



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba

“2019, Año del Caudillo del Sur, Emiliano Zapata”

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

OPCIÓN I.- TESIS

TRABAJO PROFESIONAL

“MÓDULO DE DETECCIÓN DE PATRONES DE CONSUMO
A PARTIR DEL ANÁLISIS DE CLIENTES DE SERVICIOS
DE FACTURACIÓN ELECTRÓNICA”.

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN SISTEMAS
COMPUTACIONALES

PRESENTA:

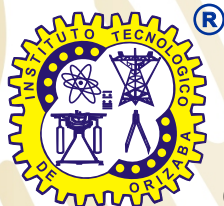
I.S.C. Luis Francisco Malpica Castro

DIRECTOR DE TESIS:

Dr. José Luis Sánchez Cervantes

CODIRECTOR DE TESIS:

M.C.E. Beatriz Alejandra Olivares Zepahua



ORIZABA, VERACRUZ, MÉXICO.

OCTUBRE 2019



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba

"2019, Año del Caudillo del Sur, Emiliano Zapata"

FECHA: 09/10/2019
DEPENDENCIA: POSGRADO
ASUNTO: Autorización de Impresión
OPCIÓN: I


C. LUIS FRANCISCO MALPICA CASTRO
CANDIDATO A GRADO DE MAESTRO EN:
SISTEMAS COMPUTACIONALES

De acuerdo con el Reglamento de Titulación vigente de los Centros de Enseñanza Técnica Superior, dependiente de la Dirección General de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

"MODULO DE DETECCION DE PATRONES DE CONSUMO A PARTIR DEL ANALISIS DE CLIENTES DE SERVICIOS DE FACTURACION ELECTRONICA".

Comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

A T E N T A M E N T E


MARIO LEONCIO ARRIOJA RODRIGUEZ
JEFE DE LA DIV. DE ESTUDIOS DE POSGRADO



Avenida Oriente 9 Núm. 852, Colonia Emiliano Zapata, C.P. 94320 Orizaba, Veracruz, México

Tel. 01 (272) 7 24 40 96, Fax. 01 (272) 7 25 17 28 e-mail: orizaba@itorizaba.edu.mx

www.orizaba.tecnm.mx





EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba

"2019, Año del Caudillo del Sur, Emiliano Zapata"

FECHA : 23/09/2019

ASUNTO: Revisión de Trabajo Escrito

C. MARIO LEONCIO ARRIJOJA RODRIGUEZ
JEFE DE LA DIVISION DE ESTUDIOS
DE POSGRADO E INVESTIGACION.
P R E S E N T E

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C. :

LUIS FRANCISCO MALPICA CASTRO

la cual lleva el título de:

**"MODULO DE DETECCION DE PATRONES DE CONSUMO A PARTIR DEL ANALISIS DE
CLIENTES DE SERVICIOS DE FACTURACION ELECTRONICA".**

Y concluyen que se acepta.

A T E N T A M E N T E

PRESIDENTE : DR. JOSE LUIS SANCHEZ CERVANTES

FIRMA

SECRETARIO : MCE BEATRIZ ALEJANDRA OLIVARES

FIRMA

VOCAL : DR. GINER ALOR HERNANDEZ

FIRMA

VOCAL SUP. : M.R.T. IGNACIO LOPEZ MARTINEZ

FIRMA

EGRESADO(A) DE LA MAESTRIA EN SISTEMAS COMPUTACIONALES

OPCION: I Tesis



Avenida Oriente 9 Núm. 852, Colonia Emiliano Zapata, C.P. 94320 Orizaba, Veracruz, México

Tel. 01 (272) 7 24 40 96, Fax. 01 (272) 7 25 17 28 e-mail: orizaba@itorizaba.edu.mx

www.orizaba.tecnm.mx



A mis abuelos, que en vida me protegieron y amaron...

Para siempre Bartolo y Evangelina

Francisco y Amada

AGRADECIMIENTOS

A Dios por darme la vida y unos padres amorosos.

A la virgen de Guadalupe, la que siempre me ha cuidado y bendecido en mis buenos y malos momentos.

A mis padres, José Luis y María Irene por amarme y apoyarme siempre y en todo momento de mi vida.

A mi novia Mariana, porque cuando mi luz se apagó permaneció a mi lado para volver a brillar.

A mis amigos Julio y Valdivia, porque sin su apoyo y cariño no hubiese llegado a terminar mi ingeniería.

Al doctor José Luis, por confiar en mí y por su apoyo a lo largo de mi maestría.

Al Lic. Sergio Salgado Orozco, por darme la oportunidad de llevar en alto el nombre de ESAD.

Al Instituto Tecnológico de Orizaba mi alma máter en donde viví tantas buenas experiencias y de donde me siento orgulloso de ser búho.

Al CONACyT por el apoyo económico que le otorga a los jóvenes para realizar sus estudios.

Contenido

Índice de figuras	VII
Índice de tablas	X
Índice de listados.....	XII
Resumen.....	XIV
Abstract	XV
Introducción.....	1
Capítulo 1. Antecedentes	3
1.1 Marco teórico	3
1.1.1 Descubrimiento de conocimiento en bases de datos	3
1.1.2 Proceso de Extracción, Transformación y Carga de datos	4
1.1.3 Sistema de recomendación	4
1.1.3.1 Basados en contenido.....	5
1.1.3.2 Filtrado colaborativo.....	5
1.1.3.3 Híbridos	6
1.1.3.4 Consciente de contexto.....	6
1.1.3.5 Basados en tipología.....	6
1.1.3.6 Basados en conocimiento	6
1.1.4 Minería de datos.....	7
1.1.4.1 Reglas de asociación	7
1.1.4.2 <i>Clustering</i>	7
1.1.4.3 Predicción	8
1.1.5 Facturación electrónica (<i>E-billing</i>).....	8
1.1.6 Comercio electrónico (<i>E-marketing</i>).....	8

1.2 Planteamiento del problema.....	9
1.3 Objetivo general y específicos.....	9
1.3.1 Objetivo general	9
1.3.2 Objetivos específicos.....	10
1.4 Justificación	10
Capítulo 2. Estado de la práctica	12
2.1 Trabajos relacionados	12
2.2 Análisis comparativo	27
2.3 Análisis comparativo algoritmo de series de tiempo	35
2.3.1 Estado del arte de series de tiempo.....	36
2.3.2 Evaluación de los algoritmos para series de tiempo	37
2.4 Propuesta de solución.....	38
Capítulo 3. Aplicación de la metodología.....	40
3.1 Sprint 0	40
3.1.1 Arquitectura de la aplicación.....	42
3.1.1.1 Descripción de las capas	43
3.1.1.2 Descripción de los componentes.....	44
3.1.1.3 Descripción del flujo de trabajo	45
3.1.2 Especificación de requisitos de la aplicación	48
3.1.2.1 Identificación de requerimientos.....	48
3.1.2.2 Especificaciones de caso de uso	49
3.1.2.2.1 Iniciar sesión.....	50
3.1.2.2.2 Tipificar cliente.....	51
3.1.2.2.3 Consultar clientes clasificados	54

3.1.2.2.4 Consultar tipos de clientes.....	56
3.1.2.2.5 Crear nuevo usuario	56
3.1.2.2.6 Modificar usuario	58
3.1.2.2.7 Eliminar usuario	59
3.1.2.2.8 Modificar recomendación.....	61
3.1.2.2.9 Pronosticar ventas	62
3.1.2.2.10 Pronosticar consumo de cliente	64
3.1.3 Diseño previo de la aplicación	66
3.1.3.1 Vista de inicio de sesión.....	66
3.1.3.2 Vista de inicio	67
3.1.3.3 Vistas usuario administrador	68
3.1.3.3.1 Vista gestión de usuarios.....	69
3.1.3.3.2 Vista para la gestión de las recomendaciones	71
3.1.3.4 Vistas usuario clasificador.....	73
3.1.3.4.1 Vista ver grupos.....	74
3.1.3.4.2 Vista clasificar.....	75
3.1.3.4.3 Vista ver clasificaciones	75
3.1.3.5 Vistas usuario pronosticador	76
3.1.3.5.1 Vista pronosticar cliente.....	77
3.1.3.5.2 Vista pronosticar ventas	78
3.1.4 Modelos conceptuales de la aplicación.....	79
3.1.4.1 Diagrama de clases	80
3.1.4.2 Esquema de la base de datos	81
3.1.5 Consideraciones del <i>Sprint 0</i>	82

3.2 <i>Sprint 1</i> generar modelo de <i>clustering</i>	83
3.3.1 Algoritmo <i>K-Means</i>	87
3.3.2 Algoritmo <i>EM</i>	93
3.3.3 Evaluación de los algoritmos para <i>clustering</i>	95
3.3 <i>Sprint 2</i> generar modelo de clasificación	96
3.3.1 Algoritmo C4.5 (J48).....	100
3.3.2 Algoritmo Naive Bayes	101
3.3.3 Algoritmo <i>ZeroR</i>	102
3.3.4 Algoritmo <i>Decision Table</i>	103
3.3.5 Algoritmo <i>Random Tree</i>	105
3.3.6 Evaluación de los algoritmos para clasificación	106
3.4 <i>Sprint 3</i> tipificar cliente	107
3.4.1 Interfaces	108
3.4.1.1 Interfaz ver grupos.....	108
3.4.1.2 Interfaz clasificar.....	108
3.4.1.3 Interfaz ver clasificaciones.....	109
3.4.2 Funcionalidades	110
3.4.2.1 Ver grupos.....	110
3.4.2.2 Clasificar.....	112
3.4.2.3 Ver clasificaciones	119
3.5 <i>Sprint 4</i> pronosticar ventas y consumo de cliente	123
3.5.1 Interfaces	123
3.5.1.1 Interfaz pronosticar ventas.....	123
3.5.1.2 Interfaz pronosticar consumo de cliente	124

3.5.2 Funcionalidades	124
3.5.2.1 Pronosticar ventas	124
3.5.2.2 Pronosticar consumo de cliente	131
3.6 <i>Sprint 5</i> gestión de usuarios y recomendaciones	137
3.6.1 Interfaces.....	137
3.6.1.1 Modificar recomendación	137
3.6.1.2 Gestionar usuarios	138
3.6.2 Funcionalidad	140
3.6.2.1 Modificar recomendación	140
3.6.2.2 Gestionar usuarios	145
3.7 <i>Sprint 6</i> iniciar sesión	154
3.7.1 Interfaces.....	154
3.7.1.1 Iniciar sesión	154
3.7.1.2 Inicio	154
3.7.2 Funcionalidad	155
3.7.2.1 Iniciar sesión	155
Capítulo 4. Resultados	160
4.1 Caso de estudio 1	160
4.2 Caso de estudio 2	180
4.2.1 Respuesta pregunta 1	180
4.2.2 Respuesta pregunta 2 y 3.....	185
Capítulo 5. Conclusiones y recomendaciones	191
5.1 Conclusiones	191
5.2 Recomendaciones	192

Productos académicos	193
Referencias	194
Anexo 1	XVI
Anexo 2	XVII
Anexo 3	XVIII
Anexo 4	XIX
Anexo 5	XX

Índice de figuras

Figura 3.1 Esquema general de la solución.	41
Figura 3.2 Proceso para el agrupamiento y predicción de clientes.	41
Figura 3.3 Proceso para la obtención de pronósticos mediante series de tiempo.	42
Figura 3.4 Arquitectura propuesta.	43
Figura 3.5 Diagrama de secuencia del flujo de trabajo de la arquitectura.	47
Figura 3.6 Diagrama de casos de uso.	49
Figura 3.7 Diagrama de secuencia Tipificar cliente.	53
Figura 3.8 Diagrama de secuencia pronosticar ventas.	64
Figura 3.9 Diagrama de secuencia pronosticar consumo de cliente.	66
Figura 3.10 Vista previa para inicio de sesión.	67
Figura 3.11 Vista de inicio.	68
Figura 3.12 Opciones del usuario administrador.	69
Figura 3.13 Vista general de la gestión de usuarios.	70
Figura 3.14 Vista para un nuevo usuario.	70
Figura 3.15 Vista para la modificación de un usuario.	71
Figura 3.16 Vista general de la gestión de recomendaciones.	72
Figura 3.17 Vista para la modificación de una recomendación.	73
Figura 3.18 Opciones del usuario clasificador.	74
Figura 3.19 Vista de grupos.	74
Figura 3.20 Vista para clasificar clientes.	75
Figura 3.21 Vista clasificaciones.	76
Figura 3.22 Opciones del usuario pronosticador.	77
Figura 3.23 Vista pronosticar cliente.	78
Figura 3.24 Vista pronosticar ventas.	79
Figura 3.25 Diagrama de clases de la aplicación.	80
Figura 3.26 Esquema de la base de datos de la aplicación.	81
Figura 3.27 Gráfica de resultados utilizando distancia Euclidiana con números pares.	91

Figura 3.28 Gráfica de resultados utilizando distancia Euclidiana con números impares.91

Figura 3.29 Gráfica de resultados utilizando distancia Manhattan con números pares.92

Figura 3.30 Gráfica de resultados utilizando distancia Manhattan con números impares.92

Figura 3.31 Gráfica de resultados del algoritmo EM.94

Figura 3.32 Interfaz ver grupos. 108

Figura 3.33 Interfaz clasificar. 109

Figura 3.34 Interfaz ver clasificaciones. 109

Figura 3.35 Recomendación mostrada al usuario. 112

Figura 3.36 Clientes por clasificar. 115

Figura 3.37 Confirmación para la clasificación de clientes. 115

Figura 3.38 Clasificaciones. 119

Figura 3.39 Interfaz pronosticar ventas. 123

Figura 3.40 Interfaz pronosticar consumo de cliente. 124

Figura 3.41 Formulario pronosticar ventas. 130

Figura 3.42 Resultados de pronosticar ventas. 130

Figura 3.43 Formulario pronosticar consumo de cliente. 136

Figura 3.44 Resultados de pronosticar consumo de cliente. 136

Figura 3.45 Interfaz modificar recomendación. 137

Figura 3.46 Formulario para actualizar recomendación. 138

Figura 3.47 Interfaz gestionar usuarios. 139

Figura 3.48 Formulario nuevo/actualizar usuario. 139

Figura 3.49 Resultados de búsqueda de grupo. 142

Figura 3.50 Formulario de modificación de recomendación. 143

Figura 3.51 Recomendación actualizada. 145

Figura 3.52 Resultados de buscar usuario. 148

Figura 3.53 Interfaz de modificación de un usuario. 150

Figura 3.54 Modificación del usuario. 153

Figura 3.55 Formulario para un nuevo usuario.....	153
Figura 3.56 Interfaz iniciar sesión.....	154
Figura 3.57 Inicio usuario clasificador.	155
Figura 3.58 Inicio usuario pronosticador.....	155
Figura 3.59 Inicio usuario administrador.....	155
Figura 4.1 Red Bayesiana generada en Weka.	161
Figura 4.2 Tabla de probabilidad del nodo Ciudad.	162
Figura 4.3 Sección del árbol generado para el algoritmo K-Means con distancia Euclidiana.	175
Figura 4.4 Sección del árbol generado para el algoritmo K-Means con distancia Manhattan.....	176
Figura 4.5 Árbol generado para el algoritmo EM.....	176
Figura 4.6 Red Bayesiana para el algoritmo K-Means con distancia Euclidiana.....	177
Figura 4.7 Red Bayesiana para el algoritmo K-Means con distancia Euclidiana.....	177
Figura 4.8 Red Bayesiana para el algoritmo EM.	178
Figura 4.9 Interfaz de inicio para el usuario clasificador.	181
Figura 4.10 Selección de la opción “Ver grupos”.	181
Figura 4.11 Visualización de grupos de clientes.....	182
Figura 4.12 Clasificación de clientes.	183
Figura 4.13 Confirmación de clasificación de un cliente.	183
Figura 4.14 Resultado de clasificación.	184
Figura 4.15 Clientes clasificados.....	185
Figura 4.16 Interfaz de inicio para usuario pronosticador.....	186
Figura 4.17 Selección de la opción “Ventas”.	186
Figura 4.18 Formulario pronosticar ventas.	187
Figura 4.19 Resultados pronosticar ventas.	187
Figura 4.20 Formulario para pronosticar el consumo de un cliente.	188
Figura 4.21 Resultado de la búsqueda de clientes.	189
Figura 4.22 Resultados del pronóstico de cliente.	189

Índice de tablas

Tabla 2.1 Análisis comparativo de los artículos relacionados.....	27
Tabla 2.2 Comparativa algoritmos series de tiempo.....	37
Tabla 3.1 Variables consideradas para clustering.	84
Tabla 3.2 Tipo de dato de cada variable.	85
Tabla 3.3 Variables ya transformadas	86
Tabla 3.4 Resultados utilizando distancia Euclidiana con números pares.	87
Tabla 3.5 Resultados utilizando distancia Euclidiana con números impares.	88
Tabla 3.6 Resultados utilizando distancia Manhattan con números pares.....	89
Tabla 3.7 Resultados utilizando distancia Manhattan con números impares.....	90
Tabla 3.8 Resultados del algoritmo EM.....	93
Tabla 3.9 Resultado obtenido automáticamente por EM.	94
Tabla 3.10 Distribución de clientes acorde a los grupos encontrados.	98
Tabla 3.11 Resultados del algoritmo C4.5.....	100
Tabla 3.12 Matriz de confusión del algoritmo C4.5.....	101
Tabla 3.13 Resultados del algoritmo Naive Bayes.	101
Tabla 3.14 Matriz de confusión del algoritmo Naive Bayes.	102
Tabla 3.15 Resultados del algoritmo ZeroR.	102
Tabla 3.16 Matriz de confusión del algoritmo ZeroR.	103
Tabla 3.17 Resultados del algoritmo Decision Table.....	104
Tabla 3.18 Matriz de confusión del algoritmo Decision Table.....	104
Tabla 3.19 Resultados del algoritmo Random Tree.	105
Tabla 3.20 Matriz de confusión del algoritmo Random Tree.....	105
Tabla 3.21 Resultados de los algoritmos de clasificación.....	107
Tabla 4.1 Resultados para K-Means utilizando distancia Euclidiana.	163
Tabla 4.2 Resultados para K-Means utilizando distancia Manhattan.....	165
Tabla 4.3 Resultados para EM.	168
Tabla 4.4 Resultados óptimos para clustering.....	171
Tabla 4.5 Resultados algoritmo J48.	173

Tabla 4.6 Resultados algoritmo BayesNet.174
Tabla 4.7 Características nuevos grupos de clientes.179

Índice de listados

Listado 3.1 Método de inicialización.	110
Listado 3.2 Método buscaRecomendacion.....	111
Listado 3.3 Búsqueda de clientes por clasificar.....	113
Listado 3.4 Método buscaPorClasificar.	114
Listado 3.5 Método clasificarClientes.	116
Listado 3.6 Método clasificar.	117
Listado 3.7 Método clasificaCliente.	118
Listado 3.8 Método buscaRecientes.	119
Listado 3.9 Método getClassificados.....	120
Listado 3.10 Método buscaClienteClasificado.	121
Listado 3.11 Método buscaClasificado.	122
Listado 3.12 Método pronosticarVenta.	125
Listado 3.13 Método buscaVentas.	126
Listado 3.14 Método intermedio para la pronosticación de ventas.	127
Listado 3.15 Método pronosticarVentas.	128
Listado 3.16 Método generarARFFV.....	129
Listado 3.17 Método pronosticarCliente.	131
Listado 3.18 Método buscaComprasCliente.	132
Listado 3.19 Método pronosticaCliente.	133
Listado 3.20 Método pronosticarCliente.	134
Listado 3.21 Método generarARFFC.....	135
Listado 3.22 Método buscarGrupo.	140
Listado 3.23 Método buscarGrupo.	141
Listado 3.24 Método guardaRecomendacion.	143
Listado 3.25 Método guardarRecomendacion.....	144
Listado 3.26 Método buscarUsuario.	145
Listado 3.27 Método buscarUsuario con parámetros.	147
Listado 3.28 Método eliminarUsuario.	148

Listado 3.29 Método eliminarUsuario.	149
Listado 3.30 Método guardarUsuario.	151
Listado 3.31 Método guardarUsuario.	152
Listado 3.32 Método login.	156
Listado 3.33 Método iniciarSesion.....	157
Listado 3.34 Opciones según tipo de cliente.	158
Listado 3.35 Método logout.	159

Resumen

La empresa Solución Integral de Facturación Electrónica e Informática S. A. de C. V. (SIFEI) se dedica a la venta/renta de servicios de facturación electrónica a clientes de distinta índole (personas físicas y morales); a lo largo de su estancia en el mercado, la empresa se apoya de la generación manual de campañas publicitarias con el propósito de atraer más clientes a sus servicios de facturación, sin embargo, en la actualidad dichas campañas se publican de manera general para todos los clientes, sin considerar sus necesidades o preferencias. Con relación a lo anterior y aunado a la creciente competencia, SIFEI cada vez pierde más clientes, obtiene algunos y retiene a pocos. Actualmente SIFEI almacena la información de sus clientes, sin embargo, la información almacenada no se aprovecha como apoyo para generar conocimiento que ayude con la toma de decisiones en cuanto a campañas publicitarias, ofertas, servicios complementarios, entre otros.

Teniendo en cuenta esta problemática, este trabajo de tesis desarrolló un módulo de software que implementa técnicas de minería de datos (clasificación, agrupamiento, por mencionar algunas) para obtener beneficios como la adquisición de conocimiento acerca de las preferencias y necesidades de los clientes, para atacar los problemas que conllevaron a clientes antiguos a dejar de trabajar con SIFEI, efectuar campañas publicitarias que atraigan a nuevos clientes y que los clientes ya existentes mejoren su experiencia para con la empresa. Lo anterior bajo el enfoque de la metodología SCRUM, la aplicación *Web* generada en este tema de tesis se desarrolló en JSF (JavaServer Faces), utilizando técnicas de minería de datos, guardando las recomendaciones generadas en el SGBD (Sistema Gestor de Base de Datos) PostgreSQL.

Abstract

The company Solución Integral de Facturación Electrónica e Informática S. A. de C. V. (SIFEI) is dedicated to the sale/rent of electronic invoicing services to different kinds of costumers (individuals and companies), throughout its stay in the market, the company has relied on the manual generation of advertising campaigns in order to attract more customers to its invoicing services, however, currently such campaigns are published generally for all customers, without considering their needs or preferences. In relation to the previously stated and considering its growing competition, SIFEI increasingly loses more customers and retaining a few. Currently, SIFEI stores the information of its clients; however, the stored information is not used as a support to generate knowledge that helps with decision making regarding advertising campaigns.

In order to address the problem, this thesis work aims to develop a software module that implements data mining techniques (classification, grouping, to name a few) to obtain benefits such as the acquisition of knowledge about customer preferences and needs, to address the issues that led old customers to stop working with SIFEI, to carry out advertising campaigns that attract new customers and that existing customers improve their experience with the company. Considering an approach under the SCRUM methodology, developing the module proposed in JSF (JavaServer Faces) through IDE (Integrated Development Environment) NetBeans, using data mining techniques, saving the recommendations generated in the SGBD (Database Management System) PostgreSQL.

Introducción

Actualmente se vive la llamada “era digital”, la cual repercute en el estilo y calidad de vida del ser humano, por ejemplo: si una máquina quirúrgica falla al momento de realizar un corte, perjudicará gravemente al paciente que depende de la operación que se supone debía salvar su vida, o ¿qué pasaría si el sistema de navegación de un avión tardara en comunicarse con la torre de control de un aeropuerto? Se incrementarán las posibilidades de que suceda un accidente que lastimaría a demasiadas personas. Considerando los ejemplos planteados previamente, llevar a cabo sistemas de software carentes de calidad en su diseño, desarrollo y mantenimiento, no es una opción viable.

Los sistemas que proveen recomendaciones a las personas tienen un papel muy importante en la vida diaria de éstas, ya que el tiempo que las personas pasan tomando decisiones con base en los resultados que los sistemas de recomendación les ofrecen, significa perder un minuto de tiempo o todo el día, por ejemplo, para alguien que desea ver una película o elegir algún restaurante para comer no le presta demasiada atención, pero, ¿si se tratara de saber en qué empresa comprar acciones? La rapidez y precisión de la recomendación es de suma importancia, ya que repercutiría en las finanzas de la persona interesada en invertir.

De igual forma, los sistemas de recomendación se basan en la información recabada por parte de los usuarios que utilizan, por ejemplo, Amazon®, YouTube® y Facebook®, por mencionar sólo algunos, dicha información se almacena en sistemas gestores de bases de datos o cualquier otro tipo de repositorios de información para después consultarse y de esa forma determinar las opciones que más se adapten a las necesidades de los clientes, o en su defecto, proveer artículos de posible interés con el fin de aumentar las ventas.

Por otra parte, la minería de datos busca encontrar conocimiento oculto en las grandes cantidades de datos almacenados ya sea en una base de datos relacional, almacenes de datos de distintos formatos (hojas de cálculo, documentos de texto, entre otros) y la Web, por mencionar algunos. La minería de datos utiliza algoritmos

para la obtención de modelos que interpretados por una persona (que entiende el contexto de los datos) se vuelven conocimiento útil para la toma de decisiones.

Por lo cual, en el presente trabajo de tesis se propone la creación de un módulo de software que aplique técnicas de minería de datos para el análisis y detección de patrones de consumo, para la generación de recomendaciones de servicios de facturación electrónica a partir del análisis de clientes.

La presente tesis está estructurada de la siguiente forma:

En el capítulo 1 se presentan los conceptos básicos para el entendimiento del problema, el planteamiento del mismo, el objetivo general, objetivos específicos y la justificación para este trabajo de tesis.

En el capítulo 2 se aborda el estado del arte, es decir, los artículos más relevantes relacionados con minería de datos y sistemas de recomendación, así también una tabla comparativa de los mismos.

El capítulo 3 muestra la aplicación de la metodología de desarrollo de software para la elaboración del módulo de detección de patrones de consumo a partir del análisis de clientes de servicios de facturación electrónica propuesto en este trabajo de tesis.

El capítulo 4 plasma la implementación del módulo en dos casos de estudio para determinar la eficiencia del sistema propuesto.

Por último en el capítulo 5 se muestran las conclusiones obtenidas a lo largo del desarrollo del proyecto de tesis y de igual forma algunas recomendaciones al mismo.

Capítulo 1. Antecedentes

En el presente capítulo se definen los conceptos más relevantes relacionados con el trabajo a realizar, de igual forma se presenta el problema a resolver, el objetivo general, los objetivos específicos y por último la justificación del proyecto a desarrollar.

1.1 Marco teórico

A continuación, se presentan algunos conceptos relacionados con el tema de investigación que se aborda con este proyecto de tesis.

1.1.1 Descubrimiento de conocimiento en bases de datos

Descubrimiento de Conocimiento en Bases de Datos (*Knowledge Discovery in Databases*, KDD) este proceso consta de varias fases como la preparación de datos (selección, limpieza y transformación), su exploración y auditoría, minería de datos propiamente dicha (desarrollo de modelos y análisis de datos), evaluación, difusión y utilización de modelos. Por lo anterior, se observa que en el proceso de extracción (descubrimiento) de conocimiento en bases de datos (KDD) [1], la secuencia de fases es la siguiente:

- **Selección:** A partir de unos datos iniciales que se dispone, generalmente la información que se quiere investigar sobre un cierto dominio se encuentra en bases de datos u otras fuentes diversas. Lo que se busca es una fuente de datos unificada, accesible y desconectada del trabajo transaccional.
- **Exploración:** En esta fase se utilizan herramientas de consulta (*Query*), que obtengan los datos necesarios para su análisis.
- **Limpieza:** Se busca en esta fase a los datos atípicos y valores desaparecidos.
- **Transformación:** Si la naturaleza de los datos no es la correcta para el uso de técnicas de minería de datos, esta fase se encarga de transformar los datos en valores adecuados (reemplazar datos faltantes mediante técnicas de imputación, o transformar datos mediante técnicas de discretización).

- **Minería de datos:** Esta fase se lleva a cabo a partir del desarrollo de modelos predictivos y descriptivos, y mediante el análisis de datos. Una vez recogidos los datos de interés el explorador decide qué tipo de patrón quiere descubrir y dependiendo del tipo de conocimiento que se desee extraer marcará claramente la técnica de minería de datos a utilizar.
- **Evaluación:** Para seleccionar y validar los modelos anteriores es necesaria la implementación de ésta fase, adicionalmente se utilizan herramientas estadísticas y de visualización.
- **Difusión:** Aquí se hace uso del nuevo conocimiento y se hace partícipe de él a todos los posibles usuarios.

1.1.2 Proceso de Extracción, Transformación y Carga de datos

El proceso ETL (*Extraction, Transformation and Loading*, Extracción, Transformación y Carga) permite realizar la integración de la información en una base de datos, almacén de datos o cualquier otro repositorio de información. Como su nombre lo dice, este proceso se encarga de extraer la información de diversas fuentes de datos, después de eso en el proceso de transformación los datos previamente extraídos pasan por varios procesos tales como: discretización de datos, limpieza de los datos, reemplazamiento de datos faltantes, eliminación de ruido, por mencionar algunos. Lo anterior con el fin de obtener datos confiables, En última instancia el proceso de carga almacena los datos ya transformados en un repositorio destino [2].

1.1.3 Sistema de recomendación

Los sistemas de recomendación son herramientas de software que proveen sugerencias acerca de artículos para el usuario, dichas sugerencias se ofrecen al usuario a la hora de tomar alguna decisión, por ejemplo: dónde planificar un recorrido, qué artículos se es posible adquirir, qué noticias interesantes hay para leer, por mencionar algunos [3]. El objetivo de un sistema de recomendación es desplegar recomendaciones significativas a los clientes respecto a artículos de interés personal [4].

Basado en cómo los sistemas analizan y filtran la información acorde a los requerimientos del cliente, existen varios tipos de sistemas de recomendación [3], descritos brevemente en las sub-secciones siguientes.

1.1.3.1 Basados en contenido

Son sistemas que se construyen acorde a las preferencias del usuario, basados en la descripción de artículos y el perfil de un cliente. Los artículos se describen usando palabras claves y el perfil del usuario en cuestión contiene todos los elementos que le agradan o prefiere. Este tipo de sistema utiliza técnicas de filtrado para la recuperación de información, en las cuales algunos elementos candidatos se combinan con elementos que el usuario calificó anteriormente, recomendando la mejor opción al usuario basado en el grado de emparejamiento obtenido [3], [5].

1.1.3.2 Filtrado colaborativo

Este tipo de sistema de recomendación depende de la reunión y análisis de información basada en las preferencias del usuario, y de hacer una predicción de los elementos que desearía en función de su similitud con otros usuarios, esta similitud se calcula comparando sus calificaciones con las de otros usuarios para el mismo artículo [3], [6].

Existen dos tipos de enfoques para el filtrado colaborativo:

- **Basado en el artículo:** Las relaciones de elementos se identifican usando una matriz de elementos de usuario y se usa para calcular recomendaciones para los usuarios indirectamente, proporcionando al usuario un elemento como recomendación, dependiendo de otros elementos con altas correlaciones. Este sistema predice las calificaciones del usuario para un artículo en particular según se otorgue una calificación a un artículo similar por el mismo usuario.
- **Basado en el usuario:** Recomienda un elemento en función de la opinión de otros usuarios con mentalidad similar para ese artículo. Este enfoque encuentra a los clientes con intereses comunes y los considera como los vecinos más cercanos, luego la predicción de un usuario para un artículo se

realizará con base en la calificación que los vecinos le hayan otorgado a ese elemento.

1.1.3.3 Híbridos

Estos sistemas combinan a los sistemas de recomendación basados en contenido con los sistemas de recomendación de filtrado colaborativo, dichos sistemas se implementan de tres formas: 1) Haciendo predicciones por separado y después combinándolas; 2) Añadiendo funciones de sistemas basados en contenido a un sistema colaborativo, y 3) Combinando los modelos en un único modelo [3], [7].

1.1.3.4 Consciente de contexto

El contexto es cualquier información que caracterice la situación de una entidad tal como una persona, lugar u objeto. Por ejemplo: los guías turísticos dependen en gran medida de la ubicación pero ignoran otros tipos de contexto, como resultado surge un problema de sobrecarga.

Estos sistemas toman en cuenta cinco tipos de información contextual tal como: ubicación, tiempo, clima, personalización y entretenimiento en redes sociales, para ofrecerle al usuario recomendaciones acorde a su situación [3].

1.1.3.5 Basados en tipología

El grado de tipicidad del usuario se utiliza para identificar a los vecinos más cercanos, primero agrupa los elementos en varios conglomerados, basado en cada conglomerado se forma un grupo de usuarios que tienen el mismo grado de tipicidad, después una matriz de tipicidad de usuario se crea para medir la similitud de los usuarios y luego seleccionar a sus vecinos, de acuerdo con las calificaciones de un artículo por parte de algún vecino, la calificación dada por el usuario para ese elemento se predice [3].

1.1.3.6 Basados en conocimiento

Este tipo de sistema recomienda productos basados en los requisitos del cliente y sus preferencias. Requiere un conocimiento funcional sobre el usuario para satisfacerlos mediante recomendación de un artículo en particular. Por lo tanto, existe un gran uso del conocimiento del dominio, dando como resultado que su

decisión es independiente del individuo, por lo que no es necesario recopilar información sobre un cliente en particular [3], [8].

1.1.4 Minería de datos

Han et al. [9] definen a la minería de datos como: el proceso de descubrimiento de patrones interesantes y conocimiento de grandes cantidades de datos. Las fuentes de datos incluyen a las bases de datos, almacenes de datos, la Web, otros repositorios de información, o datos que son transmitidos al sistema dinámicamente.

Para el descubrimiento de conocimiento la minería de datos se vale de diversas técnicas, a continuación se describen las más comunes.

1.1.4.1 Reglas de asociación

La esencia de este método es la extracción de asociaciones y correlaciones interesantes entre los conjuntos de datos en las bases de datos transaccionales u otro repositorio de información. Las reglas de asociación se usan extensivamente en varias áreas, una típica regla de asociación posee una implicación de la forma $A \rightarrow B$, donde A es un conjunto de artículos y B es un conjunto de artículos que contiene solamente una condición atómica [10].

1.1.4.2 Clustering

También llamado “Análisis de conglomerados”, constituye uno de los procedimientos estadísticos más utilizados hoy en día para la segmentación. La definición de ésta es: identificar grupos de sujetos lo más heterogéneos posible entre sí y lo más homogéneos posible dentro de cada grupo [11].

Define grupos tan distintos como sea posible en función de los propios datos sin especificación previa de los citados grupos, su finalidad esencial es revelar concentraciones en los datos para su agrupamiento eficiente en conglomerados según su homogeneidad, basándose en la proximidad o lejanía de unos con otros, por lo tanto, es fundamental el uso adecuado del concepto de distancia.

1.1.4.3 Predicción

Este método atrae demasiado la atención debido a las posibles consecuencias de un pronóstico acertado en algún contexto de negocio. Existen dos tipos de predicciones: la primera predice valores de datos no disponibles y en la segunda tan pronto como el modelo de clasificación se forma en un conjunto de entrenamiento, la etiqueta de clase del objeto se predice en función de los valores de los atributos del objeto. La predicción se refiere más a menudo a la previsión de valores numéricos faltantes [10].

1.1.5 Facturación electrónica (*E-billing*)

La Real Academia Española define “Facturación” como: La acción o efecto de facturar [12], y “Factura” se define como: Cuenta en que se detallan con su precio los artículos vendidos o los servicios realizados y que se entrega al cliente para exigir su pago [13].

Por lo tanto, “Facturación electrónica” se entiende como la acción o efecto de facturar por medio electrónico. Una definición formal es la que presenta el gobierno español: Una factura electrónica es, ante todo, una factura. Es decir, tiene los mismos efectos legales que una factura en papel. Una factura es un justificante de la entrega de bienes o la prestación de servicios, la cual se expide y recibe en formato electrónico. Es importante destacar que la expedición de una factura electrónica está condicionada al consentimiento de su destinatario. La factura electrónica, por tanto, es una alternativa legal a la factura tradicional en papel [14].

1.1.6 Comercio electrónico (*E-marketing*)

En [15], los autores definen al comercio electrónico como: cualquier tipo de operación comercial en la que la transacción se realiza mediante algún sistema de comunicación electrónico, por lo que no se requiere el “contacto físico” entre comprador y vendedor.

1.2 Planteamiento del problema

La empresa Solución Integral de Facturación Electrónica e Informática (SIFEI). S. A. de C. V. se dedica a la venta/renta de servicios de facturación electrónica (emisión, recepción y timbrado de comprobantes fiscales digitales) a clientes de distinta índole, a lo largo de su estancia en el mercado, la empresa se ha valido de campañas publicitarias para atraer más y mejores clientes a sus instalaciones, sin embargo, dichas campañas aún en la actualidad son publicadas para todos los clientes en general. Debido a lo anterior y aunado a la creciente competencia, SIFEI pierde clientes, gana a algunos y retiene a pocos.

Actualmente SIFEI almacena la información de sus clientes (antiguos y nuevos) en un repositorio de datos separado de la base de datos encargada de guardar los productos y servicios que ofertan (catálogo de productos). De igual forma, la información de los clientes no se use para la toma de decisiones en cuanto a campañas publicitarias, ofertas, entre otros. Provocando que el conocimiento potencial de los datos no se explote o ni siquiera se descubra.

El cambio de sistema gestor de base datos para la fusión de clientes y productos no es una opción, debido a la gran cantidad de información recabada a lo largo de los años que SIFEI se ha mantenido trabajando. El uso de técnicas de minería de datos para el descubrimiento de conocimiento y la implementación de sistemas que hagan uso de dichas técnicas es una buena opción, pero más allá de la implementación sobre un sistema que se esté desarrollando, es necesario tomar lo que ya se tiene hasta el momento y adecuarlo a las necesidades de la empresa.

1.3 Objetivo general y específicos

En las dos sub-secciones siguientes se detalla el objetivo general para el proyecto propuesto a desarrollar, así como sus objetivos específicos.

1.3.1 Objetivo general

Desarrollar un módulo que aplique técnicas de minería de datos para el análisis y detección de patrones de consumo para la generación de modelos que interpretados por personal capacitado ayude en la toma de decisiones y generación

de recomendaciones de servicios de facturación electrónica a partir del análisis de clientes.

1.3.2 Objetivos específicos

- Analizar técnicas de minería de datos para agrupamiento, clasificación y series de tiempo para determinar las óptimas para cada caso a través de un análisis comparativo.
- Analizar diferentes herramientas para el uso de técnicas de minería de datos para identificar aquellas que se adapten a la resolución del problema por medio de un análisis comparativo.
- Generar los modelos de agrupamiento, clasificación y series de tiempo para su uso mediante interfaces de usuario mediante la generación de pruebas, comparando los resultados obtenidos.
- Implementar las interfaces de usuario necesarias para la explotación de los modelos de minería de datos por medio de una aplicación *Web*.
- Realizar un caso de estudio para corroborar los resultados obtenidos.

1.4 Justificación

Debido la explotación limitada de la información recabada por parte de SIFEI y a la poca precisión de las campañas publicitarias que implementan para atraer nuevos clientes y retener a los ya existentes, este proyecto propuesto busca desarrollar un módulo que implemente técnicas de minería de datos, para solventar ese problema sin perjudicar la manera en que se trabaja en SIFEI, ya que al tratarse de un módulo, éste se integrará al sistema principal ayudando a la empresa Solución Integral de Facturación Electrónica e Informática (SIFEI) S. A. de C. V. a una mejor toma de decisiones respecto a campañas publicitarias y al entendimiento de las necesidades de sus clientes.

Con la utilización del módulo de recomendación, el usuario tendrá las herramientas necesarias mediante los resultados obtenidos, para tomar decisiones referentes a campañas publicitarias que impacten positivamente a las ganancias de la empresa hablando de dinero y de clientes satisfechos, esto último es muy importante y no sólo para SIFEI sino para toda empresa en general, ya que al tener clientes contentos, éstos recomiendan los servicios que tienen e incluso están abiertos a consumir servicios nuevos aumentando el valor de la empresa.

Debido al gran volumen de información con el que se cuenta en el sistema actual de SIFEI, no es posible realizar un tratamiento manual para el descubrimiento de conocimiento, la solución es valerse de un sistema automático que se encargue de ese procesamiento el cual será el módulo de recomendación propuesto. Con el desarrollo del módulo de recomendación, incorporando técnicas de minería de datos se obtendrán beneficios como la adquisición de conocimiento acerca de las preferencias y necesidades de los clientes de SIFEI.

Entendiendo las preferencias y necesidades de los clientes, SIFEI atacará los problemas que conllevaron a clientes antiguos dejar de trabajar con ellos, efectuar campañas publicitarias que atraigan a nuevos clientes y que los clientes ya existentes mejoren su experiencia utilizando los servicios de facturación electrónica que SIFEI provee.

Capítulo 2. Estado de la práctica

En este capítulo se dan a conocer algunos trabajos relacionados con el proyecto de tesis.

2.1 Trabajos relacionados

Los usuarios gastan casi la mitad de su tiempo buscando información relevante, los sistemas de recomendación tienen un papel importante en el filtrado de información, dichos sistemas lidian, con qué tan bien recomiendan artículos o información relevante al usuario. Las técnicas actuales de recomendación sufren por datos esparcidos y el problema del inicio frío, lo cual merma la habilidad para la recomendación de artículos al usuario, la combinación de dos técnicas da como resultado un sistema híbrido que supera en términos de precisión y eficiencia a los sistemas de recomendación simples. En [16] se realizó un estudio comparativo de varias técnicas para determinar cuál de ellas se acopla mejor en un dominio específico. Los sistemas de recomendación son una sub categoría de los sistemas de filtrado de información, los cuales recomiendan artículos basados en previas clasificaciones y predicciones realizadas por otros usuarios. Los diferentes tipos de técnicas de recomendación están clasificadas de la siguiente manera: Recomendaciones personalizadas (Filtrado basado en contenido, Filtrado colaborativo, Basado en conocimiento) y Recomendaciones no personalizadas (Enfoque agregado de opinión y Recomendador de asociación de producto). El estudio mostró mejores resultados en términos de precisión y desempeño al utilizar un enfoque híbrido.

Los sistemas de recomendación son herramientas de software que proveen sugerencias acerca de artículos de usuario, por lo anterior existen algunos métodos que manejan los desafíos provenientes de dichos sistemas, tales como: datos dispersos, escalabilidad, problema de cola larga, inicio frío, entre otros. Los sistemas de recomendación son útiles para los usuarios y los proveedores, y dependiendo de cómo los sistemas analizan y filtran la información acorde a los requerimientos del

usuario se clasifican en diferentes tipos de sistemas de recomendación, en [3] se describieron algunos, tales como:

- **Sistemas de recomendación basados en contenido:** están basados en la descripción del artículo y el perfil de usuario que se va construyendo acorde a las preferencias del mismo, este tipo de sistema de recomendación actualiza automáticamente el perfil del usuario por medio de la retroalimentación, y los resultados se despliegan emparejando las preferencias del usuario de manera **descendente**.
- **Sistema de recomendación de filtrado colaborativo:** depende de la reunión y análisis de la información basada en las preferencias del usuario haciendo una predicción de los artículos que le agradarán con base en la similitud con otros usuarios, dicha similitud se calcula comparando sus calificaciones con las de otro usuario para el mismo artículo.
- **Sistemas de recomendación híbridos:** estos sistemas combinan los dos anteriores, y son implementados en tres formas: 1) Haciendo predicciones por separado y después combinándolas; 2) Añadiendo funciones de sistemas basados en contenido a un sistema colaborativo, y 3) Combinando los modelos en un solo modelo.
- **Sistema de recomendación consciente de contexto:** un ejemplo es el turismo que mejoró gracias al contexto en aplicaciones móviles, estos sistemas toman en cuenta cinco tipos de información contextual tal como: ubicación, tiempo, clima, personalización y entretenimiento en redes sociales.
- **Sistema de recomendación basado en tipología:** el grado de tipología de un usuario en grupos de estos se usa para identificar vecinos de los usuarios, acorde a la calificación de un artículo por un vecino, la calificación dada por un usuario para ese artículo se pronostica.
- **Sistemas de recomendación basados en conocimiento:** basados en los requerimientos y preferencias del usuario, requiere de conocimiento funcional acerca del usuario para satisfacerlo recomendándole un artículo en particular,

estos sistemas almacenan información de un usuario dado de tal forma que el sistema es capaz de recomendar artículos complejos.

