

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**DESARROLLO DE UN SISTEMA WEB DE GESTIÓN
DE BRIGADAS MÉDICAS DEL ÁREA 2
DE SALUD DE PASTAZA**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

DANY MIGUEL GANCINO CORDONES

danygancino@gmail.com

EDWIN FERNANDO MARTÍNEZ NÚÑEZ

edwin.fmn@gmail.com

DIRECTOR: MSC. ING. RAÚL CÓRDOVA

raul.cordova@epn.edu.ec

Quito, Febrero 2011

DECLARACIÓN

Nosotros, Dany Miguel Gancino Cordones y Edwin Fernando Martínez Núñez, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Dany Miguel Gancino Cordones

Edwin Fernando Martínez Núñez

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Dany Miguel Gancino Cordones y Edwin Fernando Martínez Núñez, bajo mi supervisión.

MSc. Ing. Raúl Córdova
DIRECTOR DEL PROYECTO

CONTENIDO

CAPÍTULO I. ANÁLISIS DE LA GESTIÓN DE BRIGADAS MÉDICAS DEL ÁREA 2 DE SALUD DE PASTAZA.....	1
1.1. DESCRIPCIÓN DE LA SITUACIÓN ACTUAL DEL ÁREA 2 DE SALUD DE PASTAZA	1
1.1.1. ESTRUCTURA ORGÁNICA DEL MINISTERIO DE SALUD	1
1.1.2. SITUACIÓN ACTUAL DEL ÁREA 2 DE SALUD DE PASTAZA	2
1.2. SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO	4
1.2.1.4. UNIFIED MODELING LANGUAGE (UML)	8
1.2.2. EXTREME PROGRAMING (XP)	11
1.2.3. MICROSOFT SOLUTION FRAMEWORK (MSF).....	17
1.2.4. ANALISIS COMPARATIVO DE LAS METODOLOGIAS DE DESARROLLO	22
1.2.5. JUSTIFICACIÓN DE LA METODOLOGIA DE DESARROLLO	25
1.3. SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO	26
1.3.1. FRAMEWORK DE DESARROLLO.....	26
1.3.2. GESTOR DE BASE DE DATOS (DBMS)	30
1.3.2.3. Análisis Comparativo de los DBMS's	31
1.3.3. JUSTIFICACIÓN DE LAS HERRAMIENTAS DE DESARROLLO.....	32
CAPÍTULO II. ANÁLISIS Y DISEÑO DEL SISTEMA WEB.....	33
2.1. ESPECIFICACIÓN DE REQUERIMIENTOS	33
2.1.1. INTRODUCCIÓN.....	33
2.1.2. DESCRIPCIÓN GENERAL.....	34
2.1.3. ESPECIFICACIÓN DE REQUERIMIENTOS	35
2.2. ANÁLISIS	52
2.2.1. CASOS DE USO	52
2.2.2. CLASES	91
2.3. DISEÑO.....	95
2.3.1. MODELO ENTIDAD – RELACIÓN	95
2.3.2. DIAGRAMAS DE SECUENCIA.....	96
CAPÍTULO III. DESARROLLO, PRUEBAS E IMPLANTACIÓN DEL SISTEMA.....	110
3.1. DESARROLLO	110
3.1.1. DIAGRAMA DE DESPLIEGUE	110

3.1.2.	DIAGRAMA DE COMPONENTES.....	111
3.1.3.	CODIFICACIÓN	114
3.2.	PRUEBAS	116
3.3.	IMPLANTACIÓN.....	126
3.3.1.	REQUISITOS DE HARDWARE.....	126
3.3.2.	REQUISITOS DE SOFTWARE.....	126
3.3.3.	PLAN DE INSTALACIÓN.....	127
3.4.	ANÁLISIS DE RESULTADOS	128
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES.....		130
4.1.	CONCLUSIONES.....	130
4.2.	RECOMENDACIONES.....	131

INDICE DE TABLAS

Tabla 1.1	Análisis Comparativo de las Metodologías de Desarrollo	23
Tabla 1.2	Análisis Comparativo de las Metodologías de Desarrollo con respecto al proyecto	25
Tabla 1.3	Análisis comparativo del Framework de Desarrollo	28
Tabla 1.4	Análisis Comparativo del Framework de Desarrollo con respecto al proyecto.....	29
Tabla 1.5	Análisis comparativo de los DBMS´s.....	31
Tabla 2.1	Módulos y Requerimientos del Sistema	37
Tabla 2.2	Requerimientos de Administración de Usuarios	38
Tabla 2.3	Requerimientos de Administración de Unidades Operativas	38
Tabla 2.4	Requerimientos de Administración de Comunidades	39
Tabla 2.5	Requerimientos de Gestión de Recursos Humanos	40
Tabla 2.6	Requerimientos de Administración de Suministros.....	40
Tabla 2.7	Requerimientos de Administración de Familias	41
Tabla 2.8	Requerimientos de Administración de Personas	41
Tabla 2.9	Requerimientos de Gestión de Brigadas	42
Tabla 2.10	Requerimientos Técnicos	51
Tabla 2.11	Caso de Uso crearUsuario.....	62
Tabla 2.12	Caso de Uso buscarUsuario	63
Tabla 2.13	Caso de Uso modificarUsuario.....	64
Tabla 2.14	Caso de Uso eliminarUsuario	65
Tabla 2.15	Caso de Uso crearUnidadOperativa.....	66
Tabla 2.16	Caso de Uso buscarUnidadOperativa	67
Tabla 2.17	Caso de Uso modificarUnidadOperativa	68
Tabla 2.18	Caso de Uso eliminarUnidadOperativa	68
Tabla 2.19	Caso de Uso crearComunidad	70
Tabla 2.20	Caso de Uso buscarComunidad	70
Tabla 2.21	Caso de Uso modificarComunidad.....	71
Tabla 2.22	Caso de Uso eliminarComunidad.....	72
Tabla 2.23	Caso de Uso mostrarIndices	73

Tabla 2.24 Caso de Uso crearFamilia	76
Tabla 2.25 Caso de Uso buscarFamilia	76
Tabla 2.26 Caso de Uso modificarFamilia	77
Tabla 2.27 Caso de Uso eliminarFamilia	78
Tabla 2.28 Caso de Uso crearPersona	79
Tabla 2.29 Caso de Uso buscarPersona	80
Tabla 2.30 Caso de Uso modificarPersona	81
Tabla 2.31 Caso de Uso eliminarPersona	81
Tabla 2.32 Caso de Uso crearRecursoHumano	82
Tabla 2.33 Caso de Uso buscarRecursoHumano	83
Tabla 2.34 Caso de Uso modificarRecursoHumano	84
Tabla 2.35 Caso de Uso CrearSuministro	85
Tabla 2.36 Caso de Uso buscarSuministro	86
Tabla 2.37 Caso de Uso ModificarSuministro	87
Tabla 2.38 Caso de Uso EliminarSuministro	87
Tabla 2.39 Caso de Uso crearBrigada	88
Tabla 2.40 Caso de Uso buscarBrigada	89
Tabla 2.41 Caso de Uso modificarBrigada	90
Tabla 3.1 Caso de Prueba Administrar Usuario	117
Tabla 3.2 Caso de Prueba Administrar Unidad Operativa	118
Tabla 3.3 Caso de Prueba Administrar Comunidad	120
Tabla 3.4 Caso de Prueba Índices de Crecimiento	120
Tabla 3.5 Caso de Prueba Administrar Familia	121
Tabla 3.6 Caso de Prueba Administrar Persona	122
Tabla 3.7 Caso de Prueba Administrar Suministro	123
Tabla 3.8 Caso de Prueba Administrar Recurso Humano	125
Tabla 3.9 Caso de Prueba Administrar Brigada	125
Tabla 3.10 Análisis de Resultados	129

INDICE DE FIGURAS

Figura 1.1 Estructura Orgánica del Ministerio de Salud	1
Figura 1.2 Dimensiones del RUP	5
Figura 1.3 Diagramas UML	9
Figura 1.4 Metodología Extreme Programming	13
Figura 1.5 Metodología MSF	18
Figura 2.1 Modelo de interfaces	43
Figura 2.2 Paquetes para el Sistema de Gestión de Brigadas	52
Figura 2.3 Paquete Administración de Usuarios	54
Figura 2.4 Paquete Administración de Unidades Operativas	55
Figura 2.5 Paquete Administración de Comunidades	56
Figura 2.6 Paquete Administración de Familias	57
Figura 2.7 Paquete Administración de Personas	58
Figura 2.8 Paquete Gestión de Recursos Humanos	59
Figura 2.9 Paquete Administración de Suministros	60

Figura 2.10 Paquete Gestión de Brigadas	61
Figura 2.11 Diagrama de Clases	94
Figura 2.12 Diagrama Entidad - Relación	95
Figura 2.13 Diagrama de Secuencia crearUsuario.....	96
Figura 2.14 Diagrama de Secuencia buscarUsuario	97
Figura 2.15 Diagrama de Secuencia crearUnidadOperativa	98
Figura 2.16 Diagrama de Secuencia modificarUnidadOperativa	99
Figura 2.17 Diagrama de Secuencia buscarComunidad	100
Figura 2.18 Diagrama de Secuencia eliminarComunidad	101
Figura 2.19 Diagramas de Secuencia crearFamilia.....	102
Figura 2.20 Diagramas de Secuencia modificarFamilia	103
Figura 2.21 Diagramas de Secuencia buscarPersona	104
Figura 2.22 Diagramas de Secuencia modificarPersona.....	105
Figura 2.23 Diagramas de Secuencia crearRecursoHumano.....	106
Figura 2.24 Diagramas de Secuencia modificarSuministro	107
Figura 2.25 Diagramas de Secuencia eliminarSuministro	108
Figura 2.26 Diagramas de Secuencia crearBrigada	109
Figura 3.1 Diagrama de Despliegue.....	110
Figura 3.2 Diagrama de Componentes	111
Figura 3.3 Diagrama de Componentes – Administrar Usuario	112
Figura 3.4 Diagrama de Componentes – Administrar Unidad Operativa	113

CAPÍTULO I. ANÁLISIS DE LA GESTIÓN DE BRIGADAS MÉDICAS DEL ÁREA 2 DE SALUD DE PASTAZA

1.1. DESCRIPCIÓN DE LA SITUACIÓN ACTUAL DEL ÁREA 2 DE SALUD DE PASTAZA

1.1.1. ESTRUCTURA ORGÁNICA DEL ÁREA 2 DE SALUD

En la Figura 1.1, se muestra la Estructura Orgánica del Área 2 de Salud. Como se observa cada Dirección Provincial tiene Áreas de Salud también llamadas Unidad de Gestión de Área (UCA), cada Unidad de Gestión de Área está conformada por Subcentros de Salud denominados Unidades Operativas (UO), y finalmente cada Unidad Operativa está formada por Comunidades también conocidas como Áreas de Influencia.



Figura 1.1 Estructura Orgánica del Área 2 de Salud

Elaborado por: Dany Gancino, Edwin Martínez

1.1.2. SITUACIÓN ACTUAL DEL ÁREA 2 DE SALUD DE PASTAZA

El área 2 de salud de Pastaza se encarga de brindar atención integral a la población de las comunidades que se encuentran dentro del área de influencia; el proceso de atención se lo realiza enviando una brigada médica compuesta de un grupo de médicos odontólogos y enfermeras a esas comunidades, los cuales ofrecen servicio médico odontológico, aplicación de vacunas y ejecución de charlas educativas.

La brigada médica comienza cuando una de las áreas de la dirección provincial ve la necesidad de atender a una o varias comunidades, arrancando una serie de subprocesos que se los puede dividir en los siguientes:

- Designar la brigada a la entidad correspondiente
- Escoger el recurso humano para la brigada médica.
- Escoger los insumos y materiales necesarios para la brigada médica
- Registro de la brigada médica

1.1.2.1. Designar la brigada a la entidad correspondiente

Las brigadas médicas son gestionadas por las Áreas de Salud o por las Unidades Operativas.

Las Áreas de Salud gestionan todas las brigadas de mayor magnitud y son las únicas que pueden realizar brigadas a comunidades que necesita transporte aéreo. En este tipo de brigadas se consideran factores como peso del equipo médico, de los insumos, vacunas, las medicinas, el recurso humano y el número de comunidades a visitar.

Por otro lado las Unidades Operativas realizan brigadas que están dentro de su área de influencia, son comunidades cercanas que tienen acceso terrestre.

1.1.2.2. Escoger el recurso humano para la brigada médica.

La asignación del recurso humano se lo hace escogiendo al personal médico, odontológico y de enfermería perteneciente al área de influencia del Área de Salud o Unidad Operativa.

Las Unidades Operativas tienen a su disposición un conjunto de profesionales destinados a brigadas médicas, mientras que las Áreas de Salud Solicitan a sus Unidades el recurso humano necesario para sus brigadas.

1.1.2.3. Escoger los insumos y materiales necesarios para la brigada médica

Los insumos, materiales, equipos médicos que el recurso humano va a repartir o utilizar en la brigada tiene que ser registrado, esta información es importante para conocer la atención que se ha brindado en cada una de las comunidades como el peso a ser considerado al momento de contratar el transporte necesario.

1.1.2.4. Registro de actividades realizadas en la brigada médica

Cuando la brigada llega a la comunidad, se abre una ficha médica por cada familia, en la cual se registra la fecha de la visita, el nombre del jefe de familia, el nombre de cada miembro de la familia y su parentesco con el jefe de familia y finalmente los datos de la atención recibida por el médico, odontólogo y enfermeras.

Para finalizar la Brigada médica, se recolecta toda la información y se solicita al médico que realice un informe diario de la atención realizada, toda la documentación reposa en la entidad correspondiente.

1.2. SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO

Las metodologías de software son el conjunto de políticas, reglas, procedimientos que definen los pasos a seguir para llegar a la culminación de un proyecto de software garantizando la eficacia y eficiencia del desarrollo durante su ciclo.

El escoger adecuadamente la metodología a seguir durante el desarrollo de software determina el éxito o no del proyecto a realizar, por lo cual el siguiente análisis se realiza para elegir la que mejor se adapte a los requerimientos planteados por el cliente. A continuación se detallan las que se consideran más apropiadas: RUP¹ con UML², XP³ y MSF⁴.

1.2.1. RUP (RATIONAL UNIFIED PROCESS) CON UML (UNIFIED MODELING LANGUAGE)

RUP es el conjunto de procesos que acompañado de la notación UML conforman una metodología de desarrollo que sigue procesos disciplinados para asignar tareas y responsabilidades, detallando y documentando todo el proceso de desarrollo.

1.2.1.1. Dimensiones del RUP

RUP está formado por dos dimensiones una horizontal y una vertical.

- La horizontal representa el ciclo de vida, fases del proyecto de acuerdo al transcurso del tiempo
- La vertical agrupa actividades definidas lógicamente por la naturaleza del proyecto, iteraciones.

¹ RUP.- Rational Unified Process.

² UML.- Unified Modeling Language.

³ XP.- Extreme Programming.

⁴ MSF.- Microsoft Solution Framework.

En la Figura 1.2 se representa cómo varía el énfasis de cada disciplina en un cierto plazo en el tiempo y durante cada una de las fases.

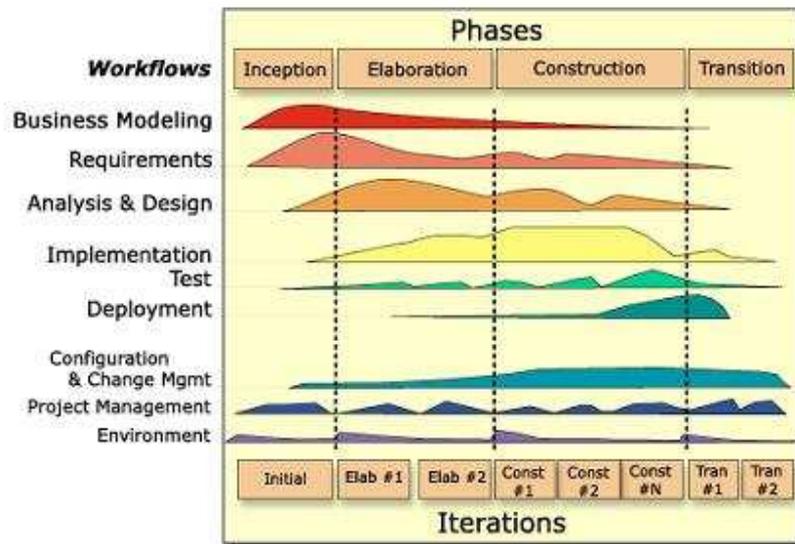


Figura 1.2 Dimensiones del RUP ⁵

1.2.1.2. Fases

1.2.1.2.1. Inicio

En esta fase se define el ámbito y objetivos del proyecto con la funcionalidad y capacidades que tendrá el producto final.

1.2.1.2.2. Elaboración

Tanto la funcionalidad como el dominio del problema se estudian en profundidad, definiendo una arquitectura básica y considerando los recursos disponibles para el proyecto.

⁵ Fuente: <http://moleculax.blogspot.com/2008/06/el-proceso-unificado-de-desarrollo-de.html>

1.2.1.2.3. Construcción

En esta fase se procede al desarrollo del producto, considerando las iteraciones del proceso se realiza el análisis, diseño e implementación que se revisaron en fases iniciales.

Se redefine de ser necesaria la arquitectura durante la programación y pruebas que forman la mayor parte de esta fase.

Se debe documentar tanto el manejo como el funcionamiento del sistema mismo.

1.2.1.2.4. Transición

Se libera un reléase del producto que es entregado al usuario para su certificación, teniendo en cuenta tareas como marketing, empaquetado atractivo, instalación, configuración, entrenamiento, soporte, mantenimiento, etc.

Se refinan los manuales para el usuario, instalación y técnicos con la información obtenida en esta fase.

1.2.1.3. Disciplinas de Desarrollo

Las disciplinas conllevan los flujos de trabajo, los cuales son una secuencia de pasos para la culminación de cada disciplina, estas disciplinas se dividen en dos grupos: las primarias y las de apoyo.

1.2.1.3.1. Disciplinas Primarias

Modelado del Negocio: Su principal objetivo es comprender el problema actual e identificar posibles mejoras utilizando el modelo de casos de uso del negocio para describir los procesos del negocio, además en esta disciplina se utilizan diagramas de objetos, actividades y de clases para mejorar el entendimiento del negocio.

Requerimientos: Se trasladan las necesidades del negocio a un sistema automatizado especificando los requisitos encontrados en la fase de modelamiento del negocio, se define un alcance, interfaces de usuario y estimación de tiempos y costos. Se utiliza casos de uso para modelar el sistema e identificar actores y de ser necesario diagramas de cada caso de uso para reconocer a profundidad su funcionalidad.

Análisis y Diseño: Se trasladan los requerimientos dentro de la arquitectura de software transformando cada caso de uso en clases y refinando el análisis para implementar en cada caso de uso.

Se usan además los diagramas de colaboración, clases de diseño, secuencia de estado y el modelo de despliegue de la arquitectura para cada caso de uso.

Implementación: Se crea el código que se ajuste a la arquitectura y que tenga el comportamiento deseado. Se prueba además cada componente individualmente, integrándolos en un sistema ejecutable.

En esta dimensión se utilizan además diagrama de componentes y de despliegue para comprender cómo se organizan y dependen unos de otros.

Pruebas: Se realizan las pruebas del sistema para asegurar que el comportamiento es el requerido.

Distribución: Hacer todo lo necesario para la salida del proyecto.

1.2.1.3.2. Disciplinas de Apoyo

- Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- Administrando el proyecto: Administrando horarios y recursos.
- Entorno: Administrando el ambiente de desarrollo.

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierten luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Los elementos del RUP son:

- **Actividades**, Son los procesos que se llegan a determinar en cada iteración.
- **Trabajadores**, Vienen a ser las personas o entes involucrados en cada proceso.
- **Artefactos**, Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.⁰

1.2.1.4. UNIFIED MODELING LANGUAGE (UML)

UML, por sus siglas en inglés, Unified Modeling Language, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad para definir un sistema, para especificar o para describir métodos o procesos y para construir y documentar un sistema.

1.2.1.4.1. Diagramas

En UML 2.0 hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, a veces es útil categorizarlos jerárquicamente, como se muestra en la Figura 1.3.

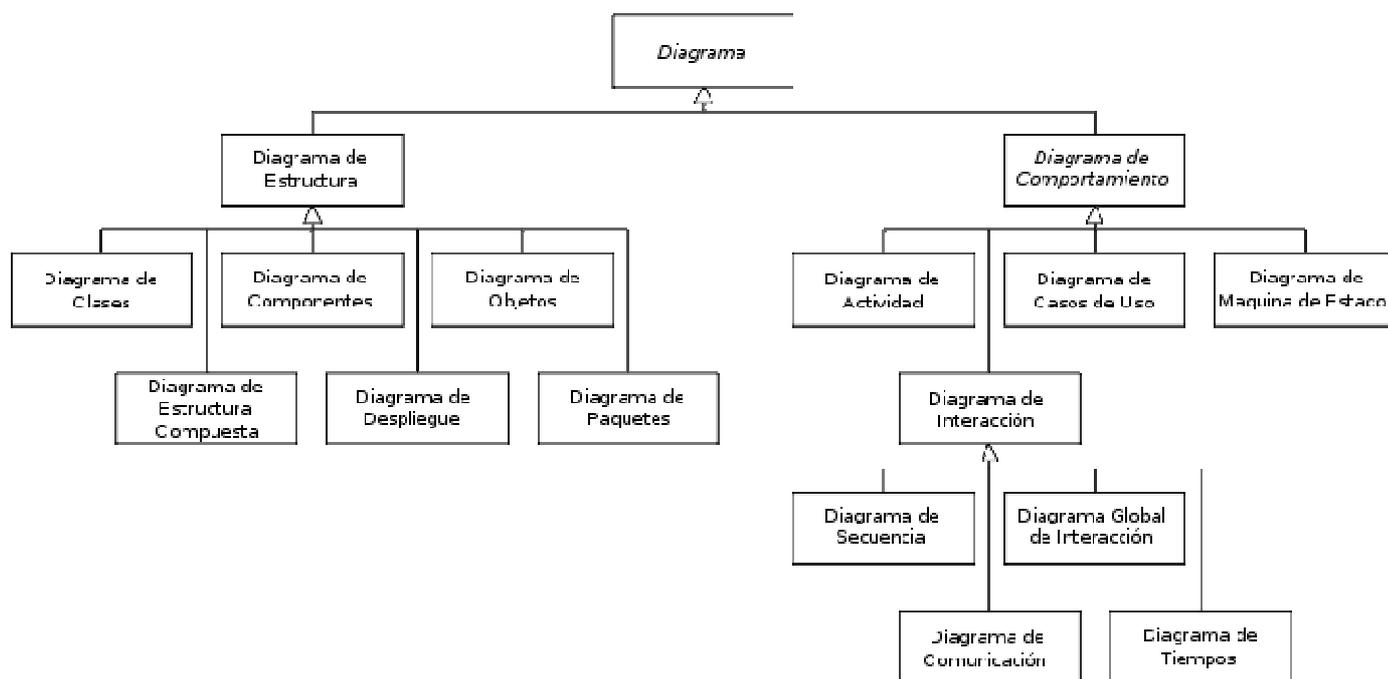


Figura 1.3 Diagramas UML

A continuación se describen brevemente los diagramas que se utilizan dentro de las disciplinas de desarrollo definidas por RUP.

Diagrama de Casos de Uso

Un diagrama de casos de uso es una especie de diagrama de comportamiento, UML no define estándares para que el formato escrito describa los casos de uso, sin embargo una notación gráfica puede solo dar una vista general simple de un caso de uso o un conjunto de casos de uso.

Los diagramas de casos de uso son a menudo confundidos con los casos de uso. Mientras los dos conceptos están relacionados, los casos de uso son mucho más detallados que los diagramas de casos de uso.

Diagrama de Clases

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

Diagrama de Secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.

Diagrama de Componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

Diagrama de Despliegue

El Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

Los elementos usados por este tipo de diagrama son nodos (representados como un prisma), componentes (representados como una caja rectangular con dos protuberancias del lado izquierdo) y asociaciones.

1.2.2. EXTREME PROGRAMING (XP)

La programación extrema o Extreme Programming fue formulado por Kent Beck en el año de 1999 basando su nueva metodología en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es más que aplicar una pura lógica.

XP es el más destacado de los procesos ágiles de desarrollo de software que prefiere adaptarse a los cambios de requerimientos en lugar de tener la previsibilidad que tanto cuesta en el proceso de planificación, los seguidores de XP defienden que los cambios sobre la marcha son un aspecto natural e inevitable y que todo gerente de proyecto debe ser capaz de resolver.

1.2.2.1. Características de XP

La metodología se basa en:

1.2.2.1.1. Desarrollo iterativo e incremental

Se comienza con un sistema con la principal funcionalidad y se va añadiendo una a una las funcionalidades restantes.

1.2.2.1.2. Pruebas Unitarias

Se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.

1.2.2.1.3. Re-fabricación

Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

1.2.2.1.4. Programación en Parejas

Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

1.2.2.1.5. Integración Del Equipo De Programación Con El Cliente

Es recomendable que el grupo de programación tenga contacto directo y continuo con el cliente.

1.2.2.1.6. Corrección de todos los errores

Se debe corregir todos los errores antes de agregar funcionalidades.

1.2.2.1.7. Refactorización del código

Si es necesario reescribir código para hacerlo más legible o mejorar su funcionalidad.

1.2.2.1.8. Propiedad del código compartida

Si creas una funcionalidad que puede ser utilizada por otras personas dentro del proyecto, compártela.

1.2.2.1.9. Simplicidad

Las cosas simples son fáciles de desarrollar, probar y mantener, de ser necesario se irá incrementando su complejidad si el cliente lo ve necesario.

