durante 5 minutos y asigna un puntaje a cada una de las categorías de la escala, seleccionando el número que más se acerque al comportamiento que observó, luego se suman los puntos obtenidos en cada categoría obteniendo el valor de dolor final que estará entre 0 y 10 [6], de acuerdo al puntaje de la FLACC se define la cantidad de dolor así: 0 no dolor, 1-3 dolor leve, 4-6 dolor moderado y de 7-10 dolor intenso.

Tabla 1. Escala de valoración del dolor FLACC.

Puntuación	0	1	2
Expresión facial	Expresión habitual Cara relajada	Arruga la nariz	Temblor del mentón Mandíbula tensa
Posibilidad de consolarle	Esta a gusto de manera espontánea	Se le puede consolar	Es difícil consolarle
Llanto	No hay llanto	Gimotea; se queja	Llanto intenso
Actividad	Esta acostado y tranquilo	Se dobla sobre su abdomen Encoje sus piernas	Esta rígido
Movimiento de las piernas	Piernas relajadas	Piernas inquietas	Pataleo intenso

3.1.2. Variables fisiológicas

Cuando se evalúa el dolor en niños se debe hacer uso de variables no subjetivas como lo son las variables fisiológicas, que pueden ser medidas con un monitor de signos vitales, ya que el cambio o ciertos valores anormales en su medición pueden ser indicadores de cambios en el paciente, en este caso asociados al dolor.

La mayoría de las investigaciones relacionadas con la evaluación del dolor en niños concuerdan en que la saturación de oxígeno, la frecuencia cardiaca, la presión arterial, entre otras, varían ante estímulos dolorosos tanto para niños neonatos como para los más maduros y aunque los indicadores fisiológicos de dolor pueden ser más variables y menos consistentes

que los indicadores comportamentales se debe tener una aproximación multidimensional para mejorar la validez de la evaluación [7].

Los signos vitales normales para niños de 0-3 años se presentan en la Tabla 1 se ha encontrado que ante un estímulo doloroso se producen los siguientes cambios fisiológicos: incremento en el ritmo cardiaco, respiraciones cortas y superficiales, aumento en el número de respiraciones (frecuencia respiratoria), jadeos, aumento en la presión arterial, pupilas dilatadas, sudoración en las manos y disminución en la saturación de oxígeno [8]. Cuando estos cambios se registran junto con las variables comportamentales se obtiene una buena valoración del dolor postquirúrgico en niños.

Tabla 2. Signos vitales normales para niños de 0-3 años [9].

Edad	Frec. Respiratoria (resp/min)	Frec. Cardiaca (lat/min)	Presión arterial (mmHg)
0-1 meses	30-60	85-190	67-84/31-45
1-12 meses	30-60	90-180	72-104/37-56
1-3 años	24-40	98-140	86-106/42-63

En cuanto a la saturación de oxígeno los valores normales oscilan entre un 95%-97% con una variación del 2%.

3.2. Bases de datos

Una base de datos se refiere a una colección organizada de información estructurada, almacenada típicamente en un sistema computacional. Una base de datos es controlada por un sistema gestor de base de datos (DBMS). Los datos, el DBMS y las aplicaciones asociadas a estos se conocen como el sistema de base de datos, frecuentemente mencionado solo como base de datos.

Lo más común es encontrar los datos organizados en filas y columnas en una serie de tablas, que hace eficiente el procesamiento y consulta de los datos. Los datos pueden ser fácilmente consultados, modificados, gestionados, actualizados, controlados y organizados. La mayoría de las bases de datos usa un lenguaje estructurado de consulta (SQL) para escribir y consultar datos [10].

3.2.1. SQLite

SQLite es una librería basada en lenguaje C que implementa, una máquina rápida, pequeña, de alta confianza y con todas las características de una máquina de bases de datos SQL, por estas cualidades es la maquina más usada en el mundo.

SQLite está integrada en todos los dispositivos móviles, a diferencia de otras bases de datos SQL, SQLite no necesita de un servidor aparte, si no que escribe y lee directamente de archivos ordinarios en el disco, una base de datos completa SQL con múltiples tablas, índices, disparadores y vistas está contenida en un solo archivo en el disco, una base de datos SQLite puede contener hasta 140 Terabytes de información y hasta un tamaño máximo de fila de 1 Gigabyte, estas características hacen que SQLite sea la maquina más popular usada en aplicaciones [11].

3.3. Aplicación Android

Una aplicación Android es una aplicación software que se puede ejecutar en la plataforma Android, es típicamente desarrollada para dispositivos móviles que corren en el Android OS, aunque una aplicación Android puede ser accesible a través de las páginas web de los desarrolladores, la mayoría de estas pueden ser publicadas en el Android Market, una tienda online dedicada a este tipo de aplicaciones.

Las aplicaciones Android están escritas en lenguaje Java y usa librerías basadas en Java, estas son compiladas a ejecutables llamados Dalvik para correr en Dalvik virtual machine, la cual está diseñada especialmente para dispositivos móviles. Para el desarrollo de estas aplicaciones se usa el kit de desarrollo de software para Android (SDK), el cual se puede descargar del Android website [12].

3.3.1. Entorno de desarrollo: Android Studio

Es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014 [12].

Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, macOS y GNU/Linux. Ha sido diseñado específicamente para el desarrollo de Android.

En la Figura 1 se muestra la interfaz general de Android Studio.

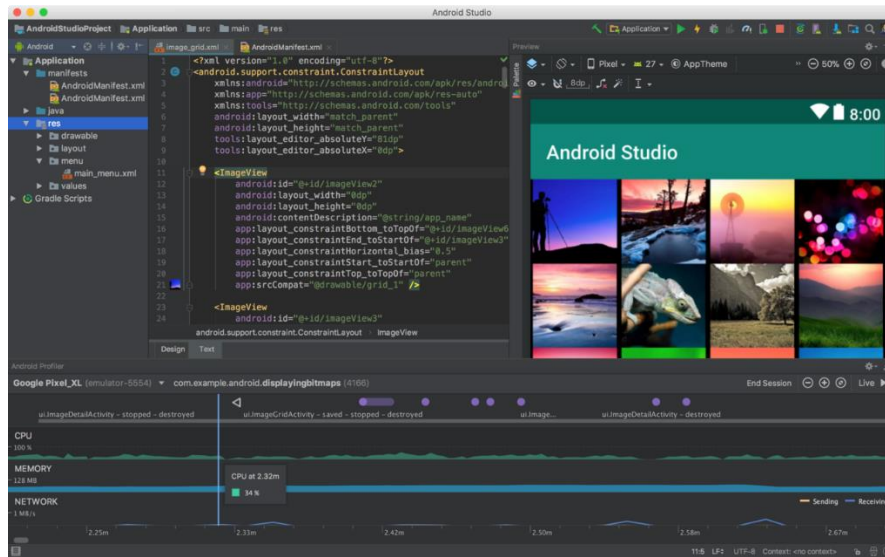


Figura 1. Interfaz de Android Studio

Este entorno tiene varias funcionalidades muy útiles a la hora de desarrollo de aplicaciones para Android OS como lo son:

1. **Editor del diseño visual:** este permite crear layouts (diseño de la aplicación) complejos, con la herramienta ConstraintLayout se pueden añadir restricciones a los objetos: como cuadros de texto, entre otros que se vayan a usar en la aplicación, también tener una vista previa en cualquier tamaño de pantalla por medio de la selección de uno o varios dispositivos [13]. En la figura 2 se muestra como luce este editor y las partes que lo componen.

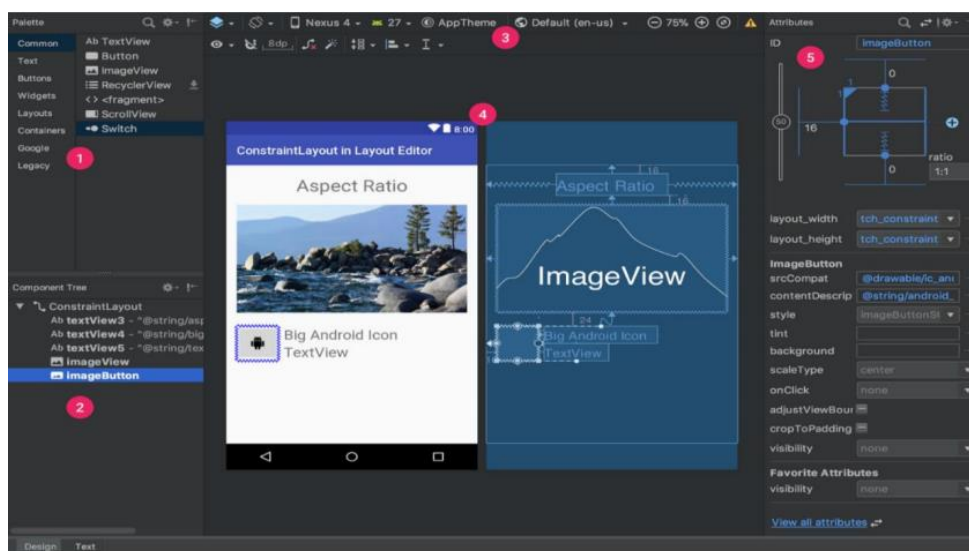


Figura 2. Elementos del editor de diseño visual. 1) Paleta, 2) Component tree, 3) Toolbar, 4) Design Editor, 5) Attributes.

2. **Analizador del APK:** Con esta herramienta se puede conocer todos los archivos que componen la APK luego de construida para analizar cómo se puede reducir su tamaño y mejorar su funcionalidad [14].
3. **Fast emulator:** Permite instalar y correr las aplicaciones más rápido que con un dispositivo físico además de simular diferentes configuraciones y características, incluyendo el ARcore, la plataforma de Google para construir realidad aumentada [15] .



Figura 3. Fast emulator proporcionado por Android Studio.

4. **Editor de código inteligente:** Editor de código inteligente que permite compilar en lenguajes Kotlin, Java y C/C++ [13].

3.3.2. Flujo para la creación de una aplicación

El flujo de trabajo para desarrollar una aplicación Android es conceptualmente el mismo que para desarrollar cualquier aplicación móvil. Para construir una aplicación bien diseñada para Android se deben usar herramientas especializadas y seguir el siguiente flujo de trabajo [16]:

1. Configurar el espacio de trabajo

Esta fase consiste en buscar e instalar las herramientas necesarias para el desarrollo de la aplicación como Android Studio, además de crear el proyecto.

2. Escribir el código

Esta es la fase más importante ya que es en esta donde se construye el código de acuerdo con las necesidades de la aplicación que se va a desarrollar, aquí se usan los diferentes recursos que ofrece Android Studio, como el NDK para añadir código fuente de C/C++ a la aplicación, se crean recursos para los diferentes tipos de dispositivos, se construye la interfaz, los widgets y las funciones.

3. Construir y correr

Durante esta fase se construye el proyecto en un APK sobre la que se puede hacer debugging y que puede ser instalada en un emulador o en un dispositivo Android, en base a esto se puede comenzar a personalizar la construcción de la APK para de esta manera reducir el código y ahorrar recursos logrando que la APK sea más pequeña y eficiente en los dispositivos.

4. Debuggiar, perfilar y probar

Esta es la fase iterativa en la cual se sigue sobrescribiendo sobre la aplicación, pero ya con un objetivo diferente, el de eliminar los bugs (fallas en la compilación) y optimizar el rendimiento de la aplicación. Para esto se pueden crear test para que ayuden en estas tareas de probar la aplicación.

5. Publicar

En esta fase la aplicación ya está lista para ser lanzada a los usuarios, en esta etapa solo quedan un par de cosas por considerar tales como colocar la versión de la aplicación y firmarla con una llave. Durante el paso de preparación, se compila una versión de lanzamiento de la aplicación que los usuarios pueden descargar e instalar en sus dispositivos con sistema operativo Android.

En la Figura 4. Flujo para la creación de una aplicación se muestra esquemáticamente este flujo.

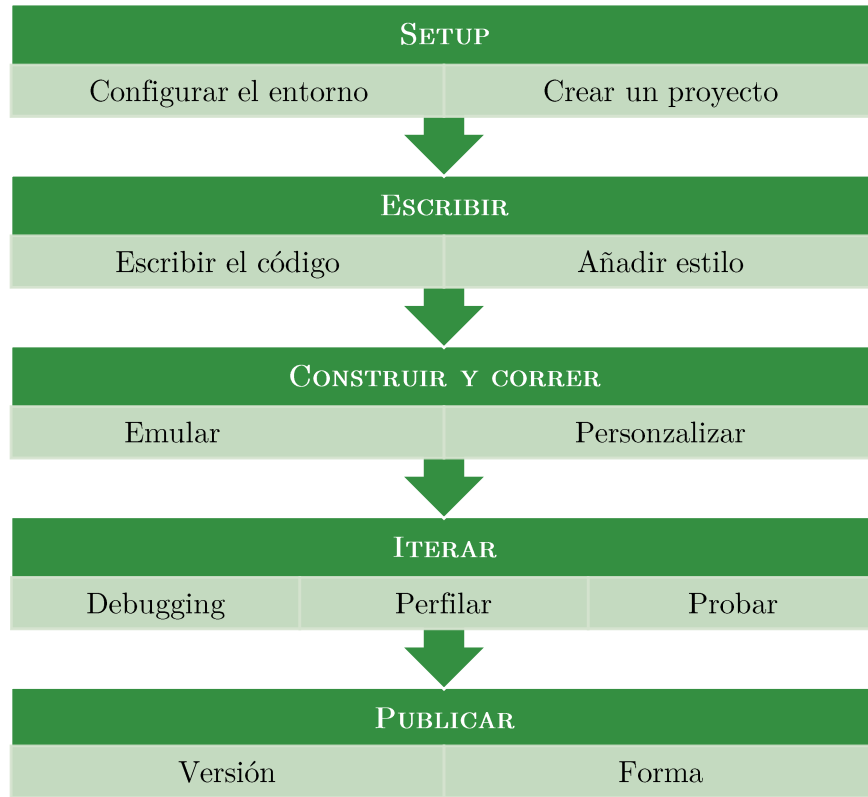


Figura 4. Flujo para la creación de una aplicación

3.4. Inteligencia Artificial

La inteligencia artificial (IA) está definida como la habilidad de un sistema para interpretar correctamente datos externos, de aprender de estos datos y usar estos aprendizajes para lograr metas específicas a través de adaptación flexible [17].

