

**Universidad de las Ciencias Informáticas  
Facultad 1**



**Título:** Análisis y diseño del Sistema de Control de Tiempo en proyectos de desarrollo de Software en el Centro de Identificación y Seguridad Digital.

**Trabajo de Diploma para optar por el título de Ingeniero Informático**

**Autor:** Sissy Ramos López

**Tutor:** Ing. Yasef Barbán Freixas

**Ciudad de la Habana**

**Junio, 2009**



*El placer supremo es obtener lo que se anhela.*

*Thales De Mileto*

# *Declaración de Autoría*

---

## **DECLARACIÓN DE AUTORÍA**

Declaramos que somos los únicos autores del trabajo titulado:

**Análisis y diseño de un Sistema de Control de Tiempo en proyectos de desarrollo de Software en el Centro de Identificación y Seguridad Digital**

y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Sissy Ramos López

\_\_\_\_\_  
Ing. Yasef Barbán Freixas

# *Opinión del Tutor*

---

## **OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA**

Título:

Autor:

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de \_\_\_\_\_.

\_\_\_\_\_ Días del mes de \_\_\_\_\_ del año 2009.

\_\_\_\_\_  
Ing. Yasef Barbán Freixas

# Agradecimientos

---

## AGRADECIMIENTOS

*A esta Universidad por haberme formado como profesional.*

*Mamita linda, quiero agradecerte por siempre darme todo lo que he querido y brindarme tu amor, apoyo, preocupación, ocupación, sabiduría y demás virtudes que te hacen adorable. Gracias por todas las malas noches que pasaste junto a mí y los corretajes por cualquier problema concerniente a mi vida. Eres lo más grande que tengo y espero que lo sepas.*

*A mi papi Willy, por tener ese corazón tan grande y de oro. Por dejarme ser tu única hijita; la más consentida, chiquitica, la luz de tus ojos. Siempre que necesites a alguien con quien confabularte sabes que me tienes a mí. Gracias por preocuparte y ocuparte tanto de mí.*

*Rodnito, eres el mango de mi vida. El destino te ha alejado de mí por un tiempo pero sabes que siempre te llevo en el corazón y te quiero muchísimo. Espero que siempre me dejes jalarte las orejas cuando hagas algo mal aunque sea la hermana menor. A ti bro, por todas las peleas y consejos que me enseñaron mucho.*

*A mi papa Raúl, aunque no estés aquí te tengo siempre presente. Gracias por tantas ocupaciones para conmigo y darme cariño, aunque sea de lejos aquí llega.*

*A mis tíos Nancy, Nelson y Raúl.*

*A todos los Ramos que aunque casi nunca nos vemos han aportado granitos de arena en mi vida.*

*A Danielita por ser la hermana que no tuvo mi mamá, por tantos años compartiendo estudios, discusiones, ruedas de casino, travesuras en bicicleta, disco, chismes, historias, secretos, en fin miles de cosas que te hacen ser especial.*

*A bebu, porque en este corto tiempo me has enseñado muchas cosas. Gracias por tus cariños, preocupaciones, risas. Ya no tendrás que decirme más "Donte para la tesis".*

*A mis primos.*

*Al pikete de la secundaria, Diadra, Ceci, Fanet, Ivet, Joan y Xavier. ¡Miren que grandes ya nos hemos hecho!!!!... Todas esas fiestas y miles de anécdotas.*

# Agradecimientos

---

*A todos mis otros amigos que son muchos ya que soy una muchacha tan sociable!!...jajaja...Gracias a todos por permitirme compartir con ustedes tantos momentos. En especial a Maikel por aguantarme todas mis malacrianzas.*

*A Gelsy por consentirme y dejarme hacer todo lo que quiero sin pedir nada a cambio. A toda esa familia tan buena gente, agradable, preocupada. Gracias por todos los vasos de leche.*

*A Dayeisi, Fudeikis, Osneidis, loquitas ustedes hacen que no me quiera ir de aquí. Gracias por su apoyo en este cortísimo tiempo en que nos hemos conocido, que lastima no haberlas conocido antes.*

*A mi tutor y amigo, eres terrible pero sabes que te tengo tremendo aprecio. Gracias por tu apoyo y ayuda.*

*A las locas de este y todos los apartamentos donde he vivido.*

*En general a todos los que de una forma u otra me han apoyado en toda mi vida, si no los menciono es porque saben que soy un poco despistada y ando siempre como loca.*

*Gracias a todos.*

# Dedicatoria

---

## DEDICATORIA

*A mi abuela querida, estés en donde estés, sé que hubieras estado orgullosa de ver en lo que me he convertido y todo lo que he sido capaz de hacer.*

*A mis padres por siempre estar ahí para mí, por ser siempre mis guías y ejemplo a seguir, por su paciencia y dedicación y por su apoyo incondicional en todo.*

*A mi hermano que lo adoro, espero siempre ser para ti más que una hermana, una amiga.*

## **RESUMEN**

La gestión de proyectos es una disciplina reciente que en las últimas décadas del siglo pasado se dedicó al estudio e investigación basados en la planificación del trabajo y control de la ejecución. En el área de Gestión del Tiempo del Proyecto, inmersa en la gestión de proyectos, se realiza toda la planificación de las tareas, recursos, entre otros procesos imprescindibles para la culminación de un proyecto en tiempo. La gestión de tareas de un proyecto es uno de los procesos de mayor envergadura dentro de la gestión de proyectos. Es importante dirigir el trabajo a la integración de estos elementos para así obtener un producto con la calidad requerida. El presente trabajo tiene como objetivo el diseño de un sistema para el control del tiempo de desarrollo de los proyectos pertenecientes al Centro de Identificación y Seguridad Digital.

La investigación consiste en la realización del análisis y diseño de una aplicación de escritorio desarrollada sobre la plataforma J2EE, con conexión mediante RSS al gestor de tareas que contiene las tareas asignadas a cada desarrollador y cuyas duraciones están previstas a partir de un cálculo razonable del tiempo necesario para cada una. El sistema llevará un control del tiempo invertido por cada trabajador en cada aplicación o software utilizado. Al finalizar cada sesión de trabajo, posibilitará generar reportes con información leal del tiempo utilizado por cada trabajador en el desarrollo de sus tareas.

## **PALABRAS CLAVE:**

Gestión de Proyectos, Gestión del Tiempo del Proyecto, Gestión de Tareas, Control del Tiempo.

# ÍNDICE

<b>INTRODUCCION</b> .....	1
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA</b> .....	6
1. Introducción.....	6
1.1. Conceptos asociados al dominio del problema .....	6
1.1.1. Gestión de Proyectos .....	6
1.1.2. Gestión del Tiempo.....	8
1.2. Uso de la gestión de tiempo .....	10
1.3. PSP (Proceso de Software Personal) .....	10
1.4. Gestión de Tareas.....	12
1.4.1. Integración de la gestión de tareas con otras áreas .....	13
1.5. Los trabajadores en el Proceso de desarrollo de software. ....	13
1.6. Sistemas para la gestión o control de tiempo.....	14
1.6.1. Sistema para el control de tiempo de producción en los proyectos productivos de la universidad de las Ciencias Informáticas.....	14
1.6.2. Sistema de gestión de tiempo de los estudiantes vinculados al proyecto PROCYON. ....	15
1.7. Herramientas de gestión de tareas que presentan elementos para la gestión de tiempo. ....	15
1.7.1. Time Tracker 4.4 .....	15
1.7.2. Process Dashboard .....	16
1.8. Metodología seleccionada para la solución del problema.....	17
1.9. Lenguaje de Modelado .....	20
1.10. Herramienta CASE a utilizar .....	21
1.10.1. Visual Paradigm.....	21
1.11. Plataforma de desarrollo.....	22
1.11.1. Java .....	22
1.11.2. Java 2 Platform, Enterprise Edition (J2EE).....	23
Conclusiones Parciales.....	24
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA</b> .....	25
2. Introducción.....	25
2.1. Situación Problemática.....	25
2.2. Flujo actual de los procesos involucrados en el campo de acción. ....	25
2.3. Debilidades de los procesos actuales. ....	26
2.4. Sindicación Simple Real (RSS).....	26
2.5. Propuesta Solución .....	27
2.6. Modelo de Dominio .....	27
2.6.1. Realización del Modelo de Dominio .....	28
2.6.2. Conceptos asociados al dominio.....	29
2.7. Especificación de los Requisitos del Software.....	30
2.7.1. Requerimientos funcionales.....	30
2.7.2. Requerimientos no funcionales .....	31
2.8. Modelo de Casos de Usos del Sistema.....	32
2.8.1. Actores del Sistema.....	32
2.8.2. Casos de Uso del Sistema.....	32
2.8.3. Diagrama de Casos de Uso del Sistema.....	33
2.8.4. Especificación textual extendida de los casos de uso.....	34
Conclusiones Parciales.....	43
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA</b> .....	44
3. Introducción.....	44
3.1. Modelo de Análisis .....	44
3.1.1. Diagramas de clases del análisis .....	44
3.2. Modelo del Diseño.....	47
3.2.1. Definiendo la Arquitectura.....	47
3.2.1.1. Estilo Arquitectónico usado.....	48
3.2.2. Estructura en Paquetes del Diseño .....	49

3.2.3.	Diagramas de Clases del Diseño.....	51
3.2.4.	Diagramas de secuencia del Diseño.....	51
3.2.5.	Patrones utilizados en el diseño.....	51
3.3.	Modelo de Datos.....	52
3.4.	Modelo de Despliegue.....	53
	Conclusiones Parciales.....	54
	<b>Conclusiones Generales del Trabajo.....</b>	<b>55</b>
	<b>Recomendaciones.....</b>	<b>56</b>
	<b>Referencia Bibliográfica.....</b>	<b>57</b>
	<b>Bibliografía.....</b>	<b>59</b>
	<b>Glosario de Términos.....</b>	<b>61</b>
	<b>Anexos.....</b>	<b>65</b>

## ÍNDICE DE TABLAS

Tabla 2.1 Conceptos del dominio .....	30
Tabla 2.2 Definición de los actores del sistema .....	32
Tabla 2.3 Descripción CUS Obtener Tarea .....	36
Tabla 2.4 Descripción CUS Registrar software y uso.....	37
Tabla 2.5 Descripción CUS Guardar Información .....	39
Tabla 2.6 Descripción CUS Ordenar Información .....	40
Tabla 2.7 Descripción CUS Obtener Reporte .....	42

## ÍNDICE DE FIGURAS

Figura 1.1 Descripción general de la Gestión del Tiempo del Proyecto.....	9
Figura 1.2 Fases e iteraciones de la Metodología RUP .....	19
Figura 2.1 Modelo de Dominio .....	29
Figura 2.2 Diagrama de Casos de Uso del Sistema.....	33
Figura 3.1 Diagrama de Clases del Análisis CU Obtener Tarea.....	45
Figura 3.2 Diagrama de Clases del Análisis CU Registrar Software y Uso.....	45
Figura 3.3 Diagrama de Clases del Análisis CU Ordenar Información.....	46
Figura 3.4 Diagrama de Clases del Análisis CU Guardar Información.....	46
Figura 3.5 Diagrama de Clases del Análisis CU Obtener Reporte .....	47
Figura 3.6 Arquitectura en capas del sistema.....	48
Figura 3.7 Diagrama de Paquetes.....	49
Figura 3.8 Modelo Físico de Datos.....	53
Figura 3.9 Diagrama de Despliegue .....	53

## INTRODUCCION

La gestión de proyectos es una disciplina relativamente reciente que comenzó a forjarse en los años sesenta como respuesta a las necesidades de la industria militar y seguidamente las de otras industrias. Aparecieron técnicas determinadas como cronogramas, los conceptos de ciclo de vida del proyecto y descomposición en tareas.

La gestión de proyectos, desarrollada en las últimas décadas del siglo pasado, se basa en la planificación del trabajo y en el posterior seguimiento y control de la ejecución. La planificación se realiza sobre un análisis detallado del trabajo que se quiere realizar y su descomposición en tareas. Parte, por tanto, de lo que se quiere hacer. Sobre esa información se desarrolla un plan adecuado de los recursos y tiempos disponibles y, durante la construcción, se sigue de cerca la ejecución para detectar posibles desviaciones y tomar medidas para corregirlas. (PALACIOS, 2006)

Se pueden mencionar como organizaciones de mayor relevancia en la rama las siguientes:

- ✓ International Project Management Association (IPMA), fundada en 1965.
- ✓ Project Management Institute (PMI), constituido en 1965.
- ✓ Prince2, que comenzó a trabajar en 1989.

IPMA y PMI surgieron como organizaciones profesionales para el desarrollo de conocimientos, metodologías y procesos para la gestión de proyectos. Prince2 ha tenido la evolución inversa, comenzó siendo una metodología, y se ha ido creando una organización. (PALACIOS, 2006)

La asignación de tiempo para el desarrollo de tareas es un proceso del que depende en gran medida la calidad del producto a desarrollar.

Se puede evidenciar que en la mayoría de las ocasiones los proyectos de desarrollo de software, aún haciendo una correcta gestión del tiempo del mismo, se atrasan debido, entre otros factores, a que determinar cuánto tiempo es realmente utilizado en función de la producción se hace difícil y esto influye en que un gran por ciento del tiempo de máquina destinado a realizar las tareas del proyecto no se esté aprovechando debidamente.

# Introducción

---

El surgimiento de la Universidad de las Ciencias Informáticas (UCI) y el empleo de las nuevas tecnologías, conllevó a realizar una ardua tarea para lograr de la misma un polo productivo con la participación activa de los estudiantes que en ella se forman. La misión de la UCI es producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación.

Actualmente, en los proyectos de desarrollo de software, del Centro de Identificación y Seguridad Digital, se realiza la estimación del tiempo y el cronograma de actividades, teniendo en cuenta el criterio de que, los trabajadores utilizan a plenitud el horario de trabajo que se les asigna para el desarrollo de las actividades. Debido a disímiles cuestiones a las que pueden estar sometidos muchos de los que trabajan en el mismo, sucede, la mayor parte del tiempo de desarrollo de un producto, que las actividades del cronograma se atrasan y, una de las causas es que no existe un sistema que controle el tiempo de desarrollo destinado a las tareas productivas y los líderes desconocen lo que hacen los trabajadores en ese tiempo.

Teniendo en cuenta esta **Situación problemática** que dio origen a dicha investigación, surge la **necesidad** de obtener el análisis y diseño de un sistema con el fin de controlar eficientemente el tiempo de desarrollo de las tareas de los proyectos productivos y brinde una solución aplicable a los problemas presentes en dicho centro.

Dada esta situación el **Problema a resolver** es el siguiente: ¿Cómo controlar el uso de tiempo en las diferentes actividades asignadas a los trabajadores en un proceso de desarrollo de software en el Centro de Identificación y Seguridad Digital?

Por lo que se define como **Objeto de estudio**: Procesos de gestión del tiempo de proyectos de desarrollo de software. Por cuanto se especifica el siguiente **Campo de Acción**: Procesos de gestión del tiempo de proyectos de desarrollo de software del Centro de Identificación y Seguridad Digital.

Teniendo en cuenta lo anterior, se plantea la siguiente **Idea a defender**:

*El desarrollo del análisis y diseño de un sistema que controle el tiempo de desarrollo de las actividades de un proyecto productivo permitirá sentar la base para la implementación de un sistema informático capaz de controlar el uso del tiempo.*

# Introducción

---

Determinándose como **Objetivo general** el siguiente: Realizar el análisis y diseño de un sistema que controle el uso del tiempo de proyectos de desarrollo de software del Centro de Identificación y Seguridad Digital.

Se trazan los siguientes **Objetivos específicos**:

- ✓ Aplicar los procesos del Área de Conocimiento Gestión del Tiempo del Proyecto implicada en la Gestión de Proyectos.
- ✓ Identificar soluciones que gestionen y controlen el uso del tiempo en el desarrollo de software.
- ✓ Concebir una propuesta de solución de software para la gestión y evaluación del tiempo de los estudiantes involucrados en el desarrollo de software.
- ✓ Desarrollar el análisis y diseño del prototipo funcional de la propuesta de solución.

Para darle cumplimiento a los objetivos de ejecutarán las siguientes **Tareas a cumplir**:

- ✓ Análisis de los procesos contenidos en la Gestión del Tiempo del Proyecto.
- ✓ Comparación de soluciones existentes para la Gestión del Tiempo del Proyecto.
- ✓ Descripción de los requisitos que debe cumplir el sistema.
- ✓ Realización del análisis de los procesos a automatizar.
- ✓ Desarrollo del diseño del sistema.
- ✓ Modelación de los diferentes diagramas que describen el diseño del sistema.

Se tienen como **Posibles resultados**: Obtener el análisis y diseño de un sistema que controle el tiempo de los trabajadores implicados en el desarrollo de productos de software para llevar a cabo distintas actividades.

Posibilitará la posterior implementación de un sistema de control del tiempo.

La **Población** que se aborda son los tres proyectos de desarrollo de software del Centro de Identificación y Seguridad Digital. Para esto se ha estudiado una **muestra** del proyecto dirigido a la línea de desarrollo de SOA y BPM.

**Aportes de la propuesta:**

# *Introducción*

---

El desarrollo del análisis y diseño del sistema informático contribuirá a que se realice la implementación del mismo y de esta manera facilitará el control del tiempo en proyectos productivos. Proporcionará una vía de integración entre los elementos de registro de tiempo, la gestión de tareas y el desarrollo de las actividades diarias.

Ofrecerá información real del cumplimiento de cada tarea por lo que ayudará al desarrollo de cronogramas más ajustados y al desenvolvimiento personal de cada trabajador. Posibilitará adicionar distintas funcionalidades que contribuirán al inicio de un entorno de gestión de tiempo dentro del desarrollo de software.

**Estrategia de investigación:** Exploratoria.

Ayuda a entender el proceso de evaluación y control del tiempo de un proyecto de desarrollo de software, además a conocer las causas que originan el desarrollo del trabajo.

## **Métodos investigativos**

### **Métodos teóricos:**

**Modelación:** Una de las tareas es realizar el análisis de los procesos a automatizar y se puede utilizar este método para explicar lo que pasa de una manera lógica mediante modelos y diagramas que ayudan a comprender el objeto en su totalidad.

**Histórico:** Se utilizará este método para conocer la trayectoria histórica del objeto de estudio y posibilitará conocer los antecedentes de los sistemas automatizados de control del tiempo de un proyecto.

**Análisis:** Este método sirve para comprender como funciona cada uno de los elementos que conforman el sistema. Para realizar esto, se dividirán estos elementos en componentes y así entender cómo funcionan cada uno individualmente.

### **Métodos empíricos:**

**Entrevista:** Este método permite obtener información a partir de experiencias, puntos de enfoque y posibles ideas de personas con experiencia en el objeto de investigación, ayuda a definir mejor los objetivos y el desarrollo del trabajo.

# *Introducción*

---

El contenido del trabajo está estructurado en tres capítulos de la siguiente forma:

**Capítulo 1:** Se realizará la fundamentación teórica donde se hará un estudio del estado del arte, así como un análisis del por qué de las herramientas, metodologías, lenguajes y tecnologías utilizadas para desarrollar el análisis y diseño del sistema propuesto.

**Capítulo 2:** Se establecerán los requisitos funcionales y no funcionales para crear un sistema automatizado que cumpla con las expectativas de los usuarios finales y así comenzar con el modelado del mismo. Se hace, además, una descripción de los casos de uso y se muestra el diagrama de casos de uso del sistema.

**Capítulo 3:** Se realiza el modelo del análisis y de diseño del sistema. Se obtiene, a través de los casos de uso del sistema, el diagrama de análisis. Como resultado del análisis se obtiene el diseño de la aplicación, el cual queda expresado en el diagrama de clases del diseño, los diagramas de secuencia y la descripción de las clases. Se establecerán los patrones de diseño y la arquitectura a desarrollar.

## **CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA**

### **1. Introducción**

Desde el surgimiento de la computación, muchos creadores se han propuesto la meta de informatizar toda actividad relacionada con el desenvolvimiento humano. Al crearse la gestión de proyectos y, dentro de ella la gestión del tiempo del proyecto, emerge la necesidad de utilizar un sistema capaz de llevar un mejor control del tiempo utilizado por los miembros de un proyecto.

Por eso se hace necesario profundizar en conceptos y aspectos relacionados con el Área de conocimiento que ocupa el trabajo, en este caso Gestión del Tiempo del Proyecto. Para realizar un análisis exhaustivo sobre el tema que se ocupa se tiene en cuenta el estudio de algunos de los sistemas de control del tiempo existentes en la actualidad, tanto en el mundo como en Cuba y específicamente en el Centro de Identificación y Seguridad Digital. Así se tiene un mejor conocimiento par poder definir las metodologías de trabajo, técnicas y herramientas que se usarán en el desarrollo del sistema informático.

#### **1.1. Conceptos asociados al dominio del problema**

A continuación se mencionaran algunos conceptos abordados en el trabajo y que están asociados al problema.

**Tarea:** Actividad más pequeña de un proceso de un proyecto.

**Tiempo:** Duración limitada de las cosas.

**Proyecto:** Proceso único consistente en un conjunto de actividades coordinadas y controladas con fechas de inicio y de finalización, llevadas a cabo para lograr un objetivo conforme con requisitos específicos, incluyendo las limitaciones de tiempo, costo y recursos. Cada proyecto tiene objetivos y características propias y únicas.(COSTA, 2006)

##### **1.1.1. Gestión de Proyectos**

El desarrollo de productos, la prestación de servicios (PALACIOS, 2006) entre otras funciones son trabajos que pueden tomar la forma de proyectos. Presentan tres características:

- ✓ Son realizados por trabajadores.

## *Fundamentación Teórica*

---

- ✓ Para realizarlo se necesitan de recursos.
- ✓ Están guiados por un plan de desarrollo.

