



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD CIENCIAS DE LA INGENIERÍA
CARRERA INGENIERÍA EN SISTEMAS
CARRERA INGENIERÍA EN TELEMÁTICA

Proyecto de Investigación previo a la
obtención del título de Ingeniera en
Telemática e Ingeniero en Sistemas.

Título del Proyecto de Investigación:
“APLICACIÓN MÓVIL INTELIGENTE PARA LA PREDICCIÓN DE LA
PÉRDIDA DE SEÑAL EN REDES DE TELECOMUNICACIÓN A PARTIR DE
VARIABLES METEOROLÓGICAS”

Autores:

Angie Sthepanie Aguilar Zambrano
Gabriel Fernando Nuñez Villalba

Director de Proyecto de Investigación:

Ph.D. Pavel Novoa Hernández

Quevedo- Los Ríos- Ecuador

2017

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, **Gabriel Fernando Nuñez Villalba**, declaro que el desarrollo de la aplicación móvil aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación personal; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Gabriel Fernando Nuñez Villalba
C.I. 050371508-8

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, **Angie Sthepanie Aguilar Zambrano**, declaro que el trabajo de la minería de datos aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación personal; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Angie Sthepanie Aguilar Zambrano
C.I. 120616040-8

CERTIFICACIÓN DE CULMINACIÓN DEL PROYECTO DE INVESTIGACIÓN

El suscrito, Ph.D. Pavel Novoa Hernández, docente de la Universidad Técnica Estatal de Quevedo, certifica que la Srta. Angie Sthepanie Aguilar Zambrano y el Sr. Gabriel Fernando Nuñez Villalba, realizaron el Proyecto de Investigación de Grado Titulado: **“APLICACIÓN MÓVIL INTELIGENTE PARA LA PREDICCIÓN DE LA PÉRDIDA DE SEÑAL EN REDES DE TELECOMUNICACIÓN A PARTIR DE VARIABLES METEOROLÓGICAS”**, previo a la obtención del título de Ingeniera en Telemática e Ingeniero en Sistemas, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

Ph.D. Pavel Novoa Hernández
DIRECTOR DE PROYECTO DE INVESTIGACIÓN

CERTIFICADO DEL REPORTE DE LAS HERRAMIENTA DE PREVENCIÓN DE COINCIDENCIA Y/O PLAGIO ACADÉMICO

Sr.

Ing. Jorge Patricio Murillo Bayas, M.Sc.

DECANO DE LA FACULTAD CIENCIAS DE LA INGENIERÍA

En su despacho.

De mi consideración. -

Yo, Ph.D. Pavel Novoa Hernández, en calidad del Director de Proyecto de Investigación: **“APLICACIÓN MÓVIL INTELIGENTE PARA LA PREDICCIÓN DE LA PÉRDIDA DE SEÑAL EN REDES DE TELECOMUNICACIÓN A PARTIR DE VARIABLES METEOROLÓGICAS”** de la autoría de la Srta. Angie Sthepanie Aguilar Zambrano y el Sr. Gabriel Fernando Nuñez Villalba, egresados de la carrera de Ingeniería en Telemática e Ingeniería en Sistemas, certifico que han cumplido con las correcciones pertinentes, y su proyecto de investigación ha sido ingresada al sistema **URKUND** para determinar el porcentaje de similitud existente con otras fuentes. La evaluación realizada en el sistema **Urkund** determinó en su informe que existe un 0% de similitud.

Atentamente,



Ph.D. Pavel Novoa Hernández

DIRECTOR DEL PROYECTO DE INVESTIGACIÓN

CERTIFICACIÓN DE REDACCIÓN TÉCNICA DEL PROYECTO DE INVESTIGACIÓN

La suscrita, **Geol. Ernestina Clemencia Coello León**, Docente de la Universidad Técnica Estatal de Quevedo, certifica que Srta. **Angie Sthepanie Aguilar Zambrano** y el Sr. **Gabriel Fernando Nuñez Villalba**, se le procedió a la respectiva revisión y a su vez las correcciones realizadas por el estudiante de su Proyecto Titulado **“APLICACIÓN MÓVIL INTELIGENTE PARA LA PREDICCIÓN DE LA PÉRDIDA DE SEÑAL EN REDES DE TELECOMUNICACIÓN A PARTIR DE VARIABLES METEOROLÓGICAS”** previo a la obtención del título de Ingeniera en Telemática e Ingeniero en Sistemas bajo mi revisión, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

Geol. Ernestina Clemencia Coello León
REDACCIÓN TÉCNICA



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD DE CIENCIAS DE LA INGENIERÍA
CARRERA INGENIERÍA EN SISTEMAS
CARRERA INGENIERÍA EN TELEMÁTICA

PROYECTO DE INVESTIGACIÓN

Título:

“Aplicación móvil inteligente para la predicción de la pérdida de señal en redes de telecomunicación a partir de variables meteorológicas”

Presentado a la Comisión Académica como requisito previo a la obtención del título de Ingeniero en Sistemas e Ingeniería en Telemática.

Aprobado por:

PRESIDENTE DEL TRIBUNAL DE TESIS

Ing. Byron Oviedo Bayas Ph.D.

MIEMBRO DEL TRIBUNAL DE TESIS

Lic. Amilkar Puris Cáceres, Ph.D.

MIEMBRO DEL TRIBUNAL DE TESIS

Ing. Eduardo Samaniego Mena M.Sc

AGRADECIMIENTO

En el presente proyecto investigativo deseamos agradecer aquellas personas que han contribuido en nuestra formación personal y profesional a lo largo de nuestras vidas, permitiéndonos crecer como personas e incentivándonos para alcanzar nuevas metas e inculcándonos a progresar en nuestro vivir diario. Por ello, agradecemos en lo particular a: Papá Dios, por habernos permitido concluir nuestros estudios, escuchando cada una de nuestras oraciones, permitirnos ser fuertes para superar esta etapa de nuestras vidas, y que el día de hoy gracias a él hemos concluido, dándonos la oportunidad de reconocer su presencia a través de personas admirables en nuestra vida personal.

A nuestros padres Isabel Zambrano, Iván Aguilar; Gladys Villalba, Guido Nuñez estamos eternamente agradecidos, por su cariño, comprensión, preocupación, por su enorme sacrificio y amor incondicional al brindarnos la oportunidad de superarnos en nuestra vida personal y profesional. A nuestros hermanos mayores, por ofrecernos la amistad que necesitábamos en los momentos más difíciles, por ser nuestros mejores amigos, cuidarnos y protegernos en cada instante de nuestras vidas y a nuestros demás familiares.

A los docentes de la carrera de Ingeniería en Telemática e Ingeniería en Sistemas, por brindarnos sus conocimientos, amistad, permitiéndonos alcanzar el máximo potencial al momento de estudiar o alcanzar un objetivo e incentivándonos a seguir superándonos en nuestra vida profesional.

A nuestros amigos Willy, Liseth, Lizbeth, Karlita, Virginia, Carlitos y Alejandrita, que sin importar el tiempo que nos conozcamos, nuestras amistades se han fortalecido, creando lazos de amistad para toda una vida, permitiéndonos decir “Quien encuentra un amigo, encuentra un tesoro”.

A todos ustedes muchas gracias, pues en estos momentos en que las palabras suficientes para expresar lo que sentimos, simplemente queda decir aquello que por su significado extenso y sin límites es GRACIAS.

DEDICATORIA

El presente proyecto investigativo está dedicado a nuestro Padre Jehová que nos ha guiado por el buen camino, al darnos la fuerza y la perseverancia para alcanzar nuestras metas y no desmayar en los problemas que se nos han presentado, enseñándonos a enfrentar las adversidades sin perder la fe y confianza ni desfallecer en el intento.

A nuestros amados padres Isabel Zambrano e Iván Aguilar; Gladys Villalba e Guido Nuñez, por su apoyo, consejos, amor, comprensión incondicional, su ayuda en los momentos difíciles, por incentivarnos a estudiar y progresar en nuestras vidas profesionales, a nuestros hermanos y sobrinos por confiar en nuestra capacidad de superación por sus enseñanzas siguiendo un ejemplo a seguir.

Atentamente.

ANGIE AGUILAR ZAMBRANO

GABRIEL NUÑEZ VILLALBA

RESUMEN EJECUTIVO Y PALABRAS CLAVES

El presente proyecto de investigación tiene por objetivo desarrollar una aplicación móvil que permita predecir la pérdida de señal en función de variables meteorológicas, sirviendo así de apoyo a la toma de decisiones por parte de los especialistas en telecomunicaciones. Para el desarrollo del proyecto se utilizó la información existente en investigaciones previas.

Se realizó un pre-procesamiento de los datos; eliminando outliers y correlaciones entre variables. Asimismo, se analizaron diferentes técnicas de clasificación de minería de datos empleando un enfoque de optimización. Como resultado, se determinó que el algoritmo de clasificación basado en Redes Neuronales Artificiales resulta ser el más preciso con un índice de exactitud de 100%.

Con la intención de aprovechar de manera práctica el modelo de clasificación obtenido durante el proceso de minería de datos, este fue representado en el lenguaje *PMML* y posteriormente procesado con tecnología Java en el desarrollo de una aplicación móvil. La aplicación lleva por nombre *SignalPred* y predice posibles pérdidas de señal a partir de la lectura, tanto manual como automatizada, de las variables meteorológicas.

Abstract and Keywords

The aim of this research project is to develop a mobile application to predict signal loss in terms of meteorological variables, serving as support for decision-making by telecommunication specialists. For the development of the project was used the information of previous research.

During the project a pre-processing of the data was performed; removing outliers and correlations between variables. Also, different data mining classification techniques were analyzed using an optimization approach. As a result, it was determined that the classification algorithm based on Artificial Neural Networks turns out to be the most accurate with an accuracy level of 100%.

With aims of applying the classification model obtained during the process of data mining in real-life scenarios, it was represented using the PMML language and subsequently processed with Java technology for the development of a mobile application. The application is named SignalPred and predicts possible signal losses from both manual and automated reading of meteorological variables.

Tabla de Contenido

CAPÍTULO I	CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN.....	1
1.1.	Introducción.....	1
1.2.	Problema de Investigación	2
1.2.1.	Planteamiento del Problema	2
1.2.2.	Formulación del Problema.....	3
1.2.3.	Sistematización del Problema.....	3
1.3.	Objetivos	4
1.3.1.	Objetivo General.....	4
1.3.2.	Objetivo Específicos.....	4
1.4.	Justificación.....	5
CAPÍTULO II	FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN	6
2.1.	Marco Conceptual	7
2.1.1.	Redes de Telecomunicaciones.....	7
2.1.2.	Redes Alámbricas	7
2.1.3.	Redes Inalámbricas	7
2.1.4.	Proceso de Extracción de Conocimiento	8
2.1.5.	Técnicas de Minería de Datos	8
2.1.6.	Árbol de Decisión.....	9
2.1.7.	Red Neuronal.....	9
2.1.8.	Análisis discriminante	9
2.1.9.	Herramienta de minería de datos	9
2.1.10.	RapidMiner	10
2.1.11.	Lenguaje de Marcado de Modelos de Predicción	10

2.1.12.	Plataforma Móvil	10
2.1.13.	Entorno de Desarrollo Integrado.....	11
2.2.	Marco Teórico	11
2.2.1.	Redes de Telecomunicaciones Inalámbricas	11
2.2.2.	Proceso de Extracción de conocimientos, Pre-procesamiento y Sistemas de minería de datos.....	12
2.2.3.	Lenguaje de Marcado de Modelos de Predicción.....	19
2.2.4.	Tecnologías para el Desarrollo Móvil	21
2.3.	Marco Legal	25
2.3.1.	Constitución Política del Ecuador	25
2.3.2.	Ley Orgánica de Telecomunicaciones.....	26
2.3.3.	Ley Orgánica de Comunicación	26
2.4.	Marco Referencial	27
CAPÍTULO III MÉTODOLOGÍA DE LA INVESTIGACIÓN.....		29
3.1.	Localización	30
3.2.	Tipo de Investigación	30
3.2.1.	Investigación Exploratoria.....	30
3.2.2.	Investigación Documental	30
3.3.	Métodos de Investigación.....	31
3.3.1.	Método de Extracción de Conocimiento de Conjuntos de Datos	31
3.3.2.	Método de Desarrollo Programación Extrema	32
3.4.	Fuentes de Recopilación de la Información.	33
3.4.1.	Fuentes Primarias	33
3.4.2.	Fuentes Secundarias	33
3.5.	Diseño de la Investigación.	33

3.6.	Instrumento de Investigación	34
3.7.	Tratamiento de Datos	34
3.8.	Recursos Humanos y Materiales.	34
3.8.1.	Equipo Humanos	35
3.8.2.	Materiales	35
3.9.	Presupuesto.....	37
CAPÍTULO IV RESULTADOS Y DISCUSIÓN.....		39
4.1.	Minería de Datos.	40
4.1.1.	Obtención del Conjunto de Datos.....	40
4.1.2.	Pre-procesamiento de los datos	45
4.1.3.	Optimización de Algoritmos	54
4.1.4.	Generación del Modelo Optimizado.....	63
4.1.5.	Discusión de la Minería.....	66
4.2.	Aplicación Móvil.....	66
4.2.1.	Exportar el modelo de minería de datos.	66
4.2.2.	Desarrollo de la Aplicación Móvil	68
4.2.3.	Discusión de los Resultados	83
CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES.....		85
5.1.	Conclusiones	86
5.2.	Recomendaciones.....	87
CAPÍTULO VI BIBLIOGRAFÍA.....		88
CAPÍTULO VII ANEXOS		92
7.1.	Solicitudes de Información – CNT.....	93
7.2.	Solicitudes de Información – Transtelco S.A.....	94
7.3.	Anexo de la Metodología de Desarrollo de Aplicación	95

7.4.	Anexo de los Operadores de Modelos a utilizar.....	97
7.5.	Diagramas de Casos de Uso.	99
7.6.	API de Android Studio	103
7.7.	Código clase usePMML.java	104
7.8.	Código inicio.java	107
7.9.	Código de ingreManual.java	111
7.10.	Manuales de la Aplicación y Servicio de Respuesta.....	114
CAPÍTULO VIII GLOSARIO.....		123

Índice de Tabla

Tabla 1:	Características de la clasificación de las Técnicas de Minería de datos.	15
Tabla 2:	Cuadro Comparativo de los Sistemas de Minería de Datos	16
Tabla 3:	Enlaces URL de Herramientas de Minería de Datos.....	17
Tabla 4:	Extensión PMML en RapidMiner	21
Tabla 5:	Temas de Marco Referencial.....	28
Tabla 6:	Recursos Hardware.....	35
Tabla 7:	Recursos-Software.....	36
Tabla 8:	Recursos- Otros	37
Tabla 9:	Presupuesto.....	38
Tabla 10:	Descripción de variables meteorológicas	41
Tabla 11:	Parámetros del Operador Select Attributes	46
Tabla 12:	Parámetros del Operador Select Value.....	46
Tabla 13:	Parámetros del Operador Numerical to Binominal	47
Tabla 14:	Parámetros del Operador Set Role	47
Tabla 15:	Parámetros del Operador-Remove Correlated Attributes	48
Tabla 16:	Parámetros del Operador- Remove Duplicates	49
Tabla 17:	Parámetros del Operador-Detect Outlier.....	49
Tabla 18:	Parámetros de Optimización-Evolutionary.	55

Tabla 19: Resultado del Proceso de Optimización de los datos sin balancear en RapidMiner	57
Tabla 20: Resultado del Proceso de Optimización de los datos con sobre-muestreo en RapidMiner.....	59
Tabla 21: Resultado del Proceso de Optimización de los datos con sub-muestreo en RapidMiner.....	61
Tabla 22: Parámetros Optimizado del Modelo Neural Net	64
Tabla 23: Requerimientos mínimo para el funcionamiento de la aplicación	70
Tabla 24: librerías con el URL de descarga	75
Tabla 25: Ingreso de Datos con el servidor sin conexión.....	80
Tabla 26: Ingreso de datos sin acceso a Internet	82
Tabla 27: Metodologías para el desarrollo de software.....	95
Tabla 28: Características de los operadores a analizar.....	97
Tabla 29: Caso de Uso - Ingresar Datos	99
Tabla 30: Caso de Uso - Ver Historial	100
Tabla 31: Caso de Uso - Iniciar Proceso	102
Tabla 32: Código clase PMML.Java	104
Tabla 33: Código Inicio.java	107
Tabla 34: Código ingreManual.java	111
Tabla 35: Glosario de Acrónimos.....	124
Tabla 36: Glosario de Palabras.....	126

Índice de Ilustración

Ilustración 1: Estadísticas del mercado de los teléfonos inteligentes	23
Ilustración 2: Descripción General del Conjunto de Datos elaborado en Weka.	44
Ilustración 3: Operadores de RapidMiner.	45
Ilustración 4: Descripción General del nuevo conjunto de datos en Weka.	51
Ilustración 5: Distribución General de los datos realizando Sobre-Muestro en Weka.....	52
Ilustración 6: Distribución General de los datos realizando Sub-Muestro en Weka.....	53
Ilustración 7: Proceso de Optimización de los algoritmos a utilizar en RapidMiner	54
Ilustración 8: Generación del modelo optimizado en RapidMiner.....	63

Ilustración 9: Parámetros Optimizados del Modelo Neural Net en RapidMiner	64
Ilustración 10: Representación Gráfica del modelo Neural Net en RapidMiner.....	65
Ilustración 11: Extensión PMML en RapidMiner	67
Ilustración 12: Uso del operador Write PMML en RapidMiner	67
Ilustración 13: Diagrama de Flujo - Ingreso Manual de valores	71
Ilustración 14: Diagrama de Flujo - Ingreso Automático de valores	72
Ilustración 15: Diagrama de Flujo - Retroalimentación	73
Ilustración 16: Arquitectura Cliente / Servidor	74
Ilustración 17: Método para cargar un archivo PMML.....	77
Ilustración 18: Definición de Campo de Variables	78
Ilustración 19: Evaluador del modelo de minería de datos	78
Ilustración 20: Pasos para ingresar valores al modelo.....	78
Ilustración 21: Obtención del resultado del modelo y probabilidad del resultado	79
Ilustración 22: Resultado del Ingreso de Datos con el servidor sin conexión en el simulador de Android Studio.....	81
Ilustración 23: Resultado del Ingreso de Datos sin acceso a Internet en el simulador de Android Studio	83
Ilustración 24: Solicitud para Información en CNT	93
Ilustración 25: Solicitud para Información en la empresa Transtelco S.A.....	94
Ilustración 26: Caso de Uso - Ingresar Datos	100
Ilustración 27: Caso de Uso - Ver Historial	101
Ilustración 28: Caso de Uso - Iniciar Proceso	102
Ilustración 29: API de Android Studio	103
Ilustración 30: Ingreso a Google Play y Búsqueda de la App.....	114
Ilustración 31: Permisos para el funcionamiento de SignalPred	115
Ilustración 32: Módulos de SignalPred	116
Ilustración 33: Ingreso Manual de variables.....	117
Ilustración 34: Localización del GPS	118
Ilustración 35: Ingreso Automático de variables	118
Ilustración 36: Resultado Obtenido	119
Ilustración 37: Retroalimentación	120

Ilustración 38: Historial	121
Ilustración 39: Servicio de Respuesta.....	122

Índice de Gráficos

Gráfico 1: Clasificación de las Técnicas de Clasificación.	14
Gráfico 2: Fases de la Metodología XP	32

Código Dublín

Título:	Aplicación móvil inteligente para la predicción de la pérdida de señal en redes de telecomunicación a partir de variables meteorológicas.				
Autores:	Aguilar Zambrano, Angie Sthepanie y Nuñez Villalba, Gabriel Fernando				
Palabras clave:	Telecomunicaciones	Minería de Datos	Aplicación Móvil	Extensión	
	Pérdida de Señal	RapidMiner, Modelo	App	PMML	
Fecha de publicación:					
Editorial:	Editorial: Quevedo: UTEQ,2017				
Resumen:	<p>Resumen</p> <p>El proyecto de investigación tiene como finalidad desarrollar una aplicación móvil que permitirá predecir la pérdida de señal en función de variables meteorológicas. Se emplearán técnicas de clasificación de minería de datos y mediante un proceso de optimización se obtendrá un modelo que será usado de manera práctica en el desarrollo de la aplicación.</p> <p>Abstract</p> <p>The research project aims to develop a mobile application that will predict signal loss depending on meteorological variables. Data mining classification techniques will be used and through an optimization process you will obtain a model that will be used in a practical way in the development of the application.</p>				
Descripción:	Formato: A4 29cm x 21 cm				

CAPÍTULO I

CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN

1.1.Introducción

En las redes de telecomunicaciones inalámbricas las comunicaciones se efectúan mediante dispositivos conectados dentro de un área de cobertura. Las señales transmitidas por estos dispositivos recorren una cierta distancia donde se ven influenciadas por variables meteorológicas que pueden ocasionar sobre ellas: pérdida de señal o afecta la calidad de la señal.

Hoy en día los teléfonos móviles inteligentes son una herramienta útil en la vida cotidiana. Debido a que facilitan tareas como: realizar compras online, utilizar la banca móvil, verificar el clima y permitir visualizar contenidos personalizables como: iconos para aplicaciones, fondos de pantallas animados, temas, entre otros.

El presente proyecto de investigación surge con la finalidad de desarrollar una aplicación móvil que permita predecir la pérdida de señal en función de variables meteorológicas. Para el desarrollo del proyecto se utilizara la información existente de la tesis[1] con la cual se efectuará un estudio de minería de datos y se procederá a utilizar el modelo optimizado de forma práctica.

Para el estudio de minería de datos se emplearán técnicas de pre-procesamiento de datos para eliminar datos atípicos (*outliers*), datos duplicados y correlación entre variables en el caso de existir dando como resultado un nuevo conjunto de datos. Posteriormente se realizará un proceso de optimización de parámetros con diferentes algoritmos de minería de datos, para obteniendo como resultado un modelo con un índice de exactitud aceptable que será exportado a un estándar de modelado.

Se desarrollará una aplicación móvil que estará destinada a predecir la pérdida de señal, la cual interactuará con un servidor remoto para procesar la información y permitirá el ingreso de datos de forma manual o automática.

1.2.Problema de Investigación

En el siguiente apartado se estableció el problema que se pretende solucionar, el diagnóstico con una descripción general, el pronóstico en caso de no haber solución, la formulación y sistematización del problema.

1.2.1. Planteamiento del Problema

En la actualidad existente muchos trabajos de investigación donde la meteorología influye en diversas áreas. El artículo [2] identifica que las variables meteorológicas tienen influencia directa sobre la calidad de las redes inalámbricas dentro del área de telecomunicaciones. Estas redes se han convertido en el medio más común para las comunicaciones entre los usuarios y diversas plataformas, dependiendo así de la disponibilidad total de estas redes.

Debido a la influencia de las variables meteorológicas en las redes de telecomunicaciones inalámbricas, se puede presentar pérdida de señal dentro de la red, lo que ocasionará una serie de inconvenientes como: limitaciones de acceso a Internet, pérdida de envíos de datos, entre otras falencias.

1.2.1.1. Diagnóstico

La pérdida de señal es un problema que afecta de manera directa a las redes de telecomunicaciones inalámbricas. Esto se puede presentar por varias causas como: atenuación, absorción atmosférica, ruido y también pueden ser afectadas por las variables meteorológicas.

1.2.1.2. Pronóstico

Uno de los inconvenientes que continuará presentándose por la pérdida de señal en redes de telecomunicaciones inalámbricas es la limitación al acceso a Internet por largos períodos de tiempo lo que ocasionará incomunicación entre servicios de alto tráfico, falla de sincronización entre aplicaciones, entre otros.

1.2.2. Formulación del Problema

¿Cómo se puede predecir eficientemente la pérdida de señal digitales en base a la información meteorológica?

1.2.3. Sistematización del Problema

- ¿Qué tecnologías existen para realizar un estudio de minería de datos?
- ¿Cómo preprocesar de manera efectiva los datos existentes para predecir adecuadamente la pérdida de señal?
- ¿Cómo determinar el algoritmo que mejor predice la pérdida de señal a partir de variables meteorológicas?
- ¿De qué manera se puede implementar el modelo de minería de datos?
- ¿Cómo aprovechar el mejor algoritmo de predicción?

1.3.Objetivos

En la presente sección se establece el objetivo general y los objetivos específicos que se proponen a cumplir en el desarrollo del proyecto de investigación.

1.3.1. Objetivo General

- Desarrollar una aplicación móvil inteligente que permita predecir la pérdida de señal en redes de telecomunicaciones inalámbricas en función de variables meteorológicas.

1.3.2. Objetivo Específicos

- Identificar la herramienta de minería de datos más adecuada para realizar el análisis inteligente de los datos
- Seleccionar las variables meteorológicas más relevantes para la predicción a partir de los datos disponibles.
- Optimizar los diferentes tipos de algoritmos predictivos de acuerdo a su exactitud.
- Transformar el modelo de minería de datos generado a un objeto de *JAVA* aplicando un estándar de modelado y predicción.
- Elaborar una aplicación móvil inteligente que permita interactuar con el modelo de minería de datos.

1.4.Justificación

En las redes de telecomunicaciones inalámbricas, existen muchos factores que disminuyen el rendimiento en las redes como: atenuación, distorsión, ruido, absorción atmosférica, entre otros. Estos fenómenos ocasionan una alta probabilidad de pérdida de señal afectando considerablemente diversas áreas en la vida cotidiana.

En la actualidad hay estudios orientados a la minería de datos que resuelven varios problemas en diversas áreas, pero estos estudios no son implementados en las áreas que pretenden solucionar. También existen investigaciones aplicadas al uso de las tecnologías en redes de telecomunicaciones, que tienen como objetivo mejorar la calidad de la señal, pero, son pocos los estudios que empiezan con un Proceso de Extracción de Conocimiento y concluyan con la implementación de un modelo.

Por el motivo antes mencionado, se procedió a realizar un nuevo estudio de minería de datos de las variables meteorológicas obtenidas de la tesis de maestría [1]. Para luego ser implementado en una aplicación móvil inteligente que brinde resultados en base a un modelo de minería de datos.

La presente investigación abarca un estudio de minería de datos para la obtención de un modelo de predicción de pérdida de señal con un índice de exactitud aceptable, el mismo que posteriormente será incorporado en una aplicación móvil inteligente que permitirá predecir la pérdida de señal en función de las variables meteorológicas.

CAPÍTULO II

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

2.1. Marco Conceptual

En el apartado Conceptual se establecen los conceptos y definiciones de los temas necesarios para el desarrollo del proyecto de investigación. Esta información sirve como guía para entender los procesos y actividades que se realizarán en el proyecto de investigación.

2.1.1. Redes de Telecomunicaciones

Las redes de telecomunicaciones se encuentran formadas por un sistema de transmisión y recepción de señal. Las redes de telecomunicación se construyen con el fin de brindar servicios de comunicación a los usuarios que se conecten a ella según lo menciona el autor [3].

2.1.2. Redes Alámbricas

El Arcotel en su boletín estadístico en el sector de las Telecomunicaciones[4], menciona que las redes alámbricas utilizan una vía de acceso físico(cable) para conectar los dispositivos a la red como: los cables de cobre, DSL, HFC y/o fibra óptica, la selección del cableado varía dependiendo del área geográfica, la cantidad de hosts, entre otros.

2.1.3. Redes Inalámbricas

Las Redes Inalámbricas permiten conectar diversos dispositivos dentro del área de cobertura de la red Inalámbrica sin utilizar una conexión física. Estas redes son altamente afectadas debido a que las señales viajan libremente y al presentarse más señales compartiendo el mismo medio se presentan las colisiones afectando la calidad de la señal según lo menciona [3] en su boletín estadístico.

2.1.4. Proceso de Extracción de Conocimiento

El autor [5] fundamenta que el *Proceso de Extracción de Conocimientos*(KDD¹) se describe como un proceso no intrascendental de identificación de patrones, legibles y útiles, el cual consta de varias fases:

- *Selección.* Se recopilan los datos a investigar determinando si la información es útil, se seleccionan las variables más relevantes y se aplican las técnicas de muestreo adecuadas.
- *Exploración.* Cuando la información proviene de diferentes fuentes, es necesario la exploración de los datos utilizando técnicas de análisis con el fin de deducir la distribución de datos.
- *Limpieza.* En esta fase se procesa a detectar y examinar los datos atípicos que se encuentren, imputando la información carente y eliminando o rectificar los datos erróneos.
- *Transformación.* En el caso de ser necesario se procederá a la transformación de los datos, mediante técnicas de reducción tales como: discretización y/o numeración.
- *Minería de datos.* Se aplican las técnicas necesarias sobre un conjunto de datos pre-procesado.
- *Evaluación.* Se interpretan de los resultados y se evalúan los modelos, si es necesario se vuelve a las fases anteriores para obtener resultados diferentes.

