



TRABAJO PARA OPTAR POR EL TÍTULO DE INGENIERÍA EN CIENCIAS  
INFORMÁTICAS

## **“SISTEMA DE GESTION DE TRABAJADORES”**

**AUTOR**

Heiner A. Dalby Martínez

**TUTOR**

Ms.C Héctor Rodríguez Figueredo

Ciudad de la Haba Julio 2008



## **DECLARACIÓN DE AUTORÍA**

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Heiner Dalby Martínez

---

## **DEDICATORIA**

Dedico este trabajo a mi familia, a mis amigos y a todas las personas que creen que toda meta es posible.

## **AGRADECIMIENTOS**

A mis amigos por soportarme todos estos años y porque se que me soportaran por muchos mas.

A mis padres por haber estado a mi lado en todo momento y siempre brindarme apoyo en cualquier decisión que tomara.

A la Revolución, por permitirme ser partícipe de este gran proyecto.

A mi tutor por todo el apoyo y la confianza depositada en mi.

## INDICE DE CONTENIDO

<b>RESUMEN</b> .....	<b>1</b>
<b>INTRODUCCIÓN</b> .....	<b>2</b>
<b>CAPITULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>6</b>
<b>INTRODUCCIÓN</b> .....	<b>6</b>
<b>GESTIÓN DE LA INFORMACIÓN</b> .....	<b>6</b>
CONCEPTOS BÁSICOS .....	6
<b>SISTEMAS DE INFORMACIÓN</b> .....	<b>7</b>
<b>SISTEMAS EXISTENTES DE GESTIÓN DE LA INFORMACIÓN</b> .....	<b>8</b>
EJEMPLOS DE SOFTWARE QUE REALIZAN FUNCIONES SIMILARES: .....	9
<b>METODOLOGÍA DE DESARROLLO</b> .....	<b>9</b>
METODOLOGÍA SELECCIONADA.....	14
<b>HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO DEL SISTEMA</b> .....	<b>17</b>
SISTEMA OPERATIVO .....	17
<b>LENGUAJES DE PROGRAMACIÓN</b> .....	<b>18</b>
LENGUAJE DE PROGRAMACIÓN WEB .....	18
LENGUAJE SELECCIONADO .....	18
LENGUAJE DE DISEÑO WEB.....	19
Lenguaje XML.....	19
<b>PROTOCOLOS</b> .....	<b>19</b>
PROTOCOLO SOAP (SIMPLE OBJECT ACCESS PROTOCOL) .....	19
SERVICIO WEB ( <i>WEB SERVICE</i> ).....	19
WSDL .....	20
UDDI .....	20
<b>CMS</b> .....	<b>20</b>
DEFINICIÓN .....	20
SISTEMAS OPEN SOURCE .....	22
CMS DRUPAL .....	23
<b>SERVIDOR APACHE</b> .....	<b>24</b>
<b>DISEÑO DE LA INTERFAZ QUANTA PLUS</b> .....	<b>25</b>
<b>LENGUAJE UNIFICADO DE MODELACIÓN</b> .....	<b>26</b>
HERRAMIENTA CASE .....	26
HERRAMIENTA CASE VISUAL PARADIGM .....	27
<b>CONCLUSIONES</b> .....	<b>27</b>



**CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA..... 28**

**INTRODUCCIÓN..... 28**  
**OBJETO DE ESTUDIO ..... 28**  
 SITUACIÓN PROBLÉMICA ..... 28  
 OBJETIVO DE AUTOMATIZACIÓN ..... 29  
 INFORMACIÓN QUE SE MANEJA..... 29  
**PROPUESTA DE SISTEMA ..... 29**  
**MODELO DEL NEGOCIO ..... 30**  
 ACTORES DEL NEGOCIO ..... 31  
 TRABAJADORES DEL NEGOCIO..... 31  
 DIAGRAMA DE CASO DE USO DEL NEGOCIO ..... 32  
 DESCRIPCIÓN DE CASO DE USOS DEL NEGOCIO ..... 33  
 DIAGRAMA DE ACTIVIDAD ..... 37  
 MODELO DE OBJETOS..... 41  
**ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE ..... 41**  
 REQUISITOS FUNCIONALES ..... 42  
 REQUISITOS NO FUNCIONALES ..... 42  
**MODELO DE CASOS DE USO DEL SISTEMA..... 43**  
 DEFINICIÓN DE LOS ACTORES Y CASOS DE USO DEL SISTEMA. .... 43  
 ACTORES DEL SISTEMA ..... 43  
 CASOS DE USO DEL SISTEMA ..... 43  
 DIAGRAMA DE CASOS DE USO DEL SISTEMA A AUTOMATIZAR ..... 54  
**CONCLUSIONES ..... 55**

**CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA..... 56**

**INTRODUCCIÓN..... 56**  
**ANÁLISIS..... 56**  
 MODELO DE ANÁLISIS ..... 56  
 MODELO DE CLASES DE ANÁLISIS ..... 57  
 DIAGRAMA DE CLASES DE ANÁLISIS ..... 57  
**MODELO DE DISEÑO ..... 60**  
 DIAGRAMA DE INTERACCIÓN..... 61  
 DIAGRAMA DE CLASES DEL DISEÑO WEB ..... 65  
**PATRONES DE DISEÑO DE DRUPAL ..... 69**  
**ARQUITECTURA CANDIDATA..... 71**  
**CONCLUSIONES ..... 72**

----- INDICE -----



<b>CONCLUSIONES .....</b>	<b>73</b>
<b>RECOMENDACIONES .....</b>	<b>74</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>75</b>
<b>BIBLIOGRAFÍA.....</b>	<b>77</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>78</b>
<b>ANEXOS .....</b>	<b>79</b>



## **RESUMEN**

En este trabajo se muestra un estudio del desarrollo de los principales elementos que conforman el Sistema de Gestión de Trabajadores de la UCI. Se hace un análisis de dichos elementos con el objetivo de lograr una organización estructural del Sistema de Acceso a Datos de los Trabajadores, expresada en la relación entre sus componentes, conectores y restricciones. Se analizan las decisiones arquitectónicas tomadas durante su desarrollo, definiendo las características fundamentales de las tecnologías y aspectos esenciales de diseño.

### **Palabras Claves**

Sistemas de gestión, acceso a la información.



## Introducción

Los aportes tecnológicos en el campo de la informática hacen que ésta se haya convertido en una herramienta imprescindible en el manejo de la información y los servicios. Por la gran utilidad que tienen, los sistemas de gestión de información son muy utilizados en entornos académicos, corporativos y empresariales, donde se manejan grandes volúmenes de información. En la Universidad de las Ciencias Informáticas (UCI), con más de 6000 trabajadores, se genera un cúmulo importante de información y por lo tanto se hace imprescindible contar con una herramienta útil para la dirección administrativa de la Universidad a la hora de gestionar los datos fundamentales de los trabajadores.

Este sistema permitirá un mejor uso de los recursos de información de la organización. En la UCI ya existen aplicaciones que recopilan, guardan y manipulan los datos de todo el personal, tanto docente como no docente. A pesar de que los datos del personal que trabaja en la UCI están automatizados, estos se encuentran separados en distintas aplicaciones y su acceso es muy limitado debido a la sensibilidad de los mismos. Por lo antes expuesto se hace necesario diseñar un software capaz de brindar a los trabajadores y directivos información relacionada con los datos recopilados en estas aplicaciones.

El manejo de estos datos es de importante relevancia, ya que el desconocimiento de los mismos causa problemas, tales como:

- Durante los primeros años de funcionamiento de la UCI el manejo de la información salarial no era un problema grave para los trabajadores ya que el sueldo se entregaba de forma personal. Con el objetivo de humanizar el trabajo la Universidad entregó tarjetas magnéticas para el pago salarial a los trabajadores. De esta forma al depositar directamente el salario en la tarjeta magnética los trabajadores carecían de información importante como el desglose de la nómina de pagos o las deducciones al salario, por citar algunos ejemplos.
- Si un directivo necesita información sobre un trabajador subordinado, por ejemplo: salario, categoría docente, grado científico, dirección particular, antigüedad, etc., no tiene una forma rápida y eficiente de hacerlo.



Inmersa en un amplio proceso de informatización, la UCI necesita crear un sistema eficiente que ayude al uso de las nuevas tecnologías que se van introduciendo en la universidad.

Lo antes expuesto llevó a identificar el siguiente **problema**:

¿Cómo optimizar el acceso a la información relacionada con el personal que labora en la universidad?

Para enfocar la investigación se trazó como **objetivo** diseñar una aplicación que, a partir de una interfaz web sea capaz de autenticar y permitir el acceso a los datos del personal de la UCI a través de la integración de servicios web.

De este **objetivo general** se derivan los siguientes **objetivos específicos**:

- Obtener una aplicación de fácil manejo para el usuario.
- Alcanzar la integración idónea entre los servicios web existentes en la UDDI.
- Lograr un nivel de acceso de acuerdo con la autenticación.

El **objeto de estudio** lo conforma el proceso de Gestión de Acceso a Datos de Trabajadores.

El **campo de acción** en el que se centrará la investigación es el desarrollo de un sistema que gestione el acceso de los trabajadores y directivos a los datos del personal de la UCI.

La incorporación de un modulo de gestión de acceso a la información relacionada con el personal laboral de la UCI optimizara el tiempo empleado por los trabajadores y directivos en la búsqueda de dicha información y agilizará el trabajo con los datos estratégicos de los directivos evitando procesos lentos y engorrosos es la **idea a defender** de este trabajo.

La investigación se desarrolló a través de las siguientes **tareas**:

1. Consultar la bibliografía científica disponible sobre sistemas de gestión para estimar el grado de novedad de los posibles resultados.
2. Definir las herramientas a utilizar para el diseño y construcción de la aplicación.
3. Definir la estrategia de desarrollo de la aplicación.
4. Análisis de las tecnologías que se utilizan para desarrollar Aplicaciones Web.

## 5. Desarrollo del modelo del problema real.

Para realizar las tareas se emplearon los siguientes **métodos**:

### **Métodos teóricos:**

- Análisis y Síntesis: para el procesamiento de la información y arribar a las conclusiones de la investigación, así como para precisar las características del modelo propuesto.
- Histórico - Lógico: Para determinar las tendencias actuales de desarrollo de los modelos y enfoques de la gestión.
- Inducción - Deducción: A partir del estudio de distintos modelos de gestión de información y sistemas específicos.
- Método Sistémico: Para determinar los componentes y definir las relaciones entre estos.

### **Métodos empíricos:**

- Observación: Para la percepción y determinación del problema.

El **aporte práctico esperado del trabajo** es el diseño de una aplicación que integre de manera eficiente los servicios web de la UDDI y permita un fácil acceso a los datos del personal de la UCI.

### **Este trabajo está estructurado en cuatro capítulos:**

#### **Capítulo 1:**

Este capítulo se titula **Fundamentación Teórica** e incluye una revisión del estado del arte de las tendencias, técnicas, tecnologías, metodologías y software usados en la gestión de la información a nivel internacional, nacional y de la Universidad. Este estudio no es una reproducción de referencias aunque se haga un uso extensivo de ellas, estas fueron analizadas y escogidas con sentido crítico.

#### **Capítulo 2:**

Este capítulo se titula **Características del Sistema** y en él se describen los procesos del negocio, realizándose también el análisis del sistema a desarrollar. Se acomete un estudio para capturar los requisitos funcionales y no funcionales que debe cumplir el

sistema, permitiendo la construcción del diagrama de Casos de Uso y la relación de los actores al interactuar con el sistema.

### **Capitulo 3:**

Este capitulo se titula **Análisis y Diseño del Sistema**. En el se abordan aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de clases de análisis, de diseño y se plantea el modelo de datos.

## Capítulo 1: Fundamentación Teórica

### Introducción

El propósito principal de este capítulo es brindar información acerca de los aspectos relacionados con la gestión de la información haciendo referencia a conceptos básicos del tema. Se aborda el estado actual en el ámbito internacional, nacional y en la Universidad de las Ciencias Informáticas del desarrollo de aplicaciones similares al sistema que se pretende desarrollar, así como un estudio de las herramientas a utilizar en nuestra aplicación.

### Gestión de la Información

#### Conceptos básicos

**Gestión:** Es el proceso mediante el cual se obtiene, despliega o utiliza una variedad de recursos básicos para apoyar los objetivos de la organización [1].

**Información:** Elemento fundamental que manejan los ordenadores en forma de datos binarios. Tras la revolución industrial, se habla de la revolución de la información, que se ha convertido en el mayor valor de las empresas y de las personas. El auge, proliferación y universalización de sistemas de interconexión global como Internet, ha llevado a hablar de la sociedad de la información como el nuevo paradigma del mundo en que vivimos. Factor cualitativo que designa la posición de un sistema, y que eventualmente es transmitido por este sistema a otro. [2]

**Acceso:** Operación de almacenamiento y recuperación de la información en la que el sistema accede a la memoria en base a un parámetro preestablecido. [3].

**Gestión de información:** Es todo lo relacionado con la obtención de la información adecuada, en la forma correcta, para la persona indicada, al costo adecuado, en el tiempo oportuno, en el lugar apropiado, para tomar la acción correcta. [4]

**Sistema de Información:** es la composición de elementos que operan unidos en orden de capturar, procesar, almacenar y distribuir información. Esta información se utiliza generalmente para tomar decisiones, la coordinación, el control y el análisis en una organización. Frecuentemente, el propósito básico del sistema es la gestión de la información.

Conjunto u ordenación de elementos organizados para llevar a cabo algún método, procedimiento o control mediante el proceso de información. [5].

**Sistema de Gestión de Información:** Es el conjunto de políticas y normas relacionadas entre sí que se establecen para el acceso y tratamiento de los recursos de información. Incluye los registros administrativos y los archivos, el soporte tecnológico de los recursos y el público a que se destina. [6]

## **Sistemas de Información**

Los sistemas de información:

- Gestionan los datos que almacenan.
- Gestionan los usuarios que utilizan la información, que además pueden agregarla.
- Poseen una interfaz en correspondencia con la información que contienen.
- Integran no sólo datos e información, sino también programas y otros sistemas de información.

Los sistemas de gestión de información para su distribución, deben seguir una representación cómoda para el usuario. El sistema debe poseer un registro de la producción de sus usuarios. En este entorno, se pueden integrar varios programas que ayudan al desarrollo del sistema de información, incluso puede un sistema de información ser la conjunción de varios sistemas que se integran en uno general, con un formato de presentación común que lo identifique, precisamente este es el tipo de sistema que se quiere desarrollar.

El punto de la interfaz es muy importante para la presentación ante el usuario, porque esta puede ayudar o dañar el contacto. Una interfaz amigable, discreta, fácil de usar y de navegación simple tiene muchas ventajas frente a una cargada de información mal delimitada, y confusa.

La información incluida en un sistema de información debe procesarse por sus responsables antes de su incorporación. Los encargados del funcionamiento del sistema deben recibir la información, así como verificar su procedencia y los programas utilizados para su confección, en aras de incorporarla para su posterior uso.

El hecho de que los sistemas de información automatizados dependen en gran medida de personas que procesan la información, autorizan su inclusión y de otros que la publican, comenzó a conspirar contra su efectividad y rápida evolución, porque, debido a estos procesos, la información se demoraba en estar a disposición de los usuarios finales, al tener que circular por varios especialistas; además, ocurría también que luego de su caducidad, continuaba a la vista en el sitio. Un sistema de información que pierde el hilo del desarrollo y el flujo de trabajo de la institución a la que pertenece, puede considerarse mediocre o inservible. [7]

## **Sistemas existentes de Gestión de la Información**

Hoy en día se manejan grandes volúmenes de información, lo cual ha posibilitado que los sistemas de gestión de la información tengan un auge impresionante. Existe un gran número de empresas que se dedican sólo a crear y diseñar este tipo de sistemas.

La información ha alcanzado un valor extraordinario y muchas empresas tratan de obtener beneficios de la necesidad que tienen otras compañías de automatizarse y proteger la información que generan, pero es incuestionable la necesidad de usar los avances tecnológicos para ser competitivo.

El desarrollo de la economía cubana está vinculado en los momentos actuales a los avances relacionados con las nuevas tecnologías de la información y las comunicaciones. La producción de *software* se convierte en una actividad con más y más demanda, lo que provoca que las empresas dedicadas a la producción de bienes y servicios informáticos tiendan a buscar las vías idóneas para el perfeccionamiento de la gestión empresarial donde la gestión de la información ejerce un rol importante.

