



**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERA  
EN SISTEMAS Y COMPUTACIÓN**

**“DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE  
CLIENTES Y PROVEEDORES PARA LA EMPRESA ALDERA DISEÑO”**

**AUTOR:**

**EDGAR ALEXANDER ZURITA ULLOA**

**DIRECTOR:**

**ING. DAMIÁN NICOLALDE**

**QUITO, Septiembre 2017**

**CONTENIDO**

ÍNDICE DE FIGURAS .....	iv
ÍNDICE DE TABLAS .....	ix
RESUMEN.....	x
DEDICATORIA .....	xi
AGRADECIMIENTO .....	xii
CAPÍTULO 1. MARCO TEÓRICO .....	1
1.1. LA EMPRESA .....	1
1.1.1. El giro del negocio .....	1
1.1.2. Misión .....	1
1.1.3. Visión .....	1
1.1.4. Organigrama de la empresa.....	1
1.1.5. Proveedores .....	2
1.1.6. Clientes.....	2
1.2. SISTEMA .....	3
1.2.1. Aplicación de escritorio .....	3
1.2.2. Lenguaje de programación.....	4
1.2.3. Motores de bases de datos.....	5
1.3. INGENIERÍA DE REQUERIMIENTOS .....	7
1.3.1. Historias de Usuarios .....	7
1.4. METODOLOGÍA .....	9
1.4.1. Extreme Programming (XP) .....	9
1.5. PROGRAMACIÓN EN CAPAS .....	19
1.6. BASE DE DATOS RELACIONAL.....	19
1.7. DIAGRAMAS DE CLASES.....	21
1.8. DIAGRAMAS DE SECUENCIA .....	22

1.9.	PRUEBAS DE SOFTWARE .....	23
1.10.	HERRAMIENTAS DE DESARROLLO .....	25
1.10.1.	JAVA.....	25
1.10.2.	PostgreSQL .....	26
CAPÍTULO 2.	DEFINICIÓN DE REQUERIMIENTOS .....	30
2.1.	INTERACCIÓN CON EL CLIENTE.....	30
2.1.1.	Historias de Usuarios .....	30
2.2.	REQUERIMIENTOS .....	33
2.2.1.	Diagramas .....	33
CAPÍTULO 3.	PLANIFICACIÓN Y ELABORACIÓN .....	58
3.1.	PLANIFICACION .....	58
3.1.1.	ITERACION 1 .....	58
3.1.2.	ITERACION 2 .....	59
3.1.3.	ITERACION 3 .....	59
3.1.4.	ITERACION 4 .....	60
3.1.5.	ITERACION 5 .....	60
3.1.6.	ITERACION 6 .....	61
3.2.	ELABORACIÓN DEL SISTEMA .....	62
3.2.1.	Modelo Físico .....	62
3.2.2.	Modelo Relacional .....	63
3.3.	ITERACIÓN 1 – Ingresar al sistema.....	63
3.3.1.	Diseño .....	63
3.3.2.	Desarrollo.....	64
3.3.3.	Pruebas .....	66
3.4.	ITERACIÓN 2 – Administrar Clientes .....	69

3.4.1.	Diseño .....	69
3.4.2.	Desarrollo.....	71
3.4.3.	Pruebas .....	73
3.5.	ITERACIÓN 3 – Administrar Proveedores .....	81
3.5.1.	Diseño .....	81
3.5.2.	Desarrollo.....	83
3.5.3.	Pruebas .....	84
3.6.	ITERACIÓN 4 – Administrar Productos .....	91
3.6.1.	Diseño .....	92
3.6.2.	Desarrollo.....	93
3.6.3.	Pruebas .....	94
3.7.	ITERACIÓN 5 – Administrar Ventas .....	101
3.7.1.	Diseño .....	101
3.7.2.	Desarrollo.....	103
3.7.3.	Pruebas .....	105
3.8.	ITERACIÓN 6 – Administrar Compras.....	111
3.8.1.	Diseño .....	111
3.8.2.	Desarrollo.....	113
3.8.3.	Pruebas .....	115
CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES .....		122
4.1.	CONCLUSIONES.....	122
4.2.	RECOMENDACIONES .....	123
BIBLIOGRAFÍA .....		125

## ÍNDICE DE FIGURAS

Figura 1-1. Organigrama de la empresa (Zurita U, 2017).....	2
Figura 1-2. Aplicación de escritorio (Zurita U, 2017). ....	3
Figura 2-1. Diagrama general (Zurita U, 2017). ....	33
Figura 2-2. Ingresar al Sistema (Zurita U, 2017). ....	34
Figura 2-3. Administrar Clientes (Zurita U, 2017). ....	35
Figura 2-4. Ingresar Cliente (Zurita U, 2017). ....	35
Figura 2-5. Consulta General (Zurita U, 2017). ....	36
Figura 2-6. Consulta por Parámetro (Zurita U, 2017). ....	37
Figura 2-7. Modificar Cliente (Zurita U, 2017). ....	38
Figura 2-8. Eliminar Cliente (Zurita U, 2017). ....	39
Figura 2-9. Administrar Proveedores (Zurita U, 2017). ....	40
Figura 2-10. Ingresar Proveedores (Zurita U, 2017). ....	40
Figura 2-11. Consulta General (Zurita U, 2017). ....	41
Figura 2-12. Consultar por Parámetros (Zurita U, 2017). ....	42
Figura 2-13. Modificar Proveedor (Zurita U, 2017). ....	43
Figura 2-14. Eliminar Proveedor (Zurita U, 2017). ....	44
Figura 2-15. Administrar Productos (Zurita U, 2017). ....	45
Figura 2-16. Ingresar Producto (Zurita U, 2017). ....	45
Figura 2-17. Modificar Producto (Zurita U, 2017). ....	46
Figura 2-18. Consulta General (Zurita U, 2017). ....	47
Figura 2-19. Consulta por Parámetro (Zurita U, 2017). ....	48
Figura 2-20. Eliminar Producto (Zurita U, 2017). ....	49
Figura 2-21. Administrar Ventas (Zurita U, 2017). ....	50
Figura 2-22. Generar Venta (Zurita U, 2017). ....	50
Figura 2-23. Consulta General (Zurita U, 2017). ....	51
Figura 2-24. Consulta por Parámetro (Zurita U, 2017). ....	52
Figura 2-25. Anular Venta (Zurita U, 2017). ....	53
Figura 2-26. Administrar Compras (Zurita U, 2017). ....	54
Figura 2-27. Generar Compra (Zurita U, 2017). ....	54
Figura 2-28. Consulta General (Zurita U, 2017). ....	55

---

---

Figura 2-29. Consulta por Parámetro (Zurita U, 2017).....	56
Figura 2-30. Anular Compra (Zurita U, 2017).....	57
Figura 3-1. Modelo Físico (Zurita U, 2017). ....	62
Figura 3-2. Modelo Relacional (Zurita U, 2017). ....	63
Figura 3-3. Diagrama de clases GUI Iteración 1 (Zurita U, 2017). ....	63
Figura 3-4. Diagrama de clases DP Iteración 1 (Zurita U, 2017). ....	64
Figura 3-5. Diagrama de clases MD Iteración 1 (Zurita U, 2017). ....	64
Figura 3-6. GUI Prototipo Inicio (Zurita U, 2017). ....	64
Figura 3-7. GUI Prototipo Principal (Zurita U, 2017). ....	65
Figura 3-8. GUI Inicio (Zurita U, 2017). ....	66
Figura 3-9. GUI Principal (Zurita U, 2017). ....	66
Figura 3-10. Prueba Unitaria Ventana Inicio (Zurita U, 2017).....	67
Figura 3-11. Prueba Unitaria UsuarioDP (Zurita U, 2017).....	67
Figura 3-12. Prueba Unitaria UsuarioMD (Zurita U, 2017). ....	68
Figura 3-13. Prueba Unitaria Ventana Principal (Zurita U, 2017).....	68
Figura 3-14. Prueba Caja Negra Ingreso (Zurita U, 2017). ....	69
Figura 3-15. Diagrama de clases GUI Iteración 2 (Zurita U, 2017). ....	70
Figura 3-16. Diagrama de clases DP Iteración 2 (Zurita U, 2017). ....	70
Figura 3-17. Diagrama de clases MD Iteración 2 (Zurita U, 2017).....	71
Figura 3-18. GUI Prototipo Clientes (Zurita U, 2017).....	71
Figura 3-19. GUI Prototipo Consulta (Zurita U, 2017). ....	72
Figura 3-20. GUI Cliente (Zurita U, 2017). ....	73
Figura 3-21. GUI Consulta (Zurita U, 2017). ....	73
Figura 3-22. Prueba Unitaria Ventana Cliente (Zurita U, 2017).....	74
Figura 3-23. Prueba Unitaria ClienteDP (Zurita U, 2017).....	74
Figura 3-24. Prueba Unitaria ClienteMD (Zurita U, 2017). ....	75
Figura 3-25. Prueba Caja Negra Ingreso Cliente Datos (Zurita U, 2017).....	76
Figura 3-26. Prueba Caja Negra Ingreso Cliente Confirmar (Zurita U, 2017). ....	76
Figura 3-27. Prueba Caja Negra Ingreso Cliente Respuesta (Zurita U, 2017).....	76
Figura 3-28. Prueba Caja Negra Ingreso Cliente BDD (Zurita U, 2017). ....	77
Figura 3-29. Prueba Caja Negra Modificar Cliente Datos (Zurita U, 2017).....	77

---

Figura 3-30. Prueba Caja Negra Modificar Cliente Confirmar (Zurita U, 2017). ....	77
Figura 3-31. Prueba Caja Negra Modificar Cliente Datos (Zurita U, 2017).....	78
Figura 3-32. Prueba Caja Negra Modificar Cliente BDD (Zurita U, 2017). ....	78
Figura 3-33. Prueba Caja Negra Consulta General (Zurita U, 2017).....	78
Figura 3-34. Prueba Caja Negra Consulta General Respuesta (Zurita U, 2017). ....	79
Figura 3-35. Prueba Caja Negra Consultar Cliente por parámetro (Zurita U, 2017). 79	
Figura 3-36. Prueba Caja Negra Consultar Cliente por parámetro respuesta (Zurita U, 2017). ....	80
Figura 3-37. Prueba Caja Negra Eliminar Cliente (Zurita U, 2017).....	80
Figura 3-38. Prueba Caja Negra Eliminar respuesta (Zurita U, 2017).....	80
Figura 3-39. Prueba Caja Negra Eliminar BDD (Zurita U, 2017). ....	81
Figura 3-40. Diagrama de clases GUI Iteración 3 (Zurita U, 2017). ....	82
Figura 3-41. Diagrama de clases DP Iteración 3 (Zurita U, 2017). ....	82
Figura 3-42. Diagrama de clases MD Iteración 3 (Zurita U, 2017).....	83
Figura 3-43. GUI Prototipo Proveedores (Zurita U, 2017).....	83
Figura 3-44. GUI Proveedores (Zurita U, 2017).....	84
Figura 3-45. Prueba Unitaria Ventana Proveedor (Zurita U, 2017).....	85
Figura 3-46. Prueba Unitaria ProveedorDP (Zurita U, 2017).....	85
Figura 3-47. Prueba Unitaria ProveedorMD (Zurita U, 2017). ....	86
Figura 3-48. Prueba Caja Negra Ingreso Proveedor Datos (Zurita U, 2017).....	86
Figura 3-49. Prueba Caja Negra Ingreso Proveedor Confirmar (Zurita U, 2017). ....	87
Figura 3-50. Prueba Caja Negra Ingreso Proveedor Respuesta (Zurita U, 2017).....	87
Figura 3-51. Prueba Caja Negra Ingreso Proveedor BDD Datos (Zurita U, 2017). ..	87
Figura 3-52. Prueba Caja Negra Modificar Proveedor Datos (Zurita U, 2017).....	88
Figura 3-53. Prueba Caja Negra Modificar Proveedor Confirmar (Zurita U, 2017). 88	
Figura 3-54. Prueba Caja Negra Modificar Proveedor Respuesta (Zurita U, 2017)..	88
Figura 3-55. Prueba Caja Negra Modificar Proveedor BDD (Zurita U, 2017).....	89
Figura 3-56. Prueba Caja Negra Consulta General Proveedor (Zurita U, 2017). ....	89
Figura 3-57. Prueba Caja Negra Consulta Parámetro Proveedor (Zurita U, 2017). ..	89
Figura 3-58. Prueba Caja Negra Consulta Parámetro Proveedor Resultado (Zurita U, 2017). ....	90

---

Figura 3-59. Prueba Caja Negra Eliminar Proveedor (Zurita U, 2017). .....	90
Figura 3-60. Prueba Caja Negra Eliminar Proveedor Confirmar (Zurita U, 2017). ..	90
Figura 3-61. Prueba Caja Negra Eliminar Proveedor Resultado (Zurita U, 2017). ...	91
Figura 3-62. Prueba Caja Negra Eliminar Proveedor BDD (Zurita U, 2017).....	91
Figura 3-63. Diagrama de Clases GUI Iteración 4 (Zurita U, 2017). .....	92
Figura 3-64. Diagrama de Clases DP Iteración 4 (Zurita U, 2017). .....	92
Figura 3-65. Diagrama de Clases MD Iteración 4 (Zurita U, 2017).....	93
Figura 3-66. GUI Prototipo Productos (Zurita U, 2017).....	93
Figura 3-67. GUI Productos (Zurita U, 2017).....	94
Figura 3-68. Prueba Unitaria Ventana Producto (Zurita U, 2017).....	95
Figura 3-69. Prueba Unitaria Producto DP (Zurita U, 2017).....	95
Figura 3-70. Prueba Unitaria Producto MD (Zurita U, 2017).....	96
Figura 3-71. Prueba Caja Negra Ingresar Producto Datos (Zurita U, 2017).....	96
Figura 3-72. Prueba Caja Negra Ingresar Producto Confirmar (Zurita U, 2017). ....	97
Figura 3-73. Prueba Caja Negra Ingresar Producto Confirmar (Zurita U, 2017). ....	97
Figura 3-74. Prueba Caja Negra Ingresar Producto BDD (Zurita U, 2017). .....	97
Figura 3-75. Prueba Caja Negra Modificar Producto Datos (Zurita U, 2017).....	98
Figura 3-76. Prueba Caja Negra Modificar Producto Confirmar (Zurita U, 2017). ..	98
Figura 3-77. Prueba Caja Negra Modificar Producto Respuesta (Zurita U, 2017)....	98
Figura 3-78. Prueba Caja Negra Consulta General Producto (Zurita U, 2017). .....	99
Figura 3-79. Prueba Caja Negra Consulta Parámetro Producto (Zurita U, 2017). ....	99
Figura 3-80. Prueba Caja Negra Consulta Parámetro Producto Respuesta (Zurita U, 2017). .....	100
Figura 3-81. Prueba Caja Negra Eliminar Producto Confirmar (Zurita U, 2017). ..	100
Figura 3-82. Prueba Caja Negra Eliminar Producto Resultado (Zurita U, 2017). ...	100
Figura 3-83. Prueba Caja Negra Eliminar Producto BDD (Zurita U, 2017).....	101
Figura 3-84. Diagrama de Clases GUI Iteración 5 (Zurita U, 2017). .....	102
Figura 3-85. Diagrama de Clases DP Iteración 5 (Zurita U, 2017). .....	102
Figura 3-86. Diagrama de Clases MD Iteración 5 (Zurita U, 2017).....	103
Figura 3-87. Prototipo GUI Ventas (Zurita U, 2017).....	103
Figura 3-88. GUI Ventas (Zurita U, 2017). .....	105

---

Figura 3-89. Prueba Unitaria Ventana Venta (Zurita U, 2017).....	105
Figura 3-90. Prueba Unitaria Venta DP (Zurita U, 2017).....	106
Figura 3-91. Prueba Unitaria Venta MD (Zurita U, 2017). ....	106
Figura 3-92. Prueba Caja Negra Generar Venta Datos (Zurita U, 2017).....	107
Figura 3-93. Prueba Caja Negra Generar Venta Confirmar (Zurita U, 2017). ....	107
Figura 3-94. Prueba Caja Negra Generar Venta Resultado (Zurita U, 2017). ....	108
Figura 3-95. Prueba Caja Negra Generar Venta Reporte (Zurita U, 2017). ....	108
Figura 3-96. Prueba Caja Negra Anular Venta Confirmar (Zurita U, 2017). ....	109
Figura 3-97. Prueba Caja Negra Anular Venta Resultado (Zurita U, 2017).....	109
Figura 3-98. Prueba Caja Negra Consulta General Ventas (Zurita U, 2017). ....	110
Figura 3-99. Prueba Caja Negra Consulta Parámetro Venta (Zurita U, 2017). ....	110
Figura 3-100. Diagrama de Clases GUI Iteración 6 (Zurita U, 2017). ....	112
Figura 3-101. Diagrama de Clases DP Iteración 6 (Zurita U, 2017). ....	112
Figura 3-102. Diagrama de Clases MD Iteración 6 (Zurita U, 2017).....	113
Figura 3-103. Prototipo GUI Compras (Zurita U, 2017). ....	113
Figura 3-104. GUI Compras (Zurita U, 2017). ....	114
Figura 3-105. Prueba Unitaria Ventana Compras (Zurita U, 2017).....	115
Figura 3-106. Prueba Unitaria Compra DP (Zurita U, 2017). ....	115
Figura 3-107. Prueba Unitaria Compra MD (Zurita U, 2017). ....	116
Figura 3-108. Prueba Caja Negra Generar Compra Datos (Zurita U, 2017). ....	117
Figura 3-109. Prueba Caja Negra Generar Compra Confirmar (Zurita U, 2017)....	117
Figura 3-110. Prueba Caja Negra Generar Compra Resultado (Zurita U, 2017)....	118
Figura 3-111. Prueba Caja Negra Generar Compra BDD (Zurita U, 2017). ....	118
Figura 3-112. Prueba Caja Negra Generar Compra Reporte (Zurita U, 2017). ....	118
Figura 3-113. Prueba Caja Negra Anular Compra Confirmar (Zurita U, 2017).....	119
Figura 3-114. Prueba Caja Negra Anular Compra Resultado (Zurita U, 2017). ....	119
Figura 3-115. Prueba Caja Negra Anular Compra BDD (Zurita U, 2017).....	120
Figura 3-116. Prueba Caja Negra Consulta General Compra (Zurita U, 2017).....	120
Figura 3-117. Prueba Caja Negra Consulta Parámetro Compra (Zurita U, 2017)...	121

## ÍNDICE DE TABLAS

Tabla 2-1. Historia de Usuario 1 (Zurita U, 2017).....	30
Tabla 2-2. Historia de Usuario 2 (Zurita U, 2017).....	30
Tabla 2-3. Historia de Usuario 3 (Zurita U, 2017).....	31
Tabla 2-4. Historia de Usuario 4 (Zurita U, 2017).....	31
Tabla 2-5. Historia de Usuario 5 (Zurita U, 2017).....	32
Tabla 2-6. Historia de Usuario 6 (Zurita U, 2017).....	32
Tabla 3-1. Prueba de Aceptación 1 (Zurita U, 2017).....	69
Tabla 3-2. Prueba de Aceptación 2 (Zurita U, 2017).....	81
Tabla 3-3. Prueba de Aceptación 3 (Zurita U, 2017).....	91
Tabla 3-4. Prueba de Aceptación 4 (Zurita U, 2017).....	101
Tabla 3-5. Prueba de Aceptación 5 (Zurita U, 2017).....	111
Tabla 3-6. Prueba de Aceptación 6 (Zurita U, 2017).....	121

## **RESUMEN**

El presente documento de disertación está basado en el desarrollo de un sistema informático para la administración de Clientes y Proveedores, así como la administración de Compra y Venta de productos para la empresa Aldera Diseño; dedicada al diseño y producción de muebles en madera y derivados. El sistema tiene como objetivo el registro de datos informativos importantes para la empresa con el objetivo de asegurar el almacenamiento y respaldo de información en una sola base de datos.

Para la elaboración del sistema, se utilizó una metodología de desarrollo ágil, Extreme Programming, esta obtiene los requerimientos, que el sistema debe satisfacer, por medio de las Historias de Usuario y su desarrollo se lo realiza por iteraciones con la constante interacción con el cliente; para el desarrollo se utilizó el lenguaje de programación JAVA por medio del IDE Netbeans y el motor de base de datos PostgreSQL.

Con la elaboración del sistema, se puede concluir que el manejo de la información de la empresa es precisa y confiable, además asegura el correcto almacenamiento de la misma.

## **DEDICATORIA**

Dedico esta disertación a mi familia.

A mis padres, Hilda y Raúl quienes, desde el inicio han estado para mí, con su apoyo incondicional, y gracias a ellos, me encuentro cumpliendo mis metas y objetivos; Por ellos siempre he dado lo mejor y mi mayor esfuerzo para no defraudarlos.

A mi hermano, Pablo quién, a pesar de la poca diferencia de edad que tenemos, ha sido un gran ejemplo, una inspiración y un guía para mi vida, desde pequeño siguiendo sus pasos.

A mi mejor amigo, Esteban quién, ha sido un gran apoyo, no solo en mis estudios, y más que un amigo, es un hermano; una de las mejores personas que he conocido en mi vida.

## **AGRADECIMIENTO**

Agradezco a mis padres por todo lo que hicieron para que yo cumpla con mi sueño, con mi objetivo, porque gracias a ellos sacaba las fuerzas necesarias para no rendirme día tras día en una carrera de gran sacrificio, siempre pensando en ellos cuando tenía que desvelarme y amanecerme, siempre pensando en no defraudarlos y demostrarles lo mucho que los amo por medio de mis estudios. Agradezco también a todas las personas que me apoyaron durante todo este tiempo para culminar mis estudios.

## **CAPÍTULO 1. MARCO TEÓRICO**

En este capítulo se da a conocer la empresa para la que será desarrollado el sistema; explicaremos que es una aplicación de escritorio, su funcionamiento, sus características y los recursos que se utilizará para su desarrollo.

También se revisará, de que forma se van a presentar los requerimientos del sistema según las historias de usuarios y por último la metodología que se aplicará para el desarrollo del sistema.

### **1.1. LA EMPRESA**

#### **1.1.1. El giro del negocio**

Aldera Diseño, es una empresa dedicada al diseño y producción de muebles en madera y derivados, ofreciendo sus productos desde el año 2010 al por mayor y menor en el mercado nacional, con proveedores de materia prima y clientes en diferentes puntos del país.

#### **1.1.2. Misión**

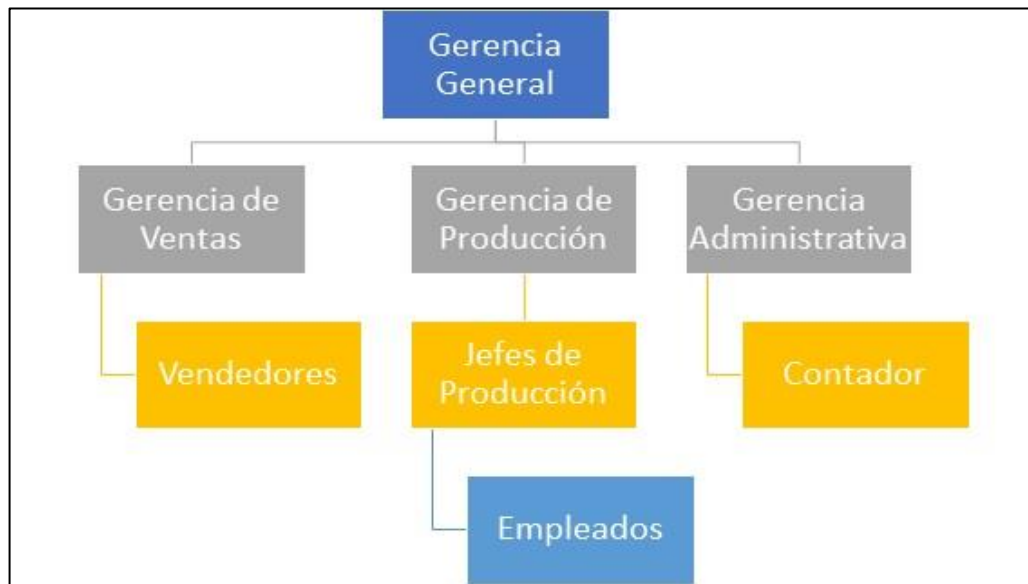
Desarrollar productos innovadores y de alta calidad para cada uno de nuestros clientes en los diferentes materiales que se realiza, brindando servicios de instalación de los mismos.

#### **1.1.3. Visión**

En los próximos 2 años se espera que el 100% de las obras sean entregadas a tiempo, manteniendo la calidad en los productos.

#### **1.1.4. Organigrama de la empresa**

En la Figura 1-1 vemos cual es la estructura de la empresa.



*Figura 1-1. Organigrama de la empresa (Zurita U, 2017).*

### **1.1.5. Proveedores**

Se define al proveedor como la persona empresa que suministra a otras empresas con materias primas, que serán transformadas para ser vendidas y distribuidas como un producto terminado (Lizarazo, 2017).

Las materias primas que son adquiridas están dirigidas directamente a la actividad o negocio principal de la empresa que compra esos elementos.

En el caso de Aldera Diseño necesita proveedores de madera, telas y otros para poder desarrollar su actividad principal que es la creación de muebles de madera.

### **1.1.6. Clientes**

El cliente es la entidad que adquiere de forma voluntaria productos o servicios que necesita o desea para sí mismo o para terceros (Ortiz, 2013).

Aldera Diseño se enfoca en que sus clientes siempre estén satisfechos con la adquisición de sus productos.

## 1.2. SISTEMA

El sistema que se va a desarrollar será el encargado de la administración de clientes, proveedores y de los procesos de compra y venta. Se ha determinado que el sistema será una aplicación de escritorio, que se desarrollará con un lenguaje de programación, que trabajará en conjunto con un motor de base de datos.

### 1.2.1. Aplicación de escritorio

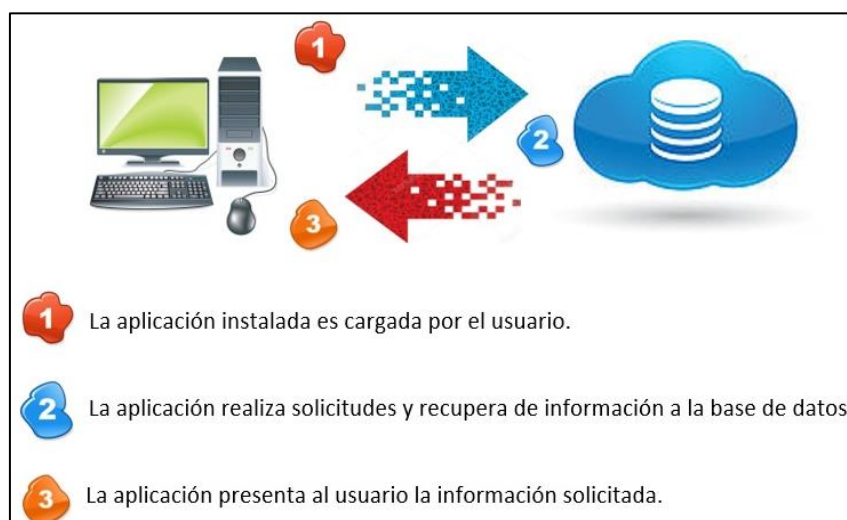
#### Definición.

Una aplicación de escritorio se define como un programa desarrollado con la finalidad de realizar, solucionar, satisfacer y/o cumplir con un objetivo/tarea específica; diseñada para interactuar con cualquier usuario. La ejecución y rendimiento de una aplicación de escritorio depende del sistema operativo y las características de hardware del computador.

Al momento del desarrollo debemos considerar el lenguaje de programación que vamos a utilizar, el entorno en el que se va a ejecutar y cuál es la base de datos que se ajuste mejor según las características que debe cumplir el programa.

#### Funcionamiento.

La Figura 1-2 se presenta como se da el funcionamiento de una aplicación de escritorio.



*Figura 1-2. Aplicación de escritorio (Zurita U, 2017).*

### **Ventajas.**

- Las aplicaciones de escritorio no dependen de un tercero para su funcionamiento (Mora, 2014).
- Son rápidas y ágiles debido a un mayor aprovechamiento del CPU, por lo tanto, mayor velocidad de la aplicación (López, 2013).
- Se puede incorporar todos los controles y eventos del escritorio (Velásquez, 2012).
- Como su ejecución se produce de forma local, el procesamiento es más rápido (Buyto, 2009).
- Se puede elaborar herramientas eficaces, robustas y estables.
- Al desarrollar una aplicación de escritorio se consta con mayor capacidad para programar funcionalidades más complejas (Buyto, 2009).

### **Desventajas.**

- Si se lleva a cabo una actualización, esta debe ser instalada a cada una de las estaciones de trabajo donde se ha instalado la aplicación (Velásquez, 2012).
- El acceso solo puede ser desde las estaciones de trabajo donde se ha instalado (Velásquez, 2012).
- Su correcta ejecución depende del sistema operativo y las características de hardware y software (Mora, 2014).
- La instalación debe ser personalizada al igual que sus actualizaciones (López, 2013).

### **1.2.2. Lenguaje de programación.**

Es el idioma en el que se expresan las funciones que serán ejecutadas por dispositivos tecnológicos. A través de éste se crean programas que determinan el funcionamiento físico y lógico de los dispositivos (IsraelCCM, 2017).

Un lenguaje de programación se compone de símbolos y reglas sintácticas y semánticas, que determinan su estructura. El lenguaje debe ser escrito, depurado y compilado; a esto es a lo que denominamos la programación (Lutz, 2010).

Al programar estamos enseñando a una computadora los datos que debe operar, de qué forma deben ser almacenados o transmitidos y que acciones deben ser ejecutadas según las circunstancias que se pueden presentar.

Según la forma en la que se procesan los comandos los lenguajes de programación son imperativos y funcionales (IsraelCCM, 2017).

La programación de un lenguaje imperativo se la lleva a cabo mediante comandos que se agrupan en bloques y se componen de órdenes condicionales. Estos lenguajes carecen de flexibilidad por la secuencia de instrucciones.

La programación en un lenguaje funcional o procedimental, se la realiza mediante funciones, que devuelven un estado de resultado según los datos de entrada que tienen. Si en este proceso una función se llama a sí misma se denomina como recursividad.

Según (Ruby, 2015), al momento de programar, en cualquier lenguaje que decidamos hacerlo, debemos considerar algunas características. Un programa tiene que ser correcto, claro, eficiente y portable.

- **Correcto.** Se considera que un programa es correcto si cumple todas las funciones como se establece en las fases previas a la programación.
- **Claro.** La claridad en el programa permite que tanto su desarrollo como mantenimiento sean fáciles. La estructura debe ser lo más simple y lógica posible, de esta forma se simplificará las modificaciones y correcciones.
- **Eficiente.** La eficiencia de un programa se determina por el correcto uso de los recursos. En un programa la eficiencia está dada por el tiempo que tarda en realizar las operaciones y la cantidad de memoria que utiliza.
- **Portable.** Se considera que un programa es portable cuando puede funcionar en varias plataformas.

### 1.2.3. Motores de bases de datos.

Una base de datos es un conjunto de información que pertenece a un contexto en común y es accesible, gestionable y actualizada (Maldonado, 2017).

Las bases de datos relacionales son las más aplicadas a los sistemas informáticos, son bases de datos que se componen de una o más tablas que a su vez con conformadas por una o más filas y columnas, sus datos pueden ser reorganizados y presentados de diferentes maneras (Alonso, 2017). Entre las principales características de las bases de datos podemos considerar:

- **Redundancia mínima.** La redundancia se presenta cuando la misma información es almacenada más de una vez. Por medio de los motores de bases de datos podemos reducir esta redundancia al mínimo. (Aparicio, 2009)
- **Acceso concurrente de múltiples usuarios.** Varios dispositivos pueden acceder a la información almacenada en la base, así también pueden almacenar más datos o modificarlos. Estas funciones pueden ser limitadas con permisos de acceso según el usuario que entra a la base de datos. (Aparicio, 2009)
- **Integridad de datos.** Esta característica está dada por la veracidad de la información que se encuentra almacenada en la base de datos (Aparicio, 2009).
- **Consultas optimizadas.** La optimización en las consultas permite rapidez en su ejecución. Esto se logra cuando los códigos de búsqueda no son tan largos y son claros y precisos. (Aparicio, 2009)
- **Seguridad de acceso y auditorías.** Esto se refiere al acceso a la base de datos por parte de sus usuarios. La auditoría es un control que se mantiene para saber quién ha ingresado y quién ha realizado alguna modificación y cuando lo ha hecho. (Aparicio, 2009)
- **Respaldo de recuperación.** Es la capacidad de las bases de datos de recuperar su estado en un momento antes de que se produjera una pérdida de datos, para lo cual se debe contar con un respaldo periódico. (Aparicio, 2009)

Por medio de las bases de datos podemos almacenar grandes cantidades de información, recuperamos la información almacenada de forma rápida y flexible, podemos organizar y reorganizar la información según nuestras necesidades y podemos presentar y distribuir la información de diferentes maneras.

Los motores de bases de datos dan a los usuarios la capacidad de controlar el acceso de lectura y escritura, generar informes y realizar análisis.

El lenguaje SQL (Structured Query Language) es el lenguaje estándar para realizar consultas y actualizar las bases de datos.

### **1.3. INGENIERÍA DE REQUERIMIENTOS**

Podemos definir la ingeniería de requerimientos como la ciencia y la disciplina de identificar, documentar, verificar y administrar los requerimientos de un sistema (Rodríguez, 2011). Mediante la ingeniería de requerimientos se puede establecer el objetivo del desarrollo de un sistema, sus necesidades y funcionalidades, las cuales tienen el propósito de definir el comportamiento del sistema.

En la ingeniería de requerimientos se distingue los “requerimientos del sistema” y “requerimientos del software”. Los primeros se refieren a transformar las necesidades de los clientes y usuarios en requerimientos del sistema, mientras que los segundos se refieren a descomponer el problema en varios problemas pequeños o “subsistemas”, en distribuir los requerimientos en dichos subsistemas y en transformar los requerimientos en una especificación más detallada apropiada para los diseñadores y arquitectos de software (Rodríguez, 2011).

#### **1.3.1. Historias de Usuarios**

Es la forma en la que los clientes transmiten los requisitos, que debe cumplir el sistema, a los desarrolladores. Estas historias son presentadas en lenguaje común y en el caso de extreme programming deben ser escritas por el cliente (Tersek, 2015).

Por medio de las historias de usuarios se llega a la determinación de los requerimientos sin la necesidad de realizar varios documentos formales que requieren de mucho tiempo.