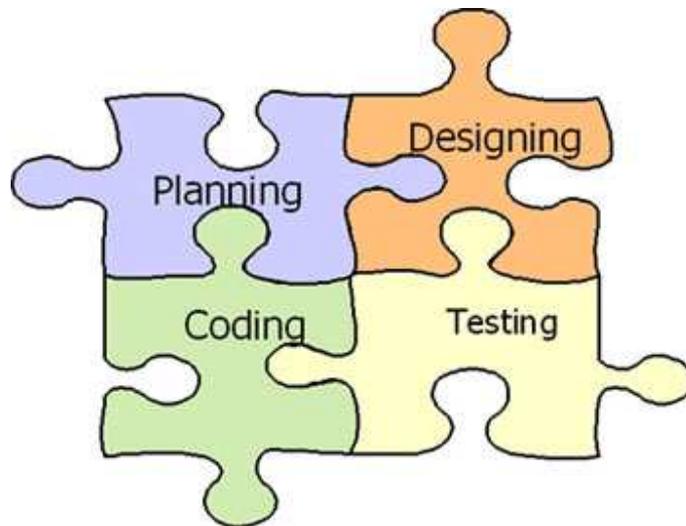


Figura 1.4 Metodología Extreme Programing

1.2.2.2. Los Cinco Valores de XP

1.2.2.2.1. Comunicación

XP fomenta la comunicación entre los miembros del grupo del desarrollo como con el cliente, una buena comunicación permite un desarrollo sincronizado entre el pequeño grupo que suele conformar el equipo de trabajo en XP, una mala comunicación al contrario permite una desorganización, discusión, mala comprensión de requerimientos e incluso enfados entre las partes.

1.2.2.2.2. Sencillez

El sistema debe ser funcional sin ningún añadido como colores, sonidos u otros que no estén dentro de los requerimientos, si al terminar toda la funcionalidad principal hay tiempo suficiente se puede aplicar cualquier otro requerimiento secundario.

1.2.2.2.3. Retroalimentación

El sistema al ser funcional y teniendo la suficiente comunicación con el cliente se puede lograr que la información que se necesite, sea el mismo sistema quien la responda, es una manera de que el cliente pruebe un sistema que está en desarrollo dando un índice de fiabilidad.

1.2.2.2.4. Valentía

XP nos advierte que existirán cambios durante el desarrollo, y el impacto que esto significa es proporcional a que tan cambiante sea el código del sistema, esto nos obliga a tomar decisiones que pueden ser cruciales para el proyecto.

1.2.2.2.5. Respeto

El respeto a cada integrante del grupo pues todos son importantes para la culminación del proyecto. El respeto en XP se refiere a comprender que el trabajo de los integrantes está ligado entre si y que cualquier retraso puede acarrear al atraso de otras personas.

1.2.2.3. Fases de la Programación Extrema

En el caso de la programación extrema se identifican diferentes roles: un equipo de gestión o de diseño, uno de desarrollo y los clientes finales. La relación entre los equipos de diseño y desarrollo son distintas a lo definido en las metodologías tradicionales.

1.2.2.3.1. Fase 1: Planificación del proyecto

El primer paso para cualquier proyecto que siga XP, son las *Historias de Usuario*, éstas tienen la misma finalidad que los casos de uso pero con algunas diferencias: Son escritas por el cliente en 3 o 4 líneas utilizando un lenguaje no técnico, sin detalles y sin hacer referencias a diseños o algoritmos para la codificación.

Las Historias de Usuario son usadas para estimar tiempos y también en la fase de pruebas para verificar que el sistema cumple con lo deseado.

Luego de definir las historias, es preciso elaborar un plan de publicaciones o *Release Plan*, que indicará las historias de usuario que se crearán para cada versión del programa y las fechas en las que se publicarán estas versiones.

Todo proyecto que utilice XP se ha de dividir en *Iteraciones* de 3 semanas de duración, al inicio de cada iteración se definirá el Release Planing y al final se determinarán las Historias de Usuario que no pasaron las pruebas de aceptación.

La *Velocidad del Proyecto* es la medida de la rapidez con que se desarrolla el proyecto; la estimación es sencilla, basta con contar el número de historias de usuario que se pueden implementar por una iteración.

XP como metodología recomienda la *Programación en Pareja*, pues incrementa la productividad y la calidad del software desarrollado, este trabajo involucra a dos programadores en el mismo equipo.

Es necesario que los programadores mantengan *Reuniones Diarias* para que se expongan sus problemas, soluciones e ideas de forma conjunta, las reuniones tienen que ser fluidas y todo el mundo tiene que tener voz y voto.

1.2.2.3.2. Fase 2: Diseño

La metodología XP sugiere que se manejen *Diseños Simples* que procuren hacer todo lo menos complicado posible para conseguir un diseño entendible y fácil de implementar.

Usar *Glosarios de Términos* y una correcta especificación de nombres de clases y métodos ayudará a comprender el diseño y facilitará una futura reutilización de código. No se debe añadir *Funcionalidad Extra* al programa aunque se piense en un futuro que ésta será utilizada, esto implica un desperdicio de tiempo y recursos.

Refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Es muy común usar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos.

1.2.2.3.3. Fase 3: Codificación

El cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de XP. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas.

Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen las pruebas que verifiquen que la historia implementada cumple la funcionalidad especificada.

La optimización del código siempre se debe dejar para el final. Hay que hacer que funcione y que sea correcto, más tarde se puede optimizar. XP afirma que la mayoría de los proyectos que necesiten más tiempo extra que el planificado para ser finalizados no podrán ser terminados a tiempo se plantea la solución de realizar un

nuevo "Release plan" para concretar los nuevos tiempos de publicación y de velocidad del proyecto.

1.2.2.3.4. Fase 4: Pruebas

Uno de los pilares de la metodología XP es el uso de pruebas para comprobar el funcionamiento de los códigos que vayamos implementando. Las pruebas permiten verificar que un cambio en la estructura de un código no tiene por qué cambiar su funcionamiento.

Además las pruebas sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario. Para asegurar el funcionamiento final de una determinada historia de usuario se deben crear "Test de aceptación"; estos test son creados y usados por los clientes para comprobar que las distintas historias de usuario cumplen su cometido.

1.2.3. MICROSOFT SOLUTION FRAMEWORK (MSF⁶)

MSF es un framework flexible e interrelacionado con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos, que con una notación y un proceso de desarrollo se lo puede aplicar como metodología⁷.

⁶ MSF.- Microsoft Solution Framework.

⁷ Basado en <http://arevalomaria.wordpress.com/2010/01/07/msf-microsoft-solution-framework/>



Figura 1.5 Metodología MSF⁸

1.2.3.1. Fases de MSF

El MSF consiste cinco fases las cuales tienen definido su objetivo, las actividades, su documentación y su marca final o punto de control que debe ser aprobada por el cliente para avanzar a la siguiente fase.⁹

1.2.3.1.1. Visión

Se reúne un equipo del proyecto. Define la visión y el ámbito de una solución que cumplirá los objetivos del cliente. El equipo organiza entonces el proyecto y proporciona un documento de visión/ámbito aprobado. Los clústeres de funciones de administración de productos y administración de programas toman el mando en esta fase.

⁸ Fuente: <http://vchamorro334.blogspot.es/tags/MSF/>

⁹ Tomado de: <http://technet.microsoft.com/es-es/library/bb490157.aspx>

1.2.3.1.2. Planeación

Se desarrollan los procesos de diseño conceptual, lógico y físico, así como la especificación funcional. El clúster de funciones de administración de programas crea planes de proyecto que tratan el desarrollo, la comunicación y otras tareas; y cada función proporciona los datos para crear la programación del proyecto. El clúster de funciones de administración de programas toma el mando durante esta fase.

1.2.3.1.3. Desarrollo

El equipo crea y prueba la solución. El clúster de funciones de desarrollo toma el mando durante esta fase.

1.2.3.1.4. Estabilización

El equipo crea la solución piloto en preparación para el lanzamiento de producción. El clúster de funciones de prueba toma el mando durante esta fase.

1.2.3.1.5. Implementación

El equipo implementa la solución en todos sitios y comprueba que es estable y utilizable. La responsabilidad pasa entonces a los equipos de operaciones de TI y soporte. El clúster de funciones de administración de lanzamientos toma el mando durante esta fase.

1.2.3.2. Características de MSF

1.2.3.2.1. Adaptable

Es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.

1.2.3.2.2. Escalable

Puede organizar equipos pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.

1.2.3.2.3. Flexible

Es utilizada en el ambiente de desarrollo de cualquier cliente.

1.2.3.2.4. Tecnología Agnóstica

Puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

1.2.3.3. Modelos de MSF

1.2.3.3.1. Modelo de Arquitectura del Proyecto

MSF es diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.

1.2.3.3.2. Modelo de Equipo

Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.

1.2.3.3.3. Modelo de Proceso

Este modelo se ha diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de equipo.

1.2.3.3.4. Modelo de Gestión del Riesgo

MSF está diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.

1.2.3.3.5. Modelo de Diseño del Proceso

MSF está diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.

1.2.3.3.6. Modelo de Aplicación

Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores.

1.2.4. ANALISIS COMPARATIVO DE LAS METODOLOGIAS DE DESARROLLO

1.2.4.1. Consideraciones para el análisis comparativo de las metodologías de desarrollo con RUP con UML.

RUP y UML a pesar de no ser inicialmente diseñados para el desarrollo de aplicaciones web es una de las metodologías más aceptadas en la gestión y desarrollo de proyectos de este tipo, se considera a favor de esta metodología la integración de extensiones como WAE (Web Application Extension) en el modelamiento de elementos web, OOHDM (Object Oriented Hypermedia Design Method) que permite definir los aspectos de navegabilidad-presentación y W2000 en la definición de navegabilidad que han fortalecido el diseño web que carecía inicialmente UML.

1.2.4.2. Consideraciones para XP en el análisis comparativo de las metodologías de desarrollo

La metodología XP se adapta al desarrollo de aplicaciones web y mantiene características deseables para el proyecto actual como la programación en pareja y poco deseables como la necesidad del cliente durante el ciclo de vida del desarrollo, poco apoyo en la gestión de proyectos y un mínimo manejo en el levantamiento de requerimientos.

1.2.4.3. Consideraciones para MSF en el análisis comparativo de las metodologías de desarrollo.

Los puntos fuertes a considerar en MSF son la gestión de proyectos, gestión de Riesgos y la facilidad en el desarrollo de aplicaciones web, pero se toma en cuenta la poca experiencia en las metodologías basadas en MSF como MSF4ASD (MSF for Agile Software Development).

Para facilitar la elección de la metodología listamos un conjunto de características que servirán para comparar las ventajas y desventajas de RUP con UML, XP y MSF.

Metodología Característica	RUP con UML	XP	MSF
Desarrollo de aplicaciones web.	Alto	Alto	Alto
Interacción del cliente con el equipo de desarrollo.	Medio	Alto	Medio
Gestión de proyectos.	Alto	Medio	Alto
Definición de requerimientos.	Alto	Bajo	Alto
Facilidad para desarrollos de corta duración.	Alto	Alto	Medio
Uso de herramientas libres.	Alto	Alto	Bajo
Experiencia del equipo de desarrollo.	Media	Baja	Baja

Tabla 1.1 Análisis Comparativo de las Metodologías de Desarrollo

Si bien la Tabla 1.1 muestra la comparativa entre metodologías, hay que considerar si las características son o no deseables dependiendo de la situación del proyecto, para aclarar más el panorama se describe cada característica de la Tabla 1.1 tomando como referencia el proyecto a desarrollar.

1.2.4.3.1. Desarrollo de aplicaciones web.

Al tratarse el proyecto de una aplicación web, es deseable el apoyo de la metodología durante el desarrollo de proyectos de este tipo.

1.2.4.3.2. Interacción del cliente con el equipo de desarrollo.

El proyecto BRIGAMED se va a implementar para el Área 2 de Salud de Pastaza, razón por la cual nuestro cliente labora y reside en esta misma provincia, este es el

motivo de la necesidad de una metodología en la cual la participación del cliente sea baja.

1.2.4.3.3. Gestión de proyectos.

Si bien las metodologías de software pueden carecer de la gestión de proyectos, es una característica que es deseable para llevar el correcto control del tiempo y alcance planteados.

1.2.4.3.4. Definición de requerimientos.

El proyecto BRIGAMED se va a implementar para el Área 2 de Salud de Pastaza, razón por la cual nuestro cliente labora y reside en esta misma provincia, este es el motivo de la necesidad de una metodología que recopile los requerimientos de forma clara y completa en el menor número de reuniones.

1.2.4.3.5. Facilidad para desarrollos de corta duración.

Al ser un proyecto corto, es deseable que la metodología facilite este tipo de desarrollos.

1.2.4.3.6. Uso de herramientas libres.

Una de las peticiones de Área 2 de Salud de Pastaza fue que el proyecto esté apegado al decreto 1014, el cual dispone el uso del software libre en las instituciones públicas.

1.2.4.3.7. Experiencia del equipo de desarrollo.

Si fuera el caso de implementar el proyecto con una metodología desconocida por el equipo de desarrollo, los recursos que se necesitarían serían mayores que si se hiciera con una metodología conocida.

Para poder justificar la elección de la metodología, en la Tabla 1.2 damos una puntuación tomando como referencia cada característica descrita y la Tabla 1.1, tomando en cuenta que los rangos para la calificación son los siguientes:

Alto: 10 -7 **Medio:** 6-4 **Bajo:** 3-0

Metodología Característica	RUP con UML	XP	MSF
Desarrollo de aplicaciones web	10	10	10
Interacción del cliente con el equipo de desarrollo	5	0	5
Gestión de proyectos	10	3	10
Definición de requerimientos	10	0	10
Facilidad para cambio de requerimientos	5	10	5
Facilidad para desarrollos de corta duración	8	10	5
Uso de herramientas libres	10	10	0
Experiencia del equipo de desarrollo	5	0	0
Puntaje	63	43	45

Tabla 1.2 Análisis Comparativo de las Metodologías de Desarrollo con respecto al proyecto

1.2.5. JUSTIFICACIÓN DE LA METODOLOGIA DE DESARROLLO

Considerando la situación actual, como el análisis de las metodologías hemos llegado a la conclusión que RUP con UML es la mejor alternativa que se acopla a las

necesidades de nuestro proyecto, descartando XP principalmente por no tener el apoyo constante del cliente debido a la distancia geográfica, MSF por apoyarse en herramientas privativas y la experiencia del equipo de trabajo en desarrollo con la metodología RUP con UML.

1.3. SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO

1.3.1. FRAMEWORK DE DESARROLLO

1.3.1.1. JBoss SEAM

JBoss Seam es un framework desarrollado por JBoss, una división de Red Hat, que combina las dos tecnologías Enterprise JavaBeans EJB¹⁰ y JavaServerFaces JSF¹¹; por lo tanto el lenguaje de programación al cual está orientado completamente es Java.

JBoss Seam ofrece una solución completa para el desarrollo web que abarca todas las capas, desde la de presentación hasta la de integración, posibilitando el acceso a cualquier componente EJB^[8].

El desarrollo WYSIWYG¹² es facilitado a través del uso de las JBoss Tools, que es un conjunto de plug-ins diseñados para el entorno integrado de desarrollo Eclipse. Seam puede ser integrado con las bibliotecas de componentes JSF JBoss RichFaces o con ICEsoft ICEFaces; ambas bibliotecas poseen soporte para AJAX¹³.

¹⁰ EJB.- Enterprise Java Beans

¹¹ JSF.- Java Server Faces

¹² WYSIWYG.- What You See Is What You Get

¹³ AJAX.- Asynchronous JavaScript and XML

Es posible generar automáticamente una aplicación web de CRUD¹⁴ a partir de una base de datos existente el plug-in que se integra a Eclipse con la instalación del JBoss Tools.

1.3.1.2. GWT ¹⁵

GWT es un framework creado por Google que permite ocultar la complejidad de varios aspectos de la tecnología AJAX. Es compatible con varios navegadores, lo cual es importante ya que cada navegador necesita código específico para lograr una correcta presentación en una aplicación web.

GWT está orientado a crear aplicaciones AJAX con lenguaje Java usando casi cualquier entorno de desarrollo. Además que proporciona un conjunto de herramientas que permiten desarrollar funcionalidades Javascript siendo necesario la programación para obtener un CRUD^[9].

GWT se encarga únicamente de la capa de presentación para desarrollar sobre la capa de integración es necesario añadir varios plug-ins de código abierto al framework.

1.3.1.3. Symfony

Symfony es un framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web.

Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más

¹⁴ CRUD.- Acrónimo de Crear, Obtener, Actualizar, Borrar (Create, Retrieve, Update, Delete)

¹⁵ GWT.- Google Web Toolkit

comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony está desarrollado completamente en PHP¹⁶ y es comúnmente utilizado en sitios web de comercio electrónico de primer nivel^[10].

La generación de un CRUD en Symfony es posible agregando una herramienta externa al framework o plug-in llamado Propel.

1.3.1.4. Análisis Comparativo del Framework de Desarrollo

Para facilitar la elección del framework, listamos un conjunto de características que servirán para comparar las ventajas y desventajas de JBoss SEAM, GWT y Symfony.

Criterio	JBoss SEAM	GWT	Symfony
Lenguajes de desarrollo	Java	Java	Java y PHP
Generación de CRUD	Integrada	No Integrada	No Integrada
Desarrollo en capas	Si	No	Si
Experiencia del equipo de desarrollo	Alta	Media	Baja

Tabla 1.3 Análisis comparativo del Framework de Desarrollo

La Tabla 1.3 muestra la comparativa entre frameworks, hay que considerar si las características son o no deseables dependiendo de la situación del proyecto, para aclarar más el panorama se describe cada característica de la Tabla 1.3 tomando como referencia el proyecto a desarrollar.

1.3.1.4.1. Lenguaje de Desarrollo.

Al analizar los lenguajes del framework de desarrollo consideramos la experiencia en el lenguaje.

¹⁶ PHP.- PHP Hyertext Pre-processor

1.3.1.4.2. Generación de CRUD.

El proyecto a desarrollar tiene gran cantidad de información que será almacenada en diferentes módulos, cada módulo necesitará algunos CRUDS y por esto la necesidad de disponer de un framework que facilite su creación.

1.3.1.4.3. Desarrollo en Capas.

Para facilitar el desarrollo se prefiere un diseño y por ende un framework que facilite el desarrollo por capas.

1.3.1.4.4. Experiencia del Equipo de Desarrollo.

Si fuera el caso de implementar el proyecto con un framework desconocido por el equipo de desarrollo, los recursos que se necesitarían serían mayores que si se hiciera con una framework conocido.

Para poder justificar la elección del framework, en la Tabla 1.4 damos una puntuación tomando como referencia cada característica descrita y la Tabla 1.3, tomando en cuenta que los rangos para la calificación son los siguientes:

Alto: 10 -7 **Medio:** 6-4 **Bajo:** 3-0

Criterio	JBoss SEAM	GWT	Symfony
Experiencia en el lenguaje del framework.	10	10	5
Generación de CRUD.	10	0	0
Desarrollo en capas.	10	0	10
Experiencia del equipo de desarrollo en el framework.	10	5	0
Puntaje	40	15	15

Tabla 1.4 Análisis Comparativo del Framework de Desarrollo con respecto al proyecto

1.3.2. GESTOR DE BASE DE DATOS (DBMS)¹⁷

Los sistemas de gestión de base de datos son un tipo de software que se dedica a ser la interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta estructurado.^[5]

1.3.2.1. PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD¹⁸.

La licencia BSD es una licencia de software libre permisiva la cual tiene menos restricciones en comparación con otras como la GNU-GPL¹⁹ estando muy cercana al dominio público.

Mediante un sistema denominado MVCC²⁰ (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

El consumo de recursos de PostgreSQL está determinado por un parámetro llamado `shared_buffers` el cual nos indica cuanta memoria está dedicada al motor de base de datos; por defecto este valor es bajo considerando las plataformas en las cuales se encuentra instalado varía del 15% al 25% de la memoria RAM²¹.

¹⁷ DBMS.- Database Management System

¹⁸ BSD.- Berkeley Software Distribution

¹⁹ GPL.- General Public License

²⁰ MVCC.- Multivesion Concurrency Control

²¹ RAM.- Random Access Memory

1.3.2.2. MySQL

MySQL es un sistema de gestión de base de datos relacional, multi-hilo y multiusuario desarrollado como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la licencia GNU-GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso.

MySQL ha demostrado ser uno de los Sistemas de Gestión de Base de Datos más estables ya que posee en el mundo más de 6 millones de instalaciones.

Para la instalación del motor de base de datos de MySQL es necesario reservar al menos el 30% de la Memoria RAM de un equipo para el correcto funcionamiento de las librerías que lo soportan

1.3.2.3. Análisis Comparativo de los DBMS's

Criterio	PostgreSQL		MySQL	
Licencia	BSD	5	GPL / Comercial	5
Estabilidad	Alta	10	Medio	7
Consumo de Recursos	Bajo	10	Bajo	10
Experiencia del equipo de desarrollo	Alta	10	Media	5
Puntaje		35		27

Tabla 1.5 Análisis comparativo de los DBMS's

1.3.3. JUSTIFICACIÓN DE LAS HERRAMIENTAS DE DESARROLLO

Considerando los cuadros comparativos de los Frameworks y los DBMS's para el desarrollo del sistema Web, se ha seleccionado como gestor de base de datos a PostgreSQL, debido a que es uno de los más utilizados en sistemas y aplicaciones Web, la facilidad del uso, la estabilidad de su licenciamiento y el conocimiento por parte de los desarrolladores.

Como framework de desarrollo se ha seleccionado JBoss Seam considerando el lenguaje de desarrollo que utiliza, la facilidad que presenta para la generación de CRUD's, la orientación al desarrollo en capas y sobre todo la experiencia del equipo de desarrollo en el mismo.

Cabe acotar que para el desarrollo con JBoss Seam, el IDE²² más apto es Eclipse, debido a que éste posee una completa gama de herramientas para el funcionamiento de JBoss Seam.

La principal característica para la toma de estas decisiones se debe también al requerimiento por parte del cliente para adoptar el Decreto 1014 firmado por el Gobierno del Ecuador en el año 2008, que dispone a las empresas públicas al uso de herramientas de Software Libre.

²² IDE.- Integrated Development Environment

CAPÍTULO II. ANÁLISIS Y DISEÑO DEL SISTEMA WEB

En esta sección se desarrollará el documento de Especificación de Requerimientos de Software cuya plantilla se detalla en los anexos.

2.1. ESPECIFICACIÓN DE REQUERIMIENTOS

2.1.1. INTRODUCCIÓN

2.1.1.1. Propósito

El propósito de este documento es definir los requerimientos del sistema para la Gestión de Brigadas Médicas del Área 2 de Salud de Pastaza.

2.1.1.2. Ámbito

El sistema de Gestión de Brigadas Médicas, se encargará de la administración de información de las brigadas, proceso importante para el desempeño de las actividades de la institución. Esta información será manipulada por el personal administrativo, médicos y odontólogos. En consecuencia todos los datos que se manejen en el sistema a desarrollarse serán relacionados con las personas de las comunidades a visitar en cada una de las brigadas y todo lo referente a la parte médica.

2.1.1.3. Definiciones, Acrónimos y Abreviaciones

Para las definiciones, acrónimos y abreviaciones se hace referencia al Glosario de términos.

2.1.1.4. Referencias

Para las referencias del documento de Especificación de Requerimientos de Software se hace referencia a la Bibliografía.

2.1.2. DESCRIPCIÓN GENERAL

2.1.2.1. Especificación de Funcionalidades

La funcionalidad del sistema será registrar la información del Área 2 de Salud de Pastaza que permita la ejecución de las tareas para la realización de una brigada médica.

En el sistema se podrá almacenar información para la realización de las brigadas médicas como: Unidades Operativas, Recurso Humano, Comunidades, Familias, Personas, Insumos Médicos, Medicamentos, Vacunas y Equipamiento Médico del Área 2 de Salud de Pastaza.

El sistema permitirá almacenar los datos de las atenciones realizadas en las brigadas médicas como el diagnóstico por persona en fichas de brigadas. Además de mostrar en pantalla el índice de la tasa de crecimiento poblacional desde el año 2009 en adelante por cada una de las comunidades, el cálculo se lo realizará de acuerdo a la siguiente fórmula:

$$\text{Índice de Crecimiento} = \frac{\text{Población final} - \text{Población inicial}}{\text{Población inicial}} \times 100\%$$

Esta información será administrada por el personal administrativo, médicos y odontólogos del Área 2 de Salud de Pastaza.

2.1.2.2. Supuestos y Dependencias

No hay una disponibilidad de subsistemas o componentes de software que sirvan como base para este proyecto.

2.1.2.3. Acuerdos con el Cliente para la Administración de Requerimientos

Los cambios en los requerimientos serán aceptados hasta que el equipo de desarrollo culmine con el diseño del sistema. Luego de este tiempo cualquier necesidad por parte del cliente no será tomada en cuenta para el desarrollo del sistema.

En caso de que los cambios en los requerimientos se encuentren dentro del plazo límite éstos serán medidos de acuerdo a la cantidad de horas/hombre que éstos requieran y se dará a conocer al cliente el nuevo plazo que tomará el análisis y diseño del sistema.

2.1.3. ESPECIFICACIÓN DE REQUERIMIENTOS

2.1.3.1. Reportes de Casos de Uso

En el reporte de casos de uso, se definen la mayoría de los requerimientos funcionales del sistema.