En Cuba los procesos de gestión de la información se automatizan con el objetivo de organizar la información archivada en las bases de datos, haciéndola más útil. Un ejemplo de esto son los procesos de gestión desarrollados por Infomed para que el personal del sistema nacional de salud pueda brindar un mejor servicio a la población.



## **Ejemplos de software que realizan funciones similares:**

### **Tecno Doctor (Internacional)**

Permite al usuario conocer al detalle cualquier aspecto de su Historia Clínica, todo ello supervisado por los profesionales de su centro médico. Es un sistema de gestión multiplataforma, y se trata de un producto capaz de convivir con los sistemas existentes en el centro de trabajo, además de facilitar la integración con dichos sistemas gracias a servicios web ideados específicamente para ello.

### **Spec. S.A (Internacional)**

Es un ejemplo de empresa especializada en sistemas de gestión de acceso, control, presencia y de horario orientados. Sus principales productos son:

- Adonis X3: ERP de segunda generación para gestión integral de empresas.
- e-Logistic: Sistema de Gestión de Almacenes guiado por radio frecuencia
- e-Depots: Sistema de Gestión para empresas de entorno portuario.

### **SAIDO (Nacional)**

Es un sistema de información docente, uno de los primeros sistemas diseñados e implantados por el CECAM (Centro de Cibernética Aplicada a la Medicina). Actualmente está implantado al nivel de Facultad e Instituto en todo el país. [8]

### **AKADEMOS (UCI)**

Permite gestionar la información académica de los estudiantes en la Universidad de las Ciencias Informáticas. Está dividido por módulos y define niveles de acceso de tal manera que cada cual pueda visualizar solo aquella información relacionada con su función.

## **Metodología de Desarrollo**

Para el desarrollo de software se desarrollan estrategias que promueven el uso de prácticas adaptativas y su objetivo principal se centra en las personas o equipos de trabajo, orientadas sobre todo a la funcionalidad del sistema. La rama de la ingeniería de software que se encarga de este proceso es la metodología de desarrollo.

Es de relevante importancia la selección de la metodología, basada en el tiempo que se dispone, el tipo de proyecto, las herramientas que requiere.

No existe una metodología definitiva, cada una define a su manera el proceso de desarrollo ajustándose a características específicas de los proyectos que han sido utilizadas.

En algún momento se ha requerido hacer esta pregunta cuando se ha tenido que desarrollar un software. Y de hecho esta pregunta se torna muy importante, ya que se debe tener un apoyo.

Todo desarrollo de un software es riesgoso y difícil de controlar, pero si no se aplica la metodología adecuada lo que se obtiene es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos.

Sin embargo, muchas veces no se toma en cuenta la metodología adecuada, sobre todo cuando se trata de proyectos pequeños. En estos casos generalmente lo que se hace es separar rápidamente el aplicativo en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo.

Cuando los proyectos a desarrollar son de mayor envergadura, se debe basar en una metodología de desarrollo y buscar cuál sería la más apropiada para el caso. Lo cierto es que muchas veces no se encuentra la más adecuada y se termina por hacer o diseñar una propia metodología, algo que por supuesto no está mal, siempre y cuando cumpla con el objetivo propuesto.

Muchas veces se realiza el diseño del software de manera rígida, sólo con los requerimientos que el cliente solicitó. De manera que, cuando el cliente en la etapa final (etapa de prueba) solicita un cambio, se hace muy difícil realizarlo pues altera muchas cosas que no habíamos previsto. Este es uno de los factores que ocasiona atrasos en el proyecto, la incomodidad del desarrollador por no cumplir el cambio solicitado y el malestar del cliente por no tomar en cuenta su pedido. Obviamente para evitar estos incidentes se debe haber llegado a un acuerdo formal con el cliente al inicio del proyecto, de tal manera que cada cambio o modificación no perjudique al desarrollo del mismo.

Por experiencia, muchas veces los usuarios se dan cuenta de las cosas que dejaron de mencionar recién en la etapa final del proyecto, pese a que se les mostró un prototipo del software en la etapa inicial.

Los proyectos en problemas son los que se salen del presupuesto, tienen importantes retrasos, o simplemente no cumplen con las expectativas del cliente. Para dar una idea

de qué metodología se va a utilizar y cuál se adapta más a este sistema, se mencionan tres de las que se consideran más importantes: RUP, XP y MSF.

### **Extreme Programing (XP)**

Es una de las metodologías de desarrollo de software más exitosas en la actualidad y muy utilizada en proyectos de corto plazo, equipo limitado y cuya entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

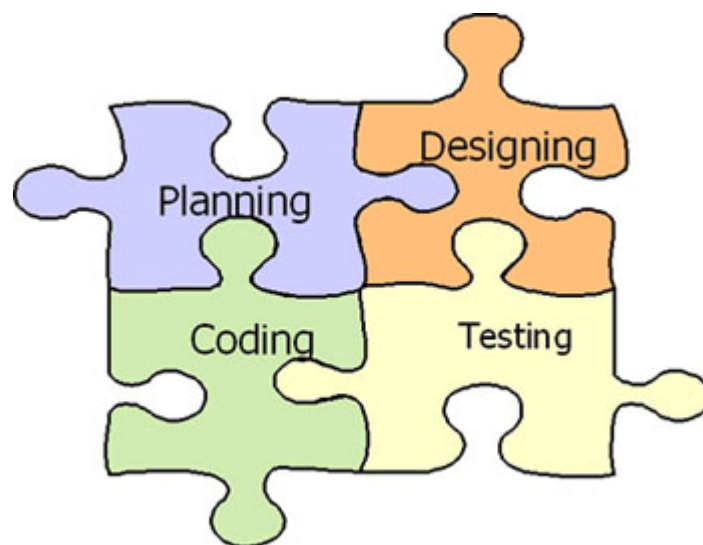


Figure 1 Estructura de XP

### **Características de XP**

La metodología se basa en:

- **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- **Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que

el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

### **¿Qué es lo que propone XP?**

- Empieza en pequeño y añade funcionalidad con retroalimentación continua
- El manejo del cambio se convierte en parte sustantiva del proceso
- El costo del cambio no depende de la fase o etapa
- No introduce funcionalidades antes que sean necesarias
- El cliente o el usuario se convierte en miembro del equipo

### **Derechos del Cliente**

- Decidir que se implementa
- Saber el estado real y el progreso del proyecto
- Añadir, cambiar o quitar requerimientos en cualquier momento
- Obtener lo máximo de cada semana de trabajo
- Obtener un sistema funcionando cada 3 o 4 meses

### **Derechos del Desarrollador**

- Decidir como se implementan los procesos
- Crear el sistema con la mejor calidad posible
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos
- Estimar el esfuerzo para implementar el sistema
- Cambiar los requerimientos en base a nuevos descubrimientos

### **Lo fundamental en este tipo de metodología es:**

- La comunicación entre los usuarios y los desarrolladores
- La simplicidad, al desarrollar y codificar los módulos del sistema
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales
-

## Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.



Figure 2 Estructura de MSF

MSF tiene las siguientes características:

- **Adaptable:** es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un lugar específico.
- **Escalable:** puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas o más.
- **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- **Tecnología Agnóstica:** porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

- **Modelo de Arquitectura del Proyecto:** Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.
- **Modelo de Equipo:** Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.

- **Modelo de Proceso:** Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de equipo.
- **Modelo de Gestión del Riesgo:** Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.
- **Modelo de Diseño del Proceso:** Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.
- **Modelo de Aplicación:** Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores.

## **Metodología seleccionada**

### **Metodología de Desarrollo RUP**

Se decide utilizar como metodología el Proceso Unificado de Modelado (RUP) por las ventajas de organización que brinda y debido a que goza de un grupo de características que hacen más dinámico el desarrollo del trabajo. Es una de las metodologías más generales y más usadas de las que existen en la actualidad, pues está pensada para adaptarse a cualquier proyecto.

Como RUP es un proceso, en su modelación define como sus principales elementos: Trabajadores que definen “el quién”, Actividades representan el “cómo”, Artefactos productos tangibles del proyecto “el qué”, y Flujo de actividades que precisan la secuencia de actividades “el cuándo”.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales y se divide además en 4 fases: Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. Estas etapas revelan que para producir una versión del producto en desarrollo se emplean todas las actividades de ingeniería. Además contempla flujos de trabajo de soporte que involucran actividades de planificación de recursos humanos tecnológicos y financieros.

Es necesario decir que el ciclo de vida de RUP se caracteriza por ser:

1. **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos.
2. **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.
3. **Iterativo e Incremental:** Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

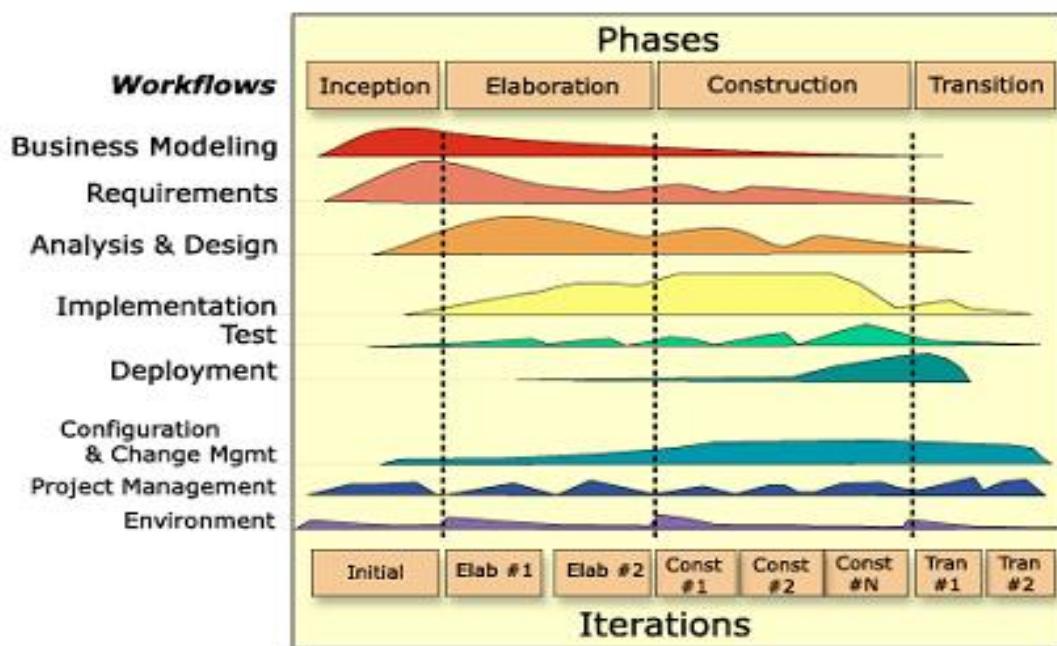


Figure 3 Estructura de RUP

## Fases de RUP

- **Inicio:** En esta etapa se describe el negocio y se determinan los límites del proyecto. Se identifican todas las entidades externas con las que interactuará el sistema y se define esta interacción a un nivel profundo. Se identifican todos los casos de uso y son descriptos los de mayor importancia.
- **Elaboración:** Se analiza el problema, definiendo la línea base de la arquitectura, es elaborado el plan del proyecto, son definidos los requerimientos (funcionales y no funcionales)
- **Construcción:** Se obtiene un producto documentado, listo para su uso y que posee un manual de usuario. Se producen además varias versiones del producto que son probadas por un reducido número de usuarios.
- **Transición:** Se obtiene un producto final listo para su uso y se le hace llegar a la comunidad de usuarios. Conlleva además a la identificación y corrección de errores.

## Flujos de Trabajo

1. **Modelación del Negocio:** Se describen los casos de usos del negocio como forma de documentar los procesos del sistema. Los casos de uso del negocio permiten identificar además a los participantes, así como las actividades que deben ser automatizadas.
2. **Requerimientos:** Durante este flujo se analizan las necesidades del negocio para poder llegar a una idea de lo que debe realizar el sistema, llegándose a un acuerdo entre desarrolladores y clientes. Son documentadas las funcionalidades del sistema y sus limitaciones.
3. **Análisis y Diseño:** En este flujo de trabajo son modelados un modelo de diseño y un modelo de análisis. Se representa una abstracción de cómo será el sistema teniendo en cuenta los requisitos impuestos en el flujo anterior.
4. **Implementación:** Este flujo de trabajo esta orientado a la creación del software prestando atención alas funcionalidades descritas y solicitadas. Genera una versión del producto.



5. **Prueba:** Garantizar que el sistema cumple con los requerimientos, que se ha realizado una correcta implementación y detectar los posibles errores antes de que el sistema sea entregado a los usuarios finales.
6. **Instalación:** El objetivo de este flujo es entregar el producto al cliente. Consiste también en brindar asesoría sobre su instalación y uso a los usuarios finales, así como brindar un servicio de soporte.

Los siguientes flujos aparecen durante todo el proceso de creación del software:

- **Administración de proyecto:**

Se vela por el cumplimiento de las fechas de entrega y los horarios de trabajo.

- **Administración de Configuración y Cambios:**

Establece un control sobre los cambios que va presentando el producto.

- **Entorno:**

Ayuda con la selección, configuración y uso de las herramientas y métodos a utilizar durante el proyecto.

## **Herramientas utilizadas para el desarrollo del sistema.**

### **Sistema Operativo**

#### **Linux**

Sistema operativo derivado de UNIX que, manteniendo casi todas las ventajas que este último ofrece, puede ser ejecutado en computadoras personales. Fue desarrollado originalmente por el estudiante finlandés de informática Linus Torvalds, que publicó su código fuente en 1990, en la forma de código abierto. Este hecho, unido a la estructura modular del sistema operativo (basado en la integración de componentes de software independientes) generó una nueva visión de desarrollo informático y ha permitido que Linux se haya expandido notablemente, gracias al trabajo, muchas veces voluntario y sin ánimo de lucro, de miles de programadores a todo lo largo del mundo. Actualmente están disponibles varias distribuciones de Linux, ofertadas por diversos proveedores, como RedHat, SuSE o Mandrake Inc.

## **Lenguajes de programación**

### **Lenguaje de programación web**

Debido a las tendencias y necesidades de las plataformas existen han ido surgiendo diferentes lenguajes de programación para desarrollar en la web.

Estos lenguajes son categorizados en dos grupos:

- **Lenguajes del Lado del Cliente:**

Ejemplo de ellos son el Java Script y el Visual Basic Script. Estos lenguajes son dinámicos ya que el código se ejecuta en el navegador antes de ser enviado al servidor, su uso sin embargo se restringe principalmente a la validación de datos.

- **Lenguajes del Lado del Servidor:**

Ejemplo de estos lenguajes son el PHP y ASP. Son lenguajes que se ejecutan en el servidor, permitiendo que se desarrolle la lógica de negocio, permiten un mejor trabajo con la base de datos.

### **Lenguaje seleccionado**

#### **Lenguaje de Programación web PHP**

PHP es un lenguaje interpretado de propósito general, ampliamente usado y que está diseñado para el desarrollo web. Este tiene la capacidad de ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en una gran cantidad de sitios web y de servidores dado por las importantes ventajas que ofrece. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

Hay que destacar también que PHP, es multiplataforma, funciona en todas las plataformas que soporten apache, además es software libre. Se puede obtener en la Web y su código esta disponible bajo la licencia GPL, también que viene acompañado por una biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, XML, creación de PDF). Permite las técnicas de Programación Orientada a Objetos, No requiere definición de tipos de variables y tiene manejo de excepciones.