Como conclusión, en [3] los autores determinaron que los distintos tipos de sistemas de recomendación se implementan acorde a las preferencias y necesidades del usuario para incrementar las ventas, y que las diferentes personalidades de los clientes requieren de distintos tipos de sistemas de recomendación para evitar la sobrecarga de información.

El enorme volumen de información tanto estructurada como no estructurada, hoy en día provee de oportunidades para diversas compañías, especialmente a aquellas que usan el comercio electrónico. Los datos se almacenan a partir de los procesos internos de los consumidores, vendedores, mercados y ambientes de negocios. En [10] se revisó la aplicabilidad de la minería de datos en el comercio electrónico, enfocándose en información estructurada y no estructurada recolectada de varias fuentes y servicios de cómputo en la nube, para justificar la importancia de la minería de datos en la toma de decisiones. Ismail et al. Describieron que la minería de datos no es específica a un único tipo de información, sin embargo, los algoritmos y técnicas varían cuando se aplican a diferentes tipos de datos, por último, la minería de datos se usa en diferentes tipos de bases de datos tales como: archivos planos, almacenes de datos, orientadas a objetos, entre otras. El objetivo primordial de la minería de datos es obtener conocimiento a partir de los datos a analizar, para ello, existen demasiadas técnicas de minería de datos, las más comunes son:

- **Reglas de asociación:** la esencia de este método es la extracción de correlaciones interesantes y asociaciones entre conjuntos de datos en las bases de datos transaccionales u otro tipo de repositorio de información.
- **Clustering (Agrupamiento):** es el proceso de agrupar un conjunto de objetos físicos o abstractos en clases de objetos similares.

- **Predicción:** atrae atención sustancial dada las posibles consecuencias de un pronóstico satisfactorio en un contexto de negocios, ya que la predicción es comúnmente referenciada a la determinación de valores numéricos faltantes.

La minería de datos para las compañías de comercio electrónico ya no es más un privilegio, sino un requerimiento para sobrevivir y mantenerse dentro del ambiente competitivo, ya que les permite desarrollar planes mercantiles, análisis de consumidores, comportamiento de ventas, entre otros. Para aquellas compañías que almacenan la información (preferencias y requerimientos) de sus clientes, la minería de datos juega un papel importante en proveer servicios orientados al consumidor para incrementar la satisfacción de los clientes y por lo tanto las ventas de la compañía.

Los recientes avances en las tecnologías de comunicación e información mejoraron drásticamente la velocidad computacional, bajo esas circunstancias los investigadores dirigen sus estudios enfocados en grandes cantidades de información almacenada en diversas fuentes. Las técnicas de minería de datos son importantes para el análisis del control de datos y su modelación. Varios métodos relacionados con la minería de datos se desarrollaron y software como Weka permite fácilmente el uso de dichos métodos, sin embargo, para estas aplicaciones generalmente se necesita seleccionar una técnica apropiada para los datos. Ishibashi et al. [17] verificaron la efectividad de la creación de un modelo predictor de ganancias basado en información de estados financieros. Los estados financieros contienen demasiadas variables, por ello su selección y la de un modelo se espera sea adecuado para la construcción de un modelo predictivo, en [17] se enfocaron en la selección de variables como primera etapa. En el estudio, la selección de variables con técnicas de minería de datos dio como resultado que el método para la selección de variables mejora la precisión de la predicción comparado con un modelo que tome en cuenta todas las variables, sin embargo, hay diferencias en el efecto de cada método, dependiendo de los conjuntos de datos que tomen como base.

Un *Call Data Record* (Registro de datos de llamada o CDR por sus siglas en inglés) es una estructura de información que almacena datos relevantes acerca de una actividad telefónica dada y que involucra a un usuario de una red de telefonía. Un CDR comúnmente contiene datos espaciales y temporales, pero también almacenan otro tipo de información útil. Anteriormente los censos se usaban para conocer el movimiento demográfico y geográfico de la población, hoy en día debido a la movilidad a corto plazo de las personas, se utilizan métodos más flexibles como varios registros y bases de datos indirectas, los CDR representan a un buen candidato en ese sentido, ya que una de sus principales ventajas es que ofrecen una representación estadísticamente precisa de la distribución de personas en un área y con esto se rastrean grandes grupos de personas heterogéneas, de tal forma que los CDR evolucionan acorde a los cambios en el comportamiento de los usuarios. En [18] propusieron dos implementaciones de un procedimiento de minería de datos: *Clustering* (agrupamiento), ya que los resultados en trabajos previos sirvieron para entender los hábitos de usuarios y de esa forma realizar el trazado de sus perfiles. La primera está basada en un novedoso sistema para el agrupamiento y el proceso de generación de conocimiento llamado LD-ABCD (*Local Dissimilarities - Agent Based Clusters Discoverer*, Disimilitudes Locales – Descubridor de Agrupamientos Basados en Agentes) capaz de recuperar grupos y al mismo tiempo descubrir automáticamente para cada uno la medida de desemejanza más apropiada. La segunda implementación se basa en PROCLUS (*PROjected CLUStering*, Agrupamiento Proyectado), el bien conocido algoritmo de sub-agrupamiento, el conjunto de datos que estuvo bajo análisis contenía información caracterizada por solamente unas pocas variables. Como resultado en [18] los autores demostraron que usando LD-ABCD los resultados son estables y confiables, incluso cuando la naturaleza de los datos es completamente desconocida y la presencia de grupos bien definidos no es clara, de esta manera se obtienen grupos de usuarios y por lo tanto sus hábitos, sin embargo, aunque es fácil

la implementación de LD-ABCD en una red de computación distribuida debido a la arquitectura multi agente, el sistema necesita grandes recursos computacionales.

El manejo de crédito es la base de la industria financiera, cuando los agentes proveen de préstamos a individuos o compañías, evalúan el riesgo crediticio de sus prestatarios para reducir la posibilidad de alguna deuda incobrable y decidir el monto del préstamo. Los prestamistas almacenan información acerca del individuo al que se le otorgó dicho bien, el intercambio de información entre prestamistas ocurre voluntariamente mediante el servicio de crédito privado o se fuerza por regulación a través de oficinas de crédito. Este intercambio de datos reduce significativamente el costo del mercado crediticio y lo hace más eficiente, en países desarrollados el sistema de crédito experimentó más de cien años de evolución, en China el sistema de crédito público está muy por detrás de su crecimiento económico, lo que provoca deficiencias en el mercado de préstamos y la necesidad de servicios crediticios llevó a la creación de servicios de crédito privado. Hoy en día muchos proveedores de servicios de crédito privado hacen uso de información almacenada en internet y otras fuentes de datos, sus prácticas proveen un buen ejemplo del uso de la tecnología de Big Data para su aplicación en la industria financiera. Zhang et al. [19] examinaron los servicios de crédito por internet a través de las prácticas de 100Credit (www.100credit.com), un nuevo proveedor de servicio de crédito en China, con la meta de profundizar el entendimiento del valor de Big Data en estos tipos de servicios. En [19] inspeccionaron las operaciones de negocios de 100Credit, la cual usa colecciones de datos heterogéneos de diversas fuentes enfocadas en actividades de usuarios tanto en línea como fuera de ella para predecir su rango de crédito, ellos resumieron cuatro valores que Big Data aportó a este tipo de compañías: 1) El valor de volumen; 2) Corrección de la predicción; 3) Variedad de servicios, y 4) Protección de la privacidad.

En campañas publicitarias, Long et al. [20] observaron que existe un intercambio comercial entre la cantidad de clientes satisfechos por las compañías y la ganancia obtenida. Simplemente satisfacer tantos clientes como sea posible o maximizar el beneficio no es deseable. Para tal fin, los autores propusieron un nuevo problema llamado *K-Satisfiability Assigment for Maximizing the Profit* (Asignación de Satisfacción para Maximizar los Beneficios, k-SAMP por sus siglas en inglés), en donde k es un parámetro de usuario, entero no negativo. Dado un conjunto P de productos y un conjunto O de consumidores, k-SAMP busca una asignación entre P y O tal que al menos k clientes sean satisfechos en dicha asignación y que la ganancia incurrida por este medio sea maximizada. El trabajo realizado por Long et al. Constó en diseñar un novedoso algoritmo llamado "Ajuste para el problema k-SAMP", dada una asignación A, el ajuste incrementa iterativamente la ganancia de A, ajustando algunas coincidencias apropiadas en A manteniendo al menos k clientes satisfechos en A. Se realizaron varios experimentos los cuales verificaron la eficiencia del algoritmo propuesto por [20], como trabajos a futuro los autores mencionan abordar dos direcciones: la primera es estudiar el problema de maximizar la ganancia (promedio) que se obtenga de un solo cliente satisfecho, y la segunda es estudiar el problema k-SAMP con una restricción adicional en donde cada usuario será emparejado a lo mucho a un tipo de producto.

El objetivo de un sistema de recomendación es desplegar recomendaciones significativas a los clientes respecto a artículos de interés personal, ya que actualmente se vive la era de la explosión de información debido al rápido desarrollo de la tecnología de la Internet. La vasta cantidad de datos no solamente provee estadísticas, sino también información y conocimiento, por ello los sistemas de recomendación proveen datos basados en las preferencias de los clientes y que anteriormente estaban escondidas en bases de datos, llamando atención sustancial. Este conocimiento escondido lo usan los sistemas de ventas para realizar recomendaciones a los clientes y promover ventas. Sin embargo, las demandas de los usuarios son dinámicas y los

productos que anteriormente eran preferidos después de un cierto periodo ya no lo serán, mientras que nuevas demandas se desarrollan gradualmente. Por lo tanto, la calidad de las recomendaciones se mejora usando la información dinámica secuencial en un enfoque continuo evolutivo. En [4] los autores implementaron la programación de red genética (GNP por sus siglas en inglés) y la optimización de colonia de hormigas (ACO por sus siglas en inglés) para resolver el problema de la extracción de reglas secuenciales para recomendaciones comerciales en bases de datos transaccionales relacionadas con el tiempo. Debido a la naturaleza cambiante de las preferencias de los clientes y a las diferencias con el enfoque tradicional “encuentra todo y después filtra”, las reglas interesantes de asociación temporal son extraídas por las metaheurísticas y los métodos genéticos basados en algoritmos de GNP. Adicionalmente un modelo útil se construye usando las reglas obtenidas para pronosticar las necesidades futuras de los clientes y un enfoque ACO para mejorar continuamente el sistema de recomendación en línea. Como resultado Zhou y Hirasawa [4] verificaron que los métodos propuestos sirven para el análisis del comportamiento del cliente, y permiten predecir con precisión algún cambio en el comportamiento del usuario construyendo el modelo de análisis propuesto. De tal forma que la precisión del mercado se ejecutará en un momento adecuado para evitar promociones ineficientes.

Bajo una fuerte competencia global, los gerentes aprovechan cualquier oportunidad que les deje como resultado grandes ganancias dentro de un margen de tiempo, con ese objetivo, se obtiene información de las operaciones diarias y se guarda dentro de almacenes de datos para el propósito de la *Customer Relationship Management* (gestión de relaciones con los clientes o CRM por sus siglas en inglés), en donde la información se transforma en conocimiento útil, el cual minimiza riesgos gerenciales e incrementa la efectividad de las estrategias CRM. Estos métodos de minería de datos mantienen la identificación de las tendencias significativas y las relaciones dentro de los datos, la identificación de los clientes más rentables y la segmentación de éstos

dependen de las variables almacenadas en los conjuntos de datos. En [21] los autores realizaron un estudio centrado en perfilar clientes hoteleros rentables mediante el análisis RFM (*Recency, Frequency and Monetary*, Frescura, Frecuencia y Monetario, las cuales son las variables por las que la información de un cliente se clasifica) que es un método de minería de datos basado en comportamiento, este análisis extrae el perfil del cliente usando pocos números de criterios reduciendo la complejidad de dicho análisis. Dursun y Caber [21] utilizaron la información CRM perteneciente a tres hoteles de cinco estrellas que operan en Antalya, Turquía. Los resultados mostraron que la mayoría de los propietarios de tarjetas de membresía son de mediana edad (35-44 años) 35.2%, parejas (37.7%) con hijos (48.2%). Siendo esta una típica situación para un hotel que atrae turistas quienes comúnmente viajan en vacaciones. De igual forma el análisis RFM arrojó 369 perfiles de clientes hoteleros divididos en ocho grupos: "Clientes leales", "Clientes leales de verano", "Clientes de compra colectiva", "Clientes de invierno", "Clientes perdidos", "Clientes con alto potencial", "Clientes nuevos" y "Clientes con alto potencial en invierno". En donde los "clientes perdidos" son el 36% y se refiere a las personas que se quedan poco tiempo. Como conclusión Dursun y Caber [21] demostraron que el análisis RFM agrupa eficientemente a los clientes, llevando a los altos directivos de los hoteles a generar nuevas estrategias para incrementar sus habilidades en CRM.

Desde la creación de la Internet la cantidad de información aumentó rápidamente impulsando la búsqueda de recuperación de información, actualmente muchos sitios y servicios incorporan sistemas de recomendación (SR) para asistir a los usuarios en la tarea de filtrar contenido relevante o de su interés, para alcanzar los criterios de los clientes los SR comúnmente construyen modelos a partir de las preferencias extraídas en un único dominio, por ejemplo, para sugerir libros el SR examina a aquellos que el usuario ya leyó o buscó. Rodrigues et al. [22] propusieron un método para la modelación de perfiles de usuario que combina información de múltiples dominios para mejorar la precisión de las recomendaciones de un SR, dicho método permite el cruce

de metadatos de artículos en un dominio (ejemplo libros) para descubrir intereses de usuario en otros dominios (por ejemplo películas), las preferencias utilizadas se extrajeron de Facebook®. Utilizando el algoritmo BPR-Mapping basado en el marco de trabajo BPR (*Business Process Reengineering*, Reingeniería de Procesos de Negocio), este algoritmo sirve para personalizar la calificación de artículos utilizando apenas una retroalimentación implícita. Como resultado en [22] los autores lograron demostrar que en todos los experimentos realizados, el combinar los atributos de las películas con los atributos de los libros produjo mejores resultados de recomendación que al utilizar sólo atributos de películas. El enfoque de múltiples dominios en el modelado de perfiles de usuario también se mostró conveniente para lidiar con el problema de arranque en frío (uno de algunos problemas en los sistemas de recomendación), el conocimiento sobre los nuevos usuarios no es suficiente para empezar a generar recomendaciones, ya que el modelo de usuario se crea automáticamente.

Dada la importancia de los consumidores como los bienes más valiosos para las organizaciones, la retención de éstos es un requerimiento básico y esencial para cada empresa, los bancos no son la excepción a esta regla. La competitiva atmósfera dentro de la cual, los servicios de banca electrónica son provistos por diferentes bancos, aumenta la necesidad de la retención de clientes, ya que muchas organizaciones competitivas determinaron que esta técnica es una estrategia clave para sobrevivir dentro del mercado. Pero mantener a los clientes no es una tarea fácil, debido a la rotación de cliente, la cual es uno de los problemas básicos con el que lidia la industria de la banca. Debido a lo anterior, Keramati et al. [23] desarrollaron un modelo predictivo para identificar características comunes de la rotación de clientes. Usaron la metodología CRISP-DM (*Cross Industry Standard Process for Data Mining*, Proceso Estándar de la Industria Cruzada para la Minería de Datos) para la creación de su modelo, en esa metodología se usó un enfoque de seis fases:

- **Entendimiento del negocio.**
- **Entendimiento de los datos.**
- **Pre-procesamiento de los datos.**
- **Modelado.**
- **Evaluación.**
- **Despliegue.**

Del mismo modo en [23] usaron el método de árboles de decisión (DT por sus siglas en inglés) en la fase de modelado. El propósito del estudio era identificar las características de los *churners* (clientes que abandonan una compañía para ir a otra) de los servicios de banca electrónica, variables demográficas (edad, género, carrera, y nivel de educación), datos de transacciones a través de portales bancarios electrónicos (ATM, banca móvil, banca telefónica, banca por internet, entre otros), la longitud de la asociación de clientes y las quejas de éstos, se extrajeron de la base de datos del banco. Keramati et al. [23] definieron que el uso de técnicas de minería de datos para la predicción de rotación de clientes es el nuevo contexto de banca electrónica. La recolección de datos y la selección de características para predecir el abandono de los clientes en este contexto, es uno de los aspectos novedosos de su investigación.

Hoy en día a los sistemas de recomendación (SR) se les encuentra en cualquier esquina de una página Web o en recomendaciones de productos en tiendas en línea, debido a su habilidad para resolver un severo problema que se incrementa: la sobrecarga de información. Como resultado, los SR atraen numerosos investigadores de los campos de aprendizaje automático y minería de datos. Sin embargo, mientras que demasiados algoritmos se crearon y se implementaron para sugerir artículos relevantes al usuario, aún hay muchos retos que necesitan completarse antes de que los sistemas de recomendación logren alcanzar su potencial máximo. Muchos algoritmos sufren del problema de inicio frío o que los SR no logran dar recomendaciones efectivas a los nuevos usuarios o referenciar a los nuevos artículos,

ya que aún no tienen indicadores de relevancia explícitos o implícitos. Por lo anterior, en [24] los autores presentaron un marco de trabajo (Framework) de visualización interactiva, que va más allá de la precisión de los algoritmos de recomendación, enfocándose en varios factores humanos que afectan la aceptación de recomendaciones como la satisfacción del usuario, veracidad, transparencia y sentido del control. El marco de trabajo combina la recomendación con técnicas de visualización para apoyar la interacción humano-recomendador. He et al. [24] propusieron una estrecha integración de técnicas de visualización y recomendación para permitir a los usuarios finales interactuar con sistemas de recomendación y crear un ciclo de retroalimentación. En [24] los autores presentaron un análisis de 24 sistemas de recomendación interactiva existentes a lo largo de las dimensiones del marco de trabajo. Como resultado, el análisis mostró que la mayoría del trabajo existente se centra en la transparencia y el control del proceso de recomendación, y que mediante el uso de técnicas de visualización, la comprensión del usuario de la lógica de los sistemas de recomendación permite ser respaldada. Dicho conocimiento mejora la aceptación de las recomendaciones, de igual forma, al apoyar el control del usuario se tiene un impacto en la precisión de las recomendaciones.

En la era actual de los dispositivos inteligentes y la Web 2.0, gran cantidad de información es generada a partir de una gran variedad de fuentes como: sitios de redes sociales, sitios para compartir contenido multimedia, servicios alojados, correo electrónico, grupos de comunicaciones, mensajes instantáneos, entre otros. Debido a esa gran cantidad de datos, se vuelve cada vez más difícil para los usuarios encontrar contenido relevante en la Web y en muchas ocasiones esta sobrecarga de información ocasiona que los usuarios respondan al problema omitiendo algo importante o cometiendo un error en el proceso de tomar una decisión. Paradarami et al. [25] presentaron un framework (marco de trabajo) de red neuronal profunda que utiliza opiniones en adición a características basadas en contenido para generar un modelo basado en predicciones para las combinaciones del usuario de negocios. En [25] los

autores desarrollaron una nueva técnica de sistema de recomendación híbrido, la cual se basa en las capacidades proporcionadas por los enfoques tradicionales como filtrado colaborativo y filtrado basado en contenido, mediante el uso de metadatos asociados con el texto a revisar, para entrenar y construir una red neuronal artificial (ANN por sus siglas en inglés). Como resultado Paradarami et al. [25] demostraron que debido a las capacidades de aprendizaje efectivas de un marco de trabajo de ANN, y a sus características basadas en contenido y en el filtrado colaborativo, la efectividad del sistema de recomendación mejora significativamente. También demostraron que una ANN entrenada para una categoría de negocios en particular se generaliza bien y actúa como el sistema de recomendación para otras categorías, lo que mejora la escalabilidad de la aplicación.

Google® ejecuta una extensa plataforma de publicidad a través de múltiples canales que abastece billones de anuncios cada día a todos los usuarios alrededor del planeta. Información detallada asociada con cada anuncio mostrado, como: criterio de segmentación, número de impresiones, números de clicks, entre otros, es almacenada y procesada en tiempo real, dicha información es extensivamente usada por Google® para diferentes casos de uso, los cuales incluyen: informes, auditoría interna, análisis, facturación y pronóstico. La escala y naturaleza crítica del negocio de esos datos dan como resultado desafíos técnicos y operativos únicos para procesar, almacenar y consultar. Mesa es la solución de Google® para estos desafíos técnicos y operativos para la información crítica de la empresa, Mesa es un sistema distribuido, replicado y altamente disponible para procesamiento, almacenamiento y consulta de datos estructurados. En [26] los autores presentaron una descripción del sistema Mesa de Google® e informaron el rendimiento y la escala que Mesa alcanza. Como resultado, Gupta et al. [26] presentaron el diseño y la implementación del sistema de almacenamiento de datos escalable, casi en tiempo real y geo-replicado llamado Mesa, su diseño de ingeniería aprovecha ideas de investigación fundamentales en las áreas de bases de datos y sistemas distribuidos. Particularmente Mesa soporta

consultas en línea y actualizaciones, al mismo tiempo que provee fuerte consistencia y garantías de corrección transaccional, logra estas propiedades utilizando una interfaz orientada a lotes, que garantiza la atomicidad de las actualizaciones mediante la introducción de versiones transitorias de los datos, esto elimina la necesidad de sincronizar las consultas y actualizaciones basadas en bloqueos. Al ser geo-replicado a través de múltiples centros de datos, Mesa es altamente tolerante a fallas. Finalmente, dentro de cada centro de datos el marco de trabajador/control le permite distribuir el trabajo y escalar dinámicamente el cálculo requerido en un gran número de máquinas para proporcionar alta escalabilidad.

Los sistemas de recomendación (SR por sus siglas en inglés) se consideran herramientas de gran utilidad, principalmente para usuarios de sistemas en línea que se encuentran con un volumen de información mayor al que soportan de manera adecuada. La industria del software para la Web considera a los SR como un medio para atender mejor las demandas de los clientes y hacer a sus sistemas más rentables, en este contexto el procesamiento de textos (libros, revistas, artículos científicos, por mencionar algunos) representa una demanda importante a esos tipos de datos, los cuales frecuentemente son objeto de procesamiento de los sistemas de recomendación basados en contenido. El aprendizaje automático (*Machine Learning*, ML por sus siglas en inglés) se caracteriza por el desarrollo de técnicas cuyo objetivo es proveer software con la habilidad de mejorar su desempeño en una tarea aprendiendo a través de la experiencia (aprendizaje inductivo). Diferentes técnicas de ML o combinaciones de ellas se utilizan para el análisis de datos textuales en diversos contextos, el éxito obtenido al utilizar dichas técnicas demuestra que son óptimas para el procesamiento de datos textuales y, por lo tanto, son adecuadas para la construcción de sistemas de recomendación que se encargan de este tipo de información. En [27] los autores presentaron una revisión sistemática para identificar, interpretar y evaluar cómo se aplican las estrategias del aprendizaje automático, es decir, construir un panorama sobre las investigaciones más recientes en el área de los

SR basados en contenido textual construidos con base en técnicas de ML. Brunialti et al. [27] observaron que las iniciativas usando ML en SR basados en contenido textual son pocas, sin embargo, en el análisis de los estudios relacionados se percibió que las iniciativas en la intersección de estas áreas producen resultados prometedores, proporcionando una motivación para que más esfuerzos de investigación se implementen en esta área, en ese contexto, se observó una carencia significativa de propuestas dirigidas a estudiar la educación del aprendizaje inductivo no supervisado en la tarea de aprendizaje del perfil del usuario.

2.2 Análisis comparativo

La Tabla 2.1 muestra de manera abstracta la información relevante de cada artículo mencionado en la sección anterior, con el fin de realizar un análisis comparativo.

Tabla 2.1 Análisis comparativo de los artículos relacionados.

Artículo	Objetivo	Problema	Tecnologías usadas	Resultados	Estado
Pereira y Varma [16]	Realizar un caso de estudio en donde se compararon diferentes técnicas de recomendación para su combinación, y la demostración de que los sistemas híbridos funcionan mejor que los sistemas simples.	Los usuarios pasan demasiado tiempo buscando información que necesitan, y en las aplicaciones dónde buscan hacen uso de técnicas de recomendación simples, las cuales no brindan información relevante o demoran para mostrar dicha información.	Técnicas de recomendación.	La combinación de las técnicas basadas en contenido y filtrado colaborativo obtiene mejores resultados.	Finalizado.

Artículo	Objetivo	Problema	Tecnologías usadas	Resultados	Estado
Gigimol y John [3]	Establecer las diferencias de algunos sistemas de recomendación para determinar cuál usar en casos específicos.	Sobrecarga de información al utilizar un sistema de recomendación erróneo para cierto tipo de clientes y proveedores.	Ninguna/No mencionada.	Explicación de los diferentes tipos de sistemas de recomendación para seleccionar alguno según se requiera con base en las necesidades del cliente.	Finalizado.
Ismail et al. [10]	Estudiar la aplicabilidad de la minería de datos en el comercio electrónico.	Las compañías que usan el comercio electrónico no utilizan la información de sus clientes de manera correcta para generar estrategias que permitan el crecimiento y mejora de sus técnicas mercantiles.	Ninguna.	Se determinó que el uso de técnicas de minería de datos en diferentes dominios obtiene información clave para la generación de	Finalizado.

Artículo	Objetivo	Problema	Tecnologías usadas	Resultados	Estado
				estrategias mercantiles.	
Ishibashi et al. [17]	Verificar la efectividad de la creación de un modelo predictor de ganancias utilizando técnicas de minería de datos para la selección de variables.	¿Cuál técnica de minería de datos es la adecuada para la selección de variables?	C4.5 para la predicción categórica.	Si se usan las técnicas de minería de datos para la selección de variables, los resultados son favorables en comparación con modelos que utilizan todas las variables.	Mejoras a futuro.

Artículo	Objetivo	Problema	Tecnologías usadas	Resultados	Estado
Bianchi et al. [18]	Utilizar la información almacenada en CDRs para comparar dos variantes de la técnica de minería de datos: <i>Clustering</i> .	¿Se obtienen hábitos de usuario mediante el uso de minería de datos en CDRs?	LD-ABCD. PROCLUS.	LD-ABCD obtuvo mejores resultados, sin embargo, presenta deficiencias.	Trabajo a futuro.
Zhang et al. [19]	Profundizar el entendimiento del valor de Big Data en servicios de crédito por internet.	En China el crecimiento del sistema de crédito público no está a la par del crecimiento económico del país.	Ninguna.	El uso de Big Data es de vital importancia para este tipo de empresas para evitar problemas financieros a futuro.	Trabajo a futuro.

Artículo	Objetivo	Problema	Tecnologías usadas	Resultados	Estado
Long et al. [20]	Probar que el ajuste al problema k-SAMP regresa un optimizador global de ganancias.	Las compañías no mantienen un equilibrio entre las ganancias obtenidas y los clientes satisfechos.	Ajuste del problema k-SAMP.	El ajuste obtuvo buenos resultados.	Trabajo a futuro.
Zhou [4]	Implementar la GNP y la ACO para ayudar en la predicción del cambio de comportamiento de los clientes.	Predecir las nuevas preferencias de los clientes es difícil para los sistemas de recomendación.	GNP. ACO.	La combinación de la GNP y la ACO, aunadas a intervalos de tiempo para la extracción de reglas de asociación, predicen los cambios de preferencias de los clientes	Trabajo a futuro.
Dursun y Caber [21]	Perfilar clientes mediante el análisis RFM.	El uso de CRM no es suficiente para segmentar clientes hoteleros con	Análisis RFM.	El análisis RFM encontró 369 perfiles	Finalizado.

Artículo	Objetivo	Problema	Tecnologías usadas	Resultados	Estado
		base en sus preferencias y necesidades.		de cliente y los agrupó en ocho grupos.	
Rodrigues et al. [22]	Utilizar un enfoque de múltiples dominios para la modelación de perfiles de usuario.	Utilizar un solo enfoque no asegura una recomendación de calidad al usuario.	Algoritmo BPR-Mapping.	Mejores resultados utilizando un enfoque de múltiples dominios.	Trabajo a futuro.
Keramati et al. [23]	Desarrollar un modelo predictivo para la rotación de clientes.	Los servicios de banca electrónica no logran retener a sus clientes, perdiendo ganancias.	Árboles de decisión.	El uso de árboles de decisión logró el desarrollo del modelo predictivo, sin embargo, es necesario mejorar la selección de variables.	Trabajo a futuro.
He et al. [24]	Analizar algunos SR tradicionales en contra de un marco de trabajo de	Los SR tradicionales no se centran en la aceptación del usuario	Framework que combina	Darle al usuario la oportunidad de entender la lógica de los SR incrementa la	Trabajo a futuro.

Artículo	Objetivo	Problema	Tecnologías usadas	Resultados	Estado
	visualización interactiva.	respecto al artículo que se le recomienda.	técnicas de visualización.	aceptación del artículo sugerido.	
Paradaram i et al. [25]	Desarrollar un marco de trabajo de una ANN para mejorar la efectividad de los SR.	La sobrecarga de información afecta los resultados efectuados por los SR.	Red neuronal artificial.	El uso de un enfoque híbrido para la generación de una red neuronal artificial en un negocio en particular mejora la efectividad de un SR.	Trabajo a futuro.
Gupta et al. [26]	Informar acerca de las mejoras al utilizar el sistema Mesa de Google® para la resolución de problemas típicos de cualquier base de datos.	Manejo de grandes cantidades de información para análisis en sistemas de tiempo real, entre otros problemas comunes.	Sistema Mesa de Google® (almacén de datos).	El uso de un sistema geo-replicado, con una interfaz por lotes y un marco de trabajo distribuido mejora los problemas típicos de cualquier sistema en	Finalizado.

Artículo	Objetivo	Problema	Tecnologías usadas	Resultados	Estado
				tiempo real y/o base de datos.	
Brunialti et al. [27]	Realizar una revisión sistemática para identificar, evaluar e interpretar técnicas de ML en SR basados en contenido textual.	La no explotación de la combinación de técnicas de ML en SR basados en contenido textual.	Ninguna.	Pocas iniciativas sobre la incorporación de dichas áreas, sin embargo, tales iniciativas producen resultados prometedores, para motivar la investigación de este tema.	Finalizado.

Después de analizar los artículos citados, se concluye que el desarrollo de sistemas de recomendación es altamente eficaz en cualquier dominio que tenga como objetivo atraer la atención de clientes potenciales, así como de retener a los ya existentes. De igual forma, los RS son de vital importancia en la toma de decisiones por parte de los usuarios, ya que les brindan información relevante de acuerdo a sus necesidades, ahorrándoles tiempo y mejorando su experiencia para con la empresa.

De igual forma, las técnicas de minería de datos son esenciales para el descubrimiento de conocimiento, pues la información que se almacena y no se explota no es más que simples datos, sin embargo, al utilizar técnicas de minería de datos, esos simples datos se convierten en conocimiento útil para cualquier empresa que busque mejorar sus servicios, aumentar sus clientes, obtener mayores ganancias y/o no perder clientes mediante estrategias mercantiles. Por lo anterior, se considera que el trabajo en conjunto entre los sistemas de recomendación y técnicas de minería de datos es un área de interés con diversas propuestas. Debido a lo anterior la implementación de técnicas de minería de datos para la generación de modelos que ayuden a SIFEI para la toma de decisiones publicitarias y mercantiles resulta ser una opción viable para la resolución de la problemática planteada en el capítulo anterior y que el módulo propuesto en este trabajo de tesis se acopla perfectamente para darle solución al problema antes mencionado.

2.3 Análisis comparativo algoritmo de series de tiempo

Una de las grandes incógnitas a las que se enfrenta toda empresa es el futuro, dado que es incierto, sin embargo existen medidas (matemáticamente hablando) para conocer comportamientos en los datos. Las series de tiempo precisamente buscan encontrar un resultado esperado con base en la actividad o comportamiento de la información, es decir, se pronostica un monto (probable) de una compra en una fecha en el futuro.

Debido a la naturaleza del problema (el futuro), a continuación se presenta el análisis comparativo de dos algoritmos para la implementación de series de tiempo, para determinar el óptimo a utilizar en el sistema propuesto en este trabajo de tesis. Los algoritmos son:

- **Regresión lineal.**
- ***Multilayer perceptron.***

2.3.1 Estado del arte de series de tiempo

Aguirre et al.[28] presentaron los resultados obtenidos utilizando tres modelos de predicción para pronosticar los niveles de ozono y dióxido de nitrógeno en tiempo de real con hasta 8 horas de antelación en cuatro estaciones de Bilbao (España). Fueron dos modelos basados en *Multilayer perceptron* y uno basado en regresión lineal múltiple. Se utilizaron variables de tráfico, variables meteorológicas y niveles por hora de nitrógeno y dióxido de nitrógeno como datos de entrada que se midieron desde 1993 hasta 1994. El desempeño de esos tres modelos se comparó con la persistencia de los niveles y los valores observados, los resultados indicaron un mejor desempeño para los modelos basados en *Multilayer perceptron* sobre el modelo de regresión múltiple, adicionalmente se comprobó que la inserción de cuatro variables de temporada adicionales en el modelo basado en *Multilayer Perceptron* aumenta la precisión de la predicción.

Cankurt et al.[29] realizaron métodos estadísticos y de aprendizaje automático para estimar la llegada de turistas con el fin de mejorar el fenómeno llamado turismo sostenible. En el estudio se consideraron la implementación de la regresión lineal y de *Multilayer Perceptron* para la creación de un pronóstico multi-variable de turismo en Turquía. Como resultado obtuvieron que el método de *Multilayer Perceptron* encuentra la mejor precisión para su pronóstico, sin embargo los autores mencionan que en ciertas ocasiones la regresión lineal obtiene mejores resultados que *Multilayer Perceptron*, y aunque se trata de casos en específico realizarán más pruebas para determinar el fenómeno.

Nassif et al.[30] realizaron un novedoso modelo basado en regresión lineal logarítmica basado en el modelo de casos de uso para calcular el esfuerzo de

software basado en diagramas de casos de uso, de igual forma utilizaron un enfoque de lógica difusa para calibrar el factor de productividad en el modelo de regresión, de igual forma desarrollaron un modelo *Multilayer Perceptron* para predecir el esfuerzo del software en función del tamaño del mismo y de la productividad del equipo. Al final los autores realizaron una comparación entre *Multilayer Perceptron* y la regresión lineal logarítmica tomando como pruebas diferentes tamaños de proyectos, demostrando que el modelo *Multilayer Perceptron* supera al modelo de regresión en proyectos pequeños, pero el modelo de regresión obtiene mejores resultados cuando de proyectos más grandes se trata.

2.3.2 Evaluación de los algoritmos para series de tiempo

La Tabla 2.2 muestra el algoritmo que mejor se desempeñó en los artículos descritos anteriormente.

Tabla 2.2 Comparativa algoritmos series de tiempo.

ARTÍCULO	REGRESIÓN LINEAL	MULTILAYER PERCEPTRON
AGUIRRE ET AL.[28]		✓
CANKURT ET AL.[29]	✓	✓
NASSIF ET AL.[30]	✓	✓

Como se observa en el primer artículo *Multilayer Perceptron* obtuvo el mejor resultado, sin embargo los siguientes artículos aunque ésta tuviera el mejor desempeño, los autores no descartan la implementación de la regresión lineal dado que es mejor en algunos casos utilizarla.

Por lo anterior se determina que el algoritmo *Multilayer Perceptron* es el indicado, esto con base en el análisis anterior, aunque de igual forma no se descarta la posibilidad de utilizar regresión lineal.

2.4 Propuesta de solución

Esta sección tiene como objetivo presentar la elección de solución que ofrece el camino más viable para dar solución a la problemática planteada en este trabajo de tesis.

La decisión tomó fundamento en el análisis de la combinación de las tecnologías de la información (especificadas a continuación) para el desarrollo del proyecto propuesto:

- **Software para la implementación de técnicas de minería de datos:** *Weka* es una de las herramientas para minería de datos muy completa, desarrollada en el lenguaje de programación Java, el cual es de propósito general y se implementa en un gran número de dispositivos. Por otra parte, cuenta con herramientas para el pre-procesamiento de la información a minar, así como también algoritmos de minería de datos (abordados en el capítulo II).
- **Sistema gestor de base de datos:** Se seleccionó a PostgreSQL como SGBD ya que cuenta con una amplia documentación para su correcto uso y configuración (a diferencia de MySQL y MariaDB), es robusto y ampliamente utilizado en el mundo, además de ser totalmente compatible con ACID, lo que provee seguridad de que la información guardada está segura.
- **Lenguaje de programación:** Debido a que la naturaleza del proyecto propuesto es *Web* y a la compatibilidad y comunicación nativa que *Weka* implementa con Java, JSF es el lenguaje (Framework de JSP) seleccionado. A diferencia de PHP, JSF implementa el patrón arquitectónico MVC naturalmente, cuenta con validaciones propias del servidor en el cual se está ejecutando, la implementación de JSF hace que los programadores en Java no se percaten de que están desarrollando para la Web al “ocultar” dicha parte delegándosela al servidor. Por último, al tratarse de Java su combinación con *Weka* es ideal para el proyecto propuesto.
- **IDE:** Aunque Eclipse luce bastante bien al ser un IDE ligero y extensible, el hecho de que se deba configurar el servidor por separado para después

integrarse, y que cuente solamente con lo necesario para desarrollar aplicaciones sencillas, hizo que NetBeans se seleccionara para ser el IDE para este proyecto, debido a que cuenta con una versión específica que contiene el servidor en donde se ejecuta JSF (GlassFish) sin necesidad de configurar nada aparte, de igual forma se trata de un IDE ampliamente utilizado por desarrolladores.

- **Metodología:** Debido al tiempo de desarrollo y al hecho de no ignorar demasiado la documentación, se seleccionó a SCRUM como metodología a utilizar, ya que al regirse bajo un enfoque ágil permite cierta flexibilidad para la inclusión de los artefactos que la ingeniería de software recomienda para sustentar los requerimientos y características de la solución planteada, enfocándose en la entrega del producto de software a desarrollar de forma iterativa e incremental sin abandonar la documentación.

Capítulo 3. Aplicación de la metodología

En este capítulo se presenta el diseño y desarrollo del proyecto de tesis aplicando la metodología propuesta en la sección anterior, de igual forma se presenta la arquitectura de la aplicación, sus capas, componentes y flujo de trabajo de la aplicación. También se describen las partes más relevantes referente a diseño y desarrollo del proyecto de tesis.

3.1 *Sprint 0*

Este primer sprint no aporta ningún valor de negocio, es decir, este paso no tiene como objetivo producir una o alguna parte del producto “tangible” y funcional. Al contrario, busca plasmar y cimentar el análisis que sustentará el producto de software esperado. Se inicia con actividades básicas pero fundamentales, tales como: análisis de requerimientos, diseños previos, modelo conceptual, entre otros.

En esta sección tanto el diseño como la arquitectura creada se realizan de manera minimalista dando cabida a cambios durante el tiempo de vida del desarrollo del producto de software. Haciendo lo anterior se obtiene un panorama del producto a realizar.

El presente trabajo hace uso de técnicas de minería de datos para la elaboración de modelos de *clustering* y series de tiempo. La Figura 3.1 muestra el comportamiento de la aplicación, el cual consiste en un módulo independiente del sistema principal de SIFEI elaborado en Java, dicho módulo implementa las bibliotecas de WEKA las cuales proveen los algoritmos necesarios para la generación de modelos de minería de datos, de igual forma la aplicación proporciona las vistas necesarias para que el usuario observe los resultados.

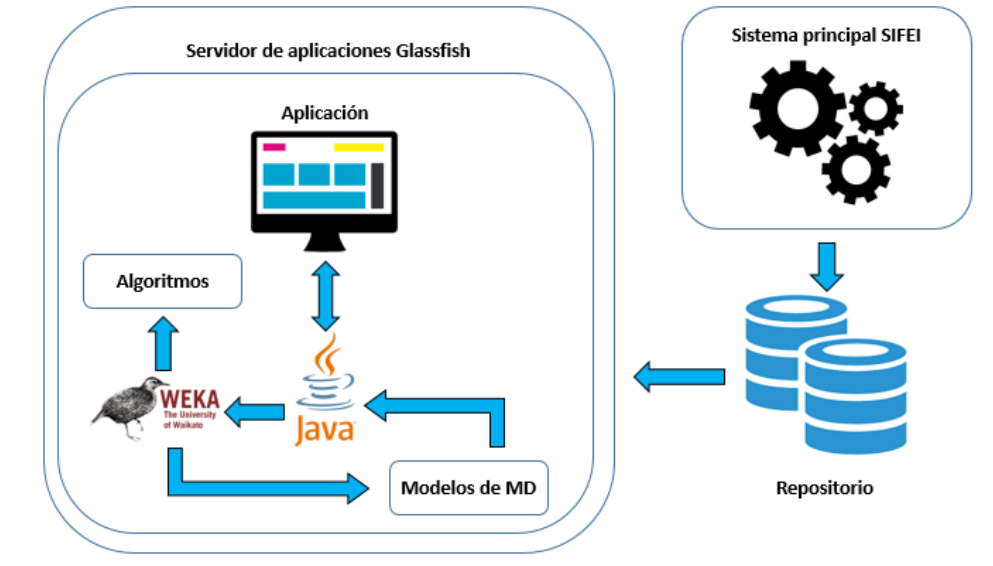


Figura 3.1 Esquema general de la solución.

Dado que SIFEI no tiene tipificados (agrupados) a sus clientes se creó un modelo de agrupamiento para encontrar dichos grupos de clientes, habiendo realizado lo anterior se desarrolló un modelo de clasificación (predicción) para ubicar a nuevos clientes en alguno de esos grupos. La Figura 3.2 representa lo anterior.

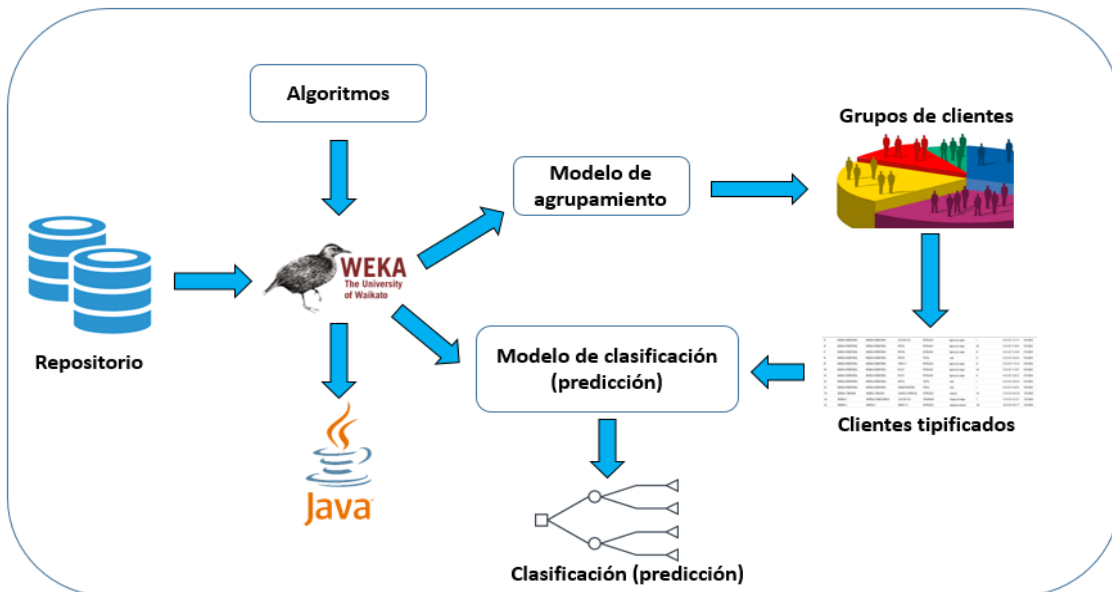


Figura 3.2 Proceso para el agrupamiento y predicción de clientes.