Módulo	Caso de uso
Administrar usuario	Crear usuario

	Buscar usuario
	Modificar usuario
	Eliminar usuario
Administrar unidad operativa	Crear unidad operativa
	Buscar unidad operativa
	Modificar Unidad Operativa
	Eliminar unidad operativa
Administrar comunidad	Crear comunidad
	Buscar comunidad
	Modificar comunidad
	Eliminar comunidad
	Mostrar índices de crecimiento
Administrar familia	Crear familia
	Buscar familia
	Modificar familia
	Eliminar familia
Administrar persona	Crear persona
	Buscar persona
	Modificar persona
	Eliminar Persona
Administrar recurso humano	Crear recurso humano
	Buscar recurso humano
	Modificar recurso humano
Administrar suministro	Crear suministro
	Buscar suministro
	Modificar Suministro
	Eliminar suministro
Administrar brigada	Crear brigada
	Buscar brigada

	Modificar brigada
--	-------------------

Tabla 2.1 Módulos y Requerimientos del Sistema

2.1.3.2. Requerimientos Funcionales

- 1.- Administración de Usuarios
- 2.- Administración de Unidades Operativas
- 3.- Administración de Comunidades
- 4.- Administración de Recursos Humanos
- 5.- Administración de Suministros
- 6.- Administración de Familias
- 7.- Administración de Personas
- 8.- Administración de Brigadas

2.1.3.2.1. Funcionalidad N° 1: Administración de Usuarios

DESCRIPCIÓN	
El sistema permitirá la administración de la información de los usuarios del sistema.	
NÚMERO	REQUERIMIENTO
Req-01	El acceso al sistema será por medio de un usuario y contraseña.
Req-02	El sistema permitirá <i>Crear</i> un nuevo usuario definiendo el nombre de usuario, la contraseña, el email, el nombre completo y el perfil.
Req-03	El sistema permitirá <i>Modificar</i> la información del usuario.
Req-04	El sistema permitirá <i>Buscar y Listar</i> a todos los usuarios registrados en el sistema mediante el nombre de usuario, una vez ejecutado este

	requerimiento, el sistema deberá permitir un enlace con los requerimientos Req-03 y Req-05
Req-05	El sistema permitirá <i>Eliminar</i> los usuarios registrados.

Tabla 2.2 Requerimientos de Administración de Usuarios

2.1.3.2.2. Funcionalidad N° 2: Administración de Unidades Operativas

DESCRIPCIÓN	
El sistema permitirá la administración de la información de las unidades operativas del Área 2 de Salud de Pastaza.	
CÓDIGO	REQUERIMIENTO
Req-06	El sistema permitirá <i>Crear</i> un nuevo registro de una unidad operativa, se especificará el nombre de la unidad operativa y el responsable de la misma.
Req-07	El sistema permitirá <i>Modificar</i> la información de una unidad operativa.
Req-08	El sistema permitirá <i>Buscar y Listar</i> a todas las unidades operativas registradas en el sistema mediante el nombre, una vez ejecutado este requerimiento, el sistema deberá permitir un enlace con los requerimientos Req-07 y Req-09.
Req-09	El sistema permitirá <i>Eliminar</i> las unidades operativas registradas.

Tabla 2.3 Requerimientos de Administración de Unidades Operativas

2.1.3.2.3. Funcionalidad N° 3: Administración de Comunidades

DESCRIPCIÓN	
El sistema permitirá la administración de la información de las comunidades del Área 2 de Salud de Pastaza.	
CÓDIGO	REQUERIMIENTO
Req-10	El sistema permitirá <i>Crear</i> un nuevo registro de una comunidad, se

	especificará la unidad operativa a la cual pertenece la comunidad, nombre de la comunidad, distancia de la comunidad a la unidad operativa más cercana, extensión territorial, tiempo de llegada desde la unidad operativa más cercana, tipo de acceso a la comunidad, riesgo de la comunidad, medios de comunicación de la comunidad.
Req-11	El sistema permitirá <i>Modificar</i> la información de una comunidad a excepción del nombre de la comunidad.
Req-12	El sistema permitirá <i>Buscar y Listar</i> a todas las comunidades registradas en el sistema mediante el nombre de la comunidad, una vez ejecutado este requerimiento, el sistema deberá permitir un enlace con los requerimientos Req-11 y Req-13.
Req-13	El sistema permitirá <i>Eliminar</i> las comunidades registradas.
Req-14	El sistema mostrará el índice de crecimiento poblacional de las comunidades.

Tabla 2.4 Requerimientos de Administración de Comunidades

2.1.3.2.4. Funcionalidad N° 4: Administración de Recursos Humanos

DESCRIPCIÓN	
El sistema permitirá la gestión de la información del recurso humano del Área 2 de Salud de Pastaza	
CÓDIGO	REQUERIMIENTO
Req-15	El sistema permitirá <i>Crear</i> un nuevo registro de un recurso humano, se especificará el nombre, número de cédula, género, fecha de nacimiento, teléfono, correo electrónico, domicilio y la fecha de ingreso al MSP, cargo, estado y especialidad del recurso humano.
Req-16	El sistema permitirá <i>Modificar</i> la información del recurso humano a excepción del número de cédula y el nombre.
Req-17	El sistema permitirá <i>Buscar y Listar</i> a todos los recursos humanos registrados en el sistema mediante el número de cédula o el nombre,

	una vez ejecutado este requerimiento, el sistema deberá permitir un enlace con los requerimientos Req-15.
--	-----------------------------------------------------------------------------------------------------------

Tabla 2.5 Requerimientos de Gestión de Recursos Humanos

2.1.3.2.5. Funcionalidad N° 5: Administración de Suministros

DESCRIPCIÓN	
El sistema permitirá la administración de la información de los suministros que se manejan en el Área 2 de Salud de Pastaza.	
CÓDIGO	REQUERIMIENTO
Req-18	El sistema permitirá <i>Crear</i> un nuevo registro de un suministro, se especificará el nombre, el tipo, la dosis, la presentación y el responsable.
Req-19	El sistema permitirá <i>Modificar</i> la información de un suministro.
Req-20	El sistema permitirá <i>Buscar y Listar</i> a todos los suministros registrados en el sistema mediante el nombre, una vez ejecutado este requerimiento, el sistema deberá permitir un enlace con los requerimientos Req-18 y Req-20.
Req-21	El sistema permitirá <i>Eliminar</i> los suministros registrados.

Tabla 2.6 Requerimientos de Administración de Suministros

2.1.3.2.6. Funcionalidad N° 6: Administración de Familias

DESCRIPCIÓN	
El sistema permitirá la administración de la información de las familias del Área 2 de Salud de Pastaza.	
CÓDIGO	REQUERIMIENTO
Req-22	El sistema permitirá <i>Crear</i> un nuevo registro de una familia, se especificará el apellido, los riesgos biológicos, los riesgos sanitarios,

	los riesgos socio-económicos, los servicios básicos y el compromiso.
Req-23	El sistema permitirá <i>Modificar</i> la información de una familia.
Req-24	El sistema permitirá <i>Buscar y Listar</i> a todas las familias registradas en el sistema mediante el apellido, una vez ejecutado este requerimiento, el sistema deberá permitir un enlace con los requerimientos Req-22 y Req-24.
Req-25	El sistema permitirá <i>Eliminar</i> las familias registradas.

Tabla 2.7 Requerimientos de Administración de Familias

2.1.3.2.7. Funcionalidad N° 7: Administración de Personas

DESCRIPCIÓN	
El sistema permitirá la administración de la información de las personas pertenecientes a una familia.	
CÓDIGO	REQUERIMIENTO
Req-26	El sistema permitirá <i>Crear</i> un nuevo registro de una persona, se especificará el nombre, el número de cédula, la fecha de nacimiento, la nacionalidad, el género, la escolaridad, el parentesco con el jefe de familia y la familia a la que pertenece.
Req-27	El sistema permitirá <i>Modificar</i> los datos de la persona, a excepción del número de cédula.
Req-28	El sistema permitirá <i>Buscar y Listar</i> a las personas registradas en el sistema mediante el número de cédula o el nombre, una vez ejecutado este requerimiento, el sistema deberá permitir un enlace con los requerimientos Req-26 y Req-28.
Req-29	El sistema permitirá <i>Eliminar</i> las personas registradas.

Tabla 2.8 Requerimientos de Administración de Personas

2.1.3.2.8. Funcionalidad N° 8: Administración de Brigadas

DESCRIPCIÓN

El sistema permitirá la gestión de la información de las brigadas en el Área 2 de Salud de Pastaza.	
CÓDIGO	REQUERIMIENTO
Req-30	El sistema permitirá <i>Crear</i> un nuevo registro de brigada especificando el nombre, la fecha inicial, la fecha final, las comunidades, el recurso humano, los medicamentos, insumos, equipo médico y vacunas asignadas a la brigada.
Req-31	El sistema permitirá <i>Modificar</i> la información de una brigada al término de la misma donde se registrará únicamente el diagnóstico de las personas atendidas en la brigada.
Req-32	El sistema permitirá <i>Buscar y Listar</i> las brigadas registradas en el sistema mediante la fecha de inicio y el nombre, una vez ejecutado este requerimiento, el sistema deberá permitir un enlace con el requerimiento Req-30.

Tabla 2.9 Requerimientos de Gestión de Brigadas

2.1.3.3. Requerimientos de Interfaces

2.1.3.3.1. Interfaz Principal

Cada una de las funcionalidades especificadas en el tema anterior, deberán estar listadas en la interfaz principal.

Para facilidad de los usuarios cada funcionalidad se mostrará en forma de vínculos, que serán identificados con un nombre que describa su uso en el sistema que llevarán a cada una de las pantallas de manejo del sistema, dependiendo del vínculo seleccionado por el usuario.

El sistema permitirá realizar una autenticación con el nombre de usuario y contraseña, además de mostrar un vínculo para cerrar la sesión del usuario y salir del sistema.

Todas las interfaces serán desarrolladas basándose en el diseño que se muestra en la Figura 2.1.

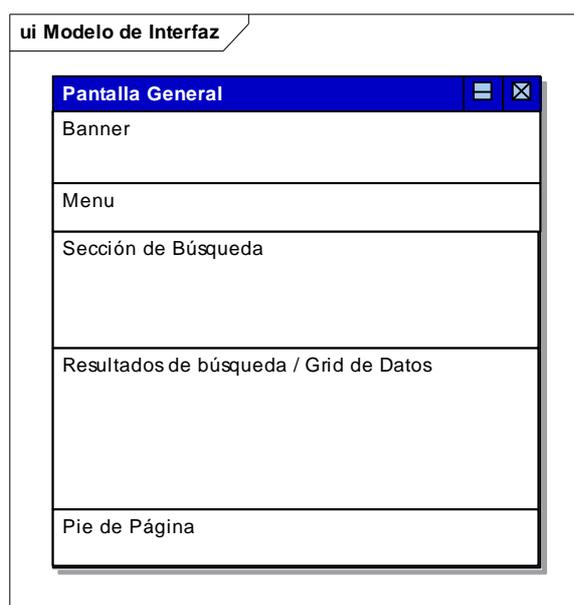


Figura 2.11 Modelo de interfaces

2.1.3.3.2. *Interfaz de Administración de Usuarios*

En esta funcionalidad el sistema permitirá la creación de un nuevo usuario, solicitando los siguientes datos: nombre de usuario que es el nombre con el que ingresa al sistema, contraseña de usuario, correo electrónico del usuario, nombre completo de la persona y perfil de usuario

Se podrá realizar la búsqueda de usuarios considerando el nombre de usuario.

Luego de elegir el usuario a ser actualizado se podrán modificar todos los campos excepto el campo nombre de usuario que se encontrará bloqueado.

2.1.3.3.3. *Interfaz de Administración de Unidades Operativas*

En esta funcionalidad el sistema permitirá la creación de una nueva unidad operativa, solicitando los siguientes datos: nombre de la unidad operativa, responsable de la unidad operativa y el área de salud a la cual pertenece, la cual podrá ser seleccionada de una lista desplegable.

Se podrá realizar la búsqueda de la comunidad considerando el nombre de la unidad operativa, el resultado se presentará en una tabla donde se visualizarán las unidades operativas que coincidan con el nombre ingresado.

Para realizar una actualización se deberá elegir la unidad a modificar, en la cual se podrán realizar cambios a todos los campos.

2.1.3.3.4. Interfaz de Administración de Comunidades

En esta funcionalidad el sistema permitirá la creación de una nueva comunidad, solicitando los siguientes datos: nombre de la comunidad, distancia de la comunidad a la unidad operativa más cercana, extensión territorial de la comunidad que será expresadas en km², ubicación geográfica, medios de comunicación de los cuales dispone la comunidad para contactarse con el exterior, estas opciones se podrán seleccionar de una lista desplegable, riesgo de la comunidad, tiempo de llegada desde la unidad operativa más cercana, acceso fluvial a la comunidad, existen varios tipos de acceso fluvial los cuales se podrán seleccionar de una lista desplegable, acceso terrestre a la comunidad, existen varios tipos de acceso terrestre los cuales se podrán seleccionar de una lista desplegable, tipo de acceso a la comunidad y la unidad operativa a la que pertenece la comunidad la cual se podrá seleccionar de una lista desplegable.

Se podrá realizar la búsqueda de la comunidad considerando el nombre de la comunidad, el resultado se presentará en una tabla donde se visualizarán las comunidades que coincidan con el nombre ingresado.

Para realizar una actualización se deberá elegir la comunidad a modificar, en la cual se podrán realizar cambios a todos los campos.

Se podrá visualizar el listado de las comunidades registradas en el sistema con su respectiva población histórica y el índice de crecimiento poblacional para cada una de ellas.

2.1.3.3.5. Interfaz de Administración de Recursos Humanos

En esta funcionalidad el sistema permitirá la creación de un nuevo recurso humano, solicitando los siguientes datos: nombre, cédula, género, fecha de nacimiento, dirección, teléfono, correo electrónico, fecha de ingreso al MSP, estado, cargo, especialidad y tipo de contrato del recurso humano. Los campos de género, estado, cargo, especialidad y tipo de contrato son textos que deberán ser escogidos de una lista desplegable donde estarán todos los datos de su respectivo tipo.

Se podrá realizar la búsqueda del recurso humano considerando su nombre y cédula, el resultado se presentará en una tabla donde se visualizarán los recursos humanos que coincidan con el nombre ingresado.

Para realizar una actualización se deberá primero elegir el recurso humano a modificar, en la cual se podrá realizar la actualización de todos los campos excepto el nombre y cédula del recurso humano que estarán bloqueados para evitar cualquier cambio de identidad.

2.1.3.3.6. Interfaz de Administración de Suministros

En esta funcionalidad el sistema permitirá la creación de un nuevo suministro, solicitando los siguientes datos: el nombre, el tipo de suministro, en caso de ser un medicamento se seleccionará la presentación y se ingresará la fuente, en el caso de ser un insumo se definirá la presentación y en caso de tratarse de una vacuna, se

definirá la dosis. El tipo del suministro será escogido de una lista desplegable donde estarán todos los tipos de suministros, la presentación será seleccionada de una lista desplegable donde estarán todas las presentaciones, tanto de medicamentos como de insumos.

Se podrá realizar la búsqueda del suministro considerando su nombre, el resultado se presentará en una tabla donde se visualizarán los suministros que coincidan con el nombre ingresado.

Para realizar una actualización se deberá primero elegir el suministro a modificar, en el cual se podrá realizar la actualización de todos los campos excepto el código de identificación del suministro que estará bloqueado.

2.1.3.3.7. Interfaz de Administrar Familias

En esta funcionalidad el sistema permitirá la creación de una nueva familia, solicitando los siguientes datos: el apellido de la familia, servicios básicos, compromiso de la familia, los riesgos biológicos, sanitarios y socioeconómicos y la comunidad a la cual pertenece. Los riesgos biológicos, sanitarios y socioeconómicos son datos fijos de la familia que ya están reconocidos por el Área 2 de Salud de Pastaza y que deberán ser calificados en su respectivo campo.

Se podrá realizar la búsqueda de la familia considerando su apellido, el resultado se presentará en una tabla donde se visualizarán las familias que coincidan con el apellido ingresado.

Para realizar una actualización se deberá primero elegir la familia a modificar, en la cual se podrá realizar la actualización de todos los campos excepto el código de identificación de la familia que estará bloqueado para evitar cualquier cambio.

2.1.3.3.8. Interfaz de Administrar Personas

En esta funcionalidad el sistema permitirá la creación de una nueva persona, solicitando los siguientes datos: cédula, nombre, fecha de nacimiento, nacionalidad, género el cual se podrá seleccionar de una lista desplegable, escolaridad la cual se podrá seleccionar de una lista desplegable, enfermedades de impacto, parentesco con el jefe de familia el cual se podrá seleccionar de una lista desplegable y la fecha de la salida de la comunidad.

Se podrá realizar la búsqueda de la persona considerando el nombre de la persona, el resultado se presentará en una tabla donde se visualizarán las personas que coincidan con el nombre ingresado.

Para realizar una actualización se deberá elegir la persona a modificar, en la cual se podrán realizar cambios a todos los campos.

2.1.3.3.9. Interfaz de Administrar Brigadas

En esta funcionalidad el sistema permitirá la creación de una nueva brigada, para realizar su ingreso el usuario deberá seleccionar de un listado de unidades operativa para luego escoger las comunidades que recibirán la atención médica.

Seleccionamos de un listado los recursos humanos, medicamentos, vacunas, insumos y equipos médicos que serán utilizados en la brigada, todos estos campos serán visualizados en pestañas individuales para facilitar al usuario el ingreso de los datos.

Se podrá realizar la búsqueda de la brigada considerando su fecha de inicio, el resultado se presentará en una tabla donde se visualizarán las brigadas que coincidan con la fecha ingresada.

2.1.3.3.10. Interfaz de Modificar Brigadas

En esta funcionalidad el sistema permitirá actualizar la información de la brigada médica, como datos de la familia, de la persona y el diagnóstico después de haber recibido la atención médica.

Para realizar la actualización deberá seleccionar de un listado una de las comunidades visitadas, con esto se desplegarán las familias que viven en la comunidad, al escoger una familia se presentarán sus datos y las personas que conforman esa familia.

Se permitirá la modificación de los datos tanto de la familia como de la persona, de la familia se podrá modificar el apellido, jefe de familia, servicios básicos, compromiso y todos los riesgos que se describen a continuación:

Riesgos biológicos

1. Personas con vacunación incompleta
2. Personas con mala nutrición (sobre peso, desnutrición)
3. Personas con enfermedades de impacto
4. Embarazadas con problemas
5. Personas con discapacidad
6. Personas con problemas mentales

Riesgos sanitarios

7. Consumo de agua insegura
8. Mala eliminación de basura y excreta
9. Mala eliminación de desechos líquidos
10. Impacto geológico por industrias
11. Animales intra-domiciliarios

Riesgos socioeconómicos

12. Pobreza
13. Desempleo o empleo informal del jefe de familia
14. Analfabetismo del padre o de la madre
15. Desestructuración familiar
16. Violencia, alcoholismo y/o drogadicción
17. Malas condiciones de la vivienda
18. Hacinamiento

De la persona se podrá modificar solo su diagnóstico.

2.1.3.4. Requerimientos no Funcionales

Para describir la operatividad del sistema a continuación se detallan los requerimientos no funcionales.

2.1.3.4.1. Disponibilidad

El sistema se encontrará disponible los días de trabajo del Área 2 de Salud de Pastaza, ya que se instalará en los equipos de la misma y la información que se manejará solo deberá ser modificada los días laborables.

2.1.3.4.2. Estabilidad

El sistema brindará estabilidad en su uso ya que ha sido diseñado para que varios usuarios interactúen al mismo tiempo sin que se produzcan bloqueos. Esta estabilidad está garantizada también por el motor de base de datos que proporciona concurrencia de varios usuarios sin problemas.

2.1.3.4.3. Portabilidad

BRIGAMED se desarrollará en un lenguaje de programación multiplataforma y se diseñará completamente en herramientas de software libre lo cual facilitará su compatibilidad con cualquier sistema operativo, además el modelo de la base de datos se estructurará de una manera tal que puede ser rediseñado por cualquier herramienta case.

2.1.3.4.4. Rendimiento

El sistema brindará un servicio óptimo, ya que ha sido diseñado para que funcione en un ambiente web sólo dentro de la intranet del Área 2 de Salud. El diseño de la red interna permitirá buenos tiempos de respuesta que hacen que el sistema pueda ser aprovechado al máximo por parte de los usuarios.

2.1.3.4.5. Mantenimiento y Escalabilidad

BRIGAMED está diseñado pensando en la posibilidad de crecimiento del sistema, por ejemplo, es fácilmente acoplable para agregar un módulo de información

geográfica de las comunidades y el manejo de índices de crecimiento poblacional y enfermedades frecuentes en las comunidades que se encuentran dentro del Área 2 de Salud.

BRIGAMED se desarrollará bajo estrictos estándares de programación que facilitarán su mantenimiento.

2.1.3.5. Requerimientos Técnicos

Como requerimientos técnicos se describe las características de software que deberán ser parte del ambiente del sistema.

Software	Versión	Descripción
JBoss	5.1	Servidor de aplicaciones J2EE de código abierto implementado en Java.
Seam Framework	2.0	Framework de desarrollo y librerías implementadas en J2EE.
Sun Java JDK	1.6	Máquina virtual independiente del sistema operativo para leer el código compilado.
PostgreSQL	8.4	Motor de base de datos.
Internet Explorer	6 o más	Navegador web que permite la visualización del sistema.
Mozilla Firefox	2 o más	
Google Chrome	7 o más	
Windows XP	2002	Permite la administración eficaz de sus recursos.

Tabla 2.10 Requerimientos Técnicos

2.1.3.6. Requerimientos de Proceso

Para desarrollar el sistema nos referiremos a los procesos definidos por la metodología RUP, la cual divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones y en las que se hace un mayor hincapié en las distintas actividades. Para los requerimientos de procesos, esta tesis se basa en la descripción de la metodología RUP descrita en el primer capítulo.

2.2. ANÁLISIS

2.2.1. CASOS DE USO

2.2.1.1. Diagrama de Casos de Uso

Para el desarrollo del Sistema de Brigadas Médicas del Área 2 de Salud de Pastaza se han definido los siguientes paquetes.

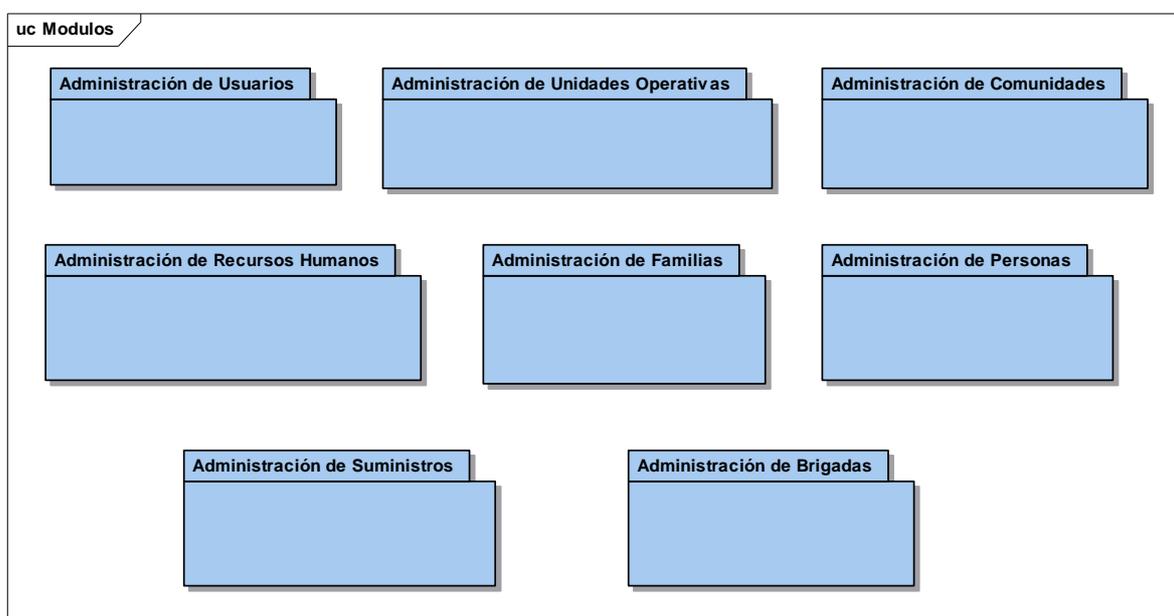


Figura 2.22 Paquetes para el Sistema de Gestión de Brigadas

El paquete de Administración de Usuarios describe el siguiente modelo de Casos de Uso.

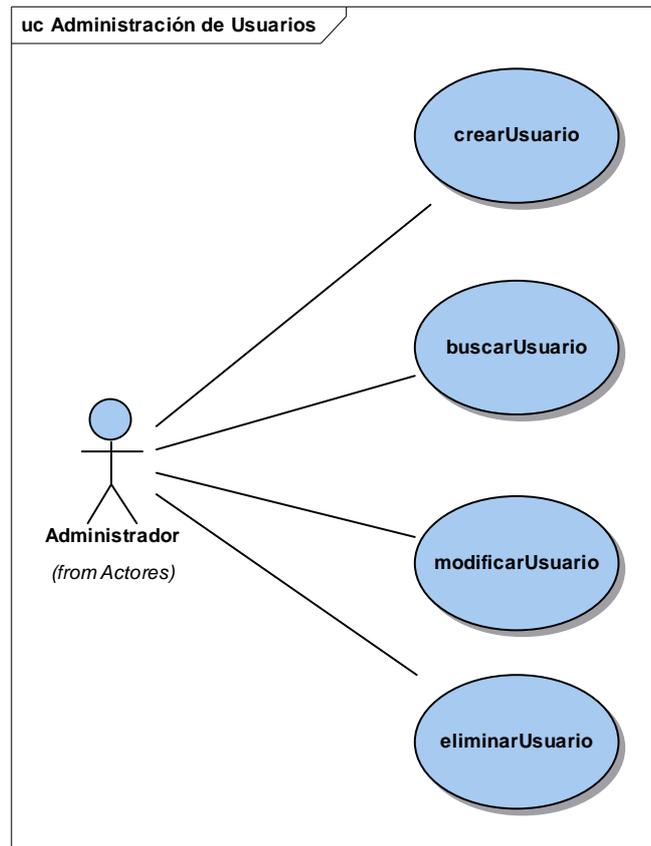


Figura 2.23 Paquete Administración de Usuarios

El paquete Administración de Unidades Operativas describe el siguiente modelo de Casos de Uso.