## Lenguaje de diseño web

### Lenguaje XML

Es un subconjunto simplificado del SGML el cual fue diseñado principalmente para documentos Web. Deja a los diseñadores crear sus propias “etiquetas” o "tags" (Ej.: <libro>), habilitando la definición, transmisión, validación, y la interpretación de datos entre aplicaciones y entre organizaciones. Un punto que es importante aclarar, es que el HTML y el XML tienen funciones diferentes. El HTML tiene por objeto mostrar información, mientras que el XML se ocupa de la información propiamente dicha (el contenido). [9]

### Protocolos

#### Protocolo SOAP (Simple Object Access Protocol)

Es un protocolo de comunicación, el cual permite la comunicación entre aplicaciones a través de mensajes por medio de Internet. Es independiente de la plataforma, y del lenguaje. Está basado en XML y es la base principal de los servicios web. Los mensajes SOAP son documento XML propiamente dicho. [10]

#### Servicio web (*web service*)

La World Wide Web Consortium, W3C especifica en su documento Web Services Glossary, los servicios Web (Web Services) son sistemas software pensados para dar soporte a interacciones entre máquinas. Para que estas interacciones sean posibles todo servicio Web posee una interfaz descrita en un formato procesable por una máquina (utilizando el lenguaje WSDL, Web Service Description Language). En esta descripción se indica cómo otros sistemas pueden interactuar con el servicio Web: qué mensajes se intercambian, qué parámetros deben pasarse al servicio para que lleve a cabo su función, qué resultados se obtienen o qué fallos pueden producirse. Normalmente la invocación remota a un servicio Web se lleva a cabo utilizando mensajes SOAP, serializados utilizando XML y transmitidos mediante el protocolo HTTP (HyperText Transfer Protocol). Los Web Services son una tecnología adecuada cuando se pretenden desarrollar arquitecturas orientadas a servicio, SOA (Service Oriented Architecture). En estas arquitecturas distintos componentes interactúan intercambiando mensajes para conseguir llevar a cabo una tarea compleja. Gracias a la utilización de Web Services, que pueden

ser accedidos remotamente, componentes distintos pueden ejecutarse en distintas máquinas distribuidas por toda la red.

### **WSDL (Web Services Description Language)**

Es un protocolo basado en XML que describe los accesos al servicio web. Podríamos decir que es el manual de operación del servicio web, porque nos indica cuales son las interfaces que provee el Servicio web y los tipos de datos necesarios para la utilización del mismo.[11]

### **UDDI (Universal Discovery Description and Integration)**

Es un modelo de directorios para servicios web. Es una especificación para mantener directorios estandarizados de información acerca de los servicios web, sus capacidades, ubicación, y requerimientos en un formato reconocido universalmente. UDDI utiliza WSDL para describir las interfaces de los servicios web. Es un lugar en el cual podemos buscar cuales son los Servicios web disponibles, una especie de directorio en el cual podemos encontrar los servicios web. [12]

## **CMS**

### **Definición**

La idea básica que existe detrás de un sistema de gestión de contenidos es separar la gestión del contenido de su diseño. El diseño de las páginas se almacena en plantillas mientras que el contenido puede almacenarse en bases de datos o ficheros separados. Cuando un usuario solicita una página Web, las partes se combinan para producir una página HTML estándar. La página Web resultante puede incluir contenido de múltiples fuentes.

En este entorno, los usuarios crean sus páginas, con diseños específicos y los administradores de los sitios, en caso de que necesiten cambiar la apariencia del sitio, sólo deben trabajar en las plantillas, sin necesidad de alterar el contenido ni los diseños individuales.

Otro criterio es que “Un sistema de gestión de contenidos sirve para que la gestión de un sitio Web, por pequeño que sea, no se vaya de las manos: permite tener una apariencia y

navegación uniforme en todo el sitio, así como actualizar y gestionar su contenido fácilmente”.

Según el punto de vista de *Bob Boiko*, la gestión de contenidos es el proceso de tratar de emparejar lo que usted tiene con lo que ellos desean; donde usted es una organización con información y funcionalidad como valor y ellos son un sistema de audiencias definibles que desean ese valor. La gestión de contenidos puede parecer una manera de crear grandes sitios Web, pero en un examen más cercano, es en hecho, un proceso total para recoger, manejar y publicar contenido.

Las definiciones antes expuestas se podrían completar aún más si se les suma que los gestores de contenido proporcionan un entorno que posibilita la actualización, mantenimiento y ampliación del Web con la colaboración de múltiples usuarios.

En la definición de sistemas de gestión de contenidos, pueden encontrarse varios puntos, que si se analizan, coinciden con los que se mencionaron cuando se hablaba de sistemas de información: capturar, actualizar, integrar, consultar y analizar información pertinente para una institución (tabla1). Aunque en este sentido, los procesos en los sistemas de información engloban varios de los puntos de incidencia de los sistemas de gestión de contenidos.

Tabla 1. Integración de sistemas de información y sistemas de gestión de contenidos

<b>Sistemas de información</b>	<b>Sistemas de gestión de contenidos</b>
Capturar	Agrupación
Integrar	Transformación Catalogación
Analizar	Agregación Autorización
Actualizar	Presentación
Consultar	Distribución

Se puede constatar, además, que, en el caso de la gestión de contenidos, los procesos son mucho más explícitos y delimitados, y ello lleva a un mejor manejo y uso de la información.

Una de las características más importantes de los sistemas de gestión de contenidos es el hecho de que se pueda mantener el sistema, con la colaboración de múltiples usuarios, es decir, que siempre habrá elementos que consultar, sin esperar a que los especialistas publiquen la información, según sus posibilidades y con la prioridad que entiendan, cuando puede que no sea la requerida.

Un sistema de gestión de contenidos es la confluencia de la captación, procesamiento, actualización, representación y consulta de información en una institución, para la toma de decisiones en ambiente Web; estas tareas se realizan con la colaboración de todos los usuarios, y el sistema no depende sólo de información aislada, sino también de su dinámica, que tiene carácter continuo, que convierte el proceso en un ciclo productivo, resultante de su retroalimentación.

En resumen, los sistemas de gestión de contenidos se emplean para crear, procesar, compartir y organizar contenido en forma cooperativa entre los autores y el sistema, donde este último es capaz de ofrecer posibilidades de integrar los contenidos, con independencia de su formato o procedencia, así como guiar su flujo dentro del mismo, y ello ofrece la oportunidad de que las instituciones naveguen satisfactoriamente mediante la gestión de la información hacia el aprendizaje institucional, como vía invaluable en el cumplimiento de su misión y sus objetivos estratégicos, así como para la reducción de los costos de fricción de este proceso. Los costos de fricción comprenden los costos directos e indirectos asociados a la implementación y puesta en funcionamiento de un sistema o tecnología sea este informático o de otro tipo. [13]

## **Sistemas Open Source**

Cuando surgieron los sistemas de gestión de contenidos, si bien vinieron a solucionar varios problemas en el mundo de la gestión empresarial, también trajeron un problema consigo: los costos de las herramientas que posibilitaban su desarrollo. Al principio, había que pagar por obtener las herramientas; sin embargo, poco a poco se abrieron paso herramientas “open source”, hasta el punto que, en estos momentos, el mundo de las herramientas que soportan los sistemas de gestión de contenidos está prácticamente dividido entre las herramientas que se deben pagar y las de código fuente abierto. No obstante, los programas “open source” se ganan, cada vez más, la confianza de los consumidores; sus ventajas sobre los programas comerciales son ostensibles.

El movimiento “open source” engloba todos los aspectos necesarios para conducir el desarrollo abierto de las aplicaciones informáticas. Al disponerse del código fuente de un programa en la comunidad de técnicos, creadores de tecnologías y usuarios en general, es posible, mediante el trabajo colectivo, perfeccionar constantemente los programas, compartir soluciones y perfeccionar la herramienta; “open source” es una nueva filosofía de trabajo en el mundo tecnológico.

## **CMS Drupal**

Hay muchas características que hacen de Drupal un buen gestor de contenidos:

- Amplia comunidad de usuarios que lo utilizan y lo mantienen, lo que implica que el desarrollo de la aplicación tenga un gran dinamismo y esté mejorándose continuamente.
- Está basado en software libre: el código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros gestores de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.
- Entorno modular: al estar estructurado en módulos permite una gran flexibilidad, permitiendo crear desde el portal web de un Instituto hasta un sitio de blogs. Drupal tiene disponibles una gran cantidad de módulos que pueden descargarse de su página web e instalarlos en el sistema ampliando su funcionalidad.
- En la gestión de usuarios, además de autenticarse en local, los usuarios pueden hacerlo contra un sistema externo como Jabber, Blogger u otro sitio Drupal. Además tiene un completo conjunto de permisos y roles para que se pueda determinar fácilmente lo que puede hacer cada usuario.
- Objetos de Contenido (Nodos): Cualquier contenido creado en Drupal es, funcionalmente, un objeto (Nodo). Esto permite estandarizar la gestión, ya que todos tendrán una serie de propiedades comunes. Además incorpora un gestor de versiones que permite guardar un histórico de las actualizaciones que se realicen sobre el contenido.
- Múltiples herramientas de trabajo colaborativo: comentarios enlazados jerárquicamente, foros de discusión, encuestas, libros colaborativos, que permiten a muchos usuarios trabajar conjuntamente en la elaboración del contenido.

- Soporte de múltiples idiomas y localización: Drupal permite tener el contenido en varios idiomas, lo cual, para comunidades autónomas bilingües, es un requisito importante.
- El módulo de Registros y el de Estadísticas permiten realizar un seguimiento de la actividad, páginas más visitadas, usuarios conectados, popularidad del contenido, búsquedas realizadas, además de informar de todos los sucesos o incidencias que puedan ocurrir en el sistema.
- Drupal es multiplataforma desde sus inicios. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X.
- El código generado es muy limpio y cumple los estándares W3C de accesibilidad.

Todas estas características y muchas otras hacen de Drupal un buen gestor de contenidos que posee todo lo necesario para crear una página web de un centro permitiendo formar comunidades muy activas de estudiantes y profesores que colaboren en un proyecto común. [14]

## Servidor Apache

El **servidor HTTP Apache** es un software de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta entre otras características, mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Tiene amplia aceptación en la red, es uno de los servidores HTTP más usado. Alcanzó su máxima cota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo. La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales.

El servidor Apache es un software que esta estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

1. **Módulos Base:** Módulo con las funciones básicas del Apache
2. **Módulos Multiproceso:** son los responsables de la unión con los puertos de la máquina, acepando las peticiones y enviando a los hijos a atender a las peticiones



3. **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

### **Diseño de la interfaz Quanta Plus**

Quanta Plus o Quanta+ es una herramienta libre que se utiliza para el desarrollo y el diseño de Páginas web, impulsado principalmente por el proyecto KDE. Forma parte del paquete Kdewebdev.

Quanta+ es el símil para Linux de Dreamweaver para Windows, está creado para que se pueda migrar de manera muy rápida y casi instantánea. Quanta+ esta diseñado para ser eficiente, rápido y sencillo.

Alguna de las características que tiene Quanta+ son:

- Usa KIO para FTP y SSH, recordando que KIO es una API en la arquitectura KDE.
- Posee asistentes para la creación de tablas, enlaces y páginas web.
- Reconoce las sintaxis de HTML, Javascript, CSS, PHP y otros lenguajes más.
- Permite incorporar una gran cantidad de plugins para llevar a Quanta+ el máximo rendimiento posible e incorporando otras herramientas adicionales, control de versiones, editor de imágenes, etc.

Otra particularidad es la pre visualización de los documentos sin salir de la misma aplicación. [15]

## Lenguaje Unificado de Modelación

Se utiliza UML porque es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema, y que permite además la modelación de sistemas con tecnología orientada a objetos.

### Herramienta CASE

Estas herramientas consisten en un grupo de aplicaciones destinadas a mejorar la productividad en el desarrollo y mantenimiento del software. Ayudan además en el desarrollo de la documentación, generación de código y pruebas de errores, facilitando además el uso de las metodologías propias de la ingeniería del software.

De una manera breve se puede decir que una herramienta CASE esta compuesta por los siguientes elementos:

**Repositorio:** Es donde se almacenan los elementos definidos o creados por la herramienta, y cuya gestión se realiza mediante el apoyo de un Sistema de Gestión de Base de Datos (SGBD) o de un sistema de gestión de ficheros.

**Meta modelo:** Constituye el marco para la definición de las técnicas y metodologías soportadas por la herramienta.

**Carga o descarga de datos:** Son facilidades que permiten cargar el repertorio de la herramienta CASE con datos provenientes de otros sistemas, o bien generar a partir de la propia herramienta esquemas de base de datos o programas que pueden, a su vez, alimentar otros sistemas. Este elemento proporciona así un medio de comunicación con otras herramientas.

**Comprobación de errores:** Facilidades que permiten llevar a cabo un análisis de la exactitud, integridad y consistencia de los esquemas generados por la herramienta.

**Interfaz de usuario:** Consta de editores de texto y herramientas de diseño gráfico que permitan, mediante la utilización de un sistema de ventanas, iconos y menús, con la ayuda del ratón, definir los diagramas y matrices, que incluyen las distintas metodologías.

Las herramientas CASE se clasifican teniendo en cuenta los siguientes parámetros:

- Las plataformas que soportan.
- Las fases del ciclo de vida del desarrollo de sistemas que cubren.

- La arquitectura de las aplicaciones que producen.
- Su funcionalidad.

Las Herramientas CASE se clasifican por su amplitud en: TOOLKIT, WORKBENCH, además también se pueden dividir teniendo en cuenta las fases del ciclo de vida que automatizan en: UPPER CASE, MIDDLE CASE, LOWER CASE.

### **Herramienta CASE 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. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

[16]

### **Conclusiones**

En este capítulo se trataron conceptos y herramientas importantes relacionados con el tema con el objetivo de profundizar en el mismo. Se hizo un estudio de las aplicaciones existentes de gestión de información. Se realizó un estudio de las tecnologías a utilizar a lo largo del desarrollo del sistema. Se fundamentó el lenguaje de programación, el sistema manejador de contenido, la metodología de desarrollo del software que se va a utilizar. Todo con el propósito de que se entienda el por qué usar estas herramientas.

## **Capítulo 2. Características del Sistema.**

### **Introducción**

El principal propósito de este capítulo es describir como se realizan los procesos involucrados en la obtención de información relacionada con el salario, las vacaciones acumuladas, lugar de residencia, área de trabajo y si alguien ha causado baja en la Universidad. Se realiza el modelado del negocio, se definen los casos de uso y los actores que intervienen en el proceso. Por último se muestra el diagrama de casos de uso resultante y se mencionan los requerimientos funcionales y no funcionales del sistema.

### **Objeto de Estudio**

#### **Situación Problemática**

En la sociedad actual, la buena administración de la información es un recurso esencial para la formación de valores en las empresas e instituciones públicas. La documentación gestionada estructura la memoria de la organización y permite un mejor funcionamiento de la misma.

En la Universidad de las Ciencias Informáticas, la Dirección de Capital Humano tiene sistemas que gestionan y manipulan gran parte de la información de sus trabajadores como lo son el Registro de Identidad UCI y el Registro de Personal, por solo poner dos ejemplos. Pero si un trabajador necesita consultar toda la información que hay almacenada sobre su persona tendría que buscar en cada uno de estos sistemas y algunas informaciones como su salario y las vacaciones acumuladas no las encontraría ni tan siquiera en estos. El trabajador tendría que ir a las oficinas de el Departamento de Economía para que este le de la información salarial. Los trabajadores no tenían problemas con su información salarial antes de la implementación del pago a través de las tarjetas magnéticas, ya que al cobrar por caja tenían esta información.

Si algún directivo necesita obtener información específica de un trabajador o un grupo de estos tendría que realizar una búsqueda tortuosa de cada uno de ellos por los sistemas antes mencionados o ir a la Dirección de Capital Humano. Si, por ejemplo, deseara agregar o modificar un dato como el salario tendría que ir personalmente al Departamento de Economía, tramite que le tomaría determinado tiempo.

## **Objetivo de Automatización**

### Gestión de la información salarial y vacaciones acumuladas de los trabajadores

Se quiere automatizar el proceso que deben realizar los trabajadores para obtener su información salarial y las vacaciones acumuladas que tienen, desde que el sueldo de los mismos es depositado en las tarjetas magnéticas hasta que les es dada la información por un por el Departamento de Economía en sus oficinas.

### Agrupación de la información existente en otros sistemas de gestión de los trabajadores

Se desea automatizar la búsqueda a través de los sistemas de gestión existentes que tienen que realizar los trabajadores para poder obtener toda la información que hay recopilada acerca de ellos en la UCI.

### **Información que se maneja.**

En los dos procesos de automatización la información que se maneja esta relacionada con los datos de los trabajadores.

### Información salarial y vacaciones acumuladas de los trabajadores

Cuando se deposita el salario de los trabajadores este dato es guardado en las bases de datos de el Departamento de Economía y el sistema Assets calcula las vacaciones acumuladas y este dato es guardado en las mismas bases de datos que el salario.

