

UNIVERSIDAD DEL BÍO - BÍO  
FACULTAD DE CIENCIAS EMPRESARIALES  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y TECNOLOGÍAS DE  
INFORMACIÓN



**“SISTEMA PARA SERVICIO DE ATENCIÓN PRIMARIA DE URGENCIA  
(SAPU) Y TRIAGE (PRIORIZACIÓN DE LA ATENCIÓN) PARA LA  
PLATAFORMA VIRTUAL DEL CESFAM VIOLETA PARRA”**

Francisco Javier Garrido Sandoval

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN  
INFORMÁTICA

CHILLÁN, MARZO DE 2012

UNIVERSIDAD DEL BÍO - BÍO  
FACULTAD DE CIENCIAS EMPRESARIALES  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y TECNOLOGÍAS DE  
INFORMACIÓN



**“SISTEMA PARA SERVICIO DE ATENCIÓN PRIMARIA DE URGENCIA  
(SAPU) Y TRIAGE (PRIORIZACIÓN DE LA ATENCIÓN) PARA LA  
PLATAFORMA VIRTUAL DEL CESFAM VIOLETA PARRA”**

Francisco Javier Garrido Sandoval

PROFESOR(A) GUIA: SRA. MARCELA PINTO FERNANDEZ  
PROFESOR(A) INFORMANTE : SR. CLAUDIO MUÑOZ SEPÚLVEDA

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN  
INFORMÁTICA

CHILLÁN, MARZO DE 2012

## **Agradecimientos**

*Quiero dar las gracias a todas las personas que formaron parte durante todo mi proceso universitario.*

*Quiero agradecer a mi familia por ser incondicional en los momentos difíciles y por siempre brindar el apoyo que necesite.*

*Agradezco con especial énfasis a mi padre Rolando Garrido y a mi madre Ema Sandoval, por darme la oportunidad de estudiar y brindarme todo lo que necesite para llevar a cabo mi meta de egresar de mi carrera.*

*Agradezco también a mi polola Rocío que siempre me apoya y motiva a seguir adelante en los momentos que más lo necesito.*

*Agradezco a los profesores por brindarme las herramientas necesarias para poder lograr ser un buen profesional.*

*Y finalmente sin dejar de lado a mis compañeros que siempre se preocuparon, ayudaron y dieron apoyo cuando los necesite en esas largas noches de estudio que pasamos juntos.*

*Francisco Garrido Sandoval*

## Resumen

Hoy en día las tecnologías de la información han adquirido gran relevancia en las diversas organizaciones agilizando, ordenando y ayudando en la gestión de la información recabada por cada organización.

El Centro de Salud Familiar (CESFAM) Violeta Parra, es una entidad dedicada a la atención primaria de salud en la provincia de Ñuble y ha notado la importancia de adquirir nuevas tecnologías para ayudar a la gestión de la información, por ende adquirió una plataforma virtual donde considera la reservas de Servicio de Orientación Médico y Estadístico (SOME), las visitas domiciliarias y los factores de riesgos. Gracias a esta plataforma virtual se han logrado resultados satisfactorios, pero también queda en evidencia la necesidad de ampliar esta plataforma a más servicios que son prestados por el CESFAM creando nuevos módulos que sean integrados a la plataforma actualmente existente. Una de las áreas que posee el CESFAM en las cuales aún no abarca la plataforma se encuentra el Servicio de Atención Primaria de Urgencia (SAPU), y más específicamente la hoja diaria de atención SAPU, esto conlleva a una demora en las atenciones de usuarios, además, de un lento manejo de la información, la cual es recabada por el área y a través del mecanismo anteriormente mencionado.

En el presente informe se da a conocer la construcción de un módulo que se integrará a la plataforma virtual actualmente existente en el CESFAM. El cual consiste en el llenado de la hoja diaria de SAPU y el posterior manejo de la información obtenida a través de esta. La metodología de desarrollo utilizada fue la iterativa incremental con el enfoque de orientación a objetos, utilizando la arquitectura MVC (Modelo Vista Controlador), UML para el modelado del problema y Java para su implementación.

El proyecto dio como resultado un módulo que satisface las necesidades y requerimientos de los usuarios.



## Índice General

### **CAPÍTULO I: DESCRIPCIÓN GENERAL DE LA ORGANIZACIÓN Y ASPECTOS TÉCNICOS** 15

<b><u>1.1</u></b>	<b><u>INTRODUCCIÓN</u></b> .....	<b>16</b>
<b><u>1.2</u></b>	<b><u>DEFINICIÓN DE LA EMPRESA O INSTITUCIÓN</u></b> .....	<b>16</b>
1.2.1	DESCRIPCIÓN DE LA EMPRESA.....	16
1.2.2	PERSONAS.....	17
1.2.3	HISTORIA DE LA ORGANIZACIÓN .....	17
1.2.4	ORGANIZACIÓN .....	17
1.2.5	MISIÓN. ....	21
1.2.6	VISIÓN .....	21
1.2.7	VALORES.....	21
1.2.8	SITUACIÓN ACTUAL DE LA ORGANIZACIÓN.....	22

### **CAPÍTULO II: MARCO TEÓRICO**..... 25

<b><u>2.1</u></b>	<b><u>INTRODUCCIÓN</u></b> .....	<b>26</b>
<b><u>2.2</u></b>	<b><u>PARADIGMA DE PROGRAMACIÓN</u></b> .....	<b>27</b>
2.2.1	ORIENTACIÓN A OBJETOS .....	27
2.2.2	UML .....	29
<b><u>2.3</u></b>	<b><u>MODELO DE DESARROLLO DE SOFTWARE</u></b> .....	<b>33</b>
2.3.1	ARQUITECTURA DE SOFTWARE .....	34
2.3.2	ARQUITECTURA MVC.....	35
<b><u>2.4</u></b>	<b><u>PATRONES DE DISEÑO</u></b> .....	<b>37</b>
2.4.1	DATA ACCESS OBJECT .....	37
<b><u>2.5</u></b>	<b><u>TECNOLOGÍAS UTILIZADAS</u></b> .....	<b>37</b>
2.5.1	JAVA .....	37

2.5.2	J2EE.....	38
2.5.3	MYSQL.....	38
2.5.4	STRUTS 2.....	39
2.5.5	JAVASERVER PAGES (JSP).....	41
2.5.6	JAVASCRIPT.....	41
2.5.7	JQUERY.....	42
2.5.8	HIBERNATE.....	42
2.5.9	AJAX.....	45
2.5.10	APACHE TOMCAT.....	45
2.5.11	IREPORT.....	45
2.5.12	JASPER REPORT.....	45
<b>2.6</b>	<b><u>HERRAMIENTAS.....</u></b>	<b>46</b>
2.6.1	MYECLIPSE.....	46
<b><u>CAPÍTULO III: DESCRIPCIÓN DEL PROBLEMA Y LA SOLUCIÓN.....</u></b>		<b>47</b>
<b>3.1</b>	<b><u>INTRODUCCIÓN.....</u></b>	<b>48</b>
<b>3.2</b>	<b><u>DESCRIPCION GENERAL DEL PROBLEMA.....</u></b>	<b>48</b>
<b>3.3</b>	<b><u>DESCRIPCIÓN DE LA PROBLEMÁTICA.....</u></b>	<b>49</b>
<b>3.4</b>	<b><u>PROCESOS DE NEGOCIOS ACTUAL.....</u></b>	<b>50</b>
<b>3.5</b>	<b><u>OPORTUNIDADES.....</u></b>	<b>52</b>
	EXISTE UNA PLATAFORMA EN FUNCIONAMIENTO.....	52
	EXISTENCIA DE TECNOLOGÍA COMPUTACIONAL.....	52
	APOYO DE PARTE DE LOS FUNCIONARIOS DEL CESFAM VIOLETA PARRA.....	52
<b>3.6</b>	<b><u>DESCRIPCIÓN DE LA SOLUCIÓN.....</u></b>	<b>52</b>
<b>3.7</b>	<b><u>LIMITACIONES.....</u></b>	<b>53</b>
<b><u>CAPÍTULO IV: ANÁLISIS DE LA SOLUCIÓN.....</u></b>		<b>54</b>
<b>4.1</b>	<b><u>INTRODUCCIÓN.....</u></b>	<b>55</b>

<b><u>4.2</u></b>	<b><u>TITULO QUE IDENTIFICA EL PROYECTO .....</u></b>	<b><u>55</u></b>
<b><u>4.3</u></b>	<b><u>OBJETIVOS GENERALES Y ESPECIFICOS DEL PROYECTO.....</u></b>	<b><u>55</u></b>
4.3.1	OBJETIVO GENERAL.....	55
4.3.2	OBJETIVOS ESPECÍFICOS .....	55
<b><u>4.4</u></b>	<b><u>PROCESOS DE NEGOCIOS FUTUROS.....</u></b>	<b><u>56</u></b>
<b><u>4.5</u></b>	<b><u>DESCRIPCIÓN DE LOS ASPECTOS FUNDAMENTALES DE LA METODOLOGIA A UTILIZAR</u></b>	<b><u>58</u></b>
	<b><u>58</u></b>	
4.5.1	PRIMER INCREMENTO .....	58
4.5.2	SEGUNDO INCREMENTO .....	58
4.5.3	TERCER INCREMENTO .....	58
4.5.4	CUARTO INCREMENTO.....	59
<b><u>4.6</u></b>	<b><u>DETERMINACIÓN DE REQUERIMIENTOS DEL PROYECTO.....</u></b>	<b><u>59</u></b>
4.6.1	REQUERIMIENTOS NO FUNCIONALES. ....	59
<b><u>CAPÍTULO V: ANÁLISIS DE LA SOLUCIÓN PRIMER INCREMENTO.....</u></b>		<b><u>60</u></b>
<b><u>5.1</u></b>	<b><u>INTRODUCCIÓN .....</u></b>	<b><u>61</u></b>
<b><u>5.2</u></b>	<b><u>REQUERIMIENTOS FUNCIONALES.....</u></b>	<b><u>61</u></b>
<b><u>5.3</u></b>	<b><u>REQUERIMIENTOS NO FUNCIONALES.....</u></b>	<b><u>63</u></b>
<b><u>5.4</u></b>	<b><u>CASOS DE USO .....</u></b>	<b><u>64</u></b>
<b><u>5.5</u></b>	<b><u>ESPECIFICACIÓN DE MODULO.....</u></b>	<b><u>64</u></b>
<b><u>5.6</u></b>	<b><u>MODELO DE DATOS .....</u></b>	<b><u>65</u></b>
<b><u>5.7</u></b>	<b><u>DISEÑO FISICO DE LA BASE DE DATOS .....</u></b>	<b><u>66</u></b>
<b><u>CAPÍTULO VI: ANÁLISIS DE LA SOLUCIÓN SEGUNDO INCREMENTO.....</u></b>		<b><u>68</u></b>
<b><u>6.1</u></b>	<b><u>INTRODUCCIÓN .....</u></b>	<b><u>69</u></b>
<b><u>6.2</u></b>	<b><u>REQUERIMIENTOS FUNCIONALES.....</u></b>	<b><u>69</u></b>



<b>6.3</b>	<b><u>ESPECIFICACIÓN DE MODULO.....</u></b>	<b><u>71</u></b>
	<b><u>CAPÍTULO VII: ANÁLISIS DE LA SOLUCIÓN TERCER INCREMENTO .....</u></b>	<b><u>72</u></b>
<b>7.1</b>	<b><u>INTRODUCCIÓN .....</u></b>	<b><u>73</u></b>
<b>7.2</b>	<b><u>REQUERIMIENTOS FUNCIONALES.....</u></b>	<b><u>73</u></b>
<b>7.3</b>	<b><u>ESPECIFICACIÓN DE MODULO.....</u></b>	<b><u>75</u></b>
	<b><u>CAPÍTULO VIII: ANÁLISIS DE LA SOLUCIÓN CUARTO INCREMENTO .....</u></b>	<b><u>76</u></b>
<b>8.1</b>	<b><u>INTRODUCCIÓN .....</u></b>	<b><u>77</u></b>
<b>8.2</b>	<b><u>REQUERIMIENTOS FUNCIONALES.....</u></b>	<b><u>77</u></b>
<b>8.3</b>	<b><u>ESPECIFICACIÓN DE MODULO.....</u></b>	<b><u>79</u></b>
	<b><u>CAPÍTULO IX: FACTIBILIDAD DEL PROYECTO.....</u></b>	<b><u>80</u></b>
<b>9.1</b>	<b><u>INTRODUCCIÓN .....</u></b>	<b><u>81</u></b>
<b>9.2</b>	<b><u>FACTIBILIDAD.....</u></b>	<b><u>81</u></b>
<b>9.2.1</b>	<b><u>FACTIBILIDAD TÉCNICA.....</u></b>	<b><u>81</u></b>
<b>9.2.2</b>	<b><u>FACTIBILIDAD OPERATIVA.....</u></b>	<b><u>82</u></b>
<b>9.2.3</b>	<b><u>FACTIBILIDAD ECONÓMICA.....</u></b>	<b><u>83</u></b>
<b>9.2.4</b>	<b><u>CONCLUSIÓN DE LA FACTIBILIDAD .....</u></b>	<b><u>85</u></b>
	<b><u>CAPÍTULO X: PLAN DE PRUEBAS.....</u></b>	<b><u>86</u></b>
<b>10</b>	<b><u>PLAN DE PRUEBAS .....</u></b>	<b><u>87</u></b>
<b>10.1</b>	<b><u>PROPÓSITO.....</u></b>	<b><u>87</u></b>
<b>10.2</b>	<b><u>ÁMBITO.....</u></b>	<b><u>87</u></b>
<b>10.3</b>	<b><u>REQUERIMIENTOS DE LAS PRUEBAS.....</u></b>	<b><u>87</u></b>
<b>10.3.1</b>	<b><u>PRUEBAS DE INTEGRIDAD DE LOS DATOS:.....</u></b>	<b><u>87</u></b>
<b>10.3.2</b>	<b><u>PRUEBAS DE FUNCIONALIDAD:.....</u></b>	<b><u>88</u></b>
<b>10.3.3</b>	<b><u>PRUEBAS DE INTERFAZ DE USUARIO:.....</u></b>	<b><u>89</u></b>
<b>10.4</b>	<b><u>ESTRATEGIA DE PRUEBA.....</u></b>	<b><u>90</u></b>

<b>10.4.1 HERRAMIENTAS .....</b>	<b>90</b>
<b>10.4.2 RECURSOS .....</b>	<b>91</b>
<b>10.4.2.1 RECURSOS HARDWARE .....</b>	<b>91</b>
<b>10.4.2.2 RECURSOS SOFTWARE .....</b>	<b>91</b>
<b>10.4.2.3 RECURSOS HUMANOS.....</b>	<b>92</b>
<b>10.5 CASOS DE PRUEBA .....</b>	<b>93</b>
<b><u>11 CONCLUSIONES.....</u></b>	<b><u>97</u></b>
<b><u>12 BIBLIOGRAFÍA .....</u></b>	<b><u>99</u></b>
<b><u>ANEXO A: DOCUMENTO DE REQUERIMIENTOS.....</u></b>	<b><u>101</u></b>
<b><u>1 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE .....</u></b>	<b><u>102</u></b>
<b>1.1 RESTRICCIONES.....</b>	<b>102</b>
<b>1.2 OBJETIVO DEL SOFTWARE.....</b>	<b>103</b>
<b>1.3 DESCRIPCIÓN GLOBAL DEL PRODUCTO .....</b>	<b>103</b>
INTERFAZ DE USUARIO .....	103
INTERFACES DE COMUNICACIÓN .....	103
<b>1.4 REQUERIMIENTOS ESPECÍFICOS.....</b>	<b>104</b>
REQUERIMIENTOS FUNCIONALES DEL SISTEMA .....	104
<b>1.4.1 MÓDULO TÉCNICO PARAMÉDICO. ....</b>	<b>104</b>
<b>1.4.2 MÓDULO ENFERMERA.....</b>	<b>105</b>
<b>1.4.3 MÓDULO MÉDICO. ....</b>	<b>106</b>
<b>1.4.4 MÓDULO ESTADISTA .....</b>	<b>107</b>
<b>1.4.5 MÓDULO PORTAL .....</b>	<b>107</b>
INTERFACES EXTERNAS DE ENTRADA .....	107
INTERFACES EXTERNAS DE SALIDA .....	108
ATRIBUTOS DEL PRODUCTO .....	109
<b>1.5 CARACTERÍSTICAS DE LOS USUARIOS.....</b>	<b>110</b>
<b>1.6 REQUISITOS DE RENDIMIENTO. ....</b>	<b>110</b>

<b><u>ANEXO B: DIAGRAMA DE CASOS DE USO Y ESPECIFICACIÓN .....</u></b>	<b><u>111</u></b>
<b><u>DIAGRAMA DE CASOS DE USO .....</u></b>	<b><u>112</u></b>
ACTORES.....	113
ESPECIFICACIÓN DE LOS CASOS DE USO .....	117
<b><u>ANEXO C: DISEÑO DE INTERFAZ Y NAVEGACIÓN .....</u></b>	<b><u>124</u></b>
<b><u>DISEÑO DE INTERFAZ Y NAVEGACIÓN .....</u></b>	<b><u>125</u></b>

## Índice Tablas

Tabla 1: Población inscrita hasta el 13 de Agosto de 2010.....	19
Tabla 2 Requerimientos Funcionales del Primer Incremento, Funciones Técnico Paramédico.....	62
Tabla 3 Función “Login” .....	62
Tabla 4 Función “Buscar paciente por folio”.....	62
Tabla 5 Función “Mostrar datos paciente”.....	62
Tabla 6 Función “Ingresa datos paciente”.....	62
Tabla 7 Requerimientos no funcionales.....	63
Tabla 8 Requerimientos Funcionales del Segundo Incremento, Funciones Enfermera .....	69
Tabla 9 Función “Login” .....	69
Tabla 10 Función “Buscar paciente por folio”.....	70
Tabla 11 Función “Mostrar datos paciente”.....	70
Tabla 12 Función “Ingresa datos paciente”.....	70
Tabla 13 Requerimientos Funcionales del Tercer Incremento, Funciones Médico .	73
Tabla 14 Función “Login” .....	73
Tabla 15 Función “Buscar paciente por folio”.....	74
Tabla 16 Función “Mostrar datos paciente”.....	74
Tabla 17 Función “Ingresa datos paciente”.....	74
Tabla 18 Requerimientos Funcionales del Cuarto Incremento, Funciones Estadista	77
Tabla 19 Requerimiento Funcionales del Cuarto Incremento, Funciones Portal .....	77
Tabla 20 Función “Login” .....	78
Tabla 21 Función “Buscar paciente por folio”.....	78
Tabla 22 Función “Muestra lista de atención” .....	78
Tabla 24: Hardware y software necesario.....	81
Tabla 25: Costo de personal.....	83

## Índice Figuras

Figura 1: <i>Radio de acción CESFAM Violeta Parra</i> .....	18
Figura 2: <i>Organigrama CESFAM Violeta Parra</i> .....	20
Figura 3: <i>Ingresos del CESFAM Violeta Parra</i> .....	22
<i>Figura 4: Jerarquía de los diagramas UML 2.0, mostrados como un diagrama de clases</i> .....	29
Figura 5: <i>Modelo Iterativo Incremental, (Larman, 2003)</i> .....	34
Figura 6: Modelo – Vista – Controlador en Struts 2 .....	40
Figura 7: Configuración básica de Hibernate .....	43
Figura 8: Proceso de negocios actual. ....	51
Figura 9: Nuevo proceso de negocios. ....	57
Figura 10: Modelo Entidad Relación. ....	65
Figura 11 Porción de base de datos a utilizar .....	66
Figura 16: Caso de uso, ingresa datos. ....	112
Figura 17: Caso de uso, genera informes.....	113
Figura 13: Diseño de interfaz búsqueda de pacientes.....	125
Figura 14: Diseño de interfaz Enfermero/Médico.....	126
Figura 15: Diseño de interfaz Técnico Paramédico .....	127



# **CAPÍTULO I: DESCRIPCIÓN GENERAL DE LA ORGANIZACIÓN Y ASPECTOS TÉCNICOS**

---

## 1.1 INTRODUCCIÓN

---

En este capítulo se dará a conocer una descripción general del Centro de Salud Familiar Violeta Parra, organización en la cual se desarrolla el proyecto.

Para el desarrollo de este proyecto se cuenta directamente con la autorización de los siguientes profesionales: Directora del CESFAM Violeta Parra, Sra. Pamela Zamudio Villarroel; Jefe del departamento de informática, Don Claudio Torres Añazco; Jefe del departamento de estadística, Don Mario Leal Carrasco.