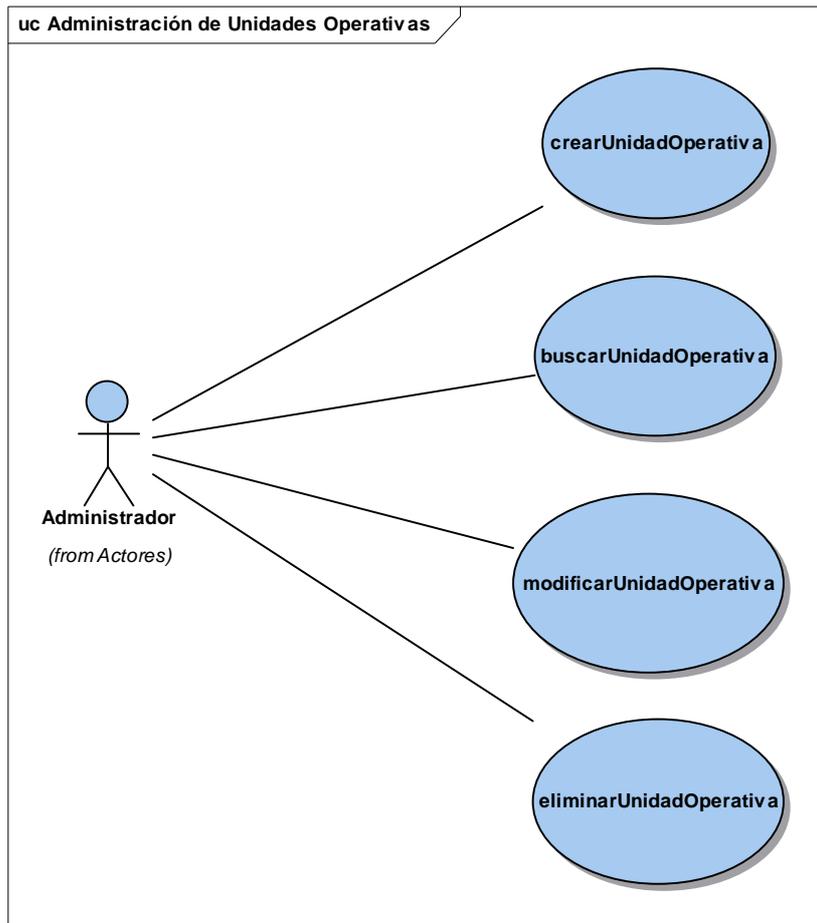


Figura 2.24 Paquete Administración de Unidades Operativas

El paquete Administración de Comunidades describe el siguiente modelo de Casos de Uso.

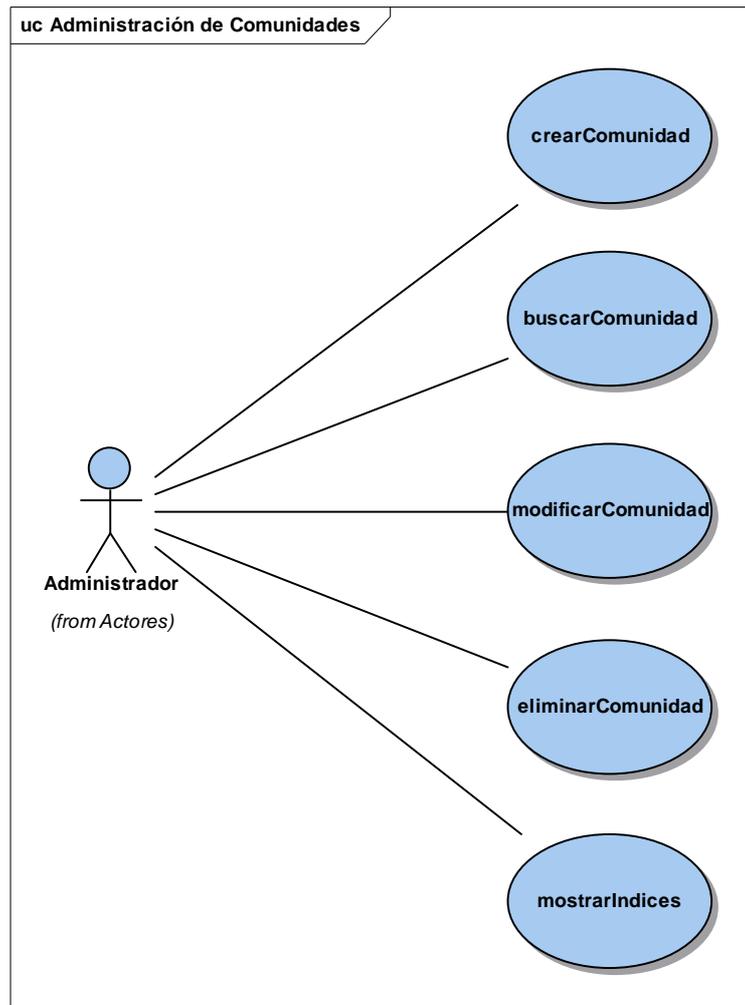


Figura 2.25 Paquete Administración de Comunidades

El paquete Administración de Familias describe el siguiente modelo de Casos de Uso.

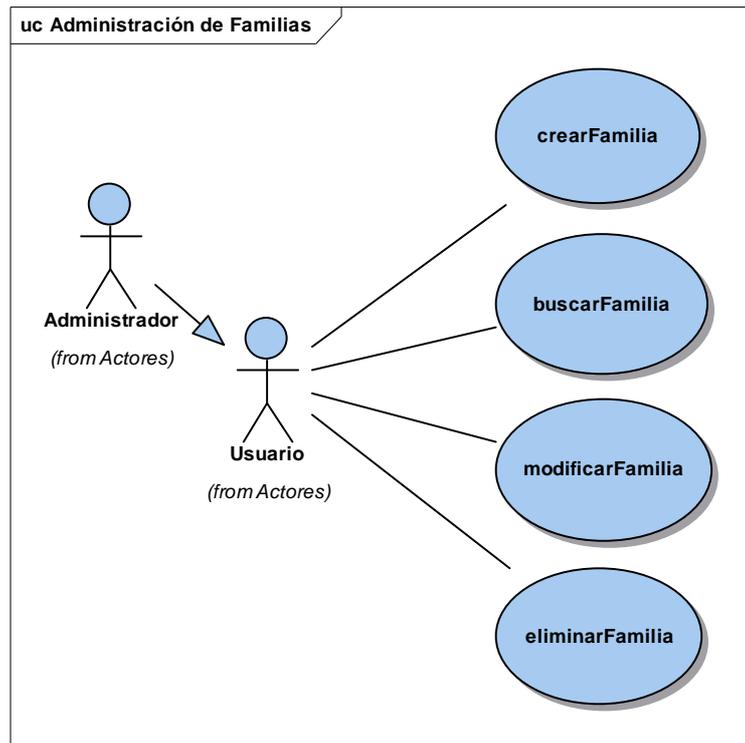


Figura 2.26 Paquete Administración de Familias

El paquete Administración de Personas describe el siguiente modelo de Casos de Uso.

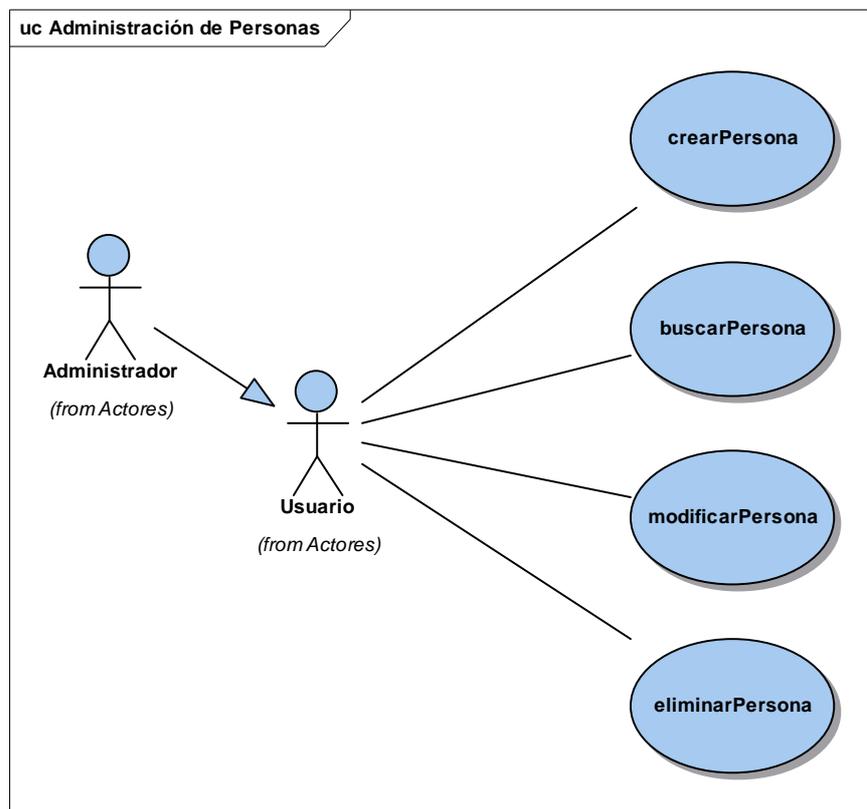


Figura 2.27 Paquete Administración de Personas

El paquete Administración de Recursos Humanos describe el siguiente modelo de Casos de Uso.

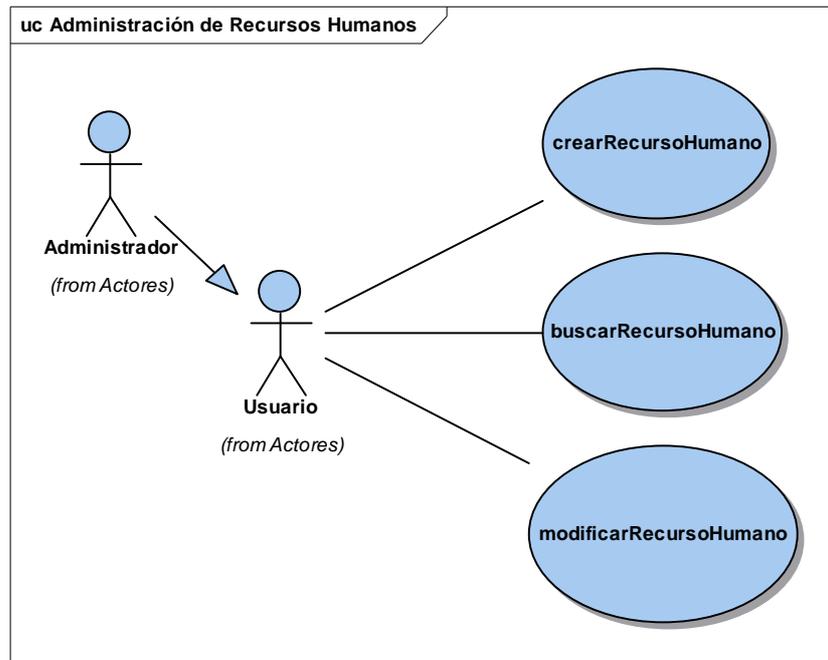


Figura 2.28 Paquete Administración de Recursos Humanos

El paquete Administración de Suministros describe el siguiente modelo de Casos de Uso.

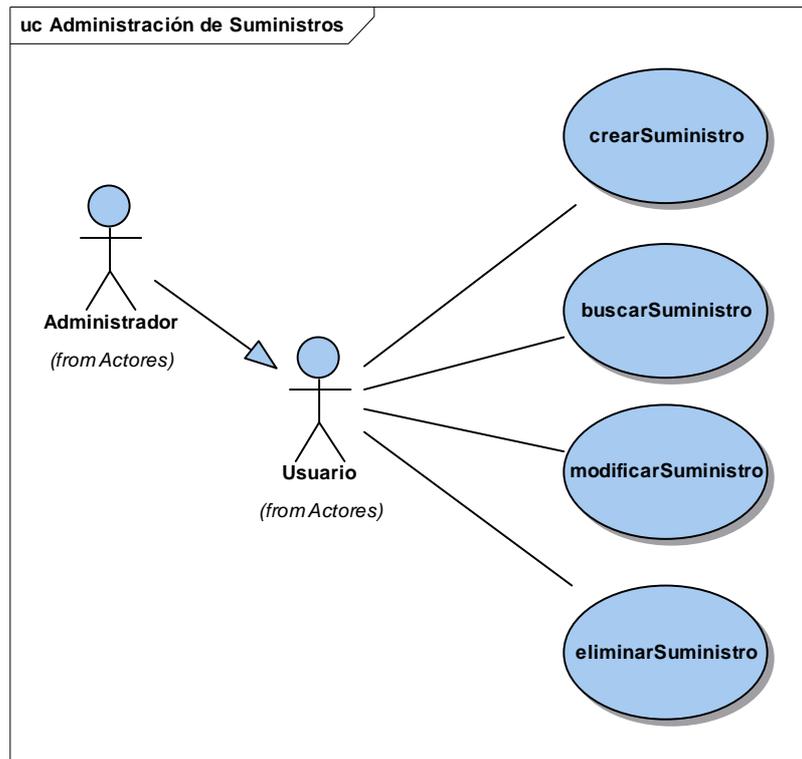


Figura 2.29 Paquete Administración de Suministros

El paquete Administración de Brigadas describe el siguiente modelo de Casos de Uso.

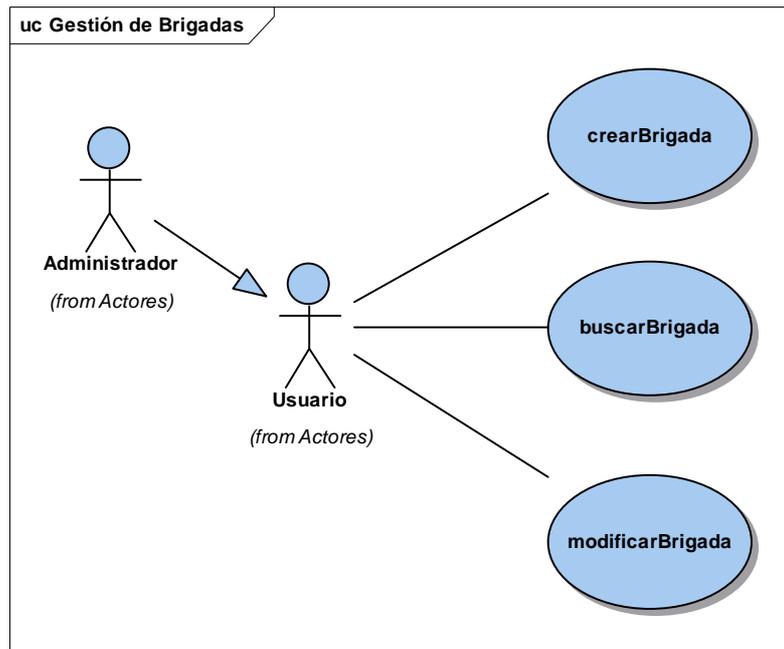


Figura 2.210 Paquete Administración de Brigadas

2.2.1.2. Detalle de casos de uso

2.2.1.2.1. Paquete Administración de Usuarios

crearUsuario

Nombre	crearUsuario	
Pre-condición	Existe un Usuario a ser creado.	
Post-condición	El Usuario queda creado en el sistema.	
Descripción	Realiza la creación de un Usuario, controlando la existencia del elemento en el sistema.	
Actor	Sistema	
Flujo Principal		
<ol style="list-style-type: none"> 1. Ingresar datos de Usuario: nombre, nombreUsuario, contraseña, email y seleccionar el perfil de una lista desplegable. 2. Ejecutar función Guardar. 	<ol style="list-style-type: none"> 3. Verificar que no exista el usuario a crear. 4. Si el usuario no existe, entonces se realiza la creación de Usuario. 	
Flujo Alternativo 1		
<ol style="list-style-type: none"> 1. Ingresar datos de Usuario: nombre, nombreUsuario, contraseña, email y seleccionar el perfil de una lista desplegable. 2. Ejecutar función Guardar. 	<ol style="list-style-type: none"> 3. Verificar que no exista el usuario a crear. 4. Si el usuario existe, entonces muestra un mensaje de error "Usuario ya existe". 	

Tabla 2.11 Caso de Uso crearUsuario

buscarUsuario

Nombre	buscarUsuario	
Pre-condición	Existe un Usuario a ser buscado en el sistema.	
Post-condición	El Usuario es encontrado, o no existía el Usuario en el sistema.	
Descripción	Realiza la búsqueda de un Usuario en el sistema, controla que exista el Usuario en el sistema.	
	Actor	Sistema
Flujo Principal		
1. Ingresar nombreUsuario. 2. Ejecuta la función Buscar		3. Realiza la búsqueda según el nombreUsuario como criterio. 4. Si el usuario existe, entonces muestra los atributos del Usuario.
Flujo Alterno 1		
1. Ingresar nombreUsuario. 2. Ejecuta la función Buscar		3. Realiza la búsqueda según el nombreUsuario como criterio. 4. Si el usuario no existe, entonces muestra mensaje de error "Usuario no existe".

Tabla 2.12 Caso de Uso buscarUsuario

modificarUsuario

Nombre	modificarUsuario	
Pre-condición	Existen atributos de un Usuario que deben ser modificados.	
Post-condición	El Usuario modificado queda registrado en el sistema, o el Usuario a modificar no existía en el sistema.	
Descripción	Realiza la modificación de los atributos de un Usuario.	
	Actor	Sistema
Flujo Principal		
1. Ingresar nombreUsuario. 2. Ejecuta la función Buscar.		3. Realiza la búsqueda del usuario. 4. Si el usuario existe, entonces muestra los atributos del usuario.
5. Modifica los atributos del Usuario, excepto el código. 6. Ejecuta la función Guardar.		7. Realiza la actualización del Usuario.
Flujo Alternativo 1		
1. Ingresar nombreUsuario 2. Ejecuta la función Buscar		3. Realiza la búsqueda del usuario. 4. Si el usuario no existe, entonces muestra un mensaje de error "Usuario no existe".

Tabla 2.13 Caso de Uso modificarUsuario

eliminarUsuario

Nombre	eliminarUsuario
Pre-condición	Existe un Usuario a ser eliminado.

Post-condición	El Usuario fue eliminado del sistema, o el Usuario a eliminar no existía en el sistema.	
Descripción	Realiza la eliminación de un Usuario, controlando la existencia del elemento en el sistema.	
Actor		Sistema
Flujo Principal		
1. Ingresar nombreUsuario. 2. Ejecuta la función Buscar. 5. Ejecuta la función Eliminar.	3. Realiza la búsqueda del usuario. 4. Si el usuario existe, entonces muestra los atributos del usuario. 6. Elimina el usuario.	
Flujo Alternativo 1		
1. Ingresar nombreUsuario. 2. Ejecuta la función Buscar.	3. Realiza la búsqueda del usuario. 4. Si el usuario no existe, entonces muestra un mensaje de error "Usuario no existe".	

Tabla 2.14 Caso de Uso eliminarUsuario

2.2.1.2.2. Paquete Administración de Unidades Operativas

crearUnidadOperativa

Nombre	crearUnidadOperativa	
Pre-condición	Existe una Unidad Operativa a ser creada.	
Post-condición	La Unidad Operativa queda creada en el sistema.	
Descripción	Realiza la creación de una Unidad Operativa, controlando la existencia del elemento en el sistema.	
Actor		Sistema
Flujo Principal		
1. Ingresar datos de Unidad Operativa: nombre de la unidad operativa y el nombre del responsable. 2. Ejecuta función Guardar.	3. Verifica que no exista la unidad	

	operativa a crear. 4. Si la Unidad Operativa no existe, entonces se realiza la creación de la Unidad Operativa.
Flujo Alterno 1	
1. Ingresar datos de Unidad Operativa: nombre de la unidad operativa y el nombre del responsable. 2. Ejecuta función Guardar.	3. Verifica que no exista la unidad operativa a crear. 4. Si la Unidad Operativa existe, entonces muestra mensaje de error "Unidad Operativa ya existe".

Tabla 2.15 Caso de Uso crearUnidadOperativa

buscarUnidadOperativa

Nombre	buscarUnidadOperativa	
Pre-condición	Existe una Unidad Operativa a ser buscada en el sistema.	
Post-condición	La Unidad Operativa es encontrada, o no existía la Unidad Operativa en el sistema.	
Descripción	Realiza la búsqueda de una Unidad Operativa en el sistema, controla que exista la Unidad Operativa en el sistema.	
Actor	Sistema	
Flujo Principal		
1. Ingresar nombreUnidadOperativa. 2. Ejecuta la función Buscar	3. Realiza la búsqueda según el nombreUnidadOperativa como criterio. 4. Si la unidad operativa existe,	

	entonces muestra los atributos de la Unidad Operativa.
Flujo Alternativo 1	
1. Ingresar nombreUnidadOperativa. 2. Ejecutar la función Buscar	3. Realiza la búsqueda según el nombreUnidadOperativa como criterio. 4. Si la unidad operativa no existe, entonces muestra mensaje de error "Unidad Operativa no existe".

Tabla 2.16 Caso de Uso buscarUnidadOperativa

modificarUnidadOperativa

Nombre	modificarUnidadOperativa	
Pre-condición	Existen atributos de una Unidad Operativa que deben ser modificados.	
Post-condición	La Unidad Operativa modificada queda registrada en el sistema, o la Unidad Operativa a modificar no existía en el sistema.	
Descripción	Realiza la modificación de los atributos de una Unidad Operativa.	
Actor		Sistema
Flujo Principal		
1. Ingresar nombreUnidadOperativa. 2. Ejecutar la función Buscar.	3. Realiza la búsqueda de la unidad operativa. 4. Si la unidad operativa existe, entonces muestra los atributos de la unidad operativa.	
5. Modifica los atributos de la Unidad Operativa. 6. Ejecuta la función Guardar.	7. Realiza la actualización de la Unidad Operativa.	
Flujo Alternativo 1		

2.2.1.2.3. Paquete Administración de Comunidades

crearComunidad

Nombre	crearComunidad	
Pre-condición	Exista una Unidad Operativa para que la Comunidad a ser creada sea asignada a dicha Unidad Operativa.	
Post-condición	La comunidad se encuentra creada en el sistema.	
Descripción	Realiza la creación de una comunidad, controlando la existencia del elemento en el sistema.	
Actor		Sistema
Flujo Principal		
1. Ingresar datos de la comunidad: nombre de la comunidad, distancia de la comunidad a la unidad operativa, extensión territorial, ubicación geográfica, medios de comunicación, riesgo de la comunidad, tiempo de llegada, acceso fluvial a la comunidad, acceso terrestre a la comunidad, acceso aéreo a la comunidad y la unidad operativa a la que pertenece la comunidad. 2. Ejecutar función Guardar.	3. Verificar que no exista la comunidad a crear. 4. Si la comunidad no existe, entonces se realiza la creación de la Comunidad.	
Flujo Alternativo 1		
1. Ingresar datos de la comunidad: nombre de la comunidad, distancia de la comunidad a la unidad operativa, extensión territorial, ubicación geográfica, medios de comunicación, riesgo de la		

<p>comunidad, tiempo de llegada, acceso fluvial a la comunidad, acceso terrestre a la comunidad, acceso aéreo a la comunidad y la unidad operativa a la que pertenece la comunidad.</p> <p>2. Ejecuta función Guardar.</p>	<p>3. Verifica que no exista la comunidad a crear.</p> <p>4. Si la comunidad existe, entonces muestra mensaje de error "Comunidad ya existe".</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 2.19 Caso de Uso crearComunidad

buscarComunidad

Nombre	buscarComunidad	
Pre-condición	Existe una Comunidad a ser buscada en el sistema.	
Post-condición	La Comunidad es encontrada o no existía en el sistema.	
Descripción	Realiza la búsqueda de una Comunidad en el sistema, controla que exista la Comunidad en el sistema.	
Actor	Sistema	
Flujo Principal		
<p>1. Ingresar nombreComunidad.</p> <p>2. Ejecuta la función Buscar.</p>	<p>3. Realiza la búsqueda según el nombreComunidad como criterio.</p> <p>4. Si la comunidad existe, entonces muestra los atributos de la Comunidad.</p>	
Flujo Alternativo 1		
<p>1. Ingresar nombreComunidad.</p> <p>2. Ejecuta la función Buscar.</p>	<p>3. Realiza la búsqueda según el nombreComunidad como criterio.</p> <p>4. Si la comunidad no existe, entonces muestra mensaje de error "Comunidad no existe".</p>	

Tabla 2.20 Caso de Uso buscarComunidad

modificarComunidad

Nombre	modificarComunidad	
Pre-condición	Exista la Comunidad a ser modificada.	
Post-condición	La Comunidad modificada queda registrada en el sistema.	
Descripción	Realiza la modificación de los atributos de una Comunidad.	
	Actor	Sistema
Flujo Principal		
1. Ingresar nombreComunidad. 2. Ejecuta la función Buscar.	5. Modifica los atributos de la comunidad a excepción del código. 6. Ejecuta la función Guardar.	3. Realiza la búsqueda de la comunidad. 4. Si la comunidad existe, entonces muestra los atributos de la comunidad. 7. Realiza la actualización de la Comunidad.
Flujo Alternativo 1		
1. Ingresar nombreComunidad. 2. Ejecuta la función Buscar.		3. Realiza la búsqueda de la comunidad. 4. Si la comunidad no existe, entonces muestra un mensaje de error "Comunidad no existe".