2.1.5. Técnicas de Minería de Datos

El autor [6], establece que las técnicas de minería de datos, son utilizadas hace algunos años para obtener la extracción de la información y de patrones. Existen varios tipos de técnicas entre ellas las técnicas: predictivas que basan en la predicción, descriptivas que se basan en la descripción y auxiliares que basan en la verificación.

¹ KDD= Por sus siglas en ingles “Knowledge Discovery in Databases”

2.1.6. Árbol de Decisión

Los árboles de decisión son utilizados para predecir la pertenencia de objetos en diferentes clases, teniendo en cuenta los valores que corresponden a sus atributos. Además, el método de árbol de decisión es una de las principales técnicas de minería de datos donde la flexibilidad de esta técnica la hace particularmente atractiva, sobre todo porque presenta la ventaja de una visualización muy sugestiva según lo menciona el autor [7].

2.1.7. Red Neuronal

El autor [8] menciona que las redes neuronales son capaces de resolver problemas que tienen patrones imprecisos de datos que contengan información incompleta y ruidosa con un gran número de variables. Esta característica de tolerancia a fallos es atractiva para problemas de minería de datos porque los datos reales suelen estar sucios y no siguen estructuras de probabilidad claras que normalmente requieren los modelos estadísticos.

2.1.8. Análisis discriminante

El análisis discriminante entrega un conjunto de funciones discriminantes que separan los máximo puntos de datos pertenecientes a una cierta clase de una variable objetivo. Los resultados del análisis discriminante pueden servir de soporte para el preprocesamiento de datos, así como el ajuste de parámetros para otros algoritmos de aprendizaje según lo establece el autor [9].

2.1.9. Herramienta de minería de datos

El autor[10] opina que las herramientas de minería de datos, permiten extraer patrones, para explicar, predecir y entender los datos. Estas herramientas suelen necesitar la existencia

previa de un almacén de datos(*DW*²), el cual es la colección de datos orientada a un dominio, no volátil para ayudar en la toma de decisiones.

2.1.10. RapidMiner

El autor y los coautores [11] mencionan que fue desarrollada con el fin de ofrecer un entorno integral para la minería de datos, la cual está diseñada como un sistema de apoyo para la realización de proceso de minería de datos, como lo es la preparación de los datos, la eliminación de los outliers, la visualización de los datos, la implementación de los algoritmos permitiendo obtener los resultados de los datos.

2.1.11. Lenguaje de Mercado de Modelos de Predicción

Los autores en su artículo [12] establecen que Lenguaje de Mercado de Modelos de Predicción(*PMML*³) está basado en Lenguaje de Mercado Extensible(*XML*⁴) que permite exportar modelos descriptivos y predictivos a diferentes aplicaciones.

2.1.12. Plataforma Móvil

Según el autor [13] la plataforma móvil es un sistema operativo en los teléfonos móviles los cuales poseen características que son importantes para un desarrollo, ya que pueden cambiar los lenguajes y herramientas donde se generan las aplicaciones nativas.

² DW= Por sus siglas en inglés Data WareHouse

³ PMML= Por sus siglas en inglés Predictive Model Markup Language

⁴ XML=Por sus siglas en inglés Extensible Markup Language

2.1.13. Entorno de Desarrollo Integrado

El autor[3] menciona que el Entorno de Desarrollo Integrado(*IDE*) es software que posee una interfaz que brinda herramientas para la elaboración de aplicaciones a partir de componentes. Por lo general son: una paleta, un lienzo, compiladores, depuradores, entre otros.

2.2. Marco Teórico

En el Marco Teórico se establecen las bases teóricas que servirán de apoyo para el desarrollo de la investigación, se abordarán definiciones y conceptos de varios autores que ayudarán a formar una idea global del problema que se investiga.

2.2.1. Redes de Telecomunicaciones Inalámbricas

En la presente investigación se abordará solo las redes inalámbricas como objeto de estudio. Las Redes Inalámbricas consisten en conectar diversos dispositivos dentro de un área de cobertura, este tipo de conexión se logra mediante el uso de ondas electromagnéticas y el rango de frecuencias que puede variar dependiendo el medio de transmisión como: ondas de radio, microondas terrestres, microondas por satélites e infrarrojos.

Estas señales viajan libremente y comparten el mismo medio de transmisión ocasionando la pérdida de la señal que se ve afectada por: interferencias, ruido, atenuación, absorción atmosférica, etc. Los tipos de tecnologías inalámbricas más empleadas en el Ecuador según el ARCOTEL son:

- *Internet Inalámbrico Fijo Terrenal.* Este tipo de acceso permite conectar a Internet a los usuarios a través de un punto de red fijo, a medida que el usuario se aleje del punto de acceso la calidad de la transmisión será menor y se presentara interferencia. Dentro de este tipo de tecnología se presentan varias redes inalámbricas:
 - *Red de Área Personal Inalámbrica (WPAN).*
 - *Red de Área Local Inalámbrica (WLAN).*

- *Red de Área Metropolitana Inalámbrica (WMAN).*
- *Red de Área Ampliada Inalámbrica (WWAN)*
- *Internet Móvil. El servicio de internet móvil se encuentra destinado a las redes celulares, en el Ecuador presentan una tendencia creciente desde año 2009 debido a la evolución de la tecnología y la facilidad de acceder a internet móvil mediante un dispositivo conocido como Smartphone.*
- *Internet Satelital. El servicio de Internet Satelital es utilizado como enlace de respaldo de las operadoras fijas o móviles en caso que se presente afectaciones en alguno de sus enlaces terrestres, también es usada en zonas rurales.*

2.2.2. Proceso de Extracción de conocimientos, Pre-procesamiento y Sistemas de minería de datos

El autor [14] menciona que el Proceso de Extracción de Conocimientos es una búsqueda de conocimientos de alto nivel a partir de un conjunto de datos, el cual es utilizado para presentar y analizar datos para la toma de decisiones. Este proceso implica varias fases la cuales se mencionarán a continuación:

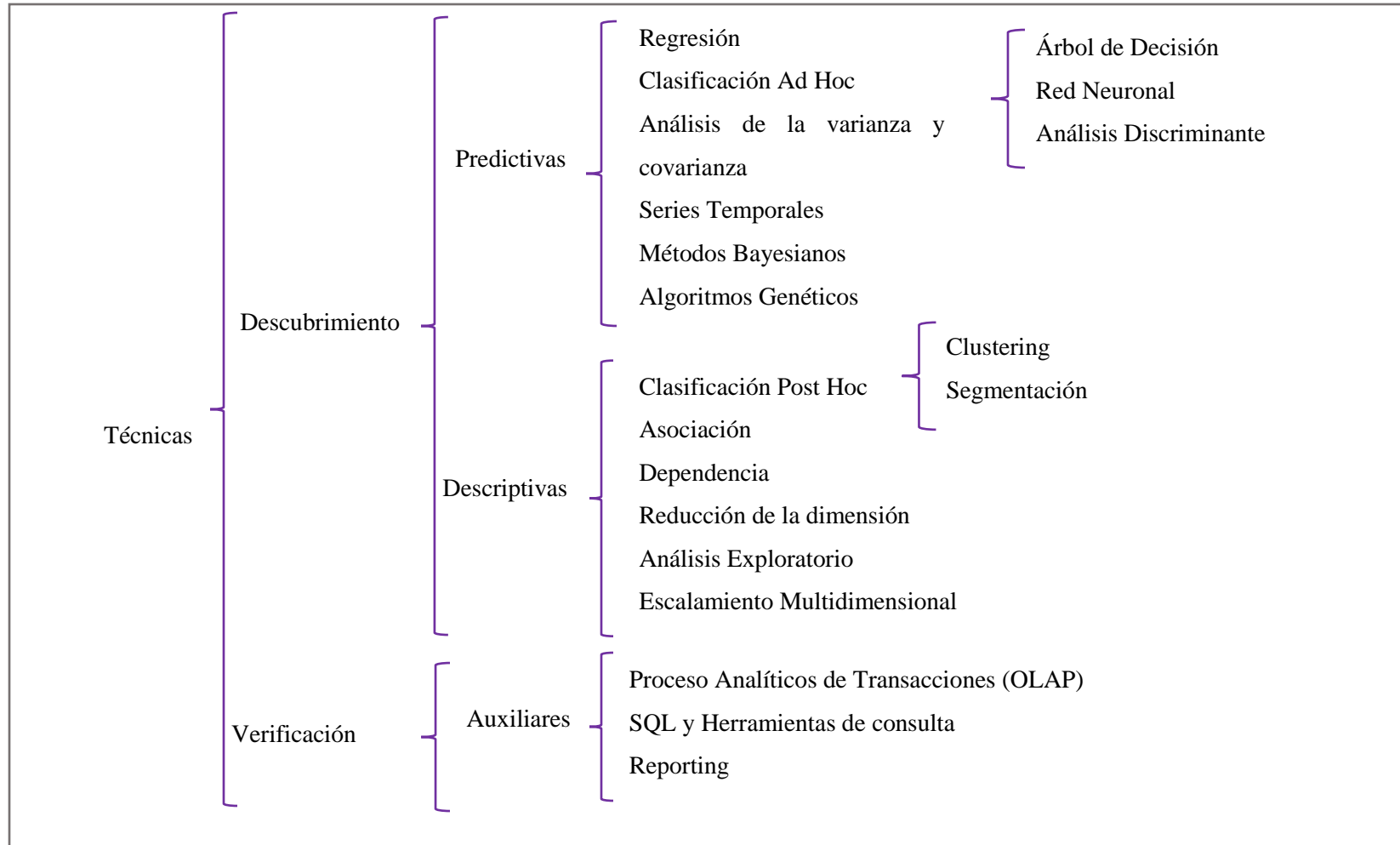
- *Selección:* En esta fase se selecciona o segmenta los datos que son relevantes.
- *Preprocesamiento:* Es la etapa de limpieza de datos donde se extrae información innecesaria del conjunto de datos.
- *Transformación:* Los datos son transformados para ser utilizados adecuadamente en la fase de minería de datos.
- *Minería de Datos:* Esta etapa se refiere a la extracción de patrones de los datos y a la generación de un modelo o resultado.
- *Interpretación y Evaluación:* Los patrones obtenidos en la etapa de minería de datos se convierten en conocimiento que posteriormente, son utilizados para apoyar la toma de decisiones.
- *Visualización de Datos:* Ayuda al analista a obtener una comprensión más profunda e intuitiva de los datos.

Las Técnicas de Minería de datos son utilizadas para la extracción de la información y descubrimiento de patrones. Los autores [10] mencionan que las técnicas se clasifican en: *Técnicas Predictivas*, *Técnicas Descriptivas* y *Técnicas Auxiliares*.

- Las *Técnicas de Clasificación Predictivas* determinan el modelo para los datos en base a un conocimiento teórico previo. Estas técnicas son conocidas como *Técnicas de clasificación Ad Hoc* debido a que se clasifican de forma individual o grupal, dentro de la clasificación Ad Hoc se encuentran los siguientes algoritmos:
 - *Árbol de Decisión.*
 - *Red Neuronal.*
 - *Análisis Discriminante.*
- Las *Técnicas de Clasificación Descriptivas* no supone la existencia de variables endógenas, exógenas y de un modelo previo para los datos. Estas técnicas suelen denominarse *Técnicas de clasificación Post Hoc*, debido a que realizan clasificaciones sin especificar grupos y describen patrones, esta clasificación consta de los siguientes algoritmos.
 - *Clustering*
 - *Segmentación*
- Las *Técnicas de Clasificación Auxiliares* son nuevos métodos basados en técnicas estadísticas, descriptivas, consultas e informes y se encuentran enfocadas hacia la verificación. Estas técnicas se clasifican en:
 - *Proceso Analíticos de Transacciones (OLAP)*
 - *SQL*
 - *Reporting*

En la Gráfico 1 se observa la clasificación de las técnicas de minería de datos.

Gráfico 1: Clasificación de las Técnicas de Clasificación.



FUENTE: MINERÍA DE DATOS: TÉCNICAS Y HERRAMIENTAS- CÉSAR PÉREZ; DANIEL SANTÍN[10].

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

En la Tabla 1, se detalla las características de las técnicas predictivas y descriptivas, las cuales se basan en el descubrimiento del conocimiento en los datos.

Tabla 1: Características de la clasificación de las Técnicas de Minería de datos.

	<i>Predictivas</i>	<i>Descriptivas</i>
<i>Características</i>	<ul style="list-style-type: none"> • Predicen el valor de los atributos de un conjunto de datos. • Su fase entrenamiento consiste en la construcción de un modelo usando subconjunto de datos. • Su fase prueba consiste en probar los modelos empleados sobre los datos. • Se encuentran orientadas a describir un conjunto de datos. • Analizan las características de una variable en función de una o varias muestras. 	<ul style="list-style-type: none"> • Descubren Patrones y tendencia en los datos. • Sirve para llevar a cabo acciones y obtener beneficios de negocios o científicos. • Se encuentran orientadas a estimar valores. • Proporciona relación entre los datos y sus características. • Agrupan conjuntos de datos en base a sus características

FUENTE: CONOCIMIENTOS PROPIOS.

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

En la Tabla 2 se muestra los diferentes sistemas de minería de datos, junto con las técnicas que pueden procesar cada uno de ellos y si soportan el lenguaje PMML⁵, el cual se estudiara posteriormente.

⁵ PMML: Lenguaje de Marcado de Modelos de Predicción.

Tabla 2: Cuadro Comparativo de los Sistemas de Minería de Datos

<i>Características</i>	<i>Weka</i>	<i>RapidMiner</i>	<i>Clemetine</i>	<i>R</i>	<i>Knime</i>	<i>SAS</i>
Tipo de Software	Libre	<i>Libre</i>	Libre	Libre	Libre	Libre
Licencia Libre	Sí	<i>Sí</i>	No	Sí	Sí	No
Licencia Estudiantil	No	<i>Sí</i>	No	No	No	No
Multiplataforma	Sí	<i>Sí</i>	No	Sí	Sí	Sí
Arquitectura Cliente/Servidor	No	<i>Si</i>	Si	No	No	No
Acceso a base de datos	Sí	<i>Sí</i>	Sí	Sí	Sí	Sí
Modelos de clasificación	Sí	<i>Sí</i>	Sí	Sí	Sí	Sí
Implementa arboles de decisión	Sí	<i>Sí</i>	Sí	Sí	Sí	Sí
Implementa redes neuronales	Sí	<i>Sí</i>	Sí	Sí	Sí	Sí
Implementa leyes bayesianas	Sí	<i>Sí</i>	Sí	No	Sí	Sí
Modelo de regresión	Sí	<i>Sí</i>	No	Sí	Sí	Sí
Permite visualización de datos	Sí	<i>Sí</i>	Sí	Sí	Sí	Sí
Clustering y agrupamiento	Sí	<i>Sí</i>	No	Sí	Sí	Sí
PMML	No	<i>Sí</i>	No	Sí	Sí	Sí

FUENTE: CONOCIMIENTOS PROPIOS.

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

A continuación, en la Tabla 3, se presentan los enlaces *URL* para descargar las diferentes herramientas de minería de datos.

Tabla 3: Enlaces URL de Herramientas de Minería de Datos

<i>Herramientas</i>	<i>Descripción</i>	<i>Enlaces URL</i>
Weka	Es un conjunto de librerías Java destinado para la extracción de conocimiento de datos según lo menciona [15].	http://www.cs.waikato.ac.nz/ml/weka/downloading.html
<i>RapidMiner</i>	Consiste en una aplicación open-source que utiliza una interfaz visual para la ejecución de procesos según lo establecen [16].	https://my.rapidminer.com/nextus/account/index.html#downloads
<i>Clementine</i>	Es una herramienta integrada de minería de datos destinada al proceso de extracción de conocimientos según lo menciona[10].	http://pscclementinepro.pl/index_en.html
<i>R</i>	Es un lenguaje que se encuentra más orientado a objetos que otras herramientas analíticas según lo menciona [17].	https://cran.r-project.org/mirrors.html / http://cran.espol.edu.ec/
<i>Knime</i>	En la página oficial de Knime [18] mencionan que es una plataforma de código abierto el cual es rápido de implementar, es escalable e intuitiva de aprender	http://www.knime.org/downloads/overview

<i>Herramientas</i>	<i>Descripción</i>	<i>Enlaces URL</i>
SAS	Es un proveedor analítico predictivo que incluye gestión de datos, herramienta de visualización, el cual ofrece muchas herramientas analíticas de minería de datos según lo menciona [17].	https://www.sas.com/en_us/software/enterprise-miner.html

FUENTE: CONOCIMIENTOS PROPIOS.

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Para llevar a cabo el pre-procesamiento de los datos, se decidió utilizar la herramienta de minería de datos *RapidMiner* la cual es de código abierto y ofrece una licencia tipo estudiantil con fines investigativos y a su vez trabaja en todos los sistemas operativos⁶.

RapidMiner utiliza una arquitectura *cliente/servidor*; realiza las operaciones de minería de datos, modelos predictivos y descriptivos, transformación de datos a nivel de analistas y permite implementar sistemas de scoring⁷, plataforma web de publicación de informes, diseño y navegación Web de informes e integridad vía servicios web a nivel de servidor.

Es una herramienta con una interfaz amigable que puede ser usada por personas con conocimientos básicos en el campo de la minería de datos, debido a que no requiere conocimientos avanzados y puede emplearse los modelos predictivos, por esta razón es favorable utilizar la herramienta de RapidMiner para realizar el pre-procesamiento y la visualización de los datos.

RapidMiner permite hacer uso del paquete PMML para exportar el modelo de mayor aceptación de pérdida de señal y de esta manera poder ser usado en el software *Android Studio*.

⁶ Multiplataforma.

⁷ Scoring: Herramienta estadística que ayuda a la toma de decisiones.

2.2.3. Lenguaje de Marcado de Modelos de Predicción.

Los autores [12], mencionan que PMML sigue una estructura intuitiva para la predicción de modelos de minería de datos y posee los siguientes componentes: *Cabecera*, *Diccionario de Datos*, *Transformación de datos*, *Modelo*, los cuales encapsulan funcionalidades diferentes en relación de los datos: entrada, modelo y salida.

Cabecera (Header): contiene información general del documento de PMML, como: la información de derechos de autor, su descripción, la información sobre el software que utiliza para generar el modelo y la versión. Además, contiene un atributo para una marca de tiempo que se puede utilizar para especificar la fecha de creación del modelo.

Diccionario de datos (Data Dictionary): contiene todas las definiciones de los posibles campos utilizados por el modelo. Este componente se define como un tipo de operación de atributo ya sea continua, categórica u ordinal.

Transformación de datos (Data Transformations): permite el mapeo de los datos del usuario para ser utilizado en la minería. PMML define varios tipos de transformación de datos como:

- *Normalización*. Valores de mapa a números, la entrada puede ser continua o discreta.
- *Discretización*. Mapa de valores continuos de valores discretos.
- *Asignación de valores*. Mapa de valores discretos de valores discretos.
- *Funciones de encargo incorporado*. Derivación de un valor mediante la aplicación de una función de uno o más parámetros.
- *Agregación*. Se utiliza para resumir o recolectar grupos de valores.

Modelo (Model): contiene la definición del modelo y posee un nombre del modelo, nombre de función ya sea de clasificación o regresión y atributos específicos. La representación del componente Modelo inicia con un esquema de minería y luego con una representación real del modelo, las cuales se describen como:

- *Esquema de minería (Mining Schema)*: enumera una lista de los campos utilizados en el modelo, esto puede ser un subconjunto de los campos según se lo determina el diccionario de datos el cual contiene información de cada campo como: *Nombre*, *Tipo de uso*, *tratamiento de valores atípicos*, *políticas de reemplazo de valores perdidos* y *tratamiento de valores perdidos*.
 - *Nombre del atributo*. Debe referirse a un campo en el diccionario de datos.
 - *Tipo de uso o atributo usageType*. Define la forma de un campo para ser utilizado en el modelo.
 - *Tratamiento de valores atípicos o atribuir valores atípicos*. Define el tratamiento de valores atípicos para ser usado⁸.
 - *Política de reemplazo de valores perdido o atributo missingValueReplacement*. Un valor que falta se sustituye automáticamente por los valores dados.
 - *Tratamiento de valor perdido o atributo missingValueTreatment*. Indica cómo se derivó la sustitución de valor que falta.
- *Objetivo (Targets)*: permite el procesamiento posterior del valor de referencia en el formato de escala, si la salida del modelo es continua.
 - *Modelo Específico (Model Specifics)*. Una vez obtenido el esquema de los datos se procede a especificar el detalle del modelo real.
 - *Salida*. Este elemento puede ser utilizado para designar a todos los campos de salida deseadas que se esperan del modelo.

El formato PMML puede ser utilizado en diferentes modelos de minería de datos dependiendo del sistema de minería de datos se puede extraen unos u otros. En la Tabla 4 se muestra la compatibilidad entre las diferentes técnicas de minería de datos que soportan PMML, la herramienta de minería de datos a utilizar y la extensión que ofrece la herramienta para manejo de dicho lenguaje.

⁸ En PMML, valores atípicos pueden ser tratados como valores que faltan, como valores extremos.

Tabla 4: Extensión PMML en RapidMiner

<i>Técnicas de Minería de Datos</i>	<i>PMML</i>	<i>RapidMiner</i>	<i>Extensión PMML - RapidMiner</i>
Algoritmos Genéticos	No	Sí	No
Análisis de la Varianza y Covarianza	No	No	No
Análisis Exploratorio	No	No	No
Árbol De Decisión	Sí	Sí	Sí
Asociación	Sí	Sí	Sí
Clustering	Sí	Sí	Sí
Dependencia	No	No	No
Análisis Discriminante	No	Sí	No
Escalamiento Multidimensional	No	Sí	No
Métodos Bayesianos	Sí	Sí	Sí
Máquinas de Soporte Vectorial	Sí	Sí	No
Redes Neuronales	Sí	Sí	Sí
Reducción de la Dimensión	No	No	No
Regresión	Sí	Sí	No
Regresión Logística	Sí	Sí	Sí
Regresión Lineal	Sí	Sí	Sí
Segmentación	No	Sí	No
Series Temporales	Sí	No	No
K-means	Sí	Sí	Sí

FUENTE: CONOCIMIENTOS PROPIOS.

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

2.2.4. Tecnologías para el Desarrollo Móvil

Para el desarrollo de aplicaciones móviles es necesario la utilización de plataformas móviles las cuales son un sistema operativo donde cada una establece sus propias reglas, requisitos y

regulaciones. A continuación se describen algunas de las plataformas para el desarrollo de *App móviles*[19]:

- *Android*: Es un sistema operativo desarrollado por *Google* y *Open Handset Alliance*(OHA⁹) el cual se encuentra destinado para ser utilizado por teléfonos móviles, google tv, tablets, entre otros dispositivos. Android está basado en un *Kernel Linux Monolítico* y utiliza una *licencia Apache 2.0* y *GNU GPL 2*.

Desde el lanzamiento de Android existen varias actualizaciones las cuales arreglan problemas que se han ido presentando y agregan nuevas funciones a los dispositivos; en la Ilustración 38, se observa la *API* más utilizada a lo largo del desarrollo del presente proyecto.

- *IOS*: Es sistema operativo propio de la multinacional *Apple Inc*, el cual está desarrollado para el uso exclusivo de los equipos *iPhone* como: teléfonos inteligentes, iPod touch y el iPad. Está basado en un Kernel híbrido *XNU*¹⁰ y utiliza una *Licencia de fuente pública de Apple*(*APSL*¹¹) y *Apple EULA*¹².
- *Fire OS*: Es un sistema operativo móvil basado en Android el cual es producido por *Amazon* y trabaja bajo las plataformas de 32 – 64 bit *ARM*¹³. Está basado en un *Kernel Linux Monolítica* y trabaja bajo una licencia *EULA* basada en *Apache License 2.0* y *GNU GPL v2* y destinado a los equipos: *Kindle Fire*, *Fire HDX*, *Amazon Fire Tv*, *Fire Phone*
- *Ubuntu Móvil*: Es un sistema operativo móvil desarrollado por *Canonical Ltd.*, la cual es la firma creadora de *Ubuntu Linux*, está basado en un *kernel Monolítico de Linux* y trabaja bajo un software libre. Este sistema está orientado a dispositivos inteligentes, tablets, ordenadores, además, puede ser ejecutado en servidores de red convirtiéndolo en uno de los servidores más populares de distribución de Linux ya que puede ejecutarse en la nube.

⁹ OHA: Confederación de compañías de software, Hardware y Telecomunicaciones.

¹⁰ XNU: X is Not Unix.

¹¹ Por sus siglas en inglés: Apple Public Source License.

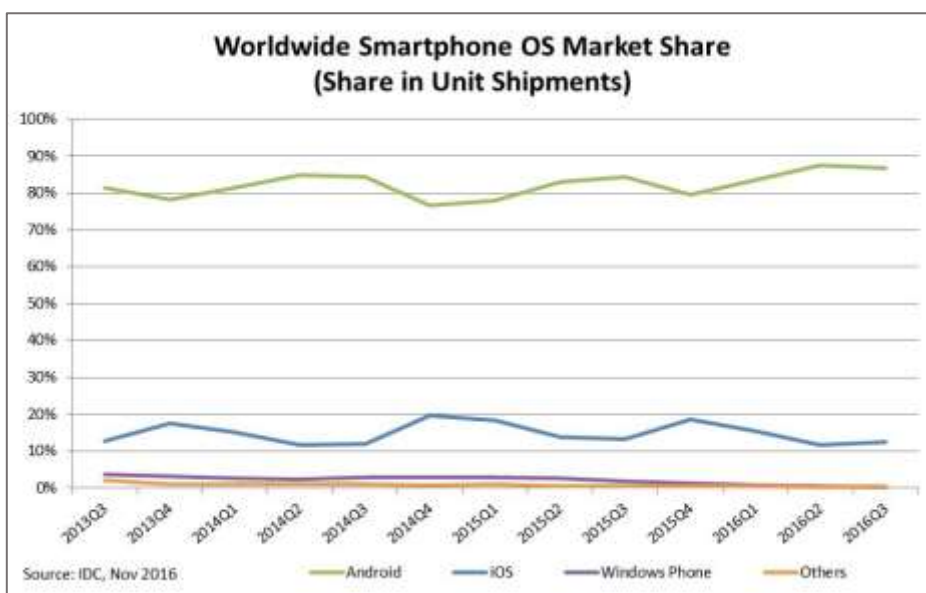
¹² Licencia de software entre el autor y el usuario del programa informático.

¹³ ARM Por sus siglas en inglés: Acorn RISC Machine.

Se decidió trabajar en la plataforma Android por las siguientes razones: su lenguaje de programación es similar a Java, la empresa auspiciante (*Google*) saca al mercado nuevas versiones de APIs periódicamente, cuenta con varias comunidades de desarrollos activos, incorpora un SDK nativo y su documentación respectiva.

Android cubre una gran parte del mercado a nivel mundial en el uso de dispositivos móviles inteligentes. Así como IDC en su página oficial[20] demuestra según sus estadísticas que en el tercer trimestre del 2016 Android domina el mercado con un 86.8%. A continuación, se muestra una ilustración de las estadísticas recopilatorias correspondientes al período del tercer trimestre del 2013 al tercer trimestre del 2016.

Ilustración 1: Estadísticas del mercado de los teléfonos inteligentes



FUENTE: CORPORACIÓN INTERNACIONAL DE DATOS [20]

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

El Entorno de Desarrollo Integrado (IDE¹⁴) es una aplicación visual utilizada para la elaboración de aplicaciones a partir de sus componentes según menciona el autor[3]. Por otra

¹⁴ Por sus siglas en inglés: Integrated Development Environment

parte el autor [21] opina que IDE se encuentra diseñado para la elaboración y ejecución de un programa en Java.

