

UNIVERSIDAD AMAZÓNICA DE PANDO

ÁREA DE CIENCIAS Y TECNOLOGÍA

CARRERA DE INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“IMPLEMENTACIÓN DEL SISTEMA DE INFORMACIÓN PARA LA
DEFENSORÍA DE LA NIÑEZ Y ADOLESCENCIA DEL MUNICIPIO DE
COBIJA”**

**PROYECTO DE GRADO PRESENTADO PARA OBTENER EL TÍTULO ACADÉMICO
DE LICENCIADO EN INGENIERÍA DE SISTEMAS**

POSTULANTE: Univ. Tony Tomas Nina Yanahuaya

TUTOR: Ing. Juan Carlos Gallardo Jiménez

ASESOR: M.Sc. Lic. Humberto Fernández Calle

Cobija - Pando – Bolivia

2018

AGRADECIMIENTOS

A Dios todo poderoso por haber iluminado mi camino y de protegerme de la caídas.

A mis queridos padres y familia por su constante apoyo moral y aliento.

A mi querida esposa Rosmery Candelaria Mamani Mendoza por estar a lado en las buenas y malas luchando por salir adelante y a mi querido hijo Eithan Yannick Nina Mamani por llegar a mi vida.

A mi Tutor y Asesor, Ing. Juan Carlos Gallardo Jiménez y M.Sc. Lic. Humberto Fernández Calle, por los consejos y la sugerencia para cumplir con la meta del proyecto.

En especial agradecer al Ing. Luis Iván Condori Calle y al Ing. Edwin Mamani Huanca, por el apoyo incondicional y constante para la presentación del proyecto.

A los señores Docentes de la Universidad Amazónica de Pando, quienes hicieron posible mi formación académica.

A mis compañeros universitarios por los años compartidos.

DEDICATORIA

Este trabajo está dedicado a todo el esfuerzo realizado por tantos días de estudio con la compañía de Dios y que junto a mi querida esposa e hijo, y anhelo de mis queridos padres por haberme dado la vida y por ser constante guía en el transcurso de mi vida y por ser la inspiración de seguir estudiando para culminar con mi esfuerzo la vida universitaria

RESUMEN

El presente proyecto se orienta a la implementación de un sistema de información para la defensoría de la niñez y adolescencia del municipio de cobija, como principal referente es la organización de la información para fines de mantener un control de las solicitudes de viaje que se lo otorga al menor, quien viajará en compañía de uno de los padres o sin ninguna compañía, o bien siendo acompañado por un responsable o apoderado y de la misma manera se podrá tener una información verídica.

Para lograr el objetivo anteriormente descrito se empleó la metodología Xp, que trabaja en base a fases de (planificación, diseño, desarrollo y las pruebas) y también haciendo el uso de las herramientas del Lenguaje de Modelado Unificado para preparar los esquemas del sistema haciendo uso de los diagramas, los cuales se diseñaron con la herramienta de software Enterprise Architect, finalmente se procedió a la implementación de la arquitectura cliente/servidor el cual centraliza los datos a un servidor, y facilita a los usuarios puedan acceder desde cualquier punto en el interior de la institución.

PALABRAS CLAVES: Sistema de Información, Defensoría de la niñez y adolescencia, Lenguaje de Modelado Unificado.

ABSTRACT

The present project is oriented to the implementation of an information system for the defense of children and adolescents of the municipality of cobija, as the main reference is the organization of information for purposes of maintaining control of travel requests that grants it the minor, who will travel in the company of one of the parents or without any company, or being accompanied by a responsible or proxy and in the same way you can have a real information.

To achieve the objective described above, the Xp methodology was used, which works on the basis of phases of (planning, design, development and testing) and also making use of Unified Modeling Language tools to prepare the system diagrams using of the diagrams, which were designed with the software tool Enterprise Architect, finally proceeded to the implementation of the client / server architecture which centralizes the data to a server, and facilitates users to access from any point in the interior of the institution.

KEY WORDS: Information System, Defender of childhood and adolescence, Unified Modeling Language.

ÍNDICE

CAPITULO I	12
1. MARCO INTRODUCTORIO	12
1.1. ANTECEDENTES	13
1.2. DESCRIPCIÓN DEL PROBLEMA.....	14
1.3. OBJETIVOS	15
1.3.1. Objetivo General.....	15
1.3.2. Objetivos Específicos	15
1.4. JUSTIFICACIÓN	15
1.4.1. Justificación Económica.....	16
1.4.2. Justificación Social	16
1.4.3. Justificación Técnica	16
1.5. ALCANCES	17
1.6. RESULTADOS OBTENIDOS.....	18
1.7. METODOLOGÍA Y HERRAMIENTAS UTILIZADAS.....	19
1.7.1. Metodología.....	19
1.7.2. Herramientas.....	20
1.7.3. Plataforma de desarrollo.....	20
2. CAPITULO II	21
2.1. MARCO INSTITUCIONAL	22
2.1.1. Marco Institucional de la Unidad de la Defensoría de la Niñez y Adolescencia del Municipio de Cobija.....	22
2.2. MARCO LEGAL.....	24
2.2.1. Legislación Internacional	24
2.2.2. Legislación Nacional	25
2.3. MARCO TEÓRICO.....	27
2.3.1. Introducción.....	27
2.3.2. Sistema de información	27
2.3.3. Redes de Datos	28

2.3.4.	Herramientas para la programación.....	28
2.3.5.	Plataforma de desarrollo.....	29
2.3.6.	Elementos de la implementación de los sistemas de información.....	29
2.4.	METODOLOGÍA DE DESARROLLO	32
2.4.1.	Metodología pesada.....	32
2.4.2.	Metodología ágiles	33
2.4.3.	Metodología XP.....	35
2.5.	LENGUAJE DE MODELADO UNIFICADO.....	38
2.6.	ENTERPRISE ARCHITECT	39
2.7.	LENGUAJE DE PROGRAMACIÓN	39
2.7.1.	Programación estructurada	40
2.7.2.	Programación Orientado a Objetos (OOP).....	41
2.8.	MÉTRICAS DE CALIDAD ISO/IEC 9126.....	43
2.8.1.	Funcionalidad	43
2.8.2.	Confiabilidad	43
2.8.3.	Usabilidad.....	44
2.8.4.	Eficiencia.....	44
2.8.5.	Capacidad de mantenimiento.....	45
2.8.6.	Portabilidad.....	45
2.9.	PRUEBA DE REQUERIMIENTOS	46
2.9.1.	Planeación.....	47
2.9.2.	Diseño de caso de prueba	48
2.9.3.	Ejecución de caso de prueba.....	49
2.9.4.	Cierre	49
3.	CAPITULO III.....	50
3.1.	MARCO APLICATIVO.....	51
3.2.	DIAGNOSTICO PRELIMINAR.....	51
3.3.	PLANIFICACIÓN DEL SISTEMA	52
3.3.1.	Identificación de usuarios.....	52
3.3.2.	Historia de usuarios	53

3.3.3.	Plan de entregas	59
3.4.	PRIMERA ITERACIÓN	60
3.4.1.	Plan de la primera iteración	60
3.4.2.	Diseño del subsistema de la primera iteración	60
3.4.3.	Desarrollo del subsistema de la primera iteración.....	66
3.4.4.	Pruebas del sistema de la primera Iteración	68
3.5.	SEGUNDA ITERACIÓN	69
3.5.1.	Plan de la segunda iteración	69
3.5.2.	Diseño del subsistema de la segunda iteración.....	70
3.5.3.	Desarrollo del subsistema de la segunda iteración	77
3.5.4.	Pruebas del sistema de la segunda Iteración.....	81
3.6.	TERCERA ITERACIÓN.....	83
3.6.1.	Plan de la tercera iteración.....	83
3.6.2.	Diseño del subsistema de la tercera iteración	83
3.6.3.	Desarrollo del subsistema de la tercera iteración	88
3.6.4.	Pruebas de la tercera iteración	88
3.7.	CUARTA ITERACIÓN.....	90
3.7.1.	Plan de la cuarta iteración.....	90
3.7.2.	Diseño del subsistema de la cuarta iteración	91
3.7.3.	Desarrollo del subsistema de la cuarta iteración	96
3.7.4.	Pruebas de la cuarta iteración	96
3.8.	ARQUITECTURA BASE DE DATOS CLIENTE/SERVIDOR	98
3.9.	DOCUMENTO DE SOPORTE DE LA ETAPA DE PRUEBAS DEL REQUERIMIENTO DE LA METODOLOGÍA	100
3.9.1.	Identificación de Necesidades	100
3.9.2.	Entrevistas	101
3.9.3.	Matrices de trazabilidad.....	101
3.9.4.	Especificación Funcional.....	101
3.9.5.	Especificación de Caso de Uso.....	102
3.9.6.	Especificación de Diagramas.....	103

3.9.7.	Planeación Prueba de Requerimientos	103
3.9.8.	Diseño de Casos de prueba.....	103
3.9.9.	Ejecución de Casos de prueba	104
3.10.	VALORACIÓN DEL SISTEMA.....	105
4.	CAPITULO IV	109
4.1.	CONCLUSIONES	110
4.2.	RECOMENDACIONES.....	111
5.	BIBLIOGRAFÍA.....	112
6.	ANEXOS.....	115
6.1.	ANEXO A.....	115
6.1.1.	Formulario Anterior de Autorización de Viajes	115
6.1.2.	Formulario Actual de Autorización de viajes.....	116
6.2.	ANEXO B.....	117
6.2.1.	METODOLOGÍA PROGRAMACIÓN EXTREMA.....	117
6.2.2.	Iteración	117
6.2.3.	Desarrollo	118
6.2.4.	Manejo Colectivo del Código.....	118
6.3.	ANEXO C.....	119
6.3.1.	Organigrama del Gobierno Autónomo Municipal De Cobija	119
6.4.	ANEXO D.....	120
6.4.1.	Inicio de las Pruebas Unitarias con el <i>framework</i> JUnit.....	120
6.5.	ANEXO E	122
6.5.1.	Evaluación del sistema de acuerdo a las normas ISO/IEC 9126	122
6.6.	ANEXO F	124
6.6.1.	Capacitación al personal de la Defensoría de la Niñez y Adolescencia	124

ÍNDICE DE ILUSTRACIÓN

ILUSTRACIÓN 2.1 ORGANIGRAMA INSTITUCIONAL SECRETARIA MUNICIPAL DE DESARROLLO HUMANO	22
ILUSTRACIÓN 2.2 FASES DE LA PROGRAMACIÓN EXTREMA.....	36
ILUSTRACIÓN 2.3 TARJETA CLASE-RESPONSABILIDAD-COLABORACIÓN	37
ILUSTRACIÓN 2.4 TABLA DE APLICACIONES DE PRUEBAS	38
ILUSTRACIÓN 2.5 FORMULARIO DE REQUERIMIENTOS.....	48
ILUSTRACIÓN 2.6 DISEÑO DE CASO DE PRUEBA	49
ILUSTRACIÓN 3.7: REGISTRO DE AUTORIZACIÓN DE VIAJES.....	51
ILUSTRACIÓN 3.8: DIAGRAMA PROCESO	52
ILUSTRACIÓN 3.9 FIGURA DIAGRAMA DE CLASES DE LA PRIMERA ITERACIÓN	63
ILUSTRACIÓN 3.10 MODELO DE LA BASE DE DATOS DE LA PRIMERA ITERACIÓN.....	64
ILUSTRACIÓN 3.11 FORMULARIO DE BÚSQUEDA DATOS DE PERSONA.....	65
ILUSTRACIÓN 3.12 FORMULARIO DE PERSONA	65
ILUSTRACIÓN 3.13 DIAGRAMA DE CASO DE USO.....	66
ILUSTRACIÓN 3.14 DIAGRAMA DE SECUENCIA	66
ILUSTRACIÓN 3.15 CODIFICACIÓN EN JAVA.....	67
ILUSTRACIÓN 3.16 CLASE PERFIL FUENTE ELABORACIÓN PROPIA.....	67
ILUSTRACIÓN 3.17 PRUEBAS DE LA PRIMERA ITERACIÓN.....	68
ILUSTRACIÓN 3.18: PRUEBA DE ACEPTACIÓN DE LA PRIMERA ITERACIÓN	69
ILUSTRACIÓN 3.19 DIAGRAMA DE CLASES DE LA SEGUNDA ITERACIÓN	72
ILUSTRACIÓN 3.20 MODELO DE LA BASE DE DATOS DE LA SEGUNDA ITERACIÓN.....	73
ILUSTRACIÓN 3.21 FORMULARIO DE MENOR.....	74
ILUSTRACIÓN 3.22 FORMULARIO DE FAMILIA/RESPONSABLE.....	75
ILUSTRACIÓN 3.23 FORMULARIO DE AUTORIZACIÓN DE VIAJES.....	76
ILUSTRACIÓN 3.24 FORMULARIO DE AUTORIZACIÓN DE VIAJES	76
ILUSTRACIÓN 3.25 DIAGRAMA DE CASO DE USO.....	77
ILUSTRACIÓN 3.26 DIAGRAMA DE SECUENCIA.....	78
ILUSTRACIÓN 3.27 CODIFICACIÓN DEL FORMULARIO AUTORIZACIÓN.....	79
ILUSTRACIÓN 3.28 CODIFICACIÓN DEL FORMULARIO FAMILIA/RESPONSABLE.....	80
ILUSTRACIÓN 3.29 CODIFICACIÓN DEL FORMULARIO MENOR	81
ILUSTRACIÓN 3.30 PRUEBAS SEGUNDA ITERACIÓN.....	82
ILUSTRACIÓN 3.31 PRUEBA DE ACEPTACIÓN SEGUNDA ITERACIÓN.....	83
ILUSTRACIÓN 3.32 DIAGRAMA DE CLASES DE LA TERCERA ITERACIÓN	85
ILUSTRACIÓN 3.33 MODELO DE LA BASE DE DATOS DE LA TERCERA ITERACIÓN.....	86
ILUSTRACIÓN 3.34 FORMULARIO DE REQUISITOS	87
ILUSTRACIÓN 3.35 FORMULARIO DE REPORTE	87
ILUSTRACIÓN 3.36 DIAGRAMA DE CASO DE USO.....	88

ILUSTRACIÓN 3.37 PRUEBA DE LA TERCERA ITERACIÓN	89
ILUSTRACIÓN 3.38 PRUEBA DE ACEPTACIÓN TERCERA ITERACIÓN.....	90
ILUSTRACIÓN 3.39 DIAGRAMA DE CLASES DE LA CUARTA ITERACIÓN.....	92
ILUSTRACIÓN 3.40 MODELO DE LA BASE DE DATOS DE LA CUARTA ITERACIÓN.....	93
ILUSTRACIÓN 3.41 FORMULARIO DE ESTADÍSTICOS	94
ILUSTRACIÓN 3.42 FORMULARIO DE GRÁFICOS.....	94
ILUSTRACIÓN 3.43 FORMULARIO DE COPIA DE SEGURIDAD.....	95
ILUSTRACIÓN 3.44 PRUEBA DE LA CUARTA ITERACIÓN.....	97
ILUSTRACIÓN 3.45 PRUEBA DE ACEPTACIÓN CUARTA ITERACIÓN.....	98
ILUSTRACIÓN 3.46 CÓDIGO FUENTE JAVA.....	99
ILUSTRACIÓN 3.47 PRIVILEGIO EN LA BASE DE DATOS.....	99
ILUSTRACIÓN 3.48 TOPOLOGIAS DE RED DE LA INSTITUCIÓN.....	100
ILUSTRACIÓN 3.49 MATRIZ DE VALORACIÓN	101
ILUSTRACIÓN 3.50 ESPECIFICACIÓN FUNCIONAL.....	102
ILUSTRACIÓN 3.51 ESPECIFICACIÓN DE CASO DE USO	102
ILUSTRACIÓN 3.52 DATOS DE ENTRADA.....	102
ILUSTRACIÓN 3.53 DATOS DE SALIDA	103
EN ESTA ETAPA SE TOMA EN CONSIDERACIÓN LOS DIAGRAMAS EMPLEADOS EN EL	
DESARROLLO DEL SISTEMA MEDIANTE LAS ETAPAS DE ITERACIÓN (VER ILUSTRACIÓN	
3.54 DIAGRAMA DE CASO DE USO).....	103
ILUSTRACIÓN 3.55 DISEÑO DE CASO DE PRUEBA	104
ILUSTRACIÓN 3.56 EJECUCIÓN DE CASO DE PRUEBA.....	104
ILUSTRACIÓN 6.57 ORGANIGRAMA GAMC.....	119
ILUSTRACIÓN 6.58 PRUEBA A UN ARCHIVO JAVA.....	120
ILUSTRACIÓN 6.59 ARCHIVO GENERADO POR JUNIT.....	120
ILUSTRACIÓN 6.60 RESULTADOS DE TESTEO.....	121

ÍNDICE DE TABLAS

TABLA 1.1 FASE DE LA METODOLOGÍA XP	19
TABLA 2.2 TABLA DE DIFERENCIA ENTRE METODOLOGÍAS AGILES Y METODOLOGÍA NO AGILES	34
TABLA 3.3 GESTIÓN DE USUARIOS.....	54
TABLA 3.4 REGISTRO DE DATOS DEL MENOR.....	54
TABLA 3.5 REGISTRO FAMILIAR, APODERADO O RESPONSABLE.....	55
TABLA 3.6 REGISTRO DE LA AUTORIZACIÓN DE VIAJE.....	56
TABLA 3.7 ANEXOS	57
TABLA 3.8 REPORTES	58
TABLA 3.9 COPIAS DE SEGURIDAD.....	58
TABLA 3.10 PLAN GENERAL DE ENTREGAS.....	59
TABLA 3.11 PLAN PRIMERA ITERACIÓN	60
TABLA 3.12 TARJETA CRC DE LA CLASE PERSONA	61
TABLA 3.13 TARJETA CRC DE LA CLASE BÚSQUEDA.....	61
TABLA 3.14 TARJETA CRC DE LA CLASE USUARIO	61
TABLA 3.15 TARJETA CRC DE LA CLASE PERFIL.....	62
TABLA 3.16 TARJETA CRC DE LA CLASE LOGEO.....	62
TABLA 3.17 PLAN DE LA SEGUNDA ITERACIÓN	69
TABLA 3.18 TARJETA CRC DE LA CLASE MENOR	70
TABLA 3.19 TARJETA CRC DE LA CLASE FAMILIA/RESPONSABLE	70
TABLA 3.20 TARJETA CRC DE LA CLASE PERSONA	71
TABLA 3.21 PLAN DE LA TERCERA ITERACIÓN.....	83
TABLA 3.22 TARJETA CRC DE LA CLASE REQUISITOS.....	84
TABLA 3.23 TARJETA CRC DE LA CLASE REPORTE	84
TABLA 3.24 PLAN DE LA CUARTA ITERACIÓN	90
TABLA 3.25 TARJETA CRC DE LA CLASE ESTADÍSTICO.....	91
TABLA 3.26 TARJETA CRC DE LA CLASE COPIA DE SEGURIDAD.....	91
CADA UNO DE LAS FUNCIONALIDAD SE DETALLA EN CADA ITERACIÓN(VER EN TABLA 3.27 PLAN GENERAL DE ENTREGAS) PARA PODER COMPRENDER LA FUNCIONALIDAD DE SISTEMA.....	102
TABLA 3.28 FUNCIONALIDAD.....	105
TABLA 3.29 FIABILIDAD	105
TABLA 3.30 USABILIDAD.....	106
TABLA 3.31 EFICIENCIA	106
TABLA 3.32 MANTENIMIENTO.....	107
TABLA 3.33 PORTABILIDAD.....	108

CAPITULO I

1. MARCO INTRODUCTORIO

1.1. ANTECEDENTES

Los sistemas informáticos, que pertenecen al grupo de tecnologías de la información, comprenden todos aquellos medios que permiten la organización, clasificación de la información, el almacenamiento y el métodos para guardar la información, la transmisión de datos como un medio de comunicación y el tratamiento de la información como conjunto de operaciones que pueden realizarse de forma rápida y segura.

“Los avances significativos que ha realizado Bolivia, para sumar a las tecnologías de la información a la administración pública, ha sido complementada con la incorporación de la firma digital, cuyo respaldo legal se encuentra en la ley N°0164 y su Decreto Supremo N°1793”. (Jemio, 2015).

En el contexto de las defensoría de la niñez y adolescencia tiene como objetivo principal el de prevenir, proteger y vigilar los derechos del niño, niña y adolescente; y es en ese sentido que realiza trabajos orientados a la prevención capacitando personas, difundiendo y promocionando su servicios como defensoría (Jhonny, 2006).

En relación al estado de arte, se evidencio el desarrollo de sistemas de información que se mencionan a continuación:

- según (Vargas, 2009), la Defensoría realiza el registro de casos y sus respectivos seguimientos de forma manual, lo que implica a la perdida de información, derivando a largas horas de inspección a sus documentos sobre todo cuando realizan sus reportes finales que presentan. Es en este sentido que se implementó un sistema de información que ayude en el registro y el seguimiento de casos dando como resultados: un acceso inmediato a los casos registrados, datos de seguimiento de casos más verídicos, menor pérdida de información y la automatización de reportes
- según (Zambrana, 2016) “la página web del Gobierno Autónomo Municipal de La Paz, toda la información se va registrando y la documentación digitalizada(fotografía de certificado de nacimiento o carnet de identidad), además se ingresa a una base de datos por primera vez, la próxima vez que se necesite el documento el trámite será más rápido, esta es una solución

para evitar largas colas y para hacer un control mucho más estricto del viaje de los niños y las niñas”.

Según el diagnóstico realizado a La defensoría de la niñez y adolescencia del Municipio de Cobija, una de las funciones que realiza es la autorización de viajes, donde el flujo de información radica en la recepción de documentos que acrediten que son progenitores del menor que comprende desde 0 a 18 años y puedan acceder a la autorización de viaje.

El procedimiento consta de la presentación de los requisitos exigidos por la defensoría, una vez entregado y verificado las copias, se registra en el sistema ofimático (Word o Excel), y archivado en distintas hojas y a pesar que son herramientas informáticas capaz de registrar y guardar información, el personal afronta problemas cuando la cantidad de solicitudes es mayor, en particular cuando se requieren reportes, pueden ocasionar demora por que se limitan en la información registrada y en la búsqueda.

En este sentido el presente proyecto de grado tiene como propósito, de la implementación del sistema de autorización de viajes para la Defensoría de la Niñez y Adolescencia del Municipio de Cobija, con el propósito de mejorar los proceso de información, aplicando la metodología XP.

1.2. DESCRIPCIÓN DEL PROBLEMA

La Defensoría de la Niñez y Adolescencia, afronta debilidades en los procesos de administración de información, que durante la investigación se ha podido evidenciar falencias, un claro ejemplo es el registro de autorización de viajes a menores de 18 años.

A pesar que se tiene un registro guardado y almacenado de forma manual, el proceso de registro de solicitudes es deficiente a la hora de acceder a los datos de gestiones anteriores, ya que estos se encuentran almacenados en diferentes hojas de excel, la autorización no cuenta con numeración de control, y además no existe medios de seguridad para evitar la falsificación de solicitudes, de acuerdo a las causas y efectos se plantea el siguiente problema principal.

“Deficiencia en el sistema de autorización de viajes para menores de 18 años, en la Defensoría de la Niñez y Adolescencia del Municipio de Cobija”

1.3. OBJETIVOS

1.3.1. Objetivo General

Implementar un sistema de información de registro y autorización de viajes para la Defensoría de la Niñez y Adolescencia del Municipio de Cobija en base a la **metodología de desarrollo de software ágil XP**.

1.3.2. Objetivos Específicos

- Planificar de acuerdo al sistema actual, el flujo que efectúa la autorización de viaje dentro de la institución con el fin de lograr las historias de cada usuario.
- Diseñar y tomar en cuenta algunos aspectos como la tarjeta CRC para un buen diseño de los modelos de vista del sistema de autorización de viajes.
- Codificar según el análisis del flujo de autorización de viajes, empleando diagramas UML, realizar las pruebas y error antes de implementar el código y verificar el funcionamiento de cada módulo del sistema.
- Probar el desarrollo, mediante la implementación y ejecución de los procedimientos en menor tiempo, mediante técnicas de desarrollo de software unitarias y las pruebas de aceptación

1.4. JUSTIFICACIÓN

Todo proceso investigativo en el área tecnológica surge a partir de una necesidad de resolver problemas prácticos o simplemente para mejorar procedimientos que están perdiendo vigencia dentro de las organizaciones de cualquier naturaleza.

1.4.1. Justificación Económica

El desarrollo del sistema de registro y autorización de viajes tendrá un costo mínimo en la elaboración y tendrá las bondades de ser modificado en un futuro, además de realizarse en un lenguaje de programación como java y código libre.

Acceder a la información de manera confiable y fácil, permite agilizar el procesamiento de información y el control de uso de los datos personales.

1.4.2. Justificación Social

El presente proyecto surge como resultado de un interés por darle solución a algunos de los problemas que se genera dentro de la Defensoría de la Niñez y Adolescencia del Municipio de Cobija, especialmente aquellas que surgen por la autorización de viajes.

Con la implementación del sistema de autorización de viajes para menores de 18 años, en la Defensoría de la Niñez y Adolescencia del Municipio de Cobija, permitirá en gran manera llevar un control eficiente y tener una relación con las redes de protección que requieren o necesitan acceder a la información actual y de gestiones anteriores para realizar cualquier actividad para precautelar los derechos del niño, niña y adolescente, de esta manera mostrar una mayor credibilidad, responsabilidad, transparencia y confianza, el cual beneficiará a la población y la sociedad.

1.4.3. Justificación Técnica

Sin embargo la implementación de sistema significa un gasto para la institución. Gastos que contemplan el POA de la Defensoría de la Niñez y Adolescencia, los cuales se detalla a continuación:

- Un Switch con 8 puertos.
- Cable UTP un aproximado de 100 mts.
- 12 unidades de Conectores JR45.

1.5. ALCANCES

La Dirección de Género, Generacional y Familia, cuenta con dos unidades:

- La unidad de la defensoría de la niñez y adolescencia.
- La unidad de servicios legales municipales.
 - ◆ Refugio transitorio de acogida municipales.

Este trabajo propone dar soporte a la unidad denominada “Defensoría de la Niñez y Adolescente”, esta unidad es responsable del precautelar la integridad del menor conjuntamente con las sub-unidades de apoyo que responden a la defensoría como Trabajo Social, Psicología y Asesoría Legal.

Aplicando la metodología XP, y tomando como referencia a una de las fases que es la planificación, menciona que para desarrollar un sistema se debe recurrir a una interacción con el cliente y el grupo de desarrollo del sistema. Donde se ha llegado a un acuerdo, que se brindarían toda la información respecto a las autorizaciones de viaje, y que el cliente está satisfecho con la idea de implementar un nuevo sistema y/o que coadyuve en mejorar el sistema anterior.

El presente proyecto de grado está contemplado en la implementación del sistema de información para la Defensoría de la Niñez y Adolescencia del Municipio de Cobija, pero de acuerdo a la fase de planificación se limitará en la elaboración del sistema de información exclusivamente para la autorización de viajes, donde los requerimientos del sistema están dentro del marco institucional y de acuerdo a la posibilidad del cliente, los mismos presentan:

- El registro de los datos personales del menor y su respectiva identificación.
- El registro y digitalización de datos de los padres, tutores y/o apoderados, esto contempla la identificación.
- Reportes personalizados, algunos sobre las solicitudes realizadas e impresiones de la autorización de viajes.
- Se gestionará usuarios y perfiles, de acuerdo a la función del personal de la institución.
- Reportes estadísticos con representaciones gráficas de tortas y barras.

- Trabajo compartido en una red local compuesta por un servidor y varias computadoras clientes, dependiendo de la infraestructura de la institución.

Se considera como opcional, la implementación de los equipos de computación para mayor seguridad como:

- La firma digital, que comprende la compra del mecanismo de seguridad con la tecnología token comercial.
- La foto y huella dactilar, que se considera como procedimiento secundario tomando en cuenta los recursos económicos de la institución para adquirir mayor equipamiento.

1.6. RESULTADOS OBTENIDOS

Una vez concluido con el desarrollo del sistema de autorización de viajes se obtuvo los siguientes resultados:

- Se consideró el flujo de procesamiento de información del sistema actual, y se procedió a la planificación para lograr la historia de cada usuario y determinar la participación en el sistema.
- Concluido con la creación de cada historia de usuario se diseñó el modelo de vista con la tarjeta CRC, pese al requerimiento de la institución se modificó constantemente el diseño hasta concluir a algo agradable para que sea amigable el sistema.
- Una vez completado con las características de cada historia de usuario y la creación de la tarjeta CRC, se codificó el sistema utilizando herramienta de programación: Netbeans, lenguaje Java, los diagramas UML y tomando en consideración realizar la prueba y error antes de implementar el código y verificar el funcionamiento de cada módulo del sistema.
- Por último se probó mediante la implementación y ejecución de los procedimientos en menor tiempo, mediante técnicas de desarrollo de software unitarias y las pruebas de aceptación en especial el Junit que está incorporado en el editor Netbeans.