En 1960, Meter Norden, del laboratorio de investigación de IBM, en su seminario de Ingeniería de Presupuesto y Control presentado ante Asociación Americana de Administración, indicó: *“Es posible relacionar los nuevos proyectos con otros pasados y terminados para estimar sus costes además se producen regularidades en todos los proyectos y es absolutamente necesario descomponer los proyectos en partes de menor dimensión para realizar planificaciones.”*(PALACIOS, 2006)

La Gestión de Proyecto consta de nueve Áreas de Conocimiento que son imprescindibles para el buen manejo de un proyecto. Estas son:

- ✓ *Gestión de la Integración del Proyecto:* En esta área se unen e integran todos los elementos de los distintos grupos que conforman el proyecto.
- ✓ *Gestión del Alcance del Proyecto:* Hace referencia a las actividades exclusivamente necesarias para completar el proyecto satisfactoriamente.
- ✓ *Gestión del Tiempo del Proyecto:* Una de las áreas más importantes dentro de la gestión de proyectos ya que toca el tema de la realización del proyecto en tiempo y así brindarle cuidado, seguridad y seriedad al producto que se desarrolle.
- ✓ *Gestión de los Costes del Proyecto:* Área primordial para controlar los costes del proyecto y que este se desarrolle y finalice dentro del presupuesto aprobado.
- ✓ *Gestión de la Calidad del Proyecto:* Trata el tema del cumplimiento de los objetivos por los que fue creado el proyecto y con la calidad requerida.
- ✓ *Gestión de los Recursos Humanos del Proyecto:* Hace referencia a las actividades y procesos que organizan y guían a los miembros del proyecto.
- ✓ *Gestión de las Comunicaciones del Proyecto:* En esta área se describe los procesos que corresponden a la gestión de la información del proyecto en tiempo y forma.
- ✓ *Gestión de los Riesgos del Proyecto:* Detalla los procesos que corresponden con el manejo de los riesgos de un proyecto.
- ✓ *Gestión de las Adquisiciones del Proyecto:* Aborda el tema de la compra o adquisición de productos, servicios o resultados, también para contratar procesos de dirección.

El Área que se abordará es Gestión del Tiempo del Proyecto para, con la ayuda de ésta, desarrollar el trabajo.

## 1.1.2. Gestión del Tiempo

El tiempo es el elemento principal en toda planificación. La Gestión del Tiempo del Proyecto contiene los pasos necesarios para lograr completar el proyecto en tiempo, la misma contiene seis procesos y cada proceso contiene a su vez diferentes entradas, herramientas y técnicas utilizadas en el proceso y salidas (Figura1). Estos procesos trabajan en conjunto y también con los procesos de las demás Áreas.(OLALDE, 2008) Los procesos incluyen lo siguiente:

- ✓ *Definición de las Actividades:* identifica las actividades específicas del cronograma que deben ser realizadas para producir los diferentes productos entregables del proyecto.
- ✓ *Establecimiento de la Secuencia de las Actividades:* identifica y documenta las dependencias entre las actividades del cronograma.
- ✓ *Estimación de Recursos de las Actividades:* evalúa el tipo y las cantidades de recursos necesarios para realizar cada actividad del cronograma.
- ✓ *Estimación de la Duración de las Actividades:* valora la cantidad de períodos laborables que serán necesarios para completar cada actividad del cronograma.
- ✓ *Desarrollo del Cronograma:* analiza las secuencias de las actividades, la duración de las actividades, los requisitos de recursos y las restricciones del cronograma para crear el cronograma del proyecto.
- ✓ *Control del Cronograma:* verifica los cambios del cronograma del proyecto.

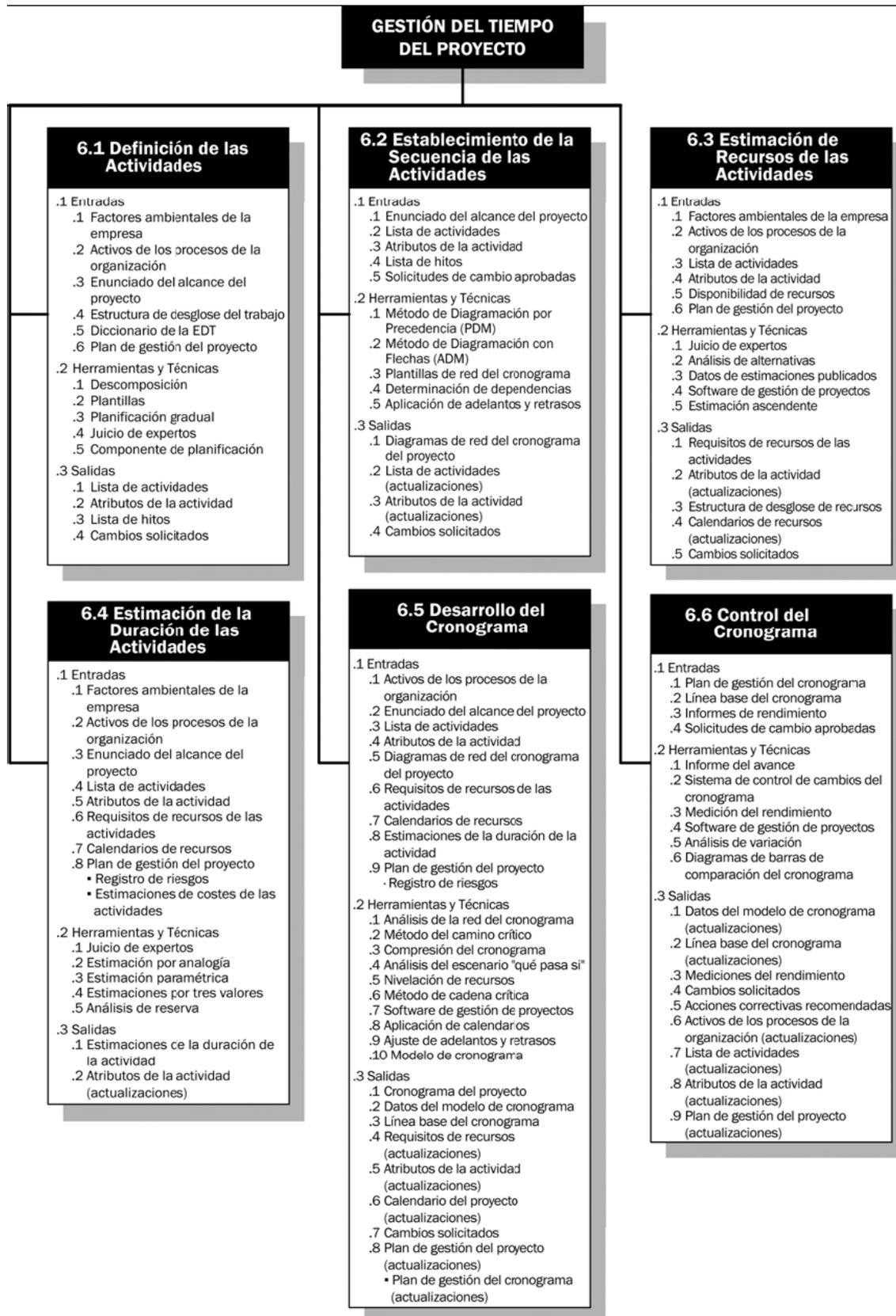


Figura 1.1 Descripción general de la Gestión del Tiempo del Proyecto

Fuente: Guía de los Fundamentos de la Dirección de Proyectos, Tercera Edición, 2004, Project Management Institute, Four Campus Boulevard.

## **1.2. Uso de la gestión de tiempo**

El objetivo fundamental de la Gestión del Tiempo del Proyecto es concluir el proyecto a tiempo, logrando el alcance del proyecto, en tiempo, costes y calidad requerida por el cliente, sin rebasar los riesgos inherentes del proyecto(OLALDE, 2008). Contribuye al buen funcionamiento de la empresa, al tener conocimiento de en qué los empleados invierten su tiempo de trabajo.

## **1.3. PSP (Proceso de Software Personal)**

Fue creado en 1995 por Watts Humphrey en el Instituto de Ingeniería de Software en Pittsburgh, Pennsylvania. El PSP provee métricas, pasos y plantillas que ayudan a los desarrolladores a mejorar sus habilidades de la ingeniería de software. Investigaciones realizadas indican que la productividad, el número de defectos y el tiempo y tamaño de las estimaciones tienden a mejorar cuando se aplica PSP(HUMPHREY, 2001).

PSP se divide en varios niveles donde en cada uno de ellos se tratan diferentes temas referentes al proyecto y a la gestión del mismo. Los niveles son los siguientes:

- ✓ **PSP 0 y 0.1** Proceso Personal de arranque. Los desarrolladores usan sus diseños y métodos de desarrollo actuales y recolectan los datos de su trabajo tales como el tiempo de demora y los defectos encontrados, luego analizan y reportan esos datos.
- ✓ **PSP 1 y 1.1** Planificación. Contiene todos los procesos de PSP 0 y se le incluye el establecimiento de los estándares de codificación, se realiza la estimación del tamaño, de recursos y de cronogramas, luego se siguen esos valores estimados y por último se propone mejorar esos procesos.
- ✓ **PSP 2 y 2.1** Calidad. Adiciona las revisiones personales de diseño y código, la utilización de métricas de calidad de costo y rendimiento y por último perfeccionar los criterios de diseño.
- ✓ **PSP 3** Proceso personal cíclico. En el mismo se realiza el desarrollo en ciclos y la publicación del seguimiento.

El PSP se centra en que el propio desarrollador mejore su desempeño y sus habilidades personales, mida su trabajo y analice los resultados, esto conlleva que la disciplina del trabajador sea de excelencia ya que debe estar consciente que de la calidad de su trabajo depende la del producto

## *Fundamentación Teórica*

---

completo. PSP brinda los datos y técnicas de análisis necesarios para determinar qué tecnologías y métodos trabajan mejor para cada ingeniero.

Pero PSP, como se explica anteriormente, no solo aborda el tema del control del tiempo sino que también incorpora una serie de métodos y formatos que son muy útiles para el análisis a fondo del producto que se desarrollará y aporta una serie de pasos que son necesarios para dar cumplimiento a los requisitos del cliente. Dichos formatos varían en dependencia del nivel en que se esté trabajando de manera que le sea más fácil y de más utilidad al trabajador, sin tener que manejar la misma información en diferentes formatos, de esta manera adapta el nivel y el formato que vaya a utilizar de acuerdo a sus necesidades.

Constituye una herramienta poderosa que puede ser empleada de varias formas, por ejemplo, ayuda a administrar el trabajo personal, evaluar los talentos y formar habilidades. Contribuye a la realización de mejores planes, para monitorear el desempeño de forma precisa y medir la calidad de los productos. Independientemente de que se diseñen programas, se desarrolle requerimientos, se escriba la documentación o se de mantenimiento al software existente, el PSP ayuda a hacer mejor el trabajo.

De acuerdo a lo que se centra este trabajo, el nivel de PSP que más aporta a la solución propuesta es el PSP 0 que mide el tiempo empleado en cada fase. PSP proporciona un cuaderno llamado Registro de Control de Tiempo donde el trabajador anota la fecha de comienzo y fin de la tarea a realizar, el tiempo de alguna interrupción que tuvo en la misma, el nombre de la actividad, entre otros datos; de igual manera sucede con el Registro de Control de Defectos en el cual se anota la fecha en que se encontró el defecto, el tipo de defecto que es, tiempo que se demoró corrigiéndolo y otros datos más de interés.

El control del tiempo es un elemento fundamental en el desarrollo de software ya que es el responsable de la rapidez, calidad y costes del producto final. Cuando se controla eficientemente el tiempo de un proyecto la calidad aumenta y los costes disminuyen considerablemente. Individualmente, PSP define varios pasos para llevar este control; el primero es definir las tareas y conocer cuánto tiempo gastas en cada una de ellas midiendo el mismo.

PSP es la mejor arma para controlar, administrar y mejorar la forma en que trabajan los miembros del equipo. De esta manera cada trabajador individualmente se preocupa por su desempeño y planifica su trabajo de acuerdo al cronograma de actividades concebido además debe medir, monitorear y analizar su trabajo para mejorar su desenvolvimiento en el desarrollo.

Actualmente en los proyectos productivos del Centro de Identificación y Seguridad Digital el control del tiempo que cada trabajador utiliza para desarrollar su tarea es prácticamente nulo; los jefes de proyectos se guían por preguntas directas a los miembros del equipo de desarrollo y sin mediación de algún sistema informático, esto conlleva a una planificación inexacta de la duración de las tareas, además no se conoce a ciencia exacta que trabajador es más indicado para realizar una actividad o que tiempo se demora realmente haciéndola, por eso, la realización de un sistema informático capaz de llevar el registro de todo ese tiempo empleado contribuirá a la estimación de tareas, tiempos y recursos.

La estimación, del tiempo y los recursos del producto, debe ser ajustada por los miembros o individuos que intervienen en el desarrollo del proyecto. PSP se basa en el tamaño y en los datos de la productividad de cada trabajador y con estos datos estima el tiempo requerido para hacer el trabajo. En conjunto con la Gestión del Tiempo del Proyecto de la Gestión de Proyectos, dicha estimación de tiempos y de recursos y la planificación de las tareas será más exacta ya que, con los formatos que brinda PSP para controlar el tiempo y los defectos, entre otras informaciones, y los procesos contenidos en el dicha área, posibilitará un mejor control del tiempo asignado para el desarrollo de tareas.

### **1.4. Gestión de Tareas**

Un producto se realiza mediante tareas o actividades. Las mismas se planifican, ejecutan y finalmente se controlan. La gestión de las tareas de un proyecto es uno de los procesos de mayor envergadura dentro de toda la gestión de proyectos. Sin duda alguna la gestión de tareas bien definidas, estimadas y priorizadas conllevan a adquirir un software con calidad.

En el mundo existen una gran variedad de sistemas para realizar la gestión de tareas de proyectos de software. En específico, se estudiará el Jira, que es una aplicación para la administración de proyectos y actividades desarrollada para facilitar el trabajo del equipo. Es una tecnología basada en el estándar J2EE.(BARRA, 2008) Tiene como características que puede contener uno o varios proyectos, es simple de usar, flexible y amigable al usuario, permite administrar tanto las tareas como el tiempo, defectos, procesos, requerimientos, ideas y otras muchas más actividades, puede enviar notificaciones mediante e-mail, es compatible con casi todas las bases de datos existentes y es multiplataforma, solo necesita para correr de una máquina virtual Java. También provee un mecanismo de permisos completo y cada usuario puede configurar su panel de control de acuerdo a las necesidades que tenga. El registro de tiempos es muy útil para llevar un estudio de los tiempos por trabajador y además permite ver para determinados procesos si se está adelantado o retrasado con respecto lo estimado.

Entre las grandes potencialidades que permite están:

- ✓ Construye y guarda reportes altamente configurables los cuales se pueden compartir con otros usuarios.
- ✓ Permite adjuntar documentos en formatos como: Acrobat, Word, Excel, entre otros.
- ✓ Brinda la posibilidad de asociar procesos o actividades que estén relacionados facilitando así la reutilización de conocimientos.
- ✓ Permite estimar y llevar un registro de los tiempos de trabajo asociados a una actividad o proceso.
- ✓ Provee un mecanismo de búsqueda de actividades o procesos y de toda la información adjunta a los mismos.

Por lo que se concluye que con la ayuda de este tipo de sistemas, la gestión de tareas es un componente viable y seguro para, integrado con otros elementos, obtener un producto robusto y con calidad.

#### **1.4.1. Integración de la gestión de tareas con otras áreas**

Si ya están las tareas planificadas también se necesita conocer cuál es el tiempo que requiere la misma para desarrollarse. La gestión del tiempo se encuentra muy ligada a la gestión de tareas ya que de una depende la otra. Sin una buena planificación del tiempo de las tareas, puede ocurrir que deje de tener valor la tarea o haya que volver a realizar la misma y esto infiere un atraso en el calendario del proyecto. Controlando el tiempo que se dedica a una tarea se puede entonces estimar, con mayor exactitud, el tiempo necesario para cada tarea y así, la gestión de las mismas se realizaría con mayor éxito.

#### **1.5. Los trabajadores en el Proceso de desarrollo de software.**

RUP define elementos dentro del proceso de desarrollo de software tales como los trabajadores, las actividades y los artefactos y cuando estos se combinan adecuadamente ayudan a una entidad a desarrollar un producto de software de manera eficiente.

Los trabajadores son la fuerza impulsora de todo proyecto y sin los mismos ningún proceso o herramienta podrían terminar por si solos un proyecto de software. Los resultados finales dependen en gran medida de la calidad de los miembros del equipo de desarrollo.

RUP ofrece conocimientos basados en la información que ayudan a los trabajadores a hacer un mejor trabajo, pero la calidad del producto depende más de la experiencia y capacidad de sus trabajadores.

Ellos van creando sus aptitudes y habilidades de los problemas que van surgiendo y que tienen que enfrentar.

### **1.6. Sistemas para la gestión o control de tiempo**

En la actualidad existen en el mundo numerosas aplicaciones y artículos que abordan este tema. La creación de sistemas de control de tiempo ha sido objetivo de numerosos proyectos de software, cada uno con características individuales, que brindan innumerables funcionalidades. Pero con esta investigación se pretende profundizar más en el tema, porque a partir de esta información brindada, se realizará un profundo análisis con el objetivo de planificar el tiempo destinado a cada actividad de acuerdo a valores reales que cumplan con el cronograma de actividades.

En una investigación realizada a las aplicaciones existente en la Universidad se estudiaron dos sistemas que ponen en práctica la temática estudiada.

#### **1.6.1. Sistema para el control de tiempo de producción en los proyectos productivos de la universidad de las Ciencias Informáticas.**

Este sistema manipula toda la información referente a los procesos de Windows, el nombre del proceso, la ventana del proceso en caso de tenerla, el tiempo de inicio, tiempo de fin, el tiempo de activación de la ventana, el usuario que creó el proceso, entre otras características. El sistema controla el horario asignado a las tareas productivas de la UCI, cuenta con una interfaz fácil y amigable. Brinda información importante para los directivos, pues estos conocen en todo momento las actividades que se están realizando, el tiempo utilizado para una actividad, la fecha y hora en que comenzó, la fecha y hora en que finalizó, así como otro tipo de información de importancia para el control de tales actividades. Para desarrollarlo se empleó Microsoft Visual Studio .NET 2003 y como lenguaje de programación Microsoft Visual C# .NET para la aplicación cliente. Para la aplicación servidor se utilizó como lenguaje de programación C++. Se recurrió al Crystal Reports como herramienta para el diseño y visualización de reportes en Microsoft .NET. Se utilizó SQL Server 2000 para la aplicación cliente y Microsoft ACCES 2000 para la aplicación servidor. Se empleó el Proceso Unificado de Desarrollo de Software (RUP) como metodología de desarrollo de software.

Con las características antes mencionadas se decidió no utilizar dicho sistema pues no se integra con un servidor o gestor de tareas para dar cumplimiento a las actividades del cronograma. Además debido al importante proceso en el que está inmerso el país como es la migración a software libre, se desea en dicha investigación utilizar herramientas y metodologías libres.

### **1.6.2. Sistema de gestión de tiempo de los estudiantes vinculados al proyecto PROCYON.**

Esta aplicación se centra entre otros aspectos en el uso de internet de los estudiantes que laboran en dicho proyecto. Ofrece además a los usuarios autorizados la información detallada sobre los proyectos en que trabaja el estudiante y la tarea que desempeña en el mismo. Permite a través de la integración del sistema con otros recursos de la red de PROCYON conocer la actividad realizada por el estudiante en Internet y en el repositorio. Además realiza evaluaciones periódicas a los estudiantes según su comportamiento y resultados en el trabajo, quedando registradas en la base de datos del sistema.

Se llega a la conclusión que el uso de internet es inevitable ya que durante el desarrollo de un producto muchas investigaciones se apoyan en sitios de internet y el objetivo del presente trabajo no es medir el tiempo de uso de la misma, por este motivo se determinó no utilizarlo.

### **1.7. Herramientas de gestión de tareas que presentan elementos para la gestión de tiempo.**

Los sistemas para gestionar tareas juegan un papel fundamental en el desarrollo de proyectos en todo el mundo, pues posibilitan optimizar este proceso que resulta, en estos tiempos, muy engorroso de realizar. En el ámbito internacional existen diversas aplicaciones que revelan numerosas utilidades y que permiten aprovechar las ventajas de las tecnologías de la informática y las comunicaciones para la gestión del tiempo. A continuación se describirán brevemente dos de los sistemas estudiados.

#### **1.7.1. Time Tracker 4.4**

Esta es una aplicación gratuita y de código abierto para Windows que permite gestionar el tiempo que se dedica a una tarea. Es una sencilla utilidad la cual monitorea la actividad que va generando cada una de las ventanas abiertas en el escritorio (ya sean de carpetas abiertas o de programas ejecutándose) informando con todo detalle del tiempo exacto que llevan ejecutándose cada una de ellas. Aparte, ofrece datos sobre los recursos del sistema que están gastando cada una de dichas ventanas así como el rendimiento y productividad de cada una de ellas.

Lo que hace principalmente **TimeTracker** es llevar la cuenta del tiempo que se usa en cada ventana abierta del sistema operativo, mostrar esa información, y almacenarla para su posterior análisis. Ese análisis es, en realidad, lo importante, ya que de ahí se puede sacar conclusiones relacionándolo con la productividad, diversión, o tiempo gastado en el uso de la computadora. Entre las principales

desventajas del Time Tracker se debe mencionar que el usuario debe introducir en el mismo las tareas a realizar, no utiliza el Registro de Windows y se puede ejecutar a partir de cualquier disco o flash. Además no se integra con un servidor central para realizar las tareas de acuerdo al cronograma.