Por esta razón se determina que el IDE es un conjunto de procesos, herramientas de programación, compilador y depuración para el desarrollo de aplicaciones. A continuación, se mencionarán algunos Entornos de desarrollo: Android Studio, NetBeans y Eclipse:

- *Android Studio*: El autor [22] menciona que Android Studio es un IDE que trabaja en un sistema operativo multiplataforma, trabaja bajo una licencia Apache 2.0. Una vez instalado el Android Studio se descarga el *Android SDK* el cual contiene el conjunto completo de herramientas de desarrollo¹⁵ y puede funcionar en cualquier sistema operativo como: *Windows, Linux, Mac OS X* según argumenta el autor[23] .
- *NetBeans*: El autor [24] opina que NetBeans es un IDE de código abierto, que es mejor que otros IDEs disponibles en el mercado para el desarrollo de aplicaciones Java. Además, ofrece una total flexibilidad y permite desarrollar aplicaciones como: aplicaciones de escritorio, aplicaciones web y aplicaciones móviles, gestionar proyectos, editar y generar código.
- *Eclipse*: es un IDE que puede ser ejecutado en varios sistemas operativos como: Windows, Mac OS X and Linux y es necesario que se encuentre instalado Java para funcionar según menciona el autor[25].

El lenguaje de programación es un lenguaje especial destinado para realizar procesos, desarrollar programas o aplicaciones dentro de un entorno de desarrollo, A continuación, se mencionará algunos de los lenguajes de programación comúnmente utilizados:

- *PHP*: Es un lenguaje interpretado del lado del servidor que se singulariza por su versatilidad, flexibilidad, robustez y potencia según lo menciona el autor [26]. Además, por su flexibilidad resulta ser fácil de aprender, es un lenguaje multiplataforma y se tiene accesibilidad al código fuente de forma gratuita.
- *SQL*: El autor [27] indica que SQL es un lenguaje estándar de alto nivel que es implementado por la mayoría de las *Sistema Gestor de Base de datos(SGBD)*¹⁶,

¹⁵ Herramientas de desarrollo: depurador, biblioteca, emulador, documentación, ejemplo de código y tutoriales.

¹⁶ SGBD: Sistema Gestor de Base de datos.

donde el usuario especifica el resultado que desea y la SGBD se encarga de decidir el orden de ejecución de las operaciones que se desean aplicar sobre los datos.

- *Java*: Es un lenguaje de programación de alta gama orientado a objetos, multiplataforma, sencillo, robusto, de alto rendimiento, orientado a objeto. El cual posee una librería de clases que permite comunicarse con programas ejecutados en ordenadores remotos y permite desarrollar programas concurrentes o multi-hilos para conseguir un aprovechamiento del procesador , según lo cita Pablo Garrido [28].
- *XML*: El lenguaje de marcado extensible está desarrollado por *Consortio Mundial de Red (W3C*¹⁷), tiene por objetivo superar las limitaciones de otros lenguajes de marcado y plantea un estándar destinado al intercambio de información entre plataformas, como lo menciona el autor [29].
- Adicionalmente a los lenguajes antes mencionados se usa *JSON*; el cual es un formato para el intercambio de datos en aplicaciones y no requiere el uso de XML. Existe un soporte para el JSON entre los lenguajes soportados se incluyen: PHP, C/C++, C#, Java, JavaScript, OCaml, Perl, ActionScript, Python, Rebol, entre otros, *según lo menciona el autor* [30]. Debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente.

2.3. Marco Legal

En el siguiente apartado se hace referencia a las leyes y artículos vigentes en el estado ecuatoriano referente al espectro radioeléctrico.

2.3.1. Constitución Política del Ecuador

En el Artículo 16 numeral 2 de la Constitución Política del Ecuador[31] establece que:” Todas las personas tienen derecho al acceso universal de las tecnologías de información y comunicación”.

¹⁷ Por sus siglas en inglés: World Wide Web Consortium.

2.3.2. Ley Orgánica de Telecomunicaciones

El **Artículo 1** [32] determina que la Ley Orgánica de Telecomunicaciones tiene como objetivo desarrollar un régimen general de telecomunicaciones y espectro radioeléctrico que comprenden la administración, control, regulación y gestión en todo el territorio ecuatoriano.

El en el **Artículo 6** [32] se define el espectro radioeléctrico como: “Un conjunto de ondas electromagnéticas que se propagan en el espacio para la prestación de servicios de telecomunicaciones, televisión, radiodifusión sonora, entre otros”.

En el **Artículo 18** [32] hace referencia que el espectro radioeléctrico es un bien de dominio público y es un recurso limitado del Estado. Su uso y explotación requiere un título habilitante emitido por el Agencia de Regulación y Control de las Telecomunicaciones (*ARCOTEL*).

El **Artículo 142** [32] establece que la Agencia de Regulación y Control de las Telecomunicaciones es la entidad encargada de administrar, regular y controlar las telecomunicaciones y el espectro radioeléctrico y su gestión. Para la asignación de frecuencia lo dispondrá la Ley Orgánica de Comunicación.

2.3.3. Ley Orgánica de Comunicación

El **Artículo 1** [33] determina que la Ley Orgánica de Comunicación tiene por objetivo desarrollar, proteger y regular el ejercicio de los derechos a la comunicación.

En el **Artículo 105** [33] se establece que la administración y aprovechamiento técnico del espectro radioeléctrico es un recurso publico estratégico que lo ejercerá el Estado Central. En el **Artículo 106** menciona que las frecuencias del espectro se encuentran destinadas al funcionamiento de estaciones de radio y televisión de señal abierta.

2.4.Marco Referencial

En el siguiente apartado se brinda una breve revisión de los trabajos más relevantes sobre el objeto de la investigación, esto es, aplicación móvil para la predicción de la pérdida de señal a partir de variables meteorológicas.

La tesis de maestría en conectividad y redes de ordenadores[1], realiza una investigación para obtener un modelo de minería de datos que caracterice la calidad de la señal inalámbrica en función de las variaciones meteorológicas. Se utilizaron dos herramientas de minería de datos: *Weka* para el pre-procesamiento de los datos y *Keel* para aplicar las reglas difusas; el algoritmo *FURIA-C* basado en lógica difusa obtuvo una *precisión de 0.8774* con una cantidad de 8 reglas y una *desviación estándar del 0.0199*.

En el año 2015, los autores Anochi y de Campos Velho [34], realizaron un estudio con el fin de predecir la variable de precipitación climática estacional en la región de Centro-Oeste de Brasil. Se empleó un método basado en la *Red Neural Artificial*¹⁸, el cual se aplica a la precipitación estacional de predicción climática utilizando el método de *Reducción de datos por la Teoría de Conjuntos Rugosos*, los resultados obtenidos muestran la efectividad de la metodología propuesta.

El autor [35] en colaboración con sus coautores, proponen un servicio de indexación de desastres climáticos urbanos el cual: detecta, predice y analiza la tendencia de varios riesgos como desastres y accidentes de seguridad. Este servicio es utilizado para la toma de decisiones en la gestión de desastres climáticos y tiene como fin determinar los riesgos basados en desastres naturales provocados por el hombre.

En la investigación enunciada por Chaudhari y Biradar [36], plantean un nuevo mecanismo de predicción en *Redes Móviles Ad-hoc*¹⁹ para predecir: tráfico de datos, espacio-búfer,

¹⁸ ANN: Red Neural Artificial.

¹⁹ MANESTs= Mobile Ad-hoc Networks.

energía y el ancho de banda necesario para la asignación de recursos. De esta manera el mecanismo propuesto servirá de apoyo a las comunicaciones en tiempo real y multimedia de los datos.

En el año 2016 los autores [37], exponen un estudio sobre la obtención de un modelo de minería de datos, se emplearon los sistemas: *Sistema Global para comunicaciones Móviles* y *Sistema de Posicionamiento Global* para recopilar los datos meteorológicos, con el fin de evaluar la capacidad de monitoreo de la lluvia, se emplearon varios algoritmos de clasificación tales como: *k-Nearest Neighbors*, *Support Vector Machine* y *Decision Tree* para identificar los tipos de precipitaciones (sin lluvia, lluvia ligera, lluvia fuerte). Con el fin de verificar la viabilidad de un sistema destinado a las alertas de lluvias.

En la Tabla 5, se observa que los temas están basados en el estudio de minería de datos empleando diferentes técnicas para la obtención de un modelo o la predicción de la precipitación climática. En el estudio [1] se realiza la predicción de pérdida de señal, pero no cuenta con una aplicación móvil. Por esta razón, es favorable el desarrollo del proyecto de investigación ya que no existe una investigación que desarrolle una aplicación móvil basado en un estudio de minería de datos destinado a la pérdida de señal.

Tabla 5: Temas de Marco Referencial

<i>Ref.</i>	<i>Autor y Coautores</i>	<i>Año</i>	<i>Minería de datos</i>	<i>Aplicación Móvil</i>	<i>Pérdida de Señal</i>
[1]	Rodolfo Najarro Quintero	2015	Sí	No	Sí
[34]	Juliana Aparecida Anochia1 y Haroldo Fraga de Campos Velho	2015	Sí	No	No
[35]	J.C. Kim, H.Jung, y K.Chung,	2016	Sí	No	No
[36]	S. Chaudhari y R. Biradar	2016	Sí	No	No
[37]	L.Brito y M. Albertini	2016	Sí	No	No

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

CAPÍTULO III

MÉTODOLOGÍA DE LA INVESTIGACIÓN

3.1.Localización

El proyecto de investigación se desarrolló en la Universidad Técnica Estatal de Quevedo, perteneciente al Cantón Quevedo, ubicada en la Avenida Quito, Km 1.5 Vía Santo Domingo. Se encuentra situada geográficamente en la Región Litoral Centro, la cual se encuentra ubicada al 1° 20' 30" de Latitud Sur y los 79° 28' 30" de Longitud occidental, dentro de una zona subtropical.

3.2.Tipo de Investigación

En el proyecto de investigación se aplicaron dos tipos de investigación: *Exploratoria* y *Documental*, las cuales se detallan a continuación:

3.2.1. Investigación Exploratoria

Se usó la Investigación Exploratoria para la obtención de nuevos resultados con un mismo conjunto de datos en base a una investigación previamente realizada.

3.2.2. Investigación Documental

El proyecto es Documental dado que se investigó los temas referentes al estudio de minería de datos para predecir la pérdida de señal y al desarrollo de aplicaciones móviles destinada a la pérdida de señal desde el año 2014 al 2016 para trabajar con estudios recientes.

3.3.Métodos de Investigación

Para el desarrollo del proyecto investigativo se utilizaron dos tipos de métodos: *Extracción de conocimiento de un conjunto de datos* destinado para el estudio de minería de datos y *Programación Extrema* para el desarrollo de la aplicación móvil.

3.3.1. Método de Extracción de Conocimiento de Conjuntos de Datos

Para el desarrollo de la minería de datos se hizo uso del Método de Extracción de Conocimientos de un Conjunto de Datos, el cual consta de cinco fases:

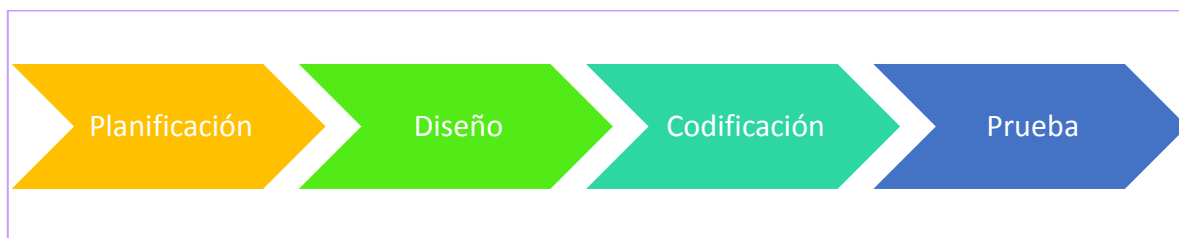
- Lo primero que se realizó fue la recopilación de datos, el conjunto obtenido corresponde a la tesis [1], en el período de enero del 2013 hasta junio del 2014, con un total de 546 registros.
- Consecuentemente, se realizó el proceso de exploración de datos y se definió la variable dependiente que determinará si existe o no la pérdida de señal. Para el procesamiento de datos, se utilizó la herramienta RapidMiner, la cual se analiza en la Tabla 2.
- Por consiguiente, se efectuó el pre-procesamiento para detectar y eliminar los valores atípicos y las correlaciones entre variables, obteniendo como resultado un nuevo conjunto de datos.
- Luego del pre-procesamiento de datos se llevó a cabo el proceso de optimización de parámetros de los diferentes modelos, permitiendo obtener los valores recomendados para generar un modelo aceptable.
- Una vez obtenido el modelo se interpretó y evaluó, permitiendo verificar que el resultado sea válido y satisfactorio. Se generaron varios modelos con el uso de distintas técnicas y se procedió a comparar los resultados de los modelos utilizando.

3.3.2. Método de Desarrollo Programación Extrema

Como se puede observar en el [Anexo 7.3](#) se presenta una descripción, fases y características de cada una de las metodologías a analizar. Para el desarrollo de la aplicación móvil se utilizó, el método de desarrollo denominada Programación Extrema (XP)²⁰, el cual se basa en una metodología liviana de desarrollo que está sujeta a frecuentes cambios a lo largo del ciclo de vida del proyecto.

Para garantizar la calidad del software se debe especificar hacia quien va dirigido el proyecto, en este caso, hacia los especialistas en telecomunicaciones; ya que a lo largo del desarrollo la estructura del proyecto puede variar, pero la finalidad se mantendrá. En el Gráfico 2 se presenta las fases de la metodología XP a utilizar:

Gráfico 2: Fases de la Metodología XP



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

²⁰ XP= Por sus siglas en inglés: *Xtreme- Programming*

3.4.Fuentes de Recopilación de la Información.

La fuente de recopilación que se empleó son las siguientes: entrevista, para la obtención de datos y la documentación de libros, artículos, entre otros.

3.4.1. Fuentes Primarias

Para la implementación del proyecto de investigación el conjunto de datos obtenido corresponde a la tesis [1] perteneciente al Ing. Rodolfo Najarro Msc. El conjunto de datos es del período de enero del 2013 hasta junio del 2014,

3.4.2. Fuentes Secundarias

- Tesis de Ingeniería utilizando minería de datos.
- Artículos Científicos acerca de minería de datos.
- Tesis de Ingeniería en diseño de aplicaciones móviles.
- Artículos Científicos sobre aplicaciones móviles.
- Artículos Científicos respecto a la utilización del paquete PMML.
- Manuales de Usuario (RapidMiner, Android Studio, entre otros).

3.5.Diseño de la Investigación.

El diseño de la investigación es experimental puesto que se encarga de la experimentación de variables independientes en un ambiente controlado, dicho experimento es definido por el investigador y tiene como fin conocer el comportamiento de la variable dependiente.

3.6.Instrumento de Investigación

A continuación, se detallan los instrumentos de investigación necesarios para recolección y entendimiento de los datos.

3.6.1.1. Entrevista

Para la implementación del proyecto de investigación se realizó la recolección de la información, entrevistando a dos empresas: Corporación Nacional de Telecomunicaciones²¹ y TELCONET S.A. Se solicitó la información actualizada de la base de datos referente a la pérdida de señal, por diferentes motivos no se obtuvo repuestas de las entidades antes mencionada, se adjunta en [Anexo 7.1](#) y [Anexo 7.2](#).

3.7.Tratamiento de Datos

Para llevar a cabo el tratamiento de los datos se utilizó dos softwares; el primero a utilizar fue Excel donde se preparó los datos y se hizo la observación inicial, posteriormente, se utilizó la herramienta de minería de datos RapidMiner la cual realizó el pre-procesamiento de los datos y optimización de los modelos.

3.8. Recursos Humanos y Materiales.

Para el desarrollo del proyecto de investigación fue imprescindible los materiales a utilizar y el equipo de trabajo que sirvieron como sustento para las diferentes fases de la investigación. A continuación, se detalla el equipo de investigación y los materiales necesarios para el desarrollo de la aplicación e investigación.

²¹ CNT

3.8.1. Equipo Humanos

El proyecto de investigación cuenta con un grupo de tres personas, dos estudiantes que desarrollaron el presente proyecto: la Srta. Angie Sthepanie Aguilar Zambrano y el Sr. Gabriel Fernando Nuñez Villalba, los cuales fueron guiados por sus amplios conocimientos por el Director del proyecto de investigación el Ph.D. Pavel Novoa Hernández. Ellos trabajaron en la recopilación de la información, generación del modelo, desarrollo de la aplicación y documentación.

3.8.2. Materiales

En la Tabla 6 se observa el material de Hardware que se utilizó en el transcurso desarrollo del proyecto.

Tabla 6: Recursos Hardware

<i>Cantidad</i>	<i>Hardware</i>	<i>Descripción</i>
1	Computador Portátil	HP Envy dv7 <ul style="list-style-type: none">• 2,4 GHz Intel Core i7-3630QM• 4 GB RAM• 320 GB Disco Duro• 1 HDMI, 1 USB 2.0, 3 USB 3.0• Monitor, Mouse, Teclado
1	Computador de Escritorio	<ul style="list-style-type: none">• 3,6 GHz Intel Core i5-4760k• 8 GB RAM• 260 GB Unidad de Estado Sólido.• MSI GTX 660.• 1 VGA, 1 HDMI, 1 USB 2.0, 3 USB 3.0

<i>Cantidad</i>	<i>Hardware</i>	<i>Descripción</i>
		<ul style="list-style-type: none"> • DVD rWriter • Monitor, Mouse, Teclado
1	Router	Huawei HG532c <ul style="list-style-type: none"> • ADSL2 • 4 puertos Ethernet • Wi-Fi 802.11n • Velocidades hasta 300Mbps • Compatible con IPv6
1	Impresora	<ul style="list-style-type: none"> • Epson L365
	Documentación	<ul style="list-style-type: none"> • Ver Bibliografía

FUENTE: CONOCIMIENTOS PROPIOS.

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

En la Tabla 7 se observa los recursos referentes a los softwares utilizados en el desarrollo del proyecto.

Tabla 7: Recursos-Software

<i>Software</i>	<i>Descripción</i>
Sistemas Operativos	<ul style="list-style-type: none"> • Windows 10
Herramienta de minería de datos	<ul style="list-style-type: none"> • RapidMiner
Sistema Gestor de Base de Datos	<ul style="list-style-type: none"> • MYSQL
Plataforma de Desarrollo	<ul style="list-style-type: none"> • Android Studio • NetBeans 8.0.1 • Workbench
Sistema Operativo Móvil	<ul style="list-style-type: none"> • Android 4.0 Ice Cream Sandwich (API 15) o superior
Programas de Oficina	<ul style="list-style-type: none"> • Microsoft Word • Microsoft Excel

<i>Software</i>	<i>Descripción</i>
	<ul style="list-style-type: none"> • Microsoft Power Point
Programa de diseño y Desarrollo de caso de uso	<ul style="list-style-type: none"> • Microsoft Visio • Draw.io

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

En la Tabla 8 se observa varios recursos también importantes para el desarrollo del proyecto.

Tabla 8: Recursos- Otros

<i>Recursos</i>	<i>Descripción</i>
Internet	<ul style="list-style-type: none"> • Acceso a Internet para la respetiva investigación, compartir documentos y carpetas.
Suministro de oficina	<ul style="list-style-type: none"> • Bolígrafos • Lápices • Memorias USB • Hojas tamaño A4 • Impresora • Carpetas

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

3.9.Presupuesto

En la Tabla 9 se detalla el presupuesto del proyecto de investigación.

Tabla 9: Presupuesto

<i>N°</i>	<i>Cantidad</i>	<i>Descripción</i>	<i>V. Unitario(USD)</i>	<i>V. Total (USD)</i>
1	2	Servicio de Internet	23.41	46.82
2	3	Resma de Papel A4	4.00	12.00
3	4	Tinta para impresora continua	12.00	48.00
4	4	Carpetas	0.75	3.00
5	2	Disco Duro Portátil	90.00	180.00
6	4	Anillado	1.00	4.00
7	3	Lapiceros	0.50	1.50
8	4	Empastados	10.50	42.00
TOTAL (USD)				\$337.32

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

Este capítulo tiene como objetivo presentar los resultados y la discusión asociada a la investigación. Primeramente, se aborda los resultados obtenidos de la minería de datos, la cual transcurrió a partir del pre-procesamiento de los datos, luego se utilizaron técnicas predictivas para realizar la optimización de los modelos mediante la herramienta de minería de datos RapidMiner.

Una vez obtenido el modelo optimizado se procedió a exportarlo haciendo uso de un estándar modelado y finalmente ser usado en el desarrollo la aplicación móvil para la plataforma Android, la cual tiene como nombre *SignalPred* y está destinada a predecir la pérdida de la señal en función de las variables meteorológicas.

4.1. Minería de Datos.

4.1.1. Obtención del Conjunto de Datos

Para la implementación del proyecto de investigación procedió a utilizar la información existente de la tesis [1]. Dentro del proceso de investigación se encontraron diferentes aspectos que se analizaron e interpretaron de un conjunto de datos referente a la pérdida de señal. A continuación, se brindará la información del conjunto de datos primarios que poseen las siguientes características:

- Número de atributos: *16 - Incluido el atributo de clase.*
- Número de registros: *546*
- Datos perdidos: *0*
- Atributos con datos perdidos: *0*

Análisis Descriptivos de las variables

En la Tabla 10 se muestra las variables meteorológicas del conjunto de datos originales a analizar, se observa el: alias, medidas, rangos originales, rangos normales en Ecuador obtenidos de [38] y su descripción. Como se puede observar la mayoría de los rangos de las variables se encuentran entre los rangos permitidos a excepción de las variables Oscilación Térmica, Tensión del vapor y la lluvia que se encuentran fuera de los rangos establecidos normalmente.

Tabla 10: Descripción de variables meteorológicas

<i>Variables</i>	<i>Alias</i>	<i>Medidas</i>	<i>Rangos Originales</i>		<i>Rangos Normales</i>		<i>Descripción</i>
			<i>Mínimo</i>	<i>Máximo</i>	<i>Mínimo</i>	<i>Mínimo</i>	
Temperatura mínima	TMIN	Celsius	17.5°	24.4°	-5°	30°	Es la menor temperatura registrada en el día. Se registra desde las 7:00 pm y las 7:00 am.
Temperatura media	TMED	Celsius	21.4°	27.8°	0°	36°	Es la temperatura promedio entre la temperatura máxima y la mínima.
Temperatura máxima	TMAX	Celsius	24.2°	35.5°	5°	42°	Es la mayor temperatura registrada en un día. Se registra desde 7:00 am y las 7:00 pm.
Oscilación térmica	TOCS	Celsius	1.6°	114°	1°	20°	Es la diferencia entre la temperatura más alta y más baja registrada en el día
Humedad relativa mínima	HMIN	Porcentaje	35%	95%	20%	80%	Es el porcentaje mínimo de humedad relativa durante el día

<i>Variables</i>	<i>Alias</i>	<i>Medidas</i>	<i>Rangos Originales</i>		<i>Rangos Normales</i>		<i>Descripción</i>
			<i>Mínimo</i>	<i>Máximo</i>	<i>Mínimo</i>	<i>Mínimo</i>	
Humedad relativa media	HMED	Porcentaje	63%	96%	40%	90%	Es la máxima cantidad de vapor de agua que puede contener el aire saturado a una determinada temperatura.
Humedad relativa máxima	HMAX	Porcentaje	80%	100%	70%	100%	Es el porcentaje máximo de humedad relativa durante el día
Tensión de vapor de agua	TENVAP	Celsius	21.6°	31.1°	20°	30°	Presión que tendría el vapor de agua si ocupara solamente el volumen del aire húmedo.
Punto de rocío	PUROC	Celsius	2.2°	24.7°	1°	28°	Temperatura en donde el aire alcanza la saturación
Heliofanía	HELIO	Horas	0 h	8.6 h	0.1 h	10 h	Cantidad de horas de brillo solar en el día.
Evaporación	EVA	Milímetros	0 mm	6.6 mm	0 mm	10 mm	Es el cambio de estado líquido a estado gaseoso sin considerar la temperatura.

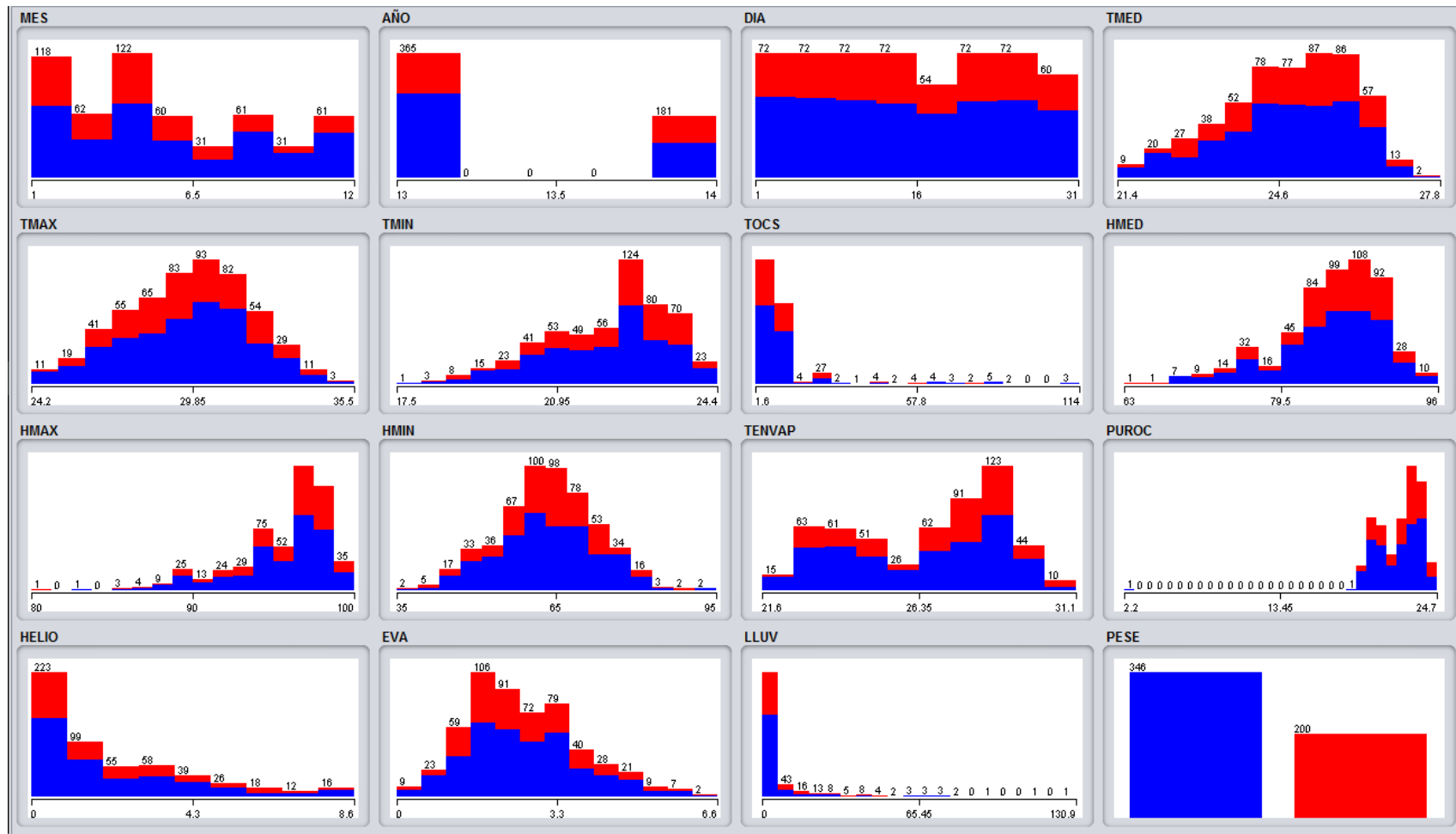
<i>Variables</i>	<i>Alias</i>	<i>Medidas</i>	<i>Rangos Originales</i>		<i>Rangos Normales</i>		<i>Descripción</i>
			<i>Mínimo</i>	<i>Máximo</i>	<i>Mínimo</i>	<i>Mínimo</i>	
Lluvia	LLUV	Milímetros	0 mm	130.9 mm	0 mm	45.4 mm	Se produce por la condensación del vapor de agua que contienen las nubes.
Pérdida de Señal.	PESE	booleano	0	1	0	1	Se produce por la disminución de la potencia por unidad de área.