La Figura 3.3 muestra el procedimiento para la obtención de modelos de series de tiempo con el fin de obtener un pronóstico que sea de utilidad para la empresa, de tal forma que al conocer el posible escenario futuro SIFEI se prepare (ya sea con alguna acción mercantil) para “atacar” el mercado y obtener beneficios.

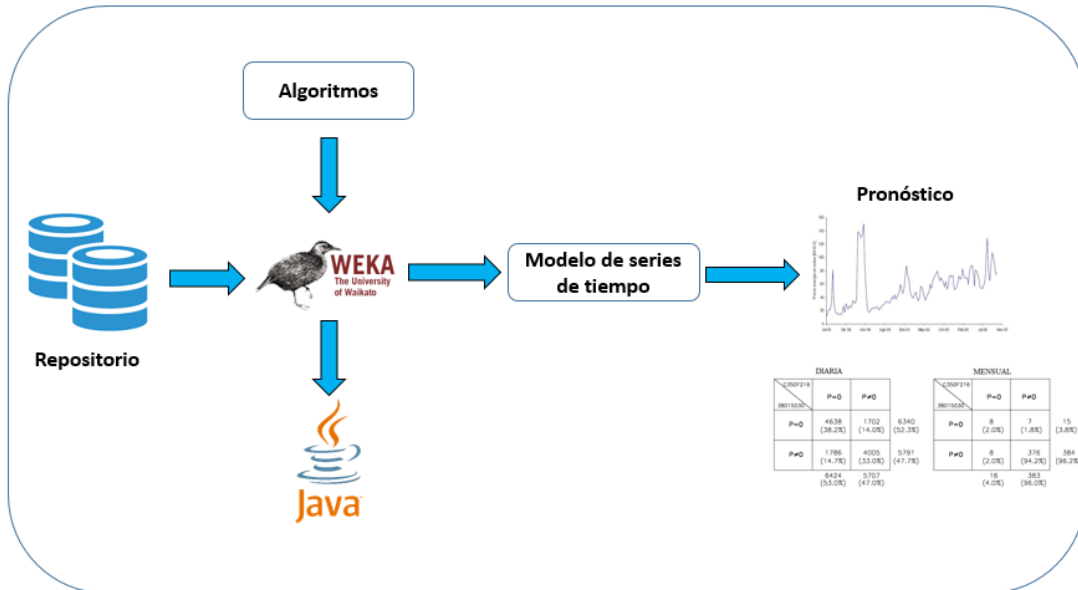


Figura 3.3 Proceso para la obtención de pronósticos mediante series de tiempo.

3.1.1 Arquitectura de la aplicación

La implementación de técnicas de minería de datos permite el descubrimiento de conocimientos ocultos en grandes cantidades de información. Con el uso del proyecto propuesto, SIFEI tomará decisiones para la generación de campañas publicitarias (o en cualquier otro ámbito) que beneficien sus ganancias gracias a los resultados derivados del análisis de la información de sus clientes.

La arquitectura propuesta se basa en capas para facilitar su organización y mantenimiento, cada capa está compuesta por módulos, y algunos de ellos contienen sub-módulos que se comunican entre sí a través de sus propias tareas. Esta comunicación crea un flujo de trabajo entre cada módulo, la Figura 3.4 muestra la arquitectura, sus componentes y las relaciones entre ellos, con el fin de entenderlo mejor.

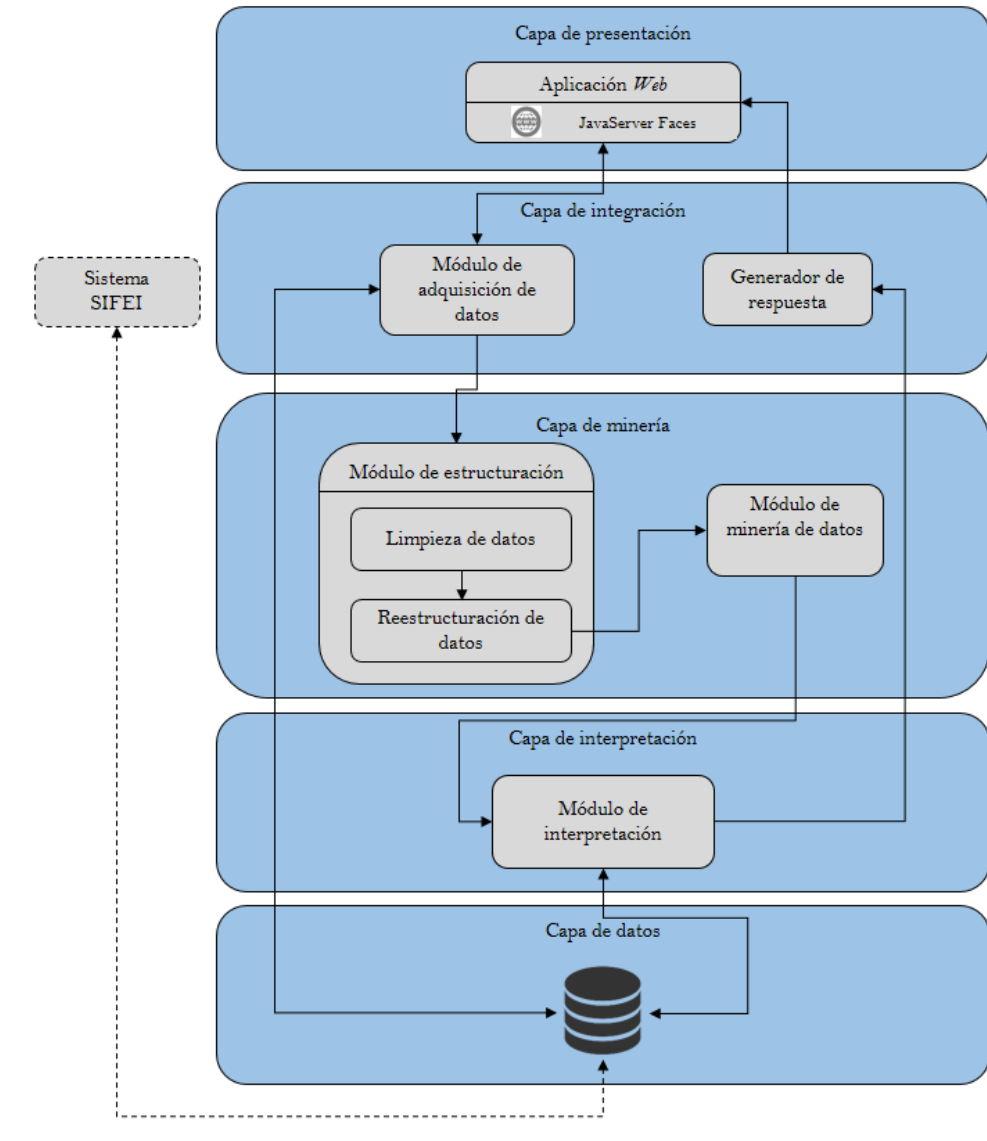


Figura 3.4 Arquitectura propuesta.

3.1.1.1 Descripción de las capas

El funcionamiento general de cada capa es descrito brevemente a continuación:

- **Capa de presentación:** Esta capa representa la interfaz Web que existe entre el usuario y la aplicación, en ella hay componentes que permiten la interacción con la aplicación.
- **Capa de integración:** Esta capa contiene los componentes necesarios para enviar y recibir información e interactuar con la capa de presentación.

- **Capa de minería:** Esta capa contiene en primer lugar un módulo encargado de la limpieza y reestructuración de la información recogida en la capa anterior, así como las bibliotecas necesarias para la implementación de técnicas de minería de datos, las bibliotecas serán proporcionadas por la suite de minería de datos Weka.
- **Capa de interpretación:** Esta capa es la encargada de interpretar el análisis elaborado por la capa anterior, teniendo en cuenta los resultados proporcionados por la implementación de técnicas de minería de datos.
- **Capa de datos:** Para esta arquitectura se consideró disponer de un repositorio de información para almacenar los datos extraídos del sistema principal de SIFEI y clasificarlos para su posterior análisis, así como para almacenar los resultados de los análisis realizados.

Aunque un sistema externo (en este caso el sistema principal de SIFEI) no se considera como una capa de la arquitectura, se representa como un componente externo esencial, ya que es a partir de ahí que se obtendrá la información una vez que se implemente el sistema.

3.1.1.2 Descripción de los componentes

Los componentes de cada capa tienen funciones específicas que determinan su comportamiento y se explican brevemente a continuación:

- **Aplicación web:** Proporciona una interfaz gráfica de usuario (GUI) desarrollada a través del framework JavaServer Faces.
- **Módulo de adquisición de datos:** Componente encargado de obtener los datos necesarios (dependiendo de lo elegido por el usuario) para su análisis.
- **Generador de respuesta:** Componente encargado de generar la vista correspondiente al usuario final, los resultados se muestran ya sea en forma de gráficos, datos tabulares o textos que describan las recomendaciones obtenidas.
- **Módulo de estructuración:** Módulo cuyo objetivo es obtener un conjunto de datos viables para la implementación de técnicas de minería de datos.

- **Limpieza de datos:** Componente encargado de limpiar los datos a analizar, ya que la información extraída de algún repositorio llega a contener ruido o datos faltantes, por mencionar un ejemplo.
- **Reestructuración de datos:** Sub-módulo encargado de aplicar el formato apropiado de acuerdo a la técnica o técnicas de minería de datos requeridas.
- **Módulo de minería de datos:** Este módulo implementa las bibliotecas de Weka y es el responsable de aplicar técnicas de minería de datos a información previamente limpiada y estructurada, resultando en un modelo que será utilizado por el módulo de interpretación.
- **Módulo de interpretación:** Este módulo se encarga de interpretar los resultados obtenidos en el módulo de minería de datos, cuando un resultado es generado (recomendación para campañas publicitarias) es almacenado en el repositorio de datos, luego envía la recomendación al constructor de la respuesta para ser mostrada al usuario final.
- **Base de datos:** En este componente se almacenan los datos requeridos en estructuras que el sistema principal de SIFEI se encarga de poblar, así como también las recomendaciones para los grupos generados.

3.1.1.3 Descripción del flujo de trabajo

Las relaciones entre los componentes de la arquitectura mostrada en la Figura 3.4 definen el flujo de trabajo para el proceso de análisis de datos, desde el inicio de sesión hasta la presentación de los resultados al usuario final. He aquí una breve descripción del flujo de trabajo de la arquitectura mostrado en la Figura 3.5.

- Mediante la aplicación Web desarrollada con JavaServer Faces, el usuario en primera instancia iniciará sesión (debido a la sensibilidad de la información que SIFEI maneja).
- Una vez que el usuario se autentica correctamente y dependiendo de su tipo de usuario, elige alguna acción a realizar (tipificar un cliente, obtener pronóstico de ventas o consumo, gestionar usuarios, entre otros).
- Habiendo el usuario seleccionado una acción, el módulo de adquisición de datos obtiene los parámetros que el usuario indicó para realizar las peticiones

necesarias para obtener la información del repositorio compartido con el sistema principal de SIFEI.

- Cuando el módulo de adquisición de datos termina su función, transfiere el control al módulo de estructuración, en donde en primera instancia el sub-módulo de limpieza de datos se encarga del “ruido”, datos faltantes u otro tipo de anomalía que evita la correcta implementación de las técnicas de minería de datos.
- Una vez que la información se limpió, ésta se transfiere el sub-módulo de reestructuración de datos en donde los datos se estructuran de acuerdo a la técnica (o técnicas) de minería de datos a aplicar. Como resultado, el módulo de estructuración obtiene un conjunto estructurado de información para su análisis en el módulo de minería de datos.
- Teniendo la información limpia y estructurada, el módulo de minería de datos es responsable de aplicar la técnica (o técnicas) adecuada para obtener un modelo, el cual se evalúa en el siguiente módulo.
- En el módulo de interpretación, el modelo generado en el módulo anterior será analizado y con los patrones encontrados se generan los resultados para ser mostrados al usuario final. Por último este módulo envía los resultados obtenidos al generador de respuesta.
- Finalmente, el generador de respuesta da formato a la respuesta obtenida en el módulo anterior, dicho formato consiste ya sea de gráficas o texto plano para la aplicación Web y ésta se muestra al usuario final.

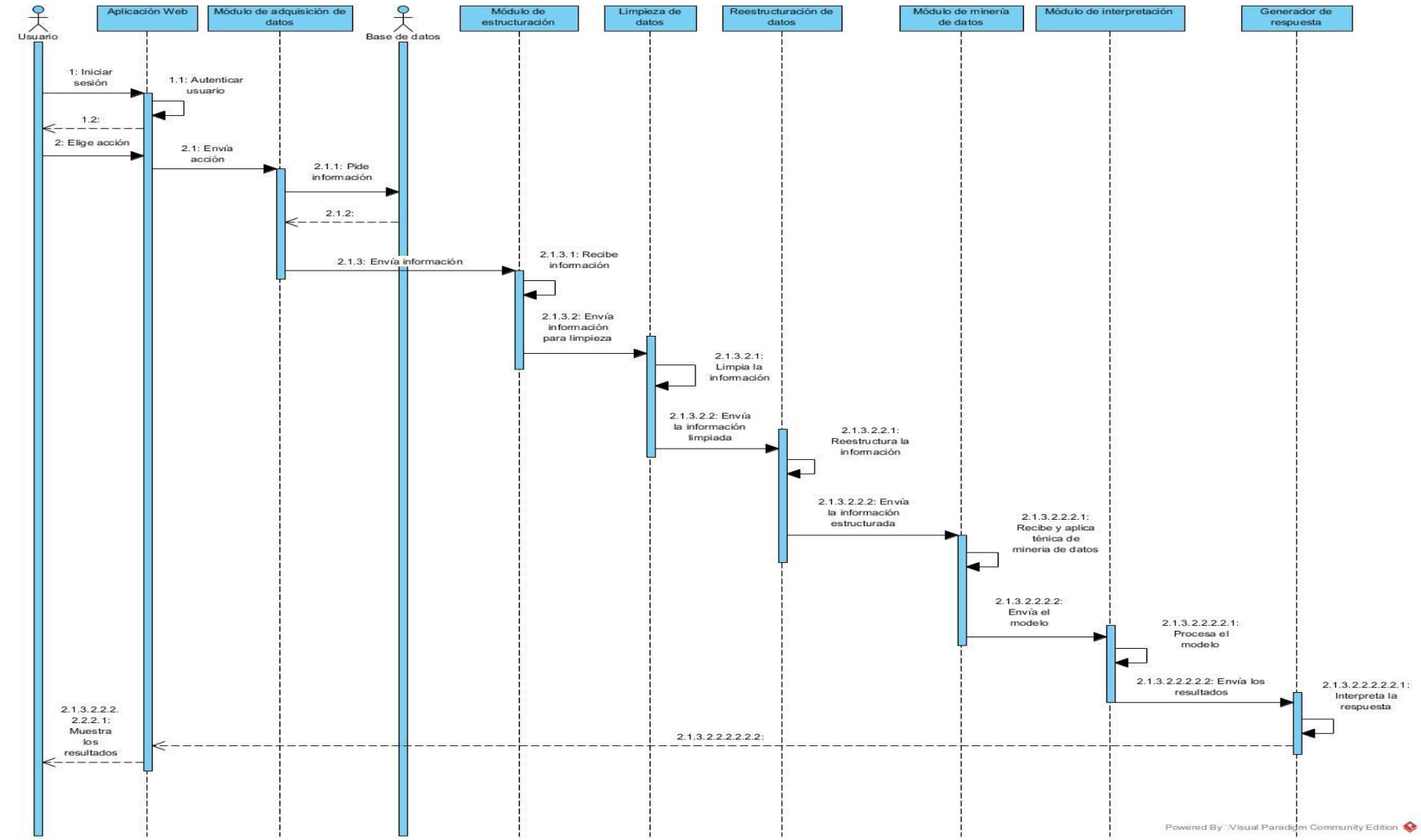


Figura 3.5 Diagrama de secuencia del flujo de trabajo de la arquitectura.

3.1.2 Especificación de requisitos de la aplicación

A continuación, se describen cada uno de los requisitos que debe cumplir la aplicación, posteriormente se representan dichos requerimientos en un diagrama de casos de uso para su mejor comprensión.

3.1.2.1 Identificación de requerimientos

La aplicación a desarrollar debe fungir como un módulo del sistema principal de la empresa Solución Integral de Facturación Electrónica e Informática S. A. de C. V. (SIFEI) de tal forma que sea un producto de software independiente por sí mismo, utilizando información provista por el propio sistema principal. El objetivo del módulo es buscar y obtener los patrones de consumo ocultos en dicha información, una vez que son extraídos deben ser mostrados al usuario final de tal manera que éste tenga las herramientas necesarias para tomar decisiones ya sea mercantiles, administrativas o preventivas, por mencionar algunas.

El usuario inicia la interacción con la aplicación ingresando su usuario y contraseña (datos definidos por la empresa mediante el usuario administrador) debido a la sensibilidad de la información a manejar se debe mantener controlado el acceso a dicha información. Una vez autenticado y dependiendo del tipo de usuario se mostrarán las acciones permitidas para realizar:

- **Administrador:** Este usuario da de alta a un nuevo usuario (clasificador o pronosticador), modificarlo o eliminarlo; de igual forma solamente modifica las recomendaciones para los grupos de clientes que se determinen.
- **Clasificador:** Este usuario tiene la facultad de consultar los grupos de clientes en donde se muestran las características principales así como también la recomendación para cada grupo de clientes; tipificar un cliente que no pertenezca a algún grupo; y por último consultar a los clientes ya clasificados para determinar alguna acción propia de la empresa.
- **Pronosticador:** Este usuario pronostica las compras de un cliente determinado bajo un intervalo de tiempo, sobre algún producto en específico; y por último pronostica las ventas por parte de la empresa sobre algún producto en específico bajo un intervalo de tiempo.

Para los usuarios de tipo Clasificador y Pronosticador, los resultados se muestran de manera clara y concisa, ya sea de forma tabular, texto plano o gráficas para que de esa forma realicen alguna acción. En la Figura 3.6 se muestran los casos de uso encontrados.

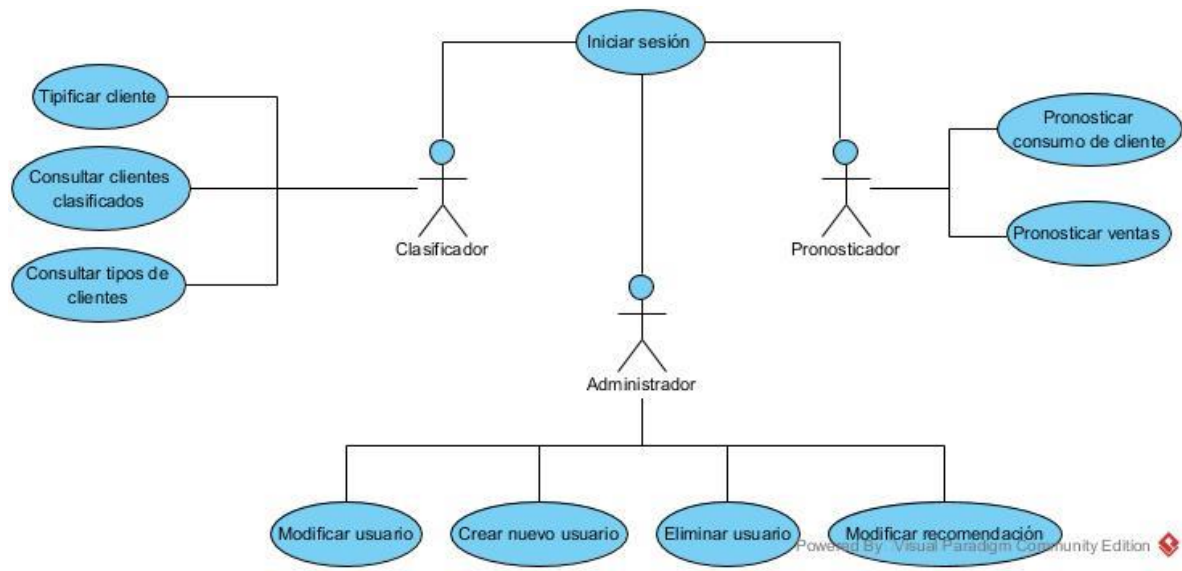


Figura 3.6 Diagrama de casos de uso.

3.1.2.2 Especificaciones de caso de uso

Tomando como base la Figura de la sección anterior, se procede a especificar cada uno de los casos de uso (CU) mostrados, con el fin de considerar todos los caminos posibles para el correcto funcionamiento del sistema propuesto. A continuación se presentan las descripciones de los casos de uso así como su correspondiente diagrama de secuencia, sin embargo solamente se definen los diagramas de secuencia asociados a los casos de uso más relevantes de la aplicación (aquellos que se asocian a la minería de datos).

3.1.2.2.1 Iniciar sesión

CU: Iniciar sesión.

Precondiciones:

- Ninguna.

Postcondiciones

- Cliente autenticado.

Descripción u objetivo: Autenticar a un usuario determinado. Una vez que éste se autenticó dependiendo de qué tipo de usuario sea, el sistema le mostrará las opciones correspondientes que tiene facultad de realizar (consultar diagrama de casos de uso).

Curso principal

Actor

Sistema

1. Indica su usuario y contraseña.
2. El sistema toma los parámetros y en caso de no recibir alguno o ningún parámetro ver Curso de Excepción 1, en caso contrario realiza una consulta a la base de datos para buscar al usuario con la contraseña determinada, en caso de no encontrar alguna concordancia con los usuarios dados de alta ver Curso de Excepción 2, en caso de encontrar al usuario, el sistema lo autentica redirigiéndolo a una nueva página.
3. Termina caso de uso

Curso de excepción 1

Actor

Sistema

1. Indica que no se le han proporcionado los parámetros requeridos.
2. Termina caso de uso

Curso de excepción 2
Actor

Sistema

1. Indica que no se ha encontrado al usuario especificado.
2. Termina caso de uso

3.1.2.2.2 Tipificar cliente

CU: Tipificar cliente.

Precondiciones:

- Usuario autenticado.

Postcondiciones

- Cliente tipificado.

Descripción u objetivo: En este caso de uso se realizan las tareas necesarias para perfilar a algún cliente.

Curso principal
Actor

Sistema

1. Selecciona la opción “Clasificar”
2. El sistema muestra una tabla en la que se encuentran todos los clientes que no se encuentran en algún grupo, así como también datos generales de éstos con el fin de diferenciarlos, en caso de no encontrar clientes por clasificar, ver Curso de Excepción 1.
3. Selecciona al cliente o clientes a perfilar y selecciona el botón “Clasificar”.
4. En caso de que el usuario no haya seleccionado ningún cliente ver Curso de Excepción 2, en caso contrario el sistema pedirá la confirmación por parte del usuario.
5. En caso de que el usuario no confirme la selección ver Curso de excepción 3, en caso contrario confirmará la selección.

6. El sistema toma al cliente o clientes seleccionados y los clasifica, al terminar muestra al usuario un mensaje avisando que todo se efectuó de manera correcta
7. Termina caso de uso.

Curso de excepción 1
Actor

Sistema

1. Indica que no se han encontrado clientes por clasificar.
2. Termina caso de uso

Curso de excepción 2
Actor

Sistema

1. Indica que no se seleccionó al menos un cliente para su clasificación.
2. Termina caso de uso

Curso de excepción 3
Actor

Sistema

1. El sistema no continúa con la clasificación.
2. Termina caso de uso.

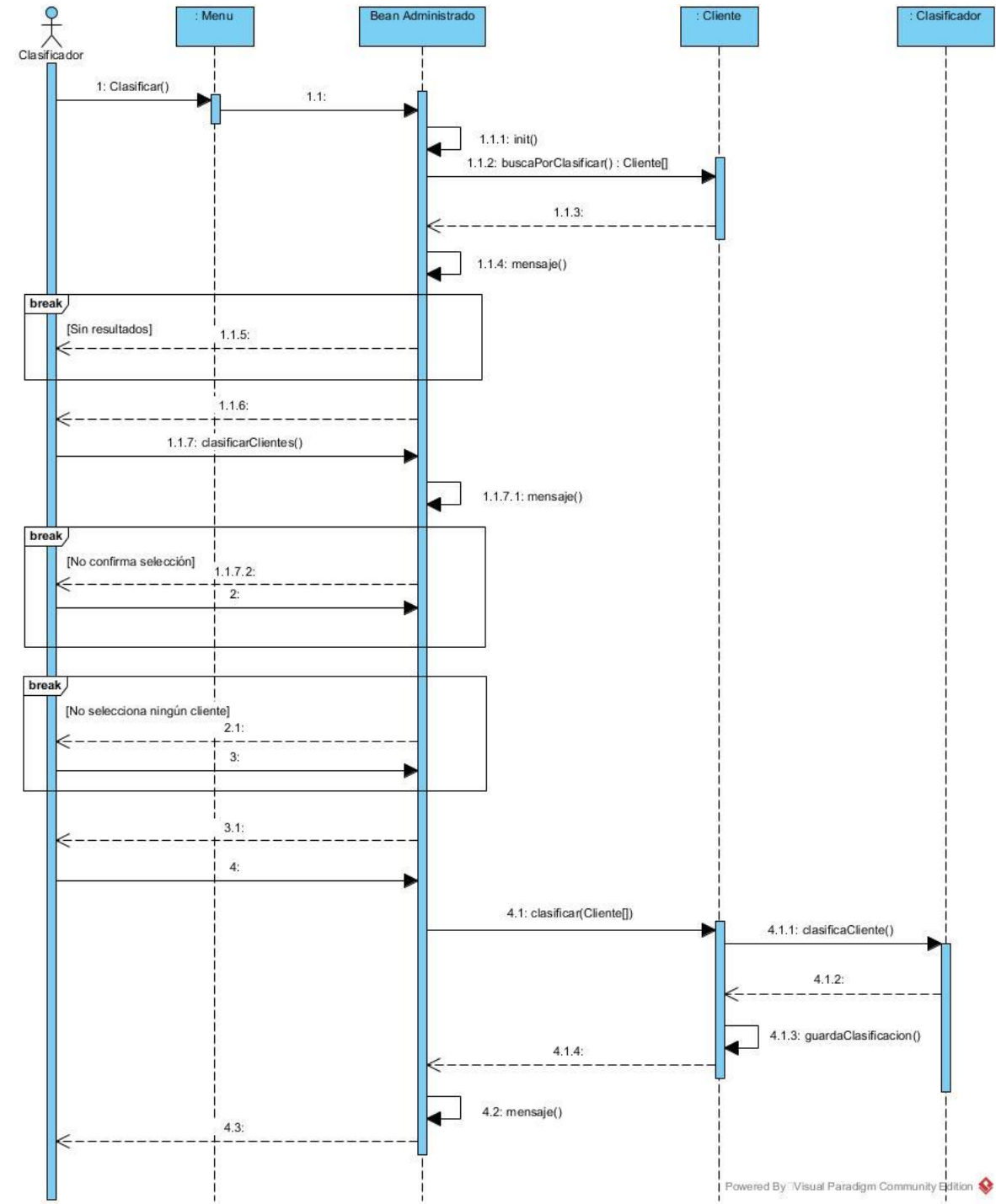


Figura 3.7 Diagrama de secuencia Tipificar cliente.

3.1.2.2.3 Consultar clientes clasificados

CU: Consultar clientes clasificados.

Precondiciones:

- Usuario autenticado.

Postcondiciones

- Ninguna.

Descripción u objetivo: El sistema le muestra al usuario los clientes ya clasificados o alguno en específico que requiera.

Curso principal

Actor

Sistema

- | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. Selecciona la opción "Ver clasificaciones". 3. En caso de que el usuario no encuentre al cliente que desee, ver Curso alternativo 1. | <ol style="list-style-type: none"> 2. El sistema muestra una tabla en la que se encuentran los clientes recientemente clasificados y a qué grupo pertenecen, así como también muestra un formulario para buscar a algún cliente en específico. 4. Termina caso de uso. |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Curso alternativo 1 (El usuario desea buscar a un cliente determinado)

Actor

Sistema

- | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. Determina bajo qué criterio de búsqueda desea encontrar al cliente deseado (búsqueda por Id o por Nombre), sea cual sea la opción deberá ingresar el cliente a buscar. | <ol style="list-style-type: none"> 2. El sistema obtiene el cliente a buscar, en caso de no recibirlo, ver Curso de Excepción 1, en caso de que el usuario seleccionó como criterio de |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

búsqueda “Id” el sistema deberá comprobar que el parámetro sea un número, en caso de no serlo, ver Curso de Excepción 2. El sistema realiza las peticiones necesarias para la búsqueda del cliente deseado, en caso de no encontrarlo, ver Curso de excepción 3, una vez encontrado al cliente se le mostrará al usuario los datos y el grupo al que pertenece dicho cliente.

3. Termina caso de uso.

Curso de excepción 1
Actor

Sistema

1. Indica que no se determinó el cliente a buscar.
2. Termina caso de uso

Curso de excepción 2
Actor

Sistema

1. Indica que el Id provisto no es de tipo numérico.
2. Termina caso de uso

Curso de excepción 3
Actor

Sistema

1. Indica que no se encontraron resultados.
2. Termina caso de uso

3.1.2.2.4 Consultar tipos de clientes

CU: Consultar tipos de clientes.

Precondiciones:

- Usuario autenticado.

Postcondiciones

- Ninguna.

Descripción u objetivo: El sistema le muestra al usuario los grupos de clientes con los que se cuentan, una característica es que también se muestran las recomendaciones para cada grupo (provistas por el usuario administrador).

Curso principal

Actor

Sistema

- | | |
|----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. Selecciona la opción “Ver Grupos” | <ol style="list-style-type: none"> 2. El sistema muestra los grupos de clientes que existen, no sin antes realizar las peticiones necesarias para obtener de la base de datos, las recomendaciones para cada uno de dichos grupos. 3. Termina caso de uso. |
|----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

3.1.2.2.5 Crear nuevo usuario

CU: Crear nuevo usuario.

Precondiciones:

- Usuario autenticado.

Postcondiciones

- Nuevo usuario para el sistema.

Descripción u objetivo: El usuario de tipo administrador, tiene la facultad de gestionar los usuarios que harán uso del sistema, este caso de uso se refiere únicamente a la creación de un nuevo usuario.

Curso principal
Actor

Sistema

- | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. Selecciona la opción “Usuarios”
 3. Selecciona el botón “Nuevo registro”
 5. Ingresar los datos correspondientes y seleccionar el botón “Guardar”. | <ol style="list-style-type: none"> 2. El sistema muestra un formulario para buscar usuarios ya registrados.
 4. Muestra un formulario para la captura de los datos requeridos para dar de alta a un nuevo usuario.
 6. El sistema verifica el usuario ingresó los datos requeridos de no ser así, ver Curso de excepción 1, una vez verificados, el sistema realiza las peticiones necesarias para dar de alta al nuevo usuario, al término le muestra al usuario un mensaje en donde se le notifica el alta del usuario proporcionado. 7. Termina caso de uso. |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Curso de excepción 1
Actor

Sistema

- | |
|--------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. Indica que no se han proporcionado datos. 2. Termina caso de uso |
|--------------------------------------------------------------------------------------------------------------------------------|

3.1.2.2.6 Modificar usuario

CU: Modificar usuario.

Precondiciones:

- Usuario autenticado.

Postcondiciones

- Usuario del sistema modificado.

Descripción u objetivo: El usuario de tipo administrador, tiene la facultad de gestionar los usuarios que harán uso del sistema, este caso de uso se refiere únicamente a la modificación de un usuario previamente dado de alta en el sistema.

Curso principal

Actor

Sistema

- | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. Selecciona la opción “Usuarios” 3. Selecciona al usuario que se desea modificar, En caso de que requiera buscar a un usuario en específico, ver Curso Alterno 1. 5. Modifica los datos correspondientes y selecciona el botón “Guardar”. | <ol style="list-style-type: none"> 2. El sistema muestra un formulario para buscar usuarios ya registrados. 4. Muestra un formulario previamente llenado con los datos del usuario a actualizar. 6. El sistema verifica el usuario haya ingresado los datos requeridos de no ser así, ver Curso de excepción 1, una vez verificados, el sistema realiza las peticiones necesarias para modificar al usuario, al término le muestra al usuario un mensaje en donde se le notifica la actualización del usuario proporcionado. 7. Termina caso de uso. |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Curso Alterno 1 (Se desea buscar a un usuario determinado)

Actor

Sistema

1. Indica el usuario a buscar y selecciona el botón "Buscar".
2. El sistema busca al usuario determinado, en caso de que el usuario no haya ingresado nada, se buscarán todos los usuarios del sistema, en caso de no encontrar resultados, ver Curso de Excepción 2. Una vez encontrado al usuario, el sistema los despliega.
3. Termina curso alternativo 1, regresa a curso principal.

Curso de excepción 1

Actor

Sistema

1. Indica que no se proporcionaron datos.
2. Termina caso de uso

Curso de excepción 2

Actor

Sistema

1. Indica que no hay resultados para mostrar.
2. Termina caso de uso

3.1.2.2.7 Eliminar usuario

CU: Eliminar usuario.

Precondiciones:

- Usuario autenticado.

Postcondiciones

- Eliminación de un usuario del sistema.

Descripción u objetivo: El usuario de tipo administrador, tiene la facultad de gestionar los usuarios que harán uso del sistema, este caso de uso se refiere únicamente a la eliminación de un usuario previamente dado de alta en el sistema.

Curso principal
Actor

Sistema

- | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. Selecciona la opción “Usuarios”
 3. Selecciona al usuario que se desea eliminar, en caso de requerir la búsqueda de un usuario en particular, ver Curso alterno 1.
 5. Confirma la eliminación del usuario seleccionado. | <ol style="list-style-type: none"> 2. El sistema muestra un formulario para buscar usuarios ya registrados.
 4. Pide la confirmación del usuario para proceder con la eliminación.
 6. El sistema realiza las peticiones necesarias para eliminar al usuario, al término le muestra al usuario un mensaje en donde se le notifica la eliminación del usuario proporcionado. 7. Termina caso de uso. |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Curso Alterno 1 (Se desea buscar a un usuario determinado)

Actor

Sistema

- | | |
|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. Indica el usuario a buscar y selecciona el botón “Buscar”. | <ol style="list-style-type: none"> 2. El sistema busca al usuario determinado, en caso de que el usuario no haya ingresado nada, se buscarán todos los usuarios del sistema, en caso de no encontrar resultados, ver Curso de Excepción 1. Una vez encontrado al usuario, el sistema los despliega. |
|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

3. Termina curso alterno 1, regresa a curso principal.

Curso de excepción 1
Actor

Sistema

1. Indica que no hay resultados para mostrar.
2. Termina caso de uso

3.1.2.2.8 Modificar recomendación

CU: Modificar recomendación.

Precondiciones:

- Usuario autenticado.

Postcondiciones

- Modificar la recomendación para un grupo de clientes específico.

Descripción u objetivo: El usuario de tipo administrador, tiene la facultad de modificar las recomendaciones (previamente dadas de forma genérica al determinar los grupos con los que se cuentan) para cada grupo de clientes, dichas recomendaciones son leídas y ejecutadas por los usuarios de tipo clasificador.

Curso principal

Actor

Sistema

- | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. Selecciona la opción "Recomendaciones"
 3. Selecciona al grupo al que desea modificar la recomendación, en caso de requerir la búsqueda de un grupo en particular, ver Curso alterno 1, selecciona el botón "Modificar" | <ol style="list-style-type: none"> 2. El sistema muestra un formulario para buscar los grupos de clientes.
 4. El sistema muestra un formulario para ingresar la nueva |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- recomendación para el grupo seleccionado.
5. Ingresa la nueva recomendación y selecciona el botón “Guardar”.
 6. El sistema realiza las peticiones necesarias para actualizar la recomendación ingresada, en caso de no ingresar nada, ver Curso de excepción 1, al término le muestra al usuario un mensaje en donde se le notifica la actualización de la recomendación.
 7. Termina caso de uso.

Curso de excepción 1
Actor

Sistema

1. Indica que no se ingresaron datos.
2. Termina caso de uso

3.1.2.2.9 Pronosticar ventas

CU: Pronosticar ventas.

Precondiciones:

- Usuario autenticado.

Postcondiciones

- Pronóstico de las ventas esperadas bajo un intervalo de tiempo.

Descripción u objetivo: Determinar las ventas esperadas en un lapso de tiempo.

Curso principal
Actor

Sistema

- | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. Selecciona la opción “Ventas”

 3. Ingresa los datos requeridos y selecciona el botón “Buscar”. | <ol style="list-style-type: none"> 2. El sistema muestra un formulario para ingresar el producto a pronosticar así como también los campos para determinar el intervalo de tiempo a evaluar.

 4. El sistema verifica que el usuario haya ingresado la información requerida, en caso de no ser así, ver Curso de excepción 1, realiza las peticiones necesarias para pronosticar las ventas, en caso de no encontrar resultados para ese periodo de tiempo, ver Curso de excepción 2, una vez encontradas las ventas le muestra al usuario una gráfica en donde se observe el comportamiento de las ventas, así como también el pronóstico esperado. 5. Termina caso de uso. |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Curso de excepción 1
Actor

Sistema

1. Indica que no se ingresaron datos.
2. Termina caso de uso

Curso de excepción 2
Actor

Sistema

1. Indica que no se encontraron resultados.
2. Termina caso de uso

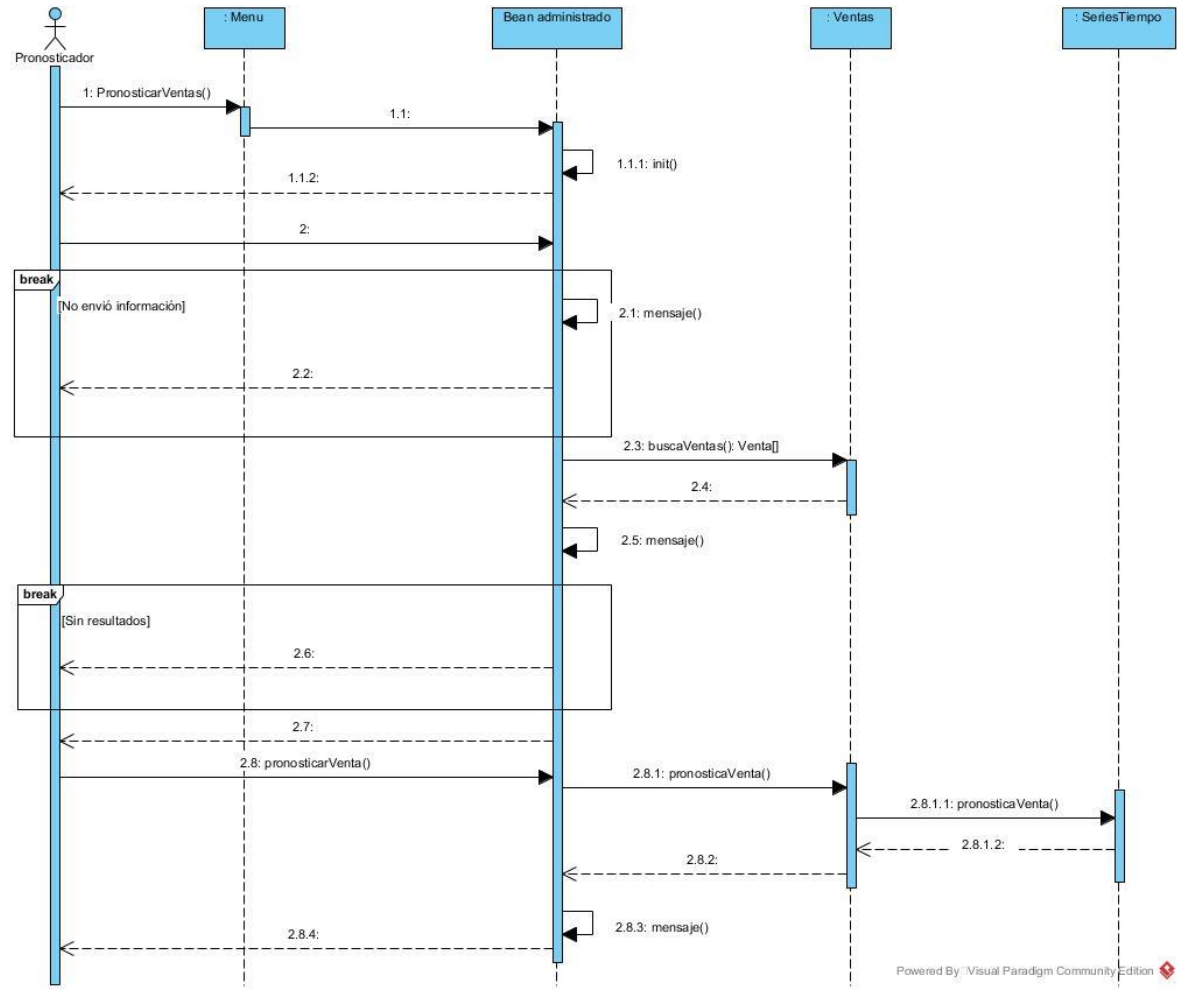


Figura 3.8 Diagrama de secuencia pronosticar ventas.

3.1.2.2.10 Pronosticar consumo de cliente

CU: Pronosticar consumo de cliente.

Precondiciones:

- Usuario autenticado.

Postcondiciones

- Pronóstico de las ventas esperadas bajo un intervalo de tiempo de un cliente en específico.

Descripción u objetivo: Determinar las ventas esperadas en un lapso de tiempo para un cliente en específico.

Curso principal
Actor

Sistema

- | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. Selecciona la opción “Ventas”

 3. Ingresa los datos requeridos y selecciona el botón “Buscar”. | <ol style="list-style-type: none"> 2. El sistema muestra un formulario para ingresar el cliente a pronosticar, el producto, así como también los campos para determinar el intervalo de tiempo a evaluar.