Para cada historia de usuario, el programador determinará el tiempo que tomará el desarrollo. En caso de que la historia de usuario sea muy compleja, esta debe ser dividida.

De acuerdo con (Tersek, 2015) al momento de elaborar las historias de usuarios debemos tomar en cuenta tres elementos: La tarjeta, la conversación y la confirmación.

**La tarjeta.** Donde el cliente expresará en sus palabras los requerimientos.

**La conversación.** Se da entre el cliente y los desarrolladores. Aquí se determina el valor de cada historia y el resultado que se espera obtener en cada una.

**La confirmación.** Es el punto donde el cliente y los desarrolladores se ponen de acuerdo en cómo se resolverá las historias de uso.

Según (Menzinsky, 2016), los elementos son:

**ID.** Es el identificador único de cada historia de usuario.

**Título.** Describe la historia de usuario.

**Descripción.** Describe la funcionalidad que se debe cumplir. Se puede seguir este formato: Como [ROL], quiero [OBJETIVO], para [BENEFICIO].

**Estimación.** Se estima de qué forma se resolverá la implementación de la historia de usuario.

**Valor.** Es un valor numérico asignado a cada historia de usuario por el cliente para determinar la importancia de la funcionalidad.

**Dependencia.** Es preferible que no exista dependencia entre las historias y en caso de que exista se la indicará aquí.

**Prueba de aceptación.** Es realizada para determinar si el desarrollo ha cumplido con lo requerido por el cliente.

## **Método INVEST**

(Tersek, 2015), define al método con las siguientes características:

### **Independiente**

Cada historia de usuario es desarrollada en cualquier orden sin que una dependa de otra. Si hay dependencia deben ser divididas o combinadas.

### **Negociable**

Las historias son acordadas entre los clientes y usuarios al momento de la conversación.

### **Valor**

Las historias de usuarios son valoradas por los clientes y los usuarios.

### **Estimable**

Se estima el tiempo del desarrollo y la implementación.

### **Pequeño**

De preferencia las historias deben ocupar el espacio de un sprint.

### **Testable**

Una vez que se ha desarrollado, las historias de usuarios deben ser probadas y el cliente con el desarrollador determinaran si están terminadas.

## **1.4. METODOLOGÍA**

La metodología se define como un conjunto de métodos, principios y reglas que permiten el desarrollo un sistema. Las metodologías se estructuran como una secuencia de pasos que inician con la definición del problema y terminan en el programa en sí.

### **1.4.1. Extreme Programming (XP)**

La programación extrema está basada en una serie de reglas y principios que se han desarrollado a lo largo de la historia de la ingeniería de software. Al aplicarlas de forma conjunta tenemos como resultado una metodología considerada ágil dentro de las metodologías de desarrollo de software (Robles, 2002, pág. 2).

Una metodología es considerada ágil cuando da prioridad al resultado final y reduce al máximo toda la burocracia, que existe alrededor del desarrollo, tanto como sea posible.

Los componentes de la programación extrema son ya conocidos dentro la ingeniería de software, muchos de estos existen desde sus comienzos. Los autores de esta metodología han seleccionado los mejores y han profundizado en la forma en que se complementan para dar como resultado esta metodología única y compacta.

La programación extrema es aplicada idealmente al momento de desarrollar software a medida. La relación con el cliente es diferente a la que se mantiene en las metodologías tradicionales que se basan en la definición de requerimientos previa al desarrollo y la validación posterior a este (Robles, 2002, pág. 3).

## **PROCESO**

### **Interacción con el cliente.**

De acuerdo con (Robles, 2002, págs. 3-4):

La metodología de programación extrema parte de una lista de características que el cliente requiere que existan en el sistema y es determinada a partir de las historias de usuarios.

El cliente está más inmerso en el proceso de desarrollo y será un apoyo para el equipo ya que tendrá como tarea expresar al equipo de desarrollo los problemas que ha encontrado y cuáles son sus sensaciones para que el equipo de desarrollo trabaje sobre ellas.

Al estar el cliente más involucrado estamos eliminando la fase de captura de requisitos ya que estos se irán definiendo claramente durante el tiempo que dure el proyecto.

En caso de que se produzcan cambios en cuanto a los requisitos el cliente siempre deberá estar predispuesto a resolver las interrogantes que se puedan presentar en el equipo de desarrollo con el fin de que se establezca de forma clara los requerimientos que debe cumplir el sistema.

Según (Robles, 2002), la definición de las historias de usuarios se da en dos fases:

**Primera Fase.** El cliente expresa en sus palabras las características que debe cumplir el sistema, a continuación, la persona que encabece el equipo de desarrollo explicará las dificultades técnicas para llevar a cabo cada una de estas. Al final, el cliente documenta el conjunto de historias y las ordena según la prioridad que tengan para él. Con esto ya se puede definir tiempos de desarrollo para cada una.

**Segunda Fase.** Se toman las primeras historias que serán implementadas y se las dividirá en tareas que serán ejecutadas. El cliente tiene en parte participación, pero la mayor responsabilidad está en el equipo de desarrollo, dando como resultado una planificación más exacta. Esta es considerada la primera iteración y en cada una se llevará a cabo esta fase considerando las historias planificadas en cada una.

Las historias de usuario se relacionan con otras técnicas como los casos de uso de UML pero su proceso de creación es muy diferente. Será el cliente quien escriba lo que quiere que el sistema realice.

Una vez establecidas las historias de usuarios, el equipo de desarrollo es el que determina la duración. Como norma se ha establecido que las historias de usuarios deben ser realizadas entre una y tres semanas de programación. Si se determina que las historias de los usuarios van a llevarse a cabo en menos tiempo, estas serán agrupadas. Así mismo si se determina que una historia de usuario va a tomar más tiempo, ésta debe ser cambiada o dividida (Robles, 2002, pág. 4).

### **Planificación del proyecto**

En esta parte del proceso planificaremos las diferentes iteraciones que se llevarán a cabo. Para una correcta planificación se debe considerar algunas premisas: (Robles, 2002, pág. 4)

- Las iteraciones deben ser realizadas lo más pronto posible y por cada una el cliente tiene que recibir una nueva versión.
- Si nos tardamos en introducir una parte esencial, menos tiempo tendremos para trabajar en ella después.
- Es preferible hacer varias entregas con mucha frecuencia.

Según (Robles, 2002) de esta manera, se determinará rápidamente si existe un error y así mismo se lo podrá corregir; prácticas como las pruebas unitarias, la integración continua y el juego de la planificación evitan que se tenga que trabajar horas extra.

Esta metodología prevé que puedan existir errores y por esto tiene previsto mecanismos de revisión. Debido a esto, es normal que cada 3 o 5 iteraciones, se deba revisar las historias de los usuarios y renegociar la planificación.

Se lleva a cabo una planificación por iteraciones y debe ser reconsiderada al cabo de unas cuantas iteraciones. Se debe tomar en cuenta que en cada iteración se realiza una planificación de la misma, a esto se lo denomina planificación iterativa. En la planificación iterativa se determina que historias de usuarios son prioritarias para su implementación y se aumenta las que no pasaron la prueba de aceptación en anteriores iteraciones.

El diseño que llevamos a cabo puede ser calificado como continuo debido a sus características. El proceso de desarrollo es ágil y evita adelantarnos en el proceso y programar tareas que aún no han sido consideradas.

A lo largo del proceso debemos concentrarnos en programar lo que corresponde a cada iteración, no es necesario que añadamos funcionalidades que no son necesarias en ese momento. Al final se determinará si es que debemos añadir funcionalidades o si podemos optimizar algún proceso.

Al llevar a cabo la planificación en iteraciones y el diseño iterativo se da paso a las discusiones diarias. De esta manera los desarrolladores pueden hablar de los problemas que se van presentando y de qué manera van cumpliendo con sus tareas.

### **Diseño, Desarrollo y Pruebas**

El desarrollo es la parte principal del proceso de programación extrema. Todas las tareas se llevan a cabo con el objetivo que el desarrollo sea rápido, sin interrupciones y de forma correcta.

El diseño es muy importante y debe ser revisado y mejorado continuamente al tiempo que se van aumentando las funcionalidades.

De esta forma no estamos limitados en caso de que el cliente quiera realizar cambios con respecto a las funcionalidades. Al contrario, se crea el ambiente propicio para que se realice una remodelación del sistema a lo largo del desarrollo (Robles, 2002, pág. 5).

Todo el proceso de desarrollo de la programación extrema gira entorno a la comunicación, de esta forma la información fluirá de forma eficiente y el resultado será un sistema acorde a las necesidades del usuario.

Durante el desarrollo se tomará en cuenta los nombres que se van a escoger para los módulos, sistemas, clases, métodos, etc. Ya que la elección correcta de los nombres se traducirá en claridad, reusabilidad y simplicidad. Tres conceptos que son importantes para la programación extrema.

El diseño está a cargo de los desarrolladores, pero se puede involucrar a profesionales con más experiencia e incluso al cliente al momento del desarrollo de las partes que se consideran más complejas. Al momento de realizar estas reuniones se utilizan un tipo de tarjetas que se denominan CRC (Clase, Responsabilidades y Colaboración) cuyo objetivo es facilitar la comunicación y documentar los resultados. Cada clase tendrá su respectivo CRC para determinar su objetivo e interacciones con otras clases. Si se realizan de forma correcta las CRC el diseño del sistema podrá ser determinado fácilmente.

La simplicidad es una de las características principales de la programación extrema. El diseño debe ser lo más simple posible. Para lograr esto acudimos al paradigma KISS (“Keep it small and simple” o “Keep it simple stupid”). Algo que debemos considerar para cumplir esto, es no añadir funcionalidades que no sean una prioridad y que posiblemente nunca sean necesarias.

Ahora se pasa a uno de los pasos más relevantes de la programación extrema, se implementan las pruebas unitarias al mismo tiempo que se hace el código de producción. Cada vez que se va a implementar una parte escribimos una prueba sencilla y el código suficiente para que pase esta prueba. Si ya ha pasado la prueba se repite el proceso con la siguiente parte.

Al aplicar de esta forma las pruebas unitarias se prioriza y se comprueba la evolución del desarrollo al mismo tiempo que se produce una realimentación inmediata. Al implementar la prueba unitaria, codificar la solución y pasar, tenemos como resultado un código simple, funcional y lo hacemos de forma rápida. Debemos considerar que las pruebas deben ser superadas al 100%.

Las pruebas unitarias no tienen que ser confundidas con las pruebas de aceptación. Las pruebas de aceptación son las realizadas por el usuario final, para que sea él quien compruebe que el sistema hace lo que debe. Si se encuentran errores, el cliente especificará la prioridad en que deben ser corregidos. Este tipo de pruebas son pruebas de caja negra.

Ya que se ha pasado por las pruebas unitarias, codificación y superación de pruebas viene la integración. En la programación extrema se realizará una integración continua. La ventaja que nos da esta integración es la ejecución de pruebas completas y una rápida detección de los problemas de compatibilidad.

Lo ideal será que los desarrolladores realicen una sincronización con el repositorio central por lo menos una vez al día, así los cambios serán realizados sobre la última versión.

En la programación extrema tiene gran importancia la refactorizar ya que de esta manera quitaremos la redundancia y eliminaremos las funcionalidades que no se utilizan. Al realizar esto, no solo tendremos un código funcional sino también un código simple (Robles, 2002, págs. 5-8).

### **Valores de la programación extrema.**

Todo el proceso de desarrollo de la programación extrema tiene su fundamento en algunos valores, considerados por algunos autores de la programación extrema como fundamentales para que el proceso de desarrollo sea exitoso. Según (Robles, 2002), estos valores son:

- Comunicación.
- Simplicidad.

- Realimentación
- Coraje.

Por medio de estos valores obtendremos diseños y códigos simples, métodos eficientes de desarrollo y lo principal clientes satisfechos.

Al hablar de coraje nos referimos al lema “si funciona, mejóralo”

### **Principios de la programación extrema**

La programación extrema se apoya en cuatro principios de acuerdo con (Robles, 2002):

- Realimentación veloz.
- Modificaciones incrementales.
- Trabajo de calidad.
- Asunción de simplicidad.

Estos principios son el puente entre los valores y la práctica, que está ligada a la técnica que se ha de seguir.

### **Prácticas de la programación extrema**

- **El juego de la planificación.**

Se define así a la interacción entre el usuario y el cliente. El equipo de desarrollo estima cuánto esfuerzo se va a requerir por cada historia de usuario y el cliente toma una decisión en cuánto al tiempo de las entregas en cada iteración (Letelier & Penadés, 2016, pág. 12).

- **Pequeñas entregas.**

El objetivo de esta metodología es entregar versiones operativas del sistema de forma rápida hasta llegar al sistema completo (Letelier & Penadés, 2016, págs. 12-13).

- **Metáfora.**

El sistema se va definiendo por medio de una metáfora o un conjunto de estas, que se comparte entre el cliente y el equipo de desarrollo. La metáfora describe el funcionamiento del sistema. Martin Fowler explica que la práctica de la metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. Este conjunto de nombres ayuda a la nomenclatura de clases y métodos del sistema (Letelier & Penadés, 2016, pág. 13).

- **Diseño simple.**

El diseño debe ser simple para que funcione y pueda ser implementado en un determinado momento. De acuerdo con Kent Beck el diseño adecuado supera todas las pruebas a las que es sometido, refleja la intención de implementación del equipo de desarrollo y tiene el mínimo número de clases y métodos (Letelier & Penadés, 2016, pág. 13).

- **Pruebas.**

El desarrollo se dirige a pruebas unitarias, estas se establecen antes de producir el código y se ejecutan constantemente con cada modificación del sistema. Los clientes escriben las pruebas para la validación de cada historia de uso (Letelier & Penadés, 2016, pág. 13).

- **Refactorización.**

Es una actividad constante en la cual el código es reestructurado, mejorado y simplificado. El diseño del sistema de software es una cosa viviente. No se puede imponer todo en un inicio, pero en el transcurso del tiempo este diseño evoluciona conforme cambia la funcionalidad del sistema (Martin, 2002).

- **Propiedad colectiva.**

Cualquier programador puede cambiar cualquier parte del código en cualquier momento (Letelier & Penadés, 2016); por lo tanto, un programador no es imprescindible.

- **Programación por parejas.**

El desarrollo o producción se lo debe realizar en parejas de programadores. Según un estudio realizado por (Cockbun & Williams, 2000) para identificar los costos y beneficios de la programación en parejas, las principales ventajas de introducir este estilo de programación son: inspección de código continua, baja tasa de errores, optimización de diseño y tamaño del código, problemas de programación se resuelven más rápido, se posibilita la transferencia de conocimientos entre el equipo, varias personas entienden diferentes partes del sistema, mejora el flujo de información y la dinámica del equipo, y finalmente, los programadores disfrutan del trabajo.

- **Integración continua.**

Según (Letelier & Penadés, 2016), cada parte del programa es integrada una vez terminada. Para que sea incorporada definitivamente al programa, todas las pruebas deben ser ejecutadas y aprobadas.

- **40 horas semanales.**

Un trabajo debe durar 40 horas por semana y no se podrá realizar horas extras en dos semanas consecutivas. Si ocurre esto, probablemente existe un problema que generará retraso al final del proyecto (Letelier & Penadés, 2016).

- **Cliente en casa.**

El cliente tiene un rol muy importante en el desarrollo del sistema, debe estar todo el tiempo disponible para el equipo. De acuerdo con Jeffries, se debe pagar un precio por perder la oportunidad de un cliente con alta disponibilidad para lo cual propone algunas recomendaciones como: conseguir un representante que siempre esté disponible, establecer visitas frecuentes, actúe como interlocutor, etc. (Jeffries, 2001).

- **Estándares de codificación.**

De acuerdo con (Letelier & Penadés, 2016), la programación extrema enfatiza la comunicación de los programadores a través del código, para lo cual es necesario seguir estándares de programación. Los estándares mantienen el código legible para los miembros del equipo, facilitando cambios.

Para aplicar la programación extrema lo mejor es hacerlo gradualmente. Los autores recomiendan primero identificar el principal problema en el proceso de desarrollo, escoger la práctica que ayude en la resolución y aplicarla, si ya dejó de ser un problema se escoge la siguiente. Lo más recomendable es que se apliquen las prácticas de dos en dos. El objetivo es que las prácticas se apoyen unas a otras ya que así es más fácil comprobar los resultados.

Al final, lo ideal es que sean aplicadas todas las prácticas como el conjunto que conforman.

**De acuerdo con (Padilla, 2013) las ventajas son:**

- Programación organizada.
- Eficiencia en el proceso de planificación y pruebas.
- Se minimiza los errores.
- Da satisfacción al programador.
- Fomenta la comunicación entre el cliente y el desarrollador.
- Es más fácil realizar cambios.
- Por la forma de su ejecución, existe ahorro en tiempo y en dinero.
- Se puede aplicar con cualquier lenguaje de programación.
- Las prioridades son definidas por el cliente.
- A lo largo del proyecto se realizan pruebas.
- Es la que más se utiliza en la implementación de nuevas tecnologías.

**De acuerdo con (Padilla, 2013) las desventajas son:**

- Recomiendan en proyectos que se los va a realizar a corto plazo.
- Si existe una falla en el producto final el costo sería muy alto.
- Se solicita rigidez en cuanto a la aplicación de sus prácticas.
- No siempre resultará más fácil que el desarrollo tradicional.
- No se puede prever todo antes de programar.

### **1.5. PROGRAMACIÓN EN CAPAS**

La programación en capas es una técnica de ingeniería de software propia de la programación orientada a objetos (Vargas del Valle & Maltes Granados, 2007). Con esta técnica el programador realiza el desarrollo de forma más organizada. Es ideal para desarrollar sistemas empresariales que se componen de módulos y subsistemas.

Esta forma de programación se la realiza generalmente en 3 capas:

**Capa de presentación o frontera.** Esta será la capa con la que interactúe directamente el usuario, por eso debe ser amigable y fácil de utilizar. Esta capa será la encargada de tomar toda la información que será procesada por el sistema.

**Capa lógica de negocio o control.** En esta capa se encuentra la lógica del funcionamiento del programa, aquí se especifica la forma en la que serán procesados los datos que el usuario ingresado mediante la capa anterior. Esta capa comunica la interfaz con la base de datos.

**Capa de datos.** En esta capa se encuentra el manejo de la base de datos. Aquí se definirá la forma en que los datos se ingresan a la base de datos, así como también la forma en que se tomarán los datos para presentarlos en la primera capa pasando por la capa lógica de negocio.

### **1.6. BASE DE DATOS RELACIONAL**

El modelo relacional de la base de datos permite establecer relaciones entre los datos que están guardados en dos tablas. El modelo relacional se acogió como un paradigma

para el desarrollo de bases de datos desde que sus bases fueron postuladas en 1970 por Edgar Frank Codd.

Entre las características, de una base de datos relacional, que podemos mencionar **Fuente especificada no válida.** están:

- La base de datos tiene varias tablas o relaciones
- No se puede repetir tablas con el mismo nombre o registro
- Cada tabla en un conjunto de registros que se compone de filas y columnas
- La relación entre tablas se da a través de las claves primarias y foráneas
- La clave primaria es la clave principal de un registro dentro de una tabla
- Las claves foráneas tienen el mismo valor que la clave primaria y por medio de estas se establecen las relaciones.

SQL (Structured Query Language) es el lenguaje más común para realizar las consultas en una base de datos relacional, es un estándar en los principales motores de bases de datos.

Entre los principales SGBD (Sistemas de Gestión de bases de Datos) tenemos: MySQL, PostgreSQL, Oracle, DB2, INFORMIX, Interbase, FireBird, Sybase y Microsoft SQL Server.

### **Elementos de la base de datos**

- Relaciones base y derivadas. Las relaciones base son las relaciones que almacenan datos y su implementación es la tabla. Las relaciones derivadas no almacenan datos, su implementación es la vista o consulta, estas relaciones reúnen la información de varias relaciones y la presenta como una.
- Restricciones. Son condiciones que se deben cumplir al momento de realizar la base de datos y de guardar los datos en esta. Estas restricciones pueden ser definidas por el usuario y hay otras que están dadas por la base de datos.
- Dominios. El dominio determina el tipo de dato que puede ser guardado en la base de datos.
- Clave única. Es el valor que identifica de forma única cada registro de la tabla, puede existir más de una clave única y una de estas será la clave primaria.

- **Clave primaria.** Es la clave única que define a todos los demás atributos de la tabla y especifica que datos serán relacionados con otras tablas. Solo hay una clave primaria por tabla y ese campo no puede tener valores NULL.
- **Clave foránea.** Es la referencia que se hace a una clave en otra tabla para determinar la relación que existe entre dos tablas.
- **Clave índice.** Se las utiliza para tener un acceso más rápido a los registros de las tablas.
- **Procedimientos almacenados.** Es código ejecutable asociado a la base de datos, estos recogen y personalizan operaciones comunes como insertar un registro, recopilar información o encapsular datos.

## 1.7. DIAGRAMAS DE CLASES

El diagrama de clases es la representación gráfica de la estructura de un sistema que va a ser implementado. Los diagramas definen el diseño del sistema una vez que se ha establecido los requisitos que va a cumplir el sistema. Estos diagramas representan las clases que va a tener el sistema, su contenido y la relación con otras clases (Gómez, 2017).

### Componentes

- **Clase.**

Representan las entidades o conceptos. Generalmente los sustantivos que aparecen en la descripción de un sistema son las clases que tendrá. Cada clase tiene sus atributos y sus métodos.

- **Atributos y métodos.**

Los atributos son mostrados con su nombre y tipo.

Los métodos son mostrados igualmente con su nombre y tipo, también se mostrará el tipo de retorno de ser el caso y sus parámetros con su nombre y tipo.

Los símbolos que están antes de los nombres de los atributos y los métodos representan la visibilidad que tienen.

(-): Atributos privados.

(+): Atributos públicos.

(\*): Atributos protegidos.

- **Relaciones.**

Existe relación entre unas clases con otras y cada una tiene un nombre para representarla. Existen relación de:

Generalización. Esta relación se da por la herencia o la extensión de una clase a partir de otra.

Asociación. Es una relación básica entre dos clases, puede ser unidireccional o bidireccional. Las relaciones pueden estar definidas uno a uno, uno a varios o varios a varios.

Agregación. Es un tipo de asociación que muestra que una clase contiene a otra.

Composición. Es también un tipo de asociación y se diferencia de la agregación porque no habría sentido en que el objeto contenedor exista si sus objetos contenidos no existen.

## **1.8. DIAGRAMAS DE SECUENCIA**

Muestran de forma gráfica los eventos que son realizados por los actores del sistema y la forma en la que se comunican los objetos entre sí. En este diagrama se muestra de forma cronológica las interacciones, y los mensajes de forma explícita (Mindiola, 2012).

Elementos:

- Objetos.

Se los define a partir de los diagramas de casos de uso y se los representa con el nombre del objeto y la clase a la que pertenece.

- Mensajes.

Es la comunicación que se da entre objetos con el propósito de que se ejecute una acción. Esto es considerado un evento, que se representa con una flecha horizontal que va del objeto que generó el mensaje al objeto que lo recibe.

- Métodos y Operaciones

Se representan con rectángulos que están sobre la línea de su respectivo objeto. La longitud determina como se va estableciendo el control sobre la secuencia, ya que el método tiene control desde el inicio hasta el final del rectángulo.

- Recursividad.

Algunas veces el objeto realizará una operación donde se invocará a si mismo. Esto es conocido como recursividad.

## **1.9. PRUEBAS DE SOFTWARE**

Una vez que se ha terminado el desarrollo de un sistema se tiene que aplicar un ejercicio de ingeniería de software que es igual de importante, antes de que se haga la entrega del producto final debe comprobarse que el sistema cumple con todos los requerimientos que se definieron en las etapas anteriores. Aquí se implementan las pruebas de software.

Uno de los aspectos más importantes al momento de realizar las pruebas de software es verificar la calidad del producto que se va a entregar. Según el estándar IEEE 6.10-1990, la calidad es el grado con el que un sistema, componente o proceso cumple con los requisitos especificados y las necesidades o expectativas del cliente o usuario. (SENA, 2015)

Si las pruebas de software son exitosas, se comprobará que se ha construido un sistema correcto.

Las pruebas de software se clasifican en dos categorías: pruebas de caja blanca y pruebas de caja negra.

### **Pruebas de Caja Blanca**

Estas pruebas se centran en cómo es la lógica del sistema y su estructura para cumplir con sus funciones. (SENA, 2015)

En estas pruebas se debe considerar estas acciones:

- Ejecutar todas las sentencias.
- Recorrer todos los caminos independientes de cada componente.
- Comprobar todas las decisiones lógicas.
- Comprobar todos los bucles.
- Implementar situaciones extremas.

### **Pruebas de Caja Negra**

En estas pruebas no se tiene en consideración el funcionamiento interno del sistema, solo se considera que de acuerdo con ciertos datos de entrada el sistema debe generar una salida.

Dos técnicas comunes en estas pruebas son:

Partición de equivalencia. En esta prueba la entrada se divide en clases de datos de donde se pueden derivar casos de prueba considerando que, si la entrada tiene un rango de valor específico, se definirá una clase de equivalencia válida y dos inválidas, y si la entrada es un valor lógico se define una clase de equivalencia válida y otra inválida (SENA, 2015).

Análisis de valores límite. Después de que se han probado los casos de partición de equivalencia, se prueban los valores límite para cada clase.

### **Estrategia de Pruebas**

Las pruebas de software se las realiza de adentro hacia afuera del sistema (SENA, 2015).

**Pruebas unitarias.** Se comprueba la lógica, funcionalidad y especificación de cada unidad de manera aislada.

**Pruebas de integración.** Se enfoca en el diseño y la arquitectura del software y analiza el flujo de datos entre las unidades.

**Pruebas de validación.** Se verifica que se cumpla con los requisitos.

**Pruebas de sistema.** Se prueba el software con todos los elementos que le componen ya como uno solo.

**Pruebas de aceptación.** Aquí el usuario es quien verifica que en realidad el sistema cumple con sus requerimientos.

## **1.10. HERRAMIENTAS DE DESARROLLO**

### **1.10.1. JAVA**

#### **¿Qué es JAVA?**

Java es lenguaje de programación de alto nivel, diseñado fundamentalmente para POO (programación orientada a objetos) que posee características completas para el desarrollo de aplicaciones autónomas e independientes; debido a que convierte su código fuente en instrucciones de máquina, bytecodes, se lo denomina como un lenguaje “interpretado”. Al ser un lenguaje interpretado lo hace totalmente independiente de la plataforma, lo que significa que la compilación no se la realiza directamente por el procesador. En su lugar, utiliza una máquina virtual de Java (JVM, Java Virtual Machine) la cual interpreta los bytecodes. (Groussard, 2012)

#### **Características de JAVA**

Caracterizado por ser un lenguaje de programación independiente de la plataforma y orientado a objetos; Java también posee otra fortaleza para el desarrollo la cual proviene de sus bibliotecas, que incorpora en los paquetes del entorno de desarrollo Java (JDK, Java Development Kit) (Groussard, 2012).

Dichos paquetes contienen centenares de clases y millones de métodos integrados, los cuales permite una mejor y optimizada programación (Groussard, 2012).

SUN caracteriza a Java como un lenguaje sencillo, orientado a objetos, distribuido, interpretado, robusto, securizado, independiente de las arquitecturas, portable, eficaz, multihilo y dinámico.

### **1.10.2. PostgreSQL**

PostgreSQL es un sistema de administración de bases de datos relacionales orientadas a objetos. Fue desarrollado por el Departamento de Ciencias Computacionales de la Universidad de California, Berkeley. (Acebedo, 2016)

Al ser orientado a objetos incluye características como herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad de transacciones. Está liberado bajo la licencia BSD. (Aliaga Ibarra & Miani Flores, 2008)

Es un proyecto open source, debido a esto su desarrollo se da por la colaboración de una comunidad de desarrolladores y organizaciones comerciales, esta comunidad se denomina PostgreSQL Global Development Group (PGDG).

### **Licencia BSD**

Es la licencia de software libre otorgada para los sistemas BSD (Berkeley Software Distribution). Es una licencia con menos restricciones comparada con otras como la GPL, a diferencia de ésta, la licencia BSD permite el uso de código fuente en software no libre. Permite la libre redistribución y modificación, incluso si los sistemas tienen propietario. (Aliaga Ibarra & Miani Flores, 2008)

Las libertades que incluyen esta licencia son:

Usar el programa por cualquier propósito. Los programas pueden ser desarrollados con propósitos educativos, culturales, comerciales, sociales, etc.

Estudiar cómo funciona el programa y adaptarlo a sus necesidades. Se tiene acceso al código fuente para saber cómo funciona.

Seremos capaces de descubrir funciones ocultas, saber cómo se llevan a cabo ciertas tareas, incluso si falta algo para cumplir con algún propósito. La adaptación a las necesidades del desarrollo que se está realizando implica quitar o aumentar las partes según como sea requerido.

Distribuir copias. Esto implica que la distribución del programa que se desarrolle está a consideración de la persona que lo está desarrollando, ya sea de forma gratuita o con algún costo.

Mejorar el programa y liberar las mejoras al público. Esto permite que los desarrolladores puedan mejorar el programa en cuanto a requerimientos de hardware, prestaciones, uso de memoria, disminución de errores y otras modificaciones y distribuirlo al público.

### **Características**

Según (Aliaga Ibarra & Miani Flores, 2008) , las características de PostgreSQL a considerar son:

**Base de datos de nivel empresarial.** Esto implica que todos los datos de la empresa están en un solo lugar, existe facilidad de ingreso a los datos por varios miembros de la empresa, evita que haya redundancia de datos, los datos son almacenados de forma segura y todo esto da mayor eficiencia en el manejo de datos por parte de la empresa.

**Multiplataforma.** Compatible con los sistemas operativos más populares como GNU/Linux, UNIX y Windows.

**Es escalable.** En cuanto a la cantidad de datos que puede manejar y al número de usuarios concurrentes que pueden interactuar.

**Cumplimiento completo ACID** (Atomicidad, Consistencia, Aislamiento, Durabilidad). Son los cuatro atributos primarios que aseguran cualquier transacción.

**Claves foráneas.** La clave foránea (foreign key) es la que relaciona a una columna o grupo de columnas de una tabla con otra columna o grupo de columnas de otra tabla. Esta relación se establece a través de las claves primarias de las columnas.

**Uniones.** La unión se da al realizar una consulta que combina los datos de dos tablas o más, esta combinación es posible con el uso de las claves foráneas.

**Vistas.** La vista es la forma en que se muestran los datos de la o las tablas por medio de una tabla virtual.

**Disparadores.** Son objetos asociados a las tablas de las bases de datos. Estos determinan una función que debe ser ejecutada al momento que se dé un evento específico en la tabla a la que se encuentra asociado en la base de datos.

**Procedimientos almacenados.** Son comandos que pueden ser ejecutados directamente en el servidor de la base de datos y no por el programa cliente. Están almacenados directamente en la base de datos y ayudan a que se mejore el rendimiento de la interacción del cliente con el servidor.

**Incluye tipos de datos SQL como:** Integer, Numeric, Boolean, Char, Var-Char, Date, Interval, Timestamp.

**Almacena objetos binarios como:** Imagen, sonido y video. Los objetos binarios son archivos de gran tamaño, que deben ser manipulados de forma diferente a como se maneja otros tipos de datos dentro de la base.

**Dispone de interfaces de programación nativas.** C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC y otros.

**Documentación.** En la documentación estará presente todos los aspectos técnicos que tiene la base de datos.

**Multi-Version Currency Control (MVCC).** Se encarga de mantener copias paralelas sobre los datos para que la escritura de datos al disco duro sea rápida. Esto busca optimizar las múltiples lecturas y escrituras que se pueden dar sobre la base de datos. A través del comando VACUUM en PostgreSQL actualiza la base de datos y se tiene control sobre la concurrencia que se da.

**Point in Time Recovery (PITR).** Esta característica permite que se recupere la base de datos en caso de que se dé un fallo en la actualización de la base de datos o que por accidente se borre una tabla.

Se determinará cada que tiempo debe hacerse un respaldo para que no se pierda la información almacenada.

**Tablespace.** Es un almacén lógico de los ficheros de la base de datos. En cada uno de estos esta uno o más ficheros donde se almacena toda la información.

**Replicación asincrónica.** Da a la base de datos tolerancia a fallos en los servidores y almacenamiento de red. Toma los cambios que se han realizado en los ficheros por parte del sistema operativo del servidor byte a byte.

**Transacciones anidadas.** Son transacciones que se dan a partir de otras que han empezado en la base de datos. Los cambios de estas transacciones no son visibles sino hasta que la transacción principal ha sido ejecutada completamente.

**Copias de respaldo en línea o en caliente.** El respaldo en caliente toma lugar el momento que la base de datos está en funcionamiento. Al momento de hacer este respaldo se copia todos los ficheros que corresponden a un tablespace

### **Limites Generales**

Tamaño máximo de base de datos: Ilimitado

Tamaño máximo de tabla: 32 TB

Tamaño máximo de fila: 1,6 TB

Tamaño máximo de campo: 1 GB

Máximo de filas por tabla: Ilimitado

Máximo de columnas por tabla: 250-1600 (depende del tipo de columnas)

Máximo de índices por tabla: Ilimitado

## CAPÍTULO 2. DEFINICIÓN DE REQUERIMIENTOS

En este capítulo nos adentramos en la interacción que vamos a tener con el cliente, primero utilizaremos las historias de usuarios que debe ser elaborada por el cliente y en base a estas definiremos los requerimientos que debe cumplir el sistema.

### 2.1. INTERACCIÓN CON EL CLIENTE

#### 2.1.1. Historias de Usuarios

##### Historia de Usuario 1

Historia de Usuario			
Número:	1	Usuario:	Cliente
Nombre Historia:	Funcionalidad Ingresar al Sistema		
Prioridad:	Alta	Riesgo en desarrollo:	bajo
Valor	1	Iteración asignada:	1
Programador responsable:	Alexander Zurita		
Descripción:			
El sistema debe ser manejado por dos tipos de usuarios (Administrador y Empleado), ya que será manejado por los			
Validación:			
El usuario podrá elegir el rol dentro del sistema para acceder a las funcionalidades.			

*Tabla 2-1. Historia de Usuario 1 (Zurita U, 2017).*

##### Historia de Usuario 2

Historia de Usuario			
Número:	2	Usuario:	Cliente
Nombre Historia:	Funcionalidad Administrar Clientes		
Prioridad:	Alta	Riesgo en desarrollo:	bajo
Valor	1	Iteración asignada:	2
Programador responsable:	Alexander Zurita		
Descripción:			
Como usuario del sistema, quiero que se guarde la información de los clientes para tener acceso de una forma más rápida			
Validación:			
El usuario puede almacenar los datos de los clientes de la empresa, puede acceder a estos en cualquier momento y utilizarlos según su conveniencia.			

