

ANÁLISIS Y DISEÑO DE UN SOFTWARE PREDICTOR DE VÍA AÉREA DIFÍCIL

OMER MANUEL SALCEDO GALVÁN

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR
FACULTAD DE INGENIERÍA, MAESTRÍA EN INGENIERÍA
CARTAGENA DE INDIAS DT Y C.**

2012

ANÁLISIS Y DISEÑO DE UN SOFTWARE PREDICTOR DE VÍA AÉREA DIFÍCIL

OMER MANUEL SALCEDO GALVÁN

TESIS DE GRADO

**MOISÉS RAMÓN QUINTANA ALVAREZ
MAGÍSTER EN INFORMÁTICA APLICADA
DIRECTOR**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR
FACULTAD DE INGENIERÍA, MAESTRÍA EN INGENIERÍA
CARTAGENA DE INDIAS DT Y C.**

2012

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Cartagena de Indias DT y C, 30 de junio de 2012

Quisiese dedicar la realización de este arduo trabajo:

A Dios como Señor y dador de nuestras vidas.

A mis familiares especialmente Kattia, mi señora esposa por su apoyo incondicional.

A mis Hijos Santiago y Oscar Miguel, motores de mi vida.

Y en especial a todas las personas que nos enseñan a nunca perder las esperanzas.

Omer Salcedo

Quisiese agradecer públicamente el apoyo brindado en la realización de esta investigación:

Al director del programa de Ingeniería de sistemas de la Universidad Tecnológica de Bolívar Msc. Moisés Quintana por sus valiosas recomendaciones en el proceso enseñanza – aprendizaje.

Al Médico Especialista Ariel Llamas por toda la colaboración ofrecida en el campo de la anestesiología, cuyas indicaciones han sido importantes en el desarrollo de la investigación.

Índice de contenido

INTRODUCCIÓN	11
1 DEFINICIÓN DEL PROBLEMA	13
1.1 ANTECEDENTES	14
1.1.1 Procedimientos.....	14
1.1.1.1 Técnicas de predicción.....	15
1.1.2 Productos.....	17
1.2 FORMULACIÓN DEL PROBLEMA	19
2 JUSTIFICACIÓN	21
3 OBJETIVOS	23
3.1 OBJETIVOS GENERAL	23
3.2 OBJETIVOS ESPECÍFICOS	23
4 MARCO TEÓRICO	24
4.1 Análisis y Diseño del software Orientado a Objetos.....	24
4.1.1 La calidad y sus factores	24
4.1.2 ¿Qué es diseño?	26
4.1.2.1 Metodologías de diseño	26
4.1.2.2 Directrices para un buen diseño	27
4.1.2.3 Principios básicos de diseño	27
4.1.2.4 Conceptos del diseño.....	28
4.1.3 UML: Lenguaje Unificado de Modelado	29
4.1.4 Programación Orientada a Objetos	30
4.1.4.1 Principios del modelo orientado a objetos	30
4.1.4.2 Ventajas.....	31
4.1.4.3 Elementos de la programación orientada a objetos	31
4.2 La minería de datos.....	31
4.2.1 Proceso.....	32
4.2.1.1 Selección.....	33
4.2.1.2 Preprocesamiento y transformación.....	33
4.2.1.3 La minería de datos.....	34
4.2.1.4 Resultados de la validación.....	35
4.2.2 Estadísticos de desempeño para un clasificador.....	36

5 DISEÑO METODOLÓGICO	40
5.1 Tipo de estudio.....	40
5.2 Población y Muestra.....	40
5.2.1 Ventana de observación.....	40
5.2.2 Población referencia.....	40
5.2.3 Universo de Estudio	40
5.2.4 Muestra	41
5.2.5 Inclusión.....	41
5.2.6 Exclusión.....	41
5.2.7 Obtención de la información.....	41
5.3 Actividades básicas.....	42
5.3.1 Cronograma de Actividades.....	42
5.3.2 Recursos y Presupuesto.....	43
6 RESULTADOS Y DISCUSIÓN	44
6.1 Variables de estudio.....	44
6.2 Obtención del modelo computacional predictivo.....	45
6.2.1 Herramientas de datamining.....	45
6.3 Generación del modelo.....	46
6.3.1 Conjunto de datos de entrenamiento.....	46
6.3.2 Protocolo experimental.....	46
6.3.3 Selección de atributos.....	47
6.3.4 Minería de datos: selección del clasificador.....	51
6.3.5 Minería de datos: Generación de Arboles de decisiones.....	53
6.3.6 Validación del modelo.....	59
6.3.6.1 Primera fase: validación cruzada con 4 capas.....	59
6.3.6.1.1 Indicadores de desempeño para los clasificadores.....	59
6.3.6.2 Segunda fase: validación sobre datos de prueba.....	60
6.3.6.2.1 Matrices de confusiones.....	61
6.3.6.2.2 Indicadores de desempeño del clasificador.....	62
6.3.6.2.3 Análisis ROC	63
6.3.7 Comparación con otros predictores.....	65
6.4 Diseño del sistema.....	66
6.4.1 Modelo de datos.....	66
6.4.2 Modelo de clases.....	67
6.4.3 Diseño arquitectónico.....	70
CONCLUSIONES.....	71
REFERENCIAS	72

Índice de tablas

Tabla 1: Matriz de confusión.....	37
Tabla 2: Tareas del principales dentro del proceso de investigación.....	43
Tabla 3: Recursos Disponibles.....	43
Tabla 4: Variables de estudio.....	44
Tabla 5: Resultados de Selección de las variables de estudio, aplicación del algoritmo Relief-F.....	50
Tabla 6: Indicadores de desempeño de los algoritmos de clasificación.....	53
Tabla 7: Indicadores de rendimientos sobre los clasificadores. Fase de entrenamiento.....	59
Tabla 8: Matriz de Confusión de la función f, procedimiento de pruebas.....	61
Tabla 9: Matriz de Confusión de la función g, procedimiento de pruebas.....	62
Tabla 10: Indicadores de rendimientos sobre los clasificadores para el valor de predicción 0. Fase de validación.....	63
Tabla 11: Indicadores de rendimientos sobre los clasificadores para el valor de predicción 1. Fase de validación.....	63
Tabla 12: Indicadores de desempeño de los diferentes clasificadores en el área de interés.	65

Índice de ilustraciones

Ilustración 1: Vista de una vía aérea típica.....	13
Ilustración 2: Kit ETV Intubation Trainer.....	18
Ilustración 3: 3B Scientific Intubation Trainer.....	19
Ilustración 4: Proceso de Descubrimiento de Conocimiento en Bases de Datos.....	33
Ilustración 5: proceso de KDD.....	45
Ilustración 6: Construcción del conjunto de datos.....	47
Ilustración 7: Proceso de Selección del algoritmo de clasificación.....	52
Ilustración 8: Árbol de decisión de la variable intubación aplicando Relief-F como método de selección de atributos.....	55
Ilustración 9: Árbol de decisión de la variable intubación aplicando Relief-F como método de selección de atributos, se excluye grosor_cuello.....	57
Ilustración 10: Análisis ROC de las funciones de clasificación, valor de predicción 0.....	64
Ilustración 11: Análisis ROC de las funciones de clasificación, valor de predicción 1.....	65
Ilustración 12: Diagrama Entidad - Relación del sistema.....	67
Ilustración 13: Diagrama de clases del modelo lógico.....	68
Ilustración 14: Diagrama de clases: motor de valoraciones.....	68
Ilustración 15: Diagrama de clases: Especificación de la interfaz de usuario.....	69
Ilustración 16: Diagrama de despliegue de la aplicación.....	70

GLOSARIO DE TÉRMINOS

Anestesiología: f. Ciencia y técnica de la anestesia.

Predictor: adj. Que predice o ayuda a predecir. U. t. c. s.

Intubación endotraqueal: procedimiento médico en el cual se coloca una cánula o sonda en la tráquea, a través de la boca o la nariz.

Fibrobroncoscopio: Tubo largo , delgado y flexible de fibra de vidrio cómo medio óptico.

Laringoscopia: Es un examen del interior de la laringe usando un espejo pequeño sostenido justo debajo de la parte posterior del paladar o un tubo de observación rígido o flexible, llamado laringoscopio, que se coloca en la boca. Dependiendo de las circunstancias, se puede necesitar anestesia.

Endotraqueal, orotraqueal: Dentro de o que atraviesa la tráquea.

Amigdalinos: referentes a las amígdalas. estas son agregados de tejido linfoide situados en la faringe y que constituyen el anillo de Waldeyer, protegiendo la entrada de las vías digestiva y respiratoria de la invasión bacteriana

Manubrio esternal: elemento superior del esternón. se trata de la porción mas ancha y gruesa del esternón. Con forma casi triangular.

Morbimortalidad: Son aquellas enfermedades causantes de la muerte en determinadas poblaciones, espacios y tiempos.

Comorbilidad: La presencia de uno o más trastornos (o enfermedades) además de la enfermedad o trastorno primario.

Software: m. Inform. Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

Minería de datos: (DM, Data Mining) consiste en la extracción no trivial de información que reside de manera implícita en los datos. Dicha información era previamente desconocida y podrá resultar útil para algún proceso. En otras palabras, la minería de datos prepara, sondea y explora los datos para sacar la información oculta en ellos.

Cirugía electiva: procedimiento quirúrgico conciliado entre pacientes y cuerpo médico.

INTRODUCCIÓN

En la disciplina de la anestesiología, se define la intubación como un procedimiento médico en el cual al paciente se le coloca una cánula o sonda en la tráquea, a través de la boca o la nariz, esto con el objeto de recibir medicamentos u oxígeno, generalmente en un procedimiento de cirugía.

En términos generales, se define la vía aérea difícil como una situación clínica en la cual un anestesista con un entrenamiento convencional experimenta dificultad para realizar el procedimiento de intubación o que el procedimiento clínico obstruya con la apropiada ventilación del paciente, convirtiéndose en un importante factor de riesgo el cual, en el peor de los casos, puede repercutir en la muerte del paciente.

Como consecuencia de lo anterior, siempre ha existido interés por lograr anticipar la ocurrencia del problema, para ello se han descrito múltiples procedimientos, en los cuales se han definido un conjunto de predictores, cuyos valores predictivos positivos no son tan altos como se desea.

Por lo tanto, al identificar aquellos pacientes que puedan estar en riesgo de tener una vía aérea de difícil acceso ayudaría en cierta manera al anestesista a realizar los procedimientos adecuados para salvaguardar la vida de sus pacientes. La valoración anestésica de la vía aérea puede ser llevada a cabo a través de medidas anatómicas y pruebas no invasivas.

El presente documento tiene como objeto mostrar el proceso investigativo relacionado con el análisis, diseño e implementación de una herramienta de apoyo para la toma de decisiones del anestesista en el momento de predecir si un paciente programado para cirugía electiva, el cual recibirá anestesia general a través del procedimiento de intubación, pueda o no tener vía aérea de difícil acceso.

Dentro del proceso se realiza una revisión de la literatura a fin de encontrar soluciones informáticas que permitan abordar el problema, en dicha revisión se tiene en consideración

diferentes procedimientos de predicción haciendo uso de pruebas no invasivas.

Con base a la revisión de la literatura y por criterio del experto, se realiza un estudio prospectivo, en la cual se definen un conjunto de variables que actúen como posibles predictores que ayuden a la resolución de la situación – problema. Sobre ese conjunto de variables se realiza un procedimiento de recolección de datos con el fin de descubrir conocimiento haciendo uso de disciplinas relacionadas para este fin.

Los criterios de análisis y diseño de la aplicación dependerán en gran medida de los resultados obtenidos por el procedimiento de generación de conocimiento definido dentro del diseño metodológico. A nivel de implementación se aplican las buenas prácticas de ingeniería del software y de aprendizaje de máquina.

1 DEFINICIÓN DEL PROBLEMA

Cuando la Asociación Americana de Anestesiología (*American Society of Anesthesiologists – ASA*) publicó su primera Guía para el manejo de la vía aérea difícil en 1993, el 28% de las muertes relacionadas con la anestesia estaban originadas por la imposibilidad de ventilar con mascarilla o de intubar. Aunque esta cifra ha disminuido en los últimos años debido al desarrollo de técnicas y aparatos diseñados para ayudar en el manejo de la vía aérea, sigue siendo una causa importante de muerte relacionada con la anestesia. La siguiente ilustración muestra una vía aérea típica [1].

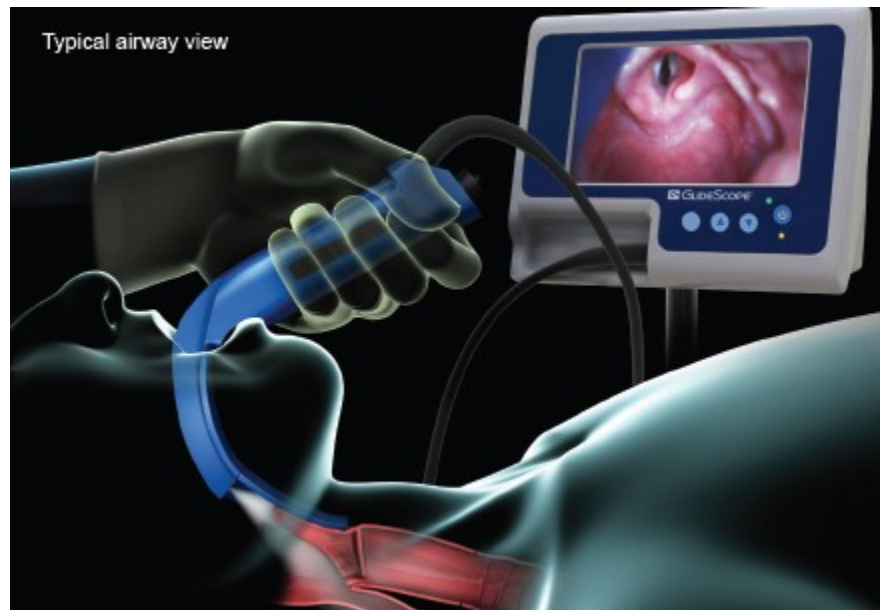


Ilustración 1: Vista de una vía aérea típica

La ASA define la vía aérea difícil como “Situación clínica en la cual un anestesista con un entrenamiento convencional experimenta dificultad para la ventilación de la vía aérea superior con una mascarilla facial, dificultad para la intubación traqueal, o ambas. Este hecho representa sin duda una compleja interacción entre factores del paciente, las

condiciones clínicas, la disponibilidad de aparatos de ayuda, el conocimiento y habilidades del operador” [2].