FUENTE: TESIS DE MAESTRÍA DE TELECOMUNICACIONES-2015 [1]

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

A continuación, se presenta gráficamente las variables meteorológicas en función de la variable clase Pérdida de Señal (*PESE*): En la Ilustración 2 se observa el conjunto total de variables con un total de: 346 casos de señal (Azul) y 200 de interferencia (Rojo).

Ilustración 2: Descripción General del Conjunto de Datos elaborado en Weka.



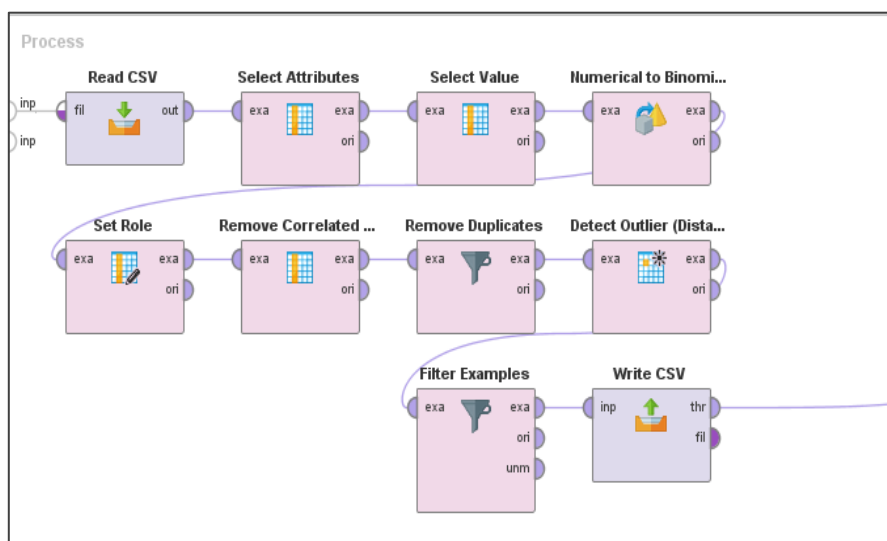
FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

4.1.2. Pre-procesamiento de los datos

El propósito de esta fase de pre-procesamiento, es la obtención de un nuevo conjunto de datos eliminando los datos atípicos y la correlación entre variables que no contribuyan favorablemente al proceso de predicción, se aplicó diversos métodos de pre-procesamiento, haciendo énfasis en la eliminación de outliers, correlaciones y datos duplicados. Para el estudio de la minería de datos se utilizó la herramienta *RapidMiner*, la cual se analiza en la Tabla 2. En la Ilustración 3, se presenta los operadores necesarios para el pre-procesamiento de los datos y su descripción.

Ilustración 3: Operadores de RapidMiner.



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- *Read CSV*. El operador *Read CSV* se utiliza para leer archivos CSV²². En este caso se fue empleado para importar la base de datos de las variables meteorológicas.
- *Select Attributes*. El operador *Select Attributes*, selecciona que atributo de un ExampleSet debe mantenerse y que atributo debe ser eliminado. En la Tabla 11 se muestra los parámetros a utilizar dentro del operador *Select Attributes*.

²² Valor separado por comas

Tabla 11: Parámetros del Operador Select Attributes

<i>Parámetro</i>	<i>Utilización</i>
<i>attribute_filter_type</i>	Se seleccionó <i>regular_expression</i> que permite especificar una expresión regular para la selección de atributo
<i>Invert selection</i>	Se habilitó la opción <i>invert_selection</i> para invertirla selección de los parámetros. En este caso se seleccionaron los atributos <i>Mes</i> , <i>Día</i> y <i>Año</i> los cuales serán descartados.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- *Select Value*: El operador *Select Value* selecciona los atributos que contengan valores numéricos. En la Tabla 12 se observa los parámetros a utilizar.

Tabla 12: Parámetros del Operador Select Value

<i>Parámetros</i>	<i>Utilización</i>
Attribute_filter_type	Nos aparece un despliegue con varios parámetros a utilizar. En este caso se seleccionó el filtro <i>value_type</i> el cual permite seleccionar todos los atributos de un tipo en particular.
Value_type	Nos aparece una lista despliegue con varios parámetros a utilizar. En este caso se seleccionó <i>attribute_value</i> .

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- *Numerical to Binominal*. El operador *Numerical to Binominal* se utiliza para cambiar el tipo de los atributos numéricos seleccionados a un tipo binominal, en este caso asignar true o false. En la Tabla 13 se observa los parámetros utilizados para el estudio de la minería.

Tabla 13: Parámetros del Operador Numerical to Binominal

<i>Parámetros</i>	<i>Utilización</i>
Attribute_filter_type	Se seleccionó el filtro <i>regular_expression</i> para especificar una expresión regular para la selección de atributos
Regular_expression	Se especificó la variable “PESE” como variable a modificar.
Min	Se especificó valor de 1.0 para establecer el límite inferior del rango
Max	Se estableció de 2.0 para establecer el límite superior del rango

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- *Set Role*. El operador *Set Role* se utiliza para señalar uno o más atributos como atributos clase u objetivo. En la Tabla 14 se muestra los parámetros empleados en este operador.

Tabla 14: Parámetros del Operador Set Role

<i>Parámetros</i>	<i>Utilización</i>
attribute_name	Se asignó el nombre “PESE” que representa a la variable “Pérdida de señal” la cual será el atributo clase.
Target_role	Este atributo será tratado como objeto de predicción en el modelo a desarrollar, por esta razón se seleccionó <i>label</i> también conocida como <i>goal_variable</i> , la cual actúa como un atributo de destino para el aprendizaje de los operadores.
Set_additional_roles	Se selecciona para modificar las funciones de uno o más atributos.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- *Remove Correlated Attributes*. El operador *Remove Correlated Attributes* retira atributos correlacionados de una *ExampleSet*. En la Tabla 15 se observa los parámetros utilizados en el operador.

Tabla 15: Parámetros del Operador-Remove Correlated Attributes

<i>Parámetros</i>	<i>Utilización</i>
correlation	Se especifica la correlación de atributos filtrados, su rango es -1 a +1 para medir el grado de asociación entre los dos atributos. Se asignará el rango 0.8.
filter_relation	Se seleccionó <i>greater_equals</i> para determinar la correlación de dos atributos se comparen a la vez.
attribute_order	Se escogió el modo <i>random</i> para calcular de manera aleatoria los atributos, calcular las correlaciones y filtrar los atributos.
use_absolute_correlation	Se seleccionó para indicar si debe usar una semilla aleatoria local.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- *Remove duplicates*. El operador *Remove duplicates*, todos los registros que contengan los mismos valores son eliminados. La idea es quedarse con casos únicos de esta forma se contribuye a disminuir el sobre aprendizaje del modelo de predicción y por tanto su sesgo²³ a una determinada clase. En la Tabla 16 se utilizaron los siguientes parámetros.

²³ Sesgo: Orientar o dirección que toma un asunto.

Tabla 16: Parámetros del Operador- Remove Duplicates

<i>Parámetros</i>	<i>Utilización</i>
attribute_filter_type	Se escogió la opción <i>all</i> para filtrar la selección de los atributos por defecto.
incluye_special_attributes	Se utilizó para incluir los atributos especiales como: <i>label, id, prediction, cluster, weight y batch</i> .
treat_missing_values_as_duplicates	Se seleccionó este parámetro para especificar los valores no deben ser tratados como duplicados o no.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- *Detect Outlier*. El operador Factores atípicos Clase²⁴ identifica los valores atípicos en lo dado ExampleSet en base a los factores atípicos .El uso de este operador es para detectar valores atípicos que se salgan de la media normal o se encuentren muy dispersos en los datos a analizar. En la presente Tabla 17 se muestra los parámetros a utilizar en el operador *Detect Outlier*.

Tabla 17: Parámetros del Operador-Detect Outlier

<i>Parámetros</i>	<i>Utilización</i>
number_of_neighbors	Específica el valor de los <i>k vecinos</i> , en este caso se decidió trabajar con <i>15 k vecinos</i>
number_of_outliers	Se especifica el número de los valores atípicos. En este caso se asignó 346.
distance_function	se seleccionó <i>angle</i> para especificar la distancia que va haber entre los dos valores.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

²⁴ COF

- *Filter Examples*. El operador *Filter Examples* selecciona los cuales los valores de una ExampleSet que deben mantenerse y que los valores deben ser eliminados. El operador filtra aquellos registros que se consideran outliers.
- *Write CSV*. Este operador se utiliza para escribir archivos CSV. Una vez hecha la limpieza, el operador escribe un nuevo archivo en el formato Microsoft Excel con los nuevos datos obtenidos.

Resultados Obtenidos del Pre-procesamiento

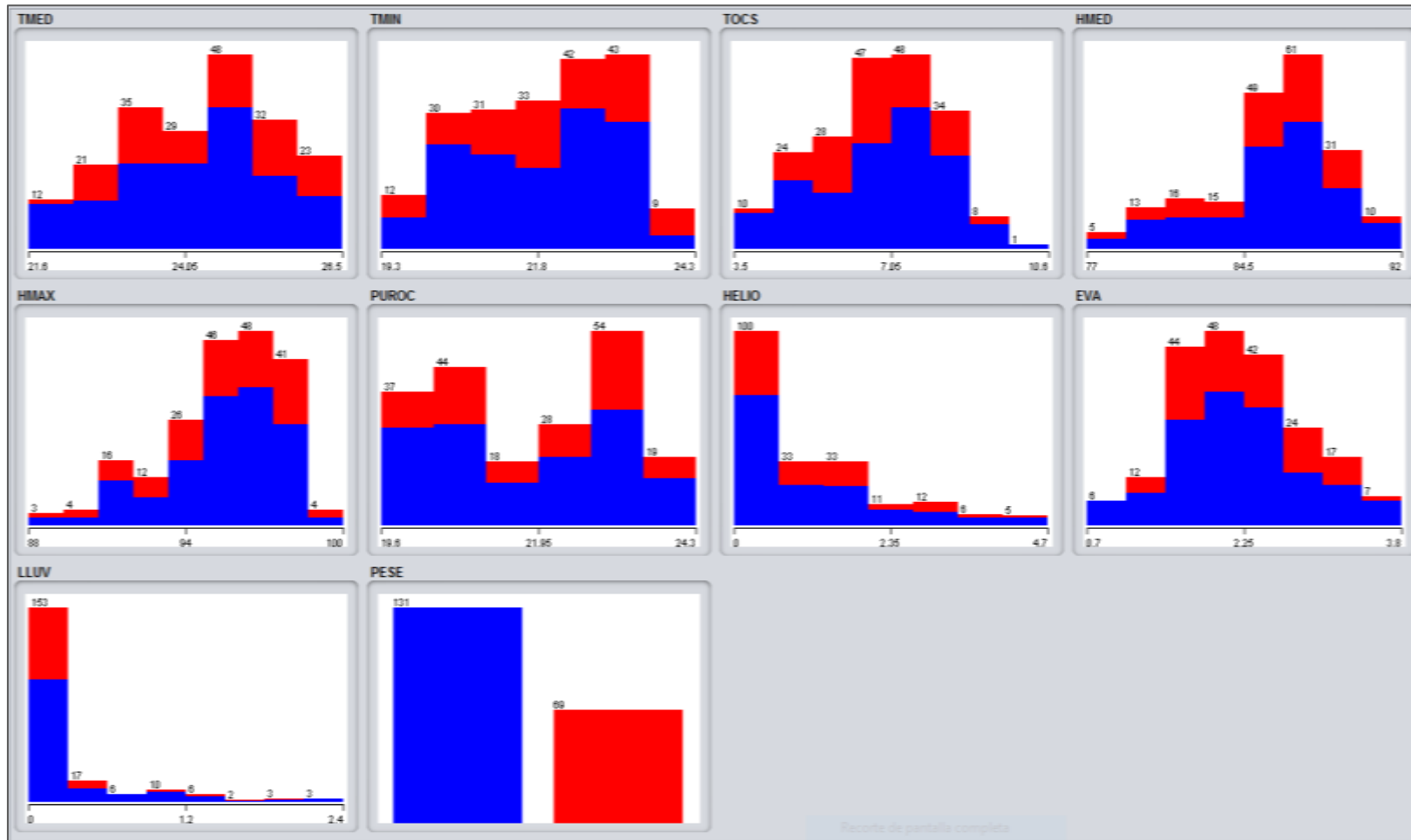
En la Ilustración 4 se muestra la nueva distribución de los datos. Se obtuvo como resultado un total 200 casos: 126 casos de Señal (Color Azul) y 74 casos de Interferencia (Color Rojo) con un total de 10 Atributos incluyendo el de tipo clase “Pérdida de Señal”. Como se puede observar la Señal supera a la Interferencia con un 63%.

Este conjunto de datos tiene una inclinación hacia los casos donde existen Señal, lo que ocasionará un sobre aprendizaje al momento de construir un modelo. Para evitar este tipo de inconvenientes se procedió a realizar un proceso de Sobre-muestreo y Sub-muestro para balancear el atributo clase y evitar el sobre aprendizaje.

En la Ilustración 5 se muestra la distribución de los datos con Sobre-muestreo. Se obtuvo como resultado un total 300 casos: 150 casos de Señal (Color Azul) y 150 casos de Interferencia (Color Rojo) con un total de 10 Atributos incluyendo el de tipo clase “Pérdida de Señal”. Como se puede observar la Señal y la Interferencia se encuentran balanceadas con un porcentaje de 50%.

El Ilustración 6 se muestra la distribución de los datos con Sub-muestreo. Se obtuvo como resultado un total 150 casos: 76 casos de Señal (Color Azul) y 74 casos de Interferencia (Color Rojo) con un total de 10 Atributos incluyendo el de tipo clase “Pérdida de Señal”. Como se puede observar la clase Señal supera a la Interferencia con un 50.66%.

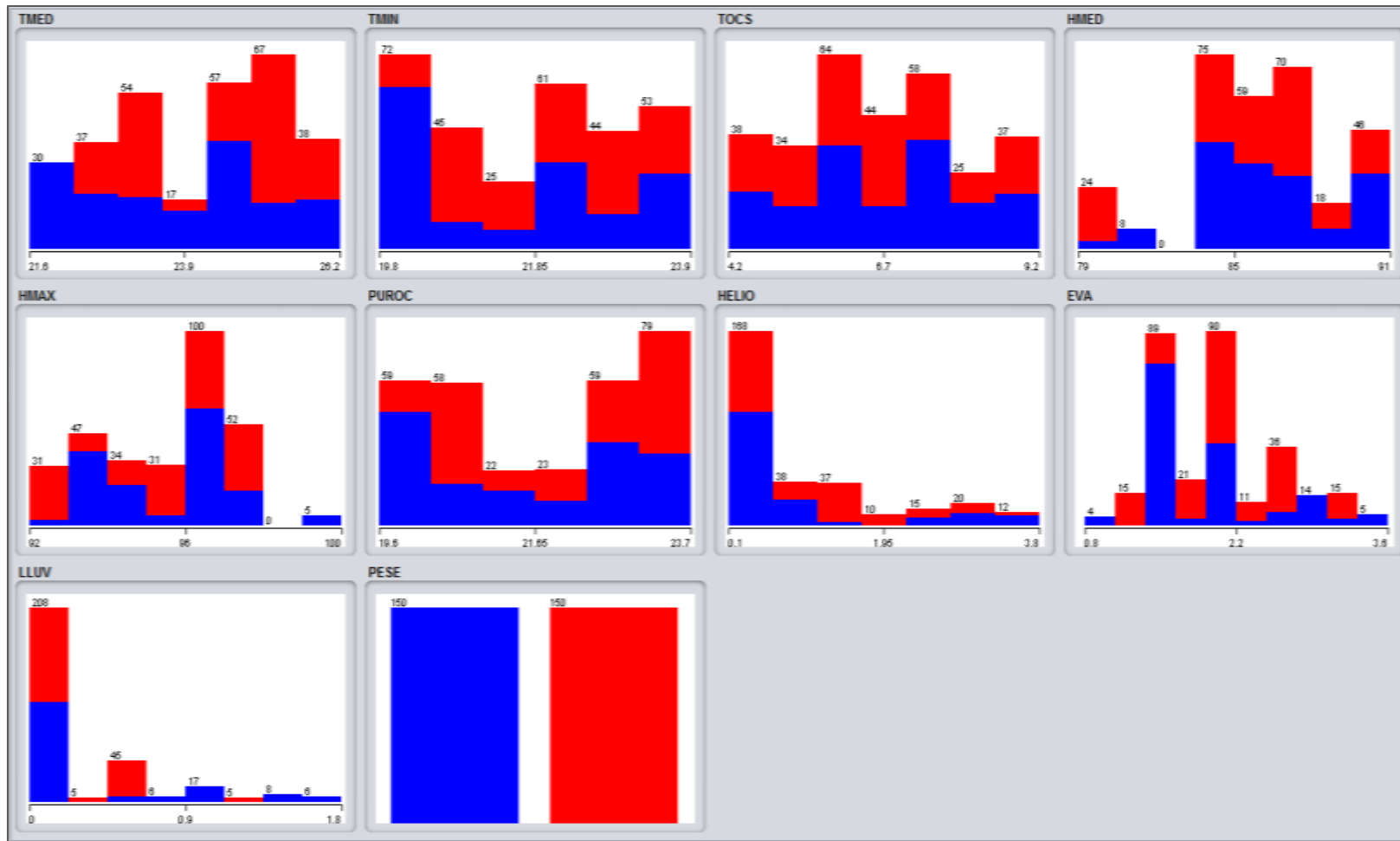
Ilustración 4: Descripción General del nuevo conjunto de datos en Weka.



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

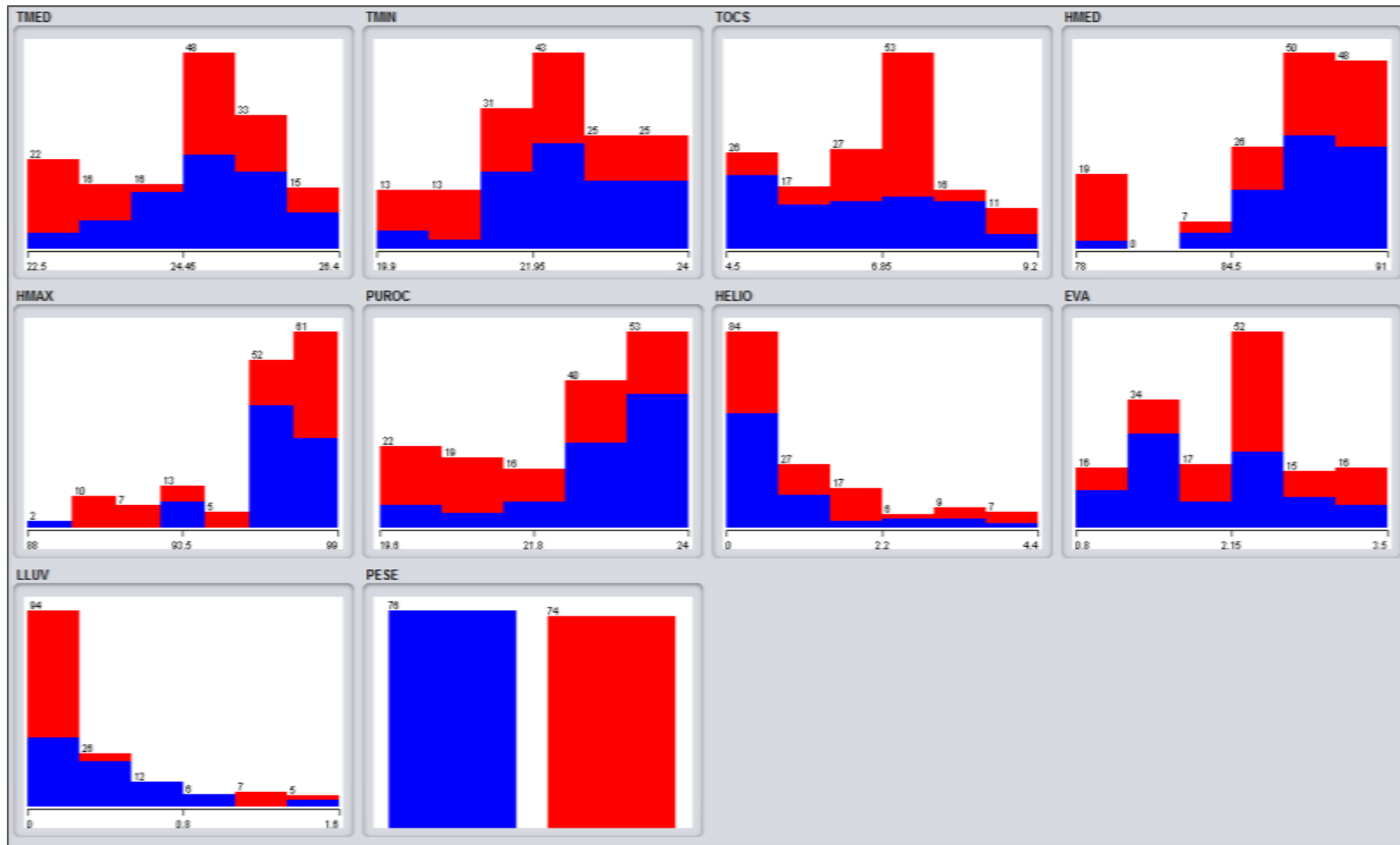
Ilustración 5: Distribución General de los datos realizando Sobre-Muestreo en Weka.



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Ilustración 6: Distribución General de los datos realizando Sub-Muestro en Weka.



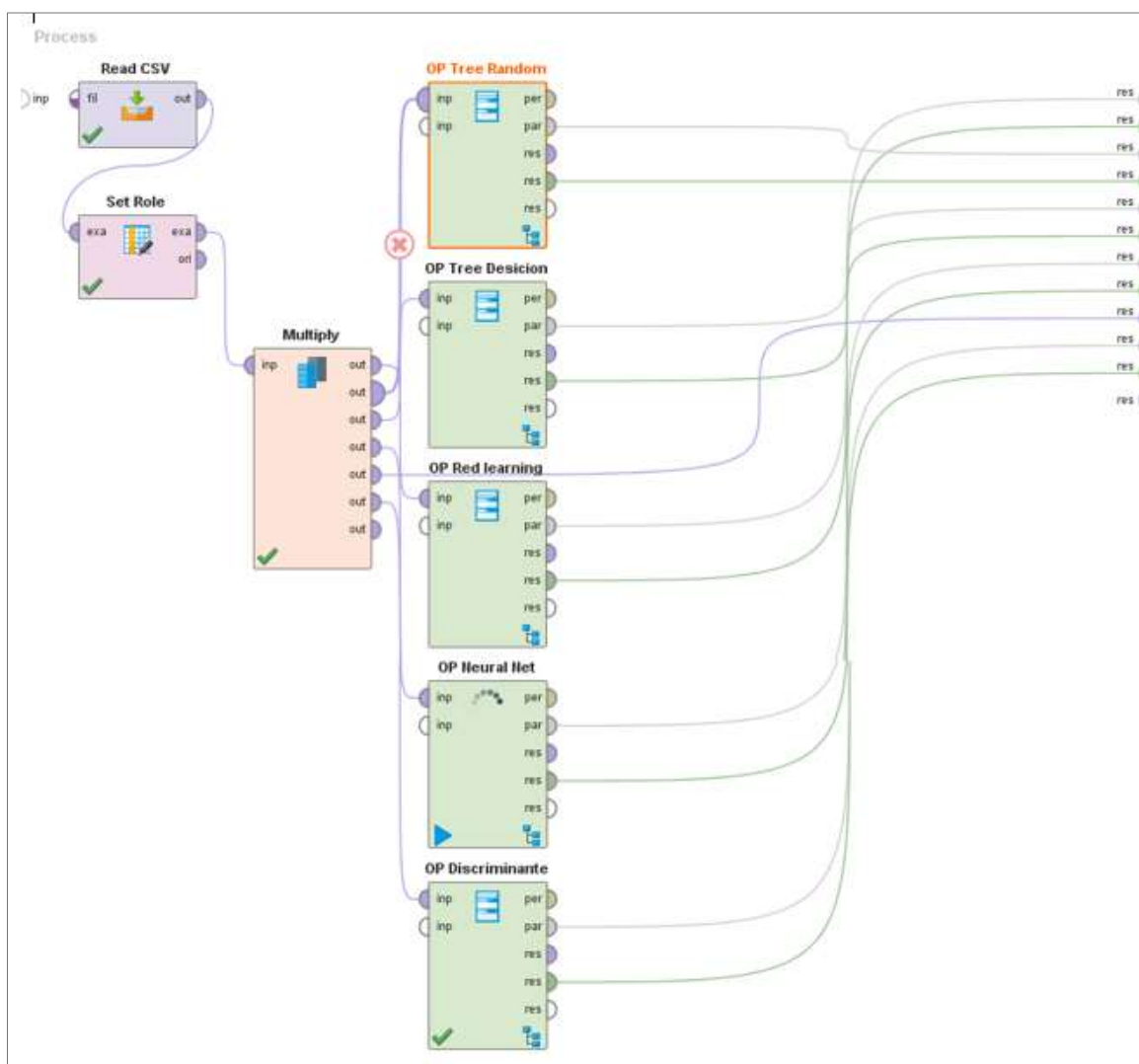
FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

4.1.3. Optimización de Algoritmos

A continuación, se detalla el proceso de optimización de los cinco tipos diferentes de algoritmos de predicción. En la Ilustración 7, se muestra los operadores a optimizar y se describe las funciones de los operadores a utilizar en el proceso de optimización. En el [Anexo 7.4](#), se detalla las características de los operadores a utilizar.

Ilustración 7: Proceso de Optimización de los algoritmos a utilizar en RapidMiner



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- *Multiply*. Se utilizó el operador *Multiply* para copiar los datos de entrada a todos los puertos de salidas conectados.
- *Optimize parameters (evolutionary)*. Permite buscar los valores óptimos para un conjunto de datos de entrada utilizando un enfoque evolutivo que reduce el número de iteraciones en base a que no se han definido en rango de variables. En la Tabla 18 se muestra los parámetros utilizados en el operador *Evolutionary*.

Tabla 18: Parámetros de Optimización-Evolutionary.

<i>Parámetros</i>	<i>Utilización</i>
Specify_popular_size	Se seleccionó para especificar el tamaño de la población.
Keep_best	Conocido también como selección elitista ²⁵ , especifica si los mejores datos se conservan.
Use_local_random_seed	Se indica para utilizar el local random
Error_handling	Permite seleccionar el método a manejar en caso de un error durante la ejecución del proceso, se seleccionó la opción <i>fail_on_error</i> para que genere un mensaje de error en el caso que se produzca una falla.
Max_generations	Se asignó el valor de <i>2000</i> para especificar el número máximo de generaciones.
Population_size	Permite establecer el tamaño de la población, en este caso el número de individuos es 5.
Mutation_type	Especifica el tipo de mutación del operador, se seleccionó el tipo <i>sparsity_mutation</i> .

²⁵ Selección elitista= Retener a los mejores individuos en una generación sin cambios en la siguiente generación

<i>Parámetros</i>	<i>Utilización</i>
selection_type	Determina el sistema de selección de los algoritmos evolutivos, se optó por utilización de la <i>técnica stochastic_universal_sampling</i>
Crossver_prob	Se asignó el valor de 0.8 para la probabilidad de que exista un cruce entre los valores a analizar
Local_random_seed	se estableció el valor 1992 para generar la aleatorización.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Resultado del Proceso de Optimización de los datos

En la Tabla 19 se muestra los resultados del proceso de optimización de los datos sin balancear. Se resaltó con ***negrita*** el mayor índice de exactitud de cada modelo analizado y se señaló con ***negrita y cursiva*** el modelo con mayor índice de exactitud a utilizar. En este caso el modelo ***Random Forest*** con un índice de exactitud del ***79.00%***.

En la Tabla 20 se muestra los resultados del proceso de optimización de los datos con sobre-muestro. Se resaltó con ***negrita*** el mayor índice de exactitud de cada modelo analizado y se señaló con ***negrita y cursiva*** el modelo con mayor índice de exactitud a utilizar. En este caso el modelo ***Neural Net*** con un índice de exactitud del ***100.00%***.