### **1.7.2. Process Dashboard**

El Process Dashboard fue desarrollado originalmente en 1998 por la fuerza aérea de Estados Unidos, y ha continuado desarrollándose bajo modelo de código abierto. Está libremente disponible para la descarga bajo condiciones de la GNU Public License (GPL: Licencia Pública de GNU). (ZULUETA, 2007)

El Process Dashboard soporta las siguientes funcionalidades:

- ✓ Recolección de datos: tiempo, defectos, tamaño; planificación contra datos actuales.
- ✓ Planeamiento: tiene los scripts de los distintos niveles, plantillas, formularios, y resúmenes, de los diversos documentos que hay que realizar; también está incorporado un modulo para utilizar PROBE.
- ✓ Seguimiento: soporte potente para los valores obtenidos.
- ✓ Análisis de Datos: genera de forma automática gráficos y reportes de ayuda para el análisis de la información histórica.
- ✓ Exportación de Datos: Permite la exportación de datos a Excel, o en formato de texto para ser utilizados con otras herramientas.

Fue desarrollado en Java, y necesita para correr en cualquier sistema de una Máquina Virtual Java. Para acceder a los formularios y guiones se requiere disponer de algún navegador web.

Como ventajas de este producto se pueden resaltar:

- ✓ **Precio:** el producto es código abierto, y se distribuye sin costo alguno. **Independiente de la plataforma:** al estar desarrollado completamente en Java, corre perfectamente en cualquier plataforma que soporte Java.
- ✓ **Flexible y extensible:** el usuario puede definir nuevos procesos e incorporarlos sin la necesidad de un programa, simplemente deberá especificarlos. Así mismo se pueden agregar o modificar los scripts, plantillas, formularios, y resúmenes, utilizando cualquier editor de HTML.

- ✓ **Disponibilidad:** los Jefes de Proyectos puedan revisar los registros de tiempo así como la ejecución de las tareas a través de la Web, sin necesidad de acudir a los puestos de trabajo de cada miembro del equipo. (SOURCEFORGE, 2006)

Presenta como inconveniente que el tiempo que requiere para procesar algunos formularios es bastante largo si se ejecuta en una PC no muy moderna. Otra de las desventajas que tiene es que los usuarios ingresan los datos manualmente y esto conlleva a una planificación irreal del tiempo invertido en cada tarea.

Por todo lo anterior expuesto se necesita realizar el análisis y diseño de un sistema que obtenga los aportes y las ventajas de estos sistemas estudiados conjuntamente con las necesidades requeridas por el Centro de Identificación y Seguridad Digital.

### 1.8. Metodología seleccionada para la solución del problema.

El **Proceso Unificado de Software**, (**RUP** por sus siglas en inglés) es un proceso de desarrollo de Software. Es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Permite especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos. Es la metodología más utilizada a nivel mundial, por las grandes potencialidades que presenta para el análisis, implementación y documentación de sistemas orientados a objetos. Está basado en componentes y utiliza el Lenguaje Unificado de Modelado (UML) para preparar los esquemas de un sistema de software y describir todo el proceso. Desde sus inicios cuenta con una documentación profunda y detallada de todo el proceso en sentido general.

Las principales características en su ciclo de vida son:

- ✓ **Guiado por lo casos de uso:** Teniendo en cuenta que la razón de ser de un sistema es brindar servicios a los usuarios, RUP define caso de uso como la secuencia de acciones realizada por un sistema que produce un resultado observable de valor para un actor particular. (ETEROVIC, 2001)
- ✓ **Centrado en la arquitectura:** La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados, entre otros.(JACABOSON, 2000) En otras palabras, la arquitectura representa la forma del sistema, la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.

## *Fundamentación Teórica*

---

- ✓ **Iterativo e incremental:** La alta complejidad de los sistemas actuales hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos. Cada mini-proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.(VALENCIA, 2007) En cada iteración los desarrolladores seleccionan un grupo de casos de uso, los cuales se diseñan, implementan y prueban. La planificación de iteraciones hace que se reduzcan los riesgos de los costos de un solo incremento, no sacar al mercado un producto en el tiempo previsto, mantener la motivación del equipo pues puede ver avances claros a corto plazo y que el desarrollo pueda adaptarse a los cambios en los requisitos.

Está compuesto por tres elementos fundamentales:

- ✓ **Actividades:** definen el “cómo”, son los procesos que se llegan a determinar en cada iteración.
- ✓ **Trabajadores:** indican el “quién”, son las personas o entes involucrados en cada proceso.
- ✓ **Artefactos:** apuntan al “qué”, un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Se divide en cuatro fases, las cuales son:

- ✓ **Inicio:** Tiene como objetivo determinar la visión del proyecto.
- ✓ **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- ✓ **Construcción:** El objetivo es obtener la capacidad operacional inicial.
- ✓ **Transición:** Debe llegar a obtenerse un entregable del producto.

Cada fase concluye con un hito bien definido donde deben tomarse ciertas decisiones.

Dentro de cada iteración de cada fase se llevan a cabo nueve flujos de trabajo principales dentro de los cuales los seis primeros son de ingeniería y los demás son de apoyo.

Flujos de Ingeniería:

- ✓ **Modelado del negocio:** Este flujo identifica los procesos de negocio, los que estarán sujetos a automatización y quiénes intervienen en los mismos.
- ✓ **Requerimientos:** Se identifican las restricciones que se imponen y lo que el sistema debe hacer.
- ✓ **Análisis y Diseño:** Describe cómo el programa será realizado y define cómo será programado.

# Fundamentación Teórica

- ✓ **Implementación:** Define cómo estarán los nodos ubicados y la ubicación de los objetos y clases en paquetes.
- ✓ **Prueba:** Se localizan los defectos del software.
- ✓ **Instalación:** Se entrega una versión operacional.

Flujos de apoyo:

- ✓ **Administración de proyecto:** Encargado de organizar el trabajo y de que se termine el proyecto en el tiempo previsto.
- ✓ **Administración de configuración y cambio:** Describe el uso y actualización concurrente de los elementos, control de versiones entre otras actividades.
- ✓ **Ambiente:** Describe los procesos y herramientas que soportarán al equipo de trabajo del proyecto.



**Figura 1.2 Fases e iteraciones de la Metodología RUP**  
Fuente: <https://www.nexteleng.es/Clientes/catalogos/imagenes/RUP4.gif>

Una peculiaridad de esta metodología es que, en cada ciclo de iteración, se insiste en el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. Esta metodología es muy utilizada en el proceso de desarrollo de software por ser flexible; además, al ser iterativa, permite que se vaya construyendo el software por ciclos, por lo cual se pueden detectar errores con tiempo de antelación. Es una metodología confiable pues desde su surgimiento ha tenido una gran aceptación.

De acuerdo a lo anteriormente descrito de la Metodología RUP, ésta ha sido la seleccionada para guiar el desarrollo del software en cuestión; además el Centro de Identificación y Seguridad Digital se propuso utilizar esta metodología para el desarrollo de todos los proyectos.

## 1.9. Lenguaje de Modelado

**Lenguaje Unificado de Modelado (UML**, por sus siglas en inglés, Unified Modeling Language) El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, hasta la implementación y configuración con los diagramas de despliegue.(ORALLO, 2008) El objetivo final es una aplicación de software robusta, flexible y escalable. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el RUP), pero no especifica en sí mismo qué metodología o proceso usar. Permite además la creación de los diferentes modelos que ofrecen las vistas necesarias para la construcción de un software de calidad y permite la comprensión del sistema que se quiere realizar tanto por parte de los usuarios finales, como de los desarrolladores que implementarán la solución. Es un estándar de la industria, pero no sólo de la industria del software sino, en general, de cualquier industria que requiera la construcción de modelos como condición previa para el diseño y posterior construcción de prototipos.

Dentro de sus *características* se pueden mencionar algunas tales como:

- ✓ Tecnología orientada a objetos.
- ✓ Viabilidad en la corrección de errores.
- ✓ Desarrollo iterativo e incremental.
- ✓ Modela estructuras complejas.
- ✓ Participación del cliente en todas las etapas del proyecto.
- ✓ Permite especificar las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- ✓ Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).

- ✓ Modela el comportamiento del sistema mediante casos de uso, diagramas de secuencia y de colaboración, que sirven para evaluar el estado de este.
- ✓ Permite documentar disímiles artefactos de un proceso de desarrollo.
- ✓ No es difícil de aprender ni de utilizar.

## **1.10. Herramienta CASE a utilizar**

Ingeniería de Software Asistida por Computadora (Computer Aided Software Engineering, CASE) es un tipo de ingeniería de software en la que se intenta aumentar la eficacia de sus procesos, al soportar la realización de las tareas con el uso de tecnologías.(PÉREZ, 1999)

Las herramientas CASE son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas permiten a los desarrolladores modelar y documentar sus artefactos, cubriendo el ciclo de vida del proceso de desarrollo de software.

### **1.10.1. Visual Paradigm**

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue(PARADIGM, 2007). Es una herramienta de modelado diseñada para un gran número de usuarios, incluyendo ingenieros de sistemas, analistas de sistemas, analistas de negocio y arquitectos de sistemas. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación(PARADIGM, 2007). Además se integra con IDEs (Entornos Integrados de Desarrollo) como Eclipse, JBuilder, NetBeans, IntelliJ IDEA, JDeveloper and WebLogic Workshop Esto ayuda a soportar la fase de implementación de desarrollo de software. A diferencia de muchas herramientas de modelado, pueden extenderse sus diagramas hechos desde el Visio y de Rational Rose. Es desarrollada como software libre, es compatible con las metodologías existentes para el desarrollo de software y muy sencilla de usar, fácil de instalar y actualizar. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios realizados por sus compañeros. Otros de los beneficios del Visual Paradigm son:

- ✓ Soporta UML versión 2.1.

- ✓ Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI.
- ✓ Presenta un Editor de Detalles de Casos de Uso.
- ✓ Soporta la Generación de objetos Java desde la base de datos.
- ✓ Importación y exportación de ficheros XMI.
- ✓ Navegación intuitiva entre el modelo visual y el código.
- ✓ Poderosa herramienta de generación de PDF/HTML a partir de diagramas UML.
- ✓ Sincronización entre el código fuente y el modelo en tiempo real.

## **1.11. Plataforma de desarrollo**

La plataforma de desarrollo es algo imprescindible en la producción de software. Es donde se definen el conjunto de herramientas y tecnologías (marcos de trabajo), versiones a usar y su integración, que intervienen en un proceso de desarrollo de software.

Dado que el presente trabajo está destinado al desarrollo del análisis y diseño de un sistema, se propone para la implementación del mismo la plataforma J2EE por sus grandes ventajas.

### **1.11.1. Java**

El lenguaje de programación Java es actualmente uno de los más utilizados en la creación de software de empresa en el mundo. Originalmente fue concebido para construir programas pequeños que se encuentren dentro de dispositivos electrónicos. Luego se empezó a utilizar en navegadores de Internet corriendo pequeños programas llamados Applets y desde entonces ha tenido un gran crecimiento y está siendo usado para construir grandes sistemas empresariales. (JAMES GOSLING, 2005)

La plataforma Java ha atraído alrededor de 4 millones de desarrolladores de software, se utiliza en los principales sectores de la industria de todo el planeta y está presente en un gran número de dispositivos, ordenadores y redes de cualquier tecnología de programación. De hecho, su versatilidad, eficiencia, portabilidad y la seguridad que aporta, la han convertido en la tecnología ideal para su aplicación a redes, de manera que hoy en día, más de 2.500 millones de dispositivos la emplean.

### **1.11.2. Java 2 Platform, Enterprise Edition (J2EE)**

La plataforma J2EE es la edición empresarial de la plataforma Java. Soporta un único lenguaje que es Java, utilizado para el desarrollo de todos los componentes y compilado por un código intermedio denominado ByteCodes que se ejecuta en un entorno de ejecución llamado JRE (Entorno de Desarrollo de Java por sus siglas en inglés) para transformar el lenguaje intermedio a código propio de la máquina en la que se corre la aplicación. Brinda soporte para múltiples sistemas operativos, debido a la portabilidad, o posibilidad de ejecutar las aplicaciones desarrolladas en cualquier sistema operativo y/o máquina del mercado. Posee una sintaxis sencilla aunque es un poco complicado de aprender y goza de una amplia gama de librerías y frameworks la mayoría gratuitas y código abierto. (PEROVICH, 2001)

Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en las tareas de mantenimiento de bajo nivel. (BODOFF, 2004)

Ejemplos de algunas herramientas de desarrollo Java de código abierto son:

- ✓ NetBeans IDE (Entorno Integrado de Desarrollo), un IDE basado en Java.
- ✓ La plataforma Eclipse, un IDE basado en Java.
- ✓ Expand, un plugin de Eclipse, para desarrollo rápido.
- ✓ JUnit, un framework para Pruebas de unidad automatizadas.
- ✓ Apache Software Foundation Apache Tomcat, un contenedor web de Servlet/JSP.
- ✓ OpenXava, un framework de código abierto para desarrollo fácil de aplicaciones de negocio J2EE.
- ✓ JDeveloper, un IDE basado en Java y desarrollado por Oracle.
- ✓ JBuilder, desarrollado por Borland.
- ✓ JavaServer Faces, un framework para desarrollar aplicaciones web EE conforme al modelo MVC, desarrollado por Sun Microsystem.

# *Conclusiones Parciales*

---

## **Conclusiones Parciales**

Una vez analizados los sistemas existentes, tanto en la universidad como fuera de ella y, determinadas las deficiencias de los mismos; para el propósito del trabajo se decidió desarrollar, en software libre, una aplicación de escritorio que sustituya los sistemas existentes y que sea capaz de registrar el tiempo de realización de cada tarea, además se integre con un sistema que controle las actividades del cronograma del proyecto.

Tras la imperiosa necesidad de migrar hacia el desarrollo del software libre, se determinó la utilización de los mismos, por lo que se propone usar como metodología RUP para el proceso de desarrollo de software, recurriendo al Visual Paradigm como herramienta de modelado, con la cual se podrá realizar el análisis y diseño de la aplicación, utilizando como lenguaje de modelado UML.

## **CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA**

### **2. Introducción**

Los procesos que se llevan a cabo actualmente en el Centro de Identificación y Seguridad Digital, con respecto al control de las tareas productivas, presentan problemas ya que, muchas de las tareas que se asignan se atrasan y, por este motivo se van a analizar los mismos con el fin de realizar la descripción de la propuesta de solución. Se define, además, el modelo de dominio y los conceptos más importantes relacionados con el tema para una mejor comprensión de la aplicación. Después de haber expuesto los procesos actuales del negocio y plantear el modelo de dominio propuesto se dispone a, capturar los requisitos funcionales y no funcionales y luego, modelar el sistema que se va a desarrollar.

#### **2.1. Situación Problemática**

Actualmente en el Centro de Identificación y Seguridad Digital no se utiliza ningún sistema de control del tiempo de producción de sus proyectos productivos. En muchas ocasiones, por diferentes motivos las entregas no se realizan en el tiempo planificado. Para estos atrasos o incumplimientos de entregas, el jefe de proyecto no sabe con precisión cuales fueron los motivos reales que provocaron tal situación, el verdadero uso del tiempo de máquina asignado a los estudiantes y la utilización de las herramientas para desarrollar dichas tareas. Los jefes de proyecto tienen, además de todas sus ocupaciones, la ardua tarea de inspeccionar y controlar la realización de las tareas y la duración del desarrollo de las mismas. Por tal motivo se desea automatizar el control de las actividades reales que realizan los estudiantes en los tiempos de máquina asignados.

#### **2.2. Flujo actual de los procesos involucrados en el campo de acción.**

En los proyectos de desarrollo de software del Centro de Identificación y Seguridad Digital actualmente el flujo de los procesos comienza cuando un usuario accede al gestor de tareas Jira, luego ubica el proyecto al que pertenece y busca sus tareas asignadas. La tarea contiene el nombre de la misma, el estado en que se encuentra y el proyecto al que pertenece, el nombre de la persona que asigna la tarea y a la persona que le es asignada la tarea, la prioridad de la tarea, el tiempo estimado de duración de la tarea y el tiempo que queda para realizarla y una pequeña descripción de la misma, entre otros aspectos. El usuario tras identificar la tarea a realizar, da comienzo a la misma y cuando ya haya terminado, cierra dicha tarea poniendo una breve descripción y finalmente la guarda en el

repositorio. Los jefes de proyecto se conectan al gestor de tarea para revisar el desarrollo de cada tarea asignada, agregar más tareas, modificar tareas y eliminar tareas entre otras posibilidades.

### **2.3. Debilidades de los procesos actuales.**

El principal problema consiste en que en el gestor de tareas Jira no se registran todas las actividades que se llevan a cabo cuando se hace una tarea, ni cuánto tiempo se demora un usuario en cada herramienta o software. Esta información es de suma importancia para hacer la planificación real del tiempo de duración de la tarea y además, saber quién está más capacitado para realizarla, ya que no todas las tareas tienen la misma complejidad y es necesario saber quién es la persona más indicada para desarrollar la misma.

Además, al no utilizar elementos de PSP en el desarrollo del producto, la calidad del mismo pudiera ser deficiente ya que no se hacen planes ajustados al no conocer el tiempo ni los recursos necesarios para realizar las tareas y no se eliminan los defectos con antelación, entre otros elementos. Esto se debe a que no se aprovechan las posibilidades que brinda PSP al proporcionar métodos detallados de planificación y estimación que muestran a los trabajadores como controlar su rendimiento. El uso de estos métodos garantiza un buen uso y aprovechamiento de los miembros del equipo de desarrollo y los recursos asignados, al mismo tiempo reduce la duración del ciclo de vida mediante la detección de los defectos en la fase inicial del proyecto y por este motivo también se reducen los costos de desarrollo.

### **2.4. Sindicación Simple Real (RSS)**

El RSS es un formato basado en XML (Lenguaje de Etiquetado Extensible) que permite encontrar aquella información que mejor se adapta a lo que el usuario desea, pero también ofrecerla de forma rápida y actualizada.

Sus siglas han cambiado múltiples veces su significado debido a nuevas versiones.

- ✓ Rich Site Summary (RSS 0.91)
  
- ✓ RDF Site Summary (RSS 0.9 y 1.0)
  
- ✓ Really Simple Syndication (RSS 2.0)

Los archivos RSS son un nuevo método para obtener y ofrecer información gracias a que contienen metadatos sobre las fuentes de información. Este formato es de gran utilidad para sitios Web que

## *Características del Sistema*

---

actualicen sus contenidos con frecuencia, ya que permite compartir la información y verla en otros sitios de forma inmediata. A este intercambio de información se le denomina "sindicación".

Sin embargo, para poder compartir esta información se necesitan unos software determinados, llamados "agregadores" o "lectores RSS", es decir, programas capaces de leer e interpretar las fuentes RSS.

El formato RSS sirve para facilitar el acceso a la información Web que se actualiza con regularidad, logrando que el usuario recupere al momento las novedades producidas en la información de su interés.

### **2.5. Solución Propuesta**

Del análisis del estado del arte y las principales funcionalidades identificadas en los programas de este tipo, se elaboró una propuesta de solución.

Para darle solución a los problemas mostrados se propone una aplicación de escritorio desarrollada sobre la plataforma J2EE.

Permitirá conectarse mediante un RSS al gestor de tareas Jira que contiene las tareas asignadas a un usuario determinado, dando la posibilidad de escoger la tarea a realizar. En el Jira se planifica la duración de cada tarea a partir de un cálculo razonable del tiempo en que se pueda demorar el desarrollo de la misma.

El sistema llevará un control exacto del tiempo que utiliza un trabajador en cada aplicación o software y además registrará el proceso del sistema operativo al que pertenece dicha aplicación.

Al finalizar cada sesión de trabajo o cerrar una tarea, posibilitará generar reportes que serán guardados en un servidor. Asimismo, el sistema brindará información leal del comportamiento y uso del tiempo de máquina que necesita cada trabajador.

Con el apoyo de elementos de PSP el registro será mucho más fácil y sencillo de hacer, ya que gracias al cuaderno de Registro de Control de Tiempo que brinda el PSP, muestra los datos de valor a recopilar.

Esto aporta a que la estimación de las tareas sea un componente mucho más seguro tras saber, a ciencia cierta, todo lo que necesita la tarea para ser desarrollada, tanto el tiempo y los recursos como el personal capacitado.

### **2.6. Modelo de Dominio**

El Modelo de Dominio (o Modelo Conceptual) es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Define un modelo de clases común para todos los implicados en el desarrollo, representadas en objetos del dominio, sirve como

# *Características del Sistema*

---

interlocutor entre clientes y desarrolladores. El propósito fundamental de este modelo es generar una terminología común y sentar las bases del entendimiento del desarrollo y no para definir el sistema completo. Cualquiera sea la solución de casos de uso que se haya elegido, los conceptos e ideas propias del dominio del problema son las mismas; un mismo modelo de dominio contempla cualquiera de las soluciones analizadas. El modelo de dominio es global, es decir se realiza para todos los casos de uso y no para uno en particular.

Características que presenta:

- ✓ No hay cronología
- ✓ No se diferencia entre dentro y fuera del sistema
- ✓ Es global, no por caso de uso
- ✓ No es completo: es más bien esquemático, las asociaciones están resumidas.

Utilizando UML, un modelo de dominio se representa con un conjunto de diagramas de clases en los que no se define ninguna operación.

Los modelos de dominio pueden mostrar:

- ✓ Objetos del dominio o clases conceptuales.
- ✓ Asociaciones entre las clases conceptuales.
- ✓ Atributos de las clases conceptuales.