Los algoritmos de aprendizaje de máquina "aprenden" directamente de los datos y mejoran su rendimiento a medida que el conjunto de estos datos es cada vez mayor y son representativos del problema que quiere imitar [18].

Hay dos técnicas en el aprendizaje automático: El aprendizaje supervisado en el cual se entrena un modelo con los datos de entrada y de salida conocidos para obtener salidas futuras y el aprendizaje no supervisado, que encuentra patrones ocultos o estructuras específicas en los datos de entrada [19].

Además de los diferentes tipos de aprendizaje también hay diversas técnicas que se ilustran en la Figura 5. Tipos y técnicas de aprendizaje de máquina.

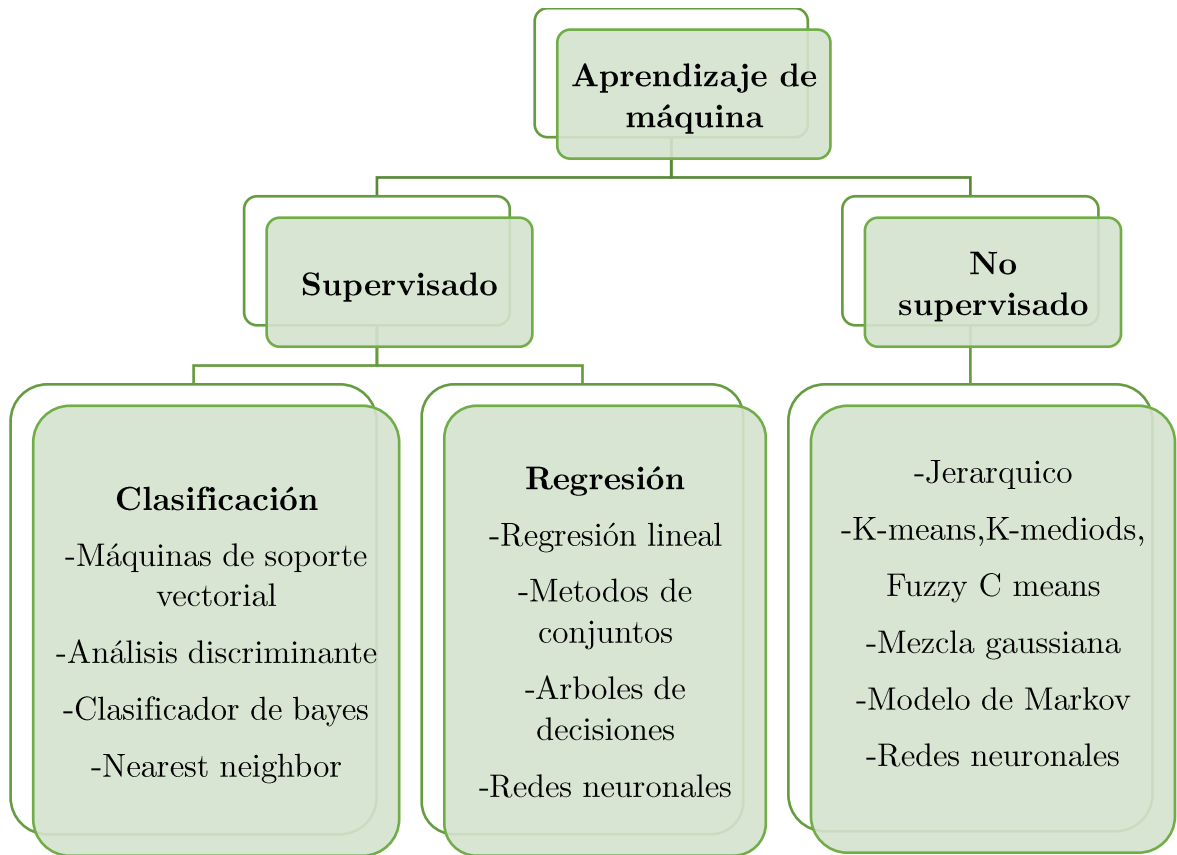


Figura 5. Tipos y técnicas de aprendizaje de máquina

3.4.1. Preprocesamientos de los datos

Antes de tratar de clasificar un conjunto de datos por medio de métodos de inteligencia artificial es importante realizar un análisis estadístico básico de los datos, esto ayuda a caracterizarlos y encontrar patrones que ayudan a decidir qué modelo y parámetros usar de acuerdo con la naturaleza de los datos. Por lo general este análisis incluye tabular la cantidad, graficar su distribución y calcular la media y la desviación estándar.

Después del análisis estadístico se debe realizar una preparación de los datos minimizando así errores cuando se entrene el modelo escogido, generalmente se siguen los siguientes pasos antes de entrenar un modelo:

- **Normalización**

Normalizar los datos es escalarlos a una misma distancia, es decir que el valor de todas las variables se encuentre en un mismo rango, normalmente entre 0 y 1, esto se hace con el

propósito de que todas las variables tengan el mismo peso en el modelo, si por ejemplo una variable se midiera en cientos y otra en miles, solo por su valor la de los miles pesaría más en el momento de entrenar el modelo [20].

Las dos técnicas de normalización más utilizadas son la normalización mínimo máximo (ec.1) donde x_{min} es el mínimo valor de una variable en un conjunto de datos, x_{max} es el máximo valor de esta misma variable y x' es el valor normalizado.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} [1]$$

La otra técnica de normalización es el *z-score* (ec.2), cuando se usa esta técnica los valores no van a oscilar entre 0 y 1 si no que habrá valores negativos, pero igualmente las variables se encontraran en una misma escala.

$$x' = \frac{x - \mu}{\sigma} [2]$$

Donde μ es la media y σ la desviación estándar.

- **Agregar datos faltantes**

Por diversas razones, muchas bases de datos poseen valores faltantes, lo que conlleva a descartar una entrada completa a la que le falte un solo valor, esto significa perder datos valiosos que pueden ayudar a que el modelo se desempeñe mejor. Para solucionar esto se han desarrollado diversas estrategias para inferir valores a partir de los valores que si se encuentran en la base de datos [21].

Una de las técnicas más simples para generar nuevos datos es generar valores aleatorios a partir de la media y la desviación estándar, otra estrategia consiste en usar el valor más frecuente como reemplazo de los valores faltantes [22].

Se han desarrollado técnicas más avanzadas como los *K nearest neighbours* [23] o los *MICE* (*Multivariate Imputation by Chained Equation*) [24].

- **Balanceo de clases**

En algunos problemas de clasificación (aprendizaje supervisado) hay una diferencia considerable entre las clases, lo que significa que una clase tiene muchas más muestras que la otra, si estas se ingresaran en estas proporciones a un clasificador probablemente este aprendería a clasificar la clase con mayor cantidad de datos, generando un problema conocido como el desbalanceo de clases [25].

Cuando este problema existe en la base de datos a utilizar, se puede solucionar usando dos técnicas principales de remuestreo. La primera es sobremuestrear, que consiste en crear una nueva base de datos replicando algunas muestras originales o creando nuevas a partir de las existentes; la otra técnica es submuestrear, que consiste en eliminar de la base de datos original datos que pertenezcan a la clase mayoritaria [26]. También hay otras técnicas más sofisticadas como “Synthetic Minority Oversampling Technique” (SMOTE) [27], que interpola varios datos cercanos de la clase minoritaria.

3.4.2. Máquinas de soporte vectorial

Las máquinas de soporte vectorial es un método de clasificación introducido en 1992 por Boser, Guyon, y Vapnik [28].

Una máquina de soporte vectorial separa los datos que se desean clasificar por medio de un hiperplano, es decir al ingresar datos etiquetados (aprendizaje supervisado), el algoritmo entrega un hiperplano óptimo que clasifique nuevos ejemplos, si fuera en un espacio de dos dimensiones este hiperplano sería una línea dividiendo los datos en dos clases, una a cada lado [29], como se observa en la Figura 6. División del espacio en dos por medio de un vector de pesos que definen un hiperplano .

Esto en términos matemáticos se puede explicar de la siguiente forma, suponiendo que se tiene un problema de clasificación de dos clases donde las etiquetas son: +1(ejemplos positivos) y -1(ejemplos negativos), que \mathbf{x} representa un vector con componentes \mathbf{x}_i , donde este denota el i -ésimo vector en un conjunto de datos compuesto por n ejemplos etiquetados $(\mathbf{x}_i, \mathbf{y}_i)$, donde \mathbf{y}_i es la etiqueta asociada a \mathbf{x}_i . Los objetos \mathbf{x}_i son llamados entradas y en este caso se asumieron como vectores, pero pueden ser cualquier objeto continuo o discreto [30].

Un concepto clave requerido para definir un clasificador lineal es el *producto punto* entre dos vectores definido como $\mathbf{w}^T \mathbf{x} = \sum_i w_i x_i$. Un clasificador lineal está basado en una *función discriminante* de la forma [30]

$$f(x) = \mathbf{w}^T \mathbf{x} + b \quad [3]$$

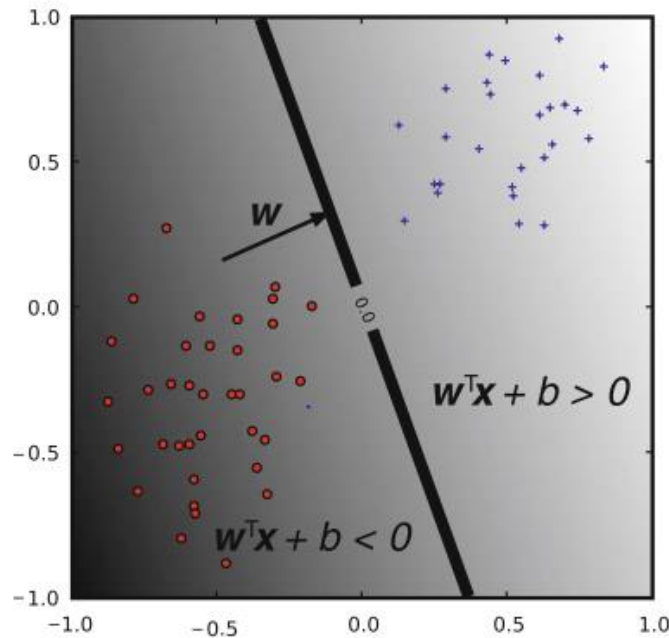


Figura 6. División del espacio en dos por medio de un vector de pesos que definen un hiperplano [30].

El vector w es conocido como el vector de pesos y b es llamado el bias. Si se considera el caso en que $b=0$, el conjunto de puntos x tales que $w^T x = 0$, son todos los puntos perpendiculares a w y que pasan por el origen, es decir una línea en dos dimensiones, un plano en tres dimensiones y más generalmente un hiperplano. El bias b traslada el hiperplano lejos del origen. Este hiperplano divide el espacio en dos como se observa en la Figura 6. División del espacio en dos por medio de un vector de pesos que definen un hiperplano ., la frontera entre las regiones clasificadas como positivas y negativas es llamada la *frontera de decisión* del clasificador. Aunque en lo anterior se ha descrito un clasificador lineal, debido a que las entradas en la definición de la ecuación 3 son lineales, en realidad puede depender de una manera no lineal como en la ecuación 4, donde ϕ es una función que mapea las entradas lineales x a un espacio no lineal [30].

$$f(x) = w^T \phi(x) + b \quad [4]$$

Una característica más que define una máquina de soporte vectorial es que no se calculan directamente los mapeos ϕ , ya que esto implica usar mayor memoria de computo, sino que se hace uso de kernels, lo que evita llevar los datos a un espacio dimensional alto.

Suponiendo que el vector de pesos se puede expresar como una combinación lineal de los ejemplos de entrenamiento, $w = \sum_{i=1}^n \alpha_i x_i$

Luego la ecuación 3 quedaría de la forma

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i \mathbf{x}_i^T \mathbf{x} + b$$

En el espacio de características, F , esta expresión toma la forma

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b \quad [5]$$

Como se observa en la ecuación 5 el espacio de características F puede ser multidimensional haciendo que este cambio sea poco práctico en términos computacionales a no ser que la función kernel $k(\mathbf{x}, \mathbf{x}')$ definida como [30]

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})\phi(\mathbf{x}')$$

pueda ser computada eficientemente. En términos del kernel la función discriminante es

$$f(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}_i) + b \quad [6]$$

La función k puede ser de cualquier forma: lineal, polinómica, gaussiana, entre otros, este es uno de los parámetros más importantes a definir cuando se usan máquinas de soporte vectorial para un problema de clasificación. En la Figura 7. Representación de como divide el espacio un hiperplano con diferentes kernel . se ilustra como se dividiría el espacio con un kernel lineal y polinomial de grado 2 y 5.