Desde siempre ha existido interés por lograr anticipar la ocurrencia del problema; como resultado se han descrito para ello múltiples predictores, cuyos valores predictivos positivos no son tan altos como se desean [3]. Se sabe que una ventilación inadecuada, una intubación esofágica, y una intubación traqueal difícil, son los mecanismos más comunes de efectos adversos relacionados con el manejo de la vía aérea en la práctica de la anestesiología. Por lo tanto, es importante identificar aquellos pacientes que puedan estar en riesgo de tener una vía aérea difícil de intubar. La valoración anestésica de la vía aérea puede ser llevada a cabo a través de medidas anatómicas y pruebas no invasivas.

1.1 ANTECEDENTES

A continuación se realiza una revisión de la literatura acerca del proceso de predicción de la vía aérea difícil en el uso de las técnicas de intubación endotraqueal, en la cual se mencionan tanto procedimientos como productos de software relacionados.

1.1.1 Procedimientos

A lo largo de la historia se han realizado una serie de guías y procedimientos anestésicos, de carácter invasivo, para realizar intubaciones cuando se presentan casos de vía aérea difícil, por ejemplo realizar la intubación traqueal guiada por fibrobroncoscopio [4], siendo una de las técnicas de elección en situaciones en que la laringoscopia es peligrosa, inadecuada o imposible, ya que permite practicar, bajo visión directa, una rápida y atraumática intubación, incluso en los casos con importantes alteraciones anatómicas.

Respecto a las guías, la más importante es la construcción de una guía para el tratamiento de intubación para pacientes con vía aérea difícil realizada por la Asociación Americana de Anestesiología en 1993 y actualizada en octubre de 2003 [2].

1.1.1.1 Técnicas de predicción

Se ha buscado la posibilidad de predecir la vía aérea difícil a través de técnicas en la mayoría de los casos no invasivas. Las revisiones de la literatura realizadas por Sunanda Gupta [5] y Jaime Escobar [3] permite resumir un conjunto de dichas técnicas de predicción de acuerdo a ciertas características clínicas de los pacientes. Los estudios clínicos relacionados con las técnicas de predicción más significativos se citan a continuación:

1. La clasificación de Mallampati [6]: Valora la visualización de estructuras anatómicas faríngeas y la lengua, con el paciente en posición sentada y la boca completamente abierta.
 1. Clase I. Visibilidad del paladar blando, úvula y pilares amigdalinos.
 2. Clase II. Visibilidad de paladar blando y úvula.
 3. Clase III. Visibilidad del paladar blando y base de la úvula.
 4. Clase IV. Imposibilidad para ver el paladar blando.
2. Distancia esternomentoniana [6]: Valora la distancia que va del borde superior del manubrio esternal a la punta del mentón, cabeza en completa extensión y boca cerrada.
 1. Clase I. Más de 13 cm.
 2. Clase II. De 12 a 13 cm.
 3. Clase III. De 11 a 12 cm.
 4. Clase IV. Menos de 11 cm.
3. Escala de Patil-Aldrete o distancia tiromentoniana [7]: Valora la distancia que existe entre el cartílago tiroideos (escotadura superior) y el borde inferior del mentón, en posición sentada, cabeza extendida y boca cerrada.
 1. Clase I. Más de 6,5 cm (laringoscopia e intubación endotraqueal muy probablemente sin dificultad).

2. Clase II. De 6 a 6,5 cm (laringoscopia e intubación endotraqueal con cierto grado de dificultad).
3. Clase III. Menos de 6 cm (intubación endotraqueal muy difícil o imposible).
4. Puntaje de Riesgo para Intubación Difícil de Wilson [8]: Se examinaron 1.500 pacientes para desarrollar este puntaje de riesgo. Se definieron 5 factores, cada uno puede valer 0, 1, ó 2 (máximo puntaje 10). Una calificación por arriba de 2 predice 75% de dificultades en la intubación, pero con un número considerable de falsos positivos [3]. Los factores observados son:
 1. Peso
 2. Movilidad de cabeza y cuello
 3. Movimiento mandibular
 4. Retroceso mandibular
 5. Dientes muy grandes y extruidos.
5. Potrución mandibular [9]: Se lleva el mentón hacia adelante lo más posible, pasando los incisivos inferiores por delante de los superiores :
 1. Clase I. Los incisivos inferiores pueden ser llevados más adelante de la arcada dental superior.
 2. Clase II. Los incisivos inferiores se deslizan hasta el nivel de la dentadura superior, es decir, quedan a la misma altura.
 3. Clase III. Los incisivos inferiores no se proyectan hacia adelante y no pueden tocar la arcada dentaria superior.
6. La clasificación de Cormack-Lehane [10]: Es una clasificación de carácter invasiva. Valora el grado de dificultad para la intubación endotraqueal al realizar la laringoscopia directa, según las estructuras anatómicas que se visualicen.
 1. Grado I. Se observa el anillo glótico en su totalidad (intubación muy fácil).

2. Grado II. Sólo se observa la comisura o mitad posterior del anillo glótico (cierto grado de dificultad).
 3. Grado III. Sólo se observa la epiglotis sin visualizar orificio glótico (intubación muy difícil pero posible).
 4. Grado IV. Imposibilidad para visualizar incluso la epiglotis (intubación sólo posible con técnicas especiales).
7. Razón entre grosor del cuello y la distancia tiromentoniana [11]: consideran como nuevo predictor la relación existente entre la distancia tiromentoniana y el grosor del cuello, este predictor tiene mayor afinidad si el paciente es obeso ($imc > 27.5$ kg/m²). Si la relación (grosor del cuello / distancia tiromentoniana) es mayor a 5 entonces el paciente tiene alto grado de probabilidad de que presente una mayor dificultad en el procedimiento de intubación.

1.1.2 Productos

En el marco de la informática aplicada, los desarrollos de aplicaciones para ayudar o asistir este procedimientos médico han sido pocos; pero se ve un creciente interés en el área con la finalidad de hacer una serie de desarrollos de softwares especializados en esas tareas.

La empresa ETVView¹ desarrolló un software que hace parte de un kit conocido como ETV Intubation Trainer [12]. Este kit proporciona una visión clara de las vías respiratorias superiores, tráquea y la carina principal para desarrollar las habilidades necesarias en el procedimiento de intubación mediante el uso de laringoscopia directa convencional, así como la intubación nasal.

1 Pagina web: <http://www.etview.com/trainer> [con acceso 2010/06/01]



*Ilustración 2: Kit ETV
Intubation Trainer*

El grupo empresarial internacional 3B Scientific² está especializado en la producción y la comercialización de material didáctico dirigido a la formación especializada en ciencias biológicas y medicina. Este también desarrolló un software incluido dentro de un KIT que tiene como meta el entrenamiento de galenos para el procedimiento de intubación. Conocido como Intubation Trainer™ [13].

El Intubation Trainer™ está dirigido a la capacitación de médicos, agentes sanitarios, enfermeras, especialistas de los pulmones y otros usuarios del sector de la medicina.

Con la ayuda de este software es posible perfeccionar la práctica de la intubación empleando el modelo tridimensional de un adulto.

Dispone de una conexión USB estándar y es compatible con los sistemas operativos Windows 98, 2000, ME y XP.

² Página web <http://www.3bscientific.es> [con acceso 2010/06/01]



Ilustración 3: 3B Scientific Intubation Trainer

Algunas Ventajas del Intubation Trainer™ son:

- Visualización y comprensión de la anatomía de las vías respiratorias superiores.
- Aprendizaje del posicionamiento correcto del tubo en la tráquea y de la manipulación correcta del laringoscopio.
- Respuesta en tiempo real de la coordinación entre el ojo y la mano.
- Aumento del nivel de participación del grupo completo de aprendizaje, durante la capacitación, gracias a función de grabación de la operación.

1.2 FORMULACIÓN DEL PROBLEMA

La vía aérea difícil es una situación clínica en la cual un anestesista con entrenamiento convencional experimenta dificultad para la ventilación de la vía aérea superior con una mascarilla facial, dificultad para la intubación traqueal, o ambas. Es sabido para la comunidad de anestesistas que dicha situación es una causa importante de muerte relacionada con la anestesia.

Mediante la revisión de la literatura, se nota el creciente interés de predecir dicha situación clínica haciendo uso de un conjunto de clasificaciones o predictores, en la mayoría de los casos no invasivos. Desafortunadamente, sus valores predictivos positivos no son tal altos como se desean.

Luego de una revisión de la literatura, no se encontró evidencia de un sistema informático que permita ayudar al anestesista en la toma de decisiones con el fin de determinar si un paciente en una consulta preanestésica pueda tener un cuadro clínico de vía aérea difícil en el procedimiento de intubación.

Con base en lo anterior ¿Será posible diseñar, implementar y validar un software que permita predecir problemas relacionados con la vía aérea difícil, en los pacientes programados para cirugía electiva?

2 JUSTIFICACIÓN

La permeabilidad de la vía aérea es una de las condiciones indispensables para el normal funcionamiento de todos los sistemas corporales. Cualquier alteración que cause obstrucción parcial o total de la vía aérea lleva a trastornos de la ventilación y la oxigenación, que en último término se traducen en acidosis metabólica, hipoxia tisular, daño neurológico y muerte del paciente.

La frecuencia con la que se presentan pacientes con un alto grado de dificultad de intubación es baja como para tener relevancia estadística; lo que sucede es que ese pequeño porcentaje representa el 100% tanto para el paciente y su familia como para el anesthesiólogo.

Existen muchas variables utilizadas para intentar predecir la dificultad de una intubación orotraqueal. Múltiples estudios han demostrado que éstas variables no tienen una sensibilidad adecuada (25 – 50%), pero tienen una muy buena especificidad (90-95%) [3], a pesar de esto estas variables se utilizan y han demostrado que en alguna forma ayudan al anesthesiólogo a disminuir la morbimortalidad de los pacientes tomando medidas preventivas para mejorar el manejo de la vía aérea difícil.

La mejor forma de prevenir comorbilidades y complicaciones, es a través de una valoración clínica adecuada en la consulta preanestésica, por lo tanto mientras más herramientas tenga el anesthesiólogo para valorar la vía aérea de un paciente, mejor van a ser los resultados para el paciente. Por lo tanto éste software puede ayudar en la toma de decisiones al anesthesiólogo al tratar de predecir si un paciente podría tener o no, una vía aérea de difícil

manejo, detectar a tiempo las posibles complicaciones, y tomar las medidas preventivas necesarias para disminuir la morbimortalidad de los pacientes.

Luego de una revisión de la literatura, se han encontrado softwares para simulaciones varias en medicina, pero no existe documentación de trabajos previos (softwares) diseñados para el pronóstico de una vía aérea difícil, basado en los predictores clínicos científicamente avalados en el área de interés.

3 OBJETIVOS

3.1 OBJETIVOS GENERAL

Desarrollar y validar un software que permita predecir problemas relacionados con la vía aérea difícil, en los pacientes programados para cirugía electiva.

3.2 OBJETIVOS ESPECÍFICOS

1. Determinar las variables a considerar para el desarrollo de un modelo computacional predictivo teniendo en cuenta las características socio demográficas y clínicas de los potenciales pacientes.
2. Diseñar un modelo computacional predictivo de vía aérea difícil con base en las variables determinadas en el objetivo específico 1 aplicando técnicas de descubrimiento de conocimientos en datos.
3. Diseñar el software predictivo de vía aérea difícil con base a la implementación del modelo computacional obtenido en el objetivo específico 2, aplicando buenas practicas a nivel de análisis y diseño de sistemas.
4. Diseñar métodos de pruebas de software con el fin de garantizar la correcta funcionalidad de sus componentes.
5. Validar los resultados alcanzados en el proceso de predicción del software con casos reales de procedimiento de intubación, a través de procedimientos de correlación.

4 MARCO TEÓRICO

4.1 Análisis y Diseño del software Orientado a Objetos

El objetivo general de la Ingeniería del Software es producir software de calidad. Por calidad se entiende la adecuación del software a los requisitos exigidos. El camino para obtener software de calidad es mediante un planteamiento riguroso del problema [14].

El proceso de desarrollo de software es aquel en el que las necesidades del usuario son traducidas en requisitos de software, estos transformados en diseño y el diseño implementado en código.

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le denomina ciclo de vida del software, cuyas etapas se mencionan a continuación:

- Análisis
- Diseño
- Implementación
- Instalación

4.1.1 La calidad y sus factores

Pueden distinguirse dos tipos de factores de calidad: factores de calidad externos y factores de calidad internos [15].