 4. El sistema verifica que el usuario haya ingresado la información requerida, en caso de no ser así, ver Curso de excepción 1, realiza las peticiones necesarias para pronosticar las compras del cliente, en caso de no encontrar resultados para ese periodo de tiempo, ver Curso de excepción 2, una vez encontradas las compras le muestra al usuario una gráfica en donde se observe el comportamiento del cliente, así como también el pronóstico esperado. 5. Termina caso de uso. |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Curso de excepción 1
Actor

Sistema

1. Indica que no se ingresaron datos.
2. Termina caso de uso

Curso de excepción 2
Actor

Sistema

1. Indica que no se encontraron resultados.
2. Termina caso de uso

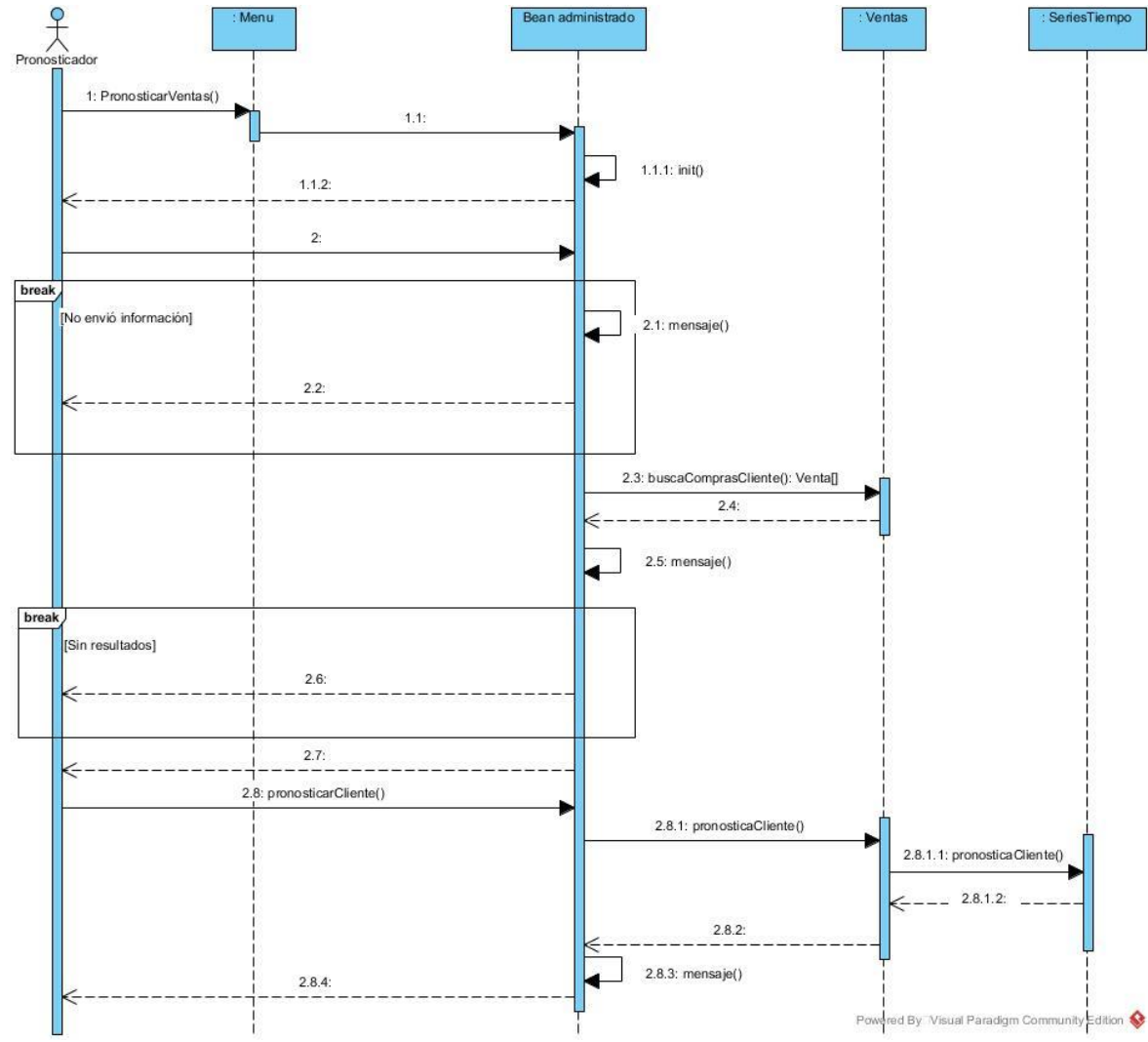


Figura 3.9 Diagrama de secuencia pronosticar consumo de cliente.

3.1.3 Diseño previo de la aplicación

Tomando como base el diagrama de casos de uso y sus descripciones, en este apartado se mostrarán diseños minimalistas de la aplicación a desarrollar con el fin de tener una guía que facilite y agilice el tiempo de desarrollo.

3.1.3.1 Vista de inicio de sesión

Para esta pantalla se pretende contar con dos campos de texto para ingresar el usuario y contraseña del usuario que desee ingresar a la aplicación, así como un botón que efectúe dicho proceso, sin olvidar el logo de la empresa así como el

nombre de la aplicación (dicho logo y nombre deberán ser vistos en todas las pantallas). La Figura 3.10 refleja lo anterior.



Figura 3.10 Vista previa para inicio de sesión.

3.1.3.2 Vista de inicio

Una vez que el usuario se autenticó, el sistema desplegará la vista de inicio, en donde se deberá observar una leyenda dándole la bienvenida al usuario, de igual forma se deberá mostrar un menú en el cual todos los tipos de usuario visiten de nuevamente la vista de inicio y tengan la opción de salir del sistema. Sin olvidar claro, que dependiendo del tipo de usuario son las opciones específicas que realicen o tengan acceso. La Figura 3.11 plasma lo anterior.

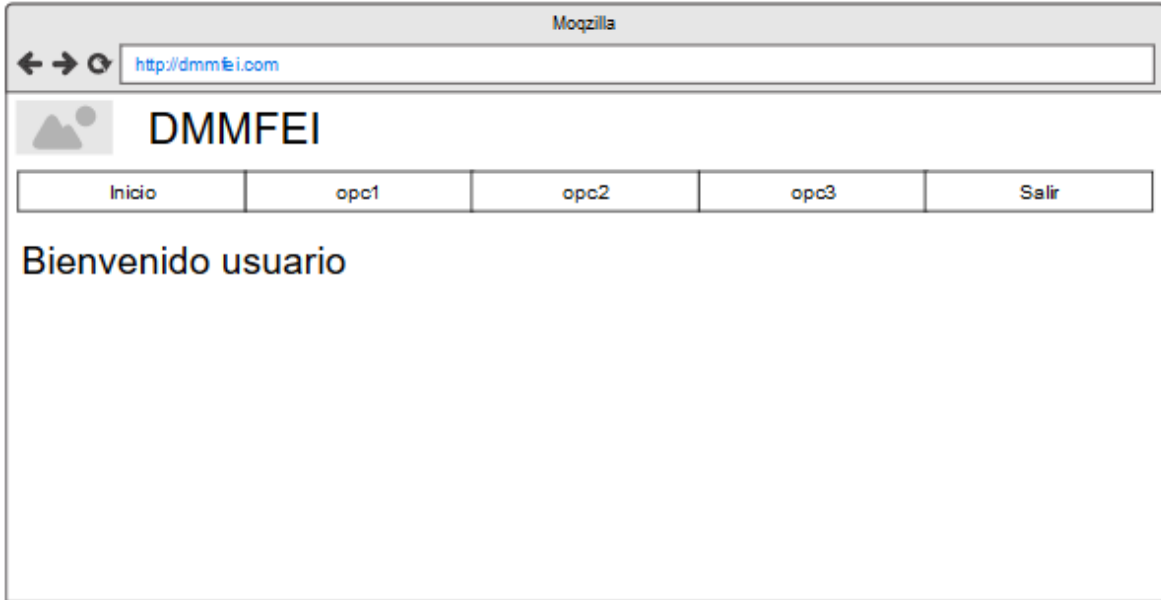


Figura 3.11 Vista de inicio.

Como se observa en la Figura 3.11, entre las opciones “Inicio” y “Salir” se encuentran tres opciones (“opc1”, “opc2”, “opc3”) Las cuales representan las acciones que los usuarios realizan dado su tipo. “opc1” corresponde al tipo de usuario administrador, “opc2” al usuario clasificador y “opc3” al tipo pronosticador.

3.1.3.3 Vistas usuario administrador

Este tipo de usuario tiene la capacidad de realizar a grandes rasgos dos tipos de tareas:

- **Gestionar usuarios:** Crear, modificar y eliminar usuarios.
- **Gestionar recomendaciones:** modificar recomendaciones de los grupos de clientes.

Lo anterior se observa en la Figura 3.12.

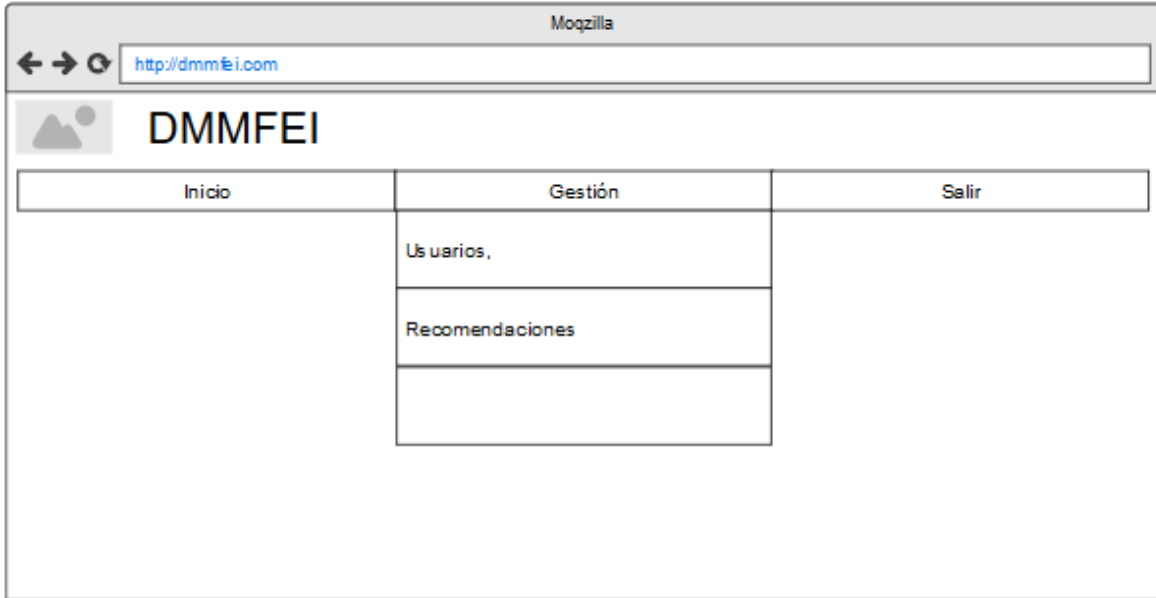


Figura 3.12 Opciones del usuario administrador

3.1.3.3.1 Vista gestión de usuarios

Dada la tecnología empleada en el desarrollo del módulo propuesto, se implementaron las capacidades de la misma, una de esas capacidades son las llamadas parciales, las cuales agilizan el proceso de envío de peticiones y respuesta por parte del servidor, es por eso que se espera tener todo en una misma vista, y dependiendo de las necesidades del usuario, agregar o quitar componentes.

La vista para la gestión de usuarios pretende contar con los componentes necesarios para la interacción con el usuario administrador, por lo tanto se espera contar con una tabla para la fácil y rápida selección de usuarios para su modificación y/o eliminación, de igual forma se mantiene apartada la opción para crear un nuevo usuario. La Figura 3.10 muestra la vista general de esta sección, en la cual se busca a un usuario deseado, si el administrador deseara modificar a un usuario solamente elige la opción adecuada (“Modificar”) y ya sea para modificar o para crear un nuevo usuario, en la misma vista se despliega un formulario lleno (para el caso de modificación) o vacío (para el caso de un nuevo registro). Las Figuras 3.13 a 3.15 representan lo anterior.

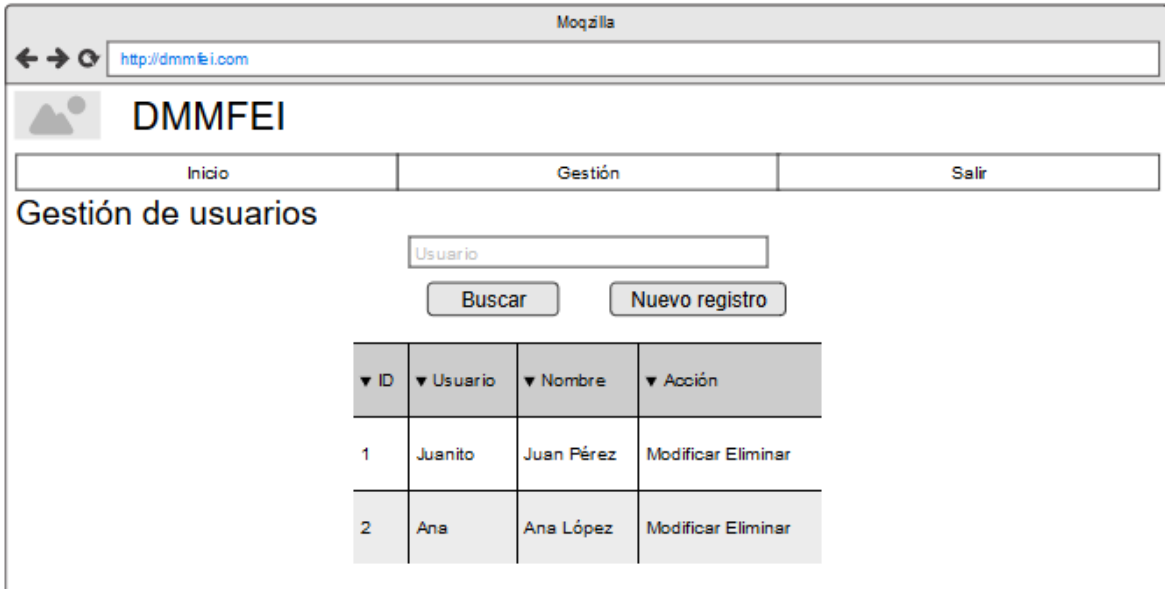


Figura 3.13 Vista general de la gestión de usuarios.

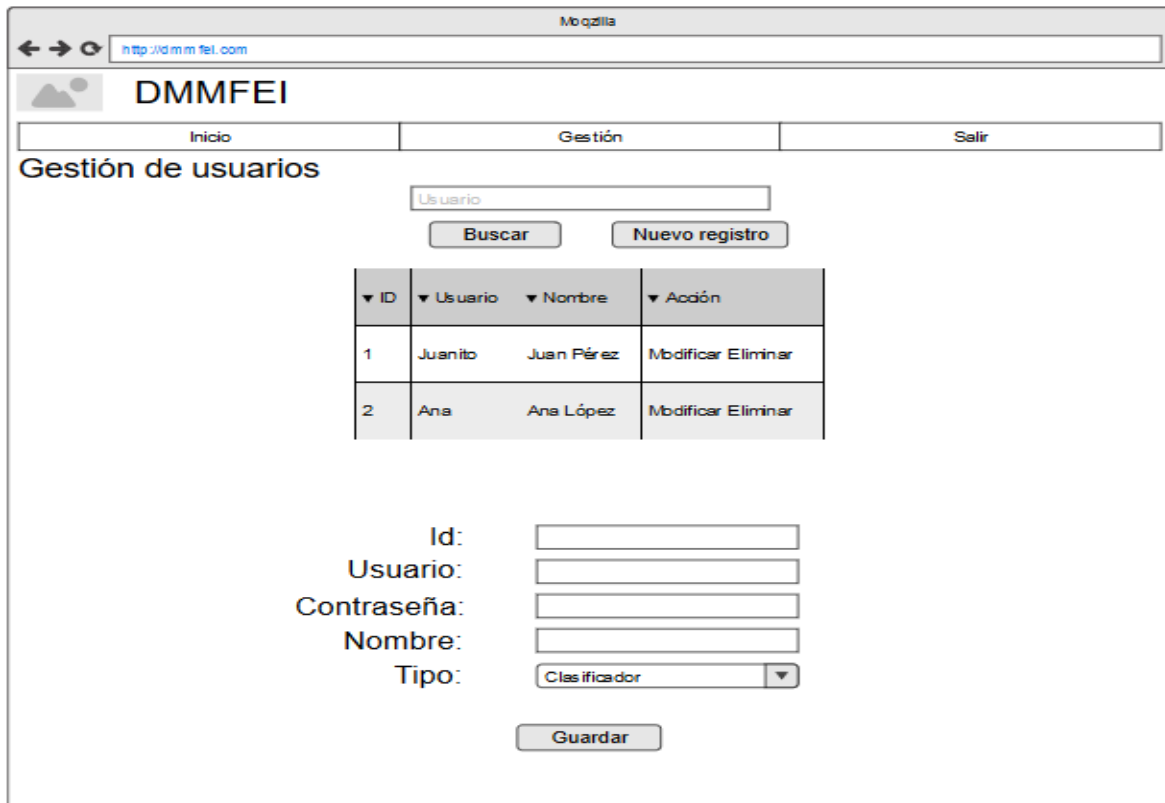


Figura 3.14 Vista para un nuevo usuario.

Inicio Gestión Salir

Gestión de usuarios

Usuario

Buscar Nuevo registro

ID	Usuario	Nombre	Acción
1	Juanito	Juan Pérez	Modificar Eliminar
2	Ana	Ana López	Modificar Eliminar

Id:

Usuario:

Contraseña:

Nombre:

Tipo:

Guardar

Figura 3.15 Vista para la modificación de un usuario.

Para la eliminación de un usuario, basta con que el administrador lo seleccione en la columna “Acción” por lo cual no se requiere de una vista en particular.

3.1.3.3.2 Vista para la gestión de las recomendaciones

Dado que los grupos de clientes estarán fijos una vez que se analice la información de la empresa (minería de datos) la única acción que el usuario administrador realiza hacia esa información, es la de determinar qué sugerencias o recomendaciones dar para cada grupo formado, con base en su experiencia en el negocio.

De igual forma que en la gestión de usuarios, la gestión de recomendaciones tiene una forma de buscar ya sea un grupo en específico o todos los grupos, muestra los resultados de forma tabular y ofrece un formulario para la edición de la recomendación del grupo seleccionado. Las Figuras 3.16 a 3.17 muestran lo anterior.

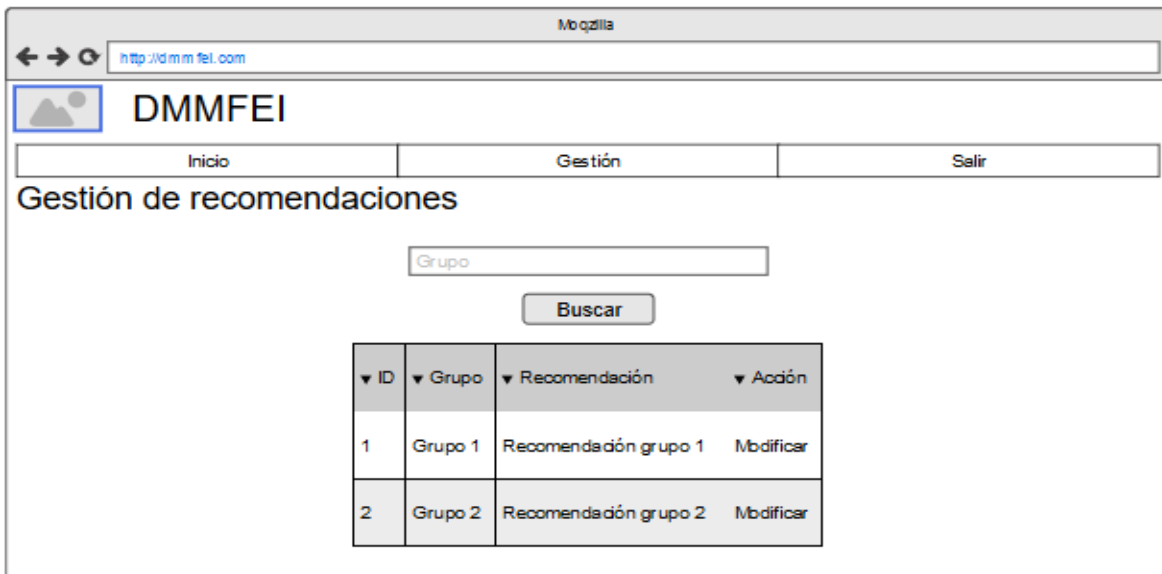


Figura 3.16 Vista general de la gestión de recomendaciones.

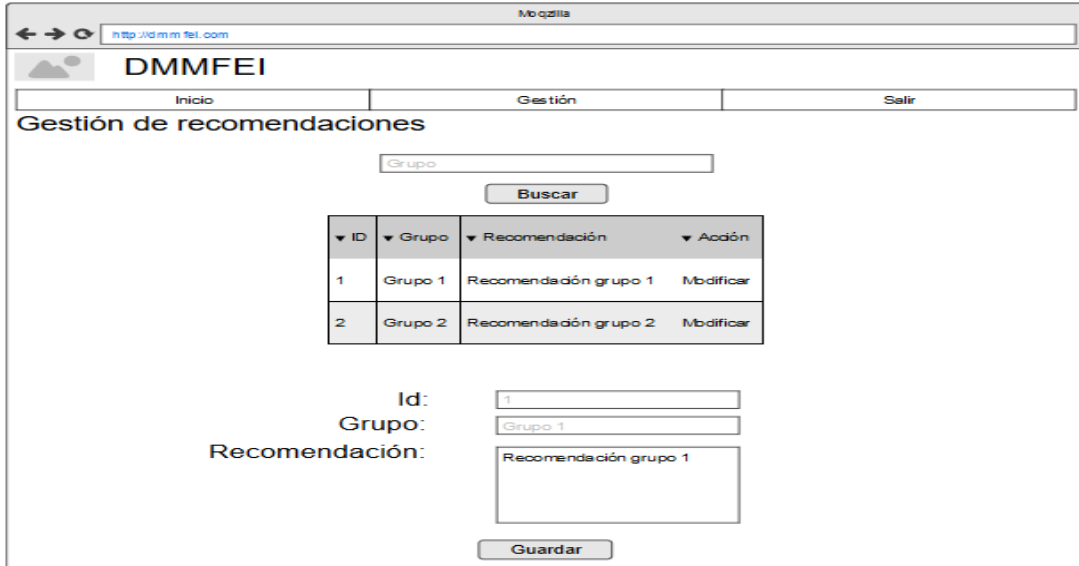


Figura 3.17 Vista para la modificación de una recomendación.

3.1.3.4 Vistas usuario clasificador

Este tipo de usuario tiene la capacidad de realizar tres tareas:

- **Ver grupos:** Características principales de cada grupo de clientes, así como la recomendación que el usuario administrador crea conveniente.
- **Clasificar:** Determinar cuál o cuáles clientes se clasifican para determinar las acciones que mejor le convengan a la empresa.
- **Ver clasificaciones:** Dado que se espera que haya más y más clientes, en caso de requerir de nuevas estrategias de mercadotecnia, esta opción permite buscar y encontrar a los clientes clasificados o a uno en específico.

Como se observa en la Figura 3.18.

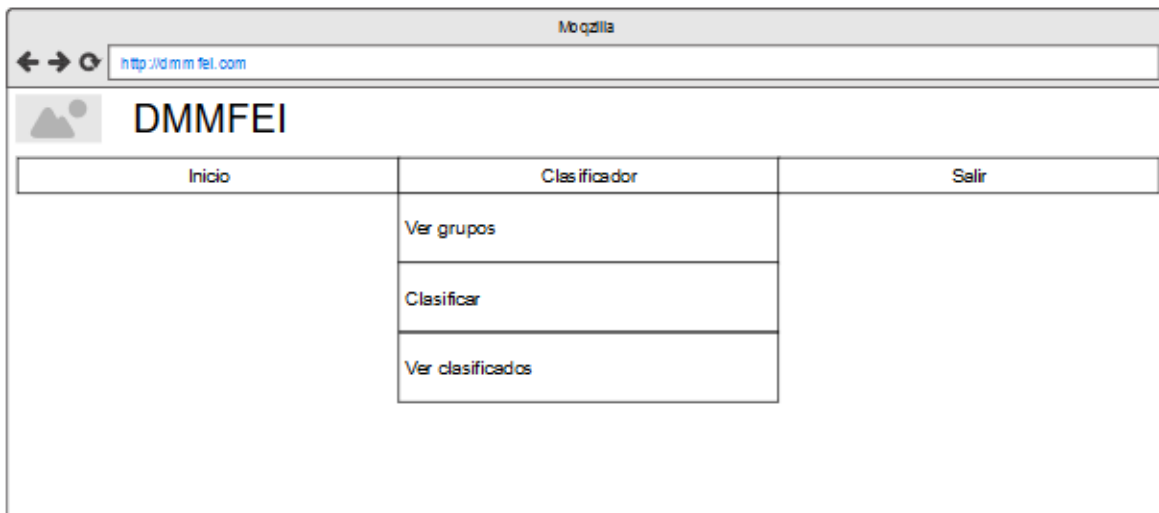


Figura 3.18 Opciones del usuario clasificador.

3.1.3.4.1 Vista ver grupos

Esta vista es meramente informativa para este usuario, dado que funge como una guía para ayudarse a generar alguna acción mercantil o administrativa para un cliente o clientes que formen parte de un grupo en específico. La Figura 3.19 muestra lo anterior.

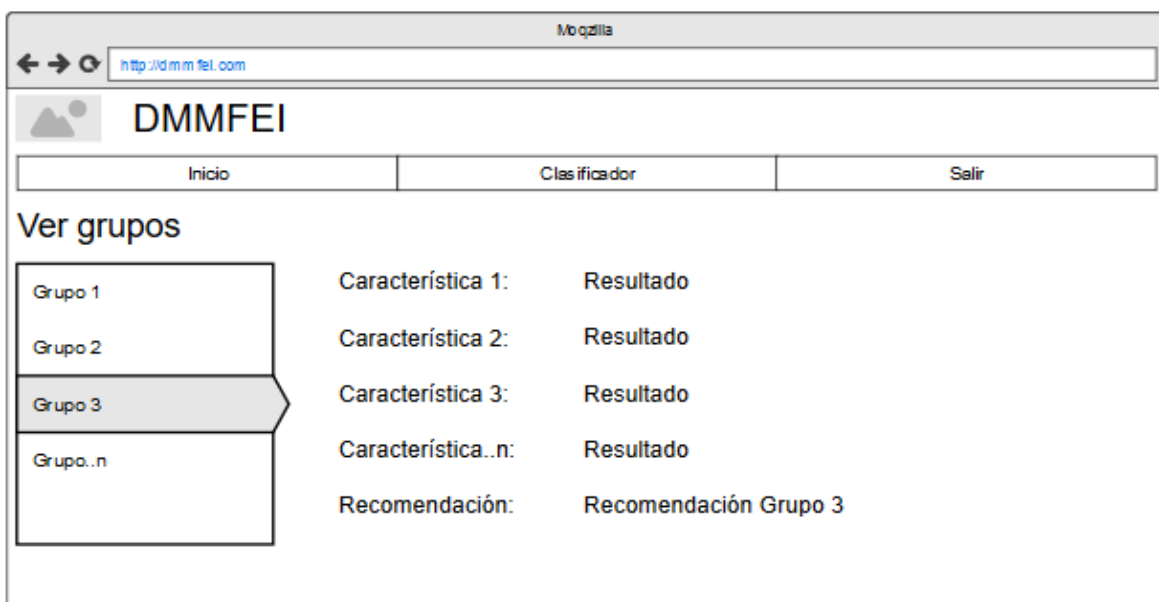


Figura 3.19 Vista de grupos.

3.1.3.4.2 Vista clasificar

La clasificación de los clientes es un proceso que el usuario clasificador efectúa, ya sea de forma simple o múltiple. Los clientes deben tipificarse se muestran de forma tabular para su mejor manejo. La Figura 3.20 plasma lo anterior.

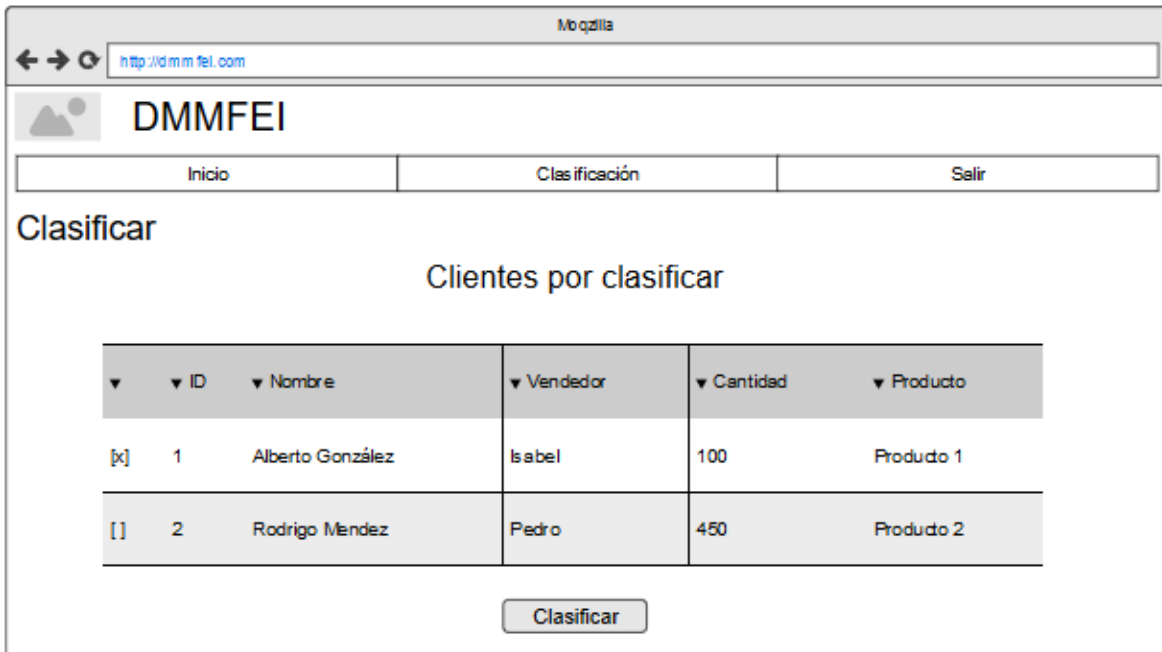


Figura 3.20 Vista para clasificar clientes.

3.1.3.4.3 Vista ver clasificaciones

Todo cliente que se clasificó correctamente, es buscado y encontrado en esta vista, por lo cual cuenta con un componente para búsqueda, y dado la naturaleza de los datos a buscar, los parámetros de búsqueda son por ID o por nombre del cliente. Los resultados se muestran de forma tabular para su mejor comprensión. La Figura 3.21 muestra lo anterior.

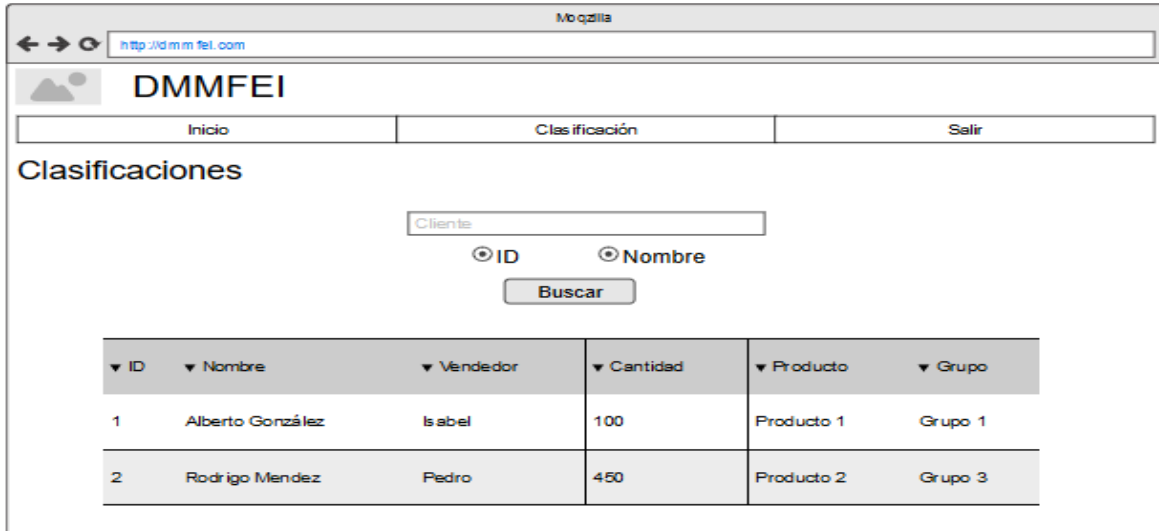


Figura 3.21 Vista clasificaciones.

3.1.3.5 Vistas usuario pronosticador

Este tipo de usuario tiene la capacidad de realizar dos tareas:

- **Pronosticar cliente:** Dado un cliente determinado, el objetivo de esta vista es mostrar el comportamiento de dicho cliente a lo largo de su estancia en la empresa así como también el pronóstico esperado.
- **Pronosticar ventas:** Con el fin de que la empresa se prepare para los días de mejores y peores ventas, el objetivo de esta vista es mostrar el comportamiento de las ventas de determinados productos que ésta oferte, así como también el pronóstico esperado.

Lo anterior se observa en la Figura 3.22.

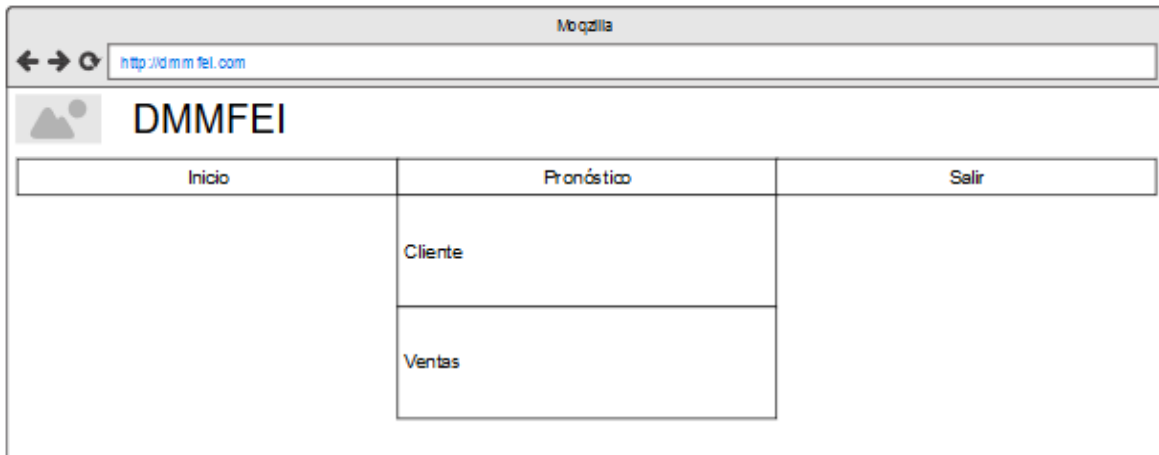


Figura 3.22 Opciones del usuario pronosticador.

3.1.3.5.1 Vista pronosticar cliente

Esta vista tiene como objetivo mostrar el comportamiento de un determinado cliente a través de su estancia en la empresa, tal objetivo se cumple con su consumo (compras) de productos que la empresa oferte. De tal forma que para su despliegue hacía el usuario pronosticador, los elementos que se requieren son: el cliente a buscar, el parámetro de búsqueda (Id o nombre), sobre qué producto se espera obtener un pronóstico de compra, y claro, el intervalo de tiempo a analizar. Los resultados se muestran de forma tabular, y al elegir el cliente a pronosticar, el sistema muestra el comportamiento del cliente y su pronóstico. La Figura 3.23 plasma lo anterior.

Inicio Pronóstico Salir

Pronosticar consumo de cliente

Cliente

ID Nombre

Producto 1

4/22/2012 Fecha inicio 4/22/2012 Fecha fin

Buscar

▼	▼ Id	▼ Cliente
<input checked="" type="radio"/>	1	Juan Pérez
<input type="radio"/>	2	Isabel Martínez
<input type="radio"/>	3	Rodolfo Luna

Seleccionar

Se espera que el cliente compre 1000 unidades

■ Cantidad comprada

Figura 3.23 Vista pronosticar cliente.

3.1.3.5.2 Vista pronosticar ventas

Esta vista tiene como objetivo mostrar las ventas de la empresa a lo largo de un periodo de tiempo, tal objetivo se cumple con las ventas de productos que la empresa oferte. De tal forma que para su despliegue hacía el usuario pronosticador, los elementos que se requieren son: el producto del que se espera obtener un pronóstico de venta, y el intervalo de tiempo a analizar. Los resultados se muestran de forma tabular, el sistema plasma el comportamiento de las ventas y su pronóstico. La Figura 3.24 plasma lo anterior.

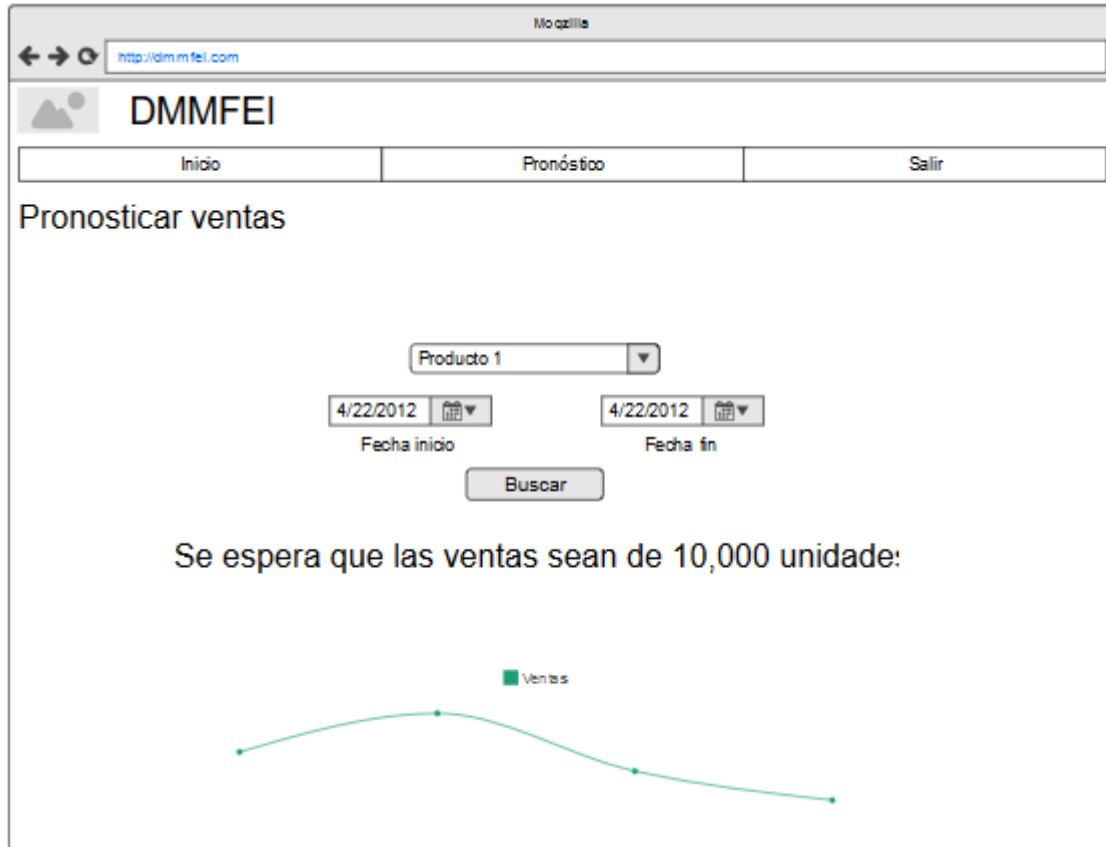


Figura 3.24 Vista pronosticar ventas.

3.1.4 Modelos conceptuales de la aplicación

En la sección anterior se definieron las vistas y sus componentes que funcionan como intermediarios entre el usuario final y la funcionalidad de la aplicación, en esta se presentan los modelos que determinan dicha funcionalidad mediante la cual, el sistema propuesto dará resolución a las necesidades de los usuarios.

Los modelos presentados en esta sección son:

- Diagrama de clases.
- Esquema de base de datos.

3.1.4.1 Diagrama de clases

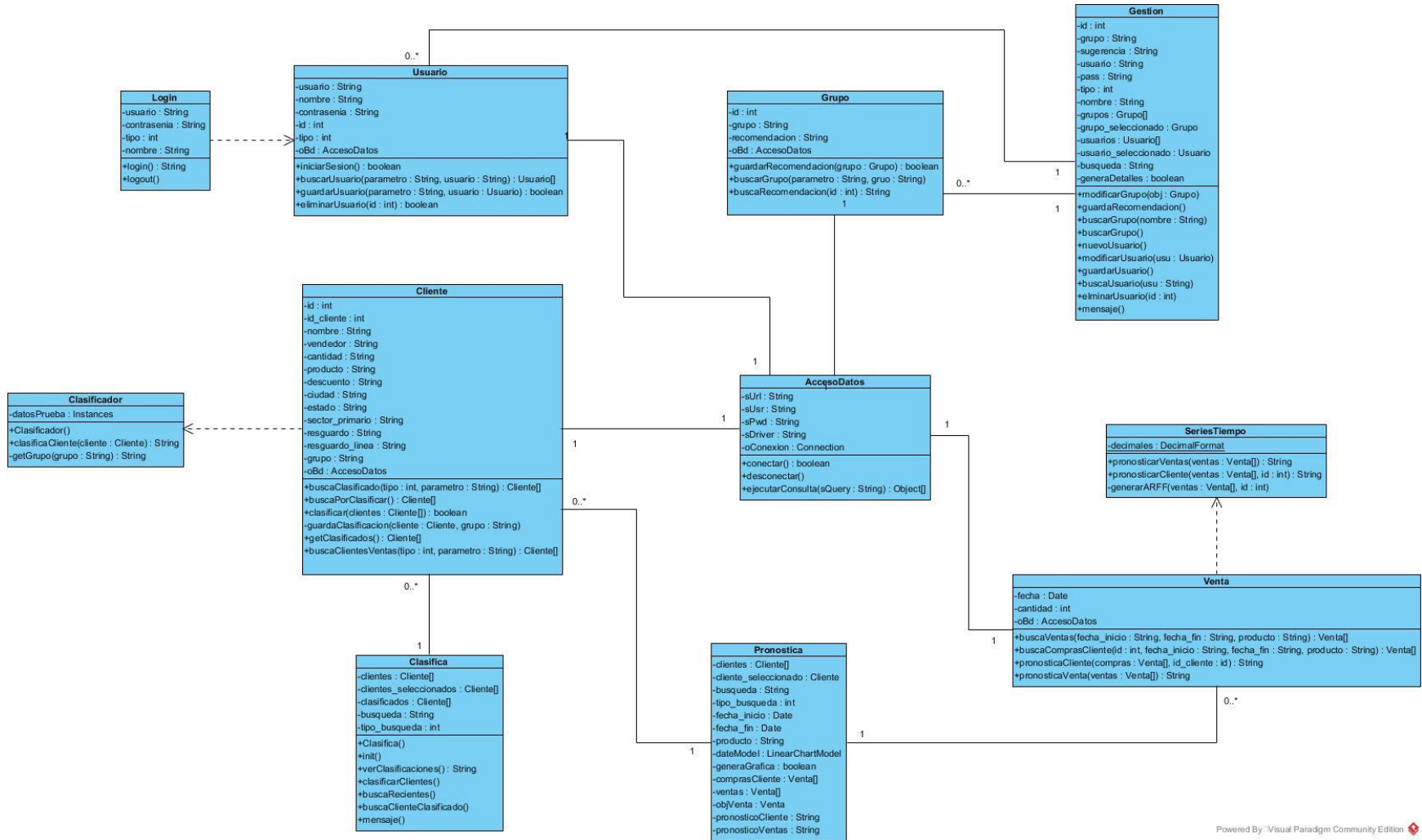


Figura 3.25 Diagrama de clases de la aplicación

La Figura 3.25 muestra la distribución e interacción de las clases en Java, las cuales realizan determinadas acciones comunicándose entre sí, ya sea para realizar una consulta a la base de datos, o efectuar algún proceso para su despliegue al usuario final.

3.1.4.2 Esquema de la base de datos

Dado que para la implementación de técnicas de minería de datos se requiere información almacenada, el esquema mostrado en la Figura 3.26 describe la estructura y relaciones de las tablas que una vez pobladas se utilizan para su análisis.



Figura 3.26 Esquema de la base de datos de la aplicación.

Existen dos tablas sin relación alguna: clasificar es la tabla en la que el sistema principal de SIFEI almacenará todos los clientes para clasificar y una vez que éstos se encuentren asignados a algún grupo, pasarán a formar parte de la tabla clasificados dejando a la tabla clasificar limpia y lista para recibir más clientes nuevos; por otro lado la tabla usuario almacena a los usuarios que podrán tener acceso a la aplicación.

3.1.5 Consideraciones del *Sprint 0*

Si bien en este *sprint* no se pretende la generación de algún módulo de software, más bien cimentar las bases para el sistema, hay algunas consideraciones que son de vital importancia mencionar.

Dado que se trata de una versión de la metodología **SCRUM** para una persona, la **Product Backlog** está representada mediante los casos de uso y sus descripciones mencionadas en secciones anteriores. De igual forma, algunos de los siguientes *sprints* no producirán necesariamente una parte del sistema propuesto debido a que su objetivo será la generación de modelos de minería de datos para su explotación (mediante interfaces de usuario).