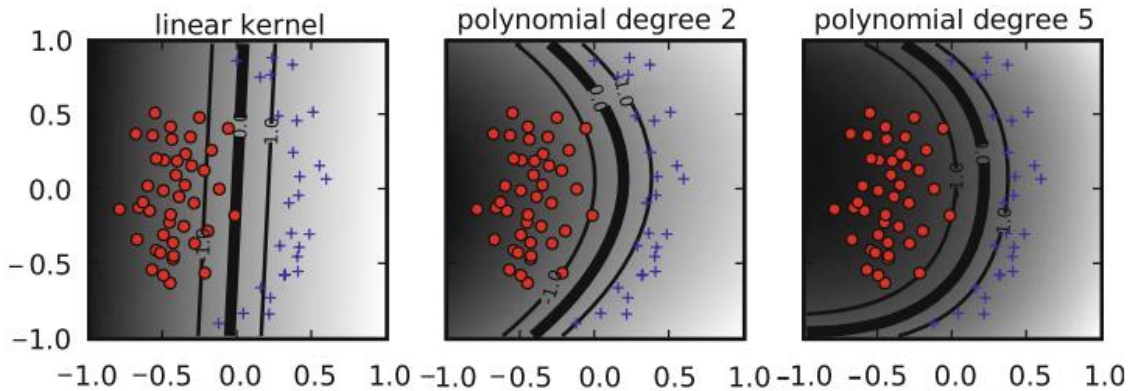


Figura 7. Representación de como divide el espacio un hiperplano con diferentes kernel [30].

La última característica de una máquina de soporte vectorial es el **máximo margen de separación**, partiendo de la **distancia** que separa la frontera (hiperplano) de los ejemplos positivos y los negativos definida como:

$$m_D(\mathbf{w}) = \frac{1}{2} \hat{\mathbf{w}}^T (\mathbf{x}_+ - \mathbf{x}_-) = \frac{1}{\|\mathbf{w}\|}$$

Donde $\hat{\mathbf{w}}$ es un vector unitario en la dirección de \mathbf{w} , \mathbf{x}_+ y \mathbf{x}_- son los puntos más cercanos al hiperplano que define la frontera y son equidistantes [28]. En la Figura 8. Margen de separación . se muestra el máximo margen de separación, los puntos encerrados serían los más cercanos al hiperplano de ambas clases.

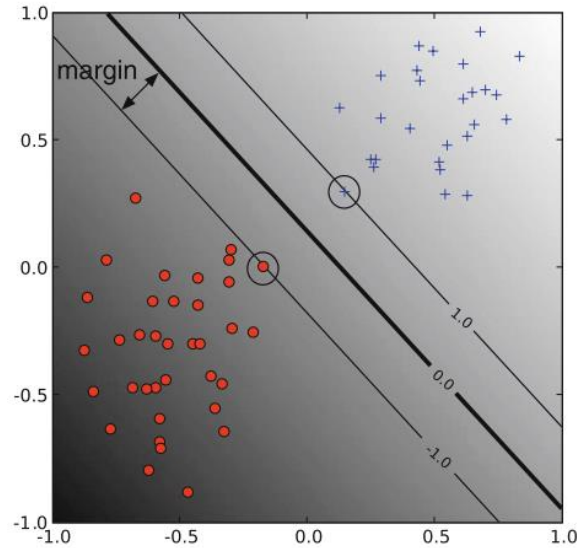


Figura 8. Margen de separación [30].

El **máximo margen de separación** sería la función que máxime la función de la distancia $\frac{1}{\|\mathbf{w}\|}$, lo que es equivalente a minimizar $\|\mathbf{w}\|^2$, minimizando esta función se está asumiendo que ningún punto puede estar mal clasificado, en la práctica esto no es posible por lo que se permiten algunos errores (ε) de clasificación para encontrar el mejor margen, a esta minimización de le añade un término C que permite penalizar los errores en la clasificación [30]. Luego el problema de optimización se convierte en:

$$\text{minimizar } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \varepsilon_i$$

En la Figura 9 se observa el efecto de la penalización de la constante C, mientras más grande sea esta, menor error permite o lo que es lo mismo el máximo margen de distancia disminuye haciendo que el error disminuya, pero acercándose más al sobre entrenamiento [31].

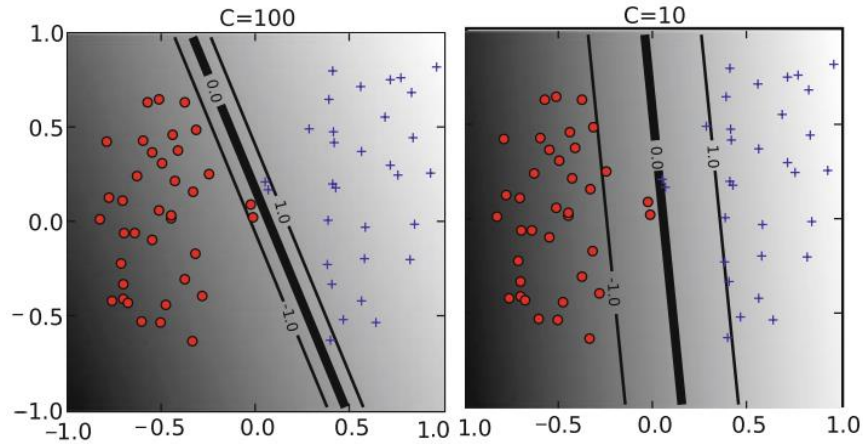


Figura 9. Efecto de la constante C en el máximo margen de separación [30].

En resumen una máquina de soporte vectorial sirve para resolver problemas de clasificación de dos clases aunque existen técnicas para llevarlo a un problema multiclases [32], estas máquinas encuentran un hiperplano que divide el espacio dimensional en dos, además encuentra la distancia máxima entre este hiperplano y los dos elementos más cercanos de ambas clases, aunque permitiendo algunos errores evitando así el sobre entrenamiento [31], y además con C como parámetro adicional para penalizar los errores que no son convenientes para el problema que se quiere resolver en cuestión.

3.4.3. Evaluación del modelo

Luego de construir un modelo y decidir sus parámetros es necesario evaluar su desempeño para elegir el más adecuado a la hora de clasificar los datos. Aunque hay muchas técnicas de evaluación, a continuación, se explican las más usadas para evaluar modelos de clasificación.

Exactitud en la clasificación

La exactitud es la razón entre el número de predicciones correctas entre el número de muestras (ec. 7). Esta métrica solo funciona bien si hay un número igual de muestras para cada clase.

$$exactitud = \frac{\text{Número de predicciones correctas}}{\text{Número total de predicciones}} [7]$$

Por ejemplo, si se considera el 98% de muestras de la clase 1 y 2% de la clase 2, luego el modelo puede obtener fácilmente el 98% de exactitud prediciendo únicamente las muestras de la clase 1, luego si cambia este porcentaje en la relación de las clases la exactitud cambiara, por lo que esta métrica por sí sola no da la información suficiente del rendimiento del modelo

y puede ser problemático cuando tiene mayor penalización no predecir la clase de menor porcentaje [33].

Matriz de confusión

La matriz de confusión permite una evaluación más completa del algoritmo, ya que permite observar como se distribuyen las predicciones del modelo respecto a los valores reales [34]. En la Figura 10 se muestra la forma general de una matriz de confusión.

		Verdad	
		Positivos	Negativos
Predicción	Positivos	Verdaderos Positivos	Falsos positivos
	Negativos	Falsos negativos	Verdaderos negativos

Figura 10. Matriz de confusión

De esta matriz se tienen cuatro términos importantes, **verdaderos positivos** son los casos predichos por el modelo como positivos que efectivamente los son; **falsos positivos** son los casos predichos por el modelo como positivos que no lo son; **falsos negativos** son los casos predichos por el modelo como negativos, pero no lo son y **verdaderos negativos** son los casos predichos como negativos que si lo son.

Con estos cuatro valores se pueden obtener dos métricas que son importantes para evaluar el modelo que son la especificidad y la sensibilidad.

La **sensibilidad** (ec. 8) da cuenta de las muestras predichas como positivas que efectivamente son positivas, es decir que tan bueno es el modelo para predecir los casos positivos [33].

$$Sensibilidad = \frac{VerdaderosPositivos}{FalsosNegativos + VerdaderosPositivos} \quad [8]$$

La **especificidad** (ec.9) corresponde a la proporción de puntos negativos que son tomados como positivos, es decir que tanto se equivoca el modelo prediciendo casos negativos [33].

$$Especificidad = \frac{VerdaderosNegativos}{VerdaderosNegativos + FalsosPositivos} \quad [9]$$

Generalmente dependiendo del propósito del modelo se le da mayor peso a una de estas dos métricas, por ejemplo, en diagnóstico es más importante que el modelo prediga correctamente los casos positivos.

Curva ROC

La curva ROC por sus siglas en ingles Receiver Operating Characteristic, es una representación gráfica de la sensibilidad versus el complemento de la especificidad(1-especificidad) para un clasificador binario y de acuerdo con cierto umbral, cuando se mide el área bajo esta curva se está midiendo que tan efectivo es el modelo distinguiendo entre las clases [35].

De una curva ROC se puede realizar tres tipos de interpretaciones, si el área bajo la curva es de 1(Figura 11) significa que el modelo distingue todas las muestras de la clase 1 como clase 1 y todas las muestras de la clase 2 como clase 2, esto no es tan bueno como parece, pues significa que el modelo esta sobre entrenado y solo funciona para los datos que se usaron en el entrenamiento, con nuevos datos podría no funcionar tan bien, en general un ROC de 0.8 a 0.9 es un buen resultado; si el área bajo la curva es de 0.5 (Figura 12) el modelo es incapaz de distinguir una clase de otra y si el área es de cero (Figura 13) significa que el modelo confunde las clases, la clase 1 la clasifica como clase 2 y la clase 2 como clase 1.

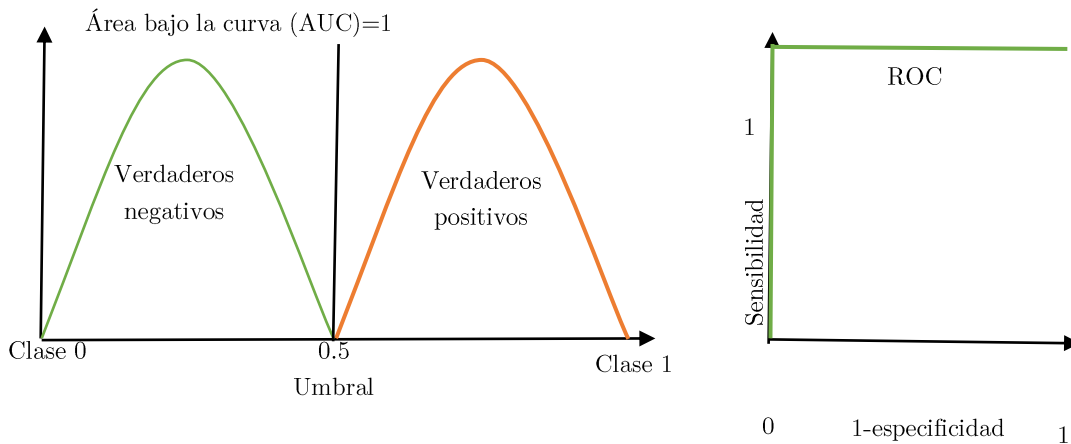


Figura 11. Curva ROC, con AUC=1

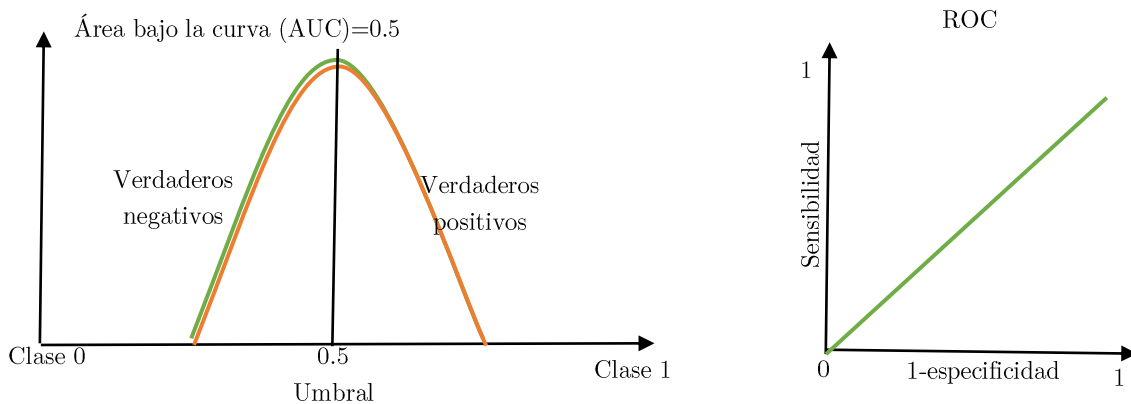


Figura 12. Curva ROC para un AUC=0.5

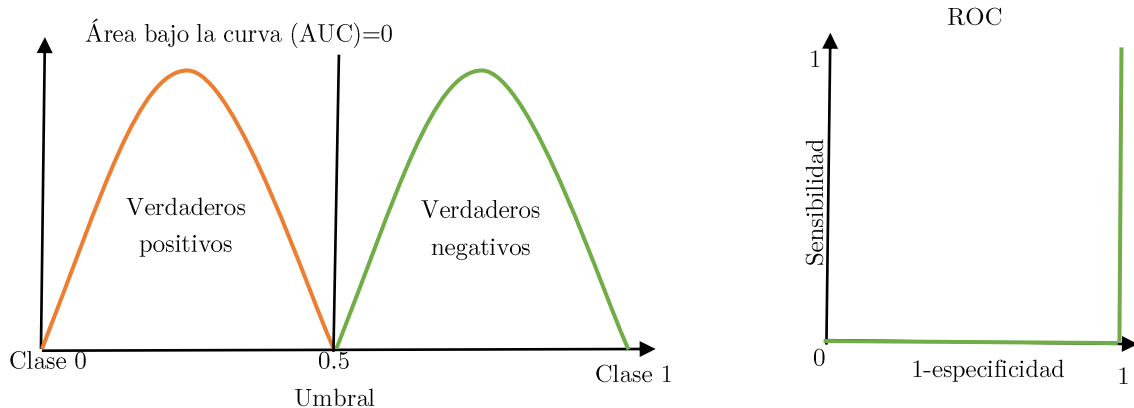


Figura 13. Curva ROC para una $AUC=0$

3.4.4. Scikit-learn

Scikit-learn es un módulo de Python que integra un amplio rango de algoritmos de *machine learning* de última generación para problemas de aprendizaje supervisado y no supervisado de mediana escala. Este paquete se enfoca en hacer accesible las técnicas de *machine learning* a personas no especializadas por medio del uso de un lenguaje de alto nivel. Con foco en un uso fácil, buen rendimiento, documentación y consistencia en la API [36].