## **2.6.1. Realización del Modelo de Dominio**

Teniendo en cuenta que la definición de procesos y roles del negocio se hace difícil encontrarlos, se ve a simple vista la necesidad de describir el funcionamiento de la aplicación mediante una serie de conceptos, entidades y sus relaciones, agrupándose en un modelo de dominio con el fin del fácil entendimiento de la aplicación.

El objetivo del modelado del dominio es contribuir a la comprensión del contexto del sistema, y por lo tanto también contribuir a la comprensión de los requisitos del sistema que se desprenden de este contexto.

## Características del Sistema

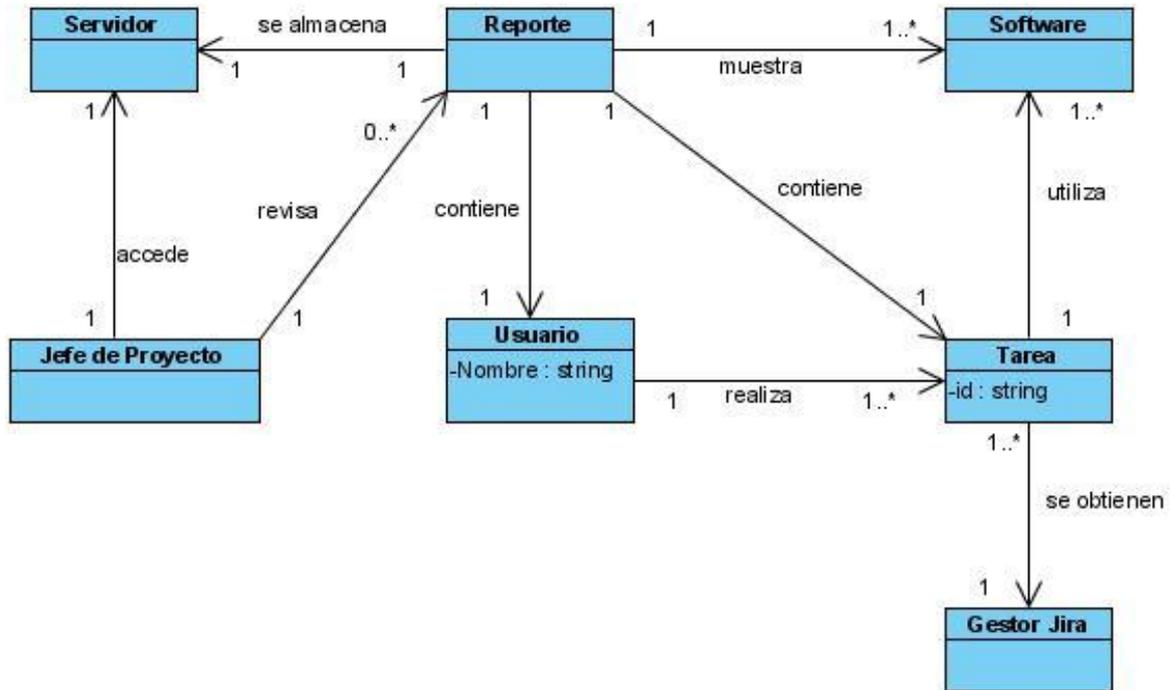


Figura 2.1 Modelo de Dominio  
Fuente: Elaboración propia.

### 2.6.2. Conceptos asociados al dominio

Conceptos	Descripción
Usuario	Representa a la persona que interactúa con la PC para desarrollar las tareas.
Jefes de Proyecto	Persona clave en la planificación, ejecución y control del proyecto y es el motor que ha de impulsar el avance del mismo. Tiene autoridad para revisar el trabajo de los miembros del proyecto.
Software	Aplicaciones usadas por el usuario para desarrollar una tarea. Éstas serán registradas por el sistema para que el Jefe de Proyecto controle el uso de las mismas.
Gestor de tareas	Herramienta que ayuda a gestionar fácilmente y de forma visual, las actividades y el trabajo de los miembros del proyecto.
Tarea	Actividad que el usuario debe hacer dentro de un tiempo determinado para llegar a conseguir el objetivo del proyecto y son controladas por los directivos del proyecto. Las mismas son asignadas en el Gestor de Tareas y se muestran en el Sistema para que el usuario seleccione en la que desea trabajar.
Reporte	Contiene los datos de las herramientas utilizadas para desarrollar la tarea, además incluye los datos del usuario que realizó la tarea.

# Características del Sistema

---

Servidor	Contiene todos los reportes de tiempo generados por los miembros del equipo de desarrollo.
----------	--

**Tabla 2.1 Conceptos del dominio**

## 2.7. Especificación de los Requisitos del Software

El propósito general de la captura de requisitos es obtener una descripción correcta de lo que debe de hacer el sistema y delimitar su alcance, es decir, que debe y que no debe hacer. Los requisitos juegan un papel importante durante el ciclo de vida de un programa. Durante la fase de inicio se identifican la mayoría de los casos de uso para delimitar el sistema y detallar los más importantes

Para ello, se enumeran a través de requerimientos funcionales y no funcionales, las acciones que el sistema deberá ser capaz de realizar.

### 2.7.1. Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Los mismos no alteran la funcionalidad del producto, esto quiere decir que los requerimientos funcionales se mantienen invariables sin importarle con que propiedades o cualidades se relacionen.

Los requisitos funcionales desglosados son los siguientes:

**RF 1-** Permitir obtener tareas abiertas o asignadas.

**RF1.1-** Establecer la conexión con el gestor de tareas Jira.

**RF 1.1.1-** Autenticar usuario.

**RF 1.1.2-** Mostar las tareas abiertas por el usuario.

**RF 1.1.3-** Seleccionar tarea a ejecutar.

**RF 1.2-** Cerrar tarea.

**RF 2-** Registrar las herramientas en uso por el usuario y el tiempo de ejecución de las mismas.

**RF 3-** Ordenar el registro de las herramientas por Tarea, Tiempo de ejecución, Porcentaje de utilización y hora de inicio.

**RF 4-** Generar registro de control del tiempo.

**RF 5-** Obtener el reporte de tiempo de las herramientas utilizadas por un usuario.

## 2.7.2. Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir, una vez se conozca lo que el sistema debe hacer se podrá determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

- ✓ **Requerimientos de software:** El sistema será multiplataforma y solo necesita para funcionar la máquina virtual de Java JDK 6.
- ✓ **Requerimientos de hardware:** Como requerimientos mínimos debe tener un procesador Pentium IV a 2.81 GHz o superior y 256 MB de RAM.
- ✓ **Requerimientos de Diseño e implementación:** Para el análisis y el diseño del sistema se empleará la metodología RUP y como lenguaje de modelado UML. La herramienta consecuente con el lenguaje de modelado será el Visual Paradigm. La aplicación será implementada en la plataforma J2EE y como lenguaje de programación se utilizará Java.
- ✓ **Requerimientos de apariencia o interfaz externa:** El sistema deberá estar oculto o minimizado todo el tiempo que se ejecuta y solo mostrará una interfaz con las opciones disponibles en consecuencia con los pedidos del usuario.
- ✓ **Requerimientos de Seguridad:**
  - **Confidencialidad:** La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.
  - **Integridad:** la información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos.
  - **Disponibilidad:** a los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

# Características del Sistema

---

- ✓ **Requerimientos de Usabilidad:** La aplicación podrá ser utilizada por usuarios con conocimientos mínimos de computación.
- ✓ **Requerimientos Legales:** La aplicación deberá distribuirse bajo al licencia GPL.

## 2.8. Modelo de Casos de Usos del Sistema

Un caso de uso es un pedazo o fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante, los casos de uso representan los requisitos funcionales y guían el diseño, la implementación y pruebas en el proceso de desarrollo.

Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones. El modelo de casos de uso permite que los desarrolladores y los clientes lleguen a un acuerdo sobre los requisitos, es decir sobre las condiciones y posibilidades que debe cumplir el sistema. El modelo de casos de uso describe lo que hace el sistema para cada tipo de usuario. Cada usuario se representa mediante uno o más actores.

### 2.8.1. Actores del Sistema

Los actores representan terceros fuera del sistema que colaboran con el sistema. Al identificar los actores del sistema se identifica el entorno externo del sistema.

Actor del Sistema	Descripción
Usuario	Accede mediante el sistema a las tareas que tiene abiertas o asignadas en el gestor de tareas y guarda el reporte de las herramientas utilizadas y el uso de las mismas en un repositorio.
Jefe de Proyecto	Utiliza el sistema para chequear los registros guardados por los miembros del proyecto

**Tabla 2.2 Definición de los actores del sistema.**

### 2.8.2. Casos de Uso del Sistema

Los casos de uso son el componente clave del modelado. Su propósito es ilustrar como un sistema permite a un actor cumplir una meta, ilustrando todos los posibles caminos apropiados que ellos

## Características del Sistema

pueden tomar para cumplirla, así como las situaciones que podrían hacerlo fallar. Los casos de uso son los siguientes:

- Obtener Tarea.
- Registrar software y uso.
- Ordenar Información.
- Guardar información.
- Obtener Reporte.

### 2.8.3. Diagrama de Casos de Uso del Sistema

Un diagrama de casos de uso del sistema representa gráficamente las funcionalidades principales del sistema y su interacción con los actores.

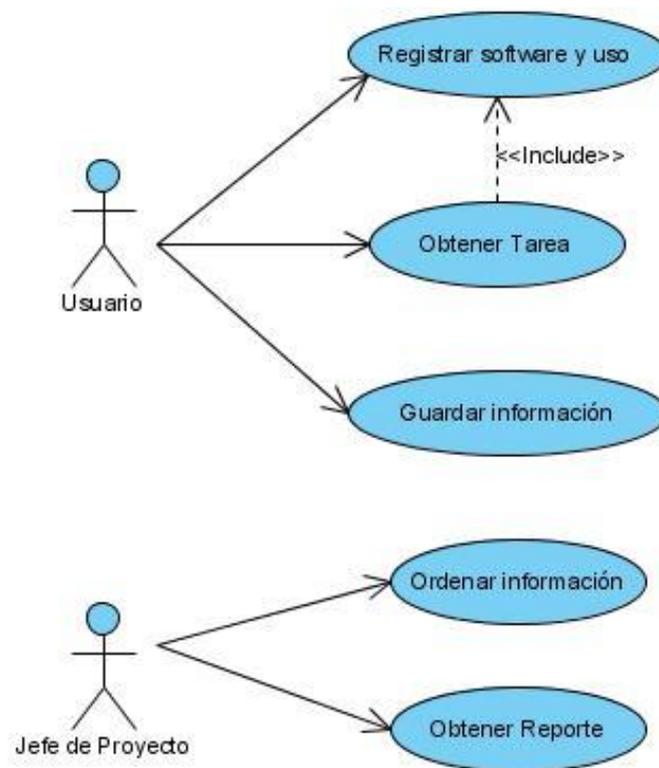


Figura 2.2 Diagrama de Casos de Uso del Sistema  
Fuente: Elaboración propia

## Características del Sistema

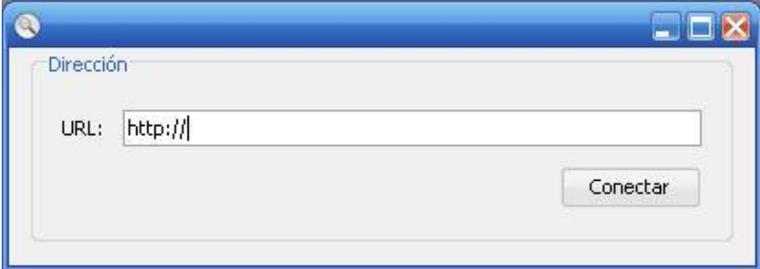
### 2.8.4. Especificación textual extendida de los casos de uso

Un caso de uso consiste principalmente de una especificación textual (llamada Especificación del caso de uso), que contiene una descripción del flujo de eventos, describiendo la interacción entre actores y el sistema. La especificación contiene además otras informaciones, como son pre-condiciones, pos-condiciones y prioridad del caso de uso.

#### Caso de Uso Obtener Tarea.

<b>Caso de uso</b>	Obtener Tarea	
<b>Actores</b>	Usuario	
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea comenzar a realizar una tarea abierta o asignada. El caso de uso termina cuando el sistema le muestra al usuario las tareas contenidas en el Gestor de Tareas Jira.	
<b>Referencias</b>	RF 1; RF 1.1; RF 1.1.1; RF 1.1.2; RF 1.1.3; RF 1.2.	
<b>Prioridad</b>	Crítica	
<b>Precondición</b>	La URL del RSS del gestor de tarea debe de ser sobre un filtro que contenga exclusivamente las tareas que el trabajador debe de hacer.	
<b>Flujo Normal de Eventos</b>		
	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1. El usuario ejecuta la aplicación.	2. El sistema muestra en el menú principal: <ul style="list-style-type: none"><li>• Descargar tarea</li><li>• Ver reporte</li><li>• Configuración</li><li>• Salir</li></ul>
	3. Selección la opción de descargar tarea	4. Muestra la opción de conectarse al gestor de tareas. Brinda la opción de indicar la URL (Localizador Uniforme de Recursos) del RSS de sus tareas.
	5. Introduce la dirección requerida.	6. Valida que la dirección sea correcta.
		7. En caso de ser correcta, el sistema brinda la opción de autenticarse.
	8. El usuario se autentica.	9. Si los datos de la autenticación son correctos, el sistema conectará con el Gestor de Tareas Jira para descargar las tareas abiertas o asignadas por el usuario. <ul style="list-style-type: none"><li>• ID de tarea</li></ul>

## Características del Sistema

	<ul style="list-style-type: none"> <li>Nombre de tarea</li> <li>Tipo de tarea</li> </ul>
	<p>10. Descarga las tareas preparadas para comenzar a trabajar y las muestra en un listado. La información del listado será la siguiente:</p> <ul style="list-style-type: none"> <li>EL número que identifica la tarea.</li> <li>Parte del nombre de la tarea en un espacio de 30 caracteres seguidos por "...".</li> <li>Los iconos de las funcionalidades de controlar el tiempo: Comenzar, Pausar y Parar.</li> </ul>
<b>Flujo Alternativo I</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	6. En caso de que la dirección no sea correcta, el sistema muestra un mensaje de error.
	7. Vuelve a la acción 4.
<b>Flujo Alternativo II</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
	9. Si los datos de la autenticación no son correctos, el sistema muestra un mensaje de error.
	10. Vuelve a la acción 7.
	

## Características del Sistema



<b>Poscondiciones</b>	La tarea es seleccionada y lista para ser controlada.
-----------------------	---

**Tabla 2.3 Descripción CUS Obtener Tarea**

### Caso de Uso Registrar Software y uso.

<b>Caso de uso</b>	Registrar Software y uso
<b>Actores</b>	Usuario
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario selecciona la tarea abierta o asignada a realizar. El sistema registra en tiempo real todas las incidencias y herramientas en utilización y el tiempo de uso de las mismas. El caso de uso termina cuando el usuario concluye o desea cerrar la tarea.
<b>Referencias</b>	<b>RF 2.</b> <b>Incluido CUS Obtener Tarea.</b>
<b>Prioridad</b>	Crítica
<b>Precondición</b>	El sistema debe mostrar el listado con las tareas pertenecientes al usuario.
<b>Flujo Normal de Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. Escoge la tarea en la que va a trabajar. Para esto, selecciona el icono que representa la funcionalidad de control de tiempo (comenzar).	2. Crea el archivo de persistencia donde incluye los datos de las tarea: <ul style="list-style-type: none"> <li>• Usuario.</li> <li>• ID de la tarea.</li> </ul>

## *Características del Sistema*

	<ul style="list-style-type: none"> <li>• Nombre de la tarea.</li> <li>• Fecha.</li> <li>• Tipo de tarea.</li> </ul>
	3. El sistema analiza y determina el Sistema operativo.
	4. Busca la ventana activa y extrae la información <ul style="list-style-type: none"> <li>• Nombre del software.</li> <li>• Acciones que se realizan.</li> </ul>
	5. Registra el tiempo que dura activada la ventana.
6. El usuario realiza cualquier cambio que implique el cambio de foco sobre la ventana activa.	7. EL sistema registra en el archivo temporal los datos de la ventana y el tiempo controlado hasta el momento y vuelve a la acción 4.
<b>Flujo Alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El usuario no ejecuta ningún software.	2. El sistema queda en espera de la ejecución de algún software.
<b>Poscondiciones</b>	Se registran en el sistema todas las actividades que realizó el usuario para desarrollar la tarea.

**Tabla 2.4 Descripción CUS Registrar software y uso**

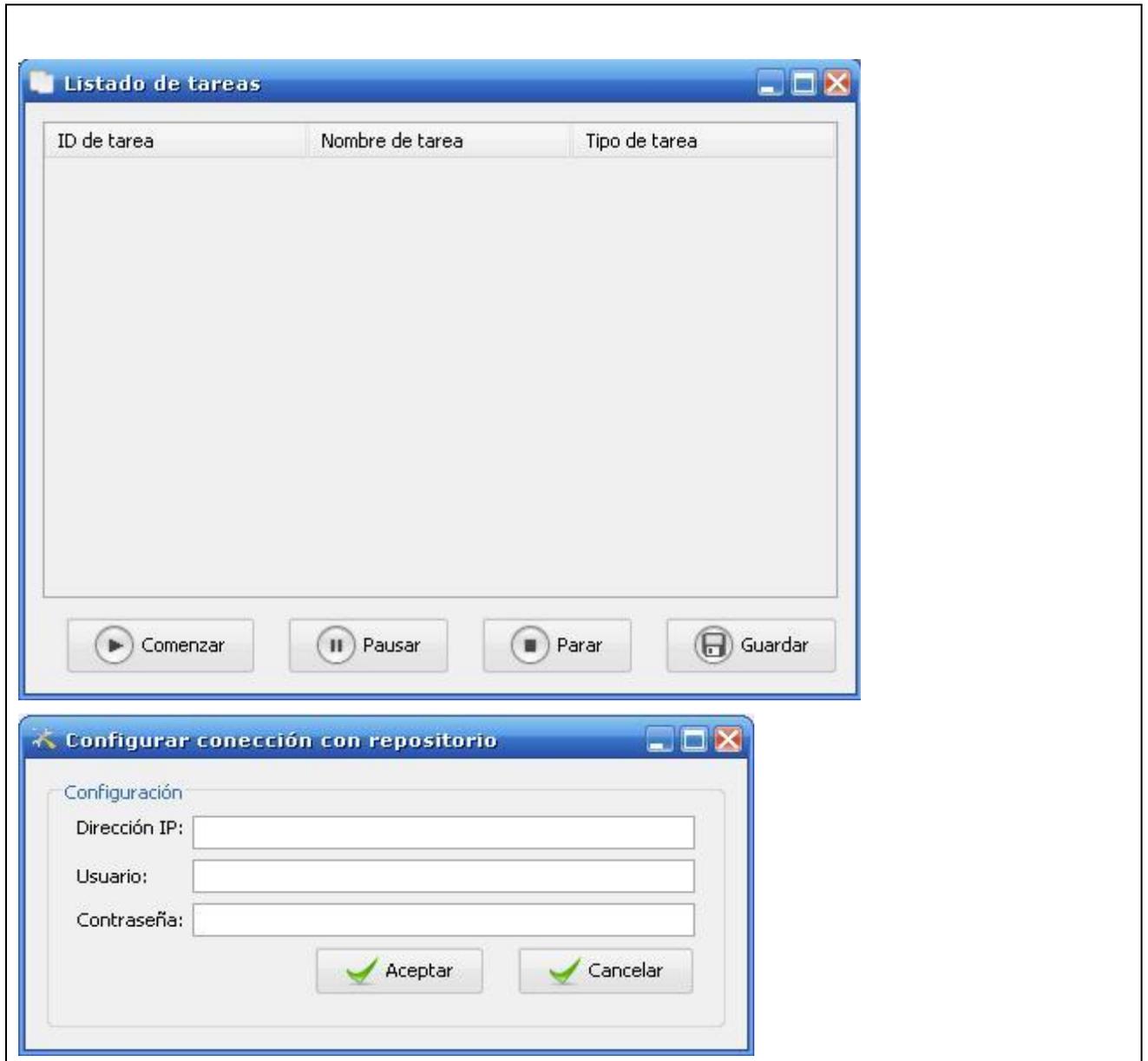
### **Caso de Uso Guardar Información.**

<b>Caso de uso</b>	Guardar Información
<b>Actores</b>	Usuario
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario concluye o cierra la tarea. El sistema conectará con el repositorio de registros y se guardará el listado con las herramientas usadas y el tiempo que empleó el usuario en cada una. El caso de uso termina cuando el listado es guardado con éxito.
<b>Referencias</b>	RF 4.
<b>Prioridad</b>	Crítica
<b>Precondición</b>	El usuario realizó la tarea
<b>Flujo Normal de Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El usuario selecciona la opción "Guardar"	2. El sistema guarda en el archivo temporal la

## *Características del Sistema*

y/o la funcionalidad de parar tarea.	<p>información de la ventana activa:</p> <ul style="list-style-type: none"> <li>• Nombre del software.</li> <li>• Acciones que se realizan.</li> </ul>
	<p>3. Cierra el archivo temporal con la información general más la específica:</p> <ul style="list-style-type: none"> <li>• Usuario.</li> <li>• ID de la tarea.</li> <li>• Nombre de la tarea.</li> <li>• Fecha.</li> <li>• Herramienta o software.</li> <li>• Tiempo de uso de la herramienta o software.</li> </ul>
	4. Se cierra el archivo
	5. Se muestra la opción para introducir la dirección de IP además del usuario y contraseña para persistir en el repositorio de información.
6. Introduce la dirección IP y las credenciales y selecciona la opción "Aceptar".	7. El sistema se conecta, verifica los permisos y guarda el archivo con el control de tiempo.
<b>Flujo Alternativo I</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
6. El usuario no introduce los datos y selecciona la opción "Aceptar".	7. Muestra un mensaje de error.
	8. Vuelve al paso 5.
<b>Flujo Alternativo II</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
6. Introduce los datos y selecciona la opción "Cancelar".	7. Se cancela el guardado y se cierra el formulario.