Tabla 2.21 Caso de Uso modificarComunidad

eliminarComunidad

Nombre	eliminarComunidad
Pre-condición	Existe una Comunidad a ser eliminada.

Post-condición	La Comunidad fue eliminada del sistema.	
Descripción	Realiza la eliminación de una Comunidad controlando la existencia del elemento en el sistema.	
Actor		Sistema
Flujo Principal		
1. Ingresar nombreComunidad. 2. Ejecutar la función Buscar. 5. Ejecutar la función Eliminar.	3. Realiza la búsqueda de la comunidad. 4. Si la comunidad existe, entonces muestra los atributos de la comunidad. 6. Elimina la Comunidad.	
Flujo Alternativo 1		
1. Ingresar nombreComunidad. 2. Ejecutar la función Buscar.	3. Realiza la búsqueda de la comunidad. 4. Si la comunidad no existe, entonces muestra mensaje de error "Comunidad no existe".	

Tabla 2.22 Caso de Uso eliminarComunidad

mostrarIndices

Nombre	mostrarIndices	
Pre-condición	Existe una Comunidad.	
Post-condición	Los índices de crecimiento poblacional son mostrados.	
Descripción	Muestra los índices de crecimiento poblacional de cada una de las Comunidades.	
Actor		Sistema
Flujo Principal		
1. Ejecutar la función MostrarIndices.	2. Realiza la búsqueda de la comunidad. 3. Realiza la búsqueda de los datos poblacionales de cada comunidad.	

	4. Calcula y muestra los índices de crecimiento por cada comunidad.
--	---------------------------------------------------------------------

Tabla 2.23 Caso de Uso mostrarIndices

2.2.1.2.4. Paquete Administración de Familias

crearFamilia

Nombre	crearFamilia
Pre-condición	Exista una Comunidad para que la Familia a ser creada sea asignada a dicha comunidad.
Post-condición	La Familia se encuentra creada en el sistema.
Descripción	Realiza la creación de una familia, controlando la existencia del elemento en el sistema.
Actor	Sistema
Flujo Principal	
<p>1. Ingresar los datos de la familia: comunidad a la que pertenece la familia, apellido, compromiso, servicios básicos y las calificaciones para cada riesgo agrupados según su respectivo tipo:</p> <p>Riesgos biológicos</p> <ul style="list-style-type: none"> • Personas con vacunación incompleta • Personas con mala nutrición (Sobrepeso, desnutrición) • Personas con enfermedades de impacto • Embarazadas con problemas • Personas con discapacidad • Personas con problemas mentales 	

<p>Riesgos sanitarios</p> <ul style="list-style-type: none"> • Consumo de agua insegura • Mala eliminación de basura y excreta • Mala eliminación de desechos líquidos • Impacto geológico por industrias • Animales intra-domiciliarios <p>Riesgos socioeconómicos</p> <ul style="list-style-type: none"> • Pobreza • Desempleo o empleo informal del jefe de familia • Analfabetismo del padre o de la madre • Desestructuración familiar • Violencia, alcoholismo, drogadicción • Malas condiciones de la vivienda • Hacinamiento <p>2. Ejecuta función Guardar.</p>	<p>3. Verifica que no exista la familia a crear.</p> <p>4. Si la familia no existe, entonces se realiza la creación de la Familia.</p>
Flujo Alterno 1	
<p>3. Ingresar los datos de la familia: comunidad a la que pertenece la familia, apellido, compromiso, servicios básicos y las calificaciones para cada riesgo agrupados según su respectivo</p>	

<p>tipo:</p> <p>Riesgos biológicos</p> <ul style="list-style-type: none">• Personas con vacunación incompleta• Personas con mala nutrición (Sobrepeso, desnutrición)• Personas con enfermedades de impacto• Embarazadas con problemas• Personas con discapacidad• Personas con problemas mentales <p>Riesgos sanitarios</p> <ul style="list-style-type: none">• Consumo de agua insegura• Mala eliminación de basura y excreta• Mala eliminación de desechos líquidos• Impacto geológico por industrias• Animales intra-domiciliarios <p>Riesgos socioeconómicos</p> <ul style="list-style-type: none">• Pobreza• Desempleo o empleo informal del jefe de familia• Analfabetismo del padre o de la madre• Desestructuración familiar• Violencia, alcoholismo,	
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

<p>drogadicción</p> <ul style="list-style-type: none"> • Malas condiciones de la vivienda • Hacinamiento <p>4. Ejecuta función Guardar.</p>	<p>5. Verifica que no exista la familia a crear.</p> <p>6. Si la familia existe, entonces se muestra mensaje de error "Familia ya existe".</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 2.24 Caso de Uso crearFamilia

buscarFamilia

Nombre	buscarFamilia	
Pre-condición	Existe una Familia a ser buscada en el sistema.	
Post-condición	La Familia es encontrada o no existía en el sistema.	
Descripción	Realiza la búsqueda de una Familia en el sistema, controla que exista la Familia en el sistema.	
Actor	Sistema	
Flujo Principal		
<p>1. Ingresar apellidoFamilia.</p> <p>2. Ejecuta la función Buscar.</p>	<p>3. Realiza la búsqueda según el apellidoFamilia como criterio.</p> <p>4. Si la familia existe, entonces muestra los atributos de la Familia.</p>	
Flujo Alternativo 1		
<p>1. Ingresar apellidoFamilia.</p> <p>2. Ejecuta la función Buscar.</p>	<p>3. Realiza la búsqueda según el apellidoFamilia como criterio.</p> <p>4. Si la familia no existe, entonces muestra un mensaje de error "Familia no existe".</p>	

Tabla 2.25 Caso de Uso buscarFamilia

modificarFamilia

Nombre	modificarFamilia	
Pre-condición	Exista la Familia a ser modificada.	
Post-condición	La Familia modificada queda registrada en el sistema.	
Descripción	Realiza la modificación de los atributos de una Familia.	
	Actor	Sistema
Flujo Principal		
1. Ingresar apellidoFamilia. 2. Ejecuta la función Buscar.		3. Realiza la búsqueda de la familia. 4. Si la familia existe, entonces muestra los atributos de la familia.
5. Modifica los atributos de la familia a excepción del código. 6. Ejecuta la función Guardar.		7. Realiza la actualización de la Familia.
Flujo Alternativo 1		
1. Ingresar apellidoFamilia. 2. Ejecuta la función Buscar.		3. Realiza la búsqueda de la familia. 4. Si la familia no existe, entonces muestra mensaje de error "Familia no existe".

Tabla 2.26 Caso de Uso modificarFamilia

eliminarFamilia

Nombre	eliminarFamilia
---------------	-----------------

Pre-condición	Existe una Familia a ser eliminada.	
Post-condición	La Familia fue eliminada del sistema.	
Descripción	Realiza la eliminación de una Familia controlando la existencia del elemento en el sistema.	
Actor		Sistema
Flujo Principal		
1. Ingresar apellidoFamilia. 2. Ejecuta la función Buscar. 5. Ejecuta la función Eliminar.	3. Realiza la búsqueda de la familia. 4. Si la familia existe, entonces muestra los atributos de la familia. 6. Elimina la Familia.	
Flujo Alterno 1		
1. Ingresar apellidoFamilia. 2. Ejecuta la función Buscar.	3. Realiza la búsqueda de la familia. 4. Si la familia no existe, entonces muestra mensaje de error "Familia no existe".	

Tabla 2.27 Caso de Uso eliminarFamilia

2.2.1.2.5. Paquete Administración de Personas

crearPersona

Nombre	crearPersona	
Pre-condición	Exista Persona a ser creada. Exista la Familia a ser asignada a la Persona.	
Post-condición	La Persona se encuentra creada en el sistema.	
Descripción	Realiza la creación de una Persona, controlando la existencia del elemento en el sistema.	
Actor		Sistema
Flujo Principal		
1. Ingresar datos de la Persona: cedula Persona, nombre de la Persona, fecha de Nacimiento,		

<p>nacionalidad, genero, escolaridad de Persona, enfermedad Impacto, parentesco Jefe Familia, fecha de Salida, familia a la que pertenece.</p> <p>2. Ejecuta la función Guardar</p>	<p>3. Verifica que no exista la persona a crear.</p> <p>4. Si la persona no existe, entonces se realiza la creación de la Persona.</p>
Flujo Alternativo 1	
<p>1. Ingresar datos de la Persona: cedula Persona, nombre de la Persona, fecha de Nacimiento, nacionalidad, genero, escolaridad de Persona, enfermedad Impacto, parentesco Jefe Familia, fecha de Salida, familia a la que pertenece.</p> <p>2. Ejecuta la función Guardar</p>	<p>3. Verifica que no exista la persona a crear.</p> <p>4. Si la persona existe, entonces se muestra un mensaje de error "Persona ya existe".</p>

Tabla 2.28 Caso de Uso crearPersona

buscarPersona

Nombre	buscarPersona	
Pre-condición	Existe una Persona a ser buscada en el sistema.	
Post-condición	La Persona es encontrada o no existía en el sistema.	
Descripción	Realiza la búsqueda de una Persona en el sistema, controla que exista la Persona en el sistema.	
Actor		Sistema
Flujo Principal		
<p>1. Ingresar nombrePersona.</p> <p>2. Ejecuta la función Buscar.</p>	<p>3. Realiza la búsqueda según el nombrePersona como criterio.</p>	

	4. Si la persona existe, entonces muestra los atributos de la Persona.
Flujo Alternativo 1	
1. Ingresar nombrePersona. 2. Ejecutar la función Buscar.	3. Realiza la búsqueda según el nombrePersona como criterio. 4. Si la persona no existe, entonces muestra un mensaje de error "Persona no existe".

Tabla 2.29 Caso de Uso buscarPersona

modificarPersona

Nombre	modificarPersona	
Pre-condición	Exista la Persona a ser modificado.	
Post-condición	La Persona modificada queda registrada en el sistema.	
Descripción	Realiza la modificación de los atributos de una Persona.	
Actor	Sistema	
Flujo Principal		
1. Ingresar nombrePersona. 2. Ejecutar la función Buscar.		3. Realiza la búsqueda de la persona. 4. Si la persona existe, entonces muestra los atributos de la persona.
5. Modifica los atributos de la persona a excepción del código. 6. Ejecuta la función Guardar.		7. Realiza la actualización de la Persona.
Flujo Alternativo 1		
1. Ingresar nombrePersona. 2. Ejecuta la función Buscar.		3. Realiza la búsqueda de la persona. 4. Si la persona no existe, entonces

	muestra mensaje de error "Persona no existe".
--	-----------------------------------------------

Tabla 2.30 Caso de Uso modificarPersona

eliminarPersona

Nombre	eliminarPersona	
Pre-condición	Existe una Persona a ser eliminada.	
Post-condición	La Persona fue eliminada del sistema.	
Descripción	Realiza la eliminación de una Persona controlando la existencia del elemento en el sistema.	
	Actor	Sistema
Flujo Principal		
3. Ingresar nombrePersona. 4. Ejecuta la función Buscar. 6. Ejecuta la función Eliminar.	5. Realiza la búsqueda de la persona. 6. Si la persona existe, entonces muestra los atributos de la persona. 7. Elimina la Persona.	
Flujo Alternativo 1		
3. Ingresar nombrePersona. 4. Ejecuta la función Buscar.	5. Realiza la búsqueda de la persona. 6. Si la persona no existe, entonces muestra mensaje de error "Persona no existe".	

Tabla 2.31 Caso de Uso eliminarPersona

2.2.1.2.6. *Paquete Administración de Recursos Humanos***crearRecursoHumano**

Nombre	crearRecursoHumano
Pre-condición	Exista Recurso Humano a ser creado.

Post-condición	El Recurso Humano se encuentra creado en el sistema.	
Descripción	Realiza la creación de un Recurso Humano, controlando la existencia del elemento en el sistema.	
Actor	Sistema	
Flujo Principal		
<ol style="list-style-type: none"> 1. Ingresar datos del recurso humano: nombre, cédula, género, fecha de nacimiento, dirección, teléfono, correo electrónico, fecha de ingreso al MSP, estado, cargo, especialidad y tipo de contrato del recurso humano. 2. Ejecuta la función guardar. 	<ol style="list-style-type: none"> 3. Verifica que no exista el recurso humano a crear. 4. Si el recurso humano no existe, entonces se realiza la creación del Recurso Humano. 	
Flujo Alternativo 1		
<ol style="list-style-type: none"> 1. Ingresar datos del recurso humano: nombre, cédula, género, fecha de nacimiento, dirección, teléfono, correo electrónico, fecha de ingreso al MSP, estado, cargo, especialidad y tipo de contrato del recurso humano. 2. Ejecuta la función guardar. 	<ol style="list-style-type: none"> 3. Verifica que no exista el recurso humano a crear. 4. Si el recurso humano existe, entonces muestra un mensaje de error "Recurso Humano ya existe." 	

Tabla 2.32 Caso de Uso crearRecursoHumano

buscarRecursoHumano

Nombre	buscarRecursoHumano
---------------	---------------------

Pre-condición	Existe un Recurso Humano a ser buscado en el sistema.	
Post-condición	El Recurso Humano es encontrado o no existía en el sistema.	
Descripción	Realiza la búsqueda de un Recurso Humano en el sistema, controla que exista el Recurso Humano en el sistema.	
Actor		Sistema
Flujo Principal		
1. Ingresar nombreRecursoHumano. 2. Ejecuta la función Buscar.	3. Realiza la búsqueda según el nombreRecursoHumano como criterio. 4. Si el recurso humano existe, entonces muestra los atributos del Recurso Humano.	
Flujo Alternativo 1		
1. Ingresar nombreRecursoHumano. 2. Ejecuta la función Buscar.	3. Realiza la búsqueda según el nombreRecursoHumano como criterio. 4. Si el recurso humano no existe, entonces muestra un mensaje de error "Recurso Humano no existe".	

Tabla 2.33 Caso de Uso buscarRecursoHumano

modificarRecursoHumano

Nombre	modificarRecursoHumano	
Pre-condición	Exista el Recurso Humano a ser modificado.	
Post-condición	El Recurso Humano modificado queda registrado en el sistema.	
Descripción	Realiza la modificación de los atributos de un Recurso Humano.	
Actor		Sistema
Flujo Principal		
1. Ingresar nombreRecursoHumano.		

<p>2. Ejecuta la función Buscar.</p> <p>5. Modifica los atributos de la persona a excepción del código y de la cédula.</p> <p>6. Ejecuta la función Guardar.</p>	<p>3. Realiza la búsqueda del recurso humano.</p> <p>4. Si el recurso humano existe, entonces muestra los atributos del recurso humano.</p> <p>7. Realiza la actualización del recurso humano.</p>
Flujo Alterno 1	
<p>1. Ingresar nombreRecursoHumano.</p> <p>2. Ejecuta la función Buscar.</p>	<p>3. Realiza la búsqueda del recurso humano.</p> <p>4. Si el recurso humano no existe, entonces muestra mensaje de error "Recurso Humano no existe".</p>

Tabla 2.34 Caso de Uso modificarRecursoHumano

2.2.1.2.7. Paquete Administración de Suministros

crearSuministro

Nombre	crearSuministro	
Pre-condición	Exista un Suministro a ser creado Exista al menos un tipo de Suministro.	
Post-condición	El Suministro queda creado en el sistema.	
Descripción	Realiza la creación de un Suministro, controlando la existencia del elemento en el sistema.	
Actor		Sistema
Flujo Principal		
1. Ingresar los datos de Suministro: nombre Suministro, tipo de		

<p>Suministro, presentación de medicamento, fuente de medicamento, presentación de insumo y dosis de vacunas.</p> <p>2. Ejecuta la función Guardar.</p>	<p>3. Verifica que no exista el suministro a crear.</p> <p>4. Si el suministro no existe, entonces se realiza la creación del Suministro.</p>
Flujo Alterno 1	
<p>1. Ingresar los datos de Suministro: nombre Suministro, tipo de Suministro, presentación de medicamento, fuente de medicamento, presentación de insumo y dosis de vacunas.</p> <p>2. Ejecuta la función Guardar.</p>	<p>3. Verifica que no exista el suministro a crear.</p> <p>4. Si el suministro existe, entonces muestra un mensaje de error "Suministro ya existe".</p>

Tabla 2.35 Caso de Uso CrearSuministro

buscarSuministro

Nombre	buscarSuministro
Pre-condición	Existe un Suministro a ser buscado en el sistema.
Post-condición	El Suministro es encontrado, o no existía el suministro en el sistema.
Descripción	Realiza la búsqueda de un Suministro en el sistema, controla que exista un Suministro en el sistema.
Actor	Sistema
Flujo Principal	
<p>1. Ingresar nombreSuministro.</p> <p>2. Ejecuta la función Buscar.</p>	<p>3. Realiza la búsqueda según el nombreSuministro como criterio.</p>

	4. Si el suministro existe, entonces muestra los atributos del Suministro.
Flujo Alternativo 1	
1. Ingresar nombreSuministro. 2. Ejecutar la función Buscar.	3. Realiza la búsqueda según el nombreSuministro como criterio. 4. Si el suministro no existe, entonces muestra un mensaje de error "Suministro no existe".

Tabla 2.36 Caso de Uso buscarSuministro

modificarSuministro

Nombre	modificarSuministro	
Pre-condición	Exista un Suministro a ser modificado	
Post-condición	El Suministro modificado queda registrado en el sistema.	
Descripción	Realiza la modificación de los atributos de un Suministro.	
Actor	Sistema	
Flujo Principal		
1. Ingresar nombreSuministro. 2. Ejecutar la función Buscar.	3. Realiza la búsqueda del suministro. 4. Si el suministro existe, entonces muestra los atributos del suministro.	
5. Modifica los atributos de la persona a excepción del código. 6. Ejecuta la función Guardar.	7. Realiza la actualización del suministro.	
Flujo Alternativo 1		
1. Ingresar nombreSuministro. 2. Ejecuta la función Buscar.		

	<ol style="list-style-type: none"> 3. Realiza la búsqueda del suministro. 4. Si el suministro no existe, entonces muestra mensaje de error "Suministro no existe".
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 2.37 Caso de Uso ModificarSuministro

eliminarSuministro

Nombre	eliminarSuministro	
Pre-condición	Exista un Suministro a ser eliminado.	
Post-condición	El Suministro fue eliminado del sistema.	
Descripción	Realiza la eliminación de un Suministro, controlando la existencia del elemento en el sistema.	
	Actor	Sistema
Flujo Principal		
<ol style="list-style-type: none"> 1. Ingresar nombreSuministro. 2. Ejecuta la función Buscar. 	<ol style="list-style-type: none"> 3. Realiza la búsqueda del suministro. 4. Si el suministro existe, entonces muestra los atributos del suministro. 	<ol style="list-style-type: none"> 5. Ejecuta la función Eliminar. 6. Elimina el suministro.
Flujo Alterno 1		
<ol style="list-style-type: none"> 3. Ingresar nombreSuministro. 4. Ejecuta la función Buscar. 	<ol style="list-style-type: none"> 5. Realiza la búsqueda del suministro. 6. Si el suministro no existe, entonces muestra un mensaje de error "Suministro no existe". 	

Tabla 2.38 Caso de Uso EliminarSuministro

2.2.1.2.8. Paquete Administración de Brigada

crearBrigada

Nombre	crearBrigada	
Pre-condición	Exista una Brigada a ser creada. Exista al menos una comunidad para ser asignada a la Brigada. Exista al menos un recurso humano para ser asignado a la Brigada. Exista al menos un medicamento para ser asignado a la Brigada. Exista al menos un insumo para ser asignado a la Brigada. Exista al menos un equipo médico para ser asignado a la Brigada. Exista al menos una vacuna para ser asignada a la Brigada.	
Post-condición	La Brigada queda creada en el sistema.	
Descripción	Realiza la creación de una Brigada, controlando la existencia del elemento en el sistema.	
	Actor	Sistema
	Flujo Principal	
	1. Ingresar los datos de la Brigada, comunidades a ser visitadas, los recursos humanos que asistirán a la Brigada, medicamentos, insumos, vacunas y equipos médicos que serán llevados a la brigada. 2. Ejecuta la función Guardar.	3. Realiza la creación de la Brigada.

Tabla 2.39 Caso de Uso crearBrigada

buscarBrigada

Nombre	buscarBrigada	
Pre-condición	Existe una Brigada a ser buscada en el sistema.	
Post-condición	La Brigada es encontrada, o no existía la Brigada en el sistema.	
Descripción	Realiza la búsqueda de una Brigada en el sistema, controla que exista la Brigada en el sistema.	
Actor		Sistema
Flujo Principal		
1. Ingresar fechalnicio. 2. Ejecuta la función Buscar	3. Realiza la búsqueda según la fechalnicio como criterio. 4. Si la brigada existe, entonces muestra los atributos de la Brigada.	
Flujo Alternativo 1		
1. Ingresar fechalnicio. 2. Ejecuta la función Buscar	3. Realiza la búsqueda según la fechalnicio como criterio. 4. Si la brigada no existe, entonces muestra mensaje de error "Brigada no existe".	

Tabla 2.40 Caso de Uso buscarBrigada

modificarBrigada

Nombre	modificarBrigada	
Pre-condición	Exista la Brigada a ser modificada.	
Post-condición	La Brigada modificada queda registrada en el sistema.	
Descripción	Realiza la actualización de los atributos de la Brigada.	
Actor		Sistema
Flujo Principal		

<ol style="list-style-type: none"> 1. Ingresar fecha inicio. 2. Ejecutar la función Buscar. 5. Modificar los atributos de la brigada. 6. Ejecutar la función Guardar. 	<ol style="list-style-type: none"> 3. Realizar la búsqueda de la brigada. 4. Si la brigada existe, entonces muestra los atributos de la brigada. 7. Realizar la actualización de la Brigada.
Flujo Alternativo 1	
<ol style="list-style-type: none"> 1. Ingresar fecha inicio. 2. Ejecutar la función Buscar. 	<ol style="list-style-type: none"> 3. Realizar la búsqueda de la brigada. 4. Si la brigada no existe, entonces muestra un mensaje de error "Brigada no existe".

Tabla 2.41 Caso de Uso modificar Brigada

2.2.2. CLASES

2.2.2.1. Diagrama de Clases

Es necesario definir una nomenclatura para cada una de las Clases, atributos, operaciones y relaciones.

2.2.2.1.1. Nomenclatura

Para las clases se definirá el nombre de acuerdo al módulo que se está dando la solución, el nombre iniciará con una letra mayúscula, si el nombre de la clase está compuesto por dos o más palabras, no tendrán una separación y cada una empezará con la primera letra mayúscula y no contendrán ningún símbolo ortográfico.

Para los atributos se define la primera letra minúscula y las siguientes palabras tendrán la primera letra mayúscula, para definir la clase a la cual pertenece se añadirá al final del atributo el nombre de la clase, el nombre no tendrá separaciones y ningún símbolo ortográfico.

Para las operaciones la primera palabra será un verbo en infinitivo que refleje la acción que la operación realiza y la segunda palabra tendrá la primera letra en mayúscula y será el nombre de la clase a la cual pertenece, sin separaciones ni símbolos ortográficos.

2.2.2.1.2. Elementos

Según el detalle de los requerimientos y los casos de uso, para el desarrollo del sistema, serán necesarios los siguientes elementos:

Clases

- Empleado
- UnidadOperativa
- Comunidad
- Familia
- Persona
- Suministro
- Brigada

Para el desarrollo del Sistema de Gestión de Brigadas Médicas del Área 2 de Salud de Pastaza se define el diagrama de clases que se muestra en la Figura 2.11.