*Tabla 2-2. Historia de Usuario 2 (Zurita U, 2017).*

### Historia de Usuario 3

Historia de Usuario			
Número:	3	Usuario:	Cliente
Nombre Historia:	Funcionalidad Administrar Proveedores		
Prioridad:	Alta	Riesgo en desarrollo:	bajo
Valor	1	Iteración asignada:	3
Programador responsable:	Alexander Zurita		
Descripción:			
Como usuario del sistema, quiero mantener y acceder a los datos informativos de nuestros proveedores para facilitar el manejo y uso de la misma.			
Validación:			
El usuario puede almacenar los datos de los proveedores de la empresa, puede acceder a estos en cualquier momento y utilizarlos según su conveniencia.			

*Tabla 2-3. Historia de Usuario 3 (Zurita U, 2017).*

### Historia de Usuario 4

Historia de Usuario			
Número:	4	Usuario:	Cliente
Nombre Historia:	Funcionalidad Administrar Productos		
Prioridad:	Alta	Riesgo en desarrollo:	bajo
Valor	3	Iteración asignada:	4
Programador responsable:	Alexander Zurita		
Descripción:			
Como usuario del sistema, quiero mantener y acceder a los datos informativos de nuestros productos para facilitar el manejo y uso			
Validación:			
El usuario puede almacenar los datos de los productos de la empresa, puede acceder a estos en cualquier momento y utilizarlos según su conveniencia.			

*Tabla 2-4. Historia de Usuario 4 (Zurita U, 2017).*

### Historia de Usuario 5

Historia de Usuario			
Número:	5	Usuario:	Cliente
Nombre Historia:	Funcionalidad Ventas		
Prioridad:	Alta	Riesgo en desarrollo:	bajo
Valor	4	Iteración asignada:	5
Programador responsable:	Alexander Zurita		
Descripción:			
Como usuario del sistema, quiero que la información de todas las ventas realizadas a nuestros clientes se mantenga en un registro que sirva de respaldo para nuestra empresa.			
Validación:			
El usuario puede almacenar los datos de las ventas de la empresa, puede acceder a estos en cualquier momento.			

*Tabla 2-5. Historia de Usuario 5 (Zurita U, 2017).*

### Historia de Usuario 6

Historia de Usuario			
Número:	6	Usuario:	Cliente
Nombre Historia:	Funcionalidad Compras		
Prioridad:	Alta	Riesgo en desarrollo:	bajo
Valor	5	Iteración asignada:	6
Programador responsable:	Alexander Zurita		
Descripción:			
Como usuario del sistema, quiero que se lleve un registro de todas las compras de materiales que realizamos como empresa a nuestros proveedores para mantener respaldada esta información.			
Validación:			
El usuario puede almacenar los datos de las compras de la empresa, puede acceder a estos en cualquier momento.			

*Tabla 2-6. Historia de Usuario 6 (Zurita U, 2017).*

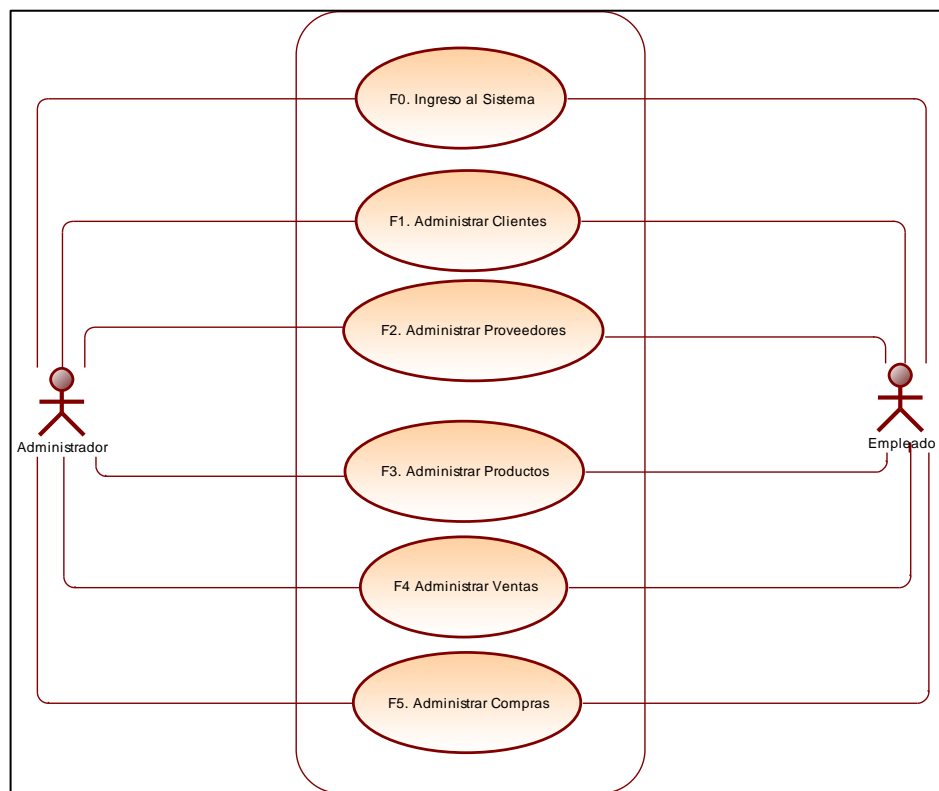
## 2.2. REQUERIMIENTOS

### 2.2.1. Diagramas

A continuación, se muestran los diagramas que indican las funcionalidades que tendrá el sistema y el acceso que tendrán los usuarios que van a interactuar con el mismo.

#### Diagrama General.

El diagrama general que se muestra en la Figura 2.1 nos muestra las funciones generales del sistema y a cuál de ellas tendrá acceso el administrador y el empleado.

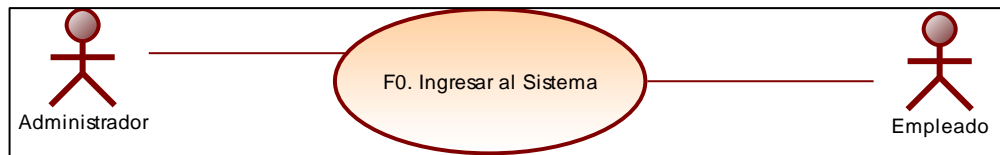


*Figura 2-1. Diagrama general (Zurita U, 2017).*

## Diagrama de Casos de Uso

### F0. Ingresar al Sistema

La función para ingreso al sistema se encuentra modelado en la Figura 2.2



*Figura 2-2. Ingresar al Sistema (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** Se presenta en una interfaz gráfica los usuarios del sistema; por medio de un cuadro de dialogo se ingresa la contraseña del usuario.

#### Flujo Principal.

1. El actor selecciona el usuario del sistema.
2. El sistema presenta en pantalla un cuadro de diálogo para la contraseña.
3. El actor ingresa la contraseña.
4. El actor presiona la opción aceptar.
5. El sistema valida la contraseña ingresada. (C1)
6. El sistema presenta mensaje de confirmación.
7. El sistema presenta la ventana principal.

#### Flujo Alternativo.

6. El sistema presenta mensaje de error.

#### Excepciones.

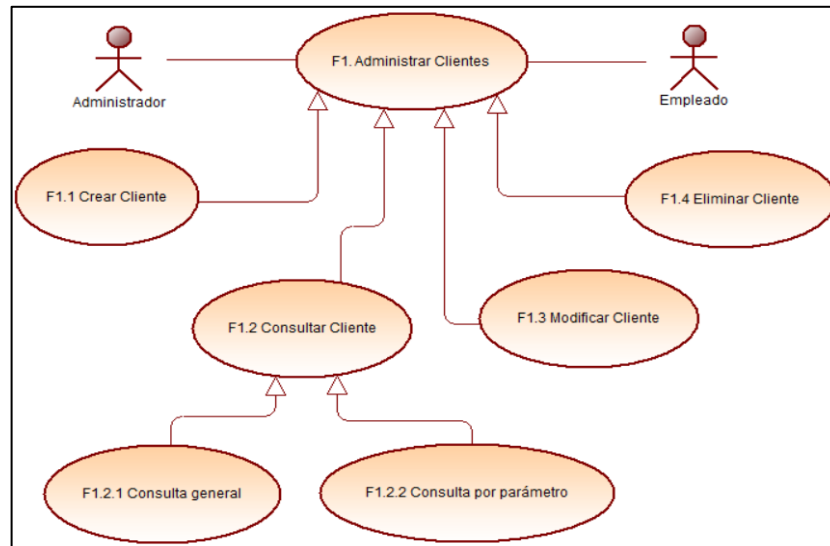
##### Código, causa, mensaje.

**C1,** Posible desconexión de la base de datos, Salga de sistema e ingrese de nuevo.

**C2,** Los datos ingresados no son correctos, Ingrese el tipo de dato correspondiente.

## F1. Administrar Clientes

El conjunto de funciones que se encuentra dentro de la Administración de Clientes esta modelado en la Figura 2-3.



*Figura 2-3. Administrar Clientes (Zurita U, 2017).*

### F1.1 Ingresar Cliente



*Figura 2-4. Ingresar Cliente (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** Mediante un formulario se ingresarán los datos que corresponden con la información de un nuevo cliente.

#### Flujo Principal.

1. El actor selecciona la opción Administrar Clientes.
2. El sistema presenta en pantalla una ventana con el formulario correspondiente.
3. El actor presiona botón Nuevo.
4. El actor ingresa los datos del cliente
5. El actor pulsa el botón guardar.

6. El sistema presenta mensaje de confirmación.
7. El actor confirma la acción.
8. En la base de datos se autogenera un código.
9. El sistema almacena la información. (C1,C2)

### **Flujo Alternativo.**

7. El sistema no continúa con la acción.

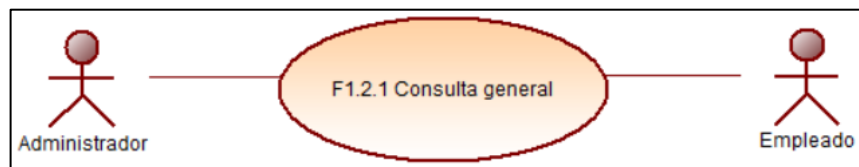
### **Excepciones.**

#### **Código, causa, mensaje.**

**C1,** Posible desconexión de la base de datos, Salga de sistema e ingrese de nuevo.

**C2,** Los datos ingresados no son correctos, Ingrese el tipo de dato correspondiente.

### **F1.2.1 Consulta general**



*Figura 2-5. Consulta General (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá realizar una consulta de la información de todos los registros de los clientes. Dicha información será presentada en pantalla.

### **Flujo Principal.**

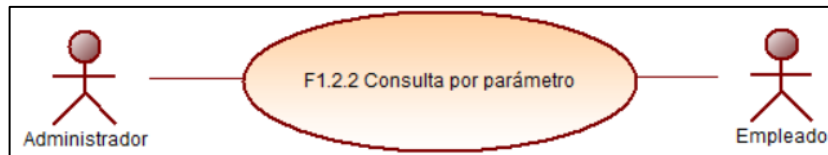
1. El actor selecciona Administrar Clientes.
2. El actor selecciona la opción Ver Todos. (C3)
3. El sistema obtiene todos los registros de clientes.
4. El sistema despliega la ventana Consulta General.
5. El sistema muestra en la ventana todos los registros de los clientes.

**Excepciones.**

**Código, causa, mensaje.**

**C3**, No existe datos en el sistema, Ingrese datos del cliente.

**F1.2.2 Consulta por parámetro**



*Figura 2-6. Consulta por Parámetro (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá realizar una consulta específica dependiendo de un parámetro dentro de los registros almacenados. Se presentará en pantalla la información de los registros que cumplan con el parámetro.

**Flujo Principal.**

1. El actor selecciona la opción Administrar Clientes.
  2. El sistema despliega la ventana Clientes.
  3. El actor establece el parámetro de búsqueda.
  4. El sistema obtiene todos los registros de clientes según el parámetro escogido.
  5. El sistema despliega la ventana Consulta General.
  6. El sistema muestra la información de los clientes según el parámetro escogido.
- (C4)

**Flujo Alternativo.**

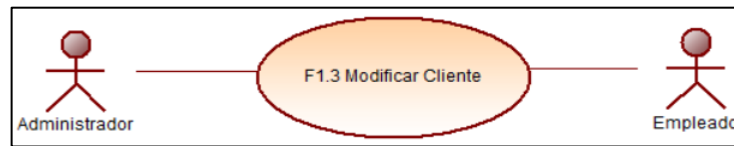
3. Ver caso de uso F1.1 Ingresar cliente.

**Excepciones.**

**Código, causa, mensaje.**

**C4**, No existen datos en el sistema según los parámetros establecidos, Ingrese datos del cliente.

### F1.3 Modificar cliente.



*Figura 2-7. Modificar Cliente (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá modificar un registro almacenado mediante los campos presentados en el formulario.

#### **Flujo Principal.**

1. El actor selecciona la opción Administrar Clientes.
2. El sistema despliega la ventana
3. El actor ingresa el número de identificación del cliente.
4. El sistema valida el número de identificación del cliente. (C5)
5. El sistema despliega la información correspondiente al número de identificación del cliente.
6. El actor modifica los datos del cliente que sean necesarios.
7. El actor pulsa el botón Guardar. (C2)
8. El sistema actualiza y almacena los datos del cliente.

#### **Flujo Alternativo.**

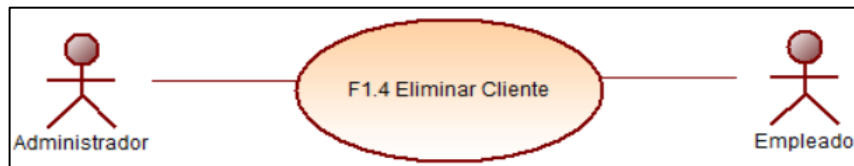
5. Ver caso de uso F1.1 Ingresar cliente.

#### **Excepciones.**

##### **Código, causa, mensaje.**

**C5,** El número de identificación del cliente no ha sido encontrado, Ingrese correctamente el número de identificación del cliente o Ingrese un nuevo registro.

### F1.4 Eliminar cliente.



*Figura 2-8. Eliminar Cliente (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá eliminar un registro almacenado de clientes.

#### **Flujo Principal.**

1. El actor selecciona la opción Administrar Clientes.
2. El sistema despliega la interfaz (ventana)
3. El actor ingresa el número de identificación del cliente.
4. El sistema valida el número de identificación del cliente. (C6)
5. El sistema muestra toda la información correspondiente al número de identificación ingresado.
6. El actor pulsa el botón Eliminar.
7. El sistema emite una alerta para confirmar si se desea eliminar el registro.
8. El sistema elimina toda la información del cliente. (C7)

#### **Flujo Alternativo.**

5. Ver caso de uso F1.1 Ingresar cliente o F1.3 Modificar cliente.

#### **Excepciones.**

##### **Código, causa, mensaje.**

**C6,** Registro no encontrado, Ingrese el número de identificación correctamente.

**C7,** No se pudo eliminar el registro, Error de conexión con la base de datos.

## F2. Administrar Proveedores

El conjunto de funciones que forman la función de Administrar Proveedores está modelado en la Figura 2.9

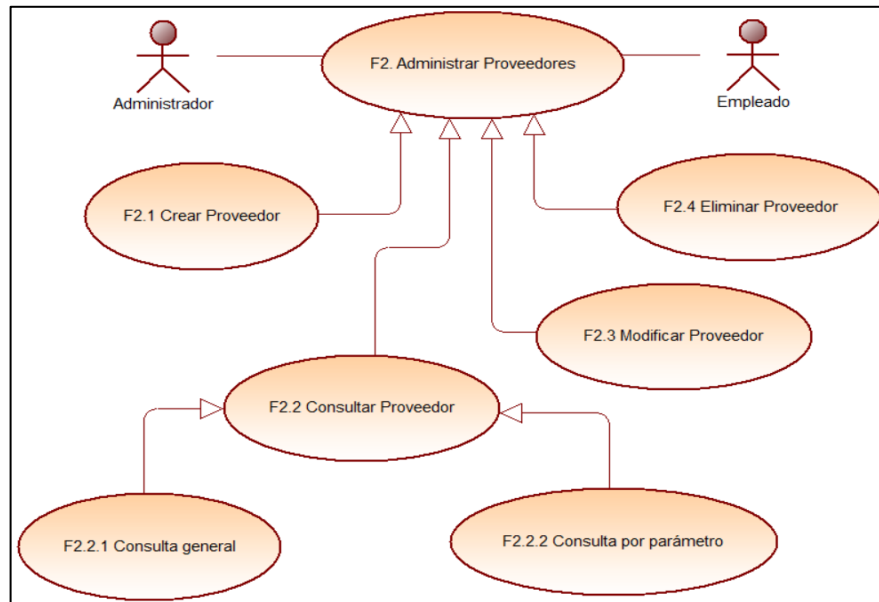


Figura 2-9. Administrar Proveedores (Zurita U, 2017).

### F2.1 Ingresar Proveedor

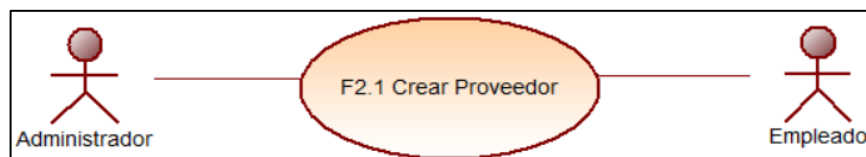


Figura 2-10. Ingresar Proveedores (Zurita U, 2017).

**Actores:** Administrador, empleado.

**Descripción:** Mediante un formulario se ingresarán los datos que corresponden con la información de un nuevo proveedor.

#### Flujo Principal.

1. El actor selecciona la opción Administrar Proveedores.
2. El sistema presenta en pantalla una ventana con el formulario correspondiente.
3. El actor presiona botón Nuevo.
4. El actor ingresa los datos del proveedor.

5. El actor pulsa el botón guardar.
6. El sistema presenta mensaje de confirmación.
7. El actor confirma la acción.
8. En la base de datos se autogenera un código.
9. El sistema almacena la información. (C1,C2)

#### **Flujo Alternativo.**

8. El sistema no continúa con la acción.

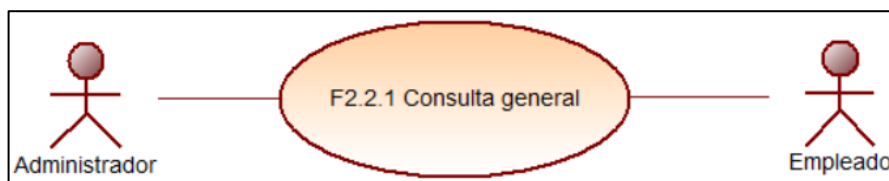
#### **Excepciones.**

##### **Código, causa, mensaje.**

**C1,** Posible desconexión de la base de datos, Salga de sistema e ingrese de nuevo.

**C2,** Los datos ingresados no son correctos, Ingrese el tipo de dato correspondiente.

#### **F2.2.1 Consulta General**



*Figura 2-11. Consulta General (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá realizar una consulta de la información de todos los registros de los proveedores. Dicha información será presentada en pantalla.

#### **Flujo Principal.**

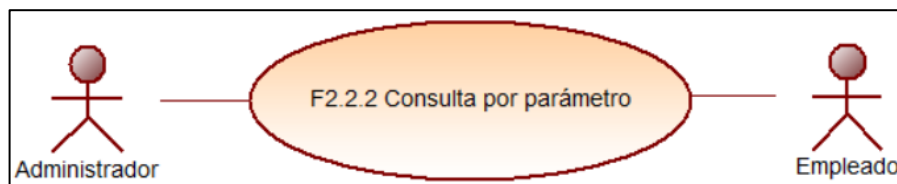
1. El actor selecciona Administrar Proveedores.
2. El actor selecciona la opción Ver Todos. (C8)
3. El sistema obtiene todos los registros de proveedores.
4. El sistema despliega la ventana Consulta General.
5. El sistema muestra en la ventana todos los registros de los proveedores.

**Excepciones.**

**Código, causa, mensaje.**

**C8**, No existe datos en el sistema, Ingrese datos en proveedores.

**F2.2.2 Consulta Por Parámetro**



*Figura 2-12. Consultar por Parámetros (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá realizar una consulta específica dependiendo de un parámetro dentro de los registros almacenados. Se presentará en pantalla la información de los registros que cumplan con el parámetro.

**Flujo Principal.**

1. El actor selecciona la opción Administrar Proveedores.
2. El sistema despliega la ventana Proveedores.
3. El actor establece el parámetro de búsqueda.
4. El sistema obtiene todos los registros de proveedores según el parámetro escogido.
5. El sistema despliega la ventana Consulta General.
6. El sistema muestra la información de los proveedores según el parámetro escogido. (C9)

**Flujo Alternativo.**

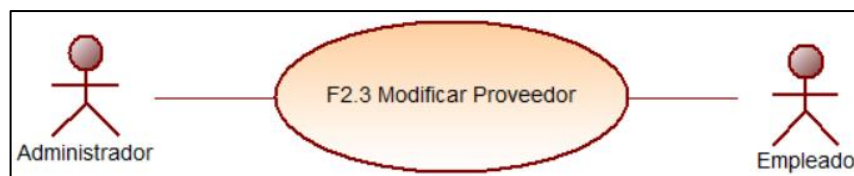
9. Ver caso de uso F2.1 Ingresar Proveedor.

### **Excepciones.**

#### **Código, causa, mensaje.**

**C9**, No existen datos en el sistema según los parámetros establecidos, Ingrese datos del proveedor.

### **F2.3 Modificar Proveedor**



*Figura 2-13. Modificar Proveedor (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá modificar un registro almacenado mediante los campos presentados en el formulario.

#### **Flujo Principal.**

1. El actor selecciona la opción Administrar Proveedores.
2. El sistema despliega la ventana
3. El actor ingresa el número de identificación del proveedor.
4. El sistema valida el número de identificación del proveedor. (C10)
5. El sistema despliega la información correspondiente al número de identificación del proveedor.
6. El actor modifica los datos del proveedor que sean necesarios.
7. El actor pulsa el botón Guardar. (C10)
8. El sistema actualiza y almacena los datos del proveedor.

#### **Flujo Alternativo.**

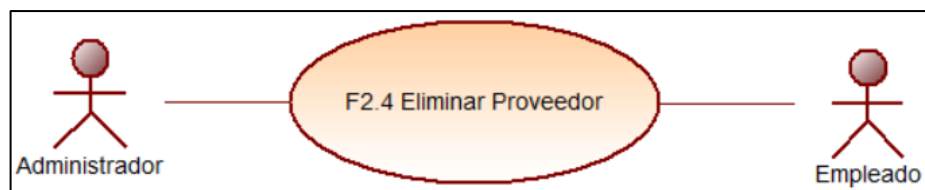
5. Ver caso de uso F2.1 Ingresar proveedor.

### **Excepciones.**

#### **Código, causa, mensaje.**

**C10**, El número de identificación del proveedor no ha sido encontrado, Ingrese correctamente el número de identificación del proveedor o Ingrese un nuevo registro.

### **F2.4 Eliminar Proveedor**



*Figura 2-14. Eliminar Proveedor (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá eliminar un registro almacenado de clientes.

#### **Flujo Principal.**

1. El actor selecciona la opción Administrar Proveedores.
2. El sistema despliega la interfaz (ventana).
3. El actor ingresa el código del proveedor.
4. El sistema valida el código. (C11)
5. El sistema muestra toda la información correspondiente al código ingresado.
6. El actor pulsa el botón Eliminar.
7. El sistema emite una alerta para confirmar si se desea eliminar el registro.
8. El sistema elimina toda la información del proveedor. (C12)

#### **Flujo Alternativo.**

6. Ver caso de uso F2.1 Ingresar Proveedor o F2.3 Modificar Proveedor.

**Excepciones.**

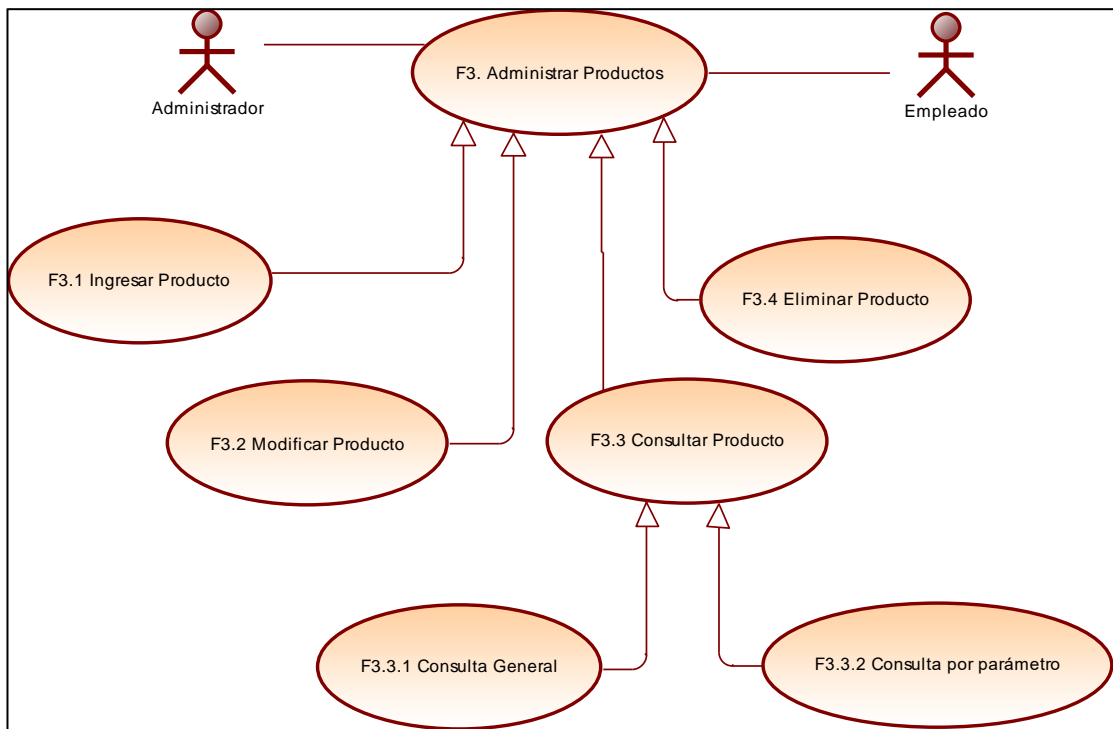
**Código, causa, mensaje.**

**C11**, Registro no encontrado, Ingrese el número de identificación correctamente.

**C12**, No se pudo eliminar el registro, Error de conexión con la base de datos.

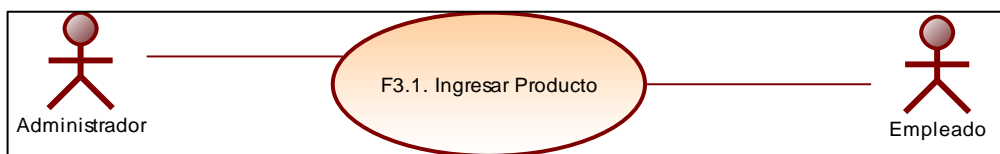
**F3. Administrar Productos**

El conjunto de funciones que forman la función de Administrar Productos está modelado en la Figura 2-15.



*Figura 2-15. Administrar Productos (Zurita U, 2017).*

**F3.1 Ingresar Producto**



*Figura 2-16. Ingresar Producto (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** Mediante un formulario se ingresarán los datos que corresponden con la información de un nuevo producto.

**Flujo Principal.**

1. El actor selecciona la opción Administrar Productos.
2. El sistema presenta en pantalla una ventana con el formulario correspondiente.
3. El actor presiona botón Nuevo.
4. El actor ingresa los datos del producto.
5. El actor pulsa el botón guardar.
6. El sistema presenta mensaje de confirmación.
7. El actor confirma la acción.
8. En la base de datos se autogenera un código.
9. El sistema almacena la información. (C1,C2)

**Flujo Alternativo.**

8. El sistema no continúa con la acción.

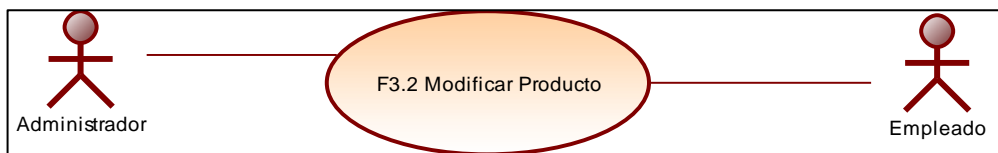
**Excepciones.**

**Código, causa, mensaje.**

**C1,** Posible desconexión de la base de datos, Salga de sistema e ingrese de nuevo.

**C2,** Los datos ingresados no son correctos, Ingrese el tipo de dato correspondiente.

**F3.2 Modificar Producto**



*Figura 2-17. Modificar Producto (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá modificar un registro almacenado mediante los campos presentados en el formulario.

**Flujo Principal.**

1. El actor selecciona la opción Administrar Productos.
2. El sistema despliega la ventana
3. El actor ingresa el número de identificación del Producto.
4. El sistema valida el número de identificación del Producto. (C10)
5. El sistema despliega la información correspondiente al número de identificación del Producto.
6. El actor modifica los datos del Producto que sean necesarios.
7. El actor pulsa el botón Guardar. (C10)
8. El sistema actualiza y almacena los datos del Producto.

**Flujo Alternativo.**

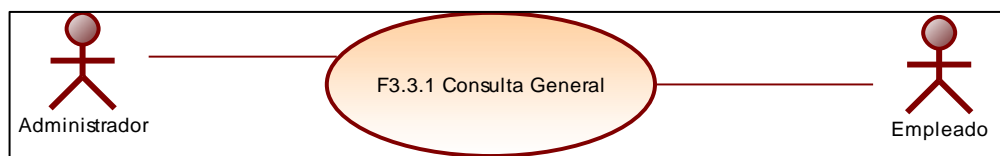
4. Ver caso de uso F3.1 Ingresar Producto.

**Excepciones.**

**Código, causa, mensaje.**

**C10,** El número de identificación del Producto no ha sido encontrado, Ingrese correctamente el número de identificación del Producto o Ingrese un nuevo registro.

**F3.3.1 Consulta General**



*Figura 2-18. Consulta General (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá realizar una consulta de la información de todos los registros de los Productos. Dicha información será presentada en pantalla.

**Flujo Principal.**

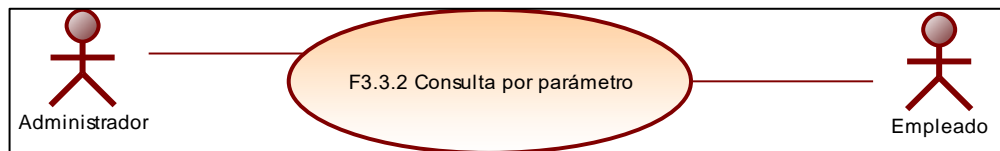
1. El actor selecciona Administrar Productos.
2. El actor selecciona la opción Ver Todos. (C8)
3. El sistema obtiene todos los registros de Productos.
4. El sistema despliega la ventana Consulta General.
5. El sistema muestra en la ventana todos los registros de los Productos.

**Excepciones.**

**Código, causa, mensaje.**

**C8,** No existe datos en el sistema, Ingrese datos en Productos.

**F3.3.2 Consulta Por Parámetro**



*Figura 2-19. Consulta por Parámetro (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá realizar una consulta específica dependiendo de un parámetro dentro de los registros almacenados. Se presentará en pantalla la información de los registros que cumplan con el parámetro.

**Flujo Principal.**

1. El actor selecciona la opción Administrar Productos.
2. El sistema despliega la ventana Productos.
3. El actor establece el parámetro de búsqueda.

4. El sistema obtiene todos los registros de Productos según el parámetro escogido.
5. El sistema despliega la ventana Consulta General.
6. El sistema muestra la información de los Productos según el parámetro escogido. (C9)

### Flujo Alternativo.

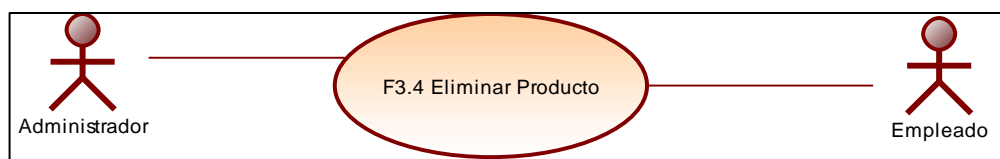
4. Ver caso de uso F3.1 Ingresar Producto.

### Excepciones.

#### Código, causa, mensaje.

C9, No existen datos en el sistema según los parámetros establecidos, Ingrese datos del Producto.

### F3.4 Eliminar Producto



*Figura 2-20. Eliminar Producto (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá eliminar un registro almacenado de clientes.

### Flujo Principal.

1. El actor selecciona la opción Administrar Productos.
2. El sistema despliega la interfaz (ventana).
3. El actor ingresa el código del Producto.
4. El sistema valida el código. (C11)
5. El sistema muestra toda la información correspondiente al código ingresado.
6. El actor pulsa el botón Eliminar.
7. El sistema emite una alerta para confirmar si se desea eliminar el registro.
8. El sistema elimina toda la información del Producto. (C12)

**Flujo Alternativo.**

- 4. Ver caso de uso F3.1 Ingresar Producto o F3.3 Modificar Producto.