## Características del Sistema



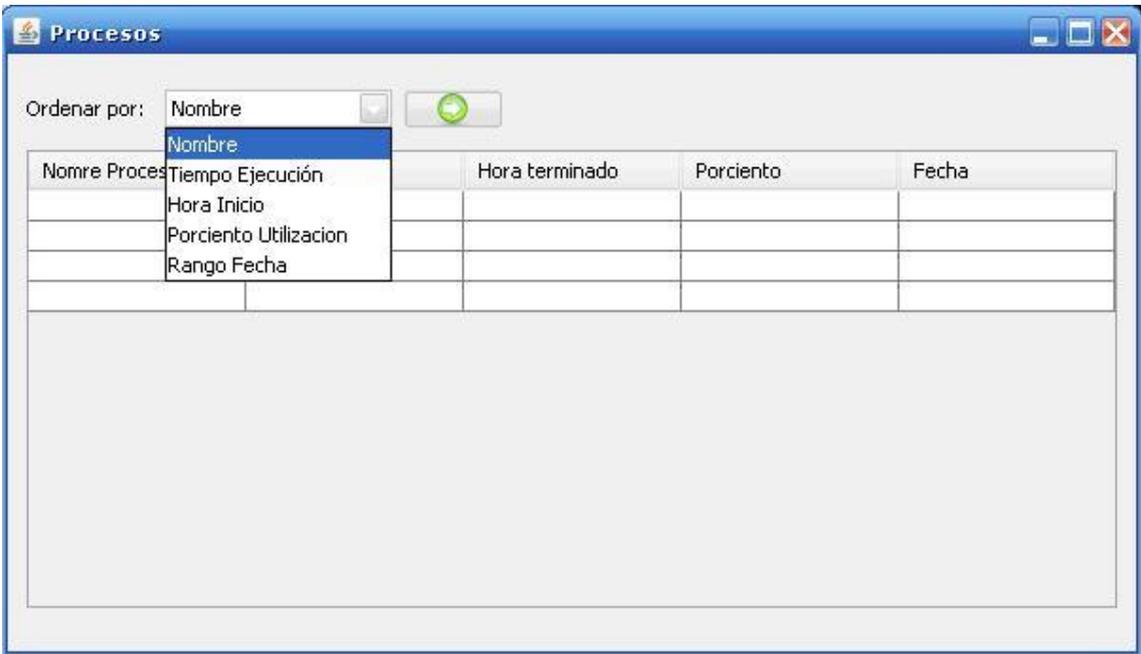
<b>Poscondiciones</b>	Se guarda el listado con la información de las herramientas usadas en el sistema y el tiempo que se empleó en cada una.
-----------------------	---

**Tabla 2.5 Descripción CUS Guardar Información**

### Caso de Uso Ordenar Información.

<b>Caso de uso</b>	<b>Ordenar Información</b>
<b>Actores</b>	Jefe de Proyecto
<b>Resumen:</b>	El caso de uso se inicia cuando el jefe de proyecto desea ordenar la información de acuerdo a criterios como: <i>Nombre, Tiempo de ejecución, Porcentaje de utilización y Hora de inicio</i> de cada herramienta. El caso de uso termina cuando es mostrada la información ordenada por el criterio seleccionado.

## Características del Sistema

<b>Referencias</b>	RF 3.	
<b>Prioridad</b>	Primaria	
<b>Precondición</b>	Debe haberse mostrado el reporte.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1. El jefe de proyecto ejecuta una de las siguientes acciones: a) Selecciona la opción "Nombre". b) Selecciona la opción "Tiempo". c) Selecciona la opción "Porcentaje". d) Selecciona la opción "Inicio". e) Selecciona la opción "por rango de fecha".	2. El sistema realiza lo siguiente: a) Muestra las herramientas y tiempo de uso ordenadas por el nombre de las mismas. b) Muestra las herramientas ordenadas por el tiempo que se ejecutaron. c) Muestra las herramientas ordenadas por el porcentaje de utilización que tuvo cada una. d) Muestra las herramientas ordenadas por la hora en que se iniciaron las mismas. e) Muestra las herramientas y tiempo de uso ordenadas por la fecha seleccionada.	
<b>Flujo Alternativo</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1. El jefe de proyecto no ejecuta ninguna acción.	2. El sistema queda en espera de alguna acción.	
		
<b>Poscondiciones</b>	Se ordena la información de las herramientas usadas para un mejor entendimiento.	

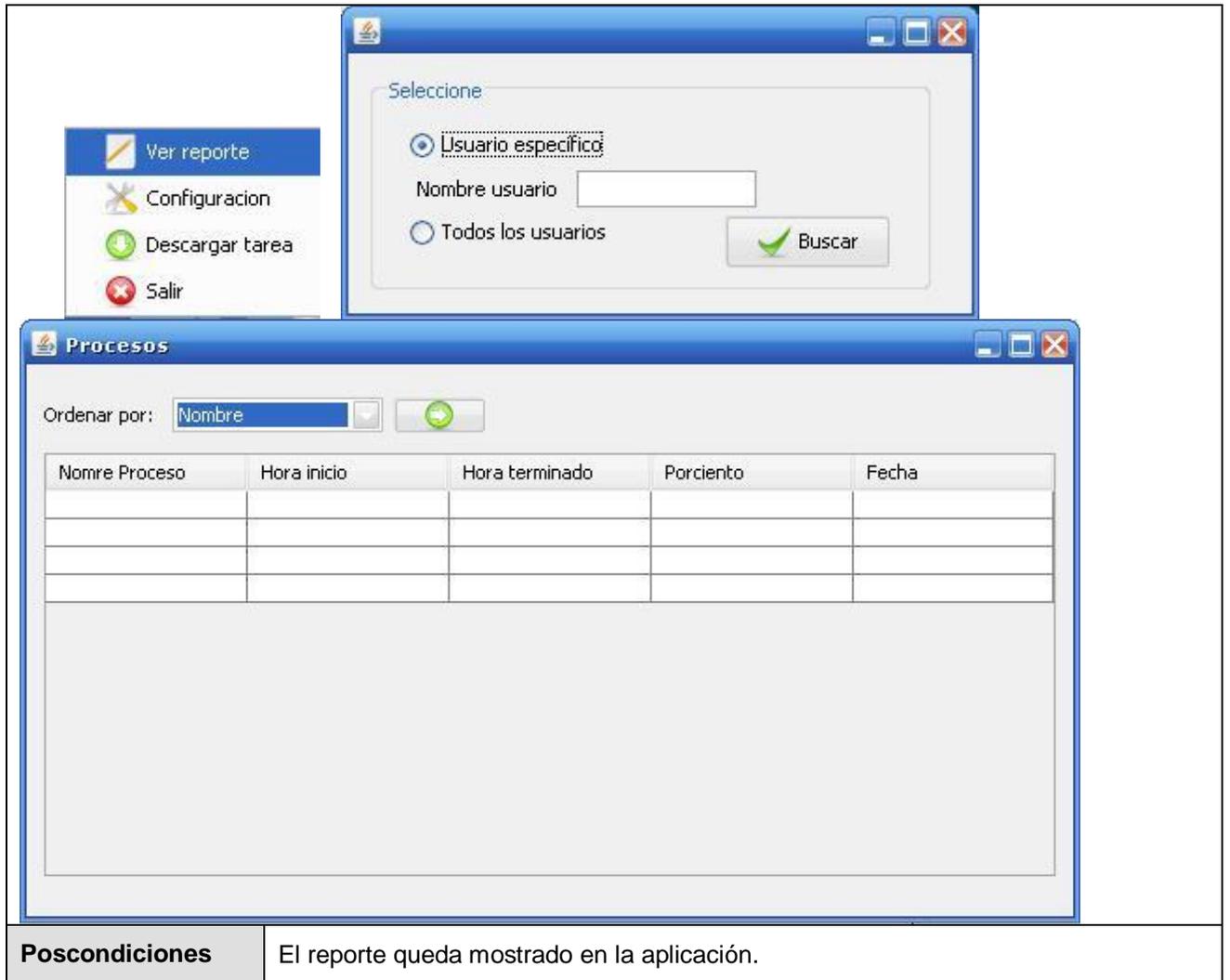
**Tabla 2.6 Descripción CUS Ordenar Información**

## *Características del Sistema*

### Caso de Uso Obtener Reporte.

<b>Caso de uso</b>	<b>Obtener Reporte</b>	
<b>Actores</b>	Jefe de Proyecto	
<b>Resumen:</b>	El caso de uso se inicia cuando el Jefe de Proyecto desea chequear la información guardada en el repositorio. El caso de uso termina cuando es mostrada la información al Jefe de Proyecto.	
<b>Referencias</b>	<b>RF 5.</b>	
<b>Prioridad</b>	Critica	
<b>Precondición</b>	Debe haberse realizado una salva de la información.	
<b>Flujo Normal de Eventos</b>		
	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1.	El jefe de proyecto selecciona la opción "Ver Reporte".	2. El sistema muestra opción para obtener : <ul style="list-style-type: none"> <li>• Usuario específico.</li> <li>• Todos los usuarios.</li> </ul>
3.	El jefe de proyecto introduce el usuario o selecciona todos los usuarios y acepta la petición.	4. El sistema se conecta al repositorio de información y carga la información del usuario o de todos los usuarios y continúa en el CUS <b>Ordenar Información.</b>
<b>Flujo Alterno</b>		
	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
		5. Si la extensión del reporte seleccionado no es el generado por la aplicación, el sistema muestra un mensaje y vuelve a la acción 2.

## Características del Sistema



The screenshot displays two overlapping windows from a software application. The top window, titled 'Seleccione', contains a search form with two radio buttons: 'Usuario específico' (selected) and 'Todos los usuarios'. Below the radio buttons is a text input field labeled 'Nombre usuario' and a 'Buscar' button with a green checkmark icon. To the left of this window is a vertical menu with four items: 'Ver reporte' (with a pencil icon), 'Configuracion' (with a wrench icon), 'Descargar tarea' (with a green download icon), and 'Salir' (with a red exit icon). The bottom window, titled 'Procesos', features a dropdown menu set to 'Nombre' and a green refresh icon. Below this is a table with five columns: 'Nomre Proceso', 'Hora inicio', 'Hora terminado', 'Porciento', and 'Fecha'. The table is currently empty. Below the table is a large, empty rectangular area.

Nomre Proceso	Hora inicio	Hora terminado	Porciento	Fecha

**Poscondiciones** El reporte queda mostrado en la aplicación.

Tabla 2.7 Descripción CUS Obtener Reporte

## *Conclusiones Parciales*

---

### **Conclusiones Parciales**

Luego de haber expuesto las principales causas que conllevan a la realización del presente trabajo, como son: la planificación irreal de las tareas, la duración de las mismas y la entrega atrasada de las tareas asignadas; la solución propuesta se basa en, registrar el uso del tiempo de máquina utilizado por un trabajador en el desarrollo de un proyecto de software, de acuerdo a sus tareas específicas; además, generará registros de tiempo para realizar una planificación real.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

### 3. Introducción

En el análisis se refinan y estructuran los requisitos obtenidos con anterioridad, profundizando en el dominio de la aplicación, lo que permitirá una mayor comprensión del problema para definir las clases y modelar la solución. Durante el diseño se toman decisiones estratégicas y tácticas para cumplir los requerimientos funcionales y no funcionales de un sistema. Uno de los objetivos del diseño es crear una entrada apropiada, creando una arquitectura estable y sólida.

#### 3.1. Modelo de Análisis

El modelo de análisis servirá para lograr una mayor comprensión del dominio del problema descrito en el capítulo anterior por un Modelo de Dominio que se elaboró. También será indispensable a su vez como punto de partida para realizar el diseño del sistema.

En el modelo de análisis no se toman en cuenta el lenguaje de programación que se va a utilizar en la construcción ni otros aspectos como la plataforma o los componentes reutilizables de otros sistemas, pues su objetivo principal es comprender, preparar, modificar y en general mantener exactamente los requisitos del software y no precisar cómo se realizará su implementación. Se considera que este modelo es la primera aproximación al Modelo de Diseño y se describe en un lenguaje comprensible para los desarrolladores.

##### 3.1.1. Diagramas de clases del análisis

El diagrama de clases del análisis (DCA) es un artefacto en el que se representan los conceptos del dominio de un problema y que representa las cosas del mundo real, no de la implementación automatizada. Está compuesto por clases del análisis y sus relaciones. Las clases del análisis están centradas en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos importantes y relaciones del dominio.

Estas clases se clasifican en:

- ✓ *Clases de interfaz*: Se utilizan para modelar la interacción entre el sistema y sus actores.
- ✓ *Clases de Control*: Representan coordinación, secuencia, transacciones, y control de otros objetos.
- ✓ *Clases de Entidad*: Se emplean para modelar información que posee una larga vida y que es a menudo persistente.

# Análisis y Diseño del Sistema

Seguidamente se muestran los diagramas de clases del análisis de los casos de uso del sistema.

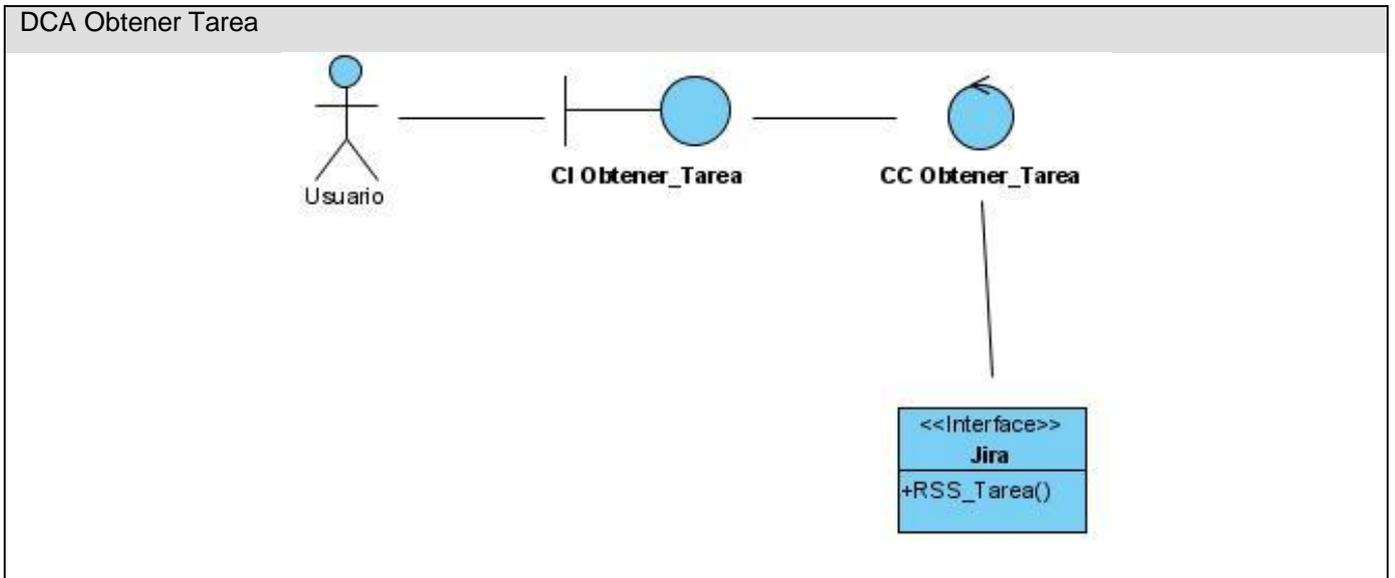


Figura 3.1 Diagrama de Clases del Análisis CU Obtener Tarea  
Fuente: Elaboración propia

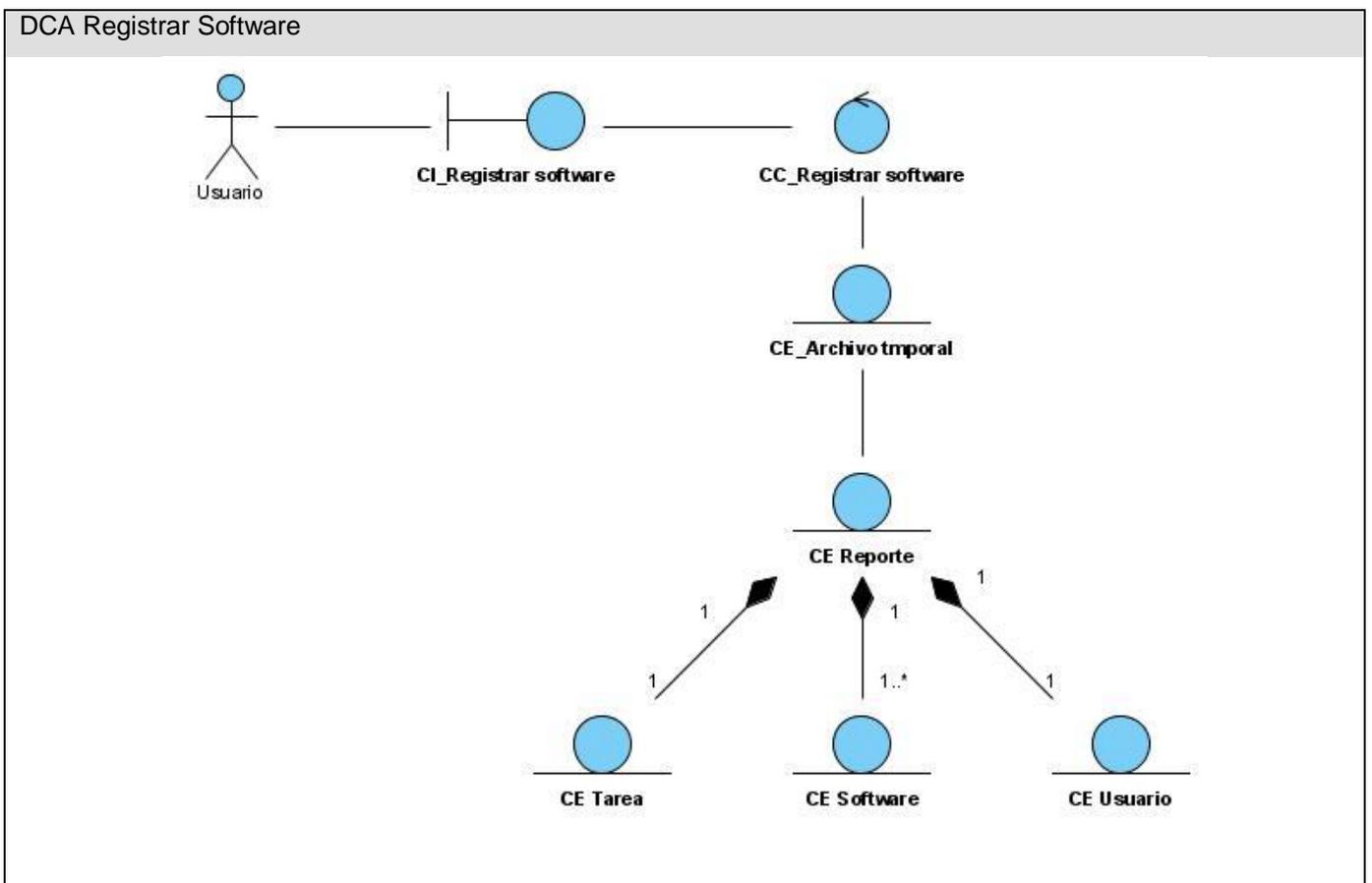


Figura 3.2 Diagrama de Clases del Análisis CU Registrar Software y Uso  
Fuente: Elaboración propia

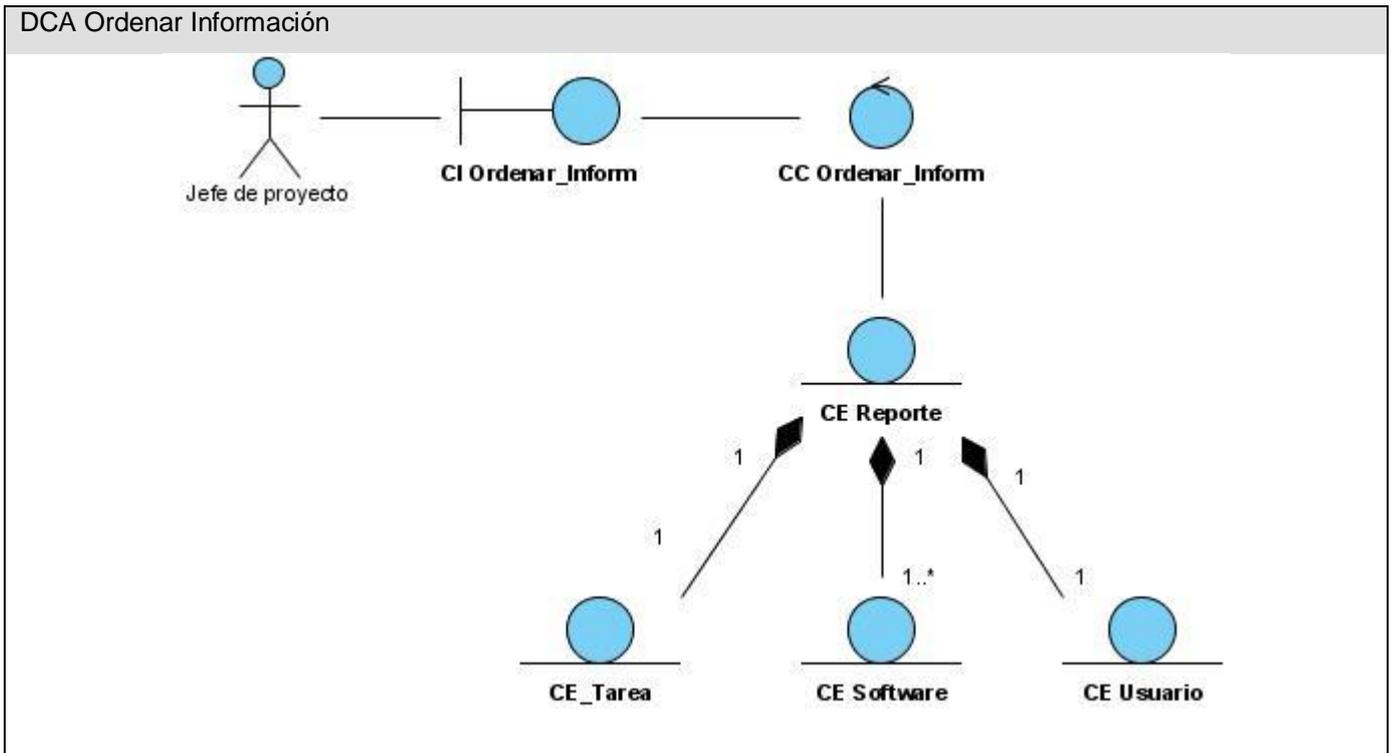


Figura 3.3 Diagrama de Clases del Análisis CU Ordenar Información  
Fuente: Elaboración propia