Scikit-learn cuenta con varios algoritmos de clasificación, regresión y análisis como máquinas de soporte vectorial, bosques aleatorios, K-means y otros.

Para usar las máquinas de soporte vectorial que facilita Scikit-learn se usan las funciones SCV, NuSVC y LinearSVC, estas reciben parámetros como el kernel y el coeficiente C.

Scikit-learn también provee las herramientas para evaluar el desempeño del modelo, con funciones como *learning.curves*, *regressions.metrics*, entre otros. Para conocer todas las funcionalidades que provee esta librería [37].

4. METODOLOGÍA

A continuación, se presenta el diagrama (Figura 14) de la metodología general propuesta para el desarrollo del proyecto. Se plantean dos metodologías diferentes una para el desarrollo de la aplicación y una para el desarrollo del modelo, estas a su vez contienen subactividades que se desarrollaron con el propósito de cumplir los objetivos definidos. Una actividad principal inicia al realizar la anterior; excepto la documentación sobre el uso de software y la revisión bibliográfica, que se realizó de manera paralela a todas las actividades.

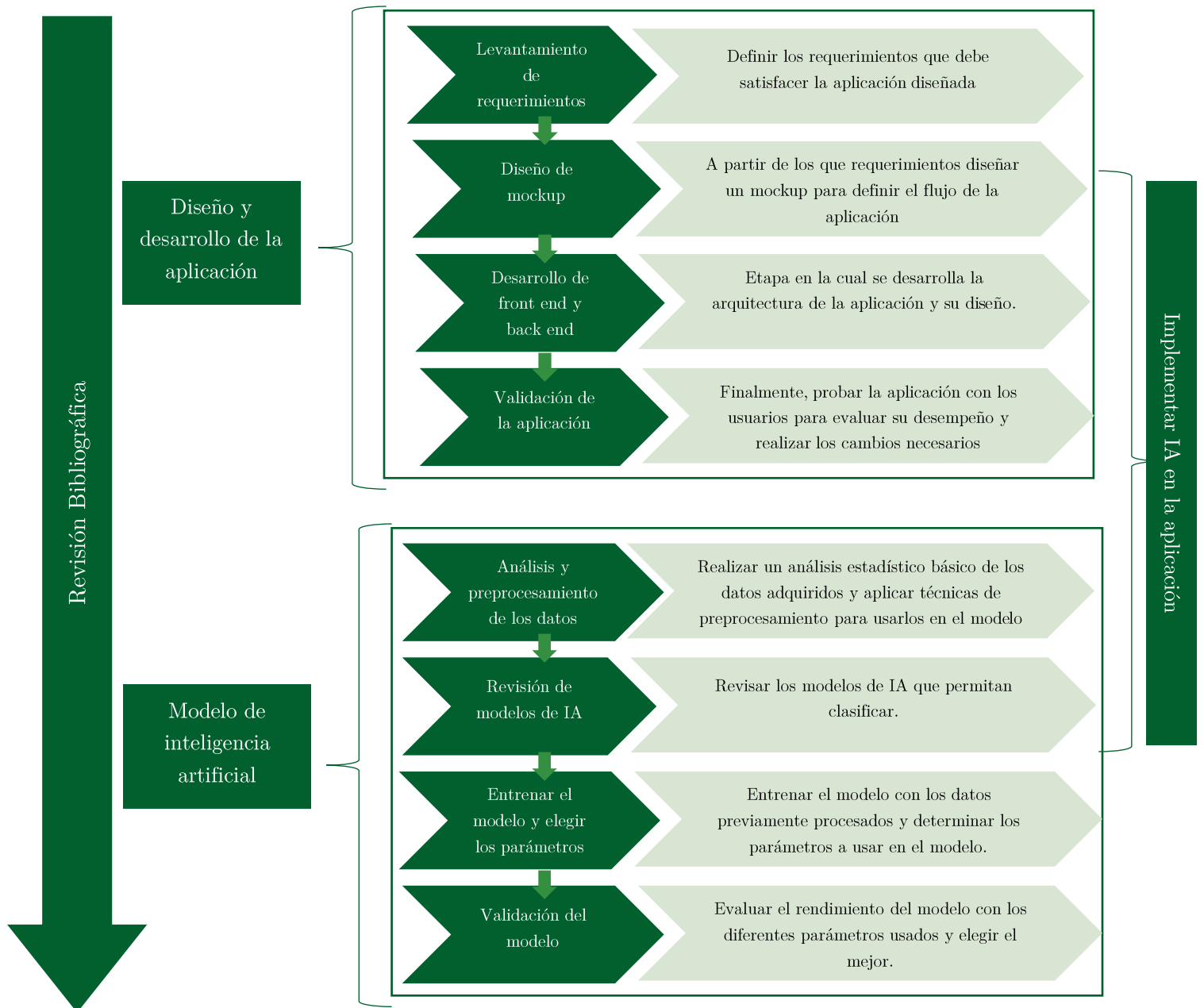


Figura 14. Metodología implementada en el proyecto

4.1. Diseño y desarrollo de la Aplicación Android

4.1.1. Levantamiento de requerimientos

Los requerimientos de la aplicación se eligieron en base al formulario definido inicialmente en el proyecto “*Desempeño y diseño de un modelo de dolor postquirúrgico en niños menores de 36 meses para la clasificación de dolor en comparación con la escala tradicional pediátrica FLACC*” que se puede ver en el anexo 1, para la recolección de datos, ya que el objetivo de la aplicación es facilitar el registro de estos datos y permitir su posterior consulta. Además, se hizo un refinamiento de las funciones en base a la opinión de los participantes en el proyecto.

Se tuvo en cuenta que la recolección de datos se realiza en 4 momentos, por lo que se debe registrar la valoración FLACC y las variables fisiológicas del paciente 4 veces, además se deben recolectar datos de información personal una vez y se deben poder consultar la información ingresada posteriormente.

En la Figura 15, se muestran los atributos que debe satisfacer la aplicación.

4.1.2. Diseño del mockup

Luego de tener definidos los requerimientos que debe satisfacer la aplicación se realizó un mockup (prototipo de la aplicación) [38], donde se determinó cuantas pantallas diferentes tendría la aplicación, qué debía contener cada una, como se relacionarían sus elementos y como se relacionaría una pantalla con otra. Para esto se hizo uso de una aplicación especial que permite prototipar estas funcionalidades.

4.1.3. Desarrollo del Front-end y Back-end

Una vez se tuvo claro el flujo de las actividades de la aplicación, se estableció la estructura de la base de datos, es decir cuantas tablas eran necesarias, qué atributos debían tener y como se tenían que relacionar estos atributos.

Luego con el diseño final se desarrolló la aplicación haciendo uso del entorno de desarrollo Android Studio, en este proceso se generó todo el código necesario para obtener el archivo .apk que permite la instalación en el dispositivo móvil donde será finalmente usada la aplicación.

4.1.4. Validación de la aplicación

Para validar la aplicación se realizó un sondeo donde se pidió evaluar la aplicación respecto a diseño, usabilidad, funcionalidad, en la Tabla 3 se muestra la estructura de la encuesta.

Esta evaluación la realizaron personas que tenían conocimiento previo de los datos y cómo funciona el proceso de recolección de estos.

Una vez realizada la evaluación se tomaron en cuenta las sugerencias para las mejorar la aplicación.

Tabla 3. Estructura de la encuesta realizada.

A continuación, se desarrolla una serie de preguntas que pretenden evaluar el funcionamiento, diseño y usabilidad de la aplicación. Por favor responder de manera objetiva las siguientes preguntas de acuerdo con la experiencia que tuvo con la aplicación, indique un nivel de satisfacción de 0 a 5. Donde 5 es muy satisfecho y 0 es insatisfecho.
Usabilidad 1) La navegación por la aplicación es intuitiva. 2) El tiempo de respuesta es adecuado 3) Los controles de la interfaz (botones, textos) funcionan correctamente 4) La aplicación funciona bien para el sistema operativo Android independiente del celular en que se esté usando
Diseño 1) Los textos son legibles 2) La interfaz gráfica facilita el uso de la aplicación 3) Es fácil entender cuáles son los datos por ingresar en la aplicación. 4) La manera como se presenta la información consultada hace más fácil su comprensión. 5) Es posible saber para cual institución fue diseñada la aplicación 5) De los elementos de la interfaz siente que hay algo fuera de lugar 6) La paleta de colores utilizada es adecuada
Funcionalidad 1) Es posible consultar un paciente de manera adecuada 2) El ingreso de los datos del paciente es de manera fácil y fluida 3) Se facilita el ingreso de los valores de la escala FLACC 4) Se facilita el ingreso de las variables fisiológicas 5) El uso de la aplicación facilita la recolección de datos del paciente

4.2. Modelo de Inteligencia Artificial

4.2.1. Preprocesamiento de datos

Para caracterizar los datos recolectados de acuerdo al protocolo definido en el proyecto “*Desempeño y diseño de un modelo de dolor postquirúrgico en niños menores de 36 meses para la clasificación de dolor en comparación con la escala tradicional pediátrica FLACC*”; se hizo un recuento de los datos, se graficaron los datos normalizados para detectar posibles patrones, se realizó un análisis estadístico básico de los datos, se encontró la correlación entre las variables, la media y la desviación estándar, con base en esta información se eligieron las variables que podrían ser más determinantes en la discriminación de las clases.

4.2.2. Revisión de modelos de inteligencia artificial

Se realizó una revisión de modelos de inteligencia artificial que sirvieran para la clasificación de datos, de acuerdo con las características de cada modelo y las características de los datos se eligió uno que se considerara apropiado para la clasificación de las clases.

4.2.3. Entrenamiento del modelo

Luego de elegir un modelo de clasificación que pudiera funcionar bien con los datos que se tienen, cambiando los parámetros del modelo se entrenó en repetidas ocasiones para obtener los mejores resultados posibles en la clasificación de los datos.

4.2.4. Validación del modelo

Para validar el modelo se encontraron los valores de exactitud, sensibilidad y especificidad a partir de la matriz de confusión, además se dibujaron las curvas ROC para cada uno de los modelos, de esta manera se evaluó qué tan efectivos eran los modelos separando las clases dolor y no dolor, verificando así si era posible detectar el dolor a partir de los signos vitales con ayuda del modelo seleccionado.

4.3. Implementar modelo de inteligencia artificial en la aplicación

Finalmente, luego de desarrollar la aplicación y tomando en cuenta el resultado del análisis de datos por medio de inteligencia artificial, se decidió no incluir en la aplicación el modelo evaluado.

5. RESULTADOS

5.1. Aplicación Android

A continuación, se presentan los resultados obtenidos en el mismo orden en que se desarrolló la metodología.

5.1.1. Levantamiento de requerimientos

Con la ayuda de los participantes del proyecto “*Desempeño y diseño de un modelo de dolor postquirúrgico en niños menores de 36 meses para la clasificación de dolor en comparación con la escala tradicional pediátrica FLACC*”; se definieron las tareas que realizará la aplicación, estas se muestran en la Figura 15, se decidió que para usar la aplicación no se necesitan permisos especiales por lo que cualquier usuario puede instalar y usar todas las funcionalidades de la aplicación, se definió que era importante que los valores de la FLACC se pudieran ingresar fácilmente, además debía contener la información sobre los atributos de la FLACC, también se estableció que la consulta de los pacientes se podría realizar de tres formas, mediante el número de historia clínica, el número de consecutivo que se le asigne o consultar todos los pacientes ingresados.

La cantidad total de datos del paciente a ingresar son: 8 manuales, la fecha automáticamente de acuerdo con el calendario del dispositivo móvil y el consecutivo también automáticamente de acuerdo con la cantidad de pacientes ingresados; para la FLACC serán 5 datos de acuerdo con sus atributos (sección 3.1.1) en 4 momentos, solo será almacenada la puntuación final y para las variables fisiológicas serán 6 datos en 4 tomas, para un total de 50 datos que debe contener la base de datos por cada registro de paciente.