**Excepciones.**

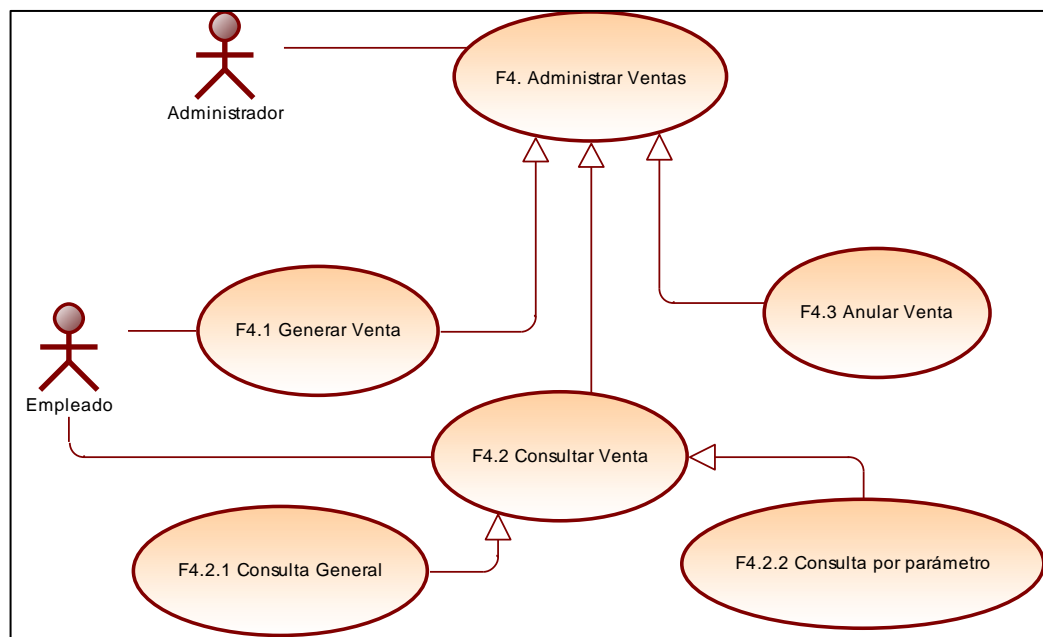
**Código, causa, mensaje.**

**C11,** Registro no encontrado, Ingrese el número de identificación correctamente.

**C12,** No se pudo eliminar el registro, Error de conexión con la base de datos.

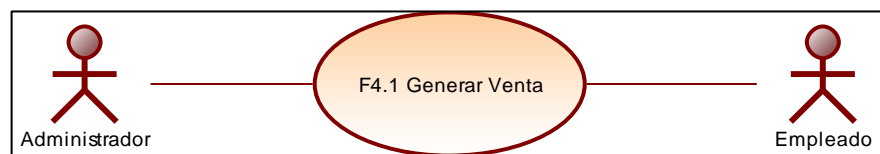
**F4. Administrar Ventas**

El modelado del conjunto de funciones que conforman la Administración de Ventas se muestra en la Figura 2.21



*Figura 2-21. Administrar Ventas (Zurita U, 2017).*

**F4.1 Generar Venta**



*Figura 2-22. Generar Venta (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** Mediante un formulario se ingresarán los datos que corresponden con la información de una nueva venta.

**Flujo Principal.**

1. El actor selecciona la opción administrar ventas.
2. El sistema presenta en pantalla una ventana con el formulario correspondiente.
3. El actor selecciona el código del cliente.
4. El sistema asigna el Estado como activo.
5. El actor ingresa la descripción del o los productos y sus precios.
6. El sistema realiza los cálculos de la venta.
7. El actor ingresa la fecha y observaciones de la venta.
8. El actor pulsa el botón guardar. (C1, C2)
9. El sistema autogenera código para el registro de venta.
10. El sistema almacena la información.
11. Se genera la factura.

**Flujo Alterno.**

3. Ver caso de uso F1.1 Ingresar Cliente.

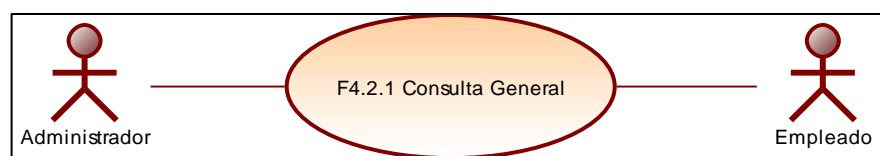
**Excepciones.**

**Código, causa, mensaje.**

**C1,** Posible desconexión de la base de datos, Salga de sistema e ingrese de nuevo.

**C2,** Los datos ingresados no son correctos, Ingrese el tipo de dato correspondiente.

**F4.2.1 Consulta General**



*Figura 2-23. Consulta General (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá realizar una consulta de la información de todos los registros de las ventas. Dicha información será presentada en pantalla.

**Flujo Principal.**

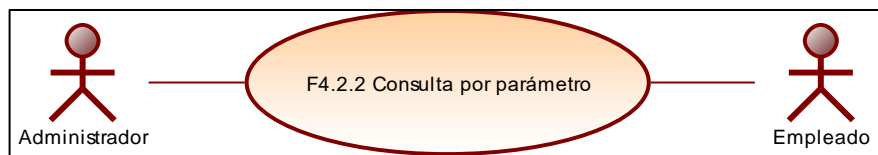
1. El actor selecciona la opción Administrar Ventas.
2. El sistema despliega la ventana Ventas.
3. El actor presiona Consulta General.
4. El sistema obtiene todos los registros de Ventas. (C11)
5. El sistema presenta la ventana Consulta.
6. El sistema muestra en pantalla todos los registros de las Ventas.

**Excepciones.**

**Código, causa, mensaje.**

**C11,** No existe datos en el sistema, Sin registro de ventas.

**F4.2.2 Consulta por Parámetro**



*Figura 2-24. Consulta por Parámetro (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá realizar una consulta específica dependiendo de un parámetro dentro de los registros almacenados. Se presentará en pantalla la información de los registros que cumplan con el parámetro.

**Flujo Principal.**

1. El actor selecciona la opción Administrar Ventas.
2. El sistema despliega la ventana.
3. El actor establece el parámetro de búsqueda.

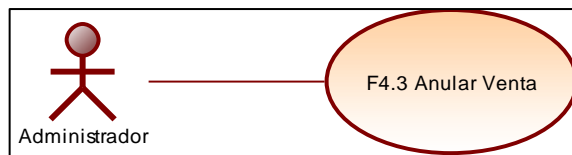
4. El sistema presenta la ventana Consulta.
5. El sistema muestra la información de las ventas según el parámetro escogido.  
(C12)

### **Excepciones.**

### **Código, causa, mensaje.**

**C12**, No existen datos en el sistema según los parámetros establecidos, Sin registro de ventas.

### **F4.3 Anular Venta**



*Figura 2-25. Anular Venta (Zurita U, 2017).*

**Actores:** Administrador.

**Descripción:** El usuario podrá anular un registro de venta.

### **Flujo Principal.**

1. El actor selecciona la opción Administrar Ventas.
2. El sistema despliega la ventana.
3. El actor ingresa el código de Venta.
4. El sistema busca el registro relacionado con el código ingresado. (C13)
5. El sistema presenta la información del registro en pantalla.
6. El actor presiona el botón Anular.
7. El sistema presenta una alerta de confirmación para anular la venta.
8. El actor confirma la operación.
9. El sistema cambia el estado de la venta a anulado.

### **Flujo Alternativo.**

8. El actor no confirma la operación.

9. El sistema regresa a la lista de registros.

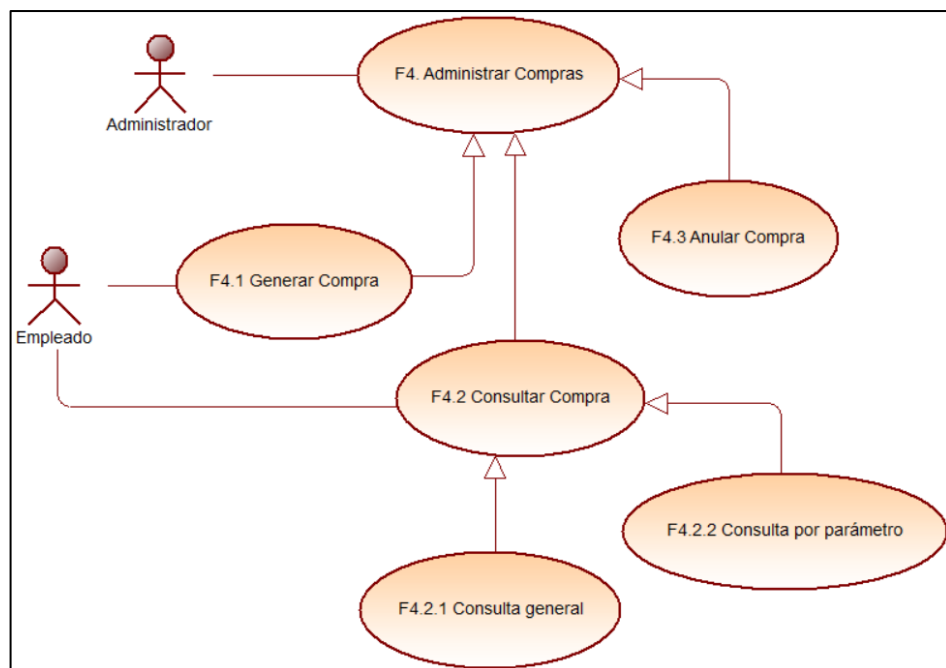
**Excepciones.**

**Código, causa, mensaje.**

**C13**, No existen datos en el sistema según los parámetros establecidos, Sin registro de ventas.

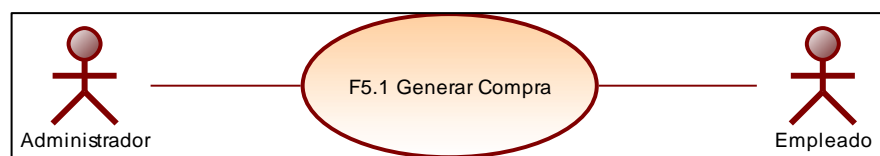
**F5. Administrar Compras**

La Figura 2-26 nos muestra el modelado de las funciones que conforman la Administración de Compras.



*Figura 2-26. Administrar Compras (Zurita U, 2017).*

**F5.1 Generar Compra.**



*Figura 2-27. Generar Compra (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** Mediante un formulario se ingresarán los datos que corresponden con la información de una nueva compra.

**Flujo Principal.**

1. El actor selecciona la opción administrar compras.
2. El sistema presenta en pantalla una ventana con el formulario correspondiente.
3. El actor selecciona el código del proveedor.
4. El sistema asigna el Estado como activo.
5. El actor ingresa la descripción del o los productos y sus precios.
6. El sistema realiza los cálculos de la compra.
7. El actor ingresa la fecha y observaciones de la compra.
8. El actor pulsa el botón guardar. (C14, C15)
9. El sistema autogenera código para el registro de compra.
10. El sistema almacena la información.
11. Se genera la orden de compra.

**Flujo Alternativo.**

3. El actor presiona el botón Nuevo Proveedor, caso de uso F2.1 Ingresar Proveedor

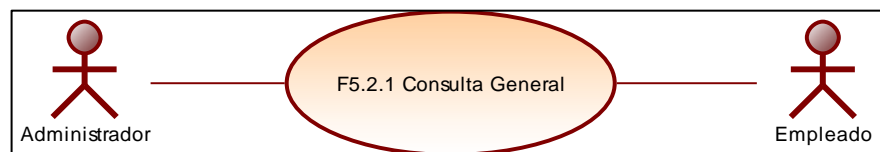
**Excepciones.**

**Código, causa, mensaje.**

**C14,** Posible desconexión de la base de datos, Salga de sistema e ingrese de nuevo.

**C15,** Los datos ingresados no son correctos, Ingrese el tipo de dato correspondiente.

**F5.2.1 Consulta General**



*Figura 2-28. Consulta General (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá realizar una consulta de la información de todos los registros de compras. Dicha información será presentada en pantalla.

**Flujo Principal.**

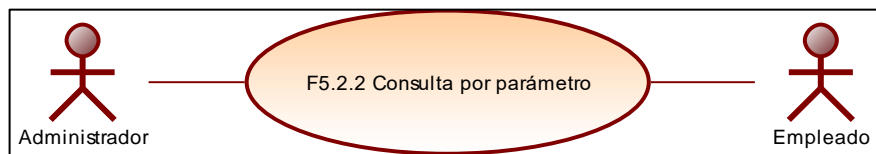
1. El actor selecciona la opción Administrar Compras.
2. El sistema despliega la ventana Compras.
3. El actor presiona Consulta General.
4. El sistema presenta la ventana Consulta.
5. El sistema obtiene todos los registros de Compras. (C16)
6. El sistema muestra en pantalla todos los registros de las Compras.

**Excepciones.**

**Código, causa, mensaje.**

**C16,** No existe datos en el sistema, Sin registro de ventas.

**F5.2.2 Consulta por Parámetro**



*Figura 2-29. Consulta por Parámetro (Zurita U, 2017).*

**Actores:** Administrador, empleado.

**Descripción:** El usuario podrá realizar una consulta específica dependiendo de un parámetro dentro de los registros almacenados. Se presentará en pantalla la información de los registros que cumplan con el parámetro.

**Flujo Principal.**

1. El actor selecciona la opción Administrar Compras.
2. El sistema despliega la ventana.
3. El actor establece el parámetro de búsqueda.
4. El sistema presenta la ventana Consulta.

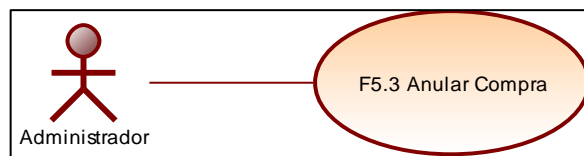
5. El sistema muestra la información de las compras según el parámetro escogido.  
(C17)

**Excepciones.**

**Código, causa, mensaje.**

**C17**, No existen datos en el sistema según los parámetros establecidos, Sin registro de compras.

**F5.3 Anular Compra**



*Figura 2-30. Anular Compra (Zurita U, 2017).*

**Actores:** Administrador.

**Descripción:** El usuario podrá anular un registro de compra.

**Flujo Principal.**

1. El actor selecciona la opción Administrar Compras.
2. El sistema despliega la ventana.
3. El actor establece el parámetro de búsqueda.
4. El sistema muestra la información de las compras según el parámetro escogido.  
(C18)
5. El usuario selecciona el registro de compra que desea anular.
6. El sistema presenta el registro seleccionado.
7. El usuario presiona el botón Anular.
8. El sistema presenta una alerta de confirmación para anular la compra.
9. El usuario confirma la operación.
10. El sistema cambia el estado de la compra a anulado.

**Flujo Alternativo.**

9. El usuario no confirma la operación.

10. El sistema regresa a la lista de registros.

**Excepciones.**

**Código, causa, mensaje.**

**C18**, No existen datos en el sistema según los parámetros establecidos, Sin registro de ventas.

### **CAPÍTULO 3. PLANIFICACIÓN Y ELABORACIÓN**

En este capítulo se da a conocer el desarrollo del sistema una vez que se han definido todos los requerimientos.

Se define el plan a seguir para cada iteración que realizaremos en el desarrollo del sistema.

Se presenta los diagramas de clases y la codificación.

Para un correcto diseño y codificación del sistema es importante seguir estándares ya establecidos, de esta manera se facilitará la legibilidad, el mantenimiento y la mejora del código.

Los estándares que se ha seguido para el desarrollo de este sistema se muestran en el Anexo 1

.

#### **3.1. PLANIFICACION**

##### **3.1.1. ITERACION 1**

**Que se va a realizar.**

Desarrollo de la funcionalidad de administrar clientes.

Desarrollo de los diagramas de clases.

Diseño de la base de datos.

Codificación.

Prueba unitaria.

Prueba de aceptación con el usuario.

**Tiempo.**

De 5 a 7 días.

**3.1.2. ITERACION 2**

**Que se va a realizar.**

Corrección de las observaciones que se tuvieron en la iteración 1 por parte del usuario.

Desarrollo de la funcionalidad de administrar proveedores.

Desarrollo de los diagramas de clases.

Diseño de la base de datos.

Codificación.

Prueba unitaria.

Integración con lo desarrollado en la iteración 1.

Prueba de aceptación con el usuario.

**Tiempo.**

De 5 a 7 días.

**3.1.3. ITERACION 3**

**Que se va a realizar.**

Corrección de las observaciones dadas por el usuario en la iteración 2.

Desarrollo de la funcionalidad del administrar productos.

Desarrollo de los diagramas de clases.

Diseño de la base de datos.

Codificación.

Prueba unitaria.

Integración con lo desarrollado en las iteraciones 1 y 2.

Prueba de aceptación con el usuario.

**Tiempo.**

De 5 a 7 días.

**3.1.4. ITERACION 4**

**Que se va a realizar.**

Corrección de las observaciones dadas por el usuario en la iteración 3.

Desarrollo de la funcionalidad del administrar ventas.

Desarrollo de los diagramas de clases.

Diseño de la base de datos.

Codificación.

Prueba unitaria.

Integración con lo desarrollado en las iteraciones 1, 2 y 3.

Prueba de aceptación con el usuario.

**Tiempo.**

De 5 a 7 días.

**3.1.5. ITERACION 5**

**Que se va a realizar.**

Corrección de las observaciones dadas por el usuario en la iteración 4.

Desarrollo de la funcionalidad del administrar compras.

Desarrollo de los diagramas de clases.

Diseño de la base de datos.

Codificación.

Prueba unitaria

Integración con lo desarrollado en las iteraciones 1, 2, 3 y 4.

Prueba general.

Prueba de aceptación con el usuario.

**Tiempo.**

De 5 a 7 días.

### **3.1.6. ITERACION 6**

**Que se va a realizar.**

Corrección de las observaciones dadas por el usuario en la iteración 5.

Desarrollo de la funcionalidad Ingresar al sistema.

Desarrollo de los diagramas de clases.

Diseño de la base de datos.

Codificación.

Prueba unitaria

Integración con lo desarrollado en las iteraciones 1, 2, 3, 4 y 5.

Prueba general.

Prueba de aceptación con el usuario.

Entrega.

## Tiempo

De 4 a 6 días.

## 3.2. ELABORACIÓN DEL SISTEMA

### 3.2.1. Modelo Físico

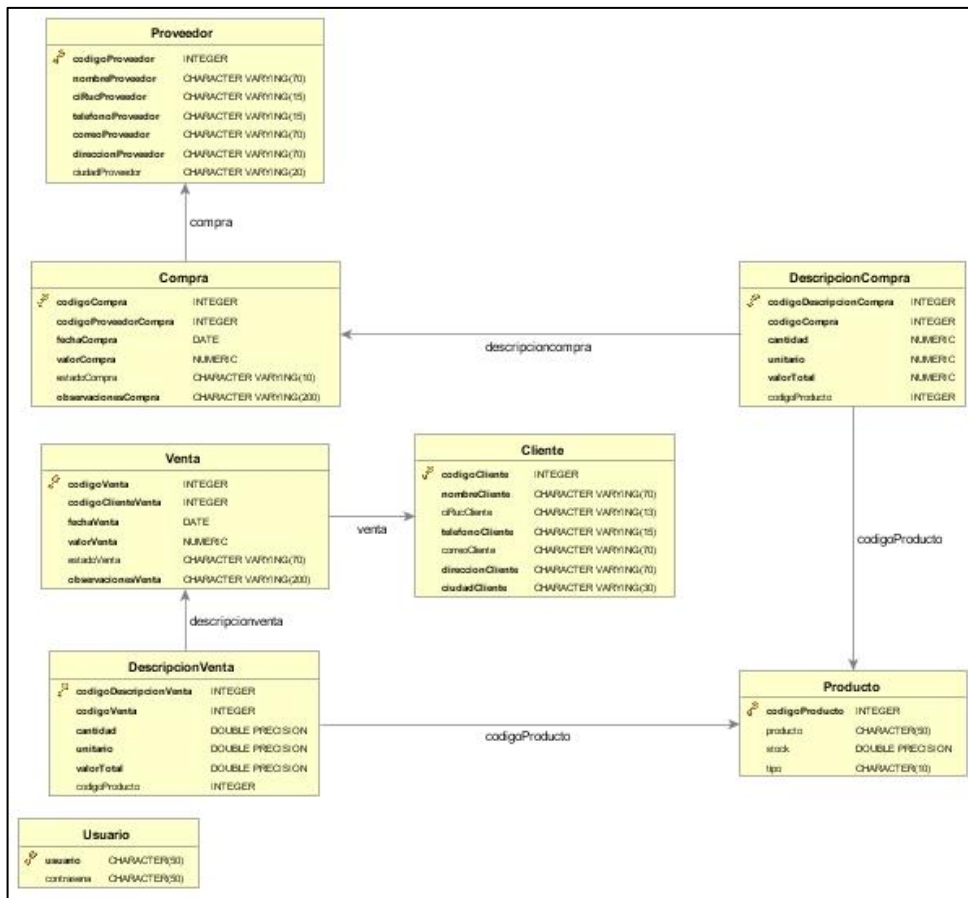


Figura 3-1. Modelo Físico (Zurita U, 2017).

### 3.2.2. Modelo Relacional

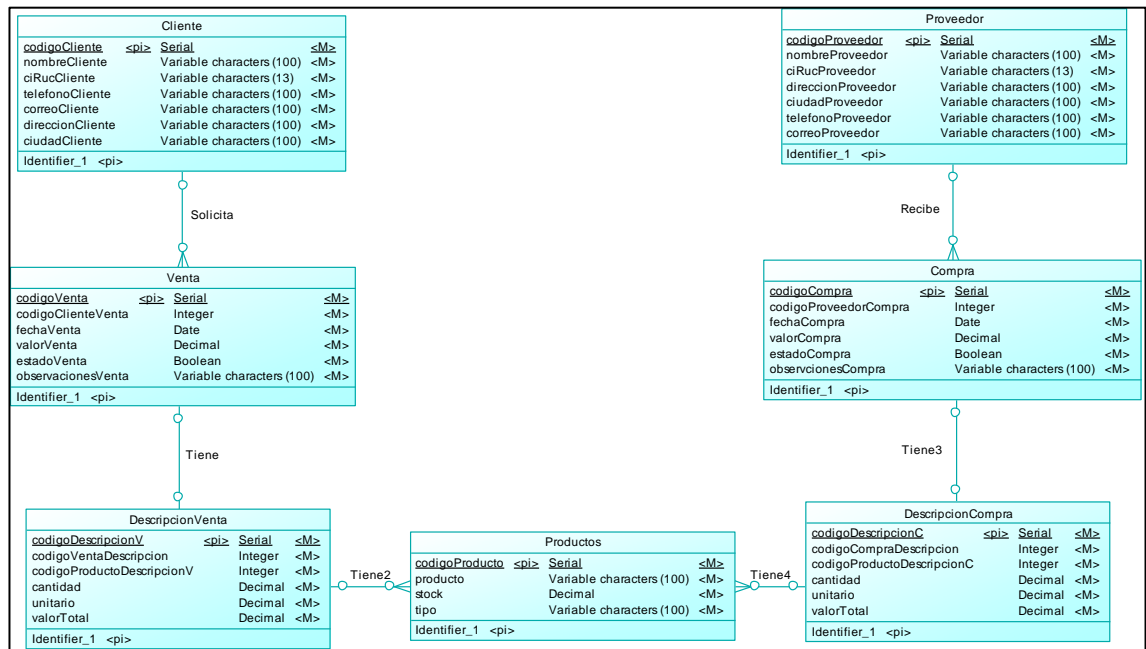


Figura 3-2. Modelo Relacional (Zurita U, 2017).

### 3.3. ITERACIÓN 1 – Ingresar al sistema

En esta iteración realizaremos el desarrollo e implementación de la funcionalidad F0. Ingresar al sistema.

#### 3.3.1. Diseño

La clase necesaria para la implementación de esta funcionalidad es Usuario.

### DIAGRAMA DE CLASES

#### Interfaz Gráfica (GUI)

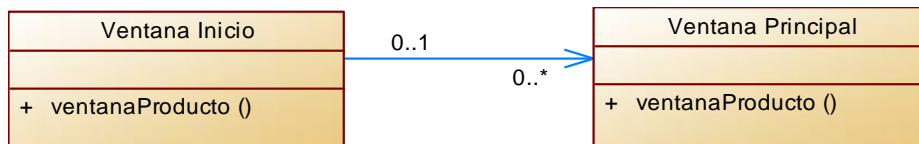
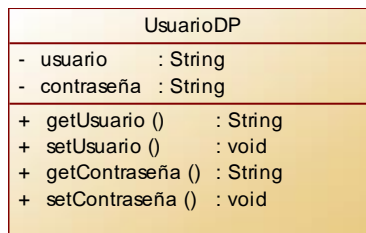


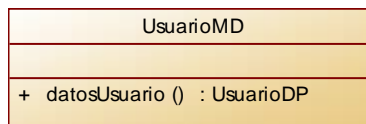
Figura 3-3. Diagrama de clases GUI Iteración 1 (Zurita U, 2017).

### **Dominio del Problema (DP)**



*Figura 3-4. Diagrama de clases DP Iteración 1 (Zurita U, 2017).*

### **Manejo de Datos (MD)**

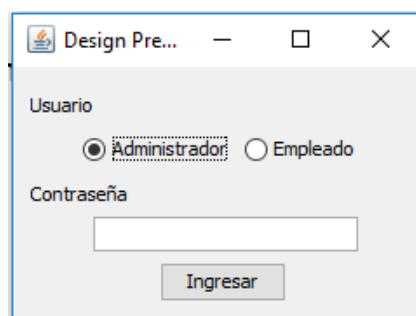


*Figura 3-5. Diagrama de clases MD Iteración 1 (Zurita U, 2017).*

### **3.3.2. Desarrollo**

Se presenta prototipos de GUI, que servirán como base y guía para la elaboración del sistema. Estos prototipos cumplen con los requerimientos solicitados por el usuario.

#### **Prototipo de GUI – Inicio**



*Figura 3-6. GUI Prototipo Inicio (Zurita U, 2017).*

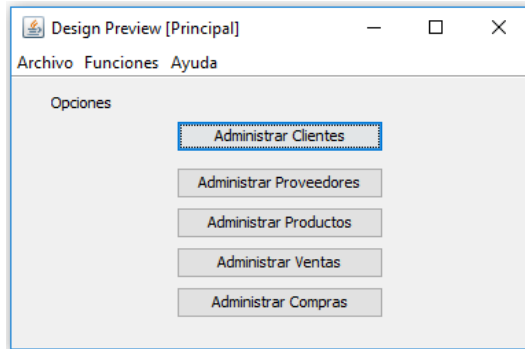
#### **Descripción:**

El prototipo de la primera ventana, nombrada “Ventana Inicio”, se la realiza en base a la historia de usuario 1.

El diseño de esta interfaz gráfica se la desarrolla sencilla y sin campos innecesarios para el ingreso al sistema, siendo amigable con el usuario y eficaz. Cumpliendo con lo solicitado por el usuario.

- Se presenta dos opciones del tipo de Usuario para el acceso al sistema.
- Un cuadro de texto para el ingreso de la contraseña.
- Un botón de acción para interactuar con el sistema.

### Prototipo de GUI – Principal



*Figura 3-7. GUI Prototipo Principal (Zurita U, 2017).*

#### Descripción:

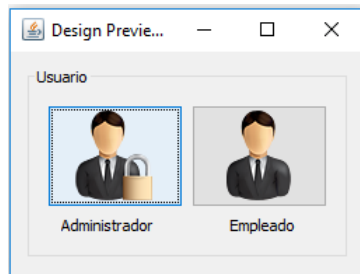
El prototipo de la segunda ventana, nombrada “Ventana Principal”, se la realiza con la finalidad de presentar las opciones que el sistema implementa; para el acceso a cada funcionalidad se coloca un botón de acción, que permite acceder a la ventana respectiva de cada función; se implementa una barra de menú el cual también permitirá el acceso a las funciones.

El diseño de la interfaz, al ser una ventana intermedia para la elección de las funcionalidades, se la desarrolla sencilla y fácil de comprender para el usuario.

- Se presentan 5 botones de acción:
  - Administrar Clientes.
  - Administrar Proveedores.
  - Administrar Productos.
  - Administrar Ventas.
  - Administrar Compras.

## Diseño final

### - Ventana Inicio



*Figura 3-8. GUI Inicio (Zurita U, 2017).*

### - Ventana Principal



*Figura 3-9. GUI Principal (Zurita U, 2017).*

## 3.3.3. Pruebas

### Pruebas Caja Blanca – Unitarias

Las pruebas unitarias se las realiza en Netbeans y su complemento Junit.

Se realiza la prueba a los métodos pertenecientes a Ventana Inicio, UsuarioDP, UsuarioMD.

## Ventana Inicio

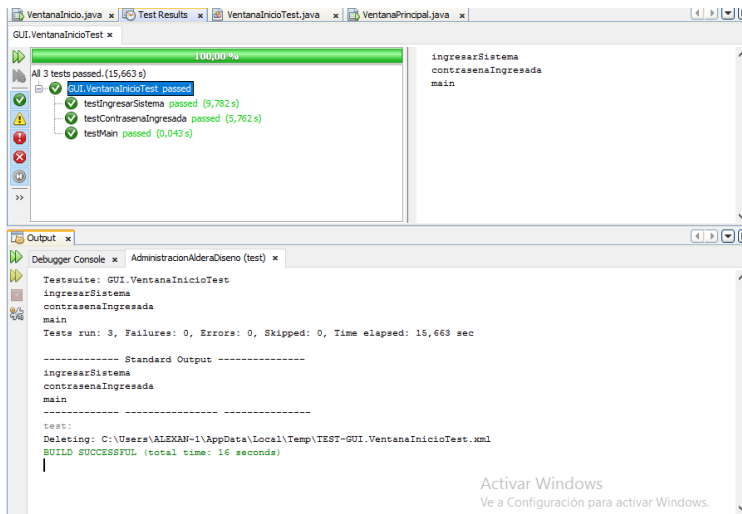


Figura 3-10. Prueba Unitaria Ventana Inicio (Zurita U, 2017).

## UsuarioDP

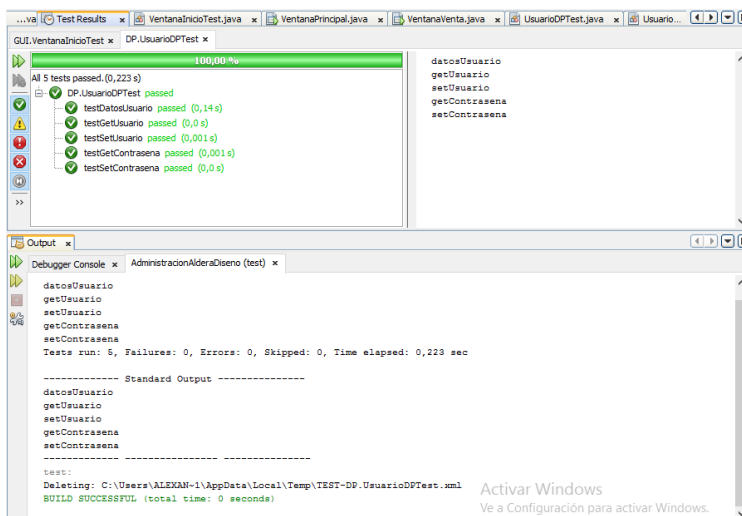


Figura 3-11. Prueba Unitaria UsuarioDP (Zurita U, 2017).

## UsuarioMD

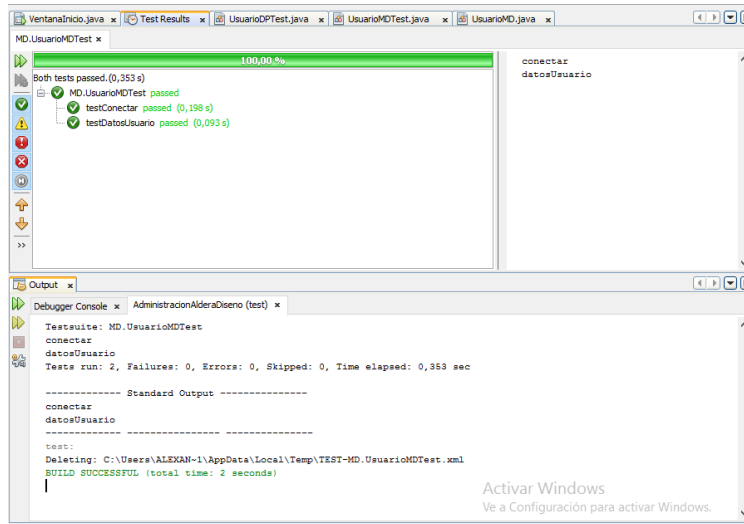


Figura 3-12. Prueba Unitaria UsuarioMD (Zurita U, 2017).

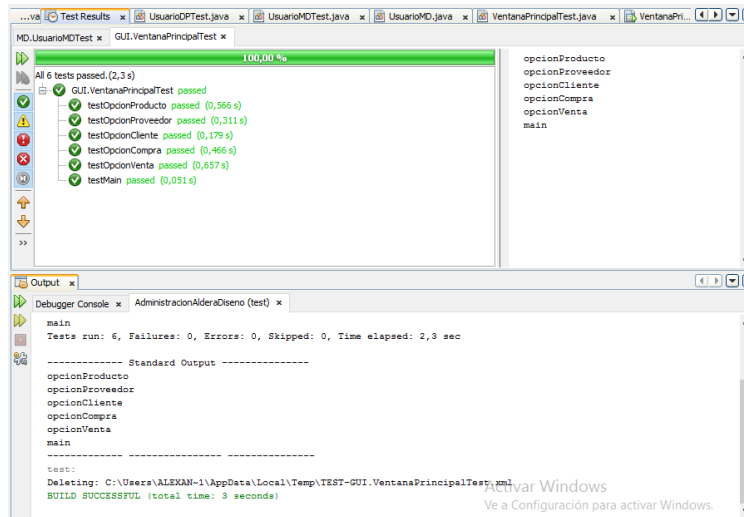


Figura 3-13. Prueba Unitaria Ventana Principal (Zurita U, 2017).

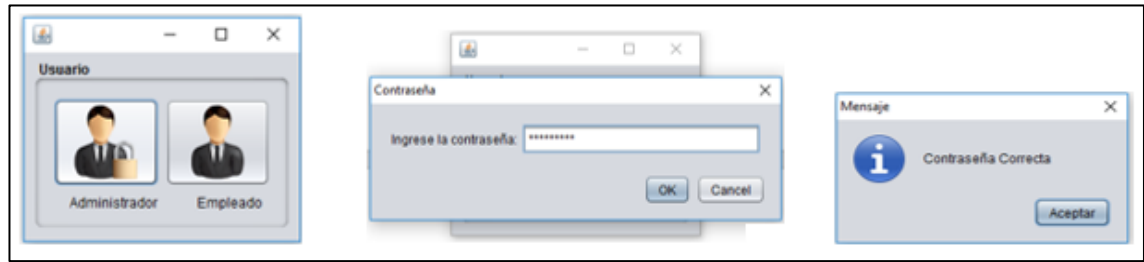
## Pruebas Caja Negra - Ingreso

Ingreso al sistema

Datos de prueba:

Usuario = Administrador

Contraseña = admin1234



*Figura 3-14. Prueba Caja Negra Ingreso (Zurita U, 2017).*

### Prueba de aceptación

PRUEBA DE ACEPTACIÓN	
Código: P1	Historia de Usuario: 1
Nombre: Ingreso al sistema	
Descripción: Prueba para la funcionalidad de ingreso al sistema	
Condiciones de ejecución: Para efectuar la prueba el sistema dará la opción de ingreso como administrador o como empleado en la pantalla de inicio. Cada uno contará con una contraseña diferente.	
Entrada/Pasos de ejecución: Una vez que se tiene las opciones de entrada, el administrador ingresará dando clic en el botón izquierdo de la pantalla donde se muestra su perfil, el empleado ingresará dando clic en el botón derecho donde se muestra su perfil.	
Resultado esperado: Los perfiles de administrador y empleado deben ingresar al sistema sin inconvenientes y con el nivel de acceso correspondiente de cada uno.	
Evaluación de la prueba: Prueba exitosa.	