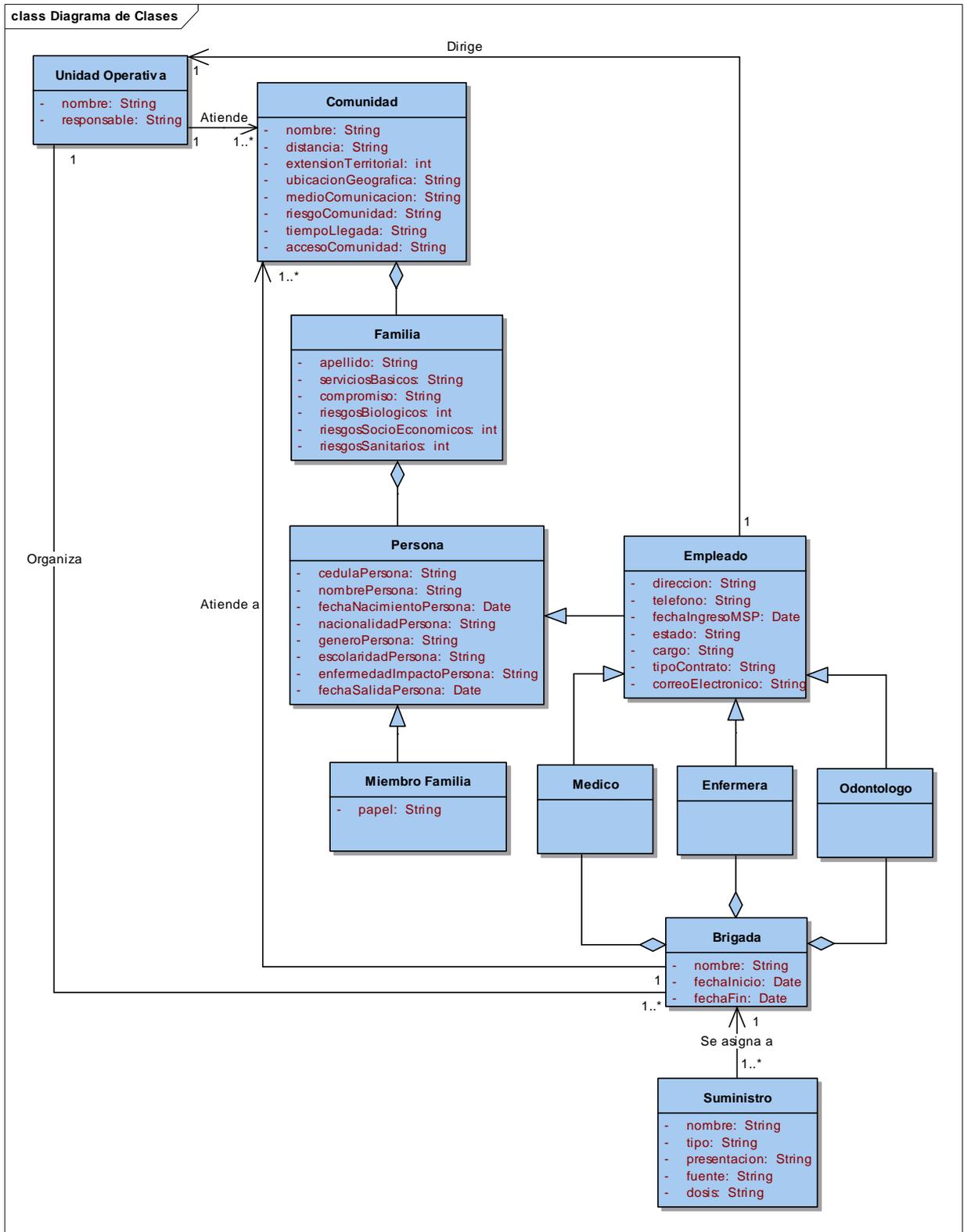


Figura 2.2.11 Diagrama de Clases

2.3. DISEÑO

2.3.1. MODELO ENTIDAD – RELACIÓN

A partir del Diagrama de Clases se crea el modelo entidad – relación que se muestra en Figura 2.12

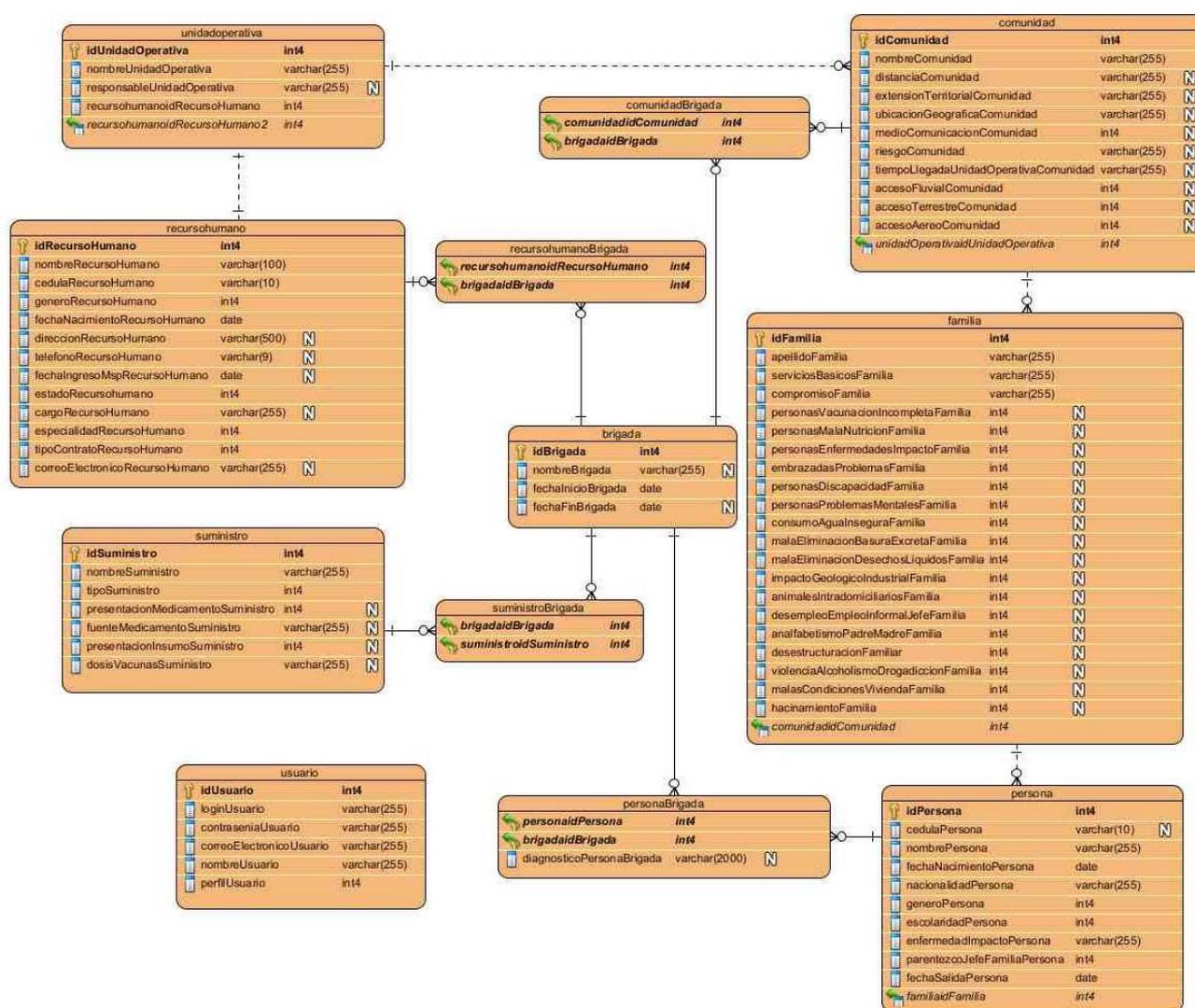


Figura 2.3.12 Diagrama Entidad - Relación

2.3.2. DIAGRAMAS DE SECUENCIA

2.3.2.1. Paquete de Administración de Usuarios

crearUsuario

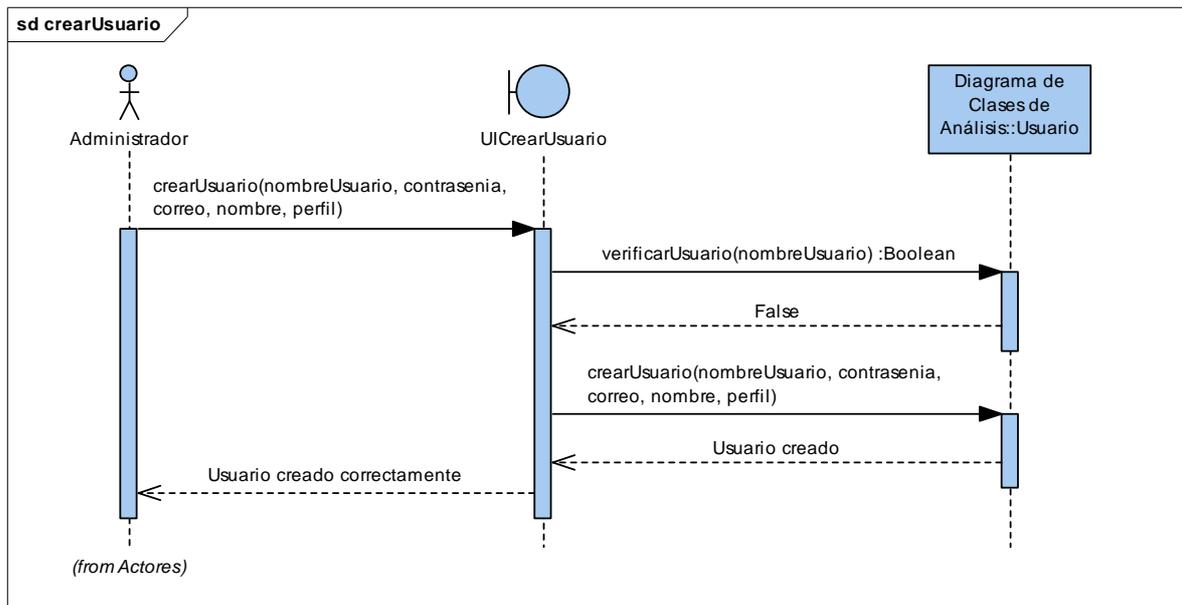


Figura 2.3.13 Diagrama de Secuencia crearUsuario

buscarUsuario

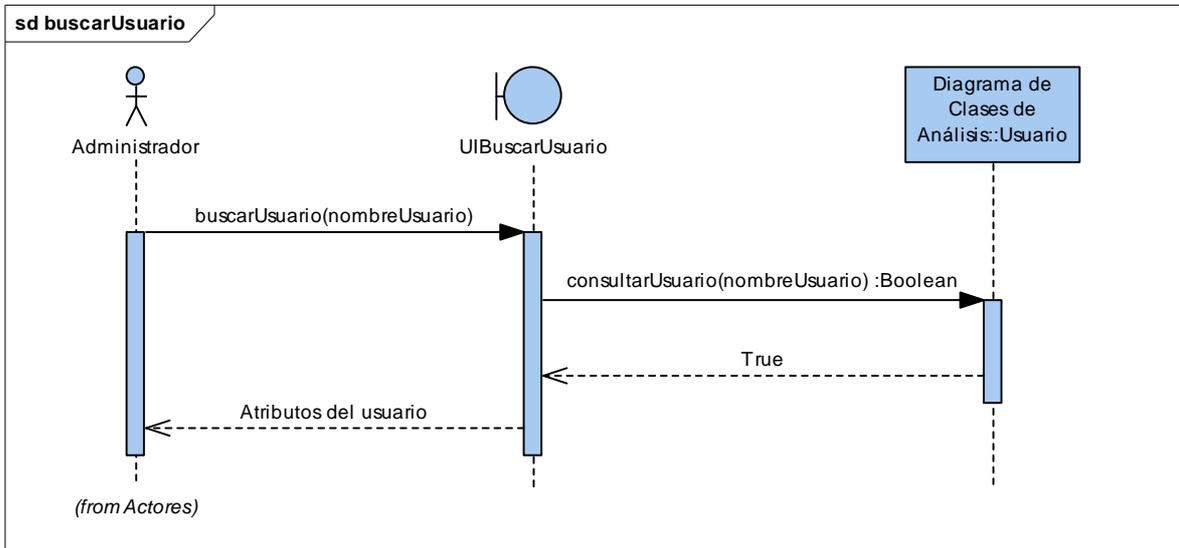


Figura 2.3.14 Diagrama de Secuencia buscarUsuario

Los demás diagramas de secuencia del Paquete de Administración de Usuarios se detallan en el Anexo II del presente trabajo.

2.3.2.2. Paquete de Administración de Unidades Operativas

crearUnidadOperativa

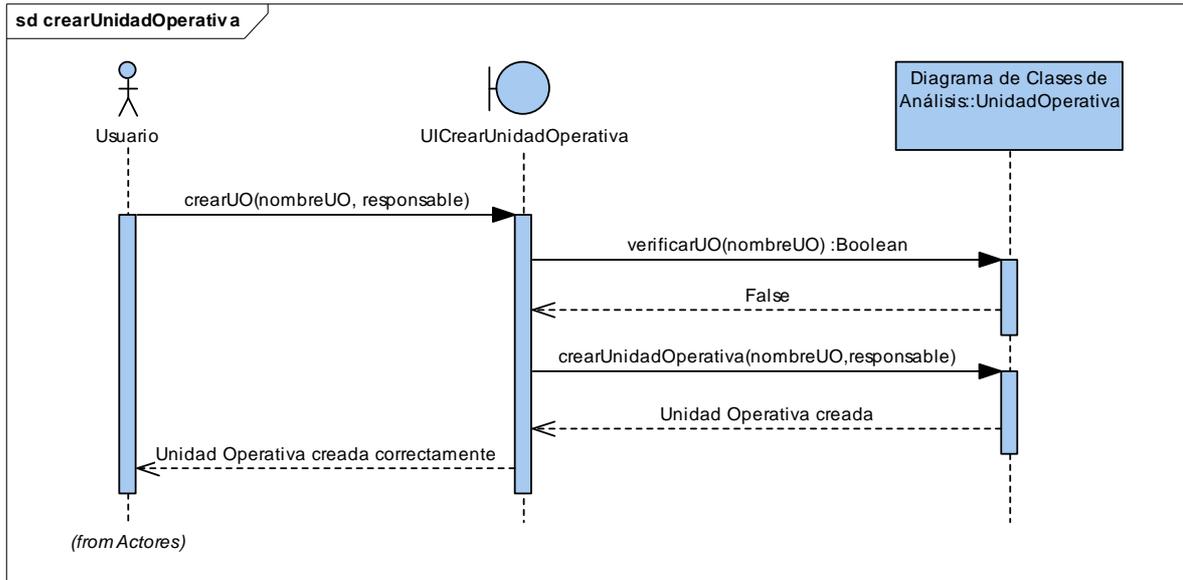


Figura 2.3.15 Diagrama de Secuencia crearUnidadOperativa

modificarUnidadOperativa

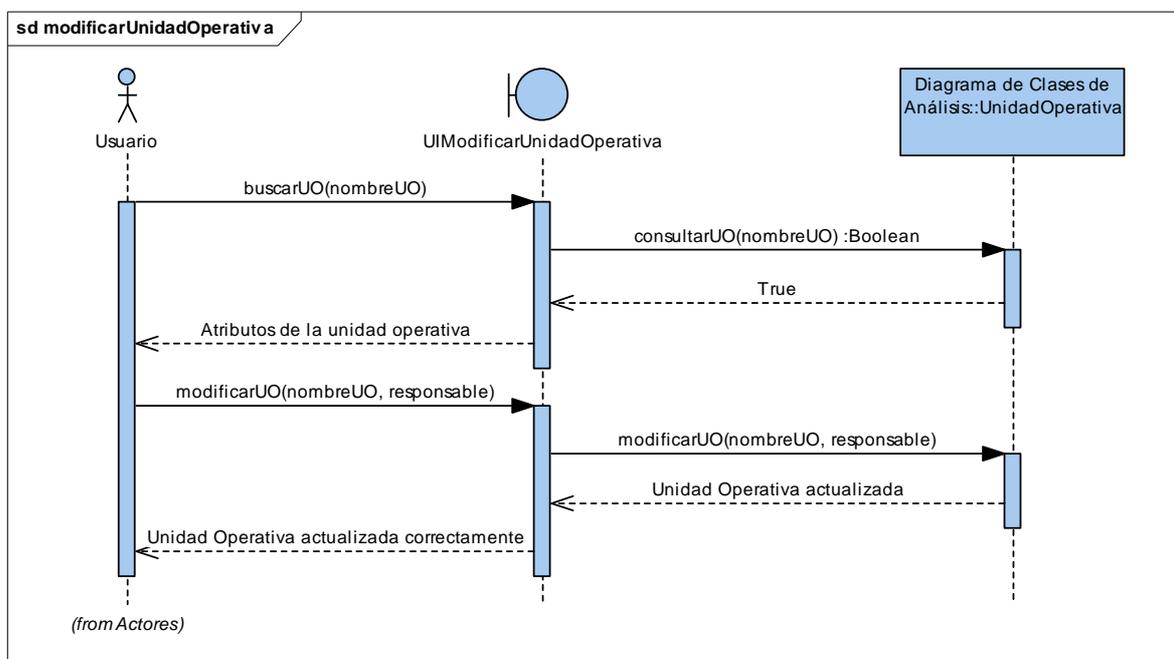


Figura 2.3.16 Diagrama de Secuencia modificarUnidadOperativa

Los demás diagramas de secuencia del Paquete de Administración de Unidades Operativas se detallan en el Anexo II del presente trabajo.

2.3.2.3. Paquete de Administración de Comunidades

buscarComunidad

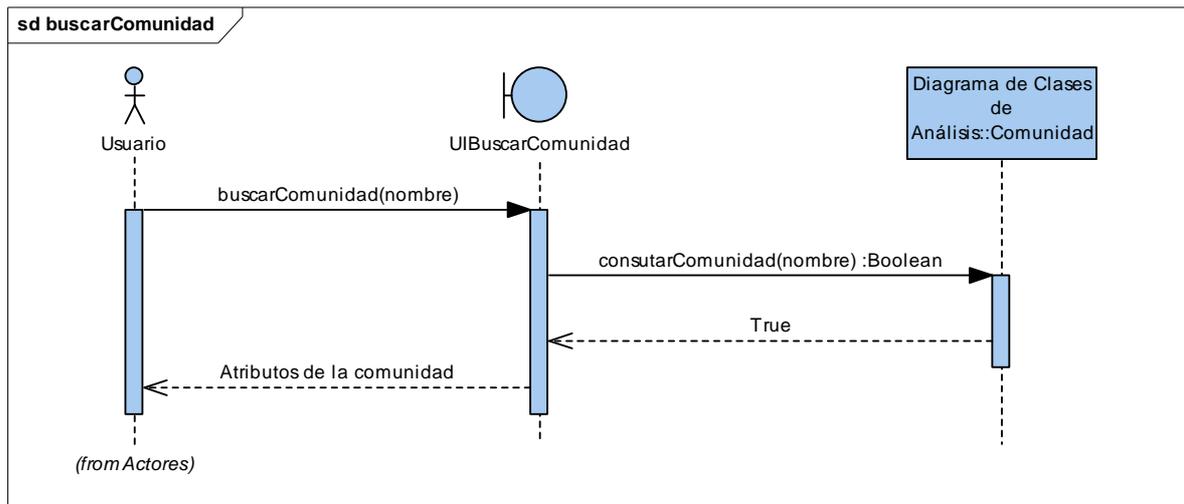


Figura 2.3.17 Diagrama de Secuencia buscarComunidad

eliminarComunidad

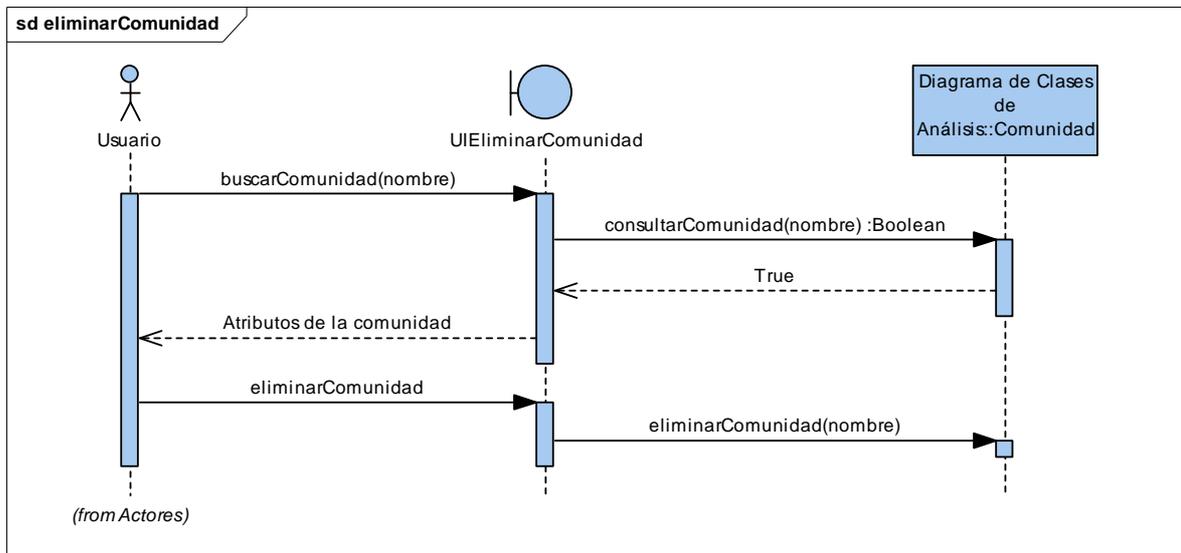


Figura 2.3.18 Diagrama de Secuencia eliminarComunidad

Los demás diagramas de secuencia del Paquete de Administración de Comunidades se detallan en el Anexo II del presente trabajo.

2.3.2.4. Paquete de Administración de Familia

crearFamilia

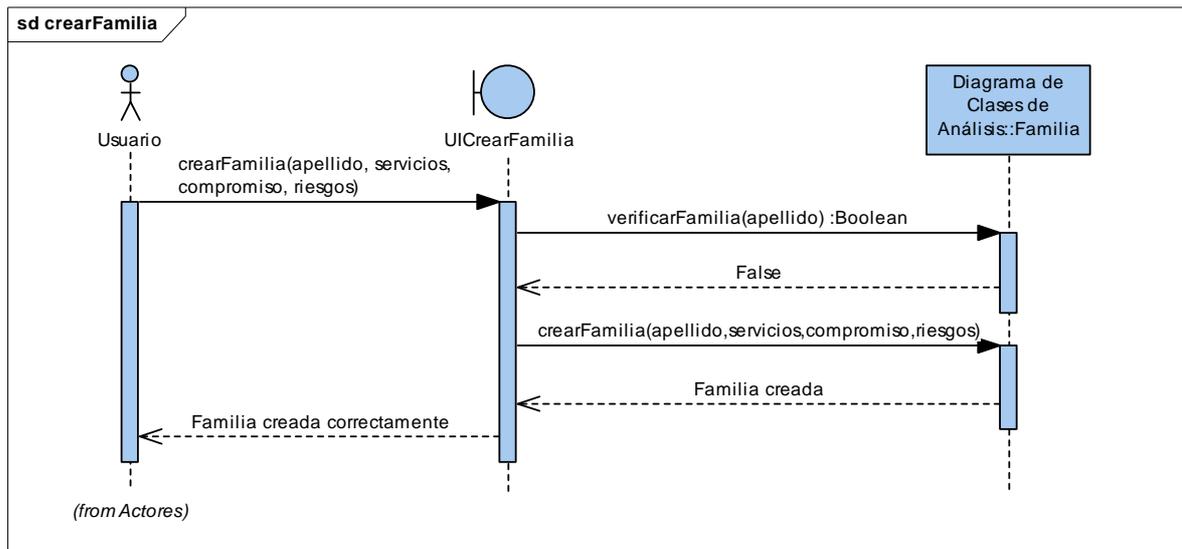


Figura 2.3.19 Diagramas de Secuencia crearFamilia

modificarFamilia

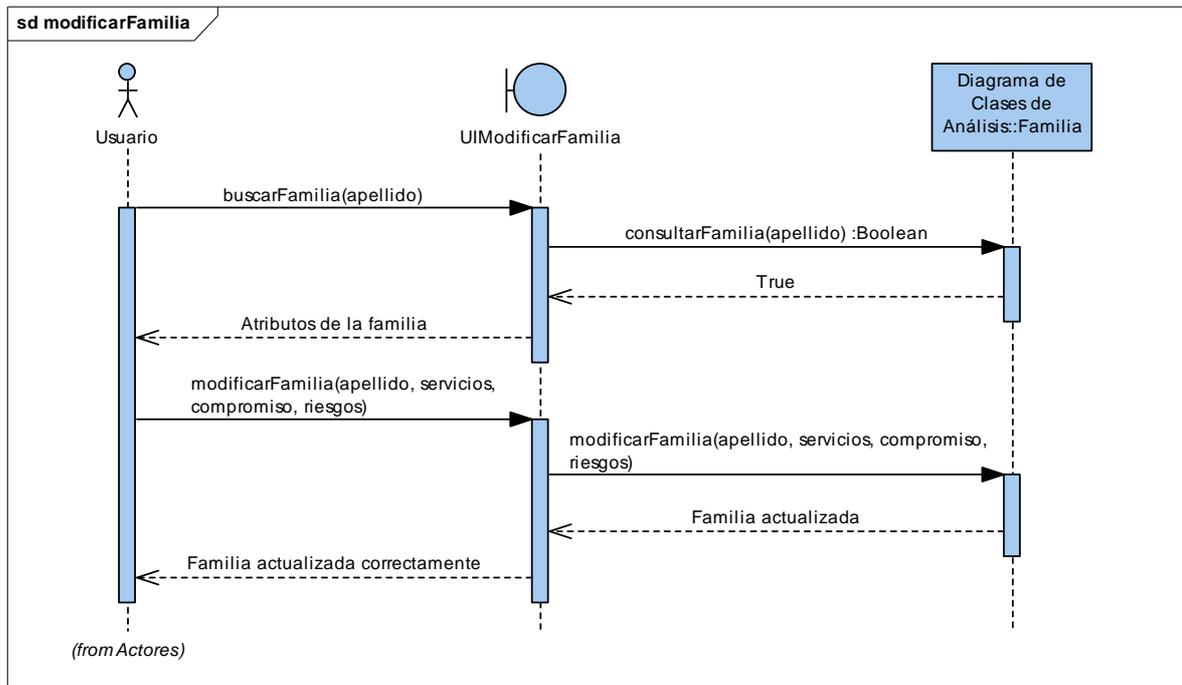


Figura 2.3.20 Diagramas de Secuencia modificarFamilia

Los demás diagramas de secuencia del Paquete de Administración de Familias se detallan en el Anexo II del presente trabajo.