En primer lugar, se realiza una descripción de la institución para comprender con claridad la historia, visión y misión que tiene esta organización. En segundo lugar, se presentan el departamento de estadística y la unidad de SAPU, los que influyen directamente en este proyecto. Describa la organización, se presenta la situación actual y el proceso de negocio que lleva a cabo.

Por último, se presentan los problemas que se detectaron en el proceso de negocios dentro de la organización.

---

## 1.2 DEFINICIÓN DE LA EMPRESA O INSTITUCIÓN

---

### 1.2.1 Descripción de la empresa

Razón Social : Centro de Salud Familiar Violeta Parra  
Giro : Salud  
Rut : 61.931.800-5  
Director(a) : Pamela Zamudio Villarroel  
Dirección : Francisco Ramírez #150, Chillán  
Teléfono : 42 – 203624



### **1.2.2 Personas**

Persona contactada : Claudio Torres Añazco.

Cargo : Jefe Departamento de Informática.

### **1.2.3 Historia de la Organización**

El Centro de Salud Familiar Violeta Parra fue construido en 1992 al costado Este del Hospital Herminda Martín siendo en un comienzo un Consultorio General Urbano (CGU) convirtiéndose en CESFAM el año 2007. Sus funciones se encuentran destinadas a la atención primaria de salud. El CESFAM, es un establecimiento de Atención Primaria de la Comuna de Chillán, dependiente técnica y administrativamente del Servicio de Salud Ñuble. Funciona como unidad presupuestaria y administrativa independiente desde mayo de 1995. Su estructura física está hecha en albañilería reforzada de reciente construcción, con una superficie de 1.665, 21 MT<sup>2</sup> construidos y ubicado en un terreno de 4.860 [m<sup>2</sup>].

Actualmente la población inscrita en el CESFAM Violeta Parra es alrededor de 77.200 personas, lo que equivale a un 43,15 % de la población de Chillán. (CESFAM Violeta Parra, 2008).

### **1.2.4 Organización**

El Centro de Salud Violeta Parra se encuentra dividido en siete sectores para brindar una mejor atención a sus pacientes los cuales son: Sector 1, Sector 2, CECOF Padre Hurtado, Sector 4, Sector 5, Sector 6 y Sector 7. La unión de éstos abarca el territorio geográfico de la ciudad de Chillán, por lo tanto, dividen a los pacientes según el sector donde habitan. Los sectores 1, 2, CECOF, 4 y 5 se limitan geográficamente de la siguiente forma:

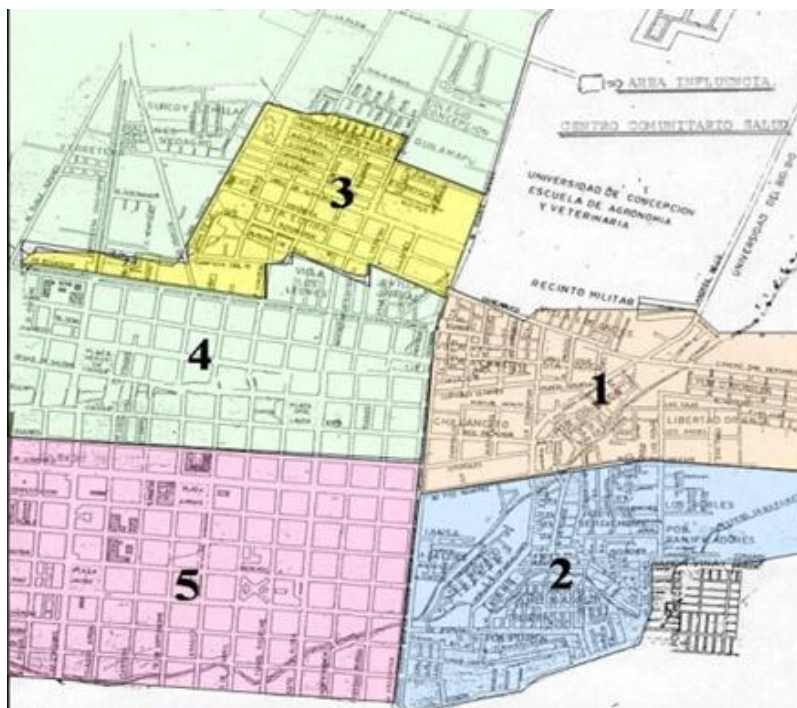
Límite norte: comprende desde el puente ferroviario sobre el Río Ñuble en su extremo sur, siguiendo por la rivera hasta la unión con el Río Cato, luego por la rivera sur de éste hasta el kilómetro 5, camino a Coihueco.

Límite Sur: comprende Avda. Collín en su lado Norte, entre avenidas Brasil y Argentina, siguiendo por calle Colón lado norte hasta el límite comunal de Coihueco.

Límite Este: con la comuna de Coihueco.

Límite Oeste: Avda. Brasil en su vereda este, hasta la esquina con Avda. Ecuador, para seguir desde allí por la línea férrea hasta el puente Ñuble.

Lo anterior se aclara en la Figura 1: Radio de acción CESFAM Violeta Parra:



**Figura 1: Radio de acción CESFAM Violeta Parra**

El Sector 6 representa a la población que se encuentra fuera del radio de acción ya mencionado y el Sector 7 constituye a la población de los sectores rurales.

En la Tabla 1: Población inscrita hasta el 13 de Agosto de 2010 se presenta el sector y la población que tiene inscrita en cada uno de éstos.

<b>Sector</b>	<b>Hombres</b>	<b>Mujeres</b>	<b>Total por Sector</b>
<b>1</b>	5.809	7.395	13.205
<b>2</b>	7.645	9.136	16.783
<b>CECOF</b>	2.865	3.366	6.231
<b>4</b>	5.606	7.448	13.058
<b>5</b>	4.798	6.102	10.905
<b>6</b>	5.872	6.682	12.560
<b>7</b>	609	686	1.302
<b>TOTAL</b>	33.204	40.815	74.019

**Tabla 1: Población inscrita hasta el 13 de Agosto de 2010**

En cada sector atienden enfermeras, matronas, trabajadores sociales, nutricionistas, psicólogos y técnicos paramédicos. Los odontólogos atienden de manera transversal, es decir, sin diferenciación de pacientes por sector al igual que el servicio de SAPU con la diferencia que el servicio SAPU abarca también a personas que no pertenecen al consultorio permanentemente. El organigrama se detalla en la Figura 2: Organigrama CESFAM Violeta Parra.

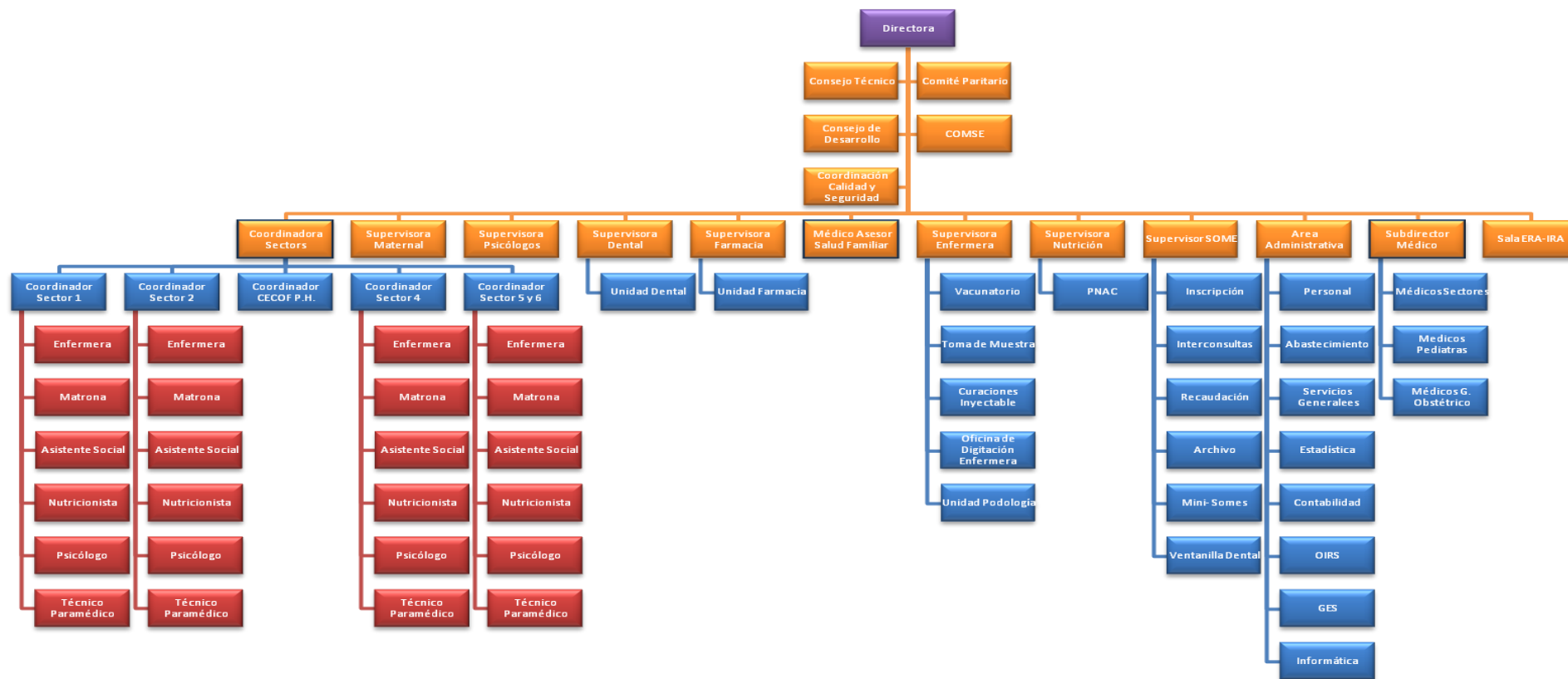


Figura 2: Organigrama CESFAM Violeta Parra

### **1.2.5 Misión.**

“Satisfacer en forma efectiva y eficiente las necesidades de salud de la población beneficiaria, en coordinación con otras instituciones sociales, priorizando a los sectores más pobres y vulnerables, maximizando la accesibilidad de la población a acciones de salud humanizadas, destinadas a la promoción, fomento, prevención y recuperación de la salud de las personas y el medio ambiente en el marco conceptual de la estrategia de la Atención Primaria”.

### **1.2.6 Visión**

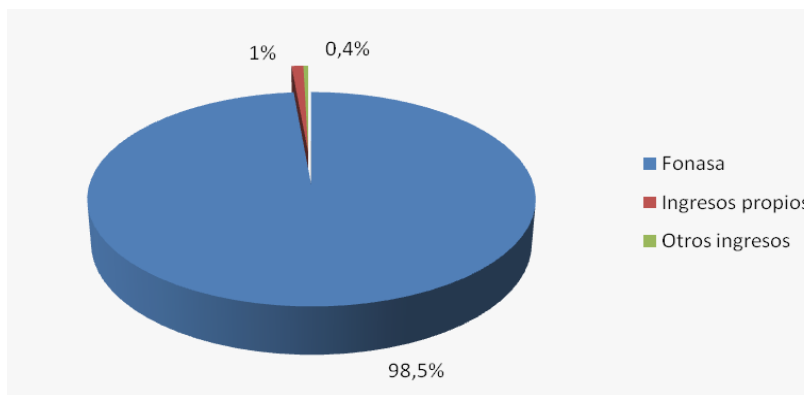
“Contribuir al bienestar de la población del sector, con el fin de lograr familias sanas, en un entorno saludable como reflejo de un satisfactorio trabajo en equipo y en red.”

### **1.2.7 Valores**

- Equidad: ofrecer igualdad de oportunidades en el acceso a la atención, de acuerdo a las prioridades sanitarias.
- Participación: trabajar en equipo tanto al interior del Consultorio como con los integrantes de la red asistencial y la comunidad de Ñuble.
- Satisfacción usuaria: otorgar un trato amable y oportuno.
- Eficiencia: obtener los mejores resultados posibles con los recursos existentes.
- Efectividad: entregar medicina basada en la mejor evidencia científica disponible.

### 1.2.8 Situación actual de la organización.

Los ingresos del CESFAM Violeta Parra son en gran parte aportados por FONASA<sup>1</sup> que financia el 98,5 % del total que recibe el centro de salud, lo restante son ingresos propios y otros ingresos que corresponden al 1% y al 0,4 % respectivamente.



**Figura 3: Ingresos del CESFAM Violeta Parra**

El dinero proporcionado por FONASA se divide en los siguientes conceptos:

- Atención Primaria: son programas financiados con metas definidas y suscritos a través de convenios, por ejemplo: Prog. Cardiovascular, Salud Mental, Resolutividad Atención Primaria, Reforzamiento Odontológico, entre otros.

---

<sup>1</sup> El Fondo Nacional de Salud, FONASA es el organismo público encargado de otorgar cobertura de atención, tanto a las personas que cotizan el 7% de sus ingresos mensuales para la salud en FONASA, como a aquellas que, por carecer de recursos propios, financia el Estado a través de un aporte fiscal directo.

Asimismo FONASA da cobertura de salud a todos sus beneficiarios, sin exclusión alguna de edad, sexo, nivel de ingreso, número de cargas familiares legales y enfermedades preexistentes, bonificando total o parcialmente las prestaciones de salud que les son otorgadas por profesionales e instituciones del sector público y privado.

Fuente: [www.fonasa.cl](http://www.fonasa.cl)

- Prestaciones Valoradas: corresponde a prestaciones definidas asociadas a un valor por actividad, por ejemplo: Actividades GES<sup>2</sup> como son: Ortesis, urgencias odontológicas, artritis reumatoidea, etc.
- Prestaciones Institucionales: Corresponde a las prestaciones habituales tabuladas a través de Estadísticas.

El dinero correspondiente a las prestaciones institucionales es recibido gracias al REM<sup>3</sup>, un documento en formato de planilla Excel proporcionado por el MINSAL<sup>4</sup>, donde se registran todas las cantidades de atenciones que realizan los profesionales de la salud. El REM debe ser completado por el Departamento de Estadística del CESFAM utilizando los datos obtenidos de la Hoja Diaria, que es un documento entregado por el Ministerio de Salud con formato determinado y que debe completar cada profesional de la salud. Allí se marcan las actividades que realiza cada profesional a cada usuario, las filas corresponden a los usuarios atendidos (pacientes) por el profesional de la salud durante el día y las columnas corresponden a los datos que se requieren de cada usuario.

De los 27 documentos del REM que debe entregar el departamento de estadística, el 85% de los ingresos de prestaciones institucionales corresponden a los tres siguientes reportes mensuales:

- Consultas médicas
- Despacho recetas a enfermos crónicos
- Atención de salud odontológica

De los tres reportes ya mencionados, el REM que indica las atenciones de salud odontológica, aporta entre un 20 y 23% de los ingresos de prestaciones institucionales.

Por otro lado, durante el año 2007, en el CESFAM Violeta Parra (en ese entonces Consultorio Violeta Parra) se implementó un sistema de apoyo a la gestión de reservas de horas para el Servicio de Orientación Médica y Estadística (SOME) que consiste en una aplicación Web para la realización de

---

<sup>2</sup> GES, Garantías explícitas de Salud. Se refiere a la protección financiera de algunos problemas de salud.

<sup>3</sup> REM, Reporte Estadístico Mensual.

<sup>4</sup> MINSAL, Ministerio de Salud.

reservas de horas de atención, ingreso y actualización de datos de los pacientes y profesionales del Centro de Salud, orientado a las personas que están inscritas en el centro. Este sistema fue implementado con éxito por un estudiante tesista de nuestra universidad, y se ha ido ampliando a medida que van surgiendo más necesidades dentro de esta institución.



## **CAPÍTULO II: MARCO TEÓRICO**

---

## 2.1 INTRODUCCIÓN

---

Para comenzar se describirán los conceptos relacionados a paradigmas de programación, entre estos, el modelo de desarrollo orientado a objetos, este nos lleva a la idea de un mundo lleno de objetos y que la resolución del problema se realiza en términos de los mismos.

Para el análisis y diseño de la aplicación será modelado utilizando U.M.L (lenguaje unificado de modelado) el cual representa los elementos del sistema a través de diagramas.

Para la implementación se utiliza el modelo iterativo incremental, que posee un enfoque de desarrollo modular, con entregas parciales del producto software denominados “incrementos” del sistema, que son escogidos según prioridades predefinidas de algún modo. Con cada incremento se agregan funcionalidades que cubren nuevos requisitos o bien se mejora la versión previamente implementada del producto software.

Además se describe la implementación del patrón de arquitectura MVC (Modelo-Vista-Controlador) que se utiliza para la separación de la capas de la aplicación, en la cual se implementan una serie de patrones J2EE propuestos por Sun Microsystem Inc. Los cuales están orientados a solucionar problemas comunes en el diseño de aplicaciones orientadas a objetos.

Para concluir, se definen las tecnologías a utilizar durante la implementación, es decir, el lenguaje de programación Java para este caso y el motor de base de datos MySQL.

---

## 2.2 PARADIGMA DE PROGRAMACIÓN

---

### 2.2.1 Orientación a objetos

Es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento. Su uso se popularizó a principios de la década de los años 90'. En la actualidad, existe una gran variedad de lenguajes de programación que soportan la orientación a objetos.

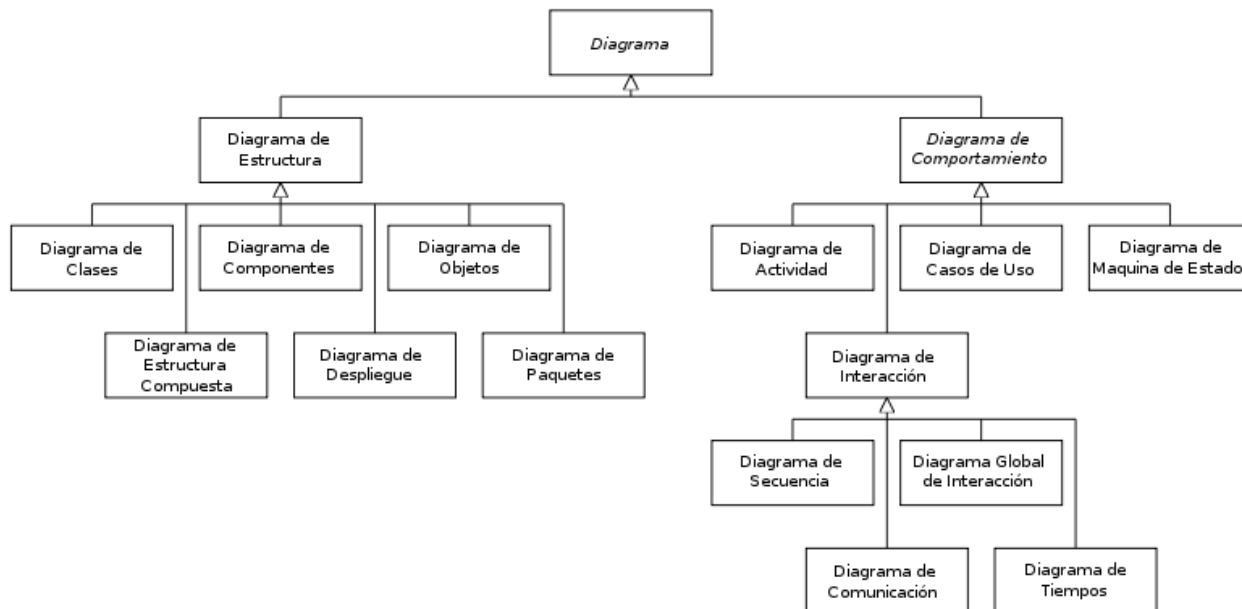
#### Conceptos Fundamentales

La programación orientada a objetos es una forma de programar que trata de encontrar una solución a estos problemas. Introduce nuevos conceptos, que superan y amplían conceptos antiguos ya conocidos. Entre ellos destacan los siguientes:

- Clase: definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.
- Herencia: (por ejemplo, herencia de la clase C a la clase D) Es la facilidad mediante la cual la clase D hereda en ella cada uno de los atributos y operaciones de C, como si esos atributos y operaciones hubiesen sido definidos por la misma D. Por lo tanto, puede usar los mismos métodos y variables públicas declaradas en C. Los componentes registrados como "privados" (private) también se heredan, pero como no pertenecen a la clase, se mantienen escondidos al programador y sólo pueden ser accedidos a través de otros métodos públicos. Esto es así para mantener hegemónico el ideal de OOP.
- Objeto: entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos) los mismos que consecuentemente reaccionan a eventos. Se corresponde con los

objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa). Es una instancia a una clase.

- Método: Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.
- Evento: Es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando, el mensaje adecuado al objeto pertinente. También se puede definir como evento, a la reacción que puede desencadenar un objeto, es decir la acción que genera.
- Mensaje: una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.
- Propiedad o atributo: contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto, esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.
- Estado interno: es una variable que se declara privada, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos). No es visible al programador que maneja una instancia de la clase.
- Componentes de un objeto: atributos, identidad, relaciones y métodos.
- Identificación de un objeto: un objeto se representa por medio de una tabla o entidad que esté compuesta por sus atributos y funciones correspondientes.