En la Tabla 21 se muestra los resultados del proceso de optimización de los datos con sub-muestreo. Se resaltó con ***negrita*** el mayor índice de exactitud de cada modelo analizado y se señaló con ***negrita y cursiva*** el modelo con mayor índice de exactitud a utilizar. En este caso el modelo ***Random Forest*** con un índice de exactitud del ***97.22%***.

Tabla 19: Resultado del Proceso de Optimización de los datos sin balancear en RapidMiner

Algoritmos		Tamaño Población	Exactitud	Error	Kappa	Precisión Clase señal	Precisión Clase Interferencia	Recall Clase Señal	Recall Clase Interferencia
Árbol de Decisiones	Random Forest	5	65.50	34.50	0.000	131	0	69	0
		10	65.50	34.50	0.000	131	0	69	0
		20	72.50	27.50	0.250	131	14	55	0
		50	73.00	27.00	0.267	131	15	54	0
		70	73.50	26.50	0.283	131	16	53	0
		100	75.00	25.00	0.332	131	19	50	0
		150	78.50	21.50	0.442	131	26	43	0
		200*	79.00*	21.00*	0.457*	131*	27*	42*	0*
	Decision Tree	5	65.50	34.50	0.000	131	0	69	0
		10	69.00	31.00	0.129	131	7	62	0
		20	69.00	31.00	0.129	131	7	62	0
		50	69.00	31.00	0.129	131	7	62	0
		70	69.00	31.00	0.129	131	7	62	0
		100	72.50	27.50	0.256	130	15	54	1
		150	69.00	31.00	0.129	131	7	62	0
		200	69.00	31.00	0.129	131	7	62	0
Red Neuronal	Neural Net	5	72.00	28.00	0.325	115	29	40	16
		10	73.50	26.50	0.368	115	32	37	16
		20	74.00	26.00	0.396	112	36	33	19
		50	74.00	26.00	0.378	116	32	37	15
		70	68.50	31.50	0.315	97	40	29	34
		100	75.50	24.50	0.389	123	28	41	8
		150	77.50	22.50	0.464	119	36	33	12

Algoritmos		Tamaño Población	Exactitud	Error	Kappa	Precisión Clase señal	Precisión Clase Interferencia	Recall Clase Señal	Recall Clase Interferencia
	Deep Learning	200	74.50	25.50	0.383	118	31	38	13
		5	65.50	34.50	0.000	131	0	69	0
		10	77.50	22.50	0.490	112	43	26	19
		20	75.00	25.00	0.447	106	44	25	25
		50	78.00	22.00	0.485	117	39	30	14
		70	65.50	34.50	0.000	131	0	69	0
		100	76.00	24.00	0.450	112	40	29	19
		150	77.50	22.50	0.460	120	35	34	11
		200	76.00	24.00	0.442	114	38	31	17
Análisis Discriminante	RDA	5	34.50	65.50	0.000	0	69	0	131
		10	34.50	65.50	0.000	0	69	0	131
		20	34.50	65.50	0.000	0	69	0	131
		50	34.50	65.50	0.000	0	69	0	131
		70	34.50	65.50	0.000	0	69	0	131
		100	34.50	65.50	0.000	0	69	0	131
		150	34.50	65.50	0.000	0	69	0	131
		200	34.50	65.50	0.000	0	69	0	131

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Tabla 20: Resultado del Proceso de Optimización de los datos con sobre-muestreo en RapidMiner

Algoritmos		Tamaño Población	Exactitud	Error	Kappa	Precisión Clase señal	Precisión Clase Interferencia	Recall Clase Señal	Recall Clase Interferencia
Árbol de Decisiones	Random Forest	5	50.00	50.00	0.000	0	150	0	150
		20	80.67	19.33	0.613	97	145	5	97
		50	91.63	8.37	0.902	130	121	24	35
		100	94.33	5.67	0.927	144	135	13	5
		150	95.14	4.86	0.907	147	139	11	3
		200	97.22	2.78	0.944	148	141	8	0
		250	98.05	1.95	0.962	148	145	7	3
		300	96.33	3.67	0.927	148	141	9	2
	Decision Tree	5	88.67	11.33	0.773	124	142	8	26
		20	93.00	7.00	0.860	140	139	11	10
		50	95.33	4.67	0.907	147	139	11	3
		100	99.67	0.33	0.993	150	149	1	0
		150	93.00	7.00	0.860	140	139	11	10
		200	93.00	7.00	0.860	140	139	11	10
		250	93.00	7.00	0.860	140	139	11	10
		300	93.00	7.00	0.860	140	139	11	10
Red Neuronal	Neural Net	5*	100.00*	0.00*	1.000*	150*	150*	0*	0*
		20	100.00	0.00	1.000	150	150	0	0
		50	100.00	0.00	1.000	150	150	0	0
		100	100.00	0.00	1.000	150	150	0	0
		150	100.00	0.00	1.000	150	150	0	0
		200	100.00	0.00	1.000	150	150	0	0
		250	100.00	0.00	1.000	150	150	0	0
		300	100.00	0.00	1.000	150	150	0	0

Algoritmos		Tamaño Población	Exactitud	Error	Kappa	Precisión Clase señal	Precisión Clase Interferencia	Recall Clase Señal	Recall Clase Interferencia
	Deep Learning	5	80.67	19.33	0.613	97	145	5	97
		20	94.00	6.00	0.880	141	141	9	9
		50	96.00	4.00	0.920	148	140	10	2
		100	96.33	3.67	0.927	148	141	9	2
		150	96.33	3.67	0.927	148	141	9	2
		200	96.33	3.67	0.927	148	141	9	2
		250	96.00	4.00	0.920	148	140	10	2
		300	96.33	3.67	0.927	148	141	9	2
Análisis Discriminante	RDA	5	50.00	50.00	0.000	0	150	0	150
		20	50.00	50.00	0.000	0	150	0	150
		50	50.00	50.00	0.000	0	150	0	150
		100	50.00	50.00	0.000	0	150	0	150
		150	50.00	50.00	0.000	0	150	0	150
		200	50.00	50.00	0.000	0	150	0	150
		250	50.00	50.00	0.000	0	150	0	150
		300	50.00	50.00	0.000	0	150	0	150

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Tabla 21: Resultado del Proceso de Optimización de los datos con sub-muestreo en RapidMiner

Algoritmos		Tamaño Población	Exactitud	Error	Kappa	Precisión Clase señal	Precisión Clase Interferencia	Recall Clase Señal	Recall Clase Interferencia
Árbol de Decisiones	Random Forest	5	53.47	46.53	0.000	77	0	67	0
		10	53.47	46.53	0.000	77	0	67	0
		20	53.47	46.53	0.000	77	0	67	0
		50	84.72	15.28	0.691	69	53	14	8
		70	97.22*	2.78*	0.944*	77*	63*	4*	0*
		100	96.53	3.47	0.930	77	62	5	0
		130	96.53	3.47	0.930	77	62	5	0
		150	95.14	4.86	0.902	75	62	5	2
	Decision Tree	5	53.47	46.53	0.000	77	0	67	0
		10	63.89	36.11	0.236	77	15	52	0
		20	61.81	38.19	0.189	77	12	55	0
		50	68.75	31.25	0.364	61	38	29	16
		70	68.75	31.25	0.364	61	38	29	16
		100	68.75	31.25	0.364	61	38	29	16
		130	68.75	31.25	0.364	61	38	29	16
		150	68.75	31.25	0.364	61	38	29	16
Red Neuronal	Neural Net	5	53.47	46.53	0.000	77	0	67	0
		10	55.56	44.44	0.073	64	16	51	13
		20	61.81	38.19	0.196	73	16	51	4
		50	63.19	36.81	0.231	70	21	46	7
		70	66.67	33.33	0.331	52	44	23	25
		100	64.58	35.42	0.266	67	26	41	10
		130	65.97	34.03	0.288	73	22	45	4

Algoritmos		Tamaño Población	Exactitud	Error	Kappa	Precisión Clase señal	Precisión Clase Interferencia	Recall Clase Señal	Recall Clase Interferencia
	Deep Learning	150	65.97	34.03	0.288	73	22	45	4
		5	65.97	34.03	0.329	43	52	15	34
		10	54.86	45.14	0.087	48	31	36	29
		20	68.06	31.94	0.376	39	59	8	38
		50	65.28	34.72	0.314	43	51	16	34
		70	61.81	38.19	0.255	34	55	12	43
		100	67.36	32.64	0.360	41	53	11	36
		130	68.75	31.25	0.377	50	49	18	27
		150	63.89	36.11	0.287	42	50	17	35
Análisis Discriminante	RDA	5	53.47	46.53	0.000	77	0	67	0
		10	53.47	46.53	0.000	77	0	67	0
		20	53.47	46.53	0.000	77	0	67	0
		50	53.47	46.53	0.000	77	0	67	0
		70	53.47	46.53	0.000	77	0	67	0
		100	53.47	46.53	0.000	77	0	67	0
		130	53.47	46.53	0.000	77	0	67	0
		150	53.47	46.53	0.000	77	0	67	0

FUENTE: CONOCIMIENTOS PROPIOS

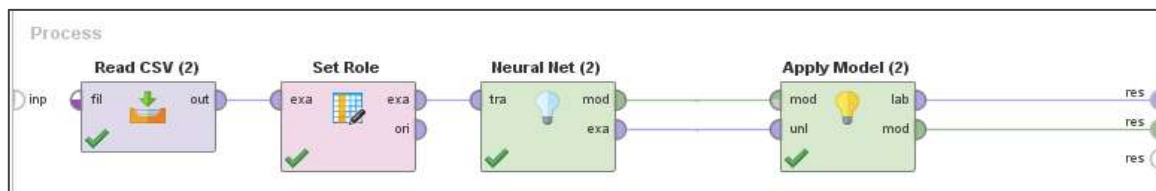
AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Como se observan en las tablas del proceso de optimización, el resultado con mayor índice de exactitud es el de la Tabla de Optimización de datos con sobre-muestro, donde el algoritmo **Neural Net** obtuvo un índice de exactitud del **100%** con un tamaño de población de **5**. Este resultado se repite para diferentes tamaños de poblaciones superiores a 5, en el resto de los algoritmos varia el resultado dependiendo del tamaño de la población.

4.1.4. Generación del Modelo Optimizado

En la Ilustración 8 se muestra el proceso para generar el modelo optimizado, en la Ilustración 9 se observa los parámetros necesarios para generar dicho modelo, en la Tabla 22 se detalla el concepto de los parámetros necesarios y en la Ilustración 10 se representa gráficamente el modelo obtenido.

Ilustración 8: Generación del modelo optimizado en RapidMiner



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

En la Ilustración 9 se exhibe los parámetros necesarios para obtener el índice de exactitud del 100% para el modelo Neural Net.

Ilustración 9: Parámetros Optimizados del Modelo Neural Net en RapidMiner

```

ParameterSet

Parameter set:

Performance:
PerformanceVector [
****accuracy: 100.00%
ConfusionMatrix:
True:  true  false
true:  150    0
false:  0    150
----classification_error: 0.00%
ConfusionMatrix:
True:  true  false
true:  150    0
false:  0    150
----kappa: 1.000
ConfusionMatrix:
True:  true  false
true:  150    0
false:  0    150
]
Neural Net.training_cycles      = 63
Neural Net.learning_rate       = 0.7628774515705214
Neural Net.momentum            = 0.681461325038434
Neural Net.error_epsilon       = 0.0
Neural Net.local_random_seed   = 74

```

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Tabla 22: Parámetros Optimizado del Modelo Neural Net

<i>Parámetros</i>	<i>Utilización</i>
Neural Net.training_cycles	Específica el ciclo de entrenamiento utilizados para la generación de redes neuronales, este ciclo se repite cada vez que se presente un error hasta que la comparación entre los valores de salida y la respuesta correcta sean las misma. Este proceso se repite N veces, en este caso $N=63$.
Neural Net.learning_rate	Determina el índice de variación de los pesos de las variables que son utilizadas en <i>training_cycles</i> , en este caso su valor es de <i>0.7628774515705214</i> .

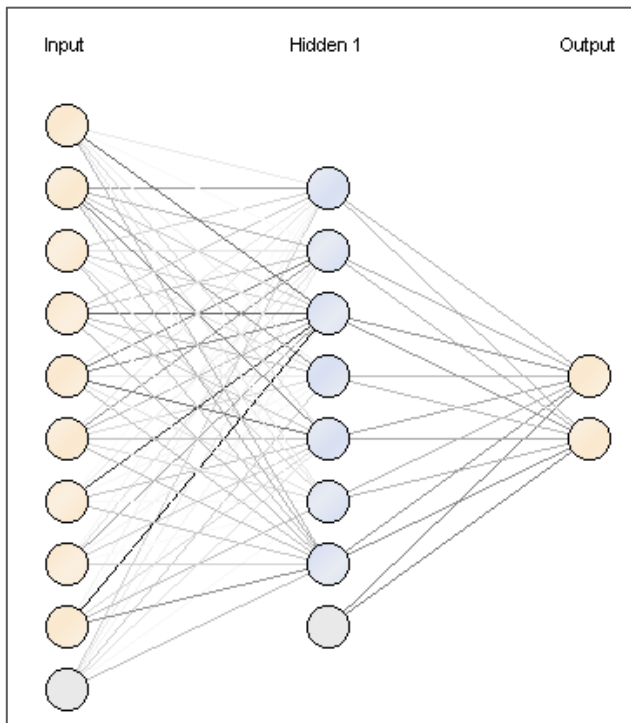
<i>Parámetros</i>	<i>Utilización</i>
Neural Net.momentum	Es utilizado para añadir una fracción del peso normal dada por el peso anterior al valor actual, se usa para evitar los máximos locales, en este caso el momentum es <i>0.681461325038434</i>
Neural Net.error_epsilon	Representa el punto de parada del algoritmo de optimización, en este caso es <i>0.0</i> .
Neural Net.local_random_seed	Específica la semilla aleatoria local, en este caso es <i>74</i> .

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Como se aprecia en la Ilustración 10 se obtuvo como resultado una Red Neural de tres capas: entrada, oculta y neuronas de salida.

Ilustración 10: Representación Gráfica del modelo Neural Net en RapidMiner



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

4.1.5. Discusión de la Minería.

El proceso de minería de datos se realizó en un período corto de tiempo. La herramienta en la que se realizó el estudio posee una interfaz amigable para estudiantes e investigadores que se encuentran comenzando a indagar dentro del área de conocimientos. Además, cuenta con una extensión que permite exportar los modelos optimizados para ser implementados de forma práctica.

Esta investigación hasta el momento es única. Como se puede observar en el Capítulo 2 en el Marco Referencial, existen muchas investigaciones que utilizan la minería de datos pero, son pocas las están destinadas a predecir la pérdida de señal, a excepción de la tesis [1] la cual realiza un estudio pero no implementa su modelo de forma práctica.

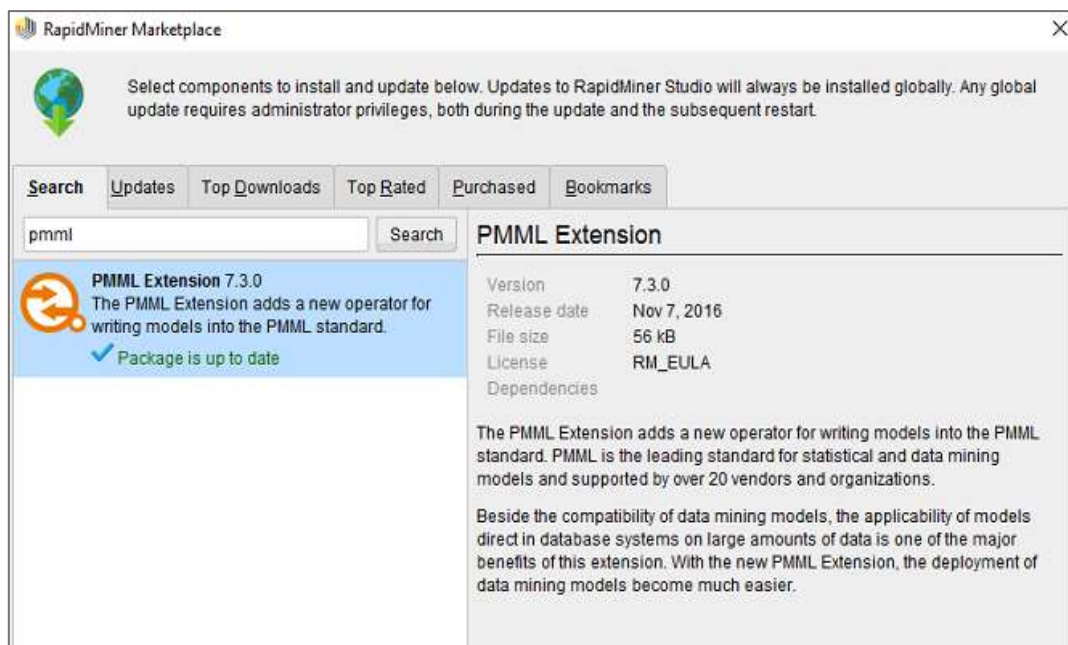
En relación con la investigación previamente mencionada, se observa que el algoritmo FURIA-C basado en Lógica Difusa obtuvo una precisión de 87.74% con una cantidad de 8 reglas. En cambio, en la presente investigación el modelo Neural Net basado en Redes Neuronales obtuvo una exactitud del 100% tomando en cuenta 9 variables meteorológicas y aplicado a un tamaño de publicación de 5. Sería ideal realizar nuevamente el proceso de minería de datos con un nuevo conjunto de datos que posea una mayor cantidad de información y está se encuentre actualizada.

4.2. Aplicación Móvil

4.2.1. Exportar el modelo de minería de datos.

Para poder manipular el modelo generado por la herramienta de minería de datos, se necesitó exportar el modelo a un objeto compatible con algún tipo de lenguaje de programación. En la Ilustración 11 se observa la extensión que ofrece RapidMiner para exportar un modelo de minería de datos a un archivo PMML.

Ilustración 11: Extensión PMML en RapidMiner

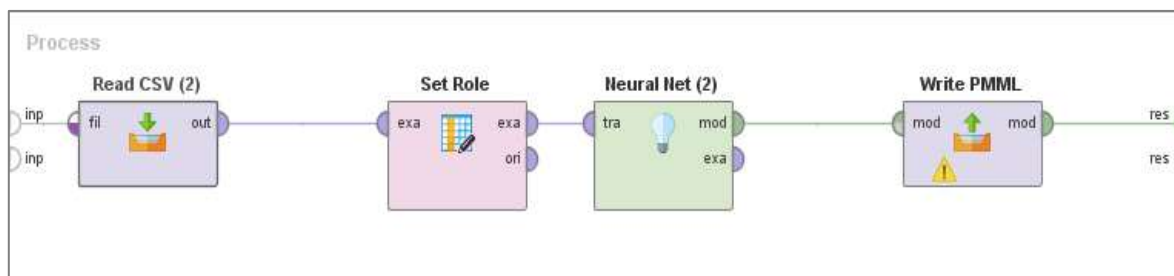


FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

En la Ilustración 12, se usa el operador Write PMML que exporta un modelo de minería de datos a un archivo con la extensión PMML. Este lenguaje está basado en XML y mediante el uso de ciertas librerías se puede transformar a un objeto de Java.

Ilustración 12: Uso del operador Write PMML en RapidMiner



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

4.2.2. Desarrollo de la Aplicación Móvil

Antes de llevar acabo el desarrollo de la aplicación móvil, se puso a prueba los diferentes métodos que brinda la documentación de PMML en las siguientes plataformas: Móvil, Web y Escritorio. Se encontraron varios inconvenientes al momento de implementar PMML en las plataformas Móvil y Web.

Debido a los contratiempos que se detallan dentro del uso de las librerías; se procedio a elaborar una aplicación de escritorio la cual se encarga de construir y evaluar el modelo de minería de datos. Para la comunicación entre las dos plataformas: Móvil y Escritorio; se usó un servidor de remoto gratuito cuyo propósito es almacenar cada transacción realizada entre las plataformas, brindando así una comunicación estable.

Para el desarrollo de la aplicación móvil *SignalPred* se empleó la metodología Extreme-Programming²⁶ la cual posee un conjuntos de fases: Planificación, Diseño, Codificación y Pruebas, como se puede observar en el Gráfico 2.

Fase de Planificación

Para el desarrollo de la aplicación móvil *SignalPred* se utilizó el entorno de desarrollo integrado *Android Studio* ya que empresa Google brinda constates actualización y soporte en caso de errores. Como se observa en el Anexo 7.6, la versión más utilizada en dispositivos móviles es la *4.0 Ice Cream Sandwich*, por esta razón se decidió trabajar a nivel de *API 15* para cubrir la mayor cantidad de dispositivos.

- **Requerimientos**

La aplicación móvil *SignalPred* cumple con los siguientes requerimientos necesarios para su correcto funcionamiento.

²⁶ XP: Extreme-Programming

Actividades

- *Inicio.* Se ejecuta solo la primera vez luego de la instalación y demuestra gráficamente su funcionamiento.
- *Ingreso de datos.* El usuario selecciona la forma de ingresar los valores para que el modelo de minería de datos los procese, existen dos formas de ingresar los valores: Manual y Automática.
- *Historial de resultados.* Esta actividad se encuentra a disposición del usuario y solo almacena los valores ya procesados por el modelo de minería de datos.
- *Visualización de resultados.* Muestra gráficamente los resultados obtenidos según el modelo de minería de datos que es procesado por una aplicación de escritorio.

Procesos

- *Rendimiento de la aplicación*
 - El modelo de minería de datos no es cargado en la aplicación móvil.
 - El tiempo máximo de espera por los resultados no debe sobrepasar los 30 segundos.
 - El tiempo promedio de entrega de los datos es de 12 segundos.
 - El modelo de minería de datos es cargado en la aplicación de escritorio la cual administra todas las peticiones de la aplicación móvil.
- *Seguridad.*
 - La primera vez que se ejecute el dispositivo le solicitará los permisos de acceso al *GPS* y acceso a manejar archivos dentro del teléfono.
 - Cada dispositivo móvil es reconocido por un identificador único(ID) proporcionado por el mismo dispositivo.
- *Disponibilidad*
 - La aplicación funcionara solo con Acceso a Internet.
 - La aplicación recibirá resultados solo si el servicio de respuesta se encuentra en línea.
- *Portabilidad.*

- La aplicación es compatible con otros dispositivos *Android* desde la *versión 4.0 o Superior*, ya sea en Tablet o Smartphone.

- ***Limitaciones***

El dispositivo móvil necesita una serie de requerimientos mínimos para funcionar correctamente, para enviar los paquetes de información al servicio de respuesta y estos puedan ser procesado se necesita una conexión a Internet ya sea: *Wifi o Plan de Datos móviles*.

En la Tabla 23 se detalla los requerimientos mínimos:

Tabla 23: Requerimientos mínimo para el funcionamiento de la aplicación

<i>Componentes</i>	<i>Descripción</i>
Procesador	1 GHz DualCore
Memoria RAM	512 MB
Pantalla	4 pulgadas
Almacenamiento	15 MB disponibles
Sistema Operativo	Android 4.0 o superior

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

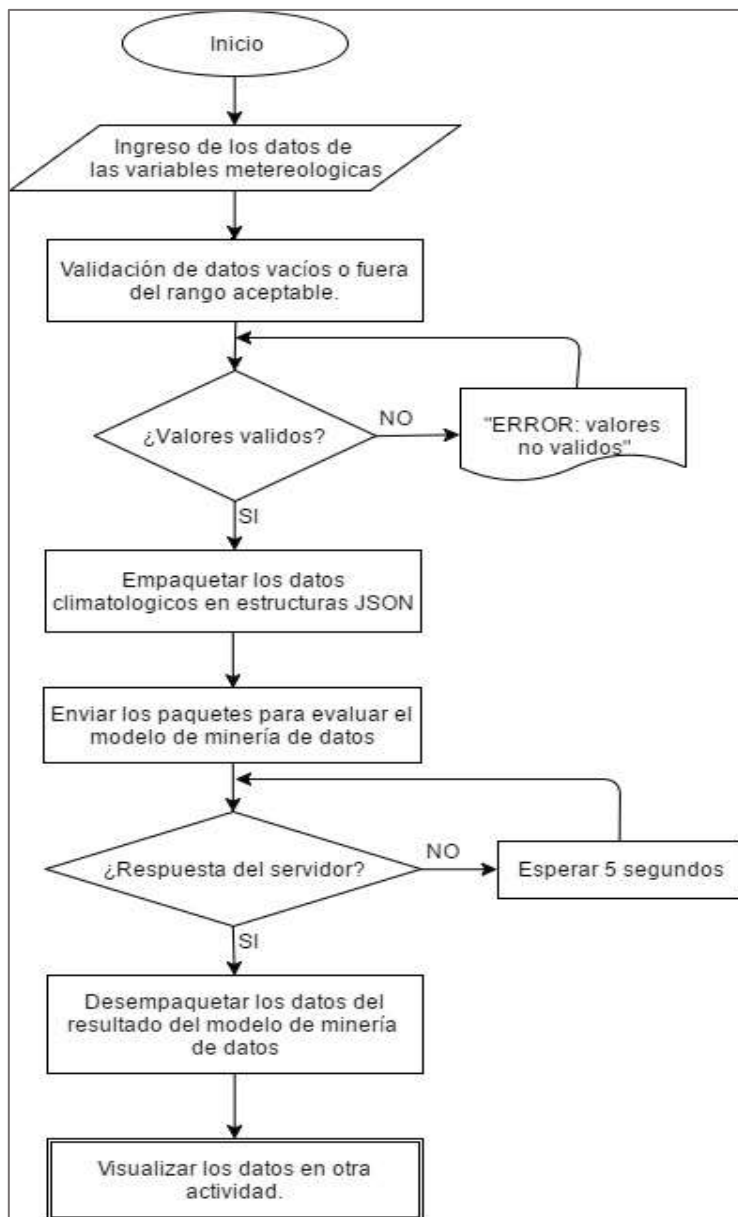
Fase de Diseño

Dentro de la fase de diseño empleó los siguientes pasos para el desarrollo de la aplicación móvil SignalPred: *Diagrama de Flujo, Arquitectura y Diagramas de caso de uso*.

Diagramas de flujo. Para un correcto funcionamiento de la aplicación móvil SignalPred fue necesario diseñar los siguientes diagramas de flujos. A continuación, se presentan los diagramas de flujos más significativos para el desarrollo de la aplicación móvil *SignalPred*.

En la Ilustración 13 se observa un diagrama de flujo, donde el usuario deberá ingresar los valores de las variables meteorológicas de forma manual, una vez ingresado los datos correctamente se procederá a enviarlos al servicio de respuesta y este responderá con el resultado de la evaluación del modelo.