*Tabla 3-1. Prueba de Aceptación 1 (Zurita U, 2017).*

### 3.4. ITERACIÓN 2 – Administrar Clientes

En esta iteración realizaremos el desarrollo e implementación de la funcionalidad F1. Administrar Cliente.

#### 3.4.1. Diseño

La clase necesaria para la implementación de esta funcionalidad es Cliente.

## DIAGRAMA DE CLASES

### Interfaz Gráfica (GUI)

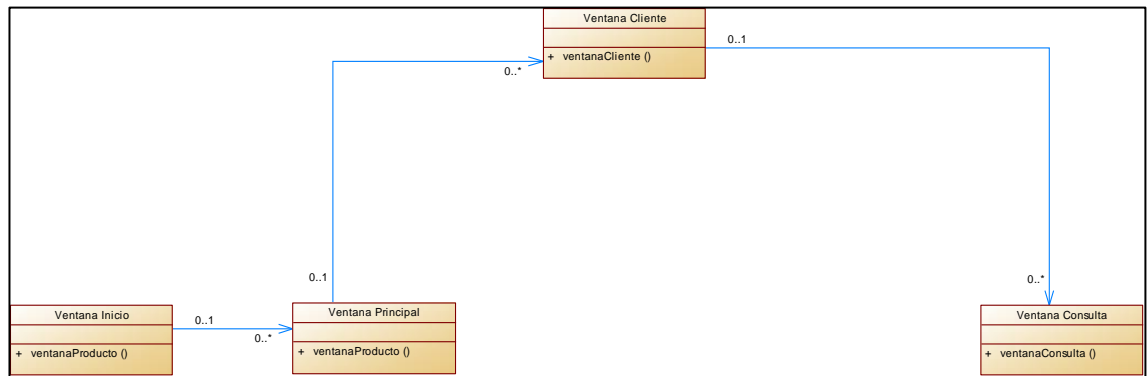


Figura 3-15. Diagrama de clases GUI Iteración 2 (Zurita U, 2017).

### Dominio del Problema (DP)

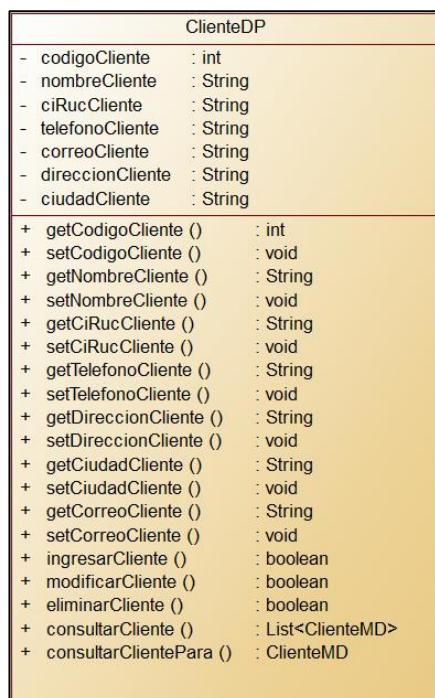


Figura 3-16. Diagrama de clases DP Iteración 2 (Zurita U, 2017).

## Manejo de Datos (MD)



Figura 3-17. Diagrama de clases MD Iteración 2 (Zurita U, 2017).

### 3.4.2. Desarrollo

#### Prototipo de GUI – Administrar Clientes

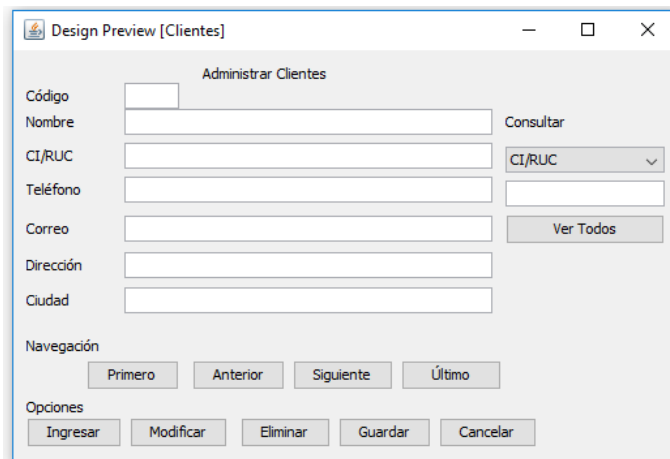


Figura 3-18. GUI Prototipo Clientes (Zurita U, 2017).

#### Descripción:

El prototipo de la tercera ventana, nombrada “Ventana Clientes”, se la realiza en base a las necesidades del usuario, cumpliendo con la Historia de Usuario 2.

El diseño de la interfaz es sencillo y con lo necesario para la administración de los registros de Clientes.

La ventana cuenta con 4 secciones:

- Datos  
Se proporciona campos para los diferentes datos de un cliente.

- Consultar

En esta sección se presenta la opción de presentar todos los registros o la búsqueda por parámetro.

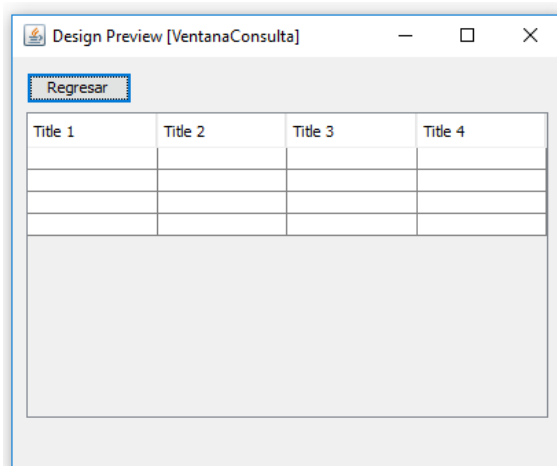
- Navegación

Su funcionalidad es permitir moverse entre registros, facilitando la presentación de la información.

- Opciones

En esta sección se presenta las acciones que se puede realizar para la administración de los registros de clientes; funciones para Ingresar, Modificar, Eliminar, Guardar y Cancelar.

### Prototipo de GUI – Consulta



*Figura 3-19. GUI Prototipo Consulta (Zurita U, 2017).*

#### **Descripción:**

El prototipo de la cuarta ventana, nombrada “Ventana Consulta”, se la realiza en base a las necesidades del usuario para presentar los registros almacenados en la base de datos.

El diseño de la interfaz es sencillo, se presenta una tabla la cual se modificará de acuerdo con los datos que se le envíe.

Esta ventana será utilizada por todas las funcionalidades del sistema, por lo tanto, no tiene un diseño definido.

## Diseño Final

### Ventana Cliente

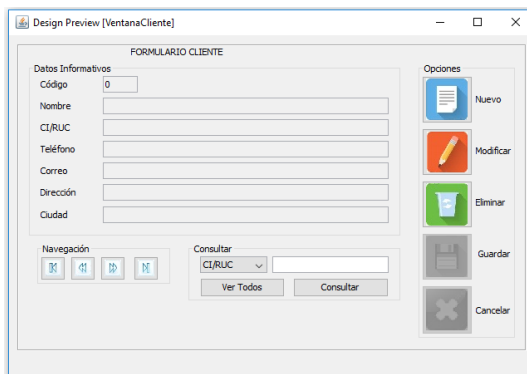


Figura 3-20. GUI Cliente (Zurita U, 2017).

### Ventana Consulta

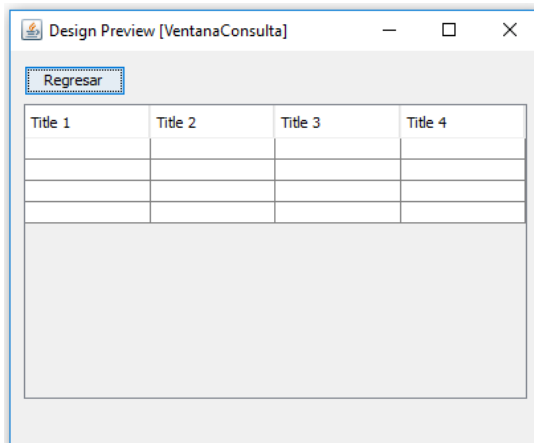


Figura 3-21. GUI Consulta (Zurita U, 2017).

## 3.4.3. Pruebas

### Pruebas Caja Blanca – Unitarias

Las pruebas unitarias se las realiza en Netbeans y su complemento Junit.

## Cliente

Se realiza la prueba a los métodos pertenecientes a Ventana Cliente, ClienteDP, ClienteMD.

## Ventana Cliente

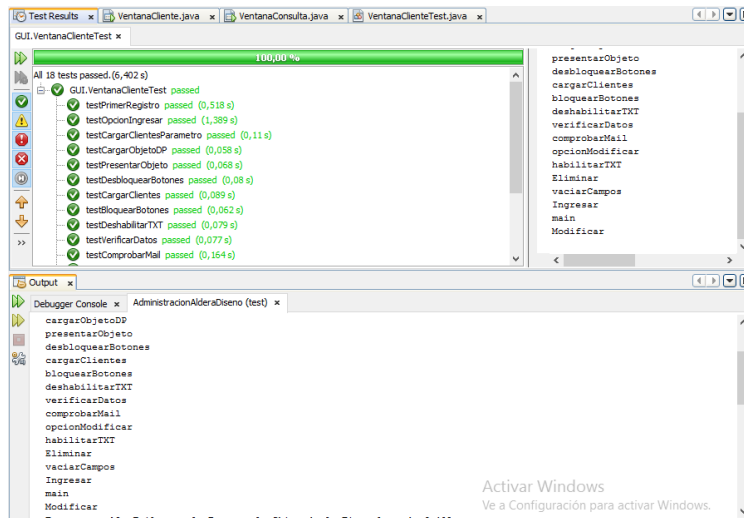


Figura 3-22. Prueba Unitaria Ventana Cliente (Zurita U, 2017).

## ClienteDP

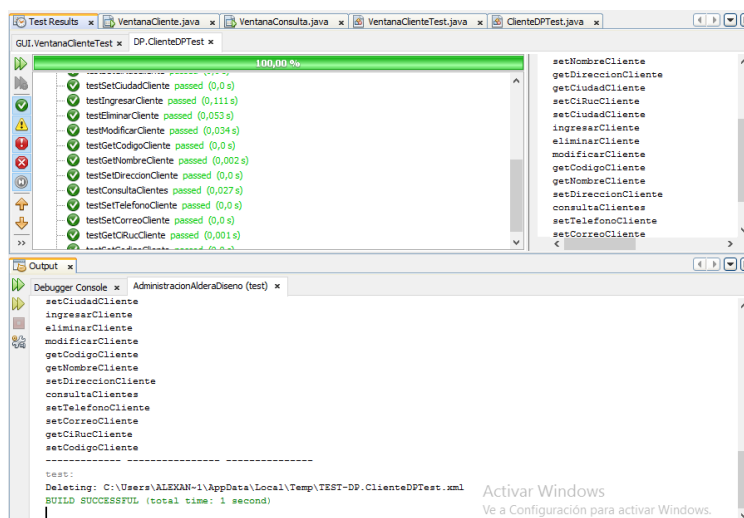


Figura 3-23. Prueba Unitaria ClienteDP (Zurita U, 2017).

## ClienteMD

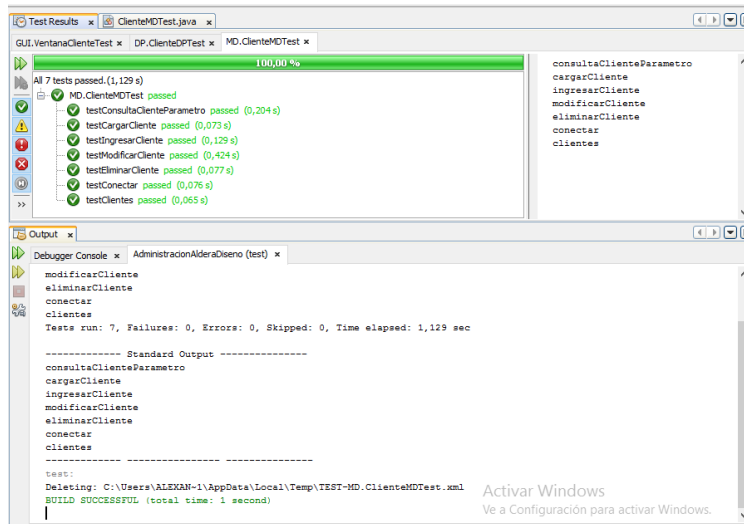


Figura 3-24. Prueba Unitaria ClienteMD (Zurita U, 2017).

## Pruebas Caja Negra

### Ingresar Cliente

Datos de prueba:

- Nombre: Esteban Romero
- CI: 1714722335
- Teléfono:0995105660
- Correo: erom611@hotmail.com
- Dirección: Vía Amaguaña
- Ciudad: Quito

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

The screenshot shows a window titled 'VENTANA CLIENTE' containing a 'FORMULARIO CLIENTE'. The form is divided into two main sections: 'Datos Informativos' and 'Opciones'.  
**Datos Informativos:** Fields for Código (23), Nombre (Esteban Romero), CI/RUC (1714722335), Teléfono (0995105660), Correo (erom611@hotmail.com), Dirección (Vía Amaguaña), and Ciudad (Quito).  
**Opciones:** Buttons for 'Nuevo', 'Modificar', 'Eliminar', 'Guardar', and 'Cancelar'.  
**Navegación:** Four small square buttons.  
**Consultar:** A dropdown menu for 'CI/RUC', a text input field, and buttons for 'Ver Todos' and 'Consultar'.

Figura 3-25. Prueba Caja Negra Ingreso Cliente Datos (Zurita U, 2017).

This screenshot is identical to the previous one, but with a confirmation dialog box overlaid in the center. The dialog box has a question mark icon and the text: '¿Esta seguro que desea guardar el nuevo registro?'. It includes 'Sí' and 'No' buttons.

Figura 3-26. Prueba Caja Negra Ingreso Cliente Confirmar (Zurita U, 2017).

This screenshot is identical to the previous ones, but with an information dialog box overlaid. The dialog box has an information icon and the text: 'Operación realizada correctamente'. It includes an 'Aceptar' button.

Figura 3-27. Prueba Caja Negra Ingreso Cliente Respuesta (Zurita U, 2017).

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

codigoCliente [PK] integer	nombreCliente character varying (70)	ciRucCliente character varying (13)	telefonoCliente character varying (15)	correoCliente character varying (70)	direccionCliente character varying (70)	ciudadCliente character vary
1	RRDC Constructores	177323581001	022449792	contacto@rrdmail.com	Beck Rollo OE7-96 y AL...	Quito
2	prueba	172238559600	null	zuritaalexander@hot...	Orquideas Bolivar Rod...	Quito
3	Pablo Israel Zurita Ulloa	1722385588001	0983246743	pablozurita@hotmail.es	Monjas	Guayaquil
4	ingreso	1818177171888	0982525252525252	zur2@otmail.com	Orquideasaaasd	Guayaquil
5	Raul Edgar Zurita	1707141642001	206216302	zuritaraul@hotmail.es	Monjas	Quito
6	Hilda Piedad Ulloa Pur...	0200864759	0992915227	hildaull1961@hotmail.es	Guanujo	Guaranda
7	Esteban Romero	1714722335	0995105660	erom611@hotmail.com	Via Amaguaña	Quito

Figura 3-28. Prueba Caja Negra Ingreso Cliente BDD (Zurita U, 2017).

### Modificar Cliente

Datos de prueba:

- Registro Código: 23.

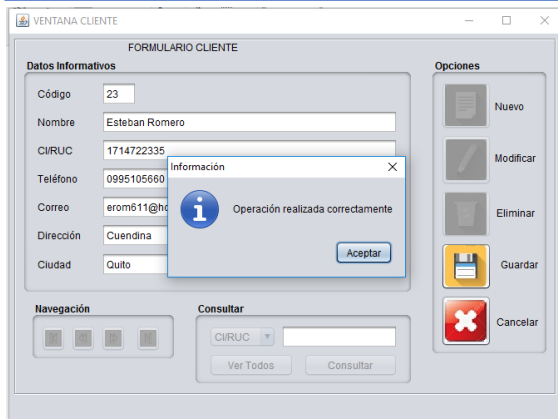
Dato por modificar:

- Dirección: Cuendina.

Figura 3-29. Prueba Caja Negra Modificar Cliente Datos (Zurita U, 2017).

Figura 3-30. Prueba Caja Negra Modificar Cliente Confirmar (Zurita U, 2017).

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño



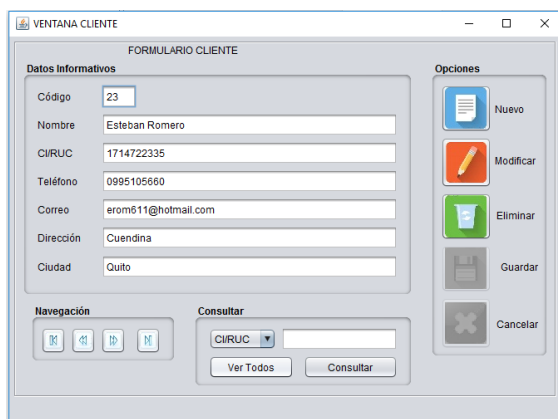
**Figura 3-31. Prueba Caja Negra Modificar Cliente Datos (Zurita U, 2017).**

	codigoCliente [PK] integer	nombreCliente character varying (70)	ciRucCliente character varying (13)	telefonoCliente character varying (15)	correoCliente character varying (70)	direccionCliente character varying (70)	ciudadCliente character varyin
1	1	RRDC Constructores	177323581001	022449792	contacto@rrdmail.com	Beck Rollo OE7-96 y Al...	Quito
2	2	prueba	172238559600	null	zuritaalexander@hot...	Orquideas Bolivar Rod...	Quito
3	4	Pablo Israel Zurita Ulloa	1722385588001	0983246743	pablozurita@hotmail.es	Monjas	Guayaquil
4	5	ingreso	1818177171888	09825252525252	zur2@otmail.com	Orqueidesaaaasd	Guayaquil
5	6	Raul Edgar Zurita	1707141642001	206216302	zuritaraul@hotmail.es	Monjas	Quito
6	7	Hilda Piedad Ulloa Pur...	0200864759	0992915227	hildaull1961@hotmail.es	Guanujo	Guaranda
7	23	Esteban Romero	1714722335	0995105660	erom611@hotmail.com	Cuendina	Quito

**Figura 3-32. Prueba Caja Negra Modificar Cliente BDD (Zurita U, 2017).**

### Consulta General

- Presionando el botón “Ver Todos”



**Figura 3-33. Prueba Caja Negra Consulta General (Zurita U, 2017).**

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

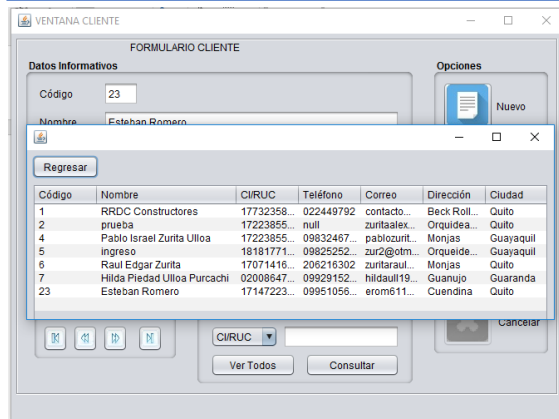


Figura 3-34. Prueba Caja Negra Consulta General Respuesta (Zurita U, 2017).

### Consulta por parámetro

Datos de prueba:

Parámetro: "CI/RUC"

Valor: 1714722335

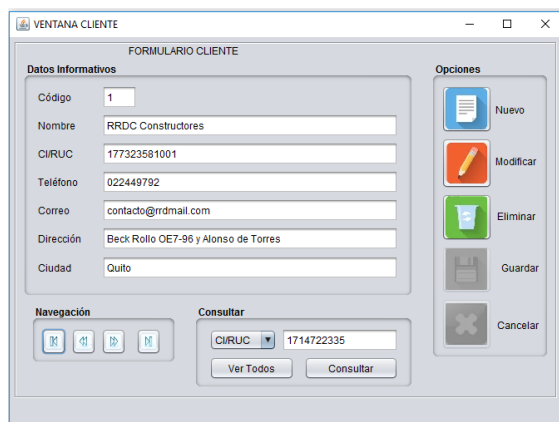


Figura 3-35. Prueba Caja Negra Consultar Cliente por parámetro (Zurita U, 2017).

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

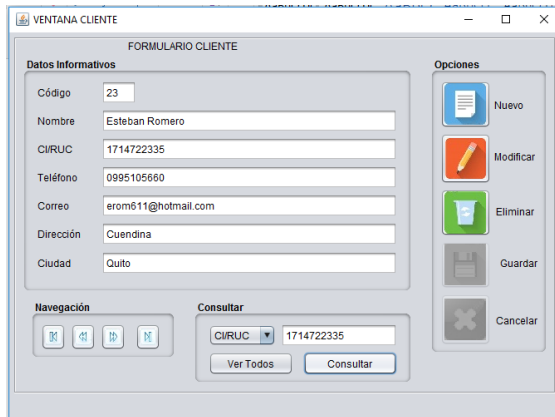


Figura 3-36. Prueba Caja Negra Consultar Cliente por parámetro respuesta (Zurita U, 2017).

## Eliminar Cliente

### Datos de Prueba

- Código: 23

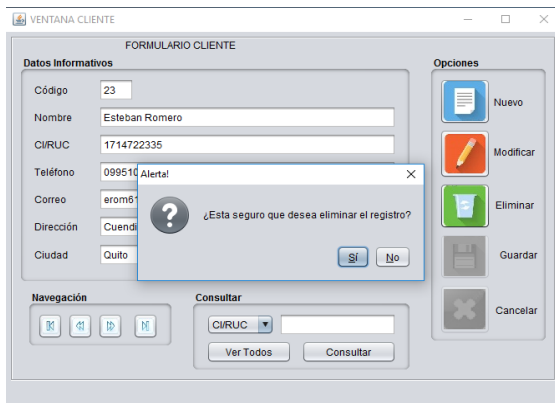


Figura 3-37. Prueba Caja Negra Eliminar Cliente (Zurita U, 2017).

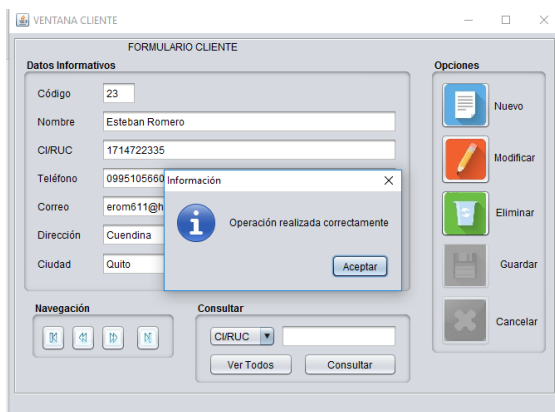


Figura 3-38. Prueba Caja Negra Eliminar respuesta (Zurita U, 2017).

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

	codigoCliente [PK] integer	nombreCliente character varying (70)	ciRucCliente character varying (13)	telefonoCliente character varying (15)	correoCliente character varying (70)	direccionCliente character varying (70)	ciudadCliente character varyin
1	1	RRDC Constructores	177323581001	022449792	contacto@rrdmail.com	Beck Rollo OE7-96 y AL...	Quito
2	2	prueba	172238559600	null	zuritaalexander@hot...	Orquídeas Bolívar Rod...	Quito
3	4	Pablo Israel Zurita Ulloa	1722385588001	0983246743	pablozurita@hotmail.es	Monjas	Guayaquil
4	5	ingreso	1818177171888	09825252525252	zur2@otmail.com	Orqueidesaaasd	Guayaquil
5	6	Raul Edgar Zurita	1707141642001	206216302	zuritaraul@hotmail.es	Monjas	Quito
6	7	Hilda Piedad Ulloa Pur...	0200864759	0992915227	hildaull1961@hotmail.es	Guanujo	Guaranda

**Figura 3-39. Prueba Caja Negra Eliminar BDD (Zurita U, 2017).**

## Prueba de aceptación

PRUEBA DE ACEPTACIÓN	
<b>Código:</b> P2	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Prueba de gestión de clientes.	
<b>Descripción:</b> Se realizará la prueba de ingreso, modificación, consultas y eliminación de perfiles de proveedores.	
<b>Condiciones de ejecución:</b> Para efectuar la prueba el administrador o el empleado deberán entrar al sistema.	
<b>Entrada/Pasos de ejecución:</b> Después que el usuario (administrador o empleado) ha entrado al sistema, se abre la siguiente pantalla. En este caso para la administración de clientes debe dar clic en el botón CLIENTES, esto lo llevará a la pantalla donde podrá ingresar datos en el formulario, podrá manejarlos por medio de los botones de OPCIONES y podrá moverse por los registros por medio de los botones de NAVEGACION, si se requiere hacer consultas podrá hacerlo por medio del menú CONSULTAR, y si lo requiere, establecer un	
<b>Resultado esperado:</b> Se guardan los datos de los clientes, se puede acceder a ellos por medio de consulta y se los puede modificar. En caso de requerirlo, los datos del cliente serán borrados.	
<b>Evaluación de la prueba:</b> Prueba exitosa.	

**Tabla 3-2. Prueba de Aceptación 2 (Zurita U, 2017).**

## 3.5. ITERACIÓN 3 – Administrar Proveedores

En esta iteración realizaremos el desarrollo e implementación de la funcionalidad F2. Administrar Proveedores.

### 3.5.1. Diseño

En esta etapa, para continuar con la implementación necesitaremos de las clases Cliente y Proveedor.

## DIAGRAMA DE CLASES

### Interfaz Gráfica (GUI)

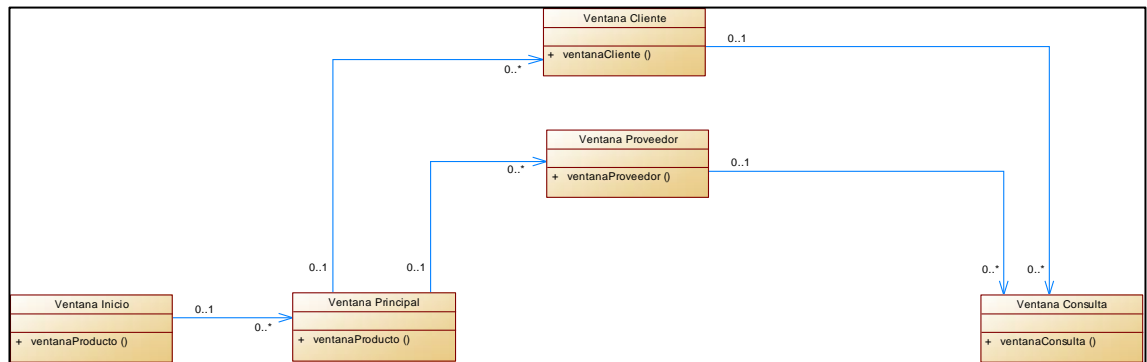


Figura 3-40. Diagrama de clases GUI Iteración 3 (Zurita U, 2017).

### Dominio del Problema (DP)

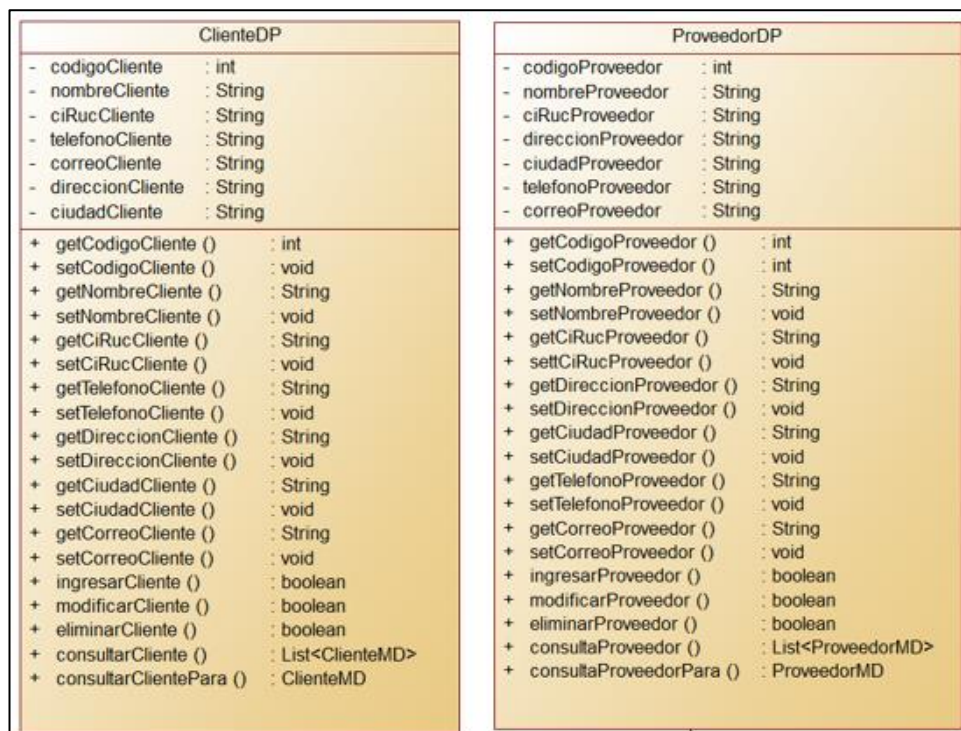


Figura 3-41. Diagrama de clases DP Iteración 3 (Zurita U, 2017).

## Manejo de Datos (MD)

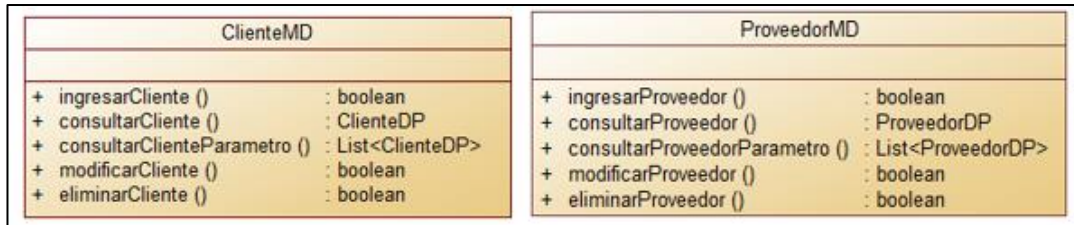


Figura 3-42. Diagrama de clases MD Iteración 3 (Zurita U, 2017).

### 3.5.2. Desarrollo

#### Prototipo de GUI – Administrar Proveedores

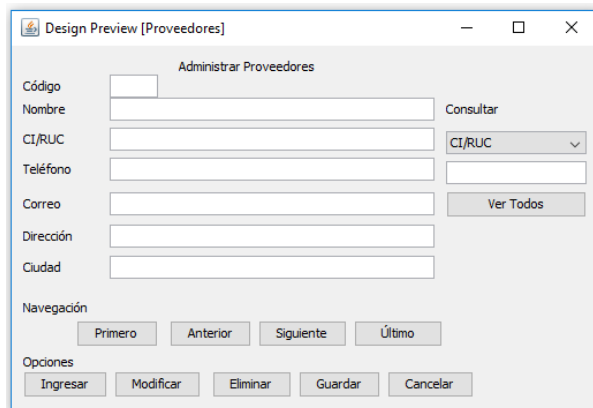


Figura 3-43. GUI Prototipo Proveedores (Zurita U, 2017).

#### Descripción:

El prototipo de la quinta ventana, nombrada “Ventana Proveedores”, se la realiza en base a las necesidades del usuario establecidas previamente, cumpliendo con la Historia de Usuario 3.

El diseño de la interfaz es sencillo y con lo necesario para la administración de los registros de Proveedores.

La ventana cuenta con 4 secciones:

- Datos

Se proporciona campos para los diferentes datos de un proveedor.

- Consultar

En esta sección se presenta la opción de presentar todos los registros o la búsqueda por parámetro.

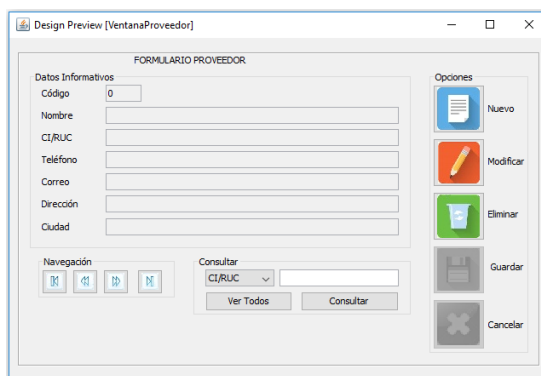
- Navegación

Su funcionalidad es permitir moverse entre registros, facilitando la presentación de la información.

- Opciones

En esta sección se presenta las acciones que se puede realizar para la administración de los registros de proveedores; funciones para Ingresar, Modificar, Eliminar, Guardar y Cancelar.

## Diseño Final



*Figura 3-44. GUI Proveedores (Zurita U, 2017).*

### 3.5.3. Pruebas

#### Pruebas Caja Blanca – Unitarias

Las pruebas unitarias se las realiza en Netbeans y su complemento Junit.

#### Proveedor

Se realiza la prueba a los métodos pertenecientes a Ventana Proveedor, ProveedorDP, ProveedorMD.