**Figura 4: Jerarquía de los diagramas UML 2.0, mostrados como un diagrama de clases**

### 2.2.2 UML

Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. (WORDPRESS.COM, 2010)

En UML 2.0 hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, a veces es útil categorizarlos jerárquicamente, como se muestra en la Figura 4.

Los Diagramas de Estructura enfatizan en los elementos que deben existir en el sistema modelado:

- **Diagrama de clases:**

Es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

- **Diagrama de componentes:**

Representa sistema de software dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

- **Diagrama de objetos:**

Se puede considerar un caso especial de un diagrama de clases en el que se muestran instancias específicas de clases (objetos) en un momento particular del sistema. Los diagramas de objetos utilizan un subconjunto de los elementos de un diagrama de clase. Los diagramas de objetos no muestran la multiplicidad ni los roles, aunque su notación es similar a los diagramas de clase. Una diferencia con los diagramas de clase es que el compartimiento de arriba va en la forma Nombre de objeto: Nombre de clase. Por ejemplo, Miguel: Persona.

- **Diagrama de estructura compuesta (UML 2.0):**

Muestra la estructura interna de una clase y las colaboraciones que esta estructura hace posibles. Esto puede incluir partes internas, puertas mediante las cuales, las partes interactúan con cada una de las otras. o mediante las instancias de las clases interactúan con las partes y con el mundo exterior. Una estructura compuesta es un conjunto de elementos interconectados que

colaboran en tiempo de ejecución para lograr algún propósito. Cada elemento tiene algún rol definido en la colaboración.

- **Diagrama de despliegue:**

Se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

- **Diagrama de paquetes:**

Muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

- **Diagrama de actividades:**

Representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. Un Diagrama de Actividades muestra el flujo de control general.

- **Diagrama de casos de uso:**

UML define una notación gráfica para representar casos de uso llamada modelo de casos de uso. UML no define estándares para que el formato escrito describa los casos de uso, y así mucha gente no entiende que esta notación gráfica define la naturaleza de un caso de uso; sin embargo una notación gráfica puede solo dar una vista general simple de un caso de uso o un conjunto de casos de uso. Los diagramas de casos de uso son a menudo confundidos con los casos de uso. Mientras los dos conceptos están relacionados, los casos de uso son mucho más detallados que los diagramas de casos de uso.

- **Diagrama de estados:**

Es un diagrama utilizado para identificar cada una de las rutas o caminos que puede tomar un flujo de información luego de ejecutarse cada proceso. Permite identificar bajo qué argumentos se ejecuta cada uno de los procesos y

en qué momento podrían tener una variación. El diagrama de estados permite visualizar de una forma secuencial la ejecución de cada uno de los procesos.

Los Diagramas de Interacción son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- **Diagrama de secuencia:**

Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista business del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos.

- **Diagrama de comunicación:**

Es una versión simplificada del diagrama de colaboración de la versión de UML 1.x.

Un diagrama de comunicación modela las interacciones entre objetos o partes en términos de mensajes en secuencia. Los diagramas de comunicación representan una combinación de información tomada desde el diagrama de clases, secuencia, y diagrama de casos de uso describiendo tanto la estructura estática como el comportamiento dinámico de un sistema.

- **Diagrama de tiempos (UML 2.0):**

Es una gráfica de formas de onda digitales que muestra la relación temporal entre varias señales, y cómo varía cada señal en relación a las demás.

- **Diagrama global de interacciones o Diagrama de vista de interacción (UML 2.0):**

Los modelos de interacción pueden llegar a ser muy grandes para sistemas complejos. Si el número de líneas de vida participantes y el número de mensajes intercambiados exceden una cierta medida, se impone “modularizar”



las interacciones y dividir en partes pequeñas, más manejables, de acuerdo a principios universales del diseño de sistemas, que también pueden ser visualizadas con la ayuda de un clásico diagrama de secuencias. La visión de conjunto de toda la interacción, de manera que la Big Picture o bien el cuadro global, puede entonces ser representada con la ayuda del diagrama global de las interacciones, provisto para eso.

---

## **2.3 MODELO DE DESARROLLO DE SOFTWARE**

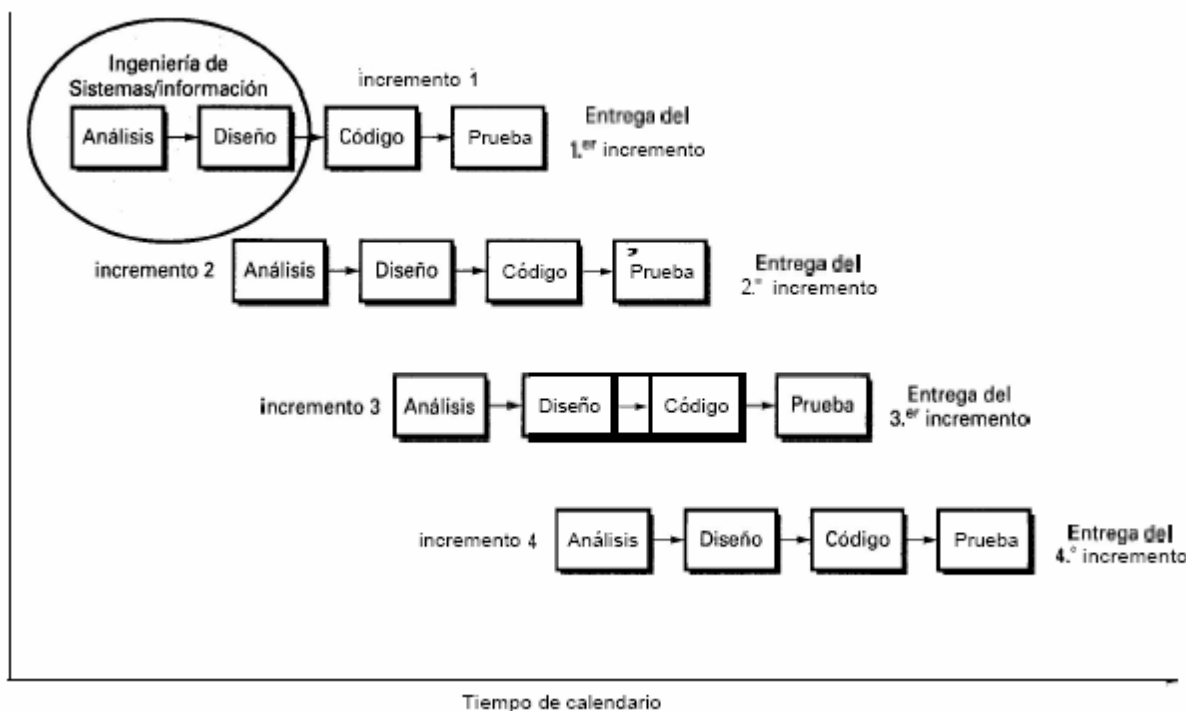
---

El modelo de desarrollo iterativo y creciente (o incremental) es un proceso de desarrollo de software, creado en respuesta a las debilidades del modelo tradicional de cascada. (Galindo Aguilar, y otros, 2002).

La idea principal detrás de mejoramiento iterativo es desarrollar un sistema de programas de manera incremental, permitiéndole al desarrollador sacar ventaja de lo que se ha aprendido a lo largo del desarrollo anterior, incrementando, versiones entregables del sistema, tal como se presenta en la Figura 4. El aprendizaje viene de dos vertientes: el desarrollo del sistema, y su uso (mientras sea posible). Los pasos claves en el proceso son comenzar con una implementación simple de los requerimientos del sistema, e iterativamente mejorar la secuencia evolutiva de versiones hasta que el sistema completo esté implementado. En cada iteración, se realizan cambios en el diseño y se agregan nuevas funcionalidades y capacidades al sistema.

El proceso en sí mismo consiste de:

- Etapa de inicialización.
- Etapa de iteración.
- Lista de control de proyecto.



**Figura 5: Modelo Iterativo Incremental, (Larman, 2003)**

### 2.3.1 Arquitectura de software

“Una arquitectura es el conjunto de decisiones significativas sobre la organización del sistema software, la selección de los elementos estructurales y sus interfaces, con los que se compone el sistema, junto con su comportamiento tal como se especifica en las colaboraciones entre esos elementos, la composición de esos elementos estructurales y de comportamiento en subsistemas progresivamente más amplios, y el estilo de arquitectura que guía esta organización –estos elementos y sus interfaces, sus colaboraciones, y su composición”. (Larman, 2003)

La arquitectura utilizada en el presente proyecto está basada en el patrón MVC.

### 2.3.2 Arquitectura MVC

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

#### Descripción del Patrón

- **Modelo:**

Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

- **Vista:**

Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

- **Controlador:**

Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, entre otro).
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.

3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

---

## 2.4 PATRONES DE DISEÑO

---

### 2.4.1 Data Access Object

Data Access Object (DAO) es un patrón de diseño que permite abstraer y encapsular todos los accesos a una base de datos en un objeto especial que utiliza las conexiones para ejecutar las consultas necesarias para obtener y almacenar datos.

Un DAO implementa los mecanismos de acceso requeridos para trabajar con la fuente de datos, que puede ser un RDBMS u otro servicio externo. El DAO debe esconder completamente los detalles de la implementación a sus clientes. Así, cualquier cambio en la base de datos no requiere cambio en los clientes si no sólo en el DAO.

Un objeto DAO sólo debe contener operaciones de lectura y modificación de datos. No debe contener operaciones con lógica de negocio. (Barrios N., 2003).

---

## 2.5 TECNOLOGÍAS UTILIZADAS

---

### 2.5.1 Java

Java es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si se construye un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo,

por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria. (Álvarez, 2010)

### **2.5.2 J2EE**

J2EE son las siglas de Java 2 Enterprise Edition que es la edición empresarial del paquete Java creada y distribuida por Oracle Corporation. Comprenden un conjunto de especificaciones y funcionalidades orientadas al desarrollo de aplicaciones empresariales.

Es una plataforma que habilita soluciones para desarrollo, uso efectivo y manejo de multicapas en aplicaciones centralizadas en el servidor.

J2EE utiliza la plataforma Java 2 SE, para brindar una completa, estable, segura, y rápida plataforma Java en el ámbito de la empresa. Habilita una plataforma que reduce de manera significativa los costos y la complejidad de desarrollo de soluciones multicapas, resultando en servicios que pueden ser desarrollados rápidamente y ampliados fácilmente. (De la Cruz García, 2006)

### **2.5.3 MYSQL**

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo:

- C
- C++
- C#
- Pascal
- Delphi (via dbExpress)
- Eiffel, Smalltalk
- Java (con una implementación nativa del driver de Java)
- Lisp
- Perl
- PHP
- Python
- Ruby
- Gambas
- REALbasic (Mac y Linux)
- FreeBASIC

Cada uno de estos utiliza una API específica. También existe una interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL. También se puede acceder desde el sistema SAP, lenguaje ABAP. (Domotica.Net, 2009)

#### **2.5.4 Struts 2**

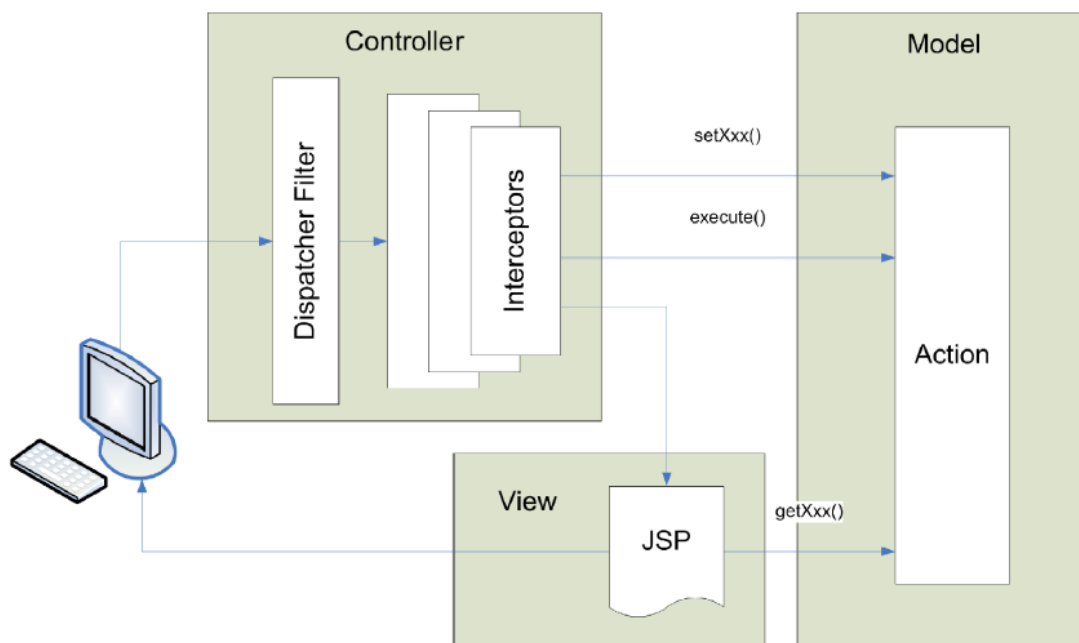
Struts 2 es la nueva versión del popular framework de desarrollo web en Java Apache Struts. Sin embargo, el código de ambos tiene poco que ver, dado que Struts 2 no se basa en el código de Struts 1.x, sino en el de otro framework de desarrollo web en Java llamado WebWork, un framework que el creador de Struts consideraba superior a Struts 1.x en varios aspectos.

Struts 2 está basado en el patrón MVC (Modelo-Vista-Controlador), una arquitectura que busca reducir el acoplamiento dividiendo las responsabilidades en 3 capas claramente diferenciadas:

- El modelo, que hace referencia a los datos que maneja la aplicación y las reglas de negocio que operan sobre ellos y que se traducen en Struts 2 en las acciones.

- La vista, encargada de generar la interfaz con la que la aplicación interactúa con el usuario. En Struts 2 equivale a los resultados.
- El controlador, que comunica la vista y el modelo respondiendo a eventos generados por el usuario en la vista, invocando cambios en el modelo, y devolviendo a la vista la información del modelo necesaria para que pueda generar la respuesta adecuada para el usuario. El controlador se implementa en Struts 2 mediante el filtro `FilterDispatcher`.

Esto se resume en la siguiente figura:



**Figura 6: Modelo – Vista – Controlador en Struts 2**

La ventaja de utilizar el framework Struts 2 es que éste se puede encargar de obtener y transformar los parámetros de las peticiones del cliente, validar los datos, buscar qué acción se debe realizar en el modelo según la petición, buscar la página a mostrar al usuario según la respuesta del modelo, proporcionar los datos del modelo a la vista, ocuparse de la internacionalización, etc.

Cabe mencionar, que un componente fundamental de Struts 2 que es utilizado en este proyecto son los interceptors, los cuales son responsables de la mayoría de los procesos hechos por el framework tales como: validación de



datos, transferencia de datos manejo de excepciones. Los interceptors mejoran inmensamente la separación de lo conceptual de la aplicación, removiendo tareas que son transversales a toda la aplicación como las mencionadas anteriormente. Por ejemplo, podemos configurar a un interceptor para que compruebe si el usuario está autenticado y si no lo está, re direccionar a la página de Login. Esto es posible debido a que los interceptors se ejecutan antes de la acción en curso. (Anónimo, 2009)

### **2.5.5 JavaServer Pages (JSP)**

JSP es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es, pues, una tecnología orientada a crear páginas web con programación en Java.

Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual. (Álvarez, 2002)

### **2.5.6 JAVASCRIPT**

Javascript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Javascript es el siguiente paso, después del HTML, que puede dar un programador de la web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para

hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica. (Álvarez, 2001)

### **2.5.7 JQuery**

JQuery es una biblioteca o framework de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC.

JQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. (Anónimo, 2010)

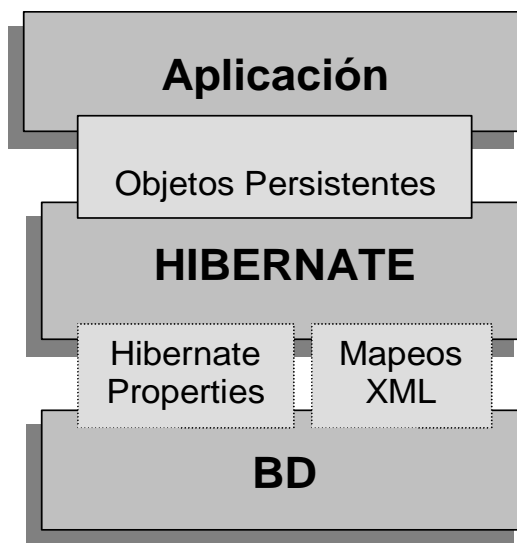
### **2.5.8 Hibernate**

Hibernate es una capa de persistencia objeto/relacional y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones y un gran número de tipos de datos. De una manera muy rápida y optimizada se pueden generar BD en cualquiera de los entornos soportados: Oracle, DB2, MySQL, Postgres, etc. y además, es open source, lo que supone, entre otras cosas, que no se debe pagar nada por adquirirlo. (Vázquez Rodríguez, 2006)

Uno de los posibles procesos de desarrollo consiste en, una vez que se tiene el diseño de datos realizado, es proceder a mapear las tablas del diseño de datos a archivos de extensión XML siguiendo la DTD de mapeo que posee

Hibernate, desde éstos podremos generar el código de nuestros objetos persistentes en clases Java y también crear base de datos (BD) independientemente del entorno escogido.

Hibernate se integra en cualquier tipo de aplicación justo por encima del contenedor de datos. Una posible configuración básica de Hibernate se muestra en la figura.



**Figura 7: Configuración básica de Hibernate**

Hibernate utiliza la base de datos y la configuración de los datos para proporcionar servicios y objetos persistentes a la aplicación que se encuentre justo por arriba de él.

Algunos beneficios de Hibernate que se pueden mencionar son:

✓ **Productividad:**

El código relacionado con la persistencia resulta tedioso para las aplicaciones de Java. Hibernate elimina la mayor parte de éste y permite concentrarse principalmente en el problema, sin importar la estrategia de

programación que se esté utilizando top-down (empezando del modelo global del sistema) o bottom-up (empezando con un esquema de base de datos existente).

✓ **Capacidad de mantenimiento:**

Menor número de LOC (Líneas de código) hacen un sistema más comprensible debido a que se da énfasis en la lógica del programa, haciendo más fácil su mantenimiento. Automatizando la persistencia de los objetos se reduce sustancialmente las LOC y de esta manera se reduce la complejidad de la aplicación independientemente de aislar el modelo relacional del orientado a objetos facilitando el empleo más elegante a la programación en Java.

✓ **Rendimiento (Performance):**

La persistencia implementada a mano puede ser tan rápida y realizar las mismas tareas como la automatizada, pero existen grandes diferencias entre éstas cuando se consideran algunas restricciones como tiempo y dinero. Cuando se realiza una tarea y es necesario manejar datos persistentes es posible realizar varias optimizaciones, a pesar de ser fácil su implementación con código escrito a mano (por ejemplo usando SQL/JDBC), la mayoría de las veces es más fácil de realizarlas con un ORM (Object/Relational Mapping). Cuando en un proyecto se está restringido por el tiempo, la persistencia codificada a mano solamente permite realizar algunas optimizaciones y únicamente bajo ciertas condiciones; Hibernate nos provee de muchas más optimizaciones que son usadas todo el tiempo. Además, la persistencia automatizada aumenta la productividad del desarrollador de manera que se cuenta con más tiempo para optimizar unos cuantos problemas restantes.

**Independencia de proveedor:**

Una gran ventaja de los ORM es que cuentan con herramientas de soporte para la mayoría de los manejadores de base de datos y esto añade un

alto nivel de portabilidad a la aplicación desarrollada, independientemente de facilitar su desarrollo.

### **2.5.9 AJAX**

Acrónimo de “Asynchronous JavaScript And XML” (JavaScript asíncrono y XML). Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. (Anónimo, 2010)

### **2.5.10 Apache Tomcat**

Tomcat funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de Java Server Pages (JSP) de Oracle Corporation. (The Apache Software Foundation, 2010)

### **2.5.11 iReport**

iReport es un diseñador de informes visual, poderoso, intuitivo y fácil de usar para JasperReports escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, etc. iReport está además integrado con JFreeChart, una de la biblioteca gráficas OpenSource más difundida para Java. Los datos para imprimir pueden ser recuperados por varios caminos, incluso múltiples uniones JDBC, TableModels, JavaBeans, XML, entre otros. (Herrera, 2005)

### **2.5.12 Jasper Report**

JasperReports es una herramienta de creación de informes Java que tiene la habilidad de entregar contenido enriquecido al monitor, a la impresora o a archivos PDF, HTML, XLS, CSV y XML.

Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones web, para generar contenido dinámico. Su propósito principal es ayudar a crear documentos de tipo páginas, preparados para imprimir en una forma simple y flexible. Ésta herramienta es utilizada comúnmente con iReport. (ITEM, 2010)

---

## 2.6 HERRAMIENTAS

---

### 2.6.1 MyEclipse

MyEclipse es un entorno de desarrollo que está disponible en el mercado de Java EE y AJAX IDE, creado y mantenido por la empresa Genuitec, miembro fundador de la Fundación Eclipse.

MyEclipse se basa en la plataforma Eclipse e integra soluciones de código abierto y propietario en el entorno de desarrollo.

MyEclipse tiene dos versiones principales (aparte de las "Blue Edition y "MyEclipse Edición Primavera"): una profesional y una edición estándar. La edición estándar incluye herramientas de base de datos, un diseñador de páginas web visual, herramientas de persistencia, herramientas de Struts y JSF, y un número de otras características para el perfil básico de Java Developer Eclipse. Compite con el Web Tools Project (Proyecto de Herramientas Web), que es una parte de Eclipse en sí, pero MyEclipse es un proyecto independiente por completo y ofrece un conjunto de características diferentes.

Más recientemente, MyEclipse ha puesto a disposición una herramienta de aprovisionamiento que mantiene los perfiles de software Eclipse, incluyendo los que utilizan MyEclipse. Además, MyEclipse está ofreciendo una versión personalizada de los productos de IBM, "MyEclipse Blue Edition", que añade soporte específico para Rational Software y el desarrollo de WebSphere. En la actualidad, MyEclipse Blue Edition está disponible para Windows y Linux, aunque para Mac no es compatible. (Anónimo, 2010)

# **CAPÍTULO III: DESCRIPCIÓN DEL PROBLEMA Y LA SOLUCIÓN**

---

## 3.1 INTRODUCCIÓN

---

En el presente capítulo se presenta la problemática que se debe abordar y como se desarrollan las actividades que conforman parte de la problemática.

Además de las oportunidades que esta genera y sus limitaciones.

---

## 3.2 DESCRIPCION GENERAL DEL PROBLEMA

---

El Servicio de Atención Primaria de Urgencia (SAPU) realiza atenciones de manera transversal a todos los usuarios del CESFAM no haciendo diferenciación por sector más aún sin hacer diferencia por consultorio de procedencia esto es una gran diferencia con respecto de la mayoría de los otros servicios que presta el CESFAM.

Esto produce una alta cantidad de atenciones y en consecuencia una gran cantidad de hojas SAPU.

En el servicio SAPU trabajan técnicos paramédicos, enfermera, médicos y administrativos en las siguientes cantidades:

- 10 Médicos
- 9 Técnicos Paramédicos
- 7 Enfermeros/as
- 9 Administrativos

Cabe destacar que las cantidades que se mencionan anteriormente corresponden a todas las personas que participan en el servicio SAPU durante el mes, pero más específicamente, las personas que participan en la atención SAPU por turno son las siguientes:

- 2 Médicos
- 3 Técnicos Paramédicos
- 1 Enfermera



Las hojas que se manejan diariamente en SAPU son variables según la cantidad de atenciones que se realicen durante el día puesto que se debe llenar una por cada atención que se realice. El número de atenciones depende de la cantidad de personas que asistan a buscar su servicio y la demora de cada una de las atenciones, pero finalmente se puede decir que en promedio se tienen entre cincuenta y sesenta hojas SAPU diariamente.

---

### **3.3 DESCRIPCIÓN DE LA PROBLEMÁTICA**

---

Hoy por hoy, el sistema que se utiliza consiste en el llenado de una hoja SAPU por cada persona que es atendida, esta consiste principalmente en un formulario pre-establecido, en el cual, se debe ingresar toda la información recaba por la atención, cabe mencionar que esta hoja viene con los datos personales del paciente ya impresos en ella, estos datos son los siguientes:

- Nombre
- Rut
- N° de ficha
- Folio
- Fono
- Fecha actual
- Hora de ingreso
- Fecha de nacimiento
- Consultorio de procedencia
- Edad
- Sexo
- Previsión

Al culminar el horario de atención del servicio SAPU todas las hojas con las atenciones que se realizaron ese día son entregadas a los administrativos que conforman el equipo SAPU, los cuales son los encargados de digitar toda la

información que estas poseen, esto es un trabajo tedioso y largo por lo cual la información no está siempre disponible por los estadísticos que realizan los REM y por ende, tampoco para los directivos del CESFAM.

---

### **3.4 PROCESOS DE NEGOCIOS ACTUAL**

---

Cada profesional que trabaja en el SAPU llena parte de cada hoja diaria según corresponda por cada uno de los pacientes que asisten a buscar el servicio diariamente, estas hojas son entregadas al departamento de estadísticas los que revisan si las hojas están correctamente rellenas. Si las hojas estas correctas estas son tabuladas en la plantilla REM, en caso contrario, son devueltas al SAPU para una revisión y corrección.

Cinco días antes que se termine el mes con los documentos ya tabulados, el encargado del departamento de estadísticas debe ir al Servicio de Salud Ñuble para entregar el REM. El servicio de Salud debe realizar un conglomerado de todos los reportes mensuales de todos los centros de salud y hospitales de la provincia de Ñuble y se envía al MINSAL. A partir de esto FONASA, que depende del ministerio envía su aporte económico a cada centro de salud y hospitales que es dinero correspondiente a las prestaciones institucionales.

El diagrama de procesos de negocios se detalla en la siguiente figura.

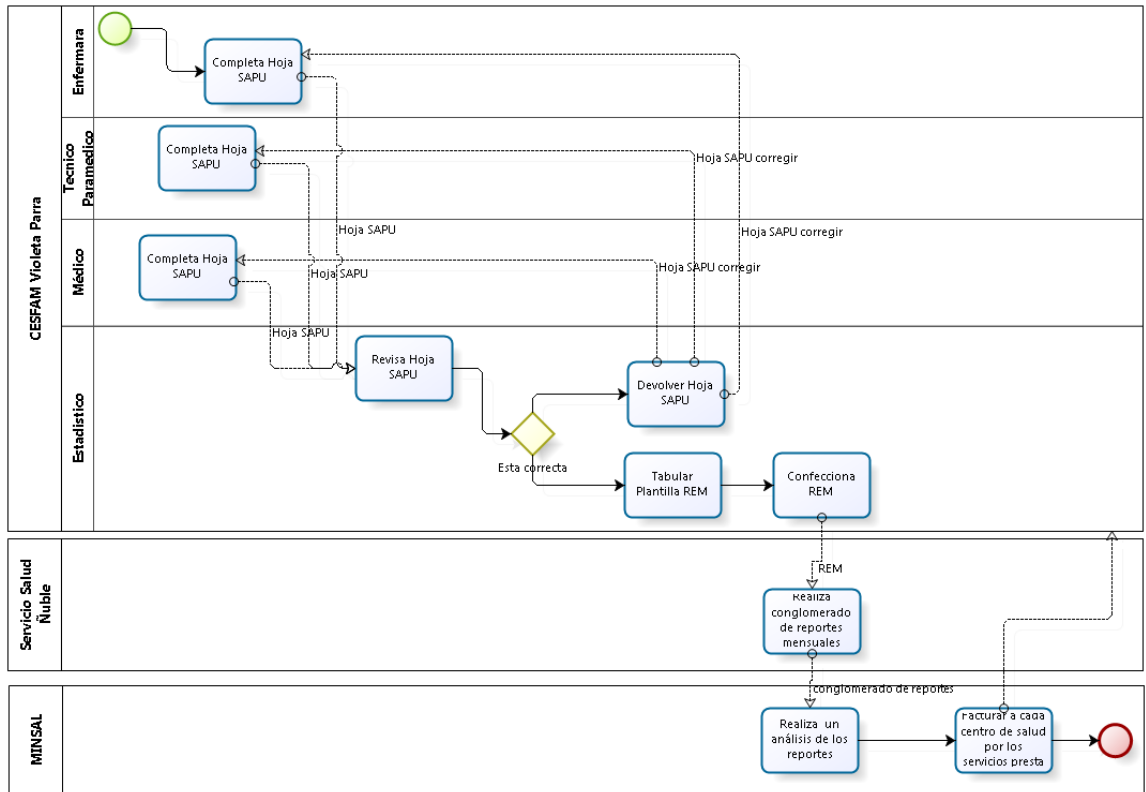


Figura 8: Proceso de negocios actual.

---

## 3.5 OPORTUNIDADES

---

### **Existe una plataforma en funcionamiento.**

El hecho de que exista una plataforma en funcionamiento es una oportunidad, ya que hay una base sobre la cual se va a trabajar. Además, dicha plataforma se creó sobre la plataforma J2EE, utilizando la arquitectura MVC.

### **Existencia de tecnología computacional.**

El departamento de informática posee cinco computadores, el departamento de estadística cuatro y en la mayoría de los box de atención hay un PC, por lo tanto se pueden lograr las condiciones óptimas para que el módulo funcione correctamente.

### **Apoyo de parte de los funcionarios del CESFAM Violeta Parra.**

Tanto los funcionarios que participaron del módulo a implementar como los que no, prestaron su ayuda cuando se solicitó. Los jefes de departamento con los que más se interactuó (estadística y unidad SAPU) siempre estuvieron dispuestos a colaborar. Además, se contó con el apoyo constante de los integrantes del departamento de informática, a los cuales se les realizaron consultas de carácter técnico.

---

## 3.6 DESCRIPCIÓN DE LA SOLUCIÓN

---

La solución del problema consiste en implementar una aplicación Web que se integre a la actual plataforma, ya que esta aplicación representará al módulo de la hoja SAPU para el CESFAM Violeta Parra. Esta aplicación debe contener el formulario correspondiente de la hoja SAPU para cada paciente.

También debe generar los datos solicitados por el departamento de estadística y la unidad dental.

---

### **3.7 LIMITACIONES**

---

Este nuevo Sistema no abarcara todas las instancias que ocurren durante una atención SAPU, se dejara de lado la inscripción de los pacientes puesto que esta se realiza actualmente con ayuda del sistema SOME y también no contempla para esta primera instancia la entrega de medicamentos ósea no tendrá integración con el Sistema de Farmacia.

Tampoco tendrá interconexión con la red MINSAL ya que será un Sistema propio del consultorio y no a nivel nacional.

# **CAPÍTULO IV: ANÁLISIS DE LA SOLUCIÓN**

---

## **4.1 INTRODUCCIÓN**

---

A continuación se mencionaran los objetivos generales y específicos del proyecto, el nuevo proceso de negocios que surgirá al implementar el nuevo sistema y la determinación de requerimientos de este.

---

## **4.2 TÍTULO QUE IDENTIFICA EL PROYECTO**

---

Se definió el siguiente nombre como Título del Proyecto:

“SISTEMA PARA SERVICIO DE ATENCIÓN PRIMARIA DE URGENCIA (SAPU) Y TRIAGE (PRIORIZACIÓN DE LA ATENCIÓN) PARA LA PLATAFORMA VIRTUAL DEL CESFAM VIOLETA PARRA”

---

## **4.3 OBJETIVOS GENERALES Y ESPECIFICOS DEL PROYECTO**

---

### **4.3.1 Objetivo general**

Implementar una mejora en el Servicio de Atención Primaria de Urgencia (SAPU) y en el manejo de los datos para el llenado del REM.

### **4.3.2 Objetivos específicos**

- Mejorar los tiempos de ingreso de los datos de cada atención realizada en el SAPU.
- Dar información a los usuarios del servicio acerca del orden de atención.

---

## 4.4 PROCESOS DE NEGOCIOS FUTUROS

---

Al analizar los procesos de negocios asociados que existe actualmente al servicio SAPU nos podemos dar cuenta de algunos cambios que sufrirán con la implementación del nuevo sistema, a continuación se explicaran los nuevos procesos de negocios y se mencionaran las diferencias que existirán con respecto a los procesos anteriores.

Cada profesional que trabaja en el SAPU llena parte de cada hoja diaria según corresponda por cada uno de los pacientes que asisten a buscar el servicio diariamente. Cinco días antes que culmine el mes el estadístico genera el REM correspondiente a la atención SAPU, una vez realizado esto el estadístico debe dirigirse al Servicio de Salud Ñuble para entregar el REM. El servicio de Salud debe realizar un conglomerado de todos los reportes mensuales de todos los centros de salud y hospitales de la provincia de Ñuble y se envía al MINSAL. A partir de esto FONASA, que depende del ministerio envía su aporte económico a cada centro de salud y hospitales que es dinero correspondiente a las prestaciones institucionales.

Podemos darnos cuenta que gracias a este nuevo proceso de negocios se disminuyen los pasos que se deben realizar, ya no será necesario realizar una revisión de cumplir con los campos mínimos necesarios de cada hoja SAPU y su posterior digitalización lo cual consume una gran cantidad de tiempo puesto que esto se realizara de manera automática en al mismo tiempo que se realiza la atención SAPU (Completa Hoja SAPU).

El nuevo diagrama de procesos de negocios se detalla en la Figura 9: Nuevo proceso de negocios..



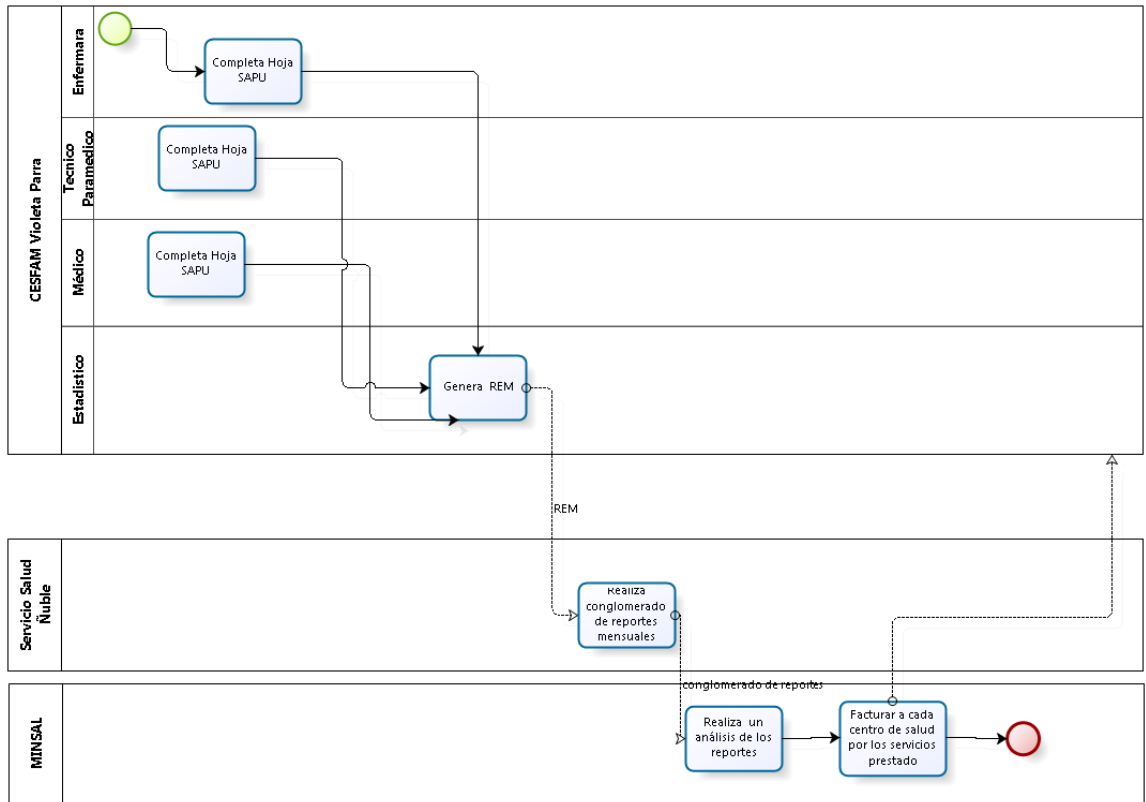


Figura 9: Nuevo proceso de negocios.

---

## **4.5 DESCRIPCIÓN DE LOS ASPECTOS FUNDAMENTALES DE LA METODOLOGIA A UTILIZAR**

---

En este proyecto, se utilizará el modelo de desarrollo incremental junto con el paradigma de desarrollo orientado a objetos, el lenguaje BPMN para modelar el negocio y el lenguaje UML para el modelado del sistema. El motivo por el cual se utiliza este modelo iterativo – incremental es tener productos tangibles en cada iteración y así mostrar el grado de avance al cliente.

### **4.5.1 Primer incremento**

En la primera iteración se implementara el módulo técnico paramédico, realizando todas las funcionalidades que se encuentran definidas para él.

Las funcionalidades del módulo se encuentran especificadas en el apartado 1.4.1 Módulo técnico paramédico. del Anexo A.

### **4.5.2 Segundo incremento**

En la segunda iteración se implementara el módulo Enferma realizando todas las funcionalidades que se encuentran definidas para él.

Las funcionalidades del módulo se encuentran especificadas en el apartado 1.4.2 Módulo enfermera. del Anexo A.

### **4.5.3 Tercer incremento**

En la tercera iteración se implementara el módulo Medico realizando todas las funcionalidades que se encuentran definidas para él.

Las funcionalidades del módulo se encuentran especificadas en el apartado 1.4.3 Módulo Médico. del Anexo A.

#### **4.5.4 Cuarto incremento**

En la cuarta iteración se implementara el modulo Portal y Estadístico realizando todas las funcionalidades que se encuentran definidas para él.

Las funcionalidades de los módulos se encuentran especificadas en los apartados 1.4.5 Módulo Portal y 1.4.5 Módulo Portal del Anexo A.

---

## **4.6 DETERMINACIÓN DE REQUERIMIENTOS DEL PROYECTO**

---

Los requerimientos son fundamentales para lograr una solución que cumpla con las exigencias del usuario. En los siguientes puntos se presentan los requerimientos para esta aplicación. El detalle de los requerimientos se encuentra en el Anexo A.

#### **4.6.1 Requerimientos no funcionales.**

Aspectos del sistema visibles para el usuario, que no están relacionados en forma directa con el comportamiento funcional del sistema.

- Se consideran tres interfaces distintas para los distintos perfiles.
- Al seguir los mismos estándares aplicados a la plataforma virtual, el módulo deberá presentar una interfaz sencilla, con menús y pantallas claras que permitan un rápido y sencillo tránsito a través de la aplicación.
- Los tiempos de respuesta del módulo no deberán ser superiores a 30 segundos al realizar algún tipo de búsqueda.

# **CAPÍTULO V: ANÁLISIS DE LA SOLUCIÓN PRIMER INCREMENTO**

---

## 5.1 INTRODUCCIÓN

---

En el presente capítulo se especifica el primer incremento realizado en el transcurso del desarrollo del proyecto, los requerimientos que se capturaron y el análisis y creación de entidades en la base de datos.

---

## 5.2 REQUERIMIENTOS FUNCIONALES

---

Los requerimientos se clasifican en las siguientes categorías según Craig Larman:

**Evidente:** Debe realizarse, y el usuario debe saber qué está realizando.

**Oculto:** Debe realizarse, aunque no es visible para los usuarios. Esto se aplica a muchos servicios técnicos subyacentes, como guardar información de un mecanismo persistente de almacenamiento. Las funciones ocultas a menudo se omiten (erróneamente) durante el proceso de obtención de requerimientos.

**Superfluo:** opcionales, su inclusión no repercute significativamente en el costo ni en otras funciones.

En la Tabla 2, Tabla 3, Tabla 4, Tabla 5 y Tabla 6 se detallan los requerimientos funcionales del primer incremento según categorías.

Gestión Técnico Paramédico		
Ref. #	Función	Categoría
R 1	Login	Evidente.
R 2	Buscar paciente por folio	Evidente.
R 3	Mostrar datos paciente	Evidente.
R 4	Ingresar datos del paciente	Evidente.

**Tabla 2 Requerimientos Funcionales del Primer Incremento, Funciones Técnico**

**Paramédico**

<b>Función “Login”</b>		
Ref. #	Función	Categoría
R 1.1	Permite ingresar el sistema previamente ingresado un usuario y contraseña	Evidente.

**Tabla 3 Función “Login”**

<b>Función “Buscar paciente por folio”</b>		
Ref. #	Función	Categoría
R 2.1	Permitir la selección de un paciente para realizar la atención.	Evidente.

**Tabla 4 Función “Buscar paciente por folio”**

<b>Función “Mostrar datos paciente”</b>		
Ref. #	Función	Categoría
R 3.1	Permite ver los datos del paciente ingresados anteriormente por SOME.	Evidente.