1.7. METODOLOGÍA Y HERRAMIENTAS UTILIZADAS

1.7.1. Metodología

La metodología ligera utilizada para el desarrollo del presente proyecto es la Programación Extrema (XP), que se basa en la simplicidad, la comunicación y la realimentación continua o reutilización del código desarrollado, Además que no se rige a un determinado lenguaje y se adapta a los requisitos en cualquier punto de vida del software.

Esta metodología XP está orientada a objetos y es una alternativa en cuanto a plazo, tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. Durante Las iteraciones de este ciclo tanto el cliente como el programador aprenden a mejorar el software.

A continuación se detalla las fases de la metodología:

Tabla 1.1 Fase de la metodología XP

Etapas	Metodología XP	Herramientas	Productos
1°	Planificación del Proyecto	Historias de Usuario	Definir las historias de los usuarios con los clientes.
		Release Planning	Se indicaran las historias de los usuarios que se crearan para cada versión del programa.
2°	Diseño del Sistema	Casos de Uso, Uso de Tarjetas CRC, diseño de diagramas UML, etc.	Documentación del análisis y diseño del sistema.
		UML, Enterprise Architect ,etc.	Diseños simples y sencillos del sistema
3°	Codificación	JAVA, MySQL, NetBeans versión 8.0.1, etc.	Módulos construidos y funcionando de acuerdo al diseño.
4°	Pruebas	Técnicas de pruebas con	Documentación de las

	JUnit	pruebas de la Caja negra y Caja blanca.
--	-------	---

Fuente: Elaboración Propia

1.7.2. Herramientas

Las herramientas utilizadas para el desarrollo del sistema de información que se emplearán son las siguientes:

- **Uml** es una notación de modelado visual, que utiliza diagramas para mostrar distintos aspectos de un sistema.
- **Java** es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems.
- **JUnit** es un API para realizar pruebas que fue diseñado para ser empleado en java.
- **JasperReport e IReport** es una combinación de librerías de java y software libre que permite el diseño e implementación de reportes.
- **MySQL** es un sistema de administración de bases de datos conjuntamente es una colección de estructurada de datos.
- La plataforma **Windows 7** sistema operativo en el cual se desarrollara el sistema por ser un entorno amigable.

1.7.3. Plataforma de desarrollo

Es el ambiente o entorno de software en el cual se desarrollara el sistema mediante:

- **Escritorio** se decidió realizar el proyecto en el entorno de escritorio ya hace uso de los recursos o un mejor aprovechamiento del hardware del equipo.

2. CAPITULO II

2.1 MARCO INSTITUCIONAL

2.2 MARCO LEGAL

2.3 MARCO TEORICO

2.3.1. INTRODUCCIÓN

2.1. MARCO INSTITUCIONAL

2.1.1. Marco Institucional de la Unidad de la Defensoría de la Niñez y Adolescencia del Municipio de Cobija

Una vez obtenida la estructura organizacional del gobierno autónomo municipal de Cobija de la gestión 2016 (ver anexo C), se podrá determinar cuál es la unidad o dirección que se tomara en cuenta para la elaboración de proyecto de grado.

De acuerdo al organigrama actualizado del Gobierno Autónomo Municipal de Cobija se considera:

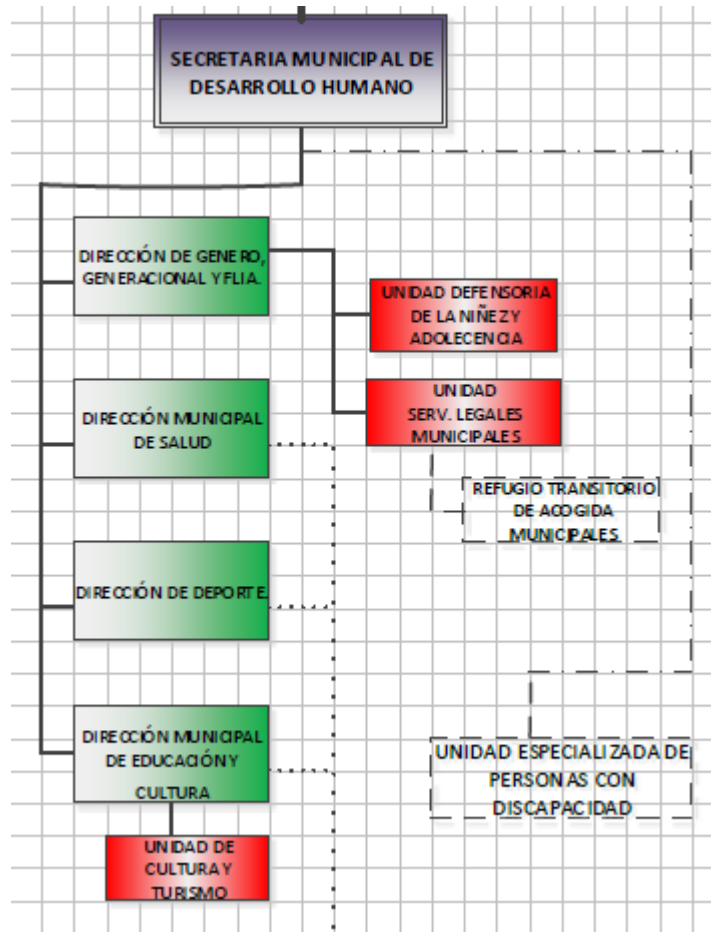


Ilustración 2.1 Organigrama Institucional Secretaría Municipal de Desarrollo Humano
Fuente: Página Oficial GAMC

La unidad de la defensoría de la niñez y adolescencia, tiene como objetivo principal:

- De prevenir, proteger y vigilar los derechos del niño, niña y adolescente.
- trabajos orientados a la prevención capacitando personas, difundiendo y promocionando sus servicios.

Del mismo modo cumple con las siguientes funciones:

- La autorización de viajes, donde el flujo de información radica en la recepción de documentos que acrediten que son progenitores del menor que comprende desde 0 a 18 años y puedan acceder a la autorización de viaje.
- El procedimiento consta de la presentación de los requisitos exigidos por la defensoría copias de documentos como el carnet de identidad, certificado de nacimiento, una vez entregado y verificado las copias, se registra en el sistema ofimático(Word o Excel),
- Elaboración de reporte de la autorización de viajes realizado mes, trimestral y anual.
- Trabaja en coordinación y colaboración con la unidad de trabajo social, psicología, asesoría legal.
- Coordinar con otros actores. contra la lucha de trata y tráfico de personas, asana, terminal de buses, seguridad ciudadana.

2.2. MARCO LEGAL

Según (Oportunidades, Guía de roles y funciones para las defensorías de la niñez y adolescencia, 2010) Los derechos de la niñez y adolescencia han dado lugar a una serie de normativas destinadas a su protección. Donde se, establece un listado de la normativa nacional e internacional de la materia:

2.2.1. Legislación Internacional

- Declaración de los Derechos y Deberes del Hombre de 1948 de la ONU.
- Convención de los Derechos Humanos, Pacto de San José de Costa Rica, 1969, ratificada por el Estado Boliviano mediante Ley N° 1430.
- Convención Interamericana sobre Conflicto en Leyes en Materia de Adopción de Menores, suscrita en la Paz Bolivia en fecha 24 de mayo de 1984.
- Convención sobre los Derechos del Niño, de Naciones Unidas de 1989, ratificada por Bolivia, mediante Ley N° 1152.
- Convención Interamericana sobre Restitución de Menores, Montevideo Uruguay 15 de Julio de 1989, adoptada y ratificada por Bolivia mediante Ley N° 1727 de 13 de Noviembre de 1996.
- Convención contra la tortura y otros tratos o penas crueles e inhumanos o degradantes suscrita en Nueva York el 14 de febrero de 1985 y ratificada por Bolivia mediante Ley N° 1939 de 10 de febrero de 1999.
- Peores Formas de Trabajo Infantil, Convenio 182, OIT, ratificado mediante Ley N° 2428 del 28 de Noviembre del 2002.
- Convención Interamericana sobre el Tráfico Internacional de Menores, de fecha 18 de marzo de 1994, ratificada por Bolivia mediante Ley N° 1725 de 13 de Noviembre de 1996.
- Estatuto de Roma de 17 de Julio de 1998, ratificada por Bolivia mediante Ley N° 2398 de 24 de mayo del 2002.
- Trabajo Forzoso, Convenio 29 OIT, ratificado mediante Ley N° 3031 del 29 de Abril 2005.

- La Abolición del Trabajo Forzoso, Convenio 105, OIT, ratificado mediante Ley N° 1119 del 1ro. de Noviembre del 1989.
- Protección del Salario, Convenio 95 OIT, ratificado por Decreto Supremo N° 14228 de 23 de Diciembre de 1976 y elevado a rango de Ley N° 2120 de 11 de Septiembre del 2000.
- Peores Formas de Trabajo Infantil, Convenio 182, OIT, ratificado mediante Ley N° 2428 del 28 de Noviembre del 2002.
- Pueblos Indígenas y Tribales, Convenio 169 de la OIT, ratificado mediante Ley 1257 de 11 de Diciembre de 1991.
- Protocolo para Prevenir, Reprimir y Sancionar la Trata de Personas, especialmente mujeres y niños, ratificado mediante Ley 2273 de 22 de Noviembre de 2001.
- Convenio para la Eliminación de la Trata de Personas y la Explotación de la Prostitución. Adhesión por Decreto Supremo N° 19777 de 13 de Septiembre de 1983 y el3vado a rango de Ley N° 2116 de 11 de Septiembre del 2000.
- Convenio de La Haya sobre Adopciones Internacionales, Ley N° 2314 de 24 de Diciembre del 2001.

2.2.2. Legislación Nacional

- Constitución Política del Estado Plurinacional, Sección V sobre los derechos de la niñez, adolescencia y juventud Ley N° 3942, de 21 de Octubre del 2008.
- Ley N° 1551 de Participación Popular, del 20 de Abril de 1994.
- Ley N° 696 de Municipalidades, del 10 de Enero de 1985.
- Ley N° 2026 CNNA, de 27 de octubre de 1999.
- Decreto Supremo N° 24447 de Reglamento al Código NNA, del 8 de abril de 2004.
- Ley N° 2616 Rectificación para el Registro de Nacimiento, del 18 de diciembre del 2003.
- Ley N° 3325 Trata y tráfico de Personas y otros Delitos Relacionados, del 18 de enero 2006.
- Ley N° 2033 De Protección a las Víctimas de Delitos contra la Libertad Sexual, de 29 de Octubre de 1999.

- Código Penal, Ley N° 1768 de 10 de Marzo de 1997. Código de Procedimiento Penal, Ley N° 1970 de 25 de marzo de 1999.
- Código Civil, Decreto Ley N° 12760 de 6 de agosto de 1975.
- Código de Procedimiento Civil, Ley N° 1760 de 28 de febrero de 1997.
- Código Familiar, Ley N° 996 de 4 de Abril de 1988.
- Ley de Violencia contra la Familia o Doméstica, N° 1674 de 15 de Diciembre de 1995.
- Ley N° 1678, Persona con Discapacidad de 15 diciembre de 1995.
- Ley General del Trabajo de 8 de Diciembre de 1942.
- Ley N° 2616, de Modificación de la Ley del Registro Civil del 18 de Diciembre de 2005.
- Ley N° 3933, de Búsqueda, Registro, y Difusión de Niños, Niñas y Adolescentes extraviados, de 18 de Septiembre de 2008.
- Ley N° 3934, de gratuidad de pruebas de ADN realizadas en las denuncias o querellas en los procesos penales por delitos de violación, abuso deshonesto, estupro, tipificados en el Código Penal cuyas víctimas sean niños, niñas y adolescentes, de 18 de septiembre del 2008.

2.3. MARCO TEÓRICO

2.3.1. Introducción

El presente proyecto de grado está dirigido a desarrollar un aplicación informática que coadyuve al registro, almacenamiento y reportes de las autorizaciones de viajes, para ello es necesario orientar el objetivo general donde se sistematizará los datos, actualizar la información de registros y contribuir al proceso de autorización de viajes de la niña, niño y adolescente, menores de 18 años.

La implementación del proyecto permitirá la interacción del personal de la defensoría con los padres, apoderados y otras redes de protección, en gran parte facilitará el seguimiento y registro de todas las autorizaciones de viajes que se le otorga a uno o varios menores de edades a solicitud de un adulto responsable, para lo cual se utiliza la metodología de desarrollo XP, donde el usuario y el programador interactúan mediante el requerimiento y necesidades de la institución, además se define las actividades para concluir el sistema. Así la defensoría de la niñez y adolescencia se beneficia con un aporte característico.

En este sentido la finalidad del sistema informático es reducir el tiempo del proceso administrativo y además se tendrá las bondades de relacionar la información con las subunidades de la Defensoría de la Niñez y Adolescencia y con otras redes de protección que velan conjuntamente el interés del menor y para evitar el traslado sin la autorización de ambos progenitores.

2.3.2. Sistema de información

Un sistema de información se define como un conjunto de procedimientos ordenados interrelacionados que obtiene, procesa, almacena y distribuye información para apoyar la toma de decisiones y el control de la institución.

También “Los sistemas de información proporcionan servicios a todos los demás sistemas de una organización y enlazan todos sus componentes en forma tal que estos trabajen con eficiencia para alcanzar el mismo objetivo” (Senn, 1999).

Los sistemas de información constan de los siguientes elementos:

- La información (datos). Conjunto de datos estructurados según los mensajes a comunicar.
- Permite el acceso controlado a una base de información(un ejemplo puede ser catálogo de bibliotecas)
- Los beneficiarios de la información. Los miembros de la organización y agentes relacionados con ella.
- Los elementos soporte. Proceso de tratamiento de información, sistemas de análisis de datos, procedimientos de comunicación o difusores de información y soportes de información.

2.3.3. Redes de Datos

Las redes de datos son fuentes centrales de información y comunicación que tienen la posibilidad de compartir con carácter universal la información entre grupos de computadoras y usuarios.

Un componente vital de la era de la información, es la generalización del ordenador o computador personal (PC) y de la red de área local (LAN), que durante la década de los 80 ha dado lugar a la posibilidad de acceder a información en bases de datos remotas.

2.3.4. Herramientas para la programación

- **Java** se trata de un poderoso y flexible lenguaje de programación con el cual se puede desarrollar desde aplicaciones para celulares hasta páginas web. Obviamente se convierte en una herramienta óptima para desarrollo de aplicaciones de escritorio.
El motivo por el que se seleccionó a java como herramienta de desarrollo, es el amplio conocimiento que ambos programadores tienen del lenguaje. En segundo lugar existe una API de pruebas desarrollada para trabajar con JAVA especialmente diseñada para proyectos desarrollados con XP.
- **NetBeans** es un IDE de licencia libre para desarrollar aplicaciones en java y de múltiple plataforma. Si bien no es el único IDE disponibles para java, a juicio de los programadores involucrados en el proyecto, se opto por la versión 8.0.1 es la más adecuada, por lo cual se convirtió en la mejor elección. Por otro lado cuenta con soporte para JUnit.

- **JUnit** es un API para realizar pruebas que fue diseñado para ser empleado en java. Un aspecto importante es que cumple con la mayoría de las recomendaciones realizadas por XP en lo que a pruebas se refiere, de las cuales se destaca el permitir hacer pruebas autónomas. Por otro lado algunos autores lo recomiendan para desarrollar aplicaciones en JAVA empleando XP.
- **JasperReport e iReport** es una combinación de librerías de java y software libre que permiten el diseño e implementación de reportes impresos. Entre las ventajas más importantes se resalta flexibilidad y facilidad de empleo.

2.3.5. Plataforma de desarrollo

Aplicación de Escritorio, son aquella que se encuentran instaladas en el ordenador o sistema de almacenamiento y podemos ejecutarlo sin internet en nuestro sistema operativo.

Algunas características de la aplicación de escritorio:

- Permite mejor aprovechamiento del hardware del equipo.
- Menor tiempo en el desarrollo
- Mejores tiempos de respuesta
- No dependencia de internet
- Si se desea y el software lo permite, puede trabajar también por internet.
- Siempre funcionará, sin dependencia del navegador web.

2.3.6. Elementos de la implementación de los sistemas de información.

2.3.6.1. Arquitectura de cliente servidor

(Hurtado, 2006), define como un Sistema donde el **Cliente** es una máquina que solicita un determinado servicio y se denomina **Servidor** a la máquina que lo proporciona.

En este modelo de trabajo, las tareas son repetidas entre el equipo Servidor y el Cliente, son un típico ejemplo las aplicaciones que solicitan, a través de interfaces, información a un servidor de Bases de Datos alojado en otro equipo y que ambas están comunicadas por una red de datos.

2.3.6.2. Base de datos

(Carlos Coronel, 2011), dice que una Base de Datos es una estructura computarizada compartida e integrada que guarda un conjunto de datos del usuario final y un conjunto de metadatos.

Además es una colección de información organizada, y relacionada entre ella, esta, pueda estar constituido de gran cantidad de información, con posibilidad de acceder en cualquier momento y de manera concurrente, este acceso generalmente se realiza a través de instrucciones denominado consultas.

Una Base de Datos cuenta con las siguientes propiedades:

- a) Representación de algún aspecto del mundo real.
- b) Es un conjunto de datos lógicamente coherente e interrelacionado.
- c) Toda base de datos se diseña con un propósito específico.

2.3.6.3. Gestor de base de datos

Es un software que permite utilizar y actualizar los datos almacenados en una o varias Base de Datos por uno o varios usuarios al mismo tiempo. Su objetivo fundamental consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, de forma que no le sea necesario conocer el modelo de almacenamiento de los datos en la computadora, ni el método de acceso empleado.

Según (Montero), “un **Sistema Gestor de Bases de Datos** o **SGBD**, también llamado DBMS (Data Base Management System) es como una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos.”

2.3.6.4. MySQL

Es un sistema de gestión de bases de datos relacionales, multihilo y multiusuario con más de seis millones de instalaciones, desarrollada como software libre. MySQL es muy utilizado en aplicaciones web, como DRUPAL o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla.

Su popularidad como aplicación web está muy ligada al lenguaje de programación PHP, que a menudo aparece en combinación con MySQL. Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.

- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

2.4. METODOLOGÍA DE DESARROLLO

Las Metodologías de Desarrollo de Software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto software.

Dichas metodologías pretenden guiar a los desarrolladores al crear un nuevo software, pero los requisitos de un software a otro son tan variados y cambiantes, que ha dado lugar a que exista una gran variedad de metodologías para la creación del software. Estas se podrían clasificar en dos grandes grupos:

- Las metodologías orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán. Estas metodologías son llamadas **Metodologías Pesadas**.
- Las metodologías orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando. Estas son llamadas **Metodologías ligeras/ágiles**.

2.4.1. Metodología pesada

Según (Acuña, 2009) “Teniendo en cuenta la filosofía de desarrollo de las metodologías, aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales o Pesadas”.

Estas metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente

a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

Entre las metodologías tradicionales o pesadas podemos citar:

- RUP (Rational Unified Procces)
- MSF (Microsoft Solution Framework)
- Win-Win Spiral Model
- Iconix

Pero los proyectos de desarrollo de software en la actualidad incluyen desafíos muy diferentes a los que se presentan al construir puentes y casas, por lo que no sorprende que los métodos tradicionales de desarrollo de software estén en crisis. Tradicionalmente los proyectos se dividen en etapas bien diferenciadas: Análisis de Factibilidad, Análisis de Requerimientos, Diseño, Programación, y Testeo.

Generalmente se trata de que haya retroalimentación (feedback en inglés) entre las etapas contiguas, de tal forma de que, por ejemplo, haya un momento en que se mejoren los Requerimientos en base a comentarios, sugerencias y necesidades de los responsables del Diseño. Sin embargo, esta forma de desarrollar software genera muy serios problemas, debido a que al comienzo del proyecto, que es cuando menos se conocen las características del problema que resolver, se toman las decisiones de mayor relevancia e impacto en el resto del proyecto.

Como propuesta de solución a estos problemas han surgido una serie de métodos ágiles de desarrollo de software y manejo de proyectos en general.

2.4.2. Metodología ágiles

Según (Sommerville, 2011), Los procesos de desarrollo del software rápido se diseñan para producir rápidamente un software útil y además se realiza una serie de incrementos, y cada uno de ello incluye una nueva funcionalidad del sistema y este proceso comparte una serie de características fundamentales:

- Los proceso de especificación, diseño e implementación están entrelazados.

- El sistema se desarrolla en diferentes versiones.
- Las interfaces de usuario del sistema se desarrollan usando con frecuencia un sistema de elaboración interactivo, que permita que el diseño de la interfaz se cree rápidamente en cuanto se dibujan y colocan icono en la interfaz.

Los métodos ágiles son métodos de desarrollo incremental donde los incrementos son mínimos y por lo general, se crean las nuevas liberación del sistema, y cada dos a tres semanas se ponen a disposición del cliente.

La diferencia que existe entre metodologías Ágiles y Tradicionales:

Tabla 2.2 Tabla de diferencia entre metodologías ágiles y metodología no ágiles

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestamente internamente (por el equipo)	Impuestas externamente
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.
Grupos pequeños y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.

Fuente: Elaboración Propia

2.4.3. Metodología XP

Según (Cruz Castillo, 2011), La programación extrema es quizás el método ágil mejor conocido y más ampliamente usado. “la programación extrema es un método ágil bien conocido que integra un rango de buenas prácticas de programación, como la liberación frecuente y mejoramiento del software y la participación del cliente en el equipo de desarrollo”, (Sommerville, 2011).

XP (Programación Extrema), es una metodología para el desarrollo de software, clasificada como uno de los más ligeros, en contraposición con las metodologías pesadas como el Proceso Unificado.

(Kent Beck, 2000), fue el desarrollador de esta metodología, en el cual se basa en la simplicidad, la comunicación y la retroalimentación o reutilización de código.

Entre los objetivos de esta metodología se encuentran la satisfacción del cliente, además de potenciar al máximo el trabajo en equipo (**clientes, desarrolladores, jefes**).

(Beck, 2000), argumenta que, Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder, el problema es la incapacidad de adaptarnos a dicho cambio cuando este tiene lugar.

2.4.3.1. Fases de la Metodología XP

La XP (Programación Extrema), define **4 fases** bien definidas que permiten cubrir todo el proceso de desarrollo de software: Planificación, Diseño, Desarrollo y Pruebas (**ver Anexo B**).

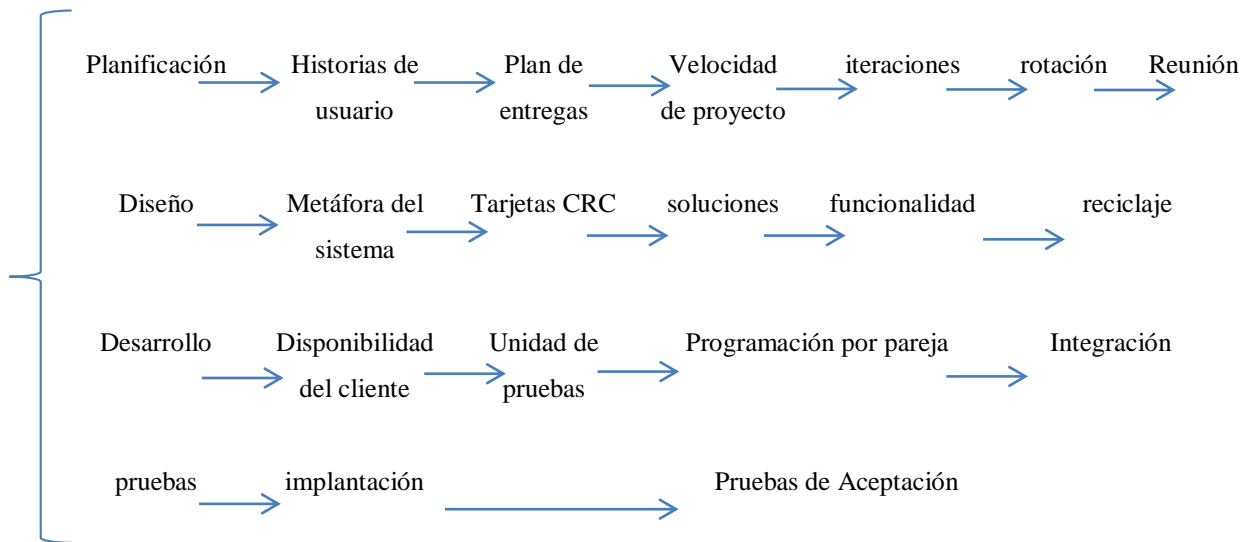


Ilustración 2.2 Fases de la Programación Extrema
Fuente: Elaboración Propia

2.4.3.2. Fase de la Planificación

Es la etapa de interacción entre el cliente con la parte técnica. En esta fase el cliente dará a conocer lo realmente necesario y el orden de prioridad del proyecto a través de las **Historias de Usuario**, se define las versiones del software y el cronograma de actividades.

(Fernández, 2002), define como tres actividades a desarrollar:

- a) **Redactar las historias de usuarios:** Es donde se describe de manera sintetizada cierto proceso, en realidad tienen el mismo propósito que los Casos de Uso de la metodología Proceso Unificado.
- b) **Se crea un plan de entregas:** Es donde se planifica determinados tiempos requeridos para el desarrollo de las **historia de usuarios**.
- c) **Se divide el proyecto en iteraciones:** Es donde una iteración corresponde a una cierta cantidad de historias de usuario.

2.4.3.3. Fase de Diseño

Para la Fase de Diseño la Programación Extrema, define los siguientes principios:

- a) **Simplicidad:** Es donde se realiza las cosas de la forma más simple posible.
- b) **Tarjetas CRC (Cargo o Clase, Responsabilidad y Colaboración):** En estas tarjetas se describen las clases, los métodos y la relación con otras clases.

Clase : Nombre de la Clase	
Responsabilidad: Método de la clase	Colaboración: Lista de clases con los que se relaciona

Ilustración 2.3 Tarjeta clase-responsabilidad-colaboración
Fuente: Elaboración Propia

- c) **No se añade funcionalidad en las primeras etapas,** es decir, trabajar en lo que se ha fijado para el día.

2.4.3.4. Fase de Desarrollo

En esta etapa se trabaja bajo los siguientes lineamientos:

- a) El cliente debe ser parte del proyecto, es decir está disponible cuando se lo necesite.
- b) Escribir el código de acuerdo a buenas prácticas de programación.
- c) Implementar la Base de Datos.
- d) Realizar las pruebas.

2.4.3.5. Fase de Pruebas

- a) Se efectuaran las siguientes pruebas:

Pruebas	Técnica de la Caja Negra	Documentación de las pruebas de Caja Negra.
	Técnica de la Caja Blanca	Documentación de las pruebas de Caja Blanca.

Ilustración 2.4 Tabla de aplicaciones de pruebas
Fuente: Elaboración propia

2.5. LENGUAJE DE MODELADO UNIFICADO

Según (Fontela, 2011), UML es el acrónimo en inglés para “Lenguaje unificado de modelado”, es una notación de modelos visual, que utiliza diagramas para mostrar distintos aspectos de un sistema, donde permite visualizar, especificar, construir y documentar los artefactos de un sistema de software.

En UML existen modelos **estáticos o estructurales** que sirven para modelar objetos, clases, relaciones y sus agrupaciones presentes de un sistema y por otro lado existen **cuestiones dinámicas o de comportamiento** que define como evolucionan esos objetos a lo largo del tiempo y cuáles son las causas de esa evolución

Estos dos modelos hacen que UML se interrelacione sustancialmente entre ambos, Aquí es donde radica su carácter de “Unificado”,

A continuación se detallara los diagramas de UML 2.2 son:

Los diagramas estructurales o estáticos:

- Diagrama de caso de uso.
- Diagrama de objeto (estático).
- Diagrama de clases.
- Diagrama de paquetes.
- Diagrama de componentes.
- Diagrama de despliegue.

- Diagrama de estructuras compuestas.

Los diagramas de comportamiento o dinámico:

- Diagrama de secuencias.
- Diagrama de comunicación (o de colaboración).
- Diagrama de máquina de estados o de estado.
- Diagrama de actividades.
- Diagrama de visión global de la interacción.
- Diagrama de tiempos.

UML ofrece una forma estándar para escribir un plano del sistema, incluyendo elementos conceptuales tales como proceso de negocios y funciones del sistema, como también elementos concretos tales como el lenguaje de programación, esquemas de base de datos y componentes de software.

2.6. ENTERPRISE ARCHITECT

Según (systems, 2016), Es una plataforma avanzada de modelado, Proceso de análisis, diseño y construcción con herramientas de modelado visual más productivas y precisas.