### Información existente en otros sistemas de gestión

Cuando una persona es empleada por la UCI todos sus datos son registrados y almacenados por los sistemas de Registro de Identidad UCI y de Registro de Personal, este proceso es llevado a cabo por el personal de la Dirección de Capital Humano.

## **Propuesta de Sistema**

### Modulo para el Sistema de Gestión de la Información de los Trabajadores de la UCI

La aplicación es diseñada con el objetivo de proporcionar a los trabajadores y directivos de la Universidad de las Ciencias Informáticas una solución que permita de forma eficaz, cómoda y veloz acceder a la información relacionada con los datos de los mismos. En la Universidad se han desarrollado aplicaciones (Registro de Identidad UCI y Registro de Personal), que permiten gestionar los datos relacionados con los trabajadores. A pesar de esto a la Universidad le es necesario desarrollar un sistema capaz de gestionar y agrupar la información disponible de los trabajadores. Se propone la publicación a través de una aplicación web de la información brindada por el Sistema Assets, que sea capaz a través

de un proceso de autenticación de dar diferentes niveles de acceso a sus usuarios, dando la posibilidad a los directivos de obtener listados con la información de sus trabajadores. Esta aplicación debe ser capaz de agrupar la información relacionada con los trabajadores que se encuentra distribuida en diferentes sistemas. Esta aplicación sin duda tendría la ventaja de optimizar del tiempo a la hora de buscar información relacionada con los trabajadores.

## **Modelo del Negocio**

En la Universidad de las Ciencias Informáticas se controla un gran volumen de información relacionada con los datos concernientes a los trabajadores.

### Información salarial y vacaciones acumuladas de los trabajadores

La información salarial y de las vacaciones es tramitada por el Departamento de Economía desde el momento que la dirección de los departamentos docentes y no docentes le envían la información de sus trabajadores sobre el cumplimiento de las tareas que le corresponden. Luego el Departamento de Economía calcula el sueldo que le corresponde a cada trabajador y es depositado en las tarjetas magnéticas, con esta información se calculan las vacaciones acumuladas de cada trabajador, luego estos datos son almacenados en las bases de datos de Assets.

Si se necesita tener acceso a la información guarda en Assets es obligatorio ir a las oficinas del Departamento de Economía y el personal de ahí brindara la información a la cual puede tener acceso.

A través de la informatización de este proceso quedaría solucionado el problema ya que de forma rápida y eficiente se tendría acceso a esta información.

### Agrupación de la información existente relacionada con los trabajadores de la UCI

La mayoría de los datos relacionados a los trabajadores de la UCI son recogidos por la Dirección de Capital Humano en el sistema Assets, y almacenados por este El Registro de Personal y El Registro de Identidad de la UCI, pero en el directorio UCI solo se pueden encontrar la información mas general. Para tramitar una información mas detallada es necesario ir a la Dirección de Capital Humano y solicitar acceso a esta. Si esta información es necesaria para hacer un listado la persona encargada del mismo tendría que pedir autorización al director de Capital Humano y este pedir un informe con los datos deseados a su personal para que los obtenga a través de Assets.

Con la creación de una aplicación que sea capaz de agrupar estos datos usando los servicios web que están implementados por los Registros de Personal y de Identidad de la UCI este problema quedaría resuelto.

### Actores del Negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. [Conferencia 2 Modelación del Negocio asignatura Ingeniería de Software UCI]

**Tabla 2.1 Actores del Negocio**

Actores del Negocio	Descripción
Trabajador	Es la persona que puede solicitar la información tanto salarial como personal al departamento de Economía y a la Dirección de Capital Humano.
Directivo	Es la persona que puede solicitar la información tanto salarial como personal de un grupo de trabajadores al departamento de Economía y a la Dirección de Capital Humano.

### Trabajadores del Negocio

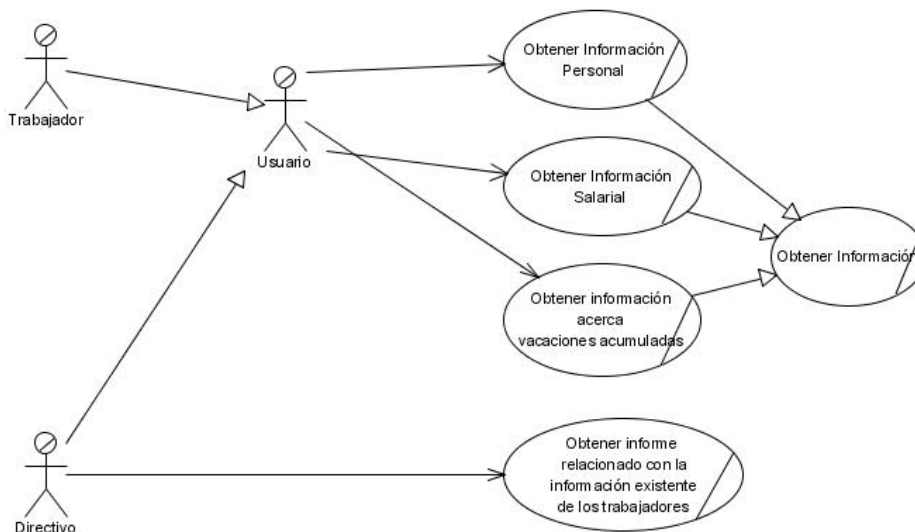
Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina, un sistema automatizado u organizaciones que están activamente implicadas en el negocio ya sea porque participan en él realizando una o varias actividades o porque sus intereses se ven afectados con los resultados del proyecto. Pueden ser los propietarios, la dirección, quienes financian, los clientes, los trabajadores, los proveedores, la comunidad local, etc.; que actúa en el negocio, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol. Se han detectado los siguientes trabajadores del negocio.

**Tabla 2.2 Trabajadores del Negocio**

Trabajadores del Negocio	Descripción
Departamento de Economía (Nómina)	Es el encargado de hacer todos los cálculos correspondientes al salario de los trabajadores e insertarlos en Assets. Se encarga también de atender a las personas que necesiten esa información.
Assets(Sistema)	Es el encargado de gestionar todo los datos relacionados con el personal de la UCI. También se encarga de archivar la información obtenida en los diferentes registros existentes. (Registros de Personal, de Identidad UCI y de Residentes)
Dirección de Capital Humano	Es el encargado de recopilar e introducir los datos de los trabajadores el los sistemas de Assets.

**Diagrama de Caso de Uso del Negocio**

Un diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio.



**Figure 4 Diagrama de Caso de Uso del Negocio**



## Descripción de Caso de Usos del Negocio

Un caso de uso del negocio representa a un proceso de negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio. Desde la perspectiva de un actor individual, define un flujo de trabajo completo que produce resultados deseables. La descripción de un caso de usos del negocio muestra cómo colaboran los trabajadores y entidades de negocio para ejecutar el proceso. Son reseñas textuales del caso de uso y explica los procesos o actividades que tienen lugar en el mismo.

**Tabla 2.3 Descripción del Caso de Uso Obtener información salarial**

Nombre de CUN		Obtener información salarial	
<b>Actores del negocio</b>		trabajador	
<b>Resumen:</b> El caso de uso se inicia cuando el trabajador desea conocer cual fue su salario. Este se presenta a la en la oficina de Nomina y comunica que desea conocer su información salarial, el personal de Nomina busca la información y se la muestra al trabajador. El caso de uso finaliza cuando el trabajador obtiene su información salarial.			
<b>Casos de Usos Asociados</b>		-	
<b>Flujo de Trabajo</b>			
<b>Acción del Actor</b>		<b>Respuesta del Negocio</b>	
1	El trabajador se presenta a la en la oficina de Nomina y solicita su información salarial.	1.1	El personal de Nomina pide solapín y carnet de identidad y verifica los datos
		1.2	El personal de Nomina busca la información en la aplicación Assets
		1.3	El personal de Nomina le entrega al trabajador la información encontrada
<b>Prioridad:</b>		<b>primario</b>	
<b>Mejoras:</b>		Se informatizara el proceso haciendo que este sea mas rápido y eficaz y eliminando carga de trabajo al personal de Nomina.	
<b>Cursos alternos:</b> 1.1 El personal de Nomina considera que el trabajador no debe tener acceso a la información.			

**Tabla 2.4 Descripción del Caso de Uso Obtener información personal**

Nombre de CUN		Obtener información personal	
Actores del negocio		trabajador	
<b>Resumen:</b> El caso de se inicia cuando el trabajador quiere obtener la información personal que esta recopilada en las bases de datos de la UCI y a demás de los datos existentes en el directorio necesita otros datos personales. El trabajador tiene que presentarse en la Dirección de Capital Humano y solicitar la información. El caso de uso termina cuando le es entregada la información al trabajador.			
<b>Casos de Usos Asociados</b>		<b>secundario</b>	
<b>Flujo de Trabajo</b>			
Acción del Actor		Respuesta del Negocio	
1	El trabajador se autentica	1.1	El sistema LA verifica que el trabajador tiene acceso al Directorio
		1.2	El sistema le permite acceder
2	El trabajador introduce sus datos	2.1	El Directorio busca la información que tiene con los datos introducidos por el trabajador
		2.2	El Directorio muestra la información obtenida al trabajador
3	El trabajador quiere obtener información adicional		
3.1	El trabajador la hace una petición a la Dirección de Capital Humano	3.2	El director de Capital Humano determina si tiene acceso a la información.
		3.3	El director pide que se le entregue la información al trabajador
		3.4	El personal de Capital Humano realiza una búsqueda en la aplicación Assets.
		3.5	El personal de Capital Humano le entrega al trabajador la información encontrada.
<b>Prioridad:</b>		-	
<b>Mejoras:</b>		Se facilitará el proceso evitándole al trabajador tantos tramites y eliminando carga de trabajo al personal de Capital Humano.	
<b>Cursos alternos:</b> 1.2 El sistema LA considera que el trabajador no debe tener acceso a la información. 3.3 El Director de Capital Humano determina que el trabajador no debe tener acceso a la información.			

**Tabla 2.5 Descripción del Caso de Uso Obtener acerca vacaciones acumuladas**

<b>Nombre de CUN</b>		<b>Obtener información acerca vacaciones acumuladas</b>	
<b>Actores del negocio</b>		Trabajador	
<p><b>Resumen:</b> El caso de uso se inicia cuando el trabajador desea conocer cuantos días de vacaciones tiene acumulado. Este se presenta a la en la oficina de Nomina y comunica que desea conocer la información relacionada con sus vacaciones, el personal de Nomina busca la información y se la muestra al trabajador. El caso de uso finaliza cuando el trabajador obtiene la información relacionada con sus vacaciones.</p>			
<b>Casos de Usos Asociados</b>		<b>Obtener información salarial</b>	
<b>Flujo de Trabajo</b>			
<b>Acción del Actor</b>		<b>Respuesta del Negocio</b>	
<b>1</b>	El trabajador se presenta a la en la oficina de Economía y solicita su información acerca de las vacaciones acumuladas.	<b>1.1</b>	El personal de Nomina pide solapín y carnet de identidad y verifica los datos
		<b>1.2</b>	El personal de Nomina busca la información en la aplicación Assets. Esta se calcula a través de la información salarial.
		<b>1.3</b>	El personal de Nomina le entrega al trabajador la información encontrada
<b>Prioridad:</b>		<b>secundario</b>	
<b>Mejoras:</b>		Se informatizara el proceso haciendo que este sea mas rápido y eficaz y eliminando carga de trabajo al personal de Nomina.	
<p><b>Cursos alternos:</b> 1.1 El personal de Nomina considera que el trabajador no debe tener acceso a la información.</p>			

**Tabla 2.6 Descripción del Caso de Uso Obtener informe relacionado con la información existente de los trabajadores.**

Nombre de CUN		Obtener informe relacionado con la información existente de los trabajadores	
<b>Actores del negocio</b>		Directivo	
<b>Resumen:</b> El caso de uso inicia cuando un directivo necesita obtener un informe con la total o parcial información existente relacionadas con sus trabajadores. Este tiene que hacerle una petición al director de la Dirección de Capital Humano. El director le pide a su personal que realice un informe y se lo envíe al directivo. El caso de uso termina cuando el directivo obtiene el informe.			
<b>Casos de Usos Asociados</b>		-	
<b>Flujo de Trabajo</b>			
Acción del Actor		Respuesta del Negocio	
1	El directivo realiza la petición de un informe con la información relacionada con los trabajadores de su área.	1.1	El director de Capital Humano determina a que nivel de información tiene acceso el directivo.
		1.2	El director de Capital Humano pide al personal de su oficina que se le entregue un informe con la información especificada al directivo.
		1.3	El personal de Capital Humano realiza una búsqueda de la información pedida en el sistema Assets.
		1.4	La aplicación Assets conforma un informe con la información encontrada.
		1.5	El personal de Capital Humano envía en informe al directivo.
<b>Prioridad:</b>		<b>primario</b>	
<b>Mejoras:</b>		Se pretende informatizar este proceso para hacer lo más eficiente y eliminarle carga de trabajo a la Dirección de Capital Humano.	
<b>Cursos alternos:</b> 1.2 El director de Capital Humano determina que el directivo no tiene acceso a la información que esta solicitando.			

## Diagrama de Actividad

El diagrama de actividad es un grafo que contiene estados en que puede hallarse una actividad. Un estado de actividad representa la ejecución de una sentencia de un procedimiento, o el funcionamiento de una actividad en un flujo de trabajo. En vez de esperar un evento, como en un estado de espera normal, un estado de actividad espera la terminación de su cómputo. Cuando la actividad termina, entonces la ejecución procede al siguiente estado de actividad dentro del grafo. Una transición de terminación es activada en un diagrama de actividades cuando se completa la actividad precedente.

Un diagrama de actividades puede contener bifurcaciones, así como divisiones de control en hilos concurrentes. Los hilos concurrentes representan actividades que se pueden realizar concurrentemente por los diversos objetos o personas en una organización. La concurrencia se presenta con frecuencia a partir de la agregación, en la cual cada objeto tiene su propio hilo concurrente. Las actividades concurrentes se pueden realizar simultáneamente o en cualquier orden. Un grafo de actividades es como un organigrama tradicional, excepto que permite el control de concurrencia además de control secuencial: una gran diferencia.

Un diagrama de actividad describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio. Es similar a un diagrama de estados en el cual todos o la mayoría de los estados son estados *de* actividad y en la cual todas o la mayoría de las transiciones se disparan al completarse las acciones en los estados fuentes precedentes.