Cabe mencionar que las primeras tres fases de *KDD* son iguales para esta sección, a continuación se explica el porqué.

- **Selección:** Para este trabajo de tesis los datos a utilizar provienen de un repositorio propio para el sistema propuesto, ya que por el momento solamente se cuentan con las estructuras de las bases de datos que la empresa utiliza, por lo tanto, los campos considerados para la aplicación de técnicas de minería de datos (*clustering*, clasificación y series de tiempo) son tomados como base para la generación de estructuras en nuestra base de datos (PostgreSQL). Dando como resultado que nuestro repositorio mantenga una estructura aislada del trabajo transaccional de la empresa, de tal forma que los datos almacenados en este repositorio son confiables para la implementación de técnicas de minería de datos. Por otro lado, la empresa se encargará de poblar las tablas definidas, dando como resultado datos consistentes y confiables. Como se menciona anteriormente los datos utilizados para este trabajo de tesis son datos representativos de ventas y clientes por parte de los ejecutivos de ventas de la empresa, de tal manera que al realizarse algún cambio en la manera de almacenar la información por parte de la empresa, se ve afectada la integridad y consistencia de los datos almacenados en nuestro repositorio.

- **Exploración:** Por lo anterior, esta fase ya no se lleva a cabo como normalmente se haría, dado que ya se tienen los datos ya se encuentran en nuestro repositorio.
- **Limpieza:** De igual forma que en el punto anterior, esta fase ya no se lleva a cabo como normalmente se haría, debido al hecho de que se generaron datos tomando en cuenta las estructuras del punto 1 y considerando el negocio de la empresa, la información generada ya mantiene consistencia y ausencia de datos faltantes.

Considerando lo anterior, para cada *sprint* que utilice el proceso *KDD* se iniciará desde la cuarta fase (**transformación**) según lo requiera la técnica de minería a utilizar.

Sin embargo al contar solamente con las estructuras de los repositorios que la empresa utiliza, se utilizó la herramienta en línea **Mockaroo**[31], para la generación de datos sintéticos.

3.2 Sprint 1 generar modelo de *clustering*

Con el fin de clasificar a un nuevo cliente cuando se requiera, el proceso para lograr ese cometido es en primera instancia contar con un conjunto de entrenamiento ya clasificado (grupo 1 hasta grupo n), sin embargo para el presente trabajo, dicho conjunto no existe por lo que se deberá generar no sin antes determinar la técnica más apropiada, a continuación se presenta el trabajo para la determinación del modelo óptimo para *clustering*.

Los algoritmos de minería de datos a utilizar serán:

- ***K-Means*.**
- **EM (*Expectation Maximisation, Expectativa de Maximización*).**

Para el uso de los algoritmos previamente mencionados, se seleccionaron tres métricas para su futura evaluación:

- **Homogeneidad:** Es la suma del error cuadrático, esta métrica se utiliza en el algoritmo *K-Means*. Esta se refiere a qué tan similares son los elementos de un grupo, entre menor sea esta métrica más confiable es el resultado.

- **Verosimilitud:** También conocida como *Log Likelihood* (Probabilidad de Registro). Se refiere a la probabilidad de que un elemento pertenezca a un grupo, entre menor sea el valor obtenido para esta métrica más confiable es el resultado. Lo anterior se utiliza en el algoritmo EM.
- **Tiempo de ejecución:** Se refiere al tiempo que le tomó al algoritmo crear los grupos.

Como se menciona en el *sprint 0*, los datos a utilizar ya se encuentran “limpios”, de tal forma que en este *sprint 1* se retomará el proceso *KDD* en la fase de **transformación**, debido a que estos algoritmos trabajan de mejor manera con datos de tipo numérico o continuos.

Las variables a ser consideradas para el agrupamiento se muestran en la Tabla 3.1.

Tabla 3.1 Variables consideradas para clustering.

VARIABLE	DESCRIPCIÓN	ESCALA	CALIFICACIÓN
VENDEDOR	Vendedor que atendió y finalizó la venta	Cualitativa	Editable
PRODUCTO	Producto o servicio que el vendedor logró comercializar	Cualitativa	Editable
CANTIDAD	Cantidad de productos o servicios vendidos	Cuantitativa	Número
DESCUENTO	Se refiere a si el cliente tuvo algún descuento en su compra	Cualitativa	True, False
CIUDAD	Ciudad hacia donde se vendió el producto o servicio	Cualitativa	Editable

VARIABLE	DESCRIPCIÓN	ESCALA	CALIFICACIÓN
ESTADO	Estado hacia donde se vendió el producto o servicio	Cualitativa	Editable
SECTOR_PRIMARIO	Se refiere a que si el cliente tiene la capacidad emitir facturas de este sector	Cualitativa	True, False
RESGUARDO	Se refiere a que si el producto o servicio permite el resguardo local de información	Cualitativa	True, False
RESGUARDO_LINEA	Se refiere a que si el producto o servicio permite resguardo en línea de información	Cualitativa	True, False

El primer paso para realizar la transformación de los datos, es determinar cuáles variables no son de tipo numérico, dando como resultado los valores mostrados en la Tabla 3.2.

Tabla 3.2 Tipo de dato de cada variable.

VARIABLE	TIPO DE DATO
VENDEDOR	VARCHAR
PRODUCTO	VARCHAR
DESCUENTO	BOOLEAN

VARIABLE	TIPO DE DATO
CIUDAD	VARCHAR
ESTADO	VARCHAR
SECTOR_PRIMARIO	BOOLEAN
RESGUARDO	BOOLEAN
RESGUARDO_LINEA	BOOLEAN

Una vez identificadas las variables que no cumplen con el requisito, se procedió a transformarlas. Cabe mencionar que para el tipo de dato Booleano, desde la consulta para obtener los valores efectuó una conversión de BOOLEAN a INTEGER (el repositorio de datos es PostgreSQL) de tal forma que si en el registro presenta True su representación a entero es 1, mientras que si presenta False su representación es 0, esto es de gran ayuda para convertir tipos de datos desde la consulta. Lo siguiente es transformar las variables restantes, la Tabla 3.3 muestra el resultado final de la conversión.

Tabla 3.3 Variables ya transformadas

VARIABLE	ESCALA	CALIFICACIÓN
VENDEDOR 1 ... VENDEDOR 10	Cuantitativa	0, 1
PRODUCTO 1 ... PRODUCTO 3	Cuantitativa	0, 1
CANTIDAD	Cuantitativa	Número
DESCUENTO	Cuantitativa	0, 1
CIUDAD 1 ... CIUDAD 10	Cuantitativa	0, 1
ESTADO 1 ... ESTADO 3	Cuantitativa	0, 1

VARIABLE	ESCALA	CALIFICACIÓN
SECTOR_PRIMARIO	Cuantitativa	0, 1
RESGUARDO	Cuantitativa	0, 1
RESGUARDO_LINEA	Cuantitativa	0, 1

Una vez teniendo la información ya transformada y lista, se procede darle uso en la implementación de algoritmos de minería de datos para *clustering*.

3.3.1 Algoritmo *K-Means*

Al final se obtuvieron 4500 registros con 31 atributos (debido a la transformación de datos). El primer algoritmo a utilizar es *K-Means* tanto con distancia Euclidiana como con distancia Manhattan, los resultados se muestran en las Tablas 3.4 a 3.7.

Tabla 3.4 Resultados utilizando distancia Euclidiana con números pares.

NÚMERO DE GRUPOS	ERROR CUADRÁTICO	TIEMPO DE EJECUCIÓN
2	15044.187078	0.05s
4	12457.173768	0.11s
6	11412.656507	0.14s
8	10657.246148	0.13s
10	10358.753237	0.42s
12	10010.596992	0.19s
14	9759.324400	0.14s
16	9478.575392	0.3s

NÚMERO DE GRUPOS	ERROR CUADRÁTICO	TIEMPO DE EJECUCIÓN
18	9197.413189	0.3s
20	8915.142130	0.59s

Tabla 3.5 Resultados utilizando distancia Euclidiana con números impares.

NÚMERO DE GRUPOS	ERROR CUADRÁTICO	TIEMPO DE EJECUCIÓN
3	14291.143953	0.04s
5	12141.084173	0.11s
7	10970.431202	0.16s
9	10376.325312	0.2s
11	10443.207996	0.17s
13	9819.057504	0.3s
15	9506.519889	0.25s
17	9238.827524	0.52s
19	8994.135930	0.55s

Tabla 3.6 Resultados utilizando distancia Manhattan con números pares.

NÚMERO DE GRUPOS	ERROR CUADRÁTICO	TIEMPO DE EJECUCIÓN
2	21445.570570	0.17s
4	18276.182182	0.11s
6	15783.288288	0.19s
8	15183.358358	0.14s
10	14828.417417	0.3s
12	14090.872872	0.19s
14	13511.815815	0.22s
16	12880.387387	0.3s
18	12764.476476	0.27s
20	12812.595595	0.3s

Tabla 3.7 Resultados utilizando distancia Manhattan con números impares.

NÚMERO DE GRUPOS	ERROR CUADRÁTICO	TIEMPO DE EJECUCIÓN
3	20084.587587	0.19s
5	17693.931931	0.13s
7	17081.595595	0.16s
9	14806.472472	0.2s
11	14775.119119	0.27s
13	13625.343343	0.17s
15	13373.779779	0.28s
17	12843.838838	0.25s
19	12938.373373	0.3s

Las Figuras 3.27 a 3.30 muestran la representación de las tablas anteriores en forma de gráficas.

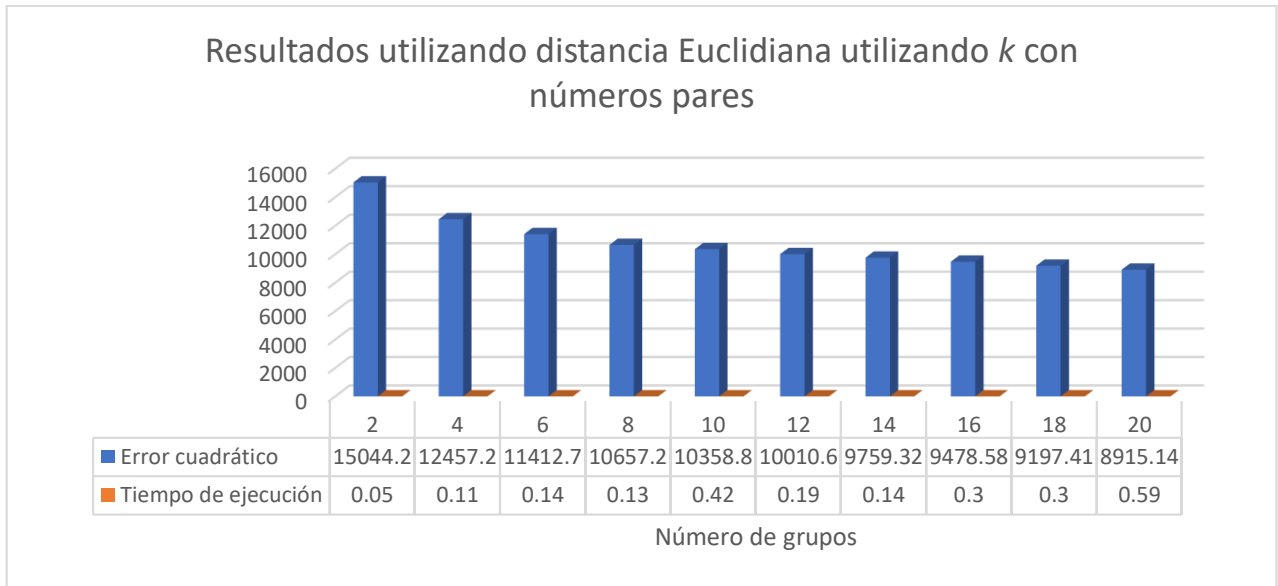


Figura 3.27 Gráfica de resultados utilizando distancia Euclidiana con números pares.

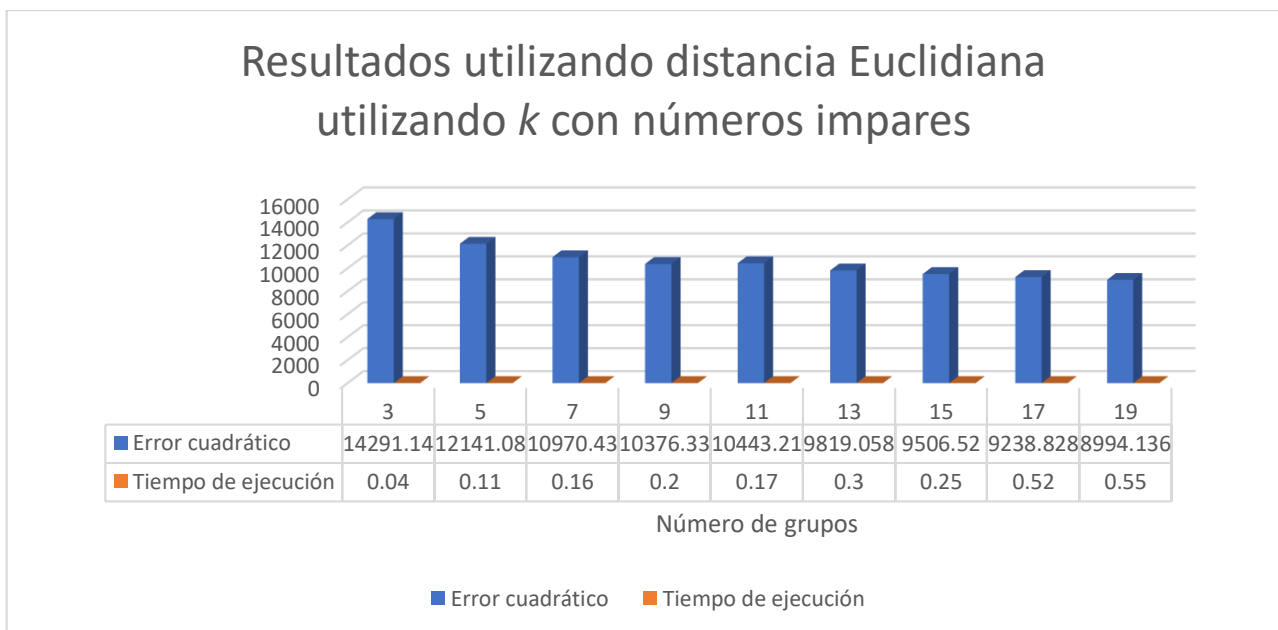


Figura 3.28 Gráfica de resultados utilizando distancia Euclidiana con números impares.

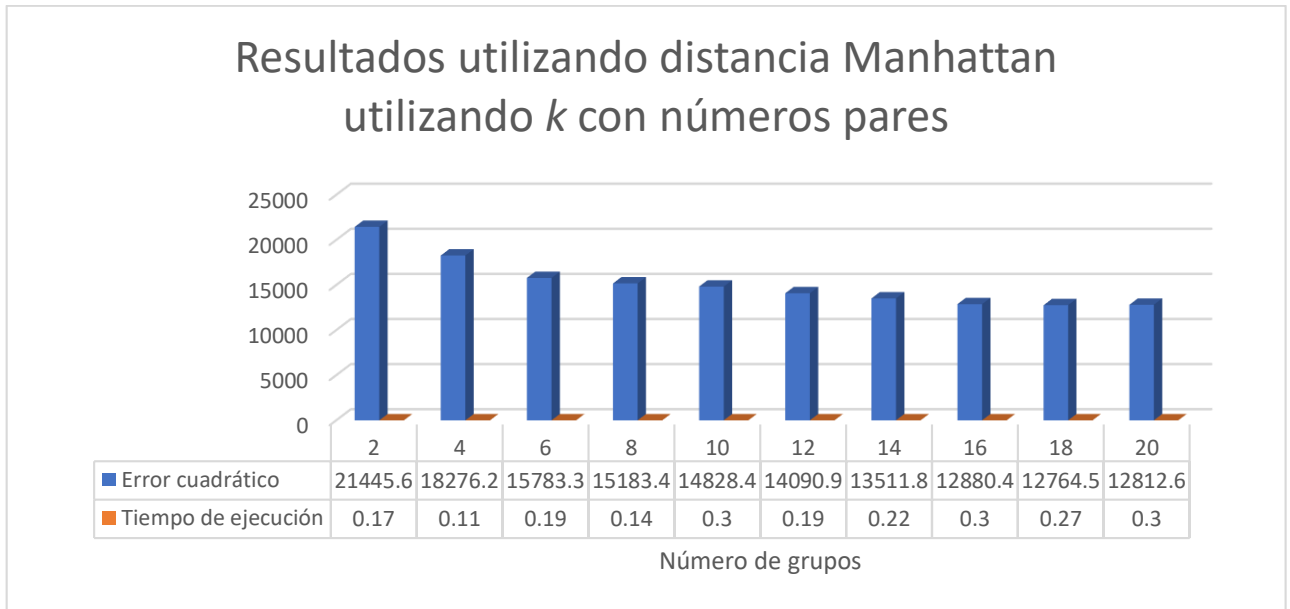


Figura 3.29 Gráfica de resultados utilizando distancia Manhattan con números pares.

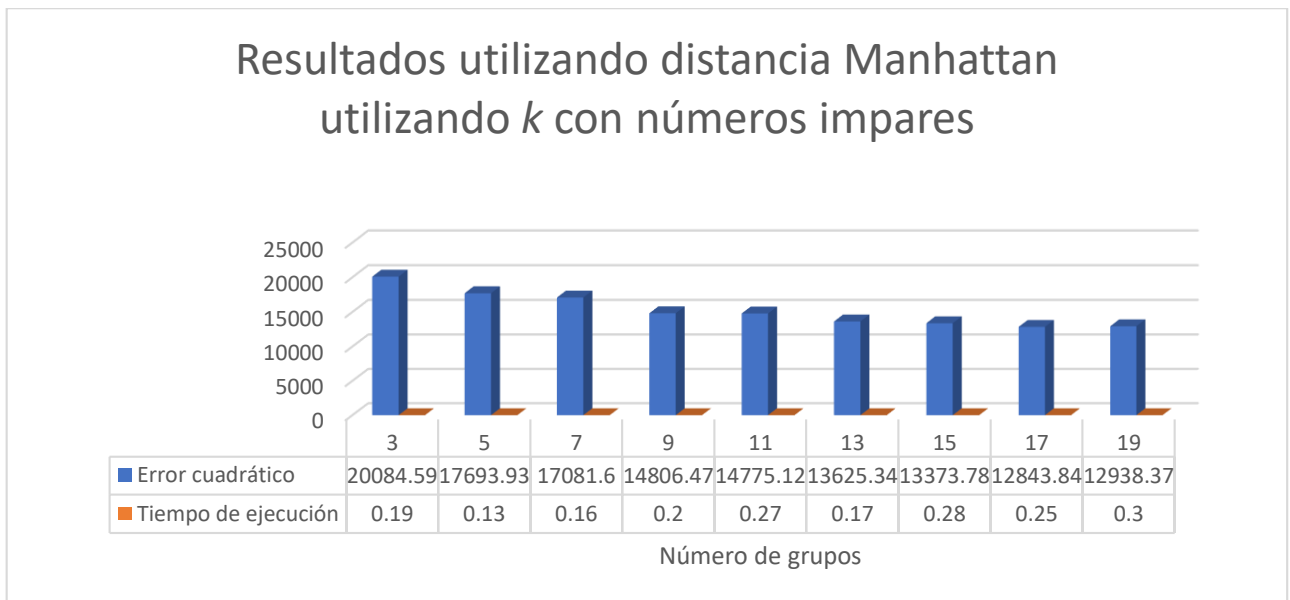


Figura 3.30 Gráfica de resultados utilizando distancia Manhattan con números impares.

Una vez realizadas las pruebas con el algoritmo *K-Means*, se procede a realizar las mismas pruebas con el algoritmo *EM*.

3.3.2 Algoritmo *EM*

Por otro lado se implementó el algoritmo *EM*, teniendo como particularidad que no se requiere asignar un número de grupos esperados como en el caso de *K-Means*, Sin embargo para fines comparativos se probará el algoritmo con 2, 4, 6, 8, 10, 12, 14, 16, 18 y 20 grupos. Cabe mencionar que las métricas a considerar para este algoritmo son verosimilitud y tiempo de ejecución. En la Tabla 3.8 se muestran los resultados obtenidos.

Tabla 3.8 Resultados del algoritmo *EM*.

NÚMERO DE GRUPOS	VEROSIMILITUD	TIEMPO DE EJECUCIÓN
2	-4.05794	0.55s
4	21.78357	0.97s
6	35.2923	1.59s
8	34.69194	2.23s
10	44.73032	2.83s
12	46.29149	4.56s
14	50.09967	3.41s
16	58.355	4.52s
18	56.43247	4.7s
20	56.7134	5.61s

De igual forma que la sección anterior, para una mejor visualización, los resultados de la tabla anterior se muestran en forma de gráfica, en la Figura 3.31 se muestran los resultados.

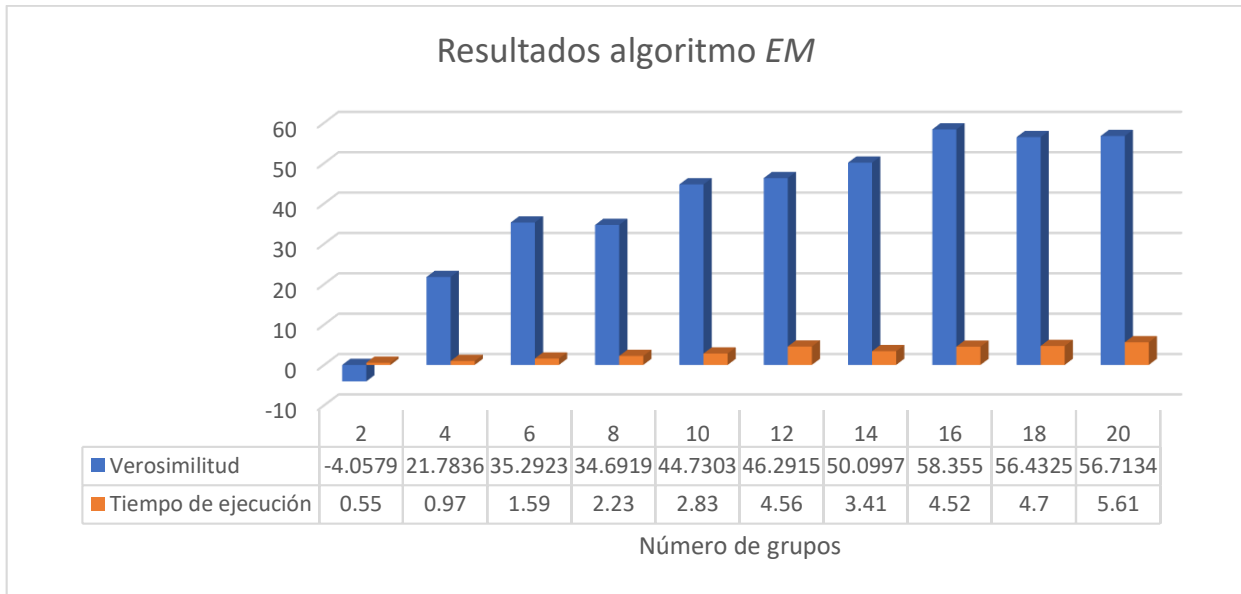


Figura 3.31 Gráfica de resultados del algoritmo EM.

En la figura anterior se aprecia que entre los números de grupos 6 y 10 existe una mejora en la verosimilitud, lo cual indica que el resultado más preciso se encuentra en ese rango, para comprobar lo anterior se implementó el algoritmo otra vez, pero dejando que él mismo determine el número óptimo de grupos. La Tabla 3.9 muestra los resultados obtenidos.

Tabla 3.9 Resultado obtenido automáticamente por EM.

NÚMERO DE GRUPOS	VEROSIMILITUD	TIEMPO DE EJECUCIÓN
8	34.69194	104.1s

Con la información obtenida se comprueba que el número óptimo de grupos es 8, debido a que la verosimilitud tiene su mejor valor en ese lugar.

3.3.3 Evaluación de los algoritmos para *clustering*

Siguiendo con el proceso *KDD*, la siguiente fase es la de **Evaluación**, para la realización de dicha fase se realiza un análisis de los algoritmos previamente implementados, considerando ciertos aspectos a valorar (métricas). En los puntos anteriores se probaron dos algoritmos en total: *K-Means* y *EM*, los valores que cada uno arrojó se presentaron de manera tabular y gráfica.

En el caso de *K-Means* existen dos distancias a evaluar, una es la distancia Euclidiana y la otra es la distancia Manhattan, de manera que para este caso, la distancia Euclidiana arroja mejores resultados con respecto a la distancia Manhattan. De tal manera que se seleccionó el algoritmo *K-Means* mediante el uso de la distancia Manhattan, dado que presenta una menor suma de error cuadrático que al utilizar como medida de distancia, la distancia Euclidiana.

Sin embargo la implementación de un algoritmo no se considera absoluta, ya que siempre es necesario tener diferentes puntos de vista, es por eso que se procedió a la aplicación de otro algoritmo llamado *EM*, en dicho algoritmo se presentaron sus resultados tomando en cuenta el número de grupos a formar de la misma manera que con el algoritmo *K-Means* (2, 4, 6... 20 grupos).

Mientras que para *K-Means* conforme se fuese aumentando el número de grupos a formar (k) la suma del error cuadrático disminuía, pero dichos grupos formados carecían de coherencia y legitimidad debido a que los grupos generados no estaban homogeneizados. Una particularidad en la implementación del algoritmo *EM* es que éste permite no definir el número de grupos a generar (k para *K-Means*), al contrario, permite dejar esa tarea al algoritmo en sí.

Al dejar que el algoritmo *EM* determinara el número óptimo de grupos, los resultados mantenían coherencia y se comprobaba la efectividad del algoritmo, ya que en los resultados al indicar el número de grupos a formar, se mostraba que en el rango de 6 – 10 grupos había una mejora en la métrica **verosimilitud**, de tal forma que al tener seis grupo la métrica tomaba el valor de **35.2923**, teniendo ocho grupos la métrica tomaba el valor de **34.69194**, y finalmente al contar con 10 grupos la métrica

tomaba el valor de **44.73032**, comprobando así que el número óptimo de grupos es ocho.

Tomando en consideración los resultados al utilizar el algoritmo *EM*, los cuales indican que el número óptimo de grupos a formar es ocho, se procedió a comparar los resultados del algoritmo *K-Means* con ese número de grupos. Utilizando la distancia Euclidiana el valor de la suma del error cuadrático es de **10657.246148**, mientras que al utilizar la distancia Manhattan el valor que toma la métrica es de **15183.358358**. Lo cual entonces indica que la mejor opción es utilizar el algoritmo *K-Means* definiendo el número de grupos (*k*) como ocho en conjunto con la distancia Euclidiana.

Y aunque se consideró la implementación del algoritmo *K-Means* con el valor de *k* de forma impar, los resultados fueron muy similares que al utilizar números pares, y de igual forma la distancia Euclidiana presentó mejores resultados respecto a la distancia Manhattan.

Se procedió a guardar el modelo de *clustering* añadiendo la etiqueta de clase *cluster* para su futura implementación para clasificar a un cliente nuevo.

3.3 Sprint 2 generar modelo de clasificación

Teniendo ya el modelo de *clustering*, éste se utiliza como conjunto de entrenamiento para los algoritmos de clasificación, la finalidad es comparar la precisión con la que estos algoritmos clasifican a un cliente en uno de los grupos encontrados en el *sprint* anterior.

Los algoritmos de minería de datos a utilizar serán:

- **C4.5 (J48 para el caso de Weka).**
- **Naive Bayes.**
- **ZeroR.**
- **Decision Table.**
- **Random Tree.**

Para el uso de los algoritmos previamente mencionados, se utilizaron dos métricas para su futura evaluación:

- **Kappa:** Determina el acuerdo de predicción con la clase verdadera, es decir, para una clase asignada 1.0 significa que la predicción de clase es verdadera (acuerdo completo).
- **Precisión:** Especifica el porcentaje de instancias que se clasificaron correctamente.

Como se menciona en el sprint 0, los datos a utilizar ya se encuentran “limpios”, y de igual forma en el *sprint* 1 se generó el modelo de *clustering* para la implementación de este tipo de algoritmos, por lo cual en este *sprint* se transformarán siempre y cuando el algoritmo a utilizar lo requiera.

El conjunto contiene un total de 4500 instancias, 31 atributos (debido a la transformación en el *sprint* anterior) y ocho etiquetas de clase. De tal forma que la variable clase que se desea predecir es la variable *cluster*, la cual indica el grupo al que pertenece un cliente.

Todos los algoritmos serán implementados al mismo conjunto de datos, con validación cruzada de 10 pliegues.

La Tabla 3.10 muestra la distribución de los clientes en los grupos formados anteriormente.

Tabla 3.10 Distribución de clientes acorde a los grupos encontrados.

ETIQUETA DE CLASE	DISTRIBUCIÓN	CARACTERÍSTICAS
CLUSTER0	10.86%	<ul style="list-style-type: none"> • Vendedor 2 es el preferido. • Producto 3 es el preferido. • Alrededor de 52 unidades compradas. • 49% con descuento. • La mayoría son de la ciudad 4. • 52% son del estado 1. • 50% son de sector primario. • 51% resguardan información localmente. • 52% resguardan información en la Web.
CLUSTER1	11.04%	<ul style="list-style-type: none"> • Vendedor 6 es el preferido. • Producto 1 es el preferido. • Alrededor de 512 unidades compradas. • Todos recibieron descuento. • 51% son de la ciudad 6. • Todos son del estado 2. • 50% pertenecen al sector primario. • Ninguno resguarda localmente su información. • Ninguno resguarda en línea su información.
CLUSTER2	11.35%	<ul style="list-style-type: none"> • Vendedor 7 es el preferido. • Producto 2 es el preferido. • Alrededor de 50 unidades compradas. • 48% recibieron descuento. • 14% son de la ciudad 7. • 52% son del estado 3. • 50% pertenecen al sector primario. • 48% resguarda localmente su información. • 51% resguarda en línea su información.
CLUSTER3	22.22%	<ul style="list-style-type: none"> • Vendedores 1, 4, 5, 6, 8 abarcan este grupo. • Producto 1 es el preferido.

ETIQUETA DE CLASE	DISTRIBUCIÓN	CARACTERÍSTICAS
		<ul style="list-style-type: none"> • Alrededor de 493 unidades compradas. • 50% recibieron descuento. • Ciudades 1, 2, 3 y 4 con alrededor de 20%. • Todos son del estado 1. • 49% pertenecen al sector primario. • Ninguno resguarda localmente su información. • Ninguno resguarda en línea su información.
CLUSTER4	11.17%	<ul style="list-style-type: none"> • Vendedor 4 es el preferido. • Producto 1 es el preferido. • Alrededor de 474 unidades compradas. • Nadie recibió descuento. • 50% son de la ciudad 6 y 49% de la ciudad 5. • Todos son del estado 2. • 51% pertenecen al sector primario. • Ninguno resguarda localmente su información. • Ninguno resguarda en línea su información.
CLUSTER5	5.77%	<ul style="list-style-type: none"> • Vendedor 10 es el preferido. • Producto 2 es el preferido. • Alrededor de 49 unidades compradas. • El 50% recibieron descuento. • 50% son de la ciudad 6 y el otro 50% de la ciudad 5. • Todos son del estado 2. • 55% pertenecen al sector primario. • 49% resguarda localmente su información. • 48% resguarda en línea su información.
CLUSTER6	5.33%	<ul style="list-style-type: none"> • Vendedor 4 es el preferido. • Producto 3 es el preferido. • Alrededor de 46 unidades compradas.

ETIQUETA DE CLASE	DISTRIBUCIÓN	CARACTERÍSTICAS
CLUSTER7	22.22%	<ul style="list-style-type: none"> • 54% recibieron descuento. • 51% son de la ciudad 6 y el 49% de la ciudad 5. • Todos son del estado 2. • 53% pertenecen al sector primario. • 50% resguarda localmente su información. • 50% resguarda en línea su información. • Vendedor 1 es el preferido. • Producto 1 es el preferido. • Alrededor de 500 unidades compradas. • 47% recibieron descuento. • Ciudades 7, 8, 9 y 10 con alrededor de 25%. • Todos son del estado 3. • 49% pertenecen al sector primario. • Ninguno resguarda localmente su información. • Ninguno resguarda en línea su información.

3.3.1 Algoritmo C4.5 (J48)

El primer algoritmo a utilizar es *C4.5 (J48)* para el caso de *Weka*. Para la implementación de dicho algoritmo no se requirió de transformación alguna hacia los datos. La Tabla 3.11 muestra los resultados obtenidos para las métricas a evaluar.

Tabla 3.11 Resultados del algoritmo C4.5.

ALGORITMO	KAPPA	PRECISIÓN
C4.5	1.0	100%

En la Tabla 3.12 se muestra la matriz de confusión para el algoritmo C4.5.

Tabla 3.12 Matriz de confusión del algoritmo C4.5.

A	B	C	D	E	F	G	H	
489	0	0	0	0	0	0	0	a = cluster 0
0	497	0	0	0	0	0	0	b = cluster 1
0	0	511	0	0	0	0	0	c = cluster 2
0	0	0	1000	0	0	0	0	d = cluster 3
0	0	0	0	503	0	0	0	e = cluster 4
0	0	0	0	0	260	0	0	f = cluster 5
0	0	0	0	0	0	240	0	g = cluster 6
0	0	0	0	0	0	0	1000	h = cluster 7

Como se observa en las tablas anteriores, el algoritmo C4.5 obtuvo un 100% de precisión y 1.0 de *Kappa*, de tal forma que clasificó correctamente todas las instancias.

3.3.2 Algoritmo Naive Bayes

De igual forma que con el algoritmo C4.5 no se requirió de ninguna transformación por parte de los datos para su implementación, los resultados se muestran en la Tabla 3.13.

Tabla 3.13 Resultados del algoritmo Naive Bayes.

ALGORITMO	KAPPA	PRECISIÓN
NAIVE BAYES	1.0	100%

En la Tabla 3.14 se muestra la matriz de confusión para el algoritmo Naive Bayes.

Tabla 3.14 Matriz de confusión del algoritmo Naive Bayes.

A	B	C	D	E	F	G	H	
489	0	0	0	0	0	0	0	a = cluster 0
0	497	0	0	0	0	0	0	b = cluster 1
0	0	511	0	0	0	0	0	c = cluster 2
0	0	0	1000	0	0	0	0	d = cluster 3
0	0	0	0	503	0	0	0	e = cluster 4
0	0	0	0	0	260	0	0	f = cluster 5
0	0	0	0	0	0	240	0	g = cluster 6
0	0	0	0	0	0	0	1000	h = cluster 7

Como se observa en las tablas anteriores, el algoritmo Naive Bayes, al igual que el algoritmo C4.5 obtuvo un 100% de precisión y 1.0 de *Kappa*, de tal forma que clasificó correctamente todas las instancias.

3.3.3 Algoritmo ZeroR

Para la implementación del algoritmo *ZeroR* no se requirió de ninguna transformación por parte de los datos, los resultados se muestran en la Tabla 3.15.

Tabla 3.15 Resultados del algoritmo ZeroR.

ALGORITMO	KAPPA	PRECISIÓN
ZEROR	0	22.22%

En la Tabla 3.16 se muestra la matriz de confusión para el algoritmo ZeroR.

Tabla 3.16 Matriz de confusión del algoritmo ZeroR.

A	B	C	D	E	F	G	H	
0	0	0	489	0	0	0	0	a = cluster 0
0	0	0	497	0	0	0	0	b = cluster 1
0	0	0	511	0	0	0	0	c = cluster 2
0	0	0	1000	0	0	0	0	d = cluster 3
0	0	0	503	0	0	0	0	e = cluster 4
0	0	0	260	0	0	0	0	f = cluster 5
0	0	0	240	0	0	0	0	g = cluster 6
0	0	0	1000	0	0	0	0	h = cluster 7

Como se observa, el algoritmo *ZeroR* tuvo el peor desempeño, debido a que clasificó todas las instancias en un solo grupo, obteniendo como resultado para *Kappa* de 0 lo que significa que la predicción de la clase no es confiable, en comparación con otros algoritmos, *ZeroR* presenta una precisión del 22.22% muy por debajo de los dos algoritmos anteriores.

3.3.4 Algoritmo *Decision Table*

De igual forma que con los algoritmos anteriores, no se requirió de ninguna transformación por parte de los datos para su implementación, los resultados se muestran en la Tabla 3.17.

Tabla 3.17 Resultados del algoritmo Decision Table.

ALGORITMO	KAPPA	PRECISIÓN
DECISION TABLE	1.0	100%

En la Tabla 3.18 se muestra la matriz de confusión para el algoritmo *Decision Table*.

Tabla 3.18 Matriz de confusión del algoritmo Decision Table.

A	B	C	D	E	F	G	H	
489	0	0	0	0	0	0	0	a = cluster 0
0	497	0	0	0	0	0	0	b = cluster 1
0	0	511	0	0	0	0	0	c = cluster 2
0	0	0	1000	0	0	0	0	d = cluster 3
0	0	0	0	503	0	0	0	e = cluster 4
0	0	0	0	0	260	0	0	f = cluster 5
0	0	0	0	0	0	240	0	g = cluster 6
0	0	0	0	0	0	0	1000	h = cluster 7

Como se observa en las tablas anteriores, el algoritmo *Decision Table*, al igual que los primeros dos algoritmos obtuvo un 100% de precisión y 1.0 de *Kappa*, de tal forma que clasificó correctamente todas las instancias.

3.3.5 Algoritmo *Random Tree*

De igual forma que con los algoritmos anteriores no se requirió de ninguna transformación por parte de los datos para su implementación, los resultados se muestran en la Tabla 3.19.

Tabla 3.19 Resultados del algoritmo *Random Tree*.

ALGORITMO	KAPPA	PRECISIÓN
RANDOM TREE	0.98	98.93%

En la Tabla 3.20 se muestra la matriz de confusión para el algoritmo *Random Tree*.

Tabla 3.20 Matriz de confusión del algoritmo *Random Tree*.

A	B	C	D	E	F	G	H	
481	0	7	1	0	0	0	0	a = cluster 0
0	486	0	0	10	0	1	0	b = cluster 1
7	0	500	1	0	1	0	2	c = cluster 2
1	0	1	998	0	0	0	0	d = cluster 3
0	8	0	0	493	0	2	0	e = cluster 4
0	0	0	0	0	260	0	0	f = cluster 5
0	2	0	0	1	2	235	0	g = cluster 6
0	0	1	0	0	0	0	999	h = cluster 7

Como se observa en las tablas anteriores, el algoritmo *Random Tree* presenta resultados distintos a los algoritmos anteriores, teniendo como *Kappa* 0.98 y precisión en un 98.93% de tal forma que de las 4500 instancias solamente 4452 se

clasificaron correctamente, mientras que 48 instancias se clasificaron incorrectamente. Los resultados son interpretados de la siguiente manera:

- En el *cluster* 0, se clasificaron incorrectamente siete instancias hacía el *cluster* 2 y una instancia hacía el *cluster* 3.
- En el caso del *cluster* 1, se clasificaron incorrectamente 10 instancias hacía el *cluster* 4 y una hacía el *cluster* 6.
- Para el *cluster* 2, se clasificaron incorrectamente siete instancias hacía el *cluster* 0, una instancia hacía el *cluster* 3, una instancia hacía el *cluster* 5 y dos instancias hacía el *cluster* 7.
- En el *cluster* 3, se clasificaron incorrectamente una instancia hacía el *cluster* 0 y una instancia más hacía el *cluster* 2.
- Para el *cluster* 4, se clasificaron incorrectamente ocho instancias hacía el *cluster* 1 y dos instancias hacía el *cluster* 6.
- En el caso del *cluster* 5, ninguna instancia se clasificó mal, siendo éste *cluster* el único clasificado correctamente.
- En el *cluster* 6, se clasificaron incorrectamente dos instancias hacía el *cluster* 1, una instancia al *cluster* 4 y dos instancias hacía el *cluster* 5.
- Por último en el *cluster* 7, se clasificó incorrectamente una instancia hacía el *cluster* 2.

3.3.6 Evaluación de los algoritmos para clasificación

Continuando con el proceso *KDD*, la siguiente fase es la de **Evaluación**, para la realización de dicha fase se realiza un análisis de los algoritmos previamente implementados, considerando ciertos aspectos a valorar (métricas). En los puntos anteriores se implementaron cinco algoritmos en total: *C4.5*, *Naive Bayes*, *ZeroR*, *Decision Table* y *Random Tree*, los valores que cada uno arrojó se presentan de manera tabular, sin embargo para su evaluación la Tabla 3.21 muestra los resultados resumidos.

Tabla 3.21 Resultados de los algoritmos de clasificación.

ALGORITMO	KAPPA	PRECISIÓN
C4.5	1.0	100%
NAIVE BAYES	1.0	100%
ZERO R	0	22.22%
DECISION TABLE	1.0	100%
RANDOM TREE	0.98	98.93%

Como se observa en la tabla anterior, *ZeroR* presenta el peor desempeño de los algoritmos evaluados, por esa razón queda descartado para su implementación al igual que el algoritmo *Random Tree* debido a su baja precisión respecto a los demás.

Ya sea que se elija *C4.5*, *Naive Bayes* o *Decision Table*, cualquier implementación será correcta, para fines de este trabajo se implementa el algoritmo *C4.5* (J48).

3.4 Sprint 3 tipificar cliente

Dado que el modelo de *clustering* para la detección de grupos de clientes será cerrado, la implementación del algoritmo *C4.5* (fase de **Difusión** de *KDD*) se realizará mediante el sistema propuesto que para fines de este *sprint* se limita a la realización de las interfaces (páginas *Web*) y funcionalidades necesarias para la tipificación de un cliente.

Tomando en consideración lo descrito en el *sprint 0*, aquí se desarrolla todo lo referente al usuario de tipo clasificador.

3.4.1 Interfaces

En esta sección se mostrarán las interfaces necesarias del usuario clasificador.

3.4.1.1 Interfaz ver grupos

La interfaz encargada de proporcionarle al usuario clasificador las características principales de cada grupo encontrado, presenta la información de manera textual dado que es una página de consulta más no de interacción. La Figura 3.32 muestra la interfaz preliminar.



Figura 3.32 Interfaz ver grupos.

Como se observa para esa vista se presentan los ocho grupos encontrados en el *sprint 1*, sin embargo aún no se muestran los valores esperados para cada característica. Cabe mencionar que el menú será el mismo para cada usuario (exceptuando las acciones que realizan en particular) en este caso tiene un botón que accede a la interfaz de inicio y otro botón encargado de terminar la sesión de trabajo, la sección “Clasificación” es propia de este tipo de usuario.

3.4.1.2 Interfaz clasificar

La interfaz encargada de mostrar al usuario clasificador una lista de clientes sin tipificar se muestra en la Figura 3.33.

Clasificar

Cientes por clasificar

(1 of 1)

	Id	Nombre	Vendedor	Cantidad	Producto	Ciudad	Estado
Sin resultados							

(1 of 1)

Figura 3.33 Interfaz clasificar.

Como se observa en la figura anterior, la interfaz presenta una tabla que muestra al usuario clasificador los clientes pendientes de asignación a alguno de los ocho grupos. La información a mostrar consta de: Id de cliente, nombre del cliente, el vendedor que lo atendió, la cantidad de producto comprado, el producto comprado, y por último la ciudad y estado del cliente.

3.4.1.3 Interfaz ver clasificaciones

La interfaz encargada de mostrar al usuario clasificador a los clientes ya tipificados se muestra en la Figura 3.34.

Clasificaciones

Cliente: *

Tipo de búsqueda: * ID Nombre

Cientes clasificados

(1 of 1)

Id	Nombre	Vendedor	Cantidad	Producto	Ciudad	Estado	Grupo
Sin resultados							

(1 of 1)

Figura 3.34 Interfaz ver clasificaciones.