Los factores de calidad externos son aquellos que son perceptibles por los usuarios:

- Corrección
- Eficiencia

- Robustez
- Portabilidad
- Extensibilidad
- Facilidad de uso
- Reutilización
- Funcionalidad
- Compatibilidad

Los factores de calidad internos son los perceptibles por los profesionales en computación:

- Modularidad
- Legibilidad

A continuación se definen cada uno de los anteriores conceptos:

Corrección: Habilidad de un sistema o producto software para desempeñar las funciones, exactamente como le fueron definidas en los requisitos y especificaciones.

Solidez o robustez: Habilidad para funcionar aún en condiciones anormales, es decir, con aquellos casos no explicitados en las especificaciones. Si se presentan el sistema termina “limpiamente”

Confiabilidad = Corrección + Robustez

Extensibilidad: Facilidad para adaptarse a los cambios en las especificaciones. Se logra haciendo simples los diseños de los módulos autónomos

Reutilización: Habilidad para utilizar de nuevo productos de software completos o partes de ellos en nuevas aplicaciones

Compatibilidad: Facilidad con la que un producto de software puede combinarse con otros, se logra homogeneidad en el diseño y estandarización en la comunicación entre programas.

Eficiencia: Facilidad de utilizar el mínimo de recursos de cómputo para conseguir mayor rapidez y menor necesidad de almacenamiento.

Portabilidad: Facilidad de transferir productos a diferentes plataformas

Facilidad de uso: Facilidad con la que personas con diferentes niveles de experiencia pueden aprender a usar los productos software y aplicarlos a resolver problemas

Funcionalidad : Conjunto de posibilidades ofrecido por un sistema

4.1.2 ¿Qué es diseño?

Es el proceso de aplicar distintas técnicas y principios con el propósito de definir un producto con los suficientes detalles como para permitir su realización física. En Ingeniería del software, con el diseño se pretende construir un sistema que:

- Satisfaga determinada especificación del sistema.
- Se ajuste a las limitaciones impuestas por el medio de destino.
- Respete requisitos sobre forma, rendimiento, utilización de recursos, costos, etc.

El diseño es la primera etapa técnica del proceso de Ingeniería del Software, consiste en producir un modelo o representación técnica del software que se va a desarrollar, siendo el proceso sobre el que se asienta la calidad del software. Este proceso es de carácter iterativo a través del cual se traducen los requisitos en una representación del software.

El diseño se representa a un alto nivel de abstracción, un nivel que se puede seguir hasta requisitos específicos de datos, funcionales y de comportamiento [14].

4.1.2.1 Metodologías de diseño

Diseño de datos: Modelo de información a estructuras de datos.

Diseño arquitectónico: Define las relaciones entre los elementos estructurales de nuestro programa.

Diseño procedimental: Se transforman los elementos estructurales de nuestro programa en una descripción procedimental del software.

Diseño de interfaz: Describe cómo se comunica el software consigo mismo y con su entorno.

4.1.2.2 Directrices para un buen diseño

El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acomodar todos los requisitos implícitos que desee el cliente.

El diseño debe ser una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el software.

El diseño debería proporcionar una completa idea de lo que es el software, enfocando los dominios de datos, funcional y de comportamiento desde la perspectiva de la implementación.

4.1.2.3 Principios básicos de diseño

El diseñador debe considerar enfoques alternativos juzgando a cada uno en base a los requisitos del problema, los resultados disponibles y los criterios de calidad interna.

Se deberían poder seguir los pasos de diseño hasta el modelo de análisis.

El diseño no va a reinventar nada que ya esté inventado.

El diseño debería presentar uniformidad e integración.

Debe estructurarse para admitir cambios.

El diseño no es escribir código y escribir código no es diseñar.

Se debería valorar la calidad del diseño mientras se crea, no después de terminado.

4.1.2.4 Conceptos del diseño

A la hora de realizar el diseño del software es necesario realizarse los siguientes interrogantes:

¿Qué criterios se pueden usar para dividir el software en componentes individuales?

¿Cómo se separan los detalles de una función o de la estructura de datos de la representación conceptual del software?

¿Existen criterios uniformes que definan la calidad técnica de un diseño de programas?

Para ayudar a resolver las anteriores preguntas se han establecido unos conceptos fundamentales del diseño del software:

- Abstracción
- Refinamiento
- Modularidad
- Arquitectura del software
- Jerarquía de control
- Partición estructural (horizontal y vertical)
- Estructura de datos
- Procedimientos
- Ocultamiento de la información

El diseño modular efectivo reduce la complejidad, facilita los cambios y produce como resultado una implementación más sencilla, permitiendo el desarrollo paralelo de las diferentes partes del sistema.

La independencia funcional se adquiere desarrollando módulos con una clara función evitando una excesiva interacción con otros módulos. Este concepto está derivado de la

modularidad, la abstracción y el ocultamiento de la información.

La independencia funcional se mide con dos criterios:

- *Cohesión*: extensión del concepto de ocultamiento de información. Un módulo cohesivo ejecuta una tarea sencilla de un procedimiento de software y requiere poca interacción con procedimientos que ejecutan otras partes de un programa.
- *Acoplamiento*: medida de la interconexión entre módulos de un programa.

Interesa una cohesión alta y un acoplamiento bajo [15].

4.1.3 UML: Lenguaje Unificado de Modelado

UML (*Unified Modeling Language*) [16] es un estándar para representar gráficamente y modelar la información relacionada en las fases de análisis y especialmente de diseño del software.

Los objetivos de UML son:

- Visualizar, expresa de ideas de diseño de sistema de forma gráfica.
- Especificar, las características de un sistema.
- Construir, a partir de los modelos especificados.
- Documentar, los propios elementos gráficos sirven de documentación.

Un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- Relaciones: relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones:
 - Diagramas de casos de uso

- Diagramas de clases
- Diagramas de objetos
- Diagramas de interacción
- Secuencia
- Colaboración
- Diagramas de actividad
- Diagramas de estados
- Diagramas de componentes

4.1.4 Programación Orientada a Objetos

Como paradigma es una filosofía de la que surge una cultura nueva que incorpora técnicas y metodologías diferentes. En ella el universo computacional está poblado por objetos, cada uno responsable de sí mismo, y comunicándose con los demás por medio de mensajes. Cada objeto representa una instancia de alguna clase, y estas clases son miembros de una jerarquía de clases unidas vía relaciones de herencia [14].

La diferencia entre un objeto y una clase es que un objeto es una entidad concreta que existe en tiempo y espacio, mientras que una clase representa una abstracción, la esencia de un objeto.

4.1.4.1 Principios del modelo orientado a objetos

Abstracción: Descripción simplificada del sistema que resalta unos detalles y suprime otros.

Encapsulamiento: Ocultar detalles de un objeto que no contribuyen a sus características esenciales.

Modularidad: Propiedad de un sistema que ha sido descompuesto en un conjunto de

módulos coherentes e independientes.

Jerarquía o herencia : Abstracciones organizadas por niveles.

4.1.4.2 Ventajas

Integra la abstracción de datos con la disciplina de programación.

Permite la reutilización de código, mantenimiento y extensión de las aplicaciones.

Permite ofrece mayor potencia del lenguaje: herencia y polimorfismo.

Permite Reflejar conceptos de problemas reales.

Más fácil de utilizar.

4.1.4.3 Elementos de la programación orientada a objetos

Objetos: atributos + métodos

- *Atributos*: determinan el estado del objeto
- *Métodos*: operaciones sobre los objetos

Clases:

- Categorías de objetos con propiedades y operaciones comunes
- Implementación total o parcial de tipos abstractos de datos

4.2 La minería de datos

La minería de datos (*Data Mining*) se puede considerar como una etapa de análisis del proceso de Descubrimiento de Conocimiento en Bases de Datos [17]. Es un campo relativamente joven e interdisciplinario de las ciencias de la computación la cual describe el proceso de descubrimiento de nuevos patrones a través de conjuntos de datos, este proceso, además de incluir métodos de las estadísticas y la inteligencia artificial, involucra un componente relacionado con la gestión de procesamiento de datos. En contraste con el

aprendizaje de máquina (*Machine Learning*), por ejemplo, el énfasis está en el descubrimiento de patrones previamente desconocidos en comparación con la generalización de los patrones conocidos de los nuevos datos.

La tarea de minería de datos es el análisis automático o semi-automático de grandes cantidades de datos con el fin de extraer patrones interesantes desconocidos, como los grupos de registros de datos (análisis de cluster), los registros no habituales (detección de anomalías) y dependencias (Asociación Minera-regla).

Estos patrones se pueden considerar como una especie de resumen de los datos de entrada, y se utiliza, por ejemplo, en el proceso de Aprendizaje de Máquina y análisis predictivo, se puede aplicar procedimientos de minería de datos a fin de identificar varios grupos en los datos, que pueden ser utilizado para obtener resultados de predicción más precisa de un sistema de apoyo.

Ni la interpretación de los datos de entrada, preparación de datos o de resultados y presentación de informes son parte de la etapa de minería de datos, pero que pertenecen al proceso de descubrimiento en base de datos en general, como medidas adicionales.

De acuerdo a lo anterior, en esta investigación se asume la minería de datos como un procedimiento dentro del proceso de descubrimiento de conocimiento en bases de datos.

4.2.1 Proceso

El proceso de *Descubrimiento de Conocimiento en Bases de Datos (KDD)* se define comúnmente con las siguientes etapas:

1. Selección
2. Preprocesamiento
3. Transformación
4. minería de datos
5. Interpretación / evaluación.

La siguiente ilustración muestra las etapas básicas del KDD [17]:

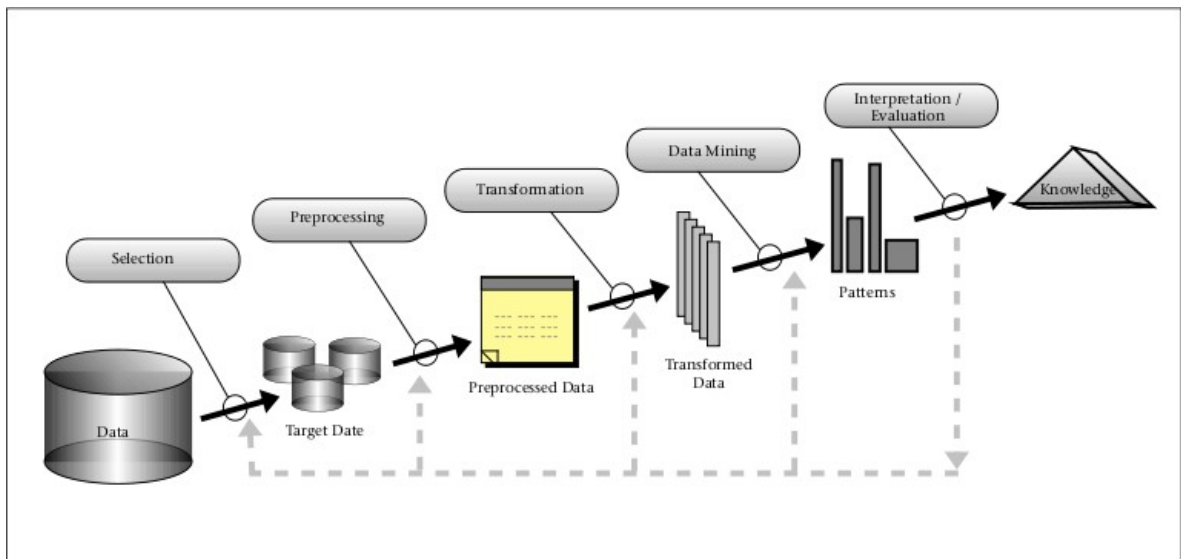


Ilustración 4: Proceso de Descubrimiento de Conocimiento en Bases de Datos

4.2.1.1 Selección

Proceso en el cual se definen el conjunto de variables de entrada así como la variable o conjunto de variables de respuesta a través de un procedimiento de extracción.

Las variables de entrada: o variables independientes, son utilizadas para el procedimiento de descubrimiento de patrones.

Generalmente las variables de respuesta que también se conocen como variables objetivos son aquellas las cuales se quieren predecir, inferir o calcular información relevante o descubrimiento de nuevos patrones.

4.2.1.2 Preprocesamiento y transformación

Antes de utilizar los algoritmos de minería de datos, se debe preparar el conjunto de datos de entrada. Como la minería de datos sólo se puede descubrir patrones presente en los datos, el conjunto de datos de destino debe ser lo suficientemente grande como para

contener estos patrones sin dejar de ser lo suficientemente conciso para ser explotado en un plazo aceptable. El Pre-proceso es esencial para analizar los conjuntos de datos multivariantes antes de aplicar los algoritmos de minería de datos.

Bajo este proceso se “limpia” el conjunto de datos en la cual se realizará de diversas formas en función del análisis previo, con el objetivo de prepararlo para aplicar la técnica de minería de datos que mejor se adapte a los datos y al problema. La limpieza de datos tiene como objeto eliminar las observaciones con ruido o ausencia de datos.

El preprocesamiento genera de forma inherente la transformación del conjunto de datos de entrada.

4.2.1.3 La minería de datos

Se puede definir como el proceso que, a través de la aplicación de técnicas de análisis de datos o un conjunto de algoritmos de descubrimientos, permitan producir un conjunto de patrones o modelos sobre los datos. La minería de datos consta de seis clases de tareas comunes:

Detección de anomalías: tiene como objetivo la identificación de los registros de datos inusuales, que podrían ser interesantes o errores de datos y requieren una mayor investigación.

Asociación de reglas de aprendizaje (modelo de dependencia): Busca relaciones entre las variables. Por ejemplo, un supermercado podría recopilar datos sobre los hábitos de compra del cliente utilizando el aprendizaje de reglas de asociación, el supermercado puede determinar qué productos se compran conjuntamente con frecuencia y usar esta información para fines de marketing. A veces esto se conoce como análisis de la cesta.