Diagramas de Actividad

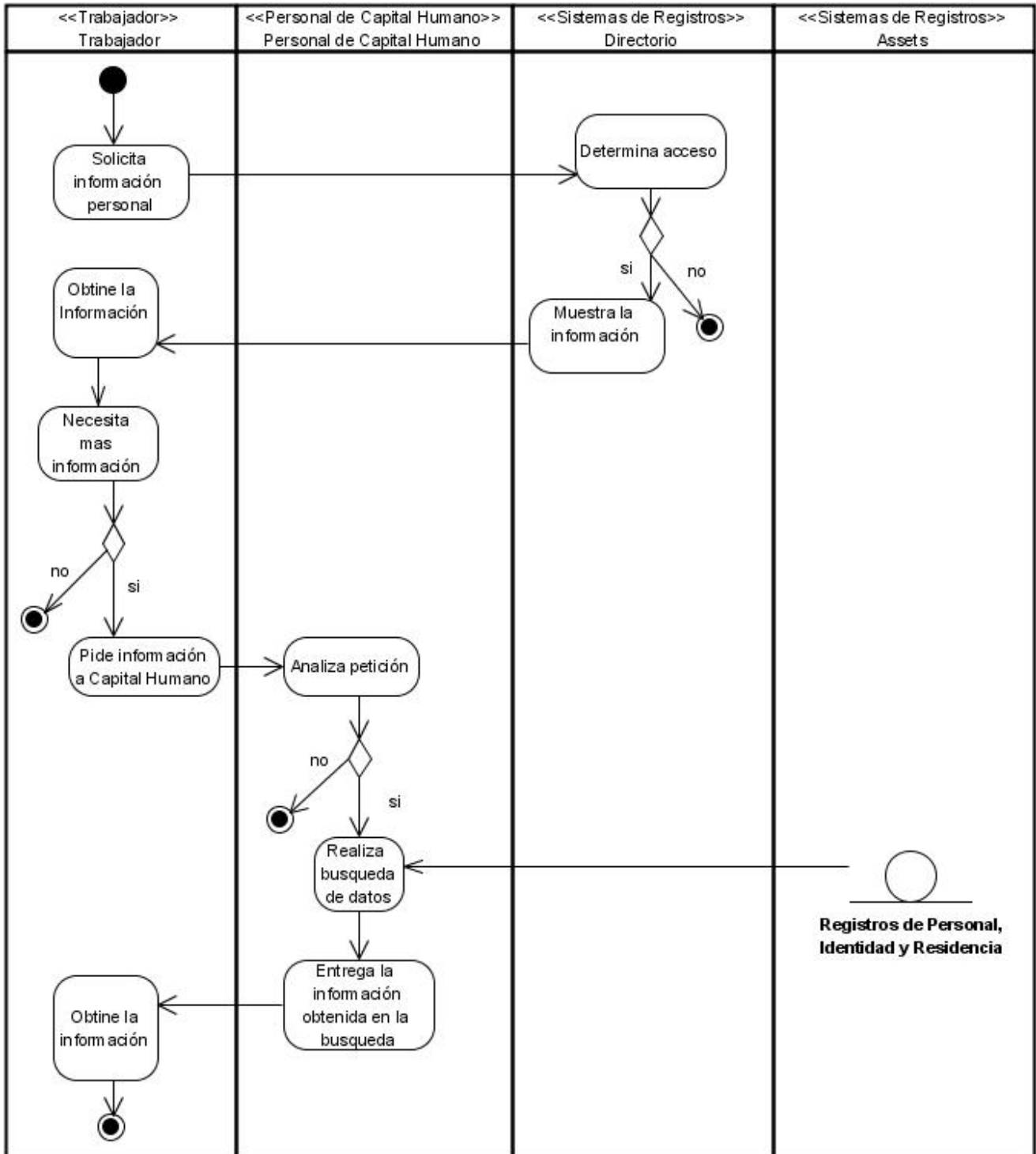


Figure 5 Diagrama de Actividad del CUN Obtener Información Personal

-----Capítulo 2 -----

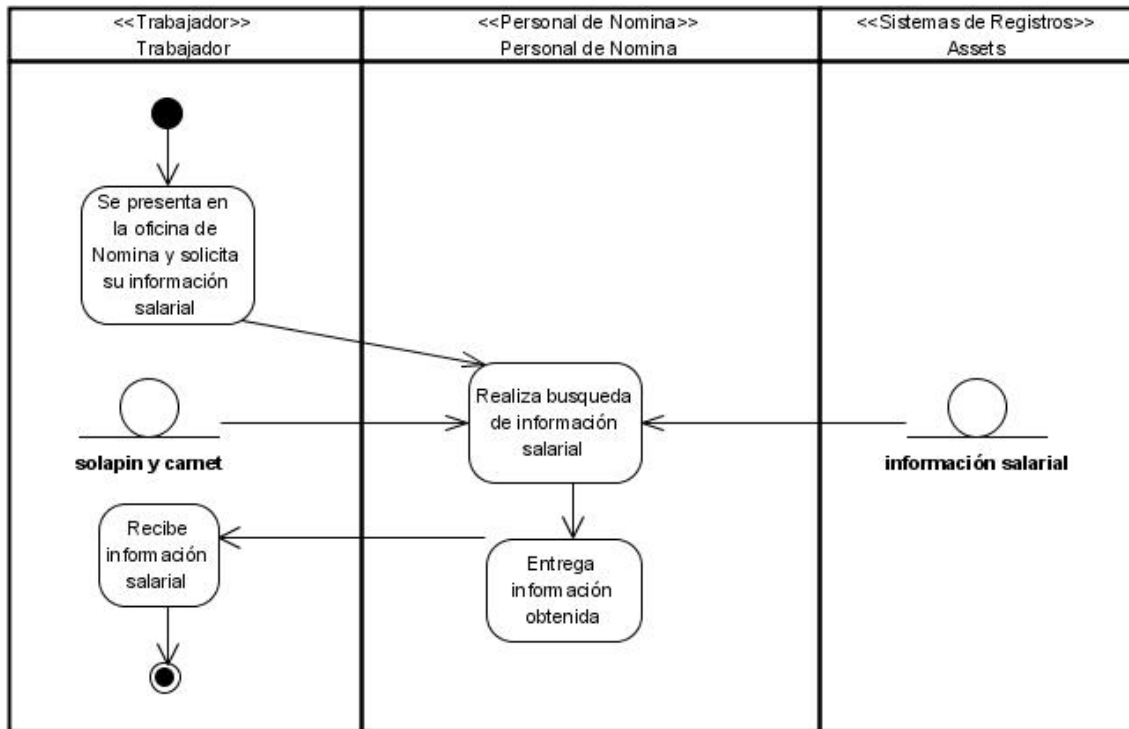


Figure 6 Diagrama de Actividad del CUN Obtener Información Salarial

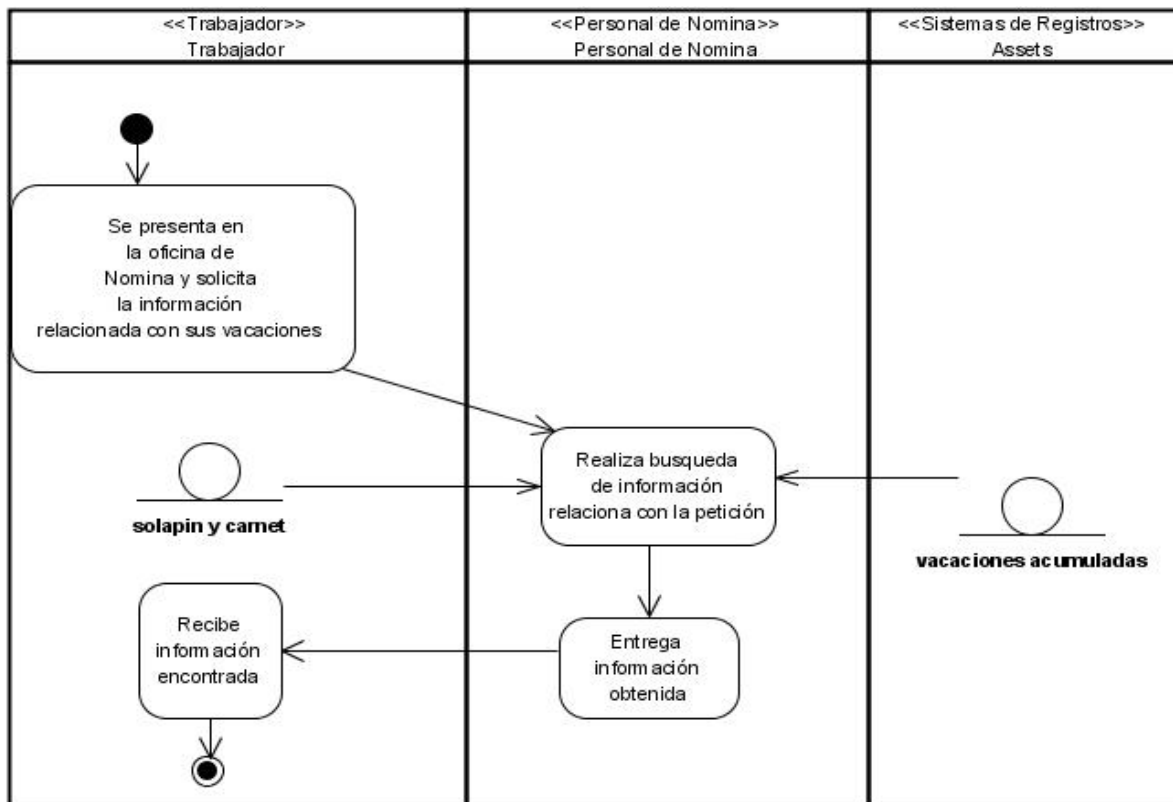


Figure 7 Diagrama de Actividad del CUN Obtener Información acerca de vacaciones acumuladas

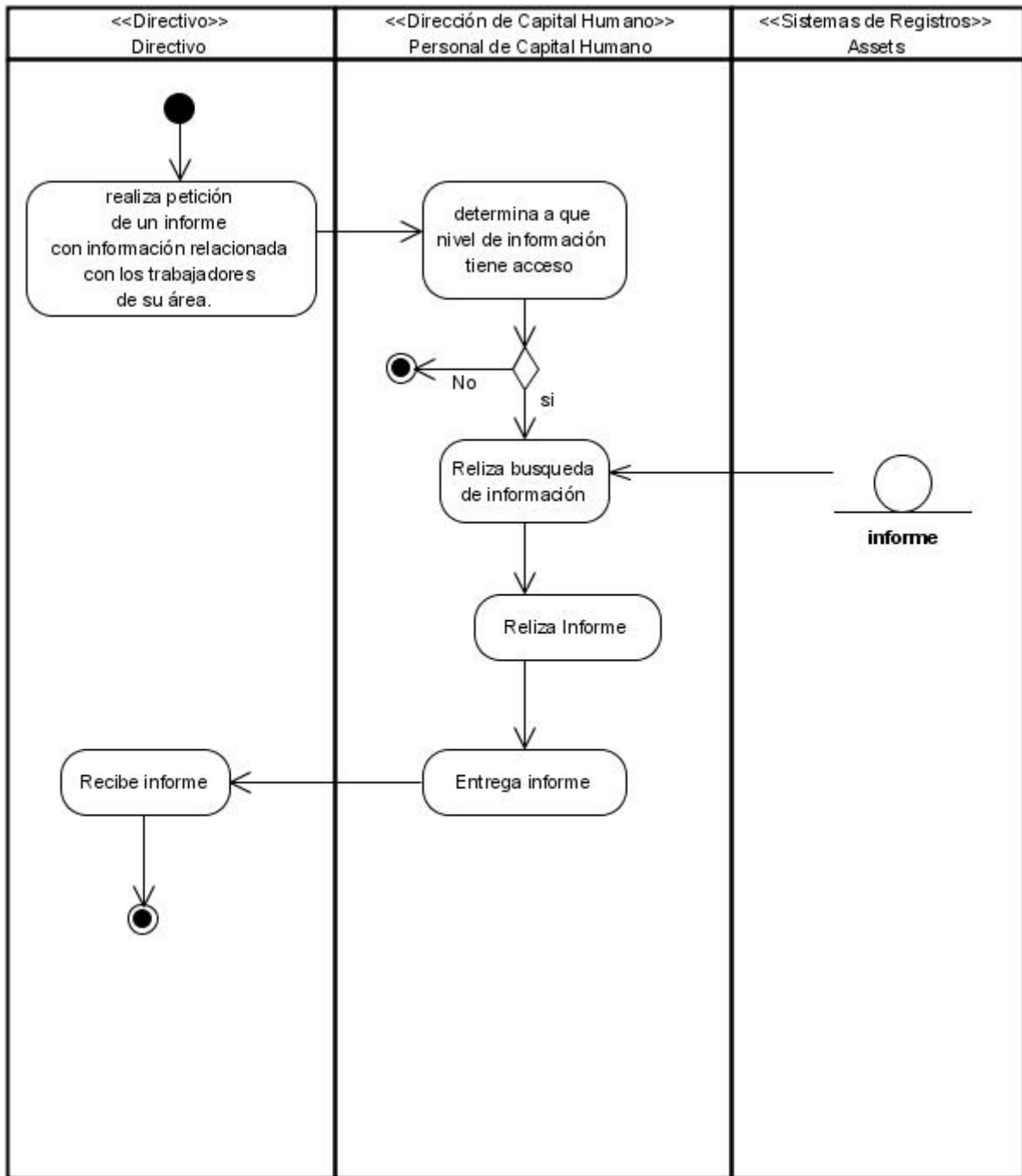


Figure 8 Diagrama de Actividad del CUN Obtener informe relacionado con la información existente de los trabajadores



## Modelo de Objetos

Representan la relación entre los trabajadores del negocio y las diferentes entidades del mismo.

El diagrama de clases, como artefacto que se construye para describir el modelo de objetos del negocio, muestra la participación de los trabajadores y entidades del negocio y la relación entre ellos. Aunque se puede construir un único diagrama, se recomienda confeccionarlo para cada caso de uso de negocio para una mejor claridad.

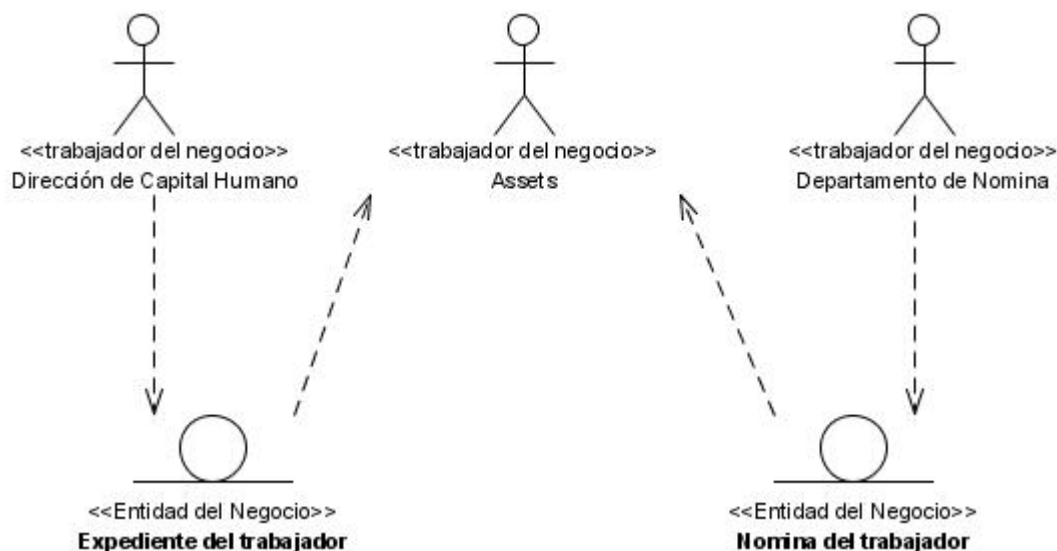


Figure 9 Modelo de Objetos

## Especificación de los Requisitos de Software

El reconocimiento de los requerimientos funcionales y no funcionales del futuro sistema es uno de los procesos más importantes en la fase de inicio. Este proceso tiene el objetivo de lograr una mejor comprensión, entendimiento y modelado del negocio y la premisa fundamental de obtener un resultado eficaz y que sea agradable al cliente.

## Requisitos Funcionales

El sistema debe cumplir con ciertas condiciones o capacidades, esos son los requerimientos funcionales.

R1- Controlar que cada usuario acceda a la información que le corresponda.

R1.1- Autenticar y autorizar a los usuarios.

R2-Obtener la información relacionada con trabajador

R2.1-Obtener la información salarial de un trabajador.

R2.2-Obtener las vacaciones acumuladas de un trabajador.

R2.3-Obtener el lugar de residencia de un trabajador.

R2.4-Obtener la categoría de un trabajador.

R3-Realizar informe con la información de los trabajadores.

R3.2- Obtener un listado del personal dada un área, los cumpleaños dada una fecha.

R3.3.-Obtener un listado de categorías del personal de un área dado un nivel de acceso.

R3.4- Obtener un listado del lugar de residencia del personal de un área dado un nivel de acceso.

R3.5- Obtener un listado con la información salarial del personal de un área dado un nivel de acceso.

R3.6- Obtener un listado con la información de las vacaciones acumuladas del personal de un área dado un nivel de acceso.

## Requisitos no Funcionales

Los requerimientos no funcionales responden a cualidades que el producto debe tener y las características para que este sea atractivo, confiable, usable y seguro. Son los aspectos visibles para el usuario del sistema, que no están relacionados de forma directa con el comportamiento funcional del sistema.

Restricciones en el diseño y la implementación. El sistema será implementado en el lenguaje de programación Web PHP.

Portabilidad. Será un sistema multiplataforma.

Seguridad. Debe garantizar que la información vista únicamente por quien tenga permiso para esto. El sistema debe permitir solo el acceso a cada uno de sus servicios al personal requerido para esto.

Confiabilidad. Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.

Disponibilidad. El sistema debe permitirle al usuario conectarse desde cualquier PC de la red.