## Ventana Proveedor

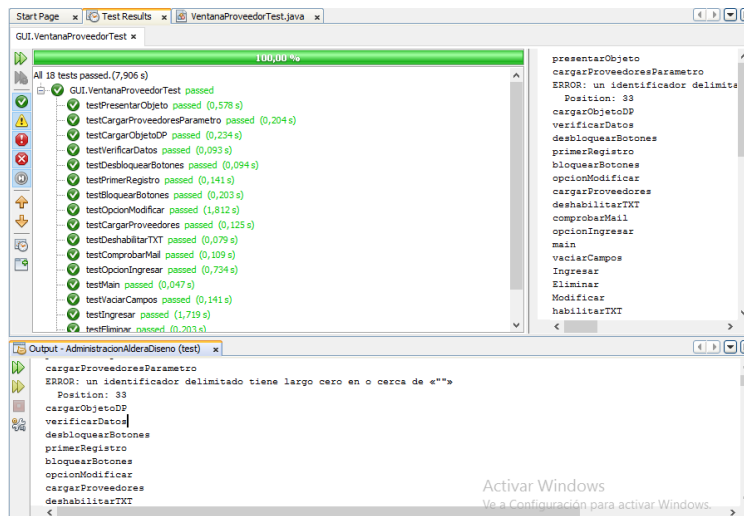


Figura 3-45. Prueba Unitaria Ventana Proveedor (Zurita U, 2017).

## ProveedorDP

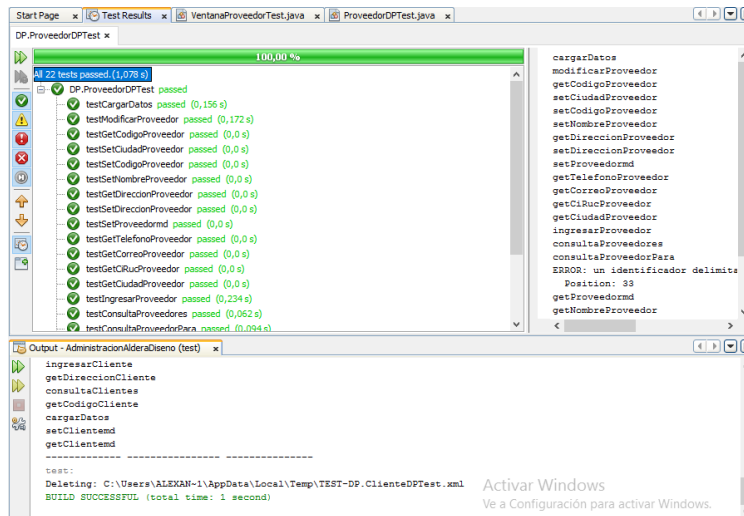


Figura 3-46. Prueba Unitaria ProveedorDP (Zurita U, 2017).

## ProveedorMD

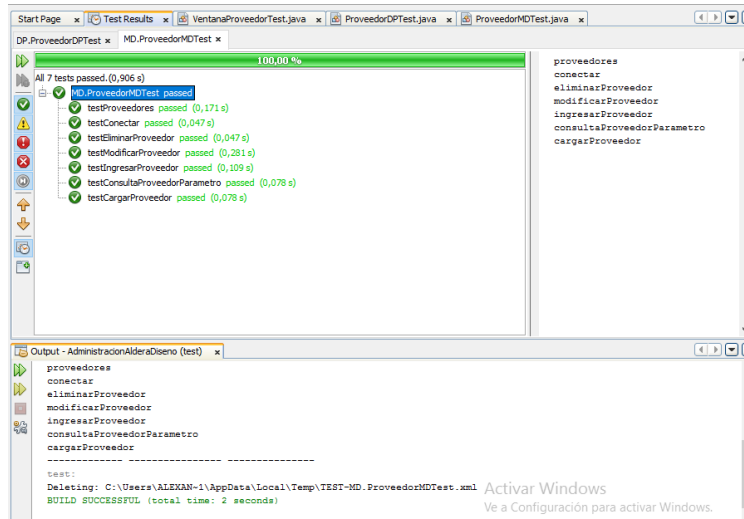


Figura 3-47. Prueba Unitaria ProveedorMD (Zurita U, 2017).

## Pruebas Caja Negra

### Ingresar Proveedor

Datos de prueba:

- Nombre: Soarsi S.A
- CI: 0200864502001
- Teléfono: 022602161
- Correo: [soarsi@hotmail.com](mailto:soarsi@hotmail.com)
- Dirección: Conocoto
- Ciudad: Quito

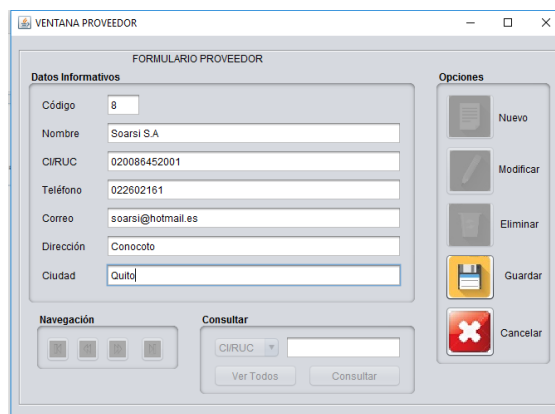


Figura 3-48. Prueba Caja Negra Ingreso Proveedor Datos (Zurita U, 2017).

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

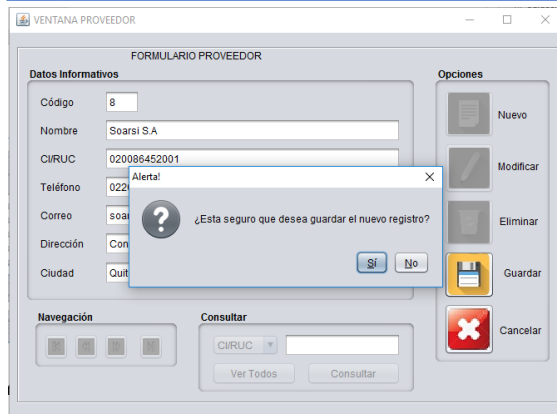


Figura 3-49. Prueba Caja Negra Ingreso Proveedor Confirmar (Zurita U, 2017).

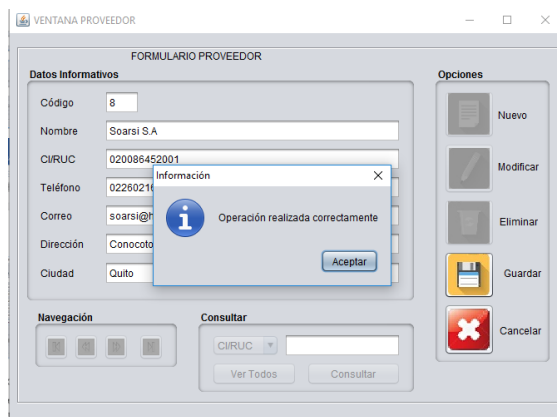


Figura 3-50. Prueba Caja Negra Ingreso Proveedor Respuesta (Zurita U, 2017).

codigoProveedor [PK] integer	nombreProveedor character varying (70)	ciRucProveedor character varying (15)	telefonoProveedor character varying (15)	correoProveedor character varying (70)	direccionProveedor character varying (70)	ciudadProv character va
1	1 IMPLEMENTOS AZ	1254125458001	022362100	iaz@prueba.com	10 de Agosto	Quito
2	2 Gimbel	0982780342	null	pedidos_gimbel@gmai...	Prolongación Moliere 4...	Guayaquil
3	3 Herramar	020086475001	0992915227	pedidos@herramar.com	Jujuy 2365 Mar de Plata	Cuenca
4	4 IMPLEMENTOS AZ	1254125458001	022362100	iaz@prueba.com	10 de Agosto	Quito
5	5 IMPLEMENTOS AZ	1254125458001	022362100	iaz@prueba.com	10 de Agosto	Quito
6	7 Raul Edgar Zurita Garr...	1707141642	026022163	raulzurita@hotmail.es	Monjas Calle A1253	Quito
7	8 Soarsi S.A	020086452001	022602161	soarsi@hotmail.es	Conocoto	Quito

Figura 3-51. Prueba Caja Negra Ingreso Proveedor BDD Datos (Zurita U, 2017).

## Modificar Proveedor

Datos de prueba:

- Registro Código: 8
- Dato por modificar: Ciudad
- Valor anterior: Quito
- Valor actual: Guayaquil

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

The screenshot shows a web application window titled 'VENTANA PROVEEDOR'. Inside, there is a form titled 'FORMULARIO PROVEEDOR' with two main sections: 'Datos Informativos' and 'Opciones'. The 'Datos Informativos' section contains several input fields: 'Código' (8), 'Nombre' (Soarsi S.A), 'CI/RUC' (020086452001), 'Teléfono' (022602161), 'Correo' (soarsi@hotmail.es), 'Dirección' (Conocoto), and 'Ciudad' (Guayaquil). The 'Opciones' section contains five buttons: 'Nuevo', 'Modificar' (highlighted with a red border), 'Eliminar', 'Guardar', and 'Cancelar'. At the bottom, there is a 'Navegación' section with four small square buttons and a 'Consultar' section with a dropdown menu for 'CI/RUC', a search input field, and 'Ver Todos' and 'Consultar' buttons.

Figura 3-52. Prueba Caja Negra Modificar Proveedor Datos (Zurita U, 2017).

This screenshot is similar to the previous one, but with a confirmation dialog box overlaid in the center. The dialog box has a question mark icon and the text 'Alerta! ¿Esta seguro que desea Modificar el registro?'. It has two buttons: 'Si' and 'No'.

Figura 3-53. Prueba Caja Negra Modificar Proveedor Confirmar (Zurita U, 2017).

This screenshot is similar to the previous ones, but with an information dialog box overlaid in the center. The dialog box has an information icon and the text 'Información Operación realizada correctamente'. It has one button: 'Aceptar'.

Figura 3-54. Prueba Caja Negra Modificar Proveedor Respuesta (Zurita U, 2017).

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

codigoProveedor [PK] integer	nombreProveedor character varying (70)	ciRucProveedor character varying (15)	telefonoProveedor character varying (15)	correoProveedor character varying (70)	direccionProveedor character varying (70)	ciudadProveedor character va
1	IMPLEMENTOS AZ	1254125458001	022362100	iaz@prueba.com	10 de Agosto	Quito
2	Gimbel	0982780342	null	pedidos_gimbel@gmai...	Prolongación Molere 4...	Guayaquil
3	Herramar	020086475001	0992915227	pedidos@herramar.com	Jujuy 2365 Mar de Plata	Cuenca
4	IMPLEMENTOS AZ	1254125458001	022362100	iaz@prueba.com	10 de Agosto	Quito
5	IMPLEMENTOS AZ	1254125458001	022362100	iaz@prueba.com	10 de Agosto	Quito
6	Raul Edgar Zurita Garr...	1707141642	026022163	raulzurita@hotmail.es	Monjas Calle A1253	Quito
7	Soarsi S.A	020086452001	022602161	soarsi@hotmail.es	Conocoto	Guayaquil

Figura 3-55. Prueba Caja Negra Modificar Proveedor BDD (Zurita U, 2017).

## Consulta General

Presionando el botón “Ver Todos”

The screenshot shows a window titled 'VENTANA PROVEEDOR' containing a 'FORMULARIO PROVEEDOR'. Inside the form, there is a 'Datos Informativos' section with a 'Regresar' button and a table listing providers. Below the table are navigation buttons and a 'Ver Todos' button.

Código	Nombre	CI/RUC	Teléfono	Correo	Dirección	Ciudad
1	IMPLEMENTOS AZ	12541254...	022362100	iaz@prue...	10 de Ago...	Quito
2	Gimbel	09827803...	null	pedidos_...	Prolongac...	Guayaquil
3	Herramar	02008647...	09929152...	pedidos_...	Jujuy 236...	Cuenca
4	IMPLEMENTOS AZ	12541254...	022362100	iaz@prue...	10 de Ago...	Quito
5	IMPLEMENTOS AZ	12541254...	022362100	iaz@prue...	10 de Ago...	Quito
7	Raul Edgar Zurita Garrido	17071416...	026022163	raulzurita...	Monjas C...	Quito
8	Soarsi S.A	02008645...	022602161	soarsi@h...	Conocoto	Guayaquil

Figura 3-56. Prueba Caja Negra Consulta General Proveedor (Zurita U, 2017).

## Consulta por parámetro

Datos de prueba:

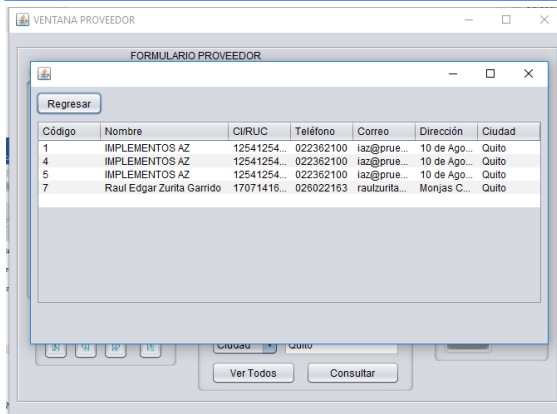
Parámetro: “Ciudad”

Valor: Quito

The screenshot shows the 'FORMULARIO PROVEEDOR' window with search filters. The 'Ciudad' dropdown is set to 'Quito'. There are buttons for 'Nuevo', 'Modificar', 'Eliminar', 'Guardar', and 'Cancelar'. A 'Consultar' button is also present, along with a 'Ver Todos' button.

Figura 3-57. Prueba Caja Negra Consulta Parámetro Proveedor (Zurita U, 2017).

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

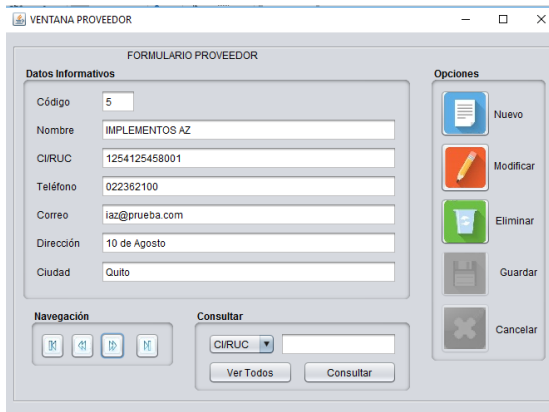


**Figura 3-58. Prueba Caja Negra Consulta Parámetro Proveedor Resultado (Zurita U, 2017).**

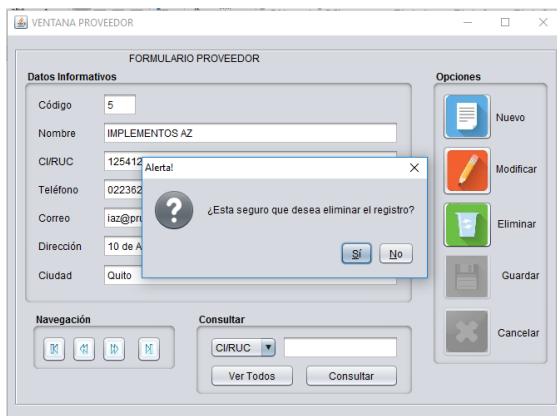
### Eliminar Proveedor

Datos de prueba:

- Código: 5



**Figura 3-59. Prueba Caja Negra Eliminar Proveedor (Zurita U, 2017).**



**Figura 3-60. Prueba Caja Negra Eliminar Proveedor Confirmar (Zurita U, 2017).**

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

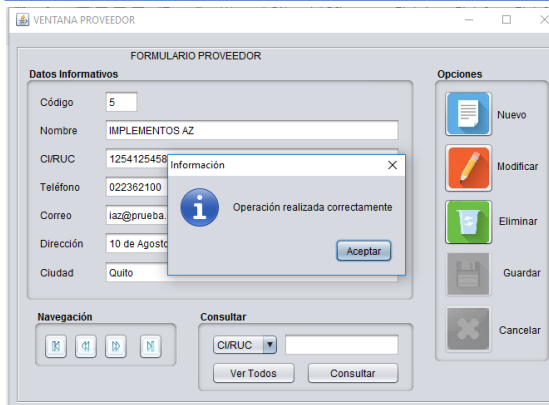


Figura 3-61. Prueba Caja Negra Eliminar Proveedor Resultado (Zurita U, 2017).

codigoProveedor	nombreProveedor	ciRucProveedor	telefonoProveedor	correoProveedor	direccionProveedor	ciudadProveedor
1	IMPLEMENTOS AZ	1254125458001	022362100	iaz@prueba.com	10 de Agosto	Quito
2	Gimbel	0982780342	null	pedidos_gimbel@gmail...	Prolongación Moliere 4...	Guayaquil
3	Herramar	020086475001	0992915227	pedidos@herramar.com	Jujuy 2365 Mar de Plata	Cuenca
4	Raul Edgar Zurita Garr...	1707141642	026022163	raulzurita@hotmail.es	Monjas Calle A1253	Quito
5	Soarsi S.A	020086452001	022602161	soarsi@hotmail.es	Conocoto	Guayaquil

Figura 3-62. Prueba Caja Negra Eliminar Proveedor BDD (Zurita U, 2017).

### Prueba de aceptación

PRUEBA DE ACEPTACIÓN	
Código: P3	Historia de Usuario: 3
<b>Nombre:</b> Prueba de gestión de proveedores.	
<b>Descripción:</b> Se realizará la prueba de ingreso, modificación, consultas y eliminación de proveedores.	
<b>Condiciones de ejecución:</b> Para efectuar la prueba el administrador o el empleado deberán entrar al sistema.	
<b>Entrada/Pasos de ejecución:</b> Después que el usuario (administrador o empleado) ha entrado al sistema, se abre la siguiente pantalla. En este caso para la administración de proveedores debe dar clic en el botón PROVEEDORES, esto lo llevará a la pantalla donde podrá ingresar datos en el formulario, podrá manejarlos por medio de los botones de OPCIONES y podrá moverse por los registros por medio de los botones de NAVEGACION, si se requiere hacer consultas podrá hacerlo por medio del menú CONSULTAR, y si lo requiere,	
<b>Resultado esperado:</b> Se guardan los datos de los proveedores, se puede acceder a ellos por medio de consulta y se los puede modificar. En caso de requerirlo, los datos del proveedor serán borrados.	
<b>Evaluación de la prueba:</b> Prueba exitosa.	

Tabla 3-3. Prueba de Aceptación 3 (Zurita U, 2017).

### 3.6. ITERACIÓN 4 – Administrar Productos

En esta iteración realizaremos el desarrollo e implementación de la funcionalidad F3. Administrar Productos.

### 3.6.1. Diseño

En esta etapa, para continuar con la implementación necesitaremos de las clases Cliente, Productos y Proveedor

## DIAGRAMA DE CLASES

### Interfaz Gráfica (GUI)

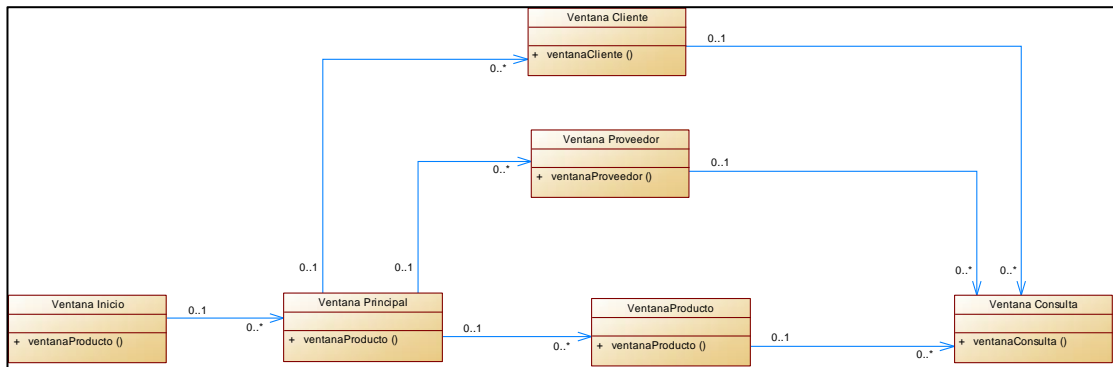


Figura 3-63. Diagrama de Clases GUI Iteración 4 (Zurita U, 2017).

### Dominio del Problema (DP)

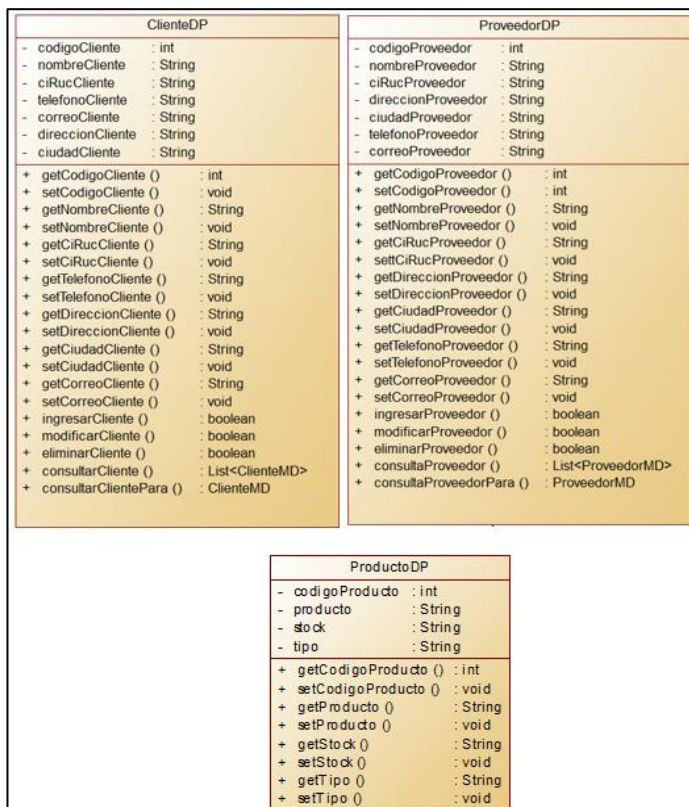


Figura 3-64. Diagrama de Clases DP Iteración 4 (Zurita U, 2017).

## Manejo de Datos (MD)

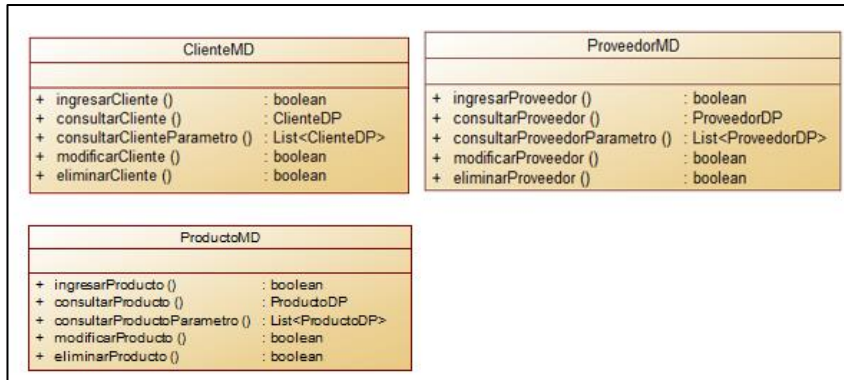


Figura 3-65. Diagrama de Clases MD Iteración 4 (Zurita U, 2017).

### 3.6.2. Desarrollo

#### Prototipo de GUI – Administrar Productos

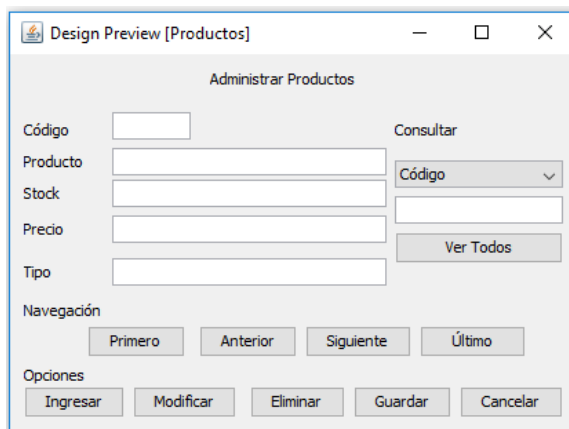


Figura 3-66. GUI Prototipo Productos (Zurita U, 2017).

#### Descripción:

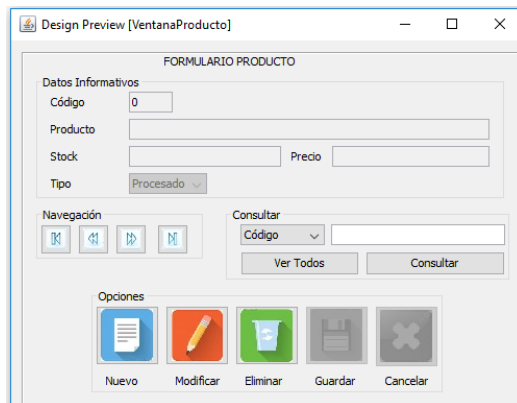
El prototipo de la sexta ventana, nombrada “Ventana Productos”, se la realiza en base a las necesidades del usuario establecidas previamente, cumpliendo con la Historia de Usuario 4.

El diseño de la interfaz es sencillo y con lo necesario para la administración de los registros de Productos.

La ventana cuenta con 4 secciones:

- **Datos**  
Se proporciona campos para los diferentes datos de un producto.
- **Consultar**  
En esta sección se presenta la opción de presentar todos los registros o la búsqueda por parámetro.
- **Navegación**  
Su funcionalidad es permitir moverse entre registros, facilitando la presentación de la información.
- **Opciones**  
En esta sección se presenta las acciones que se puede realizar para la administración de los registros de productos; funciones para Ingresar, Modificar, Eliminar, Guardar y Cancelar.

## Diseño Final



*Figura 3-67. GUI Productos (Zurita U, 2017).*

### 3.6.3. Pruebas

#### Pruebas Caja Blanca – Unitarias

Las pruebas unitarias se las realiza en Netbeans y su complemento Junit.

## Producto

Se realiza la prueba a los métodos pertenecientes a Ventana Producto, ProductoDP, ProductoMD.

## Ventana Producto

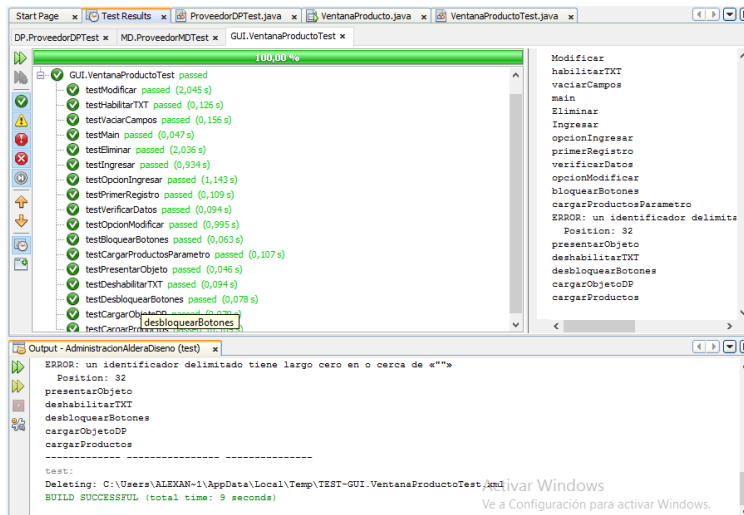


Figura 3-68. Prueba Unitaria Ventana Producto (Zurita U, 2017).

## ProductoDP

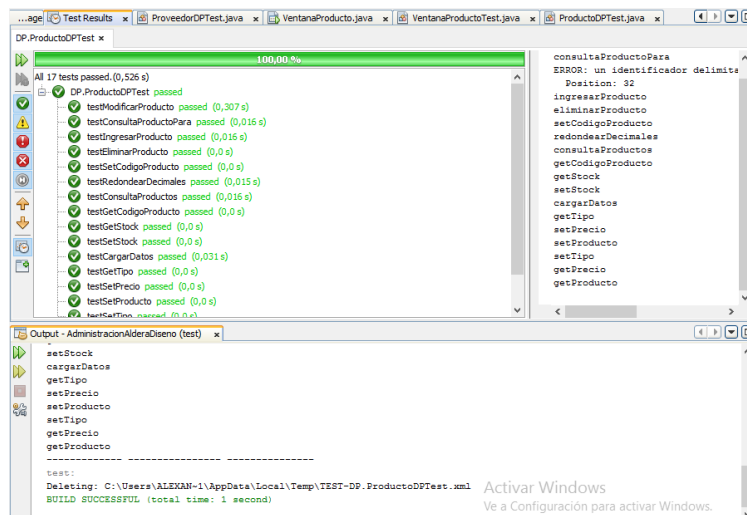


Figura 3-69. Prueba Unitaria Producto DP (Zurita U, 2017).

## ProductoMD

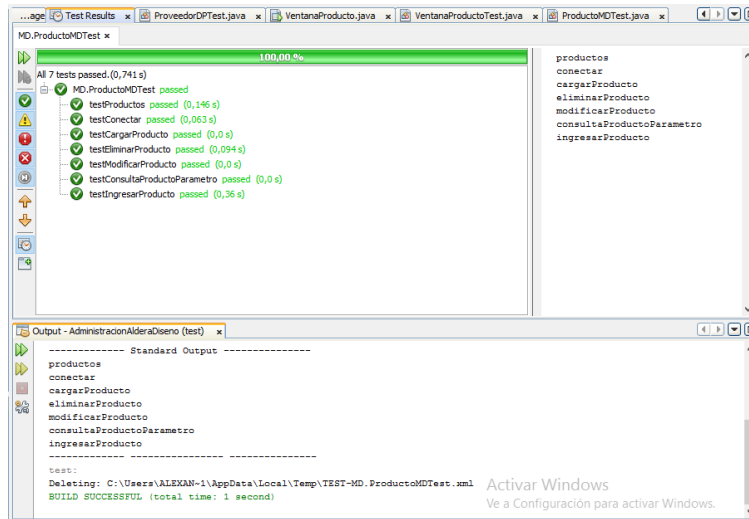


Figura 3-70. Prueba Unitaria Producto MD (Zurita U, 2017).

## Pruebas Caja Negra

### Ingresar Producto

Datos de prueba:

- Producto: Estantería
- Stock: 3
- Precio: 180.5
- Tipo: Procesado

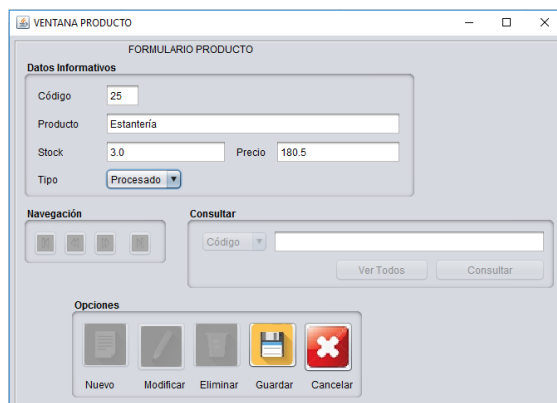


Figura 3-71. Prueba Caja Negra Ingresar Producto Datos (Zurita U, 2017).

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

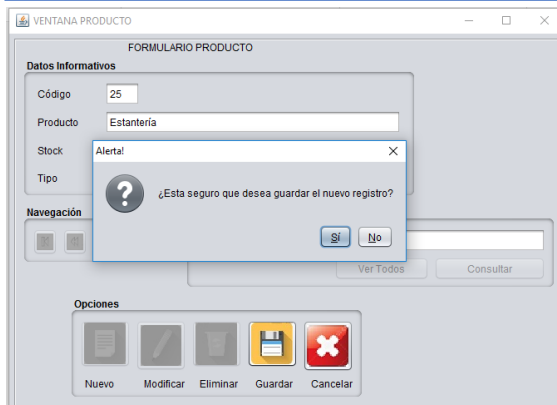


Figura 3-72. Prueba Caja Negra Ingresar Producto Confirmar (Zurita U, 2017).

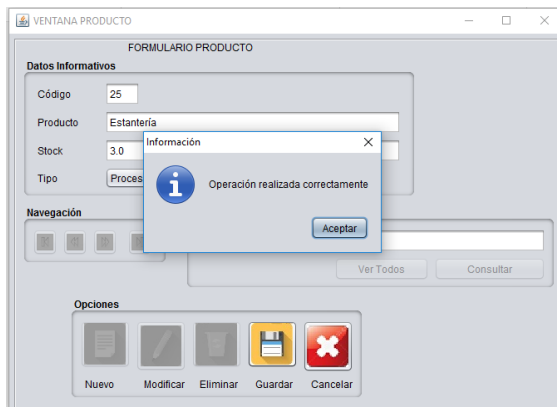


Figura 3-73. Prueba Caja Negra Ingresar Producto Confirmar (Zurita U, 2017).

codigoProducto [PK] integer	producto character (50)	stock double precision	tipo character (10)	precio double precision
1	tablas triplex	89	Prima	2.5
2	laca	12	Prima	5.3
3	Pintura Blanca	20.2	Prima	10
4	Madera Canto	57	Prima	2.63
5	Closet - Canto	233	Procesado	68.52
6	Lacado	729	Procesado	23.65
7	Mueble en mel...	141	Procesado	100.5
8	Soportes Laca...	152	Procesado	265.36
9	Madera Melam...	58.3	Prima	526.36
10	Ropero	520	Procesado	120
11	tablas triplex	225	Prima	2.5
12	madera fina	13.9	Prima	2.98
13	Estantería	3	Procesado	180.5

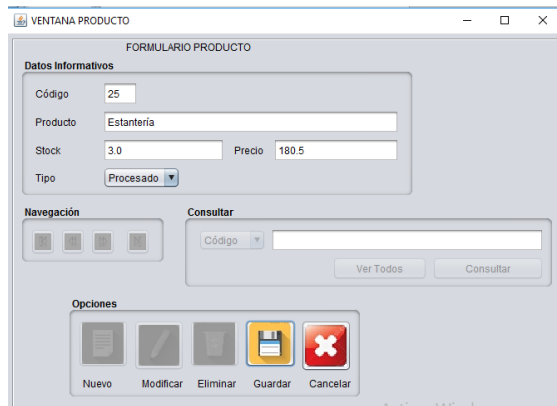
Figura 3-74. Prueba Caja Negra Ingresar Producto BDD (Zurita U, 2017).