Ilustración 13: Diagrama de Flujo - Ingreso Manual de valores

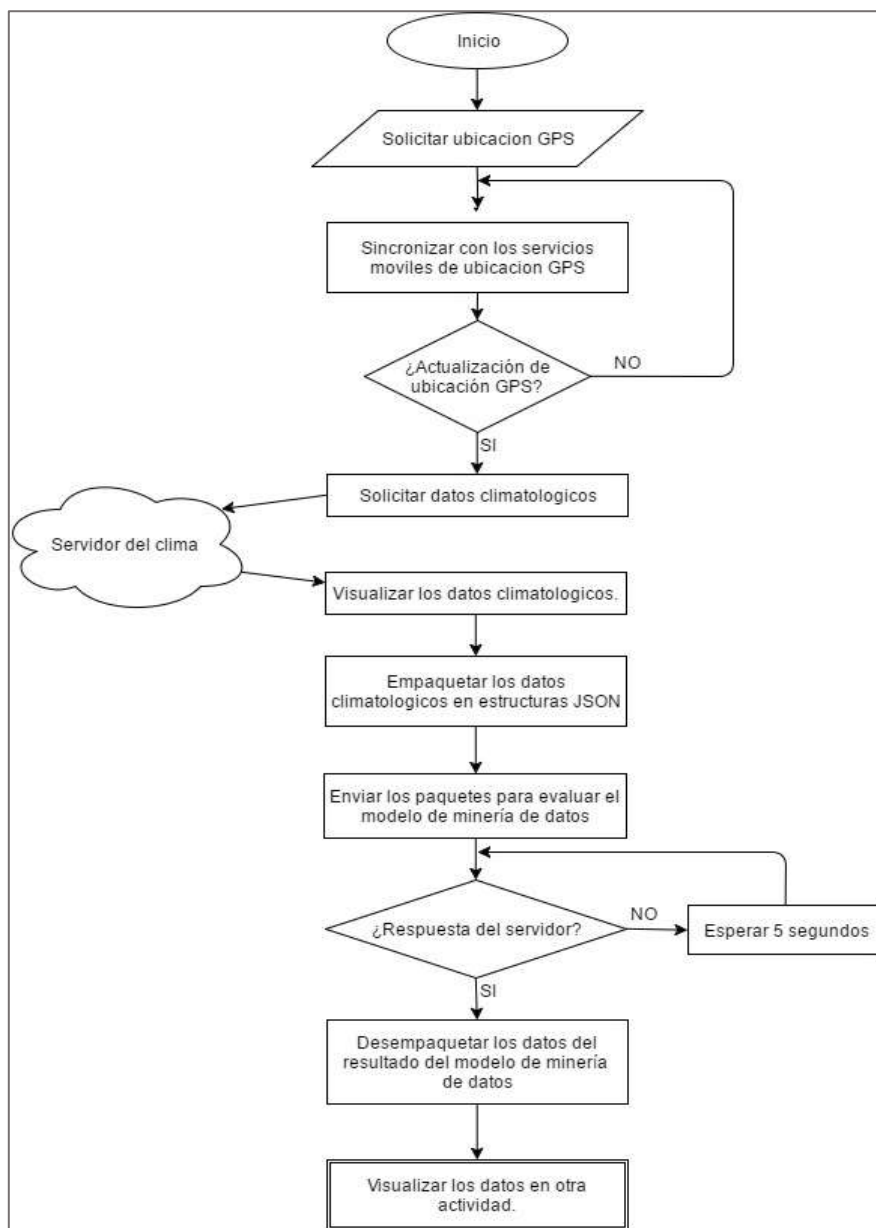


FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

En la Ilustración 14 se observa un diagrama de flujo, donde el usuario deberá proporcionar su ubicación GPS, una vez sincronizado con los servicios se visualizarán los datos meteorológicos obtenidos por un servidor climatológico externo; lo que proporcionará los valores de las variables meteorológicas y se procederá a enviarlos al servicio de respuesta y este responderá con el resultado de la evaluación del modelo.

Ilustración 14: Diagrama de Flujo - Ingreso Automático de valores

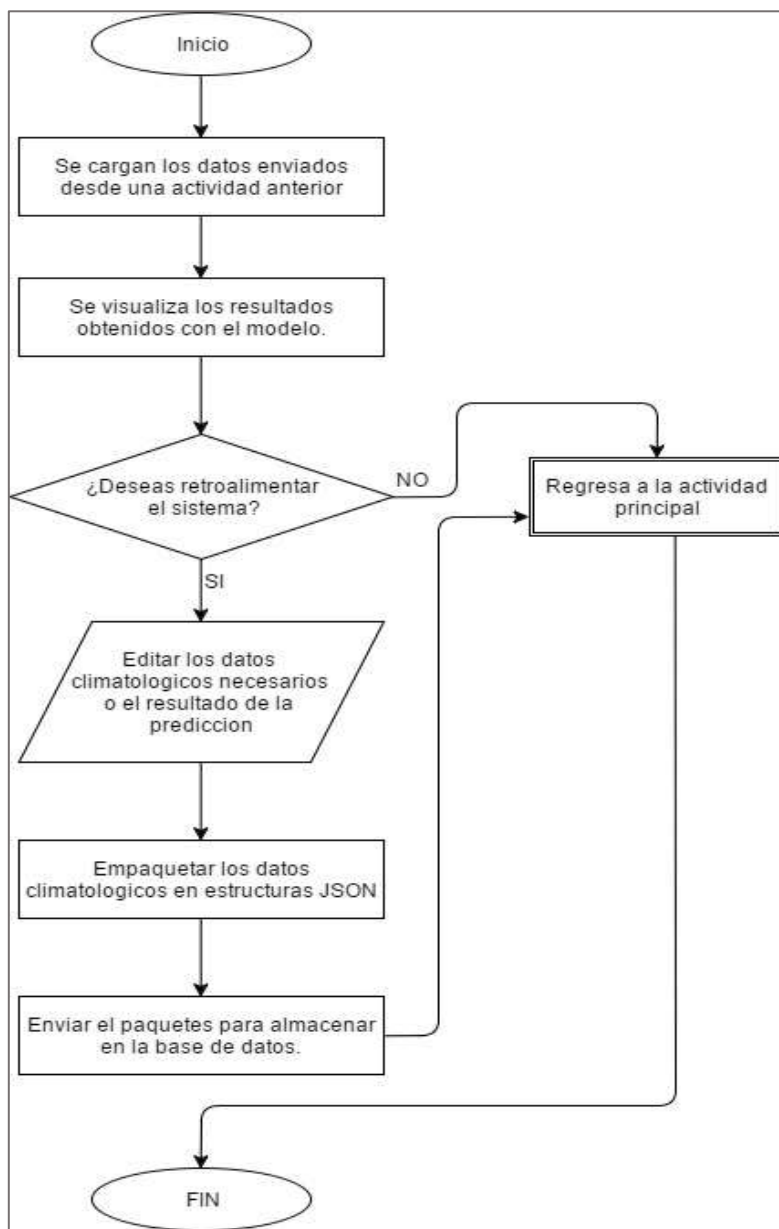


FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

En la Ilustración 15 se observa un diagrama de flujo donde consulta al usuario si desea retroalimentar el sistema con los datos obtenidos ya sea de forma manual o automática. En el caso de seleccionar retroalimentar, entonces se cargará nuevamente los datos generados anteriormente y se podrán modificar los datos climatológicos necesarios, una vez realizadas las modificaciones se procederá a retroalimentar el sistema.

Ilustración 15: Digrama de Flujo - Retroalimentación



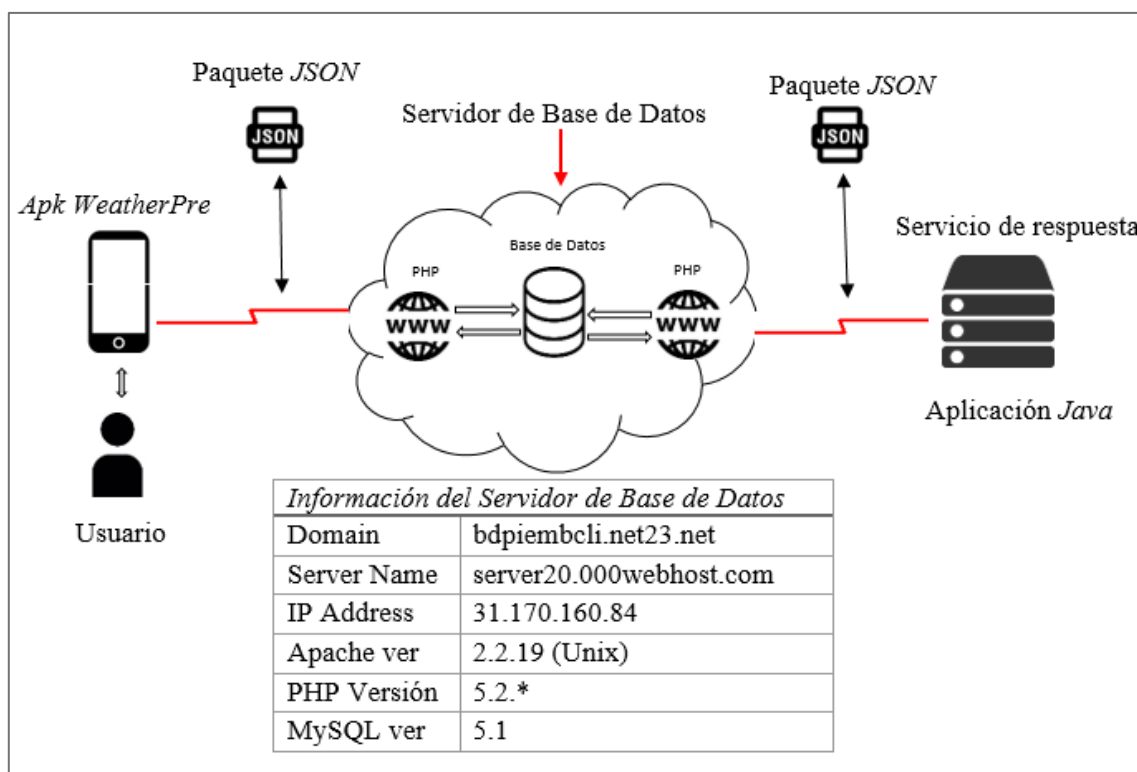
FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Arquitectura. La aplicación móvil *SignalPred* está basada en una arquitectura de *Tres Niveles: Presentación, Negocio y Datos.*

Se muestra en la Ilustración 16, la comunicación entre la aplicación móvil *SignalPred* y una *Aplicación de Escritorio* que desempeñara el rol de la lógica de negocio, adicionalmente se incorpora un *Servidor Remoto de Base de datos* que guardara la información generada.

Ilustración 16: Arquitectura Cliente / Servidor



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- *Nivel de presentación.* La aplicación móvil *SignalPred* se encarga de recolectar la información mediante el ingreso manual o automático; haciendo uso del *GPS* para adquirir información de tu posición actual, la misma que será estructurada en paquetes *JSON* para el envío por Internet.

- *Nivel de negocio.* La aplicación de escritorio se encarga de recibir la información de la base de datos y procesarla adecuadamente según las especificaciones del modelo de minería de datos, generando un resultado que es almacenado en la base de datos.
- *Nivel de datos.* El servidor de base de datos remoto es el encargado de aglomerar toda la información de las transacciones que se tenga entre la aplicación de escritorio y la aplicación móvil *SignalPred*, además de almacenar en una tabla independiente los datos que permitan retroalimentar el modelo.

Diagramas de caso de uso. Para el correcto desarrollo de la aplicación fue necesario la elaboración de casos de uso de las actividades principales a ejecutar, en el [Anexo 7.5](#), se muestran los casos de usos.

Fase de Codificación

En la presente fase, se muestra la etapa de desarrollo y codificación necesaria tanto de la parte de la aplicación móvil *SignalPred* como del servicio de respuesta. Entre los puntos a tratar se presentan los siguientes: *Librerías PMML*, *Inconvenientes encontrados con el uso de las librerías*, *Descripción del código de la capa negocio* y *Descripción del código de la capa presentación*.

Librerías PMML. Ofrece a sus usuarios una serie de librerías para manipular el objeto exportado a un objeto Java. En la Tabla 24 se muestran los repositorios de las librerías utilizadas en el desarrollo de la aplicación:

Tabla 24: librerías con el URL de descarga

<i>Librería</i>	<i>URL²⁷ de descarga.</i>
Evaluator	https://github.com/jpmml/jpmml-evaluator
Manager	http://repo1.maven.org/maven2/org/jpmml/pmml-manager/

²⁷ URL: Uniform Resource Locator / Localizador Uniforme De Recurso

<i>Librería</i>	<i>URL²⁷ de descarga.</i>
Model	https://github.com/jpmml/jpmml-model
Schema	http://repo1.maven.org/maven2/org/jpmml/pmml-schema/

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Inconvenientes encontrados con el uso de la librería. Al incorporar las librerías al SDK de Android se presentaron unas series de errores tales como:

- El SDK de Android no incorpora completamente la librería del *Javax.xml.bind*. lo que provoca un error al momento de leer el archivo del modelo.
- Durante la conversión del archivo PMML el sistema arroja un error de archivo corrupto; este caso se repitió con diferentes archivos PMML exportados de diferentes herramientas de minería de datos.
- Al intentar incorporar manualmente las librerías necesarias de Java, se logró convertir el archivo a un objeto XML, pero desafortunadamente la librería PMML solo acepta el modelo si es convertido bajo sus parámetros, caso contrario genera error de incompatibilidad.

Se intentó usar las librerías PMML en *Java Server Page (JSP)*, mediante el uso de la clase servlet para incorporar las funciones y métodos necesarios, pero se presentaron los siguientes errores:

- No se pudo convertir el modelo debido a que el *SDK 1.7* en lo que trabaja los servidores Web no reconoció la librería *pmml-manager* y *pmml-model*, las que incorporan los métodos necesarios para convertir el modelo.
- Durante la conversión de valores primitivos de Java a valores necesarios para evaluar el modelo no reconoció la librería *guava19.0*. usada para instanciar los atributos *FieldValue*.

Por último, se elaboró una aplicación de escritorio que tiene la función de convertirse en un servicio de respuesta para las solicitudes por parte de una aplicación móvil, la cual evalúa el modelo de minería de datos y genera un resultado que será devuelta a la aplicación móvil.

Esta aplicación de escritorio utiliza las siguientes librerías de PMML:

- pmml-evaluator-1.3.4.
- pmml-manager-1.1.20.
- pmml-model-1.3.5.
- pmml-schema-1.3.2.

A continuación, se detalla los puntos más importantes al momento de usar las *librerías* PMML para generar el objeto JAVA.

- Conversión de objeto. En la Ilustración 17, se define un método que necesita como parámetro la URL del archivo PMML y retorna un objeto utilizable por Java.

Ilustración 17: Método para cargar un archivo PMML

```
public static PMML loadPMML(String file){  
  
    PMML pmml = null;  
    File inputFilePath = new File( file );  
    try{  
        InputStream in = new FileInputStream( inputFilePath );  
        Source source = ImportFilter.apply(new InputSource(in));  
        pmml = JAXBUtil.unmarshalPMML(source);  
    } catch( Exception e) {  
        System.out.print(e.toString());  
    }  
    return pmml;  
}
```

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- *Obtención de campos del modelo.* Como se muestra en la Ilustración 18, se define tres variables, la primera corresponde a los campos de entrada del modelo, la segunda corresponde a los campos de destino del modelo y la última corresponde a los campos de salida del modelo.

Ilustración 18: Definición de Campo de Variables

```
private void generarCampos() {  
    List<InputField> inputFields = evaluator.getInputFields();  
    List<TargetField> targetFields = evaluator.getTargetFields();  
    List<OutputField> outputFields = evaluator.getOutputFields();  
}
```

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- *Generar un evaluador.* En la Ilustración 19, se instancia el evaluador que representa el modelo de minería de datos esperado el ingreso de valores.

Ilustración 19: Evaluador del modelo de minería de datos

```
Evaluator evaluator = modelEvaluator;
```

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- *Evaluar el modelo.* En la Ilustración 20, se define los pasos para ingresar valores a campos de entrada y almacenarlos en una lista para luego ser ingresado dentro del evaluador y generar el resultado estimado por el modelo.

Ilustración 20: Pasos para ingresar valores al modelo

```
Map<FieldName, FieldValue> arguments = new LinkedHashMap<>();  
int contador = 0;  
inputFields = evaluator.getInputFields();  
for(InputField inputField : inputFields){  
    FieldName inputFieldName = inputField.getName();  
  
    // The raw (ie. user-supplied) value could be any Java primitive value  
    Object userValue=null;  
  
    FieldValue inputFieldValue = inputField.prepare(userValue);  
    arguments.put(inputFieldName, inputFieldValue);  
}  
Map<FieldName, ?> results = evaluator.evaluate(arguments);
```

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- *Obtener resultado.* Como se muestra en la Ilustración 21, se define los pasos para extraer el resultado obtenido del modelo y la probabilidad de que ocurra ese resultado.

Ilustración 21: Obtención del resultado del modelo y probabilidad del resultado

```
targetFields = evaluator.getTargetFields();
for(TargetField targetField : targetFields){
    FieldName targetFieldName = targetField.getName();

    Object targetFieldValue = results.get(targetFieldName);
    if(targetFieldValue instanceof Computable){

        Computable computable = (Computable)targetFieldValue;
        if(targetFieldValue instanceof HasEntityId){

            HasEntityId hasEntityId = (HasEntityId)targetFieldValue;
            HasEntityRegistry<?> hasEntityRegistry = (HasEntityRegistry<?>)evaluator;
            BiMap<String, ? extends Entity> entities = hasEntityRegistry.getEntityRegistry();
            Entity winner = entities.get(hasEntityId.getEntityId());

            // Test for "probability" result feature
            if(targetFieldValue instanceof HasProbability){

                HasProbability hasProbability = (HasProbability)targetFieldValue;
                Double winnerProbability = hasProbability.getProbability(winner.getId());
                this.Probabilidad = String.valueOf(winnerProbability);
            }
        }

        Object unboxedTargetFieldValue = computable.getResult();
        this.Resultado = unboxedTargetFieldValue.toString();
    }
}
```

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Descripción del código de la capa negocio. Es necesario describir los códigos básicos que se encuentra en la capa de negocio, a continuación, se presenta los códigos: *Manejo de Objetos PMML e Inicio de Aplicación de Escritorio.*

- *Código de manejo de objetos PMML.* Ver [Anexo 7.7](#), Donde se detalle el comportamiento de la clase para el manejo de objetos *PMML* en java.
- *Código inicio de aplicación escritorio.* Ver [Anexo 7.8](#), Donde se detalle la ejecución de la aplicación, sus funciones de recepción de petición de información y respuesta de resultados.

Descripción del código de la capa presentación. Es necesario describir el código básico que se encuentra en la capa de presentación, a continuación, se presenta el código para manipular la información.

- *Código para manipulación de información.* Ver [Anexo 7.9](#), donde se detalla el comportamiento de la aplicación móvil SignalPred al momento de la recolección de valores, el método de envío de información y la recepción de información.

Fase de Pruebas

Una vez culminado la fase de codificación, se procederá a realizar pruebas para comprobar su correcto funcionamiento o verificar los posibles errores que se pueden presentar tomando en cuenta los siguientes aspectos.

- Para la primera prueba, se ingresará a la aplicación y seleccionará el modo de ingreso manual de datos, a continuación, se procederá al ingresar los valores en los espacios en blanco y al envío de información, pero con la conexión a Internet apagada.
- Para la segunda prueba, se realizará el envío de información, pero esta vez con la capa de negocios apagada.

Dentro de la fase de pruebas se realizará la ejecución del plan de pruebas. En la Tabla 25 se muestra el ingreso de datos con el servidor sin conexión.

Tabla 25: Ingreso de Datos con el servidor sin conexión

<i>Ingreso de datos - Servidor caído o sin comunicación</i>	
<i>Descripción</i>	Una vez ejecutada la aplicación y desplegando la actividad de ingresar valores manualmente, el usuario deberá ingresar todos los campos de valores solicitados, posteriormente, la información será enviada por Internet hacia el servidor remoto que la procesara y responderá a la aplicación el resultado del modelo.
<i>Condiciones de Ejecución</i>	Tener conexión a Internet

<i>Ingreso de datos - Servidor caído o sin comunicación</i>	
<i>Entrada</i>	<ul style="list-style-type: none"> • El usuario ingresa los valores de las variables meteorológicas manualmente. • El servidor remoto recibirá la información y lo procesará con el modelo de minería de datos • El servidor remoto publicara el resultado en la base de datos para acceso de los dispositivos inteligentes.
<i>Resultados esperados</i>	Tras el envío de información de las variables meteorológicas se espera el resultado del modelo en intervalos máximo de 30 segundo.
<i>Evaluación de la prueba</i>	Prueba exitosa.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Después de realizar un conjunto de 10 pruebas con distintos dispositivos móviles virtuales generados por el *Android Studio*, se constató que si se excede el tiempo máximo de respuesta el dispositivo toma la decisión de desechar ese envío de información tal como se demuestra en la Ilustración 22.

Ilustración 22: Resultado del Ingreso de Datos con el servidor sin conexión en el simulador de Android Studio



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

En la Tabla 26 se muestra el ingreso de datos sin acceso a Internet.

Tabla 26: Ingreso de datos sin acceso a Internet

<i>Ingreso de datos: Sin conexión a Internet</i>	
<i>Descripción</i>	Una vez ejecutada la aplicación y desplegando la actividad de ingresar valores manualmente, el usuario deberá ingresar todos los campos de valores solicitados, posteriormente la información será enviada por Internet hacia el servidor remoto que la procesara y responderá a la aplicación el resultado del modelo.
<i>Condiciones de Ejecución</i>	Ninguna
<i>Entrada</i>	<ul style="list-style-type: none"> • El usuario ingresa los valores de las variables meteorológicas • El servidor remoto recibirá la información y lo procesará con el modelo de minería de datos • El servidor remoto publicara el resultado en la base de datos para acceso de los dispositivos inteligentes.
<i>Resultados esperados</i>	Tras el envío de información de las variables meteorológicas se espera el resultado del modelo en intervalos máximo de 30 segundo.
<i>Evaluación de la prueba</i>	Prueba Fallida.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Después de realizar un conjunto de 10 pruebas con distintos dispositivos móviles virtuales generados por el Android Studio, se constató que los dispositivos se quedan congelados debido a la falta de conexión a Internet, los datos enviados para ser evaluados por el modelo nunca llegan al servido y consecuentemente no hay repuesta hacia el cliente.

Ilustración 23: Resultado del Ingreso de Datos sin acceso a Internet en el simulador de Android Studio



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

4.2.3. Discusión de los Resultados

A partir de los resultados mostrado anteriormente se puede concluir que el lenguaje del estándar PMML resulta eficiente para la traducción de un modelo de análisis inteligente de datos y su aprovechamiento en una aplicación móvil. Esto se puede comprobar que dado las librerías de PMML utilizadas en JAVA se incorporaron fácilmente.

A pesar de que la documentación de PMML muestra un ejemplo donde el modelo se puede ejecutar de forma nativa en los dispositivos Android, la versión *evaluator 1.3.4* y *model 1.3.5* no pudieron ser realizadas nativamente como se esperaba, por consiguiente, en futuras versiones se espera que se corrijan los inconvenientes presentados.

Durante las pruebas con el estándar PMML se logró ejecutar el modelo de análisis inteligente de datos en una aplicación de escritorio, brindando así el cumplimiento de un objetivo planteado. Para la comunicación entre la plataforma Móvil y la plataforma de Escritorio se necesitó una forma de establecer una comunicación, teniendo en cuenta que se utilizaron

servicios gratuitos de acceso a la información, la comunicación se efectuó con una base de datos remota.

El IDE Android Studio facilitó ampliamente el desarrollo de la aplicación móvil debido a que se encuentra en constante revisión y actualización, adicionalmente brinda un simulador de dispositivos móviles para realización pruebas de depuraciones de código; cabe recalcar que la empresa a cargo de las actualizaciones del IDE es Google.

La aplicación móvil *SignalPred* predice la pérdida de la señal en función de las variables meteorológicas. La aplicación se considera amigable e intuitiva ya que el usuario puede identificar las acciones a realizar dentro de la misma sirviendo de esta manera como herramienta de apoyo para la toma de decisiones.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

A partir de los resultados obtenidos se puede concluir que:

- La herramienta RapidMiner resultó ser la más apropiada para el estudio de minería de datos. Esto se pudo constatar a partir de las comparaciones realizadas con otras herramientas similares. Su utilización permitió un análisis rápido y automatizado de los diversos algoritmos, además de brindar un método para exportar un modelo de minería de datos.
- Con el análisis del pre-procesamiento de los datos realizado se concluyó que, de las 15 variables consideradas, solo 9 fueron relevantes para la predicción de la pérdida de señal.
- A partir del proceso de optimización incluido en RapidMiner y la comparación entre las tres técnicas consideradas (Árbol de Decisiones, Redes Neuronales y Análisis Discriminante) se pudo determinar que el algoritmo que predice con mayor exactitud la pérdida de señal es el Neural Net. Este índice de exactitud es de 100%.
- Mediante el uso del estándar PMML se transformó de manera efectiva y rápida el modelo de minería de datos a un objeto Java para ser aprovechado de manera práctica.
- La aplicación móvil desarrollada, SignalPred, resulta sencilla, amigable y funcional. Esta no solo incluye el modelo inteligente generado con PMML, sino que posibilita la lectura de los datos meteorológicos de manera manual y automática para predecir la pérdida de señal de manera efectiva.

5.2. Recomendaciones

El proyecto de investigación es de amplio interés para el área de telecomunicaciones, sirviendo así de apoyo a la toma de decisiones por parte de los especialistas en telecomunicaciones. No obstante, en opinión de los autores se pueden realizar las siguientes recomendaciones:

- Enriquecer el conjunto de datos actual con más información proveniente de los proveedores de servicios de Internet pública o privada.
- Realizar un estudio más profundo empleando otras técnicas de aprendizaje que sean soportadas por un estándar de modelado como PMML.
- Procesar el modelo de forma nativa en los dispositivos Android, para así no depender siempre de una conexión a un servidor de datos.
- Investigar nuevas versiones de *PMML* para el lenguaje Java Server Page que corrijan errores en el *SDK 1.7*, y de esta forma puedan ser usados en servidores web.

CAPÍTULO VI

BIBLIOGRAFÍA

- [1] I. R. Najarro, «La calidad de la conexión inalámbrica y su relación con las condiciones meteorológicas», Universidad Técnica Estatal De Quevedo, 2015.
- [2] D. Bri, M. Fernandez-Diego, M. Garcia, F. Ramos, y J. Lloret, «How the Weather Impacts on the Performance of an Outdoor WLAN», *IEEE Commun. Lett.*, vol. 16, n.º 8, pp. 1184-1187, ago. 2012.
- [3] I. R. Salavert y M. D. L. Pérez, *Ingeniería del software y bases de datos: tendencias actuales*. Ediciones de la Universidad de Castilla-La Mancha, 2000.
- [4] Agencia de Regulación y Control de las Telecomunicaciones, «Boletín estadístico del sector de telecomunicaciones». Ecuador.
- [5] Francisco G. Reyes, *Minería de patrones temporales basados en redes de restricciones*. Editorial Universidad de Almería, 2009.
- [6] M. N. Moreno García, L. A. M. Quintales, F. J. G. Peñalvo, M. José, y P. Martín, «Aplicación de Técnicas de Minería de Datos en la construcción y validación de modelos predictivos y asociativos a partir de especificaciones de requisitos de software».
- [7] F. Gorunescu, *Data Mining: Concepts, Models and Techniques*. Springer Berlin Heidelberg, 2011.
- [8] O. Maimon y L. Rokach, *Data Mining and Knowledge Discovery Handbook*. Springer US, 2010.
- [9] R. Engels, *Component-Based User Guidance in Knowledge Discovery and Data Mining*. IOS Press, Incorporated, 1999.
- [10] P. L. César y Daniel Santín Gonzalez, *Minería de datos: técnicas y herramientas*. Paraninfo Cengage Learning, 2007.
- [11] D. Ciucci, G. Wang, S. Mitra, y W. Z. Wu, *Rough Sets and Knowledge Technology: 10th International Conference, RSKT 2015, Held as Part of the International Joint Conference on Rough Sets, IJCRS 2015, Tianjin, China, November 20-23, 2015, Proceedings*. Springer International Publishing, 2015.
- [12] A. Guazzelli, M. Zeller, W. Lin, y G. Williams, «PMML: An open standard for sharing models», *R J.*, pp. 60-65, 2009.
- [13] D. DE LUCA, *Apss HTML5 para móviles - Desarrollo de aplicaciones para smartphones y tablets basado en tecnologías Web*: Alfaomega, 2013.