**Tabla 5 Función “Mostrar datos paciente”**

<b>Función “Ingresar datos paciente”</b>		
Ref. #	Función	Categoría
R 4.1	Podrá ingresar los datos que le corresponden de la hoja SAPU.	Evidente.

**Tabla 6 Función “Ingresar datos paciente”**

### 5.3 REQUERIMIENTOS NO FUNCIONALES

En la Tabla 7 se presentan los requerimientos no funcionales del sistema.

Atributos	Detalles y restricciones
Tiempos de respuesta	<p>El acceso al módulo, mediante el ingreso del rut y contraseña, no debe tardar más de 10 segundos.</p> <p>La respuesta a cualquier consulta o búsqueda, no debe tardar más de 30 segundos.</p> <p>Realizar una inserción, edición o eliminación, no debe tardar más de 15 segundos.</p> <p>La generación de informes o reportes, no debe tardar más de 15 segundos.</p>
Sistema Operativo	Windows XP o superior.
Metáfora de Interfaz	Metáfora Organizacional: Se fundamenta en la estructura existente de un grupo, sistema u organización.
Especificación Interfaz	<p>Se deben utilizar los estándares aplicados a la plataforma virtual.</p> <p>El módulo debe presentar una interfaz sencilla, con menús y pantallas claras que permitan un rápido y sencillo tránsito a través de la aplicación.</p>

**Tabla 7** Requerimientos no funcionales

## 5.4 CASOS DE USO

Los diagramas de casos de uso con su respectiva especificación se encuentran de manera detalla en el Anexo B.

## 5.5 ESPECIFICACIÓN DE MODULO

N° Módulo: 1		Nombre Módulo: Técnico paramédico	
Parámetros de entrada		Parámetros de Salida	
Nombre: Usuario	Tipo de dato: String	Nombre: nombre usuario	Tipo de dato: String
Nombre: password	Tipo de dato: String	Nombre: rut	Tipo de dato: String
Nombre: hora de ingreso	Tipo de dato: Time	Nombre: número de ficha	Tipo de dato: String
Nombre: acompañamiento	Tipo de dato: String	Nombre: folio	Tipo de dato: String
Nombre: traslado en silla de ruedas	Tipo de dato: String	Nombre: fono	Tipo de dato: String
Nombre: prestación y procedimiento	Tipo de dato: String	Nombre: fecha actual	Tipo de dato: String
		Nombre: hora ingreso	Tipo de dato: String
		Nombre: fecha nacimiento	Tipo de dato: String
		Nombre: consultorio procedencia	Tipo de dato: String
		Nombre: edad	Tipo de dato: String
		Nombre: sexo	Tipo de dato: String
		Nombre: previsión	Tipo de dato: String



## 5.6 MODELO DE DATOS

En la siguiente figura se muestra un modelo entidad relación que muestra las entidades que participan en la hoja SAPU con sus respectivas relaciones.

Se encuentra destacada con un borde más grueso la nueva entidad que fue añadida para la realización de este sistemas las demás entidades están previamente definidas.

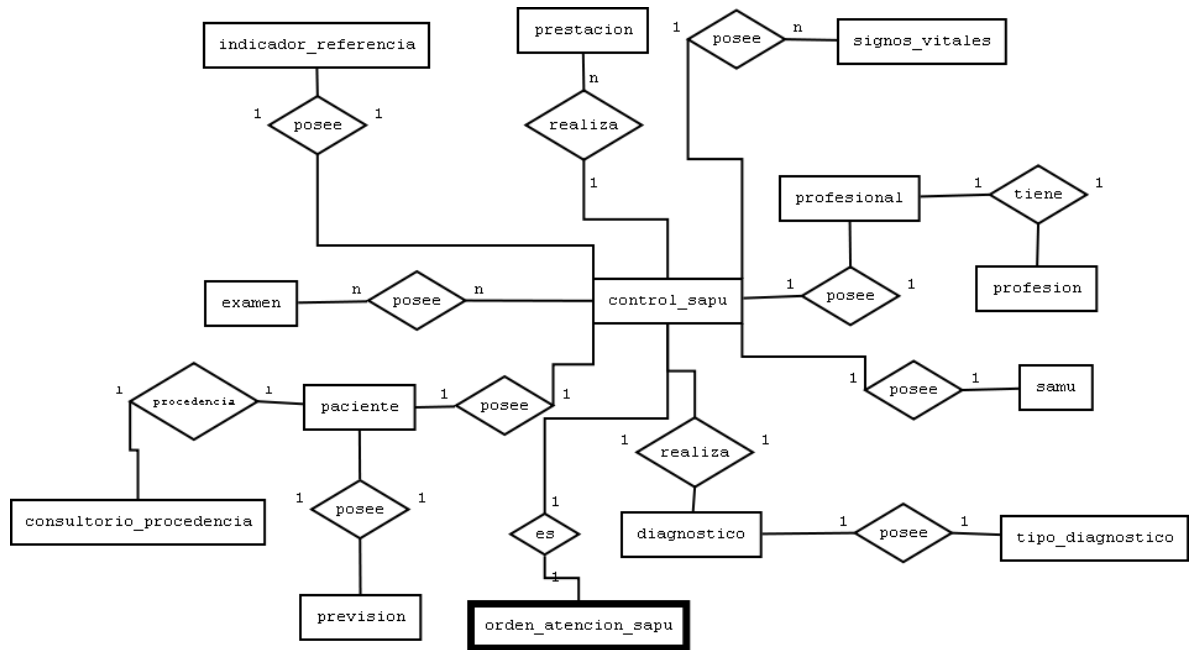


Figura 10: Modelo Entidad Relación.

## 5.7 DISEÑO FÍSICO DE LA BASE DE DATOS

En la siguiente Figura 11 Porción de base de datos a utilizar se muestra la estructura de la base de datos que se utilizara para el desarrollo del software.

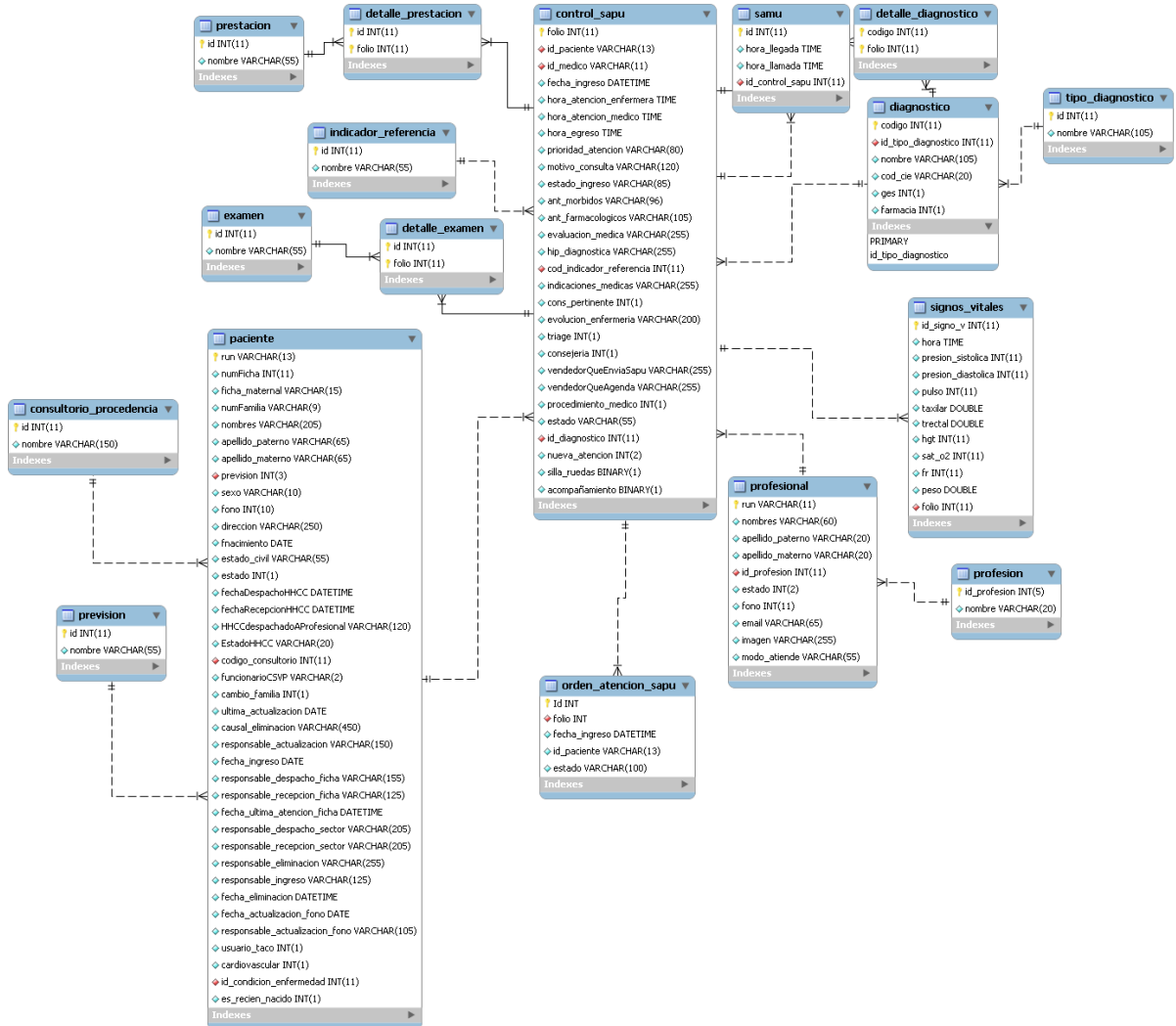


Figura 11 Porción de base de datos a utilizar

La base de datos que se utilizara para el sistema esta previamente especificada, pero cabe mencionar que se realizaron pequeñas modificaciones.

Se crea una nueva tabla llamada orden\_atencion\_sapu la cual tiene los siguientes atributos:

- id (int 11)
- folio (int 11)
- fecha\_ingreso (datetime)
- id\_paciente (varchar 13)
- estado (varchar 100)

Se agregaron columnas a la tabla control SAPU, estas columnas son las siguientes:

- silla\_ruedas (int 1)
- acompañamiento (int 1)
- accidente (varchar 100)
- diarrea (int 1)
- influenza (int 1)
- neumonia (int 1)
- varicela (int 1)
- tr\_ocular (int 1)
- parto\_prem (int 1)
- iam (int 1)
- ave (int 1)
- crisis\_hta (int 1)
- arritmia\_severa (int 1)
- dm\_desc (int 1)
- envenenamiento (int 1)
- intoxicación (int 1)

Se asoció un trigger llamado trigger\_orden\_atencion\_sapu a la tabla control\_sapu el cual después de la inserción de una nueva tupla completa los datos de la tabla orden\_atencion\_sapu, que comparte con ella misma, los datos que completa y obtiene de ella misma son los siguientes:

- folio
- fecha\_ingreso
- prioridad\_atencion
- estado

# **CAPÍTULO VI: ANÁLISIS DE LA SOLUCIÓN SEGUNDO INCREMENTO**

---

## 6.1 INTRODUCCIÓN

---

A continuación se especifica los requerimientos capturados en la realización del segundo incremento.

---

## 6.2 REQUERIMIENTOS FUNCIONALES

---

En la Tabla 8 se muestran las funciones de la enfermera.

<b>Gestión Enfermera</b>		
Ref. #	Función	Categoría
R 4	Login	Evidente.
R 5	Buscar paciente por folio	Evidente.
R 6	Mostrar datos paciente	Evidente.
R 7	Ingresar datos del paciente	Evidente.

**Tabla 8** Requerimientos Funcionales del Segundo Incremento, Funciones Enfermera

En la Tabla 9, Tabla 10, Tabla 11 y Tabla 12 se detallan las funciones de la Tabla 8.

<b>Función "Login"</b>		
Ref. #	Función	Categoría
R 4.1	Permite ingresar el sistema previamente ingresado un usuario y contraseña	Evidente.

**Tabla 9** Función "Login"

<b>Función “Buscar paciente por folio”</b>		
Ref. #	Función	Categoría
R 5.1	Permitir la selección de un paciente para realizar la atención.	Evidente.

**Tabla 10** Función “Buscar paciente por folio”

<b>Función “Mostrar datos paciente”</b>		
Ref. #	Función	Categoría
R 6.1	Permite ver los datos del paciente ingresados anteriormente por SOME.	Evidente.

**Tabla 11** Función “Mostrar datos paciente”

<b>Función “Ingresar datos paciente”</b>		
Ref. #	Función	Categoría
R 7.1	Podrá ingresar los datos que le corresponden de la hoja SAPU.	Evidente.

**Tabla 12** Función “Ingresar datos paciente”

## 6.3 ESPECIFICACIÓN DE MODULO

N° Módulo: 2		Nombre Módulo: Enfermera	
Parámetros de entrada		Parámetros de Salida	
Nombre: Usuario	Tipo de dato: String	Nombre: nombre usuario	Tipo de dato: String
Nombre: password	Tipo de dato: String	Nombre: rut	Tipo de dato: String
Nombre: hora atención enfermera	Tipo de dato: Time	Nombre: número de ficha	Tipo de dato: String
Nombre: motivo consulta	Tipo de dato: String	Nombre: folio	Tipo de dato: String
Nombre: prioridad de atención	Tipo de dato: String	Nombre: fono	Tipo de dato: String
Nombre: estado de ingreso	Tipo de dato: String	Nombre: fecha actual	Tipo de dato: String
Nombre: control de signos vitales	Tipo de dato: String	Nombre: hora ingreso	Tipo de dato: String
Nombre: antecedentes mórbidos	Tipo de dato: String	Nombre: fecha nacimiento	Tipo de dato: String
Nombre: antecedentes farmacológicos	Tipo de dato: String	Nombre: consultorio procedencia	Tipo de dato: String
Nombre: evaluación de enfermera	Tipo de dato: String	Nombre: edad	Tipo de dato: String
		Nombre: sexo	Tipo de dato: String
		Nombre: previsión	Tipo de dato: String

# **CAPÍTULO VII: ANÁLISIS DE LA SOLUCIÓN TERCER INCREMENTO**



---

## 7.1 INTRODUCCIÓN

---

A continuación se especifica los requerimientos capturados en la realización del tercer incremento

---

## 7.2 REQUERIMIENTOS FUNCIONALES

---

En la Tabla 13 se muestran las funciones del médico.

<b>Gestión Médico</b>		
Ref. #	Función	Categoría
R 4	Login	Evidente.
R 5	Buscar paciente por folio	Evidente.
R 6	Mostrar datos paciente	Evidente.
R 7	Ingresar datos del paciente	Evidente.

**Tabla 13** Requerimientos Funcionales del Tercer Incremento, Funciones Médico

En la Tabla 14, Tabla 10, Tabla 16 y Tabla 17 se detallan las funciones de la Tabla 13.

<b>Función "Login"</b>		
Ref. #	Función	Categoría
R 4.1	Permite ingresar el sistema previamente ingresado un usuario y contraseña	Evidente.

**Tabla 14** Función "Login"

<b>Función “Buscar paciente por folio”</b>		
Ref. #	Función	Categoría
R 5.1	Permitir la selección de un paciente para realizar la atención.	Evidente.

**Tabla 15** Función “Buscar paciente por folio”

<b>Función “Mostrar datos paciente”</b>		
Ref. #	Función	Categoría
R 6.1	Permite ver los datos del paciente ingresados anteriormente por SOME.	Evidente.

**Tabla 16** Función “Mostrar datos paciente”

<b>Función “Ingresar datos paciente”</b>		
Ref. #	Función	Categoría
R 7.1	Podrá ingresar los datos que le corresponden de la hoja SAPU.	Evidente.

**Tabla 17** Función “Ingresar datos paciente”

## 7.3 ESPECIFICACIÓN DE MODULO

N° Módulo: 3		Nombre Módulo: Médico	
Parámetros de entrada		Parámetros de Salida	
Nombre: Usuario	Tipo de dato: String	Nombre: nombre usuario	Tipo de dato: String
Nombre: password	Tipo de dato: String	Nombre: rut	Tipo de dato: String
Nombre: hora atención Médico	Tipo de dato: Time	Nombre: número de ficha	Tipo de dato: String
Nombre: patología centinela	Tipo de dato: String	Nombre: folio	Tipo de dato: String
Nombre: auge	Tipo de dato: String	Nombre: fono	Tipo de dato: String
Nombre: causas sistema circulatorio	Tipo de dato: String	Nombre: fecha actual	Tipo de dato: String
Nombre: accidentes	Tipo de dato: String	Nombre: hora ingreso	Tipo de dato: String
Nombre: demás causas hipótesis diag	Tipo de dato: String	Nombre: fecha nacimiento	Tipo de dato: String
Nombre: indicaciones de referencia	Tipo de dato: String	Nombre: consultorio procedencia	Tipo de dato: String
Nombre: samu	Tipo de dato: String	Nombre: edad	Tipo de dato: String
Nombre: Consulta pertinente	Tipo de dato: String	Nombre: sexo	Tipo de dato: String
Nombre: procedimiento Médico	Tipo de dato: String	Nombre: previsión	Tipo de dato: String
Nombre: indicaciones	Tipo de dato: String	Nombre: acompañamiento paciente	Tipo de dato: String
Nombre: exámenes solicitados	Tipo de dato: String	Nombre: traslado silla ruedas	Tipo de dato: String
		Nombre: motivo consulta	Tipo de dato: String
		Nombre: prioridad atención	Tipo de dato: String
		Nombre: estado de ingreso	Tipo de dato: String
		Nombre: signos vitales	Tipo de dato: String

# **CAPÍTULO VIII: ANÁLISIS DE LA SOLUCIÓN CUARTO INCREMENTO**

---

## 8.1 INTRODUCCIÓN

---

A continuación se especifica los requerimientos capturados en la realización del cuarto incremento.

---

## 8.2 REQUERIMIENTOS FUNCIONALES

---

En la Tabla 18 se muestran las funciones del estadista y en la Tabla 19 se muestran las funciones del portal.

<b>Gestión Estadista</b>		
Ref. #	Función	Categoría
R 8	Login	Evidente.
R 9	Genera REM	Evidente.

**Tabla 18** Requerimientos Funcionales del Cuarto Incremento, Funciones Estadista

<b>Gestión Portal</b>		
Ref. #	Función	Categoría
R 10	Muestra lista de atención	Evidente.

**Tabla 19** Requerimiento Funcionales del Cuarto Incremento, Funciones Portal

En la Tabla 20 y Tabla 21 se detallan las funciones de la Tabla 18.

<b>Función “Login”</b>		
Ref. #	Función	Categoría
R 8.1	Permite ingresar el sistema previamente ingresado un usuario y contraseña	Evidente.

**Tabla 20** Función “Login”

<b>Función “Genera REM”</b>		
Ref. #	Función	Categoría
R 9.1	Al ingresar un rango de fechas (fecha inicio y fecha termino) se generara en formato PDF la sección D del REM08	Evidente.

**Tabla 21** Función “Buscar paciente por folio”

En la Tabla 22 se detallan las funciones de la Tabla 19.

<b>Función “Genera REM”</b>		
Ref. #	Función	Categoría
R 10.1	Se muestra la lista de los pacientes que esperan atención en el orden que se realizaran estas.	Evidente.

**Tabla 22** Función “Muestra lista de atención”

## 8.3 ESPECIFICACIÓN DE MODULO

<b>Nº Módulo: 4</b>		<b>Nombre Módulo: Estadístico</b>	
<b>Parámetros de entrada</b>		<b>Parámetros de Salida</b>	
Nombre: Usuario	Tipo de dato: String	Nombre: Informe REM	Tipo de dato: PDF
Nombre: password	Tipo de dato: String		
Nombre: fecha de inicio	Tipo de dato: Time		
Nombre: fecha termino	Tipo de dato: Time		

<b>Nº Módulo: 5</b>		<b>Nombre Módulo: Portal</b>	
<b>Parámetros de entrada</b>		<b>Parámetros de Salida</b>	
		Nombre: nombre usuario	Tipo de dato: String
		Nombre: apellido paterno	Tipo de dato: String
		Nombre: apellido materno	Tipo de dato: String
		Nombre: orden atencion	Tipo de dato: String

# **CAPÍTULO IX: FACTIBILIDAD DEL PROYECTO**



---

## 9.1 INTRODUCCIÓN

---

En este capítulo se analiza tanto la factibilidad técnica, operativa y económica que posee el proyecto.

---

## 9.2 FACTIBILIDAD

---

### 9.2.1 Factibilidad técnica.

Para poner en marcha este proyecto se necesita hardware y software específico el cual se especifica en la Tabla 2.