### Modificar Procesado

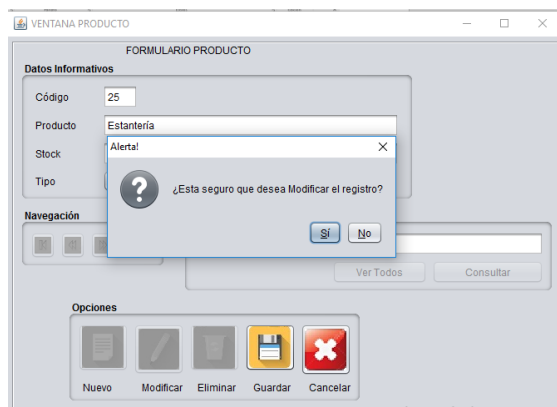
Datos de prueba:

- Código: 25

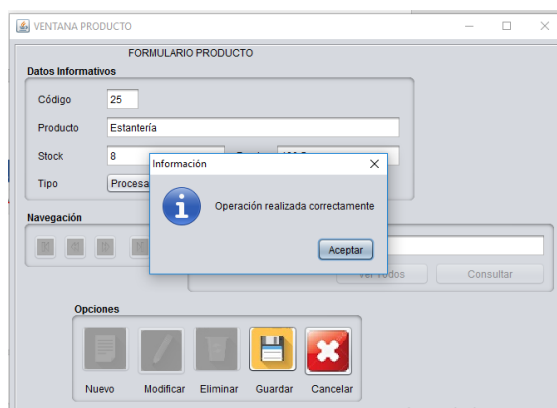
- Campo por cambiar: Stock
- Valor nuevo: 8



**Figura 3-75. Prueba Caja Negra Modificar Producto Datos (Zurita U, 2017).**



**Figura 3-76. Prueba Caja Negra Modificar Producto Confirmar (Zurita U, 2017).**



**Figura 3-77. Prueba Caja Negra Modificar Producto Respuesta (Zurita U, 2017).**

## Consulta General

Presionando el botón “Ver Todos”

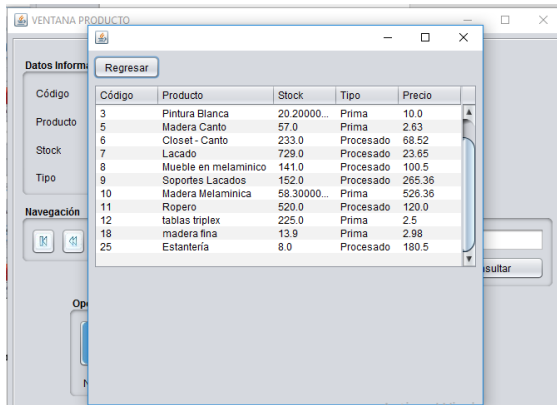


Figura 3-78. Prueba Caja Negra Consulta General Producto (Zurita U, 2017).

## Consulta por parámetro

Datos de prueba:

Parámetro: “Tipo”

Valor: “Procesado”

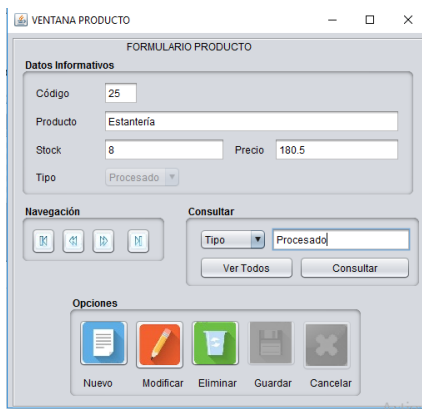


Figura 3-79. Prueba Caja Negra Consulta Parámetro Producto (Zurita U, 2017).

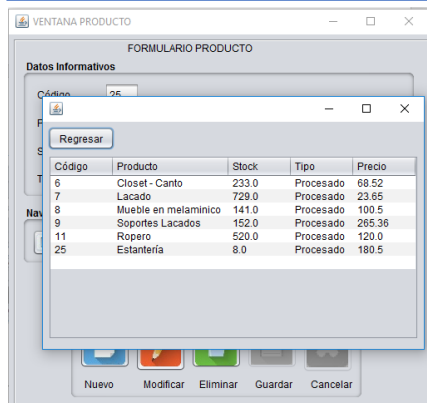


Figura 3-80. Prueba Caja Negra Consulta Parámetro Producto Respuesta (Zurita U, 2017).

## Eliminar Producto

Datos de prueba:

Código: 25.

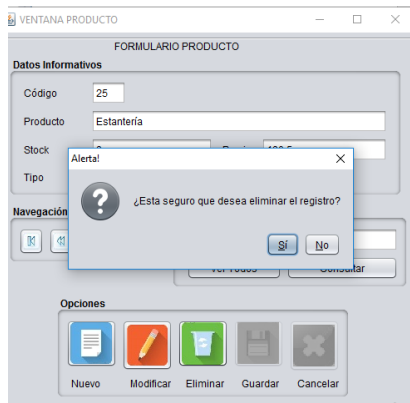


Figura 3-81. Prueba Caja Negra Eliminar Producto Confirmar (Zurita U, 2017).

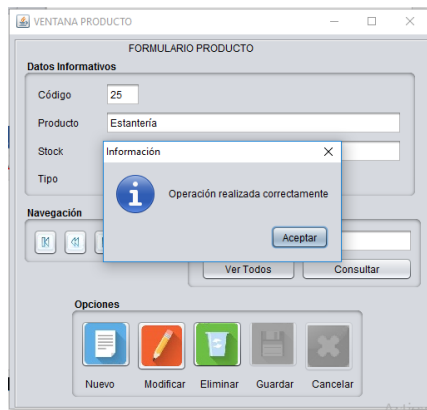


Figura 3-82. Prueba Caja Negra Eliminar Producto Resultado (Zurita U, 2017).

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

Data Output					
	codigoProducto [PK] integer	producto character (50)	stock double precision	tipo character (10)	precio double precision
1	1	tablas triplex	89	Prima	2.5
2	2	laca	12	Prima	5.3
3	3	Pintura Blanca	20.2	Prima	10
4	5	Madera Canto	57	Prima	2.63
5	6	Closet - Canto	233	Procesado	68.52
6	7	Lacado	729	Procesado	23.65
7	8	Mueble en mel...	141	Procesado	100.5
8	9	Soportes Laca...	152	Procesado	265.36
9	10	Madera Melam...	58.3	Prima	526.36
10	11	Ropero	520	Procesado	120
11	12	tablas triplex	225	Prima	2.5
12	18	madera fina	13.9	Prima	2.98

Figura 3-83. Prueba Caja Negra Eliminar Producto BDD (Zurita U, 2017).

### Prueba de aceptación

PRUEBA DE ACEPTACIÓN	
<b>Código:</b> P4	<b>Historia de Usuario:</b> 4
<b>Nombre:</b> Prueba de gestión de productos.	
<b>Descripción:</b> Se realizará la prueba de ingreso, modificación, consultas y eliminación de proveedores.	
<b>Condiciones de ejecución:</b> Para efectuar la prueba el administrador o el empleado deberán entrar al sistema.	
<b>Entrada/Pasos de ejecución:</b> Después que el usuario (administrador o empleado) ha entrado al sistema, se abre la siguiente pantalla. En este caso para la administración de productos, se da clic en el botón PRODUCTOS. El usuario ingresará los DATOS INFORMATIVOS de cada producto en el formulario correspondiente, si quiere navegar entre los registros lo realizará por medio del menú de NAVEGACIÓN y si quiere hacer una consulta lo realizará en la parte de CONSULTAR todos o po medio de un parámetro establecido. Se realizará el manejo de los datos por medio de los botones del menú OPCIONES.	
<b>Resultado esperado:</b> Se guardan los datos de los productos, se puede acceder a ellos por medio de consulta y se los puede modificar. En caso de requerirlo, los productos serán borrados.	
<b>Evaluación de la prueba:</b> Prueba exitosa.	

Tabla 3-4. Prueba de Aceptación 4 (Zurita U, 2017).

### 3.7. ITERACIÓN 5 – Administrar Ventas

En esta iteración realizaremos el desarrollo e implementación de la funcionalidad F4. Administrar Ventas.

#### 3.7.1. Diseño

En esta etapa, para continuar con la implementación necesitaremos de las clases Cliente, Venta, Descripción Venta, Productos y Proveedor.

## DIAGRAMA DE CLASES

### Interfaz Gráfica (GUI)

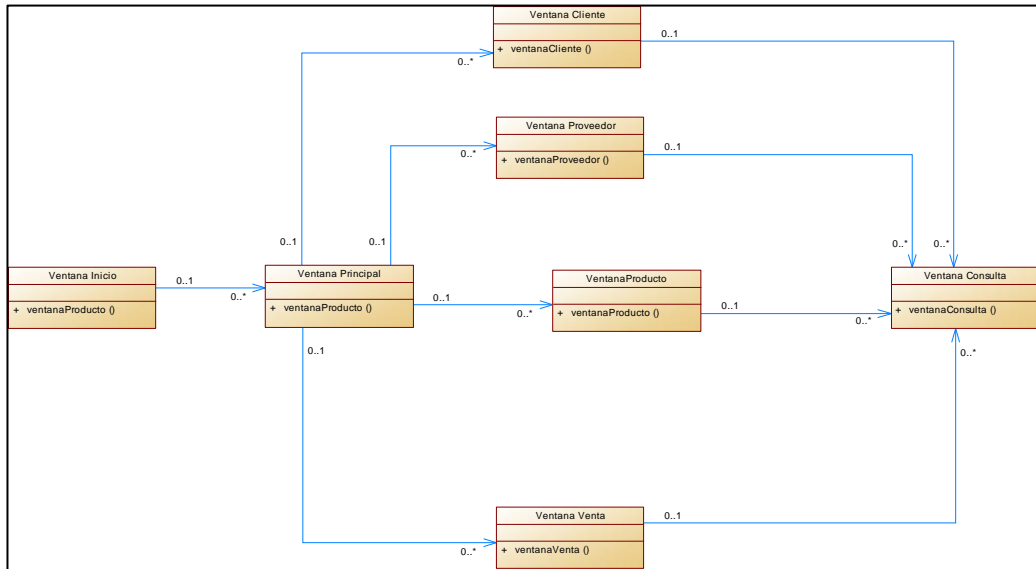


Figura 3-84. Diagrama de Clases GUI Iteración 5 (Zurita U, 2017).

### Dominio del Problema (DP)

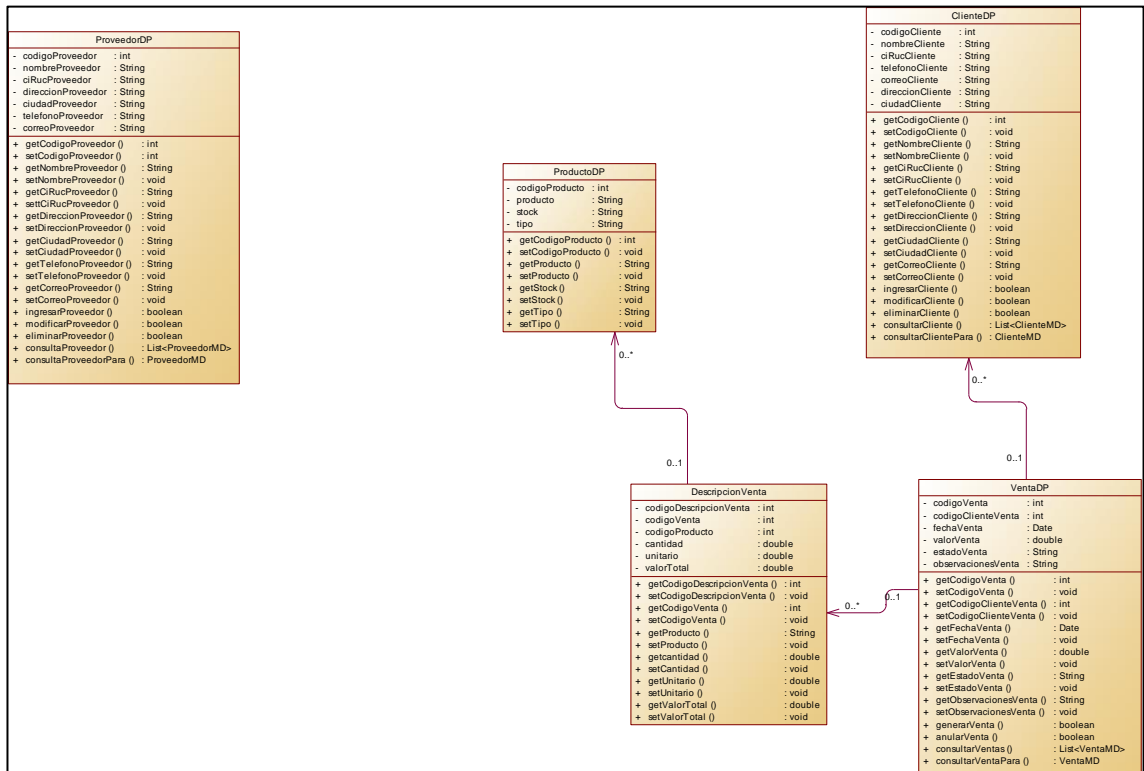


Figura 3-85. Diagrama de Clases DP Iteración 5 (Zurita U, 2017).

## Manejo de Datos (MD)

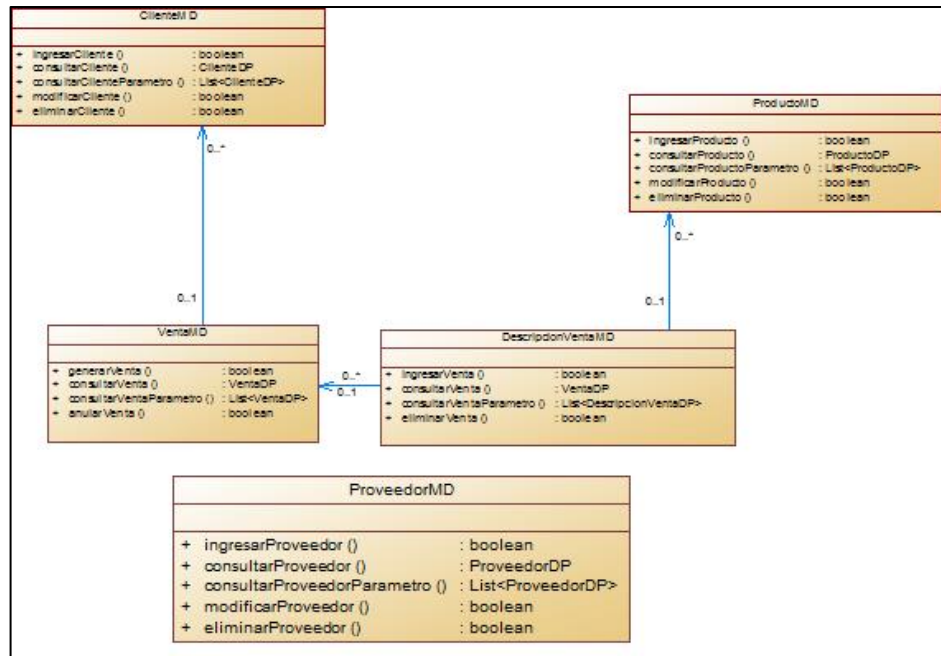


Figura 3-86. Diagrama de Clases MD Iteración 5 (Zurita U, 2017).

### 3.7.2. Desarrollo

#### Prototipo de GUI – Administrar Ventas

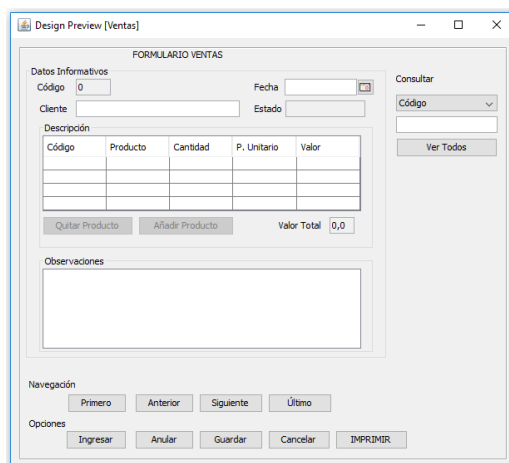


Figura 3-87. Prototipo GUI Ventas (Zurita U, 2017).

**Descripción:**

El prototipo de la sexta ventana, nombrada “Ventana Ventas”, se la realiza en base a las necesidades del usuario establecidas previamente, cumpliendo con la Historia de Usuario 5.

El diseño de la interfaz es sencillo y con lo necesario para la administración de los registros de Ventas.

La ventana cuenta con 4 secciones:

- Datos  
Se proporciona campos para los diferentes datos de una venta.
  
- Descripción  
Detalle de los productos que pertenecen a la factura.
  
- Consultar  
En esta sección se presenta la opción de presentar todos los registros o la búsqueda por parámetro.
  
- Navegación  
Su funcionalidad es permitir moverse entre registros, facilitando la presentación de la información.
  
- Opciones  
En esta sección se presenta las acciones que se puede realizar para la administración de los registros de ventas; funciones para Generar, Anular, Guardar y Cancelar.

## Diseño Final

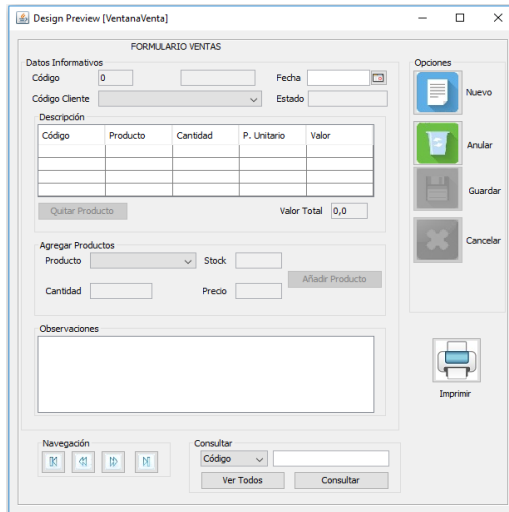


Figura 3-88. GUI Ventas (Zurita U, 2017).

### 3.7.3. Pruebas

#### Pruebas Caja Blanca – Unitarias

Las pruebas unitarias se las realiza en Netbeans y su complemento Junit.

#### Venta

Se realiza la prueba a los métodos pertenecientes a Ventana Venta, VentaDP, VentaMD.

#### Ventana Venta

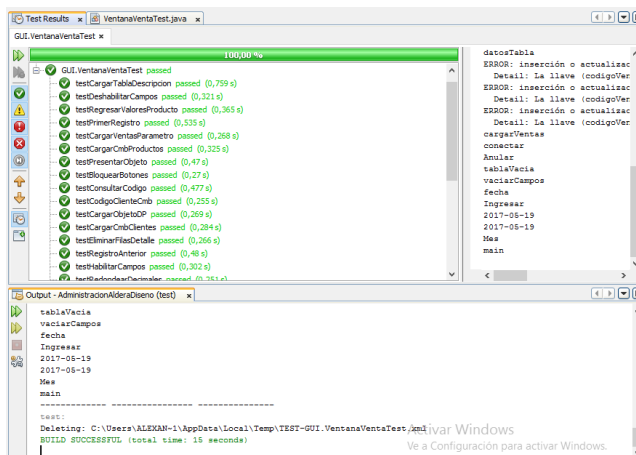


Figura 3-89. Prueba Unitaria Ventana Venta (Zurita U, 2017).

## VentaDP

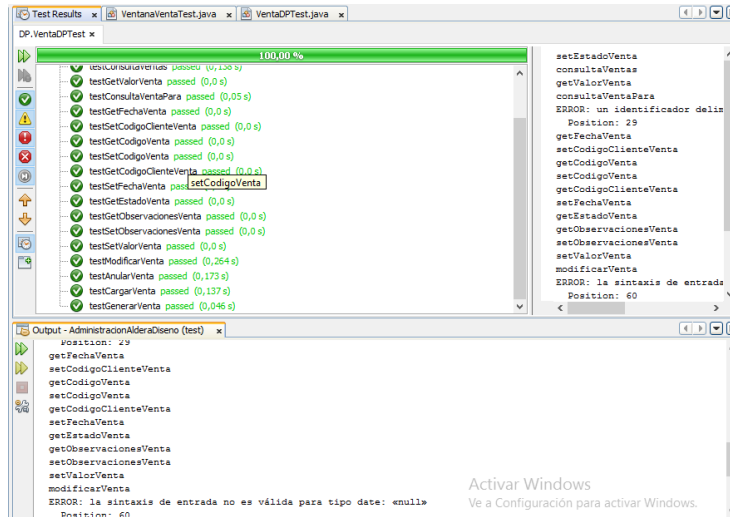


Figura 3-90. Prueba Unitaria Venta DP (Zurita U, 2017).

## VentaMD

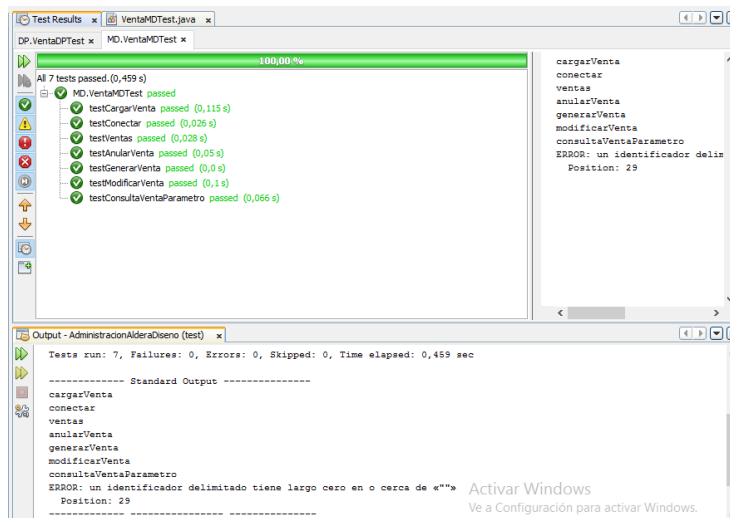


Figura 3-91. Prueba Unitaria Venta MD (Zurita U, 2017).

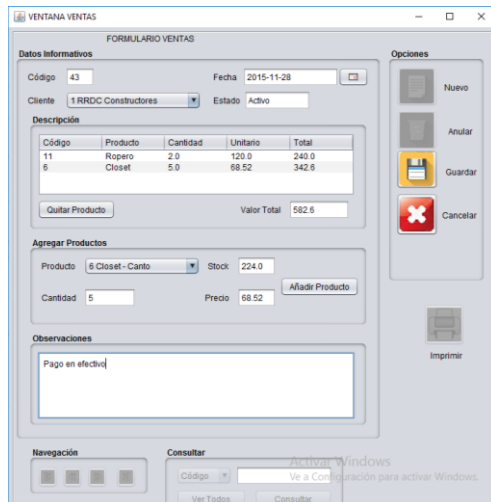
## Pruebas Caja Negra

### Generar Venta

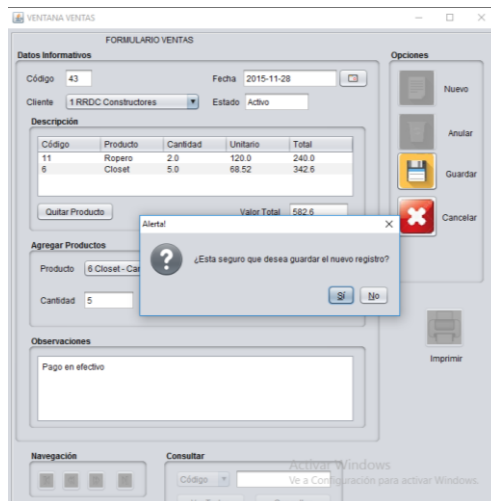
Datos de prueba:

- Cliente: RRDC Constructores
- Fecha: 2015-11-28

- Productos:
  - o Ropero – Cantidad: 2
  - o Closet – Cantidad: 5
- Observaciones: Pago en efectivo.



**Figura 3-92. Prueba Caja Negra Generar Venta Datos (Zurita U, 2017).**



**Figura 3-93. Prueba Caja Negra Generar Venta Confirmar (Zurita U, 2017).**

## Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera Diseño

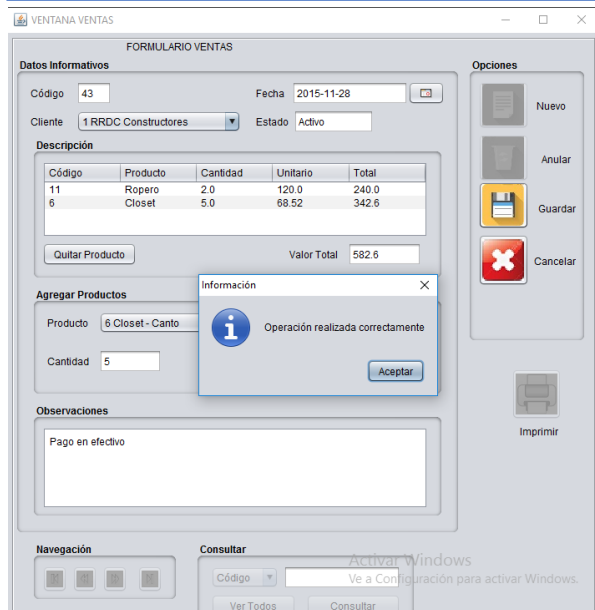


Figura 3-94. Prueba Caja Negra Generar Venta Resultado (Zurita U, 2017).

### Imprimir Factura

Presionar el botón “Imprimir”.

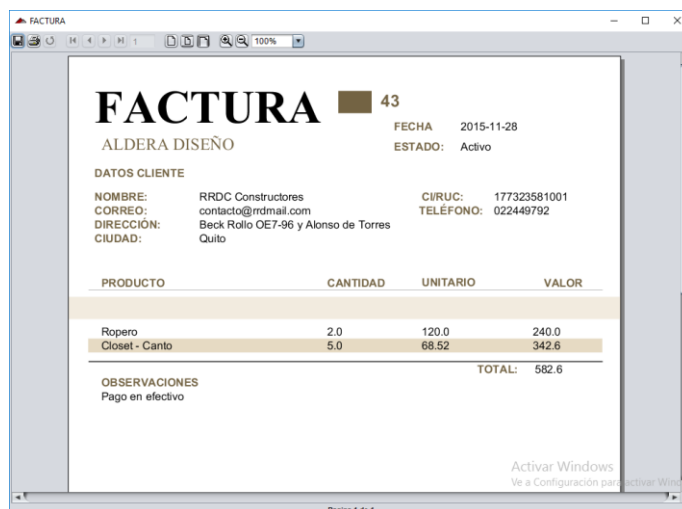


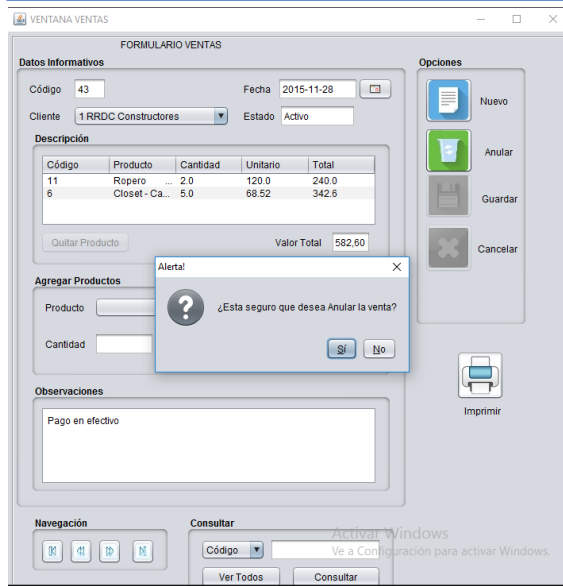
Figura 3-95. Prueba Caja Negra Generar Venta Reporte (Zurita U, 2017).

### Anular Venta

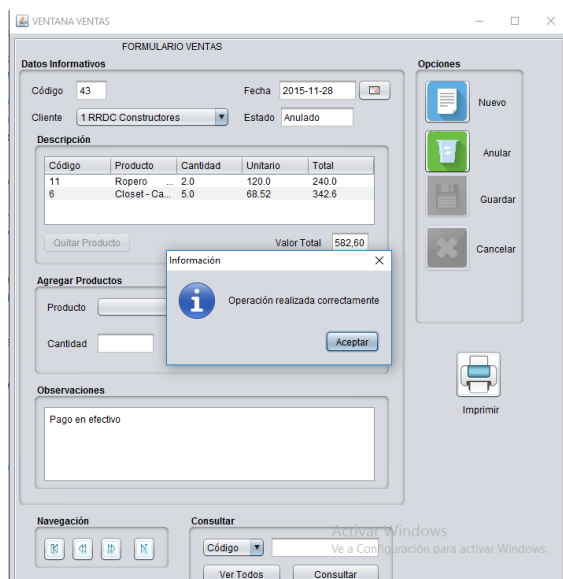
Datos de prueba:

- Código: 43

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño



**Figura 3-96. Prueba Caja Negra Anular Venta Confirmar (Zurita U, 2017).**



**Figura 3-97. Prueba Caja Negra Anular Venta Resultado (Zurita U, 2017).**

## Consulta General

Presionando el botón “Ver Todos”

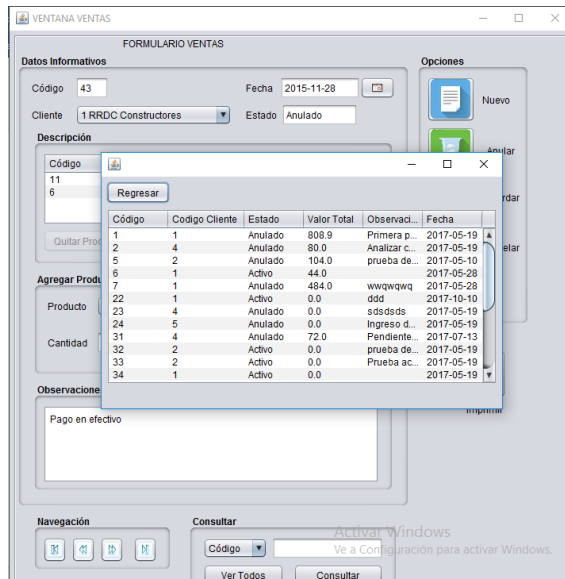


Figura 3-98. Prueba Caja Negra Consulta General Ventas (Zurita U, 2017).

## Consulta por parámetro

Datos de prueba:

- Parámetro: “Estado”; Valor: “Anulado”

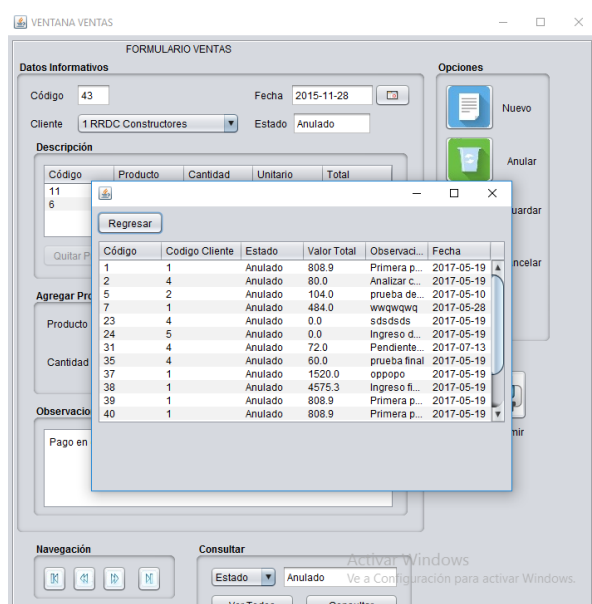


Figura 3-99. Prueba Caja Negra Consulta Parámetro Venta (Zurita U, 2017).

## Prueba de aceptación

PRUEBA DE ACEPTACIÓN	
Código: P5	Historia de Usuario: 5
Nombre: Prueba del proceso de venta.	
Descripción: Se realizará la prueba de generar, consultar y anular una venta.	
Condiciones de ejecución: Para efectuar la prueba el administrador o el empleado deberán entrar al sistema.	
Entrada/Pasos de ejecución: Después que el usuario (administrador o empleado) ha entrado al sistema, se abre la siguiente pantalla. En este caso para el proceso de ventas, se da clic en el botón VENTA. Se despliega una nueva pantalla, el usuario da clic en el botón NUEVO del menú OPCIONES. En la primera parte, el usuario escoge la fecha con el botón correspondiente y escoge un cliente de la lista correspondiente, a continuación escoge los productos y empieza a añadirlos en la descripción. Si es necesario el usuario escribirá una observación. El usuario da clic en el botón GUARDAR para guardar el registro y para imprimir la factura dará un clic en el botón IMPRIMIR, lo que le llevará a otra pantalla para visualizar el documento e imprimirlo. En esta pantalla puede navegar por los registros de venta por medio de los botones de NAVEGACION y consultar las <u>ventas que se han realizado de requerirlo se establece un parámetro de búsqueda</u>	
Resultado esperado: Se guarda la venta que se ha realizado, se puede acceder a sus datos por medio de consulta y se puede anular. La anulación solo puede ser hecha por el administrador.	
Evaluación de la prueba: Prueba exitosa.	

*Tabla 3-5. Prueba de Aceptación 5 (Zurita U, 2017).*

### 3.8. ITERACIÓN 6 – Administrar Compras

En esta iteración realizaremos el desarrollo e implementación de la funcionalidad F5. Administrar Compras.

#### 3.8.1. Diseño

En esta etapa, para continuar con la implementación necesitaremos de las clases Cliente, Venta, DescripciónVenta, Productos, Proveedor, Compra y DescripciónCompra.

## DIAGRAMA DE CLASES

### Interfaz Gráfica (GUI)

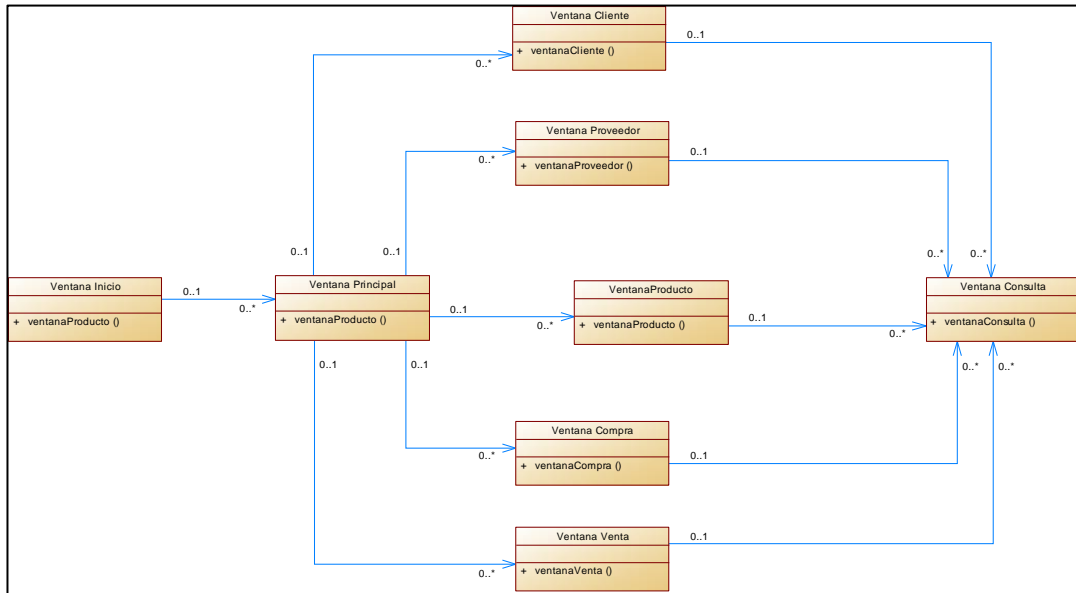


Figura 3-100. Diagrama de Clases GUI Iteración 6 (Zurita U, 2017).