Algunas de las ventajas de usar Enterprise Architect:

Único en su capacidad de reunir dominios complejos y crear una única visión de acceso unificado:

- Combina herramientas de gama alta, un compromiso con los estándares abiertos
- Tiene la habilidad para capturar y controlar la complejidad y desempeña un papel fundamental de confianza en la entrega de alta calidad de soluciones

2.7. LENGUAJE DE PROGRAMACIÓN

- Un lenguaje de programación se define como **una técnica de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora.**

2.7.1. Programación estructurada

La programación estructurada consiste en escribir un programa de acuerdo con unas reglas y un conjunto de técnicas. Las reglas son: el programa tiene un diseño modular, los módulos son diseñados descendientemente, cada módulo de programa se codifica usando tres estructuras de control (secuencia, selección e iteración); es el conjunto de técnicas que han de incorporar: recursos abstractos; diseño descendente y estructuras básicas de control. Descomponer un programa en términos de recursos abstractos consiste en descomponer acciones complejas en términos de acciones más simples capaces de ser ejecutadas en una computadora.

El diseño descendente se encarga de resolver un problema realizando una descomposición en otros más sencillos mediante módulos jerárquicos. El resultado de esta jerarquía de módulos es que cada módulo se refina por los de nivel más bajo que resuelven problemas más pequeños y contienen más detalles sobre los mismos.

Las estructuras básicas de control sirven para especificar el orden en que se ejecutarán las distintas instrucciones de un algoritmo. Este orden de ejecución determina el flujo de control del programa.

- La programación estructurada significa:
 - El programa completo tiene un diseño modular.
 - Los módulos se diseñan con metodología descendente (puede hacerse también ascendente).
 - Cada módulo se codifica utilizando las tres estructuras de control básicas: Secuenciales, selectivas y repetitivas.
 - Estructuración y modularidad son conceptos complementarios (se solapan).

2.7.2. Programación Orientado a Objetos (OOP)

Según (Cadenhead, 2012), La programación orientada a objetos aporta un nuevo enfoque a los retos que se plantean en la programación estructurada cuando los problemas a resolver son complejos, además se describe como una forma de programar elegantemente un programa informático, donde los objetos existen de forma independiente e interactúan de forma concreta y se puede combinar con otros objetos para crear algo mayor.

En el paradigma orientado a objetos, el programa se organiza como un conjunto finito de objetos que contienen datos y operaciones, que llaman a esos datos y que se comunican entre sí mediante mensajes.

Cuando se trata de resolver un problema con orientación a objetos, dicho problema no se descompone en funciones como en programación estructurada tradicional, sino en objetos. El pensar en términos de objetos tiene una gran ventaja: se asocian los objetos del problema a los objetos del mundo real.

2.7.2.1. Conceptos básicos

- **Clases:** Las clases son como plantillas o modelos que describen como se construyen ciertos tipos de objeto. Cada vez que se construye un objeto de una clase, se crea una instancia de esa Clase ("instance"). Una clase es una colección de objetos similares y un objeto es una instancia de una clase. Se puede definir una Clase como un modelo que se utiliza para describir uno o más objetos del mismo tipo.
- **Objeto:** Los objetos son las cosas físicas y conceptuales que encontramos en el universo alrededor de nosotros. Hardware, software, documentos, seres humanos, y existe variada información. Los objetos como tal se crean por medio de una clase de objetos como plantillas.
- **Mensaje:** Representa una acción a tomar por un determinado objeto.
- **Instancia:** Es la concreción de una clase.
- **Método:** consiste en la implementación de una clase de un protocolo de respuesta a los mensajes dirigido a los objetos de la misma.

2.7.2.2. Propiedades fundamentales de la orientación a objetos:

Las propiedades que se suelen considerar, no son exclusivas de este paradigma, ya que pueden existir en otros paradigmas, pero en su conjunto definen claramente los lenguajes orientados a objetos. Estas propiedades son:

- a) **Abstracción (tipos abstractos de datos y clases):** La abstracción es la propiedad de los objetos que consiste en tener en cuenta sólo los aspectos más importantes desde un punto de vista determinado y no tener en cuenta los restantes aspectos.
- b) **Encapsulado de datos:** El encapsulado o encapsulación de datos es el proceso de agrupar datos y operaciones relacionadas bajo la misma unidad de programación. En el caso de que los objetos posean las mismas características y comportamiento se agrupan en clases (unidades o módulos de programación que encapsulan datos y operaciones).
- c) **Ocultación de datos:** La ocultación de datos permite separar el aspecto de un componente, definido por su interfaz con el exterior, de sus detalles internos de implementación. Los términos ocultación de la información (information hiding) y encapsulación de datos (data encapsulation) se suelen utilizar como sinónimos, pero no siempre es así y muy al contrario son términos similares pero distintos.
- d) **Herencia:** Es una propiedad que permite a un objeto (clases) heredar los atributos y el comportamiento de unos ya existentes. Esto permite crear "Sub-Clases" denominadas clases derivadas que comparten las propiedades de la Clase de la cual derivan (Clase base). Las Clases derivadas heredan código y datos de la clase base, asimismo incorporan su propio código y datos especiales. Se puede decir que la herencia permite definir nuevas Clases a partir de las Clases ya existentes.
- e) **Polimorfismo:** La propiedad de polimorfismo es aquella en que una operación tiene el mismo nombre en diferentes clases, pero se ejecuta de diferentes formas en cada clase. Así, por ejemplo, la operación abrir se puede dar en diferentes clases: abrir una puerta, abrir una ventana, abrir un periódico, abrir un archivo, abrir una cuenta corriente en un banco, abrir un libro, etc. En cada caso se ejecuta una operación diferente aunque tiene el mismo nombre en todos ellos abrir. El polimorfismo es la propiedad de una operación de ser interpretada

sólo por el objeto al que pertenece. Y en un sentido literal, Polimorfismo significa la cualidad de tener más de una forma.

2.8. MÉTRICAS DE CALIDAD ISO/IEC 9126

La norma ISO 9126 es un estándar internacional para la evaluación del software. Esta supervisado por el proyecto SQuaRe, ISO 25000:2005, el cual sigue los mismos conceptos.

El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso.

El modelo de calidad establecido en la primera estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y sub características de la siguiente manera:

2.8.1. Funcionalidad

Es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas.

- **Adecuación.** La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.
- **Exactitud.** La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.
- **Interoperabilidad.** La capacidad del software de interactuar con uno o más sistemas específicos.
- **Seguridad.** La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados

2.8.2. Confiabilidad

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizando en condiciones específicas.

- **Madurez.** La capacidad que tiene el software para evitar fallas cuando encuentra errores.
- **Tolerancia a errores.** La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.
- **Recuperabilidad.** La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.

2.8.3. Usabilidad

La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva.

- **Entendimiento.** La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.
- **Aprendizaje.** La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.
- **Operabilidad.** La manera como el software permite al usuario operarlo y controlarlo.
- **Atracción.** La presentación del software debe ser atractivo al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario.

2.8.4. Eficiencia

La eficiencia del software es la forma del desempeño adecuado, de acuerdo a al número recursos utilizados según las condiciones planteadas. Se debe tener en cuenta otros aspectos como la configuración de hardware, el sistema operativo, entre otros.

- **Comportamiento de tiempos.** Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas. Ejemplo, ejecutar el procedimiento más complejo del software y

esperar su tiempo de respuesta, realizar la misma función pero con más cantidad de registros.

- **Utilización de recursos.** La capacidad del software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo requerimientos o condiciones establecidas. Ejemplo, los recursos humanos, el hardware, dispositivos externos.

2.8.5. Capacidad de mantenimiento

La capacidad de mantenimiento es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales.

- **Capacidad de ser analizado.** La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.
- **Cambiabilidad.** La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.
- **Estabilidad.** La forma como el software evita efectos inesperados para modificaciones del mismo.
- **Facilidad de prueba.** La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.

2.8.6. Portabilidad

La capacidad que tiene el software para ser trasladado de un entorno a otro.

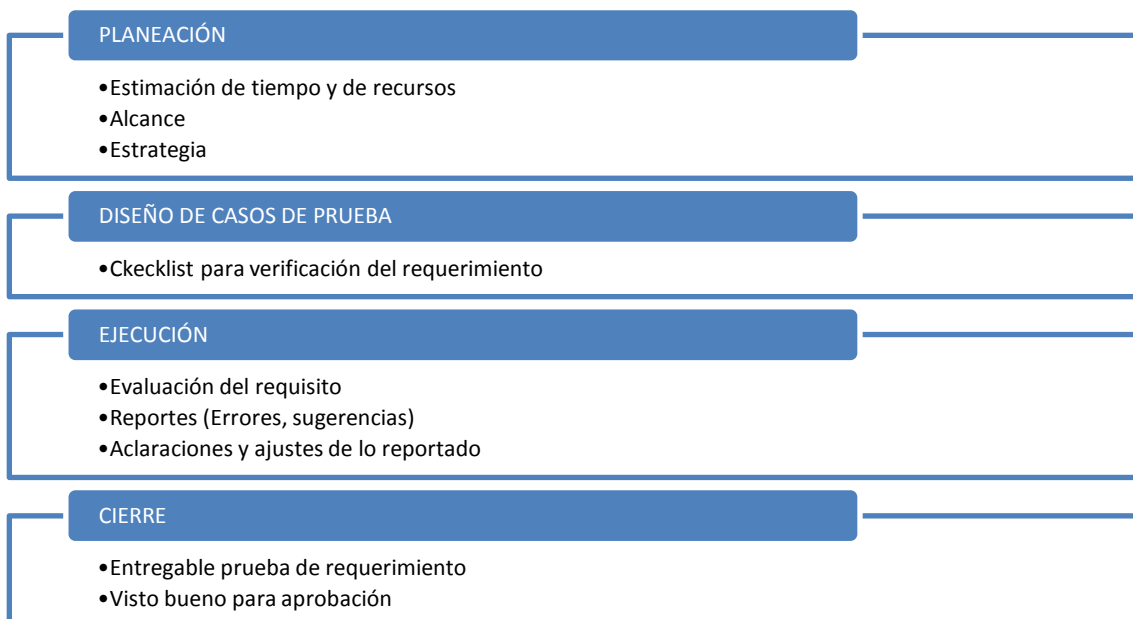
- **Adaptabilidad.** Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).

- **Facilidad de instalación.** La facilidad del software para ser instalado en un entorno específico o por el usuario final.
- **Coexistencia.** La capacidad que tiene el software para coexistir con otro o varios software, la forma de compartir recursos comunes con otro software o dispositivo.
- **Reemplazabilidad.** La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo. Ejemplo, la reemplazabilidad de una nueva versión es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor.

2.9. PRUEBA DE REQUERIMIENTOS

El objetivo de las pruebas de verificación es buscar discrepancias entre los requerimientos y la ejecución del software.

Partiendo de lo anterior se proponen las siguientes actividades para que las pruebas realizadas a los productos desarrollados mediante esta metodología sean correcta, se tendrán en cuenta las siguientes fases:



2.9.1. Planeación

En esta fase se define el alcance y limitaciones de la prueba, la estrategia utilizada para la ejecución de la prueba, los requerimientos para poder realizar las pruebas (documentación, recurso humano y recurso tecnológico), los tiempos de estimados de duración de la misma y los criterios para terminación.

ETAPAS	FORMATO	OBJETIVO
1	FMR_001_Identificación de Necesidades	Presentar una descripción de lo que se requiere desde la perspectiva del cliente para resolver la necesidad u oportunidad de mejora identificada.
2	FMR_002_Entrevista	Sugerir preguntas que permitan detallar los requerimiento..
3	FMR_003_Matrices_Trazabilidad	Relación de documentación, Revisión de pares, registró de control de cambio.
4	FMR_004_1_00X_Especificacion Funcional	Especificar detalladamente la funcionalidad(Requerimiento funcionales y no funcionales) de los componente del sistema y sus iteraciones
	FMR_004_2_00X_Caso de Uso	
	FMR_004_3_00X_Especificacion de Diagramas	
5	FMR_005_1_Planeacion Prueba de requerimiento	Presentar estrategias, alcance, estimación de tiempo, de la prueba de requerimiento.
	FMR_005_2_Diseño de Casos de Prueba	Identificar posibles casos de prueba de requerimientos

	FMR_005_3_Ejecucion de caso de prueba	Presentar resultados de ejecución de los casos
	FMR_005_4_Informe Final de Prueba	Presentar informe de cierre con los aspectos más relevantes de la ejecución
6	FMR_006_Control de Cambios	Describir la situación de cambio solicitada por el cliente

Ilustración 2.5 Formulario de Requerimientos
Fuente: Metodología Gestión de Requisitos

2.9.2. Diseño de caso de prueba

A continuación se presenta algunas de las características para evaluar dichos requerimientos:

Características	Descripción
Completo	Todo los ítems necesarios para la especificación de la solución están incluidos
Correcto	Cada ítem es libre de errores
Preciso	No ambiguo y claro. Cada ítem es exacto y no vago; hay una sola interpretación; ítem es entendido; la especificación es fácil de entender
Consistente	Ningún ítem entra en conflicto con otro de la especificación
Relevante	Cada ítem es pertinente al problema y su solución
Probable(testable)	Durante el desarrollo del programa y las pruebas de aceptación, es posible determinar si el ítem ha quedado satisfecho.
Factible	Cada ítem puede ser implementado con las técnicas, herramientas, recursos y personal disponible y dentro de las limitaciones de costo y calendarización.
Registrabilidad(Traceable)	Cada ítem puede ser seguido durante cada etapa.
Libre de detalle de diseño	La especificación de requerimiento son declaración de los requerimientos que se deben satisfacer por la solución del problema y no se deben ocultar por soluciones propuestas al problema.

Manejable	Los requerimientos son expresados de tal manera que cada ítem pueda ser cambiado sin causar un gran impacto a los demás ítems.
------------------	--

Ilustración 2.6 Diseño de Caso de Prueba
Fuente: Metodología Gestión de Requisitos

2.9.3. Ejecución de caso de prueba

En esta fase se evaluará cada uno de los requerimientos (casos de uso) de acuerdo al checklist definidos como casos de prueba (conciso, contraposición, ambigüedad y completitud, entre otras) se reportarán las consideraciones, errores, sugerencias, y se realizarán reuniones para hacer aclaraciones y definir si las consideraciones planteadas van a ser solucionadas o si el requerimiento es correcto como está.

2.9.4. Cierre

En esta fase una vez se haya completado la ejecución con resultados satisfactorios y ajustes correspondientes, se realizará el cierre de la prueba donde se dará el visto bueno sobre los requisitos evaluados.

3. CAPITULO III

3.1 MARCO APLICATIVO

3.1. MARCO APLICATIVO

El desarrollo del Sistema de Información, comienza con la elaboración de un diagnóstico y la situación actual de la institución, luego se procede a realizar las historias de usuarios, con el objeto de obtener el plan de entregas, posteriormente se divide el sistema en cuatro partes, que conforman las iteraciones en base al planificación, diseño, desarrollo y prueba de funcionamiento.

3.2. DIAGNOSTICO PRELIMINAR

La Defensoría de la Niñez y Adolescencia es una unidad dependiente de la Dirección de Género, Generacional y Familia, quienes están encargadas de velar la seguridad del menor de acuerdo a las normas y reglamento estipulado en la ley.

Dada la información proveniente por parte de la defensoría, se menciona que una de las actividades que realiza la defensoría es la autorización de solicitud de viajes para que un menor pueda viajar a cualquier parte de Bolivia.

Según la posibilidad de proveer información dentro de la defensoría la parte encargada de la autorización de viajes, se vio más interesada en la implementación de un sistema de autorización de viajes es ahí donde empieza el diagnóstico y la preposición de dotar información del flujo de procedimiento donde se registran los datos personales y la documentación para obtener la autorización como ser: copia de carnet o copia de certificado de nacimiento, documentos del padre o la madre que no viaja y del responsable a cargo del menor, los mismos son requisitos indispensables y dicha documentación es registrada, guardada en hoja de cálculo Excel, y se tiene una variada copia de autorizaciones almacenadas y al instante de buscar la información específica, es demoroso encontrarlo

SECRETARIA MUNICIPAL DE DESARROLLO HUMANO
DIRECCION DE GENERO, GENERACIONAL Y FAMILIA
Defensoría de la Niñez y Adolescencia

AUTORIZACION DE VIAJE AL INTERIOR
Cobija, 3 ENERO 2017
D.N.N.A. CITE No. 59 / 2017

Conforme a las disposiciones contenidas en los Arts. 2,3,4,11 y 14 de la Convención Internacional sobre los derechos de los Niños y los Arts. 158, 196 y 207 inc. 2 y 3 de la Ley 2026, la Defensoría Municipal de la Niñez y Adolescencia, con el fin de evitar el traslado de Niños, Niñas y Adolescentes al interior sin la autorización de

POR TANTO SE AUTORIZA EL VIAJE DE :

R..... D..... T..... V..... DE 9 AÑOS DE EDAD
D..... DE 2 AÑOS DE EDAD

Quien realizará el viaje a la ciudad de : COBIA-LA PAZ-ORURO-COCHIABAMBA- Por el Tiempo de : 90 DIAS

Nombre	Fecha de modifica...	Tipo	Tamaño
THIAGO TORREZ HURTADO 52	03/01/2017 04:00 ...	Hoja de cálculo d...	55 KB
CRISS DANIEL MONTENEGRO NAGATA 53	03/01/2017 04:04 ...	Hoja de cálculo d...	55 KB
JACQUELIN COPPE MEDINA 55	03/01/2017 04:08 ...	Hoja de cálculo d...	55 KB
HNAS JUSTINIANO MACUAPA 56	03/01/2017 04:14 ...	Hoja de cálculo d...	55 KB
CORINA ANAHE PAREDES ALLAGA 58	03/01/2017 04:17 ...	Hoja de cálculo d...	55 KB
HNOS.SANCHEZ GUAYAO	03/01/2017 04:23 ...	Hoja de cálculo d...	54 KB
ALEJANDRA MALALE BARROZO 60	03/01/2017 04:24 ...	Hoja de cálculo d...	55 KB
MARIELA QUISPE 61	03/01/2017 04:27 ...	Hoja de cálculo d...	55 KB
ALEX DAVID ARAUJO CORDERO 18	03/01/2017 04:35 ...	Hoja de cálculo d...	54 KB
HNOS.CHOQUE OCHOA	03/01/2017 04:36 ...	Hoja de cálculo d...	54 KB
DEVORA SARAH TICONA MASCO 63	03/01/2017 04:36 ...	Hoja de cálculo d...	55 KB
NAHOMI LOPEZ SOSA 65	03/01/2017 04:44 ...	Hoja de cálculo d...	55 KB
ROSALIA CAMARA GONZALES 67	03/01/2017 04:52 ...	Hoja de cálculo d...	55 KB

Ilustración 3.7: Registro de Autorización de Viajes

Fuente: Defensoría de la Niñez y Adolescencia

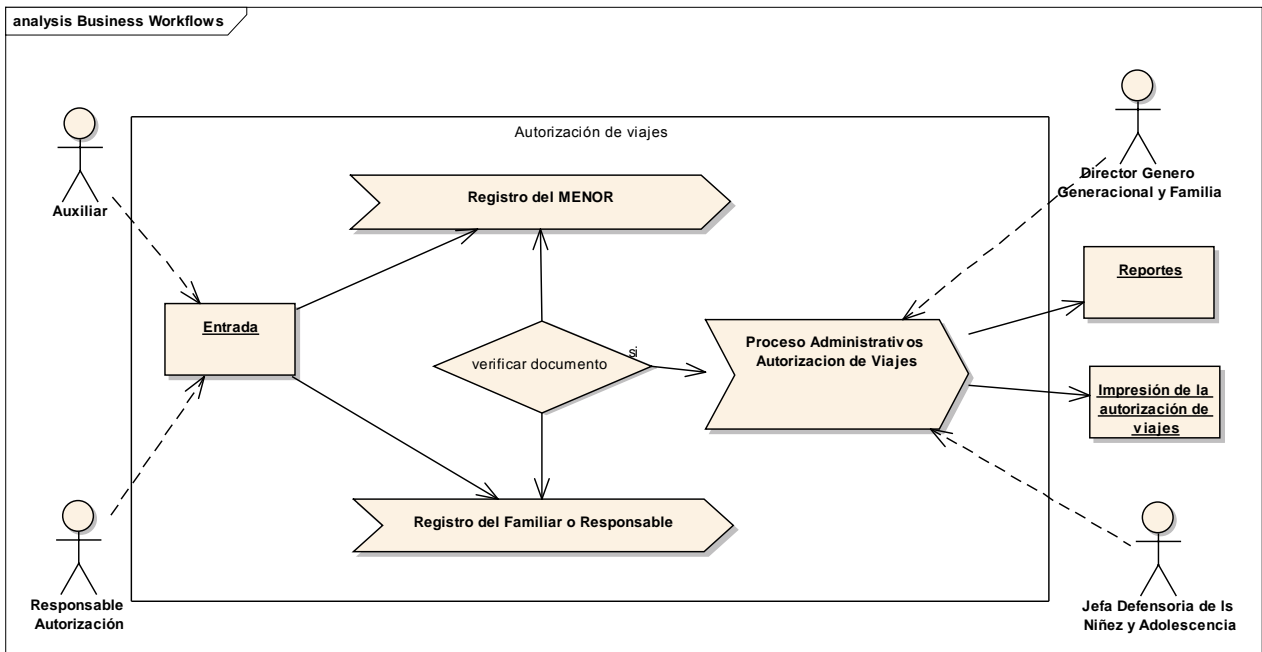


Ilustración 3.8: Diagrama proceso

Fuente: Elaboración Propia

De acuerdo la metodología XP, la planificación se plantea entre el dialogo permanentemente entre la parte institucional y técnica del proyecto, ahí es donde se terminaron los alcances hasta donde llegara la implementación del proyecto.

3.3. PLANIFICACIÓN DEL SISTEMA

3.3.1. Identificación de usuarios

Siguiendo la metodología XP, se logró identificar a los siguientes usuarios que representara el sistema de autorización de viajes.

3.3.1.1. Usuario Principales

De acuerdo a la metodología y sus fases de logro identificar a los siguientes usuarios:

- **Administrador del Sistema:** es la persona responsable de administrar el sistema, también es el responsable de la base de datos y tiene acceso a toda la información existente del sistema.
- **Director de Genero Generacional y Familia:** es la persona encargada de verificar, revisar los reportes de las autorizaciones de viajes.
- **Jefe de la Defensoría de Niñez y Adolescencia:** es la persona de verificar los requisitos y remitir los reportes de las autorizaciones de viajes.
- **Operador Secretaria:** es la persona responsable del control físico, verificar la documentación o requisitos de menor que viaja para poder emitir la autorización.
- **Operador Auxiliar de secretaria:** verificar la documentación o requisitos de menor que viaja para poder emitir la autorización

3.3.1.2. Usuarios secundarios

- **Policía Nacional.-** coordina con la defensoría en cualquier situación que afecte al menor.
- **Seguridad Ciudadana.-** apoyo en la vigilancia y colaboración a la defensoría de la niñez y adolescencia.
- **Assana.-** encargado de la seguridad aeroportuaria y en colaboración con defensoría de la niñez y adolescencia controlan el viaje que realizar un menor.
- **Terminal.-** encargado de la terminal de buses, control y verificando salidas que realizar un menor.

3.3.2. Historia de usuarios

Después de la identificación de usuarios se tomara como base, para el desarrollo de historia de usuarios.

3.3.2.1. Gestión de usuarios:

Un usuario es la persona que tendrá acceso al sistema de información. El mismo deberá realizar su respectiva autenticación de acuerdo a su rol signado: Administrador, Director, Jefa, Secretaria Auxiliar de secretaria y otros usuarios especiales por ejemplo (Assana, Terminal y Seguridad Ciudadana).

Tabla 3.3 Gestión de usuarios

Historia de Usuario	
Numero: 01	Usuario: Administrador
Nombre de historia: Gestión de usuarios	
Prioridad Alta	Riesgo de desarrollo: Alta
Puntos estimados:	Iteración asignada:
Programador responsable: Tony Tomas Nina Yanahuaya	
Descripción: Realiza el registro mediante una interfaz donde los usuarios puedan ingresar nombre de usuario y contraseña asignados para poder asignar los permisos según el rol.	
Observación: Cada usuario estará registrado en la base de datos con un usuario y contraseña	

Fuente: Elaboración Propia

3.3.2.2. Registro de menor:

Un registro ocurre cuando un menor viajara con la compañía de uno de los padres y el responsable o la secretaria es persona encargada es la que lleva a cabo el registro.

Tabla 3.4 Registro de datos del menor

Historia de Usuario	
Numero: 02	Usuario: Secretaria o Auxiliar

Nombre de historia: Registro del Menor	
Prioridad Alta	Riesgo de desarrollo: Media
Puntos estimados:	Iteración asignada:
Programador responsable: Tony Tomas Nina Yanahuaya	
Descripción: Realiza el registro de la menor y lo asigna una identidad que corresponde.	
Observación: Cada registro tiene la opción de agregar modificar y dar de baja.	

Fuente.: Elaboración Propia

3.3.2.3. Registro familiar, apoderado o responsable:

Una registrado datos del menor viaja se procede a registrar de los padres, apoderados y/o responsable.

Tabla 3.5 Registro familiar, apoderado o responsable

Historia de Usuario	
Numero: 03	Usuario: Secretaria o Auxiliar
Nombre de historia: Registro de los padres o Familiar, responsable o apoderado	
Prioridad Alta	Riesgo de desarrollo: Media
Puntos estimados:	Iteración asignada:
Programador responsable: Tony Tomas Nina Yanahuaya	
Descripción: Realiza el registro de la familia, responsable o apoderado y lo asigna un perfil que corresponde.	

Observación:

Cada registro tiene la opción de agregar, modificar y dar de baja.

Fuente: Elaboración Propia

3.3.2.4. Registro de la autorización de viaje:

El registro de autorización de lleva a cabo una vez registrado los datos del menor como también de los padres o familiares, responsable o apoderados y así completar los datos de la autorización de viaje.

Tabla 3.6 Registro de la autorización de viaje

Historia de Usuario	
Numero: 04	Usuario: Secretaria o Auxiliar
Nombre de historia: Registro de autorización de viaje	
Prioridad Alta	Riesgo de desarrollo: Media
Puntos estimados:	Iteración asignada:
Programador responsable: Tony Tomas Nina Yanahuaya	
Descripción: Realiza el registro de la autorización de viaje donde se carga los datos de la base de datos de los padres, responsable o apoderados.	
Observación: Cada registro tiene la opción de agregar y modificar.	

Fuente: Elaboracion Propia

3.3.2.5. Documentos adjuntos (anexos):

Un anexo, permite completar los datos que se generan cuando se emite una autorización por el sistema, que permite anexas información de los documentos del menor, de la familia y del responsable.

Tabla 3.7 Anexos

Historia de Usuario	
Numero: 05	Usuario: Administrador, secretaria
Nombre de historia: Anexos	
Prioridad Alta	Riesgo de desarrollo: Alta
Puntos estimados:	Iteración asignada:
Programador responsable: Tony Tomas Nina Yanahuaya	
Descripción: El sistema genera una formulario donde se debe completar los requisitos de la autorización de viaje: <ul style="list-style-type: none"> • Fotocopias de carnet de identidad • Fotocopia de certificado de nacimiento • Y otros documentos 	
Observación: No es necesario registrar en un día, el sistema puede acumular anexo.	

Fuente: Elaboración Propia

3.3.2.6. Reportes:

Un reporte es un documento generado por el sistema, que nos permite presentar de manera estructurada o resumida de una información que puede ser de forma gráfica o literal, esos datos relevantes generados por el mismo sistema.

Uno de los objetivos del sistema de información es proporcionar reportes de manera precisa y oportuna, que sirva de apoyo para el proceso de toma de decisiones, bajo esa propuesta el sistema proporciona una mejor representación de la información almacenada en la base de datos.

Tabla 3.8 Reportes

Historia de Usuario	
Numero: 06	Usuario: Administrador, director, jefa de la defensoría, secretaria y auxiliar.
Nombre de historia: Reportes	
Prioridad Alta	Riesgo de desarrollo: Alta
Puntos estimados:	Iteración asignada:
Programador responsable: Tony Tomas Nina Yanahuaya	
Descripción: El sistema debe ser capaz de proporcionar los siguientes reportes: Reporte de autorización de viajes	
Observación: Cada reporte de almacenara si es necesario.	

Fuente: Elaboración Propia

3.3.2.7. Copias de Seguridad(Backup):

Una copia de seguridad, también llamado “backup” es un copia de los datos originales que se realizan con el fin de disponer de un medio para recuperarlos en caso de su perdida.