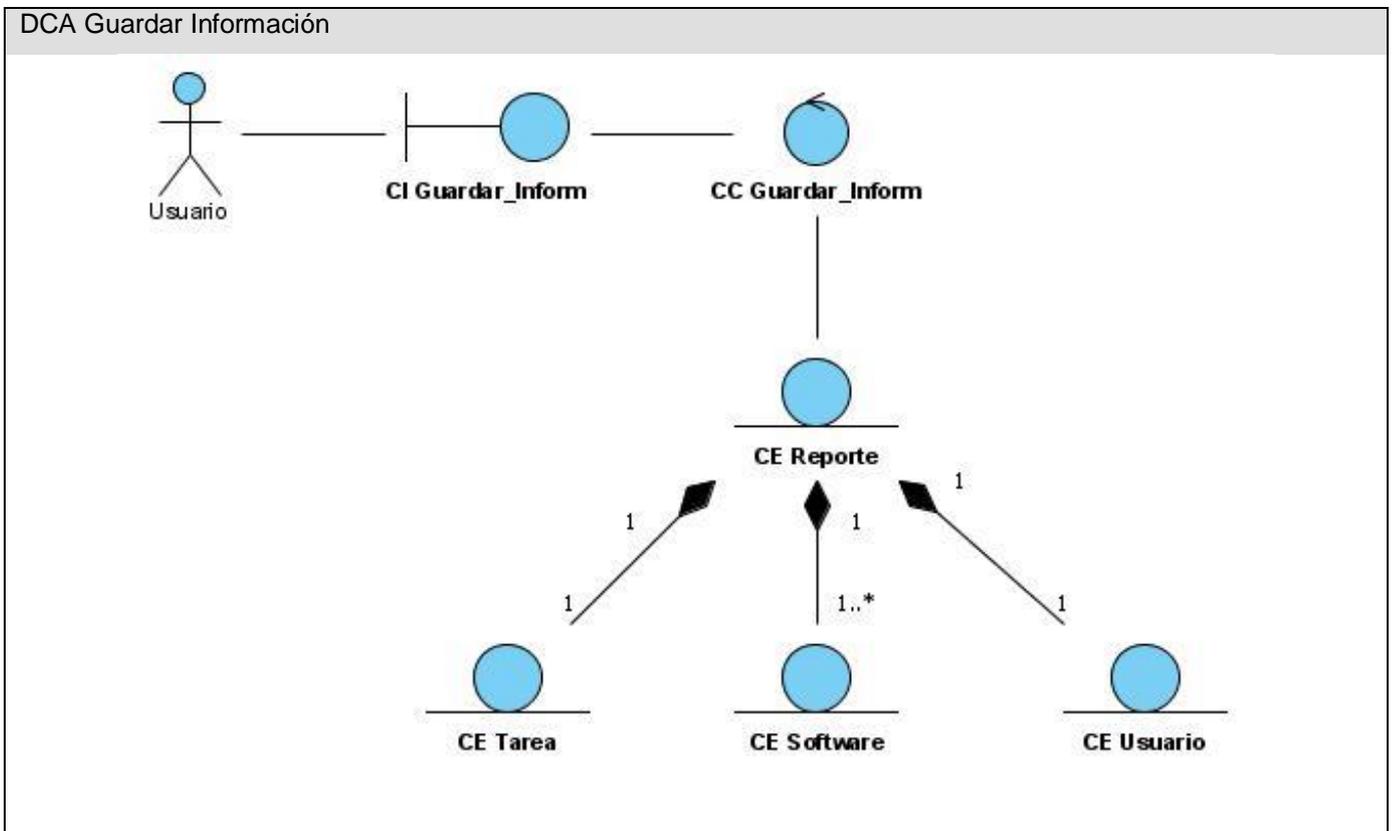


Figura 3.4 Diagrama de Clases del Análisis CU Guardar Información

Fuente: Elaboración propia

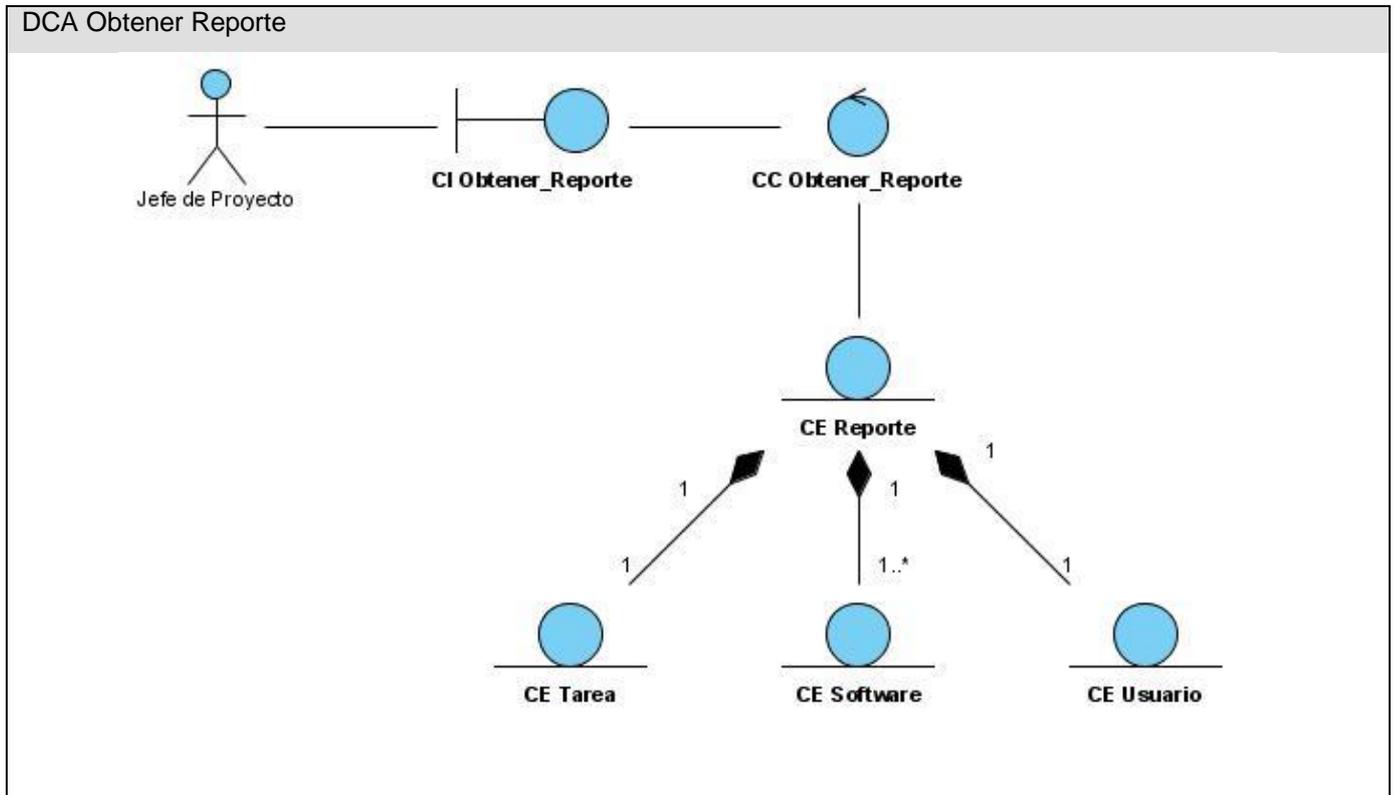


Figura 3.5 Diagrama de Clases del Análisis CU Obtener Reporte  
Fuente: Elaboración propia

## 3.2. Modelo del Diseño

Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis que proporciona una comprensión detallada de los requisitos. En el diseño se modela el sistema y se encuentra su forma, incluyendo la arquitectura, para que soporte todos los requisitos funcionales, no funcionales y las restricciones que se le suponen. Es específico para una implementación y depende del lenguaje de programación.

### 3.2.1. Definiendo la Arquitectura

Se tomará como definición de Arquitectura de Software la siguiente:

La Arquitectura de Software de un programa o sistema es la estructura o estructuras del sistema, la cual comprende los componentes de software, las propiedades externas visibles de estos elementos y las relaciones entre ellos.(AMBLER, 2008)

## 3.2.1.1. Estilo Arquitectónico usado

La arquitectura a proponer se basa en el estilo N-Capas, específicamente contará con 3 capas: Interfaz Usuario, Lógica del Negocio y Persistencia. La misma se encuentra dentro de la clasificación de los estilos de llamada y retorno. (APRENDIZAJE, 2009)

Algunas ventajas de este estilo arquitectónico son:(ZYNCRODIGMA)

- ✓ Aplicaciones más robustas debido al encapsulamiento.
- ✓ Mantenimiento y soporte más sencillo pues es más sencillo hacer cambios en una capa que modificar la aplicación completa.
- ✓ Mayor flexibilidad porque se pueden añadir nuevos módulos para dotar al sistema de nuevas funcionalidades.
- ✓ Alta escalabilidad pues se puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware.



**Figura 3.6** Arquitectura en capas del sistema  
Fuente: Elaboración propia

Capa de Interfaz Usuario

En esta capa se va a encontrar el paquete Interface que va a contener las clases de las interfaces que interactúan con el usuario. Esta es la mediadora entre la lógica del negocio y el usuario, mostrando la respuesta que le proporcione la capa inmediatamente inferior Lógica del Negocio a las peticiones del usuario.

## Capa de Lógica del Negocio

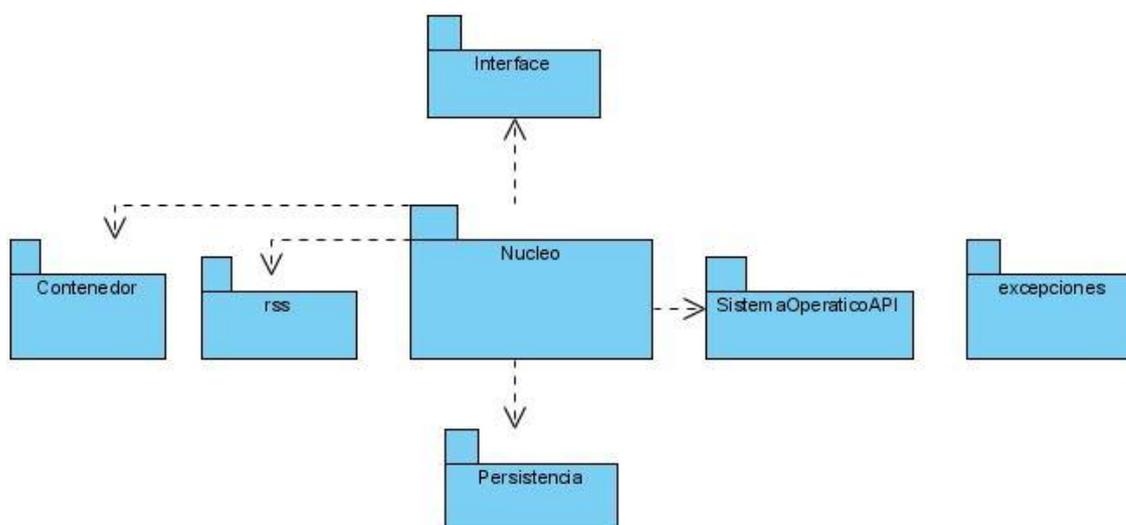
Ésta capa va a contener los paquetes Núcleo, Contenedor, SistemaOperativoAPI y Conexión RSS. Aquí es donde recae el mayor peso de la aplicación ya que va a contener la clase núcleo que es la que realiza la mayoría de las actividades apoyándose en las demás clases para lograr obtener las tareas, registrarlas y controlarlas y finalmente realizando una salva evidenciando así la relación con la Capa Persistencia.

## Capa de Persistencia

La capa de Persistencia recibe las peticiones de la capa inmediatamente superior Lógica del Negocio que en este caso sería simplemente realizar la salva de datos en ficheros XML tanto local como en un servidor previamente conocido.

### 3.2.2. Estructura en Paquetes del Diseño

Los paquetes del diseño se usan para reflejar la organización de paquetes y sus elementos. Los usos más comunes de los diagrama de paquetes son, para organizar diagramas de casos de uso y diagramas de clases; estos paquetes son como grandes contenedores de clases.(MARTIN, 2008)



**Figura 3.7 Diagrama de Paquetes**  
Fuente: Elaboración propia

## *Análisis y Diseño del Sistema*

---

En la Figura 10 se representan los paquetes que contienen todas las clases que el sistema utiliza. La aplicación está compuesta de un paquete principal Núcleo que es imprescindible para el funcionamiento del sistema, éste se relaciona con los demás paquetes para solucionar las necesidades de los usuarios.

- ✓ El paquete Núcleo es el encargado de realizar todas las operaciones vinculadas al sistema apoyándose en los demás paquetes por lo que lo convierte en el paquete principal del sistema y se relaciona con los demás paquetes. Mediante las clases que contiene se ejecutan todas las peticiones del usuario como: conectarse al gestor de tareas para obtener las tareas, mostrar las tareas, registrar el uso de las aplicaciones utilizadas y guardar el reporte de ese tiempo y la tarea, entre otros.
- ✓ Por otra parte el paquete Interface está compuesto por las interfaces, formularios, ventanas, con las que tiene que trabajar el usuario para poder realizar todas las operaciones anteriormente expuestas. Esta es la mediadora entre el usuario y la Lógica del Negocio.
- ✓ En el paquete SistemaOperativoAPI se encuentran las clases que son las encargadas de verificar qué sistema operativo es el que se está usando y así, obtener los datos necesarios para registrar el uso de las aplicaciones que corren en el sistema. También están las clases que devuelven el estado de las ventanas y de los procesos en ejecución.
- ✓ Para poder realizar la conexión con el gestor de tareas y descargar las tareas de un usuario en específico se utiliza el paquete RSS. En el mismo se lleva a cabo la conexión con el RSS y luego el intérprete de RSS reconoce el documento XML enviado por el gestor de tareas en este caso Jira.
- ✓ Luego se efectúa la persistencia mediante el paquete PersistenciasXML, que en sí lo que hace es, cargar la información de las tareas y los procesos y la guarda en un fichero XML. Si no existe dicho fichero, el sistema brinda la oportunidad de crearlo.
- ✓ En el paquete Contenedor van a estar agrupadas todas las clases referentes al registro del tiempo, los datos de la tarea y los procesos involucrados. Los procesos van a tener un tiempo de vida que es desde que se inicio el proceso hasta que se termino y también un rango de tiempo activo que suma la cantidad de veces en que estuvo ese proceso en actividad. También está la clase Rectángulo que brinda las dimensiones de la ventana en actividad.
- ✓ El paquete de Excepciones se va a encargar de gestionar todos los errores que lanza el sistema operativo.

### **3.2.3. Diagramas de Clases del Diseño.**

El diagrama de Clases de diseño (DCD) describe los tipos de objetos que hay en el sistema y las diversas clases de relaciones que existen entre ellos [Ver Anexo 1]. Además muestra los atributos y operaciones de una clase y las restricciones a que se ven sujetos, según la forma en que se conecten los objetos. Un diagrama de clases está compuesto por los siguientes elementos: Clase (Atributos, métodos y visibilidad) y Relaciones (Herencia, Agregación, Asociación y Composición).

### **3.2.4. Diagramas de secuencia del Diseño**

Los diagramas de secuencia destacan el orden temporal de los mensajes, o sea, la forma en que las operaciones se producen en el tiempo. Se crean durante la elaboración, cuando es útil identificar los detalles de los eventos del sistema para poner en claro cuáles son las operaciones que se deben diseñar que gestione el sistema, escribir los contratos de las operaciones del sistema y posiblemente, dar soporte a la estimación.

Para modelar los aspectos dinámicos del sistema se utilizaron los diagramas de secuencia por cada escenario de caso de uso. Debido al gran tamaño que adoptan los mismos, no es posible mostrarlos en este epígrafe, ya que se verá afectada la legibilidad de los mismos [Ver Anexo 2].

### **3.2.5. Patrones utilizados en el diseño**

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño es una solución a un problema de diseño.

Para que una solución sea considerada un patrón debe poseer ciertas características:

- ✓ Son soluciones concretas y técnicas.
- ✓ Se utilizan en situaciones frecuentes.
- ✓ Favorecen la reutilización de código.
- ✓ Es difícil reutilizar la implementación de un patrón.

Un patrón de diseño proporciona un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos.

**Ventajas de los patrones de diseño:** La principal ventaja es que proponen una forma de reutilizar la experiencia de expertos en el proceso de desarrollo de software, en busca de no cometer los mismos errores. Por ello clasifican formas de solucionar problemas que ocurren de forma frecuente en el desarrollo y se describen como experiencias reales, probadas y funcionales.

En el diseño de la herramienta se tuvieron en cuenta los patrones GRASP (*Patrones Generales de Software para Asignación de Responsabilidades*). Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Estos fueron los patrones que se aplicaron para realizar el diseño de la aplicación que propone la investigación.

- ✓ *Experto:* Es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. El mismo expresa que siempre se debe asignar una responsabilidad al experto en información, o sea, a la clase que cuenta con la información necesaria para llevar a cabo la funcionalidad
- ✓ *Controlador:* Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. La mayor parte de los sistemas reciben eventos de entrada externa, En cualquiera de los casos que puedan existir, si se recurre a un diseño orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada.

En búsqueda de un diseño claro, flexible y reutilizable también se utilizaron patrones de diseño GOF. El patrón Singleton (instancia única) garantiza que solo haya una única instancia de una clase y esta sea accesible desde cualquier punto de la aplicación. Es necesario cuando hay clases que tienen que gestionar de manera centralizada un recurso.

### 3.3. Modelo de Datos

El modelo físico de los datos contiene un conjunto de tablas que conforman la base de datos. Este modelo constituye entonces la representación lógica y física del modelo de clases persistentes. Como la aplicación no presenta una base de datos sino que lo único que realiza es una salva de ficheros XML y las demás entidades utilizadas son temporales y no se guardan en una base de datos, solo se representa la tabla con los campos que contiene el fichero XML guardado en el servidor.

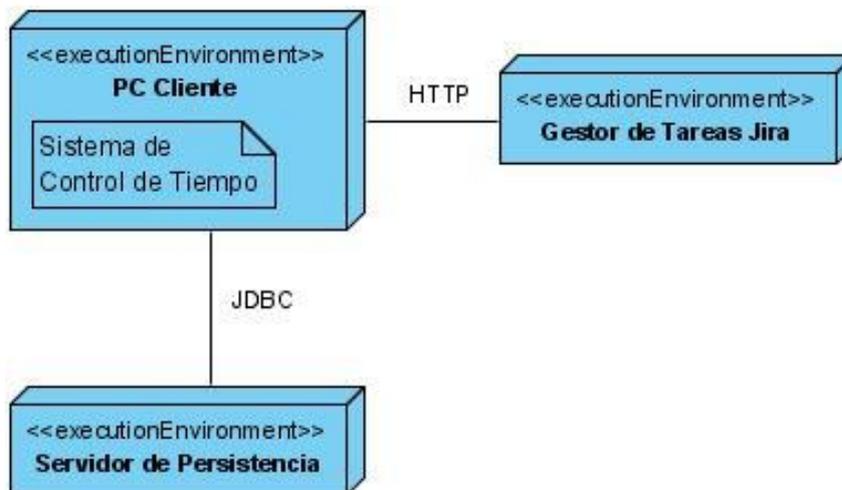
Archivo Temporal		
usuario	char(15)	Nullable = false
id_tarea	integer(15)	Nullable = false
nombre_tarea	integer(25)	Nullable = false
fecha	date	Nullable = false
software_utilizado	char(30)	Nullable = false
tiempo_uso	integer(30)	Nullable = true

**Figura 3.8 Modelo Físico de Datos**  
Fuente: Elaboración Propia

### 3.4. Modelo de Despliegue

RUP propone que en el diseño del sistema se realice un primer acercamiento al diagrama de despliegue; el modelo de despliegue representa la ubicación física que tendrá el sistema cuando se vaya a desplegar.

Teniendo en cuenta que la solución propuesta está compuesta por las terminales de las áreas de trabajo, un servidor de persistencia y el gestor de tareas, en el diagrama de despliegue del sistema, se representan tres nodos. Uno de los nodos es la PC Cliente, que representa los ordenadores de los usuarios que están utilizando el sistema, los mismos acceden al Gestor de Tareas Jira mediante el protocolo HTTP para la descarga de tareas. Y en el otro nodo se encuentra ubicado el servidor para realizar la persistencia de la información, se podrá acceder a éste mediante el protocolo JDBC ya que es una aplicación desarrollada en java y utiliza este tipo de protocolos para conectarse con servidores o bases de datos.