### Dominio del Problema (DP)

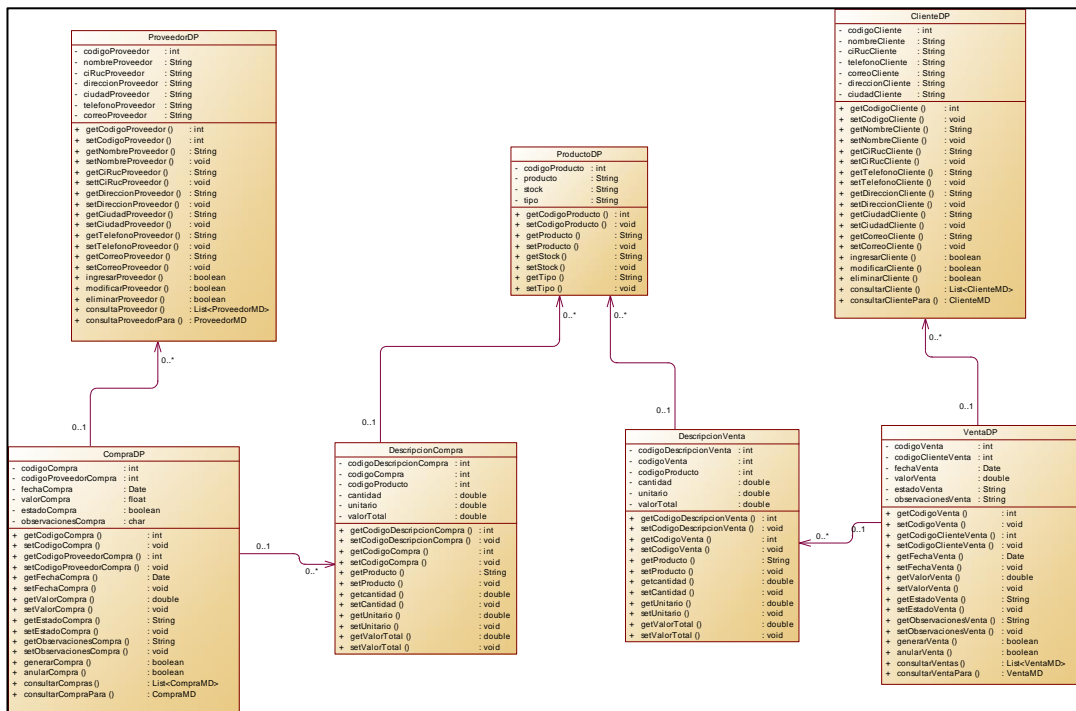


Figura 3-101. Diagrama de Clases DP Iteración 6 (Zurita U, 2017).

## Manejo de Datos (MD)

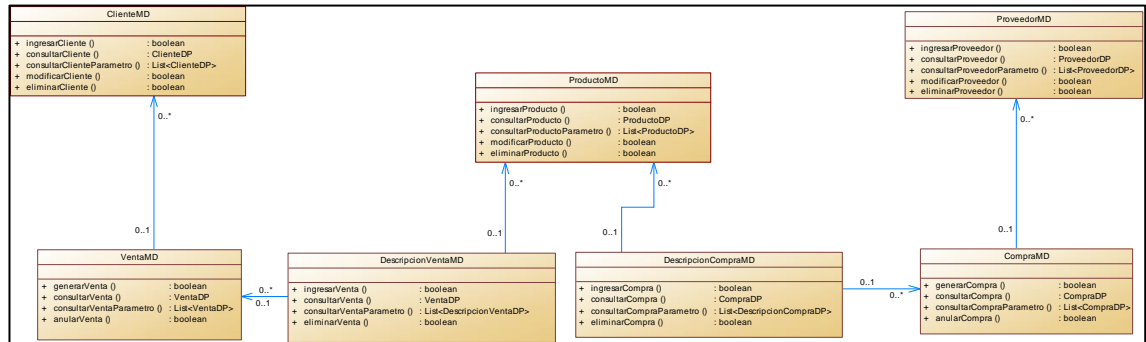


Figura 3-102. Diagrama de Clases MD Iteración 6 (Zurita U, 2017).

### 3.8.2. Desarrollo

#### Prototipo de GUI – Administrar Compras

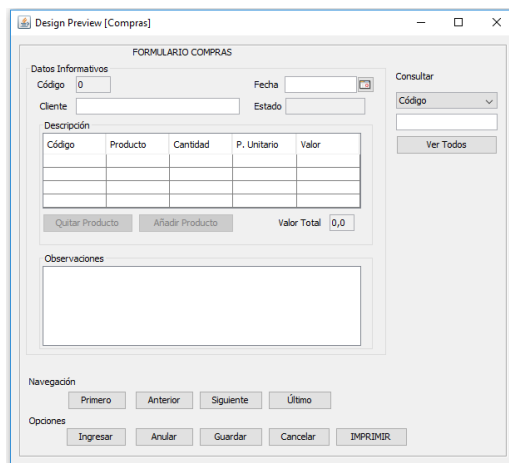


Figura 3-103. Prototipo GUI Compras (Zurita U, 2017).

#### Descripción:

El prototipo de la sexta ventana, nombrada “Ventana Compras”, se la realiza en base a las necesidades del usuario establecidas previamente, cumpliendo con la Historia de Usuario 6.

El diseño de la interfaz es sencillo y con lo necesario para la administración de los registros de Compras.

La ventana cuenta con 4 secciones:

- Datos  
Se proporciona campos para los diferentes datos de una compra.
- Descripción  
Detalle de los productos que pertenecen a la compra.
- Consultar  
En esta sección se presenta la opción de presentar todos los registros o la búsqueda por parámetro.
- Navegación  
Su funcionalidad es permitir moverse entre registros, facilitando la presentación de la información.
- Opciones  
En esta sección se presenta las acciones que se puede realizar para la administración de los registros de compras; funciones para Generar, Anular, Imprimir, Guardar y Cancelar.

## Diseño Final

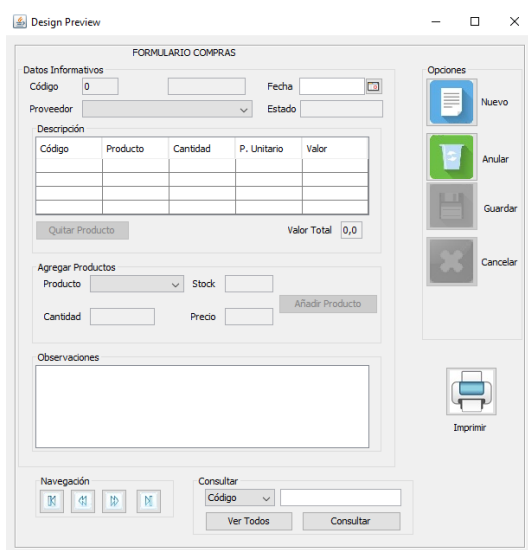


Figura 3-104. GUI Compras (Zurita U, 2017).

### 3.8.3. Pruebas

#### Pruebas Caja Blanca – Unitarias

Las pruebas unitarias se las realiza en Netbeans y su complemento Junit.

#### Compra

Se realiza la prueba a los métodos pertenecientes a Ventana Compra, CompraDP, CompraMD.

#### Ventana Compra

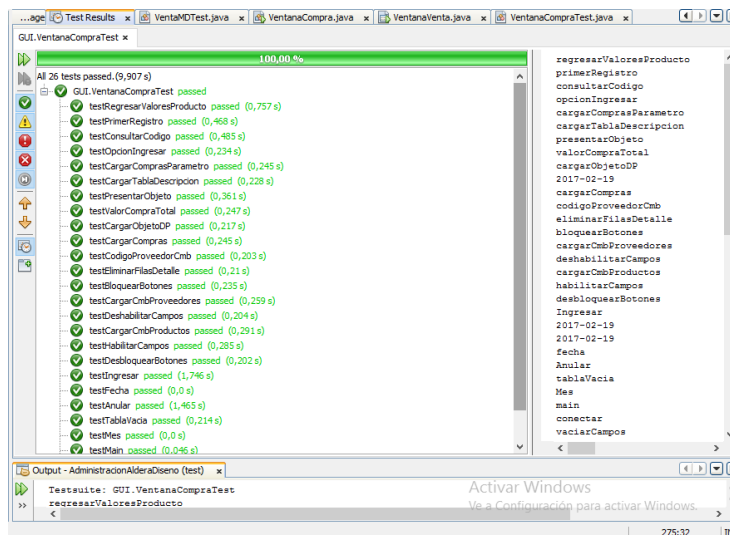


Figura 3-105. Prueba Unitaria Ventana Compras (Zurita U, 2017).

#### CompraDP

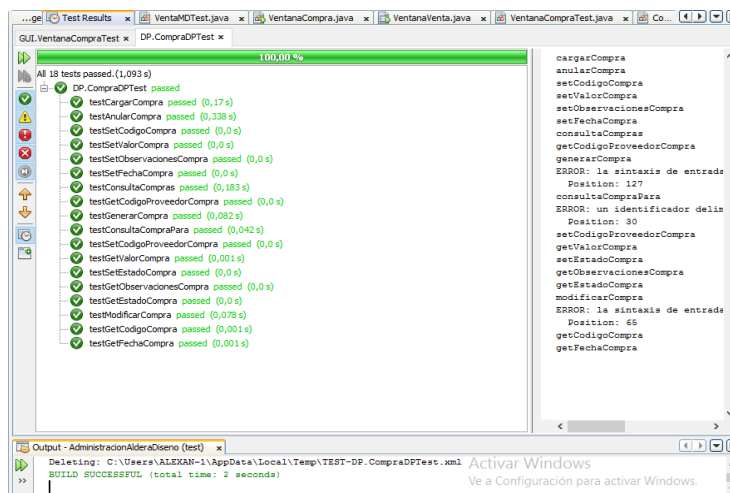


Figura 3-106. Prueba Unitaria Compra DP (Zurita U, 2017).

## CompraMD

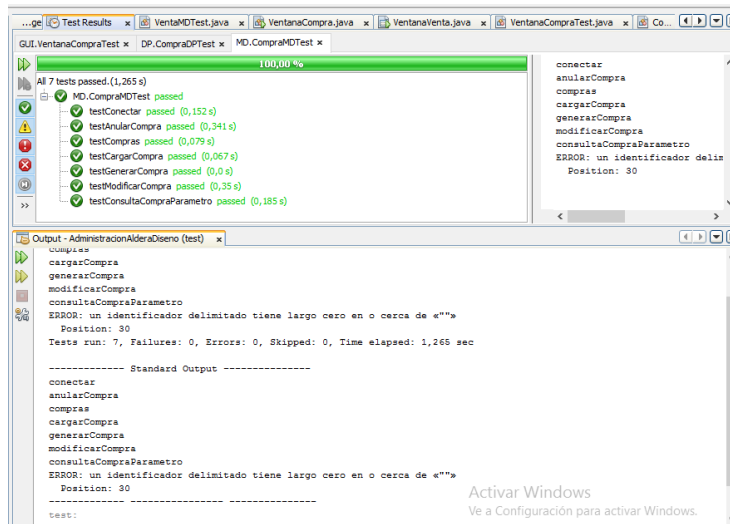


Figura 3-107. Prueba Unitaria Compra MD (Zurita U, 2017).

## Pruebas Caja Negra

### Generar Compra

Datos de prueba:

- Proveedor: Soarsi S.A
- Fecha: 2017-06-20
- Productos:
  - o Madera Canto – Cantidad: 30
  - o Madera Fina – Cantidad: 25
- Observaciones: Pago a crédito 6 meses.

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

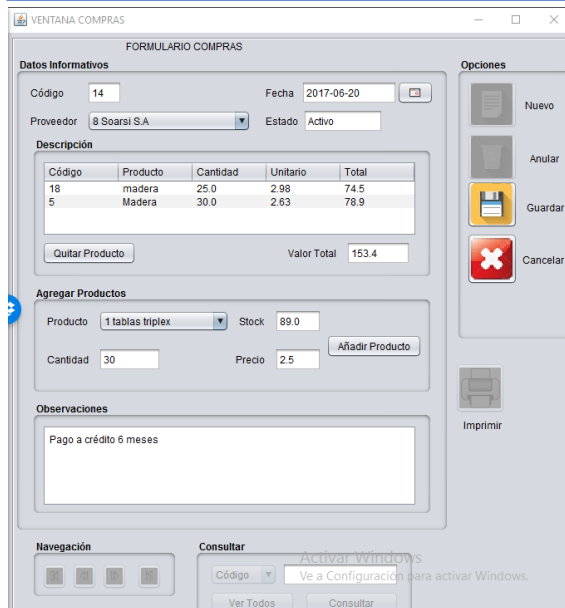


Figura 3-108. Prueba Caja Negra Generar Compra Datos (Zurita U, 2017).

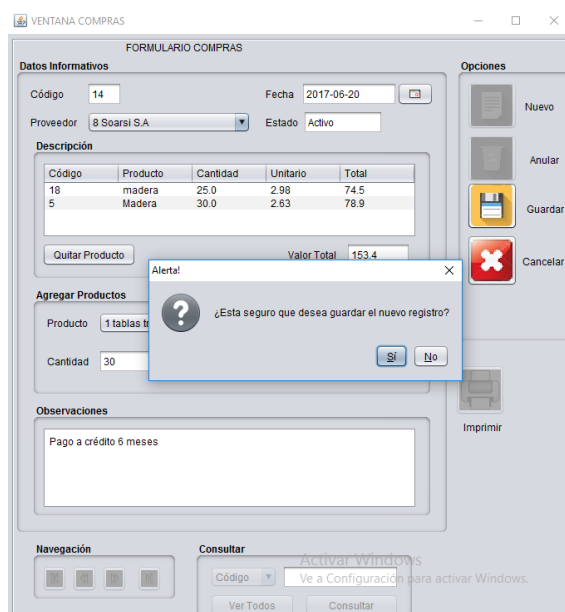


Figura 3-109. Prueba Caja Negra Generar Compra Confirmar (Zurita U, 2017).

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

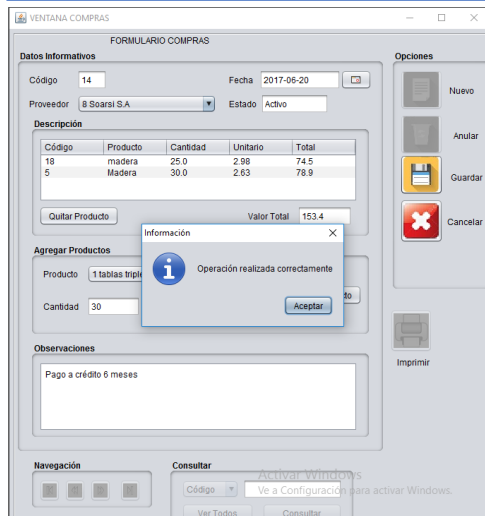


Figura 3-110. Prueba Caja Negra Generar Compra Resultado (Zurita U, 2017).

codigoCompra [PK] integer	codigoProveedorCompra integer	fechaCompra date	valorCompra numeric	estadoCompra character varying (10)	observacionesCompra character varying (200)
1	1	2017-02-19	50.0	Activo	prueba 1
2	2	2017-09-09	300.0	Anulado	prueba 2
3	4	2015-11-28	180.0	Activo	mkkkjk
4	5	2015-11-28	164.0	Activo	kskldem,df
5	6	2017-02-19	23.0	Activo	sddsd
6	7	2003-07-09	512.0	Anulado	Prueba de ingreso 11/0.
7	9	2017-02-19	50.0	Activo	zxczc
8	10	2017-02-19	1200.0	Activo	
9	11	2017-02-19	3561.1	Activo	3561.10
10	12	2017-02-19	50.0	Activo	prueba 1
11	13	2017-02-19	50.0	Activo	prueba 1
12	14	2017-06-20	153.4	Activo	Pago a crédito 6 meses

Figura 3-111. Prueba Caja Negra Generar Compra BDD (Zurita U, 2017).

## Imprimir Reporte

Presionando el botón “Imprimir”

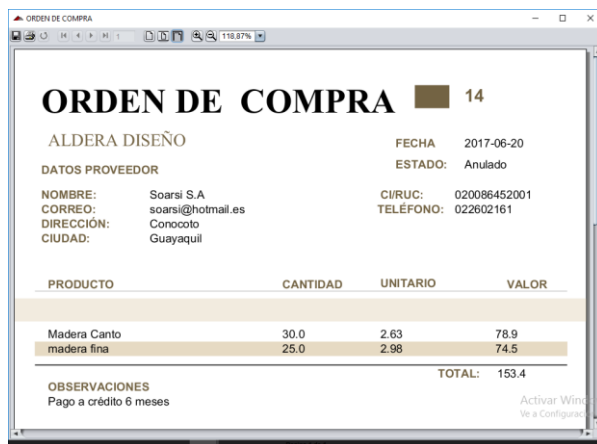
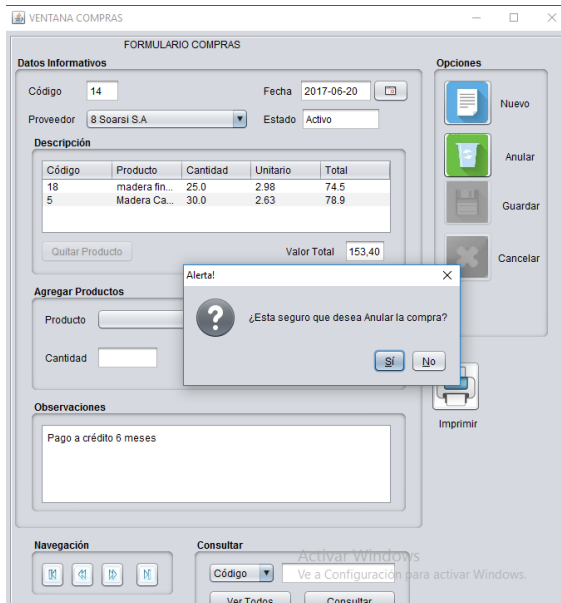


Figura 3-112. Prueba Caja Negra Generar Compra Reporte (Zurita U, 2017).

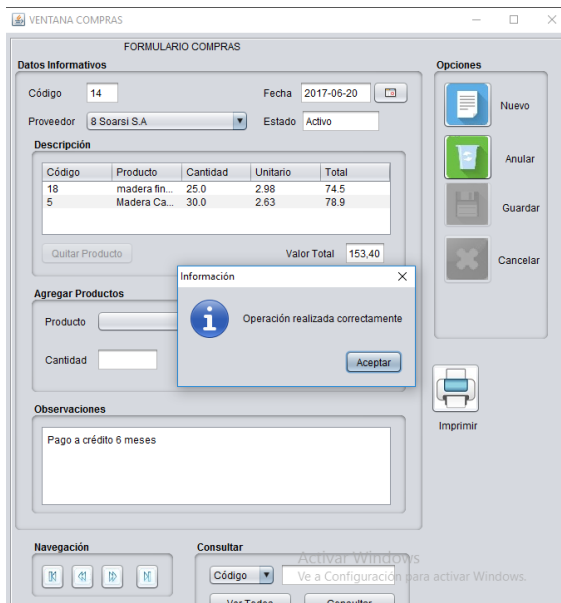
## Anular Compra

Datos de prueba:

- Código: 14



**Figura 3-113. Prueba Caja Negra Anular Compra Confirmar (Zurita U, 2017).**



**Figura 3-114. Prueba Caja Negra Anular Compra Resultado (Zurita U, 2017).**

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

	codigoCompra [PK] integer	codigoProveedor integer	fechaCompra date	valorCompra numeric	estadoCompra character varying (10)	observacionesCompra character varying (200)
1	1		2017-02-19	50.0	Activo	prueba 1
2	2		2017-09-09	300.0	Anulado	prueba 2
3	4		2015-11-28	180.0	Activo	mkkjkjk
4	5		2015-11-28	164.0	Activo	kskldsm,df
5	6		2017-02-19	23.0	Activo	sddsd
6	7		2003-07-09	512.0	Anulado	Prueba de ingreso 11/0...
7	9		2017-02-19	50.0	Activo	zxczc
8	10		2017-02-19	1200.0	Activo	
9	11		2017-02-19	3561.1	Activo	3561.10
10	12		2017-02-19	50.0	Activo	prueba 1
11	13		2017-02-19	50.0	Activo	prueba 1
12	14		2017-06-20	153.4	Anulado	Pago a crédito 6 meses

Figura 3-115. Prueba Caja Negra Anular Compra BDD (Zurita U, 2017).

### Consulta General

Presionando el botón “Ver Todos”

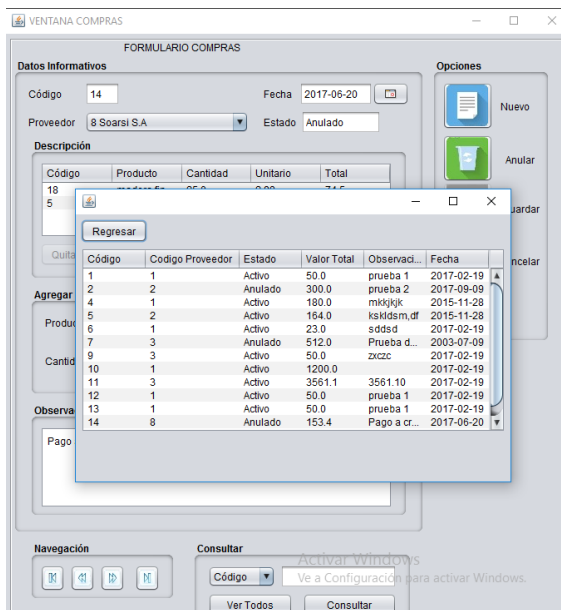


Figura 3-116. Prueba Caja Negra Consulta General Compra (Zurita U, 2017).

### Consulta por parámetro

Datos de prueba:

- Parámetro: Estado
- Valor: “Activo”

Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera  
Diseño

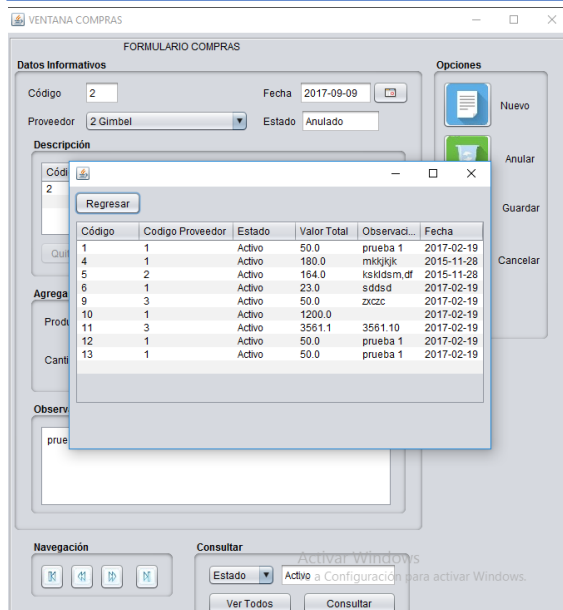


Figura 3-117. Prueba Caja Negra Consulta Parámetro Compra (Zurita U, 2017).

### Prueba de aceptación

PRUEBA DE ACEPTACIÓN	
<b>Código:</b> P6	<b>Historia de Usuario:</b> 6
<b>Nombre:</b> Prueba del proceso de compra.	
<b>Descripción:</b> Se realizará la prueba de generar, consultar y anular una orden de compra.	
<b>Condiciones de ejecución:</b> Para efectuar la prueba el administrador o el empleado deberán entrar al sistema.	
<b>Entrada/Pasos de ejecución:</b> Después que el usuario (administrador o empleado) ha entrado al sistema, se abre la siguiente pantalla. En este caso para el proceso de compras, se da clic en el botón COMPRA. Se despliega una nueva pantalla, el usuario da clic en el botón NUEVO del menú OPCIONES. En la primera parte, el usuario escoge la fecha con el botón correspondiente y escoge un proveedor de la lista correspondiente, a continuación escoge los productos y empieza a añadirlos en la descripción. Si es necesario el usuario escribirá una observación. El usuario da clic en el botón GUARDAR para guardar el registro y para imprimir la orden de compra da un clic en el botón IMPRIMIR, lo que le llevará a otra pantalla para visualizar el documento e imprimirlo. En esta pantalla puede navegar por los registros de compra por medio de los botones de <u>NAVEGACION</u> y <u>consultar las ventas que se han realizado de requerirlo se establece un parámetro de</u>	
<b>Resultado esperado:</b> Se guarda la compra que se ha realizado, se puede acceder a sus datos por medio de consulta y se puede anular. La anulación solo puede ser hecha por el administrador.	
<b>Evaluación de la prueba:</b> Prueba exitosa.	

Tabla 3-6. Prueba de Aceptación 6 (Zurita U, 2017).

## **CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES**

En este capítulo presento las conclusiones y las recomendaciones a las que se ha llegado, durante todo el proceso que se ha llevado a cabo para el desarrollo e implementación de este sistema.

### **4.1. CONCLUSIONES**

- Como objetivo principal, desarrollar un sistema capaz de almacenar, recuperar y manejar los datos de los clientes y proveedores de la empresa Aldera Diseño. Este objetivo se ha logrado, e incluso se ha ampliado ya que, por medio del sistema, se está manejando los datos de los clientes, proveedores y productos, y se está llevando a cabo los procesos de venta y compra.
- Dentro del desarrollo y acorde a la metodología aplicada, por medio de las historias de usuario y una constante interacción con el cliente, se determinaron de forma correcta los requerimientos que está cumpliendo el sistema para satisfacer a las necesidades de la empresa.
- Para una interacción fácil y eficaz se desarrolló una interfaz totalmente amigable con el usuario con ventanas sencillas e íconos que facilitan la comprensión de las funciones del sistema.
- Durante el desarrollo de un sistema informático, la interacción con el usuario permite un diseño eficaz y óptimo del sistema cumpliendo con lo necesario para satisfacer los requerimientos.
- La fase de pruebas en todas sus iteraciones nos da la posibilidad de encontrar defectos y corregirlos a tiempo antes que el sistema entre en funcionamiento. Esto evita que se generen costos de revisión y cambios.
- Las pruebas unitarias permiten verificar el correcto funcionamiento de las funciones que forman parte del código fuente del sistema por medio de estas se puede concluir la eficacia del sistema.
- La automatización de los procesos compra y venta, y la gestión de clientes y proveedores de la empresa permite al cliente mantener una base de datos unificada que respalde y garantice la veracidad de la información.

- Una correcta planificación de las iteraciones en la metodología Extreme Programming es fundamental para la elaboración del sistema reduciendo tiempo y costos en la elaboración del sistema.
- Las interfaces gráficas con un diseño amigable, sencillo y comprensible para el usuario, facilita el funcionamiento del sistema, evitando errores por confusión del usuario cuando realice un proceso.
- El trabajo en equipo junto con el usuario permite un desarrollo optimizado del sistema evitando las funciones innecesarias, así como el cumplimiento de los requerimientos del usuario.

#### **4.2. RECOMENDACIONES**

- Como todo sistema, éste ha sido desarrollado para que sea susceptible de ser mejorado. Una importante mejora para pensar en un futuro es el control del proceso de producción, con esto se tendrá un control mucho más certero de nuestra materia prima y el uso que se da a la misma al momento de la fabricación.
- Motivar al usuario para que al momento de las pruebas de aceptación realice todas las pruebas, considerando los posibles escenarios en los que puede llegar a desenvolverse el programa, para verificar que responde de forma correcta y que está hecho a prueba de errores.
- Mantener una constante comunicación con el cliente durante todo el proceso de elaboración del sistema ya que esto evitaría elaborar funciones que el usuario no comprenda o no requiera.
- Elaborar una planificación de acuerdo con los requerimientos del usuario previamente establecido por medio de las historias de usuario para evitar contratiempos o retrasos en el cumplimiento con el cliente.
- Realizar las pruebas al finalizar cada iteración para definir los cambios necesarios que debe realizarse en el sistema antes de pasar a la siguiente etapa.

- Al desarrollar un sistema mantener un patrón o plantilla de diseño para que el usuario no tenga complicaciones en el manejo del sistema, siendo este de forma intuitiva y sin complicaciones.
- Aplicar las metodologías de desarrollo ágiles para la elaboración de un sistema enfocándose mas en el resultado final del producto y menos la documentación

## **BIBLIOGRAFÍA**

- Acebedo, L. F. (21 de Julio de 2016). *Read The Docs*. Recuperado el 20 de Junio de 2017, de <https://media.readthedocs.org/pdf/pgsqtutorial/9.1.0/pgsqtutorial.pdf>
- Aliaga Ibarra, A., & Miani Flores, M. A. (21 de Enero de 2008). *IES San Vicente*. Recuperado el 20 de Junio de 2017
- Alonso, P. (17 de Junio de 2017). *Sigmur*. Obtenido de [http://www.um.es/geograf/sigmur/sigpdf/temario\\_9.pdf](http://www.um.es/geograf/sigmur/sigpdf/temario_9.pdf)
- Aparicio, S. (13 de mayo de 2009).  *analisis692*. Recuperado el 18 de 06 de 2017, de <http:// analisis692.blogspot.com/2009/05/bases-de-datos.html>
- Buyto. (2009). *diferencias entre aplicaciones web y aplicaciones desktop*. Obtenido de <http://www.buyto.es/general-diseno-web/diferencias-entre-aplicaciones-web-y-aplicaciones-desktop>
- Cockbun, A., & Williams, L. (2000). *The Costs and Benefits of Pair Programming. Humans and technology Technical Report*.
- Gómez, V. (19 de Marzo de 2017). *Instinto Binario*. Recuperado el 29 de Agosto de 2017, de <https://instintobinario.com/diagrama-de-clases/>
- Groussard, T. (2012). *JAVA 7: Los fundamentos del lenguaje Java*. Ediciones Eni.
- IsraelCCM. (27 de Marzo de 2017). *CCM*. Recuperado el 27 de 06 de 2017, de Lenguajes de programación: <http://es.ccm.net/contents/304-lenguajes-de-programacion>
- Jeffries, R. (2001). *Extreme Programming in practice*. Addison-Wesley.
- Letelier, P., & Penadés, C. (2016). *eXtreme Programming*. Obtenido de upv.es: <http://users.dsic.upv.es/assignaturas/eui/lds/doc/masyxp.pdf>
- Lizarazo, F. (2017). *repository.usta.edu.co*. Recuperado el 19 de 04 de 2017, de <http://repository.usta.edu.co/bitstream/handle/11634/4726/LizarazoOsorioLauraFernanda2017.pdf>

- López, K. (27 de octubre de 2013). *aplicaciones-de-escritorio*. Obtenido de PREZI:  
[https://prezi.com/\\_2ftznleklm0/aplicaciones-de-escritorio/](https://prezi.com/_2ftznleklm0/aplicaciones-de-escritorio/)
- Lutz, M. (2010). *Learning Python* (Cuarta ed.). O'Reilly Media. Recuperado el 15 de junio de 2017
- Maldonado, F. (2017). *medic*. Obtenido de <http://www.medic.ula.ve/tsues/BASEDDATOS/LAS%2520BASES%2520DE%2520DATOS.pdf>
- Martin, R. (2002). *Continuous Care vs. Initial Design*. Obtenido de [www.objectmentor.com/resources/articles/Continuous\\_Care.pdf](http://www.objectmentor.com/resources/articles/Continuous_Care.pdf)
- Mindiola, J. (13 de Septiembre de 2012). *Ingeniería del Software II*. Recuperado el 28 de Agosto de 2017, de <http://jams001.blogspot.com/2012/09/diagrama-de-secuencia-y-colaboracion.html>
- Mora, R. (29 de julio de 2014). *Aplicaciones de escritorio*. Obtenido de [https://es.slideshare.net/romimo33/mdulo-4-informtica?next\\_slideshow=1](https://es.slideshare.net/romimo33/mdulo-4-informtica?next_slideshow=1)
- Ortiz, E. (2013). *ANÁLISIS DEL ENTORNO Y PROSPECTIVA*. Washington D.C.: Instituto Mediterráneo Publicaciones.
- Padilla, E. (2013). *Programación extrema*. Recuperado el 27 de junio de 2017, de <http://isw-itson.wix.com/ingenieriaensoftware#!programacion-extrema>
- Robles, G. (10 de octubre de 2002). *Programación extrema*. Recuperado el 28 de junio de 2017, de revistas electrónicas: <http://revistaselectronicas.ujaen.es/formacion/index.php/TTTT/article/viewFile/222/182>
- Rodriguez, G. (28 de Febrero de 2011). *analisis y diseño de sistemas*. Recuperado el 26 de 06 de 2017, de <http://sites.google.com/site/analdisist/analisis-y-diseno-de-sistemas/INGENIERIADEREQUERIMIENTOS.doc>
- Ruby. (11 de febrero de 2015). *ISSUU*. Recuperado el 17 de 06 de 2017, de <https://issuu.com/ruby68/docs/actividad3-m2-etapa1tic2-virtuanl.p>

- SENA. (2015). Obtenido de [https://senaintro.blackboard.com/bbcswebdav/institution/semillas/228106\\_2\\_VIRTUAL-2015/contenido/oaaps/oaap9/aa1/oa\\_pruebas\\_final/utilidades/descargable.pdf](https://senaintro.blackboard.com/bbcswebdav/institution/semillas/228106_2_VIRTUAL-2015/contenido/oaaps/oaap9/aa1/oa_pruebas_final/utilidades/descargable.pdf)
- Vargas del Valle, R. J., & Maltes Granados, J. P. (2 de 2007). *Di-Mare*. Recuperado el 7 de 9 de 2017, de <http://www.di-mare.com/adolfo/cursos/2007-2/pp-3capas.pdf>
- Velásquez, D. (30 de noviembre de 2012). *www.webprogramacion.com*. Obtenido de aplicaciones-web-vs-aplicaciones-de-escritorio: <https://www.webprogramacion.com/356/blog-informatica-tecnologia/aplicaciones-web-vs-aplicaciones-de-escritorio.aspx>
- Zurita U, E. A. (2017). *Desarrollo de un sistema para la administración de clientes y proveedores para la empresa Aldera Diseño*. Quito: Pontificia Universidad Católica del Ecuador. Recuperado el 2017