## Modelo de Casos de Uso del Sistema

### Definición de los actores y casos de uso del sistema.

Se han definido los actores y casos de uso del sistema teniendo en cuenta que un caso de uso representa una secuencia de acciones que producen un resultado observable para ciertos actores y que un actor representa cualquier cosa que interactúe con el sistema.

### Actores del Sistema

Tabla 2.7 Actores del Sistema

Actores del Sistema	Justificación
<b>Trabajador</b>	Son los usuarios que interactuar con el sistema para hacer solicitudes relacionadas con información personal.
<b>Directivo</b>	Son los usuarios que interactuar con el sistema para hacer solicitudes de listados relacionadas con información del personal de su área.

### Casos de Uso del Sistema

Seguidamente se muestran los casos de uso seleccionados para satisfacer los requerimientos funcionales de sistema:

**Tabla 2.8 CUS Autenticarse**

<b>CU-1</b>	<b>Autenticarse</b>
<b>Actor</b>	Trabajador, Directivo
<b>Descripción:</b>	Los usuarios introducen sus credenciales del dominio para que el sistema los verifique y poder acceder a las funcionalidades correspondientes a su rol.
<b>Referencia:</b>	RN 1,1.1

**Tabla 2.9 CUS Generar informe de un trabajador**

<b>CU-2</b>	<b>Generar informe de un trabajador</b>
<b>Actor</b>	Trabajador
<b>Descripción:</b>	Los usuarios acceden a todo la información que esta registrado en la UCI relacionada con ellos.
<b>Referencia:</b>	RN 2,2.1,2.2,2.3,2.4

**Tabla 2.10 CUS Generar listado de cumpleaños**

<b>CU-3</b>	<b>Generar listado</b>
<b>Actor</b>	Directivo
<b>Descripción:</b>	Los usuarios acceden a una interfaz que les permite elegir un listado de todas las personas de su área dada una determinada característica.

<b>Referencia:</b>	RN 3.1,3.2,3.3,3.4,3.5
--------------------	------------------------

**Tabla 2.11 CUS Generar informe general.**

<b>CU-4</b>	<b>Generar informe general</b>
<b>Actor</b>	Directivo
<b>Descripción:</b>	Los usuarios acceden a un listado con la toda información relacionada con las personas de su área.
<b>Referencia:</b>	RN3,3.2,3.3,3.4, 3.5

**Descripción expandida de los Casos de uso del sistema.**

**Tabla 2.12 Descripción del CUS Autenticar**

<b>Caso de Uso:</b>	<b>Autenticarse</b>
<b>Actores:</b>	Trabajador, Directivo(inicia)
<b>Propósito:</b>	Permite registrar los usuarios al sistema, otorgar privilegios y niveles de acceso.
<b>Resumen:</b> El caso de uso se inicia cuando un usuario se registra en el sistema, si los datos (nombre de usuario y contraseña), son correctos se le permite el acceso al sistema.	
<b>Referencias:</b> R1,R1.1	
<b>Precondiciones:</b>	

<b>Poscondiciones:</b>			
1 Permitir acceso al sistema del usuario.			
<b>Curso normal de los eventos:</b>			
<b>Acción del Actor</b>		<b>Respuesta del proceso del Sistema</b>	
1	El trabajador introduce nombre de usuario y contraseña.	1.1	El sistema verifica si los datos son correctos.
		1.2	El sistema permite acceso al usuario.
<b>Curso alternativo de los eventos:</b>			
<b>Acción 1.1</b>		Si los datos no son correctos el sistema muestra un mensaje de error.	
<b>Prioridad:</b>		Critico	

**Tabla 2.13 Descripción del CUS Generar informe de un trabajador**

<b>Caso de Uso:</b>	<b>Generar informe de un trabajador</b>
<b>Actores:</b>	Trabajador(inicia)
<b>Propósito:</b>	Permitir que un trabajador tenga acceso a toda la información relacionada con el.
<b>Resumen:</b>	
El caso de uso se inicia cuando un trabajador elige la opción ver expediente teniendo acceso a la información relacionada con el.	
<b>Referencias:</b>	
R2,R2.1,R2.2,R2.3,R2.4	

<b>Precondiciones:</b>			
1 El trabajador debe tener permisos para acceder a esta parte del sistema.			
<b>Poscondiciones:</b>			
1 Permitir acceso a la información del trabajador.			
<b>Curso normal de los eventos:</b>			
<b>Acción del Actor</b>		<b>Respuesta del proceso del Sistema</b>	
1	El trabajador elige la opción de ver su expediente.	1.1	El sistema realiza una búsqueda de la información relacionada con el trabajador.
		1.2	El sistema muestra la información encontrada en la búsqueda al trabajador.
<b>Curso alternativo de los eventos:</b>			
<b>Acción :</b>		-	
<b>Prioridad:</b>		Primario	

**Tabla 2.14 Descripción del CUS Generar informe general**

<b>Caso de Uso:</b>	<b>Generar informe general</b>
<b>Actores:</b>	Directivo(inicia)
<b>Propósito:</b>	Permitir que un directivo pueda obtener un informe general de toda la información relacionada con las personas de su área.
<b>Resumen:</b>	

El caso de uso se inicia cuando un directivo elige la opción ver informe general, teniendo acceso a toda la información relacionada con los trabajadores de su área.

**Referencias:**  
R3,R3.1,R3.2,R3.3,R3.4,R3.5

**Precondiciones:**  
1 El directivo debe tener permisos para acceder a esta parte del sistema.

**Poscondiciones:**  
1 Permitir acceso al informe con los datos relacionados con los trabajadores de su área.

**Curso normal de los eventos:**

Acción del Actor		Respuesta del proceso del Sistema	
1	El directivo elige la opción de ver su informe general.	1.1	El sistema realiza una búsqueda de la toda la información relacionada con los trabajadores de su área.
		1.2	El sistema genera un informe con la información relacionada con los trabajadores de su área encontrada en la búsqueda.
		1.3	El sistema muestra al directivo el informe generado.

**Curso alternativo de los eventos:**

<b>Acción :</b>	-
<b>Prioridad:</b>	Primario



**Tabla 2.22 Descripción del CUS Generar listado.**

<b>Caso de Uso:</b>		<b>Generar listado</b>	
<b>Actores:</b>		Directivo(inicia)	
<b>Propósito:</b>		Permitir que un directivo pueda obtener un listado con la información relacionada a las personas de su área dada una característica.	
<b>Resumen:</b>			
El caso de uso se inicia cuando un directivo elige una de las opciones de listado. El sistema realiza una búsqueda y genera un listado de los trabajadores del área del directivo con los datos de la característica seleccionada.			
<b>Referencias:</b>			
R3.1, 3.2,3.3,3.4,3.5			
<b>Precondiciones:</b>			
1 El directivo debe tener permisos para acceder a esta parte del sistema.			
<b>Poscondiciones:</b>			
1 Mostrar listado con la característica elegida por el directivo.			
<b>Curso normal de los eventos:</b>			
<b>Acción del Actor</b>		<b>Respuesta del proceso del Sistema</b>	
1	El directivo elige la opción de ver listado.	1.1	El sistema le da la posibilidad de elegir el tipo de listado.
<b>Escenario 1: El directivo elige listado de cumpleaños</b>			

1	El directivo elige la opción de ver cumpleaños.	1.1	El sistema le muestra al directivo la opción de elegir fecha.
2	El directivo elige la fecha de la cual quiere obtener el listado.	2.1	El sistema realiza una búsqueda de la fecha de nacimiento y calcula cuales de los trabajadores cumple años en esa fecha.
		2.2	El sistema genera un listado con el nombre de los trabajadores de su área que cumplen año en esa fecha encontrada el la búsqueda.
		2.3	El sistema le muestra al directivo el listado generado.

**Curso alternativo de los eventos:**

<b>Acción :</b>	2.2 En caso de que no aparezca ningún trabajador se le muestra un mensaje al directivo diciéndole que nadie de su área cumple años en esa fecha.
<b>Prioridad:</b>	Secundario

**Escenario 2: El directivo elige listado de categorías**

<b>Acción del Actor</b>		<b>Respuesta del proceso del Sistema</b>	
1	El directivo elige la opción de ver listado de categorías.	1.1	El sistema realiza una búsqueda de la categoría correspondiente a cada trabajador en esa área.
		1.2	El sistema genera un listado con el nombre de los trabajadores y la categoría de cada trabajador del área

			del directivo encontrada el la búsqueda.
		<b>2.3</b>	El sistema le muestra al directivo el listado generado.

**Curso alternativo de los eventos:**

**Acción :**

-

**Prioridad:**

Secundario

**Escenario 3: El directivo elige listado de residencia**

**Curso normal de los eventos:**

<b>Acción del Actor</b>		<b>Respuesta del proceso del Sistema</b>	
<b>1</b>	El directivo elige la opción de ver listado de residencia.	<b>1.1</b>	El sistema realiza una búsqueda de la residencia correspondiente a cada trabajador en esa área.
		<b>1.2</b>	El sistema genera un listado con el nombre de los trabajadores y la residencia de cada trabajador del área del directivo encontrada el la búsqueda.
		<b>2.3</b>	El sistema le muestra al directivo el listado generado.

**Curso alternativo de los eventos:**

**Acción :**

-

**Prioridad:**

Secundario

<b>Escenario 4: El directivo elige listado de salario</b>			
<b>Curso normal de los eventos:</b>			
<b>Acción del Actor</b>		<b>Respuesta del proceso del Sistema</b>	
<b>1</b>	El directivo elige la opción de ver listado de salario.	<b>1.1</b>	El sistema realiza una búsqueda de el salario correspondiente a cada trabajador en esa área.
		<b>1.2</b>	El sistema genera un listado con el nombre de los trabajadores y el salario de cada trabajador del área del directivo encontrado el la búsqueda.
		<b>2.3</b>	El sistema le muestra al directivo el listado generado.
<b>Curso alternativo de los eventos:</b>			
<b>Acción :</b>		-	
<b>Prioridad:</b>		Secundario	
<b>Escenario 5: El directivo elige listado de vacaciones acumuladas.</b>			
<b>Curso normal de los eventos:</b>			
<b>Acción del Actor</b>		<b>Respuesta del proceso del Sistema</b>	
<b>1</b>	El directivo elige la opción de ver listado de vacaciones acumuladas.	<b>1.1</b>	El sistema realiza una búsqueda de las vacaciones acumuladas correspondiente a cada trabajador en esa área.

		<b>1.2</b>	El sistema genera un listado con el nombre de los trabajadores y las vacaciones acumuladas de cada trabajador del área del directivo encontrada el la búsqueda.
		<b>2.3</b>	El sistema le muestra al directivo el listado generado.
<b>Curso alternativo de los eventos:</b>			
<b>Acción :</b>	-		
<b>Prioridad:</b>	Secundario		

**Diagrama de Casos de Uso del Sistema a automatizar**

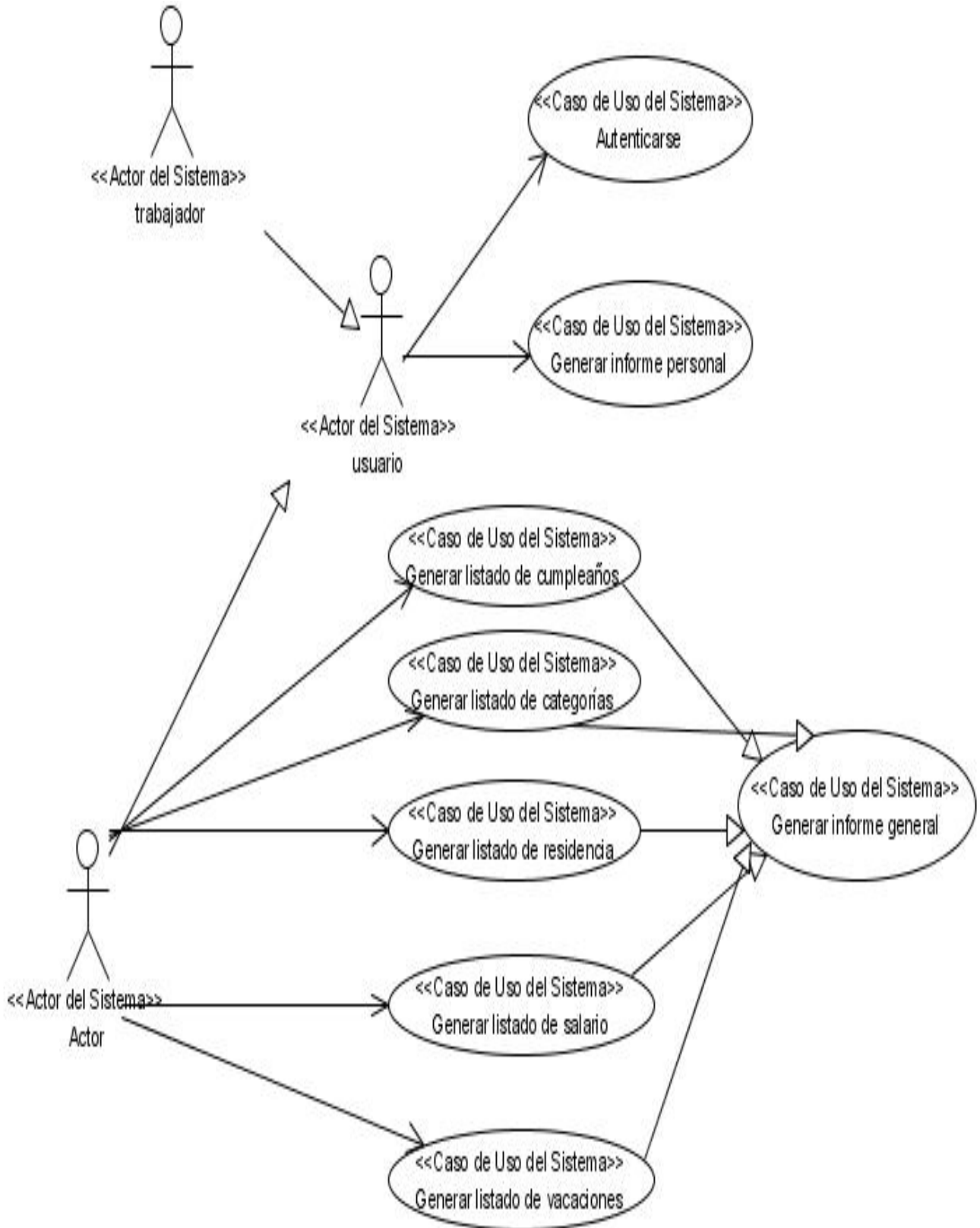


Figure 10 Diagrama de Caso de Uso de Sistema

## Conclusiones

En este capítulo se hace una descripción del negocio a automatizar con el objetivo de ganar en comprensión del problema que se ha de tratar, además de adentrarse un poco en el desarrollo de la propuesta de solución obteniéndose una lista de las funcionalidades que debe tener el sistema final, representada mediante el Diagrama de CU del Sistema creándose la base para comenzar a construir la aplicación.

## **Capítulo 3: Análisis y Diseño del Sistema**

### **Introducción**

En este capítulo se muestran los resultados obtenidos en el flujo de trabajo Análisis y Diseño así como una breve explicación de los artefactos que se construyeron durante el mismo en la búsqueda de una óptima solución final. El propósito fundamental es lograr claridad en relación a la propuesta final del sistema así como lograr una implementación que satisfaga las necesidades de los usuarios.

### **Análisis**

#### **Modelo de Análisis**

En el modelo del análisis hay un refinamiento de los requisitos del software, de manera que facilita su comprensión, su preparación, su modificación y en general su mantenimiento, el objetivo fundamental es lograr una completa comprensión de los mismos. En este flujo se refinan y estructuran los requisitos obtenidos con anterioridad, profundizándose en el dominio de la aplicación. Todo esto permite una mayor comprensión del problema para modelar la solución. [17]

En el análisis se analizan los requisitos con mayor profundidad y se utiliza el lenguaje del desarrollador para describir los resultados. Esto da la ventaja de razonar más sobre los aspectos internos del sistema, y por tanto resolver elementos relativos a la interferencia de casos de uso.

Los propósitos a conseguir con el Análisis son los siguientes:

- Conseguir una comprensión más precisa de los requisitos, refinarlos y estructurarlos.
- Utilizar el lenguaje de los desarrolladores para analizar con profundidad los requisitos funcionales.
- Proporcionar una visión general del sistema.