Hardware / Software
Servidor de aplicaciones
Servidor de base de datos
Red de área local
Computadores
Apache tomcat
MySQL

**Tabla 23: Hardware y software necesario.**

El CESFAM Violeta Parra posee un servidor que cumple las funciones de servidor de aplicaciones y servidor de base de datos. Este servidor está encargado de soportar cuatro aplicaciones hasta hoy las cuales son:

- Reserva SOME
- Sistema Farmacia
- Sistema Leche
- Hoja odontológica

En consecuencia por lo anteriormente mencionado queda en evidencia que se satisface una de las necesidades de servidores.

Cabe destacar que este servidor tiene instalado el software necesario para soportar las aplicaciones web que están en existencia hoy en día, el cual es el mismo que se requiere para el correcto funcionamiento de este nuevo sistema.

Además es de relevancia mencionar que el CESFAM Violeta Parra consta con una red de área local que abarca a la totalidad de las instalaciones de este, lo cual no limitaría la comunicación de los clientes (Computadores del servicio) con el servidor.

Por todo lo anteriormente dicho se llega a la conclusión que la factibilidad técnica es favorable en este proyecto ya que se consta con todo lo necesario para un buen desarrollo y posterior implementación del sistema.

### **9.2.2 Factibilidad operativa.**

La implantación de este nuevo sistema tendrá un impacto en la cantidad de personas que hoy conforman el SAPU como se mencionó en la descripción de la empresa SAPU consta con 35 personas que equivalen a 280 horas/hombre diaria, y después de la puesta en marcha del sistema solo se necesitara contar con 248 horas/hombre, se ahorran 32 horas/hombre las cuales podrán ser utilizadas para cumplir otras labores dentro del mismo CESFAM.

Los tiempos de realización de la sección D del REM 08 bajara considerablemente, normalmente se trabaja durante todo el mes para tener los resultados a fin de este, gracias a este nuevo sistema solo se tendrá que hacer la consulta el mismo día que se quiera obtener los resultados y se tendrán en menos de 5 minutos.

La atención deberá tomar un orden lógico en la realización de los pasos que se deben realizar, este orden será consensuado entre los profesionales, esto puede producir molestias entre los profesionales ya que deberán cambiar su habitual orden de atención.

### 9.2.3 Factibilidad económica.

#### 9.2.3.1 Costos de implementación e inversión.

Al ya existir todos los recursos técnicos que son necesarios para poner en marcha el sistema no es necesario adquirir nuevos equipos y/o herramientas para la implementación del sistema por lo cual no se debe incurrir en gastos.

Para desarrollar la aplicación por completo se necesitara contratar un diseñador gráfico y un analista programador al realizar estas contrataciones se incurrirá en los siguientes gastos:

Asumiendo que el diseñador gráfico deberá trabajar durante 20 días 8 horas diarias cobrando \$3.500 por hora y el analista programador 40 días 8 horas diarias cobrando \$4.000 por hora la tabla de costo nos quedara de la siguiente manera.

<b>Personal</b>		
<b>Cargo</b>	<b>Número de horas</b>	<b>Total Pesos</b>
Diseñador Gráfico	160	\$560.000
Analista Programador	320	\$1.280.000

**Tabla 24: Costo de personal**

Estos gastos se ven reflejados como inversión en el año cero.

#### 9.2.3.2 Estimación de ingresos o beneficios.

Al tener disponible 32 horas/hombre en el servicio se produce un ahorro de para el CESFAM ya que esas horas las podrán destinar a realizar otras actividades.

Asumiendo que la hora de trabajo tiene un costo de \$1.000 se tendría un ahorro de 32 horas/hombre x 20 días del mes x12 meses = 15360 horas en consecuencia \$7.680.000 anual.

### 9.2.3.3 Determinación de flujos netos de caja.

Para determinar la factibilidad económica del proyecto se utilizara el indicador Valor Actual Neto (VAN), que permite obtener valor de decisión frente al costo de ésta.

Este análisis tendrá las siguientes consideraciones:

- Se estima un tiempo de vida útil del proyecto de 4 años.
- Se sabe que el consultorio por ser una institución pública, no paga un impuesto al gobierno.
- El proyecto se evaluará a una tasa de descuento del 12%, debido a que con esta tasa se evalúan la mayoría de los proyectos de esta índole.

El cálculo del VAN se dará con la siguiente formula:

$$\sum_{i=1}^n \frac{FC_i}{(1 - K)^i} - I_0$$

Dónde: n, es el total de años de vida útil del proyecto, en este caso 4,

i, representa el año correspondiente,

FC<sub>i</sub>, son cada uno de los flujos netos de caja,

K, es la tasa de interés, en este caso 12%,

I<sub>0</sub>, es la inversión inicial.

Por ende tenemos

$$VAN (12\%) = \frac{7.680.000}{(1 + 0.12)^1} + \frac{7.680.000}{(1 + 0.12)^2} + \frac{7.680.000}{(1 + 0.12)^3} + \frac{7.680.000}{(1 + 0.12)^4} - 1.840.000$$

$$VAN (12\%) = 21.486.842$$

Como conclusión del estudio de la factibilidad económica, se puede ver que el cálculo del VAN, nos da un valor positivo, lo que nos dice que se alcanza a recupera la inversión en los 4 años de vida útil estimados del proyecto. Por ende se deduce que la realización del proyecto es factible económicamente.

#### **9.2.4 Conclusión de la factibilidad**

Por lo expuesto en los de factibilidad técnica, operativa y económica se puede ver claramente que es un proyecto factible, tanto en la realización como implantación de este.

# **CAPÍTULO X: PLAN DE PRUEBAS**

---

## 10 PLAN DE PRUEBAS

---

### 10.1 Propósito

Este documento describe el Plan de pruebas para el sistema de hoja SAPU que se integrará a la Plataforma Virtual del CESFAM Violeta Parra. Define los siguientes objetivos específicos:

- Identifica los elementos que se van a probar.
- Describe la estrategia de pruebas que se va a seguir en el proceso de prueba.
- Identifica los recursos necesarios para llevar a cabo el proceso de prueba y estima los esfuerzos que conlleva.
- Lista los resultados que se obtienen de las actividades de prueba.

### 10.2 Ámbito

Este Plan de Pruebas describe las pruebas de unidad, integración y del sistema que se aplicarán al sistema software desarrollado.

El objetivo es probar todos los requisitos definidos en la especificación de requisitos y en el modelo de casos de uso.

### 10.3 Requerimientos de las pruebas

La lista que se proporciona en esta sección identifica los elementos (casos de uso, requisitos funcionales y requisitos no funcionales) que son objetivos de las pruebas. Es decir, los elementos que se van a probar.

#### 10.3.1 Pruebas de integridad de los datos:

- Verificar la recuperación correcta de las modificaciones realizadas en la base de datos.
- Verificar accesos simultáneos de lectura de datos.

Pruebas de integridad de los datos.

<b>Objetivos de la prueba</b>	Comprobar que los procedimientos y métodos de acceso a los datos funcionan correctamente.
<b>Técnicas</b>	Invocar cada procedimiento o método de acceso a los datos con entradas válidas e inválidas. Inspeccionar los datos para asegurar que estos son los previstos
<b>Criterios de finalización</b>	Todos los procedimientos y métodos de acceso funcionan como se diseñaron y sin ningún error en los datos.

**10.3.2 Pruebas de funcionalidad:**

- Verificar el caso de uso Autenticar
- Verificar el caso de uso Ingresa datos de la hoja SAPU
- Verificar el caso de uso Lista datos ingresados anteriormente
- Verificar el caso de uso Ingresa datos de discriminación
- Verificar el caso de uso Genera informes

Las pruebas de funcionalidad se deberían centrar en cualquier requisito que pueda ser trazado directamente de los casos de uso y reglas de negocio. El objetivo de estas pruebas es verificar la aceptación, procesamiento y recuperación de datos y la adecuada implementación de las reglas de negocio. Este tipo de pruebas están basadas en técnicas de caja negra, es decir, verificar la aplicación interaccionando a través de las interfaces de usuario y analizando los resultados.



Objetivos de la prueba	Asegurar la navegación correcta de la aplicación, la entrada de datos, su procesamiento y recuperación.
Técnicas	Ejecutar cada caso de uso y flujo del caso de uso con datos válidos e inválidos para verificar lo siguiente: <ul style="list-style-type: none"> <li>• Cuando se utilizan datos correctos se obtienen los resultados esperados.</li> <li>• Cuando se utilizan datos incorrectos se obtienen los mensajes de error o advertencias adecuadas.</li> <li>• Cada regla de negocio se ha aplicado correctamente.</li> </ul>
Criterios de finalización	Todas las pruebas planificadas se han ejecutado. Todos los defectos identificados se han considerado.
Consideraciones	Ninguna.

### 10.3.3 Pruebas de interfaz de usuario:

- Verificar que la navegación a través de un conjunto de pantallas es rápida.
- Navegar a través de todos los casos de uso, verificando que cada interfaz de usuario se comprenda en su totalidad.

Las pruebas de interfaz de usuario verifican la interacción del usuario con el sistema software. El objetivo de esta prueba es asegurar que la interfaz de usuario permite al usuario acceder y navegar a través de toda la funcionalidad de la aplicación.

Objetivos de la prueba	<p>Verificar los siguientes objetivos:</p> <ul style="list-style-type: none"> <li>• La navegación a través de la aplicación refleja adecuadamente las reglas de negocio y los requisitos incluyendo ventana a ventana, campo a campo y métodos de acceso (tabulador, movimientos del ratón y teclas de función).</li> <li>• Las ventanas y sus características, como menús, tamaño, posición y estado cumplen los estándares.</li> </ul>
Técnicas	<p>Crear o modificar pruebas para cada ventana con el objetivo de verificar la correcta navegación y su estado.</p>
Criterios de finalización	<p>Cada ventana se ha verificado con éxito y es consistente con la versión de referencia o con los estándares utilizados.</p>
Consideraciones	<p>Ninguna.</p>

## 10.4 Estrategia de Prueba

En esta sección presentamos el enfoque que vamos a utilizar para probar el sistema software. En la sección anterior hemos descrito qué elementos del sistema software vamos a probar, y en esta sección se define cómo se realizarán las pruebas.

### 10.4.1 Herramientas

Las siguientes herramientas se usarán para llevar a cabo el proceso de prueba:

Tipo de Prueba	Herramienta
Herramienta DBMS	MySQL
Interfaz de usuario	Mozilla Firefox

## 10.4.2 Recursos

En esta sección describimos los recursos necesarios para realizar el proceso de prueba, sus principales responsabilidades y características.

### 10.4.2.1 Recursos hardware

Recurso	Cantidad	Nombre y Tipo
PC	1	Diseño de las pruebas
PC	1	Ejecución de las pruebas

### 10.4.2.2 Recursos software

Nombre del elemento software	Tipo y otras notas
Mozilla Firefox	Interfaz de usuario
Microsoft Word	Registro de resultados
MySQL	Gestor de Base de datos
Adobe Reader	Lector de documentos en formato PDF para los reportes que genere el sistema.

### 10.4.2.3 Recursos humanos

RECURSOS HUMANOS		
Rol	Mínimos recursos recomendados	Responsabilidades específicas o comentarios
Gestor de prueba	1	<p>Proporcionar una gestión adecuada. Responsabilidades:</p> <ul style="list-style-type: none"> <li>• Proporcionar una dirección técnica.</li> <li>• Adquirir los recursos apropiados.</li> <li>• Informar de la gestión.</li> </ul>
Diseñador de prueba	1	<p>Identificar, priorizar e implementar los casos de prueba.</p> <p>Responsabilidades:</p> <ul style="list-style-type: none"> <li>• Generar el Plan de pruebas.</li> <li>• Diseñar los Casos de prueba.</li> <li>• Evaluar el esfuerzo de prueba.</li> </ul>
Probador (Tester)	1	<p>Ejecutar las pruebas.</p> <p>Responsabilidades:</p> <ul style="list-style-type: none"> <li>• Ejecutar pruebas.</li> <li>• Recuperar los errores.</li> <li>• Documentar los defectos.</li> </ul>

## 10.5 Casos de prueba

Caso de Prueba de Autenticar	
Tipo de prueba: Integridad de los datos/ Funcionalidad	
Código de Prueba: 1	Código Caso de Uso: 1.
<p>Descripción de la Prueba:</p> <p>Esta Prueba intentará realizar ingresos al sistema con distintos usuarios mostrando las opciones que posee asociado cada usuario, además de impedir el acceso a usuarios no registrados.</p>	
<p>Condiciones de Ejecución:</p> <p>Se debe estar en la página Login del sistema.</p>	
<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> <li>1. Se ingresa un usuario</li> <li>2. Se ingresa una contraseña</li> <li>3. Se selecciona el botón ingresar.</li> </ol>	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> <li>1. Se ingresa al perfil que corresponde al usuario y clave ingresa.</li> <li>2. Si el usuario o la clave no se encuentran en la base de datos o son incorrectos se debe mostrar información haciendo alusión a esto.</li> </ol>	
<p>Evaluación de la Prueba:</p> <p>El sistema no permite el ingreso a usuarios ingresados incorrectamente y que no se encuentran registrados.</p> <p>Permite el ingreso de usuarios registrados y direcciona según el perfil que posee cada uno de manera correcta</p>	

Caso de Prueba de Selecciona paciente atención	
Tipo de prueba: Integridad de los datos/ Funcionalidad	
Código Caso de Prueba: 2	Código Caso de Uso: 2.
<p>Descripción de la Prueba:</p> <p>Esta Prueba intentará seleccionar un paciente para realizar la respectiva atención, desplegando sus datos y los campos que se deben llenar con información.</p>	
<p>Condiciones de Ejecución:</p> <p>Se debe estar autenticado.</p>	
<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> <li>1. Se selecciona al primer paciente que se encuentra en la lista de atenciones.</li> <li>2. Se despliega la información que se encuentra asociada a este hasta ese momento y los campos de información requerida para la atención correspondiente.</li> </ol>	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> <li>1. Los datos asociados hasta ese momento del paciente y los campos de ingreso de información requerida para ese momento de la atención.</li> <li>2. Si el paciente ya fue atendido anteriormente por el mismo profesional que desea ver su información debe desplegar un mensaje asociado a esto.</li> </ol>	
<p>Evaluación de la Prueba:</p> <p>El sistema despliega los datos asociados hasta ese momento del paciente y los campos de ingreso de información.</p> <p>Despliega un mensaje en pantalla si el profesional ya atendió anteriormente al paciente.</p>	

Caso de Prueba de Ingreso Datos de la hoja SAPU.	
Tipo de prueba: Integridad de los datos/ Funcionalidad	
Código Caso de Prueba: 3	Código Caso de Uso: 3.
<p>Descripción de la Prueba:</p> <p>Esta Prueba intentará ingresar los datos requeridos del paciente para cada perfil de los profesionales.</p>	
<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> <li>• Se debe estar autenticado.</li> <li>• Se debe haber seleccionado un paciente para realizar la atención.</li> </ul>	
<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> <li>1. Se deben llenar cada campo con la información requerida y confirmarlo por separado.</li> <li>2. Se debe desplegar un mensaje haciendo alusión al ingreso correcto o incorrecto.</li> </ol>	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> <li>1. La información ingresada debe quedar guardada en la base de datos.</li> <li>2. Si no se ingresa ningún campo de información debe enviar un mensaje asociado a esto.</li> <li>3. Se debe redirigir a la página lista pacientes después de confirmar la atención.</li> </ol>	
<p>Evaluación de la Prueba:</p> <p>La información ingresa se ve reflejada correctamente en la base de datos.</p> <p>El sistema muestra un mensaje cuando se desea confirmar la atención sin haber ingresado ningún dato del paciente.</p> <p>Al confirmar la atención se redirige a la pantalla donde se encuentra la lista con los pacientes que aún falta por atender.</p>	

Caso de Prueba de Imprime Hoja SAPU.	
Tipo de prueba: Integridad de los datos/ Funcionalidad	
Código Caso de Prueba: 4	Código Caso de Uso: 5.
<p>Descripción de la Prueba:</p> <p>Esta Prueba intentará lanzar la hoja SAPU completa con los datos del paciente en formato PDF para imprimir.</p>	
<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> <li>• Se debe estar autenticado.</li> <li>• Se debe haber seleccionado un paciente para realizar la atención.</li> <li>• Se debe haber realizado la atención por todos los participantes de esta.</li> </ul>	
<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> <li>1. Se presiona el botón de imprimir hoja SAPU</li> </ol>	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> <li>1. Hoja SAPU en formato .PDF con todos los datos de la atención del paciente.</li> </ol>	
<p>Evaluación de la Prueba:</p> <p>La prueba tiene resultados satisfactorios ya que la hoja SAPU que es generada por el sistema refleja de manera precisa lo realizado durante la atención del paciente.</p>	



---

## 11 CONCLUSIONES

---

La implementación del nuevo módulo Hoja SAPU en la plataforma virtual del centro de salud familiar violeta parra, provoca tres cambios muy significativos.

En primer lugar, el complemento de la hoja SAPU pasa a ser un procedimiento digital, cambiando lápiz y papel por un computador personal conectado a la intranet que posee el establecimiento, logrando ingresar los datos correspondientes a cada atención SAPU directamente a la base de datos, durante el mismo tiempo que se produce la atención. En segundo lugar, la forma de obtener los datos es más explícita y rápida, ya que el estadístico puede realizar consultas, especificando los parámetros para poder visualizar los resultados a través de un reporte exportable. Finalmente en tercer lugar la manera que se utiliza para dar a conocer el orden de atención que tendrá cada paciente se transparenta a través del portal digital que se mantendrá en la sala de espera disminuyendo la incertidumbre de las personas.

Para la construcción del módulo, en un comienzo se realizó una descripción actual de la organización y diversas reuniones con el jefe de departamento de Informática y la jefa de la unidad SAPU, para la captura de requerimientos. Gracias a esto se logró conocer el funcionamiento de los sistemas de información empleados hasta ese momento. En esa instancia se pudo observar y comprender la necesidad de implementar un módulo que optimizara el tiempo y acceso a la información, puesto que los existentes que estaban relacionados con el módulo mencionado, carecían de eficiencia y los procedimientos demandaban mucho tiempo.

Con respecto a la forma en que se llevó a cabo el desarrollo del proyecto, se optó por utilizar el modelo de desarrollo iterativo-incremental, con un total de cuatro incrementos. El hecho de haber utilizado esta metodología de desarrollo

permitió mostrar al usuario versiones del software previas a la última. Así los usuarios pudieron apreciar avances funcionales del software y así generar retroalimentación para continuar con el desarrollo de éste.

Las ventajas que tendrá la institución con este proyecto serán la reducción de errores en el llenado de la hoja SAPU sin necesidad de que los médicos, enfermeros y técnicos paramédicos pierdan tiempo en recurrir al encargado de estadística para saber cómo deben completar el documento. También implicará una notable reducción del tiempo en cuanto a realizar las consultas necesarias para poder completar el reporte estadístico mensual; así desaparecerán errores en cuanto al conteo que realiza el departamento de estadística para obtener las consultas necesarias.

Las dificultades que se presentaron durante la realización del proyecto fueron, en primer lugar, conocer a cabalidad el funcionamiento de los procesos previos a la implementación del módulo. En segundo lugar, para construir el modelo entidad-relación fue necesario integrar las entidades existentes en la plataforma hasta este minuto.

Por último, mencionar que en este proyecto se pudo poner en práctica los conocimientos obtenidos durante todos estos años de carrera en la universidad, además de adquirir nuevos conocimientos, asociados a aspectos técnicos, durante el desarrollo de este proyecto. Se considera importante el hecho de poder negociar con clientes reales, realizando un trabajo que será un aporte concreto al CESFAM Violeta Parra, lo cual es notable ya que el producto obtenido en este proyecto será utilizado en el futuro.

---

## 12 BIBLIOGRAFÍA

---

**Álvarez, Miguel Ángel.** desarrolloweb.com. [En línea] 2010. Disponible en <<http://www.desarrolloweb.com/articulos/497.php>>[Consultado: 16 diciembre 2011.] .

Introducción a Javascript. [En línea] 2001. Disponible en <<http://www.desarrolloweb.com/articulos/490.php> > [Consultado: 17 diciembre 2011.] .

**Anónimo.** Qué es JSP. [En línea] 2002. Disponible en <<http://www.desarrolloweb.com/articulos/831.php> > [Consultado: 17 diciembre 2012.] .

**Anónimo.** Actualidad JQuery. [En línea] 2010. Disponible en <<http://www.actualidadjquery.es/acerca-de> >[Consultado:: 20 diciembre2011.] .

Ajax: Asynchronous JavaScript And XML. [En línea] 2010. Disponible en <<http://tipsgeeks.blogspot.com/2009/06/ajax-asynchronous-javascript-and-xml.html>.> [Consultado: 20 diciembre 2011.] .

Struts 2. [En línea] 2009. Dispobible en <<http://mundogeek.net/archivos/2009/02/08/struts-2/>> [Consultado: 5 diciembre 2011.] .

worldlingo.com. [En línea] 2010. Dispobible en <<http://www.worldlingo.com/ma/enwiki/es/MyEclipse>> [Consultado 25 diciembre 2011.] .