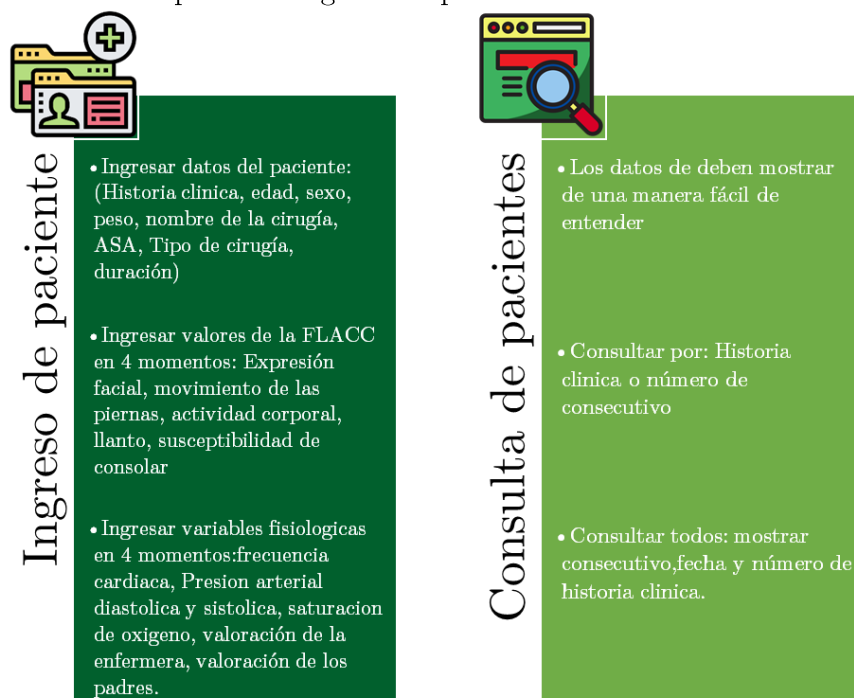


Figura 15. Definición de requerimientos

5.1.2. Diseño del mockup

A partir de los requerimientos se diseñó el flujo de las actividades y los elementos que cada pantalla iba a contener, además de cómo se relacionarían estas pantallas, en la Figura 16 se puede observar el mockup diseñado. A partir de este diseño preliminar se hicieron las adecuaciones pertinentes para mejorar el diseño final y pasar al desarrollo de la aplicación.



Figura 16. Mockup diseñado.

5.1.3. Desarrollo del Front-End y el Back-End

Se decidió usar una base de datos relacional con dos entidades (tablas) una que contiene todos los datos del paciente y otra donde están los cuatro valores de la FLACC para cada toma y las variables fisiológicas, estas entidades se relacionan mediante el número de consecutivo que se le asigna a cada paciente. En la Figura 17 se puede observar la estructura final de la base de datos.

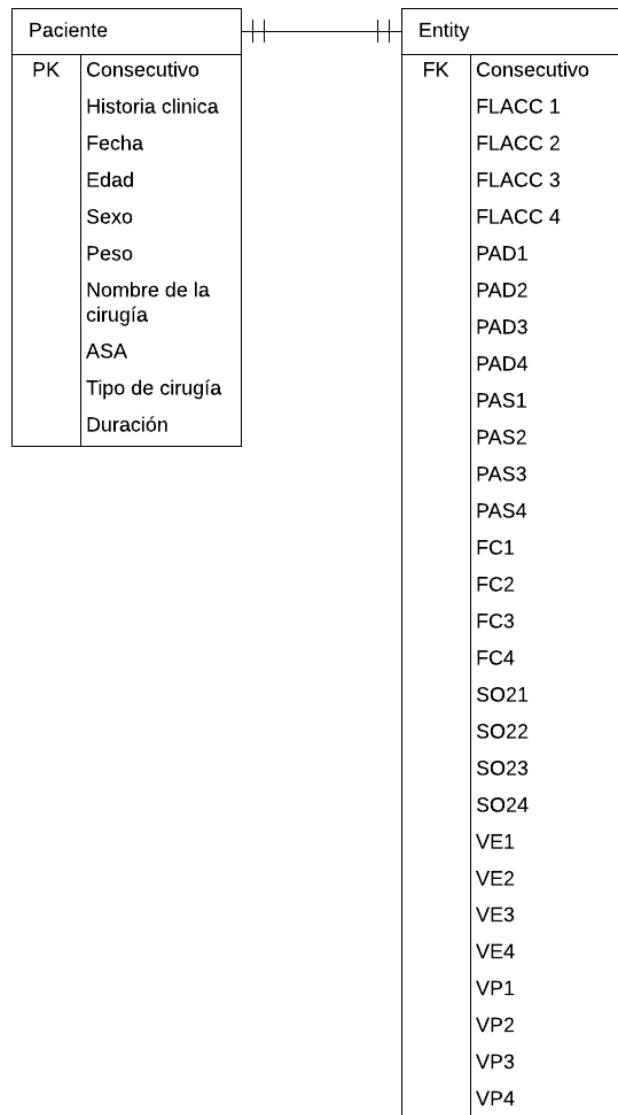


Figura 17. Diseño de la base de datos implementada.

Finalmente, a partir del diseño del mockup se añadieron 2 pantallas más que son consulta por consecutivo y consultar todos, la aplicación final cuenta con 16 pantallas, el nombre que se decidió para la aplicación fue FLACC.

En la Figura 18 aparece la pantalla inicial de la aplicación que cuenta con 2 botones uno para consultar los pacientes que ya se encuentran en la base de datos y otro para ingresar un nuevo paciente.



Figura 18. Pantalla principal de la aplicación.

Al presionar el botón ingreso de paciente aparece la pantalla que se muestra en la Figura 19, esta cuenta con nueve campos para ingresar datos y dos menús desplegables para elegir el sexo y el ASA además cuenta con un botón que al presionarlo lleva a la primera toma de la FLACC.

Figura 19. Ingreso datos del paciente.

En la Figura 20 se muestra la pantalla de ingreso de los valores de la FLACC, hay cuatro pantallas que son idénticas a estas para los cuatro momentos en los que se realiza esta toma, esta pantalla cuenta con cinco *radiogroups* que permiten elegir un valor de 0 a 2 y son mutuamente excluyentes, además cuenta con botones de información que permiten al usuario saber cuál valor es más apropiado para elegir y un botón de guardar que al presionarlo lleva al ingreso de las variables fisiológicas.



Figura 20. Ingreso de los valores de la FLACC

En la Figura 21 se muestra el ingreso de las variables fisiológicas, hay cuatro pantallas idénticas a esta para los cuatro ingresos, la pantalla cuenta con seis campos para ingreso de datos, dos botones de información y un botón guardar que al presionarlo lleva a la siguiente toma de la FLACC o en el caso del cuarto ingreso regresa de nuevo a la pantalla principal.

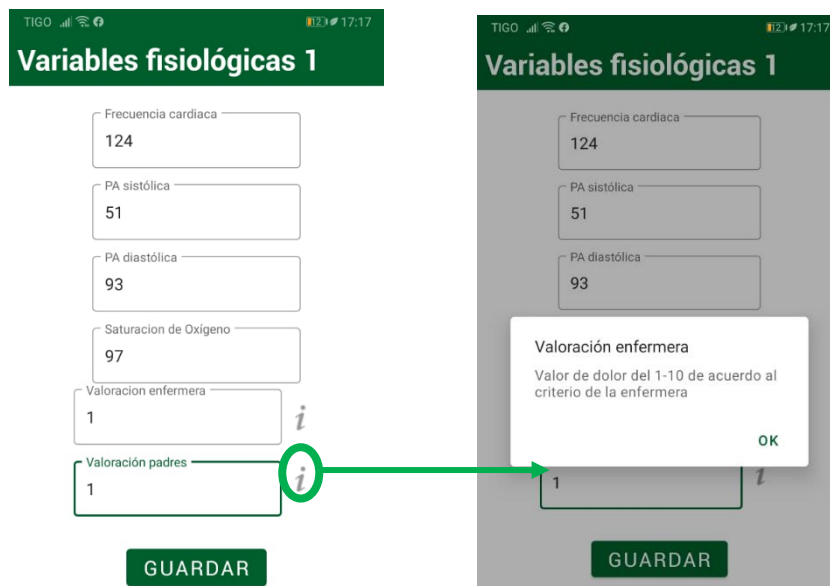


Figura 21. Ingreso de variables fisiológicas

Es importante anotar que todos los campos ingresados son validados, además de que todos son obligatorios y si alguno no se ingresa la aplicación genera mensajes de alerta y no permite que continúe la navegación.

Cuando en la pantalla principal se presiona el botón consulta de paciente aparece una pantalla como la que se muestra en la Figura 22, esta cuenta con tres botones uno para consultar todos los pacientes, uno para consultar con el número de historia clínica y el último para consultar con el número del consecutivo.



Figura 22. Consulta de pacientes.

Al presionar consultar por historia clínica aparece la pantalla de la figura 23.a que cuenta con un campo para ingresar el número de historia clínica y un botón para consultar, en la figura 23.b se muestra la pantalla a la que conduce el botón consulta por consecutivo, esta cuenta con un campo para ingresar el número de consecutivo y un botón para consultar, cuando se presiona este botón se dirige a la pantalla donde se muestran los datos.



Figura 23. a) Consulta por historia clínica. b) Consulta por consecutivo

Finalmente al presionar consultar todos aparece la pantalla mostrada en la figura 24.a, esta despliega una lista de todos los pacientes que se encuentran en la base de datos con su consecutivo, fecha e historia clínica, cuando se presiona cualquiera de estos pacientes se lleva a la pantalla mostrar datos (figura 24.b), esta pantalla contiene ocho campos donde se muestran datos, una tabla donde se muestra la información condensada de los cuatro registros de la FLACC y las variables fisiológicas y un botón para regresar a la pantalla principal.

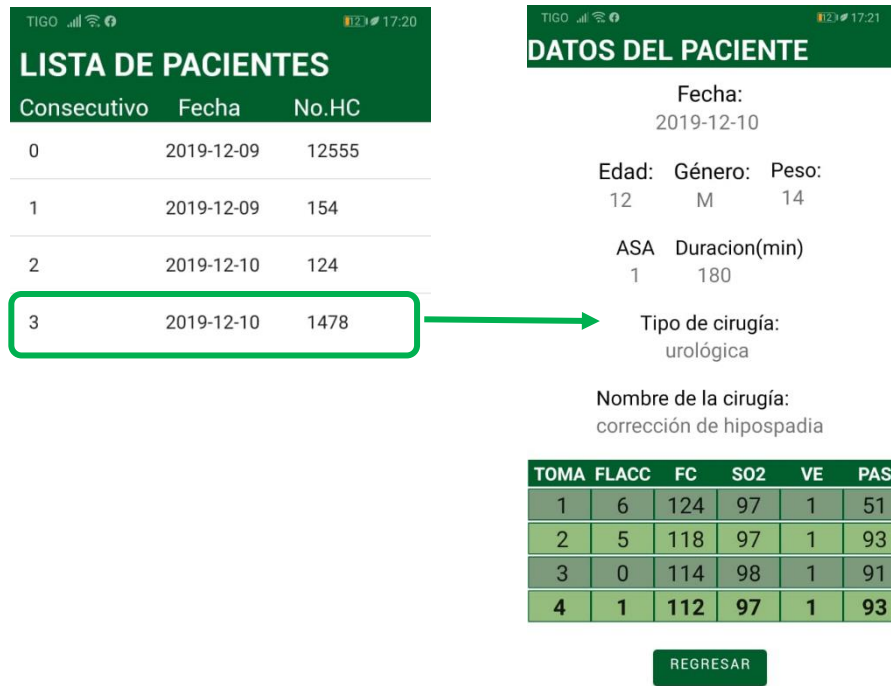


Figura 24. a) Listado de pacientes. b) Datos del paciente

5.1.4. Validación de la aplicación

La encuesta implementada se puede consultar en el siguiente [enlace](#), esta fue contestada por 7 personas familiarizadas con el proyecto y los objetivos que debía cumplir la aplicación.

En las figuras 25 y 26 se pueden observar los resultados obtenidos. La figura 25 presenta un promedio de la calificación de cada pregunta por categoría para cada voluntario que evaluó la aplicación. Mientras que la figura 26 presenta el promedio de calificaciones de todos los voluntarios por cada pregunta realizada.

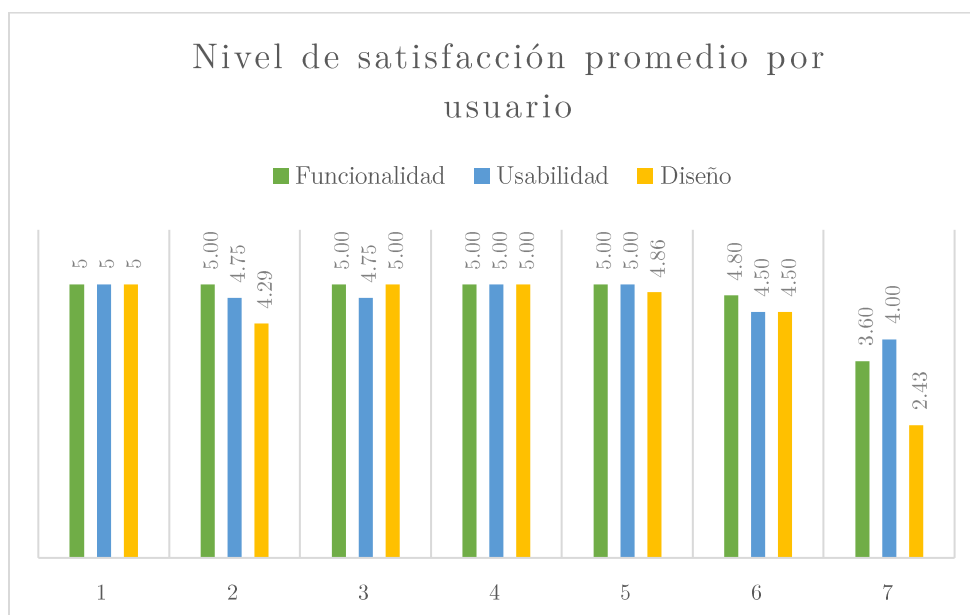


Figura 25. Nivel de satisfacción promedio por usuario.