## Modelo de clases de Análisis

El modelo de análisis está compuesto por clase de análisis, que representan una abstracción de una o varias clases o subsistemas del diseño del sistema. Posee un conjunto de características tales como: se centra en el tratamiento de los requisitos funcionales y pospone los no funcionales, define atributos y participa en relaciones, entre otras.

Las clases de análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio.

Entidad: Modelan información que posee larga vida y que es a menudo persistente.

Interfaz: Modelan la interacción entre el sistema y sus actores.

Control: Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

## Diagrama de clases de Análisis

El diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. A continuación se representan los diagramas de clases de análisis por CUS de la situación propuesta.

**E\_Trabajador:** Es una entidad instantánea que está conformada por los datos del trabajador que está desempeñándose como usuario y que son extraídos de los web services de Assets.

**E\_Trabajador:** Es una entidad instantánea que está conformada por los datos de los trabajadores del área del directivo y que son extraídos de los web services de Assets.

### CUS Autenticarse

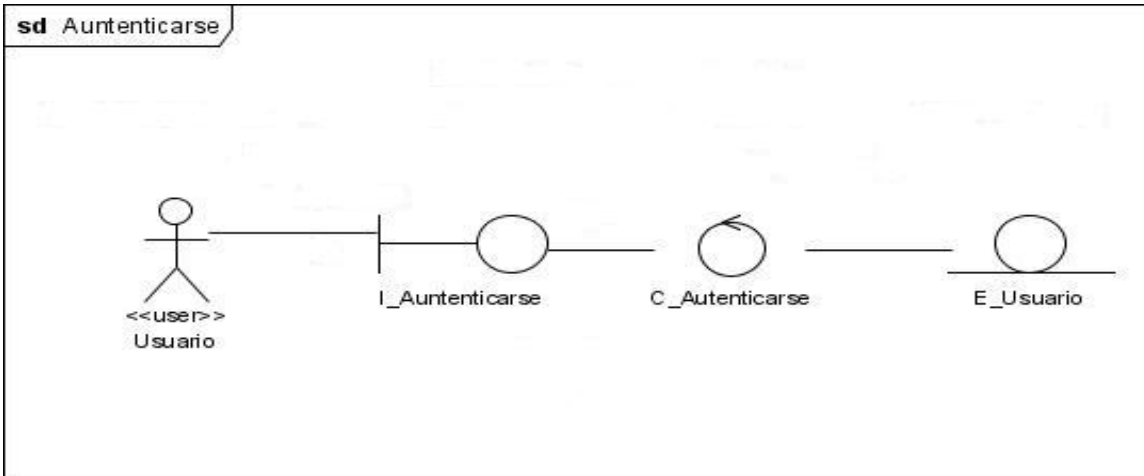


Figure 11 Diagrama de clases CUS Aumenticarse

### CUS Generar informe de un trabajador

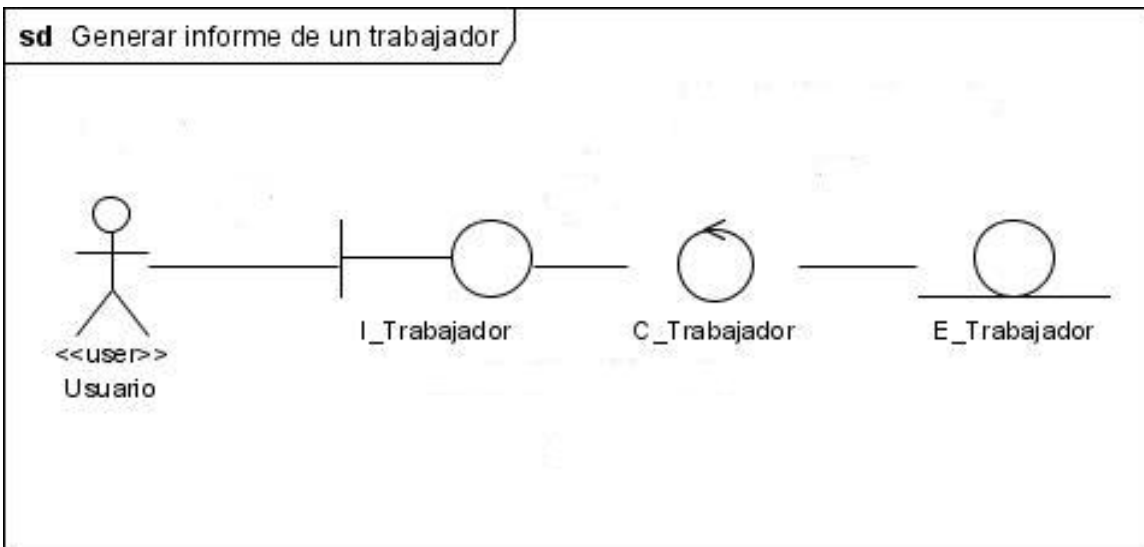


Figure 12 Diagrama de clases CUS Generar informe de un trabajador

### CUS Generar informe general

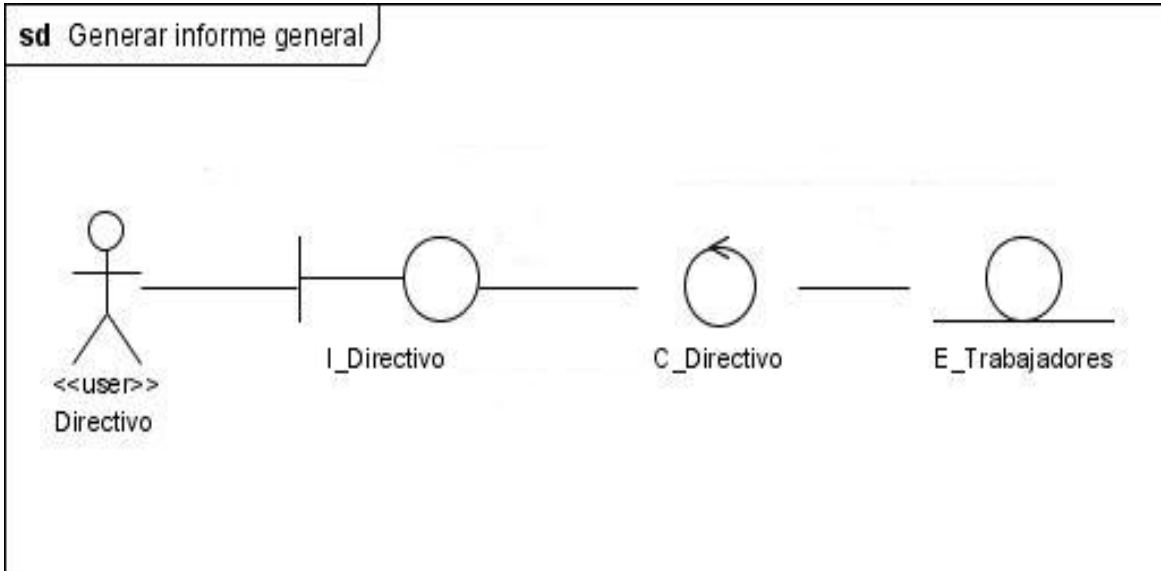


Figure 13 Diagrama de clases CUS Generar informe general

### CUS Generar listado

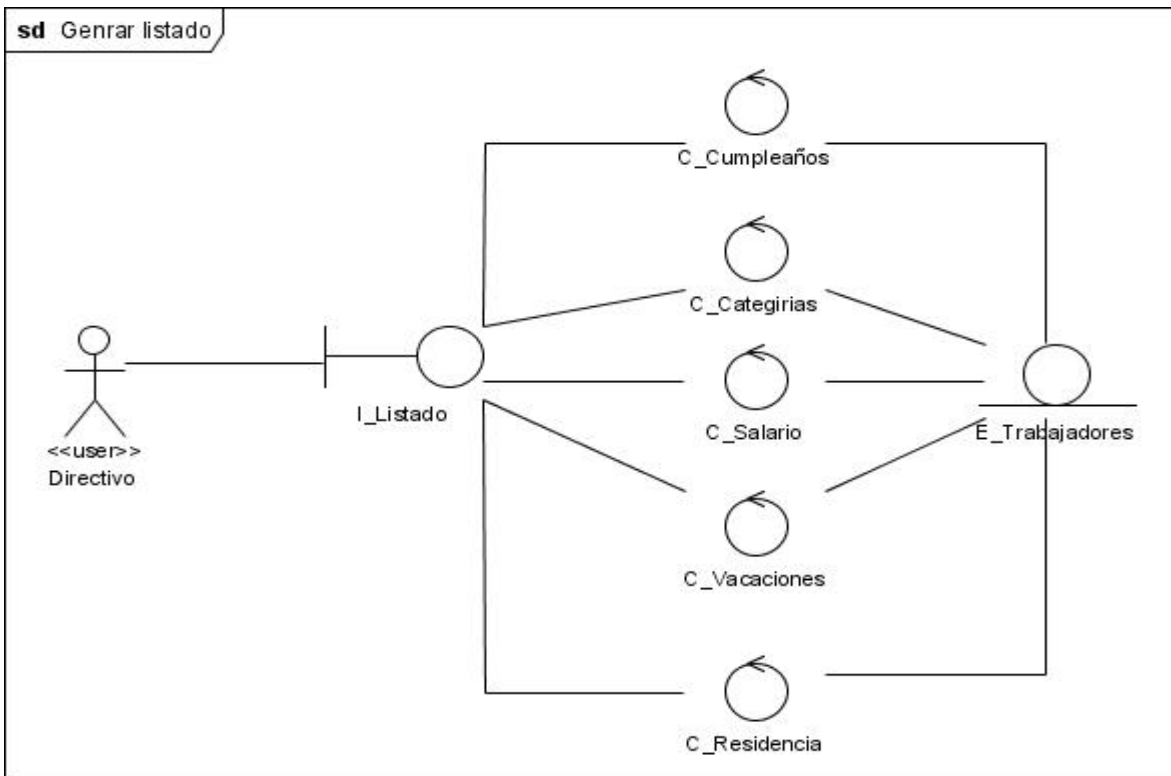


Figure 14 Diagrama de clases CUS Generar Listado



Los diagramas de clases de los distintos escenarios del CUS Generar Listado se encuentran en el Anexo1.

## **Modelo de Diseño**

En el diseño se modela el sistema y se determina la arquitectura para que soporte todos los requisitos, funcionales y no funcionales, así como las restricciones que se le suponen. 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.

El modelo del diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además el modelo de diseño sirve de abstracción de la implementación del sistema, y es de ese modo, utilizada como una entrada fundamental de las actividades de implementación y prueba.

El modelo del Diseño pretende esencialmente:

- Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.
  - Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.
  - Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
  - Capturar las interfaces entre los subsistemas antes en el ciclo de vida del software, lo cual es muy útil cuando utilizamos interfaces como elementos de sincronización entre diferentes equipos de desarrollo. [Jacobson, Ivar; Booch, Grady y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Editorial Félix Varela, La Habana, 2004]
- Para una mejor comprensión describimos a continuación las clases que intervienen en la realización de nuestro sistema.

## Diagrama de interacción

Los diagramas de interacción modelan aspectos dinámicos del sistema, muestra una interacción, que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que pueden enviarse entre ellos.

-Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes.

-Un diagrama de colaboración es un diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes.

Para modelar los aspectos dinámicos de este sistema se utilizaron diagramas de secuencia y colaboración por cada caso de uso.

