

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA DE SISTEMAS**

**DESARROLLO UN SISTEMA DE REGISTRO HOSPITALARIO PARA  
SOLCA MANABÍ NÚCLEO DE PORTOVIEJO**

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

**CARLOS GUSTAVO CASTILLO CHANGUAN**

**calini121@yahoo.es**

**CAROLINA TERESA OÑATE BRITO**

**onate\_carolina@yahoo.es**

**DIRECTORA: ING. SHEILA NOBOA**

**gaiashen@hotmail.com**

**Quito, Febrero 2010**

## **DECLARACIÓN**

Nosotros, Carlos Gustavo Castillo Changuan y Carolina Teresa Oñate Brito, 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.

---

**Carlos Castillo**

---

**Carolina Oñate**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Carlos Gustavo Castillo Changuan y Carolina Teresa Oñate Brito, bajo mi supervisión.

---

**Ing. Sheila Noboa**

**DIRECTORA DE PROYECTO**

## **DEDICATORIA**

El presente Proyecto de Titulación va dedicado a mis padres, hermanos y mis sobrinos/as quienes de una u otra forma me han ayudado, apoyado, comprendido y enseñado a luchar día a día para alcanzar mis propósitos y metas.

A Dios quien siempre ha estado presente en mi vida siendo un soporte y llenándome de satisfacciones y bendiciones.

**Carlos**

## DEDICATORIA

Este proyecto de Titulación está dedicado a todas las personas de una u otra manera me apoyaron de manera incondicional, con sus palabras de aliento cuando lo necesitaba y así poder cumplir con esta meta que tanto anhele.

A mi mamá, a mi abuelita y a mis hermanos, que siempre me han apoyado y han estado a mi lado en todo momento ayudándome a superar cualquier dificultad que se ha presentado en mi vida

A mi tío Pepe y mi tía Aurora que gracias a ellos he podido superarme y que han sabido darme palabras de aliento y sus consejos.

A Carlos que siempre ha estado a mi lado de manera incondicional, que me ha apoyado en todo lo que hemos emprendido juntos.

A mi hija Camila que es el pilar de mi vida y que es por quien sigo luchando para salir adelante, que es mi fuerza y mis ganas de superarme.

A Dios y a la Virgencita que me bendicen día a día dándome salud y vida para culminar con todas las metas que me proponga

**Carolina**

## **AGRADECIMIENTOS**

Para realizar este proyecto hubo muchas personas que me apoyaron directa e indirectamente, sin su ayuda no hubiera sido posible la terminación del mismo.

Agradezco a mi madre, mi hermana, por todo su apoyo recibido a lo largo de mi carrera, siempre estando pendiente de mis necesidades y nunca dejando que me descuide de mis obligaciones del día a día.

A mis hermanos, mí cuñada Ana y mis sobrinos/as, quien siempre ha estado a mi lado acompañándome en cualquier desafío o travesía que se me ocurra enfrentar.

A mi esposa Carolina, quién ha estado a mi lado en las buenas y en las malas ayudándome a sobrellevar todo de la mejor manera.

A Jaime Vinuesa, por su ayuda desinteresada y sus conocimientos para comenzar a direccionar mi profesión.

A mi hija Camila que me da cada día la fortaleza para seguir superándome en mi vida, ya que es la alegría de mi vida.

A la ingeniera Sheila Noboa por su guía en la realización de este proyecto.

A todos mis amigos de la Universidad muchas gracias.

**Carlos**

## AGRADECIMIENTOS

Quiero agradecer primero a mi mamá y mi abuelita quienes me apoyaron en todas las etapas de mi vida y que me supieron orientar con sus consejos, a mis hermanos que me han ayudado y me han apoyado en todo lo que han podido.

A mi Tío Pepe y mi Tía Aurora que me han apoyado siempre en todo y que gracias a ellos y podido obtener este título y culminar esta etapa en mi vida, y que son muy especiales en mi vida.

A Sheila por su ayuda y guía en este proceso y en nuestra vida personal, a su familia que también nos ha ayudado y apoyado en todo momento.

A Carlos que a más de ser compañero en esta etapa ha sido mi compañero en la vida que esta siempre a mi lado con su amor y cariño incondicional y que gracias a esto hemos podido culminar juntos esta etapa.

A todos mis amigos que en su momento supieron con sus palabras de aliento ayudarme en momentos difíciles y superar todos los obstáculos que se nos presentaron.

A Dios por sus bendiciones y por darme mi hermosa hija Camila que a pesar de ser tan pequeña con una sonrisa me ayuda a superar cualquier dificultad.

Gracias a todos.

***Carolina***

## CONTENIDO

CAPITULO I. INTRODUCCIÓN.....	1
1.1    OBJETIVOS DEL PROYECTO.....	1
1.1.1    OBJETIVO GENERAL.....	1
1.1.2    OBJETIVOS ESPECÍFICOS.....	1
1.2    JUSTIFICACIÓN DEL PROYECTO.....	2
1.3    DESCRIPCIÓN DEL PROBLEMA.....	2
CAPÍTULO II.....	4
ASPECTOS METODOLÓGICOS Y HERRAMIENTAS A USAR.....	4
2.1    ASPECTOS METODOLÓGICOS.....	4
2.1.1    PROCESO UNIFICADO DE DESARROLLO.....	4
2.1.2    LENGUAJE UNIFICADO DE MODELADO (UML).....	5
2.1.3    GUÍA PARA EL DESARROLLO DEL SISTEMA.....	7
2.1.4    ESTÁNDARES DE PROGRAMACIÓN.....	11
2.2    HERRAMIENTAS A USAR.....	14
CAPITULO III.    DESARROLLO DE LA SOLUCIÓN.....	16
3.1.    OBJETIVOS DEL SISTEMA.....	16
3.2    ALCANCE DEL SISTEMA.....	16
3.3    MODELAMIENTO DEL SISTEMA.....	17
3.3.1    MODELAMIENTO DEL NEGOCIO.....	17
3.3.2    MODELO DE REQUERIMIENTOS.....	19
3.3.2.1    Matriz Procesos Administrativos incluyen requerimientos.....	19
3.3.2.2    Lista de Requerimientos funcionales.....	21
3.3.2.3    Lista de Requerimientos no funcionales.....	21

3.3.3	MODELO DE ANÁLISIS .....	24
3.3.4	MODELO DE DISEÑO .....	27
3.3.4.1	Diseño Funcional.....	27
3.3.4.2	Diseño Estructural .....	72
3.3.4.3	Diseño Dinámico .....	74
3.3.5	MODELO DE DESPLIEGUE .....	87
3.4	IMPLEMENTACIÓN DEL SISTEMA .....	89
3.5	DOCUMENTACIÓN DE PRUEBA PILOTO DE IMPLANTACIÓN .....	91
3.5.1	VALIDACIÓN DEL CUMPLIMIENTO DE REQUERIMIENTOS FUNCIONALES.....	91
3.5.1.1	Validación de Requerimientos Funcionales vs Diseño Funcional ...	91
3.5.1.2	Validación del Software .....	93
3.5.2	VALIDACIÓN DEL CUMPLIMIENTO DE REQUERIMIENTOS NO FUNCIONALES.....	101
3.5.2.1	Recuperación .....	101
3.5.2.2	Rendimiento .....	102
3.5.2.3	Multiusuario .....	104
3.5.2.4	Facilidad de uso .....	104
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES .....		105
4.1	CONCLUSIONES .....	105
4.2	RECOMENDACIONES.....	107
BIBLIOGRAFÍA.....		108

## **ANEXOS**

**Anexo Cap II. A1:** Características del PUD

**Anexo Cap II. A2:** Punto De Vista De Procesos Del PUD

**Anexo Cap II. A3:** Punto De Vista Metodológico Del PUD

**Anexo Cap II A4:** Diagramas UML

**Anexo Cap III A1:** Descripción de Requerimientos

**Anexo Cap III A2:** Prototipo de Interfaces

**Anexo Cap III A3:** Diccionario de datos

## **TABLAS**

**Tabla 1:** Diagrama vs Procesos

**Tabla 2:** Guía para el desarrollo del sistema

**Tabla 3:** Matriz de procesos administrativos

**Tabla 4:** Matriz de procesos administrativos incluye requerimientos funcionales

## RESUMEN

Este Proyecto se desarrollo bajo principio de ingeniería de software para preparar un sistema que recoge los datos relevantes de exámenes y tratamientos de tumores malignos de pacientes atendidos en SOLCA Manabí Núcleo de Portoviejo, con el fin de obtener información relacionada a la calidad de atención al paciente y a la calidad del diagnóstico que apoyen tanto al personal médico como al personal administrativo.

El Capitulo I presenta una introducción que permitirá al lector ubicarse en el tema

El Capítulo II Los aspectos metodológicos que guían al desarrollo con una orientación a objetos (OO) y hace referencia a las herramientas a utilizar.

El Capítulo III describe los pasos que se ejecutaron para el desarrollo de la aplicación práctica, detallando los productos que se fueron obteniendo en cada paso.

El Capítulo IV presenta las conclusiones y recomendaciones.

# **CAPITULO I. INTRODUCCIÓN**

## **1.1 OBJETIVOS DEL PROYECTO**

### **1.1.1 OBJETIVO GENERAL**

Desarrollar un sistema de Registro Hospitalario para SOLCA Manabí Núcleo de Portoviejo que respete estándares de recolección de datos establecidas por las normas de los registros hospitalarios y que facilite la entrega de información a personal médico, personal técnico y autoridades del Hospital como soporte a las decisiones que procuren el mejoramiento de la calidad del servicio.

### **1.1.2 OBJETIVOS ESPECÍFICOS**

Recopilar los requerimientos de información que tengan los médicos-técnicos, los registradores hospitalarios y las autoridades respecto de: los casos tratados, del nivel y de la calidad del servicio brindado en el hospital.

Analizar y diseñar un software que satisfaga los requerimientos mencionados.

Implementar el software diseñado considerando los criterios y códigos internacionales para las diferentes variables que se recopilen.

Implantar el software utilizando casos que entregue el hospital para una prueba piloto.

## **1.2 JUSTIFICACIÓN DEL PROYECTO**

Es preocupación de médicos y autoridades de SOLCA Manabí Núcleo de Portoviejo el evaluar el nivel y calidad del servicio que presta a los pacientes. Esta evaluación se dificulta en vista de que en los últimos años ha incrementado el número de pacientes y se imposibilita hacer el tratamiento de estos datos manualmente o con ayuda de paquetes de escritorio, puesto que no cuentan con un sistema especializado para el procesamiento.

El personal de Registro Hospitalario ha expresado insistentemente la necesidad de contar con un sistema computarizado para registrar los casos y preparar información confiable, consistente para la toma de decisiones orientada al mejoramiento de la calidad en la atención al paciente.

Los datos que requieren procesar de los casos se relacionan, entre otros, con: identificación del paciente, contactos para ubicarlo, antecedentes; características del tumor; exámenes para llegar al diagnóstico y estadiaje; tratamientos recibidos; seguimiento.

## **1.3 DESCRIPCIÓN DEL PROBLEMA.**

El registro de datos de las atenciones a los pacientes en todos los servicios de SOLCA se los lleva detalladamente y de manera transaccional. Estos datos son demasiado específicos, disgregados y orientados a cada uno de los servicios.

Estos datos no son adecuados para conocer al paciente de una manera integral que permita:

- Llevar un control específico de la evolución de su enfermedad.
- Resumir los tratamientos que recibe el paciente por cada tumor que se le diagnosticará.

- Hacer seguimiento de las atenciones recibidas con el fin de cuidar que el paciente no abandone su tratamiento.

Estos datos tampoco son adecuados para:

- Obtener estadísticas para conocer la cobertura del servicio de SOLCA.
- Obtener listados relacionados al paciente, tumor, tratamiento y seguimiento con el fin de brindar un mejor servicio al paciente.

Por lo antes expuesto SOLCA requiere contar con sistema de registro hospitalario que básicamente:

- Registre datos de paciente.
- Registre datos de tumor.
- Registre datos de tratamiento.
- Registre datos de seguimiento.
- Registro de tablas códigos estandarizados.
- Emita listados de pacientes relacionados al seguimiento, tumor y tratamiento.
- Emita estadísticas relacionados al seguimiento, tumor y tratamiento.

## CAPÍTULO II.

### ASPECTOS METODOLÓGICOS Y HERRAMIENTAS A USAR

#### 2.1 ASPECTOS METODOLÓGICOS

Para el modelamiento del sistema se basó en el Lenguaje Unificado de Modelado (UML) y en Proceso Unificado de Desarrollo (PUD).

##### 2.1.1 PROCESO UNIFICADO DE DESARROLLO

El Proceso Unificado de desarrollo es un marco de trabajo genérico que permite adaptabilidad a cualquier organización o sistemas de software sin importar el tamaño del sistema.

El Proceso Unificado define los roles, tiempo y actividades que se usarán mediante el desarrollo de un producto de software, para que este sea efectivo y disminuyan considerablemente los riesgos.

El Proceso Unificado de Desarrollo Software se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental para mayor detalle sobre las características del PUD **ver Anexo Cap II A1.**

El PUD permite la gestión del proyecto, es decir actúa como proceso de control del avance del proyecto, marcando iteraciones por al menos una vez por cada fase del PUD: inicio, elaboración, construcción y transición, el detalle de estas fases consta en el **Anexo Cap II A2 “Punto de vista de procesos del PUD”.**

El PUD se aplica también como una guía metodológica que establece las actividades y los productos que los desarrolladores deben conseguir con la ejecución de cada uno de los flujos de trabajo el detalle de estos flujos de trabajo consta en el **Anexo Cap2 A3 “Punto de vista metodológico del PUD”**.

### 2.1.2 LENGUAJE UNIFICADO DE MODELADO (UML)

Lenguaje Unificado de Modelado (Unified Modeling Language) es un lenguaje gráfico para la especificación, visualización, construcción y documentación de modelos producidos durante el proceso de desarrollo de software.

UML provee un marco arquitectónico de diagramas para trabajar sobre análisis y diseño orientado a objetos, así como también el modelamiento de negocios y otros sistemas que no son software.

El UML consta con varios tipos de diagrama que permiten el modelamiento de los siguientes aspectos:

DIAGRAMA	PARA MODELAR	
Actividades	Procesos Administrativos	Negocio
Casos de uso	Procesos	Funcionalidad
Clases	Datos	Estructura
Objetos	Datos	
Actividades	Controles	Dinámica
Secuencia	Controles	
Colaboración	Controles	
Estados	Controles	

Componentes	Despliegue	Instalación
Despliegue	Despliegue	

**Tabla 1: Diagrama vs Procesos**

Para conocer las características de los diagramas mencionados se presenta en el **Anexo Cap2 A4** el detalle de las mismas.

### 2.1.3 GUÍA PARA EL DESARROLLO DEL SISTEMA

A continuación se presenta una matriz que en este proyecto se utilizará como guía metodológica para desarrollo del sistema, respeta la estructura del PUD. Considera sus flujos de trabajo y en ella se establecen las actividades que se van realizando y los productos que se obtienen a lo largo del tiempo es decir conforme se cumplen las fases.

En cada porcentaje habrá un margen de error de más menos 10% ya que se dedujo debido a la experiencia en otros proyectos realizados y a la interpretación a las curvas del PUD.

FASES		INICIACIÓN	ELABORACIÓN	CONSTRUCCIÓN	TRANSICIÓN
FLUJOS DE TRABAJO	ACTIVIDADES	PRODUCTOS	% de Madurez de los entregables en cada fase		
	<b>Modelo del Negocio</b>		<ul style="list-style-type: none"> <li>Reconocer los procesos administrativos relacionados al negocio para el que se va a desarrollar el software.</li> <li>Hacer un seguimiento de las actividades relacionadas al negocio que se va a modelar.</li> <li>Validar modelo del negocio.</li> </ul>	<ul style="list-style-type: none"> <li>Matriz procesos administrativos</li> </ul>	90%
<b>Requisitos</b>	<ul style="list-style-type: none"> <li>Para cada proceso pertinente del negocio reconocer los requisitos funcionales (opciones que pueden constar en el</li> </ul>	<ul style="list-style-type: none"> <li>Matriz procesos administrativos incluye requerimientos funcionales</li> </ul>	80%	100%	

	<p>software que faciliten la ejecución de la actividad)</p> <ul style="list-style-type: none"> <li>• Obtener la lista de requisitos funcionales reconocidos en el modelo del negocio.</li> <li>• Especificar los requerimientos funcionales</li> <li>• Definir los requerimientos no funcionales.</li> </ul>	<ul style="list-style-type: none"> <li>• Lista de requerimientos no funcionales.</li> <li>• Lista de requerimientos funcionales</li> <li>• Diagrama de especificación de requerimientos.</li> </ul>	80%	100%		
			80%	100%		
			80%	100%		
<b>Análisis</b>	<ul style="list-style-type: none"> <li>• En base a los requerimientos se agrupan de manera consistente para definir los módulos del sistema.</li> <li>• Con los usuarios se define las clases de datos principales del sistema.</li> </ul>	<ul style="list-style-type: none"> <li>• Diagrama de actores</li> <li>• Diagrama de casos de uso a nivel contextual.</li> <li>• Diagrama de clases de análisis.</li> </ul>	100%			
			90%	100%		
			90%	100%		
<b>Diseño</b>	<ul style="list-style-type: none"> <li>• Se transforma los requerimientos principales del sistema en casos de uso básicos y se relaciona el resto de requerimientos a los casos de uso diferenciando casos de uso incluye o extend según sea el caso.</li> <li>• Se describe la lógica</li> </ul>	<p><b>Modelo Funcional</b></p> <ul style="list-style-type: none"> <li>• Diagrama de casos de uso por módulo</li> <li>• Descripción de casos de uso</li> </ul>	10%	80%	95%	100%
			10%	80%	95%	100%
		<p><b>Modelo Estructural</b></p> <ul style="list-style-type: none"> <li>• Diagrama de clases de diseño.</li> <li>• Diccionario de datos del</li> </ul>	10%	95%	97%	100%
				85%		

	<p>básica de los casos de uso.</p> <ul style="list-style-type: none"> <li>• Se detallan las clases de datos</li> <li>• Se reconocen los atributos para cada clase de datos.</li> <li>• Se relacionan las clases de datos según las reglas del negocio.</li> <li>• Desde el punto de vista de controles se reconocen los casos de uso complejos y para cada uno se especifica su lógica.</li> <li>• Se reconoce los estados que puedan adquirir los objetos complejos</li> <li>• Se reconoce la interrelación de los objetos y sus funcionalidades.</li> <li>• Se define los componentes a desarrollar e instalar.</li> </ul>	<p><b>diagrama de clases.</b></p> <p><b>Modelo Dinámico</b></p> <ul style="list-style-type: none"> <li>• Diagrama de actividades de casos de uso complejos.</li> <li>• Diagrama de estados de objetos que cambian de estados.</li> <li>• Diagrama de secuencia de objetos relacionados a través de funcionalidad.</li> </ul> <p><b>Modelo de Despliegue</b></p> <ul style="list-style-type: none"> <li>• Diagrama de componentes.</li> <li>• Diagrama de despliegue.</li> </ul>				
				90%	95%	100%
				90%	95%	100%
				90%	95%	100%
				90%	100%	
				90%	100%	
<b>Implementación</b>	<ul style="list-style-type: none"> <li>• Documentar Estándares de programación</li> <li>• Codificar en base al</li> </ul>	<ul style="list-style-type: none"> <li>• Estándares de Programación</li> <li>• Código del Sistema</li> </ul>		100%		100%
				40%	90%	100%

	<p>diseño.</p> <ul style="list-style-type: none"> <li>Realizar pruebas unitarias y de integración considerando el desarrollo del software.</li> <li>Preparar documentación orientada al usuario.</li> </ul>	<ul style="list-style-type: none"> <li>Código ejecutable del Sistema</li> <li>Manual de Usuario.</li> </ul>		40%	100%	
<b>Pruebas</b>	<ul style="list-style-type: none"> <li>Validar requisitos frente al análisis y diseño.</li> <li>Realizar pruebas de componentes.</li> <li>Validar el código frente al diseño.</li> </ul>	<ul style="list-style-type: none"> <li>Documentación de las pruebas.</li> </ul>	20%	60%	90%	100%
<b>Despliegue</b>	<ul style="list-style-type: none"> <li>Instalar el sistema</li> <li>Desarrollar documentación para instalación del sistema.</li> </ul>	<ul style="list-style-type: none"> <li>Código ejecutable instalado</li> </ul>				100%

**Tabla 2: Guía para el desarrollo del sistema**

#### 2.1.4 ESTÁNDARES DE PROGRAMACIÓN

Los estándares que deberán ser utilizados en la programación de este sistema hacen referencia a:

Estándares para asignar nombres:

- Tablas y atributos de la base de datos
- Clases y atributos de la clase
- Funciones
- Objetos de programación en Interfaces

Estándar para estructurar el proyecto:

A continuación se detallan estos estándares.

#### ESTÁNDARES PARA ASIGNAR NOMBRES

**Tablas y atributos de la base de datos:** Se nombrará a las tablas y a los atributos con palabras describan el contenido de las mismas. Si alguna tabla requiere ser nombrada con dos o más palabras serán separadas con un guión bajo.

**Clases y atributos de la clase:** Se nombrará a las clases con palabras que representan lo que hacen con la primera letra en mayúscula y las demás letras en minúsculas; si la clase consta de dos o más palabras la primera letra de las palabras posteriores palabra comienzan con mayúsculas. En el caso de los atributos de las clases se nombra en minúscula y en caso de tener dos o más palabras la primera letra de las siguientes palabras empieza con mayúsculas. En el caso de que el atributo sea de tipo identificadores estos serán de tipo numérico y su nombre debe comenzar con "id" seguido del nombre de la tabla, respetando las reglas para nombrar atributos. Ejemplo: idPaciente.

**Funciones:** Todas las funciones deben tener un nombre relacionado con la función que desempeñan y si consta de dos o más palabras las posteriores palabras comenzarán con letra en mayúsculas. Ejemplo: guardarPaciente.