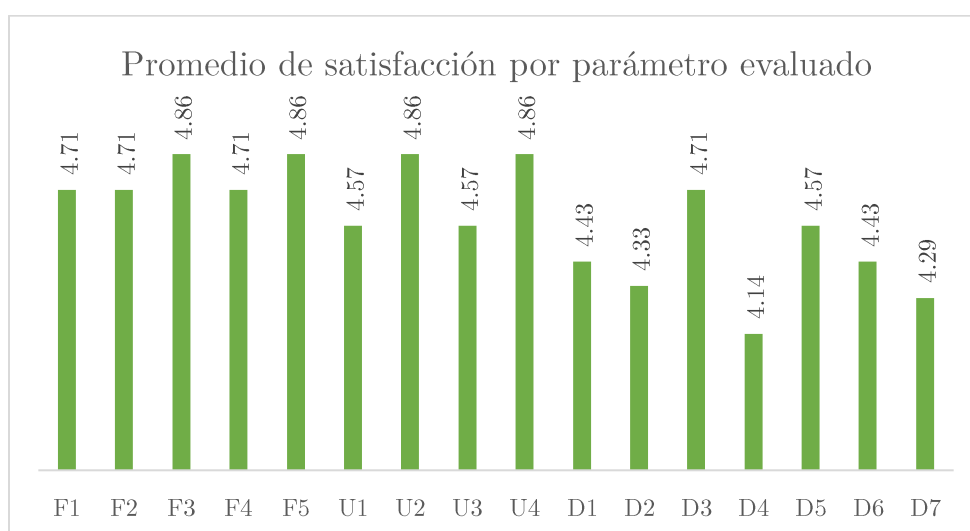


Figura 26. Promedio de satisfacción por parámetro evaluado

5.2. Modelo de Inteligencia Artificial

5.2.1. Preprocesamiento de los datos

Descripción de los datos

Los datos usados para el análisis son los recolectados en el Hospital San Vicente Fundación en el proyecto “*Desempeño y diseño de un modelo de dolor postquirúrgico en niños menores de 24 meses para la clasificación de dolor en comparación con la escala tradicional pediátrica FLACC*”, los datos son recolectados en cuatro momentos para cada niño que ha sido sometido a un procedimiento quirúrgico de acuerdo con los criterios establecidos en el proyecto, todos son niños entre 0 y 36 meses, para el modelo se usaron 78 niños, cada sujeto tiene cuatro tomas de datos, por lo que en total son 312 datos. En la Tabla 4 se muestra como están divididos estos datos de acuerdo con la clasificación FLACC (sección 3.1.1).

Tabla 4. Cantidad de sujetos clasificados según la FLACC.

Tipo de dolor según FLACC	Cantidad de datos
Dolor severo	8
Dolor moderado	29
Dolor leve	64
Sin dolor	211

La distribución porcentual de las clases se observa en la Figura 27.

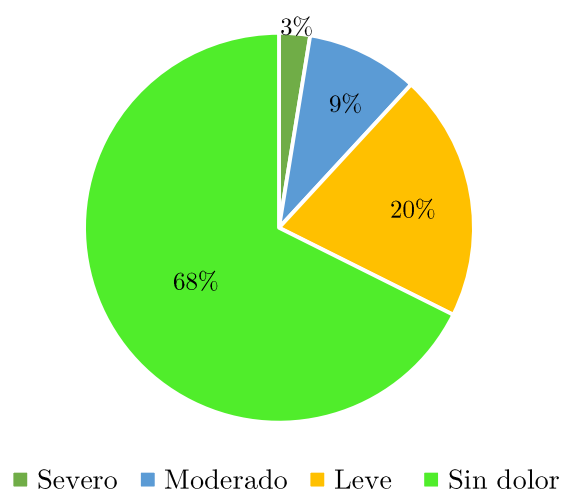


Figura 27. Porcentajes de las clases de dolor.

En la Figura 28 se presenta la matriz de correlación entre las variables, frecuencia cardiaca, presión arterial diastólica, presión arterial sistólica y saturación de oxígeno, todas normalizadas con el método de mínimo-máximo (sección 3.4.1).

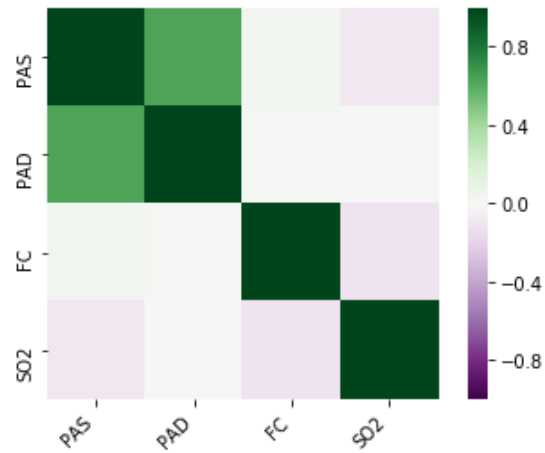


Figura 28. Matriz de correlación

En la Tabla 5 se presenta la media y desviación estándar para las variables normalizadas.

Tabla 5. Media y desviación estándar de las variables normalizadas.

	PA sistólica	PA diastólico	Frecuencia cardiaca	Saturación de oxígeno
Media	0.62	0.52	0.63	0.88
Desviación estándar	0.13	0.10	0.16	0.09

En la Figura 29 se muestra la distribución de las variables para la clase sin dolor versus la clase dolor leve.

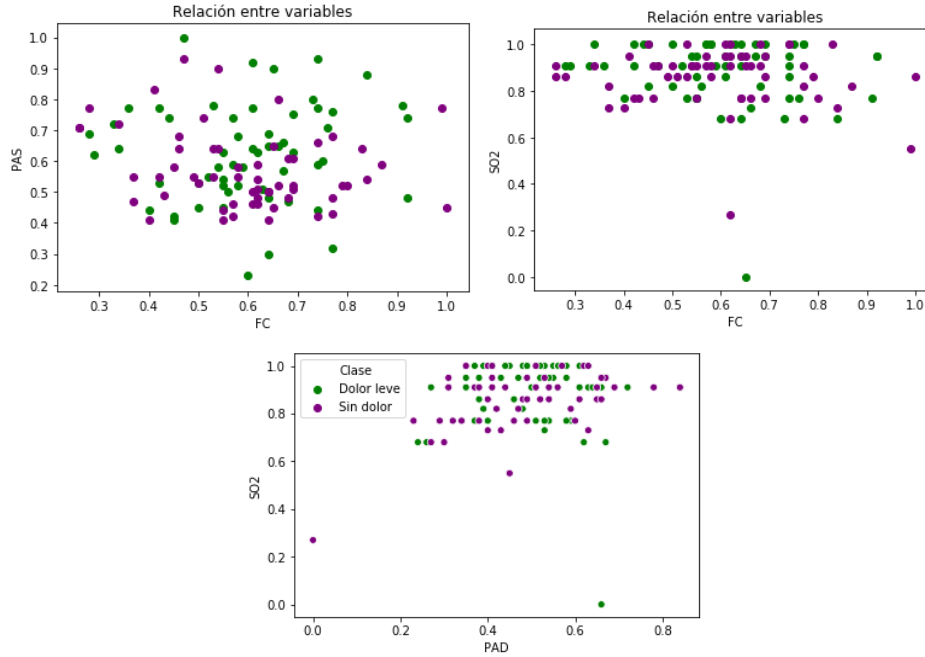


Figura 29. Distribución de los datos para la clase de no dolor (morado) vs dolor leve (verde). a) Presión arterial sistólica vs frecuencia cardiaca. b) Frecuencia cardiaca vs Saturación de oxígeno. c) Presión arterial diastólica vs Saturación de oxígeno.

En la Figura 30 se muestra la distribución de las variables para la clase sin dolor vs la clase dolor moderado.

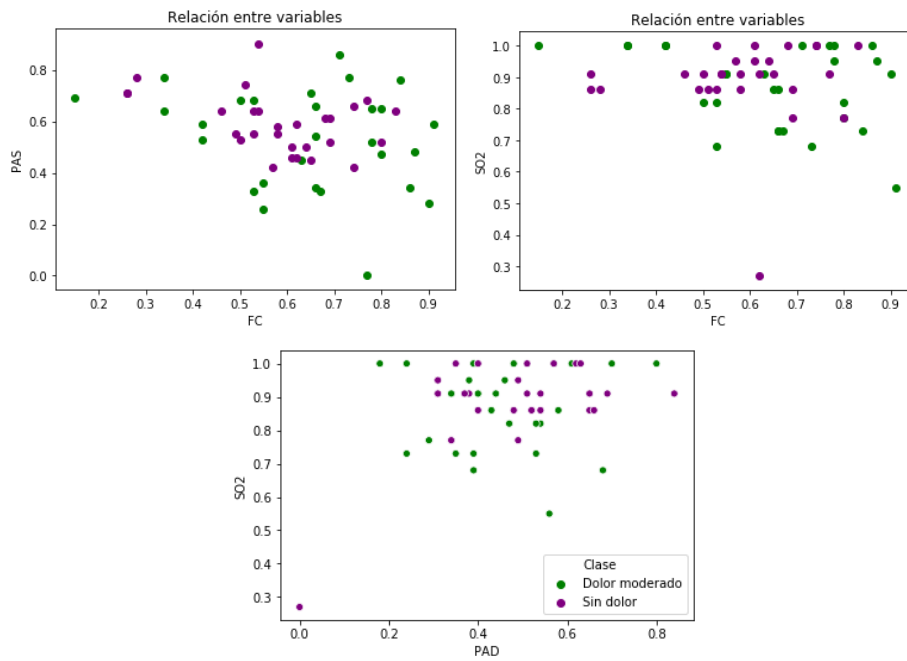


Figura 30. Distribución de los datos para la clase de no dolor vs dolor moderado. a) Presión arterial sistólica vs frecuencia cardiaca. b) Frecuencia cardiaca vs Saturación de oxígeno. c) Presión arterial diastólica vs Saturación de oxígeno.

Debido a que solo hay 8 sujetos en la clase dolor severo estas gráficas se omiten debido a que no es una muestra suficiente para observar algún patrón.

También se graficó la distribución de no dolor versus cualquier tipo de dolor (leve, severo o moderado) esto se muestra en la Figura 31.

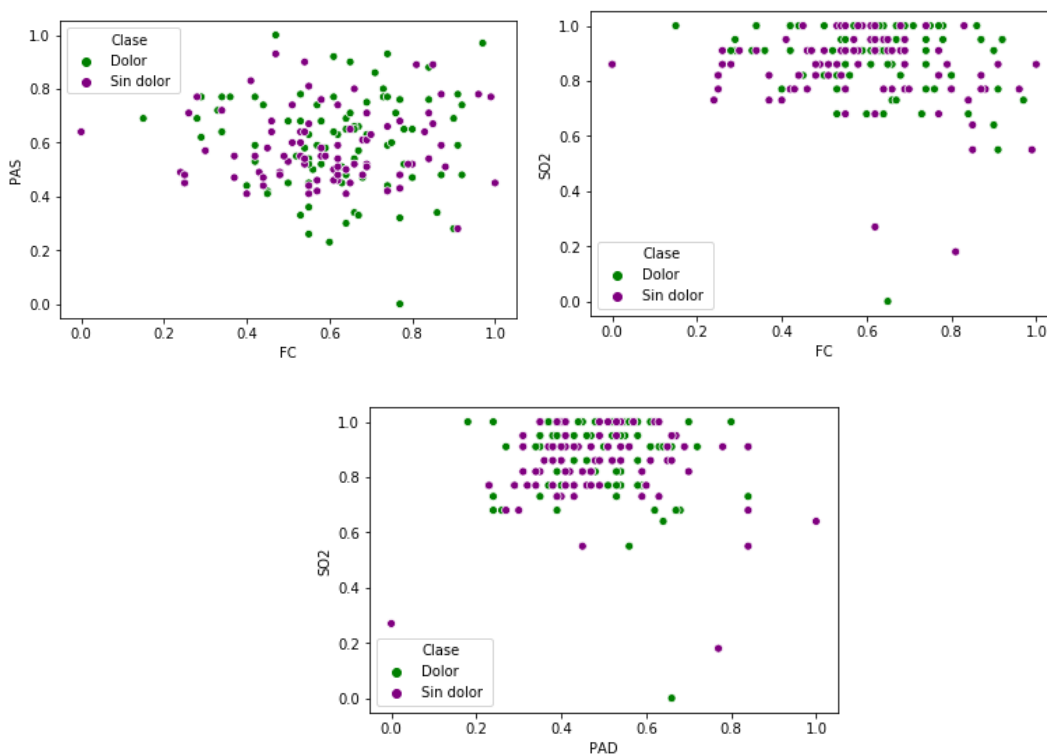


Figura 31. Distribución de los datos para la clase de no dolor vs dolor. a) Presión arterial sistólica vs frecuencia cardiaca. b) Frecuencia cardiaca vs Saturación de oxígeno. c) Presión arterial diastólica vs Saturación de oxígeno

Además de esto según los rangos normales para las variables fisiológicas (Tabla 2) se determinó que sujetos tenían las variables alteradas por encima de los rangos normales ya que esta es la alteración que se ha probado puede ser un indicador del dolor. En la Tabla 6 se muestran estos porcentajes.

Tabla 6. Porcentaje de alteración en las variables fisiológicas para cada clase de dolor.

Tipo de dolor	Frec. cardiaca	PA sistólica	PA diastólica	Saturación de oxígeno
Leve	14.1%	39%	50%	40%
Moderado	27.6%	24.1%	34.5%	44.8%
Severo	25%	75%	50%	37.5%

También se analizó cuantas variables fisiológicas se alteran al mismo tiempo para cada clase de dolor, estos porcentajes se muestran en la Tabla 7.

Tabla 7. Cantidad de variables alteradas por cada clase de dolor.

Tipo de dolor	Ninguna variable alterada	Una variable alterada	Dos variables alteradas	Tres o más variables alteradas
Sin dolor	46%	29.4%	16.6%	5.2%
Leve	18.8%	32.8%	28.1%	15.6%
Moderado	24.1%	20.7%	48.3%	3.4%
Severo	0%	37.5%	25%	37.5%

5.2.2. Revisión, entrenamiento y validación del modelo

Debido a lo analizado en el preprocesamiento de los datos se decidió usar máquinas de soporte vectorial debido a que es un clasificador fácil de implementar y que funciona bien en datasets que no contienen mucha cantidad de datos, además que es preciso y no se necesita que los datos sean separables linealmente debido al uso de kernels [39].