En la interfaz “ver clasificaciones” se muestra un pequeño formulario para la búsqueda de un cliente en específico, el tipo de búsqueda (por ID o nombre) y un botón para realizar la petición. En la parte de abajo se presenta una tabla en la cual se desplegarán los resultados obtenidos y en la última columna de dicha tabla se encuentra el grupo al cual el cliente pertenece.

3.4.2 Funcionalidades

En esta sección se presentan las partes más relevantes de la funcionalidad de la aplicación (código) para el correcto funcionamiento de las interfaces previamente mostradas.

3.4.2.1 Ver grupos

La forma en que se obtienen las recomendaciones para cada grupo, es realizar una petición a la base de datos, dicha petición se realiza cuando el usuario clasificador ingresa a la página *Web* por primera vez, el Listado 3.1 muestra el método encargado de lo anterior.

```

1 public void init(){
2     rec1 = oGrupo.buscaRecomendacion(1);
3     rec2 = oGrupo.buscaRecomendacion(2);
4     rec3 = oGrupo.buscaRecomendacion(3);
5     rec4 = oGrupo.buscaRecomendacion(4);
6     rec5 = oGrupo.buscaRecomendacion(5);
7     rec6 = oGrupo.buscaRecomendacion(6);
8     rec7 = oGrupo.buscaRecomendacion(7);
9     rec8 = oGrupo.buscaRecomendacion(8);
10    mensaje("Actualizado", clientes.size() + " clientes por clasificar");
11 }

```

Listado 3.1 Método de inicialización.

Como se observa, el método recibe como parámetro el identificador del grupo a buscar su recomendación, el Listado 3.2 muestra la petición a la base de datos para extraer la recomendación del grupo deseado.

```

1 public String buscaRecomendacion(int grupo){
2     String resultado = "";
3     String sQuery = "SELECT sugerencia FROM grupo WHERE id_grupo = " + grupo + "";
4     ArrayList rst = null, arrLin = null;
5     int i = 0;
6     try{
7         oBd = new AccesoDatos();
8         if (oBd.conectar()){
9             rst = oBd.ejecutarConsulta(sQuery);
10            oBd.desconectar();
11        }
12    }catch(Exception ee){
13        System.out.println("Error de conexion");
14    }
15    try{
16        if (rst != null) {
17            for (i = 0; i < rst.size(); i++) {
18                arrLin = (ArrayList)rst.get(i);
19                resultado = (String)arrLin.get(0);
20            }
21        }
22    }catch(Exception e){
23        System.out.println(e);
24        System.out.println("Error de lectura");
25    }
26    return resultado;
27 }

```

Listado 3.2 Método buscaRecomendacion.

Se realiza la petición buscando la recomendación para el grupo indicado, devuelve el resultado, y por último se almacena en una variable para su posterior despliegue al usuario clasificador. La Figura 3.35 muestra el resultado.

Grupos de clientes

Grupo 1	Producto:	Producto 1
Grupo 2	Cantidad:	Baja
Grupo 3	Descuento:	Sí
Grupo 4	Ciudad:	Ciudad 1
Grupo 5	Estado:	Estado 1
Grupo 6	Sector primario:	No
Grupo 7	Resguardo:	No
Grupo 8	Resguardo en línea:	No
	Recomendación:	Ofrecer el Producto 1

Figura 3.35 Recomendación mostrada al usuario.

Las características principales de cada grupo se mantienen fijas dado la naturaleza del modelo, sin embargo la recomendación es obtenida de la base de datos, dicha recomendación el usuario administrador es quien se encarga de modificarla según le convenga.

3.4.2.2 Clasificar

La forma en que se obtienen a los clientes por clasificar, es realizar una petición a la base de datos, dicha petición se realiza cuando el usuario clasificador ingresa a la página *Web* por primera vez, el Listado 3.3 muestra el método encargado de lo anterior.

```
1 public void init(){
2     clientes = new Cliente().buscaPorClasificar();
3     rec1 = oGrupo.buscaRecomendacion(1);
4     rec2 = oGrupo.buscaRecomendacion(2);
5     rec3 = oGrupo.buscaRecomendacion(3);
6     rec4 = oGrupo.buscaRecomendacion(4);
7     rec5 = oGrupo.buscaRecomendacion(5);
8     rec6 = oGrupo.buscaRecomendacion(6);
9     rec7 = oGrupo.buscaRecomendacion(7);
10    rec8 = oGrupo.buscaRecomendacion(8);
11    mensaje("Actualizado", clientes.size() + " clientes por clasificar");
12 }
```

Listado 3.3 Búsqueda de clientes por clasificar.

Como se observa en el Listado anterior, la petición (línea número 2) se realiza al igual que la carga de recomendaciones para cada grupo y asignando a los clientes a la variable clientes, al finalizar se lanza un mensaje con la cantidad de clientes por clasificar (línea número 11). El Listado 3.4 muestra la petición a la base de datos para extraer a los clientes por clasificar.

```

1 public ArrayList<Cliente> buscaPorClasificar(){
2     ArrayList<Cliente> resultados = new ArrayList<>();
3     String sQuery = "SELECT id_clasificar, id_cliente, nombre, vendedor, cantidad, "
4         + "producto, descuento, ciudad, estado, "
5         + "sector_primario, resguardo, resguardo_linea FROM clasificar";
6     Cliente obj;
7     ArrayList rst = null, arrLin = null;
8     int i = 0;
9     try{
10         oBd = new AccesoDatos();
11         if (oBd.conectar()){
12             rst = oBd.ejecutarConsulta(sQuery);
13             oBd.desconectar();
14         }
15     }catch(Exception ee){
16         System.out.println("Error de conexion");
17     }
18     try{
19         if (rst != null) {
20             for (i = 0; i < rst.size(); i++) {
21                 obj = new Cliente();
22                 arrLin = (ArrayList)rst.get(i);
23                 obj.setId(((Double)arrLin.get(0)).intValue());
24                 obj.setId_cliente(((Double)arrLin.get(1)).intValue());
25                 obj.setNombre((String)arrLin.get(2));
26                 obj.setVendedor((String)arrLin.get(3));
27                 obj.setCantidad((String)arrLin.get(4));
28                 obj.setProducto((String)arrLin.get(5));
29                 obj.setDescuento((String)arrLin.get(6));
30                 obj.setCiudad((String)arrLin.get(7));
31                 obj.setEstado((String)arrLin.get(8));
32                 obj.setSector_primario((String)arrLin.get(9));
33                 obj.setResguardo((String)arrLin.get(10));
34                 obj.setResguardo_linea((String)arrLin.get(11));
35                 resultados.add(obj);
36             }
37         }
38     }catch(Exception e){
39         System.out.println(e);
40         System.out.println("Error de lectura");
41     }
42     return resultados;
43 }

```

Listado 3.4 Método buscaPorClasificar.

Una vez que los clientes pendientes de clasificación son extraídos de la base de datos, se despliegan al usuario clasificador en forma tabular. La Figura 3.36 muestra a los clientes por clasificar.

Clasificar

Clientes por clasificar <input type="text" value="φ"/>							
(1 of 1) << 1 >> 10							
<input type="checkbox"/>	Id	Nombre	Vendedor	Cantidad	Producto	Ciudad	Estado
<input type="checkbox"/>	19	Ciente 1	Vendedor1	Media	Producto1	Ciudad1	Estado1
<input type="checkbox"/>	20	Ciente 4	Vendedor1	Alta	Producto2	Ciudad6	Estado2
<input type="checkbox"/>	21	Ciente 6	Vendedor9	Baja	Producto2	Ciudad4	Estado1
<input type="checkbox"/>	22	Ciente 2	Vendedor2	Alta	Producto1	Ciudad2	Estado1
<input type="checkbox"/>	24	Ciente 5	Vendedor5	Alta	Producto1	Ciudad9	Estado3

(1 of 1) << 1 >> 10

Clasificar

Figura 3.36 Clientes por clasificar.

Cuando el usuario desea clasificar ya sea a uno o a varios clientes, los selecciona y oprime el botón “Clasificar” el sistema requiere de la confirmación, como se muestra en la Figura 3.37.



Figura 3.37 Confirmación para la clasificación de clientes.

Cuando el usuario lo confirma, el sistema procede a enviar a los clientes seleccionados para su clasificación, como se muestra en el Listado 3.5 en donde verifica que se haya seleccionado al menos a un cliente.


```
1 public void clasificarClientes(){
2     if(clientes_seleccionados.size() <= 0){
3         mensaje("Error", "Ningún cliente seleccionado");
4     }else{
5         if(new Cliente().clasificar(clientes_seleccionados)){
6             mensaje("Correcto", "Clientes clasificados: "
7                 + clientes_seleccionados.size());
8             init();
9             mensaje("Correcto",
10                 "Resultados en la sección 'Ver clasificaciones'");
11         }else
12             mensaje("Error", "Error del módulo"
13                 + clientes_seleccionados.size());
14     }
15 }
```

Listado 3.5 Método clasificarClientes.

Al comprobar que se seleccionó al menos un cliente, se envía el arreglo de clientes a un método que su función es fungir como intermediario entre el método de la vista y el método que implementa las técnicas de minería de datos. El Listado 3.6 muestra el método intermediario.

```
1 public boolean clasificar(ArrayList<Cliente> clientes){
2     int cluster;
3     Clasificador clasificador;
4     if(clientes.isEmpty())
5         return false;
6     else{
7         clasificador = new Clasificador();
8         for (Cliente item : clientes) {
9             try {
10                cluster = clasificador.clasificaCliente(item);
11                guardaClasificacion(item, cluster);
12            } catch (Exception ex) {
13                System.out.println(ex.toString());
14                ex.printStackTrace();
15            }
16        }
17        return true;
18    }
19 }
```

Listado 3.6 Método clasificar.

De nueva cuenta se valida la cantidad de clientes a clasificar, y finalmente se envía al método encargado de su clasificación, cuando lo clasifica (devuelve el grupo al que pertenece) el cliente es enviado junto a su grupo para su almacenamiento en la base de datos. El Listado 3.7 muestra el proceso para clasificar un cliente.

```

1 public int clasificaCliente(Cliente cliente) throws Exception {
2     int resultado;
3     Classifier cl = null;
4     Instance porClasificar = null;
5     double valor;
6     //Llenar instancia
7     porClasificar = new DenseInstance(9);
8     porClasificar.setValue(datosPrueba.attribute(0), cliente.getVendedor());
9     porClasificar.setValue(datosPrueba.attribute(1), cliente.getProducto());
10    porClasificar.setValue(datosPrueba.attribute(2), cliente.getCantidad());
11    porClasificar.setValue(datosPrueba.attribute(3), cliente.getDescuento());
12    porClasificar.setValue(datosPrueba.attribute(4), cliente.getCiudad());
13    porClasificar.setValue(datosPrueba.attribute(5), cliente.getEstado());
14    porClasificar.setValue(datosPrueba.attribute(6),
15        cliente.getSector_primario());
16    porClasificar.setValue(datosPrueba.attribute(7), cliente.getResguardo());
17    porClasificar.setValue(datosPrueba.attribute(8), cliente.
18        getResguardo_linea());
19    datosPrueba.add(porClasificar);
20
21    cl = (Classifier) weka.core.SerializationHelper.read(
22        Clasificador.class.getResourceAsStream("j48Model.model"));
23
24    valor = cl.classifyInstance(datosPrueba.lastInstance());
25    resultado = getGrupo(datosPrueba.classAttribute().value((int) valor));
26    return resultado;
27 }

```

Listado 3.7 Método clasificaCliente.

Se genera una nueva instancia para su clasificación (línea 4), se le asignan los valores o parámetros (línea 8 a 18) y mediante las bibliotecas de *Weka* se invoca al clasificador (en este caso el árbol J48) para que determine el grupo al que pertenece (línea 21 y 22), devolviendo como resultado el grupo del cliente en cuestión (línea 24).

Una vez que ha realizado la clasificación y guardado el resultado (línea 25) el control regresa al usuario para que continúe con su trabajo ya sea clasificar a más clientes nuevos o determinar nuevas acciones mediante las recomendaciones que el usuario administrador genere.

3.4.2.3 Ver clasificaciones

Cuando el usuario clasificador desee ver a los clientes ya tipificados, selecciona la opción correspondiente y en primera instancia se le muestra un formulario para buscar a un cliente en específico y una tabla previamente llenada con los últimos 50 clientes clasificados, tal y como se muestra en la Figura 3.38.

Clasificaciones

Cliente: *

Tipo de búsqueda: * ID Nombre

Buscar

Clientes clasificados							
(1 of 1) < << 1 >> > 10							
Id	Nombre	Vendedor	Cantidad	Producto	Ciudad	Estado	Grupo
24	Cilente 5	Vendedor5	Alta	Producto1	Ciudad9	Estado3	Grupo 2
22	Cilente 2	Vendedor2	Alta	Producto1	Ciudad2	Estado1	Grupo 1
21	Cilente 6	Vendedor9	Baja	Producto2	Ciudad4	Estado1	Grupo 5
20	Cilente 4	Vendedor1	Alta	Producto2	Ciudad6	Estado2	Grupo 4
19	Cilente 1	Vendedor1	Media	Producto1	Ciudad1	Estado1	Grupo 1

(1 of 1) |< << 1 >> >| 10

Figura 3.38 Clasificaciones.

El método encargado de la búsqueda de los últimos 50 registros se muestra en el Listado 3.8.

```

1 public void buscaRecientes(){
2     clasificados = new Cliente().getClasificados();
3     mensaje("Actualizado", clasificados.size()
4         + " Cliente(s) encontrado(s)");
5 }

```

Listado 3.8 Método buscaRecientes.

Dicho método realiza una petición para que el sistema acceda a la base de datos, busque y regrese a los clientes encontrados. Tal y como se muestra en el Listado 3.9.

```

1 public ArrayList<Cliente> getClasificados(){
2     ArrayList<Cliente> resultados = new ArrayList<>();
3     String sQuery = "SELECT id_cliente, nombre, vendedor, cantidad, "
4         + "producto, ciudad, estado, grupo.grupo "
5         + "FROM clasificados JOIN grupo on grupo.id_grupo = "
6         + "clasificados.grupo ORDER BY id_clasificados DESC LIMIT 50";
7     Cliente obj;
8     ArrayList rst = null, arrLin = null;
9     int i = 0;
10    try{
11        oBd = new AccesoDatos();
12        if (oBd.conectar()){
13            rst = oBd.ejecutarConsulta(sQuery);
14            oBd.desconectar();
15        }
16    }catch(Exception ee){
17        System.out.println("Error de conexion");
18    }
19    try{
20        if (rst != null) {
21            for (i = 0; i < rst.size(); i++) {
22                obj = new Cliente();
23                arrLin = (ArrayList)rst.get(i);
24                obj.setId(((Double)arrLin.get(0)).intValue());
25                obj.setNombre((String)arrLin.get(1));
26                obj.setVendedor((String)arrLin.get(2));
27                obj.setCantidad((String)arrLin.get(3));
28                obj.setProducto((String)arrLin.get(4));
29                obj.setCiudad((String)arrLin.get(5));
30                obj.setEstado((String)arrLin.get(6));
31                obj.setGrupo((String)arrLin.get(7));
32                resultados.add(obj);
33            }
34        }
35    }catch(Exception e){
36        System.out.println(e);
37        System.out.println("Error de lectura");
38    }
39    return resultados;
40 }

```

Listado 3.9 Método getClasificados.

En el Listado anterior se muestra el proceso de obtención de los clientes recientemente clasificados, se realiza una petición a la base de datos (líneas 3 a 6 y 13) y los resultados encontrados se almacenan en un arreglo (líneas 21 a 33) que es devuelto (línea 39) y desplegado (Figura 3.45). Cuando el usuario requiera de un cliente en particular, la petición la procesa el método mostrado en el Listado 3.10.

```

1 public void buscaClienteClasificado(){
2     clasificados.clear();
3     if (tipo_búsqueda == 1) {
4         try {
5             Integer.parseInt(búsqueda);
6             clasificados = new Cliente().buscaClasificado(tipo_búsqueda,
7                 búsqueda);
8         } catch (NumberFormatException e) {
9             mensaje("Error", "ID debe ser numérico");
10        }
11    } else {
12        clasificados = new Cliente().buscaClasificado(tipo_búsqueda,
13            búsqueda);
14    }
15    if(clasificados.isEmpty())
16        mensaje("Sin resultados", "No se encontraron resultados");
17    else
18        mensaje("Correcto", clasificados.size()
19            + " Cliente(s) encontrado(s)");
20 }

```

Listado 3.10 Método buscaClienteClasificado.

Cuando realiza las consideraciones apropiadas para el criterio de búsqueda, se invoca el método encargado de buscar a un cliente clasificado en particular (línea 6 o 12) enviando como parámetros el tipo de búsqueda (por id o por nombre) y el nombre o el id según se elija el tipo de búsqueda. El Listado 3.11 muestra dicho método.

```

1 public ArrayList<Cliente> buscaClasificado(int tipo, String parametro){
2     ArrayList<Cliente> resultados = new ArrayList<>();
3     String sQuery = "";
4     if(tipo == 1){
5         sQuery = "SELECT id_cliente, nombre, vendedor, cantidad, producto, "
6                 + "ciudad, estado, grupo.grupo "
7                 + "FROM clasificados JOIN grupo on grupo.id_grupo "
8                 + "= clasificados.grupo WHERE id_cliente = " + parametro
9                 + " ORDER BY id_clasificados DESC";}
10    else{
11        sQuery = "SELECT id_cliente, nombre, vendedor, cantidad, producto, "
12                + "ciudad, estado, grupo.grupo "
13                + "FROM clasificados JOIN grupo on grupo.id_grupo "
14                + "= clasificados.grupo WHERE nombre LIKE '%" + parametro
15                + "%' ORDER BY id_clasificados DESC";}
16    Cliente obj;
17    ArrayList rst = null, arrLin = null;
18    int i = 0;
19    try{
20        oBd = new AccesoDatos();
21        if (oBd.conectar()){
22            rst = oBd.ejecutarConsulta(sQuery);
23            oBd.desconectar();
24        }
25    }catch(Exception ee){
26        System.out.println("Error de conexion");
27    }
28    try{
29        if (rst != null) {
30            for (i = 0; i < rst.size(); i++) {
31                obj = new Cliente();
32                arrLin = (ArrayList)rst.get(i);
33                obj.setId(((Double)arrLin.get(0)).intValue());
34                obj.setNombre((String)arrLin.get(1));
35                obj.setVendedor((String)arrLin.get(2));
36                obj.setCantidad((String)arrLin.get(3));
37                obj.setProducto((String)arrLin.get(4));
38                obj.setCiudad((String)arrLin.get(5));
39                obj.setEstado((String)arrLin.get(6));
40                obj.setGrupo((String)arrLin.get(7));
41                resultados.add(obj);
42            }
43        }
44    }catch(Exception e){
45        System.out.println(e);
46        System.out.println("Error de lectura");
47    }

```

Listado 3.11 Método buscaClasificado.

Ciertamente el tipo de búsqueda es lo más relevante para el resultado, dado que dependiendo del valor que tome es la consulta a ejecutar hacia la base de datos, una vez que se ejecuta dicha consulta, se procede a asignar cada resultado a un arreglo para devolverlo y acto seguido desplegarlo al usuario (Figura 3.38).

3.5 *Sprint 4* pronosticar ventas y consumo de cliente

Habiendo seleccionado el algoritmo apropiado para el pronóstico de ventas y consumo (Sección 2.3 de este documento de tesis) se procede con la generación de las interfaces y funcionalidades para su explotación.

En este *sprint* se lleva a cabo la realización de las interfaces necesarias para el usuario de tipo pronosticador así como las funcionalidades necesarias para cubrir sus necesidades.

3.5.1 Interfaces

En esta sección se muestran las interfaces necesarias para el usuario pronosticador.

3.5.1.1 Interfaz pronosticar ventas

La interfaz pronosticar ventas es la encargada de proporcionarle al usuario pronosticador los campos necesarios para acotar su búsqueda y proporcionarle un escenario que refleje el estado de las ventas de la empresa. La Figura 3.39 muestra la interfaz preliminar.

Pronosticar ventas

The image shows a web form titled "Pronosticar ventas". It contains three input fields: a dropdown menu for "Producto: *" with "Todos" selected, a text input for "Fecha inicio: *" and another for "Fecha Fin: *". Below these fields is a "Buscar" button.

Figura 3.39 Interfaz pronosticar ventas.

La Figura anterior presenta un formulario que se encarga de delimitar la búsqueda a realizar, dado que permite determinar el producto a buscar, así como también el

periodo de tiempo a evaluar. Por el momento no se presenta ninguna gráfica dado que ésta se generará con los resultados de las ventas que al usuario le interesen.

3.5.1.2 Interfaz pronosticar consumo de cliente

Esta interfaz se encarga a grandes rasgos de las mismas actividades que la interfaz anterior, con la diferencia de que se debe buscar y seleccionar el cliente al cual se le desee pronosticar su consumo. La Figura 3.40 presenta la interfaz antes mencionada.

Pronosticar consumo de cliente

Cliente: *

Tipo de búsqueda: * ID Nombre

Producto: *

Fecha inicio: * Fecha Fin: *

Buscar

Clientes	
Id	Nombre
Sin resultados	

Seleccionar

Figura 3.40 Interfaz pronosticar consumo de cliente.

En la interfaz anterior, se presentan algunos campos adicionales, tales como el cliente a buscar y el tipo de búsqueda para encontrar a dicho cliente, sin olvidar el producto y el periodo de tiempo a evaluar. En la parte inferior se observa una tabla que será la encargada de desplegar al cliente o clientes encontrados bajo los parámetros de búsqueda.

3.5.2 Funcionalidades

En esta sección se presentan las partes más relevantes de la funcionalidad de la aplicación (código) para el correcto funcionamiento de las interfaces previamente mostradas.

3.5.2.1 Pronosticar ventas

El usuario que desee pronosticar las ventas sobre algún producto (o todos) ingresa a la vista correspondiente y completa la captura de los campos pulsando al final el

botón “Buscar” (Figura 3.39), una vez hecho eso, el sistema ejecuta un método que realiza las peticiones correspondientes. El Listado 3.12 presenta el método previamente mencionado.

```
1 public void pronosticarVenta(){
2     SimpleDateFormat formmat1 = new SimpleDateFormat("yyyy-MM-dd");
3     ventas = objVenta.buscaVentas(formmat1.format(fecha_inicio),
4         formmat1.format(fecha_fin), producto);
5     pronosticoVentas = objVenta.pronosticaVenta(ventas);
6     generaGraficaV(ventas);
7 }
```

Listado 3.12 Método pronosticarVenta.

Una vez que es invocado el método de la figura anterior, inicializa el formato de fecha esperado para la implementación de series de tiempo, acto seguido le asigna un arreglo de ventas a la variable correspondiente mediante la invocación del método “buscaVentas” que recibe como parámetros la fecha inicio, la fecha fin y el producto a buscar. El Listado 3.13 presenta el método “buscaVentas”.

```

1 public ArrayList<Venta> buscaVentas(String fecha_inicio, String fecha_fin, String producto){
2     String sQuery = "";
3     ArrayList<Venta> resultados = new ArrayList<>();
4     if(producto.equalsIgnoreCase("")){
5         sQuery = "SELECT fecha_compra as fecha, sum(cantidad_comprada) "
6                 + "as compra FROM ventas WHERE fecha_compra between '"
7                 + fecha_inicio + "' AND '" + fecha_fin + "' "
8                 + "GROUP BY fecha ORDER BY fecha ASC"; }
9     else{
10        sQuery = "SELECT fecha_compra as fecha, sum(cantidad_comprada) "
11                + "as compra FROM ventas WHERE producto = '"
12                + producto + "' AND fecha_compra"
13                + " between '" + fecha_inicio + "' AND '" + fecha_fin
14                + "' GROUP BY fecha ORDER BY fecha ASC"; }
15    Venta obj;
16    ArrayList rst = null, arrLin = null;
17    int i = 0;
18    try{
19        oBd = new AccesoDatos();
20        if (oBd.conectar()){
21            rst = oBd.ejecutarConsulta(sQuery);
22            oBd.desconectar();
23        }
24    }catch(Exception ee){
25        System.out.println("Error de conexion");
26    }
27    try{
28        if (rst != null) {
29            for (i = 0; i < rst.size(); i++) {
30                obj = new Venta();
31                arrLin = (ArrayList)rst.get(i);
32                obj.setFecha((Date)arrLin.get(0));
33                obj.setCantidad(((Double)arrLin.get(1)).intValue());
34                resultados.add(obj);
35            }
36        }
37    }catch(Exception e){
38        System.out.println(e);
39        System.out.println("Error de lectura");
40    }
41    return resultados;
42 }

```

Listado 3.13 Método buscaVentas.

El método de la Figura anterior, muestra el proceso para la obtención de las ventas efectuadas por la empresa, ya sea cuando se desee las ventas de un producto en específico o de todos (primera condición *if*), ejecuta la petición y cada resultado se almacena en un arreglo para su posterior envío.

Al regresar el control, el sistema reutiliza las ventas obtenidas para pronosticar una venta en el futuro (definida por el intervalo de tiempo que el usuario determinó), mediante el método `pronosticaVentas` (Listado 3.12) que le asigna el resultado del pronóstico para su despliegue al usuario. Los Listados 3.14 a 3.15 muestran el proceso para el pronóstico de las ventas a partir de la reutilización del arreglo de ventas.

```
1 public String pronosticaVenta(ArrayList<Venta> ventas){  
2     String resultado = new SeriesTiempo().pronosticarVentas(ventas);  
3     return resultado;  
4 }
```

Listado 3.14 Método intermedio para la pronóstico de ventas.

El método `pronosticaVenta` toma el arreglo provisto y lo redirige al método correcto para que éste retorne el resultado esperado.

```

1 private static DecimalFormat decimales = new DecimalFormat("##.00");
2
3 public String pronosticarVentas(ArrayList<Venta> ventas) {
4     String resultado = "";
5     try {
6         generarARFFV(ventas);
7         Instances dataset = new Instances(new BufferedReader(
8             new FileReader("C:\\temporal\\consultaVentas.arff")));
9         dataset.sort(0);
10        WekaForecaster forecaster = new WekaForecaster();
11        forecaster.setFieldsToForecast("cantidad");
12        forecaster.setBaseForecaster(new MultilayerPerceptron());
13        forecaster.getTSLagMaker().setTimeStampField("fecha");
14        forecaster.getTSLagMaker().setMinLag(1);
15        forecaster.getTSLagMaker().setMaxLag(12);
16        forecaster.buildForecaster(dataset);
17        forecaster.primeForecaster(dataset);
18        List<List<NumericPrediction>> forecast = forecaster.forecast(12,
19            System.out);
20        List<NumericPrediction> predsAtStep = forecast.get(0);
21        NumericPrediction predForTarget = predsAtStep.get(0);
22        resultado = "" + decimales.format(predForTarget.predicted());
23        System.out.println("" + decimales.format(predForTarget.predicted())
24            + " ");
25    } catch (Exception ex) {
26        ex.printStackTrace();
27    }
28    return resultado;
29 }

```

Listado 3.15 Método pronosticarVentas.

El método del Listado anterior como primer paso genera un archivo con extensión **.arff** (Listado 3.16) el cual es el formato que las bibliotecas de *Weka* requiere, después procede con la lectura correspondiente de dicho archivo, ordena el contenido por medio del primer atributo (fecha), se asigna el atributo a pronosticar (cantidad), se determina el algoritmo a utilizar, se genera el pronosticador y se ejecuta el método que regresa un arreglo de arreglos los cuales contienen las

predicciones (según se configure) de los datos proporcionados, casi al final se aplica el formato al resultado obtenido a dos decimales (mediante el uso de un formato decimal definido en la parte superior) y acto seguido se regresa el resultado.

```

1 private void generarARFFV(ArrayList<Venta> ventas) {
2     try (PrintWriter writer = new PrintWriter(
3         new File("C:\\temporal\\consultaVentas.arff"));){
4         StringBuilder sb = new StringBuilder();
5         sb.append("@relation consulta");
6         sb.append('\n');
7         sb.append('\n');
8         sb.append("@attribute fecha date 'yyyy-MM-dd'");
9         sb.append('\n');
10        sb.append("@attribute cantidad numeric");
11        sb.append('\n');
12        sb.append("@data");
13        sb.append('\n');
14        for (Venta venta : ventas) {
15            sb.append(venta.getFecha());
16            sb.append(',');
17            sb.append(venta.getCantidad());
18            sb.append('\n');
19        }
20        writer.write(sb.toString());
21    } catch (Exception e) {
22        System.out.println(e.getMessage());
23    }
24 }

```

Listado 3.16 Método generarARFFV.

El método mostrado en el Listado anterior, toma como parámetro el arreglo de ventas, y genera un archivo con extensión **.arff** al cual se le asigna la cabecera dándole un nombre a la relación, dando de alta los atributos que lo conforman y por último un ciclo para agregar todas y cada una de las ventas encontradas.

Una vez que se genera el pronóstico, se crea el modelo de la gráfica a utilizar con los datos de ventas obtenidos (Listado 3.13) para ser desplegados al usuario. Las figuras 3.41 a 3.42 muestran el resultado de lo anterior.

Pronosticar ventas

Producto: * Todos

Fecha inicio: * 1/01/18 Fecha Fin: * 1/01/19

Buscar

Figura 3.41 Formulario pronosticar ventas.

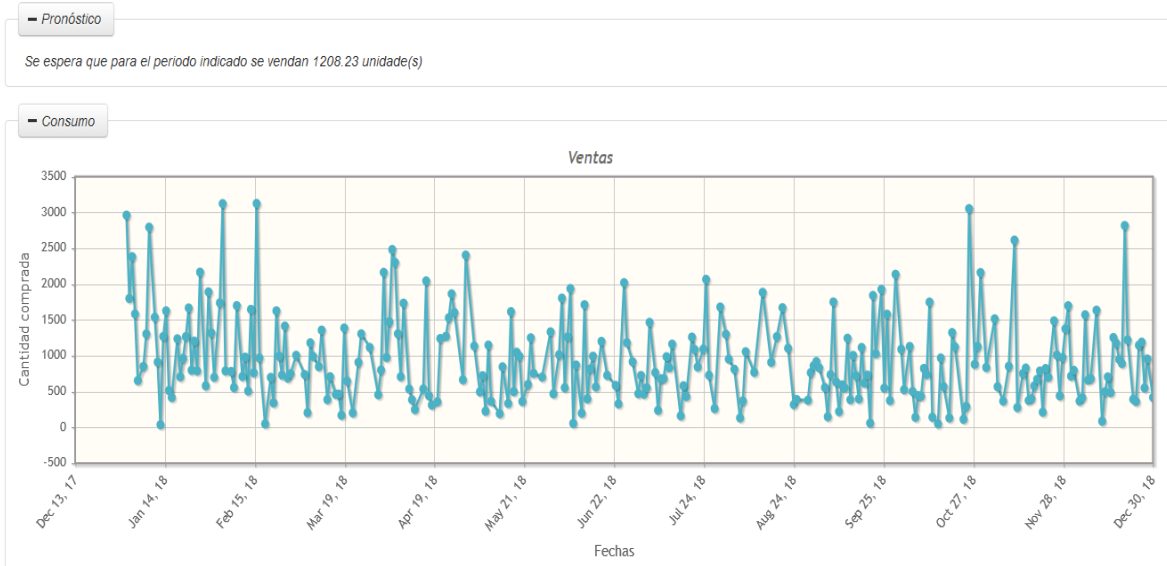


Figura 3.42 Resultados de pronosticar ventas.

Como se observa, se presenta una leyenda que menciona el posible escenario para ese determinado periodo de tiempo y una gráfica que presenta las fluctuaciones de dichas ventas a lo largo del periodo de tiempo indicado.

El resultado anterior será de gran utilidad para el usuario pronosticador dado que al conocer un posible escenario, determinará las acciones correspondientes para hacerle frente o en su defecto ayudarse.

3.5.2.2 Pronosticar consumo de cliente

El usuario que desee pronosticar las compras de un cliente determinado hacía algún producto (o todos) entra a la vista correspondiente y completa la captura de los campos pulsando al final el botón “Buscar” (Figura 3.40), una vez hecho eso el sistema ejecuta un método que realiza las peticiones correspondientes. El Listado 3.17 presenta el método previamente mencionado.

```
1 public void pronosticarCliente(ActionEvent ae) {
2     SimpleDateFormat formmat1 = new SimpleDateFormat("yyyy-MM-dd");
3     comprasCliente = objVenta.buscaComprasCliente(
4         cliente_seleccionado.getId_cliente(),
5         formmat1.format(fecha_inicio),
6         formmat1.format(fecha_fin), producto);
7     pronosticoCliente = objVenta.pronosticaCliente(comprasCliente,
8         cliente_seleccionado.getId_cliente());
9     generaGraficaC(comprasCliente);
10 }
```

Listado 3.17 Método pronosticarCliente.

Una vez que es invocado el método de la figura anterior, inicializa el formato de fecha esperado para la implementación de series de tiempo, acto seguido le asigna un arreglo de ventas a la variable correspondiente mediante la invocación del método “buscaComprasCliente” que recibe como parámetros el id del cliente, la fecha inicio, la fecha fin y el producto a buscar. El Listado 3.18 presenta el método “buscaComprasCliente”.


```

1 public ArrayList<Venta> buscaComprasCliente(int id, String fecha_inicio,
2     String fecha_fin, String producto){
3     String sQuery = "";
4     ArrayList<Venta> resultados = new ArrayList<>();
5     if(producto.equalsIgnoreCase("*"))
6         sQuery = "SELECT fecha_compra as fecha, sum(cantidad_comprada) "
7             + "as compra "
8             + "FROM ventas WHERE cliente = " + id + " "
9             + "AND fecha_compra between '" + fecha_inicio + "' AND '"
10            + fecha_fin + "' GROUP BY fecha ORDER BY fecha ASC";
11     else
12         sQuery = "SELECT fecha_compra as fecha, sum(cantidad_comprada) "
13             + "as compra "
14             + "FROM ventas WHERE cliente = " + id + " AND producto = '"
15             + producto + "' AND fecha_compra between '"
16             + fecha_inicio + "' AND '" + fecha_fin + "' "
17             + "GROUP BY fecha ORDER BY fecha ASC";
18     Venta obj;
19     ArrayList rst = null, arrLin = null;
20     int i = 0;
21     try{
22         oBd = new AccesoDatos();
23         if (oBd.conectar()){
24             rst = oBd.ejecutarConsulta(sQuery);
25             oBd.desconectar();
26         }
27     }catch(Exception ee){
28         System.out.println("Error de conexion");
29     }
30     try{
31         if (rst != null) {
32             for (i = 0; i < rst.size(); i++) {
33                 obj = new Venta();
34                 arrLin = (ArrayList)rst.get(i);
35                 obj.setFecha((Date)arrLin.get(0));
36                 obj.setCantidad(((Double)arrLin.get(1)).intValue());
37                 resultados.add(obj);
38             }
39         }
40     }catch(Exception e){
41         System.out.println(e);
42         System.out.println("Error de lectura");
43     }
44     return resultados;
45 }

```

Listado 3.18 Método buscaComprasCliente.

El método de la figura anterior, muestra el proceso para la obtención de las compras efectuadas por un cliente en particular, ya sea cuando se desee las compras de un producto en específico o de todos (primera condición *if*), ejecuta la petición y cada resultado se almacena en un arreglo para su posterior envío.

Al regresar el control, el sistema reutiliza las compras obtenidas para pronosticar una compra en el futuro (definida por el intervalo de tiempo que el usuario determinó), mediante el método `pronosticaCliente` (Listado 3.17) que le asigna el resultado del pronóstico para su despliegue al usuario. Los Listados 3.19 a 3.20 muestran el proceso para el pronóstico de las ventas a partir de la reutilización del arreglo de compras.

```
1 public String pronosticaCliente(ArrayList<Venta> compras, int id_cliente){
2     String resultado = new SeriesTiempo().pronosticarCliente(compras,
3         id_cliente);
4     return resultado;
5 }
```

Listado 3.19 Método pronosticaCliente.

El método `pronosticaCliente` toma el arreglo provisto y lo redirige al método correcto para que éste retorne el resultado esperado.

```

1 public String pronosticarCliente(ArrayList<Venta> ventas, int id) {
2     String resultado = "";
3     try {
4         generarARFFC(ventas, id);
5         Instances dataset = new Instances(new BufferedReader(
6             new
7             FileReader("C:\\temporal\\consulta"+id+".arff"));
8         dataset.sort(0);
9         WekaForecaster forecaster = new WekaForecaster();
10        forecaster.setFieldsToForecast("cantidad");
11        forecaster.setBaseForecaster(new MultilayerPerceptron());
12        forecaster.getTSLagMaker().setTimeStampField("fecha");
13        forecaster.getTSLagMaker().setMinLag(1);
14        forecaster.getTSLagMaker().setMaxLag(12);
15        forecaster.buildForecaster(dataset);
16        forecaster.primeForecaster(dataset);
17        List<List<NumericPrediction>> forecast = forecaster.forecast(12,
18            System.out);
19        List<NumericPrediction> predsAtStep = forecast.get(0);
20        NumericPrediction predForTarget = predsAtStep.get(0);
21        resultado = "" + decimales.format(predForTarget.predicted());
22        System.out.println("" + predForTarget.predicted() + " ");
23    } catch (Exception ex) {
24        ex.printStackTrace();
25    }
26    return resultado;
27 }

```

Listado 3.20 Método pronosticarCliente.

El método de la figura anterior como primer paso genera un archivo con extensión **.arff** (Listado 3.21) el cual es el formato que las bibliotecas de *Weka* requiere, después procede con la lectura correspondiente de dicho archivo, ordena el contenido por medio del primer atributo (fecha), se asigna el atributo a pronosticar (cantidad), se determina el algoritmo a utilizar, se genera el pronosticador y se ejecuta el método que regresa un arreglo de arreglos los cuales contienen las

predicciones (según se configure) de los datos proporcionados, casi al final se formatea el resultado obtenido a dos decimales (mediante el uso de un formato decimal definido en la parte superior) y acto seguido se regresa el resultado.

```

1 private void generarARFFC(ArrayList<Venta> ventas, int id) {
2     try (PrintWriter writer = new PrintWriter(
3         new File("C:\\temporal\\consulta"+id+".arff"));){
4         StringBuilder sb = new StringBuilder();
5         sb.append("@relation consulta");
6         sb.append('\n');
7         sb.append('\n');
8         sb.append("@attribute fecha date 'yyyy-MM-dd'");
9         sb.append('\n');
10        sb.append("@attribute cantidad numeric");
11        sb.append('\n');
12        sb.append("@data");
13        sb.append('\n');
14        for (Venta venta : ventas) {
15            sb.append(venta.getFecha());
16            sb.append(',');
17            sb.append(venta.getCantidad());
18            sb.append('\n');
19        }
20        writer.write(sb.toString());
21    } catch (Exception e) {
22        System.out.println(e.getMessage());
23    }
24 }

```

Listado 3.21 Método generarARFFC.

El método mostrado en la figura anterior, toma como parámetro el arreglo de ventas, y genera un archivo con extensión **.arff** al cual se le asigna la cabecera dándole un nombre a la relación, dando de alta los atributos que lo conformarán y por último un ciclo para agregar todas y cada una de las ventas encontradas.

Una vez que genera el pronóstico se crea el modelo de la gráfica a utilizar con los datos de compras obtenidos (Listado 3.17) para ser mostrados al usuario. Las Figuras 3.43 a 3.44 muestran el resultado de lo anterior.

Pronosticar consumo de cliente

Cliente: *
 Tipo de búsqueda: * ID Nombre
 Producto: *
 Fecha inicio: * Fecha Fin: *

Clientes	
Id	Nombre
<input type="radio"/> 1	Cliente 1
<input type="radio"/> 4	Cliente 4
<input type="radio"/> 6	Cliente 6
<input type="radio"/> 2	Cliente 2
<input type="radio"/> 5	Cliente 5
<input type="radio"/> 3	Cliente 3

Figura 3.43 Formulario pronosticar consumo de cliente.

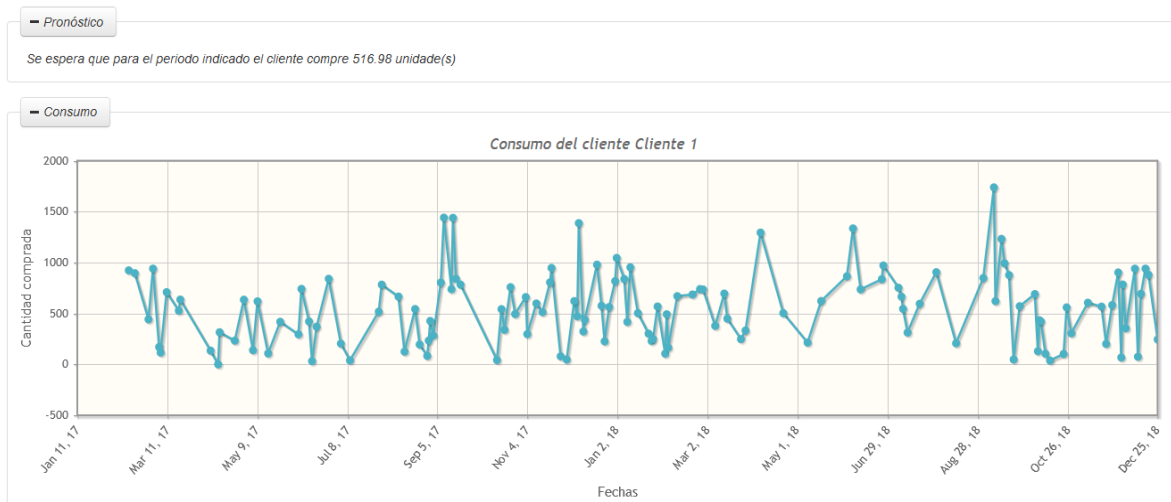


Figura 3.44 Resultados de pronosticar consumo de cliente.

Como se observa, se presenta una leyenda que menciona el posible escenario para ese determinado periodo de tiempo y una gráfica que presenta las fluctuaciones de dichas compras a lo largo del periodo de tiempo indicado.

El resultado anterior será de gran utilidad para el usuario pronosticador dado que al conocer un posible escenario, determinará las acciones correspondientes para hacerle frente o en su defecto ayudarse.

3.6 *Sprint 5* gestión de usuarios y recomendaciones

Una vez teniendo las interfaces y funcionalidades para el usuario clasificador y pronosticador, se debe tener una manera de gestionarlos, dicha responsabilidad es del usuario administrador, añadiéndole la funcionalidad de modificar las recomendaciones para los grupos y que el usuario clasificador consulta constantemente para la toma de decisiones. Por lo anterior, en este *sprint* se lleva a cabo la realización de las interfaces para el usuario administrador, así como las funcionalidades necesarias para cubrir sus necesidades.

3.6.1 Interfaces

En esta sección se muestran las interfaces necesarias para el usuario administrador.

3.6.1.1 Modificar recomendación

La interfaz modificar recomendación es la encargada de proporcionarle al usuario una forma sencilla de buscar y seleccionar el grupo al que requiera modificar su respectiva recomendación. La Figura 3.45 muestra la interfaz preliminar.

Gestión de recomendaciones

Grupo:

Grupos de clientes			
Id	Nombre	Recomendación	Acción
Sin resultados			

Figura 3.45 Interfaz modificar recomendación.

La interfaz presentada en la figura anterior, presenta un campo de texto para ingresar el nombre del grupo a buscar, o bien buscar todos los grupos dados de alta. Para la modificación de la recomendación, se debe seleccionar un grupo y se muestra un formulario para actualizar la recomendación. La Figura 3.46 presenta dicha interfaz.



El formulario muestra un botón de pestaña "Detalles" en la esquina superior izquierda. A continuación, hay tres campos de entrada: "Id:" con el valor "1", "Grupo" con el valor "Grupo 1", y "Recomendación:" con un campo de texto grande que contiene "Recomendación del Grupo 1". En la parte inferior del formulario hay un botón "Guardar".

Figura 3.46 Formulario para actualizar recomendación.

Dado que la sugerencia (recomendación) es el único campo a actualizar, el id y el grupo permanecen deshabilitados para su edición.