**Objetos de programación en interfaces:**

- Para las cajas de texto (textbox) se asignará un nombre que se relacione a su contenido antepuesto la palabra “txt”, cada palabra usada para este nombre debe comenzar con mayúsculas. Ejemplo: txtNombrePaciente.
- Para los membretes (label) se utilizará la palabra “lbl” seguido de las palabras necesarias para describirlo con la que cada palabra será con mayúsculas.
- Para los combos la palabra “cmb” seguido del nombre del combo con la primera letra en mayúsculas y si consta de dos o más palabras la primera letra en mayúsculas.
- Para los CheckList la palabra “chk” seguido del nombre con la primera letra en mayúsculas y si consta de dos o más palabras la primera letra en mayúsculas.
- Para los botones se asigna el nombre antepuesto de la palabra “btn” y si consta de dos o más palabras las posteriores palabras comenzarán con letra en mayúsculas

**ESTÁNDAR PARA ESTRUCTURAR EL PROYECTO**

El proyecto deberá estar organizado en una carpeta que haga las veces de área de trabajo (workspace), el nombre de esta carpeta debe ser el nombre del proyecto.

El proyecto constará de la siguiente estructura:

- main
  - dao
  - daoHibernate
  - domain
  - dto

- services
- serivicesImpl
- web
- conf
- lib
- src

**main** que contendrá las clases que realizan las funciones de conectores entre la base de datos y el sistema o el sistema hacia las paginas y contiene a su vez las subcarpetas:

- **dao**: contendrá las clases con las descritas las funciones que va a realizar cada clase.
- **daoHibernate**: contendrá el desarrollo (select, insert, update, etc) de las funciones que va a desarrollar la clase.
- **domain**: contendrá las clase con el mapeo de las mismas es decir la conexión entre la base de datos y el sistema
- **dto**: contendrá las clases y con los nombres que serán utilizados para ser llamados en las jsp.
- **services**: contendrá las descripciones de las funciones que serán llamadas desde los action.
- **serivicesImpl**: contendrá las funciones que serán invocadas desde un action para ser ejecutadas en el daoHibernate.
- **web** que contendrá las clases que realizan las funciones de acciones invocadas desde una página para ser ejecutas.
- **conf**: constará los archivos de conexión del sistema con la base de datos para su interrelación.
- **lib**: contendrá las librerías que se utilizaran para el sistema.
- **src**: contendrá las páginas de java script (jsp) desarrolladas que se utilizaron en el sistema.

## 2.2 HERRAMIENTAS A USAR

Como herramienta case para el moldeamiento del sistema se empleará Enterprise Architect, ya que ésta ofrece soporte para el modelamiento basado en el proceso Unificado de Desarrollo (PUD o RUP) y el lenguaje de Modelamiento Unificado (UML) y se dispone a nivel académico de demos.

*Se escogió desarrollar en plataforma de software libre, porque SOLCA es una institución pública y se debe regir al Decreto ejecutivo 1040 que básicamente se refiere a:*

*“Software Libre.- Las empresas públicas en función de sus requerimientos tecnológicos harán uso, desarrollarán o contratarán según sea el caso, sistemas informáticos basados en software no propietario (software libre), y podrán trabajar con software propietario cuando no exista solución y no sea posible desarrollarla en software libre.”*

Se utilizará la arquitectura J2EE y Eclipse como ambiente de desarrollo por considerar que es software gratuito y conocemos como codificar en esta herramienta, seleccionando como herramientas para codificación:

- Java
- Jasper Reports como código de reportes e iReport para diseñar reportes.
- Html dentro de las jsp (java script page) para diseño de las páginas.
- Java script dentro de las jsp para las validaciones en el ambiente web.
- Ajax para las validaciones especiales en las jsp para evitar la recarga completa de las páginas.
- Hibernate como capa de acceso a datos.

Para base de datos se eligio:

- Oracle XE para base de datos, por que SOLCA tiene licencias para este motor de basey asi no incurrirá en gastos.

## **CAPITULO III. DESARROLLO DE LA SOLUCIÓN**

El desarrollo de la solución se lo ejecuto basado en la Guía para el desarrollo del sistema que consta en el Capítulo II.

Durante el desarrollo se fue controlando el avance del trabajo y la madurez del producto.

Todos los productos que se presentan en este documento son productos finales es decir que tienen una madurez del 100%.

### **3.1. OBJETIVOS DEL SISTEMA**

En reuniones de trabajo con personal del Registro Hospitalario, aplicando técnicas de entrevista y lluvia de ideas se establecen los objetivos y el alcance del sistema “*Sistema de Registro Hospitalario para SOLCA Manabí Núcleo de Portoviejo*” con el fin de determinar hacia donde está encaminado el desarrollo del mismo.

El objetivo del sistema es apoyar al Registro Hospitalario en la recopilación de datos de pacientes con cáncer tratados en SOLCA, de sus tumores, tratamientos y seguimientos.

### **3.2 ALCANCE DEL SISTEMA**

Desde el punto de vista tecnológico el sistema es en plataforma web pero SOLCA verá cuando lo pone en un servidor web.

Desde el punto de vista administrativo el sistema será para uso del Registro Hospitalario, administración y gestión, para consulta del personal técnico médico de SOLCA.

Desde el punto de vista funcional el sistema se orientará a los requerimientos funcionales del Registro Hospitalario, y no formará parte del sistema integrado.

### **3.3 MODELAMIENTO DEL SISTEMA.**

A continuación se documenta las actividades realizadas en base a la matriz previamente presentada en el Capítulo II “GUÍA PARA EL DESARROLLO DEL SISTEMA”.

Los productos intermedios que se van preparando se discuten con el personal usuario del sistema pero no se presentan como documento del presente proyecto de titulación, constan únicamente los productos finales de cada flujo de trabajo.

#### **3.3.1 MODELAMIENTO DEL NEGOCIO**

Se reconocieron los procesos administrativos que se ejecutan dentro del registro hospitalario.

Se revisaron las tareas que realiza el personal para cada uno de los procesos administrativos, reconociendo específicamente cuales eran las actividades de responsabilidad de cada uno de los cargos.

Se reconoció el procedimiento de los procesos mediante la explicación dada por el personal y la realización de una matriz donde se reflejan los responsables y las fuentes de datos para cada uno de esos procesos.

En la misma matriz se reconocen los procesos informáticos que apoyarían la solución de los procesos administrativos

### MATRIZ DE PROCESOS ADMINISTRATIVOS

PROCESOS	Obtención archivo unificado fuentes principales por periodo	Búsqueda de casos de fuentes secundarias	Eliminación duplicados en periodo	Confrontación y eliminación duplicados con base	Elaboración archivos suspensos	Ingreso de Casos	Seguimiento de pacientes	Control de calidad	Elaboración de Informes
RESPONSABLES	Registrador	Registrador	Registrador	Registrador	Registrador	Registrador	Seguimiento	Coordinador	Coordinador
FUENTE DE DATOS		Quimio ambulatoria	Casos recopilados en fuentes secundarias	Base histórica de casos	Casos sin duplicados	Casos suspensos	Contactos con pacientes y familiares	Casos en el Registro Hospitalario	Casos en el Registro Hospitalario
		Radioterapia		Casos del periodo sin duplicados			Casos en Registro Hospitalario	Historia clínicas pacientes	
		Laboratorio clínico							
		Genética							

Tabla 3: Matriz de procesos administrativos

### **3.3.2 MODELO DE REQUERIMIENTOS.**

Se reconocieron los requisitos funcionales necesarios para cada proceso del negocio que requiere apoyo informático.

Se reconocieron los requisitos funcionales ligados a cada proceso.

Se obtuvo la lista de requisitos funcionales.

Se especificó junto con el personal del Registro Hospitalario cuales son los requerimientos funcionales.

De igual manera se determinaron los requerimientos no funcionales relacionados al sistema.

#### **3.3.2.1 Matriz Procesos Administrativos incluyen requerimientos**

### MATRIZ DE PROCESOS ADMINISTRATIVOS INCLUYE REQUERIMIENTOS FUNCIONALES

PROCESOS	Obtención archivo unificado fuentes principales por periodo	Búsqueda de casos de fuentes secundarias	Eliminación duplicados en periodo	Confrontación y eliminación duplicados con base	Elaboración archivos suspensos	Ingreso de Casos	Seguimiento de pacientes	Control de calidad	Elaboración de Informes
RESPONSABLES	Registrador	Registrador	Registrador	Registrador	Registrador	Registrador	Seguimiento	Coordinador	Coordinador
REQUERIMIENTOS FUNCIONALES A NIVEL GENERAL						Coordinador			
						Registrar Paciente.			Generar reportes:Pacientes según estado seguimiento.
						Registrar Tumor.			Generar reportes:Pacientes según fecha pase control
						Registrar Tratamiento.			Generar reportes:Número de casos según tumor por estado seguimiento.
						Registrar Seguimiento.			Generar reportes:Número de casos según tumor por sexo.
									Generar reportes:Número de casos según tumor por tipo tratamiento.

Tabla 4: Matriz de procesos administrativos incluye requerimientos funcionales

### **3.3.2.2 Lista de Requerimientos funcionales**

A continuación se presenta la lista de requerimientos funcionales planteados por parte del personal médico, técnico.

- Registrar Paciente.
- Registrar Tumor.
- Registrar Tratamiento.
- Registrar Seguimiento.
- Generar reportes:
  - Reportes
    - Pacientes según estado seguimiento.
    - Pacientes según fecha pase control
    - Número de casos según tumor por estado seguimiento.
    - Número de casos según tumor por sexo.
    - Número de casos según tumor por tipo tratamiento.

### **3.3.2.3 Lista de Requerimientos no funcionales**

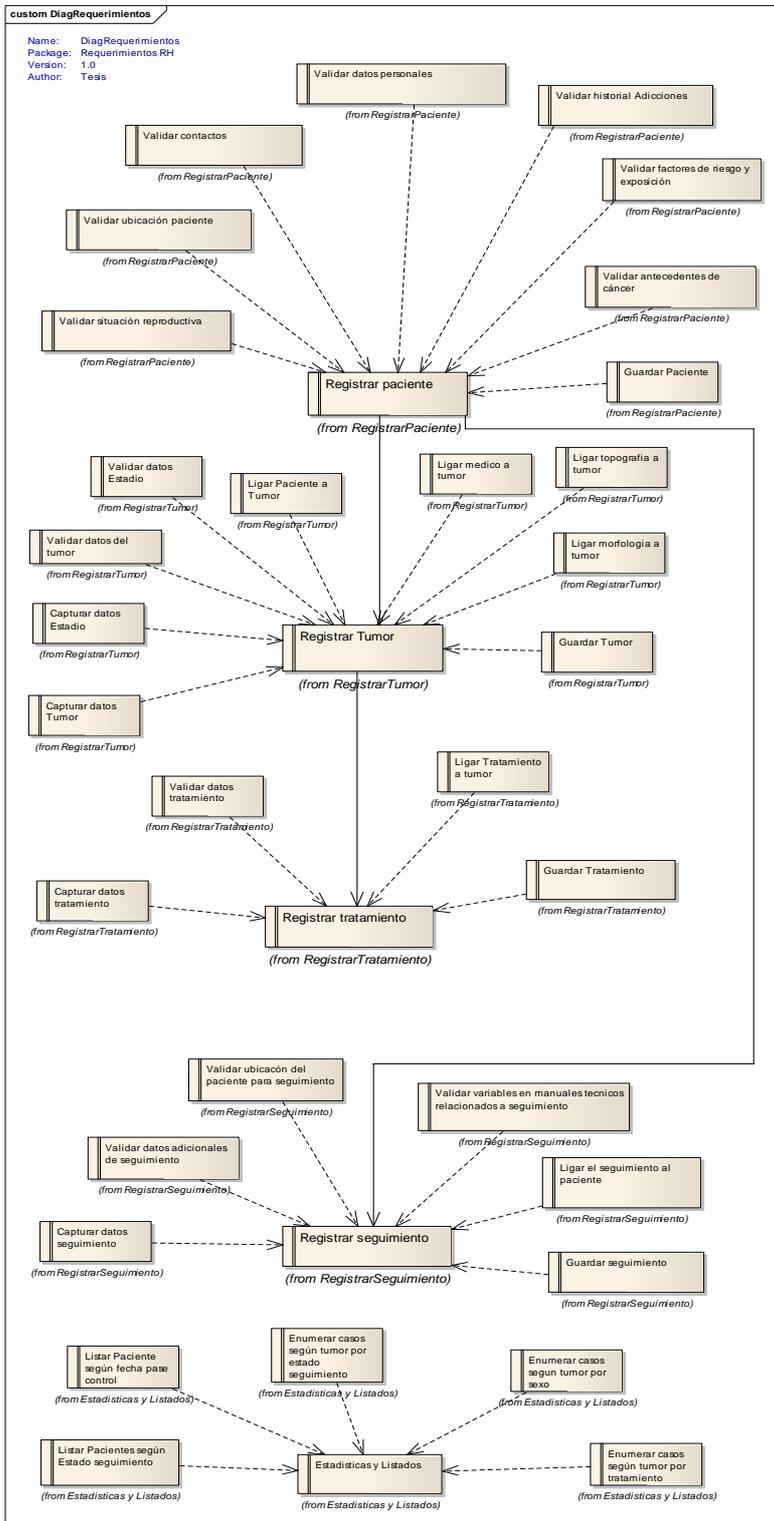
- A prueba de falla de energía eléctrica.
- Que se pueda recuperar los datos al dañarse el servidor total o parcialmente.
- Que el consumo de recursos del computador por parte del sistema no sea excesivo y que no afecte el correcto funcionamiento de otras aplicaciones.
- Multiusuario.
- Interfaces de usuario sencillas y fáciles de operar.

## **Especificación de Requerimientos**

Por cada uno de los requerimientos funcionales reconocidos con los usuarios se especificaron a mayor detalle con el fin de reconocer las funcionalidades, los atributos, las fuentes de datos y las formas de validación que deben aplicarse según las reglas del negocio.

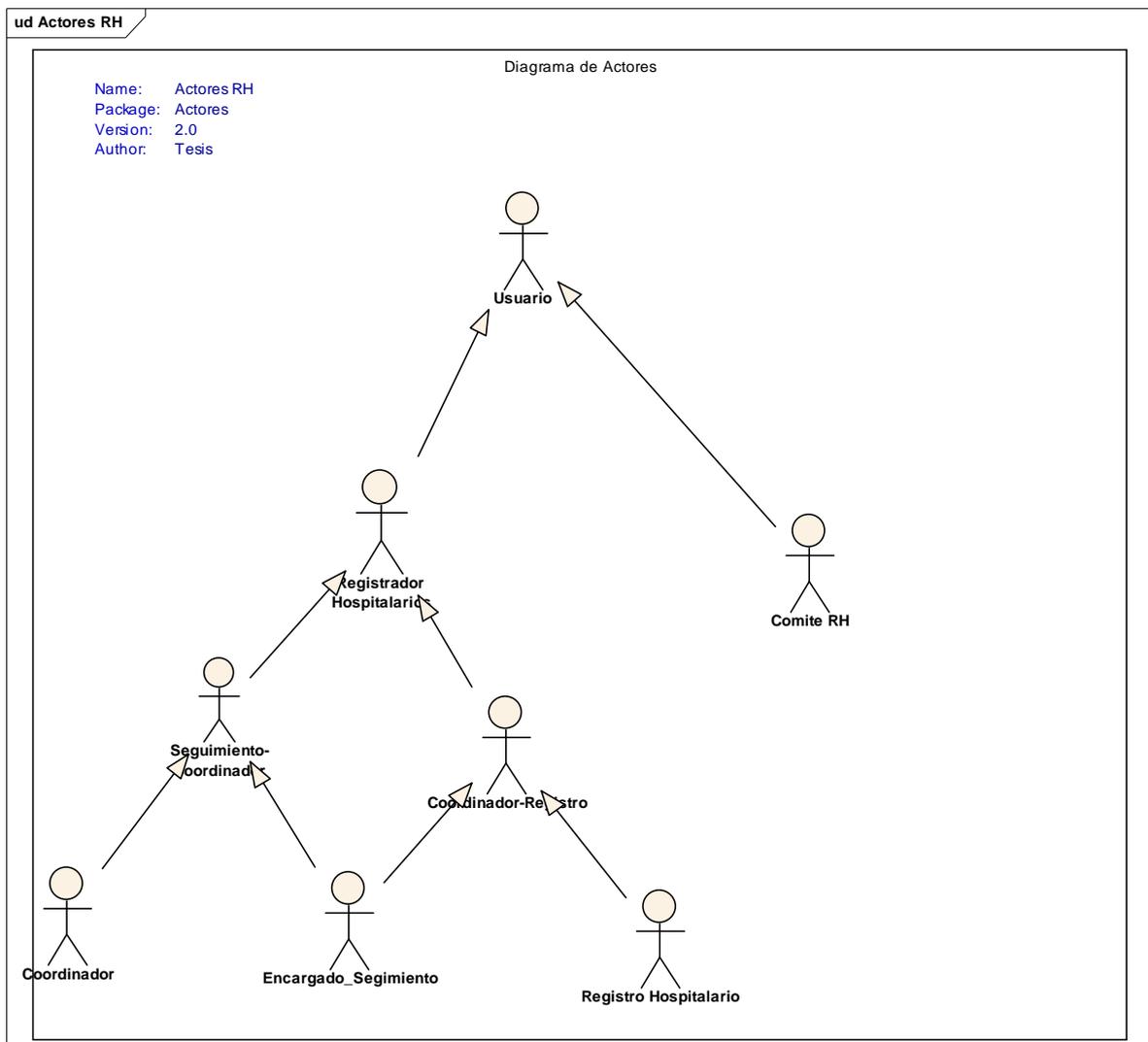
De la explotación de los requerimientos funcionales se consigue detallar los mismos lo cual se puede observar en el siguiente diagrama.

Adicionalmente la descripción de cada requerimiento consta en el **Anexo CapIII A1**.

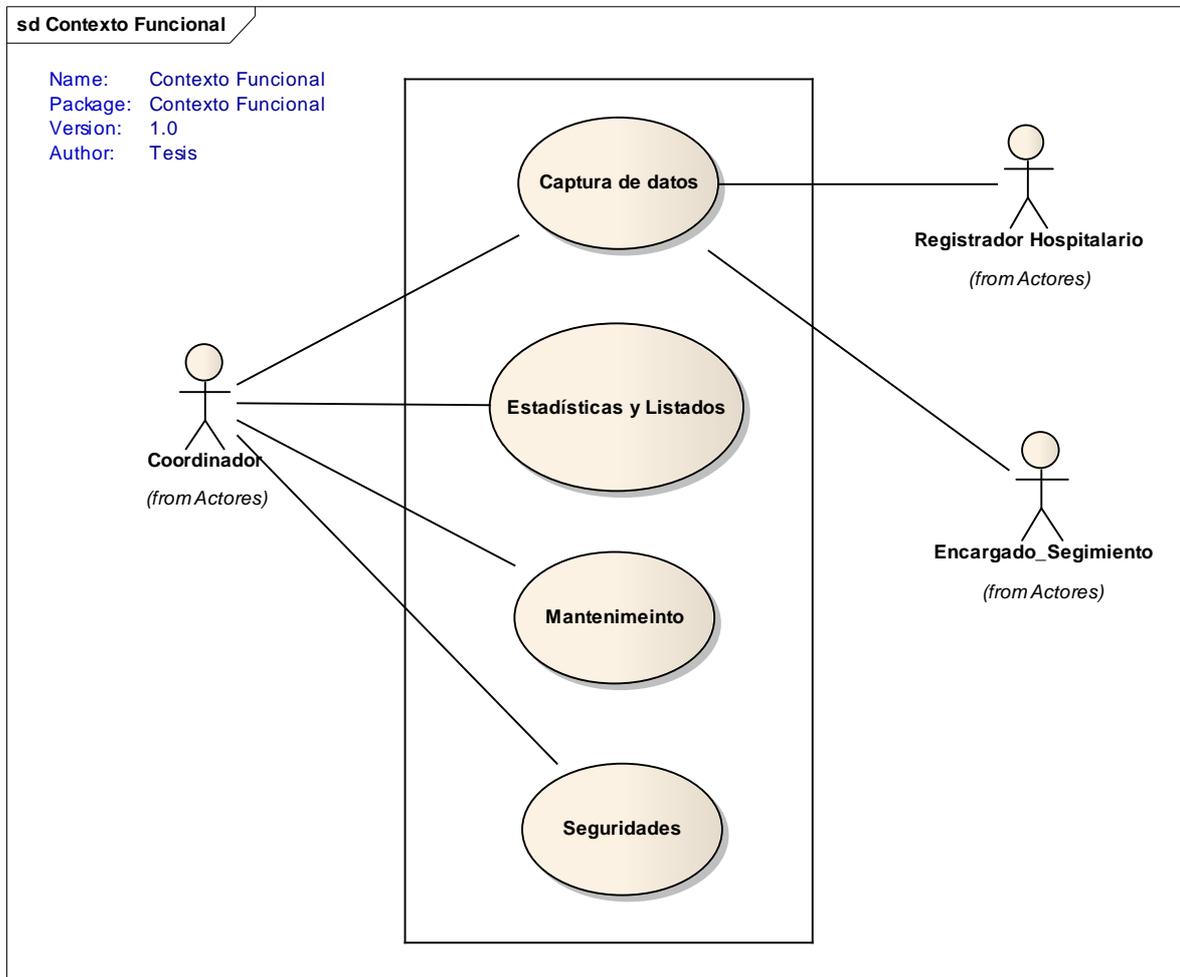


### 3.3.3 MODELO DE ANÁLISIS

Se reconocieron los usuarios de sistema en base de los roles o cargos responsables que constan en el diagrama de actividades, esto se determina mediante el diagrama de actores.