Tabla 3.9 Copias de seguridad

Historia de Usuario	
Numero: 07	Usuario: Administrador, jefa de la defensoría
Nombre de historia: Copias de seguridad	
Prioridad Alta	Riesgo de desarrollo: Alta
Puntos estimados:	Iteración asignada:

Programador responsable: Tony Tomas Nina Yanahuaya
Descripción: El sistema muestra una interfaz donde se pueda realizar las copias de seguridad de la base de datos.
Observación:

Fuente: Elaboración Propia

3.3.3. Plan de entregas

Teniendo los requerimientos del sistema en las historias de usuario, estas fueron implementadas en cada entrega del sistema, además considerado el orden de prioridad se decidió dividir el desarrollo del sistema en cuatro iteraciones, considerando cuatro meses como máximo.

Tabla 3.10 Plan General de Entregas

Iteración	MES	1 DE MARZO HASTA EL 30 DE JUNIO															
		MARZO				ABRIL				MAYO				JUNIO			
	SEMANA	Semana				Semana				Semana				Semana			
	HISTORIA DE USUARIO	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	1. Gestión de usuarios	■															
	2. Registro de Personas		■														
2	3. Registro menor			■	■												
	4. Registro familia y responsable				■	■											
	5. Registro Autorización de Viajes					■	■	■									
3	6. Anexos								■	■							
	7. Reporte										■	■					
4	8. Estadística											■	■				
	9. Copias de Seguridad														■	■	
		Tiempo Estimado de 4 meses como máximo															

Fuente: Elaboración Propia

3.4. PRIMERA ITERACIÓN

Una vez concluido el plan general, tal como se describió en la (Tabla Plan general de entregas), se precedió a realizar las actividades comprendidas a la primera iteración.

3.4.1. Plan de la primera iteración

En el plan de entrega de la primera iteración se establecieron las tareas a realizar en cada historia de usuario.

Tabla 3.11 Plan primera iteración

Historia de usuario	Tarea a realizar	Programador	Tiempo Estimado
Gestión de usuarios	Formulario administración	Tony Nina Yanahuaya	4 hrs.
	Formulario usuario	Tony Nina Yanahuaya	4 hrs.
	Formulario perfil	Tony Nina Yanahuaya	4 hrs.
	Formulario de Logeo o bienvenida	Tony Nina Yanahuaya	8 hrs.
Registro de Persona	Formulario persona	Tony Nina Yanahuaya	4 hrs.
	Búsqueda y modificación	Tony Nina Yanahuaya	8 hrs.
Total horas			32 hrs.

Fuente: Elaboración Propia

3.4.2. Diseño del subsistema de la primera iteración

Una vez determinado el plan de la iteración, se procede a construir cada una de las tarjetas “Clase-Responsabilidad-Colaboración”

3.4.2.1. Clase-Responsabilidad-Colaboración

Los diagramas que se presentan a continuación son parte de la primera iteración:

- **Clase Persona.** Representa a registro de personas para luego ser parte del formulario Menor y del formulario Familia o Responsable.

Tabla 3.12 Tarjeta CRC de la clase persona

Clase: Persona	
Responsabilidad:	Colaboración:
Agregar persona Modificar persona Eliminar	

Fuente: Elaboración Propia

- **Clase Búsqueda y Modificación.** Representa la búsqueda de una persona registrado en el sistema.

Tabla 3.13 Tarjeta CRC de la clase búsqueda

Clase: Búsqueda	
Responsabilidad:	Colaboración:
Buscar por el número de identificación: <ul style="list-style-type: none"> • El carnet de identidad • O el certificado de nacimiento Agregar datos a la base de datos Modificar si existe en la base de datos	Persona

Fuente: Elaboración Propia

- **Clase Usuario.** Representa el registro de los usuarios.

Tabla 3.14 Tarjeta CRC de la clase Usuario

Clase: Usuario	
Responsabilidad:	Colaboración:
Agregar nombre de usuario	Persona
Agregar contraseña de usuario	Perfil
Modificar usuario	
Eliminar usuario	

Fuente: Elaboración Propia

- **Clase Perfil.** Representa la asignación de un perfil al usuario.

Tabla 3.15 Tarjeta CRC de la clase Perfil

Clase: Perfil	
Responsabilidad:	Colaboración:
Agregar perfil Actualizar perfil	

Fuente: Elaboración Propia

- **Clase Logeo.** Representa la bienvenida y acceso al sistema de acuerdo al perfil de usuario.

Tabla 3.16 Tarjeta CRC de la clase Logeo

Clase: Logeo	
Responsabilidad:	Colaboración:
Agregar usuario Agregar contraseña	Usuario Perfil

Fuente: Elaboración Propia

3.4.2.2. Diagrama de Clases de la primera iteración

Una vez obtenido las tarjetas CRC de la primera iteración, se procede a construir el diagrama de clases en base al análisis de las mismas.

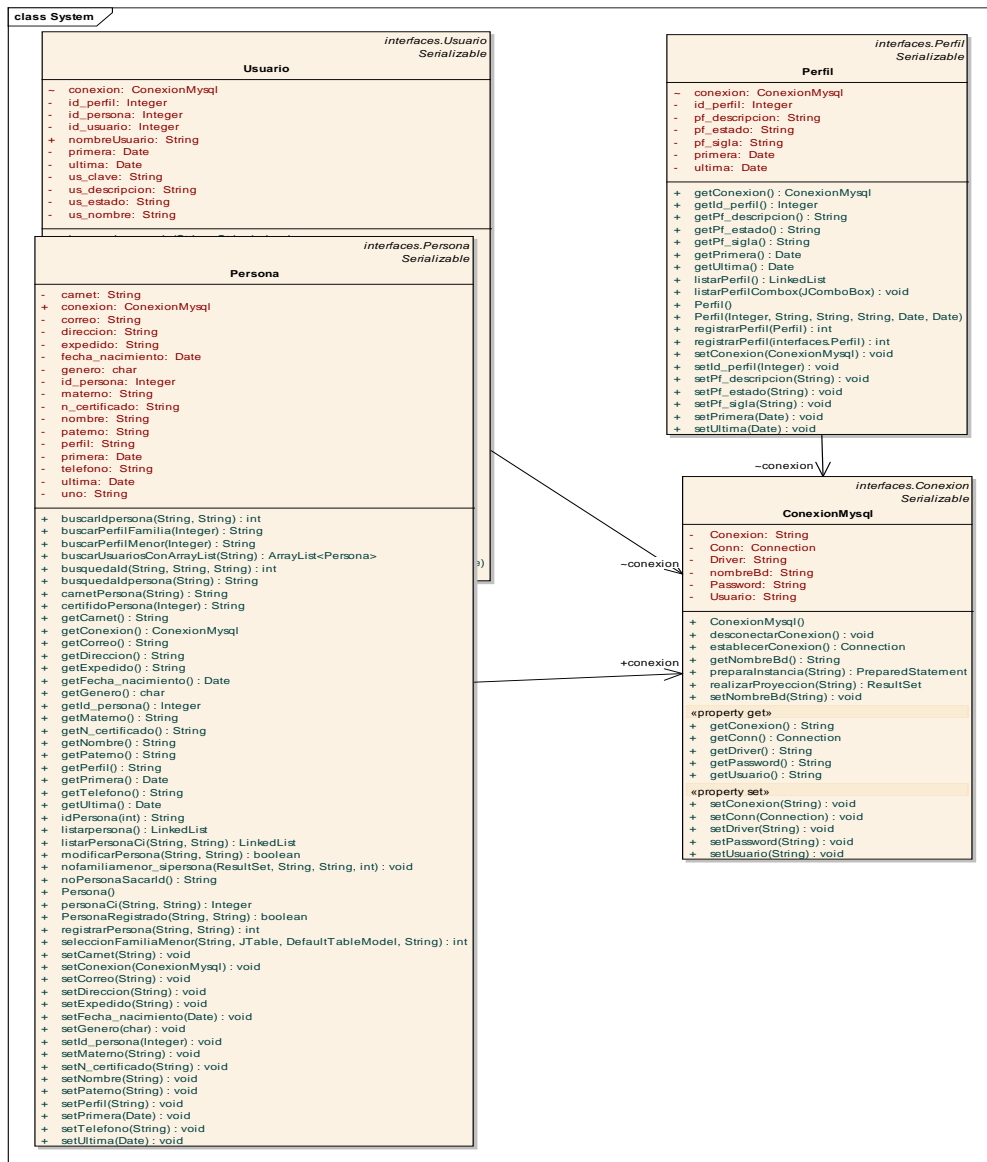


Ilustración 3.9 Figura diagrama de clases de la primera iteración
Fuente: Elaboración Propia

3.4.2.3. Diseño conceptual de la base de datos

La siguiente figura representa el diseño de la base de datos de la primera iteración.

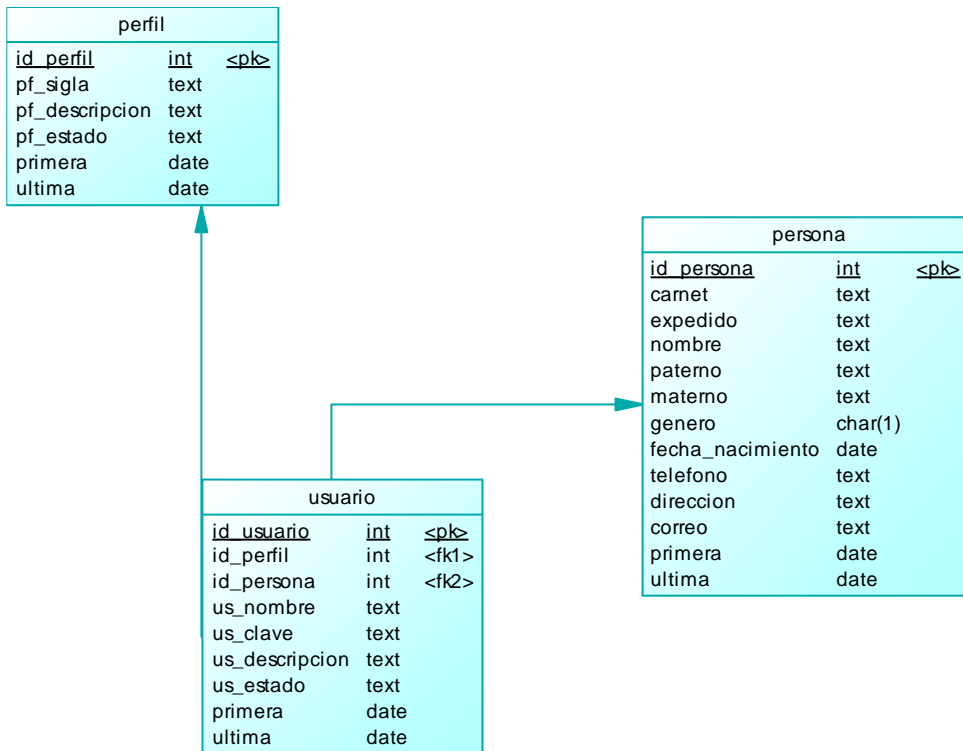


Ilustración 3.10 modelo de la base de datos de la primera iteración
Fuente: Elaboración Propia

3.4.2.4. Funcionalidad del sistema

Las partes fundamentales de sistema de describen a continuación:

- El formulario de búsqueda, permite la búsqueda de los datos personales de la persona en la base de datos, mediante el número de carnet de identidad o como también el número de certificado de nacimiento, si se encuentra en la base de datos muestra los datos que están registrado caso contrario solo muestra en número de identificación ingresado.

BUSCAR LA INFORMACION EN LA BASE DE DATOS

Por el Carnet de Identidad N°:

Caso contrario de no tener Carnet de identidad

Por el codigo del Certificado N°:

Buscar en la Base de Datos:

Ilustración 3.11 Formulario de búsqueda datos de Persona
Fuente: Elaboración Propia

- El formulario de registro, permite registrar los datos personales de la persona, pero si se encontraron los datos personales se pueden modificar.

* Nombre:

* Apellido Paterno:

Apellido Materno:

* Genero: Femenino Masculino

Carnet de Identidad:

Nro Certificado:

Telefono o Celular:

Dirección:

* Fecha de Nacimiento:

Ilustración 3.12 Formulario de Persona
Fuente Elaboración Propia

3.4.3. Desarrollo del subsistema de la primera iteración

Aplicando el diagrama de caso de uso de la primera iteración

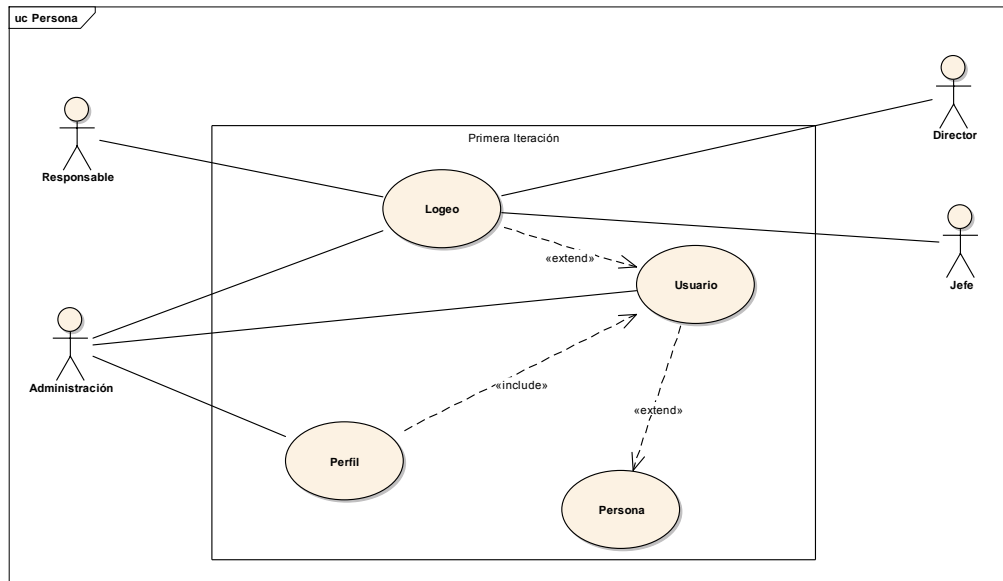


Ilustración 3.13 Diagrama de caso de uso
Fuente: Elaboración Propia

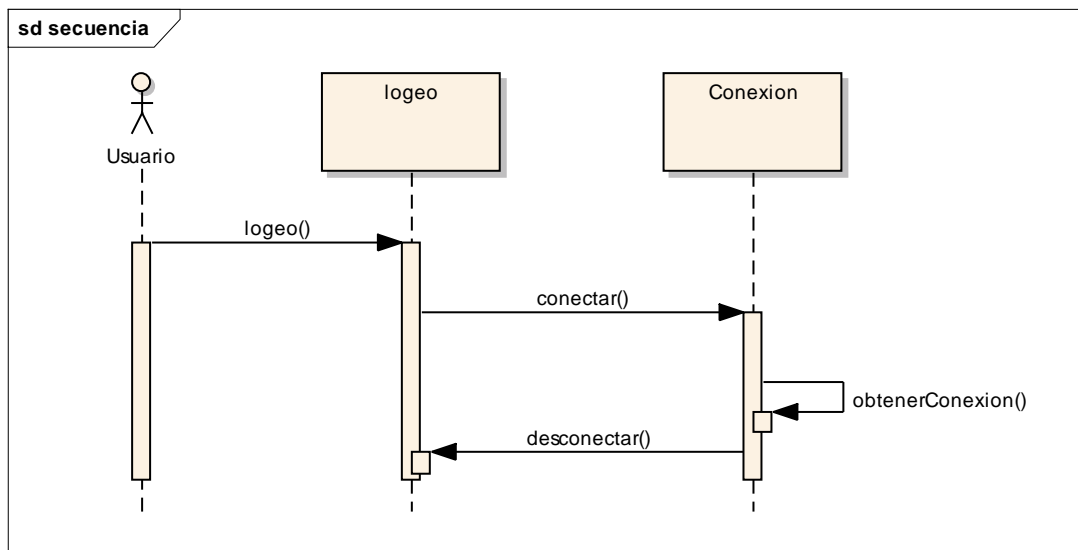


Ilustración 3.14 Diagrama de Secuencia
Fuente: Elaboración Propia

```

39 //
40 }
41 public ConexionMysql() throws IOException {
42     this.Driver = "com.mysql.jdbc.Driver";
43     //
44     this.ip = "192.168.3.2:3306";
45     this.nombreBd = "nna4";
46     this.Conexion = "jdbc:mysql://" + ip + "/" + this.nombreBd;
47     //
48     this.Usuario = "defensoria";
49     this.Usuario = "root";
50     this.Password = "defensoria";
51 }
52 //=====
53 @Override
54 public Connection establecerConexion() {
55     try {
56         Class.forName(Driver);
57         this.Conn = DriverManager.getConnection(this.Conexion, this.Usuario, this.Password);
58     } catch (ClassNotFoundException | SQLException e) {
59         //JOptionPane.showMessageDialog(null, "Error al conectar a mysql" + e.getMessage());
60     }
61     return this.Conn;
62 }
63
64 @Override
65 public void desconectarConexion() throws SQLException {
66     this.Conn.close();
67 }

```

Ilustración 3.15 Codificación en java
Fuente Elaboración Propia

```

16 * @author Kosmery
17 */
18 public class Perfil implements interfaces.Perfil, Serializable {
19
20     private Integer id_perfil;
21     private String pf_sigla;
22     private String pf_descripcion;
23     private String pf_estado;
24     private Date primera;
25     private Date ultima;
26     public boolean terminar;
27     ConexionMysql conexion;
28
29     public Perfil() throws IOException {
30         this.id_perfil = new Integer("0");
31         this.pf_sigla = new String();
32         this.pf_descripcion = new String();
33         this.pf_estado = new String();
34         this.primera = new Date(0, 0, 0);
35         this.ultima = new Date(0, 0, 0);
36         this.terminar = false;
37         this.conexion = new ConexionMysql();
38     }

```

Ilustración 3.16 Clase perfil
Fuente Elaboración Propia

3.4.4. Pruebas del sistema de la primera Iteración

3.4.4.1. Pruebas unitarias

Aplicando el *framework* JUnit se realiza las pruebas unitarias de forma automática, que se realiza al módulo “*Vbuscar.java*”. (Ver anexo D) Luego de ejecutar se genera un paquete de pruebas, dentro de la carpeta se encuentra el archivo “*VbuscarTest.java*”, después se procede a realizar la ejecución del archivo “*VbuscarTest.java*”, dando como resultado una vista de resultados de las pruebas y resolviendo cada uno de los errores se procede a ejecutar nuevamente el archivo “*VbuscarTest.java*” y muestra sin errores como lo muestra en la ilustración .15.

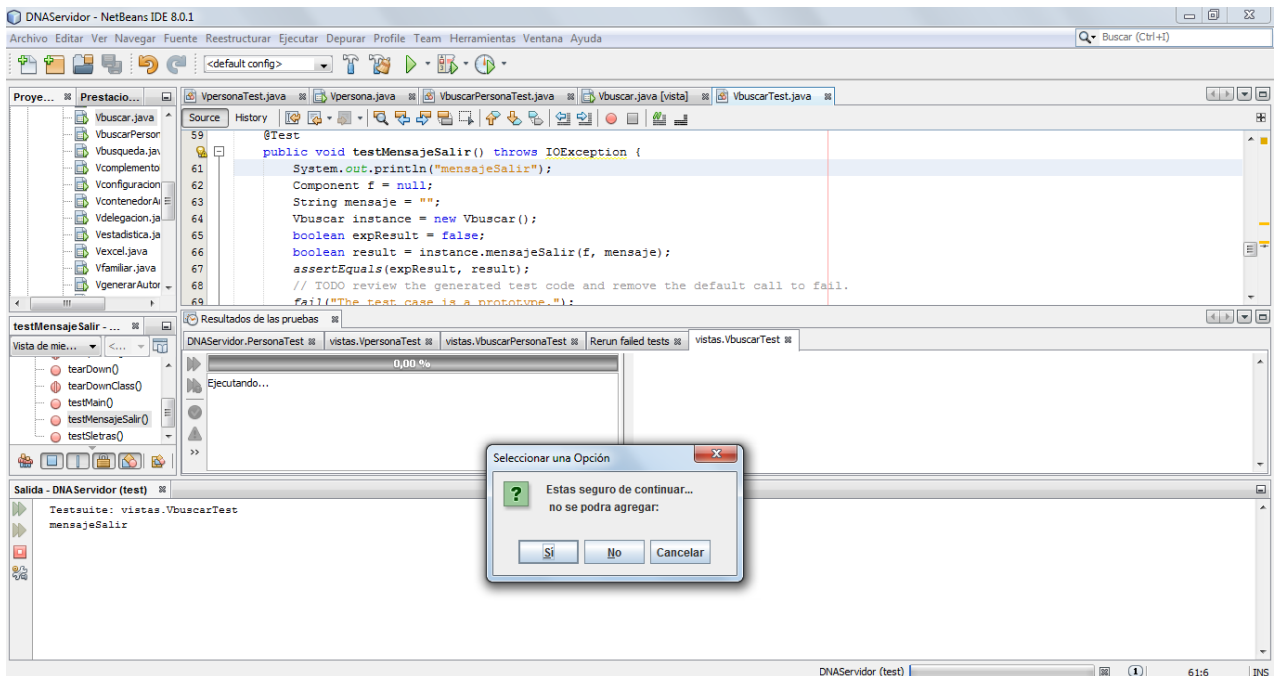


Ilustración 3.17 Pruebas de la primera iteración
Fuente: Elaboración Propia

3.4.4.2. Pruebas de aceptación

A la conclusión de la primera iteración se procedió a efectuar las pruebas a través de la técnica de la caja negra, para verificar las condiciones sobre las cuales trabajará el sistema.

CASO DE PRUEBA DE ACEPTACIÓN	
HISTORIA DE USUARIO	01 Gestión de Usuarios
DESCRIPCIÓN: Registrar los datos personales y efectuar un rol de usuario.	
CONDICIONES DE EJECUCIÓN	
ENTRADA/ PASOS DE EJECUCIÓN	
ENTRADA: Realiza el registro mediante una interfaz donde los usuarios puedan ingresar: <ul style="list-style-type: none"> • nombre de usuario • contraseña asignados para poder asignar los permisos según el rol.	
PASOS DE EJECUCIÓN: Datos Personales >> Persona	
RESULTADOS ESPERADOS: Cada usuario está asignado a un rol	
EVALUACIÓN DE PRUEBAS: Superado con éxito.	

Ilustración 3.18: Prueba de Aceptación de la Primera Iteración
Fuente: Elaboración Propia

3.5. SEGUNDA ITERACIÓN

3.5.1. Plan de la segunda iteración

Una vez concluido la primera iteración, se procede a realizar el plan de entrega de la segunda iteración donde se establecerá las tareas que se realizara en cada historia de usuario.

Tabla 3.17 Plan de la segunda iteración

Historia de usuario	Tarea a realizar	Programador	Tiempo Estimado
Registro Menor	Formulario menor	Tony Nina Yanahuaya	8 hrs.
Registro Familia – Responsable	Formulario Familia/Responsable	Tony Nina Yanahuaya	8 hrs.

Registro Autorización	Formulario de autorización de viaje	Tony Nina Yanahuaya	8 hrs.
			Total horas
			24 hrs.

Fuente: Elaboración Propia

3.5.2. Diseño del subsistema de la segunda iteración

Una vez determinado el plan de la iteración, se procede a construir cada una de las tarjetas “Clase-Responsabilidad-Colaboración”

3.5.2.1. Clase-Responsabilidad-Colaboración

Los diagramas que se presentan a continuación son parte de la segunda iteración:

- **Clase Menor.** Representa al registro de identidad del menor donde ya incluye el formulario persona.

Tabla 3.18 Tarjeta CRC de la clase menor

Clase: Menor	
Responsabilidad:	Colaboración:
Buscar por el carnet de identidad o nro. de certificado de nacimiento	Persona
Agregar menor	Identidad
Modificar menor	
Actualizar menor	

Fuente: Elaboración Propia

- **Clase Familia/Responsable.** Representa al registro de los familiares y/o responsable del menor donde ya incluye el formulario persona.

Tabla 3.19 Tarjeta CRC de la clase Familia/Responsable

Clase: Familia/Responsable	
Responsabilidad:	Colaboración:
Buscar por el carnet de identidad	Persona
Agregar familia o responsable	Perfil_Familiar
Modificar familia o responsable	

Actualizar familia o responsable	
----------------------------------	--

Fuente: Elaboración Propia

- **Clase Autorización.** Representa al registro la autorización de viaje y su impresión.

Tabla 3.20 Tarjeta CRC de la clase persona

Clase: Autorización	
Responsabilidad:	Colaboración:
Incluye los datos menor	Menor
Seleccionar familia o responsable	Familia
Agregar Autorización	
Guardar e Imprimir Autorización	

Fuente: Elaboración Propia

3.5.2.2. Diagrama de Clases de la segunda iteración

Una vez obtenido las tarjetas CRC de la segunda iteración, se procede a construir el diagrama de clases en base al análisis de las mismas.

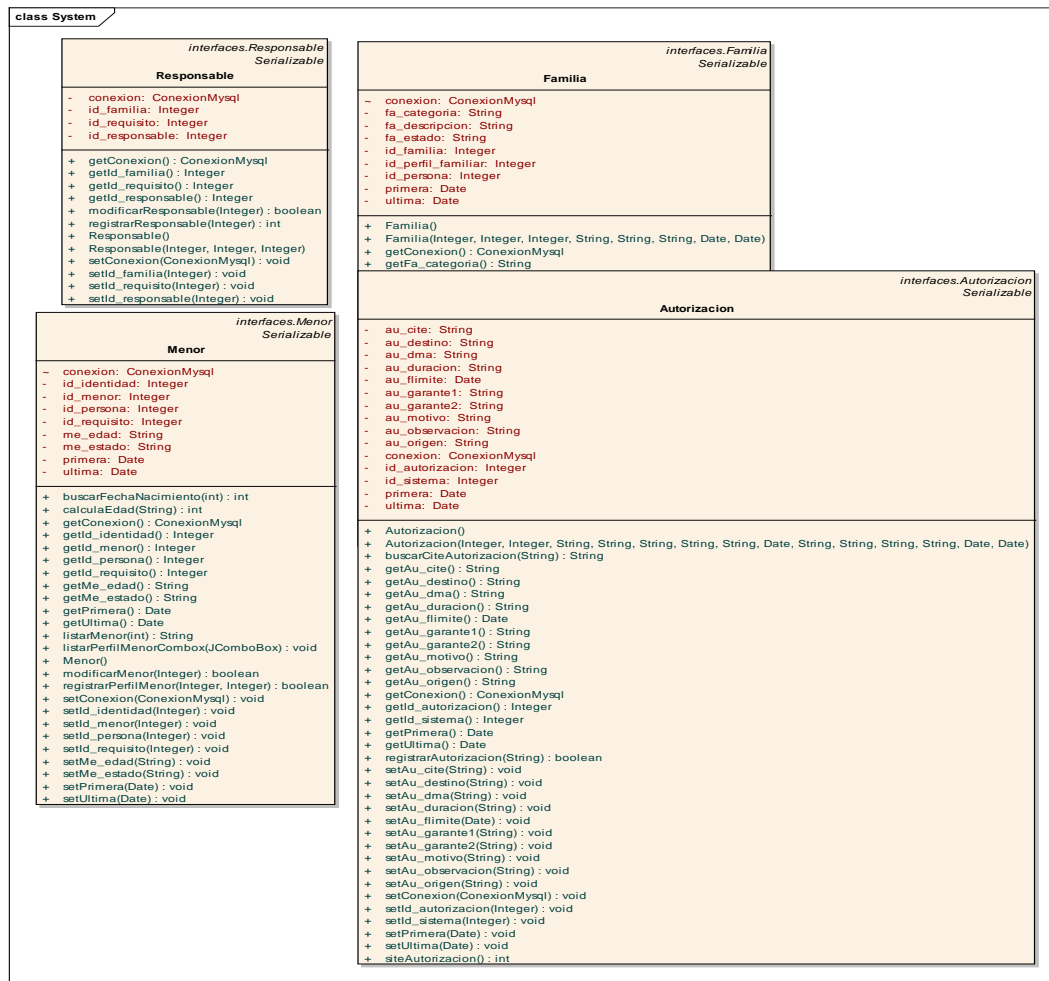


Ilustración 3.19 diagrama de clases de la segunda iteración
Fuente: Elaboración Propia

3.5.2.3. Diseño conceptual de la base de datos

La siguiente figura representa el diseño de la base de datos de la segunda iteración.