**Figura 3.9 Diagrama de Despliegue**  
Fuente: Elaboración propia

## *Conclusiones Parciales*

---

### **Conclusiones Parciales**

Realizar el modelo de análisis contribuye en gran medida a comprender mejor el diseño del sistema. Los diagramas de clases del análisis por casos de uso realizados explican de una mejor forma el funcionamiento del sistema.

El modelo del diseño es la entrada principal a las actividades de la implementación brindando una profundización en las características del sistema y en las acciones que debe realizar el mismo. Se construyeron los diagramas de clases del diseño de cada caso de uso y los diagramas de paquetes correspondientes al diseño, demostrando la vinculación con la arquitectura definida.

La arquitectura definida fue la de N-Capa, especificando 3 capas, la capa Interface, Lógica del Negocio y Persistencia. La capa Interface va a contener todas las clases de interfaz que interactúan con el usuario. En la capa Lógica del Negocio va a estar el centro del sistema, la clase núcleo es la encargada de realizar todas las operaciones y dar respuesta a las peticiones del usuario. La capa Persistencia realiza simplemente la salva de los ficheros XML con la información obtenida del registro de tiempo.

Los patrones expuestos guiarán y darán solución a problemas frecuentes que se encontrarán los programadores en el desarrollo de la implementación. Los patrones de asignación de responsabilidades utilizados fueron el Experto y el Controlador y como patrón de diseño el singleton.

# *Conclusiones del Trabajo*

---

## **Conclusiones Generales del Trabajo**

Como conclusiones generales de la investigación se tiene que, el trabajo realizado durante las etapas de Modelo de Dominio, Análisis y Diseño constituye una aproximación a tener en cuenta a la hora de implementar un Sistema de Control del Tiempo.

- ✓ Durante el modelo de dominio se identificaron todos los conceptos y sus relaciones involucrados en el contexto del negocio, para el desarrollo de un sistema de control de tiempo.
- ✓ La etapa del análisis de la solución propuesta permitió estructurar los requisitos definidos previamente obteniendo un mayor entendimiento de los mismos y a su vez, brinda una mejor comprensión sobre el funcionamiento interno de la aplicación propuesta. Esta etapa constituye un acercamiento al diseño de esta aplicación.
- ✓ Para la realización del modelo de diseño se tuvieron en cuenta las descripciones de los casos de usos definidos, lo que permitió el cumplimiento de los requisitos funcionales y no funcionales. En este modelo se definió la arquitectura del sistema a desarrollar.

Se demuestra con todo lo anterior expuesto que, se le dio cumplimiento y solución a los requerimientos, los cuales tributaban a los objetivos propuestos, satisfaciendo así las necesidades del cliente.

## **Recomendaciones**

Se recomienda:

- ✓ Implementar el Sistema de Control de Tiempo teniendo en cuenta el análisis y diseño propuesto en esta investigación.
- ✓ Continuar el estudio de los procesos que se llevan a cabo con el objetivo de agregarle nuevas funcionalidades al sistema.

# Referencia Bibliográfica

---

## Referencia Bibliográfica

1. AMBLER, S. W. *Agile Modeling (AM)*. 2008, Disponible en: <http://www.agilemodeling.com/>.
2. APRENDIZAJE, E. V. D. *Conferencia 6. tendencias arquitectónicas*. 2009, Disponible en: [http://eva.uci.cu/file.php/259/CURSO\\_2008-2009/Materiales\\_Basicos/Semana\\_11/Conf/Conf\\_6.\\_Tendencias\\_arquitectonicas.pdf](http://eva.uci.cu/file.php/259/CURSO_2008-2009/Materiales_Basicos/Semana_11/Conf/Conf_6._Tendencias_arquitectonicas.pdf).
3. BARRA, J. M. D. L. *Atlassian Jira* Última actualización: 2009. Disponible en: <http://www.inflexa.cl/inflexa/soluciones/atlassianjira/>.
4. BODOFF, S. *The J2EE Tutorial*. 2004.
5. COSTA, R. *Dirección de Proyectos: principales definiciones* Disponible en: <http://ramoncosta.blogspot.com/2006/01/direccin-de-proyectos-principales.html>.
6. ETEROVIC, Y. *El Proceso Unificado (RUP): Técnicas Modernas para Desarrollar Aplicaciones*. 2001, Disponible en: <http://www2.ing.puc.cl/~iic3194/rup.ppt>.
7. HUMPHREY, W. S. *Introducción al Proceso Software Personal*. 2001.
8. JACABOSON, I., BOOCH, G., RUMBAUGH J. *El Proceso Unificado de Desarrollo de Software*. 2000.
9. JAMES GOSLING, B. J., GUY STEELE, Y GILAD BRACHA. *The Java language specification, tercera edición*. 2005.
10. MARTIN, M. D. *El CoDiGo K* Disponible en: <http://elcodigok.blogspot.com/2008/03/diagramas-de-paquetes.html>.
11. OLALDE, K. *Gestión del Tiempo del Proyecto*. 2008, Disponible en: <http://www.ehu.es/Degypi/Gestion/gespro1va.htm>.
12. ORALLO, E. H. *El Lenguaje Unificado de Modelado (UML)*. 2008, Disponible en: [http://eva.uci.cu/file.php/102/Curso\\_2008-2009/Materiales\\_Complementarios/Materiales\\_Complementarios\\_Conf\\_1/UML.pdf](http://eva.uci.cu/file.php/102/Curso_2008-2009/Materiales_Complementarios/Materiales_Complementarios_Conf_1/UML.pdf).
13. PALACIOS, J. *Origen de la gestión de proyectos*. 2006, Disponible en: [http://www.navegapolis.net/files/s/NST-001\\_01.pdf](http://www.navegapolis.net/files/s/NST-001_01.pdf).
14. PARADIGM, V. *Paradigma visual para UML* [Consultado el: enero de 2009]. Disponible en: [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%5Bcuenta\\_de\\_Plataforma\\_de\\_Java\\_14715\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/).
15. PÉREZ, M. *Arquitectura para Ambientes CASE Integrados*. Universidad Central de las Villas, 1999.
16. PEROVICH, L. R. V. D. *Gestión de Transacciones en la Plataforma J2EE*. 2001, Disponible en: <http://www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR0312.pdf>.
17. SOURCEFORGE. *Process Dashboard*, Disponible en: <http://processdash.sourceforge.net/>.

## Referencia Bibliográfica

---

18. VALENCIA, U. P. D. *Introducción a RUP*. 2007, Disponible en:  
<https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20a%20RUP.doc>.
19. ZULUETA, I. Y. *Introducción de técnicas del Personal Software Process desde los primeros años en la formación del ingeniero informático*. 2007, Disponible en:  
<http://www.inf.udec.cl/revista/ediciones/edicion14/zulueta.pdf>.
20. ZYNCRODIGMA. *Arquitectura n-capas* Disponible en: <http://zyncrodigma.com.ar/Partners.htm>.

## Bibliografía

1. ESTRADA, D. A. F. Introducción al Proceso de Software Personal. Disponible en: [http://eva.uci.cu/file.php/256/CLASES/CONFERENCIAS/Conferencia\\_2/Otros\\_documentos/Introduccion\\_al\\_PSP.ppt](http://eva.uci.cu/file.php/256/CLASES/CONFERENCIAS/Conferencia_2/Otros_documentos/Introduccion_al_PSP.ppt).
2. GAMMA, E., Y OTROS. Design Patterns: Elements of Reusable Object- Oriented Software. 1998.
3. HUMPHREY, W. S. Introducción al Proceso Software Personal. 2001.
4. JACABOSON, I., BOOCH, G., RUMBAUGH J. El Proceso Unificado de Desarrollo de Software. 2000.
5. KALOYVER. Ingeniería de Software – RUP - UML Disponible en: <http://www.mmug.cl/articulos.php?id=287&appl=otr>.
6. KENDALL, K. Y. K., J. . Análisis y Diseño de Sistemas. 1997.
7. LARMAN, C. UML y Patrones. Introducción al análisis y diseño orientado a objetos. 1999.
8. MANUAL-JAVA. Utilizando el patrón Singleton Disponible en: <http://www.manual-java.com/codigos-java/utilizando-patron-singleton.html>.
9. NIURYS ÁLVAREZ DELGADO, L. G. G. SISTEMA DE GESTIÓN DE TIEMPO DE LOS ESTUDIANTES VINCULADOS AL PROYECTO PROCYON. 2007.
10. ONEL ROSELLÓ REYES, C. S. A. Sistema para el control de tiempo de producción en los proyectos productivos de la universidad de las Ciencias Informáticas. 2007.
11. P.LETELIER. Introducción a RUP. 2006, Disponible en: <https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20a%20RUP.doc>.
12. PMI, P. M. I. Guía de los Fundamentos de la Dirección de Proyectos, Tercera Edición. 2004, n<sup>o</sup>
13. PRESSMAN, R. Ingeniería del Software: un enfoque práctico. 2002.
14. PROGRAMACIÓN, L. D. Programación Java Disponible en: <http://www.lenguajes-de-programacion.com/programacion-java.shtml>.
15. PROGRAMASGRATIS. Time Tracker 4.4 Disponible en: <http://timetracker.programas-gratis.net/>.
16. RINCÓNDELVAGO. Trabajo de Auditoría “Herramientas Case” Disponible en: [http://html.rincondelvago.com/herramientas-case\\_2.html](http://html.rincondelvago.com/herramientas-case_2.html).
17. RÍOS, S. S. Análisis y UML. Modelo conceptual. 2007, Disponible en: [http://www.uvmsf.cl/~ssanchez/images/Metodologias/Unidad4\\_MAD.pdf](http://www.uvmsf.cl/~ssanchez/images/Metodologias/Unidad4_MAD.pdf).
18. SACRISTÁN, L. Comparativa entre J2EE, ASP.NET y PHP Disponible en: <http://sentidoweb.com/2007/07/24/comparativa-entre-j2ee-aspnet-y-php.php>.

## *Bibliografía*

---

19. SVENSSON, P. D. H. Incorporating the Personal Software Process into the Rational Unified Process. 2001.

## **Glosario de Términos**

### **A**

Applet (de java)

Componente de una aplicación que corre en el contexto de otro programa.

Artefactos

Pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades; representa un área de responsabilidad, y es candidata a ser tenida en cuenta para el control de la configuración. Un artefacto puede ser un modelo, un elemento de un modelo, o un documento.

### **B**

Bytecode

El bytecode es un código intermedio más abstracto que el código máquina. Habitualmente se lo trata como a un fichero binario que contiene un programa ejecutable similar a un módulo objeto, que es un fichero binario que contiene código máquina producido por el compilador.

### **C**

Código abierto

Término con el que se conoce al software distribuido y desarrollado libremente.

### **F**

Framework

En el desarrollo de software, un Framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, librerías y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

### **G**

GoF

Del acrónimo en inglés Gang of Four: Es el nombre con el que se conoce a los autores del libro Design Patterns, referencia en el campo del diseño orientado a objetos. La 'Banda de los cuatro' se compone de los autores: Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides.

### **H**

Hardware

Conjuntos de componentes que integran la parte material de una computadora. Componentes físicos de una tecnología.

HTML

## *Glosario de Términos*

---

El HTML, acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Explorer o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

### **HTTP**

(HyperText Transfer Protocol, protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la Web. El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceso a una página y la respuesta con el contenido.

### **I**

#### **IDE**

Un entorno de desarrollo integrado o en inglés Integrated Development Environment ('IDE') es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

#### **Interfaz de Usuario**

Parte del programa informático que permite el flujo de información entre el programa y el usuario u otro sistema.

### **J**

#### **JDBC**

Es el acrónimo de Java Database Connectivity, un API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema de operación donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.

### **L**

#### **Licencia GNU/GPL (General Public License)**

Es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software Libre.

### **N**

#### **.NET**

## *Glosario de Términos*

---

.NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma y que permita un rápido desarrollo de aplicaciones.

### **P**

#### Plataforma de Desarrollo

Es el entorno de software común en el cual se desenvuelve la programación de un grupo definido de aplicaciones. Comúnmente se encuentra relacionada directamente a un sistema operativo; sin embargo, también es posible encontrarla ligada a una familia de lenguajes de programación o a una Interfaz de programación de aplicaciones.

#### Plugin

Aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica.

#### Protocolo

Conjunto de reglas que controlan la secuencia de mensajes que ocurren durante una comunicación entre entidades que forman una red.

### **S**

#### Scripts

Un conjunto de comandos escritos en un lenguaje interpretado para automatizar ciertas tareas de aplicación.

#### Sistema Operativo

Conjunto de programas que se integran con el hardware para facilitar al usuario, el aprovechamiento de los recursos disponibles.

### **T**

#### TCP/IP

La familia de protocolos de Internet es un conjunto de protocolos de red que implementa la pila de protocolos en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras. En ocasiones se la denomina conjunto de protocolos TCP/IP, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP).

### **U**

#### URL

Uniform Resource Locator (localizador uniforme de recurso). Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

### **X**

## *Glosario de Términos*

---

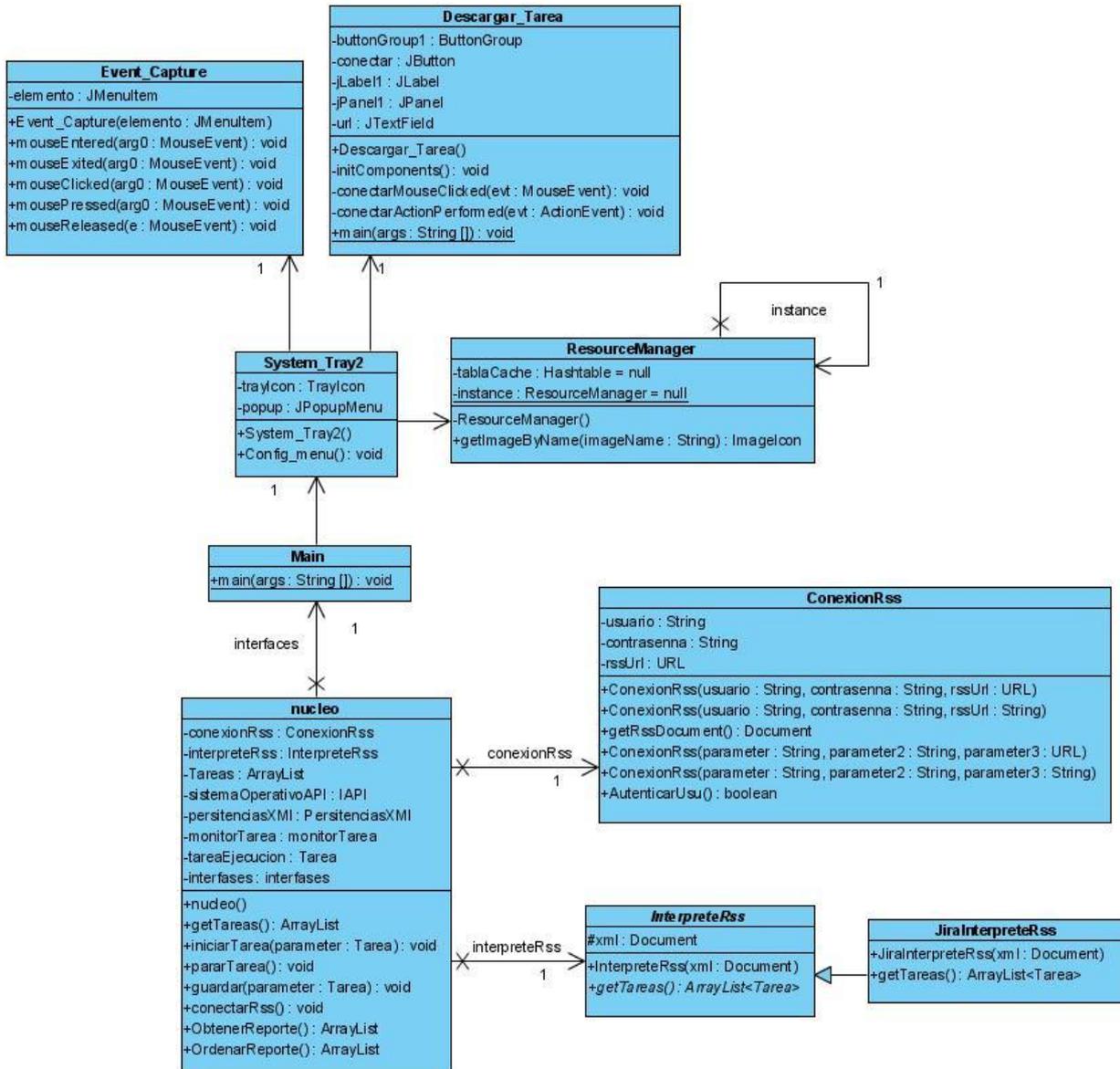
### XML

El XML es el acrónimo del inglés Extensible Markup Language (lenguaje de marcado ampliable o extensible) desarrollado por el World Wide Web Consortium (W3C). Es muy simple de gran utilidad en el intercambio de datos, ya que permite describirlos sin mostrarlos al usuario. Permite la lectura de datos a través de diferentes aplicaciones.

## Anexos

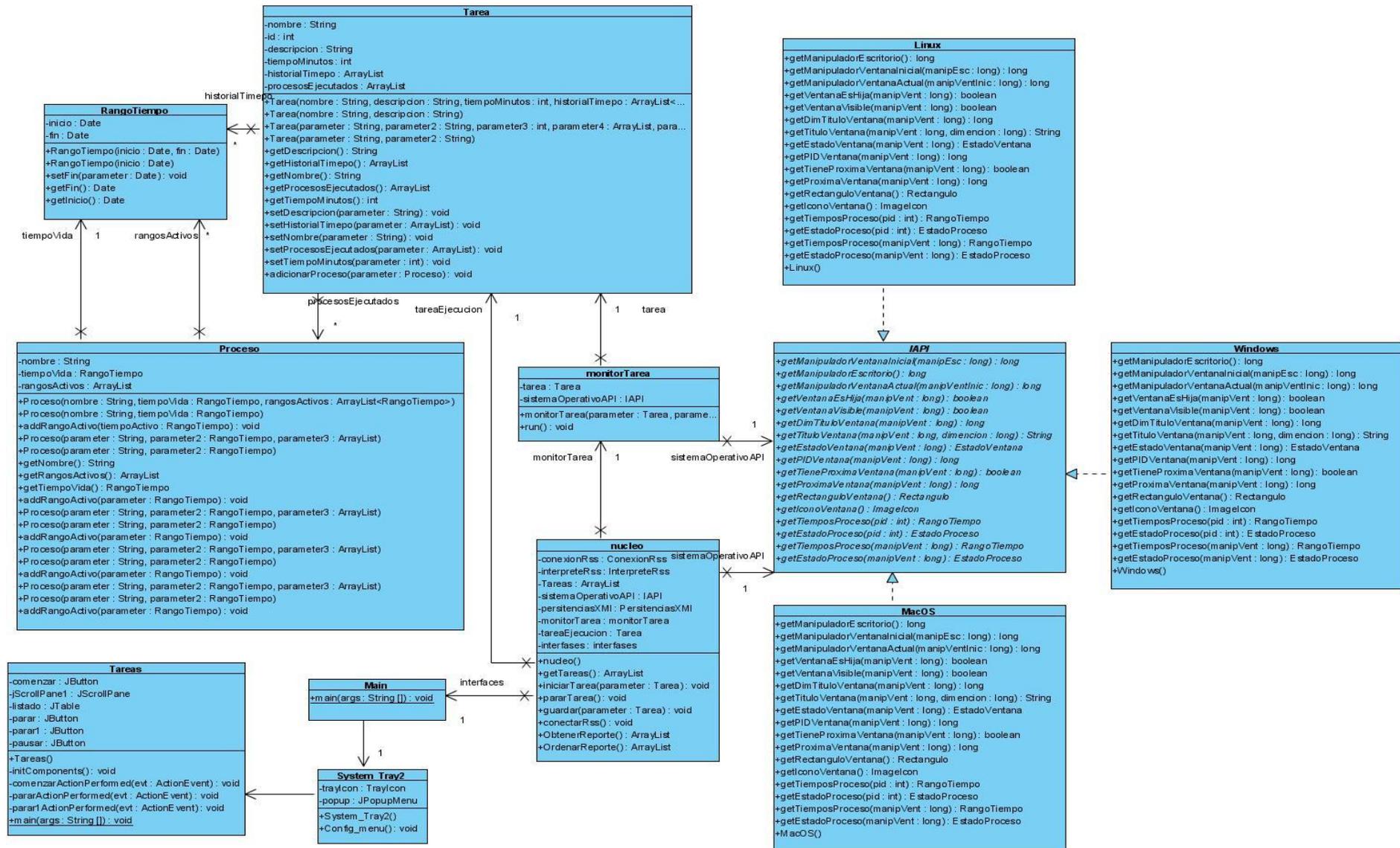
### Anexo 1 Diagramas de Clases del Diseño