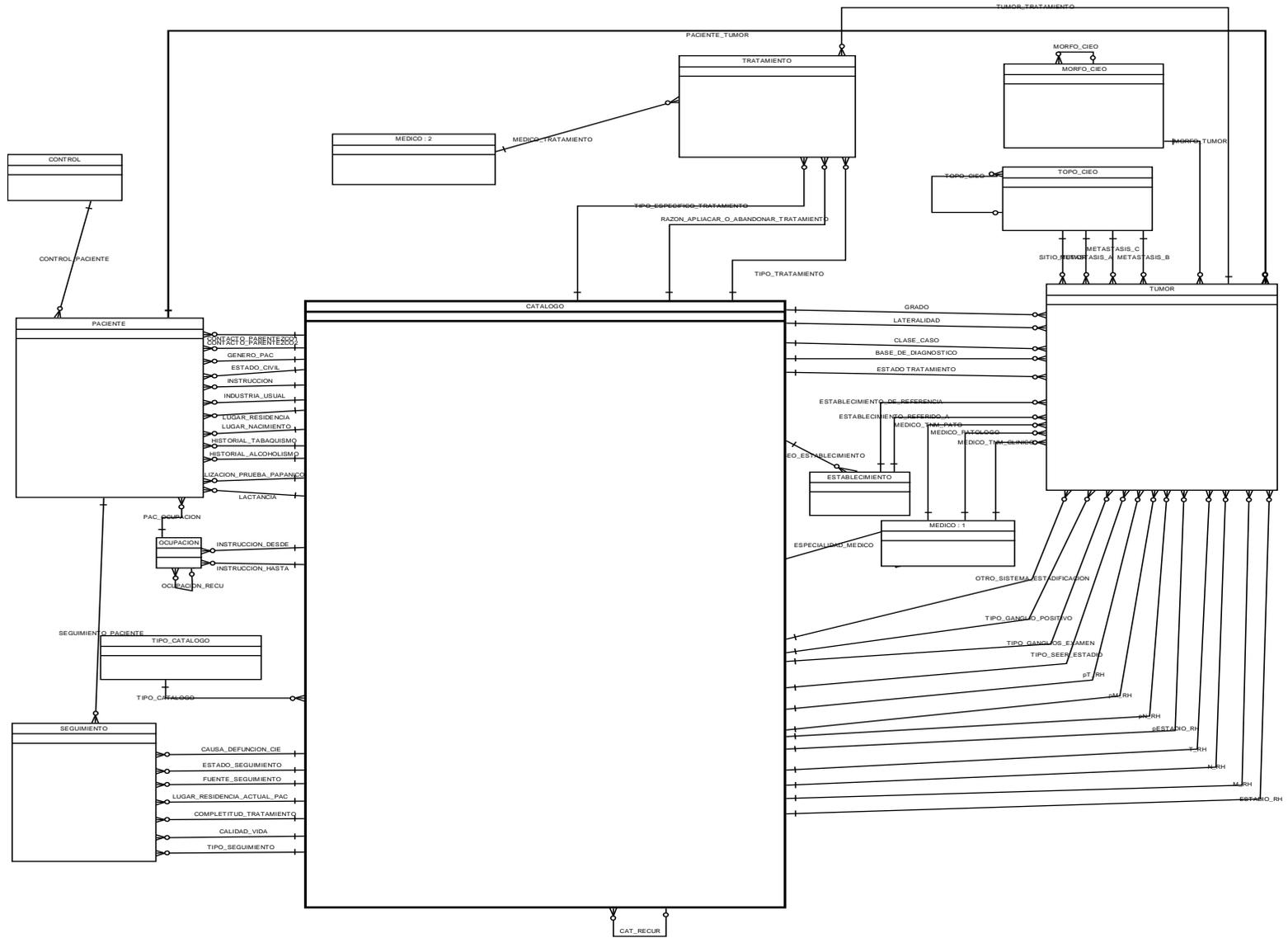
Se agruparon en forma consistente los requerimientos para definir los casos de uso que representan los módulos del mismo.



### Modelo clases de Análisis

Se definieron junto con los miembros del Registro Hospitalario las principales clases de datos del sistema y sus relaciones básicas. Esto se representa a través del diagrama de clases de análisis.

### Modelo de clases de Análisis



### **3.3.4 MODELO DE DISEÑO**

El diseño comienza al final de la fase de iniciación, toma mayor énfasis en la fase de elaboración y se desarrolla hasta el principio de la fase de transición.

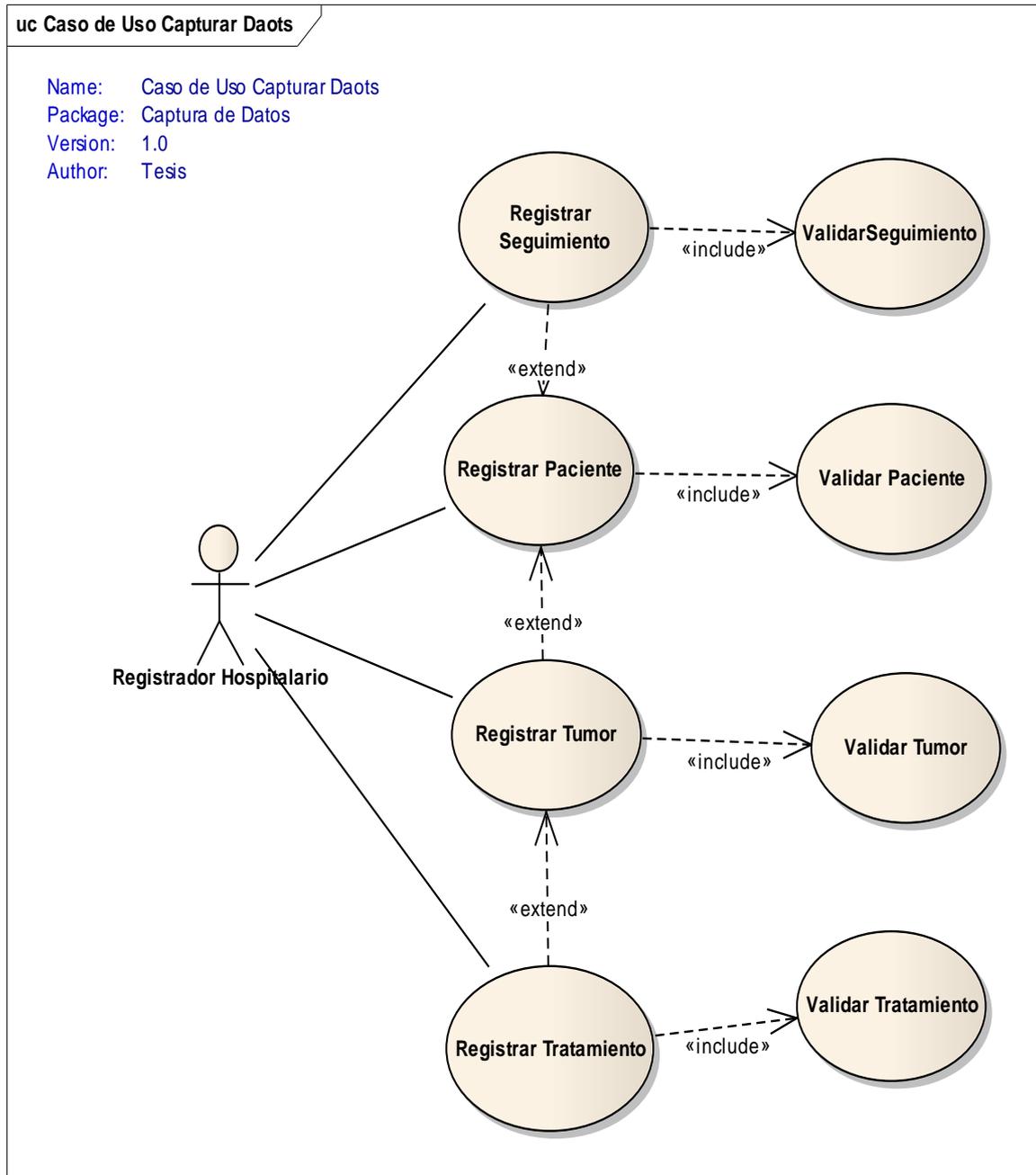
Desde el punto de vista funcional, estructural, dinámico y despliegue se diseño el sistema.

#### **3.3.4.1 Diseño Funcional**

- Se transformaron los requerimientos principales del sistema en casos de uso base, incluidos y extensores.
- Se relacionaron los requerimientos a los casos de uso.
- Se describió la lógica básica del los casos de uso.
- Se realizó diagrama de casos de uso por módulos, sirve para reconocer los casos de uso de acuerdo al módulo y a la interacción con los actores del sistema.
- Se realizó el diagrama de casos de uso general del sistema, todos los casos de uso que fueron reconocidos en los diferentes módulos los presenta en un solo plano.
- Adicionalmente se trabajó con los usuarios con el fin de definir de manera general las interfaces de usuario que satisfagan la facilidad de navegabilidad de acuerdo a los procesos administrativos que manejan en el Registro Hospitalario,



## Diagramas de casos de uso de módulo Captura de Datos



**A: Registrar Pacientes**

*Descripción del caso de uso:*

<b>CASO DE USO:</b> REGISTRAR PACIENTES
<b>ACTORES:</b> REGISTRADOR HOSPITALARIO
<b>PRE-CONDICIÓN:</b> El paciente se presente por primera vez al Registro Hospitalario.
<b>POST-CONDICIÓN:</b> Tener registrado o ingresado el paciente.
<p><b>DESCRIPCIÓN:</b> Mantener un registro de todos los pacientes del Registro Hospitalario.</p> <p><b>ESCENARIO NORMAL:</b></p> <p>10 El sistema despliega los criterios de selección para buscar paciente.</p> <p>20 El sistema captura los criterios de selección.</p> <p>30 El sistema busca los pacientes que cumplan con criterios de selección.</p> <p>40 El Registrador Hospitalario da un clic en el botón nuevo para que pueda ingresar la información necesaria.</p> <p>50 El sistema despliega interfaz vacía para llenar paciente.</p> <p>60 El sistema despliega los datos del paciente.</p> <p>70 El Registrador Hospitalario ingresa los datos personales del paciente</p> <ul style="list-style-type: none"> <li>- Nombre1</li> <li>- Nombre2</li> <li>- Apellido1</li> <li>- Apellido2</li> <li>- Cédula</li> <li>- Fecha nacimiento</li> </ul>

- *Estado civil*
- *Género*
- *Dirección*
- *Ciudad*
- *País*
- *Lugar de nacimiento*
  - *provincia*
  - *cantón*
  - *parroquia*
- *Lugar de residencia*
  - *provincia*
  - *cantón*
  - *parroquia*
- *Instrucción*
- *Ocupación*
- *Industria usual*

*80 El Registrador Hospitalario ingresa los datos relacionados con cáncer del paciente*

- *Antecedentes personales*
- *Antecedentes familiares*
- *Edad primera relación sexual*
- *Número cesáreas*
- *Número embarazos*
- *Realiza papanicolaou*
- *Lactancia*
- *Historial alcoholismo*
- *Estado vital*

*90 El Registrador Hospitalario ingresa los datos de contactos del paciente*

**CONTACTO1**

- *Nombre*
- *Ciudad*
- *Teléfono1*
- *Telefono2*
- *Dirección*
- *Parentesco*

**CONTACTO2**

- *Nombre*
- *Ciudad*
- *Teléfono1*
- *Telefono2*
- *Dirección*
- *Parentesco*
- *Fecha pase control*
- *Observación*
- *Institución control*

*100. El Registrador Hospitalario confirma el ingreso con el botón guardar.*

*110. El sistema ejecuta validar paciente (caso de uso incluido).*

*120. Se crea un nuevo registro en la base de datos.*

**ESCENARIO EXCEPCIÓN:**

**ACCIÓN 30:**

- 10 El registrador Hospitalario escoge paciente para actualización
- 20 El sistema despliega datos de paciente
- 30 El sistema regresa a la acción 70 y continua con las acciones posteriores.

**ACCIÓN 100:**

10. El Registrador Hospitalario puede presiona el botón cancelar.
20. El caso de uso termina con la acción no realizada.

**ACCIÓN 110:**

10. Si la validación falla entonces el sistema despliega un mensaje de error que permita identificar las fallas.
20. Regresa a la opción 30 del escenario normal.

**B: Validar paciente**

*Descripción del caso de uso:*

<b>CASO DE USO:</b> VALIDAR PACIENTE
<b>ACTORES:</b> REGISTRADOR HOSPITALARIO
<b>PRE-CONDICIÓN:</b> haber ingresado los datos de paciente.
<b>POST-CONDICIÓN:</b> tener los datos ingresados del paciente validados.
<b>DESCRIPCIÓN:</b> validar los registros realizados de pacientes.
<b>ESCENARIO NORMAL:</b>

10. El Registrador Hospitalario da clic en el botón guardar para registrar los datos ingresados de paciente.

20. El sistema ejecuta el método validarpaciente(), donde se valida los siguientes datos:

Controlar en todos los campos enviando un mensaje de error si se ingresan más caracteres que los permitidos a ingresar en la base de datos

Los siguientes datos son obligatorios:

- Nombre1
- Apellido1
- Fecha nacimiento
- Estado civil
- Género
- Dirección
- Lugar de nacimiento
  - provincia
  - cantón
  - parroquia
- Lugar de residencia
  - provincia
  - cantón
  - parroquia
- Instrucción
- Ocupación
- Industria usual
- Estado vital

CONTACTO1

- Nombre

- *Ciudad*
- *Teléfono 1*
- *Dirección*
- *Parentesco*

**ESCENARIO EXCEPCIÓN:**

**ACCIÓN 20:**

10. *El sistema encuentra fallas de validación de campos obligatorios y/o máximo de valores permitidos.*

20. *El sistema muestra el mensaje “campo obligatorio” y/o “En este campo debe ingresar máximo este numero de caracteres” en los campos donde haya detectado las fallas.*

**C: Registrar Seguimiento**

*Descripción del caso de uso:*

<b>CASO DE USO:</b> REGISTRAR SEGUIMIENTO
<b>ACTORES:</b> REGISTRADOR HOSPITALARIO
<b>PRE-CONDICIÓN:</b> tener registrado un paciente.
<b>POST-CONDICIÓN:</b> tener registrado o ingresado un seguimiento.
<b>DESCRIPCIÓN:</b> mantener un registro de todos los seguimientos a los pacientes del Registro Hospitalario.

**ESCENARIO NORMAL:**

10. El sistema verifica si la llamada es desde paciente o desde el menú.
20. Si la llamada es desde el menú despliega los criterios de selección de buscar seguimiento.
30. El sistema captura criterios de selección.
40. Buscar seguimiento que cumplan con criterios de selección.
50. Desplegar interfaz vacía para llenar datos de seguimiento.
60. El sistema captura los datos de seguimiento.
  - *Año seguimiento*
  - *Fecha último contacto*
  - *Fecha registro seguimiento*
  - *Fecha defunción*
  - *Fuente seguimiento*
  - *Lugar residencia*
    - *provincia*
    - *cantón*
    - *parroquia*
  - *Tipo seguimiento*
  - *Calidad de vida*
  - *Causa defunción cie*
  - *Completa tratamiento*
  - *Num edición cie seguimiento*
  - *Observaciones*
70. El Registrador Hospitalario confirma el ingreso con el botón guardar.
80. El sistema ejecuta validar seguimiento (caso de uso incluido).

90. Se crea un nuevo registro en la base de datos.

**ESCENARIO EXCEPCIÓN:**

**ACCIÓN 10:**

10 Si la llamada viene desde paciente buscar paciente por idPaciente.

20 Ir a acción 40.

**ACCIÓN 30:**

10 El registrador Hospitalario escoge seguimiento para actualización

20 El sistema despliega datos de seguimiento

30 El sistema regresa a la acción 60 y continua con las acciones posteriores.

**ACCIÓN 70:**

10. El Registrador Hospitalario puede presiona el botón cancelar.

20. El caso de uso termina con la acción no realizada.

**ACCIÓN 80:**

10. Si la validación falla entonces el sistema despliega un mensaje de error que permita identificar las fallas.

20. Regresa a la opción 30 del escenario normal.

**D: Validar seguimiento**

*Descripción del caso de uso:*

<b>CASO DE USO:</b> VALIDAR SEGUIMIENTO
<b>ACTORES:</b> REGISTRADOR HOSPITALARIO
<b>PRE-CONDICIÓN:</b> haber ingresado los datos de seguimiento.
<b>POST-CONDICIÓN:</b> tener los datos ingresados del seguimiento validados.
<p><b>DESCRIPCIÓN:</b> validar los registros realizados de seguimiento.</p> <p><b>ESCENARIO NORMAL:</b></p> <p>10. El Registrador Hospitalario da clic en el botón guardar para registrar los datos ingresados de seguimiento.</p> <p>20. El sistema ejecuta el método validarSeguimiento(), donde se valida los siguientes datos:</p> <p>Controlar en todos los campos enviando un mensaje de error si se ingresan más caracteres que los permitidos a ingresar en la base de datos</p> <p>Los siguientes datos son obligatorios:</p> <ul style="list-style-type: none"> <li>- Año seguimiento</li> <li>- Fecha último contacto</li> <li>- Fecha registro seguimiento</li> <li>- Fuente seguimiento</li> <li>- Lugar residencia <ul style="list-style-type: none"> <li>- provincia</li> <li>- cantón</li> <li>- parroquia</li> </ul> </li> </ul>

**ESCENARIO EXCEPCIÓN:***ACCIÓN 20:*

10. El sistema encuentra fallas de validación de campos obligatorios y/o máximo de valores permitidos.

20. El sistema muestra el mensaje “campo obligatorio” y/o “En este campo debe ingresar máximo este numero de caracteres” en los campos donde haya detectado las fallas.

**E: Registrar Tratamiento**

*Descripción del caso de uso:*

**CASO DE USO:** REGISTRAR TRATAMIENTO

**ACTORES:** REGISTRADOR HOSPITALARIO

**PRE-CONDICIÓN:** tener registrado un tumor y un paciente.

**POST-CONDICIÓN:** tener registrado o ingresado un tratamiento.

**DESCRIPCIÓN:** mantener un registro de todos los tratamientos a los pacientes del Registro Hospitalario.

**ESCENARIO NORMAL:**

10. El sistema verifica si la llamada es desde paciente o desde el menú.

20. Si la llamada es desde el menú despliega los criterios de selección de buscar

tratamiento.

30. El sistema captura criterios de selección.

40. Buscar tratamiento que cumplan con criterios de selección.

50. Desplegar interfaz vacía para llenar datos de tratamiento.

60. El sistema captura los datos de tratamiento.

- *Elige un paciente.*
- *Elige un tumor*
- *Fecha tratamiento*
- *Descripción tratamiento*
- *Razón abandona tratamiento*
- *Tipo tratamiento*
- *Médico*
- *Observación tratamiento*

70. El *Registrador Hospitalario* confirma el ingreso con el botón guardar.

80. El sistema ejecuta validar tratamiento (caso de uso incluido).

90. Se crea un nuevo registro en la base de datos.

#### **ESCENARIO EXCEPCIÓN:**

*ACCIÓN 10:*

*10 Si la llamada viene desde paciente buscar paciente por idPaciente.*

*20 Ir a acción 40.*

*ACCIÓN 30:*

*10 El registrador Hospitalario escoge tratamiento para actualización*

*20 El sistema despliega datos de tratamiento*

*30 El sistema regresa a la acción 60 y continua con las acciones posteriores.*

**ACCIÓN 70:**

*10. El Registrador Hospitalario puede presiona el botón cancelar.*

*20. El caso de uso termina con la acción no realizada.*

**ACCIÓN 80:**

*10. Si la validación falla entonces el sistema despliega un mensaje de error que permita identificar las fallas.*

*20. Regresa a la opción 30 del escenario normal.*

## **F: Validar Tratamiento**

*Descripción del caso de uso:*

**CASO DE USO:** VALIDAR TRATAMIENTO

**ACTORES:** REGISTRADOR HOSPITALARIO

**PRE-CONDICIÓN:** haber ingresado los datos de tratamiento.

**POST-CONDICIÓN:** tener los datos ingresados del tratamiento validados.

**DESCRIPCIÓN:** validar los registros realizados de tratamiento.

**ESCENARIO NORMAL:**

*10. El Registrador Hospitalario da clic en el botón guardar para registrar los datos ingresados de tratamiento.*

20. El sistema ejecuta el método `validarTratamiento()`, donde se valida los siguientes datos:

Controlar en todos los campos enviando un mensaje de error si se ingresan más caracteres que los permitidos a ingresar en la base de datos

Los siguientes datos son obligatorios:

- Elige un paciente.
- Elige un tumor
- Fecha tratamiento

#### **ESCENARIO EXCEPCIÓN:**

ACCIÓN 20:

10. El sistema encuentra fallas de validación de campos obligatorios y/o máximo de valores permitidos.

20. El sistema muestra el mensaje "campo obligatorio" y/o "En este campo debe ingresar máximo este numero de caracteres" en los campos donde haya detectado las fallas.

#### **G: Registrar Tumor**

Descripción del caso de uso:

<b>CASO DE USO:</b> REGISTRAR TUMOR
<b>ACTORES:</b> REGISTRADOR HOSPITALARIO
<b>PRE-CONDICIÓN:</b> tener registrado un paciente.
<b>POST-CONDICIÓN:</b> tener registrado o ingresado un tumor.