3.6.1.2 Gestionar usuarios

De igual forma que la interfaz para la modificación de las recomendaciones, esta interfaz provee de un formulario para la búsqueda de un usuario determinado o todos los usuarios, pero con la particularidad de proveer al administrador un mecanismo para la creación de un nuevo usuario. La Figura 3.47 muestra la vista preliminar.

Gestión de usuarios

Usuario:

Usuarios			
Id	Usuario	Nombre	Acción
Sin resultados			

Figura 3.47 Interfaz gestionar usuarios.

Además del formulario para la búsqueda o creación de un usuario, se añade una tabla en la cual se muestran los resultados obtenidos además de una serie de acciones disponibles para cada usuario definido con anterioridad.

En caso de requerir una actualización o una creación, los atributos a considerar son los mismos para cualquier caso. En la Figura 3.48 se muestra el formulario correspondiente.

Id:

Usuario:

Contraseña:

Nombre:

Tipo:

Figura 3.48 Formulario nuevo/actualizar usuario.

En la figura anterior, se presenta un campo deshabilitado el cual pertenece al id del usuario (0 para uno nuevo) así como la posibilidad de agregar/modificar: nombre, contraseña, su nombre y por último, el tipo de usuario.

3.6.2 Funcionalidad

Teniendo las interfaces correspondientes, en esta sección se aborda la funcionalidad de la aplicación aunada a las vistas generadas, para dar solución a las peticiones del usuario administrador.

3.6.2.1 Modificar recomendación

Una vez que el administrador accede correctamente a la vista correspondiente, tiene la facultad de buscar un grupo en particular o a todos, mediante el campo de texto definido, sea cual sea el caso, el botón que ejecuta la petición se comunica con el método buscarGrupo. El Listado 3.22 muestra dicho método.

```

1 public void buscarGrupo(){
2     generaDetalles = false;
3     Grupo obj = new Grupo();
4     String param = "";
5     if(getBusqueda().equalsIgnoreCase("") ||
6         getBusqueda().equalsIgnoreCase(" ") || getBusqueda() == null)
7         param = "todo";
8     grupos = obj.buscarGrupo(param, busqueda);
9     if(grupos.size() > 0){
10        mensaje("Correcto", "Se encontraron " + grupos.size()
11            + " resultados");
12    }else{
13        mensaje("Sin resultados", "Sin resultados para mostrar");
14    }
15 }

```

Listado 3.22 Método buscarGrupo.

En caso de no ingresar nombre alguno a buscar, se realiza una petición para traer todos los resultados posibles, en caso contrario se busca al grupo requerido, el método encargado de dicho proceso es buscarGrupo pero una clase diferente, el cual recibe como parámetro el tipo de búsqueda (todos o uno) y el nombre del grupo a buscar. En el Listado 3.23 se detalla el segundo método buscarGrupo.

```

1 public ArrayList<Grupo> buscarGrupo(String parametro, String grupo){
2     ArrayList<Grupo> resultados = new ArrayList<>();
3     String sQuery = "";
4     if(parametro.equalsIgnoreCase("todo")){
5         sQuery = "SELECT id_grupo, grupo, sugerencia FROM grupo";
6     }else{
7         sQuery = "SELECT id_grupo, grupo, sugerencia FROM grupo "
8             + "WHERE grupo LIKE ('%" + grupo + "%')";
9     }
10    Grupo obj;
11    ArrayList rst = null, arrLin = null;
12    int i = 0;
13    try{
14        oBd = new AccesoDatos();
15        if (oBd.conectar()){
16            rst = oBd.ejecutarConsulta(sQuery);
17            oBd.desconectar();
18        }
19    }catch(Exception ee){
20        System.out.println("Error de conexion");
21    }
22    try{
23        if (rst != null) {
24            for (i = 0; i < rst.size(); i++) {
25                obj = new Grupo();
26                arrLin = (ArrayList)rst.get(i);
27                obj.setId(((Double)arrLin.get(0)).intValue());
28                obj.setGrupo((String)arrLin.get(1));
29                obj.setRecomendacion((String)arrLin.get(2));
30                resultados.add(obj);
31            }
32        }
33    }catch(Exception e){
34        System.out.println(e);
35        System.out.println("Error de lectura");
36    }
37    return resultados;

```

Listado 3.23 Método buscarGrupo.

Al inicio del método, se realiza la diferenciación de la consulta a efectuar, es decir, si se buscan a todos los grupos o a uno en particular, sin importar el caso, se realiza la petición a la base de datos, y ésta regresa como resultado a los grupos (o grupo) que serán almacenados en un arreglo para después ser devuelto.

Cuando el arreglo de grupos es devuelto, se asigna a una variable para su despliegue al usuario administrador. La Figura 3.49 muestra los resultados.

Gestión de recomendaciones

Grupo:

Grupos de clientes			
Id	Nombre	Recomendación	Acción
2	Grupo 2	Ofrecer más descuentos a Orizaba	<input type="button" value="Modificar"/>
4	Grupo 4	Ofrecer descuentos del 10 al 15%	<input type="button" value="Modificar"/>
3	Grupo 3	Dar timbres gratis a clientes de Orizaba	<input type="button" value="Modificar"/>
5	Grupo 5	Recomendar la versión Web	<input type="button" value="Modificar"/>
1	Grupo 1	Recomendación del Grupo 1	<input type="button" value="Modificar"/>

Figura 3.49 Resultados de búsqueda de grupo.

Cuando se hayan desplegado los grupos, el usuario administrador que pulsar el botón “Modificar” para acceder a un formulario que modifica la recomendación del grupo elegido. La Figura 3.50 presenta dicho formulario.

Figura 3.50 Formulario de modificación de recomendación.

Una vez que se ingresada la nueva recomendación, y presionar el botón “Guardar” el sistema invoca a un método encargado de la actualización de la recomendación. El Listado 3.24 presenta el método antes descrito.

```

1 public void guardaRecomendacion(){
2     grupo_seleccionado.setRecomendacion(sugerencia);
3     if(new Grupo().guardarRecomendacion(grupo_seleccionado)){
4         setGeneraDetalles(false);
5         buscarGrupo(grupo_seleccionado.getGrupo());
6         mensaje("Correcto", "Recomendación actualizada");
7     }
8     else{
9         mensaje("Error", "Fallo al guardar recomendación");
10    }
11 }

```

Listado 3.24 Método guardaRecomendacion.

Al grupo seleccionado se le asigna la nueva recomendación, y se le indica que se actualice mediante el método guardarRecomendación. El Listado 3.25 muestra dicho método.

```
1 public boolean guardarRecomendacion(Grupo grupo){
2     boolean flag = false;
3     String sQuery = "UPDATE grupo SET sugerencia = '"
4         + grupo.getRecomendacion()+ "' "
5         + "WHERE id_grupo = " + grupo.getId() + """;
6     try {
7         int nRet = 0;
8         oBd = new AccesoDatos();
9         if (oBd.conectar()){
10            nRet = oBd.ejecutarComando(sQuery);
11            oBd.desconectar();
12        }
13        if(nRet > 0)
14            flag = true;
15    } catch (Exception ex) {
16        System.out.println(ex);
17    }
18    return flag;
19 }
```

Listado 3.25 Método guardarRecomendacion.

La figura anterior muestra el método encargado de la actualización de la recomendación, la cual simplemente realiza la petición adecuada y regresa un valor indicando la correcta actualización (*true*) o algún error (*false*), una vez que termina, se le muestra al usuario el grupo con su nueva recomendación. La Figura 3.51 muestra el resultado.

Gestión de recomendaciones

Grupo:

Grupos de clientes			
Id	Nombre	Recomendación	Acción
1	Grupo 1	Ofrecer descuentos de 30%	<input type="button" value="Modificar"/>

*Figura 3.51 Recomendación actualizada.***3.6.2.2 Gestionar usuarios**

Una vez que el administrador accede correctamente a la vista correspondiente, una de sus actividades que tiene permitido realizar, es buscar un usuario en particular o a todos, mediante el campo de texto definido, sea cual sea el caso, el botón que ejecuta la petición se comunica con el método buscarUsuario. El Listado 3.26 muestra dicho método.

```

1 public void buscarUsuario(){
2     generaDetalles = false;
3     Usuario obj = new Usuario();
4     String param = "";
5     if(getBusqueda().equalsIgnoreCase("") ||
6         getBusqueda().equalsIgnoreCase(" ") || getBusqueda() == null)
7         param = "todo";
8     usuarios = obj.buscarUsuario(param, busqueda);
9     if(usuarios.size() > 0){
10        mensaje("Correcto", "Se encontraron " + usuarios.size()
11            + " resultados");
12    }else{
13        mensaje("Sin resultados", "Sin resultados para mostrar");
14    }
15}

```

Listado 3.26 Método buscarUsuario.

En caso de no ingresar nombre alguno a buscar, se realiza una petición para traer todos los resultados posibles, en caso contrario se busca al grupo requerido, el método encargado de dicho proceso es `buscarUsuario` pero una clase diferente, el cual recibe como parámetro el tipo de búsqueda (todos o uno) y el nombre del usuario a buscar. En el Listado 3.27 se detalla el segundo método `buscarUsuario`.

```

1 public ArrayList<Usuario> buscarUsuario(String parametro, String usuario){
2     ArrayList<Usuario> resultados = new ArrayList<>();
3     String sQuery = "";
4     if(parametro.equalsIgnoreCase("todo")){
5         sQuery = "SELECT id_usuario, usuario, pass, tipo, nombre "
6                 + "FROM usuario WHERE tipo <> 3";
7     }else{
8         sQuery = "SELECT id_usuario, usuario, pass, tipo, nombre "
9                 + "FROM usuario "
10                + "WHERE usuario LIKE ('%" + usuario + "%') AND tipo <> 3";
11    }
12    Usuario obj;
13    ArrayList rst = null, arrLin = null;
14    int i = 0;
15    try{
16        oBd = new AccesoDatos();
17        if (oBd.conectar()){
18            rst = oBd.ejecutarConsulta(sQuery);
19            oBd.desconectar();
20        }
21    }catch(Exception ee){
22        System.out.println("Error de conexion");
23    }
24    try{
25        if (rst != null) {
26            for (i = 0; i < rst.size(); i++) {
27                obj = new Usuario();
28                arrLin = (ArrayList)rst.get(i);
29                obj.setId(((Double)arrLin.get(0)).intValue());
30                obj.setUsuario((String)arrLin.get(1));
31                obj.setContrasenia((String)arrLin.get(2));
32                obj.setTipo(((Double)arrLin.get(3)).intValue());
33                obj.setNombre((String)arrLin.get(4));
34                resultados.add(obj);
35            }
36        }
37    }catch(Exception e){
38        System.out.println(e);
39        System.out.println("Error de lectura");
40    }
41    return resultados;
42}

```

Listado 3.27 Método buscarUsuario con parámetros.

Al inicio del método, se realiza la diferenciación de la consulta a efectuar, es decir, si se buscan a todos los usuarios o a uno en particular, sin importar el caso, se realiza la petición a la base de datos, y ésta regresa como resultado a los usuarios (o usuario) que serán almacenados en un arreglo para después ser devuelto.

Cuando el arreglo de usuarios es devuelto, se asigna a una variable para su despliegue al usuario administrador. La Figura 3.52 muestra los resultados.

Gestión de usuarios

Usuario:

Usuarios			
Id	Usuario	Nombre	Acción
2	MANUEL	MANUEL MALDONADO GARCÍA	<input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
1	LUIS	LUIS FRANCISCO MALPICA LÓPEZ	<input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>

Figura 3.52 Resultados de buscar usuario.

Teniendo la lista de usuarios disponibles para su acceso al sistema, las dos acciones posibles a realizar, son: modificar o eliminar al usuario, en caso de elegir su eliminación, basta con seleccionar la opción adecuada y el sistema realizará la petición correspondiente. El Listado 3.28 muestra la invocación de un método al pulsar el botón “Eliminar”.

```

1 public void eliminarUsuario(int id){
2     generaDetalles = false;
3     if(new Usuario().eliminarUsuario(id)){
4         mensaje("Correcto", "Usuario eliminado");
5         busqueda = "";
6         buscarUsuario();
7     }
8     else
9         mensaje("Error", "Error al eliminar usuario");
10}

```

Listado 3.28 Método eliminarUsuario.

Se envía el Id del usuario a eliminar (dado que es lo único que se necesita) al método eliminar usuario descrito en el Listado 3.29.

```
1 public boolean eliminarUsuario(int id){
2     boolean flag = false;
3     String sQuery = "DELETE FROM usuario WHERE id_usuario = " + id + ";
4     try {
5         int nRet = 0;
6         oBd = new AccesoDatos();
7         if (oBd.conectar()){
8             nRet = oBd.ejecutarComando(sQuery);
9             oBd.desconectar();
10        }
11        if(nRet > 0)
12            flag = true;
13    } catch (Exception ex) {
14        System.out.println(ex);
15    }
16    return flag;
17}
```

Listado 3.29 Método eliminarUsuario.

En el método eliminar Usuario (Listado 3.29) se muestra una sencilla petición para eliminar al usuario con el id enviado, una vez que se ejecuta y que devuelve el control, el sistema actualiza la tabla de usuarios demostrando la eliminación del usuario seleccionado.

Por otra parte si se elige una modificación del usuario, se desplegará un formulario con la información hasta el momento del usuario seleccionado, tal y como se muestra en la Figura 3.53.

Gestión de usuarios

Usuario:

Usuarios			
Id	Usuario	Nombre	Acción
2	MANUEL	MANUEL MALDONADO GARCÍA	<input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
1	LUIS	LUIS FRANCISCO MALPICA LÓPEZ	<input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>

Id:

Usuario:

Contraseña:

Nombre:

Tipo: ▼

Figura 3.53 Interfaz de modificación de un usuario.

El campo para el id es el único que no se ve habilitado para su edición, pero los demás campos sí, una vez que se efectuaron los cambios deseados, se selecciona el botón “Guardar” el cual invoca al método mostrado en el Listado 3.30.

```
1 public void guardarUSuario(){
2     String param = "";
3     if(id == 0){
4         param = "nuevo";
5     }
6     usuario_seleccionado.setUsuario(usuario.toUpperCase());
7     usuario_seleccionado.setContrasenia(pass);
8     usuario_seleccionado.setTipo(tipo);
9     usuario_seleccionado.setNombre(nombre.toUpperCase());
10    if(new Usuario().guardarUsuario(param, usuario_seleccionado)){
11        setGeneraDetalles(false);
12        buscaUsuario(usuario_seleccionado.getUsuario());
13        mensaje("Correcto", "Usuario actualizado");
14    }
15    else{
16        mensaje("Error", "Fallo al guardar usuario");
17    }
18
19}
```

Listado 3.30 Método guardarUsuario.

En caso de referirse a un usuario nuevo (id = 0) el parámetro a enviar toma el valor de “nuevo”, pero en este caso se trata de una actualización por lo que el parámetro se envía sin ninguna palabra salvo una cadena de texto vacía hacia el método mostrado en el Listado 3.31.

```

1 public boolean guardarUsuario(String param, Usuario usu){
2     boolean flag = false;
3     String sQuery = "";
4     if(param.equalsIgnoreCase("nuevo"))
5         sQuery = "INSERT INTO usuario (usuario, pass, tipo, nombre) "
6                 + "VALUES ('" + usu.getUsuario()+ "', '"
7                 + usu.getContrasenia()+ "', "
8                 + usu.getTipo()+ "', '" + usu.getNombre()+ "')";
9     else
10        sQuery = "UPDATE usuario SET usuario = '" + usu.getUsuario()+ "', "
11                + "pass = '" + usu.getContrasenia()+ "', tipo = "
12                + usu.getTipo()+ "', "
13                + "nombre = '" + usu.getNombre()+ "' WHERE id_usuario = "
14                + usu.getId() + "'";
15    try {
16        int nRet = 0;
17        oBd = new AccesoDatos();
18        if (oBd.conectar()){
19            nRet = oBd.ejecutarComando(sQuery);
20            oBd.desconectar();
21        }
22        if(nRet > 0)
23            flag = true;
24    } catch (Exception ex) {
25        System.out.println(ex);
26    }
27    return flag;
28}

```

Listado 3.31 Método guardarUsuario.

El método mostrado en el Listado anterior, se encarga ya sea de crear a un nuevo usuario o de modificar a alguno existente, regresando el estado de dicha acción.

Por último al usuario se le muestra al usuario seleccionado con sus cambios ya efectuados. La figura 3.54 muestra los resultados.

Gestión de usuarios

Usuario:

Buscar

Usuarios			
Id	Usuario	Nombre	Acción
2	MAN	MANUEL MALDONADO GARCÍA	<input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>

Figura 3.54 Modificación del usuario.

La última acción del usuario administrador es crear a un nuevo usuario, para ello deberá seleccionar el botón “Nuevo registro” con el cual se despliega el formulario mostrado en la Figura 3.53 con la excepción de encontrarse vacío tal y como se muestra en la Figura 3.55.

Gestión de usuarios

Usuario:

Buscar

Usuarios			
Id	Usuario	Nombre	Acción
Sin resultados			

Id:

Usuario:

Contraseña:

Nombre:

Tipo:

Figura 3.55 Formulario para un nuevo usuario.

Una vez que se completa el formulario mostrado en la figura anterior, se pulsa el botón “Guardar” y efectúa el mismo procedimiento mostrado en los Listados 3.30 a 3.31, el cual es guardar a un nuevo usuario, y después desplegarlo para su verificación por parte del usuario administrador.

3.7 *Sprint* 6 iniciar sesión

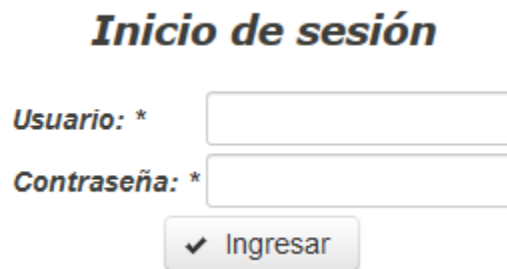
Teniendo ya las interfaces y funcionalidades para cada tipo de usuario, en este último *sprint* se lleva a cabo la generación del proceso para iniciar sesión en la aplicación.

3.7.1 Interfaces

En esta sección se mostrarán las interfaces necesarias para el inicio de sesión y el acceso al sistema.

3.7.1.1 Iniciar sesión

Esta interfaz cuenta con dos campos para el ingreso del usuario y la contraseña de la persona que requiera ingresar al sistema. La Figura 3.56 muestra la interfaz preliminar.

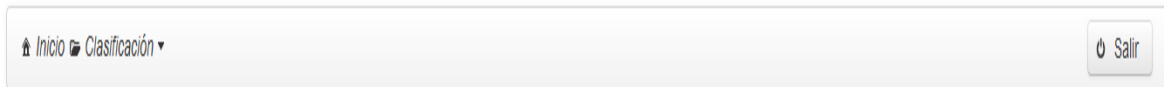


La imagen muestra una interfaz de inicio de sesión con el título "Inicio de sesión" en un estilo de letra cursiva y azul. Debajo del título, hay dos campos de entrada de texto. El primer campo está etiquetado como "Usuario: *" y el segundo como "Contraseña: *". Debajo de estos campos hay un botón con un icono de checkmark y el texto "Ingresar".

Figura 3.56 Interfaz iniciar sesión.

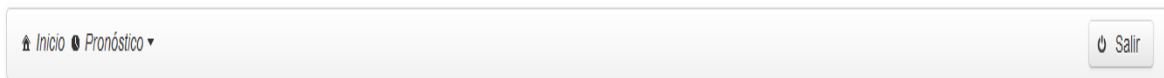
3.7.1.2 Inicio

La interfaz de inicio muestra un mensaje de bienvenida al usuario que inició sesión correctamente, de igual forma las opciones mostradas varían según el tipo de usuario: clasificador (Figura 3.57), pronosticador (Figura 3.58) o administrador (Figura 3.59).



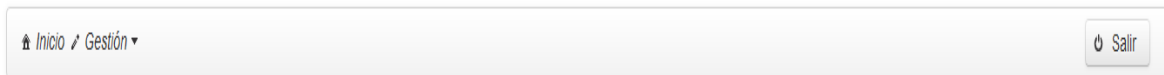
Bienvenido MANUEL MALDONADO GARCÍA

Figura 3.57 Inicio usuario clasificador.



Bienvenido LUIS FRANCISCO MALPICA LÓPEZ

Figura 3.58 Inicio usuario pronosticador.



Bienvenido ALBERTO CONTRERAS MEJÍA

Figura 3.59 Inicio usuario administrador

Para todos se muestra un botón para salir de la aplicación, así como la opción de volver a la página de inicio.

3.7.2 Funcionalidad

Teniendo las interfaces correspondientes, en esta sección se describe la funcionalidad de las vistas para el inicio de sesión de cualquier usuario.

3.7.2.1 Iniciar sesión

Cuando se ingresa el usuario y la contraseña, el botón “Ingresar” realiza una petición a un método llamado *login* que se encarga de validar la información proporcionada. El método *login* se muestra en el Listado 3.32.


```

1 public String login() {
2     FacesContext facesContext = FacesContext.getCurrentInstance();
3     Usuario oUsu = new Usuario();
4     String sRet = "sessions/inicio";
5     Firmado oFirm;
6     oUsu.setUsuario(getUsuario().toUpperCase());
7     oUsu.setContrasenia(getContrasenia());
8     try {
9         if (oUsu.iniciarSesion()) {
10            this.setNombre(oUsu.getNombre());
11            this.setTipo(oUsu.getTipo());
12            this.setContrasenia("");
13            oFirm = new Firmado();
14            oFirm.setUsu(oUsu);
15            /*Firmado no se encuentra bajo el control de JSF, por eso se
16            obtiene directamente HttpSession*/
17            HttpSession session = (HttpSession)
18                facesContext.getExternalContext().getSession(false);
19            session.setAttribute("oFirm", oFirm);
20        } else {
21            sRet = "index";
22        }
23    } catch (Exception e) {
24        sRet = "/index";
25        e.printStackTrace();
26    }
27    return sRet;
28}

```

Listado 3.32 Método login.

Se obtiene el usuario y contraseña ingresados y se realiza una petición al método `iniciarSesion` (Listado 3.33) el cual determina si existe o no el usuario dado con su contraseña dada, en caso de ser verdad se genera un objeto y se guarda en la sesión actual de tal forma que si se quisiera ingresar a alguna parte de la aplicación sin pasar por ahí, no se mostraría nada. En caso de ser un usuario con contraseña válida, se redirige a la página de iniciar sesión.

```

1 public boolean iniciarSesion() throws Exception {
2     boolean bRet = false;
3     ArrayList rst = null, arrLin = null;
4     String sQuery = "SELECT id_usuario, usuario, tipo, nombre FROM usuario "
5         + "WHERE usuario = '" + this.getUsuario() + "' AND pass = '"
6         + this.getContrasenia() + "'";
7     oBd = new AccesoDatos();
8     if (this.oBd.conectar()) {
9         rst = this.oBd.ejecutarConsulta(sQuery);
10        this.oBd.desconectar();
11    }
12    if (rst != null && rst.size() == 1) {
13        arrLin = (ArrayList) rst.get(0);
14        this.setId(((Double)arrLin.get(0)).intValue());
15        this.setUsuario((String) arrLin.get(1));
16        this.contrasenia = "";
17        this.setTipo(((Double)arrLin.get(2)).intValue());
18        this.setNombre((String) arrLin.get(3));
19        bRet = true;
20    }
21    return bRet;
22}

```

Listado 3.33 Método iniciarSesion.

El método mostrado en el Listado anterior, realiza una consulta a la base de datos para encontrar al usuario requerido, en caso de encontrarlo rellena los datos que le faltan (nombre y tipo) y regresa un valor booleano (*true* en caso de encontrarlo o *false* en caso de no encontrarlo), por último se redirecciona al usuario a la página de inicio (Figura 3.57 a 3.57 según el tipo de usuario) y se le muestran las opciones propias de su tipo de usuario (Listado 3.34).

```

1 <h:form>
2     <p:menubar>
3         <p:menuitem value="Inicio" action="inicio"
4             icon="ui-icon-home"/>
5         <p:submenu label="Pronóstico"
6             icon="ui-icon-clock"
7             rendered="#{oLogin.tipo eq 1}">
8             <p:menuitem value="Cliente"
9                 action="pronosticarCliente"
10                icon="ui-icon-person"/>
11            <p:menuitem value="Ventas"
12                action="pronosticarVentas"
13                icon="ui-icon-flag"/>
14        </p:submenu>
15        <p:submenu label="Clasificación"
16            icon="ui-icon-folder-open"
17            rendered="#{oLogin.tipo eq 2}">
18            <p:menuitem value="Ver grupos"
19                action="gruposClientes"
20                icon="ui-icon-calculator"/>
21            <p:menuitem value="Clasificar"
22                action="clasificar"
23                icon="ui-icon-pencil"/>
24            <p:menuitem value="Ver clasificaciones"
25                action="#{oClasificador.
26                    verClasificaciones}"
27                icon="ui-icon-circle-zoomout"/>
28        </p:submenu>
29        <p:submenu label="Gestión" icon="ui-icon-pencil"
30            rendered="#{oLogin.tipo eq 3}">
31            <p:menuitem value="Usuarios"
32                action="gestionUsuarios"
33                icon="ui-icon-person"/>
34            <p:menuitem value="Recomendaciones"
35                action="gestionRecomendaciones"
36                icon="ui-icon-comment"/>
37        </p:submenu>
38        <f:facet name="options">
39            <p:commandButton value="Salir"
40                action="#{oLogin.logout}"
41                icon="ui-icon-power" />
42        </f:facet>
43    </p:menubar>
44</h:form>

```

Listado 3.34 Opciones según tipo de cliente.

Como se observa en el Listado anterior, una vez que se carga la interfaz de inicio, se realiza una distinción del tipo de cliente, de tal forma que dependiendo de éste, son las opciones a mostrar (pronóstico para el tipo 1, clasificación para el tipo 2 o gestión para el tipo 3).

En caso de efectuar la salida de la aplicación, el botón “Salir” ejecuta el método llamado *logout* (Listado 3.35).

```
1 public void logout() throws IOException {
2     FacesContext facesContext = FacesContext.getCurrentInstance();
3     HttpSession session;
4     ServletContext oServletContext = (ServletContext)
5         facesContext.getExternalContext().getContext();
6     String nombreProyecto = oServletContext.getContextPath()
7         + File.pathSeparator;
8     session = (HttpSession)
9         facesContext.getExternalContext().getSession(false);
10    if (session != null) {
11        session.invalidate();
12    }
13    facesContext.getExternalContext().redirect(nombreProyecto);
14}
```

Listado 3.35 Método logout.

El método mostrado en la figura anterior, simplemente destruye la sesión actual y redirige al usuario a la pantalla de iniciar sesión para que un nuevo usuario acceda de nuevo.

Capítulo 4. Resultados

En este capítulo se presentan los resultados de la implementación de dos casos de estudio hacia el sistema desarrollado. El primer caso de estudio aborda un punto de vista distinto al tomado en el capítulo anterior, mientras que en el segundo caso de estudio se implementa el sistema propuesto para dar respuesta a una serie de preguntas provistas por SIFEI.

4.1 Caso de estudio 1

Se realizó un periodo de estancias comprendido del 01 de Mayo de 2019 al 31 de Mayo de 2019 en el Centro de Investigación en Inteligencia Artificial (CIIA) de la Universidad Veracruzana (UV) en la ciudad de Xalapa de Enríquez, Veracruz., México. Bajo la dirección del Dr. Nicandro Cruz Ramírez, ahí se mostró el avance realizado hasta el momento, sin embargo debido a que la herramienta que se utilizó para la generación de datos para *clustering* no presenta bajo qué distribución lo hace, se procedió a atacar el problema desde otro punto de vista mediante la generación pseudoaleatoria de datos bajo una distribución Bayesiana.

Se inició con la creación de una red Bayesiana que contiene los atributos a generar. Se utilizó el editor de redes Bayesiana de *Weka*, la cual provee de una interfaz gráfica que permite la creación de nodos (atributos) y la relación entre ellos. La Figura 4.1 muestra la red Bayesiana generada.

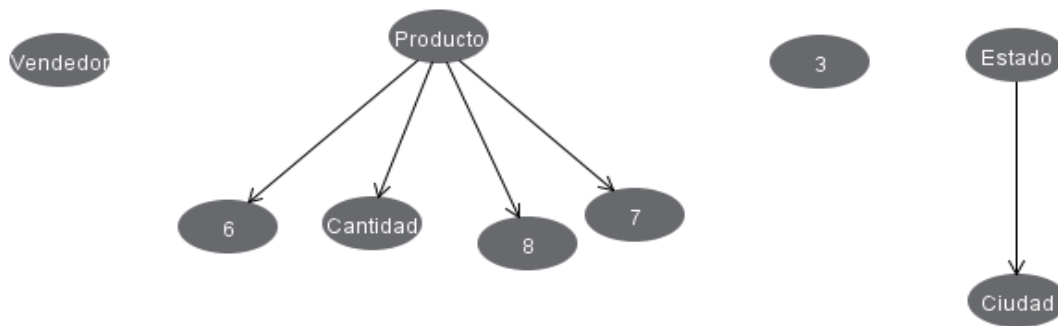
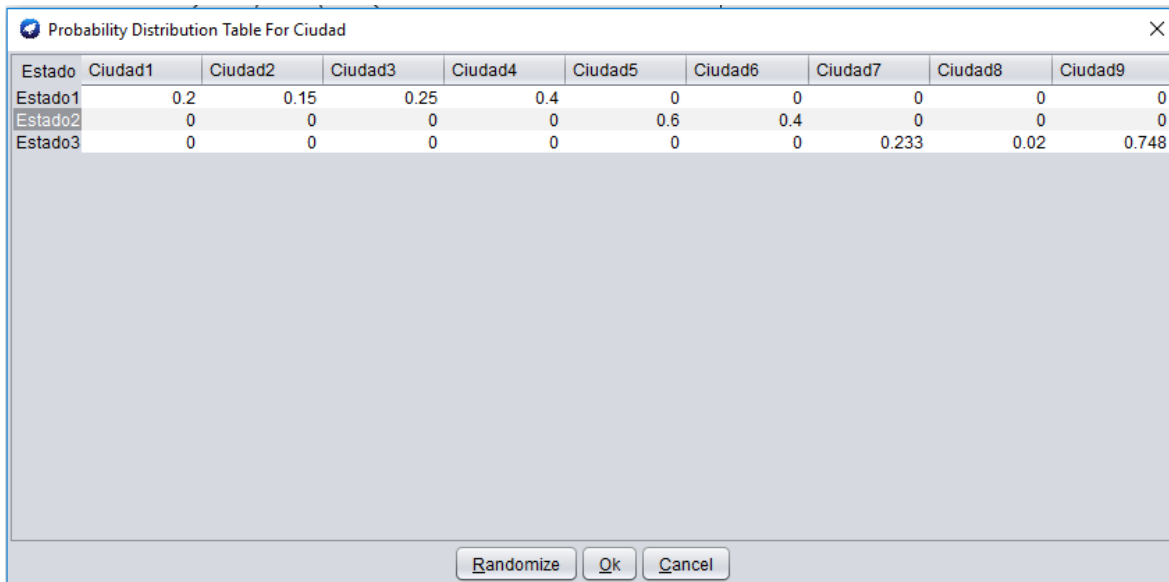


Figura 4.1 Red Bayesiana generada en Weka.

Como se observa en la figura anterior, existen nodos sin relación con alguno otro, y nodos que determinan a otros. El nodo **Vendedor** y el **número 3** (descuento, dado que la herramienta así lo muestra) no tienen relación con ningún nodo, sin embargo no dejan de ser importantes para la generación de datos. Por otro lado el nodo **Estado** determina al nodo **Ciudad** y eso en la realidad es correcto. El nodo **Producto** determina a los nodos: **Cantidad**, **6 (Sector_primario)**, **7 (Resguardo)** y **8 (Resguardo_linea)**. Dado que esa determinación es correcta tanto en la realidad como en el modelo de negocio de SIFEI.

Para cada nodo existe una tabla de probabilidad en la cual se determina los posibles valores del atributo así como la probabilidad de que cada uno aparezca. La Figura 4.2 muestra la tabla de probabilidad para el nodo **Ciudad** dado que se observa con más claridad la premisa de: el estado determina la ciudad.



Estado	Ciudad1	Ciudad2	Ciudad3	Ciudad4	Ciudad5	Ciudad6	Ciudad7	Ciudad8	Ciudad9
Estado1	0.2	0.15	0.25	0.4	0	0	0	0	0
Estado2	0	0	0	0	0.6	0.4	0	0	0
Estado3	0	0	0	0	0	0	0.233	0.02	0.748

Figura 4.2 Tabla de probabilidad del nodo Ciudad.

Como se observa en la figura anterior, existen (para este caso) tres estados y nueve ciudades posibles, sin embargo la probabilidad de que la Ciudad1 aparezca en dos estados no representa un modelo real (por ejemplo que la ciudad de Xalapa pertenezca a Veracruz y a Puebla), es por eso que las ciudades 1 a 4 pertenecen al Estado1; las ciudades 5 y 6 al Estado2 y por último las ciudades 7 a 9 al Estado3.

Para consultar el resto de tablas de probabilidad, ver Anexo 2.

Tomando como base la red Bayesiana (Figura 4.1) se procedió con la generación de 10 conjuntos de datos con **100,000** registros cada uno, dando como resultado **1,000,000** de datos de pruebas. Para cada conjunto los valores de las tablas de probabilidad se generaron datos pseudoaleatorios con el fin de tener como resultado 10 posibles escenarios de tipos de clientes para SIFEI.

A cada conjunto de datos se le aplicaron las mismas pruebas que en el *sprint* 1 (implementación del algoritmo *K-Means* con distancia Euclidiana y Manhattan y la implementación del algoritmo EM) Las tablas 4.1 a 4.3 muestran los resultados de dichas implementaciones sobre el conjunto 1.

Tabla 4.1 Resultados para K-Means utilizando distancia Euclidiana.

NÚMERO DE GRUPOS	ERROR CUADRÁTICO	TIEMPO DE EJECUCIÓN	DISTRIBUCIÓN DE GRUPOS	
2	322502.0	1.09s	64051 (64%)	35949 (36%)
3	300849.0	0.66s	56365 (56%)	13086 (13%)
			30549 (31%)	
4	259363.0	0.94s	42191 (42%)	10590 (11%)
			23660 (24%)	23559 (24%)
5	249976.0	1.27s	30618 (31%)	20460 (20%)
			25364 (25%)	14858 (15%)
			8700 (9%)	
6	240812.0	1.45s	18740 (19%)	9005 (9%)
			20372 (20%)	21989 (22%)
			12015 (12%)	17879 (18%)

NÚMERO DE GRUPOS	ERROR CUADRÁTICO	TIEMPO DE EJECUCIÓN	DISTRIBUCIÓN DE GRUPOS	
7	244555.0	1.28s	25285 (25%)	23860 (24%)
			18385 (18%)	9545 (10%)
			5874 (6%)	8241 (8%)
			8810 (9%)	
8	226154.0	1.19s	26295 (26%)	21544 (22%)
			17077 (17%)	5120 (5%)
			6529 (7%)	7773 (8%)
			7912 (8%)	7750 (8%)
9	215418.0	2.36s	23257 (23%)	7374 (7%)
			16554 (17%)	6573 (7%)
			6981 (7%)	7626 (8%)
			7792 (8%)	8336 (8%)

NÚMERO DE GRUPOS	ERROR CUADRÁTICO	TIEMPO DE EJECUCIÓN	DISTRIBUCIÓN DE GRUPOS	
10	208068.0	1.49s	15507 (16%)	
			21518 (22%)	7373 (7%)
			15869 (16%)	6555 (7%)
			7288 (7%)	10303 (10%)
			4350 (4%)	8150 (8%)
			9544 (10%)	9050 (9%)

Tabla 4.2 Resultados para K-Means utilizando distancia Manhattan.

NÚMERO DE GRUPOS	ERROR CUADRÁTICO	TIEMPO DE EJECUCIÓN	DISTRIBUCIÓN DE GRUPOS	
2	322502.0	0.64s	64051 (64%)	35949 (36%)
3	300849.0	0.59s	56365 (56%)	13086 (13%)
			30549 (31%)	

NÚMERO DE GRUPOS	ERROR CUADRÁTICO	TIEMPO DE EJECUCIÓN	DISTRIBUCIÓN DE GRUPOS	
4	259363.0	0.94s	42191 (42%)	10590 (11%)
			23660 (24%)	23559 (24%)
5	249976.0	1.22s	30618 (31%)	20460 (20%)
			25364 (25%)	14858 (15%)
			8700 (9%)	
6	240812.0	1.36s	18740 (19%)	9005 (9%)
			20372 (20%)	21989 (22%)
			12015 (12%)	17879 (18%)
7	244555.0	1.09s	25285 (25%)	23860 (24%)
			18385 (18%)	9545 (10%)
			5874 (6%)	8241 (8%)
			8810 (9%)	

NÚMERO DE GRUPOS	ERROR CUADRÁTICO	TIEMPO DE EJECUCIÓN	DISTRIBUCIÓN DE GRUPOS	
8	226154.0	0.94s	26295 (26%)	21544 (22%)
			17077 (17%)	5120 (5%)
			6529 (7%)	7773 (8%)
			7912 (8%)	7750 (8%)
9	215418.0	1.28s	23257 (23%)	7374 (7%)
			16554 (17%)	6573 (7%)
			6981 (7%)	7626 (8%)
			7792 (8%)	8336 (8%)
			15507 (16%)	
10	208068.0	3s	21518 (22%)	7373 (7%)
			15869 (16%)	6555 (7%)
			7288 (7%)	10303 (10%)

NÚMERO DE GRUPOS	ERROR CUADRÁTICO	TIEMPO DE EJECUCIÓN	DISTRIBUCIÓN DE GRUPOS	
			4350 (4%)	8150 (8%)
			9544 (10%)	9050 (9%)

Tabla 4.3 Resultados para EM.

NÚMERO DE GRUPOS	VEROSIMILITUD	TIEMPO DE EJECUCIÓN	DISTRIBUCIÓN DE GRUPOS	
2	-8.15483	18.44s	44097 (44%)	55903 (56%)
3	-7.68694	22.84s	38989 (39%)	30204 (30%)
			30807 (31%)	
4	-7.54061	60.09s	18860 (19%)	30204 (30%)
			30453 (30%)	20483 (20%)
5	-7.29826	76.94s	30204 (30%)	24940 (25%)
			16705 (17%)	12733 (13%)

NÚMERO DE GRUPOS	VEROSIMILITUD	TIEMPO DE EJECUCIÓN	DISTRIBUCIÓN DE GRUPOS	
			15418 (15%)	
6	-7.23843	82.88s	30204 (30%)	24940 (25%)
			7159 (7%)	5574 (6%)
			15418 (15%)	16705 (17%)
7	-7.15513	101.52s	24763 (25%)	15427 (15%)
			16705 (17%)	12733 (13%)
			168 (0%)	16914 (17%)
			13290 (13%)	
8	-7.09197	131.91	24827 (25%)	15427 (15%)
			16705 (17%)	12733 (13%)
			104 (0%)	8129 (8%)
			13147 (13%)	8928 (9%)

NÚMERO DE GRUPOS	VEROSIMILITUD	TIEMPO DE EJECUCIÓN	DISTRIBUCIÓN DE GRUPOS	
9	-7.09535	158.62	13290 (13%)	2 (0%)
			24926 (25%)	7159 (7%)
			5574 (6%)	3 (0%)
			16914 (17%)	15427 (15%)
			16705 (17%)	
10	-7.06862	129.22	13290 (13%)	2 (0%)
			24926 (25%)	3778 (4%)
			5511 (6%)	3 (0%)
			16914 (17%)	15427 (15%)
			16705 (17%)	3444 (3%)

En las tablas anteriores se observan las distribuciones de los grupos, así como el error cuadrático (en el caso de *K-Means*), la verosimilitud (en el caso de EM) y el tiempo de ejecución del algoritmo. Para consultar resto de resultados para cada conjunto (2 al 10) ver en el Anexo 3.

Una vez terminada la implementación de los algoritmos, se determinaron los resultados óptimos para cada conjunto y cada algoritmo. La Tabla 4.4 presenta dichos resultados.

Tabla 4.4 Resultados óptimos para clustering.

CONJUNTO	K-MEANS			EM		
	DISTANCIA EUCLIDIANA	K ÓPTIMA	DISTANCIA MANHATTAN	K ÓPTIMA	VEROSIMILITU D	K ÓPTIMA
1	240812	6	240812	6	-7.54061	4
2	266736	5	286151	4	-7.40553	5
3	241636	3	241636	3	-6.87195	4
4	241636	3	241636	3	-6.87195	4
5	232530	5	241636	3	-6.87195	4
6	206557	3	206557	3	-6.18767	3

CONJUNTO	K-MEANS			EM		
	DISTANCIA EUCLIDIANA	K ÓPTIMA	DISTANCIA MANHATTAN	K ÓPTIMA	VEROSIMILITU D	K ÓPTIMA
7	234644	4	241636	3	-6.87195	4
8	241636	3	241636	3	-6.87195	4
9	232530	5	217362	6	-6.66251	5
10	241636	3	232530	5	-6.73506	6

Para cada conjunto con cada resultado óptimo se guardó con su etiqueta de clase para su posterior análisis con dos algoritmos de clasificación:

- **Árbol de decisión J48.**
- **Red Bayesiana *BayesNet*.**

Una vez realizada la implementación de los algoritmos de clasificación mencionados anteriormente, los resultados de la precisión para cada algoritmo de clasificación se muestran en las tablas 4.5 a 4.6.

Tabla 4.5 Resultados algoritmo J48.

CONJUNTO	J48					
	<i>K-Means</i> Euclidiana	<i>k</i> óptima	<i>K-Means</i> Manhattan	<i>k</i> óptima	EM	<i>k</i> óptima
1	99.90%	6	99.90%	6	100%	4
2	99.63%	5	99.77%	4	99.98%	5
3	99.99%	3	99.99%	3	100%	4
4	99.99%	3	99.99%	3	100%	4
5	99.99%	5	99.99%	3	100%	4
6	99.95%	3	99.95%	3	100%	3
7	99.98%	4	99.99%	3	100%	4
8	99.99%	3	99.99%	3	100%	4
9	99.99%	5	99.89%	6	99.99%	5
10	99.99%	3	99.99%	5	100%	6

Tabla 4.6 Resultados algoritmo BayesNet.

CONJUNTO	BAYESNET					
	<i>K-Means</i> Euclidiana	<i>k</i> óptima	<i>K-Means</i> Manhattan	<i>k</i> óptima	EM	<i>k</i> óptima
1	99.09%	6	99.09%	6	100%	4
2	98.19%	5	98.80%	4	99.94%	5
3	99.95%	3	99.95%	3	100%	4
4	99.95%	3	99.95%	3	100%	4
5	99.95%	5	99.95%	3	100%	4
6	99.68%	3	99.68%	3	100%	3
7	99.95%	4	99.95%	3	100%	4
8	99.95%	3	99.95%	3	100%	4
9	99.95%	5	99.22%	6	99.99%	5
10	99.95%	3	99.95%	5	100%	6

Todas las pruebas oscilan en un rango de precisión del 98% al 100%, sin embargo para el caso del algoritmo J48 se extrajeron los árboles formados para cada conjunto, y para el algoritmo *BayesNet* se obtuvieron las redes generadas. Lo anterior para determinar el algoritmo a utilizar, dado que una red Bayesiana representa el fenómeno encontrado en los datos, mientras que un árbol de decisión muestra los caminos posibles para un resultado.

Las figuras 4.3 a 4.5 muestran los árboles generados para el conjunto 2 por el algoritmo J48.