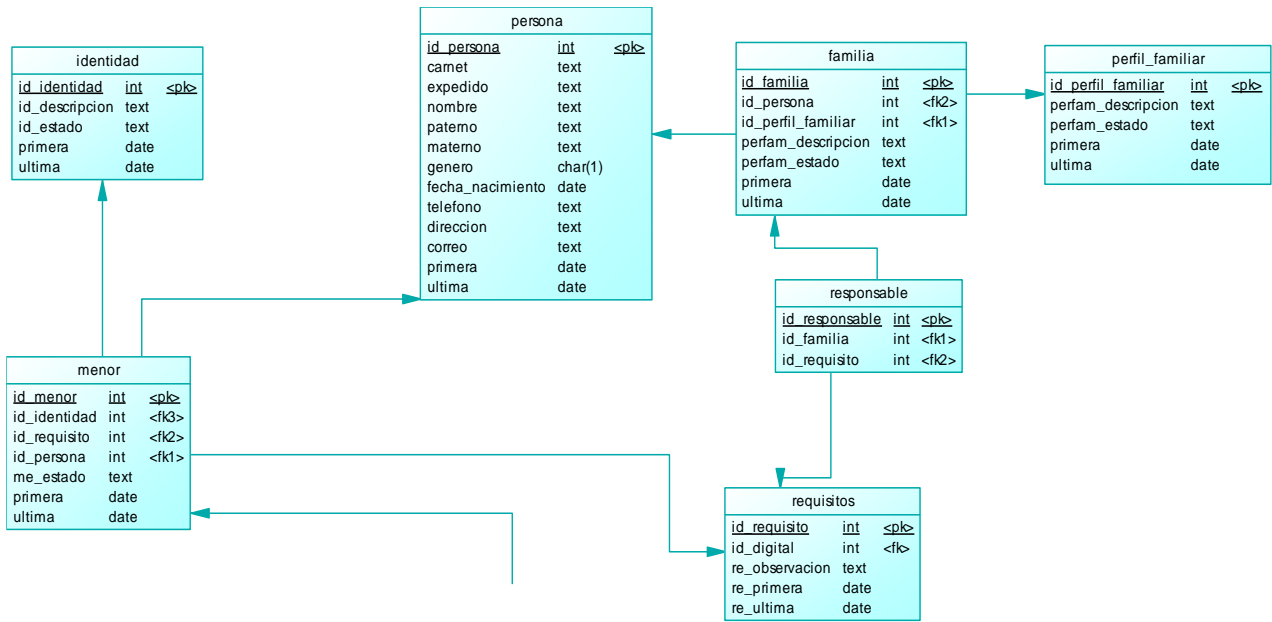


Ilustración 3.20 Modelo de la base de datos de la segunda iteración
Fuente: Elaboración Propia

3.5.2.4. Funcionalidad del sistema

Las partes fundamentales de sistema de describen a continuación:

- El formulario de registro de menor, según el formulario de búsqueda se procede a registrar los datos personales de la persona y su identidad, como también se puede modificar si existiera el menor.

Registro para la autorización

1. Paso: 2. Paso: 3. Paso:

FORMULARIO DE REGISTRO DEL MENOR

Datos del menor

* Nombre:

* Apellido Paterno:

Apellido Materno:

* Genero: Femenino Masculino

Carnet de Identidad: Expedido ▾

Nro Certificado:

Telefono o Celular:

Dirección:

* Fecha de Nacimiento:

Datos Complementarios

Seleccione la Identidad del menor:

▾

OBSERVACION:

Ilustración 3.21 Formulario de Menor
Fuente: Elaboración Propia

- El formulario de registro de la Familia/Responsable, según el formulario de búsqueda se procede a registrar los datos personales de la persona y el perfil al que corresponde de acuerdo a la responsabilidad hacia el menor.

Registro para la autorización

1. Paso: 2. Paso: 3. Paso:

FORMULARIO DE REGISTRO DE LOS FAMILIAR, APODERADO, TUTOR, GUARDADOR Y/O RESPONSABLES

Datos Personales

* Nombre:

* Apellido Paterno:

Apellido Materno:

* Genero: Femenino Masculino

Carnet de Identidad:

Nro Certificado:

Teléfono o Celular:

Dirección:

* Fecha de Nacimiento:

Datos Complementarios

Parentesco Familiar:

Responsable del Menor:

OBSERVACION: Escriba alguna Observación respecto al responsable o encargado del menor.....

Ilustración 3.22 Formulario de Familia/Responsable
Fuente Elaboración Propia

- El formulario de registro de la autorización de viajes, incluye los datos personales del menor según el registro efectuado en el formulario del menor y se procede a seleccionar la responsabilidad de o los familiar(es)/responsable(s) y completar las casillas bacías para completar la autorización de viajes.

Registro para la autorización

1. Paso: 2. Paso: 3. Paso:

REGISTRÓ DE AUTORIZACIÓN DE VIAJE

CITE N°: /2017

Campos obligados

* Se autoriza el viaje de: de años

Fecha de valides del viaje:

* Parte de la ciudad o lugar de: Con destino a:

* El motivo de: por el tiempo de: meses y día

Observacion:

Con la garantía de las siguientes Personas:

Padre/Madre/Familiar:

Responsable/Tutor/Guardador/apoderado/Testigo:

Autorizacion N°

Ilustración 3.23 Formulario de autorización de viajes
Fuente Elaboración Propia

- Una vez completado el formulario de autorización de viajes se procede a guardar, ahí es donde muestra un mensaje donde se seleccón la impresión o mostrar en formato “.pdf”, el formulario de autorización de viajes.

GOBIERNO AUTÓNOMO MUNICIPAL DE
Cobija
Jorge Limpio-Gestión Eficiente

SECRETARIA MUNICIPAL DE DESARROLLO HUMANO
DIRECCIÓN DE GENERO, GENERACIONAL Y FAMILIA
DEFENSORÍA DE LA NIÑEZ Y ADOLESCENCIA

AUTORIZACION DE VIAJE AL INTERIOR

Cobija, 08 septiembre 2017 D.N.N.A. CITE N° 3669/2017

Conforme a las disposiciones contenidas en los Arts. 2, 3, 4, 11 y 14 de la Convención Internacional sobre los derechos de los Niños y los Arts. 188, inc. ee, de la Ley 548 y Art. 58 núm. 1, 2, 3, 4; Art. 59 núm. 1, 2, 3, 4, 5, 6, 7; y art. 60 del Reglamento del Código Niña, Niño y Adolescente de la Ley 548; La Defensoría Municipal de la Niñez y Adolescencia, con el fin de evitar la trata, tráfico y el traslado de niños, niñas y adolescentes al interior del país sin la autorización de ambos progenitores:

POR LO TANTO: SE AUTORIZA EL VIAJE

NIÑO - ABIMAEI RICARDO FLORES QUISPE	1 año	Tiempo de Viaje
Nombre Completo	Su edad	7
Quien realizará el viaje de la ciudad COBIJA	SANTA CRUZ	día(s)
Origen	Destino	

Motivo del Viaje: **VACACIONES**

Observación: **VIAJA EN COMPAÑIA DE SU MADRE**

La presente autorización de viaje es emitida previa presentación de los requisitos exigidos por la Defensoría de la Niñez y Adolescencia del Municipio de Cobija.

LOURDES QUISPE MAMANI - MADRE - C.I.: 145500 Lp Padre/Madre/Familiar	LOURDES QUISPE MAMANI - MADRE - C.I.: 145500 Lp Responsable/Tutor/Guardador/Apoderado/Testigo
--	---

Ilustración 3.24 Formulario de Autorización de viajes
Fuente Elaboración Propia

3.5.3. Desarrollo del subsistema de la segunda iteración

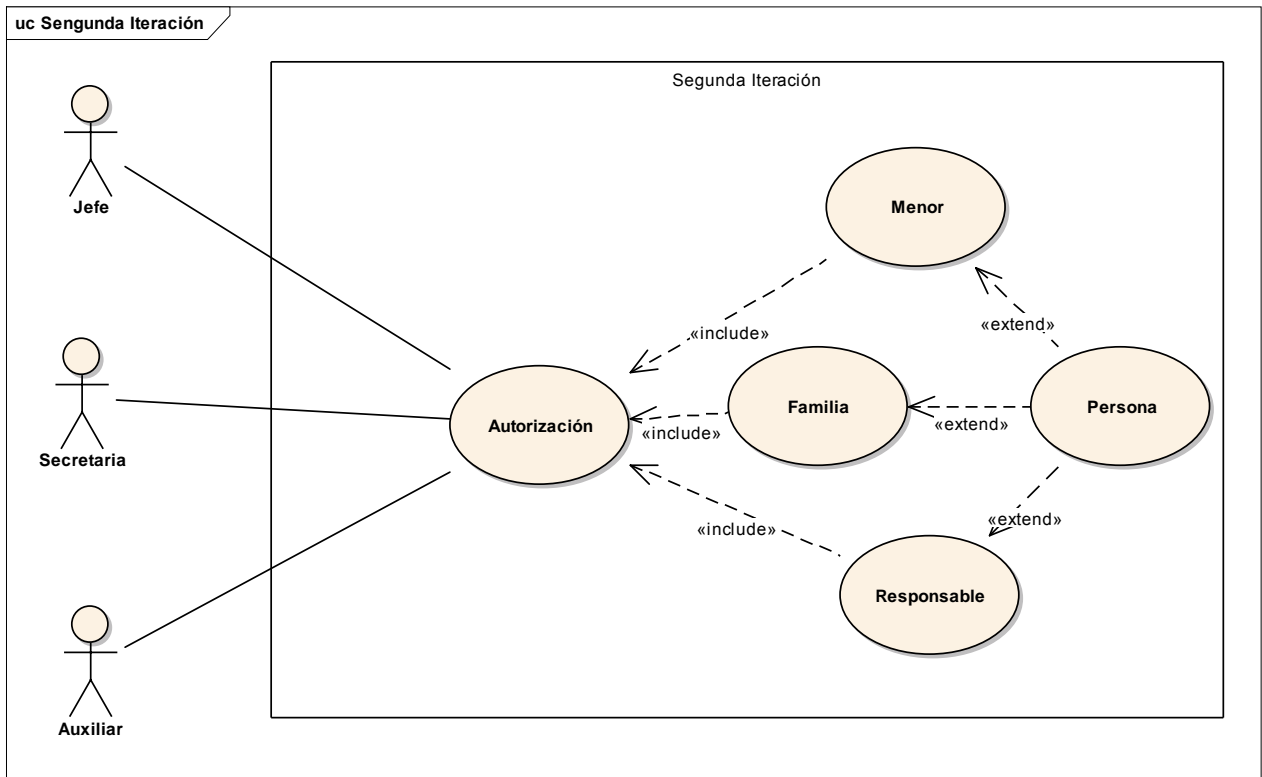


Ilustración 3.25 Diagrama de caso de uso
Fuente Elaboración Propia

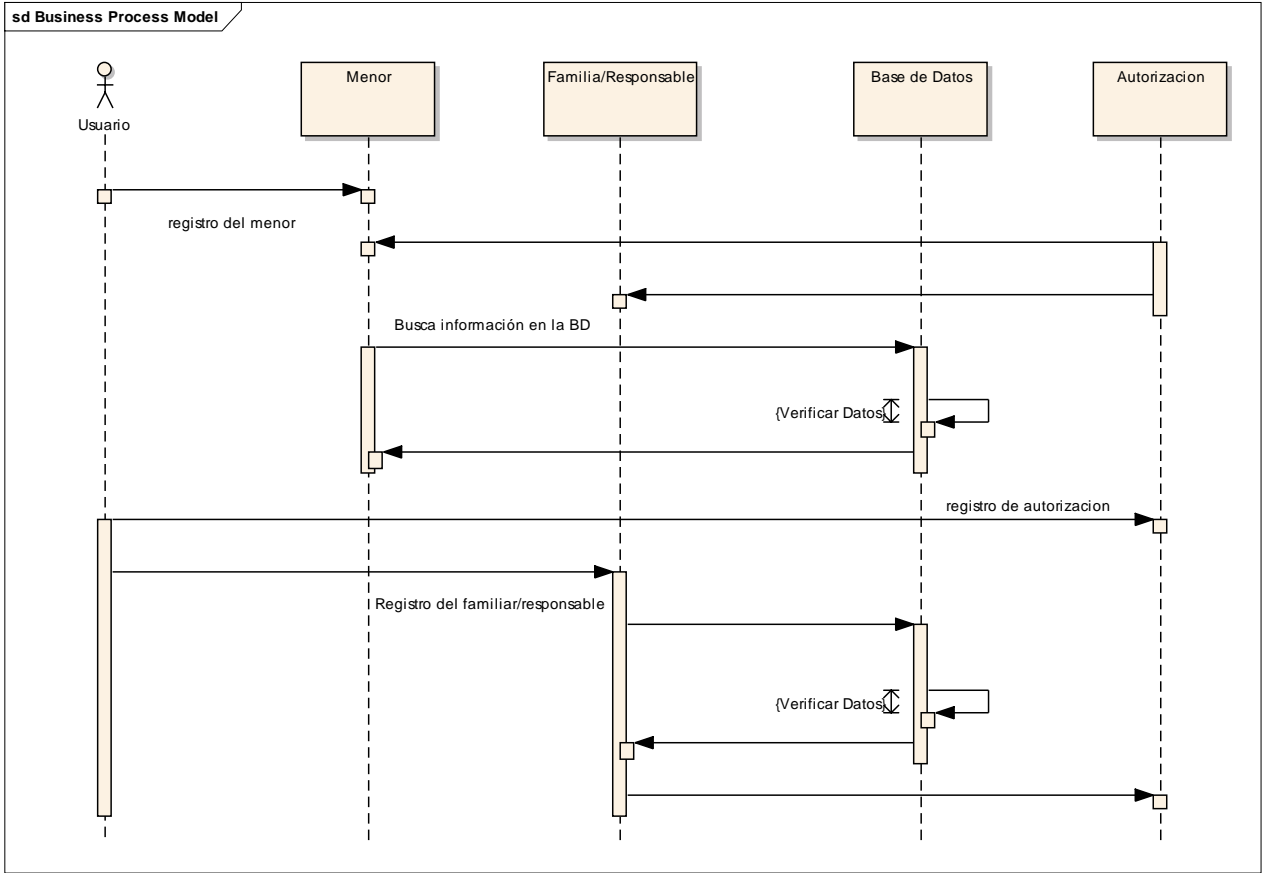
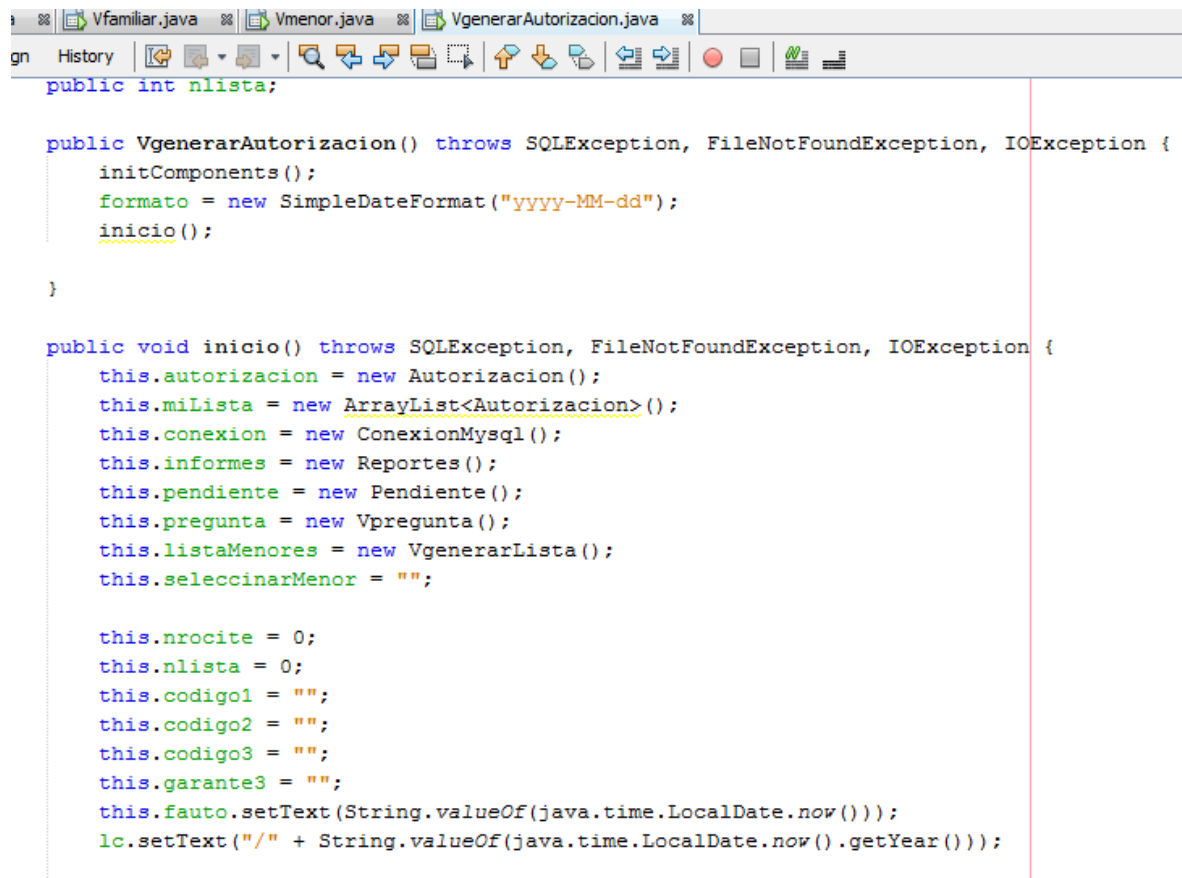


Ilustración 3.26 Diagrama de secuencia
Fuente Elaboración Propia



```
public int nlista;

public VgenerarAutorizacion() throws SQLException, FileNotFoundException, IOException {
    initComponents();
    formato = new SimpleDateFormat("yyyy-MM-dd");
    inicio();
}

public void inicio() throws SQLException, FileNotFoundException, IOException {
    this.autorizacion = new Autorizacion();
    this.miLista = new ArrayList<Autorizacion>();
    this.conexion = new ConexionMysql();
    this.informes = new Reportes();
    this.pendiente = new Pendiente();
    this.pregunta = new Vpregunta();
    this.listaMenores = new VgenerarLista();
    this.seleccinarMenor = "";

    this.nrocite = 0;
    this.nlista = 0;
    this.codigo1 = "";
    this.codigo2 = "";
    this.codigo3 = "";
    this.garante3 = "";
    this.fauto.setText(String.valueOf(java.time.LocalDate.now()));
    lc.setText("/") + String.valueOf(java.time.LocalDate.now().getYear());
}
```

Ilustración 3.27 Codificación del formulario autorización
Fuente Elaboración Propia

```
ava  Vfamiliar.java  Vmenor.java  VgenerarAutorizacion.java
esign  History  [Icons]
*/
public class Vfamiliar extends javax.swing.JFrame {

    public String mensaje;
    public Vbusqueda vbuscar;
    public Vrequisito documentos;
    public Requisito requisito;
    public Responsable responsable;
    public Familia pfamilia;
    public LinkedList<Responsable> guardarCodigoResponsable;
    public boolean activar;
    public int idpersona;
    public boolean cerrar;

    public Vfamiliar() throws SQLException, IOException {
        initComponents();
        iniciar();
        this.guardarCodigoResponsable = new LinkedList<>();

        this.pfamilia.listarPerfilFamiliaResponsableCombo(perfilResponsable, "R");
        this.pfamilia.listarPerfilFamiliaResponsableCombo(perfilFamilia, "F");
    }

    public void iniciar() throws SQLException, IOException {
        this.idpersona = 0;
        this.mensaje = "";
        this.vbuscar = new Vbusqueda();
        this.vbuscar.nDescripcion = "familia";
        this.perfilFamilia.setSelectedIndex(0);
        this.perfilResponsable.setSelectedIndex(0);
    }
}
```

Ilustración 3.28 Codificación del formulario Familia/Responsable
Fuente: Elaboración Propia

```

this.nuevo.setEnabled(false);
this.cerrar = true;
this.tdescripcion.setText("Escriba Alguna Observacion respecto al me
}

@SuppressWarnings("unchecked")
Generated Code
public int capturarRequisito(int idDigital) throws SQLException {
    requisito.setId_digital(idDigital);
    requisito.setRe_observacion(tdescripcion.getText());
    return requisito.registrarRequisito(idDigital);
}

private void guardarActionPerformed(java.awt.event.ActionEvent evt) {

    String nro = "";
    String car = "";
    String idAux = "";
    int idreq = 0;
    if (perfil.getSelectedIndex() > 0) {

        try {

            perfil.setSelectedItem(vbuscar.vpersona.persona.getPerfil());
            pfmenor.setId_identidad(perfil.getSelectedIndex());
            //if (contarIdpersona.size() > 1) {

```

Ilustración 3.29 Codificación del formulario Menor
Fuente: Elaboración Propia

3.5.4. Pruebas del sistema de la segunda Iteración

3.5.4.1. Pruebas Unitarias

Aplicando el *framework* JUnit se realiza las pruebas unitarias de forma automática, que se realiza al módulo “*Vmenor.java*”. (Ver anexo D) Luego de ejecutar se genera un paquete de pruebas, dentro de la carpeta se encuentra el archivo “*VmenorTest.java*”, después se procede a realizar la ejecución del archivo “*VmenorTest.java*”, dando como resultado una vista de resultados de las pruebas y resolviendo cada uno de los errores se procede a ejecutar nuevamente el archivo “*VmenorTest.java*” y muestra sin errores como lo muestra en la ilustración 3.28.

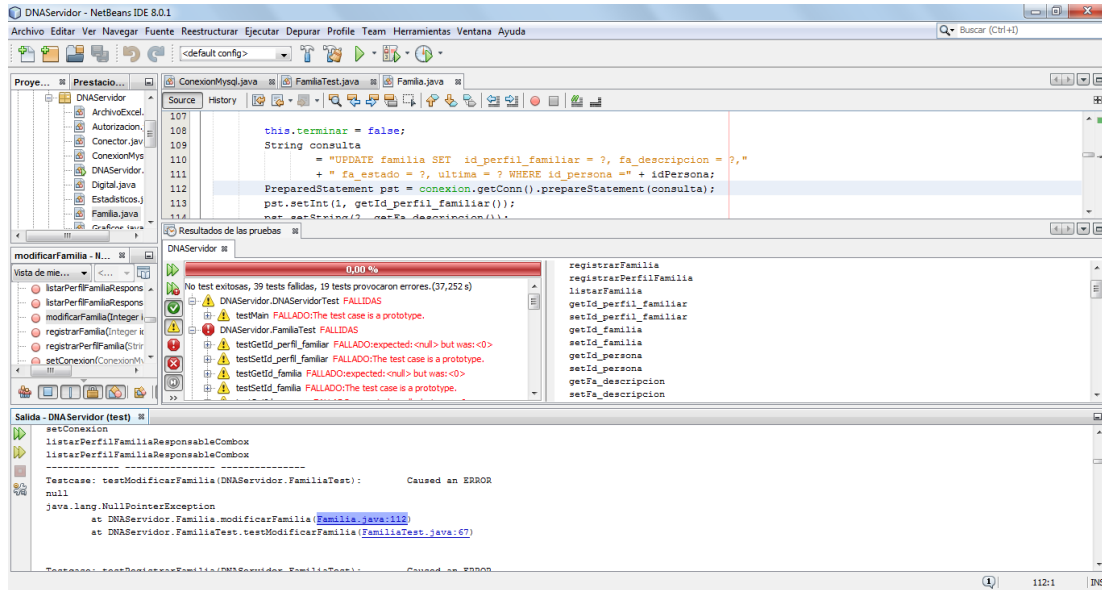


Ilustración 3.30 Pruebas segunda iteración
Fuente: Elaboración Propia

3.5.4.2. Pruebas de aceptación

A la conclusión de la primera iteración se procedió a efectuar las pruebas a través de la técnica de la caja negra, para verificar las condiciones sobre las cuales trabajará el sistema.

CASO DE PRUEBA DE ACEPTACIÓN	
HISTORIA DE USUARIO	02 Registro Menor
DESCRIPCIÓN: Verifica los datos registrados en el campo, sean numéricos y letras dependiendo de los datos que piden.	
CONDICIONES DE EJECUCIÓN	
ENTRADA/ PASOS DE EJECUCIÓN	
ENTRADA: Buscar por el número de identificación: <ul style="list-style-type: none"> • El número de carnet de identidad • El número de certificado de nacimiento.. 	

<p>Agregar datos a la base de datos.</p> <p>Modificar si existe en la base de datos.</p> <p>PASOS DE EJECUCIÓN:</p> <p>Introducir los datos según la casilla de texto donde ingresan letra no puede ingresar numéricos o viceversa.</p>
<p>RESULTADOS ESPERADOS:</p> <p>Que los campos de textos estén convalidados al ingresar los datos</p>
<p>EVALUACIÓN DE PRUEBAS:</p> <p>Superado con éxito.</p>

Ilustración 3.31 Prueba de aceptación Segunda Iteración
Fuente: Elaboración Propia

3.6. TERCERA ITERACIÓN

3.6.1. Plan de la tercera iteración

Un vez concluido el plan de la segunda iteración se procede a la tercera iteración se establecieron las tareas a realizar en cada historia de usuario.

Tabla 3.21 Plan de la tercera iteración

Historia de usuario	Tarea a realizar	Programador	Tiempo Estimado
Anexo	Formulario requisitos	Tony Nina Yanahuaya	8 hrs.
Reporte	Formulario reporte	Tony Nina Yanahuaya	8 hrs.
Total horas			16 hrs.

Fuente: Elaboración Propia

3.6.2. Diseño del subsistema de la tercera iteración

Una vez determinado el plan de la iteración, se procede a construir cada una de las tarjetas “Clase-Responsabilidad-Colaboración”

3.6.2.1. Clase-Responsabilidad-Colaboración

Los diagramas que se presentan a continuación son parte de la tercera iteración:

- **Clase Requisitos.** Representa al registro de copias escaneadas de documentos como el carnet de identidad del menor, de la familia y del responsable.

Tabla 3.22 Tarjeta CRC de la clase Requisitos

Clase: Requisitos	
Responsabilidad:	Colaboración:
Buscar por número de cite de la autorización Agregar copia escaneadas Modificar copia escaneadas	Autorización Digital Pendiente

Fuente: Elaboración Propia

- **Clase Reportes.** Representa al registro almacenado donde se muestran consultas y un resumen de la información que se desean para un informe.

Tabla 3.23 Tarjeta CRC de la clase Reporte

Clase: Reporte	
Responsabilidad:	Colaboración:
Buscar por número de identificación: <ul style="list-style-type: none"> • El carnet de identidad • O el certificado de nacimiento Agregar datos personales, Número de Cite y entre otros datos. Seleccionar tipo de consulta dependiendo de los datos introducidos.	Persona Menor Autorización

Fuente: Elaboración Propia

3.6.2.2. Diagrama de Clases de la tercera iteración

Una vez obtenido las tarjetas CRC de la tercera iteración, se procede a construir el diagrama de clases en base al análisis de las mismas.

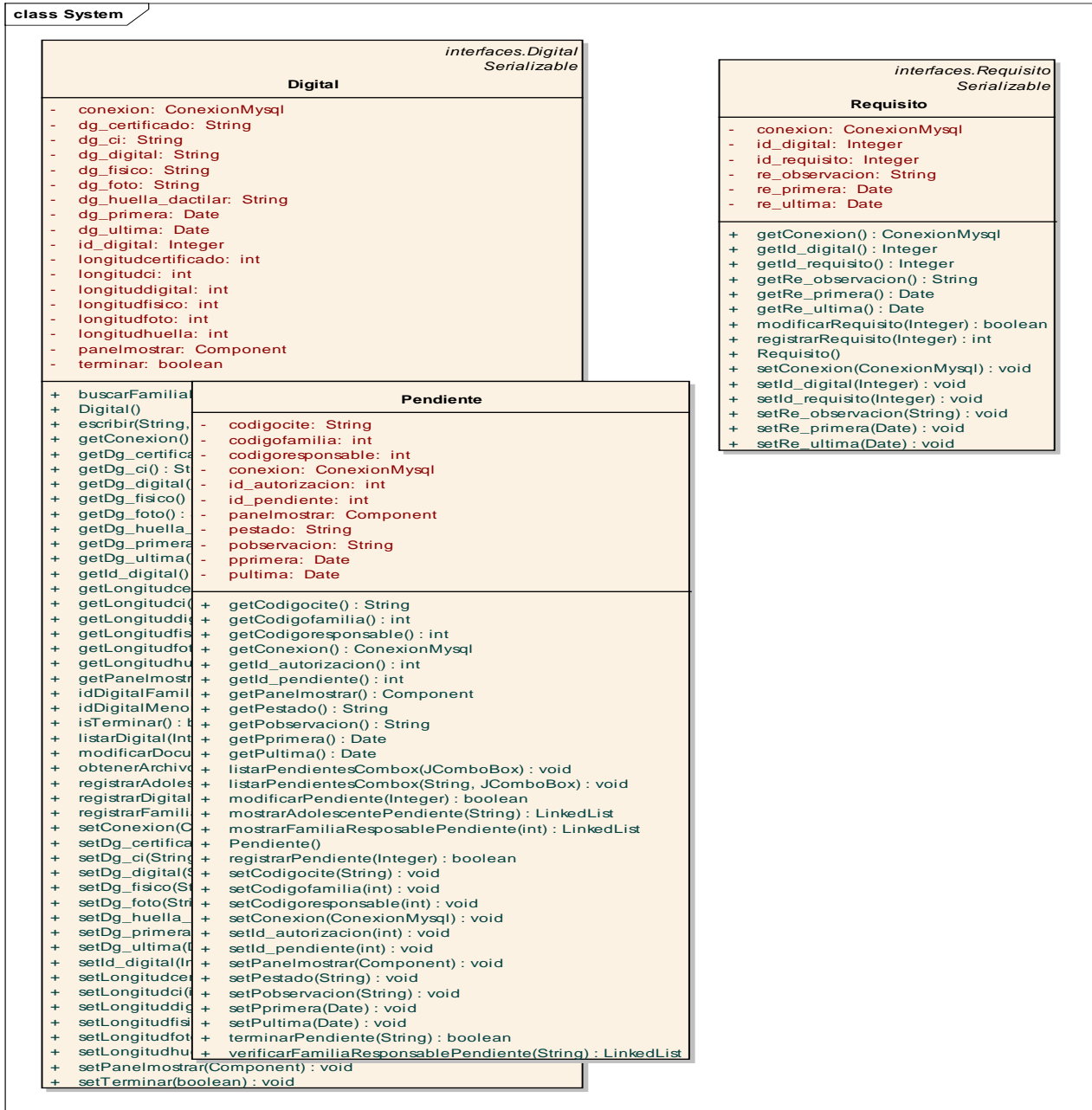


Ilustración 3.32 Diagrama de clases de la tercera iteración
Fuente: Elaboración Propia

3.6.2.3. Diseño conceptual de la base de datos

La siguiente figura representa el diseño de la base de datos de la tercera iteración.