**DESCRIPCIÓN:** *mantener un registro de todos los tumores de los pacientes del Registro Hospitalario.*

**ESCENARIO NORMAL:**

10. El sistema verifica si la llamada es desde paciente o desde el menú.
20. Si la llamada es desde el menú despliega los criterios de selección de buscar tumor.
30. El sistema captura criterios de selección.
40. Buscar tumor que cumplan con criterios de selección.
50. Desplegar interfaz vacía para llenar datos de tumor.
60. El sistema captura los datos de tumor.
  - *Elige un paciente.*
  - *Médico*
  - *Fecha diagnóstico*
  - *Edad diagnóstico*
  - *Fecha primera atención*
  - *Fecha diagnostico RNT*
  - *Fecha primer tratamiento*
  - *Clase caso*
  - *Morfología*
  - *Topo sitio tumor*
  - *Lateralidad*
  - *Base diagnóstico*
  - *Estado tratamiento*
  - *Grado*
  - *Establecimiento referido*

- *Establecimiento referencia*
- *Tipo metástasis a*
- *Tipo metástasis b*
- *Tipo metástasis c*
- *Observaciones*

*Estadio*

- *T*
- *N*
- *M*
- *Estadio*
- *Medico patólogo*
- *pt*
- *pn*
- *pm*
- *Pestadio*
- *Médico clínico*
- *Ganglios examen*
- *Ganglios positivos*
- *Otro sistema estadificación*
- *Marcador tumoral 1*
- *Marcador tumoral 2*
- *Marcador tumoral 3*
- *Edicion seer*
- *Edicion tnm*

70. El Registrador Hospitalario confirma el ingreso con el botón guardar.

80. El sistema ejecuta validar tumor (caso de uso incluido).

90. Se crea un nuevo registro en la base de datos.

**ESCENARIO EXCEPCIÓN:**

**ACCIÓN 10:**

10 Si la llamada viene desde paciente buscar paciente por idPaciente.

20 Ir a acción 40.

**ACCIÓN 30:**

10 El registrador Hospitalario escoge tumor para actualización

20 El sistema despliega datos de tumor

30 El sistema regresa a la acción 60 y continua con las acciones posteriores.

**ACCIÓN 70:**

10. El Registrador Hospitalario puede presiona el botón cancelar.

20. El caso de uso termina con la acción no realizada.

**ACCIÓN 80:**

10. Si la validación falla entonces el sistema despliega un mensaje de error que permita identificar las fallas.

20. Regresa a la opción 30 del escenario normal.

**H: Validar Tumor**

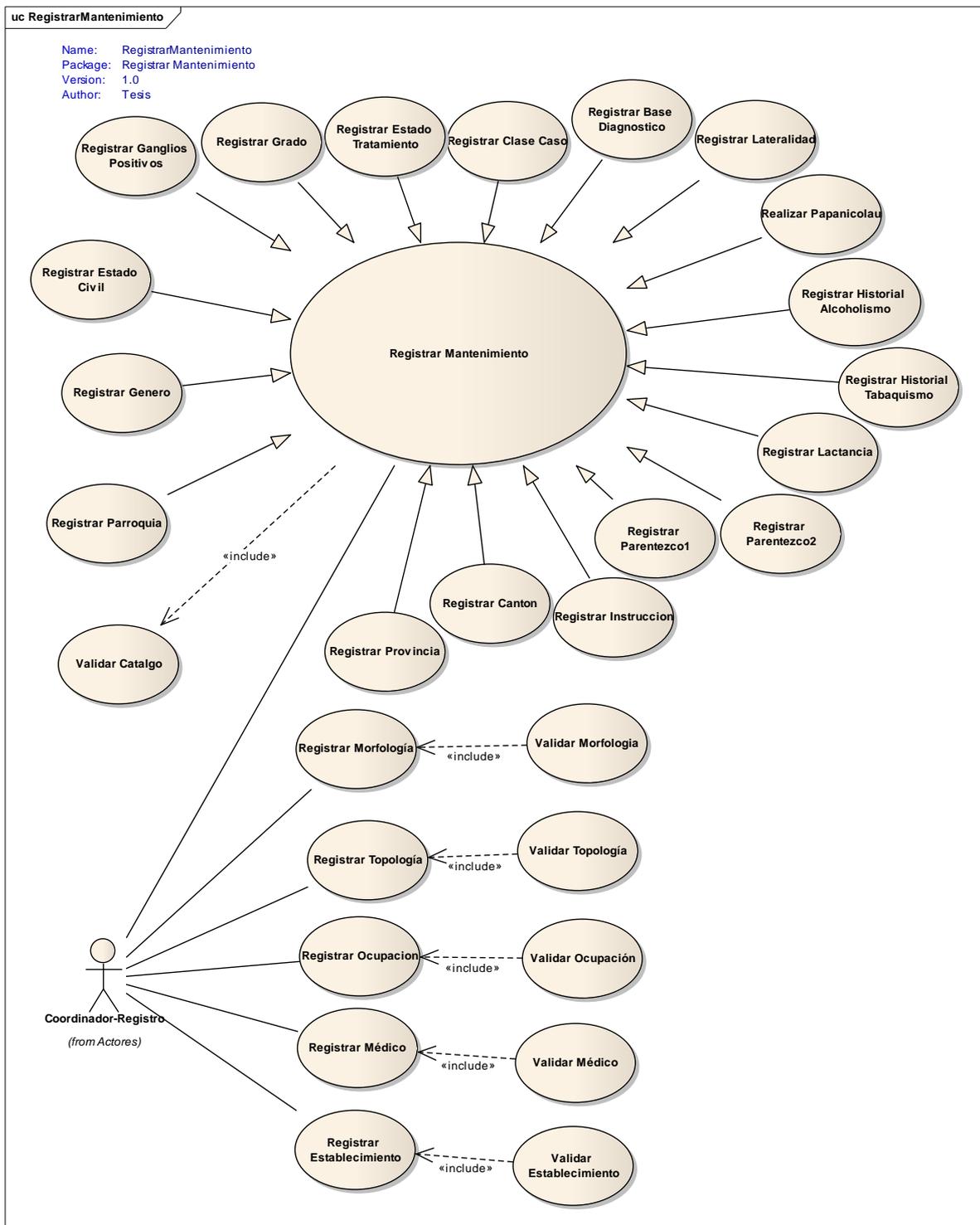
*Descripción del caso de uso:*

<b>CASO DE USO:</b> <i>VALIDAR TUMOR</i>
<b>ACTORES:</b> <i>REGISTRADOR HOSPITALARIO</i>
<b>PRE-CONDICIÓN:</b> <i>haber ingresado los datos de tumor.</i>
<b>POST-CONDICIÓN:</b> <i>tener los datos ingresados del tumor validados.</i>
<p><b>DESCRIPCIÓN:</b> <i>validar los registros realizados de tumor.</i></p> <p><b>ESCENARIO NORMAL:</b></p> <p><i>10. El Registrador Hospitalario da clic en el botón guardar para registrar los datos ingresados de tumor.</i></p> <p><i>20. El sistema ejecuta el método validarTumor (), donde se valida los siguientes datos:</i></p> <p><i>Controlar en todos los campos enviando un mensaje de error si se ingresan más caracteres que los permitidos a ingresar en la base de datos</i></p> <p><i>Los siguientes datos son obligatorios:</i></p> <ul style="list-style-type: none"> <li>- <i>Elige un paciente.</i></li> <li>- <i>Médico</i></li> <li>- <i>Fecha diagnóstico</i></li> <li>- <i>Edad diagnóstico</i></li> <li>- <i>Fecha primera atención</i></li> </ul> <p><b>ESCENARIO EXCEPCIÓN:</b></p> <p><b>ACCIÓN 20:</b></p> <p><i>10. El sistema encuentra fallas de validación de campos obligatorios y/o máximo de</i></p>

*valores permitidos.*

*20. El sistema muestra el mensaje "campo obligatorio" y/o "En este campo debe ingresar máximo este número de caracteres" en los campos donde haya detectado las fallas.*

### Diagrama de casos de uso módulo Mantenimiento



**A: Registrar Mantenimiento***Descripción del caso de uso:*

<b>CASO DE USO:</b> <i>COORDINADOR</i>
<b>ACTORES:</b> <i>COORDINADOR</i>
<b>POST-CONDICIÓN:</b> <i>tener registrado o ingresado un mantenimiento.</i>
<p><b>DESCRIPCIÓN:</b> <i>mantener un registro de todos los catalogos pertenecientes al registro hospitalario.</i></p> <p><b>ESCENARIO NORMAL:</b></p> <p>10. <i>El coordinador escoge a que catalogo quiere ingresar un nuevo dato.</i></p> <ul style="list-style-type: none"> <li>- <i>Registrar estado civil.</i></li> <li>- <i>Registrar ganglios positivos</i></li> <li>- <i>Registrar grado</i></li> <li>- <i>Registrar estado tratamiento</i></li> <li>- <i>Registrar clase caso</i></li> <li>- <i>Registrar clase caso</i></li> <li>- <i>Registrar base diagnostico</i></li> <li>- <i>Registrar lateralidad</i></li> <li>- <i>Registrar papanicolaou</i></li> <li>- <i>Registrar historial alcoholismo</i></li> <li>- <i>Registrar historial tabaquismo</i></li> <li>- <i>Registrar lactancia</i></li> <li>- <i>Registrar parentesco2</i></li> <li>- <i>Registrar parentesco1</i></li> <li>- <i>Registrar instrucción</i></li> </ul>

- Registrar cantón
- Registrar provincia
- Registrar parroquia
- Registrar género
- 

20. El coordinador da un clic en el botón nuevo para que pueda ingresar la información necesaria.

30. El coordinador ingresa los siguientes datos:

- Código Catalogo
- Catalogo en rnt
- Descripción
- Descripción en inglés
- Catálogo padre

40. El coordinador confirma el ingreso con el botón guardar.

50. El sistema ejecuta validar catalogo (caso de uso incluido).

60. Se crea un nuevo registro en la base de datos.

#### **ESCENARIO EXCEPCIÓN:**

**ACCIÓN 40:**

10. El coordinador puede presiona el botón cancelar.

20. El caso de uso termina con la acción no realizada.

**Acción 50:**

10. Si la validación falla entonces el sistema despliega un mensaje de error que

*permita identificar las fallas.*

*20. Regresa a la opción 20 del escenario normal.*

## **B: Validar Mantenimiento**

*Descripción del caso de uso:*

<b>CASO DE USO:</b> <i>VALIDAR MANTENIMIENTO</i>
<b>ACTORES:</b> <i>COORDINADOR</i>
<b>PRE-CONDICIÓN:</b> <i>haber ingresado los datos de mantenimiento</i>
<b>POST-CONDICIÓN:</b> <i>tener los datos ingresados del mantenimiento validados.</i>
<p><b>DESCRIPCIÓN:</b> <i>validar los registros realizados de mantenimiento.</i></p> <p><b>ESCENARIO NORMAL:</b></p> <p><i>10. el coordinador da clic en el botón guardar para registrar los datos ingresados de mantenimiento.</i></p> <p><i>20. el sistema ejecuta el método validarcatalogo().</i></p> <p><i>Controlar en todos los campos enviando un mensaje de error si se ingresan más caracteres que los permitidos a ingresar en la base de datos</i></p> <p><i>Los siguientes datos son obligatorios:</i></p> <ul style="list-style-type: none"> <li>- <i>Código Catalogo</i></li> <li>- <i>Descripción</i></li> </ul> <p><b>ESCENARIO EXCEPCIÓN:</b></p>

**ACCIÓN 20:**

10. el sistema encuentra fallas de validación de campos obligatorios y/o campos de validación de solo números.

20. el sistema muestra el mensaje “campo obligatorio” y/o “debe ingresar solo números” de los campos donde haya detectado las fallas.

**C: Registrar Morfología**

*Descripción del caso de uso:*

**CASO DE USO:** COORDINADOR

**ACTORES:** COORDINADOR

**POST-CONDICIÓN:** tener registrado o ingresado de morfología.

**DESCRIPCIÓN:** mantener un registro de todos las morfologías del registro hospitalario.

**ESCENARIO NORMAL:**

10. el coordinador da un clic en el botón nuevo para que pueda ingresar la información necesaria.

20. el coordinador ingresa los siguientes datos:

- código morfología
- descripción
- descripción en inglés
- morfología padre
- orden de despliegue

30. el coordinador confirma el ingreso con el botón guardar.  
 40. el sistema ejecuta validar catalogo (caso de uso incluido).  
 50. se crea un nuevo registro en la base de datos.

**ESCENARIO EXCEPCIÓN:**

**ACCIÓN 30:**

10. el coordinador puede presiona el botón cancelar.  
 20. el caso de uso termina con la acción no realizada.

**ACCIÓN 40:**

10. si la validación falla entonces el sistema despliega un mensaje de error que permita identificar las fallas.  
 20. regresa a la opción 20 del escenario normal.

**D: Validar Morfología**

*Descripción del caso de uso:*

**CASO DE USO:** VALIDAR MORFOLOGÍA

**ACTORES:** COORDINADOR

**PRE-CONDICIÓN:** haber ingresado los datos de morfología

**POST-CONDICIÓN:** tener los datos ingresados de morfología validados.

**DESCRIPCIÓN:** validar los registros realizados de morfología.

**ESCENARIO NORMAL:**

10. el coordinador da clic en el botón guardar para registrar los datos ingresados de morfología.

20. el sistema ejecuta el método validarmorfología().

Controlar en todos los campos enviando un mensaje de error si se ingresan más caracteres que los permitidos a ingresar en la base de datos

Los siguientes datos son obligatorios:

- código morfología
- descripción

#### **ESCENARIO EXCEPCIÓN:**

*ACCIÓN 20:*

10. el sistema encuentra fallas de validación de campos obligatorios y/o campos de validación de solo números.

20. el sistema muestra el mensaje “campo obligatorio” y/o “debe ingresar solo números” de los campos donde haya detectado las fallas.

#### **E: Registrar Topología**

*Descripción del caso de uso:*

**CASO DE USO:** COORDINADOR

**ACTORES:** COORDINADOR

**POST-CONDICIÓN:** tener registrado o ingresado de topología.

**DESCRIPCIÓN:** mantener un registro de todos las topologías del registro hospitalario.

**ESCENARIO NORMAL:**

10. el coordinador da un clic en el botón nuevo para que pueda ingresar la información necesaria.

20. el coordinador ingresa los siguientes datos:

- código topología
- descripción
- descripción en inglés
- sexo
- aplica lateralidad
- aplica extensión clínica
- orden de despliegue
- topología padre

30. el coordinador confirma el ingreso con el botón guardar.

40. el sistema ejecuta validar catalogo (caso de uso incluido).

50. se crea un nuevo registro en la base de datos.

**ESCENARIO EXCEPCIÓN:****ACCIÓN 30:**

10. el coordinador puede presiona el botón cancelar.

20. el caso de uso termina con la acción no realizada.

**ACCIÓN 40:**

10. si la validación falla entonces el sistema despliega un mensaje de error que permita identificar las fallas.

*20. regresa a la opción 20 del escenario normal.*

## **F: Validar Topología**

*Descripción del caso de uso:*

**CASO DE USO:** *VALIDAR TOPOLOGÍA*

**ACTORES:** *COORDINADOR*

**PRE-CONDICIÓN:** *haber ingresado los datos de topología*

**POST-CONDICIÓN:** *tener los datos ingresados de topología validados.*

**DESCRIPCIÓN:** *validar los registros realizados de topología.*

### **ESCENARIO NORMAL:**

*10. el coordinador da clic en el botón guardar para registrar los datos ingresados de topología.*

*20. el sistema ejecuta el método validar topología ().*

*Controlar en todos los campos enviando un mensaje de error si se ingresan más caracteres que los permitidos a ingresar en la base de datos*

*Los siguientes datos son obligatorios:*

- código topología*
- descripción*
- sexo*

### **ESCENARIO EXCEPCIÓN:**

*ACCIÓN 20:*

10. el sistema encuentra fallas de validación de campos obligatorios y/o campos de validación de solo números.

20. el sistema muestra el mensaje “campo obligatorio” y/o “debe ingresar solo números” de los campos donde haya detectado las fallas.

### **G: Registrar Ocupación**

*Descripción del caso de uso:*

**CASO DE USO:** COORDINADOR

**ACTORES:** COORDINADOR

**POST-CONDICIÓN:** tener registrado o ingresado de ocupación.

**DESCRIPCIÓN:** mantener un registro de todos las ocupación del registro hospitalario.

#### **ESCENARIO NORMAL:**

10. el coordinador da un clic en el botón nuevo para que pueda ingresar la información necesaria.

20. el coordinador ingresa los siguientes datos:

- código
- descripción
- instrucción desde
- instrucción hasta
- ocupación padre

30. el coordinador confirma el ingreso con el botón guardar.

40. el sistema ejecuta validar ocupación (caso de uso incluido).

50. se crea un nuevo registro en la base de datos.

**ESCENARIO EXCEPCIÓN:**

**ACCIÓN 30:**

10. el coordinador puede presiona el botón cancelar.

20. el caso de uso termina con la acción no realizada.

**ACCIÓN 40:**

10. si la validación falla entonces el sistema despliega un mensaje de error que permita identificar las fallas.

20. regresa a la opción 20 del escenario normal.

**H: Validar Ocupación**

*Descripción del caso de uso:*

**CASO DE USO:** VALIDAR OCUPACIÓN

**ACTORES:** COORDINADOR

**PRE-CONDICIÓN:** HABER INGRESADO LOS DATOS DE OCUPACIÓN

**POST-CONDICIÓN:** tener los datos ingresados de ocupación validados.

**DESCRIPCIÓN:** validar los registros realizados de ocupación.

**ESCENARIO NORMAL:**

10. el coordinador da clic en el botón guardar para registrar los datos ingresados de

*ocupación.*

*20. el sistema ejecuta el método validar ocupación ().*

*Controlar en todos los campos enviando un mensaje de error si se ingresan más caracteres que los permitidos a ingresar en la base de datos*

*Los siguientes datos son obligatorios:*

- código*
- descripción*
- instrucción desde*
- instrucción hasta*

**ESCENARIO EXCEPCIÓN:**

*ACCIÓN 20:*

*10. el sistema encuentra fallas de validación de campos obligatorios y/o campos de validación de solo números.*

*20. el sistema muestra el mensaje “campo obligatorio” y/o “debe ingresar solo números” de los campos donde haya detectado las fallas.*

**I: Registrar Médico**

*Descripción del caso de uso:*

**CASO DE USO:** *COORDINADOR*

**ACTORES:** *COORDINADOR*

**POST-CONDICIÓN:** *tener registrado o ingresado de médico.*

**DESCRIPCIÓN:** *mantener un registro de todos los médico del registro hospitalario.*

**ESCENARIO NORMAL:**

10. el coordinador da un clic en el botón nuevo para que pueda ingresar la información necesaria.

20. el coordinador ingresa los siguientes datos:

- nombre
- especialidad

30. el coordinador confirma el ingreso con el botón guardar.

40. el sistema ejecuta validar médico (caso de uso incluido).

50. se crea un nuevo registro en la base de datos.

**ESCENARIO EXCEPCIÓN:****ACCIÓN 30:**

10. el coordinador puede presiona el botón cancelar.

20. el caso de uso termina con la acción no realizada.

**ACCIÓN 40:**

10. si la validación falla entonces el sistema despliega un mensaje de error que permita identificar las fallas.

20. regresa a la opción 20 del escenario normal.

**J: Validar Médico**

*Descripción del caso de uso:*

<b>CASO DE USO:</b> <i>VALIDAR MÉDICO</i>
<b>ACTORES:</b> <i>COORDINADOR</i>
<b>PRE-CONDICIÓN:</b> <i>haber ingresado los datos de médico</i>
<b>POST-CONDICIÓN:</b> <i>tener los datos ingresados de médico validados.</i>
<p><b>DESCRIPCIÓN:</b> <i>validar los registros realizados de médico.</i></p> <p><b>ESCENARIO NORMAL:</b></p> <p><i>10. el coordinador da clic en el botón guardar para registrar los datos ingresados de médico.</i></p> <p><i>20. el sistema ejecuta el método validar médico ().</i></p> <p><i>Controlar en todos los campos enviando un mensaje de error si se ingresan más caracteres que los permitidos a ingresar en la base de datos</i></p> <p><i>Los siguientes datos son obligatorios:</i></p> <ul style="list-style-type: none"> <li><i>- nombre</i></li> <li><i>- especialidad</i></li> </ul> <p><b>ESCENARIO EXCEPCIÓN:</b></p> <p><b>ACCIÓN 20:</b></p> <p><i>10. el sistema encuentra fallas de validación de campos obligatorios y/o campos de validación de solo números.</i></p> <p><i>20. el sistema muestra el mensaje “campo obligatorio” y/o “debe ingresar solo números” de los campos donde haya detectado las fallas.</i></p>

**K: Registrar Establecimiento***Descripción del caso de uso:*

<b>CASO DE USO:</b> <i>COORDINADOR</i>
<b>ACTORES:</b> <i>COORDINADOR</i>
<b>POST-CONDICIÓN:</b> <i>tener registrado o ingresado de establecimiento.</i>
<p><b>DESCRIPCIÓN:</b> <i>mantener un registro de todos los establecimiento del registro hospitalario.</i></p> <p><b>ESCENARIO NORMAL:</b></p> <p><i>10. el coordinador da un clic en el botón nuevo para que pueda ingresar la información necesaria.</i></p> <p><i>20. el coordinador ingresa los siguientes datos:</i></p> <ul style="list-style-type: none"> <li>- <i>código establecimiento</i></li> <li>- <i>nombre de establecimiento</i></li> <li>- <i>cuidad</i></li> <li>- <i>provincia</i></li> <li>- <i>cantón</i></li> <li>- <i>parroquia</i></li> </ul> <p><i>30. el coordinador confirma el ingreso con el botón guardar.</i></p> <p><i>40. el sistema ejecuta validar establecimiento (caso de uso incluido).</i></p> <p><i>50. se crea un nuevo registro en la base de datos.</i></p> <p><b>ESCENARIO EXCEPCIÓN:</b></p>

**ACCIÓN 30:**

10. el coordinador puede presiona el botón cancelar.
20. el caso de uso termina con la acción no realizada.

**ACCIÓN 40:**

10. si la validación falla entonces el sistema despliega un mensaje de error que permita identificar las fallas.
20. regresa a la opción 20 del escenario normal.