#### DCD CU Obtener Tarea



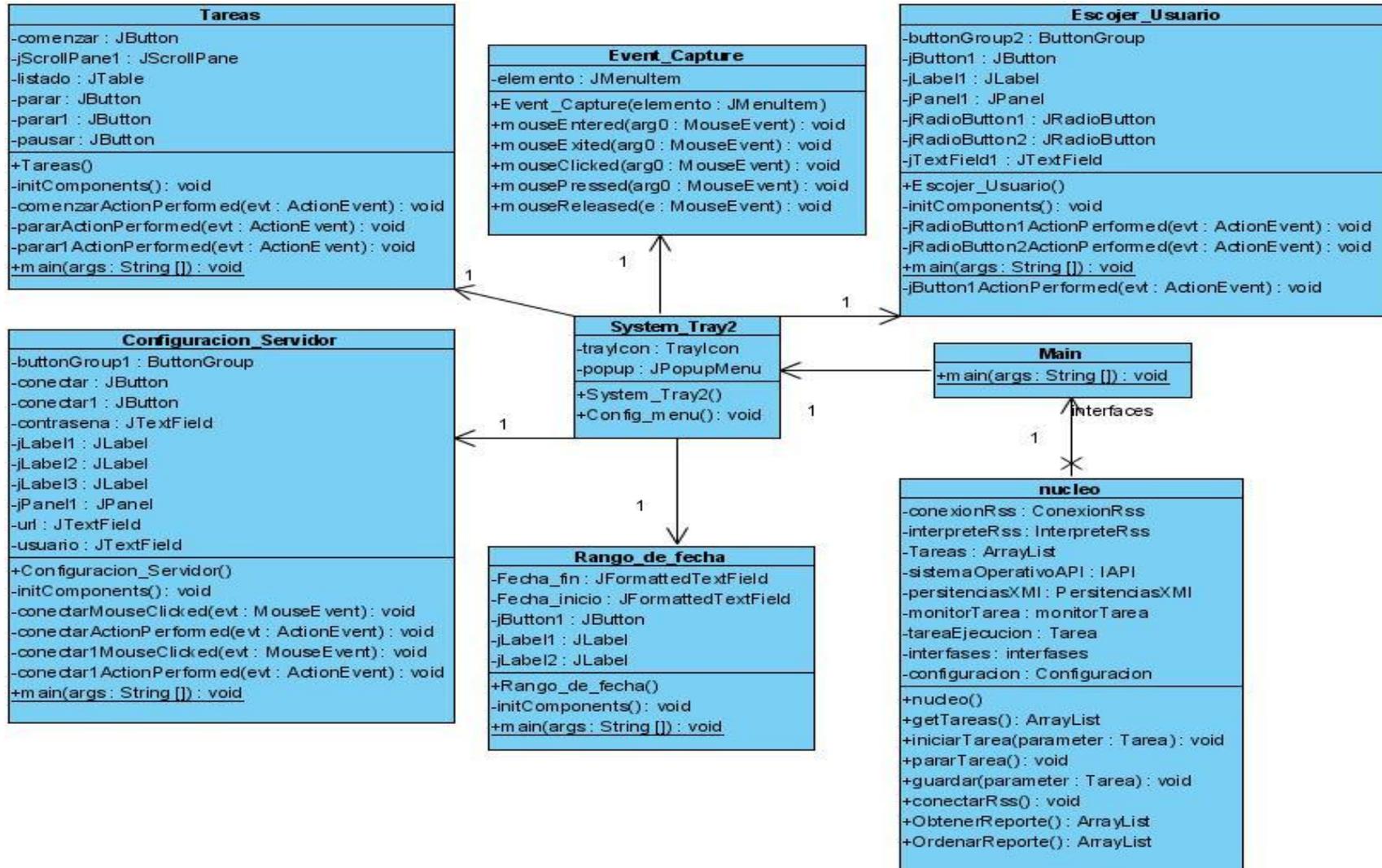
#### DCD CU Registrar software y uso

# Anexos





## DCD CU Obtener Reporte



## DCD CU Guardar información

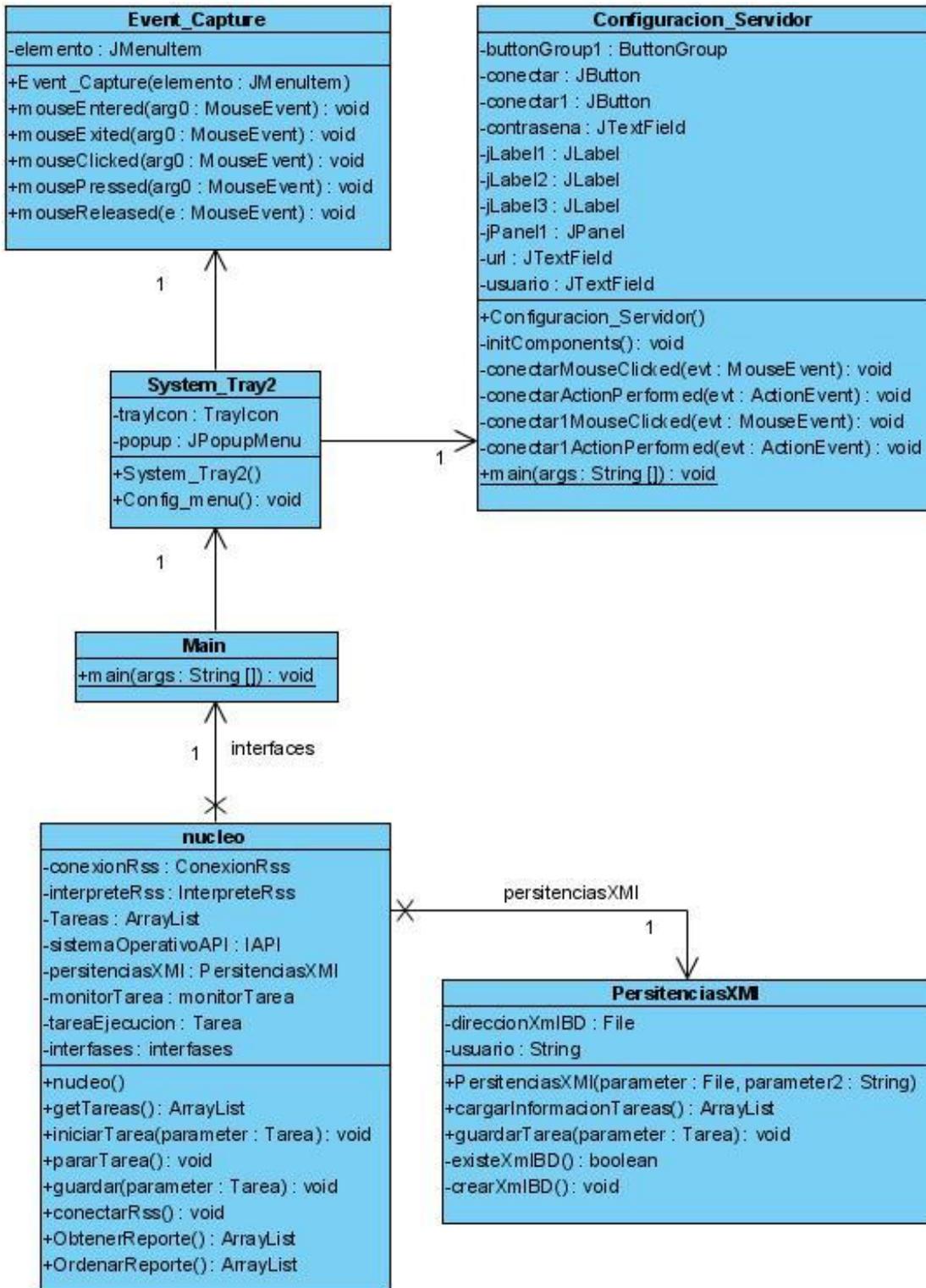
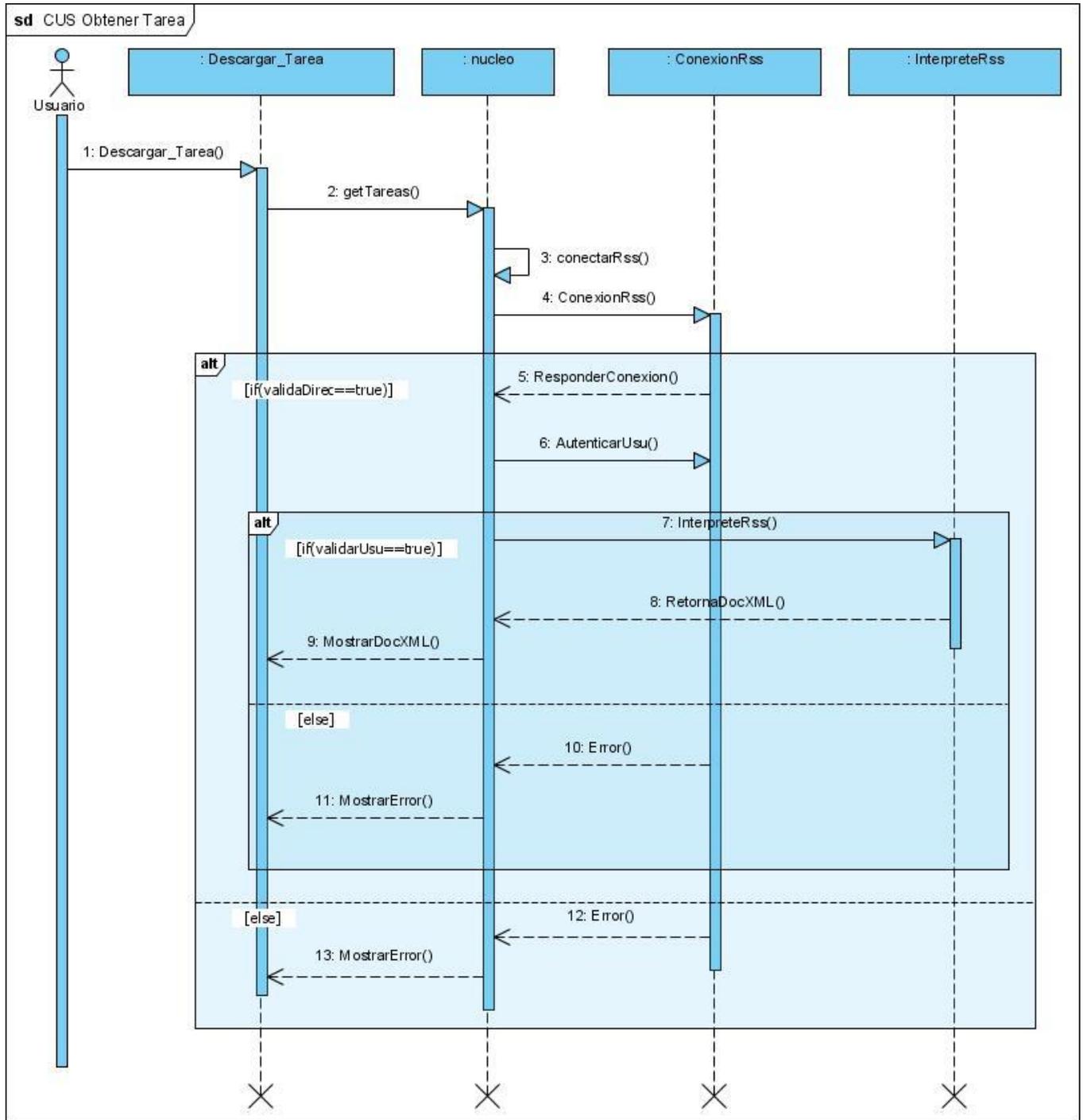


Diagrama de paquetes de clases del diseño



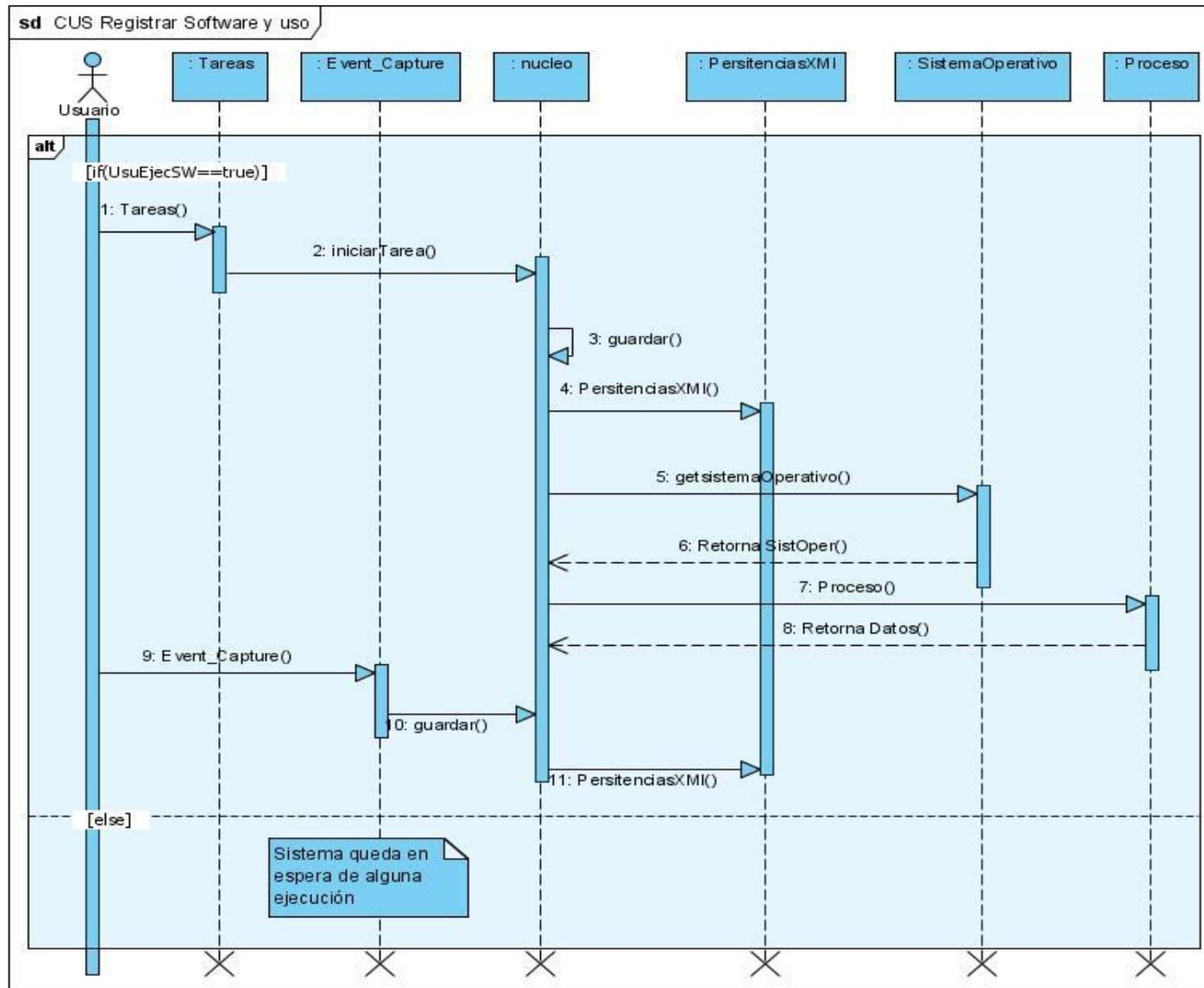
Anexo 2 Diagrama de Secuencias del Diseño

DSD CU Obtener Tarea

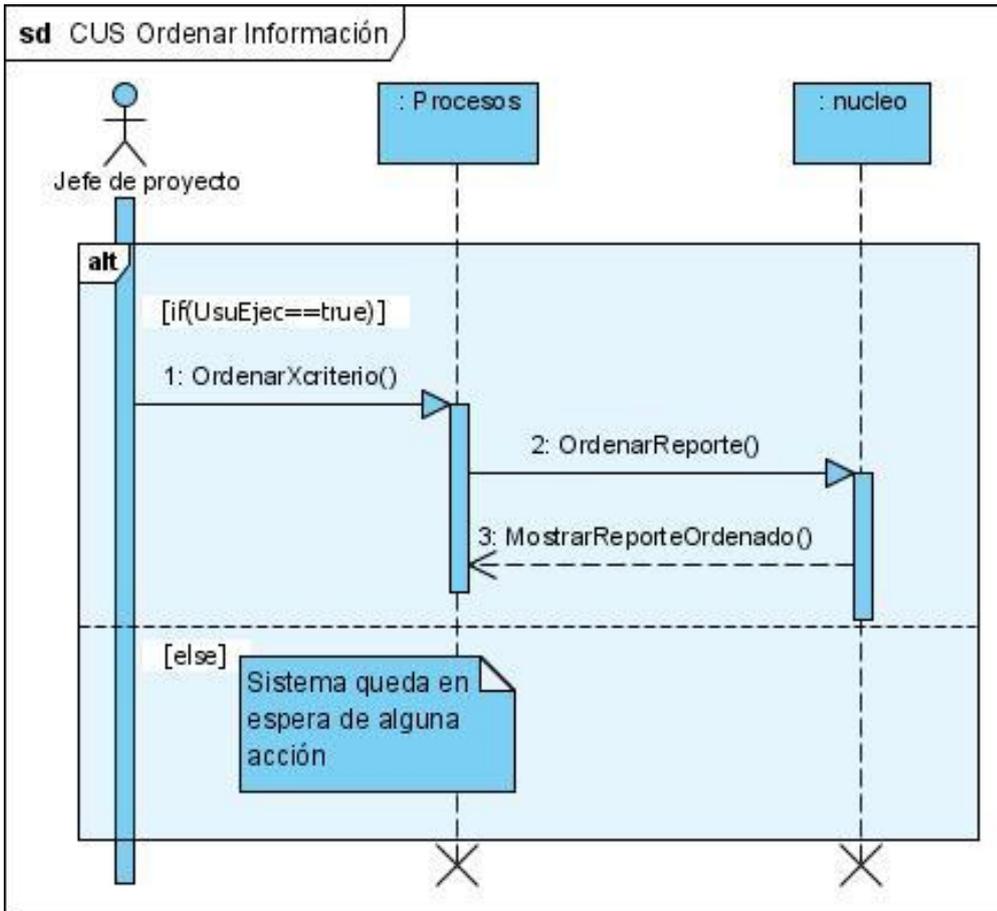


DSD CU Registrar software y uso

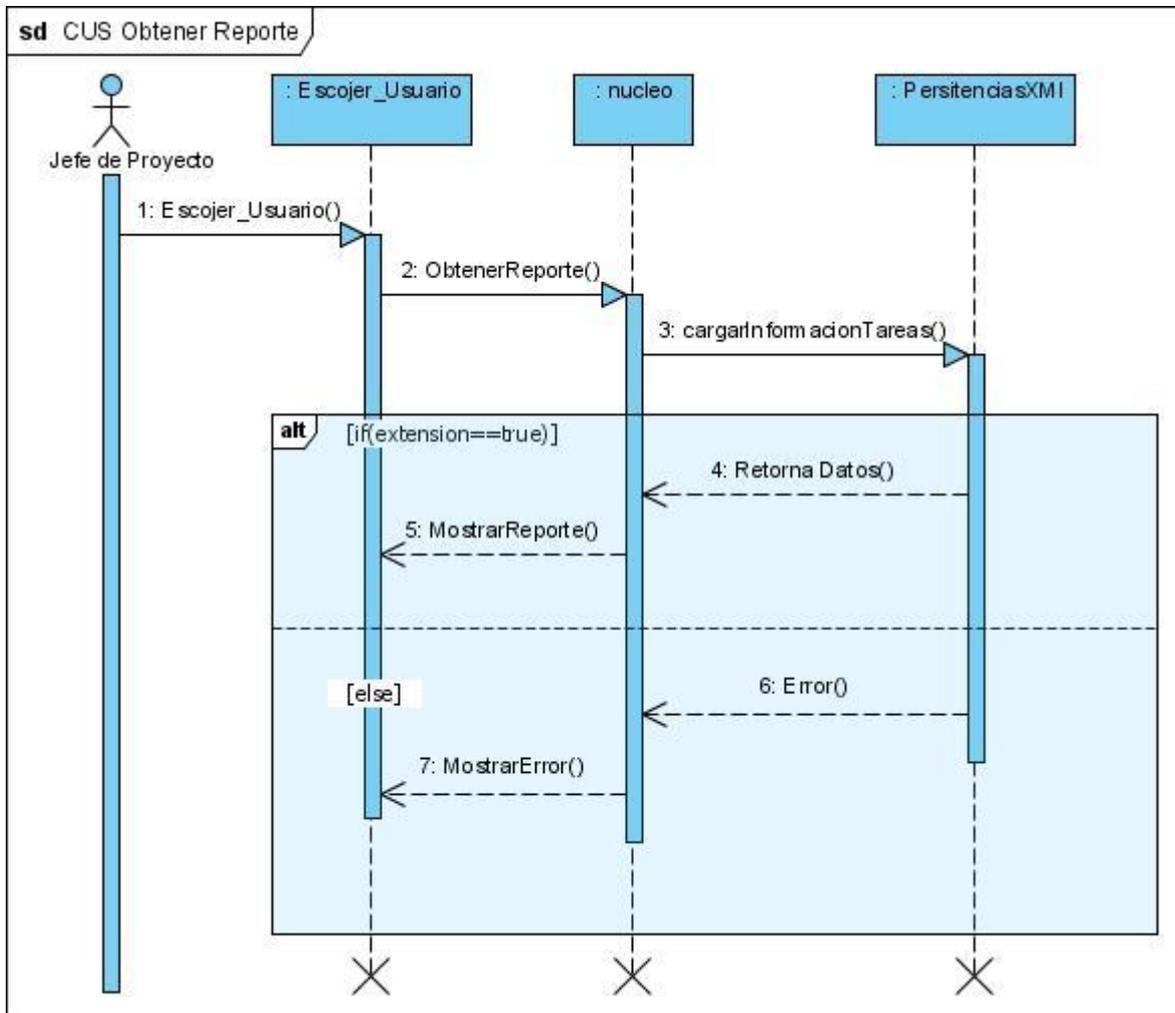
# Anexos



DSD CU Ordenar información



DSD CU Obtener Reporte



## DSD CU Guardar información