Clustering: es la tarea de descubrir grupos y estructuras en los datos que se encuentran en una u otra manera "similar", sin necesidad de utilizar las estructuras conocidas en los datos.

Clasificación: es la tarea de generalizar la estructura conocida de aplicar a los nuevos datos. Por ejemplo, un programa de correo electrónico pueden intentar clasificar un correo

electrónico como legítimo o spam.

Regresión: Los intentos de encontrar una función que modele los datos con el menor error.

Sumarización: tarea que tiene como objeto proporcionar una representación más compacta del conjunto de datos, incluyendo la visualización y generación de informes.

4.2.1.4 Resultados de la validación

El paso final de descubrimiento de conocimiento de los datos es el de verificar que los patrones producidos por los algoritmos de minería de datos puedan realizar inferencias sobre un conjunto más amplio de datos.

No todos los patrones encontrados por los algoritmos de minería de datos son necesariamente válidos. Es un error común que el conjunto de entrenamiento en los cuales se aplican los algoritmos de minería de datos a fin de encontrar patrones sean utilizados para validar el modelo de clasificación, o que el conjunto de datos de entrada sea sesgado por el experimentador para esperar obtener resultados convenientes, al realizar las anteriores acciones es posible que los patrones obtenidos por los algoritmos de datamining no estén presentes en el conjunto de datos generales. Esto se conoce como sobreajuste.

Para superar esto, la evaluación utiliza una prueba de conjunto de datos en los que el algoritmo de minería de datos no estaba capacitado. Los patrones aprendidos se aplican a este conjunto de pruebas y el resultado se compara con la salida deseada. Por ejemplo, se aplicó un algoritmo de minería de datos a fin de poder tratar de distinguir el spam del correo legítimo, El primer paso sería entrenar el algoritmo dada una muestra de mensajes de correos electrónicos. Una vez entrenados, los patrones aprendidos se aplican al conjunto de la prueba de los correos electrónicos en los que no habían sido entrenados. La precisión de estos modelos se puede medir a partir de cuántos correos electrónicos se clasifican correctamente.

Una serie de métodos estadísticos pueden ser utilizados para evaluar el algoritmo, como las curvas ROC, la cual es una representación gráfica de la sensibilidad frente a (1 –

especificidad) para un sistema clasificador binario según se varía el umbral de discriminación esta es una medida de desempeño para los clasificadores [18].

Si los patrones aprendidos no cumplen con los estándares deseados, entonces es necesario volver a evaluar y modificar el pre-procesamiento y extracción de datos. Si los patrones aprendidos cumplen con los estándares deseados entonces el último paso consiste en interpretar los patrones aprendidos y convertirlos en conocimiento.

4.2.2 Estadísticos de desempeño para un clasificador

De acuerdo con Fawcett, cuando se consideran problemas de clasificación usando solamente dos clases, formalmente, cada instancia I es mapeada a un elemento del conjunto $\{p,n\}$ de clases positivas y negativas. Un modelo de clasificación (o clasificador) tiene como objeto mapear instancias a clases de predicción, algunos modelos de clasificación producen una salida continua (por ejemplo una probabilidad). Otros modelos producen una salida discreta indicando solo clases de predicción de la instancia. para distinguir entre la clase actual y la clase predicha se usan etiquetas $\{y,n\}$ para las clases de predicción producidas por el modelo [18].

Dado un clasificador y una instancia, hay 4 posibles resultados. si la instancia es positiva y fue clasificada como positiva, esta se cuenta como verdadero positivo (True Positive -TP); si es clasificada como negativa, se cuenta como falso negativo (False Negative -FN). Si la instancia es negativa y fue clasificada como negativa, se cuenta como verdadero negativo (True Negative -TN); si es clasificada como positiva es contada como un falso positivo (False Positive -FP).

Dado un clasificador y un conjunto de instancias (del conjunto de pruebas) se pueden ilustrar los conceptos definidos por Fawcett [18] en una matriz de 2x2 llamada matriz de confusión (o también llamada matriz de contingencia) de la siguiente forma:

		Predicciones del modelo		Total
		n	y	
Clases correctas	n	Verdadero Negativo (TN)	Falso Positivo (FP)	N
	p	Falso Negativo (FN)	Verdadero Positivo (TP)	P
Total		N'	P'	

Tabla 1: Matriz de confusión

Teniendo en cuenta la anterior matriz de contingencia se pueden definir los siguientes indicadores de rendimiento para un clasificador [18]:

Verdaderos Positivos (TP) : o también éxitos

Verdaderos Negativos (TN): o también rechazos correctos

Falsos Positivos (FP) : o también falsas alarmas o Error tipo I

Falsos Negativos (FN): o también , Error de tipo II

Sensibilidad o Razón de Verdaderos Positivos (TPR): Es el número de ejemplos positivos detectados en todos los ejemplos positivos, por ejemplo, la proporción de los enfermos correctamente diagnosticados como enfermos.

$$TPR = TP / P = TP / (TP + FN)$$

Razón de Falsos Positivos (FPR) : o también razón de falsas alarmas o fall-out en recuperación de información.

$$FPR = FP / N = FP / (FP + TN)$$

Precisión o Exactitud (Accuracy) (ACC) : es la proporción de ejemplos correctamente clasificados.

$$ACC = (TP + TN) / (P + N)$$

Especificidad (SPC): o Razón de Verdaderos Negativos, es la proporción de ejemplos negativos detectados en todos los ejemplos negativos, por ejemplo, la proporción de sanos correctamente reconocido como saludable.

$$SPC = TN / N = TN / (FP + TN) = 1 - FPR$$

Valor Predictivo Positivo (PPV): o también "precisión" en recuperación de información, es el número de ejemplos positivos en todos los ejemplos clasificados como ejemplo positivo, el número de enfermos entre los diagnosticados como enfermos, o un número de documentos relevantes de todos los documentos recuperados.

$$PPV = TP / (TP + FP)$$

Valor Predictivo Negativo (NPV) :

$$NPV = TN / (TN + FN)$$

Razón de Falsos Descubrimientos (FDR):

$$FDR = FP / (FP + TP)$$

Recall: es la misma medida que la sensibilidad, salvo que este último término es más común en la medicina. El termino *Recall* proviene de la minería de texto, donde significa la proporción de los documentos pertinentes que se recuperan.

Índice de Brier: mide la precisión de las evaluaciones de probabilidad, es la desviación media entre las probabilidades predichas y las reales.

Medida F (F-measure): es una media armónica de precisión y *Recall*.

$$F1 = 2TP / (P + P') = 2TP / (2TP + FP + FN)$$

Coefficientes de correlación de Matthews (MCC): Es una medida de la calidad para los clasificadores binarios. Tiene en consideración los verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Se considera generalmente como una medida equilibrada que puede ser usado incluso si las clases son de tamaños muy diferentes. El MCC es, en esencia, un coeficiente de correlación entre las clasificaciones binarias observadas y previstas, este coeficiente devuelve un valor entre -1 y 1. Un coeficiente de 1 representa una predicción perfecta, 0 un valor de predicción al azar y -1 una predicción inversa.

$$MCC = (TP * TN - FP * FN) / \sqrt{P * N * P' * N'}$$

5 DISEÑO METODOLÓGICO

Para conseguir los objetivos puestos en esta investigación se estimaron un conjunto de actividades definidas en la metodología “*Design Science Research Methodology in IS*” [19] cuyas actividades permite que los diseñadores puedan abordar los problemas de diseño "sistemáticamente", mirando a nivel funcional los problemas tales como los objetivos, requisitos, y así sucesivamente, de esta forma se consigue avanzar hacia soluciones más específicas.

5.1 Tipo de estudio

Observacional, descriptivo, prospectivo. De acuerdo al número de pacientes se podría considerar como serie de casos.

5.2 Población y Muestra

5.2.1 Ventana de observación

Por criterio del experto con el objeto de cumplir con los tiempos necesarios para realizar a satisfacción dicha investigación el periodo para obtener las diferentes series de casos comprende desde 1 de mayo del 2010 hasta el 26 de mayo del 2011.

5.2.2 Población referencia

Todos pacientes programados para cirugía electiva, que requieran anestesia general como primer plan anestésico.

5.2.3 Universo de Estudio

Todos pacientes programados para cirugía electiva, en el Hospital Universitario del Caribe y la Clínica Universitaria San Juan de Dios.

5.2.4 Muestra

No se calculara muestra, se trabaja con todos los pacientes mayores o igual de 18 años y menores de 65 años.

5.2.5 Inclusión

Pacientes con edad > 18 años y < de 65 años.

Pacientes programados para cirugía electiva en el Hospital Universitario del Caribe y la Clínica Universitaria San Juan de Dios.

Pacientes que requieran anestesia general como primer plan anestésico.

5.2.6 Exclusión

Pacientes menores de 18 años y mayores de 65 años. La razón obedece en el caso de los infantes y/o adolescentes a las dimensiones morfológicas de los huesos y órganos comprometidos debido a su crecimiento; en cambio en el caso de los pacientes mayores de 65 años en la mayoría de los casos se presentan comorbilidades que pueden afectar significativamente el grado de dificultad de la vía aérea para dichos pacientes.

Pacientes que ingresan a cirugía de urgencias.

Pacientes a quienes se les administre anestesia regional o local.

5.2.7 Obtención de la información

De forma preoperatoria, se detecta el caso, además se hace revisión de la valoración preanestésica en la historia clínica. La revisión de la historia de la vía aérea del paciente debe ser realizada, antes de iniciar el procedimiento anestésico y el manejo de la vía aérea en todos los pacientes.

5.3 Actividades básicas

De acuerdo con Peffers [19] se resaltan las siguientes actividades como las básicas dentro del proceso:

- Identificación del problema y motivación.
- La definición de los objetivos de la solución.
- Diseño y desarrollo:
 - El diseño y posterior validación del modelo computacional predictivo.
 - La Abstracción de los requerimientos e identificación de componentes dentro del sistema (especificación de requerimientos, diseño de sistemas y modelado en UML).
 - El diseño del sistema cuyo producto final sea el aplicativo que cumpla con los requerimientos estipulados y encaminados a resolver el problema de investigación.
- Demostración.
- Evaluación.
- Comunicación.

5.3.1 Cronograma de Actividades

A continuación se relaciona el cronograma de las principales tareas que son fundamentales en el desarrollo del proyecto de investigación:

Tasks

WBS	Name	Start	Finish	Priority	Complete
1	Compilación de Información	06/05/10	21/03/11	500	100,00%
2	Diseño del Modelo Computacional	07/04/11	22/04/11	500	100,00%
3	Implementación del modelo computacional y primer prototipo	27/04/11	14/06/01	500	100,00%
4	Pruebas y validación del prototipo	08/06/11	21/09/11	500	100,00%
5	Refinación del sistema y puesta en preproducción	28/09/11	26/10/11	500	100,00%
6	Análisis de resultados	07/04/11	16/11/11	500	100,00%

Tabla 2: Tareas del principales dentro del proceso de investigación

5.3.2 Recursos y Presupuesto

A continuación se relacionan los recursos utilizados y el presupuesto necesario para la ejecución del proyecto.

Resources

Name	Type	Group	Email	Cost
Computador	Material			\$1.500.000,00
Software: JDK 1.6	Material			\$0,00
Software: MySQL 5.1	Material			\$0,00
Software: NetBeans 7.0	Material			\$0,00

Tabla 3: Recursos Disponibles

6 RESULTADOS Y DISCUSIÓN

6.1 Variables de estudio

De acuerdo a la revisión de la literatura y por criterio del experto, las variables de estudio, cuya información se recolecta para cada paciente son las siguientes:

Variable	Tipo	Unidad
Sexo	char	{M, F}
Edad	int	Años
Peso	float	Kg
Talla	float	Cm
Imc	float	Kg/m ²
Mallampati	int	{1,2,3,4}
Distancia tiromentoniana	float	Cm
Distancia esternomentoniana	float	Cm
Apertura oral	float	Cm
Ángulo de movilidad de la cabeza	float	Grados sexagesimales
Grosor de cuello	float	Cm
Translación mandibular	float	Cm
Intubación	int	{0,1}

Tabla 4: Variables de estudio

6.2 Obtención del modelo computacional predictivo

El objeto de la recolección de los datos teniendo en cuenta las variables de estudio relacionadas anteriormente es obtener una función f que relacione la posible interacción de las variables bajo estudio respecto al grado de dificultad de la intubación:

$$\text{intubación} = f(\text{sexo}, \text{edad}, \text{peso}, \text{talla}, \text{imc}, \text{mallampati}, \text{d_tiromentoniana}, \text{d_esternomentoniana}, \text{apertura_oral}, \text{a_m_cabeza}, \text{grosor_cuello}, \text{translación_mandibular})$$

El proceso de generar conocimiento a partir de un conjunto de datos haciendo uso de técnicas de minería de datos se ilustra a continuación:

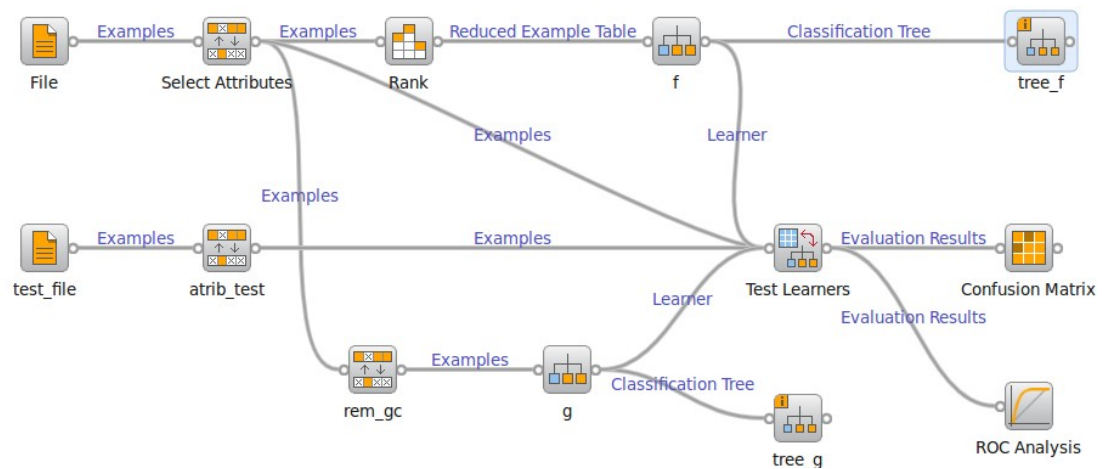


Ilustración 5: proceso de KDD

6.2.1 Herramientas de datamining

Se usó la herramienta Orange³ para todo el procedimiento de selección, clasificación, pruebas y validación de los modelos de clasificación que permitieron desarrollar el modelo

³ Orange. 2011. Version 2.0b Ljubljana (Slovenia): University of Ljubljana [Programa Informático] disponible en internet. <<http://orange.biolab.si/>>

computacional predictivo. En cambio, a nivel de implementación y por compatibilidad a nivel de lenguajes de programación se usó la herramienta WEKA⁴, Una herramienta de datamining desarrollada en el lenguaje de programación JAVA la cual dispone de una API de los diferentes algoritmos de datamining los cuales se desean usar en nuestra aplicación.