**L: Validar Establecimiento**

*Descripción del caso de uso:*

**CASO DE USO:** VALIDAR ESTABLECIMIENTO

**ACTORES:** COORDINADOR

**PRE-CONDICIÓN:** haber ingresado los datos de establecimiento

**POST-CONDICIÓN:** tener los datos ingresados de establecimiento validados.

**DESCRIPCIÓN:** validar los registros realizados de establecimiento.

**ESCENARIO NORMAL:**

10. el coordinador da clic en el botón guardar para registrar los datos ingresados de establecimiento.
  20. el sistema ejecuta el método validar establecimiento ().
- Controlar en todos los campos enviando un mensaje de error si se ingresan más caracteres que los permitidos a ingresar en la base de datos

*Los siguientes datos son obligatorios:*

- *código establecimiento*
- *nombre de establecimiento*

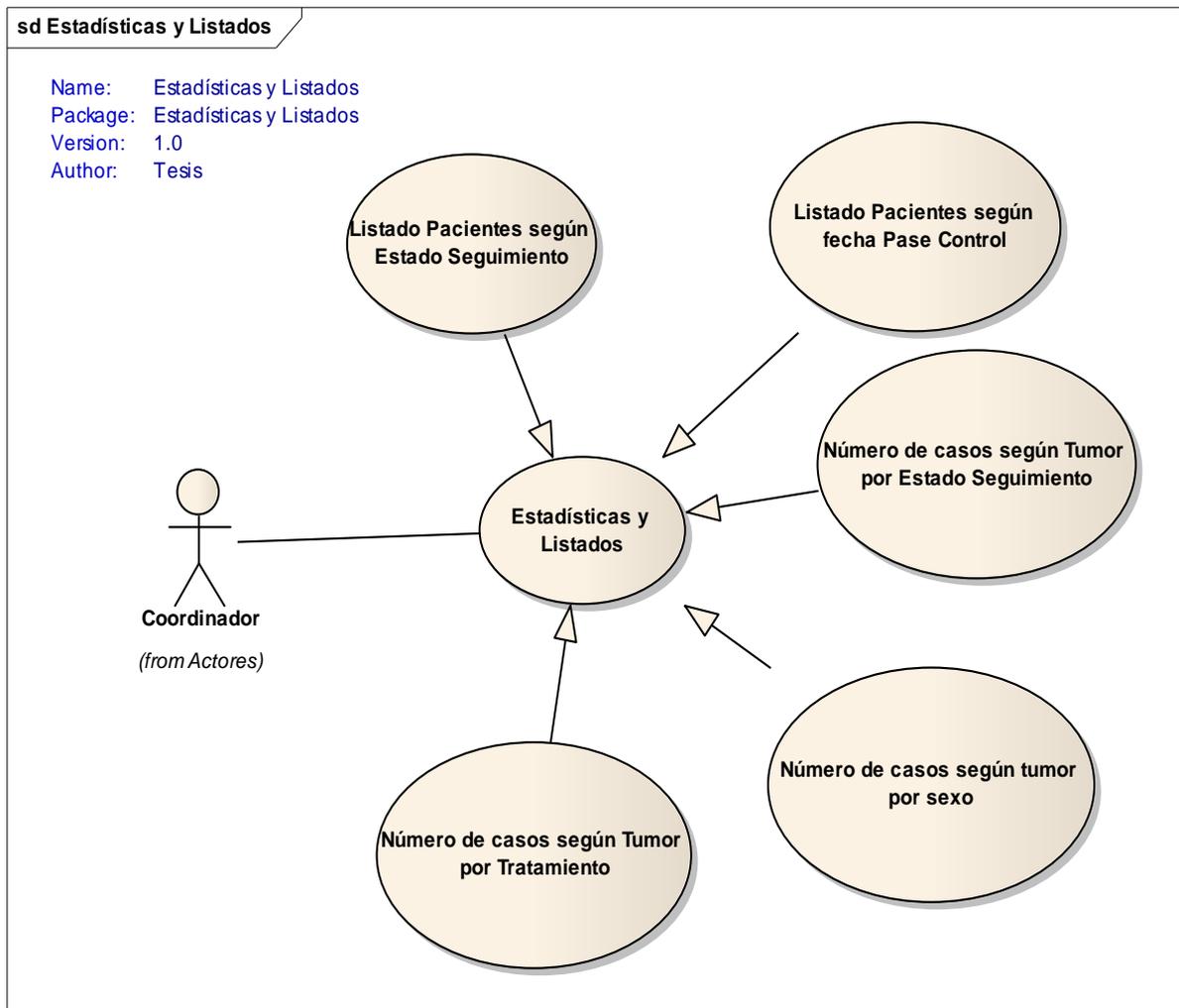
**ESCENARIO EXCEPCIÓN:**

*ACCIÓN 20:*

*10. el sistema encuentra fallas de validación de campos obligatorios y/o campos de validación de solo números.*

*20. el sistema muestra el mensaje “campo obligatorio” y/o “debe ingresar solo números” de los campos donde haya detectado las fallas.*

## Diagramas de casos de uso de Estadísticas y Listados



## A: Listado Pacientes según estado seguimiento

*Descripción del caso de uso:*

<b>CASO DE USO:</b> <i>LISTADO PACIENTES SEGÚN ESTADO SEGUIMIENTO</i>
<b>ACTORES:</b> <i>COORDINADOR</i>
<b>PRE-CONDICIÓN:</b> <i>seleccionar la opción reportes de listados.</i>
<b>POST-CONDICIÓN:</b> <i>visualizar el reporte de acuerdo al tipo de selección.</i>
<p><b>DESCRIPCIÓN:</b> <i>visualización de los reportes de tipo listado pacientes según estado seguimiento.</i></p> <p><b>ESCENARIO NORMAL:</b></p> <p><i>10. el coordinador selecciona el reporte listado pacientes según estado seguimiento.</i></p> <p><i>20. el sistema ejecuta la consulta.</i></p> <p><i>30. el sistema despliega el reporte solicitado.</i></p> <p><b>ESCENARIO EXCEPCIÓN:</b></p> <p><b>ACCIÓN 10:</b></p> <p><i>10. el coordinador puede presiona el botón salir.</i></p> <p><i>20. el caso de uso termina con la acción no realizada.</i></p>

**B: Listado Pacientes según fecha pase a control**

*Descripción del caso de uso:*

<b>CASO DE USO:</b> <i>LISTADO PACIENTES SEGÚN FECHA PASE A CONTROL</i>
<b>ACTORES:</b> <i>COORDINADOR</i>
<b>PRE-CONDICIÓN:</b> <i>seleccionar la opción reportes de listados.</i>
<b>POST-CONDICIÓN:</b> <i>visualizar el reporte de acuerdo al tipo de selección.</i>
<b>DESCRIPCIÓN:</b> <i>visualización de los reportes de tipo listado pacientes según fecha pase a control.</i>
<b>ESCENARIO NORMAL:</b>
<i>10. el coordinador selecciona el reporte listado pacientes según fecha pase a control</i>
<i>20. el sistema ejecuta la consulta.</i>
<i>30. el sistema despliega el reporte solicitado.</i>
<b>ESCENARIO EXCEPCIÓN:</b>
<b>ACCIÓN 10:</b>
<i>10. el coordinador puede presiona el botón salir.</i>
<i>20. el caso de uso termina con la acción no realizada.</i>

**C: Número de casos según tumor por estado seguimiento.**

*Descripción del caso de uso:*

<b>CASO DE USO:</b> <i>NÚMERO DE CASOS SEGÚN TUMOR POR ESTADO SEGUIMIENTO.</i>
<b>ACTORES:</b> <i>COORDINADOR</i>
<b>PRE-CONDICIÓN:</b> <i>seleccionar la opción reportes de estadísticas número de casos según tumor por estado seguimiento.</i>
<b>POST-CONDICIÓN:</b> <i>visualizar el reporte de acuerdo al tipo de selección.</i>
<p><b>DESCRIPCIÓN:</b> <i>visualización de los reportes de tipo estadística.</i></p> <p><b>ESCENARIO NORMAL:</b></p> <p><i>10. el coordinador selecciona el reporte de estadísticas número de casos según tumor por estado seguimiento.</i></p> <p><i>20. el sistema ejecuta la consulta.</i></p> <p><i>30. el sistema despliega el reporte solicitado.</i></p> <p><b>ESCENARIO EXCEPCIÓN:</b></p> <p><b>ACCIÓN 10:</b></p> <p><i>10. el coordinador puede presiona el botón salir.</i></p> <p><i>20. el caso de uso termina con la acción no realizada.</i></p>

**D: Número de casos según tumor por sexo.**

*Descripción del caso de uso:*

<b>CASO DE USO:</b> <i>NÚMERO DE CASOS SEGÚN TUMOR POR SEXO.</i>
<b>ACTORES:</b> <i>COORDINADOR</i>
<b>PRE-CONDICIÓN:</b> <i>seleccionar la opción reportes de estadísticas número de casos según tumor por sexo.</i>
<b>POST-CONDICIÓN:</b> <i>visualizar el reporte de acuerdo al tipo de selección.</i>
<p><b>DESCRIPCIÓN:</b> <i>visualización de los reportes de tipo estadística.</i></p> <p><b>ESCENARIO NORMAL:</b></p> <p><i>10. el coordinador selecciona el reporte de estadísticas número de casos según tumor por sexo.</i></p> <p><i>20. el sistema ejecuta la consulta.</i></p> <p><i>30. el sistema despliega el reporte solicitado.</i></p> <p><b>ESCENARIO EXCEPCIÓN:</b></p> <p><b>ACCIÓN 10:</b></p> <p><i>10. el coordinador puede presiona el botón salir.</i></p> <p><i>20. el caso de uso termina con la acción no realizada.</i></p>

**E: Número de casos según tumor por tratamiento.**

*Descripción del caso de uso:*

<b>CASO DE USO:</b> <i>NÚMERO DE CASOS SEGÚN TUMOR POR TRATAMIENTO.</i>
<b>ACTORES:</b> <i>COORDINADOR</i>
<b>PRE-CONDICIÓN:</b> <i>seleccionar la opción reportes de estadísticas número de casos según tumor por tratamiento.</i>
<b>POST-CONDICIÓN:</b> <i>visualizar el reporte de acuerdo al tipo de selección.</i>
<p><b>DESCRIPCIÓN:</b> <i>visualización de los reportes de tipo estadística.</i></p> <p><b>ESCENARIO NORMAL:</b></p> <p><i>10. el coordinador selecciona el reporte de estadísticas número de casos según tumor por tratamiento.</i></p> <p><i>20. el sistema ejecuta la consulta.</i></p> <p><i>30. el sistema despliega el reporte solicitado.</i></p> <p><b>ESCENARIO EXCEPCIÓN:</b></p> <p><b>ACCIÓN 10:</b></p> <p><i>10. el coordinador puede presiona el botón salir.</i></p> <p><i>20. el caso de uso termina con la acción no realizada.</i></p>

### **Diseño de prototipos de interfaces**

Se trabajó con los usuarios con el fin de definir de manera general las interfaces de usuario, en este sentido, se presenta un prototipo de interfaces que ayuda a entender al usuario de manera general como quedaría el sistema ver anexo **Anexo CapIII A2:**

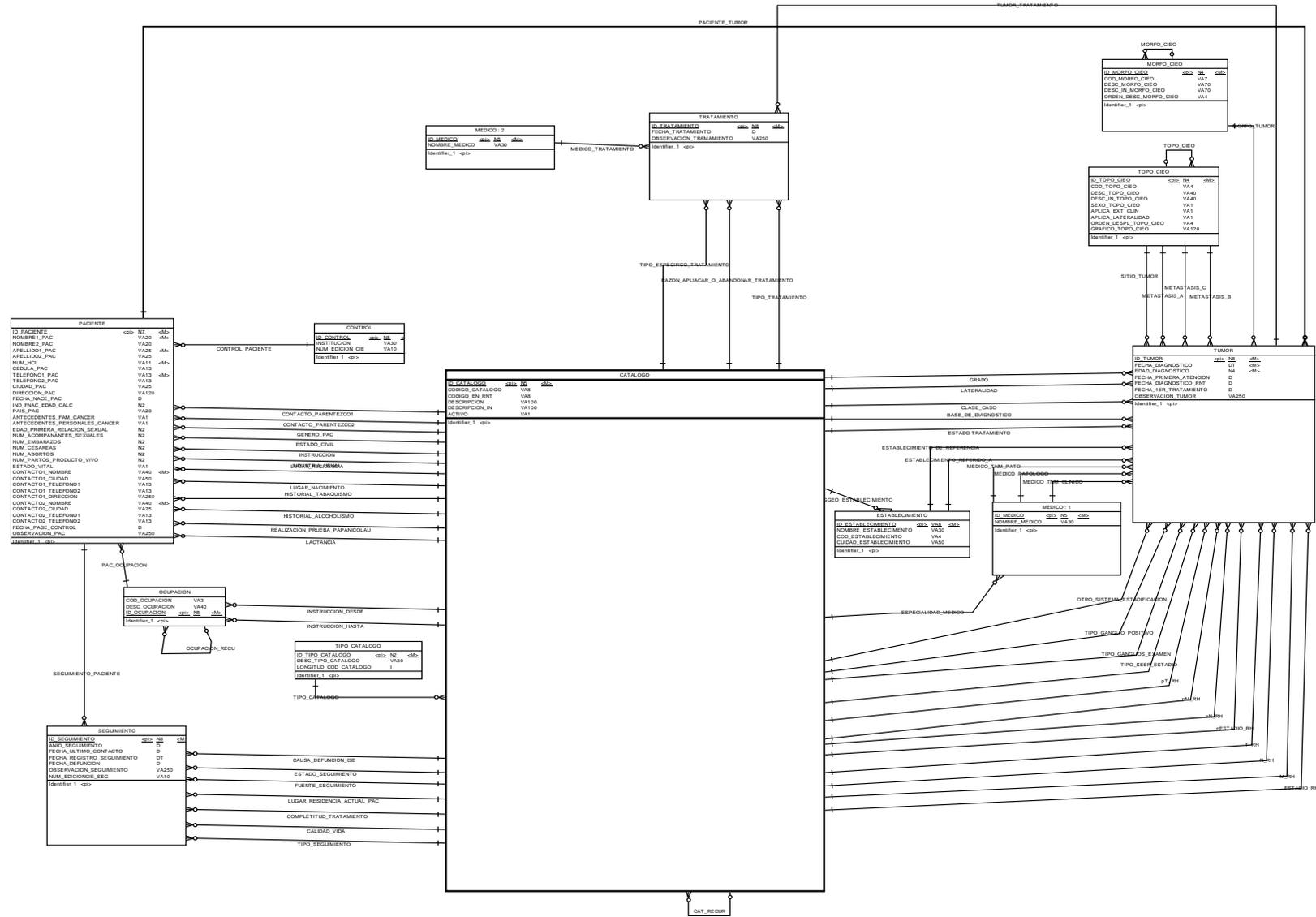
- Prototipo de interface pantalla de Login
- Prototipo de interface Menú del sistema
- Prototipo de interface para navegabilidad en opciones de carga de datos
- Prototipos de interface de carga de datos

### 3.3.4.2 Diseño Estructural

- Se reconocieron los atributos correspondientes a cada clase de datos.
- Se verificaron las relaciones de las clases de datos según las reglas del negocio.
- Se presentó el diagrama de clases de diseño.
- Se preparó el diccionario de datos del diagrama de clases de diseño.

El diccionario de datos se encuentra detallado en el **Anexo Cap III A 3**.

# Modelo Clases de Diseño



### 3.3.4.3 Diseño Dinámico

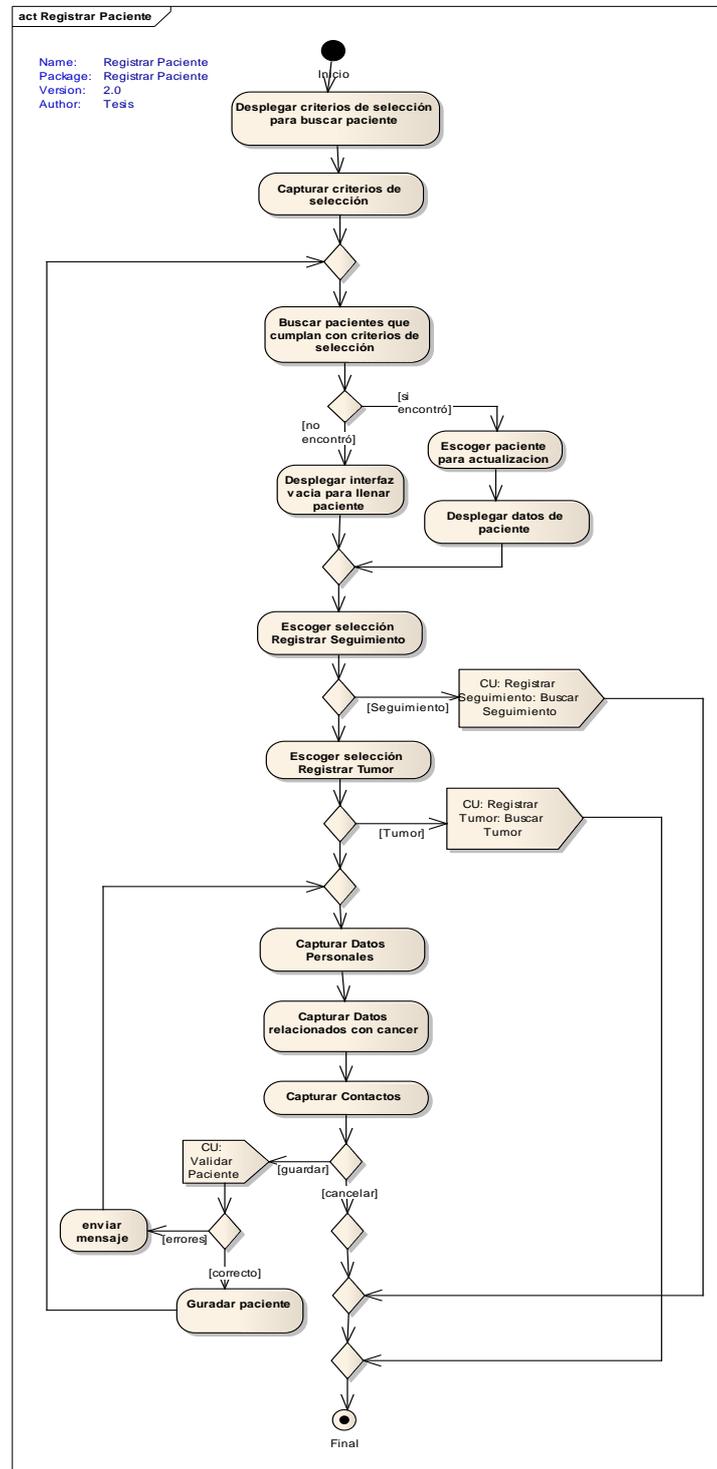
- Se reconocieron los casos de uso complejos tomando en cuenta los controladores y se especificó su lógica a través de diagramas de actividades.
- Se reconocieron las interrelaciones entre los objetos mediante diagrama de objetos
- Se realizó los diagramas de secuencia, se reconocieron los objetos y relaciones del sistema.

#### ***Diagramas de actividades***

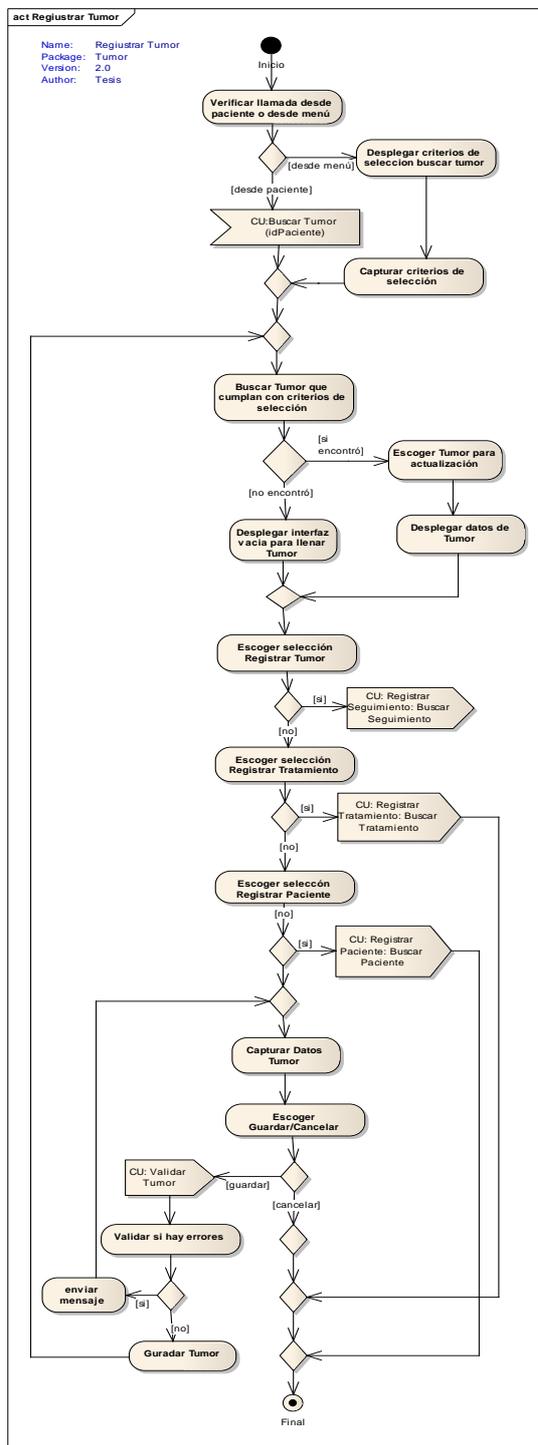
Se presentan los diagramas de actividades de los siguientes casos de uso:

- Registrar Paciente
- Registrar Tumor
- Registrar Seguimiento
- Registrar Tratamiento