- [14] A. K. Pujari, *Data Mining Techniques*. Universities Press, 2001.
- [15] *Avances en Informática y Sistema Computacionales Tomo I (CONAIS 2006)*. Univ. J. Autónoma de Tabasco.
- [16] V. autores, *XXVIII Congreso de Marketing. AEMARK 2016 León*. 2016.
- [17] L. J. Aguilar, *Big Data, Análisis de grandes volúmenes de datos en organizaciones*. Alfaomega, 2016.
- [18] «KNIME | Analytics Platform KNIME». [En línea]. Disponible en: <http://www.knime.org/knime-analytics-platform>. [Accedido: 09-feb-2017].
- [19] F. Luna, *Desarrollo web para dispositivos móviles: Herramientas para diseñar y programar WebApps*. RedUsers, 2016.
- [20] IDC, «IDC: Smartphone OS cuota de mercado de 2016, 2015». [En línea]. Disponible en: <http://www.idc.com/promo/smartphone-market-share/os>. [Accedido: 04-abr-2017].
- [21] D. Bell, M. Parr, A. V. R. Elizondo, y D. B. M. P. A. V. R. E. G. L. Gutiérrez, *Java para estudiantes*. Pearson Educación, 2003.
- [22] N. Smyth, *Android Studio 2.2 Development Essentials - Android 7 Edition*: CreateSpace Independent Publishing Platform, 2016.
- [23] M. Qiu, W. Dai, y K. Gai, *Mobile Applications Development with Android: Technologies and Algorithms*. CRC Press, 2016.
- [24] A. Palandurkar, *Instant NetBeans IDE How-to*. Packt Publishing, Limited, 2013.
- [25] E. Burnette, *Eclipse IDE Pocket Guide: Using the Full-Featured IDE*. O'Reilly Media, 2005.
- [26] Á. Cobo, *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos, S.A., 2005.
- [27] M. M. Andrés, *SQL y desarrollo de aplicaciones en Oracle 8*. Universitat Jaume I, 2001.
- [28] P. P. G. Abenza, *Comenzando a programar con JAVA*: Universidad Miguel Hernández, 2015.
- [29] F. Minera, *Desarrollador Web Profesional*. Usershop.
- [30] Á. Arias, M. Arias, A. Durango, y I. T. C. Academy, *Curso de Desarrollo Web: 2ª Edición*. CreateSpace Independent Publishing Platform, 2016.

- [31] Constitución del Ecuador, «Constitución del Ecuador - 2008», *Regist. Of.*, vol. 449, n.º Principios de la participación Art., p. 67, 2008.
- [32] R. Correa y H. del P. Barrezueta, «Ley Orgánica de Telecomunicaciones», *Regist. Of. Órgano del Gob. del Ecuador*, vol. Tercer Sup, pp. 1-40, 2015.
- [33] I. N. G. Hugo y D. E. L. Pozo, «Ley Orgánica de Comunicación», pp. 1-24, 2013.
- [34] Juliana Aparecida Anochia1 y Haroldo Fraga de Campos Velho, «Meteorological data mining for climate precipitation prediction using neural networks», 2015.
- [35] J.-C. Kim, H. Jung, y K. Chung, «Mining Based Urban Climate Disaster Index Service According to Potential Risk», *Wirel. Pers. Commun.*, vol. 89, n.º 3, pp. 1009-1025, 2016.
- [36] S. S. Chaudhari y R. C. Biradar, «Resource prediction using wavelet neural network in mobile ad-hoc networks», en *2014 International Conference on Advances in Electronics Computers and Communications*, 2014, pp. 1-6.
- [37] L. F. A. Brito y M. K. Albertini, «Data Mining of Meteorological-related Attributes from Smartphone Data», *INFOCOMP J. Comput. Sci.*, vol. 15, n.º 2, pp. 1-9, 2016.
- [38] INAMHI, «Anuario Meteorológico», 2012.
- [39] L. Borrero, *Tecnologías de La Informacion En Internet*. Editorial Norma.
- [40] J. B. de Areba, *Metodología del análisis estructurado de sistemas*. Universidad Pontificia Comillas (Publicaciones), 2001.
- [41] «Metodologías del Desarrollo de Software». [En línea]. Disponible en: <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/>. [Accedido: 24-abr-2017].

CAPÍTULO VII

ANEXOS

7.1. Solicitudes de Información – CNT

Ilustración 24: Solicitud para Información en CNT



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

7.2. Solicitudes de Información – Transtelco S.A

Ilustración 25: Solicitud para Información en la empresa Transtelco S.A



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

7.3. Anexo de la Metodología de Desarrollo de Aplicación

Tabla 27: Metodologías para el desarrollo de software

<i>Metodologías</i>	<i>Conceptos</i>	<i>Fases de los Modelos</i>	<i>Características</i>
Programación Extrema XP	La metodología XP es un modelo ágil, el cual se basa en realimentar continuamente los cambios que se presenten en el ciclo de vida del proyecto. Es adecuada para proyectos con requisitos imprecisos y que pueden someterse a cambios según lo menciona [39] .	<ul style="list-style-type: none"> • Planificación. • Diseño. • Codificación. • Prueba. 	<ul style="list-style-type: none"> • Basada en Prueba y Error. • Reduce el costo del ciclo de vida. • Fundamenta Valores y Practicas • Está basada en 12 practicas. • Garantiza la calidad del software.
Desarrollo Rápido de Aplicaciones RAD	El modelo RAD está basado en las iteraciones y manejo de los prototipos y cuenta con las <i>herramientas case</i> ²⁸ las cuales permite acelerar el proceso según lo menciona [40].	<ul style="list-style-type: none"> • Modelado de Gestión. • Modelado de Datos. • Modelado de Proceso. • Generación de aplicaciones. • Prueba y Volumen. 	<ul style="list-style-type: none"> • Sus prototipos son iterativos y evolucionario. • Herramientas Especializadas. • Menor codificación manual. • Interfaz gráfica estándar.

²⁸ Herramienta CASE: Sirve para realizar la actividad de codificación del software para que puede ser entendible por la máquina.

Espiral	El modelo Espiral tiene como objetivo reducir la cantidad de riesgos. Es un modelo evolutivo ,el cual irá incrementando según la cantidad de código fuente, el incremento en el tiempo de ejecución y su planificación según lo menciona [41].	<ul style="list-style-type: none"> • Determinar Objetivo. • Análisis de Riesgo. • Desarrollar, Validar y Probar. • Planificación. 	<ul style="list-style-type: none"> • Puede ser adaptado y aplicado en lo largo de la vida del proyecto. • Permite aplicar un enfoque de construcción de prototipo en cualquier parte de la evolución del producto.
---------	--	---	--

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

7.4. Anexo de los Operadores de Modelos a utilizar

Tabla 28: Características de los operadores a analizar.

<i>Algoritmos</i>	<i>Operadores</i>	<i>Características</i>
<i>Árbol de Decisión</i>	Random Forest	<ul style="list-style-type: none"> • Genera un conjunto de árboles al azar. • Es fácil de interpretar. • Crea un modelo de clasificación que predice el valor de un atributo de destino también conocido como clase. • Posee un método eficaz para la estimación de los datos faltantes. • Posee un método para la detección de interacciones de variable. • Es eficiente con bases de datos de mayor tamaño.
	Tree Decision	<ul style="list-style-type: none"> • Crea un modelo de clasificación que predice el valor de un atributo de destino. • Es fácil de entender e interpretar. • Requiere poca preparación de datos. • Es robusto. • Es mucho más preciso con grandes cantidades de datos. • Admite atributos discretos y continuos.
<i>Red Neuronal</i>	Neural Net	<ul style="list-style-type: none"> • Usada para el reconocimiento de patrones, predicción, optimización. • Sirve para buscar elementos afines dentro de un conjunto. • Aprende un modelo por medio de una red neural de alimentación directa entrenado por

<i>Algoritmos</i>	<i>Operadores</i>	<i>Características</i>
		<p>un algoritmo de propagación hacia atrás-perceptrón multicapa.</p> <ul style="list-style-type: none"> • Es un sistema adaptativo.
	Deep Learning	<ul style="list-style-type: none"> • Se basa en una red neural artificial de alimentación hacia adelante de múltiples capas que está entrenado con descenso de gradiente estocástico utilizando retropropagación. • Conjunto de algoritmos que intentan modelar abstracciones de alto nivel en datos.
<i>Análisis Discriminante</i>	Regularized Discriminant Analysis - RAD	<ul style="list-style-type: none"> • Realiza un análisis discriminante regularizado(RDA). • Determina qué variables discriminan entre dos o más grupos. • Se puede usar para evaluar la adecuación de la clasificación. • Se puede asignar objetos a uno de un número de grupos. • Puede tener un descriptivo o un objetivo predictivo. • Puede ser empleado como un complemento para el análisis de conglomerados o Análisis de componentes principales.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

7.5. Diagramas de Casos de Uso.

La aplicación está compuesta de dos actores: *Usuario* y *Servicio*, por lo tanto, los diagramas de casos de uso en la aplicación son los siguientes:

USUARIO

- **Ingresar Datos**

En la Tabla 29 se puede apreciar las características que presenta el caso de Ingresar datos

Tabla 29: Caso de Uso - Ingresar Datos

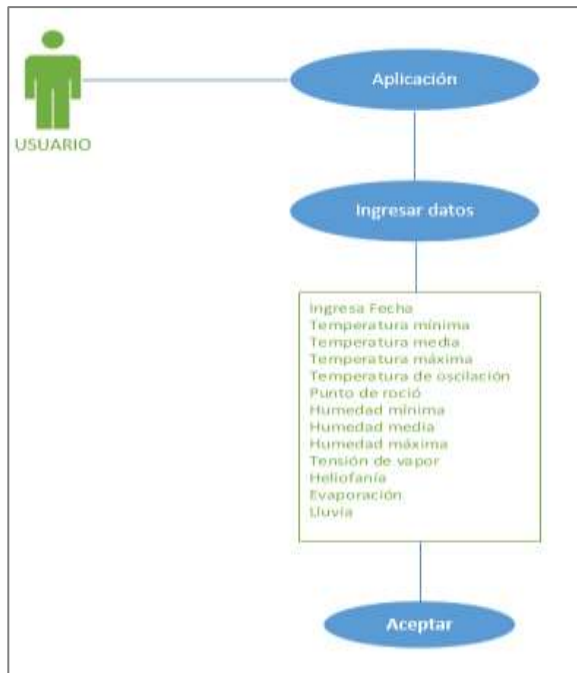
<i>CU-01</i>	<i>Ingresar Datos</i>
Resumen	El caso de uso correspondiente a la acción de Ingresar dato es realizada por el usuario en la aplicación.
Actor	Usuario.
Descripción	El usuario descarga la aplicación <i>SignalPred</i> y procede a ingresar los datos a analizar.
Resultado	El usuario ingresa los datos y espera el resultado.
Excepciones	Si el usuario no ingresa todos los valores establecidos se procederá a enviar un mensaje de error.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

A continuación, se mostrará el caso de uso obtenido de la Tabla anterior, donde podemos ver que el usuario puede observar ingresar los datos.

Ilustración 26: Caso de Uso - Ingresar Datos



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

- **Ver Historial**

Como se puede observar en la Tabla 30, se puede apreciar las características que presenta el caso Ver historial.

Tabla 30: Caso de Uso - Ver Historial

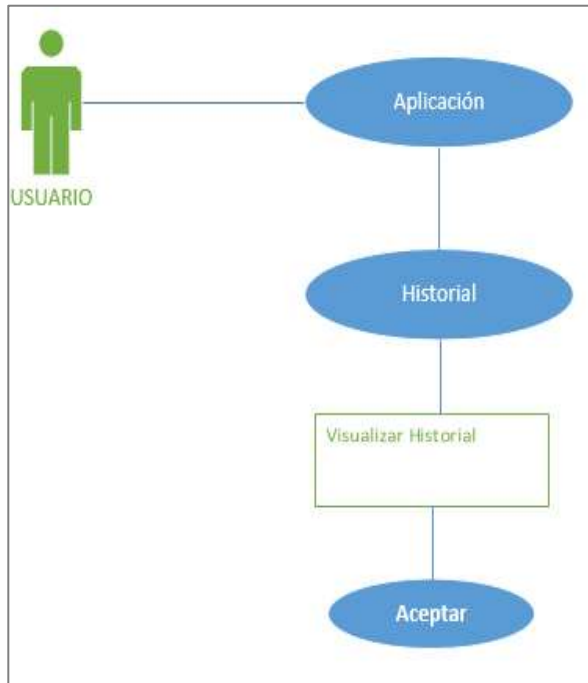
<i>CU-02</i>	<i>Ver Historial</i>
Resumen	El caso de uso correspondiente ver historial, el usuario podrá observar su historial de predicciones.
Actor	Usuario.
Descripción	El usuario podrá observar el historial de predicciones.
Resultado	Acceder al historial de predicción.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

En la Ilustración 27 se mostrará el caso de uso obtenido de la Tabla anterior, donde se ver observar el acceso al historial.

Ilustración 27: Caso de Uso - Ver Historial



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

SERVICIO

A continuación, se detalla el caso de uso a nivel de aplicación de escritorio, el cual funcionará como servicio de respuesta de las solicitudes de la aplicación móvil *SignalPred*.

- **Inicio del proceso**

En la Tabla 31, se puede apreciar las características que presenta el caso Inicio del proceso.

Tabla 31: Caso de Uso - Iniciar Proceso

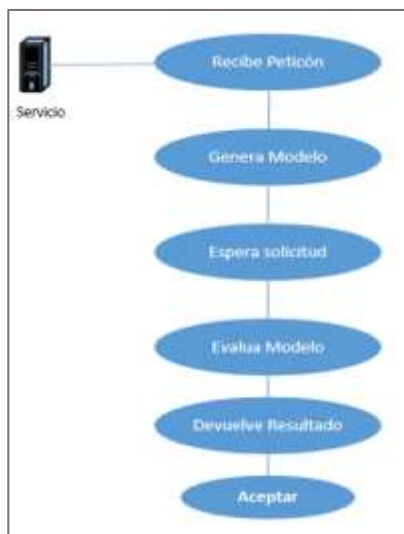
<i>CU-03</i>	<i>Inicio del proceso</i>
Resumen	El caso de uso correspondiente iniciaría los diferentes procesos necesarios a ejecutar.
Actor	Servicio
Descripción	Arranca la aplicación y construye el modelo <i>PMML</i> y luego que queda en modo espera hasta recibir información, una vez recibida la información procese a evaluar el modelo y generar un resultado. Estos resultados retornaran a la aplicación móvil <i>SignalPred</i> .
Resultado	Inicializa los servicios y genera un resultado.
Excepciones	Si el servicio de respuesta no está activo, no se podrá generar un resultado.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Como se observa la Ilustración 28 se muestra el caso de uso obtenido de la Tabla anterior, donde podemos observar el inicio del proceso.

Ilustración 28: Caso de Uso - Iniciar Proceso

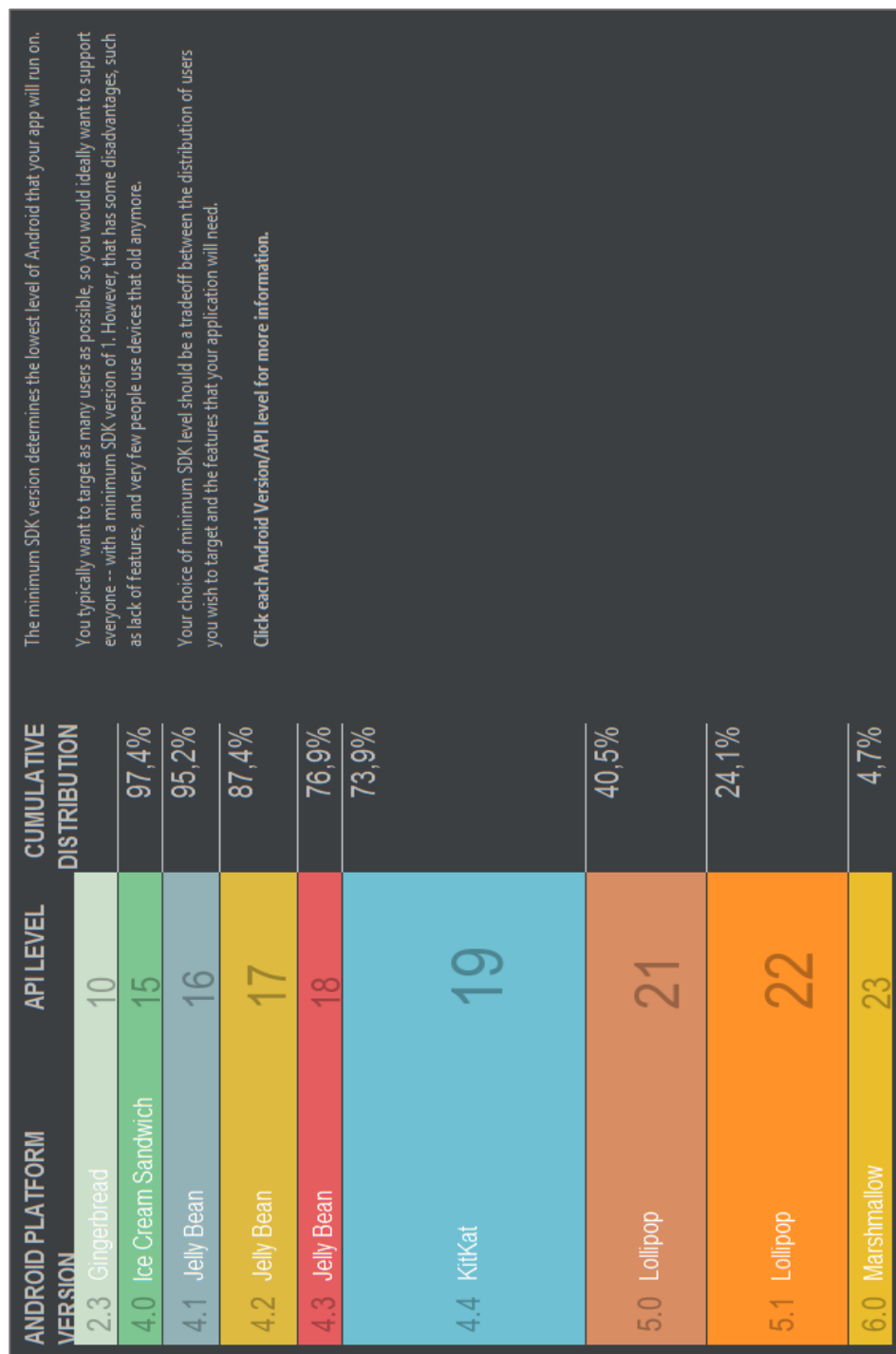


FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

7.6. API de Android Studio

Ilustración 29: API de Android Studio



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

7.7. Código clase usePMML.java

Tabla 32: Código clase PMML.Java

```
package classes;

import java.io.Serializable;

import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;

import org.dmg.pmml.Value;
import org.jpmmml.evaluator.Computable;
import org.jpmmml.evaluator.Evaluator;
import org.jpmmml.evaluator.FieldValue;
import org.jpmmml.evaluator.FieldValueUtil;
import org.jpmmml.evaluator.HasEntityId;
import org.jpmmml.evaluator.HasEntityRegistry;
import org.jpmmml.evaluator.HasProbability;
import org.jpmmml.evaluator.InputField;
import org.jpmmml.evaluator.ModelEvaluator;
import org.jpmmml.evaluator.ModelEvaluatorFactory;
import org.jpmmml.evaluator.OutputField;
import org.jpmmml.evaluator.TargetField;

import org.dmg.pmml.Entity;
import org.dmg.pmml.FieldName;
import org.dmg.pmml.PMML;

import com.google.common.collect.BiMap;
import com.google.common.collect.RangeSet;
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;
import javax.xml.transform.Source;

public class usePMML implements Serializable {
    private static final long serialVersionUID = 186535865133154191L;

    private PMML pmml;
    private Evaluator evaluator;
    private ModelEvaluator<?> modelEvaluator;
    private List<InputField> inputFields;
    public List<TargetField> targetFields;
    public List<OutputField> outputFields;
    Map<FieldName, FieldValue> arguments;
    public String Resultado;
    public String Probabilidad;

    public usePMML (PMML pmml){
        ModelEvaluatorFactory modelEvaluatorFactory = ModelEvaluatorFactory.newInstance();
        modelEvaluator = modelEvaluatorFactory.newModelEvaluator(pmml);

        Evaluator evaluator = modelEvaluator;

        this.pmml = pmml;
        camposActivos();
        camposObjetivos();
        camposResultados();
    }

    private void camposActivos(){
        List<InputField> inputFields = evaluator.getInputFields();

        for(InputField inputField : inputFields){
            org.dmg.pmml.DataField pmmlDataField = (org.dmg.pmml.DataField)inputField.getField();
            org.dmg.pmml.MiningField pmmlMiningField = inputField.getMiningField();

            org.dmg.pmml.DataType dataType = inputField.getDataType();
            org.dmg.pmml.OpType opType = inputField.getOpType();
        }
    }
}
```

```

        switch(opType){
            case CONTINUOUS:
                RangeSet<Double> validArgumentRanges = FieldValueUtil.getValidRanges(pmmlDataField);
                break;
            case CATEGORICAL:
            case ORDINAL:
                List<Value> validArgumentValues = FieldValueUtil.getValidValues(pmmlDataField);
                break;
            default:
                break;
        }
    }
    this.inputFields = inputFields;
}

private void camposObjetivos(){
    List<TargetField> targetFields = evaluator.getTargetFields();
    for(TargetField targetField : targetFields){
        org.dmg.pmml.DataField pmmlDataField = targetField.getDataField();
        org.dmg.pmml.MiningField pmmlMiningField = targetField.getMiningField(); // Could be null
        org.dmg.pmml.Target pmmlTarget = targetField.getTarget(); // Could be null

        org.dmg.pmml.DataType dataType = targetField.getDataType();
        org.dmg.pmml.OpType opType = targetField.getOpType();

        switch(opType){
            case CONTINUOUS:
                break;
            case CATEGORICAL:
            case ORDINAL:
                List<Value> validResultValues = FieldValueUtil.getValidValues(pmmlDataField);
                break;
            default:
                break;
        }
    }
    this.targetFields = targetFields;
}

private void camposResultados(){...15 lines }

public void preparar (Double [] Valores, String [] Campos){...22 lines }

public void evaluar(){
    Map<FieldName, ?> results = evaluator.evaluate(arguments);

    targetFields = evaluator.getTargetFields();
    for(TargetField targetField : targetFields){
        FieldName targetFieldName = targetField.getName();

        Object targetFieldValue = results.get(targetFieldName);
        if(targetFieldValue instanceof Computable){

            Computable computable = (Computable)targetFieldValue;
            if(targetFieldValue instanceof HasEntityId){

                HasEntityId hasEntityId = (HasEntityId)targetFieldValue;
                HasEntityRegistry<?> hasEntityRegistry = (HasEntityRegistry<?>)evaluator;
                SetMap<String, ? extends Entity> entities = hasEntityRegistry.getEntityRegistry();
                Entity winner = entities.get(hasEntityId.getEntityId());

                // Test for "probability" result feature
                if(targetFieldValue instanceof HasProbability){

                    HasProbability hasProbability = (HasProbability)targetFieldValue;
                    Double winnerProbability = hasProbability.getProbability(winner.getId());
                    this.Probabilidad = String.valueOf(winnerProbability);
                }
            }
        }
    }
}

```

Activar Wind
Va a Configuraci

Activar Wind

```

        Object unboxedTargetFieldValue = computable.getResult();
        this.Resultado = unboxedTargetFieldValue.toString();
    }
}

}

public long id(){
    return serialVersionUID;
}

public void inicializar(){
    camposActivos();
    camposObjetivos();
    camposResultados();
}

public static PMML loadPMML(String file){
    PMML pmml = null;
    File inputFilePath = new File( file );
    try{
        InputStream in = new FileInputStream( inputFilePath );
        Source source = ImportFilter.apply(new InputSource(in));
        pmml = JAXBUtil.unmarshalPMML(source);
    } catch( Exception e) {
        System.out.print(e.toString());
    }
    return pmml;
}

try{
    InputStream in = new FileInputStream( inputFilePath );
    Source source = ImportFilter.apply(new InputSource(in));
    pmml = JAXBUtil.unmarshalPMML(source);
} catch( Exception e) {
    System.out.print(e.toString());
}
return pmml;
}

public static PMML loadPMML(InputStream file){
    PMML pmml = null;
    try{
        Source source = ImportFilter.apply(new InputSource(file));
        pmml = JAXBUtil.unmarshalPMML(source);
    } catch( Exception e) {
        System.out.print(e.toString());
    }
    return pmml;
}
}

```

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

En la clase *use PMML.java* se encuentra en la aplicación de escritorio cumple la función manipular el objeto *PMML*

- Crear un objeto *PMML* a partir de un archivo externo.
- Genera un evaluador que sirve para ejecución del modelo.

- Extraer los campos de entrada del modelo.
- Ingresa datos al modelo.
- Evalúa y Genera el resultado del modelo.

7.8. Código inicio.java

Tabla 33: Código Inicio.java

```
package ventanas;

import clases.usePMML;
import clases.valoresModelo;
import static com.google.common.net.HttpHeaders.USER_AGENT;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.LinkedList;
import java.util.List;
import java.util.Locale;
import javax.swing.Timer;
import org.json.simple.JSONArray;

public class Inicio extends javax.swing.JFrame {

    //private Timer cronometro = new Timer(WIDTH, sucesos);
    private usePMML file_pmml=null;
    private String name="C:\\\\pmmlEjemplo.pmml";
    private String URL_subirRespuesta="";
    private String URL_descargarSolicitud="http://bdpiemboli.net23.net/PredWeather/descargarValoresPC.php";
    private String URL_comprobarSolicitud="http://bdpiemboli.net23.net/PredWeather/descargarRespuestaPC.php";
    private String URL_subirValor="http://bdpiemboli.net23.net/PredWeather/subirValoresPC.php";
    private int ID=0,cont=0,NDescargas=0;
    private Timer timer;
    private List<valoresModelo> aux;
    private int tiempoEnMilisegundos = 1000, repeti_conteo=15; // 1 seg

    /**
     * Creates new form Inicio
     */
    public Inicio() {
        initComponents();
        this.setLocationRelativeTo(null);
        //pba_barraEspera.setMaximum(repeti_conteo);
        timer = new Timer (tiempoEnMilisegundos, new ActionListener () {
            public void actionPerformed(ActionEvent e) {

                if(cont == repeti_conteo) {
                    aux = resivirJSON();
                    if(aux != null){

```



```

        hora+"\t"+
        "ID Dispositivo: "+matriz[c].getID_dis()+
        "\tResultado: "+file_pddl.Resultado+
        "\tProbabilidad: "+file_pddl.Probabilidad+"\t"+ID+"\n"
        :

        enviarJSON(Double.parseDouble(file_pddl.Resultado), Double.parseDouble(file_pddl.Probabilidad), matriz[c],
        JTextArea1.setText(texto);
    }
    matriz=null;
    JTextArea1.setText(JTextArea1.getText()+"Proceso Terminado, esperando Resultados\n");
    cont =0;
}
else {
    cont++; pba_barraEspera.setValue(cont);
    pba_barraEspera.setStringPainted(true);
}

}

});
}

private void cbx_respuestaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    if(cbx_respuesta.isSelected()){
        //----- Esperando respuesta
        lbl_estado.setText("Procesando");
        timer.start();
    }else{
        //----- No hago nada
        lbl_estado.setText("En Espera");
        timer.stop();
        cont=0; pba_barraEspera.setValue(cont);
    }

    private void enviarJSON(Double valor, Double proba, int idDisp ){
        try {
            // open a connection to the site
            URL url = new URL(URL_subirValor);
            URLConnection con = url.openConnection();
            // activate the output
            con.setDoOutput(true);
            PrintStream ps = new PrintStream(con.getOutputStream());
            // send your parameters to your site
            String texto ="valor="+valor+"&predic="+proba+"&idDisp="+idDisp;
            ps.print(texto);
            // we have to get the input stream in order to actually send the request
            con.getInputStream();
            // close the print stream
            ps.close();
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private List<valoresModelo> recibirJSON(){
        StringBuffer response = null;
        JSONObject jsonObj = new JSONObject();
        jsonObj.put("ID",ID);
        List l = new LinkedList();
        l.addAll(Arrays.asList(jsonObj));
        String jsonString = JSONValue.toJSONString(l);
        try {
            //Crea un nuevo objeto URL con la url donde pedir el JSON
            URL obj = new URL(URL_comprobarSolicitud);
            jsonString = URLEncoder.encode(jsonString, "UTF-8");
            //Crea un objeto de conexión

```

Activar


```

//Añadimos la cabecera
con.setRequestMethod("POST");
con.setRequestProperty("User-Agent", USER_AGENT);
con.setRequestProperty("Accept-Language", "es-ES,en;q=0.5");
//Creamos los parámetros para enviar
String urlParameters = "ID="+ID;
// Enviamos los datos por POST
con.setDoOutput(true);
DataOutputStream wr = new DataOutputStream(con.getOutputStream());
wr.writeBytes(urlParameters);
wr.flush();
wr.close();
//Capturamos la respuesta del servidor
int responseCode = con.getResponseCode();
BufferedReader in = new BufferedReader(
    new InputStreamReader(con.getInputStream()));
String inputLine;
response = new StringBuffer();

while ((inputLine = in.readLine()) != null) {
    response.append(inputLine);
}
//cerramos la conexión
in.close();
List<valoresModelo> temp = null;
JSONObject jsonObject = (JSONObject) JSONValue.parse(corregir(response.toString()));
if (((Long) jsonObject.get("success")).intValue() == 1) {
    int aux = Integer.parseInt((String) jsonObject.get("cantidad"));
    NDescargas = aux-ID;
    temp = recuperar(ID);
    ID = aux;

    return temp;
} else {
    //TextArea1.setText((String) jsonObject.get("message")+"n");
    e.printStackTrace();
}
return null;
}

private String corregir(String json){
    int aux=0;
    String nuevo="";
    for(int c=0;c<json.length();c++){
        if(json.toCharArray()[c] == ','){
            aux = c ;
        }
        for(int c2=0;c2<aux;c2++){
            nuevo+=json.toCharArray()[c2];
        }
        return nuevo;
    }
}

private List<valoresModelo> recuperar(int index){

    StringBuffer response = null;
    JSONObject jsonObj = new JSONObject();
    //Añadimos el nombre, apellidos y email del usuario
    jsonObj.put("cantidad",index);
    //Creamos una lista para almacenar el JSON
    List l = new LinkedList();
    l.addAll(Arrays.asList(jsonObj));
    //Creamos el String JSON
    String jsonString = JSONValue.toJSONString(l);
    try {
        //Creamos un nuevo objeto URL con la URL donde pedir el JSON
        URL obj = new URL(URL_DescargasSolitud);
        jsonString = URLEncoder.encode(jsonString, "UTF-8");
        //Creamos un objeto de conexión

```

```

//Añadimos la cabecera
con.setRequestMethod("POST");
con.setRequestProperty("User-Agent", USER_AGENT);
con.setRequestProperty("Accept-Language", "en-US,en;q=0.5");
//Creamos los parametros para enviar
String urlParameters = "cantidad="+index;
// Enviamos los datos por POST
con.setDoOutput(true);
DataOutputStream wr = new DataOutputStream(con.getOutputStream());
wr.writeBytes(urlParameters);
wr.flush();
wr.close();
//Capturamos la respuesta del servidor
int responseCode = con.getResponseCode();
BufferedReader in = new BufferedReader(
    new InputStreamReader(con.getInputStream()));
String inputLine;
response = new StringBuffer();

while ((inputLine = in.readLine()) != null) {
    response.append(inputLine);
}
//Mostramos la respuesta del servidor por consola
in.close();
List<valoresModelo> nuevo = new ArrayList<valoresModelo>();
JSONObject jsonObject = (JSONObject) JSONValue.parse(corregir(response.toString()));
if (((Long)jsonObject.get("success")).intValue() == 1){
    JSONArray array=(JSONArray)jsonObject.get("lista");
    for(int c=0; c<array.size();c++){
        JSONObject bal = (JSONObject)array.get(c);
        valoresModelo aux = new valoresModelo(bal);
        nuevo.add(aux);
    }
}
return nuevo;

```

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

La clase inicio.java se encarga de:

- Se definen las variables a utilizar, además de inicializar las direcciones web de donde se descargará la información.
- Se inicializa la clase definiendo un *Timer* que se utiliza para tener la aplicación a la escucha en caso de recibir peticiones de procesos.
- Eventos de un *ChekBox* para controlar cuando la aplicación está escuchando o no la recepción de datos.
- Enviar los datos procesados a cada uno de los solicitantes, empaquetando la información en *JSON*.
- Recibir los datos pendientes de procesos empaquetados en *JSON* de parte del servidor de datos.
- Definir un método que en el caso de comunicación con *PHP* se descarga una sentencia adicional de texto que debe ser eliminada.