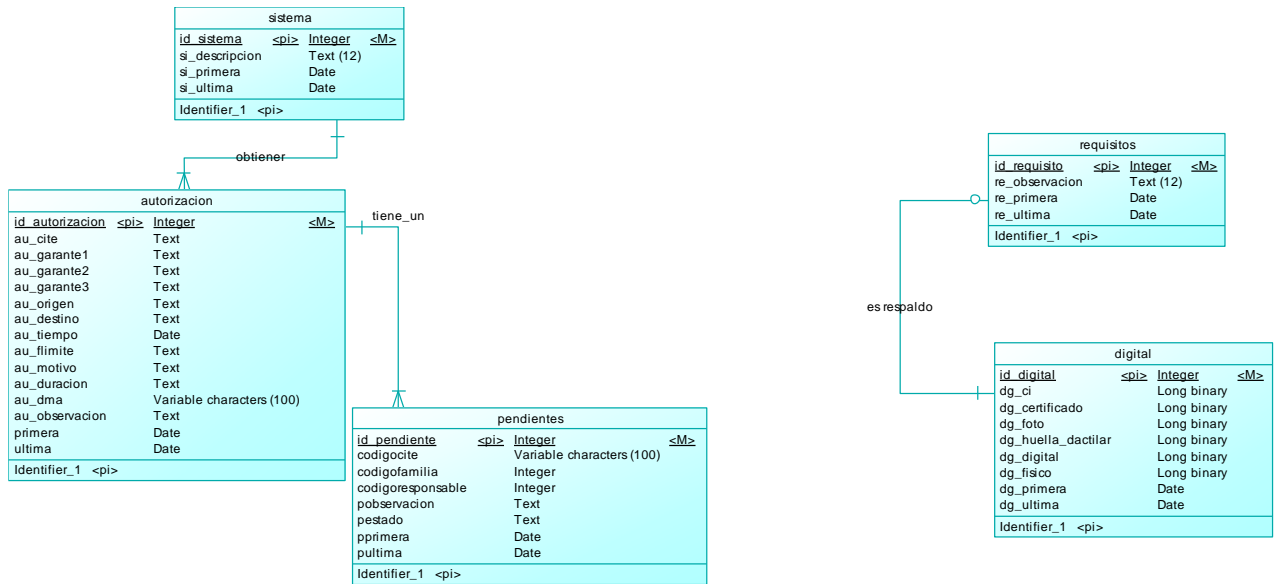


Ilustración 3.33 Modelo de la base de datos de la tercera iteración
Fuente: Elaboración Propia

3.6.2.4. Funcionalidad del sistema

Las partes fundamentales de sistema de describen a continuación:

- El formulario de requisitos, permite adicionar datos complementarios de la solicitud de autorización de viaje, permite agregar copias escaneadas de documentos del menor, familia y responsables.

REQUISITO INDISPENSABLE PERSONAL

Completar Información mediante el Nro. de:

Buscar por el:

Fecha de viaje:

Dirección del Archivo:

Para Agregar una Fotografía y/o Huella Dactilar
Selección la imagen haga clic en la imagen que corresponda

Datos del menor | Datos de la Familia | Datos del responsable

Datos personales

Nombres:

Ap. Paterna:

Ap. Materno:

Edad:

Completar Información mediante el escaneo del archivo:

Carnet de Identidad:

Certificado de Nacimiento:

Documentos Digital:

Documentos Físicos:

Ilustración 3.34 Formulario de Requisitos
Fuente: Elaboración Propia

- El formulario de reportes, permite ver un resume de la información almacenada en la base de datos de los menores, familiares, Responsables y según el tipo de datos que selecciona el usuario.

Características del reporte

Nro.:

Muestra todas las autorizaciones según:

Cite(s) Acompañante

Responsable y/o Familiar

Motivo Destino/Retorno

Dias permitido

Muestran todas las autorizaciones que realizó el menor:

Reporte anual de autorización de viajes

Ingresar:

Carnet de Identidad	Nombre	Apellido Paterno	Apellido Materno	Fecha de Nacimiento

Ilustración 3.35 Formulario de Reporte
Fuente: Elaboración Propia

3.6.3. Desarrollo del subsistema de la tercera iteración

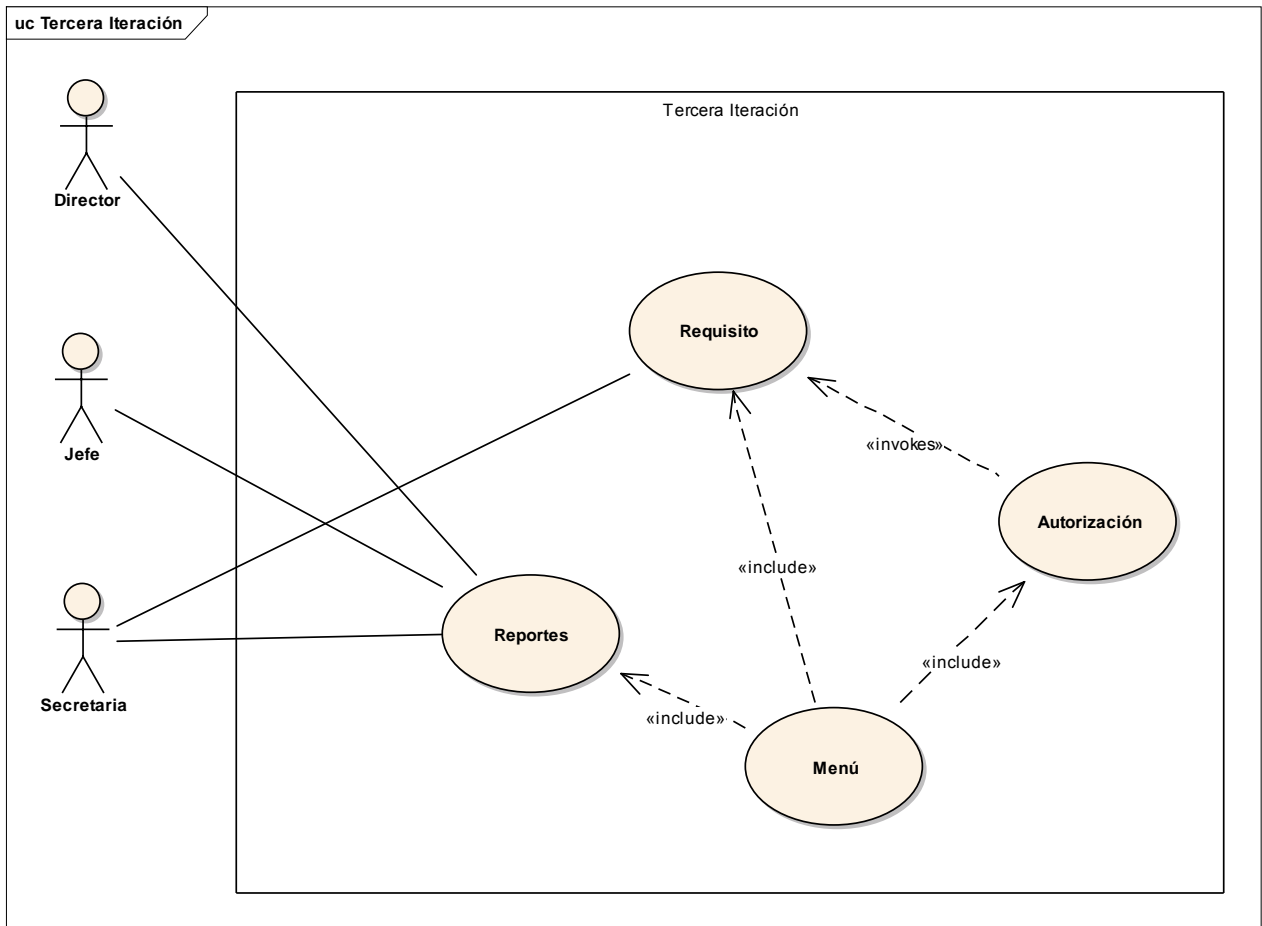


Ilustración 3.36 Diagrama de caso de uso
Fuente: Elaboración Propia

3.6.4. Pruebas de la tercera iteración

3.6.4.1. Pruebas Unitarias

Aplicando el *framework* JUnit se realiza las pruebas unitarias de forma automática, que se realiza al módulo “*Vrequisitos.java*”. (Ver anexo D) Luego de ejecutar se genera un paquete de pruebas, dentro de la carpeta se encuentra el archivo “*VrequisitosTest.java*”, después se procede a realizar la ejecución del archivo “*VrequisitosTest.java*”, dando como resultado una vista de resultados de las

pruebas y resolviendo cada uno de los errores se procede a ejecutar nuevamente el archivo “*VrequisitosTest.java*” y muestra sin errores como lo muestra en la ilustración 3.35.

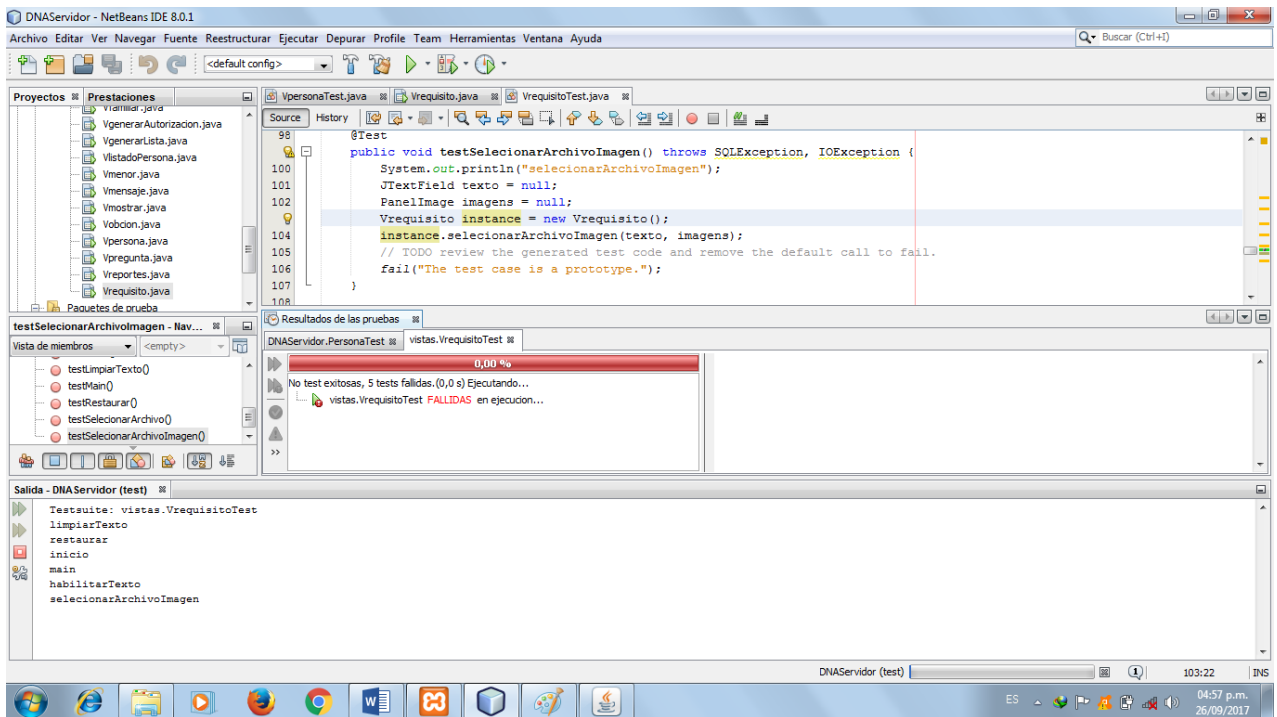


Ilustración 3.37 Prueba de la tercera iteración
Fuente: Elaboración propia

3.6.4.2. Pruebas de aceptación

A la conclusión de la primera iteración se procedió a efectuar las pruebas a través de la técnica de la caja negra, para verificar las condiciones sobre las cuales trabajará el sistema.

CASO DE PRUEBA DE ACEPTACIÓN	
HISTORIA DE USUARIO	06 Anexos
DESCRIPCIÓN: El sistema genera una formulario donde se debe completar los requisitos de la autorización de viaje:	
CONDICIONES DE EJECUCIÓN	

Una vez registrado la autorización de viajes se puede agregar los documentos escaneados
<p>ENTRADA/ PASOS DE EJECUCIÓN</p> <p>ENTRADA:</p> <p>Adjuntar documentos:</p> <ul style="list-style-type: none"> • Fotocopias de carnet de identidad • Fotocopia de certificado de nacimiento • Y otros documentos <p>Agregar datos a la base de datos.</p> <p>PASOS DE EJECUCIÓN:</p> <p>Registrar la autorización de viajes y luego se puede agregar los documentos escaneados</p>
<p>RESULTADOS ESPERADOS:</p> <p>Registrar la documentación, No es necesario registrar en un día, el sistema puede acumular anexo.</p>
<p>EVALUACIÓN DE PRUEBAS:</p> <p>Superado con éxito.</p>

Ilustración 3.38 Prueba de aceptación Tercera Iteración
Fuente: Elaboración Propia

3.7. CUARTA ITERACIÓN

3.7.1. Plan de la cuarta iteración

Una vez concluido con el plan de la tercera iteración se procederá con el último plan de entrega que corresponde a la cuarta iteración y se establecieron las tareas para cada historia de usuario.

Tabla 3.24 Plan de la cuarta iteración

Historia de usuario	Tarea a realizar	Programador	Tiempo Estimado
Estadísticos	Formulario estadísticos	Tony Nina Yanahuaya	8 hrs.
Copia de seguridad	Formulario copias de seguridad	Tony Nina Yanahuaya	8 hrs.

3.7.2. Diseño del subsistema de la cuarta iteración

Una vez determinado el plan de la iteración, se procede a construir cada una de las tarjetas “Clase-Responsabilidad-Colaboración”

3.7.2.1. Clase-Responsabilidad-Colaboración

Los diagramas que se presentan a continuación son parte de la cuarta iteración:

- **Clase Estadísticos.** Representa los registros almacenados en la base de datos y que pueden ser utilizados como referencia y reportes para cualquier tipo de informe.

Tabla 3.25 Tarjeta CRC de la clase Estadístico

Clase: Estadístico	
Responsabilidad:	Colaboración:
Adicionar datos para obtener gráficos estadísticos	Autorización SeleccionDatos Graficos JFreeChart

Fuente: Elaboración Propia

- **Clase Copia de seguridad.** Representa a la copia de seguridad de la base de datos.

Tabla 3.26 Tarjeta CRC de la clase Copia de seguridad

Clase: Copia de seguridad	
Responsabilidad:	Colaboración:
Crear la copia de seguridad Restaurar copia de seguridad	

Fuente:Elaboración Propia

3.7.2.2. Diagrama de Clases de la cuarta iteración

Una vez obtenido las tarjetas CRC de la cuarta iteración, se procede a construir el diagrama de clases en base al análisis de las mismas.

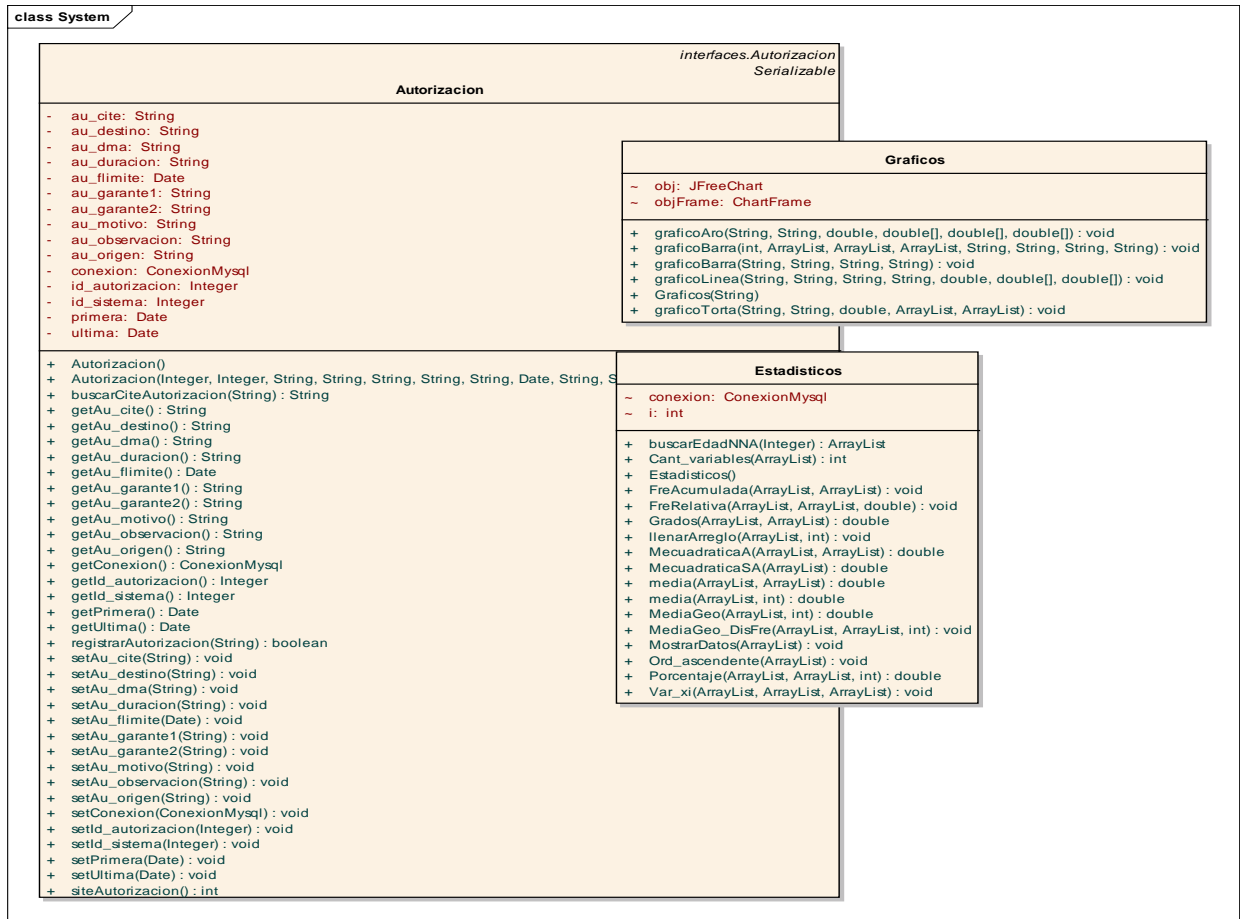


Ilustración 3.39 Diagrama de clases de la cuarta iteración
Fuente: Elaboración Propia

3.7.2.3. Diseño conceptual de la base de datos

La siguiente figura representa el diseño de la base de datos de la cuarta iteración.

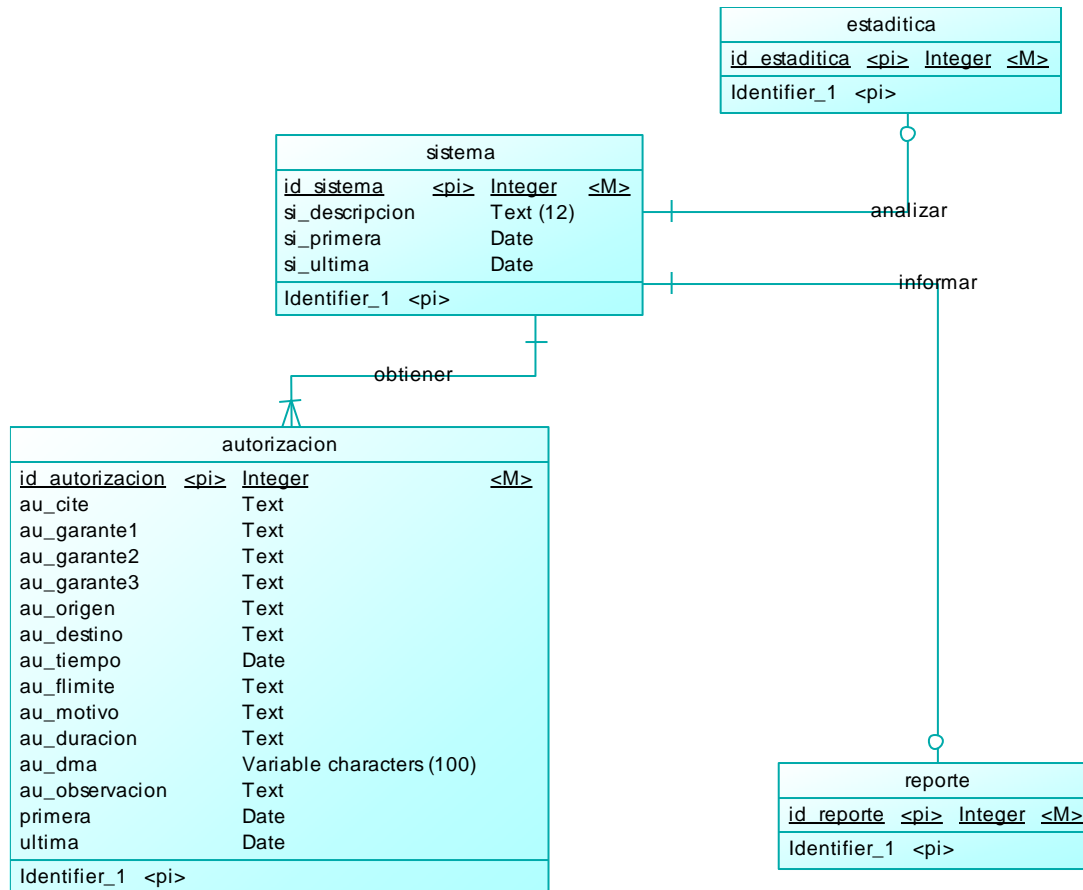


Ilustración 3.40 Modelo de la base de datos de la cuarta iteración
Fuente: Elaboración Propia

3.7.2.4. Funcionalidad del sistema

Las partes fundamentales de sistema de describen a continuación:

- El formulario de datos estadísticos, permite seleccionar el tipo de grafico que se mostrara para la toma de decisiones según las características de los datos introducidos.

CONSULTAR MEDIANTE LA BASE DE DATOS

Graficar reportes

Rango de edades: a: En los meses de:

Datos personales de los familiares:

Datos personales de los responsables:

Graficar reportes

Graficar las edades de los menor por la medio de la:

Reporte de solicitudes realizado en:

Rango de viajes realizado en fechas: a:

Ilustración 3.41 Formulario de Estadísticos
Fuente: Elaboración Propia

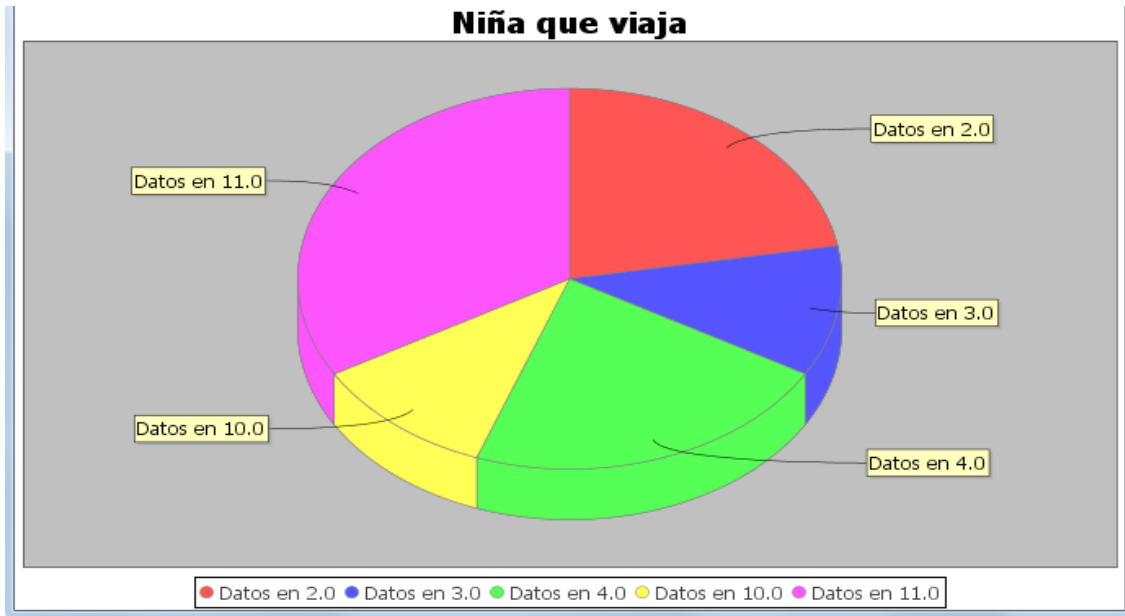


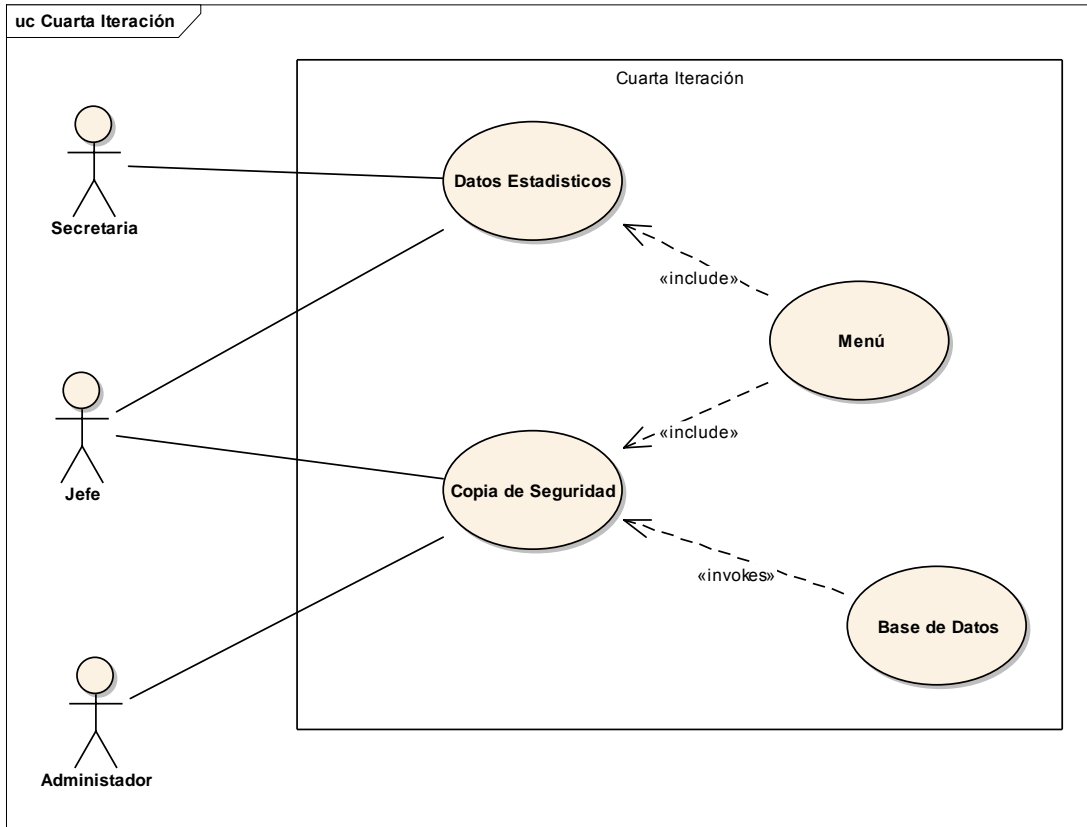
Ilustración 3.42 Formulario de Gráficos
Fuente: Elaboración Propia

- El formulario de copia de seguridad, permite conservar una copia de seguridad de la base de datos.