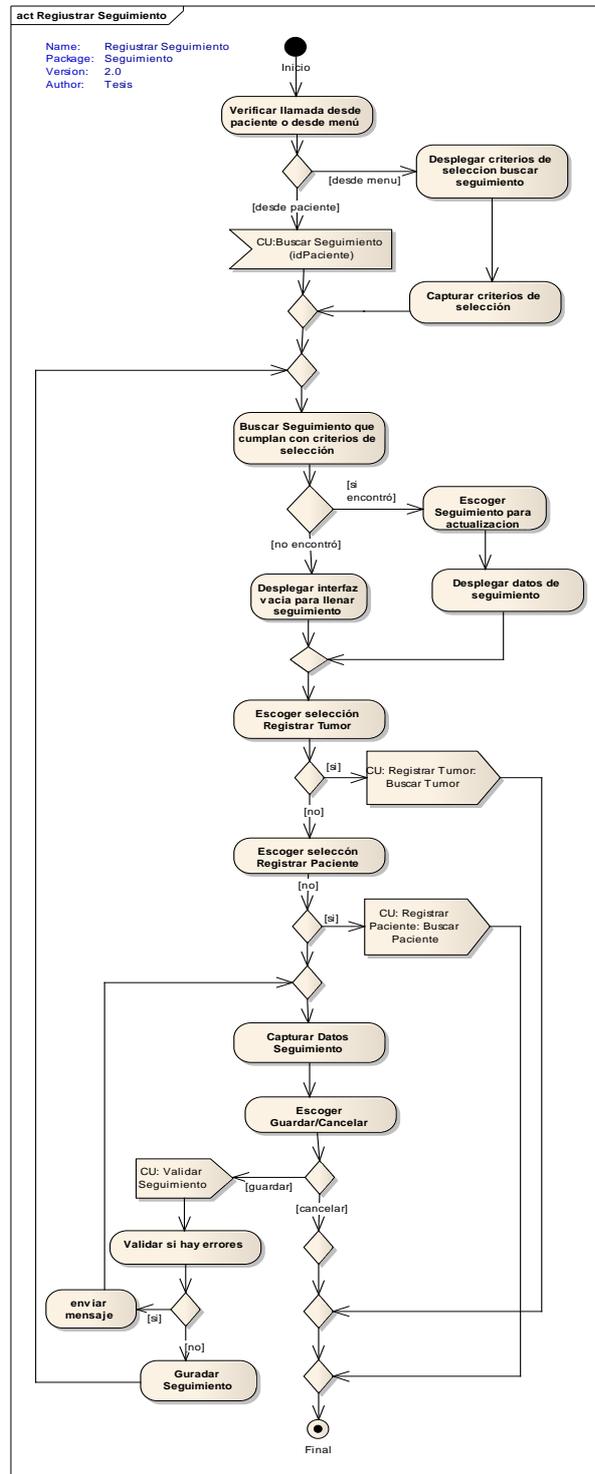
- Registrar paciente



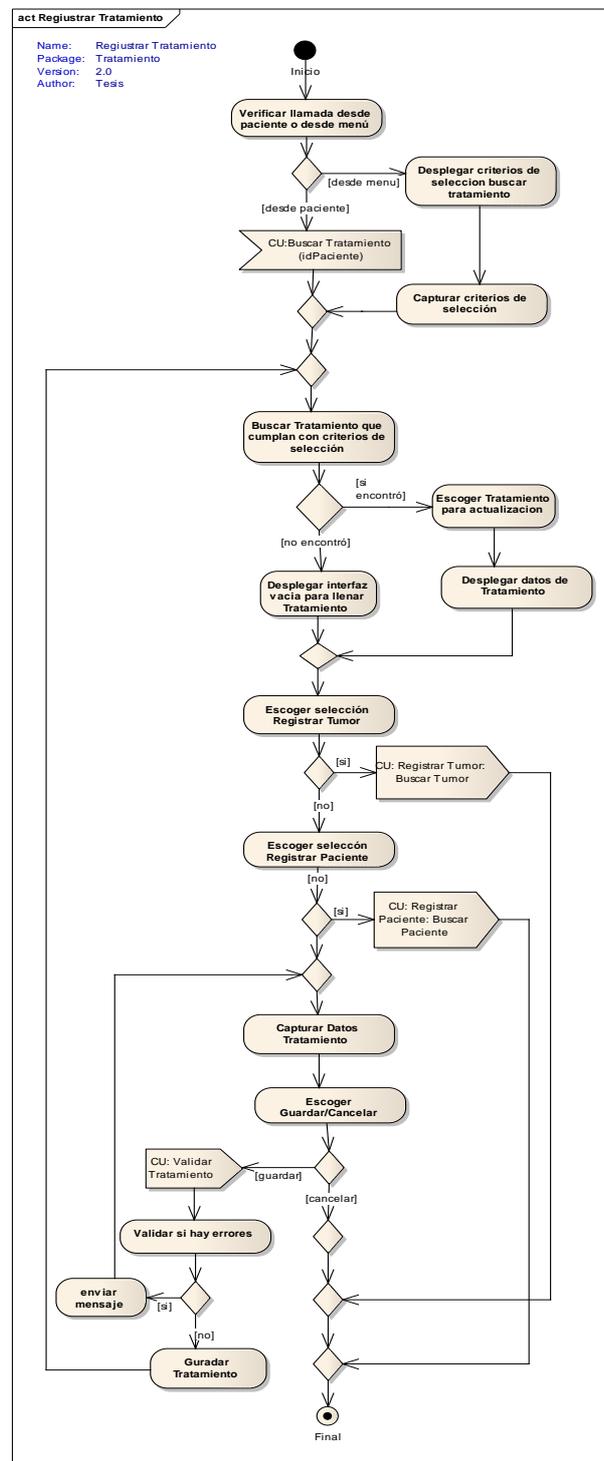
- Registrar tumor



- Registrar seguimiento

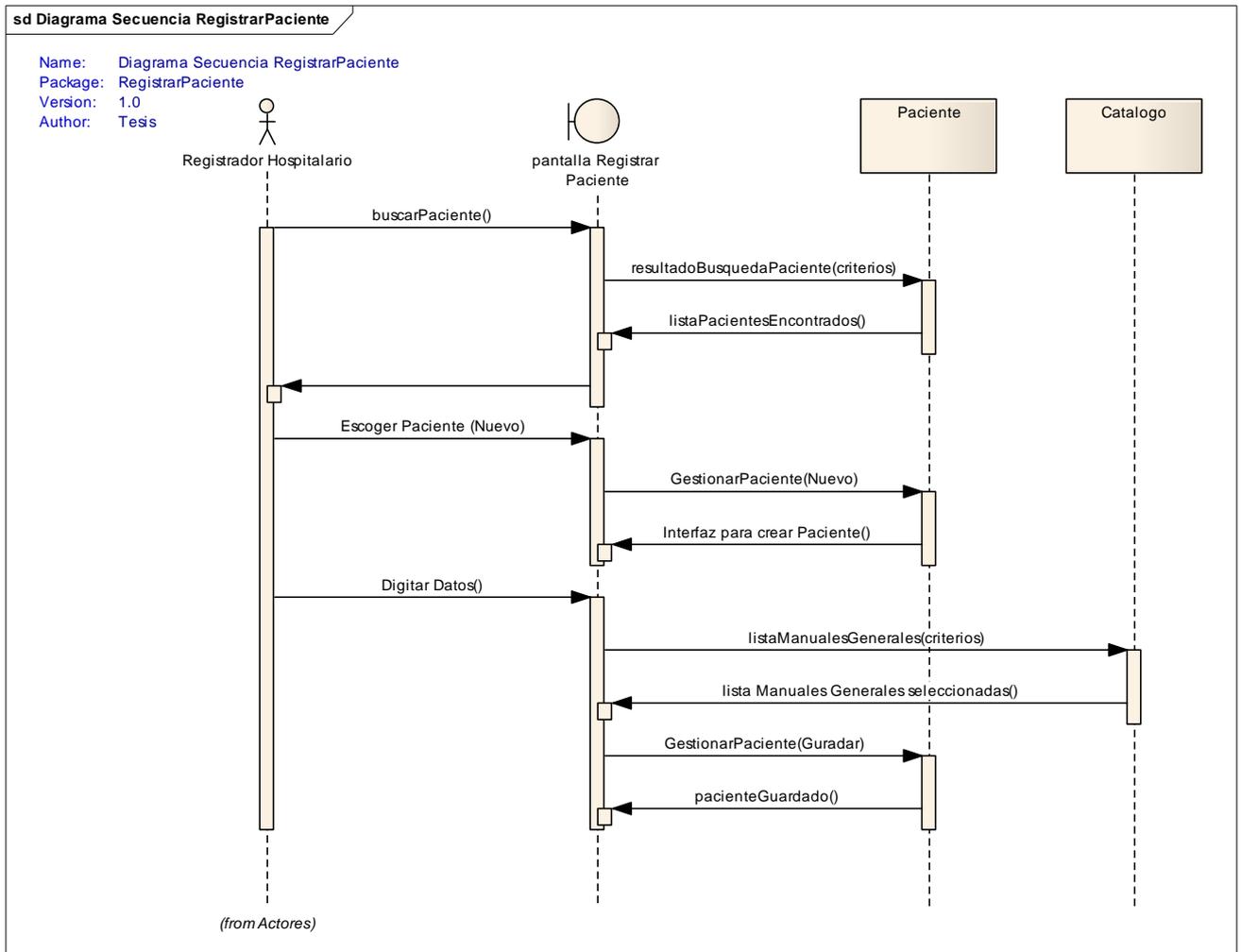


- Registrar Tratamiento

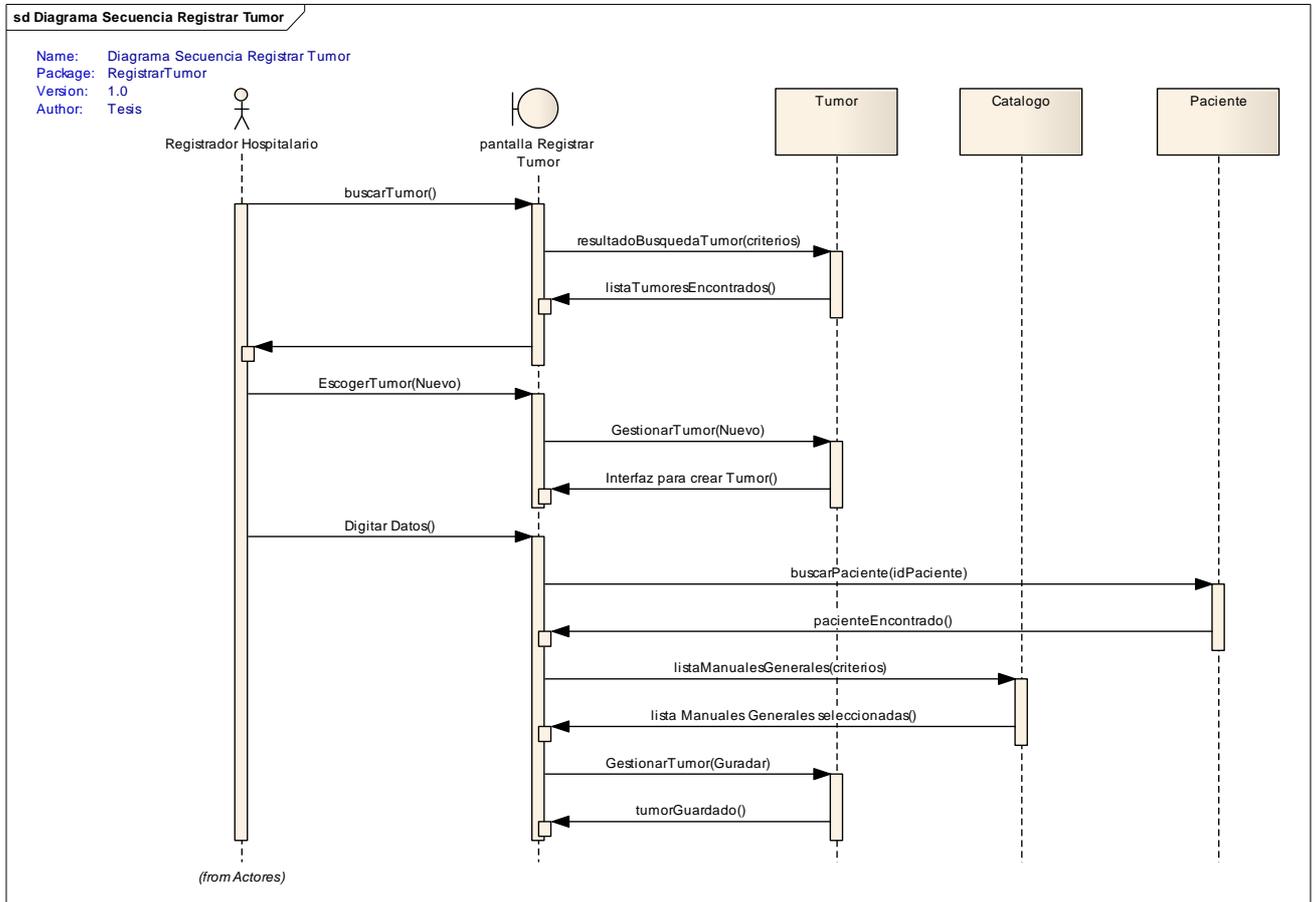


## Diagrama de secuencia

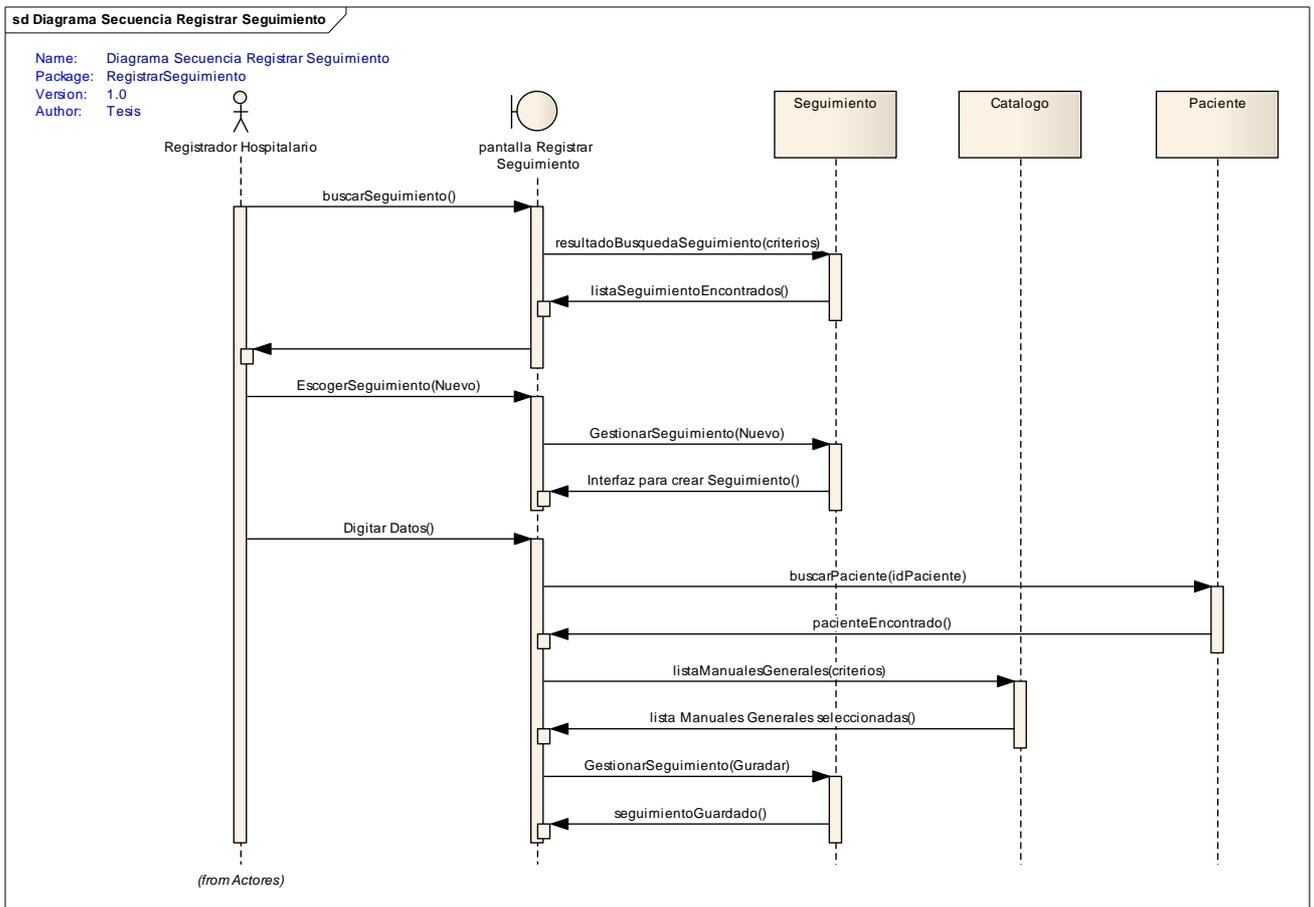
### • Registrar paciente



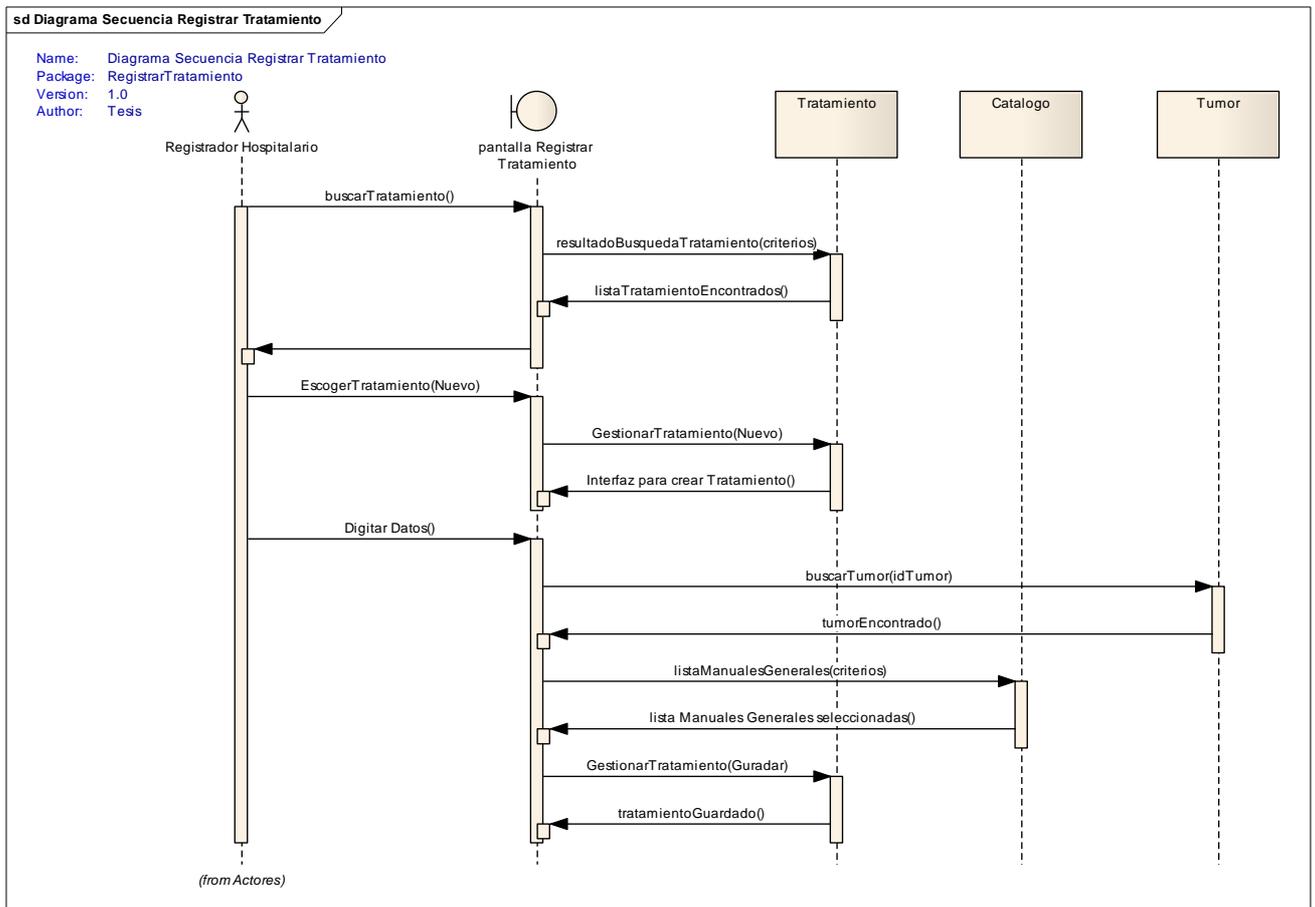
- Registrar tumor



## • Registrar seguimiento

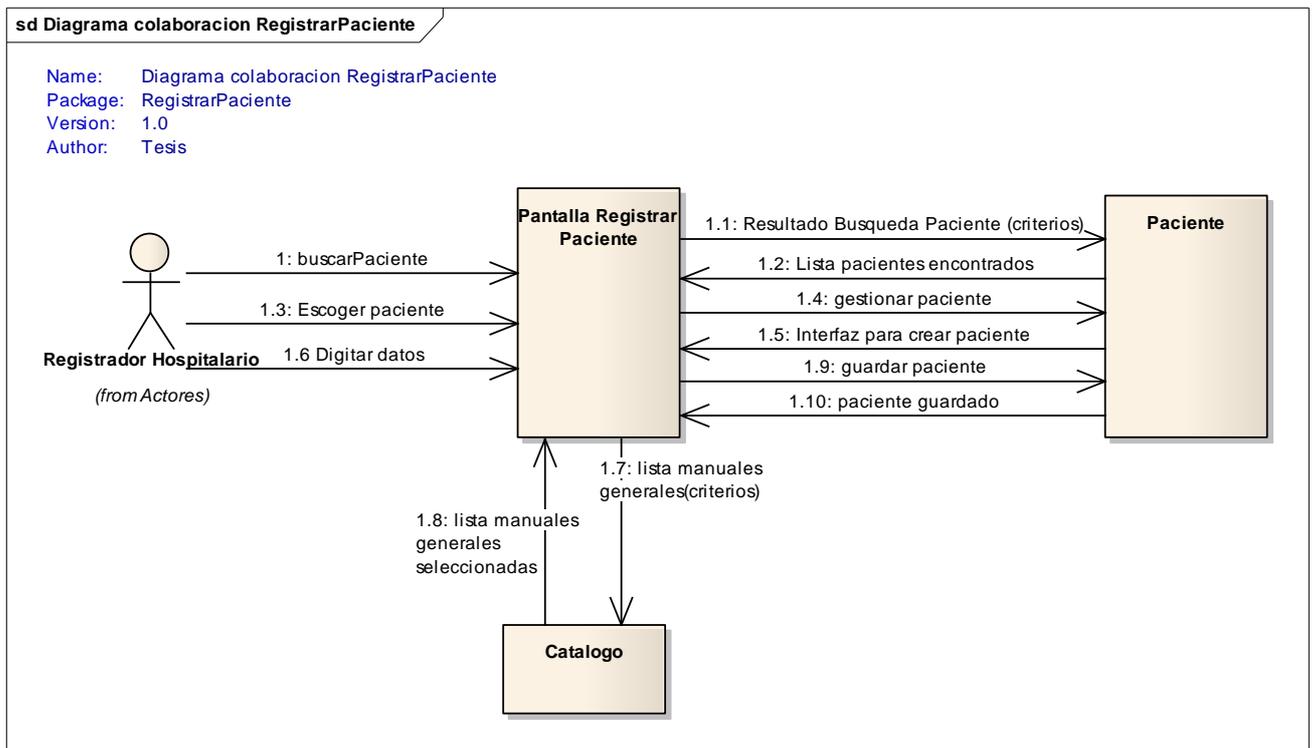


## • Registrar Tratamiento

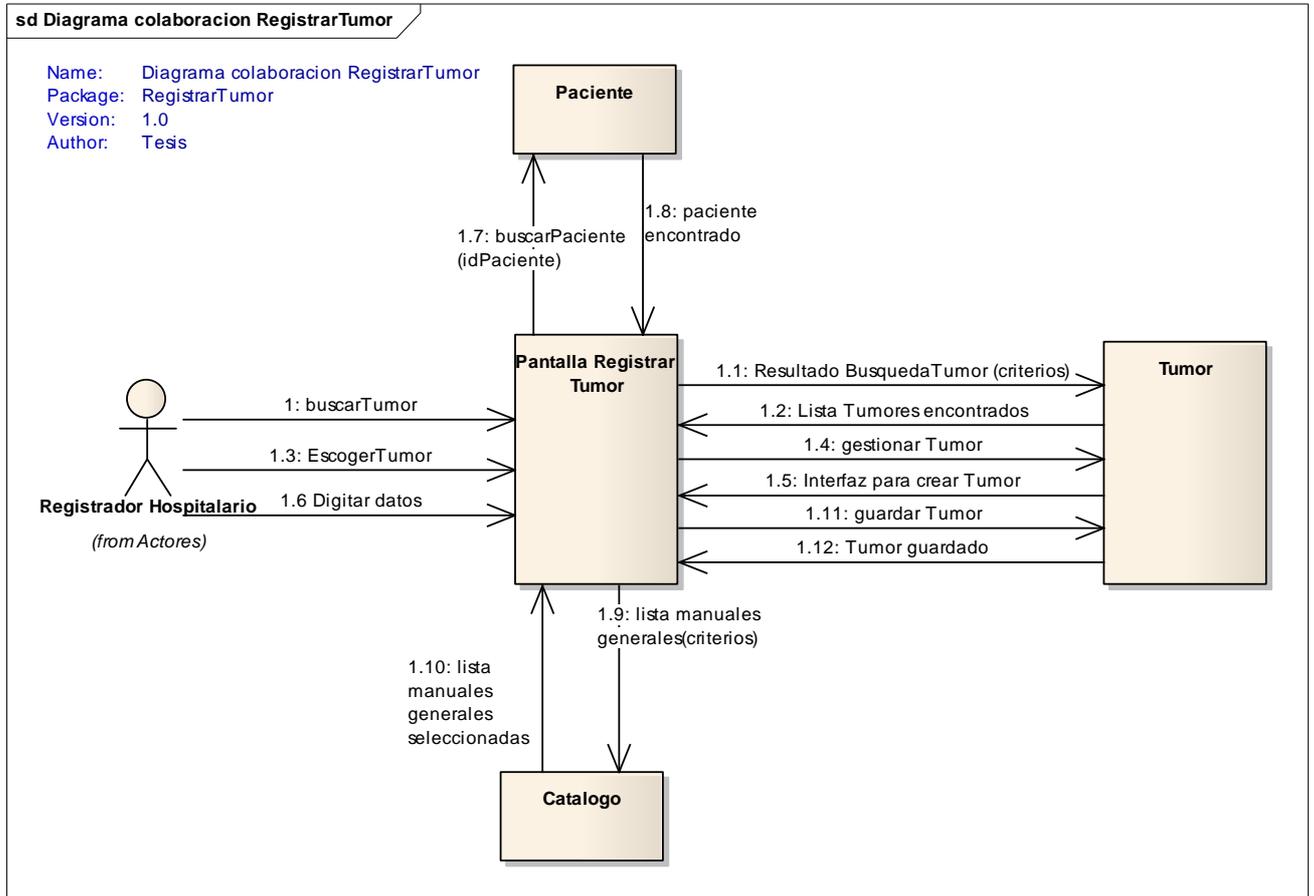


## Diagramas de colaboración

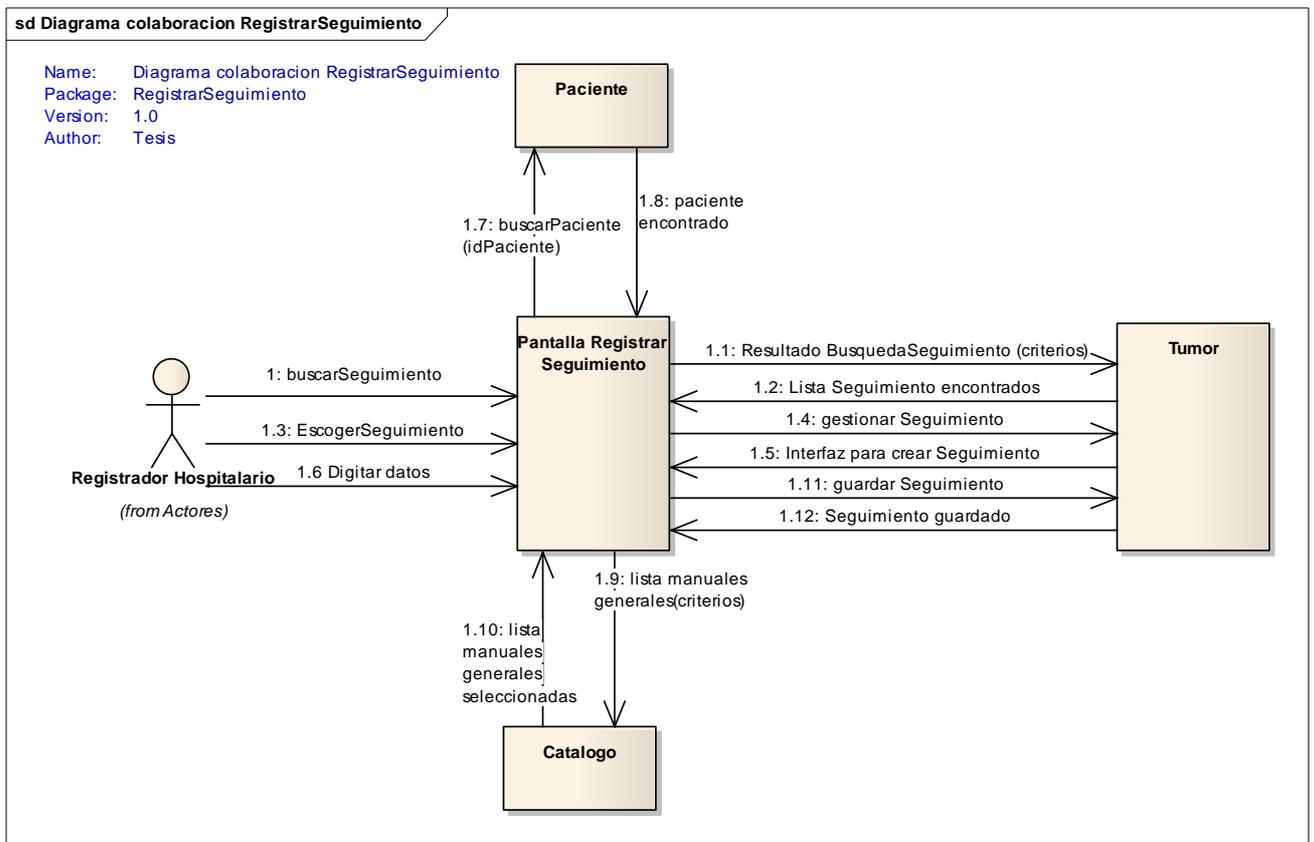
- Registrar paciente



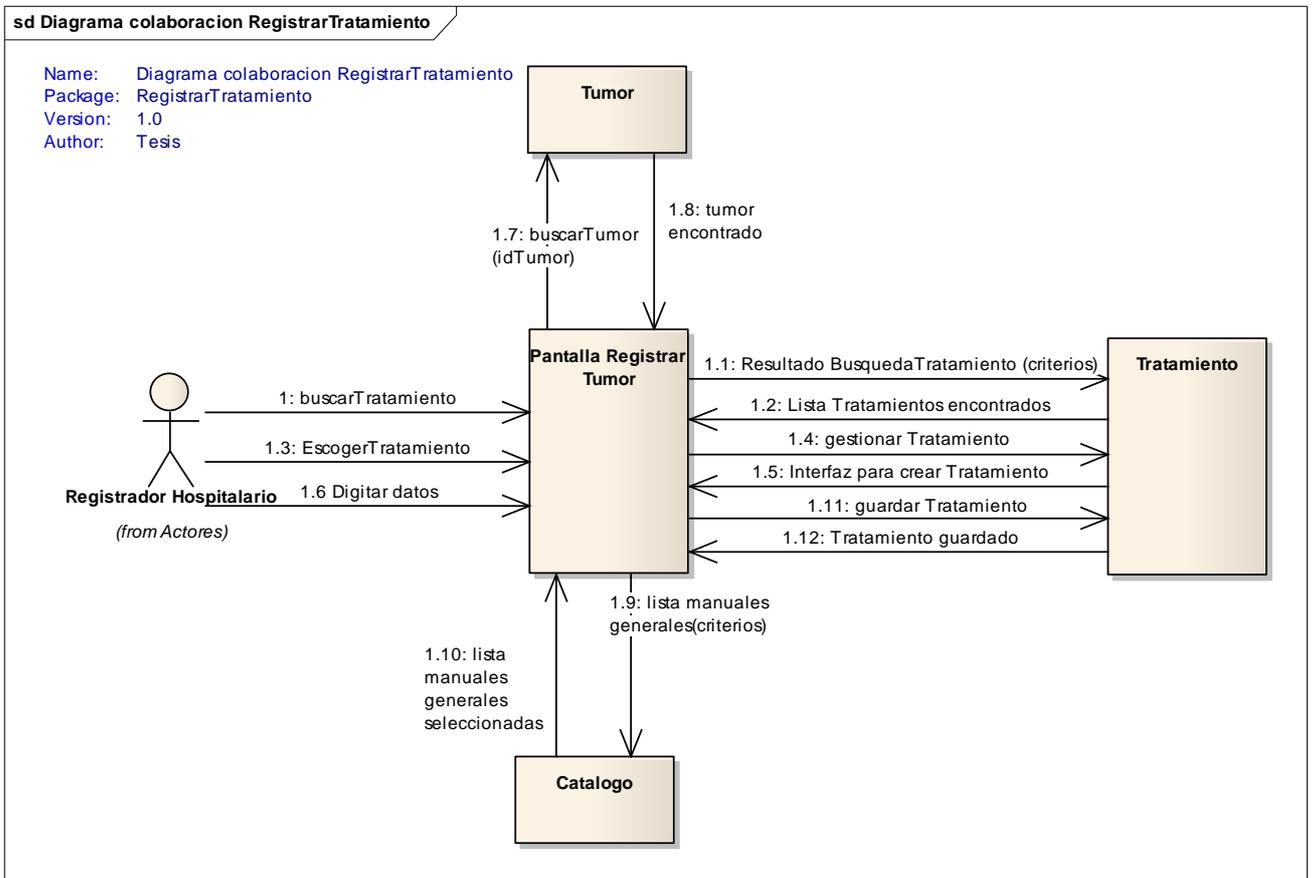
- Registrar tumor