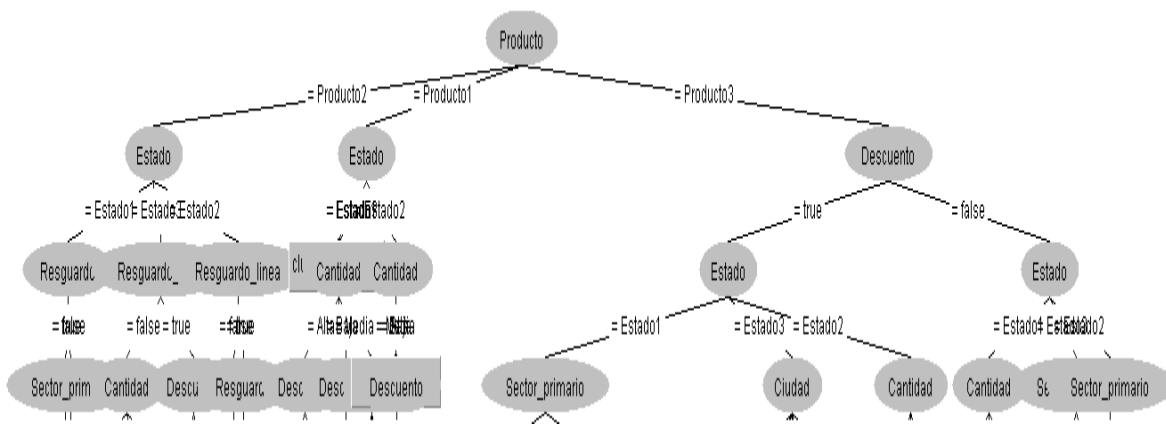


Figura 4.3 Sección del árbol generado para el algoritmo K-Means con distancia Euclidiana.

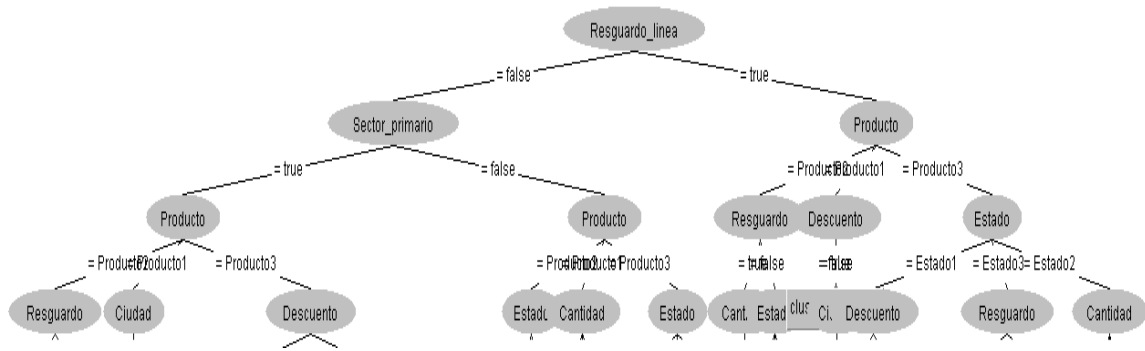


Figura 4.4 Sección del árbol generado para el algoritmo K-Means con distancia Manhattan.

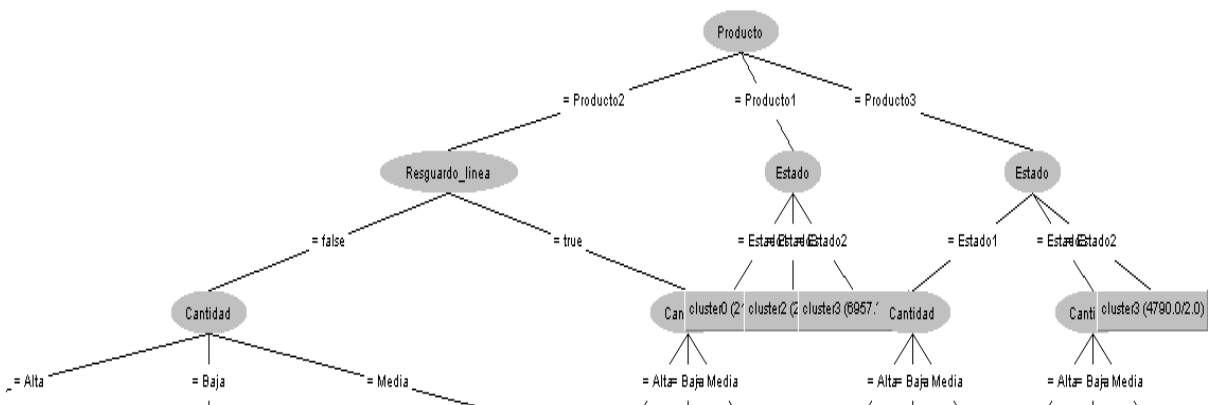


Figura 4.5 Árbol generado para el algoritmo EM

Por otro lado las figuras 4.6 a 4.8 muestran las redes generadas para el conjunto 2 por el algoritmo *BayesNet*.

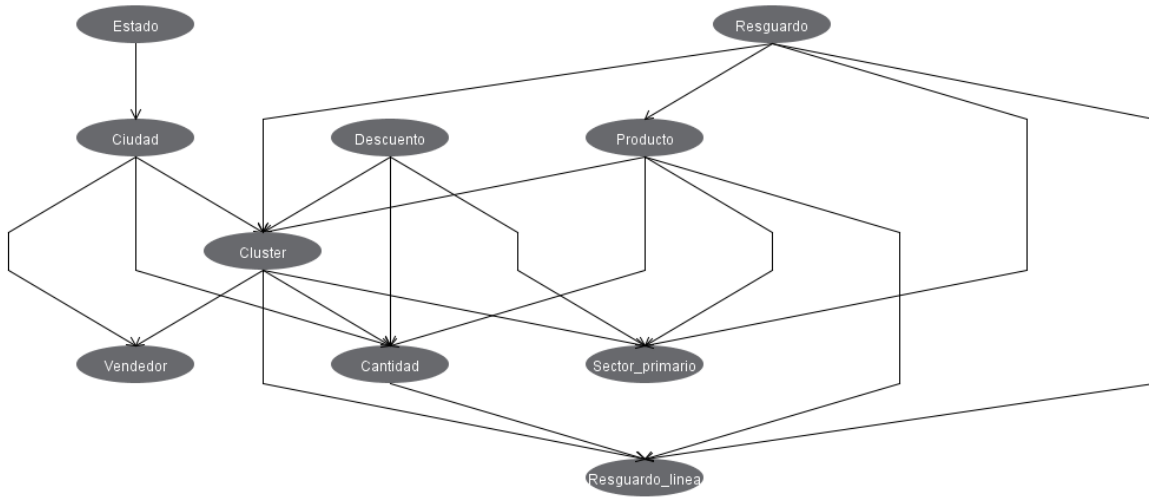


Figura 4.6 Red Bayesiana para el algoritmo K-Means con distancia Euclidiana.

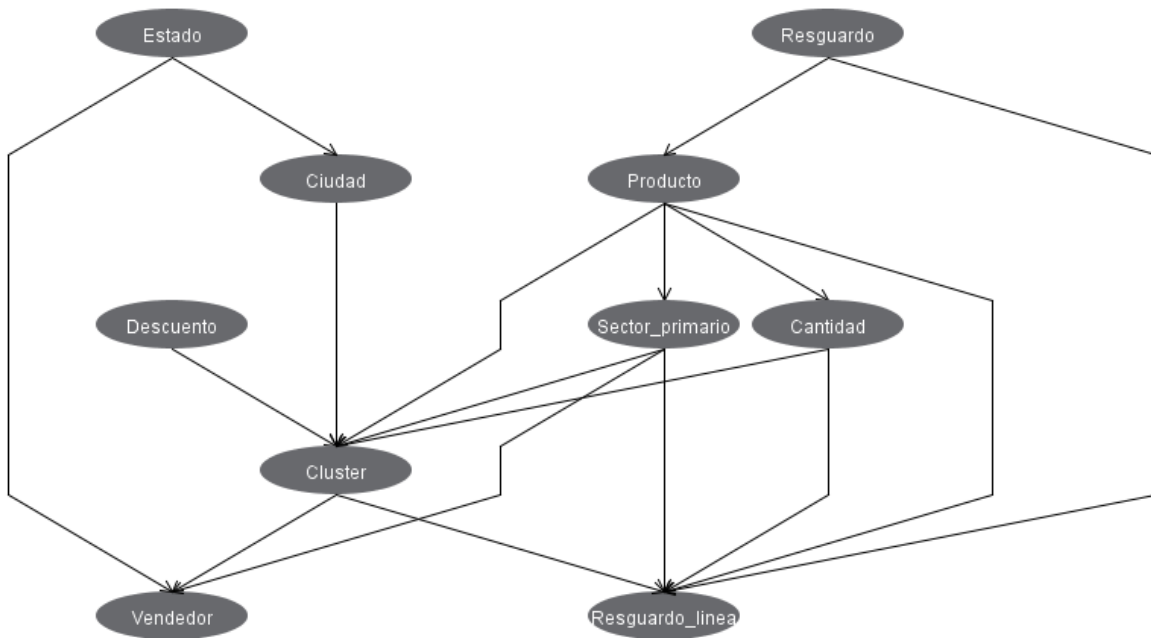


Figura 4.7 Red Bayesiana para el algoritmo K-Means con distancia Euclidiana.

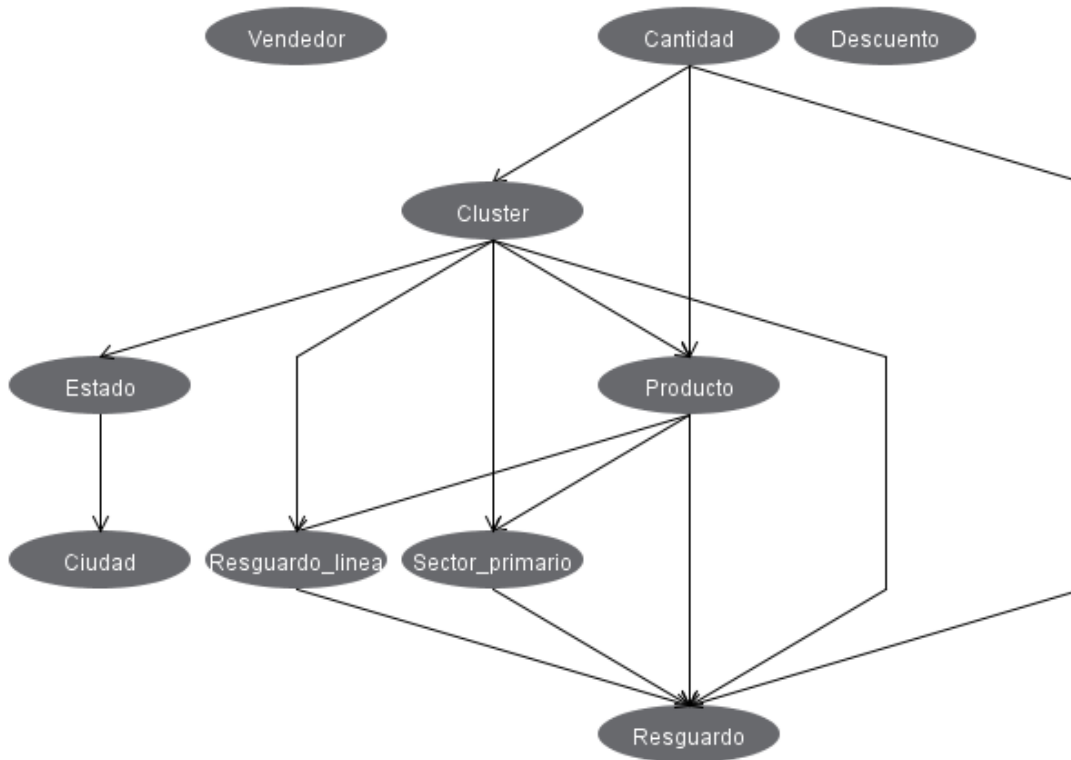


Figura 4.8 Red Bayesiana para el algoritmo EM.

En las figuras anteriores se hace mención de que pertenecen al conjunto número 2, y esto se debe a que es el conjunto seleccionado para reemplazar al conjunto establecido en el *sprint* 1, tal cambio se realiza debido a que presenta mejores (y más confiables) resultados que el anterior. Ver Anexo 4 para consultar los árboles generados para cada conjunto. Ver Anexo 5 para consultar las redes Bayesianas generadas para cada conjunto.

Para determinar qué variante del conjunto 2 se le dará uso, se utilizaron como guías las redes Bayesianas de dicho conjunto, las cuales representan el fenómeno del comportamiento de los datos. La red elegida es la que pertenece al resultado de *K-Means* utilizando la distancia Manhattan, debido a que plasma la realidad del modelo de negocio de SIFEI, es decir, que se asemeja a la realidad que experimenta

la empresa con sus clientes. Y dado que el mejor tiempo de respuesta para este conjunto se obtuvo con el algoritmo J48 se utilizará éste.

Todo lo anterior dio como resultado la modificación de los grupos de clientes, dado que en un inicio eran ocho, ahora son 5 con las características mostradas en la Tabla 4.7.

Tabla 4.7 Características nuevos grupos de clientes.

ATRIBUTO	GRUPO				
	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5
PRODUCTO	Producto 1	Producto 3	Producto 2	Producto 3	Producto 3
CANTIDAD	Baja	Media	Alta	Media	Media
DESCUENTO	Sí	No	Sí	Sí	Sí
CIUDAD	Ciudad 1	Ciudad 7	Ciudad 9	Ciudad 9	Ciudad 9
ESTADO	Estado 1	Estado 3	Estado 3	Estado 3	Estado 3
SECTOR_PRIMARIO	No	No	Sí	Sí	No
RESGUARDO	No	No	Sí	Sí	No
RESGUARDO_LINEA	No	No	Sí	No	No

Y aunque ahora son cinco grupos, las funciones previamente elaboradas no sufrieron cambios radicales, dado que se atendieron a la brevedad y no presentaron complicación alguna dado que todo se desarrolló de forma genérica y no ajustada.

En el caso de las series de tiempo no se requirió de ninguna modificación ya que como se mencionó en el *sprint 4*, la eficacia del algoritmo a utilizar radica en la

naturaleza de los datos a pronosticar, de tal forma que tanto las interfaces como sus funcionalidades permanecen intactas.

4.2 Caso de estudio 2

Como segundo caso de estudio se realizó la prueba del sistema DMMFEI el cual es el resultado de lo desarrollado en los *sprints* pasados y las mejoras obtenidas en el caso de estudio previo.

Como prueba para el sistema, SIFEI planteó una serie de preguntas que requieren de su respuesta mediante el uso de la aplicación DMMFEI, dichas preguntas son:

1. ¿Qué grupos de clientes existen?
2. ¿Cuánto consumirá un cliente o cuando se incrementará su consumo?
3. ¿Cuánto se venderá o cuando se incrementarán las ventas?

Con base en las preguntas anteriores supóngase que el analista de SIFEI Asley Alberto Cristales Pavón requiere analizar los datos de nuevos clientes, así como de clientes ya existentes en la empresa. Cabe mencionar que para fines de este caso de estudio, el analista inició sesión con usuarios predefinidos en la base de datos y solamente se muestran las interfaces necesarias según el tipo de acción a realizar.

4.2.1 Respuesta pregunta 1

Para dar respuesta a la pregunta número uno el analista utiliza el sistema DMMFEI ha iniciado sesión como usuario de tipo clasificador, mostrando la interfaz presentada en la Figura 4.9.

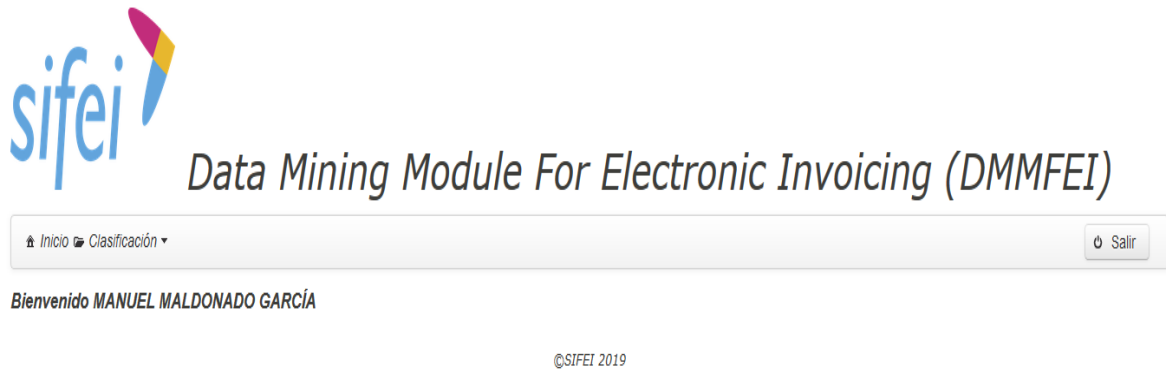


Figura 4.9 Interfaz de inicio para el usuario clasificador.

Dado que el analista requiere consultar los tipos de clientes, seleccionó la opción “Ver grupos” del menú “Clasificación”, tal y como se muestra en la Figura 4.10.



Figura 4.10 Selección de la opción “Ver grupos”.

Una vez que seleccionó la opción correspondiente, el sistema le muestra los diferentes grupos de clientes existentes. La Figura 4.11 muestra el resultado.



Data Mining Module For Electronic Invoicing (DMMFEI)

Inicio Clasificación Sair

Grupos de clientes

Grupo 1	Producto:	Producto 3
Grupo 2	Cantidad:	Baja
Grupo 3	Descuento:	Sí
Grupo 4	Ciudad:	Ciudad 4
Grupo 5	Estado:	Estado 1
	Sector primario:	Sí
	Resguardo:	Sí
	Resguardo en línea:	Sí
	Recomendación:	Ofrecer descuentos de 30%

Figura 4.11 Visualización de grupos de clientes.

Como se observa en la figura anterior, se presentan los cinco grupos de clientes encontrados en el caso de estudio anterior, así como sus características principales y la recomendación que el usuario administrador determine (en este caso esa recomendación está predefinida).

Si bien hasta este punto la respuesta se respondió, el sistema DMMFEI también tiene la capacidad de clasificar a un nuevo cliente, de tal forma que al determinar el grupo al que pertenece se llegan a efectuar acciones mercantiles que beneficien tanto al cliente como a SIFEI. Por tal motivo la Figura 4.12 muestra la interfaz de clasificación de clientes en la cual el analista de SIFEI tipifica a un cliente nuevo.



Data Mining Module For Electronic Invoicing (DMMFEI)

Actualizado
2 clientes por clasificar

Inicio Clasificación Sair

Clasificar

Cientes por clasificar

(1 of 1) 1 10

	Id	Nombre	Vendedor	Cantidad	Producto	Ciudad	Estado
<input type="checkbox"/>	24	Cliente 7	Vendedor5	Alta	Producto2	Ciudad9	Estado3
<input type="checkbox"/>	9	Cliente 8	Vendedor2	Media	Producto3	Ciudad1	Estado1

(1 of 1) 1 10

Clasificar

©SIFEI 2019

Figura 4.12 Clasificación de clientes.

El analista determina si quiere clasificar a todos los clientes o solamente a uno, en este caso clasificará solamente al Cliente 7, seleccionándolo y pulsando el botón “Clasificar” tal y como se muestra en las figura 4.13 y 4.14.

sifei Data Mining Module For Electronic Invoicing (DMMFEI)

Inicio Clasificación Sair

Clasificar

Confirmación ✕

¿Clasificar cliente(s) seleccionado(s)?

	Id	Nombre	Vendedor	Cantidad	Producto	Ciudad	Estado
<input checked="" type="checkbox"/>	24	Cliente 7	Vendedor5	Alta	Producto2	Ciudad9	Estado3
<input type="checkbox"/>	9	Cliente 8	Vendedor2	Media	Producto3	Ciudad1	Estado1

(1 of 1) 1 10

Clasificar

©SIFEI 2019

Figura 4.13 Confirmación de clasificación de un cliente.

sifei Data Mining Module For Electronic Invoicing

Inicio Clasificación

Clasificar

Cientes por clasificar

(1 of 1) 1 10

<input type="checkbox"/>	Id	Nombre	Vendedor	Cantidad	Producto	Ciudad	Estado
<input type="checkbox"/>	9	Cliente 8	Vendedor2	Media	Producto3	Ciudad1	Estado1

(1 of 1) 1 10

Clasificar

©SIFEI 2019

Correcto
Clientes clasificados: 1

Actualizado
1 clientes por clasificar

Correcto
Resultados en la sección 'Ver clasificaciones'

Figura 4.14 Resultado de clasificación.

Una vez que el analista confirma la selección, el sistema automáticamente realiza los procesos necesarios para tipificar a los clientes seleccionados, mostrando al analista el número de clientes clasificados, cuantos faltan de clasificación y en donde observar el resultado de dicha clasificación.

Teniendo al cliente ya tipificado, el analista comprueba el resultado en la opción “Ver clasificaciones” mostrada en la Figura 4.15

Inicio Clasificación Salir

Clasificaciones

Cliente: *

Tipo de búsqueda: * ID Nombre

Buscar

Clientes clasificados							
(1 of 1) 1 10							
Id	Nombre	Vendedor	Cantidad	Producto	Ciudad	Estado	Grupo
30	Cliente 7	Vendedor5	Alta	Producto2	Ciudad9	Estado3	Grupo 2
25	Cliente 3	Vendedor8	Baja	Producto3	Ciudad7	Estado3	Grupo 2
24	Cliente 5	Vendedor5	Alta	Producto1	Ciudad9	Estado3	Grupo 2
22	Cliente 2	Vendedor2	Alta	Producto1	Ciudad2	Estado1	Grupo 1
21	Cliente 6	Vendedor9	Baja	Producto2	Ciudad4	Estado1	Grupo 4
20	Cliente 4	Vendedor1	Alta	Producto2	Ciudad6	Estado2	Grupo 3
19	Cliente 1	Vendedor1	Media	Producto1	Ciudad1	Estado1	Grupo 1

(1 of 1) 1 10

©SIFEI 2019

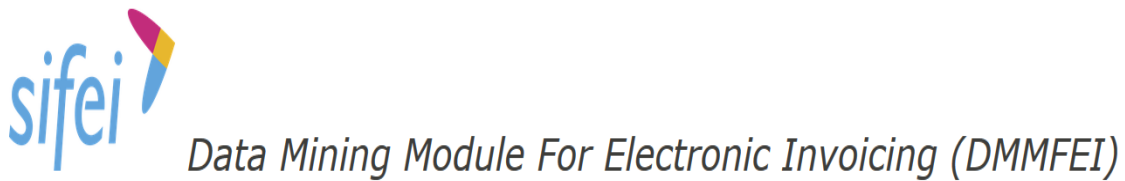
Figura 4.15 Clientes clasificados.

En la figura anterior, se muestran los clientes previamente clasificados, ordenados ascendentemente para su manejo, como primer registro se aprecia el Cliente 7 se clasificó en el Grupo 2.

Con lo anterior el analista tiene las herramientas necesarias para tomar las acciones que más le convengan para el trato hacía el cliente, ya que al conocer los tipos de grupos con sus respectivas recomendaciones realiza su trabajo con eficacia.

4.2.2 Respuesta pregunta 2 y 3

Para dar respuesta a la pregunta número dos el analista utiliza el sistema DMMFEI ha iniciado sesión como usuario de tipo pronosticador, mostrando la interfaz presentada en la Figura 4.16.



Bienvenido LUIS FRANCISCO MALPICA LÓPEZ

©SIFEI 2019

Figura 4.16 Interfaz de inicio para usuario pronosticador

Dado que el analista requiere pronosticar (conocer el comportamiento de las ventas), basta con seleccionar la opción “Ventas” del menú “Pronóstico”, tal y como se muestra en la Figura 4.17.

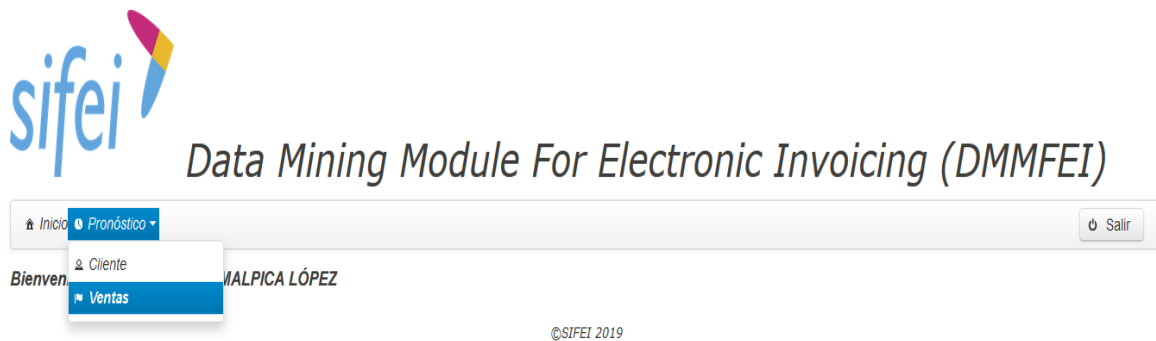


Figura 4.17 Selección de la opción “Ventas”.

Una vez que se ha seleccionado la opción correspondiente, el sistema muestra un formulario para determinar el pronóstico a buscar. La Figura 4.18 presenta la interfaz antes mencionada.



Data Mining Module For Electronic Invoicing (DMMFEI)

Inicio ● Pronóstico ▾ Salir

Pronosticar ventas

Producto: * Todos ▾

Fecha inicio: * Fecha Fin: *

Buscar

©SIFEI 2019

Figura 4.18 Formulario pronosticar ventas.

El asistente de SIFEI ingresa los parámetros solicitados y selecciona el botón “Buscar”, el sistema automáticamente realiza las peticiones y procesos correspondientes para desplegar los resultados obtenidos. La Figura 4.19 muestra dichos resultados.

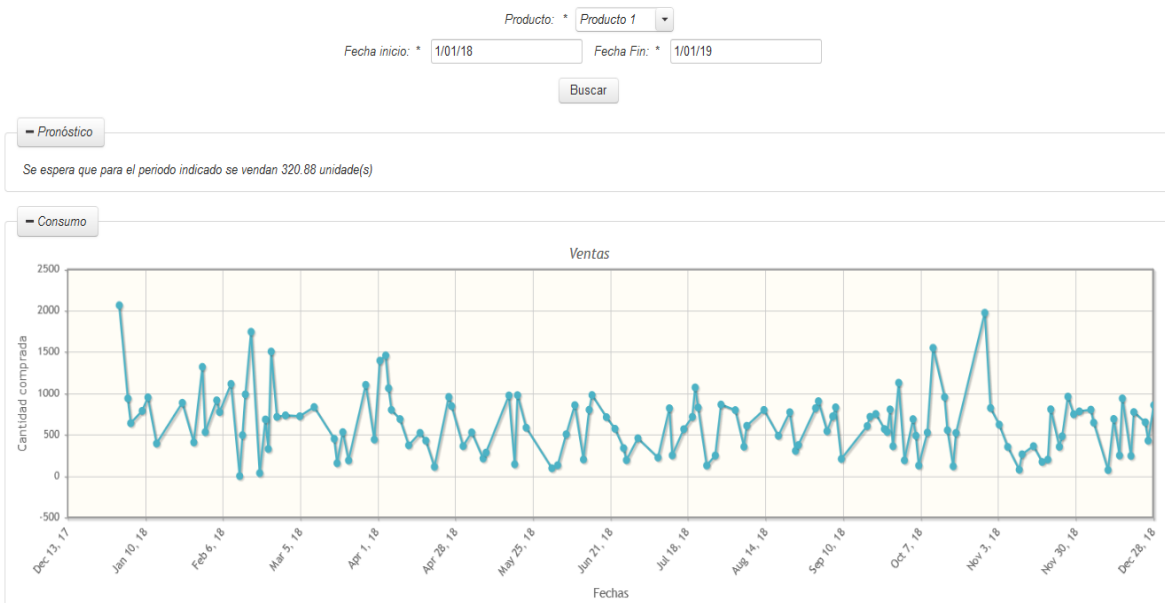


Figura 4.19 Resultados pronosticar ventas.

En la figura anterior se presentan los resultados que constan de un pronóstico y una gráfica que muestra el comportamiento de las ventas para el Producto 1 durante el

periodo de tiempo del 01 de Enero de 2017 al 01 de Enero de 2019, cabe señalar que el sistema solamente encuentra los registros comprendidos en ese intervalo de tiempo si es que existen, y el pronóstico mostrado deberá ser analizado por personal capacitado.

Para responder la tercera pregunta, el asistente seleccionará la opción “Cliente” para ingresar a la interfaz correcta, tal como se muestra en la Figura 4.20.

sifei Data Mining Module For Electronic Invoicing (DMMFEI)

Inicio Pronóstico Salir

Pronosticar consumo de cliente

Cliente: *

Tipo de búsqueda: * ID Nombre

Producto: *

Fecha inicio: * Fecha Fin: *

Buscar

Clientes	
Id	Nombre
Sin resultados	

Seleccionar

Figura 4.20 Formulario para pronosticar el consumo de un cliente.

El asistente de SIFEI ingresa los parámetros requeridos para la búsqueda del cliente a pronosticar y acto seguido selecciona el botón “Buscar”, el sistema realiza las peticiones y procesos necesarios para buscar al cliente que el asistente requiera. La Figura 4.21 muestra el resultado de la búsqueda.



Data Mining Module For Electronic Invoicing (DMMFEI)

Inicio ● Pronóstico ◻ Salir

Pronosticar consumo de cliente

Cliente: *

Tipo de búsqueda: * ID Nombre

Producto: *

Fecha inicio: * Fecha Fin: *

Buscar

Clientes	
Id	Nombre
2	Cliente 2

Seleccionar

Figura 4.21 Resultado de la búsqueda de clientes.

Para determinar el probable consumo del cliente buscado, el asistente selecciona a dicho cliente y acto seguido pulsa el botón “Seleccionar” con esa acción, el sistema ejecuta los procesos necesarios para mostrarle al asistente el resultado esperado, tal y como lo muestra la Figura 4.22.

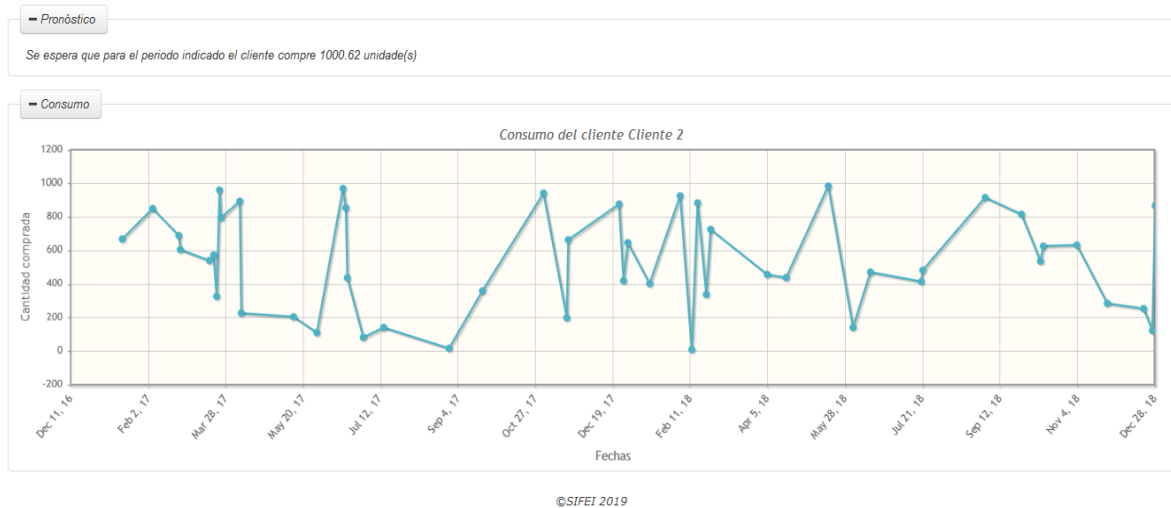


Figura 4.22 Resultados del pronóstico de cliente.

La Figura anterior muestra el consumo esperado del Cliente 2 y de igual forma muestra el comportamiento a lo largo del periodo de tiempo indicado.

Las preguntas dos y tres se contestaron por medio del uso del sistema DMMFEI, otorgándole al asistente las herramientas necesarias para prever escenarios futuros y analizar el comportamiento tanto de las ventas en general, como de un cliente en específico.

Capítulo 5. Conclusiones y recomendaciones

En este capítulo se describen las conclusiones y se indican algunas recomendaciones a futuro para este trabajo de tesis.

5.1 Conclusiones

Dado que en la actualidad la era tecnológica va en aumento, cada vez más y más empresas generan enormes cantidades de información acerca de sus clientes, los cuales sin saberlo desarrollan patrones que con las técnicas adecuadas llegan a ser descubiertos, explotados y reutilizados para conveniencia de la empresa y del cliente en cuestión. La minería de datos es la encargada de encontrar esos patrones inmersos en los repositorios de información, sin embargo existen inconvenientes que retrasan la exploración, tal es el caso de: datos faltantes, datos inconsistentes, ruido, por mencionar algunos.

Las empresas que se dedican al rubro de la facturación electrónica también guardan información de sus clientes, que a su vez generan patrones que la minería de datos llega a encontrar y que interpretados por un experto de negocio se conviertan en conocimiento útil para la empresa. SIFEI es una empresa dedicada a la venta/renta de servicios de facturación electrónica, y como toda organización presenta altas y bajas en sus ventas pero debido a la cantidad de clientes y transacciones que presenta, es una tarea casi imposible el determinar patrones tanto en el comportamiento de sus clientes como de sus ventas, por tal motivo se desarrolló el sistema DMMFEI que incorpora técnicas de minería de datos para ser un apoyo en la toma de decisiones mercantiles o administrativas que los directivos de SIFEI consideren prudentes para mejorar sus servicios y campañas publicitarias.

Se realizó un análisis exhaustivo para determinar las opciones tecnológicas adecuadas para la realización del sistema DMMFEI, que hasta este punto se considera una opción genérica ya que los datos a analizar se llegan a encontrar en repositorios de empresas que se dediquen a la facturación electrónica, sin embargo se adecuó para su uso con SIFEI, ya que las estructuras de base de datos que se utilizan para la implementación de minería de datos serán pobladas por el sistema

principal de SIFEI asegurando que la información proporcionada se encuentre “limpia”. También cabe mencionar que al tratarse de un sistema independiente del trabajo transaccional de los repositorios de SIFEI, el sistema DMMFEI no presenta riesgo alguno para la información que la empresa almacene, y por lo tanto el mantenimiento se facilita dada la estructura arquitectónica del sistema DMMFEI (capas).

Con el uso del sistema propuesto aunado a técnicas de minería de datos y por los resultados obtenidos en los dos casos de estudio realizados, la empresa SIFEI tendrá un apoyo para la toma de decisiones que tenga un sustento profesional dadas las personas involucradas en este tema de tesis (director, co-director y director de estancias).

5.2 Recomendaciones

Debido a que SIFEI no proporcionó información referente a sus clientes y sus ventas, se tuvo que implementar la generación de datos sintéticos, en primera instancia con una herramienta *Web* y después mediante redes Bayesianas, por tal motivo el tiempo de análisis y desarrollo se extendió más de lo considerado, sin embargo se obtuvieron buenos resultados puesto que se respondieron las preguntas que SIFEI necesitaba.

Se recomienda proveer de más información para mejorar los modelos de minería de datos ya que se pudo haber omitido alguno importante.

Productos académicos

En congreso nacional

Malpica-Castro Luis Francisco, Sánchez-Cervantes José Luis, Olivares-Zepahua Beatriz Alejandra, Alor-Hernández Giner, López-Martínez Ignacio. “*Module for the detection of consumption patterns based on customer analysis of electronic invoicing services*”, 2nd Conference on Computer Science and Computer Engineering (COCSCE), 2019, Tijuana, B.C.N.

Revista Aristas: Investigación Básica y Aplicada, Vol. 7, Núm. 14.

ISSN 2007-9478

Referencias

- [1] C. P. López, *Minería de datos: técnicas y herramientas*. Editorial Paraninfo, 2007.
- [2] W. Astriani and R. Trisminingsih, “Extraction, transformation, and loading (ETL) module for hotspot spatial data warehouse using Geokettle,” *Procedia Environ. Sci.*, vol. 33, pp. 626–634, 2016.
- [3] S. Gigimol and S. John, “A Survey on Different Types of Recommendation Systems,” *Eng. Sci.*, vol. 1, no. 4, pp. 111–113, 2016.
- [4] H. Zhou and K. Hirasawa, “Evolving temporal association rules in recommender system,” *Neural Comput. Appl.*, Oct. 2017.
- [5] M. J. Pazzani and D. Billsus, “Content-Based Recommendation Systems,” in *The Adaptive Web: Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 325–341.
- [6] J. Ben Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative Filtering Recommender Systems,” in *The Adaptive Web: Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 291–324.
- [7] R. Burke, “Hybrid Web Recommender Systems,” in *The Adaptive Web: Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 377–408.
- [8] C. C. Aggarwal, “Knowledge-Based Recommender Systems,” in *Recommender Systems: The Textbook*, Cham: Springer International Publishing, 2016, pp. 167–197.
- [9] J. Han, J. Pei, and M. Kamber, “Data mining: concepts and techniques,” 3th ed., Waltham, Massachusetts: Elsevier, 2011, p. 8.
- [10] M. Ismail, M. M. Ibrahim, Z. M. Sanusi, and M. Nat, “Data Mining in Electronic Commerce: Benefits and Challenges,” *Int. J. Commun. Netw. Syst. Sci.*, vol. 8, no. 12, p. 501, 2015.
- [11] M. Pérez, “Minería de datos a través de ejemplos,” 20th ed., Madrid, España: RC Libros, 2014, pp. 343–344.
- [12] R. A. Española, “Facturación,” 2018. [Online]. Available:

- <http://dle.rae.es/?id=HTrr7EH>. [Accessed: 17-Feb-2018].
- [13] R. A. Española, “Factura,” 2018. [Online]. Available: <http://dle.rae.es/?id=HTrRse1>. [Accessed: 17-Feb-2018].
- [14] G. de España, “Facturación electrónica,” 2018. [Online]. Available: <http://www.facturae.gob.es/factura-electronica/Paginas/factura-electronica.aspx>. [Accessed: 17-Feb-2018].
- [15] E. S. Balado, “La Nueva Era Del Comercio/the New Era of Commerce: El Comercio Electrónico, Las Tic’s Al Servicio De La Gestión Empresarial,” Ideaspropias Editorial SL, 2005, p. 1.
- [16] N. Pereira and S. Varma, “Survey on content based recommendation system,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 7, no. 1, pp. 281–284, 2016.
- [17] K. Ishibashi, T. Iwasaki, S. Otomasa, and K. Yada, “Model Selection for Financial Statement Analysis: Variable Selection with Data Mining Technique,” *Procedia Comput. Sci.*, vol. 96, pp. 1681–1690, 2016.
- [18] F. M. Bianchi, A. Rizzi, A. Sadeghian, and C. Moiso, “Identifying user habits through data mining on call data records,” *Eng. Appl. Artif. Intell.*, vol. 54, pp. 49–61, 2016.
- [19] S. Zhang, W. Xiong, W. Ni, and X. Li, “Value of big data to finance: observations on an internet credit Service Company in China,” *Financ. Innov.*, vol. 1, no. 1, p. 17, 2015.
- [20] C. Long, R. C.-W. Wong, and V. J. Wei, “Profit Maximization with Sufficient Customer Satisfactions,” *ACM Trans. Knowl. Discov. Data*, vol. 12, no. 2, p. 19:1--19:34, 2018.
- [21] A. Dursun and M. Caber, “Using data mining techniques for profiling profitable hotel customers: An application of RFM analysis,” *Tour. Manag. Perspect.*, vol. 18, pp. 153–160, 2016.
- [22] M. B. Rodrigues, G. O. M. da Silva, and F. A. Durão, “User Models Development Based on Cross-Domain for Recommender Systems,” in *Proceedings of the 22Nd Brazilian Symposium on Multimedia and the Web*, 2016, pp. 363–366.
- [23] A. Keramati, H. Ghaneei, and S. M. Mirmohammadi, “Developing a prediction model for customer churn from electronic banking services using data mining,” *Financ. Innov.*, vol. 2, no. 1, p. 10, 2016.
- [24] C. He, D. Parra, and K. Verbert, “Interactive recommender systems: A survey

- of the state of the art and future research challenges and opportunities,” *Expert Syst. Appl.*, vol. 56, pp. 9–27, 2016.
- [25] T. K. Paradarami, N. D. Bastian, and J. L. Wightman, “A hybrid recommender system using artificial neural networks,” *Expert Syst. Appl.*, vol. 83, pp. 300–313, 2017.
- [26] A. Gupta *et al.*, “Mesa: A Geo-replicated Online Data Warehouse for Google’s Advertising System,” *Commun. ACM*, vol. 59, no. 7, pp. 117–125, 2016.
- [27] L. F. Brunialti, S. M. Peres, V. Freire, and C. A. M. Lima, “Machine Learning in Textual Content-Based Recommendation Systems: A Systematic Review,” in *Proceedings of the Annual Conference on Brazilian Symposium on Information Systems: Information Systems: A Computer Socio-Technical Perspective - Volume 1*, 2015, p. 28:203--28:210.
- [28] E. Agirre-Basurko, G. Ibarra-Berastegi, and I. Madariaga, “Regression and multilayer perceptron-based models to forecast hourly O₃ and NO₂ levels in the Bilbao area,” *Environ. Model. Softw.*, vol. 21, no. 4, pp. 430–446, Apr. 2006.
- [29] S. Cankurt and A. Subasi, “Comparison of linear regression and neural network models forecasting tourist arrivals to Turkey,” 2012.
- [30] A. B. Nassif, D. Ho, and L. F. Capretz, “Towards an early software estimation using log-linear regression and a multilayer perceptron model,” *J. Syst. Softw.*, vol. 86, no. 1, pp. 144–160, 2013.
- [31] M. Brocato, “Mockaroo,” 2019, 2019. [Online]. Available: <https://mockaroo.com/help/about>.

Anexo 1

Carta de usuario.



ASUNTO: Carta de usuario

DR. JOSE LUIS SANCHEZ CERVANTES
 PROFESOR INVESTIGADOR
 División de Estudios de Posgrado e Investigación
 Instituto Tecnológico de Orizaba
 PRESENTE

Orizaba, Ver. 05 de julio de 2019

Respetable investigador:

Por medio de la presente me dirijo a usted en primer lugar para felicitarlo por su labor de investigación enfocada a la minería de datos, entre la cual se encuentra el desarrollo de la tesis titulada "MODULO DE DETECCION DE PATRONES DE CONSUMO A PARTIR DEL ANALISIS DE CLIENTES DE SERVICIOS DE FACTURACION ELECTRONICA". Y en segundo lugar para hacerle saber la intención de la implementación del módulo desarrollado por el Ing. Luis Francisco Malpica Castro en la tesis antes mencionada.

El módulo será utilizado para la toma de decisiones mercantiles y/o administrativas según se requiera, de igual forma se usará para la perfilación de nuestros clientes permitiéndonos mejorar nuestras campañas publicitarias enfocándolas a grupos de clientes en específico.

ATENTAMENTE

ING. ASLEY ALBERTO CRISTALES PAVON
 JEFE DE CAPITAL HUMANO Y SERVICIOS GENERALES

SIFEI



Anexo 2

Para consultar el Anexo 2 – Tablas de probabilidad del caso de estudio 1, utilizar el siguiente código QR.



Anexo 2 Figura 1 Código QR para consultar el Anexo 2.

Anexo 3

Para consultar el Anexo 3 – Tablas de distribución de grupos, utilizar el siguiente código QR.



Anexo 3 Figura 1 Código QR para consultar el Anexo 3.

Anexo 4

Para consultar el Anexo 4 – Árboles de decisión generados para cada conjunto, utilizar el siguiente código QR.



Anexo 4 Figura 1 Código QR para consultar el Anexo 4.

Anexo 5

Para consultar el Anexo 5 – Redes Bayesianas generadas para cada conjunto, utilizar el siguiente código QR.



Anexo 5 Figura 1 Código QR para consultar el Anexo 5.