The screenshot shows a web interface with a blue background and white text. The title is "IMPORTAR Y EXPORTAR BASE DE DATOS DEL SISTEMA". There are two main sections. The first section is titled "Restaurar un respaldo de una base de datos:" and contains a "Seleccionar Ruta" button, a text input field, and an "Importar" button. The second section is titled "Realizar backup de la base de datos:" and contains a "Seleccionar Ruta" button, a text input field, and an "Exportar" button. At the bottom right, there is a "Salir" button.

Ilustración 3.43 Formulario de Copia de seguridad
Fuente: Elaboración Propia

3.7.3. Desarrollo del subsistema de la cuarta iteración



3.7.4. Pruebas de la cuarta iteración

3.7.4.1. Pruebas Unitarias

Aplicando el *framework* JUnit se realiza las pruebas unitarias de forma automática, que se realiza al módulo “*CopiaSeguridad.java*”. (Ver anexo D) Luego de ejecutar se genera un paquete de pruebas, dentro de la carpeta se encuentra el archivo “*CopiaSeguridadTest.java*”, después se procede a realizar la ejecución del archivo “*CopiaSeguridadTest.java*”, dando como resultado una vista de resultados de las pruebas y resolviendo cada uno de los errores se procede a ejecutar nuevamente el archivo “*CopiaSeguridadTest.java*” y muestra sin errores como lo muestra en la ilustración 3.42.

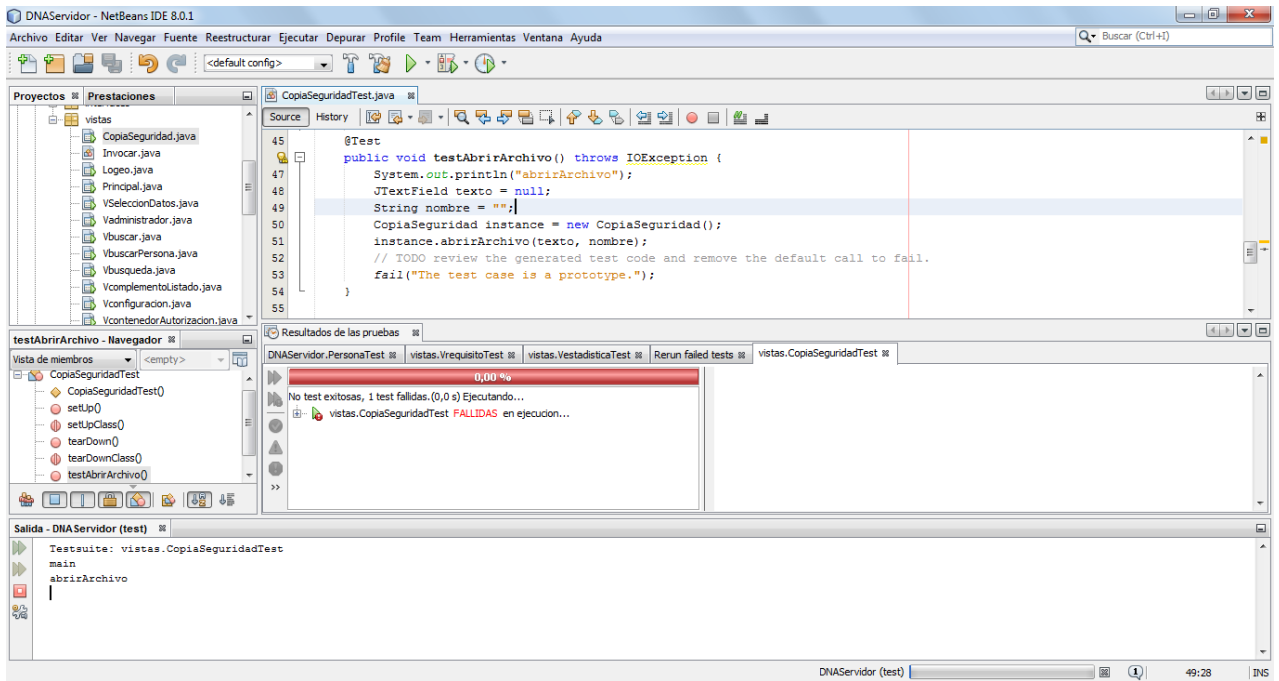


Ilustración 3.44 Prueba de la Cuarta Iteración
Fuente: Elaboración Propia

3.7.4.2. Pruebas de aceptación

A la conclusión de la primera iteración se procedió a efectuar las pruebas a través de la técnica de la caja negra, para verificar las condiciones sobre las cuales trabajará el sistema.

CASO DE PRUEBA DE ACEPTACIÓN	
HISTORIA DE USUARIO	07 Copias de Seguridad
DESCRIPCIÓN: El sistema muestra una interfaz donde se pueda realizar las copias de seguridad de la base de datos.	
CONDICIONES DE EJECUCIÓN Un medio importante para evitar la pérdida de información, es necesario realizar copias de seguridad.	

<p>ENTRADA/ PASOS DE EJECUCIÓN</p> <p>ENTRADA:</p> <p>Adjuntar documentos:</p> <ul style="list-style-type: none"> • Direccion del archivo la base de datos • Buscar el archivo con extension “.SQL" o ".sql”. <p>PASOS DE EJECUCIÓN:</p> <p>Importar el archivo >> Exportar el archivo</p>
<p>RESULTADOS ESPERADOS:</p> <p>Resguardar la copia de seguridad de la base de datos.</p>
<p>EVALUACIÓN DE PRUEBAS:</p> <p>Superado con éxito.</p>

Ilustración 3.45 Prueba de aceptación Cuarta Iteración
Fuente: Elaboración Propia

Una vez concluida con el trabajo de las iteraciones según la metodología XP, se procede aplicar la configuración de cliente/servidor y las pruebas.

3.8. ARQUITECTURA BASE DE DATOS CLIENTE/SERVIDOR

En esta parte de la arquitectura, la capacidad de proceso de información está repartida entre los clientes y el servidor, aunque con mayor importancia son las ventajas de tipo organizativo, debidas a la centralización de la gestión de la información y la separación de la responsabilidad, lo que facilita y clarifica el diseño del sistema.

A continuación se describe el procedimiento para que sistema que trabaje como cliente/servidor:

- Se tiene que codificar o modificar los parámetros de la clase **ConexionMysql.java**

SERVIDOR	CLIENTE >= 1
<pre>public ConexionMysql() throws IOException { this.Driver = "com.mysql.jdbc.Driver"; this.ip = "localhost:3306"; this.nombreBd = "nna4"; this.Conexion = "jdbc:mysql://" + ip + "/" + this.nombreBd; this.Usuario = "root"; this.Password = "*****"; }</pre>	<pre>public ConexionMysql() throws IOException { this.Driver = "com.mysql.jdbc.Driver"; this.ip = "192.168.3.2:3306"; this.nombreBd = "nna4"; this.Conexion = "jdbc:mysql://" + ip + "/" + this.nombreBd; this.Usuario = "defensoria"; this.Password = "*****"; }</pre>

Ilustración 3.46 Código fuente java
Fuente: Elaboración Propia

- Se tiene que dar privilegio en el servidor de XAMPP.

Usuarios con acceso a "nna4"

Usuario	Servidor	Tipo	Privilegios	Conceder	Acción
defensoria	%	global	ALL PRIVILEGES	Sí	
root	127.0.0.1	global	ALL PRIVILEGES	Sí	
root	::1	global	ALL PRIVILEGES	Sí	Editar los privilegios
root	localhost	global	ALL PRIVILEGES	Sí	Editar los privilegios

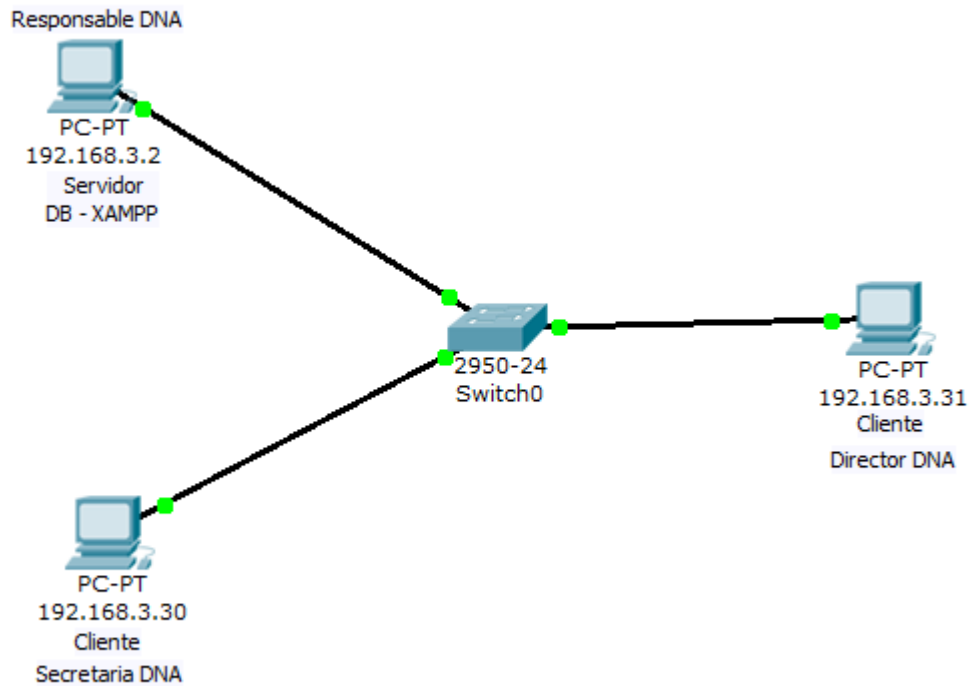
Cliente

Servidor

Ilustración 3.47 Privilegio en la base de datos
Fuente: Elaboración Propia

- Aplicando la arquitectura cliente/servidor al diseño de la red de la Defensoría de la Niñez y Adolescencia.

DEFENSORÍA DE LA NIÑEZ Y ADOLESCENCIA DEL MUNICIPIO DE COBIJA



Grupo de trabajo DEFENSORIA

Ilustración 3.48 Topologías de red de la institución
Fuente: Elaboración Propia

3.9. DOCUMENTO DE SOPORTE DE LA ETAPA DE PRUEBAS DEL REQUERIMIENTO DE LA METODOLOGÍA

Se consideró como etapas de utilizar las plantillas como herramientas de apoyo y soporte a las actividades sugeridas de la metodología.

3.9.1. Identificación de Necesidades

Presenta una descripción de lo que se quiere desde la perspectiva del cliente para resolver la necesidad u oportunidad de mejora identificada y la descripción del cómo se maneja actualmente el

proceso o actividad que va ser apoyada con el ajuste o creacion de la funcionalidad del sistema (**ver en el capitulo 2**).

3.9.2. Entrevistas

En esta parte se toma en consideración los siguientes puntos:

- la funcionalidad
 - Usuarios Finales
 - Nombre de la funcionalidad
- las características de los campos
 - nombre de campo
 - restricciones
 - validaciones
- dependencia de otros sistemas
 - información que se debe enviar a otros sistemas
 - información que se debe traer de otros sistemas

3.9.3. Matrices de trazabilidad

Matriz de Valoración y aprobacion de Requisitos			
	Cumple	No cumple	No aplica
El requerimiento se encuentra debidamente documentado y cumple con los estandares definidos	*		
El requisito no tiene errores de sintaxis y morfologicos		*	
El requisito cuenta con caso de prueba			*

Ilustración 3.49 Matriz de Valoración
Fuente: Metodología Gestión de Requisitos

3.9.4. Especificación Funcional

En esta parte de describe detalladamente cada uno de los módulos o funcionalidades con los que debe cumplir el requerimiento.

Módulo Registro de datos del menor	
Entrada	Carnet de identidad o nro. de certificado de nacimiento
Proceso	Registro en la base de datos
Salida	Registro correctamente

Ilustración 3.50 Especificación Funcional
Fuente: Metodología Gestión de Requisitos

Cada uno de las funcionalidad se detalla en cada ITERACIÓN(ver en **Tabla 3.27 Plan General de Entregas**) para poder comprender la funcionalidad de sistema.

3.9.5. Especificacion de Caso de Uso

Se considera en esta parte la descripción de los requerimientos identificados

Nombre		
Identificador de Casos de Uso	Complejidad	Prioridad
Requerimiento funcional	Obtener los requerimienton en base al cliente	Alta
Actores	Obtener todos lo actores	Alta
Caso de uso asociados	Diagramas de caso de uso	Baja
Propósito		Baja
Precondiciones		Alta

Ilustración 3.51Especificación de Caso de Uso
Fuente: Metodología Gestión de Requisitos

- Datos de entrada

Datos de Entrada						
Nombre del campo	Oblicación	Tipo	Mascara	Longitud	Restricción	Descripción
Nro. carnet de identidad		numerico		25	Numerico	Validacion de datos
Nombre completo		letras		35	Letras	Validacion de datos
Fecha de Nacimiento		date		25	Fecha	Validacion de fecha
Selección de identidad		char		15	Carácter	Validado

Ilustración 3.52 Datos de Entrada
Fuente: Metodología Gestión de Requisitos

- Datos de salida

Datos de Salida						
Nombre del campo	Oblicación	Tipo	Mascara	Longitud	Restricción	Descripción
Guardar		Boton	Icono	25		Antes se tiene que registra los datos
Eliminar		Boton	Icono	25		Se selecciona el datos
Modificar		Boton	Icono	25		Se selecciona el datos

Ilustración 3.53 Datos de Salida
Fuente: Metodología Gestión de Requisitos

3.9.6. Especificacion de Diagramas

En esta etapa se toma en consideración los diagramas empleados en el desarrollo del sistema mediante las etapas de iteración (**Ver Ilustración 3.54 Diagrama de caso de uso**).

3.9.7. Planeación Prueba de Requerimientos

En esta etapa se considera los alcances del proyecto que se plateo de manera que se realizará la actividades durante la ejecucción para obtener una finalización exitosa.

3.9.8. Diseño de Casos de prueba

Se representa con la siguiente tabla

Nombre del Requerimiento					
	Numeral	Usuario Dueño del Requisito	Caracteristica de evaluación 1	Caracteristica de evaluación 2	Caracteristica de evaluación n
Datos del Menor	1	Admin	Continua	Funcional	Continua

Datos del Responsable	2	Admin	Continua	Funcional	Continua
Operaciones de Pantalla	3	Admin	Continua	Sistemática	Continua
Reporte del Sistema	4	Admin	Continua	Funcional	Continua

Ilustración 3.55 Diseño de Caso de Prueba
Fuente: Metodología Gestión de Requisitos

3.9.9. Ejecución de Casos de prueba

Se tiene que considera los datos:

Cliente: Administrador del sistema

Proyecto: Sistema de Autorización de viajes

Requerimiento:

Responsable de ejecución: Desarrollador sistema

Fecha de inicio: 1 de Marzo

Fecha Fin: 30 de Junio

Nombre del Requerimiento								
	Numeral	Concisión	Comentar	# Error	No ambiguo	Comentar	# Error	Resultado
Datos del Menor	1		Mejorar Datos de Ingreso	0	NO		0	Corregido
Datos del Responsable	2		Mejorar Datos de Ingreso	2	NO		0	Corregido
Operaciones de Pantalla	3		Parpadeo en la Pantalla	0	NO		0	Corregido
Reporte del Sistema	4		Modificar características del reporte	0	NO		0	Corregido

Ilustración 3.56 Ejecución de Caso de Prueba
Fuente: Metodología Gestión de Requisitos

3.9.9.1. Informe final de prueba de requerimiento

En esta etapa se presenta el alcance definido inicialmente para el proyecto que finaliza.

3.10. VALORACIÓN DEL SISTEMA

Basados en la norma ISO/IEC 9126 se ha elaborado un cuestionario de evaluación (Ver Anexo E), permiten evaluar la calidad del Sistema, este resultado del software fue valorado a criterio de: un usuario identificado como administrador del sistema.

Se ha asignado un peso (valor numérico) a cada atributo del Sistema de Autorización de Viajes, de acuerdo al grado de importancia o nivel de necesidad exigida por el Gerente de la empresa, va desde 1 que significa muy poco importante, hasta el 10 que significa extremadamente importante, se ajustó cada pregunta de la evaluación a un atributo exigido, con estos resultados se aplicó una fórmula para obtener las métricas a cada atributo.

FUNCIONABILIDAD

Tabla 3.28 Funcionalidad

Ref.	Atributo	Peso	Resultado
1	Adecuación	10	5
2	Exactitud	10	5
3	Seguridad	10	5

Fuente: Elaboración Propia

$$\text{Medida Funcional} = \frac{(10*5)+(10*5)+(10*5)}{10+10+10} = 5$$

De acuerdo al resultado obtenido, se califica como valor aceptable, por lo que el Sistema Informático de facturación y ventas si cumple con la métrica de funcionalidad de acuerdo a los requerimientos exigidos.

FIABILIDAD

Tabla 3.29 Fiabilidad

Ref.	Atributo	Peso	Resultado
------	----------	------	-----------

1	Madurez	10	5
2	Recuperabilidad	10	5

Fuente: Elaboración Propia

$$\text{Medida Fiabilidad} = \frac{(10*5)+(10*5)}{10+10} = 5$$

De acuerdo al resultado obtenido, se califica como valor aceptable de la evaluación del Sistema de Autorización de Viajes (tiene las entradas validadas, se recupera la Información en caso de un incidente) lo cual significa que el sistema es seguro y recupera los datos almacenados fácilmente.

USABILIDAD

Tabla 3.30 Usabilidad

Ref.	Atributo	Peso	Resultado
6	Entendimiento	10	5
7	Aprendizaje	10	4
8	Operabilidad	10	4
9	Atracción	10	5

Fuente: Elaboración Propia

$$\text{Medida de Usabilidad} = \frac{(10*5)+(10*4)+(10*4)+(10*5)}{10+10+10+10} = 4,5$$

De acuerdo al resultado obtenido en su métrica de Usabilidad, se establece que el Sistema de Autorización de Viajes tiene una gran aceptación por parte del usuario, en el manejo de las interfaces y la facilidad de aprendizaje. Por el cual se otorga una buena puntuación, en esta fase es importante estar entre la puntuación de cuatro a cinco.

EFICIENCIA

Tabla 3.31 Eficiencia

Ref.	Atributo	Peso	Resultado
10	Comportamiento de Tiempos	10	5
11	Utilización de Recursos	10	4

Fuente: Elaboración Propia

$$\text{Medida de Eficiencia} = \frac{(10*5)+(10*5)}{10+10} = 5$$

De acuerdo al resultado obtenido, se establece que el Sistema Autorización de Viajes, cumple con los requerimientos exigidos en un determinado tiempo (hace lo que tiene que hacer, no demora mucho tiempo su ejecución), por lo que el software obtiene la puntuación aceptable de acuerdo a la métrica de eficiencia.

MANTENIMIENTO

Tabla 3.32 Mantenimiento

Ref.	Atributo	Peso	Resultado
12	Capacidad de ser analizado	10	4
13	Cambiabilidad	8	5
14	Estabilidad	5	5
15	Facilidad de prueba	5	4

Fuente: Elaboración Propia

$$\text{Medida de Mantenimiento} = \frac{(10*4)+(10*5)+(10*5)+(10*4)}{10+8+5+5} = 4.46$$

De acuerdo al resultado obtenido, se establece que el Sistema Autorización de Viajes cumple con los requerimientos mínimos exigidos para su mantenimiento y aceptabilidad. Por lo que se obtuvo una puntuación de **4.46**, lo cual significa que es aceptable.

PORTABILIDAD

Tabla 3.33 Portabilidad

Ref.	Atributo	Peso	Resultado
16	Capacidad de ser analizado	5	1
17	Cambiabilidad	5	4
18	Estabilidad	10	8

Fuente: Elaboración Propia

$$\text{Medida de Portabilidad} = \frac{(5*1)+(5*4)+(10*8)}{5+5+10} = 5.25$$

RESULTADO

Toda la información está basado en la encuesta (ver Anexo E)

Tabla 3.34 Valoración

Atributo	Puntaje
Funcionalidad	5
Fiabilidad	5
Usabilidad	4.5
Eficiencia	5
Mantenimiento	4.46
Portabilidad	5.25
PROMEDIO	4.86

Fuente: Elaboración Propia

Como se observa en la (TABLA 3.42), indica que el Sistema de Autorización de Viajes, cumple con las Métricas de Calidad de la Norma **ISO/IEC 9126**, por lo que **SÍ** es pertinente utilizarlo en la Defensoría de la Niñez y Adolescencia del Municipio de Cobija.

4. CAPITULO IV

4.1 CONCLUSIONES

4.2 RECOMENDACIONES

4.1. CONCLUSIONES

De acuerdo a los objetivos específicos, al concluir el proyecto se obtuvo lo siguiente:

Recolectar la información del procesamiento y control de la información que administra la institución, así determinar los requerimientos del sistema

En la etapa de análisis se realizó, la identificación de los actores que interactúan con el sistema, también se identificó los distintos requerimientos de la institución y de acuerdo a los requerimientos planteados se diseñó los diagramas de casos de uso en su versión inicial de los cuales se obtuvo los modelos que representan el comportamiento dinámico del sistema útil para la implementación.

En la etapa de diseño se realizó la elaboración de la línea de arquitectura del sistema se obtuvo el modelo conceptual de la base de datos, se comprendió el funcionamiento, de las interfaces de usuario, se modeló la estructura del sistema con diagramas de casos de uso y diagramas de clases con el fin de preparar la información necesaria para planificar la fase de construcción.

En la etapa de construcción se realizó el desarrollo del sistema a partir de una arquitectura basado en los diferentes modelos diseñados en la etapa de análisis y de esa manera se da paso al desarrollo del sistema donde se aplica el patrón de diseño modelo vista controlador haciendo uso de herramientas gestos de base de datos MySQL, Lenguaje de programación java.

En la etapa de implementación se realizó la instalación y configuración del sistema en base a la arquitectura cliente servidor, se capacitó a los funcionarios de la institución mencionados como actores en la etapa de análisis también se realizaron las respectivas pruebas y de mantenimientos

De esta manera se culminó con la implementación del sistema de información para la defensoría de la niñez y adolescencia del municipio de cobija y la capacitación del sistema (**ver Anexo F**).

4.2. RECOMENDACIONES

El presente proyecto de grado se culminó con la implementación del sistema de autorización de viajes para la defensoría de la niñez y adolescencia del municipio de Cobija, pero se recomienda lo siguiente:

- Cambiar algunas características al sistema de autorización de viajes y contemplar la posibilidad de modificar el diseño y coordine la Defensoría de la Niñez y Adolescencia del Municipio de Cobija con otras unidades de protección del menor.
- Implementar módulos para asesoría legal, psicología y trabajo social, para en un futuro ayude a centralizar toda la información de las unidades
- Implementar un sitio web para:
 - ✓ La socialización de la información que se genera la temática de protección y atención en la defensoría de la niñez y adolescencia en Cobija.
 - ✓ El registro de los datos menor y los datos del responsable para evitarse largas filas en las oficinas de la defensoría de la niñez y adolescencia.
- Se recomienda aplicar calidad del software y realizar un estudio de las herramientas que se pueden aplicar al Sistema de Autorización de Viaje.

5. BIBLIOGRAFÍA

Padilla, M. E., Carrasco, L., & Noya, M. (2010). <http://www.unicef.org/bolivia/>. Recuperado el 19 de 05 de 2016, de http://www.unicef.org/bolivia/Guia_de_Roles_y_Funciones_para_las_Defensorias_de_la_Ninez_y_Adolescencia.pdf

Acuña, K. B. (2009). *Biblioteca virtual*. Obtenido de <http://www.eumed.net>

Bolivia, D. d. (28 de enero de 2013). Obtenido de <http://www.portalfio.org/inicio/index.php/noticias/item/11772-bolivia-la-defensoria-lidera-el-control-en-el-traslado-de-ni%C3%B1os-ni%C3%B1as-y-adolescentes-en-el-tr%C3%B3pico-de-cochabamba.html>

Cadenhead, R. (2012). *Programacion JAVA 7*. España: ANAYA MULTIMEDIA.

Calle, I. L. (2011). *Sistema Informatico para el manejo y control de informacion de asistencias social de niños, niñas y adolescentes del sedeges Pando*. Cobija - Pando.

Carlos Coronel, S. M. (2011). *Base de datos Diseño, implementacion y administracion*. Estado de Mexico: Digital cromatica S.A. de C.V.

confidencial. (s.f.). *METODOLOGÍA*. Obtenido de Google Sites: www.sites.google.com/sites/metodologíareq/

Costas, G. (2010). *Guia de Roles y Funciones para las Defensorías de la Niñez y Adolescencia*. Bolivia: Viceministerio de Igualdad de Oportunidades, Centro Juana Azurduy, UNICEF. Obtenido de http://www.google.com.bo/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0CCMQFjAB&url=http%3A%2F%2Fwww.unicef.org%2Fbolivia%2FGuia_de_Roles_y_Funciones_para_las_Defensorias_de_la_Ninez_y_Adolescencia.pdf&ei=txCLVaTZMYeZNu_igbgI&usg=AFQjCnF4dl

Cruz Castillo, L. M. (2011). *Inegiería de software*. Mexico: PEARSON EDUCACION.

Fontela, C. (2011). *Modelado de software para profesionales*. Argentino: Alfaomega.

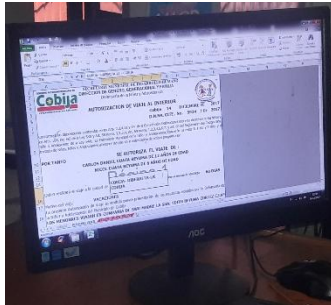
- Hill, M. (s.f.). *pedrobeltrancanessa-biblioteca.weebly.com*. Obtenido de pedrobeltrancanessa-biblioteca.weebly.com: pedrobeltrancanessa-biblioteca.weebly.com/.../sgbd-sistemas-gestores-bases-datos-primera-parte-22206-completo.pdf
- hombres, F. T. (2014). *Aproximaciones a la niñez en movimiento en España*. España: Fundación Tierra de hombres.
- Humana, P. d. (s.f.). Trata y Trafico. *Cartilla 2*, www.caritascbba.org/mobilidadhumana. Obtenido de <http://www.caritascbba.org/mobilidadhumana>
- Jemio, E. F. (2015). Exposición de tecnología de información de código abierto, software libre, gobierno electrónico y firma digital. *Boletín AE* 38, 2.
- Jhonny, A. V. (2006). Obtenido de Sistemas de Gestion y Seguimiento de Casos y Servicios de la Defensoria de la Niñez y Adolescencia de Chilimarca: http://www.cs.umss.edu.bo/rep_tesis?codigo=928&tipo_tes=2
- Montero, J. R. (s.f.). *Sistema gestorees de base de datos*. McGraw-Hill.
- Oportunidades, V. d. (2010). *Guia de roles y funciones para las defensorias de la niñez y adolescencia*. La Paz: Viceministerio de Igualdad de Oportunidades - Centro Juana Azurduy - UNICEF.
- Oportunidades, V. d. (2014). *Consultoria*. Obtenido de www.oportunidades.onu.org.bo: <http://www.oportunidades.onu.org.bo/roster/DesktopModules/UNJobs/UNGetMyFile.aspx?FileTicketId=934d8e50-db41-43c0-a798-711e5be84395>
- Plurinacional, L. A. (Julio 2014). *Codigo Niña, Niño y Adolescente*. La Paz Bolivia: SPC Impresores S. A.
- Pueblo, D. d. (julio de 2015). *Información Relevante sobre Niñez y Adolescencia en el Estado Plurinacional de Bolivia*. Obtenido de http://www.defensoria.gob.bo/sp/datos_ninas_ninos_adolescentes.asp
- pueblo, D. d. (gestion 2011). *Trata Trafico-Informe Defensorial*. Obtenido de www.defensoria.gob.bo: [www.defensoria.gob.bo/archivos/Trata Trafico - Informe Defensorial.pdf](http://www.defensoria.gob.bo/archivos/Trata_Trafico_-_Informe_Defensorial.pdf)
- Sanchez, A. M. (2009). Evaluación de métricas de calidad de software sobre un programa java. 89.
- Sánchez, A. M. (2010). Evaluación de métricas de calidad del software sobre un programa Java. *Proyecto de Fin de Máster en Programación y Tecnología Software*, 89.