Para evitar un problema multiclases se realizaron varias clasificaciones binarias, para el entrenamiento y evaluación del modelo se hizo uso del módulo de Sckit-learn SVM [37] de Python, que permite el uso de modelos pre-hechos pero deja libertad sobre los parámetros a usar, es decir el kernel y los pesos de cada clase.

El modelo se usó para clasificar entre diferentes tipos de dolor (leve, moderado, severo, sin dolor) a partir de las variables fisiológicas y usando como etiqueta el valor de la escala FLACC, debido a que en principio el problema es una clasificación de múltiples clases, se convirtió en un problema binario con las clases de la Tabla 8. Las variables que se tuvieron en cuenta para el entrenamiento del modelo fueron: frecuencia cardiaca, presión arterial, saturación de oxígeno y si el paciente tenía caudal.

Tabla 8. Clases binarias definidas.

	Clase 0	Clase 1
Modelo 1	Dolor leve + No dolor	Dolor moderado + dolor severo
Modelo 2	No dolor	Dolor leve + dolor severo + dolor moderado
Modelo 3	Dolor leve	Dolor severo + dolor moderado

Debido al desbalanceo de clases en general (más datos de la clase sin dolor) para cualquiera de las clasificaciones en la Tabla 8, se usaron dos técnicas de balanceo, se crearon nuevos datos para las clases minoritarias(dolor severo y moderado), a partir de la media y la desviación estándar (Tabla 5) y además se eliminaron datos de la clase mayoritaria(sin dolor). En la Tabla 9 se presentan la cantidad de datos totales de entrenamiento y prueba usados para cada clasificación, aclarando que el 20% son para la prueba y en esta no se usan datos generados y el 80% para el entrenamiento del modelo, todos los datos fueron normalizados entre 0 y 1.

Tabla 9. Cantidad de datos usados para el entrenamiento.

Modelo	Clase 0	Clase 1
Dolor leve + No dolor VS Dolor moderado + Dolor severo	51	31 _{BD} 20 _G
No dolor VS Dolor leve + dolor severo + dolor moderado	88	88 _{BD}
Dolor leve VS Dolor severo + dolor moderado	57	31 _{BD} 11 _G

BD: Datos tomados de la base de datos. G: Datos generados a partir de técnicas de balanceo.

Para cada problema de clasificación binario definido se cambiaron dos parámetros en la máquina de soporte vectorial, el tipo de kernel y luego con cada kernel se varía el valor de C (peso de las clases), en la Tabla 10 se muestran los valores usados.

Tabla 10. Parámetros usados en la máquina de soporte vectorial.

Tipo de kernel	Valor de C
Sigmoid	1
RBF	5
Polinomial grado 3	10

Para el primer problema de dos clases (Dolor leve + No dolor **VS** Dolor moderado + Dolor severo), se obtuvo que para cualquier variación de los parámetros el resultado era el mismo, con todos se obtuvo una exactitud del 50%, una sensibilidad del 0% y una especificidad del 100%, la matriz de confusión es la mostrada en la Figura 32.

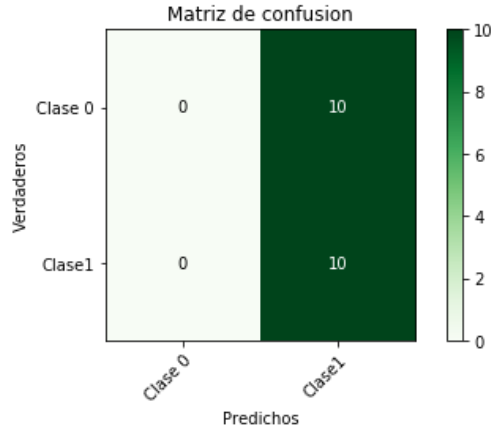


Figura 32. Matriz de confusión primera clasificación binaria.

Para el segundo modelo (no dolor vs cualquier tipo de dolor) para los kernel *RBF* y *sigmoid*, y cualquier valor de C se obtuvo una exactitud del 62.5%, una sensibilidad del 50% y una especificidad del 75%, la matriz de confusión es la mostrada en la Figura 33.

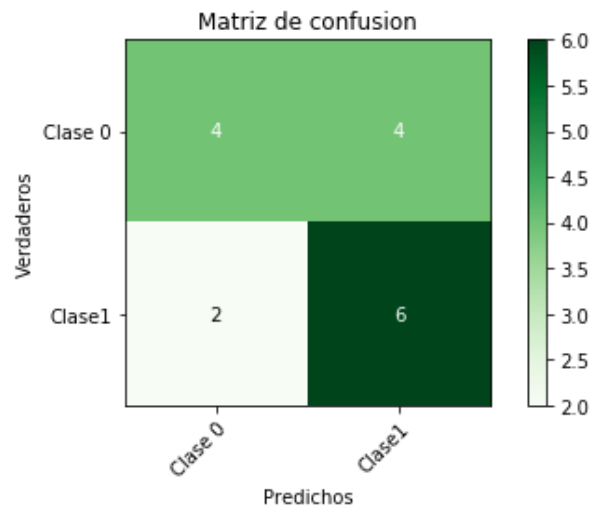


Figura 33. Matriz de confusión segunda clasificación binaria con Kernel RBR y sigmoid.

Luego con un kernel polinomial de grado 3 y para cualquier valor de C se obtuvo una exactitud del 75%, una especificidad del 50% y una sensibilidad del 100%, la matriz de confusión es la mostrada en la Figura 34.

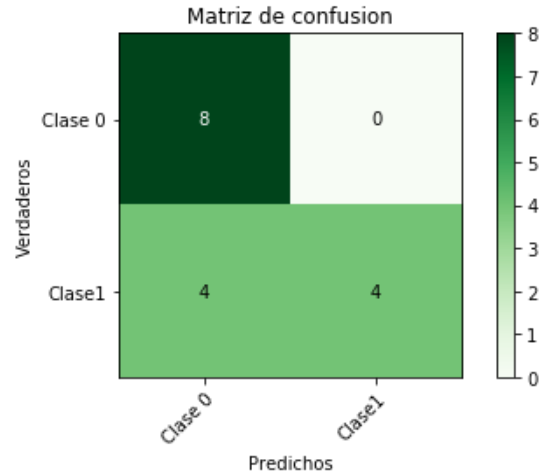


Figura 34. Matriz de confusión segunda clasificación binaria con Kernel polinomial de grado 3.

Finalmente, para la tercera clasificación (dolor leve vs dolor severo + dolor moderado) para cualquier tipo de kernel y valor de C, se obtuvo una exactitud del 50%, una sensibilidad del 0% y una especificidad del 0%, la matriz de confusión es la mostrada en la Figura 35.

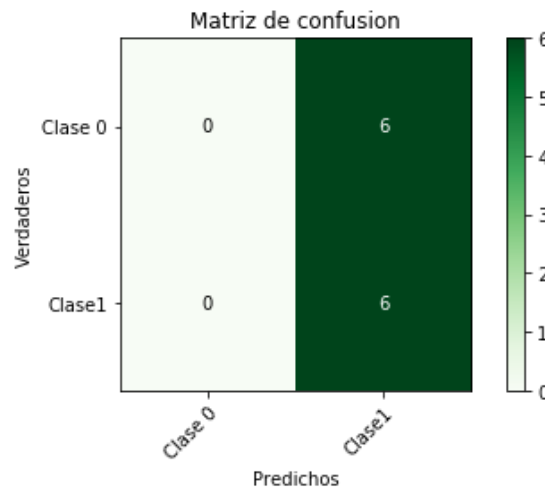


Figura 35. Matriz de confusión para la tercera clasificación.

6. ANÁLISIS DE RESULTADOS

6.1. Diseño y desarrollo de la Aplicación Android

6.1.1. Diseño del mockup

El mockup fue la herramienta que facilitó el prototipado inicial de la aplicación, de esta manera fue más fácil observar cómo podría ser una versión final de la aplicación aún sin haber desarrollado el código de funcionamiento, pudiendo hacer cambios rápidos en el diseño y la relación entre las pantallas, además se decidieron los elementos que iban a interactuar y como iban a ser estas interacciones. El mockup permitió una revisión preliminar y decidir finalmente los elementos funcionales y estéticos de la aplicación, es una herramienta muy útil que evita retrocesos en la programación de la aplicación.

6.1.2. Desarrollo de FrontEnd y BackEnd

Se desarrolló la aplicación para las tareas de consulta e ingreso de pacientes de tal manera que el usuario no necesitara ningún tipo de capacitación en su uso, sino que fuera intuitivo y de fácil manejo.

El entorno de desarrollo Android Studio permitió total libertad sobre el diseño y funcionamiento de la aplicación, además el uso de la librería *material design* [40] aportó mucho valor en el diseño visual de la aplicación ya que contiene elementos más atractivos visualmente para el usuario, lo cual es importante para que se sientan cómodos y satisfechos con el resultado final.

El uso de SQLite (sección 3.2.1) permitió que la base de datos que se iba modificando en cada uso de la aplicación fuera almacenada directamente en el disco del dispositivo en el cual estaba instalada la aplicación, facilitando así su posterior consulta, una limitación de esto es que la base de datos no se sincroniza en diferentes dispositivos así que para la recolección de datos del proyecto se limitaría el ingreso de datos a un solo dispositivo. La base de datos posee un valor único que es el consecutivo de cada paciente que permite relaciones los datos iniciales: historia clínica, fecha, nombre de cirugía, sexo, ASA, peso y tipo de cirugía con los demás datos que son las variables fisiológicas y los valores de la FLACC, teniendo así las variables de diferente naturaleza en todas tablas relacionales.

El uso de botones de información como los observados en las Figura 21 y 22 facilitan el ingreso de datos correctos según lo estipulado por la FLACC.

Todos los campos de texto son validados antes de ingresarlos en la base de datos, para esto se usan entradas de texto y número dependiendo del tipo de variable de cada campo, además se verifica que todos los campos estén llenos sino se muestra un mensaje de alerta, se encontró

que todas estas validaciones son necesarias para no generar errores en la consulta o el ingreso de los datos, también se valida que las historias clínicas no se repitan, ya que este es un campo único para cada paciente.

Finalmente, la aplicación diseñada cumple con todos los requerimientos planteados inicialmente, facilitando la recolección de los datos y el posterior análisis de estos ya que es fácil acceder al archivo SQL que los contiene desde el dispositivo.

6.1.3. Validación de la aplicación

En la Figura 25, que muestra el promedio de satisfacción por usuario, se observó que, en general, el promedio estuvo entre un rango de 4 a 5 en nivel de satisfacción. Es decir, los usuarios manifiestan estar satisfechos y muy satisfechos con relación a los parámetros evaluados.

El usuario 7, a diferencia del resto de usuarios, mostró una menor satisfacción, especialmente con el diseño sugiriendo que se cambie la forma como se muestran los datos ya que no le parece que sea fácil de entender la información presentada.

De la figura 26 donde se muestra el promedio de satisfacción por parámetro evaluado se observa que las calificaciones más altas (mayores a 4.7) son para los ítems de funcionalidad, por lo que en este aspecto la aplicación no requiere muchos cambios, mientras que las evaluaciones más bajas son para los ítems de diseño especialmente en aspectos como el texto(D1), la paleta de colores(D2) y la manera como se muestran los datos consultados (D4) por lo que sobre estos aspectos sería necesario realizar cambios para mejorar la experiencia del usuario.

Todos los usuarios sugirieron una sincronización de los datos ingresados desde cualquier dispositivo para que cualquier usuario pueda tener acceso a la información completa y no solo a la ingresada desde su dispositivo. Además de otros cambios menores en la aplicación como el alineamiento de los textos y botones, para que esta sea más prolija y mejor visualmente. No se obtuvieron sugerencias en cuanto a la usabilidad.

6.2. Modelo de Inteligencia Artificial

6.2.1. Preprocesamiento de los datos

De la Tabla 4 y la Figura 27 donde se muestra la distribución de los datos en las 4 clases de dolor, se observa un desbalanceo en las clases bastante pronunciado, la clase sin dolor representa 68% del total de los datos, mientras que la clase de dolor severo representa apenas un 3%, además de los 312 datos totales hay un 5% con variables faltantes, por lo que no pueden ser usados en el modelo.

De la Figura 28 donde se muestra la matriz de correlación de los datos, se puede observar que en general hay una correlación baja entre todas las variables, colores cercanos al blanco, solo se encuentra una correlación alta entre la presión sistólica y diastólica, algo esperado debido a que son variables que miden un fenómeno fisiológico relacionado en dos momentos diferentes, por lo que si una sube la otra también y viceversa. La baja correlación entre variables muestra que provienen de fuentes de información diferentes y por tanto todas pueden ser usadas en el modelo, ya que cada una aporta información individual. Caso contrario con las presiones diastólica y sistólica, por lo que se decidió eliminar del modelo la sistólica.

En las Figura 29,28,29 donde se representan las distribuciones de los datos para cada variable, no se observa un patrón de distribución de las variables, al parecer el valor de dolor no afecta considerablemente las variables fisiológicas o por lo menos no las agrupa hacia un mismo lugar, haciendo muy difícil que un modelo diferencie las clases.

De la Tabla 5 donde se muestra la media y desviación estándar para cada una de las variables, se puede inferir que la frecuencia cardiaca puede no ser un indicador del dolor, en todas las clases es la que presenta menor porcentaje de alteración con un 14.1% para dolor leve, 27.5% para moderado y 25% para severo, mientras que una presión arterial alta en términos generales tanto la diastólica como la sistólica se ven más relacionadas con alguna presencia de dolor ya que en alrededor de un 50% de los pacientes esta se ve alterada, además un saturación de oxígeno baja también tiene un porcentaje significativo alrededor del 40% en pacientes con dolor, se debe tener en cuenta que la muestra para paciente con dolor severo es muy pequeña apenas 8 sujetos de un total de 312, por lo que los porcentajes para esta clase probablemente no sean representativos de la realidad.