## • Registrar seguimiento



## • Registrar Tratamiento

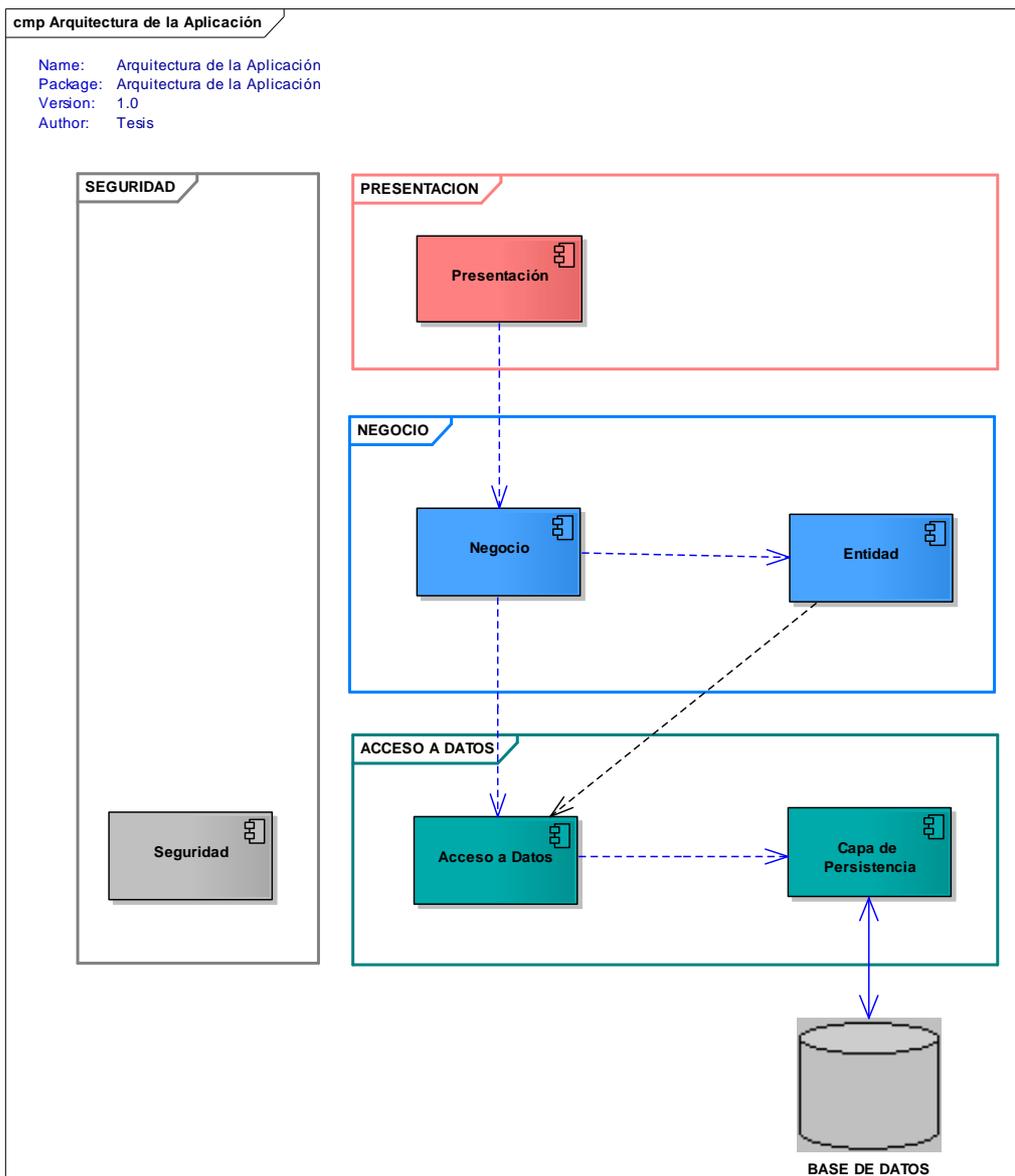


### 3.3.5 MODELO DE DESPLIEGUE

Se definió los componentes que se desarrollarán e instalarán.

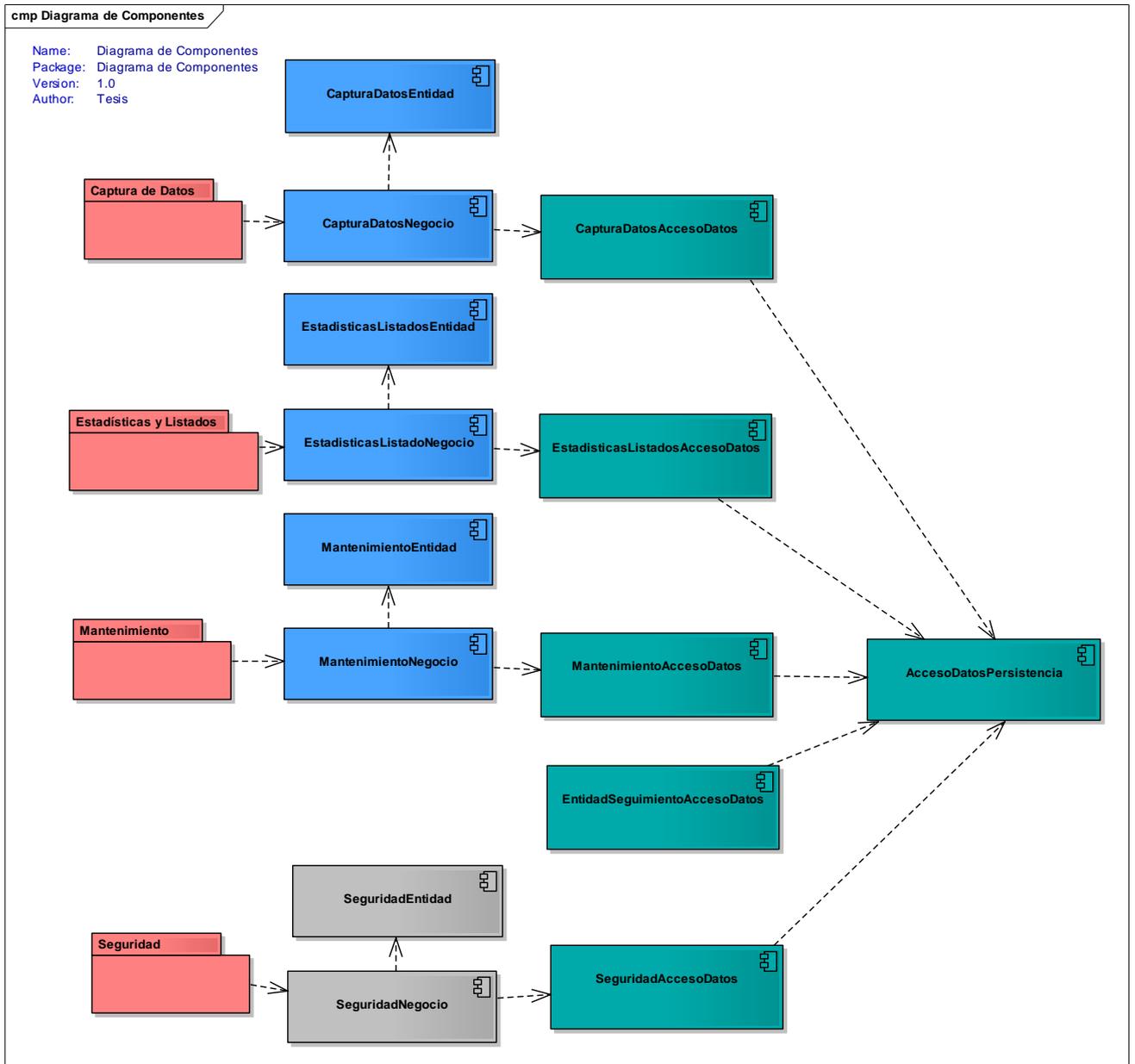
#### Arquitectura de la aplicación

Se definió la arquitectura de la aplicación en multicapas con el fin de que el sistema contenga un modelo vista controlador

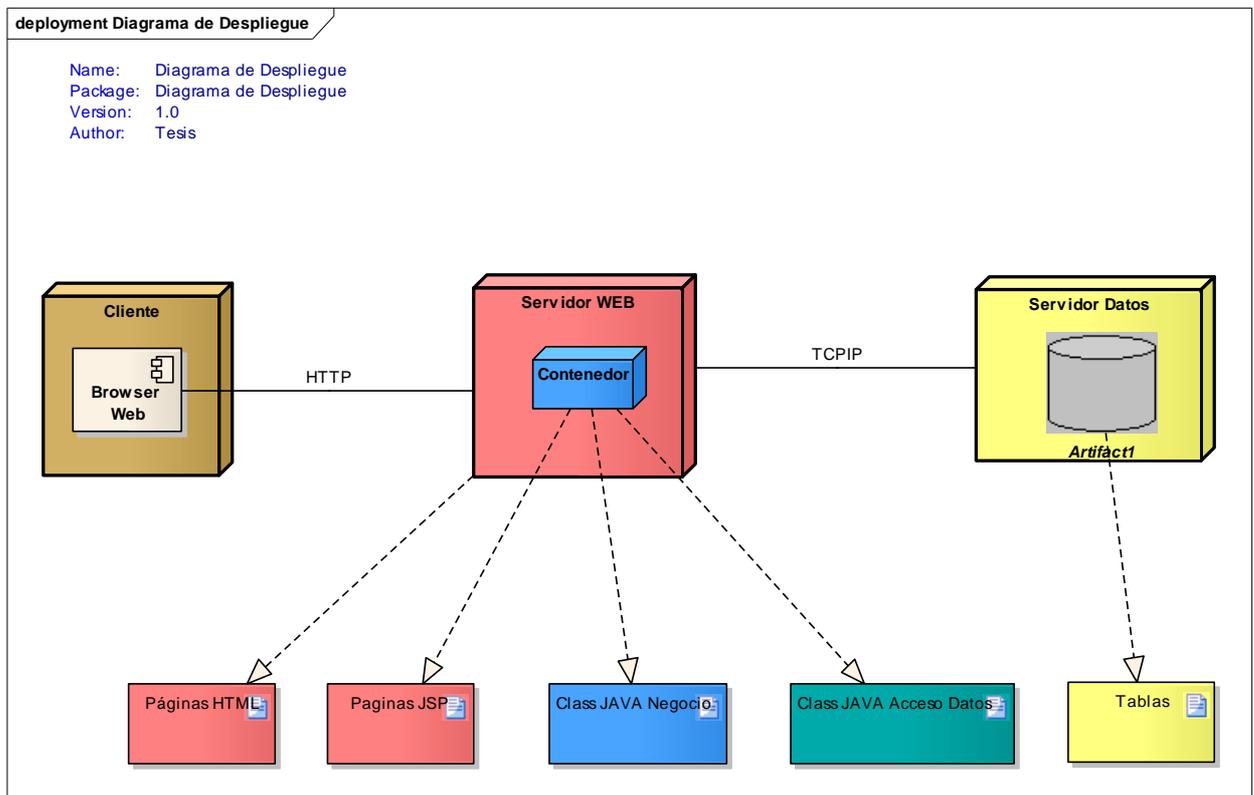


## Diagrama de componentes

Se definieron los componentes que se desarrollaran



## Diagrama de despliegue



### 3.4 IMPLEMENTACIÓN DEL SISTEMA

Poniendo en práctica los conocimientos formales de desarrollo de sistemas la implementación se hizo en base del diseño conseguido en este capítulo, guiado por casos.