**Brown, Donald, Davis, Chad Michael y Stanlick, Scott. 2008.** *Struts 2 in Action*. s.l. : Manning, 2008.

**CESFAM Violeta Parra. 2008.** 2008.

**De la Cruz García, Miguel Teodoro. 2006.** Proyecto Fin de Carrera: Sistema de Gestión con Replicación de Datos. [En línea] 2006. Disponible en <<http://www.iit.upcomillas.es/pfc/resumenes/450a66346db72.pdf>> [Consultado: 7 enero 2012.] .

**Herrera, Cristhian. 2005.** Informes en Java con IReport. [En línea] 29 de 04 de 2005. Disponible en <<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ireport>> [Consultado: 7 enero 2012.] .

ign.es. [En línea] 2010. Disponible en <[www.ign.es/ign/resources/acercaDe/aig/B.pdf](http://www.ign.es/ign/resources/acercaDe/aig/B.pdf)> [Consultado: 20 diciembre 2011.] .

**ITEM.** ITEM: Innovación Tecnológica Empresarial. [En línea] 2010. Disponible en <<http://www.item.com.ve/technology.php>> [Consultado: 27 diciembre 2011.] .

**Larman, Craig. 2003.** *UML y Patrones. Una Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado*. s.l. : Prentice Hall, 2003.

**Schmuller, J. 2000.** *Aprendiendo UML en 24 horas*. s.l. : Prentice Hall, 2000.

**The Apache Software Foundation. 2010.** Apache Tomcat. [En línea] 2010. Disponible en <<http://tomcat.apache.org/>> [Consultado: 16 diciembre 2011.].

**Vázquez Rodríguez, Josefina Adriana. 2006.** Creación de un repositorio para programas de Java utilizando la herramienta de ORM Hibernate. [En línea] 16 de 5 de 2006. Disponible en <[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/vazquez\\_r\\_ja/capitulo2.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/vazquez_r_ja/capitulo2.pdf)> [Consultado: 19 diciembre 2011.].

**WORDPRESS.COM. 2010.** MI CURSO JAVA & CO. [En línea] 2010. Disponible en <<http://micursojava.files.wordpress.com/2010/07/uml.pdf>> [Consultado: 17 diciembre 2011.] .

# **ANEXO A: DOCUMENTO DE REQUERIMIENTOS**

---

# 1 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

---

## 1.1 Restricciones

Este nuevo Sistema no abarcara todas las instancias que ocurren durante una atención SAPU, se dejara de lado la inscripción de los pacientes puesto que esta se realiza actualmente con ayuda del sistema SOME y también no contempla para esta primera instancia la entrega de medicamentos ósea no tendrá integración con el Sistema de Farmacia.

Tampoco tendrá interconexión con la red MINSAL ya que será un Sistema propio del consultorio y no a nivel nacional.

La base de datos que se utilizara durante el desarrollo de la aplicación web se encuentra previamente diseñada e implementada y esta no puede sufrir grandes modificaciones.

Los módulos deben de ser implementados bajo las mismas tecnologías con la cual está diseñada la plataforma virtual actualmente existe en el CESFAM.

Las tecnologías son:

- Programación Orientación a Objetos (POO).
- Unified Modeling Language (UML).
- Arquitectura Modelo Vista Controlador (MVC).
- Patron Data Access Object (DAO).
- Tecnologías J2EE (JSP, Struts 2, Hibernate).
- Entorno de programación MyEclipse.
- Servidor de aplicaciones Apache Tomcat.
- Motor de bases de datos MySQL.

## **1.2 Objetivo del software**

El objetivo general de este proyecto es desarrollar una aplicación web que servirá para todo el transcurso de una atención SAPU además, de generación de informes estadísticos y mostrara la priorización de la atención.

## **1.3 Descripción Global del Producto**

### **Interfaz de usuario**

Al seguir los mismos estándares aplicados a la plataforma virtual que existente hoy en día, el módulo deberá presentar una interfaz sencilla, con menús y pantallas claras que permitan un rápido y sencillo tránsito a través de la aplicación. Los colores que se deben utilizar son blanco y una gama de colores que se encuentre entre azul y celeste para los fondos de las distintas pantallas, para las letras que aparecerán en cada módulo serán de color negro para destacarlas.

### **Interfaces de comunicación**

El protocolo de comunicación que se utilizara en la comunicación del sistema es TCP/IP este protocolo es uno de los más populares y utilizados por todo el mundo.

Además, todo el equipamiento que posee el CESFAM y el protocolo de comunicación que utilizan hoy en día en sus demás sistemas es el mismo.

## 1.4 Requerimientos Específicos

### Requerimientos Funcionales del sistema

#### 1.4.1 Módulo técnico paramédico.

Id	Nombre	Descripción
01	Login	El técnico paramédico deberá tener un sistema de login con un usuario y una password.
02	Busca por folio	Podrá buscar los datos de un paciente por el folio de la hoja SAPU que corresponda
03	Mostrar datos paciente	<p>Podrá ver los datos del paciente que están previamente ingresados por el SOME</p> <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Rut</li> <li>• Sexo</li> <li>• N° Ficha</li> <li>• Fecha Nacimiento</li> <li>• Folio</li> <li>• Teléfono</li> <li>• Edad</li> <li>• Fecha Ingreso</li> <li>• Dirección</li> <li>• Procedencia</li> <li>• Previsión</li> </ul>
04	Ingresar datos del paciente	<p>Podrá ingresar los datos que le corresponden de la hoja SAPU estos datos son:</p> <ul style="list-style-type: none"> <li>• Hora de egreso</li> <li>• Acompañamiento del paciente</li> <li>• Traslado en silla de ruedas</li> <li>• Prestaciones y procedimientos</li> </ul>



### 1.4.2 Módulo enfermera.

Id	Nombre	Descripción
01	Login	La enfermera deberá tener un sistema de login con un usuario y una password.
02	Busca por folio	Podrá buscar los datos de un paciente por el folio de la hoja SAPU que corresponda
03	Mostrar datos paciente	<p>Podrá ver los datos del paciente que están previamente ingresados por el SOME</p> <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Rut</li> <li>• Sexo</li> <li>• N° Ficha</li> <li>• Fecha Nacimiento</li> <li>• Folio</li> <li>• Teléfono</li> <li>• Edad</li> <li>• Fecha Ingreso</li> <li>• Hora Ingreso</li> <li>• Dirección</li> <li>• Procedencia</li> <li>• Previsión</li> </ul>
04	Ingresar datos del paciente	<p>Podrá ingresar los datos que le corresponden de la hoja SAPU estos datos son:</p> <ul style="list-style-type: none"> <li>• Hora atención Enfermera</li> <li>• Motivo consulta</li> <li>• Prioridad de atención</li> <li>• Estado de ingreso</li> <li>• Control de signos vitales</li> <li>• Antecedentes Mórbidos</li> <li>• Antecedentes Farmacológicos</li> <li>• Evaluación de enfermería</li> </ul>

### 1.4.3 Módulo Médico.

Id	Nombre	Descripción
01	Login	El médico deberá tener un sistema de login con un usuario y una password.
02	Busca por folio	Podrá buscar los datos de un paciente por el folio de la hoja SAPU que corresponda
03	Mostrar datos paciente	<p>Podrá ver los datos del paciente que están previamente ingresados por el SOME</p> <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Rut</li> <li>• Sexo</li> <li>• N° Ficha</li> <li>• Fecha Nacimiento</li> <li>• Folio</li> <li>• Teléfono</li> <li>• Antecedentes Mórbidos</li> <li>• Estado de ingreso</li> <li>• Edad</li> <li>• Fecha Ingreso</li> <li>• Hora Ingreso</li> <li>• Dirección</li> <li>• Procedencia</li> <li>• Previsión</li> <li>• Antecedente Farmacológicos</li> <li>• Prioridad de atención</li> </ul>
04	Ingresar datos del paciente	<p>Podrá ingresar los datos que le corresponden de la hoja SAPU estos datos son:</p> <ul style="list-style-type: none"> <li>• Hora atención Médico.</li> <li>• Patologías Centinelas</li> <li>• Auge</li> <li>• Causas Sistema Circulatorio</li> <li>• Accidentes</li> <li>• Demás causas Hipótesis Diagnóstico</li> <li>• Indicaciones de referencia</li> <li>• SAMU</li> <li>• Consulta pertinente</li> <li>• Procedimiento Médico</li> <li>• Indicaciones</li> <li>• Exámenes solicitados</li> <li>• Signos Vitales</li> </ul>

### 1.4.4 Módulo Estadista

Id	Nombre	Descripción
01	Login	El estadista deberá tener un sistema de Login con un usuario y una password.
02	Genera REM	Al ingresar un rango de fechas (fecha inicio y fecha termino) se generara en formato PDF la sección D del REM08

### 1.4.5 Módulo Portal

Id	Nombre	Descripción
01	Muestra lista de atención	Se muestra la lista de los pacientes que esperan atención en el orden que se realizaran estas.

### Interfaces externas de entrada

Cada interfaz de entrada indica todos los grupos de datos que serán ingresados al sistema independiente del medio de ingreso.

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem
ID_01	Datos de la atención (técnico paramédico)	<ul style="list-style-type: none"> <li>• Hora de egreso</li> <li>• Acompañamiento del paciente</li> <li>• Traslado en silla de ruedas</li> <li>• Prestaciones y procedimientos</li> </ul>
ID_02	Datos de la atención (Enfermera)	<ul style="list-style-type: none"> <li>• Hora atención Enfermera</li> <li>• Motivo consulta</li> <li>• Prioridad de atención</li> <li>• Estado de ingreso</li> <li>• Control de signos vitales</li> <li>• Antecedentes Mórbidos</li> <li>• Antecedentes Farmacológicos</li> <li>• Evaluación de enfermería</li> </ul>
ID_03	Datos de la atención (Médico)	<ul style="list-style-type: none"> <li>• Hora atención Médico.</li> <li>• Patologías Centinelas</li> <li>• Auge</li> <li>• Causas Sistema Circulatorio</li> <li>• Accidentes</li> <li>• Demás causas Hipótesis Diagnóstico</li> <li>• Indicaciones de referencia</li> <li>• SAMU</li> <li>• Consulta pertinente</li> </ul>

		<ul style="list-style-type: none"> <li>• Procedimiento Médico</li> <li>• Indicaciones</li> <li>• Medicamento</li> <li>• Horario</li> <li>• N° Días</li> <li>• N° Dosis</li> <li>• Exámenes solicitados</li> </ul>
ID_04	Rango de fechas (Estadista)	<ul style="list-style-type: none"> <li>• Fecha de inicio</li> <li>• Fecha de fin</li> </ul>

### Interfaces externas de Salida

Se especifica cada salida del sistema, indicando en cada caso el formato o medio de salida.

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem	Medio Salida
IS_01	Datos paciente (técnico)	<ul style="list-style-type: none"> <li>• Nombre</li> <li>• Rut</li> <li>• Numero de ficha</li> <li>• Folio</li> <li>• Fono</li> <li>• Fecha actual</li> <li>• Hora de ingreso</li> <li>• Fecha de nacimiento</li> <li>• Consultorio de procedencia</li> <li>• Edad</li> <li>• Sexo</li> <li>• Previsión</li> </ul>	Pantalla
IS_02	Datos paciente (Enfermera)	<ul style="list-style-type: none"> <li>• Nombre</li> <li>• Rut</li> <li>• Numero de ficha</li> <li>• Folio</li> <li>• Fono</li> <li>• Fecha actual</li> <li>• Hora de ingreso</li> <li>• Fecha de nacimiento</li> <li>• Consultorio de procedencia</li> <li>• Edad</li> <li>• Sexo</li> <li>• Previsión</li> </ul>	Pantalla

---

IS_03	Datos paciente (Médico)	<ul style="list-style-type: none"> <li>• Nombre</li> <li>• Rut</li> <li>• Numero de ficha</li> <li>• Folio</li> <li>• Fono</li> <li>• Fecha actual</li> <li>• Hora de ingreso</li> <li>• Fecha de nacimiento</li> <li>• Consultorio de procedencia</li> <li>• Edad</li> <li>• Sexo</li> <li>• Previsión</li> <li>• Acompañamiento del paciente</li> <li>• Traslado en silla de ruedas</li> <li>• Motivo consulta</li> <li>• Prioridad de atención</li> <li>• Estado de ingreso</li> <li>• Control de signos vitales</li> <li>• Antecedentes Mórbidos</li> <li>• Antecedentes Farmacológicos</li> <li>• Evaluación de enfermería</li> </ul>	Pantalla
-------	-------------------------	--	----------

---

### Atributos del producto

- **USABILIDAD- OPERABILIDAD.** Que los mensajes de error sean claros para el usuario. Es decir que en el 100% de los mensajes de error, relacionados con el ingreso y procesamiento de datos, se indique claramente el error, la causa y la solución.
- **EFICIENCIA- TIEMPO DE EJECUCIÓN/RESPUESTA.** El sistema debe garantizar un tiempo de carga inferior a 30 segundos considerando 10 usuarios conectados.
- **FUNCIONALIDAD-SEGURIDAD.** El sistema debe mantener un control de acceso a la funcionalidad a través de login –password establecido según los perfiles definidos para los usuarios del sistema. Estos perfiles se crearan en la plataforma virtual SOME.

- Portabilidad. el sistema debe ser capaz de funcionar en cualquier servidor que contenga la base de datos pertinente y el programa Apache Tomcat instalado.

### **1.5 Características de los usuarios.**

Los usuarios que utilizaran los perfiles de enfermera, médico y técnico paramédico deben tener conocimiento acerca del llenado de la hoja SAPU, además, de conocimientos básicos acerca de la utilización de una aplicación web. Para lograr esto se realizaran capacitaciones para enseñar el llenado de manera electrónica de la hoja diaria SAPU.

Los usuarios del perfil estadístico mayoritariamente solo deberán saber ingresar los parámetros de búsqueda de manera correcta para realizar la consulta.

### **1.6 Requisitos de rendimiento.**

El tiempo de respuesta del sistema no deberá ser mayor a 30 segundos, al realizar la búsqueda de los datos del paciente, guardar los datos de las hojas diarias de SAPU y realizar las consultas correspondientes para completar la sección "D" del REM 08.

# **ANEXO B: DIAGRAMA DE CASOS DE USO Y ESPECIFICACIÓN**

## DIAGRAMA DE CASOS DE USO

En las siguientes figuras se presentan los casos de uso Hoja SAPU.

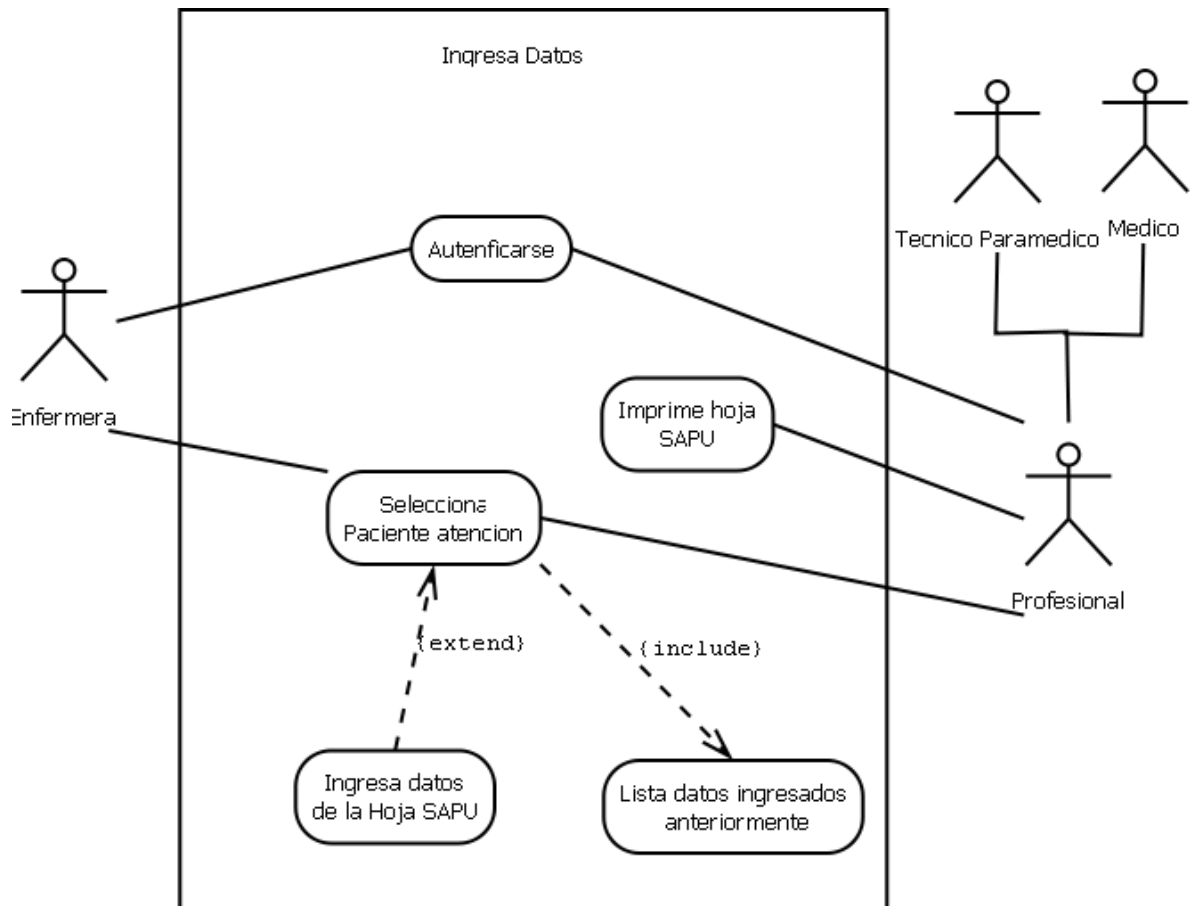
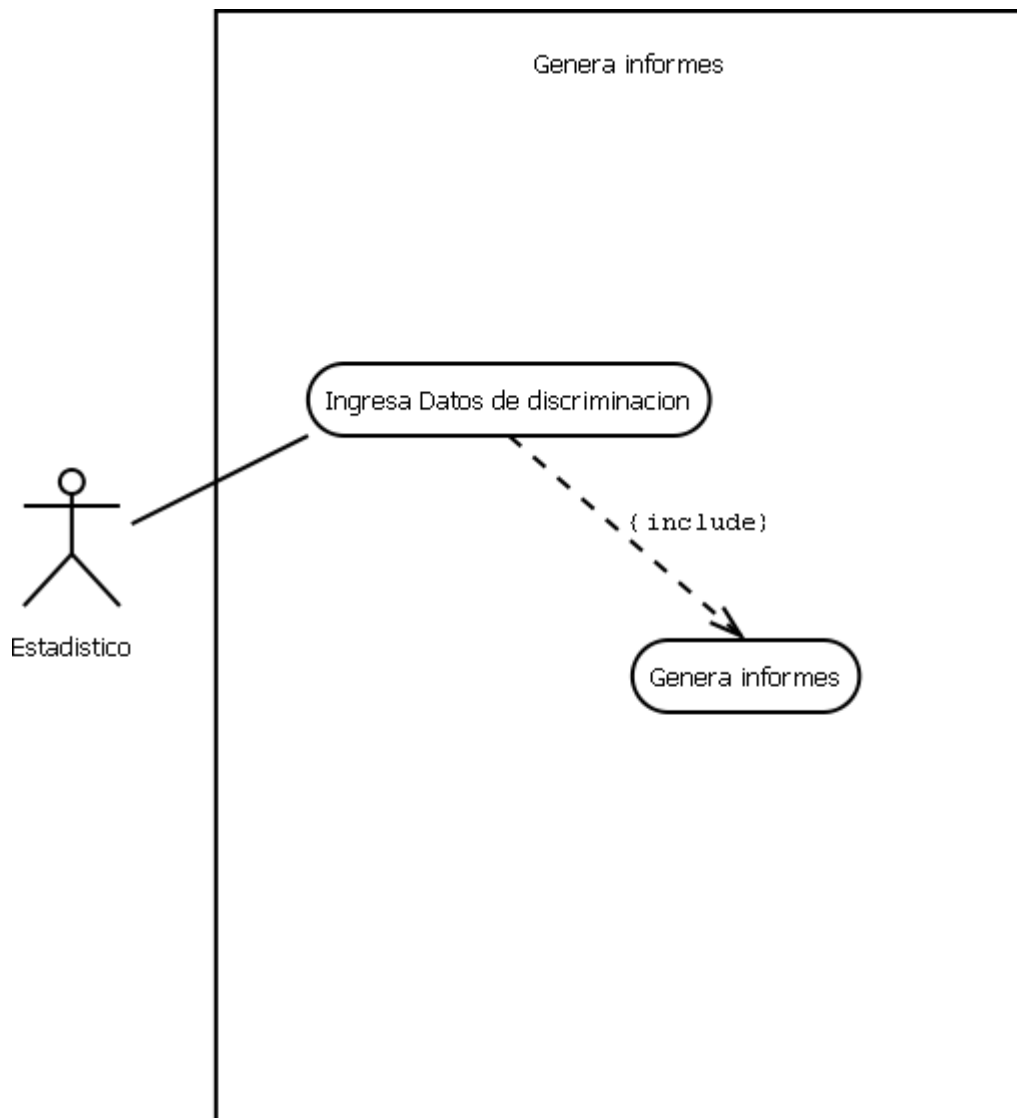


Figura 12: Caso de uso, ingresa datos.