En la Tabla 67 se puede observar que a medida que aumenta el dolor más variables se ven alteradas, para la clase sin dolor es más común que no haya ninguna variable alterada, el 46% de los sujetos tiene las variables en rangos normales y solo el 5.2% todas las variables alteradas, para los sujetos con dolor leve es más común que tengan una(32.8%) o dos variables alteradas(28.1%), para sujetos con dolor moderado lo más común es que por lo menos tengan 2 variables alteradas con 48.3% y finalmente aunque de los sujetos con dolor

severo hay muy pocas muestras es más común que tengan una o más variables alteradas, 0% para ninguna variable alterada, estos porcentajes son importantes pues según la cantidad de variables alteradas en un sujeto se puede inferir si el paciente tiene algo de dolor.

Los porcentajes presentados en la Tabla 5 y 6 describen como sufren alteraciones las variables de acuerdo con el dolor que presenta el sujeto, esto demuestra una relación entre las alteraciones y el dolor, lo que es un buen indicio para usar un modelo de inteligencia artificial que intente encontrar estos patrones, incluso con estas relaciones se puede inferir cuales variables tienen más peso en el momento de predecir el dolor como es el caso de una saturación de oxígeno baja junto con una presión arterial alta, ya que alrededor de un 40%-50% de sujetos con dolor moderado o severo presentan alteraciones en estas medidas.

6.2.2. Revisión, entrenamiento y validación del modelo

Para el entrenamiento del modelo se decidió tener en cuenta si al paciente se le había aplicado la caudal (técnica anestésica regional que consiste en la inyección de anestésicos locales en el componente sacro del espacio epidural [41]), ya que esto disminuye el dolor que el niño pueda sentir, por lo que es una variable relevante y que se considero podía aportar información adicional al modelo.

Los modelos de la primera y tercera clasificación no pueden distinguir entre las clases debido a que su exactitud es de un 50%, lo que significa que solo reconoce una clase y realmente no diferencia una clase de la otra, por lo que estos modelos no pueden ser usados para la discriminación de las clases.

El modelo para la segunda clasificación en el cual se usó un kernel polinomial es el que mejores resultados presenta con una exactitud del 75%, pero este porcentaje tampoco es suficientemente confiable para discriminar entre las clases.

Aun así, esto confirma lo que se observaba desde el análisis inicial de los datos, aparentemente según su distribución no hay una forma de separarlos fácilmente, además los resultados indican que podría ser más fácil clasificar entre cualquier tipo de "dolor" (leve, moderado o severo) y "no dolor", debido a que al clasificarlos de la forma planteada no hubo necesidad de balancear las clases creando nuevos datos.

Otro problema que puede estar afectando la clasificación es el número de datos disponibles, aunque de la clase "no dolor" presenta una cantidad considerables de datos, las otras clases tienen muy pocas muestras (en comparación con "no dolor") dificultando así que el algoritmo encuentre algún patrón en los datos que le permita separar claramente las clases.

7. CONCLUSIONES

Para lograr el objetivo general del proyecto que buscaba desarrollar un aplicativo que sirviera como herramienta para evaluar el dolor postquirúrgico en niños no comunicativos del Hospital Universitario San Vicente Fundación, se dividió la metodología en dos secciones una parte dedicada al desarrollo de la aplicación y otra dedicada a la evaluación de un modelo de inteligencia artificial para la predicción del dolor a partir de las variables fisiológicas.

Con respecto a la aplicación se puede concluir que se logró un diseño que cumpliera con el levantamiento de los requerimientos establecidos inicialmente junto con los otros participantes del proyecto, haciendo uso de herramientas que provee el entorno de desarrollo Android Studio y librerías como material design. Se implementó una base de datos relacional que facilitó la interacción con los datos, permitiendo realizar de manera óptima la consulta y escritura de datos, esto gracias al uso de SQLite.

En general los potenciales usuarios de la aplicación estuvieron satisfechos con el resultado, especialmente con los parámetros evaluados de usabilidad y funcionalidad, además propusieron mejoras que son útiles para que la aplicación escale y pueda ser usada fácilmente en todo el hospital.

Como trabajo futuro se propone implementar las sugerencias realizadas por los usuarios como permitir la sincronización de la base de datos desde diferentes dispositivos, mejorar la forma como se presentan los datos cambiando los colores y la distribución de la tabla donde se muestra, evitar que botones se superpongan a textos u otros elementos de interés en la aplicación, todo esto con el fin de que la aplicación sea más amigable con el usuario y satisfactoria en su uso.

En la segunda parte de este trabajo, se realizó un diseño y análisis de un posible modelo para predecir la presencia de dolor a partir de las variables fisiológicas, con el uso de máquinas de soporte vectorial los rendimientos no son los esperados, esto conduce a concluir que las variables fisiológicas no son tan determinantes por sí solas para decidir si un paciente cursa con dolor o no, también el desbalanceo de las clases hace que el modelo se incline por la clase cero, es decir sin dolor, una mayor cantidad de datos de las otras clases puede ser una solución para mejorar el rendimiento del modelo, o usar otras técnicas y tipos de modelos de clasificación con inteligencia artificial, o incluso aprendizaje no supervisado para encontrar patrones en las variables fisiológicas y detectar de esta manera si están relacionadas de alguna forma con el dolor que siente el paciente, con el tipo de analgesia suministrada o el tipo de cirugía.

Aunque se ha demostrado que el dolor está relacionado con un cambio en las variables fisiológicas del paciente, también hay otras situaciones sobre todo en el área postoperatoria que podría estar alterando estas variables, como la ansiedad que siente el paciente al encontrarse solo en un lugar desconocido al despertar de la cirugía (recordando que los pacientes en este proyecto son infantes de 0 a 36 meses), o los ayunos prolongados a los que se deben someter antes de los procedimientos quirúrgicos, además que los medicamentos usados para tratar el dolor también alteran estos signos vitales, por lo que, con los resultados obtenidos con el modelo y lo observado en la recolección de los datos, se concluye que los signos vitales por si solos no pueden determinar el dolor que siente un paciente, pero quizás si se asocian a otras variables como el tipo de cirugía o el ASA, puedan ser más determinantes para predecir el dolor del paciente.

8. BIBLIOGRAFÍA

- [1] J. Gómez, «Abordaje del dolor en pediatría.,» 2013. [En línea]. Available: <http://www.pfizerpro.com.co/sites/g/files/g10022836/f/201509/Consenso%20colombiano%20en%20>. [Último acceso: 16 12 2019].
- [2] C. Gancedo y F. Malmierca, «Dolor en pediatría,» Ergon, Madrid, 2008.
- [3] World Health Organization, guidelines on the pharmacological treatment of persisting pain in children with medical illnesses, Geneva: World Health Organization, 2012.
- [4] O. Cañas, T. Pabon y L. Pineda, «Fisiopatología evaluación y manejo del dolor agudo en pediatría,» *Salutem Scientia Spiritus*, pp. 25-37, 2015.
- [5] Tovar y M. Ana, «Dolor en niños,» *Colombia médica*, vol. 36, n^o 4, 2005.
- [6] S. Merkel, T. Voepel y M. Shobha, «Pain Assessment in Infants and Young Children: The FLACC scale,» *American Journal of Nursing*, vol. 102, n^o 10, pp. 55-58, 2002.
- [7] C. Johnston y B. Stevens, «Differential response to pain by very premature neonates,» *Pain*, vol. 61, n^o 3, pp. 471-479, 1995.
- [8] B. Ghai, J. Makkar y J. Wig, «Postoperative pain assessment in preverbal children and children with cognitive impairment,» *Pediatric Anesthesia*, vol. 18, pp. 462-477, 2008.
- [9] D. Cobo y P. Daza, «Signos vitales en pediatría,» *Gastrohnap*, vol. 13, n^o 1, pp. 58-70, 2011.
- [10] Oracle, «What is a database?,» [En línea]. Available: <https://www.oracle.com/database/what-is-database.html>. [Último acceso: 8 Diciembre 2019].
- [11] SQLite, «About SQLite,» [En línea]. Available: <https://www.sqlite.org/about.html>. [Último acceso: 09 Diciembre 2019].
- [12] Technopedia, «What is an Android App?,» 2018. [En línea]. Available: <https://www.techopedia.com/definition/25099/android-app>. [Último acceso: 12 12 2019].

- [13] Google, «Cómo compilar una IU con Layout Editor,» [En línea]. Available: <https://developer.android.com/studio/write/layout-editor>. [Último acceso: 12 12 2019].
- [14] Google, «Cómo analizar tu compilación con el Analizador de APK,» [En línea]. Available: <https://developer.android.com/studio/build/apk-analyzer>. [Último acceso: 12 12 2019].
- [15] Google, «Cómo ejecutar apps en Android Emulator,» [En línea]. Available: <https://developer.android.com/studio/run/emulator>. [Último acceso: 12 12 2019].
- [16] Google, «Conceptos básicos del flujo de trabajo para desarrolladores,» [En línea]. [Último acceso: 12 12 2019].
- [17] A. Kaplan y M. Haenlein, «Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence,» *Business Horizons*, vol. 62, n° 1, pp. 15-25, 2019.
- [18] I. Goodfellow, Y. Bengio y A. Courville, *Deep learning*, Massachussets: MIT, 2016.
- [19] MATHWORKS, «Machine Learning: Tres cosas que es necesario saber,» [En línea]. Available: <https://la.mathworks.com/discovery/machine-learning.html>. [Último acceso: 12 12 2019].
- [20] Code Academy, «Normalization,» [En línea]. Available: <https://www.codecademy.com/articles/normalization>. [Último acceso: 13 12 2019].
- [21] Scikit-learn, «Imputation of missing values,» Scikit-learn, [En línea]. Available: <https://scikit-learn.org/stable/modules/impute.html#impute>. [Último acceso: 13 12 2019].
- [22] W. Badr, «6 Different Ways to Compensate for Missing Values In a Dataset (Data Imputation with examples),» *Towards Data Science*, 5 Enero 2019. [En línea]. Available: <https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779>. [Último acceso: 13 12 2019].
- [23] R. Malarvizhi, «K-Nearest Neighbor in Missing Data Imputation,» *International Journal of Engineering Research and Development*, vol. 5, n° 1, pp. 5-7, 2012.
- [24] S. Van Buuren, *Flexible multivariate imputation by MICE*, TNO, 1998.

- [25] X. Wu, X. Zhu, G. Wu y W. Ding, «Data mining with big data,» *IEEE Trans Knowl Data Eng*, vol. 26, n° 1, pp. 97-107, 2014.
- [26] S. García, S. Ramírez-Gallego y J. Luengo, «Big data preprocessing: methods and prospects,» *Big Data Analytics*, 2016.
- [27] N. Chawla, K. Bowyer, L. Hall y W. Kegelmeyer, « SMOTE: synthetic minority over-sampling technique,» *J Artif Intell Res*, vol. 16, n° 1, pp. 31-57, 2002.
- [28] B. Boser, I. Guyon y V. Vapnik, «A training algorithm for optimal margin classifiers,» *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144-152, 1992.
- [29] S. Patel, «Chapter 2 : SVM (Support Vector Machine) — Theory,» *Machine Learning 101*, [En línea]. Available: <https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72>. [Último acceso: 12 12 2019].
- [30] A. Ben-Hur y J. Weston, «A User's Guide to Support Vector Machines,» *Methods in molecular biology*, pp. 223-239, 2010.
- [31] D. Hawkins, «The problem of overfitting,» *J. Chem. Inf. Comput. Sci*, vol. 44, n° 1, pp. 1-12, 2004.
- [32] C.-W. Hsu y L. Chih-Jen, «A comparison of methods for multiclass support vector machines,» *IEEE Transactions on Neural Networks*, vol. 13, n° 2, pp. 415-425, 2002.
- [33] A. Mishra, «Metrics to Evaluate your Machine Learning Algorithm,» *Towards Data Science*, 24 Febrero 2018. [En línea]. Available: <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234>. [Último acceso: 13 12 2019].
- [34] Interactive Chaos, «Matriz de confusión,» [En línea]. Available: <https://www.interactivechaos.com/manual/tutorial-de-machine-learning/matriz-de-confusion>. [Último acceso: 13 12 2019].
- [35] S. Narkhede, «Understanding AUC - ROC Curve,» *Towards Data Science*, 26 Junio 2018. [En línea]. Available: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>. [Último acceso: 13 12 2019].

- [36] F. Pedregosa, G. Varoquaux, A. Gramfort y V. Michel, «Scikit-learn: Machine Learning in Python,» *JMLR*, vol. 12, pp. 2825-2830, 2011.
- [37] Scikit-Learn, «Support Vector Machines,» [En línea]. Available: <https://scikit-learn.org/stable/modules/svm.html#svm-classification>. [Último acceso: 15 12 2019].
- [38] C. Bravo, «¿Que es un mock-up?,» 16 2 2019. [En línea]. Available: <https://estudioka.es/que-es-un-mock-up/>. [Último acceso: 16 12 2019].
- [39] B. Akella, «Building Support Vector Machine – SVM Algorithm Models Using Python and Sklearn,» IntelliPaat, 4 12 2019. [En línea]. Available: <https://intellipaat.com/blog/tutorial/machine-learning-tutorial/svm-algorithm-in-python/>. [Último acceso: 13 12 2019].
- [40] Material Design, «Material Design,» [En línea]. Available: <https://material.io/design/>. [Último acceso: 16 12 2019].