En base del diseño estructural que contienen las clases de diseño se genero la base de datos, este script se corrió en ambiente Oracle para conseguir la estructura de datos que de soporte al sistema.

La codificación se llevo a cabo guiada por casos de uso, por cada caso de uso reconocido en el diagrama general del sistema se consideró su descripción y en el caso de ser complejo se tomaron en cuenta los diagramas de actividades y los de secuencia.

El diagrama de objetos sirvió para comprender con mayor detalle los ejemplos de datos que van a hacer incluidos.

Se codificó el sistema en base al diseño que se desarrollo.

Se realizó las pruebas unitarias y de integración considerando el desarrollo de Pruebas del software.

Se preparó la documentación orientada a los usuarios del sistema

Se cargó las tablas del sistema referenciales recogidos de textos que contienen normas del Registro Hospitalario.

La implementación comienza al final de la fase de iniciación, toma mayor desempeño en la fase de construcción y termina en la fase de transición. En este flujo se define la estructura de implementación del sistema.

### **3.5 DOCUMENTACIÓN DE PRUEBA PILOTO DE IMPLANTACIÓN**

Las pruebas empiezan desde la fase inicio y se desarrollan a través de todas las fases alcanzando un 100% en la fase de transición, logrando los objetivos planteados para el desarrollo del sistema.

#### **3.5.1 VALIDACIÓN DEL CUMPLIMIENTO DE REQUERIMIENTOS FUNCIONALES**

Es necesario validar que todos los requisitos hayan sido considerados para el desarrollo del sistema y que se los haya solucionado satisfactoriamente.

##### **3.5.1.1 Validación de Requerimientos Funcionales vs Diseño Funcional**

Se debe verificar que los requerimientos funcionales concebidos en la etapa de análisis hayan sido considerados en la etapa funcional, es decir que hayan sido solucionados en los casos de uso propuestos.

Se verificó que todos los requisitos funcionales hayan sido considerados en algún caso de uso y que cada caso de uso este orientado a satisfacer al menos un requisito funcional. Esto se consigue cruzando a través de una matriz los requisitos frente a los casos de uso.

A continuación se presenta la matriz de requisitos frente a casos de uso.

**MATRIZ**

### 3.5.1.2 Validación del Software

#### **Validación software vs Diseño**

Esta validación se la realizó antes de trabajar con los usuarios es una validación que hacen estrictamente por los desarrolladores. Se tomó como ejemplo el caso de uso Registrar Paciente, para esto:

- Se verificó que el código:
  - Cumpla con los estándares de programación.
  - Responda a la lógica de los casos de uso de acuerdo a su descripción, diagrama de actividades, diagrama de secuencia.
  - Las llamadas de las funciones consten en el sitio correcto que permita la navegabilidad adecuada.
- Se verificó que la base de datos:
  - Cumpla con los estándares para nombrar tablas y atributos.
  - Consten todos los atributos que la funcionalidad del caso de uso requiere.

#### **Validación software vs Requerimientos**

Para la validación del software se procedieron a ejecutar pruebas de unidad conjuntamente con el usuario usando la técnica de pruebas de caja negra, adicionalmente se realizaron pruebas de integración.

Esto permitió hacer una validación del software vs los requerimientos con el fin de constatar que los usuarios se sienten satisfechos.

## **Pruebas de Unidad**

Las pruebas de unidad se llevaron a cabo como dice el PUD guiadas por casos de uso.

Se utilizó la técnica de caja negra para la realización de las pruebas de unidad, para la ejecución de estas pruebas participaron los desarrolladores acompañados por los usuarios.

El objetivo de estas pruebas es revisar que la interfase sea consistente y que cada opción funcione correctamente de manera independiente entregando los resultados esperados.

En estas pruebas se realizó la validación de los métodos y funciones básicas del sistema.

Se verificó el correcto funcionamiento de los diferentes opciones de sistema que ayudan a la administración de datos, es decir las acciones que nos permiten guardar, modificar, eliminar y buscar datos.

## **REGISTRAR**

### ***Paciente***

- Se ingresó en la interfaz datos que no existían previamente en la base de datos, se ejecutó la opción guardar y el sistema validó que todos los datos obligatorios hayan sido ingresados caso contrario presenta un mensaje que los datos que faltan ingresar.

- Se ingresaron datos inconsistentes, el sistema presentó el mensaje de error en el que indica que en ese campo solo se deben ingresar números o letras según el caso.
- Se realizaron modificaciones en los datos ya ingresados en la base de datos, una vez corregidos el sistema guarda los datos.
- Se eliminó una fila seleccionada del grid, el sistema muestra un mensaje verificando si el usuario quiere borrarlo, en caso de que acepte borra el registro de la base de datos.

### ***Tumor***

- Se ingresó datos que no existían previamente en la base de datos, se ejecutó la opción guardar y el sistema validó que todos los datos obligatorios hayan sido ingresados de lo contrario presenta un mensaje con los datos que faltan ingresar.
- Se ingresaron datos inconsistentes, el sistema presenta el mensaje de error en el que indica que en ese campo solo se deben ingresar números o letras según el caso.
- Se realizó modificaciones en los datos ya ingresados en la base de datos, una vez corregidos el sistema guarda los datos.
- Se eliminó una fila seleccionada del grid, el sistema muestra un mensaje verificando si el usuario quiere borrarlo, en caso de que acepte borra el registro de la base de datos.

### ***Tratamiento***

- Se ingresó datos que no existían en la base de datos, se ejecutó la opción guardar y el sistema valida que todos los datos obligatorios hayan sido ingresados de lo contrario presenta un mensaje que los datos que faltan ingresar.

- Se ingresaron datos inconsistentes, el sistema presenta el mensaje de error en el que indica que en ese campo solo se deben ingresar números o letras según el caso.
- Se realizó modificaciones en los datos ya ingresados en la base de datos, una vez corregidos el sistema guarda los datos.
- Se eliminó una fila seleccionada del gird, el sistema muestra un mensaje verificando si el usuario quiere borrarlo, en caso de que acepte borra el registro de la base de datos.

### **Seguimiento**

- Se ingresaron datos que no existan en la base de datos, se ejecutó la opción guardar y el sistema valida que todos los datos obligatorios hayan sido ingresados de lo contrario presenta un mensaje que los datos que faltan ingresar.
- Se ingresó datos inconsistentes, el sistema presenta el mensaje de error en el que indica que en ese campo solo se deben ingresar números o letras según el caso.
- Se realizó modificaciones en los datos ya ingresados en la base de datos, una vez corregidos el sistema guarda los datos.
- Se eliminó una fila seleccionada del gird, el sistema muestra un mensaje verificando si el usuario quiere borrarlo, en caso de que acepte borra el registro de la base de datos.

### **Estadísticas y Listados**

- Se ingresó a la opción y se prueba que todos los reportes se estén ejecutando de manera concisa y que los datos presentados reflejen la realidad de la base de datos.

## **Mantenimiento**

### ***Catálogo***

- Se escogió el tipo de catálogo sobre el cual se quiere ingresar, modificar o eliminar datos.
- Se ingresaron datos que no existan previamente en la base de datos, se ejecutó la opción guardar y el sistema valida que todos los datos obligatorios hayan sido ingresados de lo contrario presenta un mensaje que los datos que faltan ingresar.
- Se ingresaron datos inconsistentes, el sistema presenta el mensaje de error en el que indica que en ese campo solo se deben ingresar números o letras según el caso.
- Se realizó modificaciones en los datos ya ingresados en la base de datos, una vez corregidos el sistema guarda los datos.
- Se eliminó una fila seleccionada del grid, el sistema muestra un mensaje verificando si el usuario quiere borrarlo, en caso de que acepte borra el registro de la base de datos.

### ***Morfología***

- Se ingresó datos que no existan previamente en la base de datos, se ejecutó la opción guardar y el sistema valida que todos los datos obligatorios hayan sido ingresados de lo contrario presenta un mensaje que los datos que faltan ingresar.
- Se ingresaron datos inconsistentes, el sistema presenta el mensaje de error en el que indica que en ese campo solo se deben ingresar números o letras según el caso.
- Se realizó modificaciones en los datos ya ingresados en la base de datos, una vez corregidos el sistema guarda los datos.

- Se eliminó una fila seleccionada del gird, el sistema muestra un mensaje verificando si el usuario quiere borrarlo, en caso de que acepte borra el registro de la base de datos.

### ***Topología***

- Se ingresaron el datos que no existan previamente en la base de datos, se ejecuta la opción guardar y el sistema valida que todos los datos obligatorios hayan sido ingresados de lo contrario presenta un mensaje que los datos que faltan ingresar.
- Se ingresaron datos inconsistentes, el sistema presenta el mensaje de error en el que indica que en ese campo solo se deben ingresar números o letras según el caso.
- Se realizaron modificaciones en los datos ya ingresados en la base de datos, una vez corregidos el sistema guarda los datos.
- Se eliminaron una fila seleccionada del gird, el sistema muestra un mensaje verificando si el usuario quiere borrarlo, en caso de que acepte borra el registro de la base de datos.

### ***Ocupación***

- Se ingresaronn el datos que no existan previamente en la base de datos, se ejecutó la opción guardar y el sistema valida que todos los datos obligatorios hayan sido ingresados de lo contrario presenta un mensaje que los datos que faltan ingresar.
- Se ingresaronn datos inconsistentes, el sistema presenta el mensaje de error en el que indica que en ese campo solo se deben ingresar números o letras según el caso.
- Se realizó modificaciones en los datos ya ingresados en la base de datos, una vez corregidos el sistema guarda los datos.

- Se eliminó una fila seleccionada del gird, el sistema muestra un mensaje verificando si el usuario quiere borrarlo, en caso de que acepte borra el registro de la base de datos.

### ***Médico***

- Se ingresó datos que no existan previamente en la base de datos, se ejecuta la opción guardar y el sistema valida que todos los datos obligatorios hayan sido ingresados de lo contrario presenta un mensaje que los datos que faltan ingresar.
- Se ingresaron datos inconsistentes, el sistema presenta el mensaje de error en el que indica que en ese campo solo se deben ingresar números o letras según el caso.
- Se realizó modificaciones en los datos ya ingresados en la base de datos, una vez corregidos el sistema guarda los datos.
- Se eliminó una fila seleccionada del gird, el sistema muestra un mensaje verificando si el usuario quiere borrarlo, en caso de que acepte borra el registro de la base de datos.

### ***Establecimiento***

- Se ingresaron el datos que no existan previamente en la base de datos, se ejecuta la opción guardar y el sistema valida que todos los datos obligatorios hayan sido ingresados de lo contrario presenta un mensaje que los datos que faltan ingresar.
- Se ingresaron datos inconsistentes, el sistema presenta el mensaje de error en el que indica que en ese campo solo se deben ingresar números o letras según el caso.
- Se realizó modificaciones en los datos ya ingresados en la base de datos, una vez corregidos el sistema guarda los datos.

- Se eliminó una fila seleccionada del grid, el sistema muestra un mensaje verificando si el usuario quiere borrarlo, en caso de que acepte borra el registro de la base de datos.

## **Seguridad**

### ***Cambiar Contraseña***

- Se ingresó la contraseña actual en caso de no ser la correcta el sistema despliega un mensaje de error y no podrá cambiar la contraseña
- Se ingresó la contraseña nueva y la confirmación de la contraseña nueva en caso de no ser iguales el sistema desplegará un error, una vez que sean iguales se cambiará la contraseña.

### **Pruebas de Integración**

Se escogió el método de integración descendente, se navega en el sistema de arriba hacia abajo las diferentes opciones verificando que cumplan tanto con la profundidad y anchura de los módulos.

Se revisó íntegramente el menú verificando que todos los casos de uso relacionados a actores consten como opciones del menú y que todos los casos de uso que todos los casos de uso que contengan extensión cuenten con un acceso en las correspondientes opciones.

Al realizar estas pruebas se pudo comprobar que al ingresar los datos de seguimiento, tumor y tratamiento no correspondían al paciente elegido, se desplegaban todos los datos existentes sin ser filtrados por el paciente escogido. Se realizó la corrección debida para que al escoger un paciente solo desplieguen los datos de seguimiento, tumor y tratamiento correspondientes a ese paciente.

También se verificó que los datos ingresados se reflejen correctamente en los reportes, esta prueba es a satisfacción del usuario.

### **3.5.2 VALIDACIÓN DEL CUMPLIMIENTO DE REQUERIMIENTOS NO FUNCIONALES**

#### **3.5.2.1 Recuperación**

Las pruebas de recuperación ayudan a validar el comportamiento del sistema luego de algún fallo imprevisto externo.

Estas pruebas son esenciales para asegurar que no exista pérdida ni inconsistencia de información.

Para el requerimiento no funcional "*A prueba de falla de energía eléctrica*", se procedió a cortar la energía eléctrica al momento en que se estaba ingresando datos, luego de recuperada la energía se verificó que todas las tablas se encuentran con sus datos íntegros y consistentes, a lo sumo se perdió las transacciones que estaban en medio de captura de datos, lo cual por tratarse de un sistema que no es en tiempo real y que las decisiones no son afectadas críticamente los usuarios reconocen y están de acuerdo que lo que deben hacer es volver a digitar los casos que estuvieron inconclusos.

Para el requerimiento no funcional "*Que se pueda recuperar los datos al dañarse el servidor total o parcialmente*" se siguió el siguiente procedimiento en presencia de los usuarios:

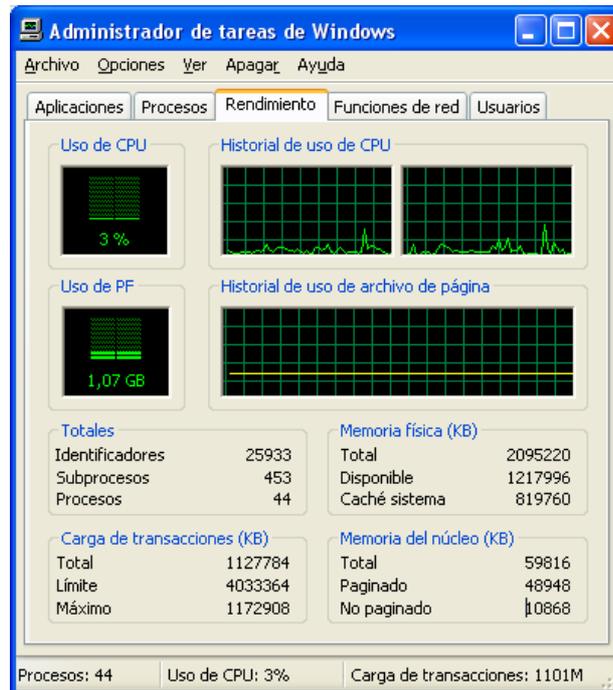
- Respaldo de la base de datos (BDD).
- Borrado de la BDD.
- Recuperación de la BDD.
- Recomendar a los usuarios la periodicidad de obtención de los respaldos y la periodicidad del reciclaje de los medios en el siguiente sentido:
  - En épocas de ingreso de grandes volúmenes de datos la periodicidad diaria y el reciclaje semanal.
  - Respaldos semanales con reciclaje mensual.
  - Respaldos mensuales con reciclaje semestral.
  - Respaldo semestrales sin reciclaje.

### **3.5.2.2 Rendimiento**

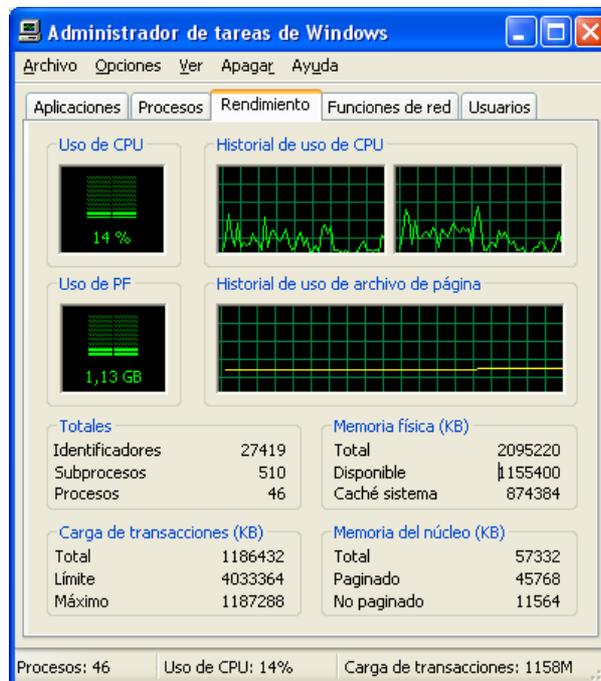
Las pruebas de rendimiento ayudan a comprobar *“Que el consumo de recursos del computador por parte del sistema no sea excesivo y que no afecte el correcto funcionamiento de otras aplicaciones”*.

Para la comprobación de este requerimiento se procedió a observar el rendimiento de la memoria y el procesador antes y después de la ejecución del sistema.

- Pantalla de rendimiento de memoria y procesador antes de la ejecución del sistema.



- Pantalla de rendimiento de memoria y procesador después de la ejecución del sistema



Las pantallas demuestran que el sistema no consume excesivos recursos del computador, con lo cual se verifica que el consumo del procesador y de la memoria es adecuado para las funciones que realiza, y ninguna aplicación se ve afectada por el mismo.

Estas pruebas se realizaron en una maquina que no tiene características de ser servidor, por tanto el desempeño del sistema en el servidor será mucho mejor.

### **3.5.2.3 Multiusuario**

Se instaló el sistema en un servidor y se ejecutó el sistema en dos usuarios entrando al mismo módulo al mismo tiempo y grabar datos de manera simultánea y ver si se graba los datos correctamente

### **3.5.2.4 Facilidad de uso**

Para validar que el sistema cuente con "*Interfaces de usuario sencillas y fáciles de operar*" se trabajo conjuntamente con los usuarios haciendo pruebas de navegabilidad por todas las opciones del sistema y recibiendo la aprobación y satisfacción de los usuarios que reconocieron que este sistema era de más fácil uso que otros con los que habían contado anteriormente

## **CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES**

### **4.1 CONCLUSIONES**

- El desarrollo del “Sistema de Registro Hospitalario para SOLCA Manabí Núcleo de Portoviejo” cumple con las necesidades de SOLCA en el ámbito administrativo, brinda al personal técnico médico y administrativo un soporte de calidad, permitiendo un control y seguimiento de los pacientes con cáncer que son tratados en SOLCA, por tanto los objetivos planteados para el proyecto se cumplieron.
- La guía metodológica utilizada en el desarrollo de este proyecto es práctica, de fácil uso y entendimiento para los desarrolladores de tal manera que se puede tener un control sobre el avance del proyecto asegurando el desarrollo de un producto de calidad.
- El desarrollo del proyecto, bajo una tutoría académica, permite a los desarrolladores aplicar la teoría de una manera correcta y orientarse en distintas cuestiones prácticas que surgen a lo largo del desarrollo, de tal manera que se construya un sistema correcto y sirva como base de experiencia para desarrollos posteriores.
- Entender los principales procesos administrativos del negocio ayuda de gran manera a reconocer los requerimientos funcionales y los actores que van interactuar con el sistema.

- Para garantizar la calidad del software a entregarse es importante revisar correctamente los requerimientos concebidos en el análisis vs en el diseño funcional y el software vs los requerimientos.

## 4.2 RECOMENDACIONES

Recomendamos:

- Que se hagan más pruebas con casos reales para validar en estado de producción al sistema y que se recojan nuevos requerimientos de reportes ya que con la gran cantidad de datos que se manejan se podría generar más información válida.
- No descartar posibles mejoras o cambios a los datos y procesos propuestos según futuros requerimientos.

## BIBLIOGRAFÍA

- ARLOW Jim, NEUSTADT Ila. UML AND THE UNIFIED PROCESS PRACTICAL OBJECT-ORIENTED ANALYSIS A DESING. Addison-Wesley. Estados Unidos de América. 2002
- Jacobson, Ivar; Booch, Grady; Rumbaugh, James. El Proceso Unificado de Desarrollo de Software; Madrid; Addison Wesley – Pearson Educación S.A.; 2000
- Rumbaugh, James; Jacobson, Ivar; Booch, Grady El Lenguaje Unificado de Modelado, Manual de Referencia; Madrid; Addison Wesley – Pearson Educación S.A.; 2000
- Pressman, Roger Ingeniería del Software; Mc.Graw Hill, Interamericana de España S.A, 5ta.
- BOOCH Grady, RUMBAUGH James, JACOBSON Iván; El Lenguaje Unificado de Modelado, Primera Edición, Addison Wesley, 2000.
- PRESSMAN, S. Roger; Ingeniería del Software un enfoque práctico, Sexta edición, MacGraw-Hill Interamericana editores S.A. 2006.
- Varios autores Documentación sobre UML, PUD, Arquitecturas n capas
- FERRÉ, Xavier; SÁNCHEZ, María. Desarrollo Orientado a Objetos con UML. <http://www.clikear.com/manuales/uml>.
- <http://www.dcc.uchile.cl/~luguerre/cc61j/recursos/clase2.ppt>.
- Estructura del Proceso Unificado de Desarrollo  
<http://www.reynox.com/publicaciones/images/metodologia-rup.jpg>