**Figura 13: Caso de uso, genera informes**

### Actores

A continuación se especificara cada actor que interviene en el proceso mencionando su rol, nivel de conocimiento, nivel de privilegio entre otras cosas.

Todos los roles que se listaran a continuación comparten la necesidad de tener un nivel de conocimiento básico acerca del funcionamiento y manejo de una aplicación web, solo debe conocer el flujo de navegación de esta.

## Enfermera.

- Es la encargada de hacer el primer acercamiento con el paciente y hacer la primera evaluación a los síntomas que posee este.
- Las funcionalidades del sistema que tendrá acceso son las siguiente:
  - Ingreso de información del paciente tal como:
    - Hora atención Enfermera
    - Motivo consulta
    - Prioridad de atención
    - Estado de ingreso
    - Control de signos vitales
    - Antecedentes Mórbidos
    - Antecedentes Farmacológicos
    - Evaluación de enfermería
  - Eliminación de información del paciente tal como:
    - Signos vitales
  - Vista de información del paciente tal como:
    - Nombre
    - Rut
    - Numero de ficha
    - Folio
    - Fono
    - Fecha actual
    - Hora de ingreso
    - Fecha de nacimiento
    - Consultorio de procedencia
    - Edad
    - Sexo
    - Previsión

#### Técnico Paramédico.

- Esta encargado de ayudar al médico en el momento de la atención al
- Las funcionalidades del sistema que tendrá acceso son las siguiente:
  - Ingreso de información del paciente tal como:
    - Hora de egreso
    - Acompañamiento del paciente
    - Traslado en silla de ruedas
    - Prestaciones y procedimientos
  - Vista de información del paciente tal como:
    - Nombre
    - Rut
    - Numero de ficha
    - Folio
    - Fono
    - Fecha actual
    - Hora de ingreso
    - Fecha de nacimiento
    - Consultorio de procedencia
    - Edad
    - Sexo
    - Previsión

#### Médico.

- Es el encargado de hacer el diagnóstico del paciente es el que culmina la atención SAPU.
- Las funcionalidades del sistema que tendrá acceso son las siguiente:
  - Ingreso de información del paciente tal como:
    - Hora atención Médico.
    - Patologías Centinelas
    - Auge
    - Causas Sistema Circulatorio

- Accidentes
- Demás causas Hipótesis Diagnóstico
- Vista de información del paciente tal como:
  - Nombre
  - Rut
  - Numero de ficha
  - Folio
  - Fono
  - Fecha actual
  - Hora de ingreso
  - Fecha de nacimiento
  - Consultorio de procedencia
  - Edad
  - Sexo
  - Previsión
  - Acompañamiento del paciente
  - Traslado en silla de ruedas
  - Prestaciones y procedimientos
  - Motivo consulta
  - Prioridad de atención
  - Estado de ingreso
  - Control de signos vitales
  - Antecedentes Mórbidos
  - Antecedentes Farmacológicos
  - Evaluación de enfermería

## Especificación de los Casos de Uso

A continuación se muestra la especificación de los casos de uso de la sección.

<b>Autenticar.</b>
<b>Id:</b> 01
<b>Descripción:</b>
<ul style="list-style-type: none"> <li>• Comprueba que un usuario registrado sea el que ingrese al sistema.</li> </ul>
<b>Actores Principales:</b>
<ul style="list-style-type: none"> <li>• Enfermera.</li> <li>• Técnico Paramédico.</li> <li>• Médico.</li> <li>• Estadístico</li> </ul>
<b>Actores secundarios:</b>
<b>Precondiciones:</b>
<ul style="list-style-type: none"> <li>• El usuario desea ingresar al sistema</li> </ul>
<b>Flujo principales:</b>
<ol style="list-style-type: none"> <li>1. El usuario debe ingresar su usuario y contraseña.</li> <li>2. El sistema realiza el proceso de validación de los datos.</li> <li>3. El sistema muestra las opciones correspondientes según el tipo de usuario.</li> </ol>
<b>Post- condiciones:</b>
<b>Flujos alternativos:</b>
<ol style="list-style-type: none"> <li>3.1. Se genera alerta de usuario o contraseña no válidos.</li> <li>3.2. Se direcciona nuevamente a la página autenticar.</li> </ol>

<b>Selecciona paciente atención.</b>
<b>Id:</b> 02
<b>Descripción:</b> <ul style="list-style-type: none"> <li>• Se selecciona el paciente al cual se le realizara la atención.</li> </ul>
<b>Actores Principales:</b> <ul style="list-style-type: none"> <li>• Enfermera.</li> <li>• Técnico Paramédico.</li> <li>• Médico.</li> </ul>
<b>Actores secundarios:</b>
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado (caso de uso Id 01).</li> <li>• Deben existir pacientes ingresados para ser atendidos.</li> </ul>
<b>Flujo principales:</b> <ol style="list-style-type: none"> <li>1. Seleccionar al paciente que se encuentre en la primera posición de la lista de atención.</li> <li>2. Se muestran los datos del paciente que según el perfil de usuario registrado que corresponda.</li> </ol>
<b>Post- condiciones:</b> <ul style="list-style-type: none"> <li>• Se debe realizar la atención al paciente.</li> </ul>
<b>Flujos alternativos:</b> <ol style="list-style-type: none"> <li>2.1. Se despliega mensaje de alerta diciendo que el paciente no puede ser atendido dos veces por el mismo profesional.</li> <li>2.2. Si el paciente no responde al llamado para realizar su atención después de tres ocasiones, se borra de la lista de atención quedando en estado NSP (No Se Presenta).</li> </ol>

<b>Ingresar Datos de la hoja SAPU.</b>
<b>Id:</b> 03
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>• Se ingresan los datos del paciente correspondientes a cada perfil de usuario.</li> </ul>
<p><b>Actores Principales:</b></p> <ul style="list-style-type: none"> <li>• Enfermera.</li> <li>• Técnico Paramédico.</li> <li>• Médico.</li> </ul>
<p><b>Actores secundarios:</b></p>
<p><b>Precondiciones:</b></p> <ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado (caso de uso Id 01).</li> <li>• El paciente debe estar previamente seleccionado (caso de uso Id 02).</li> </ul>
<p><b>Flujo principales:</b></p> <ol style="list-style-type: none"> <li>1. Se muestran los campos que se deben completar cada profesional correspondiente a la atención del paciente.</li> <li>2. Se ingresa información en cada uno de los campos que existen.</li> <li>3. Se confirma cada uno de los campos por separado.</li> <li>4. Se confirma la atención por parte del profesional.</li> </ol>
<p><b>Post- condiciones:</b></p>
<p><b>Flujos alternativos:</b></p> <ol style="list-style-type: none"> <li>4.1. Si no se ingresó ningún campo, se envía mensaje de error "Atención SAPU no realizada".</li> <li>4.2. Se retorna a la página de ingreso de datos del paciente.</li> </ol>

<b>Lista datos ingresados anteriormente</b>
<b>Id:</b> 04
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>• Muestra en pantalla los datos del paciente que fueron ingresados anteriormente.</li> </ul>
<p><b>Actores Principales:</b></p> <ul style="list-style-type: none"> <li>• Enfermera.</li> <li>• Técnico Paramédico.</li> <li>• Médico.</li> </ul>
<p><b>Actores secundarios:</b></p>
<p><b>Precondiciones:</b></p> <ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado (caso de uso Id 01).</li> <li>• Deben existir datos anteriormente ingresados del paciente.</li> <li>• El paciente debe estar previamente seleccionado (caso de uso Id 02).</li> </ul>
<p><b>Flujo principales:</b></p> <ol style="list-style-type: none"> <li>1. Se muestran los datos del paciente que está seleccionado.</li> </ol>
<p><b>Post- condiciones:</b></p> <ul style="list-style-type: none"> <li>• Los datos mostrados no deben ser modificados por un perfil distinto al que lo ingreso.</li> </ul>
<p><b>Flujos alternativos:</b></p>



<b>Imprime Hoja SAPU</b>
<b>Id:</b> 05
<b>Descripción:</b> <ul style="list-style-type: none"> <li>• Genera Hoja SAPU (imprimible) del paciente.</li> </ul>
<b>Actores Principales:</b> <ul style="list-style-type: none"> <li>• Técnico Paramédico.</li> <li>• Médico.</li> </ul>
<b>Actores secundarios:</b>
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>• Se deben haber ingresado todos los datos correspondientes a la atención del paciente.</li> </ul>
<b>Flujo principales:</b> <ol style="list-style-type: none"> <li>1. Se confirma la atención de los tres profesionales de la salud.</li> <li>2. Se realiza la consulta a la base de datos</li> <li>3. Se estructura la respuesta de la consulta al formato de la hoja SAPU</li> <li>4. Se genera el informe.</li> </ol>
<b>Post- condiciones:</b>
<b>Flujos alternativos:</b>

<b>Ingresar datos de discriminación</b>
<b>Id:</b> 06
<b>Descripción:</b> <ul style="list-style-type: none"> <li>• Se ingresa un rango de fechas valido para la generación del reporte.</li> </ul>
<b>Actores Principales:</b> <ul style="list-style-type: none"> <li>• Estadístico.</li> </ul>
<b>Actores secundarios:</b>
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>• El usuario debe estar previamente autenticado (caso de uso Id 01).</li> <li>• Debe ser un rango de fecha valido.</li> </ul>
<b>Flujo principales:</b> <ol style="list-style-type: none"> <li>1. Se ingresa un rango de fecha (fecha inicio, fecha final)</li> <li>2. Se genera el informe</li> </ol>
<b>Post- condiciones:</b>
<b>Flujos alternativos:</b> <ol style="list-style-type: none"> <li>2.1. Se genera alerta que rango de fechas no tiene atenciones SAPU asociadas.</li> </ol>

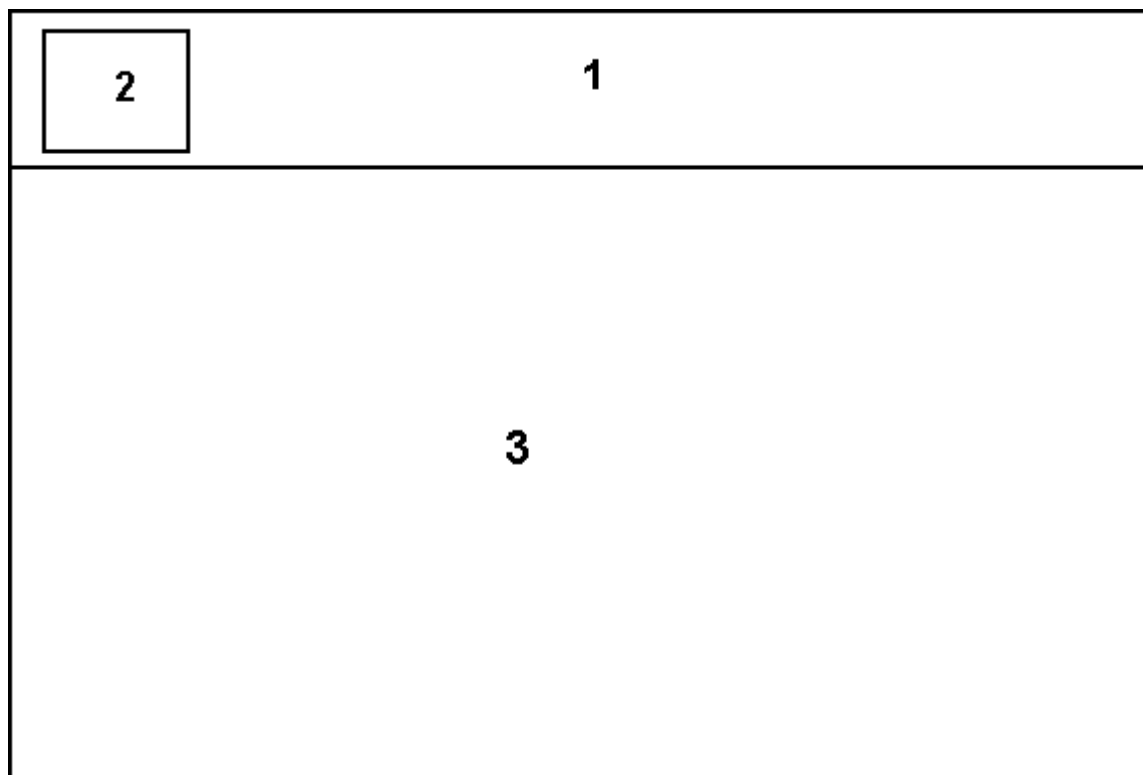
<b>Genera informes</b>
<b>Id:</b> 07
<b>Descripción:</b> <ul style="list-style-type: none"> <li>• Genera sección D del ReM08</li> </ul>
<b>Actores Principales:</b> <ul style="list-style-type: none"> <li>• Estadístico.</li> </ul>
<b>Actores secundarios:</b>
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>• Se debe haber ingresado un rango de fecha anteriormente (caso de uso 06).</li> </ul>
<b>Flujo principales:</b> <ol style="list-style-type: none"> <li>1. Se realiza la consulta a la base de datos</li> <li>2. Se estructura la respuesta de la consulta al formato de la sección D del REM08</li> </ol>
<b>Post- condiciones:</b>
<b>Flujos alternativos:</b> <ol style="list-style-type: none"> <li>2.1. Se genera alerta que rango de fechas no tiene atenciones SAPU asociadas.</li> </ol>

# **ANEXO C: DISEÑO DE INTERFAZ Y NAVEGACIÓN**

---

## DISEÑO DE INTERFAZ Y NAVEGACIÓN

---



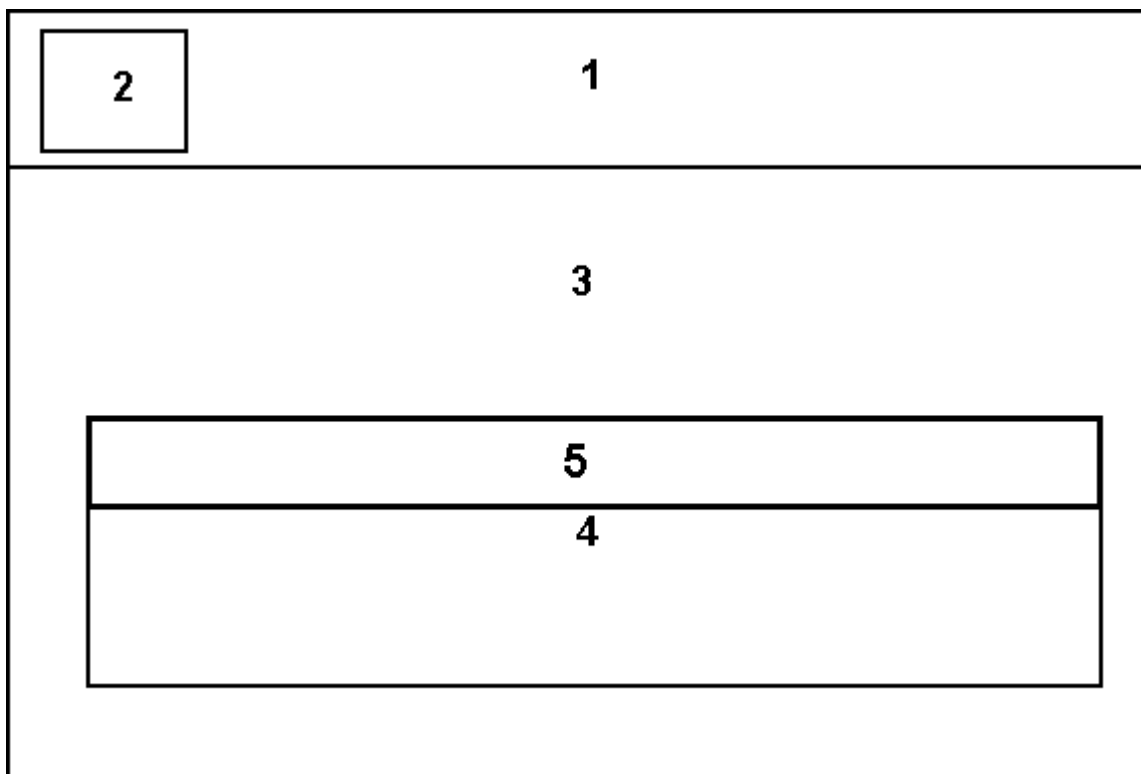
**Figura 14: Diseño de interfaz búsqueda de pacientes**

Area 1. Banner que contiene:

- Título del sistema.
- Reloj
- Nombre del usuario autenticado.

Area 2. Logo corporativo.

Area 3. Lista de pacientes y estado de atención.



**Figura 15: Diseño de interfaz Enfermero/Médico**

Area 1. Banner que contiene:

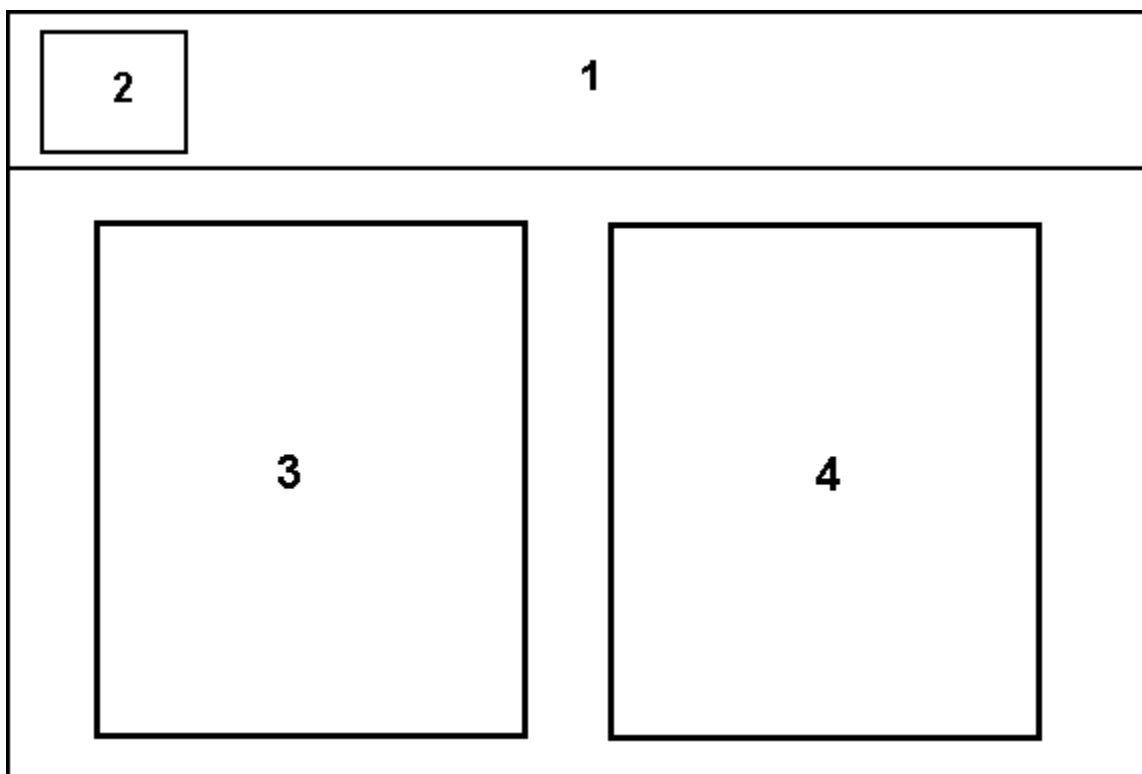
- Título del sistema.
- Reloj
- Nombre del usuario autenticado.

Area 2. Logo corporativo.

Area 3. Despliegue de muestra de datos.

Area 4. Despliegue de ingreso de datos.

Area 5. Tabs para cambio de formulario de ingreso de datos.



**Figura 16: Diseño de interfaz Técnico Paramédico**

- Area 1. Banner que contiene:
  - Título del sistema.
  - Reloj
  - Nombre del usuario autenticado.
- Area 2. Logo corporativo.
- Area 3. Despliegue de muestra de datos.
- Area 4. Despliegue de ingreso de datos.