7.9. Código de ingreManual.java

Tabla 34: Código ingreManual.java

```
package com.kinnshite.ak4.weatherpre;

import ...

public class ingreManual extends AppCompatActivity {

    private JSONParser jsonParser = new JSONParser();
    private valores tempValores;
    private static final String URL_subirDatos = "http://bdpiembcii.net23.net/PredWeather/subirValoresAndroid.php?";
    private static final String URL_descargarResultado = "http://bdpiembcii.net23.net/PredWeather/descargarValoresAndroid.php?";

    private ListView list;
    private ProgressDialog pDialog;
    private Timer timerLoopScan = new Timer();
    private TextView txt_tmin, txt_tmed, txt_tmax, txt_tcon, txt_temvap, txt_llyuvia, txt_eva, txt_puroc, txt_helio, txt_huin, txt_hmed, txt_hmax, txt_maqEspera;
    private String nameFile="index.txt", res="Tiempo excedido de espera", resultado, probabilidad, texto_error="";

    long scanPeriod = 1000;
    private int cont=0, idDescarga=0, idDispositivo=2, tiempoMuerto=4, id_error_vacio=0, id_error_rango=0;
    boolean valida = false, listoAvanzar=false;
    private Timer timer;
    private TimerTask timerTask;
    final Handler handler = new Handler();

    private String name_Ansatz="fichero.txt";

    @Override
    protected void onCreate(Bundle savedInstanceState) { ... }

    @Override
    protected void onStop() { ... }

    public void eventosPosibles(View v){

        limpiarValores();

    }

    public void eventosPosibles(View v){

        limpiarValores();
        if(generarValores()) {
            new AttemptLogin().execute(); // Genera los valores necesarios.
            // Ejecuta el proceso de envío y recepción de datos.
        }
    }

    private Boolean generarValores(){
        Boolean continuer=true;
        DatePicker fecha = (DatePicker)findViewById(R.id.datePicker);
        int dia = fecha.getDayOfMonth();
        int mes = fecha.getMonth();
        int año = fecha.getYear();
        if(comprobarValores()!=0){
            mostrarMensaje();
            continuer = false;
        } else if(comprobarRangos()!=0){
            mostrarMensaje();
            continuer = false;
        }

        if(continuer){
            tempValores = new valores(
                Double.parseDouble( String.valueOf(dia) ),
                Double.parseDouble( String.valueOf(mes) ),
                Double.parseDouble( String.valueOf(año) ),
                Double.parseDouble( txt_tmed.getText().toString() ),
                Double.parseDouble( txt_tmax.getText().toString() ),
                Double.parseDouble( txt_tmin.getText().toString() ),
                Double.parseDouble( txt_tcon.getText().toString() ),
                Double.parseDouble( txt_hmed.getText().toString() ),
                Double.parseDouble( txt_hmax.getText().toString() ),
                Double.parseDouble( txt_temvap.getText().toString() ),
                Double.parseDouble( txt_puroc.getText().toString() ) );
        }
    }
}
```

```

        if(continuar){
            tempValores = new valores(
                Double.parseDouble( String.valueOf(dia) ),
                Double.parseDouble( String.valueOf(mes) ),
                Double.parseDouble( String.valueOf(año) ),
                Double.parseDouble( txt_tmee.getText().toString() ),
                Double.parseDouble( txt_tmaz.getText().toString() ),
                Double.parseDouble( txt_tmia.getText().toString() ),
                Double.parseDouble( txt_tmoe.getText().toString() ),
                Double.parseDouble( txt_tmee.getText().toString() ),
                Double.parseDouble( txt_tmaz.getText().toString() ),
                Double.parseDouble( txt_tmenvap.getText().toString() ),
                Double.parseDouble( txt_tmproc.getText().toString() ),
                Double.parseDouble( txt_tmello.getText().toString() ),
                Double.parseDouble( txt_tmva.getText().toString() ),
                Double.parseDouble( txt_tmlluvia.getText().toString() ),
                Double.parseDouble( txt_tmhin.getText().toString() ),
                idDispositivo
            );
            continuar=true;
        }else{
            continuar= false;
        }
        return continuar;
    }

    private void mostrarMensaje() { new DialogoAlerta().show(getFragmentManager(),"a"); }

    private int comprobarVarices() {...}

    private int comprobarRangos() {...}

    Indices de la base de datos guardados en 'tat'

    Contador de rels;

    private void mostrarMensaje() { new DialogoAlerta().show(getFragmentManager(),"a"); }

    private int comprobarVarices() {...}

    private int comprobarRangos() {...}

    Indices de la base de datos guardados en 'tat'

    Contador de rels;

    public void limpiarValores() {...}

    private void presentarResultado(String prueba, String solucion, String id_res, String cadenaValores){
        Intent actividad = new Intent(getApplicationContext(), Resultado.class);
        actividad.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_SINGLE_TOP);
        actividad.putExtra("ID_PRUEBA","1");
        actividad.putExtra("PRUEBA_LIMPIA",prueba);
        actividad.putExtra("SOLUCION",solucion);
        actividad.putExtra("ID_RES",id_res);
        actividad.putExtra("VALORES",cadenaValores);
        startActivity(actividad);
        this.finish();
        //this.onDestroy();
    }

    //region Clase de cuadro de carga
    private class AttemptLogin extends AsyncTask<String, String, String> {

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            progressDialog = new ProgressDialog(IngreManual.this);
            progressDialog.setMessage("Cargando Datos al Modelo");
            progressDialog.setIndeterminate(false);
            progressDialog.setCancelable(false);
            progressDialog.show();
        }

        @Override
        protected String doInBackground(String... args) {
            int success;
            try {
                // Building Parameters
                List params = tempValores.params();
                // getting product details by making HTTP request
                JSONObject json = JsonParser.makeHttpRequest(URL_subirlatos, "POST", params);
                if(json!=null) {
                    success = json.getInt(valores.TAG_SUCCESS);
                    if (success == 1) {
                        startTimer();
                        return json.getString(valores.TAG_MESSAGE); //----- FIN DE CICLO.... OK
                    } else { //----- MENSAJE DE NO PODER OBTENER LA INFORMACION
                        return json.getString(valores.TAG_MESSAGE);
                    }
                }
            }
        }
    }

```

```

        } else {
            return "Error con la base de datos";
        }
    } catch (JSONException e) {
        stopInertask();
        e.printStackTrace();
    }
    return null;
}

@Override
protected void onPostExecute(String file_url) {
    // dismiss the dialog once product deleted
    pDialog.dismiss();
    //stopInertask();

    if (file_url.equals("Error con la base de datos")) {
        View layoutDetos = (View) findViewById(R.id.layoutDetos);
        layoutDetos.setVisibility(View.GONE);
        layoutDetos.setEnabled(false);

        View layoutCarga = (View) findViewById(R.id.layoutEspera);
        layoutCarga.setVisibility(View.VISIBLE);
    }
    Toast toast = Toast.makeText(getApplicationContext(), file_url, Toast.LENGTH_LONG);
    toast.show();
}

```

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

La clase valoresModelo.java se encarga de:

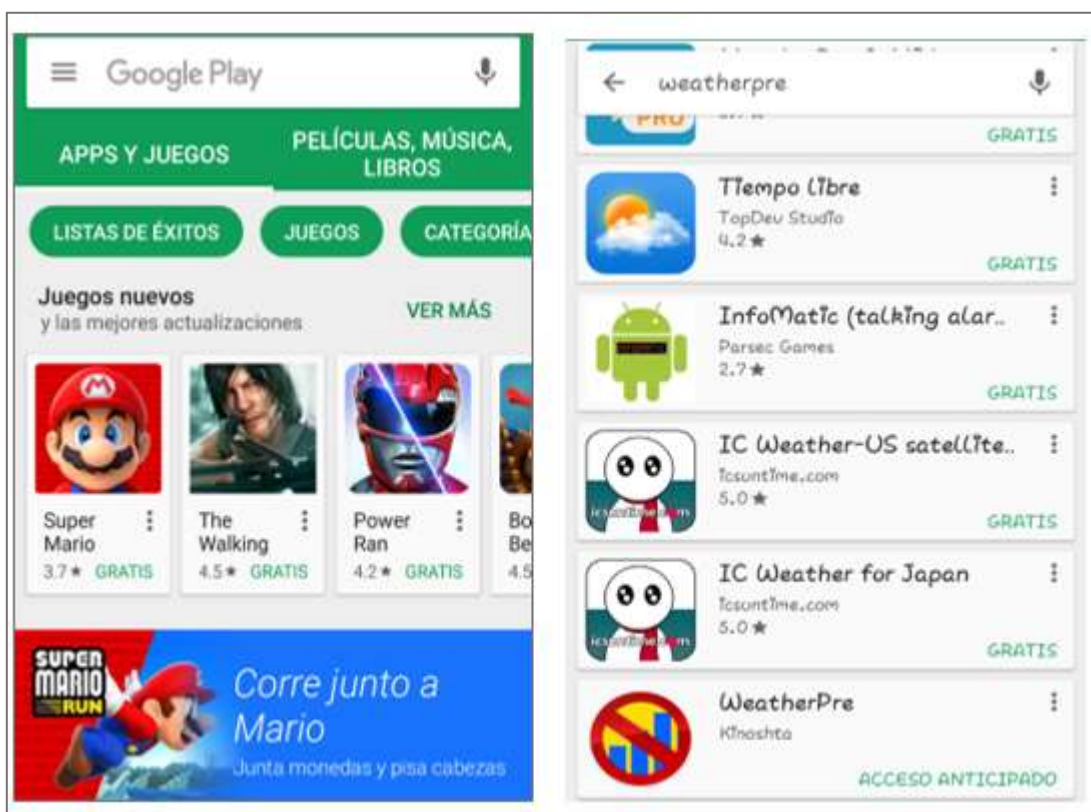
- Obtener los datos ingresados por el usuario de forma manual, en orden.
- Los datos se empaquetan en el objeto *JSON*.
- Los datos se suben a la base de datos y la aplicación se posiciona en modo espera para determinar la respuesta
- Cuando la respuesta lleva de parte del servidor, la aplicación visualizará de forma gráfica el resultado obtenido del modelo evaluado.

7.10. Manuales de la Aplicación y Servicio de Respuesta

Manual para descargar la aplicación *SignalPred*.

Para descargar la aplicación *SignalPred* se procede a abrir la tienda de *Google Play*. En el cuadro de búsqueda se escribe la palabra *SignalPred* y se selecciona para buscar la aplicación. A continuación, se mostrarán aplicaciones semejantes o de la misma rama que la aplicación que se desarrolló, se selecciona la apk *SignalPred*.

Ilustración 30: Ingreso a Google Play y Búsqueda de la App

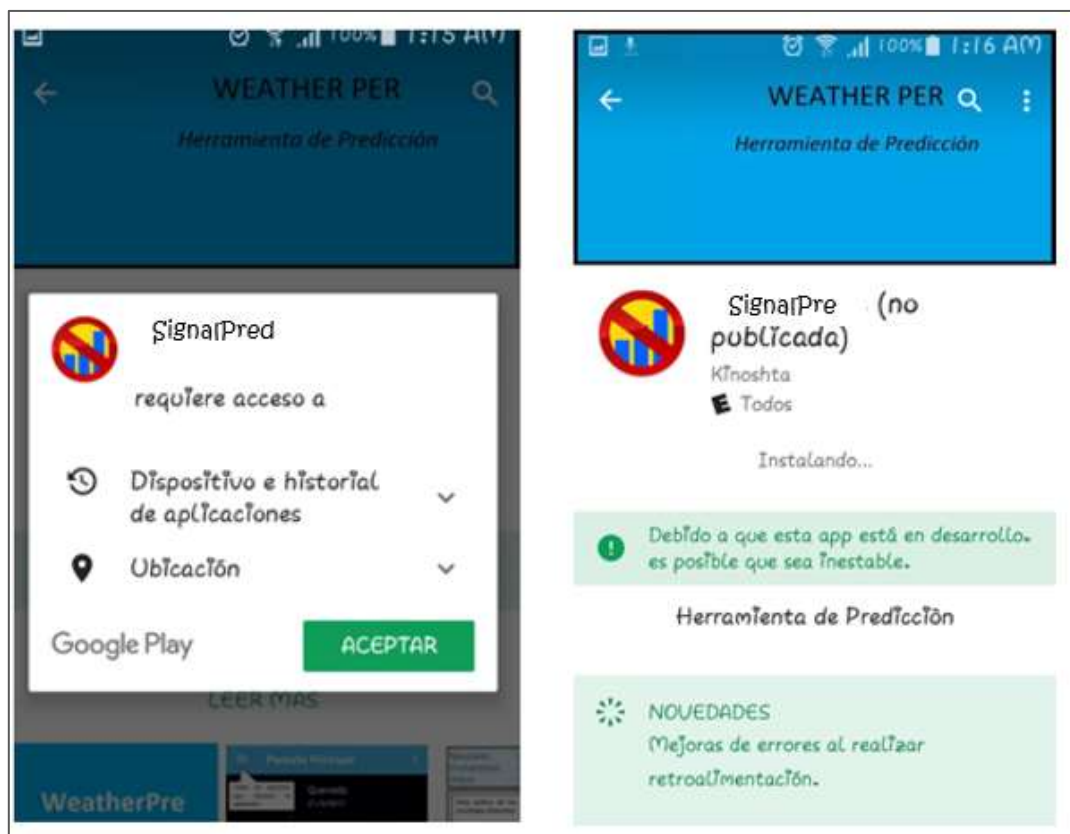


FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Una vez se encuentre la aplicación se debe seleccionar Instalar, a continuación, se mostrará los permisos necesarios para poder funcionar la aplicación y se acepte las condiciones, se procederá a instalar la aplicación.

Ilustración 31: Permisos para el funcionamiento de SignalPred



FUENTE: CONOCIMIENTOS PROPIOS

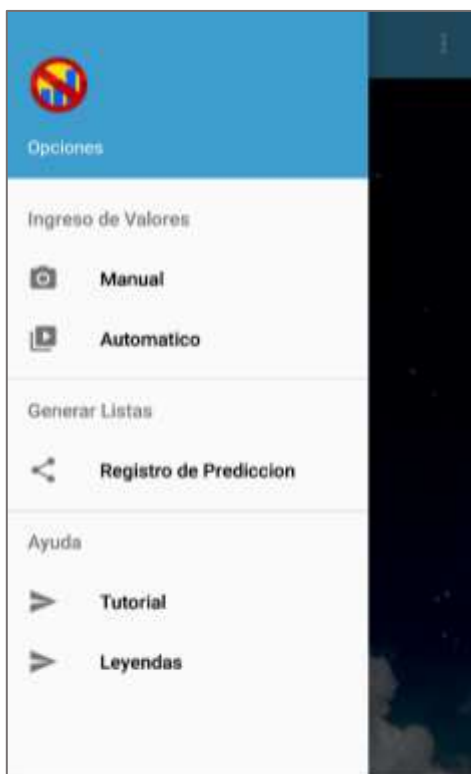
AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Manual de Usuario

La aplicación móvil *SignalPred* está destinada a la predicción de pérdida de señal en redes inalámbrica además de ser una herramienta de estudio para el mejoramiento de posibles modelos predictivos, la cual tiene como objetivo la recolección de datos de las variables meteorológicas ya se por ingreso manual o automático.

SignalPred cuenta con tres módulos: el primero para ingresar los valores y generar la pérdida de señal, el segundo para visualizar el historial de las predicciones y el tercero está orientado a la guía de uso de la aplicación.

Ilustración 32: Modulos de SignalPred



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Al ingresar por primera vez a la aplicación, se desplegará un tutorial de uso de la aplicación además de registrar el *Id* del dispositivo automáticamente en la base de datos.

Una vez dentro de la aplicación se mostrará la pantalla principal donde se observa de fondo los datos meteorológicos de la ubicación *GPS* siempre y cuando esta se encuentre disponible, en la barra lateral se aprecia los módulos disponibles.

Ingreso Manual.

En esta opción se mostrará una lista de todas las variables meteorológicas que el modelo necesita para generar un resultado.

Ilustración 33: Ingreso Manual de variables

The screenshot shows a web-based form for manual input of meteorological data. At the top, there are three input fields for date and time: '21', 'feb.', and '2016' on the first line; '22', 'mar.', and '2017' on the second line; and '23', 'abr.', and '2018' on the third line. Below these are five rows of input fields for specific meteorological variables, each with a corresponding icon and unit 'C°':

- Icon: Blue thermometer with a blue dot. Label: 'Temperatura Minima'. Unit: 'C°'.
- Icon: Green thermometer with a green dot. Label: 'Temperatura Media'. Unit: 'C°'.
- Icon: Red thermometer with a red dot. Label: 'Temperatura Maxima'. Unit: 'C°'.
- Icon: Blue thermometer with a blue wave. Label: 'Temperatura de Oscilacion'. Unit: 'C°'.
- Icon: Blue cloud with rain. Label: 'Punto de Rocio'. Unit: 'C°'.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Ingreso Automático.

Mediante el ingreso automático se mostrará las coordenadas *GPS* en cero hasta que se proceda a obtener la nueva ubicación *GPS*, posteriormente se visualiza los datos de las variables meteorológicas brindada por un servicio meteorológicos de terceros.

Ilustración 34: Localización del GPS



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Ilustración 35: Ingreso Automático de variables



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Resultados.

Esta opción se activa cuando se procede a evaluar el modelo ya se de forma manual o automática y genera un resultado que se expresa en números y gráficos, además ofrece la posibilidad de retroalimentar el modelo.

Ilustración 36: Resultado Obtenido



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Retroalimentación

Esta opción se activa cuando el usuario desea retroalimentar el sistema, este despliega todos los datos de las variables meteorológicas disponibles y también las opciones de resultados posibles.

Ilustración 37: Retroalimentación

Estos son los valores obtenidos de las variables meteorológicas, Modifique los valores que crea que son necesarios

Resultados de Predicción	
<input checked="" type="radio"/> Interferencia	
<input type="radio"/> Señal	
	Temperatura Mínima C° 13.98999
	Temperatura Media C° 18.99
	Temperatura Maxima C° 22.99
	Temperatura de Oscilacion C° 9.0
	Punto de Rocío C° 18.99

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Historial

Una vez que se elija esta opción se procederá a descargar una breve descripción de los resultados generados anteriormente.

Ilustración 38: Historial

PROBABILIDAD	
Fecha: 22/3/2017	
Señal 6.793	Interferencia 93.207
Fecha: 21/3/2017	
Señal 6.793	Interferencia 93.207
Fecha: 21/3/2017	
Señal 6.793	Interferencia 93.207
Fecha: 21/3/2017	
Señal 6.793	Interferencia 93.207
Fecha: 21/3/2017	
Señal 6.793	Interferencia 93.207

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

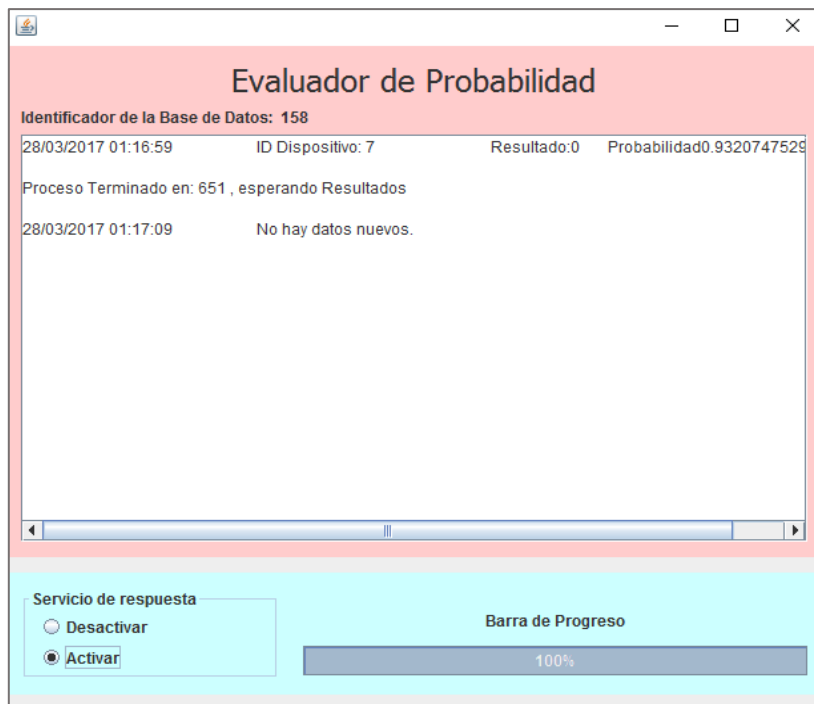
Manual del servicio de respuesta.

La aplicación de escritorio desarrollada en Java está orientada a construir y evaluar el modelo de minería de datos, esta modelo es entregado por una herramienta de minería de datos y posee una extensión PMML.

Una vez que se inicia la aplicación este cargar el modelo de minería de datos y si funciona correctamente se procede construir los servicios de respuesta.

Si los servicios de respuestas están activos, constantemente dentro de períodos de tiempo establecidos se procede a revisar la actividad de solicitud. En el caso de recibir una solicitud, se procederá a evaluar el modelo y responder con su respectivo resultado.

Ilustración 39: Servicio de Respuesta



FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

CAPÍTULO VIII GLOSARIO

Tabla 35: Glosario de Acrónimos

<i>Acrónimos</i>	<i>Concepto</i>
AD HOC	Técnicas de Clasificación Predictivas Ad Hoc
APIs	Application Programming Interface / Interfaz de Programación de Aplicaciones
ARCOTEL	Agencia de Regulación y Control de las Telecomunicaciones
CNT	Corporación Nacional de Telecomunicaciones
COF	Factores atípicos Clase
CSV	Valor separado por comas.
EVA	Evaporación
FTTx	Fibra Óptica
GPL	Licencia Pública General
HELIO	Heliofanía
HFC	Hybrid Fiber Coaxial / Cable Modem
HMAX	Humedad relativa máxima
HMED	Humedad relativa media
HMIN	Humedad relativa mínima
IDE	Integrated Development Environment.
INAMHI	Instituto Nacional de Meteorología e Hidrología
ISP	Proveedor de Servicios de Internet
JDK	Java Development Kit
JSON	JavaScript Object Notation / Notación en Objetos de JavaScript
KDD	Knowledge Discovery in Databases / Proceso de Extracción de Conocimientos
KNIME	Konstanz Information Miner
LLUV	Lluvia
MANESTs	Mobile Ad-hoc Networks / Redes Ad-hoc para Móviles
OHA	Open Handset Alliance.
PCA	Análisis de Componentes Principales
PESE	Pérdida de Señal.

<i>Acrónimos</i>	<i>Concepto</i>
PHP	PHP Hypertext Preprocessor
PMML	Predictive Model Markup Language / Lenguaje de marcado de modelos de Predicción
PUROC	Punto de roció
RDA	Regularized Discriminant Analysis
SDK	Software Development Kit- kit.
SGBD	Sistema Gestor de Base de datos.
SQL	Structured Query Language / Lenguaje de Consulta Estructurada.
TENVAP	Tensión de vapor de agua
TMAX	Temperatura máxima
TMED	Temperatura media
TMIN	Temperatura mínima
TOCS	Oscilación térmica
TSR	Protocolo de Enrutamiento de Origen basad en confianza.
URL	Uniform Resource Locator / Localizador Uniforme de Recurso
WEKA	Waikato Environment for Knowledge Analysis
WLAN	Wireless Local Area Networks / Redes de Área Local Inalámbricas
WMAN	Wireless Metropolitan Area Networks / Redes de Área Metropolitana Inalámbricas
WPAN	Wireless Personal Area Networks /Redes de Área Personal Inalámbricas
WSN	Redes de Sensores Inalámbricas
WWAN	Wireless Wide Area Networks / Redes de Área Ampliada Inalámbricas
XML	Lenguaje de marcado extensible.
XNU	X is Not Unix.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)

Tabla 36: Glosario de Palabras

<i>Palabra</i>	<i>Definición.</i>
Data Warehouse	Sistema de información central en el proceso de herramienta de minería de datos.
Multiplataforma	Software que trabaja en todos los sistemas operativos
Outliers o Datos Atípicos	Observaciones cuyos valores son muy diferentes a las otras observaciones del mismo grupo de datos.
OHA	Confederación de compañías de software, Hardware y Telecomunicaciones.
Scoring	Herramienta estadística que ayuda a la toma de decisiones.
Sesgo	Orientar o dirección que toma un asunto.
Variables Endógenas	Variables Dependientes.
Variables Exógenas	Variables Independientes.

FUENTE: CONOCIMIENTOS PROPIOS

AUTORES: AGUILAR ANGIE Y NUÑEZ GABRIEL (2017)