6.3 Generación del modelo

6.3.1 Conjunto de datos de entrenamiento

Se obtuvieron 108 muestras de carácter prospectiva de cada una de las 12 variables de estudio sin datos omitidos (*missing*), la proporción de la variable de respuesta (intubación) es:

- 0: 94,44%
- 1: 5,56%

Donde el valor 0 representa una vía de fácil acceso para el paciente, en cambio el valor 1 representa una vía de difícil acceso para el paciente en cuestión en el proceso de intubación.

6.3.2 Protocolo experimental

Por la infrecuencia de pacientes con la situación clínica bajo estudio, se obtuvo un conjunto limitados de datos. Ante esta situación es necesario enfocarse en el desarrollo del clasificador en la fase de entrenamiento.

Ante problemas de clasificación binarios con un conjunto limitado de datos, Christophe Salperwyck y Vincent Lemaire [20] sugieren dividir el conjunto de datos en conjuntos de entrenamientos y conjuntos de pruebas para la generación del modelo haciendo un procedimiento de validación cruzada de n capas (con n=10) el cual se describe a continuación:

⁴ Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

1. Dividir el conjunto de datos en n series de tamaño N/n.
2. Usar los n-1 conjuntos de datos para entrenamiento de los modelos y el resto de prueba.
3. Repetir el procedimiento para los n-1 series de datos y tener una precisión media.

La siguiente figura muestra dicho protocolo [20]:

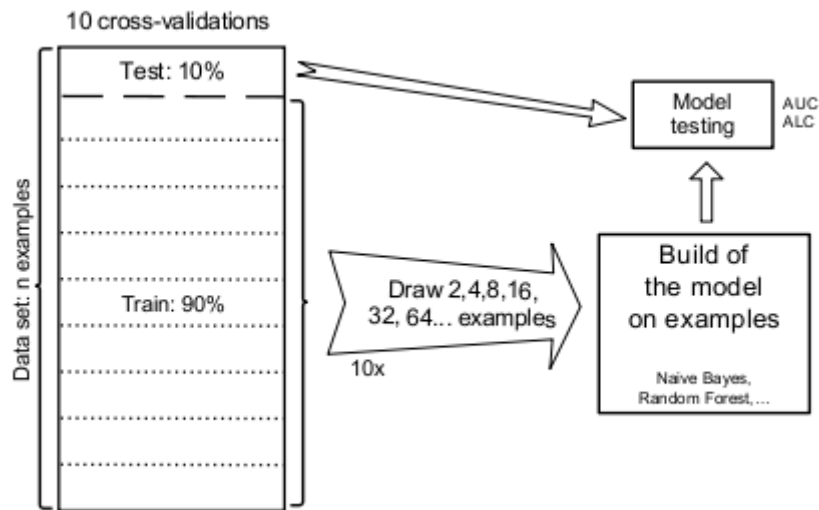


Ilustración 6: Construcción del conjunto de datos

6.3.3 Selección de atributos

La *selección de características* es un término usado habitualmente en la minería de datos para describir las herramientas y las técnicas disponibles para reducir las entradas a un tamaño apropiado para su procesamiento y análisis. La selección de características no solo implica la reducción de cardinalidad, es decir, la imposición de un límite arbitrario o predefinido en el número de atributos que se pueden considerar al crear un modelo, sino también la elección de atributos, lo que significa que el analista o la herramienta de modelado debe seleccionar o descartar activamente los atributos en función de su utilidad para el análisis [21].

El procedimiento de selección de atributos (características) se realiza antes del procedimiento de aprendizaje, y además, es independiente del algoritmo de entrenamiento.

Una de las técnicas aplicadas en Bioinformática para el procedimiento de selección de atributos es el método de eliminación recursiva de características (*Recursive Elimination of Features – Relief*).

Relief [22] es un algoritmo de selección de características inspirado en el aprendizaje basado en instancias. Dado un conjunto de datos de entrenamiento, el tamaño de los datos y unas características iniciales de relevancia, Relief detecta aquellas características las cuales son estadísticamente relevantes respecto a la variable de respuesta.

La idea principal de Relief es estimar los atributos de acuerdo a qué tan bien sus valores se distinguen entre los casos que se encuentran cerca uno del otro. Para cumplir este propósito el algoritmo Relief, para cada instancia, realiza una búsqueda entre dos de sus instancias vecinas cercanas, una de la misma clase y otra de diferente clase.

De hecho, Relief estima los pesos $W[A]$ del atributo A como una aproximación de la siguiente diferencia de probabilidades:

$$W[A] = P(\text{diferentes valores de } A | \text{Instancia cercana de la misma clase}) - P(\text{diferentes valores de } A | \text{Instancia cercana de diferente clase})$$

Kira y Rendell [22] definen el algoritmo como sigue a continuación:

algoritmo Relief es

inicio

 asignar todos los pesos $W[A]=0$;

 para $i=1$ hasta m hacer

 seleccionar al azar una instancia R ;

 buscar una instancia vecina cercana de la misma clase H ;

 buscar una instancia vecina cercana de diferente clase M ;

para A=1 hasta todos los atributos hacer

$$W[A]=W[A]-diff(A,R,H)/m + diff(A,R,M)/m;$$

fin para;

fin para;

fin;

El procedimiento $diff(atributo, instancia1, instancia2)$ obtiene la diferencia de los valores del atributos entre las dos instancias. Si los atributos son discretos retorna 1 sin son diferentes o retorna 0 en caso contrario. En cambio si son continuos obtiene la diferencia normalizada cuyo valor oscila entre 0 y 1.

El valor m representa el número de instancias las cuales se desea realizar la aproximación de probabilidades, entre mayor sea m mas reales serían las aproximaciones, este valor no puede superior al tamaño del conjunto de datos de entrenamiento.

Igor Kononenko [23] realiza un análisis del algoritmo Relief, y describe 6 extensiones del algoritmo.

Relief-F [23]: es la sexta extensión del algoritmo Relief. Respecto al algoritmo original, en vez de encontrar un vecino cercano de diferente clase M , el algoritmo encuentra una un vecino cercano de diferente clase $M(C)$ para cada clase diferente y los promedios de su contribución para la actualización de las estimaciones $W[A]$. El promedio está ponderado con la probabilidad a priori de cada clase:

$$W[A]=W[A]-diff(A,R,H)/m + \sum_{c \neq class(C)} [P(C)*diff(A,R,M(C))]/m$$

La idea es que el algoritmo debe estimar la habilidad de cada atributo para separar cada par de clases independientemente de que las dos clases son los más cercanas la una de la otra.

Relief es una heurística estimadora eficiente de atributos que es capaz de tratar con conjuntos de datos con atributos dependientes e independientes. Su extensiones incorporadas en Relief-F le permiten hacer frente a los conjuntos de datos ruidosos e

incompletos y, lo que es probablemente la contribución más importante del Relief-F, la posibilidad de hacerle frente de forma eficiente a los problemas de múltiples clases.

Al aplicar el algoritmo de selección Relief-F sobre el conjunto de datos de entrenamiento asignando como parámetros se obtuvo el siguiente resultado:

Rank

Muestras: 108

Atributos: 12 ("sexo", "edad", "peso", "talla", "imc", "mallampati", "d_tiromentoniana", "d_esternomentoniana", "apertura_oral", "a_m_cabeza", "translacion_mandibular", "grosor_cuello")

Class: "intubación"

Atributo	Cantidad de valores	ReliefF
"grosor_cuello"	Continuo	0.550
"translacion_mandibular"	3	0.531
"imc"	Continuo	0.302
"peso"	Continuo	0.261
"mallampati"	4	0.232
"talla"	Continuo	0.214
"apertura_oral"	5	0.201
"sexo"	2	0.156
"a_m_cabeza"	Continuo	0.111
"d_tiromentoniana"	Continuo	0.100
"d_esternomentoniana"	Continuo	0.081
"edad"	Continuo	0.0071

Tabla 5: Resultados de Selección de las variables de estudio, aplicación del algoritmo Relief-F

Se puede interpretar de acuerdo con los resultados del algoritmo de selección, que los atributos grosor_cuello, translacion_mandibular, imc, peso y mallampati con un ranking superior al 10%, son significativos respecto a la variable de respuesta intubación.

De acuerdo con José L. Balcázar [24], para realizar estimaciones correctas sobre clasificadores binarios, una buena práctica es tener entre 10 o mas observaciones por cada parámetro del modelo de clasificación el cual se desea estimar.

Bajo esta premisa, se excluyen las variables edad y distancia esternomentoniana por su bajo ranking o participación en la variable de respuesta, obteniendo de esta forma un modelo compuesto por 108 instancias y 10 atributos para construir el clasificador.

La exclusión de la distancia esternomentoniana (o distancia mentón – esternón) debido a su poca efectividad son confirmados por los resultados de Melendez [25] en su trabajo relacionado con la concordancia de la evaluación objetiva y subjetiva en la predicción y hallazgo en la vía aérea difícil.

6.3.4 Minería de datos: selección del clasificador

Para seleccionar el algoritmo de clasificación adecuado para el conjunto de datos, se realizaron pruebas de validación cruzada (n=10) con diferentes algoritmos de clasificación a saber: k Nearest Neighbours (kNN), Classification Tree, CN2, Naive Bayes.

El procedimiento de selección se fundamentará en seleccionar entre los algoritmos de clasificación previamente mencionados aquel cuyos indicadores de desempeño sean los mas favorables, esto permite inferir en cierta forma una mayor correlación entre el procedimiento de clasificación para el conjunto específico de datos.

No se aplicaran algoritmos de regresión a fin de respetar la integridad de los datos de la variable de respuesta (datos discretos).

El proceso de selección se ilustra a continuación:

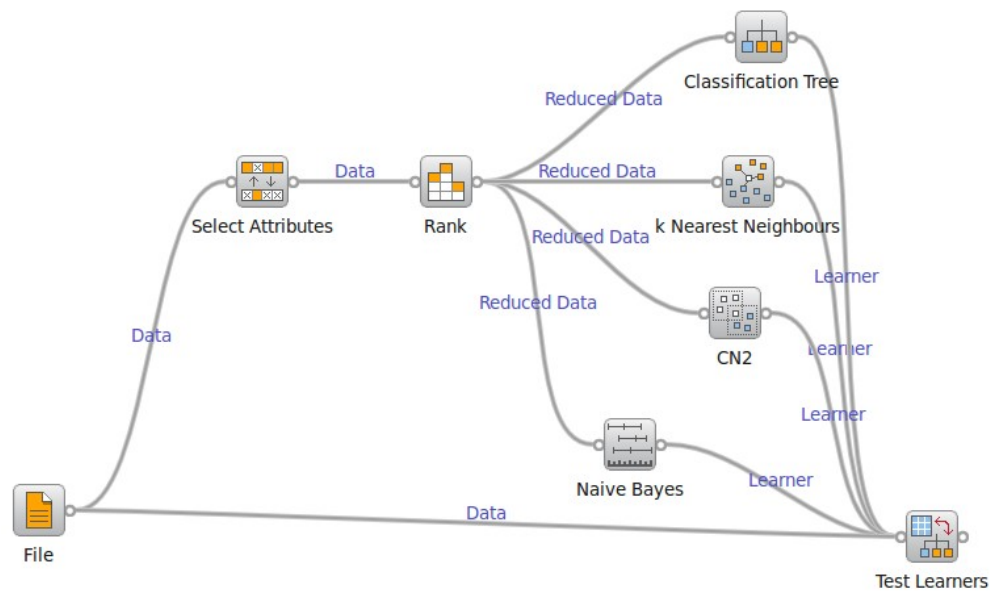


Ilustración 7: Proceso de Selección del algoritmo de clasificación

Las configuraciones de los diferentes algoritmos se describen a continuación:

- *Classification Tree*: No binarizado, selección de atributos a través de la heurística *Relief-F* para un número de 100 observaciones y 10 vecinos.
- *k Nearest Neighbours*: con $k=10$ (10 vecinos) se toma como métrica la distancia euclídeana.
- *CN2 rules*: Se aplica la regla de calidad de estimación de Laplace con $\alpha=0,05$.
- *Naive Bayes*: para la estimación de la probabilidad se tuvo en cuenta la frecuencia relativa, con 100 muestras y umbrales ajustados para los valores discretos.