2.3.2.5. Paquete de Administración de Personas

buscarPersona

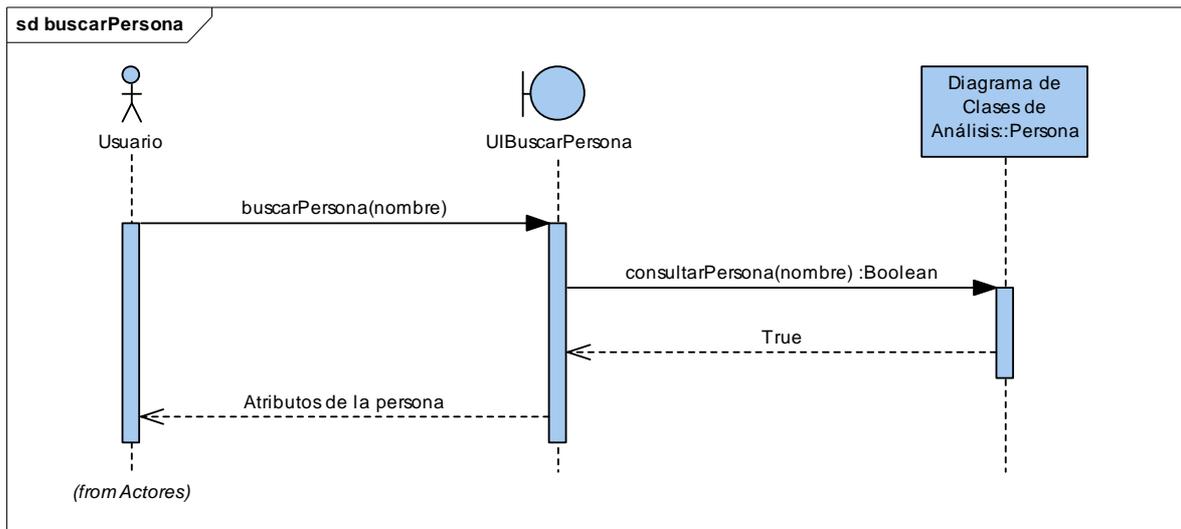


Figura 2.3.21 Diagramas de Secuencia buscarPersona

modificarPersona

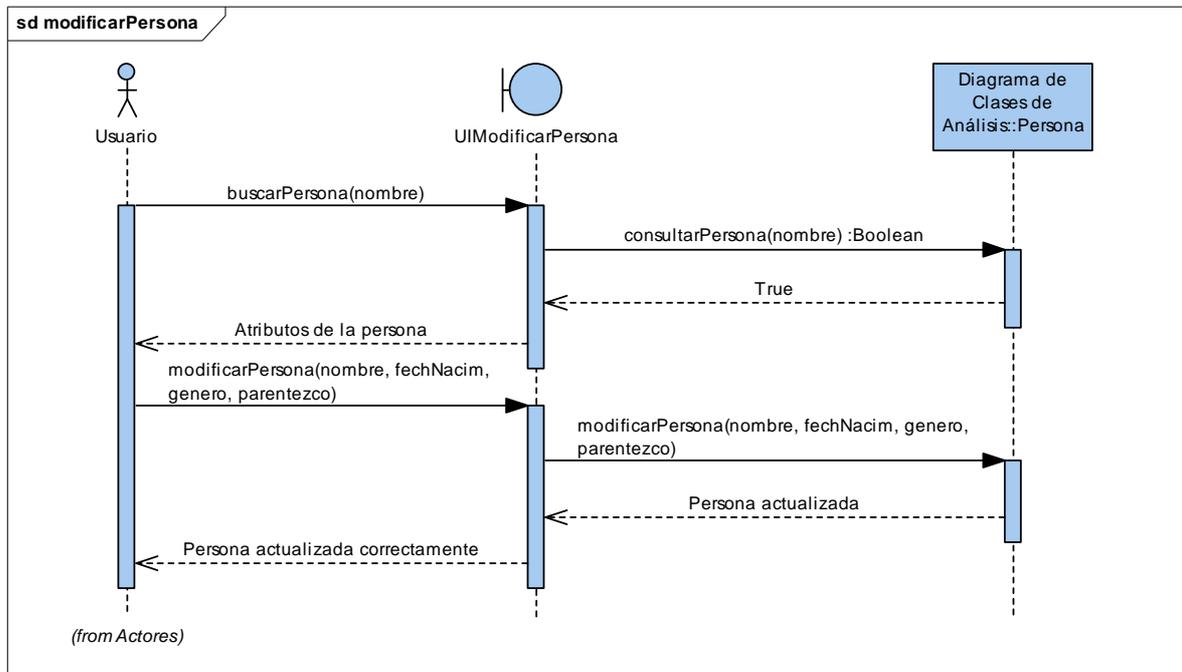


Figura 2.3.22 Diagramas de Secuencia modificarPersona

Los demás diagramas de secuencia del Paquete de Administración de Personas se detallan en el Anexo II del presente trabajo.

2.3.2.6. Paquete de Administración de Recursos Humanos

crearRecursoHumano

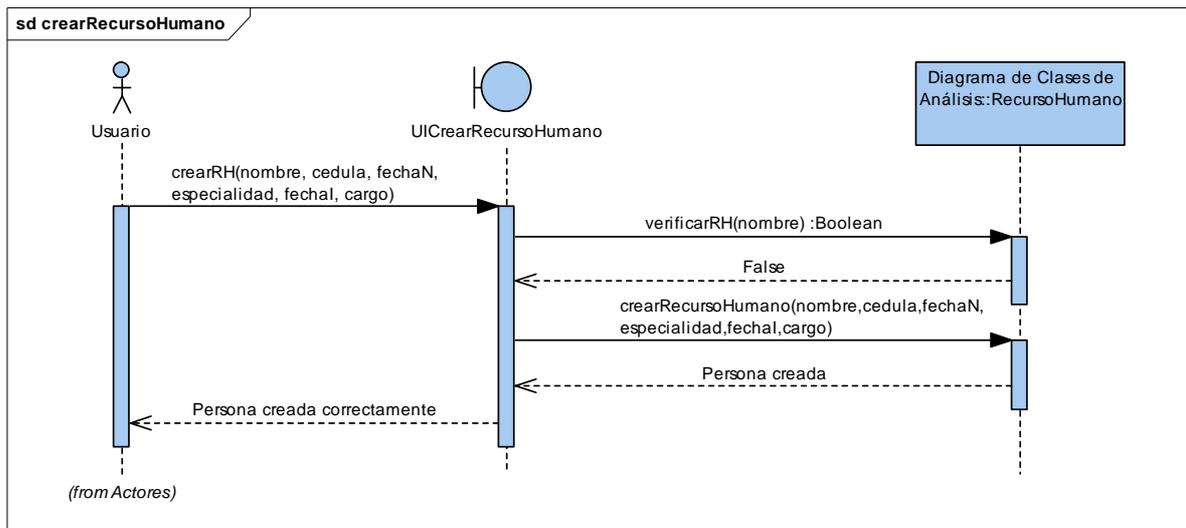


Figura 2.3.23 Diagramas de Secuencia crearRecursoHumano

Los demás diagramas de secuencia del Paquete de Administración de Recursos Humanos se detallan en el Anexo II del presente trabajo.

2.3.2.7. Paquete de Administración de Suministros

modificarSuministro

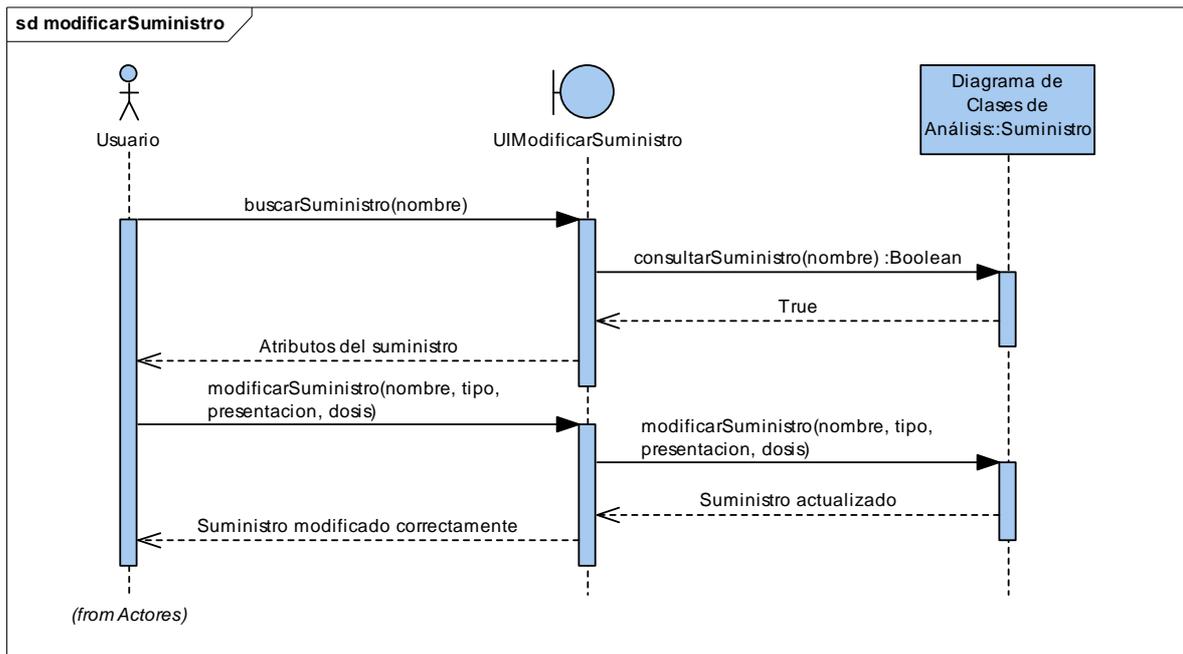


Figura 2.3.24 Diagramas de Secuencia modificarSuministro

eliminarSuministro

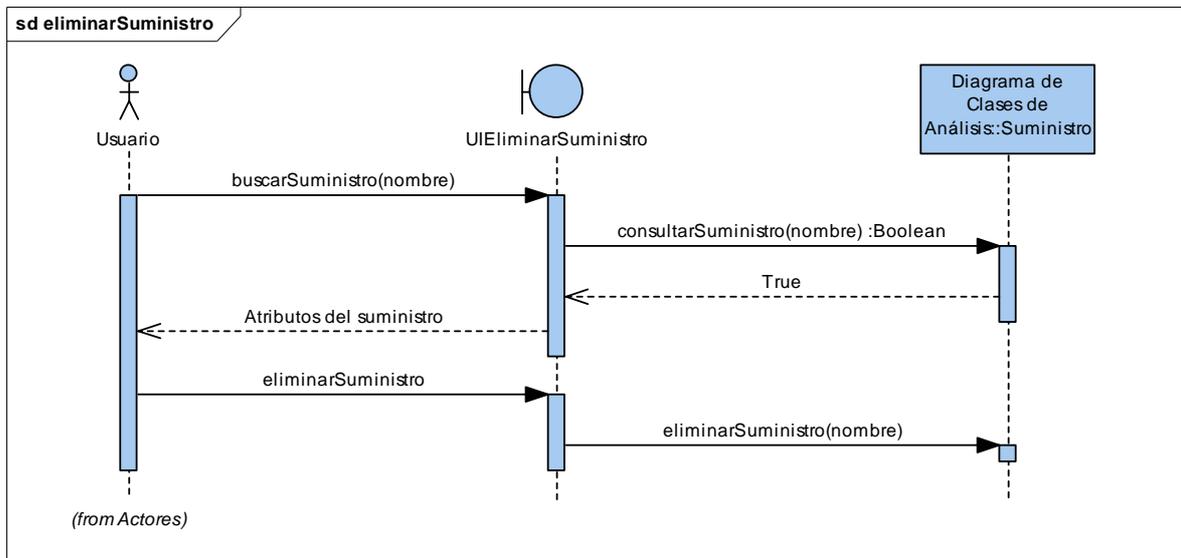


Figura 2.3.25 Diagramas de Secuencia eliminarSuministro

Los demás diagramas de secuencia del Paquete de Administración de Suministros se detallan en el Anexo II del presente trabajo.

2.3.2.8. Paquete de Administración de Brigadas

crearBrigada

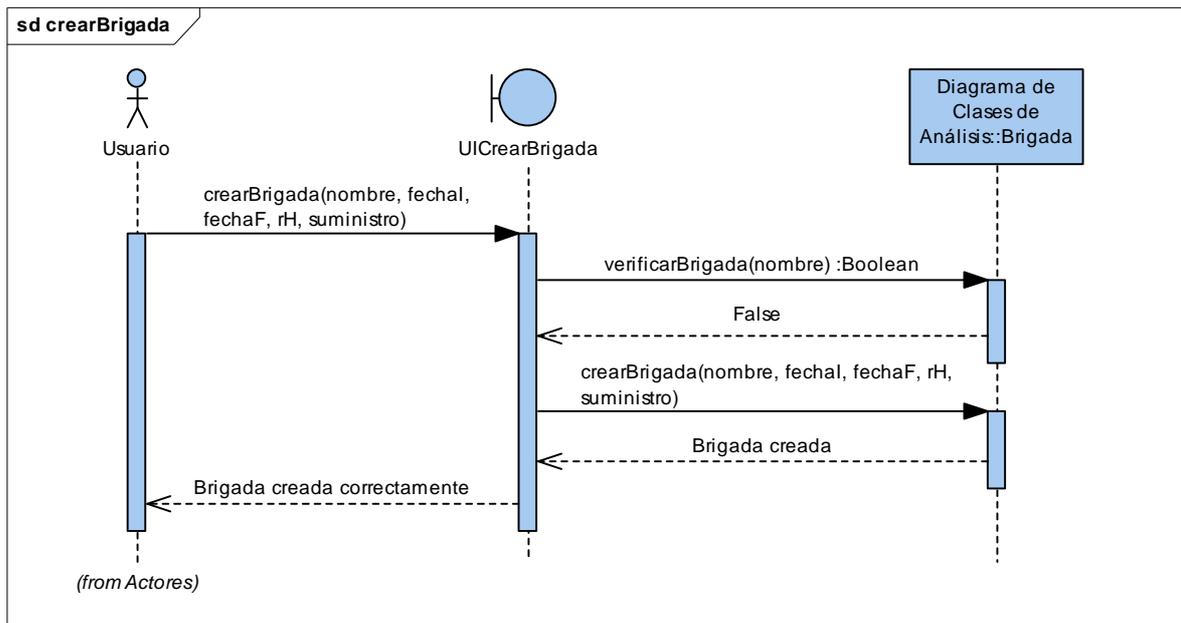


Figura 2.3.26 Diagramas de Secuencia crearBrigada

Los demás diagramas de secuencia del Paquete de Administración de Brigadas se detallan en el Anexo II del presente trabajo.

CAPÍTULO III. DESARROLLO, PRUEBAS E IMPLANTACIÓN DEL SISTEMA

3.1. DESARROLLO

3.1.1. DIAGRAMA DE DESPLIEGUE

A continuación se presenta el diagrama de despliegue que se utiliza para modelar el hardware utilizado en la implementación del sistema y las relaciones entre sus componentes.

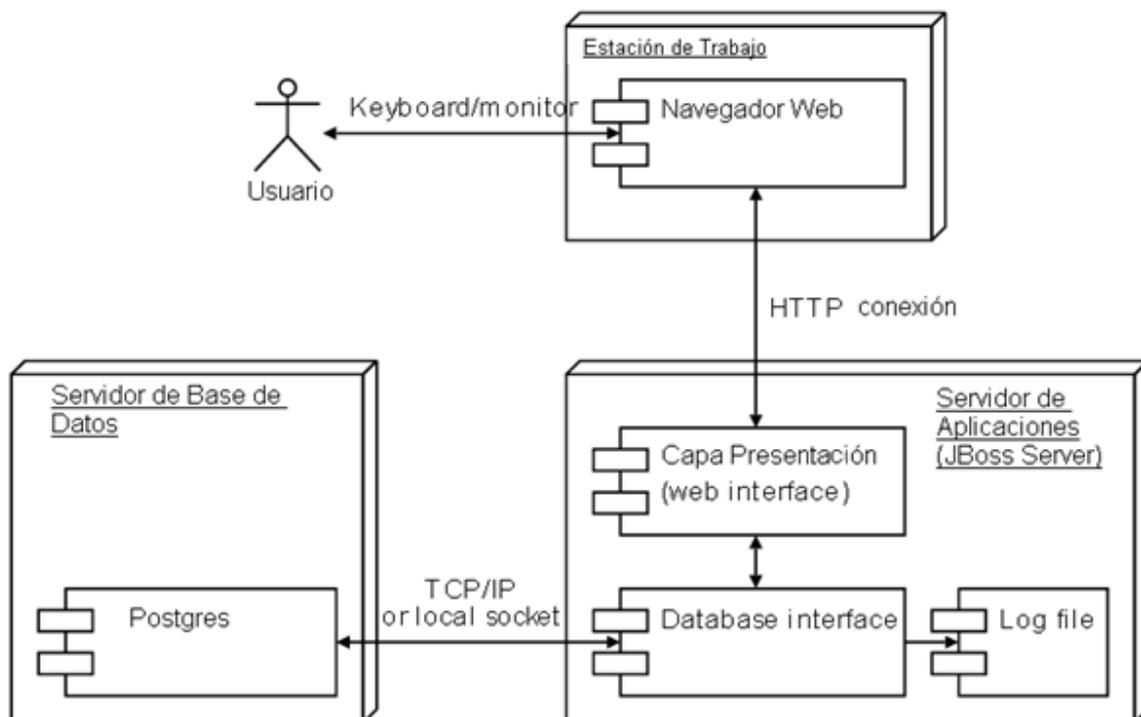


Figura 3.1 Diagrama de Despliegue

3.1.2. DIAGRAMA DE COMPONENTES

A continuación se presenta el diagrama de componentes que detalla la estructura del Sistema de Gestión de Brigadas Médicas.

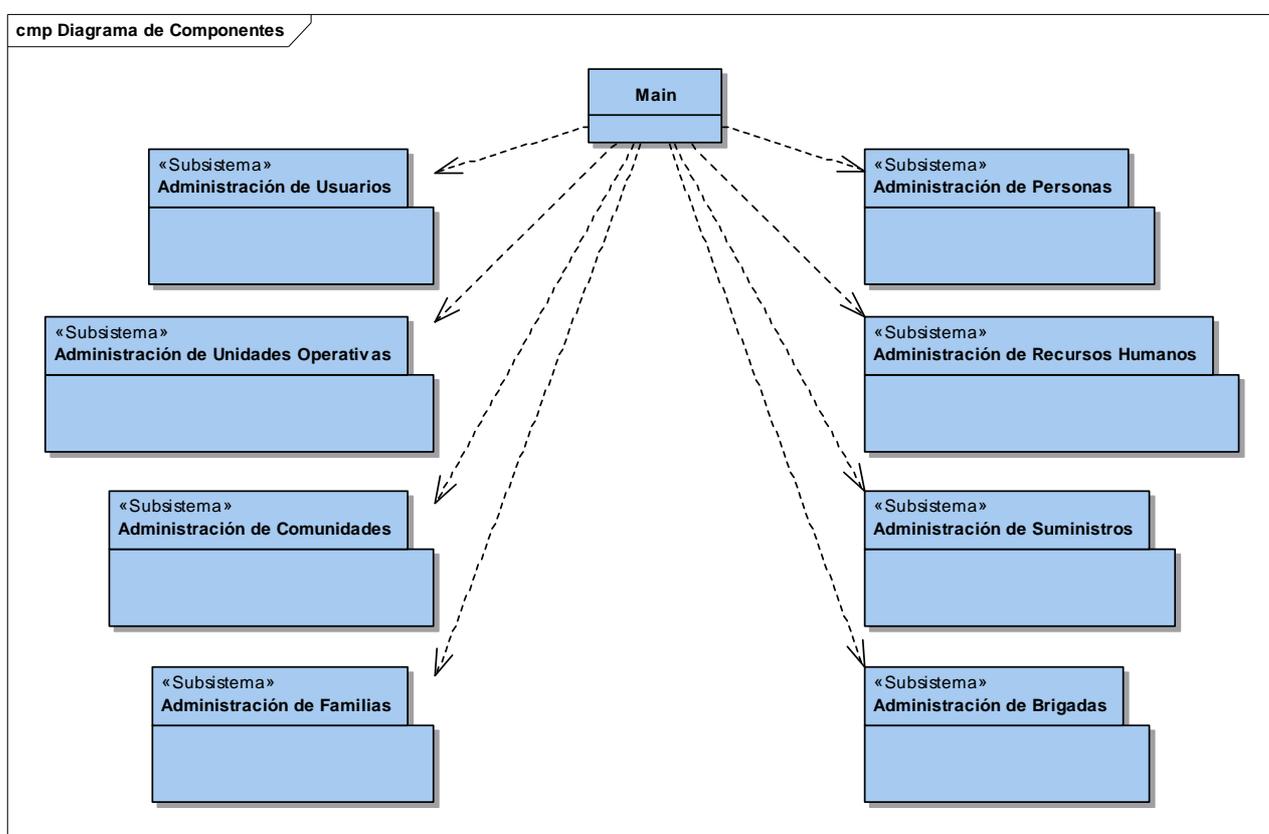


Figura 3.2 Diagrama de Componentes

Diagrama de componentes del Subsistema – Administración de Usuarios

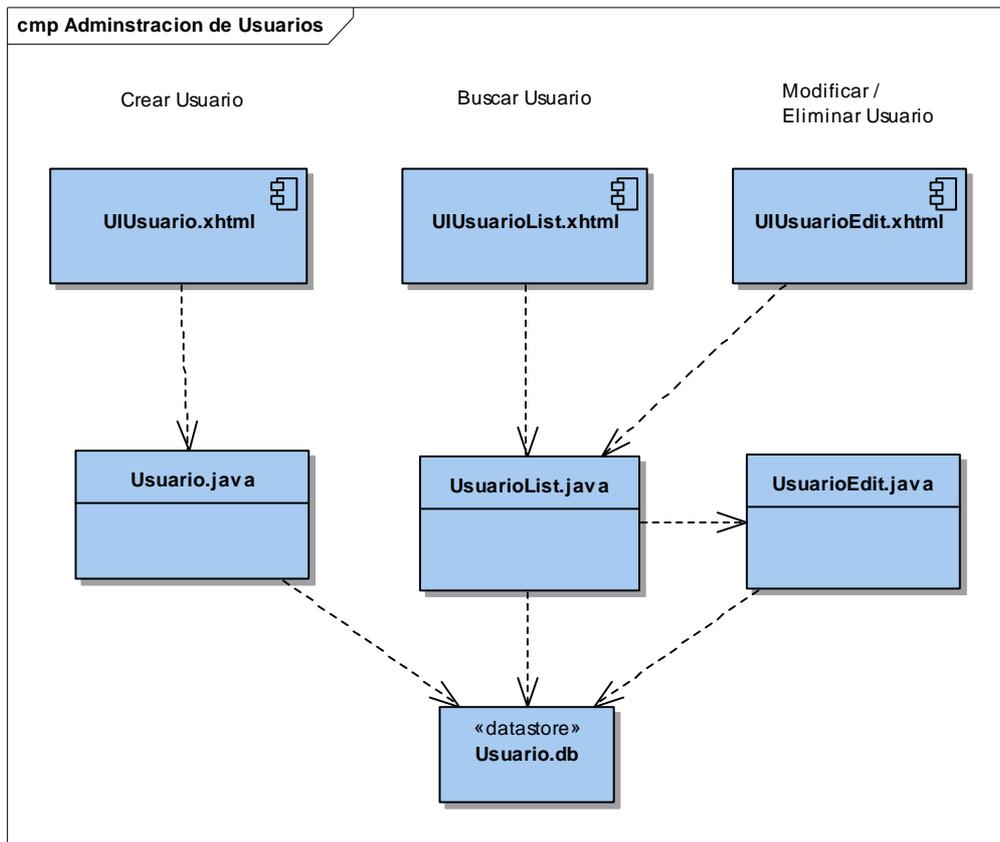


Figura 3.3 Diagrama de Componentes – Administración de Usuarios

Diagrama de componentes de Administración de Unidades Operativas

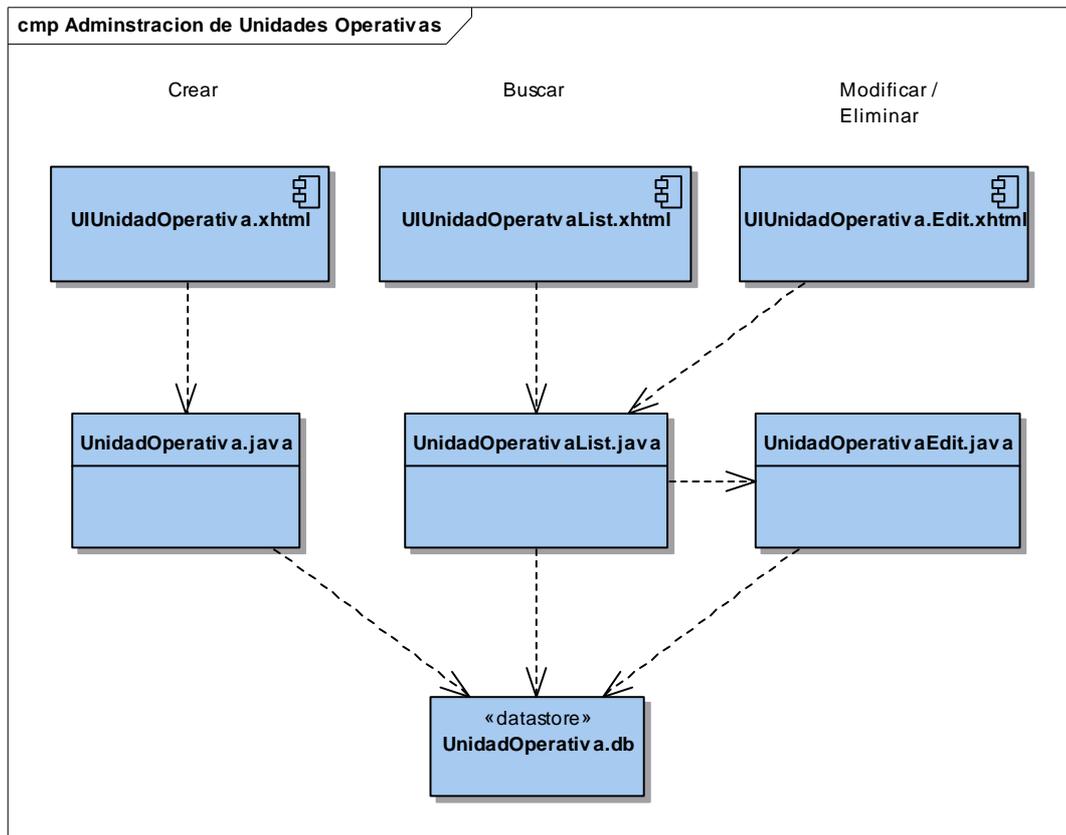


Figura 3.4 Diagrama de Componentes – Administración de Unidades Operativas

Los demás diagramas de componentes se detallan en el Anexo IV – Diagramas de Componentes.