- Senn, J. A. (1999). *Análisis y Diseño de Sistemas de Información*. Mexico: McGRAW-HILL.
- Sommerville, I. (2011). *Ingeniería de Software*. Mexico: Pearson Education.
- Sommerville, I. (2011). *Ingeniería de Software*. Mexico: PEARSON EDUCACIÓN.
- systems, s. (2016). www.sparxsystems.com.ar. Obtenido de www.sparxsystems.com.ar
- Unicef, T. d. (2012). *Trata de Personas*. Argentina: Gimol Pinto, Especialista en Protección de UNICEF - Eva Giberti, Coordinadora del Programa “Las Víctimas contra las Violencias” - Ministerio de Justicia y Derechos Humanos de la Nación. Obtenido de [www.unicef.org/argentina/spanish/Trata2012\(1\).pdf](http://www.unicef.org/argentina/spanish/Trata2012(1).pdf)
- Vargas, J. A. (2009). *Sistema de Registro y Seguimiento de Casos de la Defensoría de la Niñez y Adolescencia*. Cochabamba.
- www.frlp.utn.edu.ar/materias/info2/SI-Sistemas%20de%20Informacion.pdf. (s.f.).
- Zambrana, R. (08 de julio de 2016). *web.paginasiete.bo*. Obtenido de Director del Gobierno Electrónico y Modernización de la Alcaldía La Paz: <http://web.paginasiete.bo/sociedad/2016/7/7/alcaldia-habilita-plataforma-digital-para-permisos-viaje-para-menores-102091.html>

6. ANEXOS

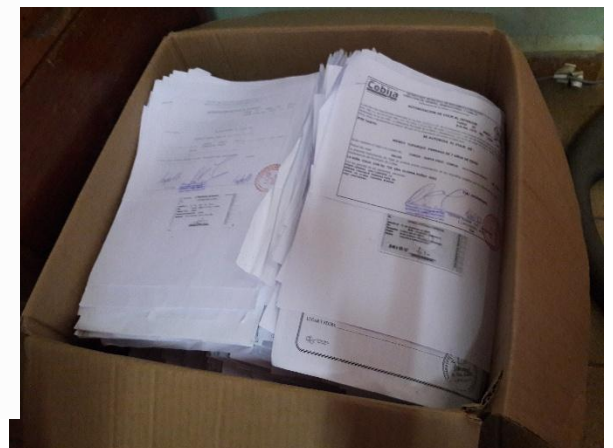
6.1. ANEXO A

6.1.1. Formulario Anterior de Autorización de Viajes

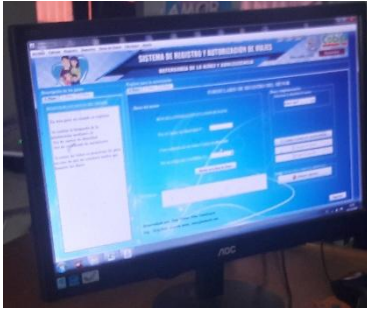


6.1.1.1. Registro almacenado en archivos excel

Nombre	Fecha de modifica...	Tipo	Tamaño
AARON USSIEL RUEDA ROCA 1271	21/02/2017 10:23 a...	Hoja de cálculo d...	48 KB
AARON MISAEL AYALA TARAUNE 1026	03/02/2017 03:49 ...	Hoja de cálculo d...	54 KB
abdiel cesar vargas ticona 1291	21/02/2017 04:59 ...	Hoja de cálculo d...	48 KB
ABIGAIL FERNANDEZ CAMACHO 1151	10/02/2017 06:45 ...	Hoja de cálculo d...	48 KB
ABNER HUMEREZ NOZA 1307	22/02/2017 04:16 ...	Hoja de cálculo d...	54 KB
ADELGHYLMER ALARCON VILLCA 1297	22/02/2017 10:26 a...	Hoja de cálculo d...	48 KB
ADRIAN MORENO ROCA 1183	14/02/2017 06:19 ...	Hoja de cálculo d...	48 KB
ADRIANA DAYERLIN MAMANI PILLCO ...	10/02/2017 05:22 ...	Hoja de cálculo d...	48 KB
ADRIANA ROJAS ROJAS 1270	21/02/2017 10:19 a...	Hoja de cálculo d...	48 KB
ADRIANA MERCIE FARIÑAS 1299	22/02/2017 09:57 a...	Hoja de cálculo d...	54 KB
ADRIANO SERGIO CORREA RAMIREZ 1...	24/02/2017 03:37 ...	Hoja de cálculo d...	54 KB
ADRIEL HERRERA MEDRANO 1131	10/02/2017 11:28 a...	Hoja de cálculo d...	48 KB
ADRIELY HUMADAY MAEDA 1290	21/02/2017 06:00 ...	Hoja de cálculo d...	48 KB
AISHA MARCELLY BARROSO 959	01/02/2017 09:09 a...	Hoja de cálculo d...	48 KB
ALAN NICOLAS CHOQUE HERRERA 12...	21/02/2017 05:13 ...	Hoja de cálculo d...	48 KB
ALAN KALED HUANCA GOMEZ 1218	16/02/2017 05:02 ...	Hoja de cálculo d...	55 KB
ALAN KALED HUANCA GOMEZ 1219HN...	16/02/2017 05:19 ...	Hoja de cálculo d...	55 KB
ALBA AYLLEN CUELLAR YANAMO 966	01/02/2017 03:33 ...	Hoja de cálculo d...	55 KB
ALBERT ESCOBAR VARGAS 1030	03/02/2017 04:20 ...	Hoja de cálculo d...	54 KB
ALEJANDRA CAMILA BAUTISTA RODRI...	03/02/2017 06:23 ...	Hoja de cálculo d...	54 KB
ALEXANDER CHOQUE CORNEJO 1037	03/02/2017 06:32 ...	Hoja de cálculo d...	54 KB
ALEXIA VALENTINA RAMOS ROJAS 1012	03/02/2017 11:24 a...	Hoja de cálculo d...	54 KB
ALICIA ALEJANDRA JOFFRE PILUY 1138	10/02/2017 04:25 ...	Hoja de cálculo d...	48 KB
AMAILY RAMIREZ ALVAREZ 1181	14/02/2017 05:50 ...	Hoja de cálculo d...	48 KB
AMANDEA JHANET ARPIO CALLISAYA ...	20/02/2017 10:17 a...	Hoja de cálculo d...	48 KB



6.1.2. Formulario Actual de Autorización de viajes



GOBIERNO AUTÓNOMO MUNICIPAL DE Cobiya
SECRETARÍA MUNICIPAL DE DESARROLLO HUMANO
DIRECCIÓN DE GENERO, GENERACIONAL Y FAMILIA
DEFENSORÍA DE LA NIÑEZ Y ADOLESCENCIA
AUTORIZACION DE VIAJE AL INTERIOR

Cobiya, 17 octubre 2017 D.N.N.A. CITE N° 45/2017

Conforme a las disposiciones contenidas en los Arts. 2, 3, 4, 11 y 14 de la Convención Internacional sobre los derechos de los Niños y los Arts. 188, inc. ee, de la Ley 548 y Art. 59 núm. 1, 2, 3, 4, Art. 59 núm. 1, 2, 3, 4, 5, 6, 7, y art. 60 del Reglamento del Código Niña, Niño y Adolescente de la Ley 548; La Defensoría Municipal de la Niñez y Adolescencia, con el fin de evitar la trata, tráfico y el traslado de niños, niñas y adolescentes al interior del país sin la autorización de ambos progenitores:

POR LO TANTO: SE AUTORIZA EL VIAJE

NIÑA - CIELITO MARBEL CARMONA ROCA	2 años	Tiempo de Viaje
Nombre Completo	Su edad	14
Quien realizará el viaje de la ciudad	COBUIA	SANTA CRUZ - COBUIA
Origen	Destino	
Motivo del Viaje:	VACACIONES	
Observación: EL MENOR VIAJA EN COMPAÑÍA DE SU PADRE		

La presente autorización de viaje es emitida previa presentación de los requisitos exigidos por la Defensoría de la Niñez y Adolescencia del Municipio de Cobiya.

ALVARO ROJAS ZENTENO - PADRE - C.I.: 6509334 Cbba Padre/Madre/Familiar	ALVARO ROJAS ZENTENO - PADRE - C.I.: 6509334 Cbba Responsable/Tutor/Guardador/Apoderado/Testigo
---	--

6.1.2.1. Registro almacenado en una base de datos

phpMyAdmin

Reciente Favoritas

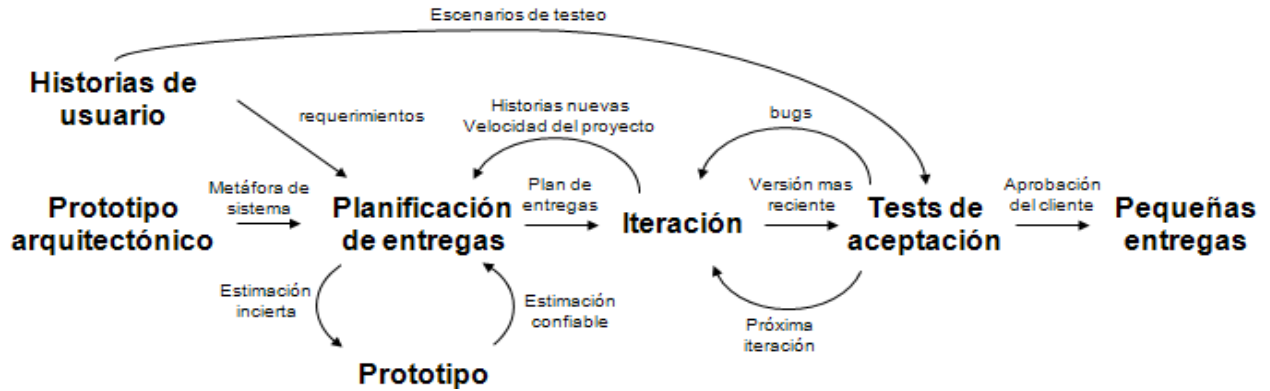
base_ueba
 cdcol
 information_schema
 mysql
 nna4

Nueva
 autorizacion
 digital
 estaditica
 familia
 identidad
 menor
 pendientes
 perfil
 perfil_familiar
 persona
 reporte
 reporteannual
 requisitos
 responsable
 sistema
 usuario
 performance_schema

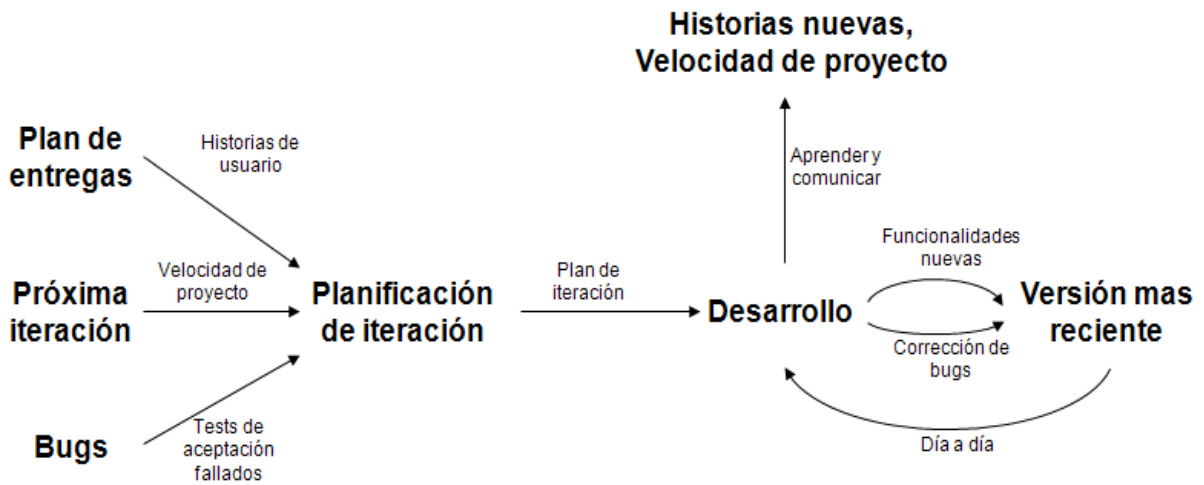
Tabla	Replicación	Acción	Filas	Tipo	Cotejamiento	Tamaño	Re de
autorizacion		Examinar Estructura Buscar Insertar Vaciar Eliminar	13	InnoDB	latin1_swedish_ci	32 KB	
digital		Examinar Estructura Buscar Insertar Vaciar Eliminar	22	InnoDB	latin1_swedish_ci	16 KB	
estaditica		Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	32 KB	
familia		Examinar Estructura Buscar Insertar Vaciar Eliminar	13	InnoDB	latin1_swedish_ci	48 KB	
identidad		Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	latin1_swedish_ci	16 KB	
menor		Examinar Estructura Buscar Insertar Vaciar Eliminar	9	InnoDB	latin1_swedish_ci	64 KB	
pendientes		Examinar Estructura Buscar Insertar Vaciar Eliminar	13	InnoDB	latin1_swedish_ci	64 KB	
perfil		Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	latin1_swedish_ci	16 KB	
perfil_familiar		Examinar Estructura Buscar Insertar Vaciar Eliminar	10	InnoDB	latin1_swedish_ci	16 KB	
persona		Examinar Estructura Buscar Insertar Vaciar Eliminar	32	InnoDB	latin1_swedish_ci	16 KB	
reporte		Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	32 KB	
reporteannual		Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	latin1_swedish_ci	16 KB	
requisitos		Examinar Estructura Buscar Insertar Vaciar Eliminar	22	InnoDB	latin1_swedish_ci	32 KB	
responsable		Examinar Estructura Buscar Insertar Vaciar Eliminar	13	InnoDB	latin1_swedish_ci	48 KB	
sistema		Examinar Estructura Buscar Insertar Vaciar Eliminar	9	InnoDB	latin1_swedish_ci	32 KB	
usuario		Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	latin1_swedish_ci	48 KB	
16 tablas	Replicación	Número de filas	170	InnoDB	latin1_swedish_ci	528 KB	

6.2. ANEXO B

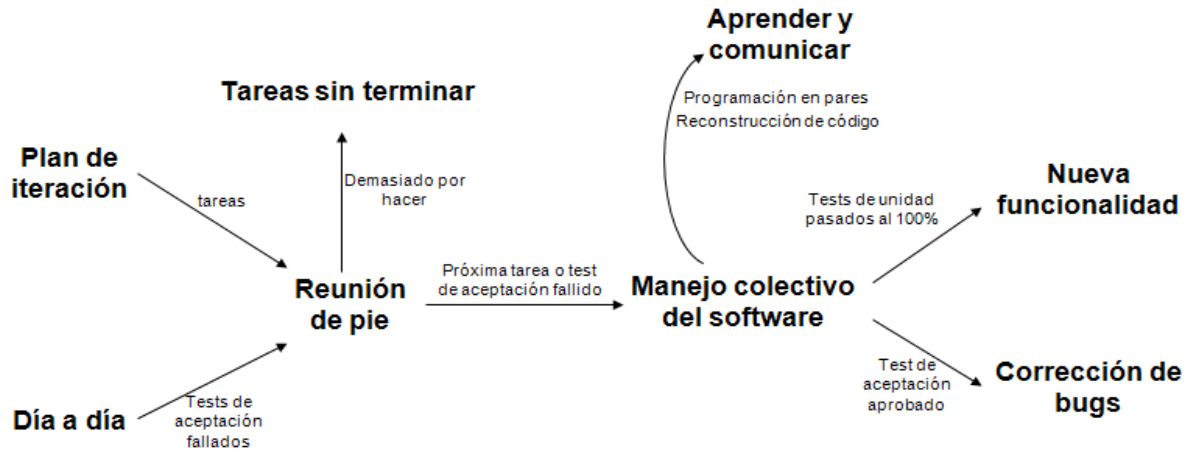
6.2.1. METODOLOGÍA PROGRAMACIÓN EXTREMA



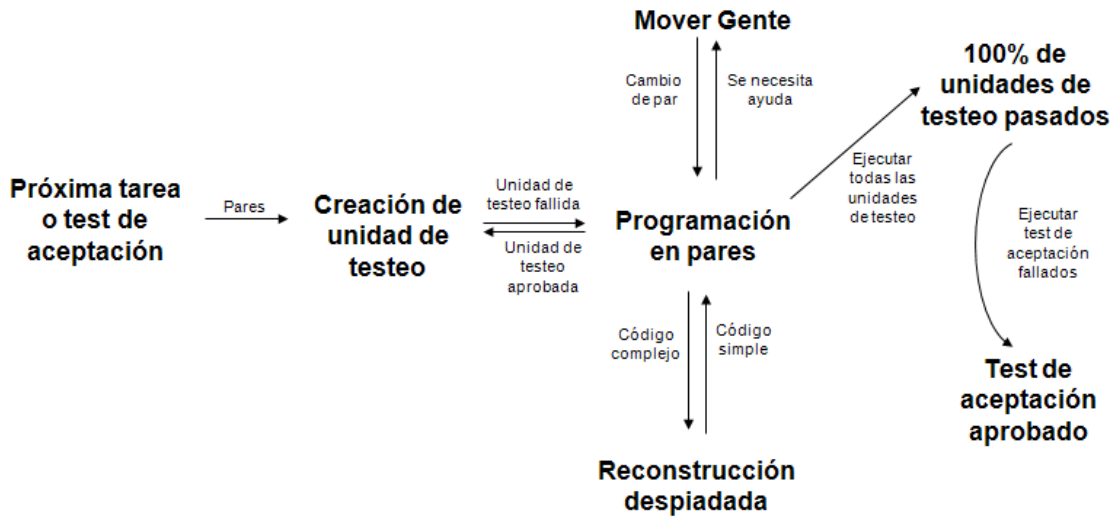
6.2.2. Iteración



6.2.3. Desarrollo



6.2.4. Manejo Colectivo del Código



6.3. ANEXO C

6.3.1. Organigrama del Gobierno Autónomo Municipal De Cobija

Como se muestra en la ilustración 2.4, el organigrama del Gobierno Autónomo Municipal de Cobija del gestión 2016.

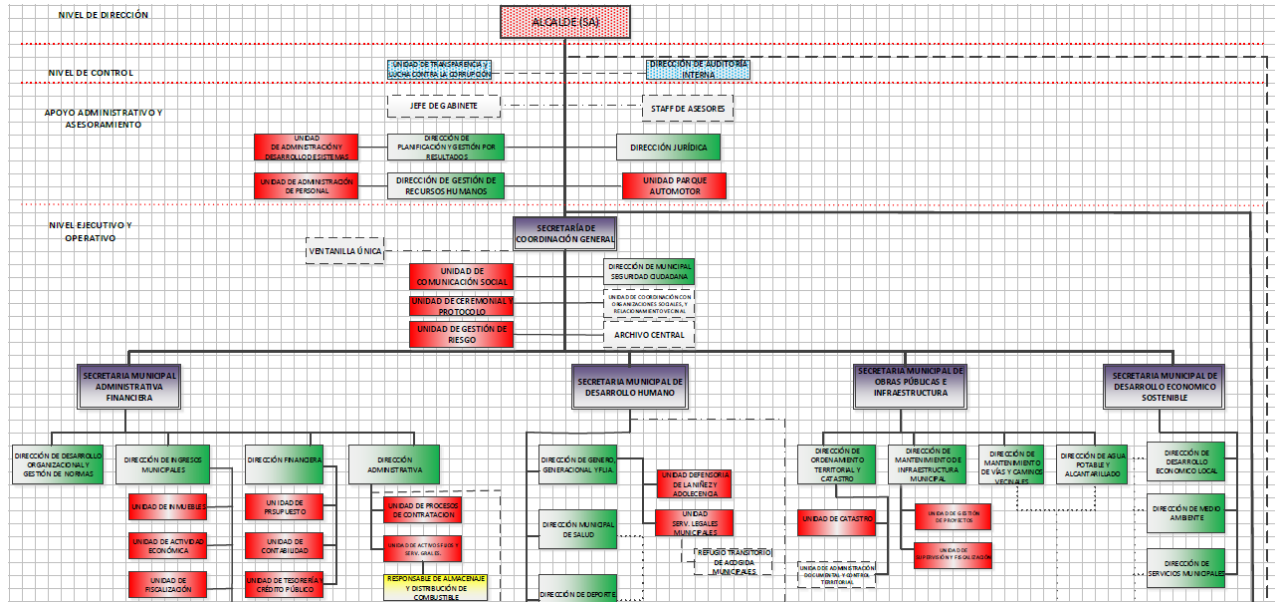


Ilustración 6.57 Organigrama GAMC
Fuente: Página Oficial GAMC

6.4. ANEXO D

6.4.1. Inicio de las Pruebas Unitarias con el *framework* JUnit

En esta parte se evalúa si el funcionamiento de cada uno de los métodos de la clase, seleccionando el archivo al que se le hará el testeo.

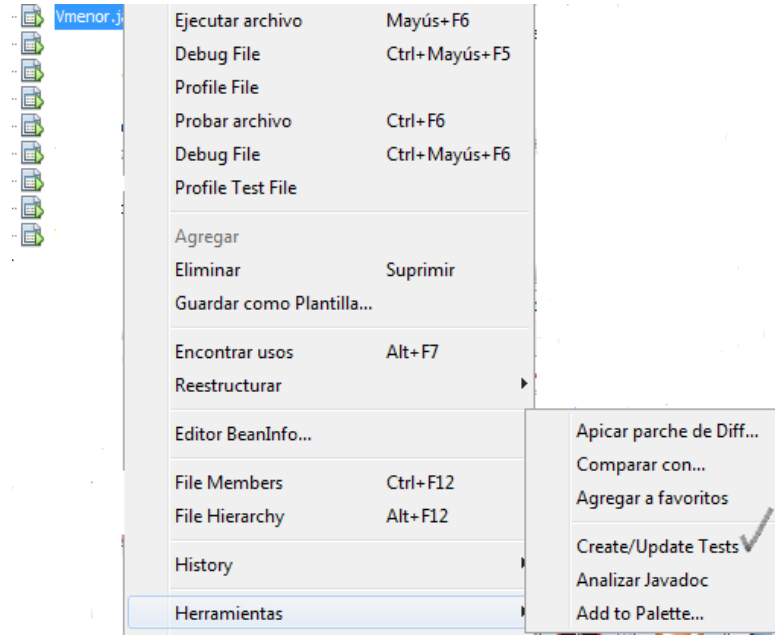


Ilustración 6.58 Prueba a un Archivo Java
Fuente: Elaboración Propia

Una vez seleccionado la opción “Create/Update Tests” se genera un archivo de paquete de prueba

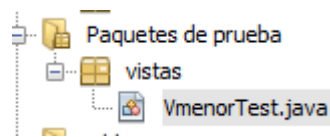


Ilustración 6.59 Archivo Generado por JUnit
Fuente: Elaboración Propia

Una vez seleccionado el archivo “**VmenorTest.java**” se ejecuta, es donde se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado

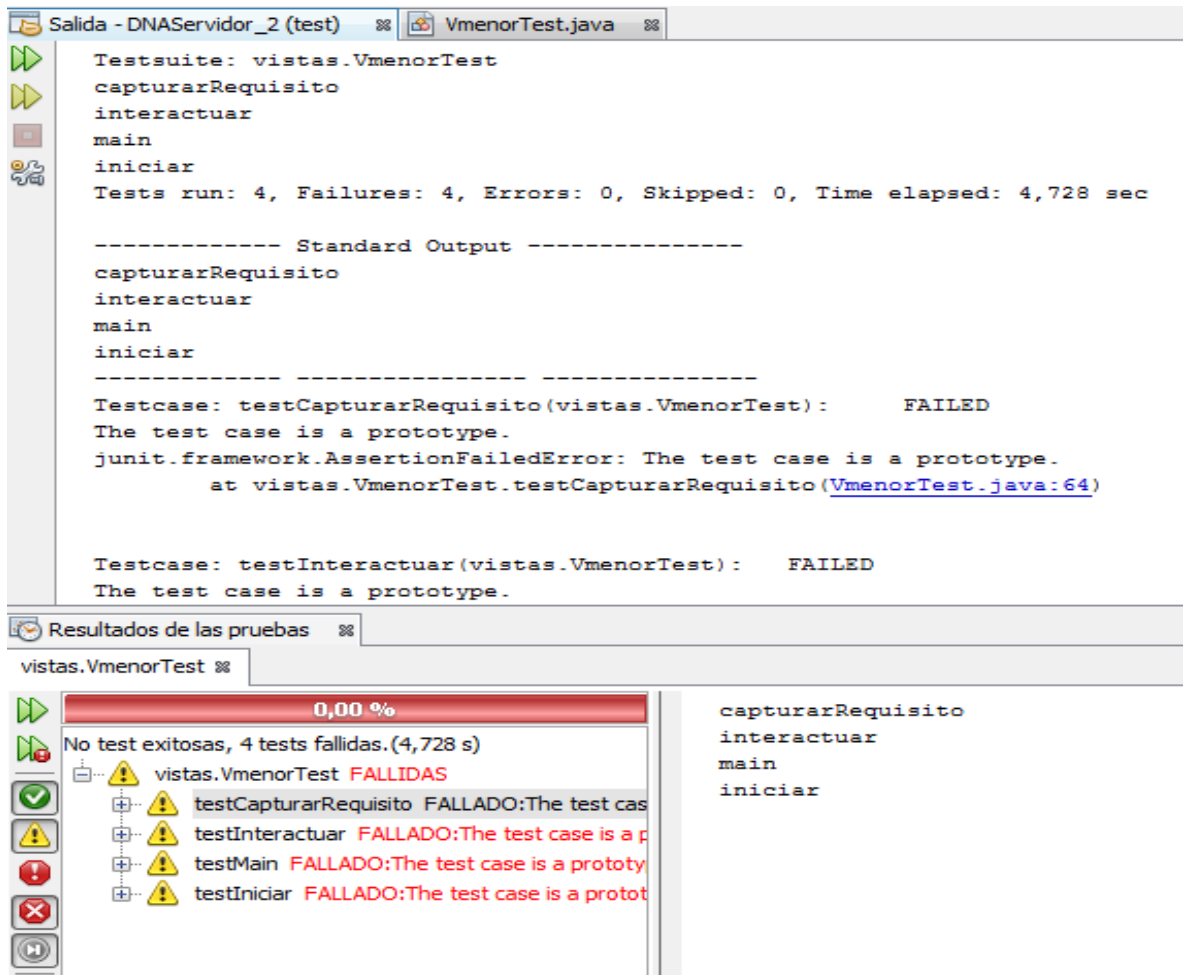


Ilustración 6.60 Resultados de testeo
Fuente: Elaboración Propia

En conclusión este es el procedimiento por el cual se apoya en verificar la calidad de pruebas unitarias de cada módulo del proyecto, además que se aplica en cada iteración según la metodología XP.

6.5. ANEXO E

6.5.1. Evaluación del sistema de acuerdo a las normas ISO/IEC 9126

CALIFICACIÓN	PUNTAJE
Muy Malo	1
Malo	2
Medio	3
Bueno	4
Muy Bueno	5

Ref.:	Atributo	Puntaje
Funcionalidad		
1	¿Cumple los requerimientos funcionales especificados en las historias de usuario?	5
2	¿El sistema hace lo acordado (Ej. un reporte) y en forma correcta?	5
3	¿El Sistema pide que se autentifique con su usuario, contraseña antes de ingresar?	5
Fiabilidad		
4	¿Las entradas al sistema están debidamente validadas?	5
5	¿En caso de un incidente grave que afecte el funcionamiento del programa su recuperación será no mayor a un día?	5
Usabilidad		
6	¿La interfaz del usuario presenta buena estética para su utilización?	5
7	¿El sistema cuenta con la documentación (impreso, descripciones en la interfaz, etiquetas) necesaria para su aprendizaje?	4
8	¿Las entradas de datos están debidamente etiquetadas?	4
9	¿Es atractivo el diseño del sistema?	5
Eficiencia		
10	¿El sistema responde de manera rápida a peticiones que realiza?	5
11	¿Toma menor tiempo para la búsqueda de información?	4
Mantenibilidad		
12	¿Existe documentación del código fuente y la base de datos los cuales faciliten realizar modificaciones en el sistema?	4
13	¿Es fácil de realizar modificaciones en el sistema?	5
14	¿Existe independencia en entre la base de datos y software?	5
15	¿Son fáciles de validar las modificaciones?	4
Portabilidad		
16	¿Es compatible con diferentes sistemas operativos?	1

17	¿Incompatibilidad con otros sistemas de información e interfieran la coexistencia?	4
18	¿El sistema cuenta con un asistente que facilite una nueva instalación?	8

6.6. ANEXO F

6.6.1. Capacitación al personal de la Defensoría de la Niñez y Adolescencia