Para la valoración de los diferentes clasificadores se tendrán en cuenta los siguientes indicadores de desempeños definidos por Fawcett [18]:

- Precisión del clasificador (*CA*)
- Sensibilidad (*Sens*)
- Especificidad (*Spec*)

- Area bajo la curva ROC (*AUC*)
- Coeficientes de correlación de Matthews (*MCC*)

La siguiente ilustración resume los resultados de los algoritmos de clasificación respecto a cada uno de los indicadores de desempeño:

	CA	Sens	Spec	AUC	MCC
kNN	0,9727	0,9902	0,5	0,8578	0,5633
Classification Tree	1	1	1	1	1
Naive Bayes	0,9818	1	0,75	1	0,8618
CN2 rules	0,9909	1	0,75	0,875	0,8618

Tabla 6: Indicadores de desempeño de los algoritmos de clasificación

Luego de analizar el rendimiento de cada algoritmo de clasificación con base al conjunto de datos bajo estudio, se selecciona el algoritmo de clasificación arboles de decisión (*Classification Tree*) teniendo como criterio de selección la heurística Relief-F.

6.3.5 Minería de datos: Generación de Arboles de decisiones

Es posible usar el algoritmo Relief-F como un estimador de los atributos en cada paso de selección dentro del procedimiento de poda de características en la generación de Arboles de decisión [26].

Las técnicas de pre - poda (*Prepruning*) y post - poda (*Postpruning*) se usan con el fin de quitar las hojas no confiables de los árboles de decisión.

Para *prepruning*, a nivel de usuario, existen tres umbrales de que están disponibles para este proceso a saber: número mínimo de instancias de capacitación, ganancia mínima información atributos y la probabilidad máxima de clase de la mayoría en el nodo actual.

Para *postpruning*, el método desarrollado por Niblett y Bratko utiliza la ley de de sucesión de Laplace para la estimación del error esperado debido a la clasificación del nodo actual cometidos por la decisión de podar o no su subárbol [26].

Respecto a la generación del árbol de decisión, para el conjunto de datos de entrenamiento,

se utilizó el algoritmo Relief-F como criterio de selección de atributos. En el proceso de pre-poda se desea dejar por lo menos dos atributos en las hojas de los árboles. En el proceso de post-poda se desea unir de forma recursiva las hojas con la misma clase mayoritaria.

Los parámetros de entrenamiento para método de clasificación *árbol de decisión* son los siguientes:

Classification Tree

Learning parameters

Attribute selection: ReliefF

ReliefF settings: 100 reference samples, 10 neighbours

Binarization: No binarization

Pruning: 2 instances in leaves

Recursively merge leaves with same majority class: Yes

Pruning with m-estimate: m=2

Data

Samples: 108

Attributes: 10 ("sexo", "peso", "talla", "imc", "mallampati", "d_tiromentoniana", "apertura_oral", "a_m_cabeza", "translacion_mandibular", "grosor_cuello")

Class: "intubación"

Los resultados se pueden representar como un grafo como se muestra a continuación:

Classification Tree Grap

Information

Node color: Majority class probability

Target class: 0

Data in nodes: Majority class, Majority class probability

Line widths: Proportion of parent's instances

Tree size: 3 nodes, 2 leaves

Tree

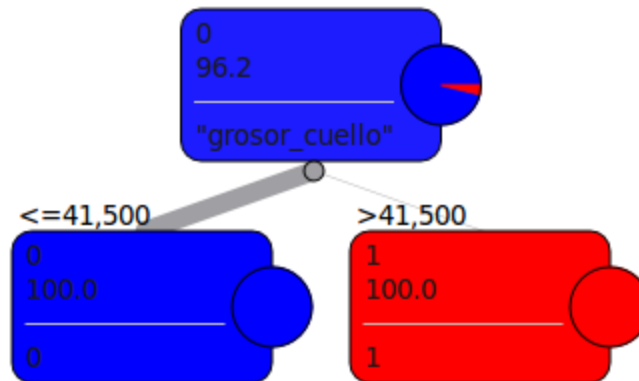


Ilustración 8: Árbol de decisión de la variable intubación aplicando Relief-F como método de selección de atributos

Los anteriores resultados son comparables con los de la investigación realizada por Kim [11] en lo que se refiere a la variable del grosor cuello. Además, confirman los resultados de Escobar [3] referente a la eficiencia de los predictores clínicos utilizados.

Se puede expresar la representación del anterior árbol de clasificación como una función f (objetivo de la investigación) como sigue a continuación:

```

función f ( sexo, edad, peso, talla, imc, mallampati, d_tiromentoniana,
d_esternomentoniana, apertura_oral, a_m_cabeza, grosor_cuello,
translación_mandibular ) retornar entero es
  intubación: entero;
inicio
  si( grosor_cuello > 41.5) entonces
    intubación=1;
  sino
    intubación=0;
  fin si;
  retornar intubación;
fin f;

```

Para reforzar la función de predicción de la dificultad de intubación, es necesario evaluar su valor predictivo, respecto a la definición del algoritmo, cuando el parámetro grosor_cuello sea desconocido (*grosor_cuello=null*). Para realizar la clasificación se excluye del análisis dicho parámetro.

Los parámetros de entrenamiento para método de clasificación árbol de decisión son los siguientes:

Classification Tree

Learning parameters

Attribute selection: ReliefF

ReliefF settings: 100 reference samples, 10 neighbours

Binarization: No binarization

Pruning: 2 instances in leaves

Recursively merge leaves with same majority class: Yes

Pruning with m-estimate: m=2

Data

Samples: 108

Attributes: 9 ("sexo", "peso", "talla", "imc", "mallampati", "d_tiromentoniana", "apertura_oral", "a_m_cabeza", "translacion_mandibular")

Class: "intubación"

Los resultados se pueden representar como un grafo como se muestra a continuación:

Classification Tree Graph

Information

Node color: Majority class probability

Target class: 0

Data in nodes: Majority class, Majority class probability

Line widths: Proportion of parent's instances

Tree size: 6 nodes, 4 leaves

Tree

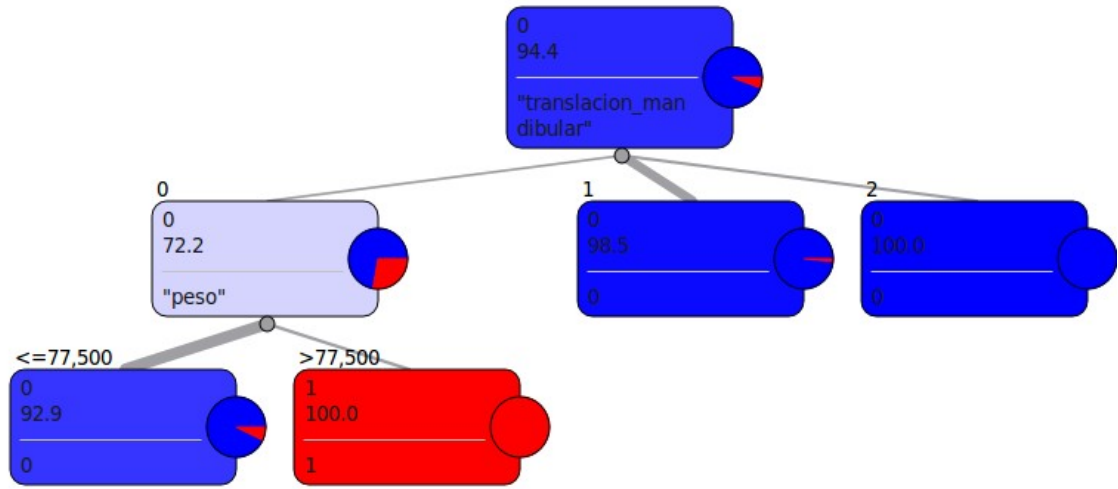


Ilustración 9: Árbol de decisión de la variable intubación aplicando Relief-F como método de selección de atributos, se excluye grosor_cuello

Es posible expresar el anterior grafo como una función g cuyo objetivo es predecir la dificultad de intubación dado los parámetros relacionados con las variables de estudio excluyendo el grosor del cuello (grosor_cuello) como sigue a continuación:

función g (sexo, edad, peso, talla, imc, mallampati, d_tiromentoniana, d_esternomentoniana, apertura_oral, a_m_cabeza, translación_mandibular) *retornar entero es*

intubación: entero;

inicio

si(translación_mandibular < 1 *y* peso >77.5) *entonces*

intubación=1;

sino

```

        intubación=0;
    fin si;
    retornar intubación;
fin g;

```

Luego de obtenida la función de contingencia g , para el caso de que el parámetro grosor_cuello sea desconocido, entonces es posible reforzar el algoritmo de predicción de la dificultad de intubación f tal como sigue a continuación:

```

función f ( sexo, edad, peso, talla, imc, mallampati, d_tiromentoniana,
    d_esternomentoniana, apertura_oral, a_m_cabeza, grosor_cuello,
    translación_mandibular ) retornar entero es
    intubación: entero;
inicio
    si( grosor_cuello > 41.5) entonces
        intubación=1;
    sino
        intubación=0;
    fin si;
    retornar intubación;
excepciones
    cuando (grosor_cuello = null) entonces
        retornar g ( sexo, edad, peso, talla, imc, mallampati, d_tiromentoniana,
            d_esternomentoniana,apertura_oral,a_m_cabeza,
            translación_mandibular );
fin f;

```

6.3.6 Validación del modelo

Para validar el procedimiento adecuado de obtención del modelo se aplicaran dos etapas de validación.

La primera etapa tiene como objeto aplicar la metodología de validación de n capas.

Luego de haber superado la primera fase el modelo se valida con un conjunto de datos nuevos, diferentes del conjunto de entrenamiento.

6.3.6.1 Primera fase: validación cruzada con 4 capas

6.3.6.1.1 Indicadores de desempeño para los clasificadores.

Validation method

Method: Cross-validation

Folds: 4

Target class: 1

Muestras: 108

Atributos: 12 ("sexo", "edad", "peso", "talla", "imc", "mallampati", "d_tiromentoniana", "d_esternomentoniana", "apertura_oral", "a_m_cabeza", "translacion_mandibular", "grosor_cuello")

Class: "intubación"

Results

Función	CA	Sens	Spec	AUC	IS	F1	Prec	Recall	Brier	MCC
f	1,000	1,000	1,000	1,000	0,310	1,000	1,000	1,000	0,000	1,000
g	0,982	0,667	1,000	0,933	0,168	0,800	1,000	0,667	0,035	0,809

Tabla 7: Indicadores de rendimientos sobre los clasificadores. Fase de entrenamiento

Para interpretar los resultados anteriores se tendrá en cuenta los valores del Indicador de

área bajo la curva ROC (*AUC*) [18]. A modo de guía para interpretar las curvas se han establecido los siguientes intervalos para los valores de *AUC*:

- [0.5, 0.6) : Test malo.
- [0.6, 0.75) : Test regular.
- [0.75, 0.9) : Test bueno.
- [0.9, 0.97) : Test muy bueno.
- [0.97, 1] : Test excelente.

La cual se puede interpretar que las funciones de predicción *f* y *g* obtenidas por la aplicación del algoritmo de generación de árboles de decisión teniendo como criterio de selección, el procedimiento de eliminación recursiva de características ReliefF, tuvieron un rendimiento excelente de acuerdo al análisis del área bajo la curva ROC (*AUC*) el cual es uno de los principales indicadores de rendimiento para clasificadores binarios.

6.3.6.2 Segunda fase: validación sobre datos de prueba

Se obtuvieron 88 muestras de carácter prospectiva de cada una de las 12 variables de estudio sin datos omitidos (*missing*), la proporción de la variable de respuesta (intubación) es:

- 0: 94,25%
- 1: 4,60%
- 2: 1,15%

Donde el valor 0 representa una vía de fácil acceso para el paciente, el valor 1 representa una vía de difícil acceso y el valor 2 (una instancia) representa una vía de imposible acceso para el paciente en cuestión en el proceso de intubación.

Respecto a la instancia cuyo valor de intubación es 2 (vía de imposible acceso) si el valor de predicción de la función es equivalente a 1 entonces se considera una buena predicción por dos razones:

- 1) El valor de intubación imposible no estuvo presente en el conjunto de datos de entrenamiento.
- 2) De acuerdo con el experto la ocurrencia de un caso de vía aérea imposible solo sucede si el cuadro clínico del paciente sugiere haber sospechas de vía área de difícil acceso.

Los resultados del proceso de validación se relacionan a continuación:

6.3.6.2.1 Matrices de confusiones

A continuación se presentan las matrices de confusión así como las curvas ROC del modelo predictivo de la dificultad de intubación:

Matriz de Confusión: Función f

Contents

Learner: Classification Tree

Data: Number of Muestras

Matrix:

	0	1	2	
0	82	0	0	82
1	0	4	0	4
2	0	1	0	1
	82	5	0	87

Tabla 8: Matriz de Confusión de la función f, procedimiento de pruebas

Nota Columnas representan resultados de predicciones, Filas representan clases reales.

Matriz de Confusión: Función g

Contents

Learner: Interactive Tree Builder

Data: Number of Muestras

Matrix

	0	1	2	
0	82	0	0	82
1	1	3	0	4
2	0	1	0	1
	83	4	0	87

Tabla 9: Matriz de Confusión de la función g, procedimiento de pruebas

Nota: Columnas representan predicciones, filas representan clases reales.