3.1.3. CODIFICACIÓN

Para la codificación del sistema Brigamed se utilizará el ambiente de programación conformado por el IDE Eclipse con los plug-ins de JBoss Tools y Jboss Seam, esta combinación permite la generación de código facilitando la creación de CRUDS.

Como ejemplo de los estándares utilizados se presenta por cada componente el código respectivo, y el resto de código se detalla en el ANEXO III.

3.1.3.1. Codificación específica de componenetes

3.1.3.1.1. Botón con Link

```
<s:button view="/PAGINAS/RecursosHumanos/RecursohumanoEdit.xhtml"
         id="create"
         propagation="none"
         value="Create recursohumano">
  <f:param name="recursohumanoIdrecursohumano" />
</s:button>
```

3.1.3.1.2. Botón simple

```
<h:commandButton id="save"
                 value="Save"
                 action="#{recursohumanoHome.persist}"
                 disabled="#{!recursohumanoHome.wired}"
                 rendered="#{!recursohumanoHome.managed}" />
```

3.1.3.1.3. Tabla consulta

```
<rich:dataTable
  id="recursohumanoList"
  var="_recursohumano"
  value="#{recursohumanoList.resultList}"
  rendered="#{not empty recursohumanoList.resultList}">
  <h:column>
    <f:facet name="header">
      <ui:include src="/layout/sort.xhtml">
        <ui:param name="entityList" value="#{recursohumanoList}" />
        <ui:param name="propertyLabel" value="Idrecursohumano" />
        <ui:param name="propertyPath"
          value="recursohumano.idrecursohumano" />
      </ui:include>
    </f:facet>
```

```

        <h:outputText value="#{_recursohumano.idrecursohumano}"/>
    </h:column>
    <h:column>
</rich:column>
</rich:dataTable>

```

3.1.3.1.4. Combo box

```

<h:selectOneMenu id="especialidadrecursohumanoUnidad"
    value="#{recursohumanoHome.instance.especialidadrecursohumano}"
    required="true">
    <s:selectItems var="catalogo"
        value="#{listaCatalogosEspecialidadrecursohumano}"
        label="#{catalogo.valorcatalogo}"
        noSelectionLabel="#{messages['Seleccione Uno']}" />
    <s:convertEntity />
    <a:support event="onblur"reRender="especialidadrecursohumanoField"
        bypassUpdates="true" ajaxSingle="true" />
</h:selectOneMenu>

```

3.1.3.1.5. Input

```

<h:inputText
    id="nombrerecursohumano"
    required="true"
    size="100"
    maxlength="100"
    value="#{recursohumanoHome.instance.nombrerecursohumano}">

```

3.1.3.1.6. Output

```

<h:outputText value="#{_recursohumano.idrecursohumano}"/>

```

3.1.3.1.7. Form

```

<h:form id="recursohumanoSearch" styleClass="edit">
</ h:form>

```

3.2. PRUEBAS

Una parte importante de la fase de construcción de RUP son las pruebas del sistema, las cuales se encargan de asegurar que el software cumple con lo estipulado en el punto 2.1 Especificación de Requerimientos.

Las pruebas se realizarán los días 7 y 8 de febrero de 2011 en las instalaciones del Área 2 de Salud. Se han definido 9 casos de pruebas que describen una funcionalidad por cada módulo, los usuarios que llevarán a cabo estas pruebas son: La Directora del Área, La Secretaria y 2 Especialistas (1 Médico y 1 Odontólogo).

Los usuarios se encargarán de seguir los procedimientos y utilizar los datos de las entradas definidas para cada Caso de Prueba, describirán los resultados obtenidos y resaltarán una observación para cada Caso.

Caso de Prueba 1

Prueba Administrar Usuario – Caso Válido	
Nombre de Proceso:	Crear Usuario
Descripción de Proceso:	Se creará un nuevo Usuario
Escenario:	En la pantalla principal el usuario seleccionará el menú Usuario
Entradas:	Nombre: Edwin Nombre de usuario: emartinez Contraseña: 199288 E-mail: edwin.fmn@gmail.com Perfil: 1
Pre-condiciones	Selección función: <i>Crear</i>
Procedimientos	<ol style="list-style-type: none"> 1. Menú inicial, seleccionar Usuario. 2. Seleccionar <i>Crear</i>. 3. Ingresar los valores indicados en las entradas. 4. Seleccionar Guardar.
Resultados	<ol style="list-style-type: none"> 1. Seleccionar Usuario. 2. Ingresar Nombre de Usuario. 3. Seleccionar <i>Buscar</i>. 4. Se visualiza el Usuario con los datos de entrada.
Observaciones	Luego de ejecutar el procedimiento en el paso 4 se puede visualizar una tabla con los datos del nuevo Usuario.

Tabla 3.1 Caso de Prueba Administrar Usuario

Caso de Prueba 2

Prueba Administrar Unidad Operativa – Caso Válido	
Nombre de Proceso:	Crear Unidad Operativa.
Descripción de Proceso:	Se creará una nueva Unidad Operativa.
Escenario:	En la pantalla principal el usuario seleccionará el menú de Unidad Operativa
Entradas:	Nombre: Unidad 1 Responsable: Juan Pérez
Pre-condiciones:	Selección de la función: <i>Crear</i>
Procedimientos:	<ol style="list-style-type: none"> 1. Menú inicial, seleccionar Unidad Operativa. 2. Seleccionar <i>Crear</i>. 3. Ingresar los valores indicados en las entradas. 4. Seleccionar <i>Guardar</i>.
Resultados:	<ol style="list-style-type: none"> 1. Seleccionar Unidad Operativa. 2. Ingresar Nombre de Unidad Operativa. 3. Seleccionar <i>Buscar</i>. 4. Se visualiza la Unidad Operativa con los datos de entrada.
Observaciones :	Luego de ejecutar el procedimiento en el paso 4 se puede visualizar una tabla con los datos de la nueva Unidad Operativa.

Tabla 3.2 Caso de Prueba Administrar Unidad Operativa

Caso de Prueba 3

Prueba Administrar Comunidad – Caso Válido																			
Nombre de Proceso:	Crear Comunidad																		
Descripción de Proceso:	Se creará una nueva Comunidad																		
Escenario:	En la pantalla principal el usuario seleccionará el menú de Comunidad																		
Entradas:	<table> <tr> <td>Nombre:</td> <td>Chiriyacu</td> </tr> <tr> <td>Extensión:</td> <td>200 Km²</td> </tr> <tr> <td>Comunicación comunidad:</td> <td>Radio</td> </tr> <tr> <td>Riesgo:</td> <td>Ninguno</td> </tr> <tr> <td>Tiempo de llegada:</td> <td>2 horas</td> </tr> <tr> <td>Acceso fluvial:</td> <td>Ninguno</td> </tr> <tr> <td>Acceso terrestre:</td> <td>Carretera</td> </tr> <tr> <td>Acceso aéreo:</td> <td>Ninguno</td> </tr> <tr> <td>Unidad Operativa:</td> <td>Unidad 1</td> </tr> </table>	Nombre:	Chiriyacu	Extensión:	200 Km ²	Comunicación comunidad:	Radio	Riesgo:	Ninguno	Tiempo de llegada:	2 horas	Acceso fluvial:	Ninguno	Acceso terrestre:	Carretera	Acceso aéreo:	Ninguno	Unidad Operativa:	Unidad 1
Nombre:	Chiriyacu																		
Extensión:	200 Km ²																		
Comunicación comunidad:	Radio																		
Riesgo:	Ninguno																		
Tiempo de llegada:	2 horas																		
Acceso fluvial:	Ninguno																		
Acceso terrestre:	Carretera																		
Acceso aéreo:	Ninguno																		
Unidad Operativa:	Unidad 1																		
Pre-condiciones:	Selección de la función: <i>Crear</i>																		
Procedimientos:	<ol style="list-style-type: none"> 1. Menú inicial, seleccionar Comunidad. 2. Seleccionar <i>Crear</i>. 3. Ingresar los valores indicados en las entradas. 4. Seleccionar <i>Guardar</i>. 																		
Resultados:	<ol style="list-style-type: none"> 1. Seleccionar Comunidad. 2. Ingresar Nombre de Comunidad. 3. Seleccionar <i>Buscar</i>. 4. Se visualiza la Comunidad con los datos de entrada. 																		
Observaciones:	Luego de ejecutar el procedimiento en el paso 4 se puede visualizar una tabla con los datos de la nueva Comunidad.																		

Tabla 3.3 Caso de Prueba Administrar Comunidad

Caso de Prueba 4

Prueba Índices de Crecimiento – Caso Válido	
Nombre de Proceso:	Mostrar Índices de Crecimiento
Descripción de Proceso:	Se mostrarán los Índices de Crecimiento Poblacional por Comunidad.
Escenario:	En la pantalla principal el usuario seleccionará el menú de Comunidad y Submenú de Índices de Crecimiento.
Entradas:	Ninguna
Pre-condiciones:	Selección del submenú: <i>Índices de Crecimiento</i>
Procedimientos:	<ol style="list-style-type: none"> 1. Menú inicial, seleccionar Comunidad. 2. Seleccionar Submenú Índices de Crecimiento
Resultados:	<ol style="list-style-type: none"> 1. Se visualiza lo datos poblacionales por cada una de las Comunidades registradas, separados por año y un porcentaje de crecimiento poblacional (índice).
Observaciones:	Luego de ejecutar el procedimiento en el paso 2 se puede visualizar una tabla con los datos de las comunidades, su población por año y el índice de crecimiento poblacional en el rango de años mostrado.

Tabla 3.4 Caso de Prueba Índices de Crecimiento

Caso de Prueba 5

Prueba Administrar Familia – Caso Erróneo	
Nombre de Proceso:	Crear Familia.
Descripción de Proceso:	Se creará una nueva Familia.
Escenario:	En la pantalla principal el usuario seleccionará el menú Familia.
Entradas:	Comunidad: Chiriyacu Apellido: Le56ma Compromiso: Ninguno Servicios Básicos: Luz Eléctrica Calificaciones: 3 (todos los campos)
Pre-condiciones:	Selección función: <i>Crear</i>
Procedimientos:	1. Menú inicial, seleccionar Familia. 2. Seleccionar <i>Crear</i> . 3. Ingresar los valores indicados en las entradas. 4. Seleccionar <i>Guardar</i> .
Resultados:	Error en los Datos, los siguientes campos tienen valores incorrectos: <ul style="list-style-type: none"> • Apellido
Observaciones:	Los campos mostrados en el resultado no tienen el formato indicado.

Tabla 3.5 Caso de Prueba Administrar Familia

Caso de Prueba 6

Prueba Administrar Persona – Caso Válido	
Nombre de Proceso:	Crear Persona
Descripción de Proceso:	Se creará una nueva Persona.
Escenario:	En la pantalla principal el usuario seleccionará el menú Persona.
Entradas:	Cedula: 1714136023 Nombre: Nail Lema Fecha de Nacimiento: 1988/03/14 Escolaridad: Ninguna Enfermedad Impacto: Ninguna Parentesco: Hijo/a Fecha de Salida: Ninguna Apellido Familia: Lema
Pre-condiciones:	Selección función: <i>Crear</i>
Procedimientos:	1. Menú inicial, seleccionar Persona. 2. Seleccionar <i>Crear</i> . 3. Ingresar los valores indicados en las entradas. 4. Seleccionar <i>Guardar</i> .
Resultados:	1. Seleccionar Persona. 2. Ingresar Nombre de Persona. 3. Seleccionar <i>Buscar</i> . 4. Se visualiza la Persona con los datos de entrada.
Observaciones:	Luego de ejecutar el procedimiento en el paso 4 se puede visualizar una tabla con los datos de la nueva Persona.

Tabla 3.6 Caso de Prueba Administrar Persona

Caso de Prueba 7

Prueba Administrar Suministro – Caso Erróneo	
Nombre de Proceso:	Crear Suministro
Descripción de Proceso:	Se creará un nuevo Suministro
Escenario:	En la pantalla principal el usuario seleccionará el menú Suministro
Entradas:	Nombre: 4ape7 Tipo: Insumo Presentación: Ninguna Fuente: Ninguna Insumo: Ninguno
Pre-condiciones:	Selección función: <i>Crear</i>
Procedimientos:	1. Menú inicial, seleccionar Suministro. 2. Seleccionar <i>Crear</i> . 3. Ingresar los valores indicados en las entradas. 4. Seleccionar <i>Guardar</i> .
Resultados:	Error en los Datos, los siguientes campos tienen valores incorrectos: <ul style="list-style-type: none"> • Nombre • Insumo
Observaciones:	Los campos mostrados en el resultado no tienen el formato indicado.

Tabla 3.7 Caso de Prueba Administrar Suministro

Caso de Prueba 8

Prueba Administrar Recurso Humano – Caso Válido	
Nombre de Proceso:	Crear Recurso Humano
Descripción de Proceso:	Se creará un nuevo Recurso Humano
Escenario:	En la pantalla principal el usuario seleccionará el menú Recurso Humano
Entradas:	Nombre: Ivan Cadena Cédula: 178345739-2 Género: Masculino Fecha de nacimiento: 1978/05/08 Dirección: Avda. Primera 7-32 Correo electrónico: icadena@gmail.com Estado: Activo Cargo: Jefe de Cirugía Especialidad: Médico Cirujano
Pre-condiciones:	Selección función: <i>Crear</i>
Procedimientos:	<ol style="list-style-type: none"> 1. Menú inicial, seleccionar Recurso Humano. 2. Seleccionar <i>Crear</i>. 3. Ingresar los valores indicados en las entradas. 4. Seleccionar <i>Guardar</i>.
Resultados:	<ol style="list-style-type: none"> 1. Seleccionar Recurso Humano. 2. Ingresar el Nombre del Recurso Humano. 3. Seleccionar <i>Buscar</i>. 4. Se visualiza la Recurso Humano con los datos de las entradas.
Observaciones:	Luego de ejecutar el procedimiento en el paso 4 se puede visualizar una tabla con los datos del nuevo Recurso Humano.

Tabla 3.8 Caso de Prueba Administrar Recurso Humano

Caso de Prueba 9

Prueba Administrar Brigada – Caso Erróneo	
Nombre de Proceso:	Crear Brigada
Descripción de Proceso:	Se creará una nueva Brigada.
Escenario:	En la pantalla principal el usuario seleccionará el menú Brigada.
Entradas:	Brigada: Brigada 1 Fecha Inicial: 2011/03/01 Recursos Humanos: - Insumos: -
Pre-condiciones:	Selección función: <i>Crear</i>
Procedimientos:	Menú inicial, seleccionar Brigada. Seleccionar <i>Crear</i> . Ingresar los valores indicados en las entradas. Seleccionar <i>Guardar</i> .
Resultados:	Error en los Datos, los siguientes campos tienen valores incorrectos: <ul style="list-style-type: none"> • Recursos Humanos • Insumos
Observaciones:	Los campos mostrados en el resultado no pueden ser vacíos.

Tabla 3.9 Caso de Prueba Administrar Brigada

3.3. IMPLANTACIÓN

3.3.1. REQUISITOS DE HARDWARE

Para el correcto funcionamiento del sistema el Área 2 de Salud de Pastaza debe cumplir los siguientes requisitos:

El servidor deberá reunir los siguientes requisitos mínimos:

- Disponer de 5 Gb de espacio libre en el Disco Duro.
- Disponer mínimo de 1 Gb de memoria RAM (2 Gb recomendado).
- Procesador doble núcleo (Core 2 Duo recomendado).

Los equipos cliente deberán cumplir los siguientes requisitos mínimos:

- Dispones Mínimo de 512 Mb de memoria RAM (1 Gb recomendado).

3.3.2. REQUISITOS DE SOFTWARE

El servidor deberá reunir los siguientes requisitos de software:

- El software requerido se detalla en el documento de especificación de requerimientos en el apartado 2.1.3.5 Requerimientos Técnicos.

Los equipos cliente deberán reunir el siguiente requisito de software:

- Navegador Web Mozilla Firefox v 2.0 o superior, Internet Explorer v 6.0 o superior, Google Chrome v 7.0 o superior.

3.3.3. PLAN DE INSTALACIÓN

Para la instalación inicial del sistema en el equipo servidor se seguirán los siguientes pasos:

- a. Tener instalado el sistema operativo.
- b. Instalar el servidor de aplicaciones JBoss con las librerías Seam.
- c. Instalar el motor de base de datos PostgreSQL v 8.4.
- d. Instalar la Máquina Virtual de Java JDK v 1.6.
- e. Realizar la restauración de la base de datos.
- f. Copiar los archivos fuentes del Sistema de Gestión de Brigadas Médicas BRIGAMED.
- g. Iniciar el servidor de aplicaciones Web JBoss.

Todo el software necesario, así como la estructura de la base de datos y los archivos fuentes del Sistema de Gestión de Brigadas Médicas se encuentran en el Kit de Instalación en el ANEXO V.

3.4. ANÁLISIS DE RESULTADOS

Para realizar el análisis de resultados se tomará en cuenta las respuestas de la encuesta aplicada el día viernes 11 de febrero de 2011 al 50% del personal (4 usuarios: La Directora, La Secretaria y 2 Especialistas) del Área 2 de Salud de Pastaza que utilizarán de ahora en adelante el sistema BRIGAMED.

El modelo de la encuesta se encuentra en el Anexo VI – Encuesta Análisis de Resultados.

N°	Pregunta	Calificación	Observación
1	Sistema intuitivo	3 Muy Bueno 1 Bueno	Los menús de cada funcionalidad del sistema están accesibles y agrupados según su concepto. Los sistemas de creación, modificación, eliminación y búsqueda de los diferentes módulos son fáciles de ubicarlos.
2	Facilidad de uso	4 Muy Bueno	El sistema se basa en el proceso normal de creación de brigadas y conserva el mismo léxico utilizado por el personal del Área.
3	Facilidad de aprendizaje	3 Muy Bueno 1 Bueno	Todos los procesos de creación, eliminación, actualización y búsqueda de los módulos son comunes, por lo cual el usuario al entender el funcionamiento de uno es capaz de realizar el ingreso en todos los módulos.
4	Tiempo de respuesta del sistema	2 Muy Bueno 1 Bueno 1 Malo	Debido a la infraestructura no pudieron observar el mismo tiempo de respuesta en las estaciones de trabajo.
5	Información coherente	4 Muy Bueno	Durante el uso del sistema, los usuarios ingresaron información real para realizar la

			simulación de una brigada, certificando que la información no sufrió alteración alguna.
6	El sistema cumple con lo deseado	3 Muy Bueno 1 Bueno	Esta pregunta permite dar una perspectiva general de la satisfacción del cliente con respecto al uso del sistema Brigamed.

Tabla 3.10 Análisis de Resultados

Los resultados arrojados por esta encuesta han demostrado que el sistema Brigamed cumple con las características para soportar el uso cotidiano del mismo.

En base a la encuesta se puede determinar que el sistema disminuye el tiempo que anteriormente el personal del Área de Salud tardaba en realizar tareas operativas, como búsqueda de archivos y consulta de información a las diferentes dependencias, lo cual implica un aumento en la productividad en la ejecución de las actividades cotidianas.

Además al mantener la información de las comunidades, como la cantidad real de habitantes y los recursos humanos que se encuentran activos, el personal del Área de Salud puede manejar un reparto más eficiente en los recursos que se asignan a las Brigadas.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- El éxito o no del desarrollo de software depende de la interacción del cliente con el equipo de desarrollo, RUP al mantener una completa planificación y documentación de cada reunión, permite tener poco contacto con el cliente y un buen aprovechamiento del tiempo de las reuniones, tan importante con el desarrollo a distancia, como es el proyecto actual.
- El uso de UML para el modelamiento del proyecto, pudo hacer más comprensible la estructura y arquitectura del sistema, al describir gráficamente cada método o proceso del sistema.
- La aplicación de la tecnología JBoss Seam facilitó la elaboración del sistema, ya que permitió la creación de CRUD's con tan solo adaptar los plug-ins necesarios y configurar una conexión a la base de datos, todo esto mediante el asistente para la creación de JBoss Server.
- Para garantizar la calidad del sistema, se puso un cuidado especial en la etapa de pruebas, en las cuales se realizó la mayor cantidad de casos de prueba para abarcar todas las funcionalidades y solucionar la mayor cantidad de errores que se suscitan por el ingreso incorrecto de información.
- La administración de brigadas que se realizaba de forma manual consumía varias horas del personal del Área 2 de Salud de Pastaza con trabajo operativo, dificultando una gestión adecuada de una brigada; al automatizar

los procesos de creación y actualización de brigadas se tornó eficiente el trabajo desempeñado por las autoridades y empleados del Área.

4.2. RECOMENDACIONES

- Se recomienda el uso cotidiano del Sistema BRIGAMED en el Área 2 de Salud de Pastaza, para llevar de una manera más organizada la información de las comunidades, recurso humano, suministros, así como las brigadas que se realizan.
- Se recomienda usar herramientas de software libre para reducir costos de licencias, conocer las vulnerabilidades del código y seguir la regla del desarrollo de aplicaciones para el sector público donde es mandatorio.
- Se recomienda tomar en cuenta el tiempo que toma la capacitación e investigación de la tecnología a utilizar por parte del equipo de desarrollo en la planificación de proyectos de esta índole.
- Se recomienda implementar Sistemas Complementarios a BRIGAMED, como un sistema de Gestión de Inventarios y Sistemas de Información Geográfica que apoyen al personal del Área en sus tareas cotidianas.
- Para realizar de una manera más fácil la recolección de datos se recomienda conocer la naturaleza de la organización para la cual se desarrollará el sistema.

- Se recomienda que tanto el equipo de desarrollo como el cliente fijen un cronograma con fechas y entregables que se respetarán, ya que cada retraso en las tareas puede generar un retraso importante en el proyecto en general.
- Se recomienda utilizar lenguajes de modelamiento como UML en las cuales se tenga experiencia, para evitar complicaciones en la elaboración de los diagramas o modelos del sistema.

BIBLIOGRAFIA

Libros:

- [1] JACBOSON, Ivar; BOOCH, Grady; RUMBAUGH, James. El Lenguaje Unificado de Modelado. Primera Edición. Pearson Education. 2000.
- [2] ARNOLD, Ken; GOSSLING, James; HOLMES, David. The Java Programming Language. Cuarta Edición. Addison Wesley. 2006

Tesis:

- [3] DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN DE INFORMACIÓN DE LA FUNDACIÓN PROYECTO SALESIANO, Chávez Andrea, Guagalango Adriano, Octubre 2008.
- [4] DESARROLLO DE UN SISTEMA DE ADMINISTRACIÓN ESTUDIANTIL PARA EL COLEGIO SEGRÉ USANDO EL DESARROLLO DIRIGIDO POR MODELOS, Calero Gerardo, Enero 2009

Publicaciones Web:

- [5] RUP vs. XP, USMP,
<http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>, 12 de agosto de 2010
- [6] Programación Extrema, WIKIMEDIA FOUNDATION INC,
http://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema,
19 de agosto de 2010
- [7] THE PHP GROUP, PHP, <http://www.php.net/>, 05 de noviembre de 2010
- [8] JBoss Seam, SEAM CITY,
<http://seamcity.madeinxpain.com/archives/comparativa-seam>,
18 de enero de 2011

- [9] GWT, W3 SUPPORT, <http://es.w3support.net/index.php?db=so&id=744515>,
18 de enero de 2011
- [10] SYMFONY, LIBROS WEB, :
http://www.librosweb.es/symfony_formularios/capitulo4/el_generador_crud.html,
18 de enero de 2011
- [11] Sistema de Gestión de Base de Datos, WIKIMEDIA FOUNDATION INC.,
http://es.wikipedia.org/wiki/Sistema_de_gestion_de_bases_de_datos,
04 de febrero de 2011
- [12] PostgreSQL versus MySQL, FERMILAB,
<http://www-css.fnal.gov/dsg/external/freeware/pgsql-vs-mysql.html>,
04 de febrero de 2011
- [13] PostgreSQL, WIKIMEDIA FOUNDATION INC.,
<http://es.wikipedia.org/wiki/PostgreSQL>,
05 de febrero de 2011
- [14] Fases de la Programación Extrema, CASTILLO Oswaldo; FIGUEROA Daniel; SEVILLA Héctor, <http://programacionextrema.tripod.com/fases.htm>,
11 de marzo de 2011

ANEXOS

- I. Plantilla de Especificación de Requerimientos – Anexo Digital
- II. Diagramas de Secuencia – Anexo Digital
- III. Código Fuente – Anexo Digital
- IV. Diagramas de Componentes – Anexo Digital
- V. Kit de Instalación – Anexo Digital
- VI. Encuesta para Análisis de Resultados – Anexo Digital