6.3.6.2.2 Indicadores de desempeño del clasificador.

Luego de realizar las respectivas matrices de confusiones para cada clasificador, se proceden a estimar los indicadores de desempeños, los cuales se observan resumidos en la siguiente tabla:

Validation method

Method: Test on test data

Data

Muestras: 108

Atributos: 12 ("sexo", "edad", "peso", "talla", "imc", "mallampati", "d_tiromentoniana", "d_esternomentoniana", "apertura_oral", "a_m_cabeza", "translacion_mandibular",

"grosor_cuello")

Class: "intubación"

<i>Función</i>	<i>CA</i>	<i>Sens</i>	<i>Spec</i>	<i>AUC</i>	<i>IS</i>	<i>F1</i>	<i>Prec</i>	<i>Recall</i>	<i>Brier</i>	<i>MCC</i>
<i>f</i>	0,989	1,000	1,000	0,995	0,285	1,000	1,000	1,000	0,023	1,000
<i>g</i>	0,977	1,000	0,800	0,897	0,164	0,994	0,988	1,000	0,047	0,889

Tabla 10: Indicadores de rendimientos sobre los clasificadores para el valor de predicción

0. Fase de validación

<i>Función</i>	<i>CA</i>	<i>Sens</i>	<i>Spec</i>	<i>AUC</i>	<i>IS</i>	<i>F1</i>	<i>Prec</i>	<i>Recall</i>	<i>Brier</i>	<i>MCC</i>
<i>f</i>	0,989	1,000	0,988	0,995	0,285	0,889	0,800	1,000	0,023	0,889
<i>g</i>	0,977	0,750	0,988	0,897	0,164	0,750	0,750	0,750	0,047	0,738

Tabla 11: Indicadores de rendimientos sobre los clasificadores para el valor de predicción

1. Fase de validación

6.3.6.2.3 Análisis ROC

A continuación se realiza el análisis de la curva ROC para los clasificadores en cada una de los valores de predicción. Se aplican las recomendaciones realizadas por [18] respecto a la valoración e interpretación de los resultados de las mismas.

Classifiers: Función *g*, Función *f*

Target class: 0

Costs: FP=500, FN=500

Prior target class probability: 93%

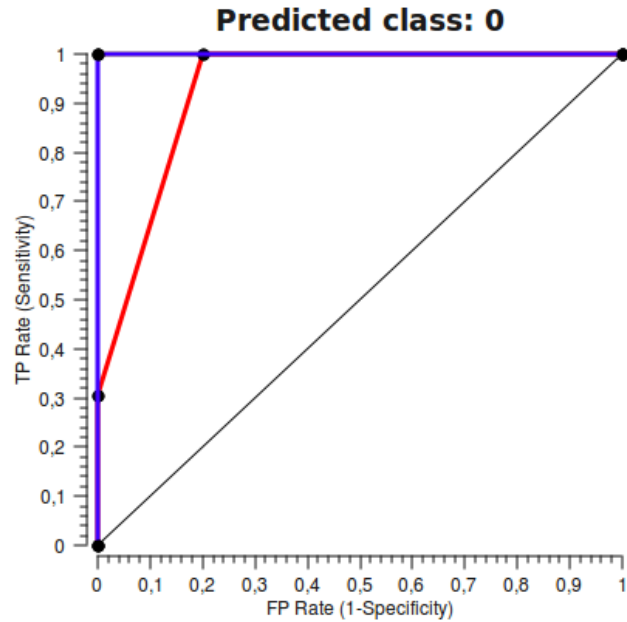


Ilustración 10: Análisis ROC de las funciones de clasificación, valor de predicción 0

Se puede interpretar, de acuerdo al análisis de la curva ROC, que tanto la función f , como la función de refuerzo, g tienen un alto valor predictivo, en el cual el valor del área bajo la curva de ambas funciones son de 0,9952 y 0,8973 respectivamente para los valores de predicción 0.

Classifiers: función g , función f

Target class: 1

Costs: FP=500, FN=500

Prior target class probability: 5%

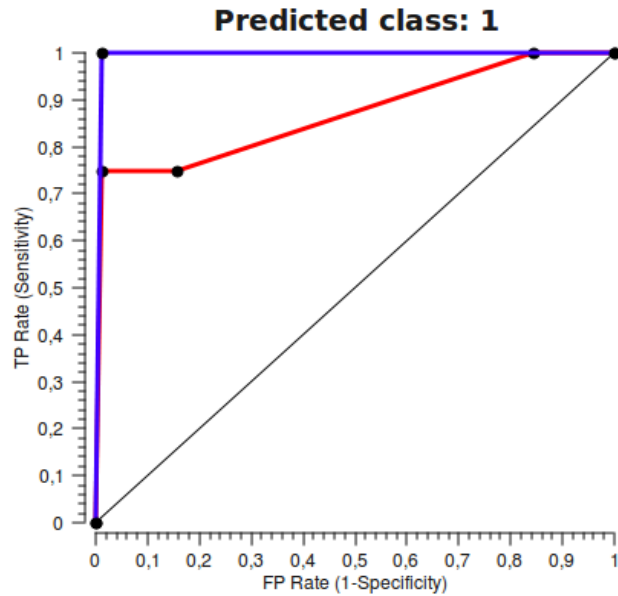


Ilustración 11: Análisis ROC de las funciones de clasificación, valor de predicción 1

Se puede interpretar, de acuerdo al análisis de la curva ROC así como los indicadores de desempeño, que tanto la función **f**, como la función de refuerzo, **g** tienen un alto valor predictivo tanto para verdaderos positivos como verdaderos negativos.

6.3.7 Comparación con otros predictores

Jaime Escobar [3] en su estudio relacionado con la efectividad de los predictores reconocidos en el área de interés, compila una serie de indicadores de desempeño lo cual permite realizar una breve comparación resumida en la siguiente tabla:

Predictor	Sensibilidad	Especificidad	VPP
Mallampati original	0,6	0,89	0,21
Mallampati modificada	0,81	0,82	0,09
Distancia tiromentoniana	0,91	0,82	0,15
Distancia esternomentoniana	0,82	0,89	0,27
Score de Wilson	0,55	0,92	0,09
Apertura bucal	0,47	0,95	0,25
Protrusión mandíbular	0,26	0,96	0,21
función f	0,99	0,99	0,8

Tabla 12: Indicadores de desempeño de los diferentes clasificadores en el área de interés

6.4 Diseño del sistema

Luego de haber definido y validado el modelo computacional predictivo, se procedió a desarrollar el sistema, el cual le dará utilidad al modelo.

Los detalles técnicos relacionados con el paradigma de programación, lenguaje y modelo de persistencia se muestran a continuación:

- Nombre del proyecto: VAD (Vía Área Difícil)
- Paradigma de Programación: Orientado a Objetos
- Lenguaje de Programación: *JAVA (Sun/Oracle)*
- Entorno de desarrollo: *NetBeans IDE 7.0*⁵
- Persistencia de los datos: Bases de datos
- Motor de Base de Datos: *MySql Server 5.1*⁶
- Plataforma de desarrollo: *Win32, Unix, Unix like.*

6.4.1 Modelo de datos

Realizando el análisis del sistema, relacionado con la persistencia de los datos, se definen 4 entidades básicas a saber:

Médicos: Entidad que administra los datos básicos de los médicos los cuales realizan las valoraciones.

Pacientes: Entidad la cual le realizan las valoraciones por parte de los médicos.

Predictores: conjunto de variables socio-demográficas y clínicas las cuales permiten realizar la valoración.

5 NetBeans IDE. 2010. Versión 7.0 Santa Clara - EEUU: Sun Microsystems [Programa Informático] disponible en línea <<http://www.netbeans.org/>> [con acceso 2009-05-04] bajo licencia CDDL

6 MySQL Community Server. 2008. Versión 5.0 Uppsala – Suecia: MySQL AB [Programa informático] en línea <<http://dev.mysql.com/downloads/mysql/5.1.html#downloads>> [con acceso 2009-04-05]. bajo licencia GPL

Valoraciones: Entidad que guarda las predicciones de un paciente realizada por un médico.

El siguiente diagrama de Entidad-Relación muestra como están relacionados las anteriores entidades:

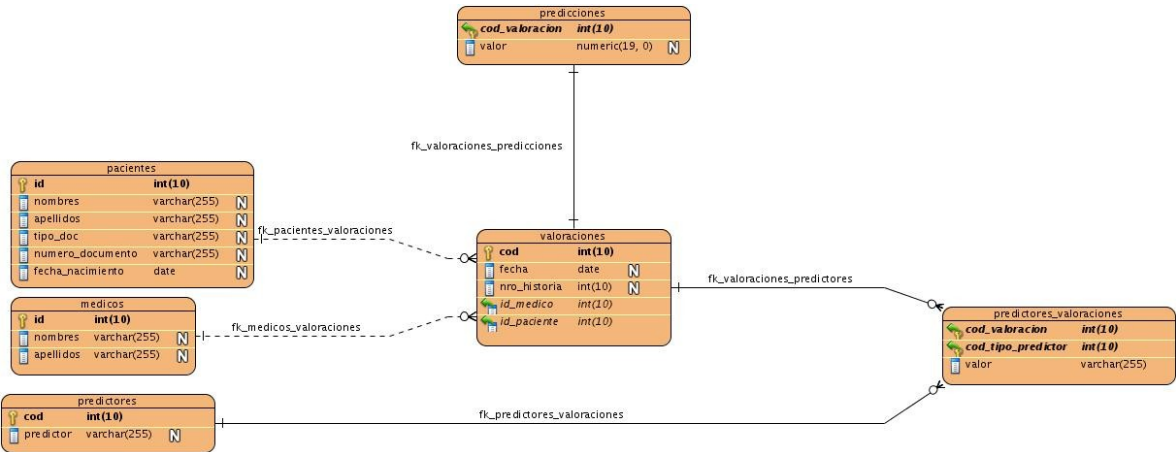


Ilustración 12: Diagrama Entidad - Relación del sistema

El objetivo de la persistencia en la aplicación es de recuperar las valoraciones realizadas por el sistema para estudios relacionados futuros.

6.4.2 Modelo de clases

Al definir las clases relacionadas al diseño lógico del sistema, se puede partir del modelo de dato utilizando la técnica *Mapeo Objeto-Relacional (ORM)* por sus siglas en inglés), la cual es una técnica de programación cuyo objeto es convertir un modelo de datos de una base de datos relacional a un sistema de tipos utilizado en un lenguaje de programación orientado a objetos, utilizando un motor de persistencia.

Para el caso específico se usó el framework *EclipseLink* para realizar el mapeo objeto-relacional al lenguaje de programación *JAVA*. El diagrama de clases, del modelo lógico del sistema se muestra a continuación:

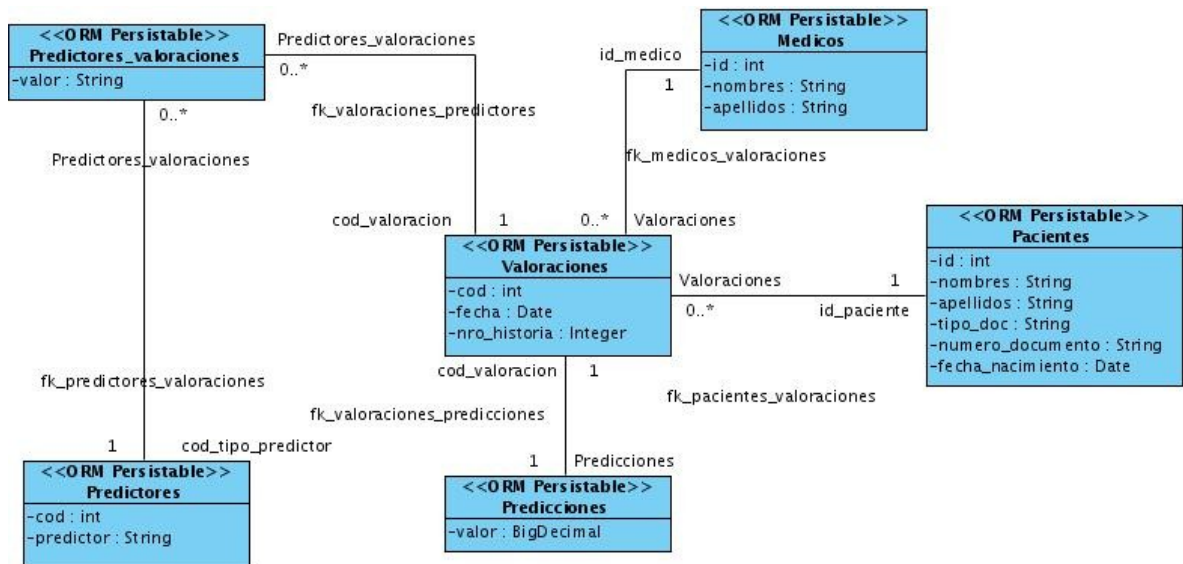


Ilustración 13: Diagrama de clases del modelo lógico

También se define la clase motor, la cual realiza el procedimiento de predicción, es en sí, la clase la cual se implementa el modelo computacional predictivo. La clase motor permite retroalimentar el modelo computacional a partir de las predicciones validadas por el anestesista. El cual se muestra a continuación:

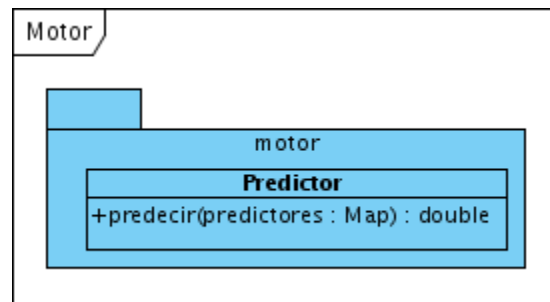


Ilustración 14: Diagrama de clases: motor de valoraciones

Se puede describir, el diseño de la interfaz como un sistema, las cuales las ventanas de la aplicación parten de una forma genérica la cual dicha forma, presenta un componente de

navegación de registros relacionados para cada ventana. El siguiente diagrama de clases describe dichas propiedades:

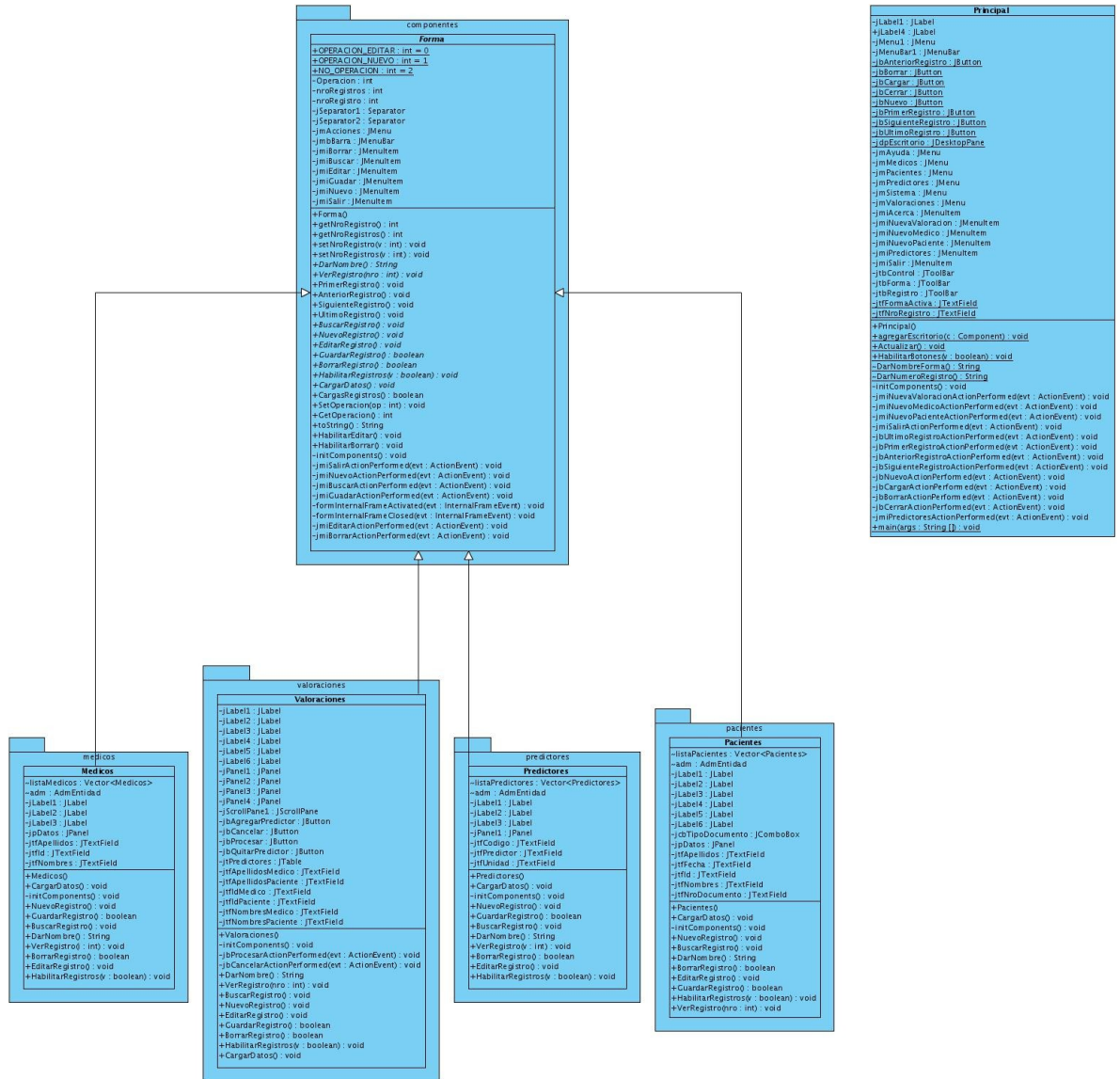


Ilustración 15: Diagrama de clases: Especificación de la interfaz de usuario

6.4.3 Diseño arquitectónico

Se puede describir el Software Predictor de Vía Aérea Difícil, como una aplicación con conectividad hacia una base de datos usando frameworks de persistencia, la conectividad puede ser de forma local o de forma remota al motor de base de datos. Para lograr la conectividad con la base de datos realizada en MySQL se hace uso de la biblioteca JDBC de MySQL, para la unidad de persistencia se usa la biblioteca EclipseLink

Desde el punto de vista de la interfaz de usuario, es una aplicación MDI (Múltiple Documents Interface), al ser desarrollada en el lenguaje de programación JAVA se usaron los componentes de Swing del paquete javax.

Se puede ilustrar la arquitectura del sistema a través de un diagrama de despliegue:

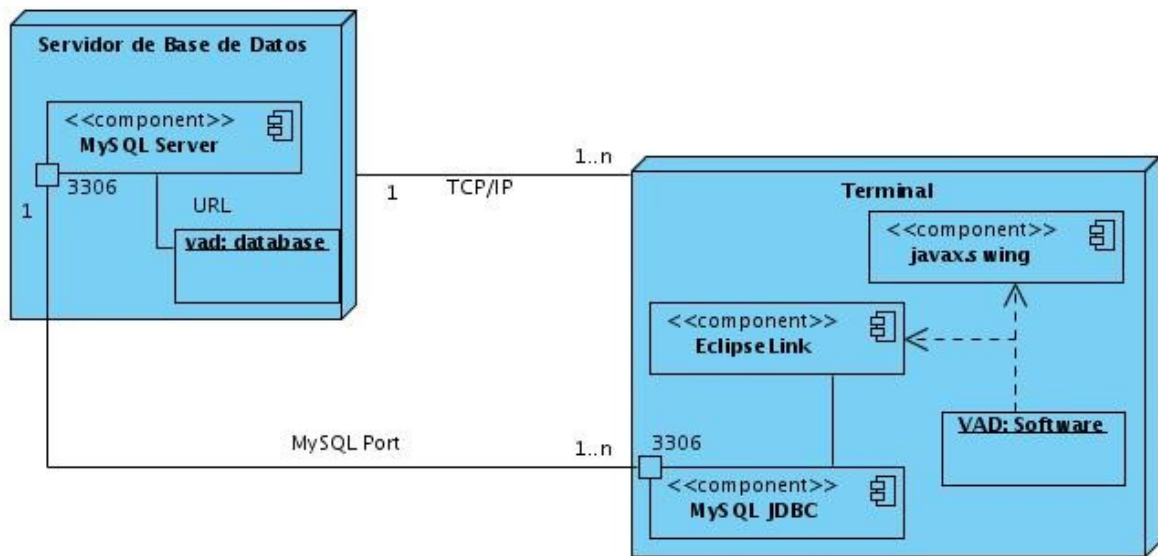


Ilustración 16: Diagrama de despliegue de la aplicación

CONCLUSIONES

Luego de realizar el procedimiento investigativo relacionado con el análisis, diseño e implementación de un software cuya objetivo principal sea actuar como una herramienta de apoyo para la toma de decisión de un anestesista en el momento de predecir si un paciente programado para cirugía electiva pueda tener una vía aérea de difícil acceso, se obtuvieron las siguientes conclusiones:

Se puede interpretar de acuerdo con los resultados del algoritmo de selección de atributos Relief-f que, de los parámetros bajo estudio, el grosor del cuello, la translación mandibular, y el peso ranking superior al 20% , son significativos respecto a la variable de respuesta intubación.

Se pudo obtener una función en termino de las variables bajo estudio, que permiten clasificar la dificultad de intubación, luego de aplicar las técnicas de datamining (Selección, extracción, clasificación y validación), la técnica de clasificación utilizada fue la generación de arboles de decisión con el algoritmo de selección relief-f como criterio de selección de atributos.

Se puede interpretar, de acuerdo a las matrices de confusiones, el análisis de la curva ROC así como los indicadores de desempeño, que la función de clasificación del la dificultad de intubación tienen un alto valor predictivo tanto para verdaderos positivos como verdaderos negativos.

Fue posible realizar una aplicación que permite a los anestesistas dar soporte a la toma de decisiones relacionada con la valoración de un paciente respecto al grado de dificultad que pueda tener este ultimo al aplicarle un procedimiento de intubación de carácter invasiva, conocido de forma previa algunas variables tanto clínicas como socio-demográficas aplicando todos los principios del análisis y diseño orientado a objetos.

REFERENCIAS

- 1: VERATHON Corp. GlideScope [Programa Informático] [en línea] <<http://www.verathon.com/Products/GlideScope.aspx>> [citado en Enero de 2011].
- 2: ASA Task Force. Practice Guidelines for Management of the Difficult Airway. EN *Anesthesiology*. 2003. vol. 8, no. 5, p. 1269 –1277.
- 3: ESCOBAR, Jaime. ¿Cuanto podemos predecir la via aerea dificil?. EN *Revista chilena de Anestesiología*. 2009. no. 38, p. 84-90.
- 4: BOTANA, M. et al. Intubación traqueal guiada por fibrobroncoscopio en pacientes con vía aérea difícil. EN *Medicina Intensiva*. 2009. vol. 33, no. 2, p. 68-73. ISSN 0210-5691.
- 5: GUPTA, S; SHARMA. R y DIMPEL, J. Airway assessment: predictors of difficult airway. EN *Indian J Anaesth*. 2005. vol. 49, p. 257-262.
- 6: MALLAMPATI SR; GATTI, SP. GUGINO, LD. A clinical sign to predict difficult tracheal intubation; a prospective study. EN *Canadian Journal of Anesthesia*. 1985. vol. 32, no. 4, p. 429-434. ISSN 0832-610X.
- 7: PATIL, P, et al. Mechanical aids for fiberoptic endoscopy. EN *Anesthesiology*. 1982. vol. 57, p. 69-70.
- 8: WILSON, ME, et al. Predicting difficult intubation. EN *British Journal of Anesthesia*. 1998. no. 61, p. 211-216.

- 9: TAMURA, M y ISHIKAWA, T. Mandibular advancement improves the laryngeal view during direct laryngoscopy performed by inexperienced. EN *Anesthesiology*. 2004. no. 100, p. 598-601.
- 10: CORMACK, RS y LEHANE, J. Difficult tracheal intubation in obstetrics. EN *Anaesthesia*. 1984. vol. 39, no. 11, p. 1105-1111.
- 11: KIM, W. H. et al. Neck circumference to thyromental distance ratio: a new predictor of difficult intubation in obese patients. EN *British Journal of Anaesthesia*. 2011. vol. 5, no. 106, p. 743–748.
- 12: ETVIEW LTD ETV Intubation Trainer [Programa Informático] [en línea] <<http://www.etview.com/trainer>> [citado en Enero de 2011]
- 13: 3B Scientific Intubation Trainer™ [Programa Informático] [en línea] <http://www.3bscientific.es/software/intubation-trainer-w22200,p_56_0_0_0_7819.html> [citado en Enero de 2011]
- 14: PRESSMAN, Roger. Ingeniería del software un enfoque práctico. 5 edición: McGraw-Hill Interamericana de España. 2002. 602 pp. ISBN 84-481-3214-9
- 15: GALLUD, José y TESORIERO, Ricardo. Diseño y Programación Orientada a Objetos [en línea] <<http://www.info-ab.uclm.es/asignaturas/42579/>> [citado en Abril de 2011]
- 16: BOOCH, Grady; JACOBSON, Ivar y RUMBAUGH, Jim. OMG Unified Modeling Language Specification. 2005.

- 17: USAMA, Fayyad, et al. From Data Mining to Knowledge Discovery in Databases. EN AI MAGAZINE. 1996. p. 37-54.
- 18: FAWCETT, Tom. ROC Graphs: Notes and Practical Considerations for Researchers, 2004
- 19: PEFFERS Ken, et al. A Design Science Research Methodology for Information Systems Research. EN Journal of Management Information Systems. 2008. vol. 24, no. 3, p. 45 - 77.
- 20: SALPERWYCK, Christophe y LEMAIRE, Vincent. Learning with few examples: an empirical study on leading classifiers. EN Neural Networks (IJCNN), The 2011 International Joint Conference on. 2011. vol. , no. , p. ISSN 2161-4393.
- 21: MICROSOFT Corp. SQL Server 2012 Library: Feature Selection (Data Mining) [en línea] <<http://technet.microsoft.com/es-es/library/ms175382.aspx>> [citado en Mayo de 2012]
- 22: KIRA, Kenji y RENDELL, Larry. A practical approach of Feature Selection. 1992. p. 249 - 256.
- 23: KONONENKO, Igor. Estimating Attributes: Analysis and Extensions of RELIEF. 1994. p. 171 – 182.
- 24: BALCAZAR, Jose L. Minería de Datos [en línea] <<http://www.mavir.net/talks/105-jlbalcazar-feb2012>> [citado en Febrero de 2012]

- 25: MELENDEZ, Hector y LEAL, Douglas. Concordancia de la evaluación objetiva y subjetiva en la predicción y hallazgo de la VAD. EN Revista colombiana de Anestesiología. 2010. vol. 38, no. 1, p. 34-49. ISSN 01203347.
- 26: KONONENKO, Igor y SIMEC, Edvard. Induction of decision trees using RELIEFF. 1995.