### Diagrama de Secuencia Autenticarse

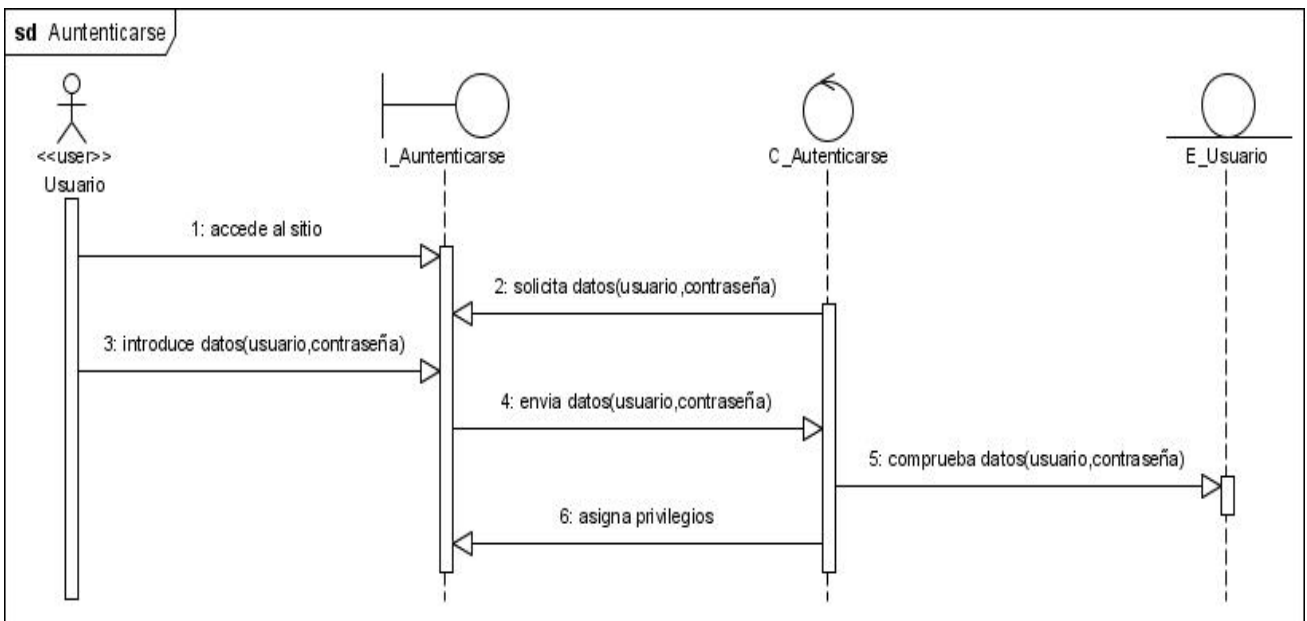


Figure 15 Diagrama de Secuencia Autenticarse

### Diagrama de Secuencia generar informe de un trabajador

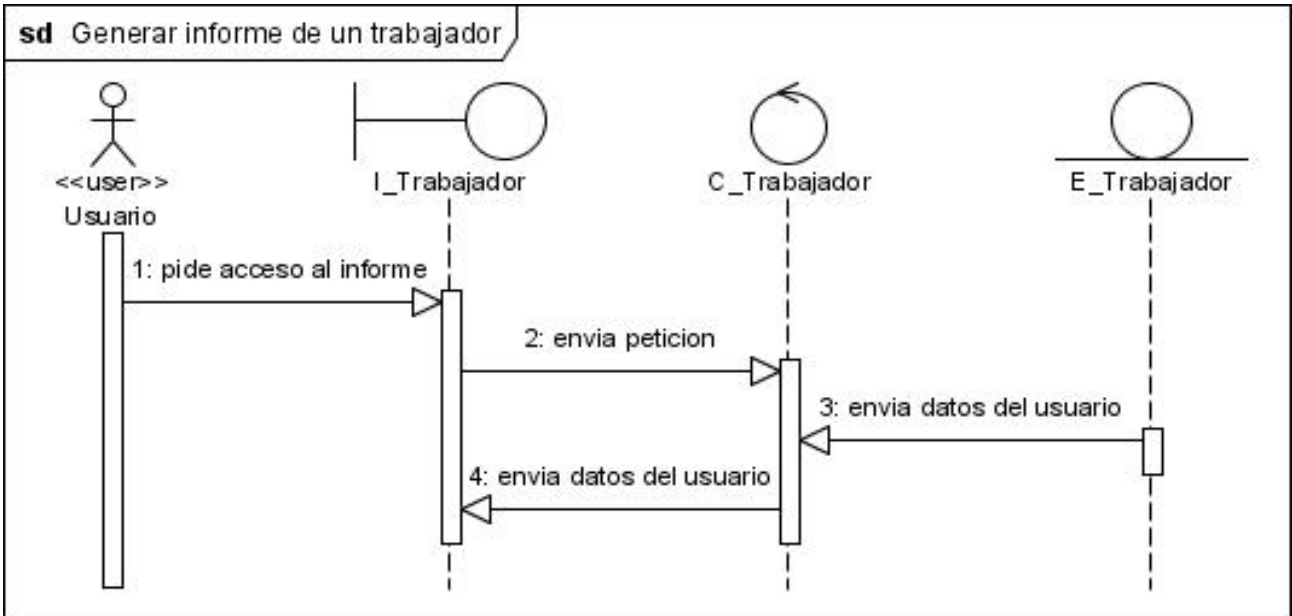


Figure 16 Diagrama de Secuencia generar informe de un trabajador

### Diagrama de Secuencia generar informe general

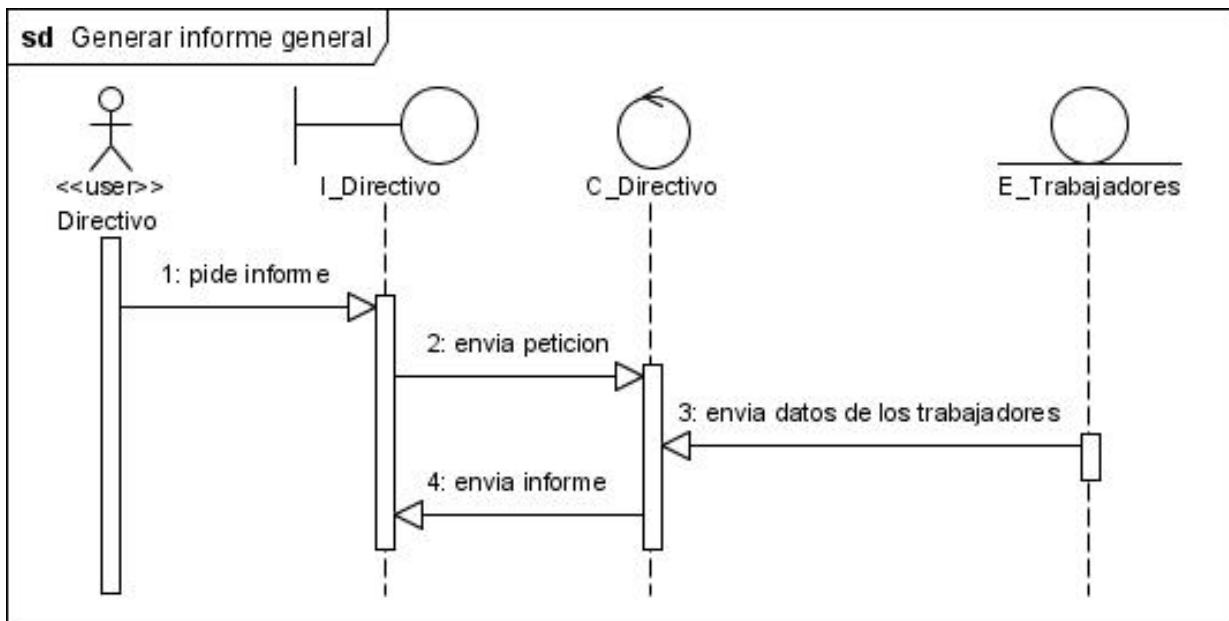


Figure 17 Diagrama de Secuencia generar informe general

### Diagrama de Secuencia de los distintos escenarios del CUS Generar Listado

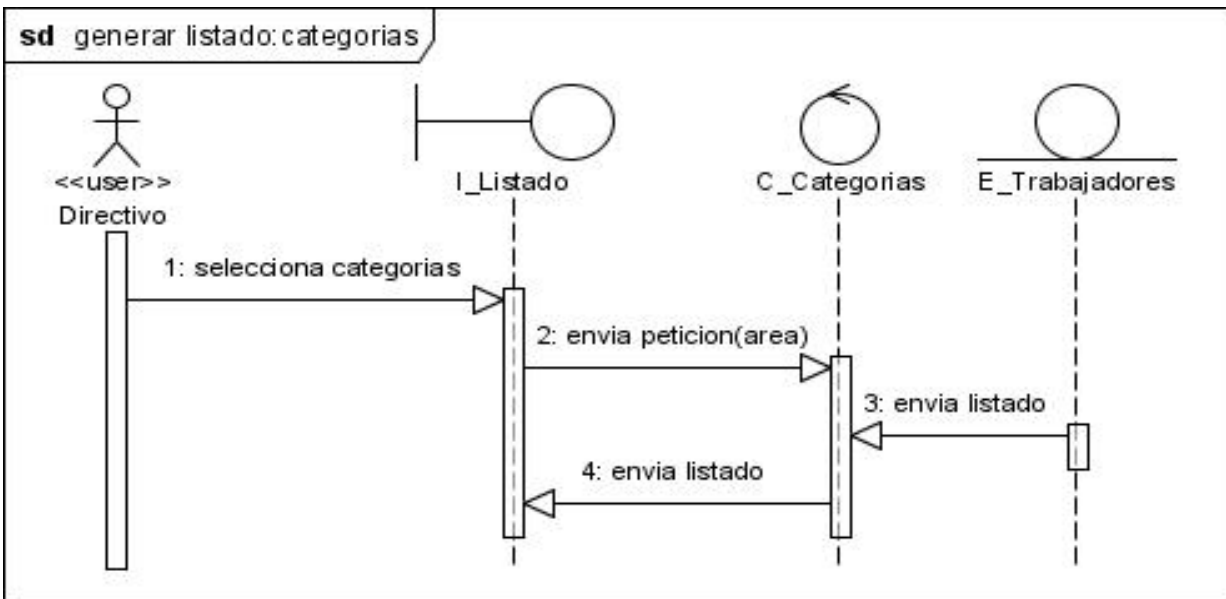


Figure 18 Diagrama de Secuencia listado de categorías

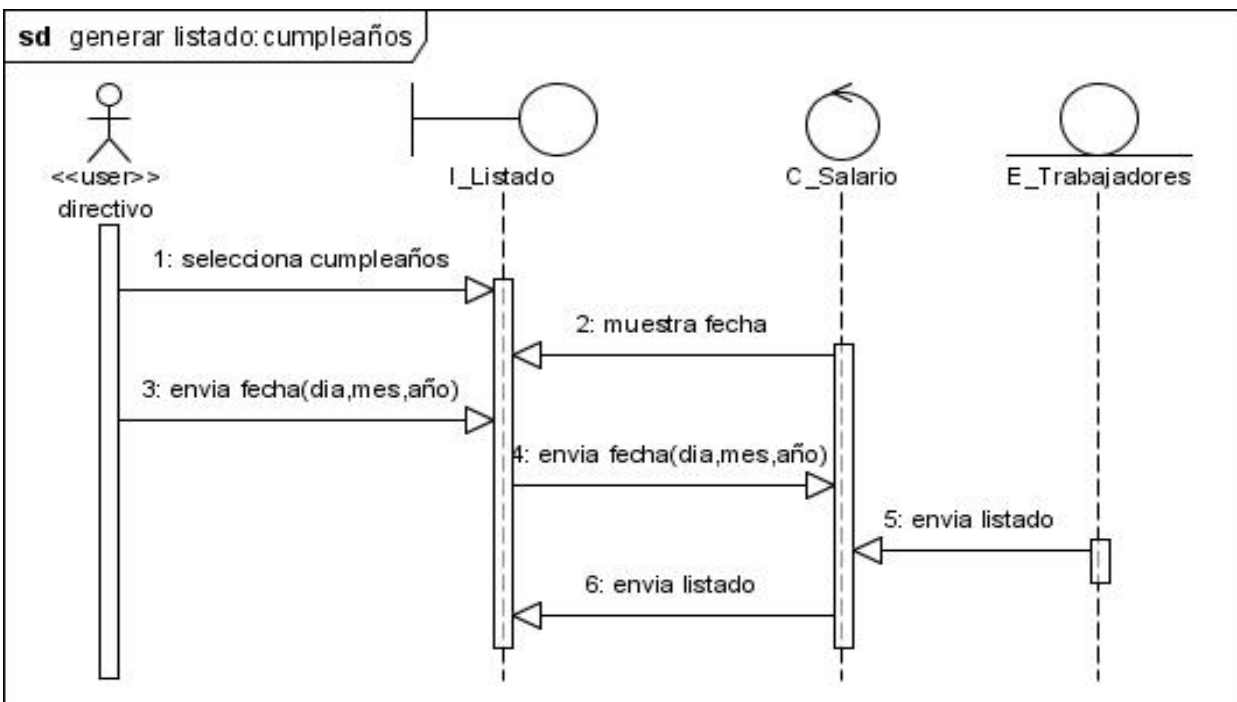


Figure 19 Diagrama de Secuencia listado de cumpleaños

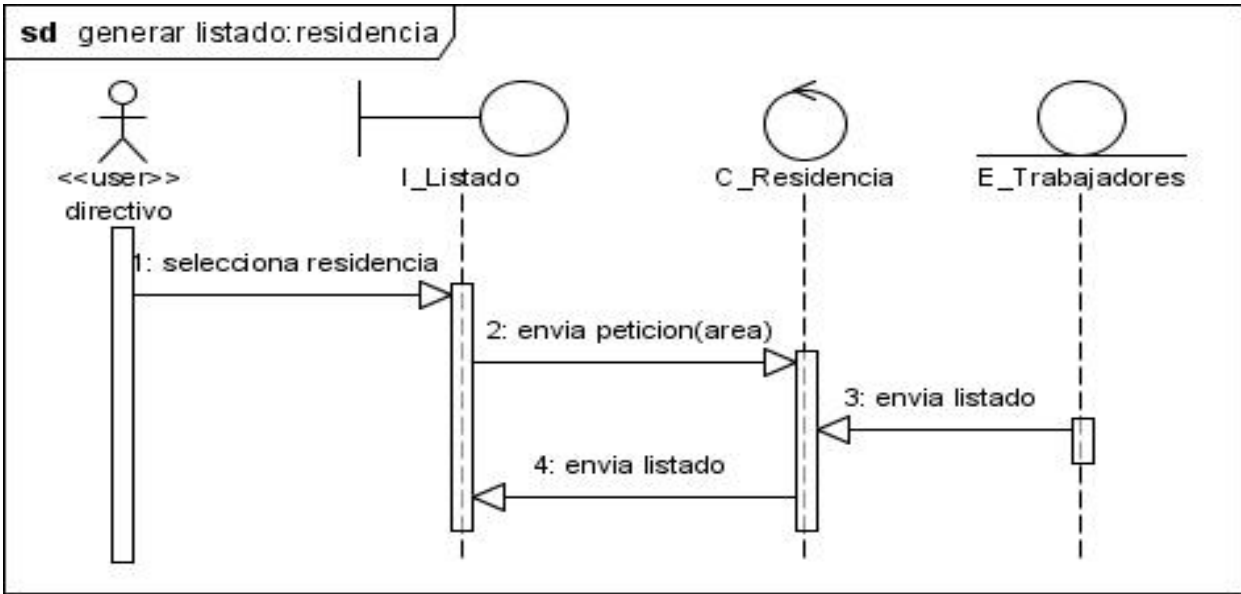


Figure 20 Diagrama de Secuencia listado de residencia

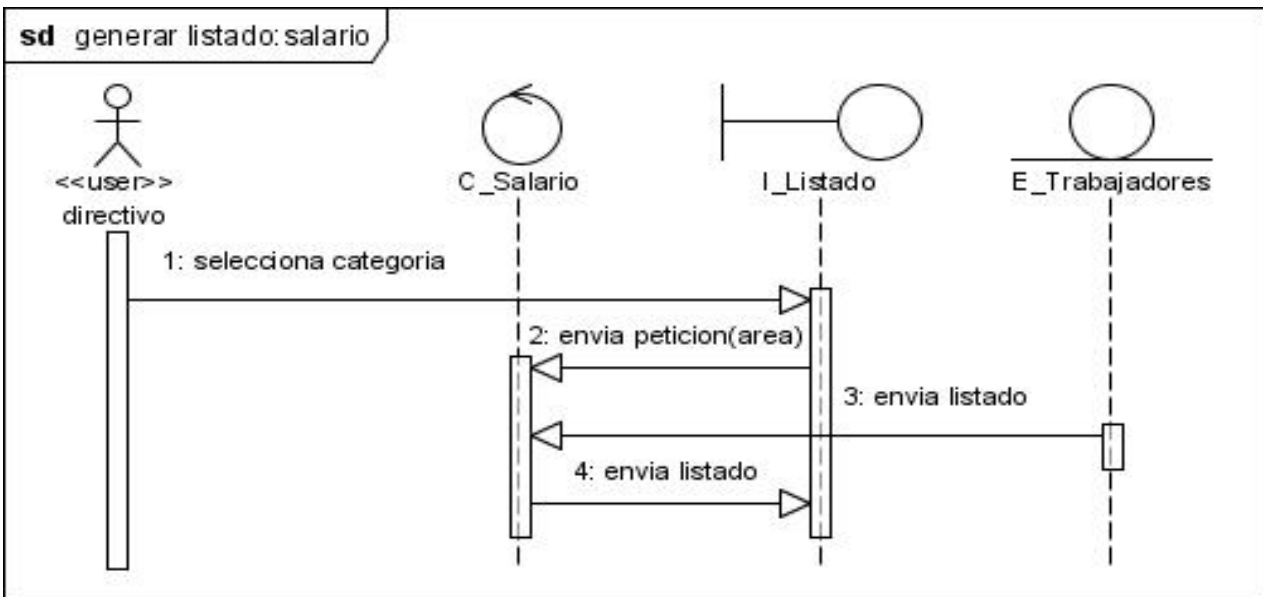


Figure 21 Diagrama de Secuencia listado de salario



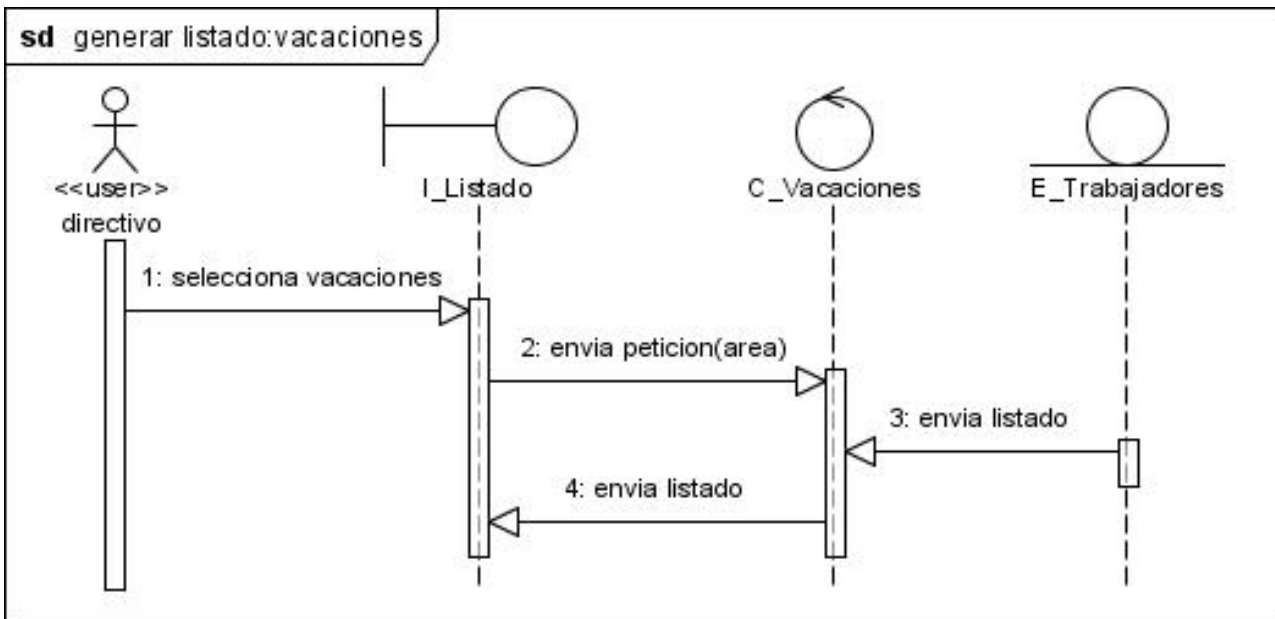


Figure 22 Diagrama de Secuencia listado de vacaciones

### Diagrama de clases del diseño web

Ya que para desarrollar el sistema se a utilizaría Drupal como CMS se explicaran de forma breve el funcionamiento del mismo y posteriormente se desarrollará el diagrama de clases del diseño web.

Drupal genera el contenido indexado en tiempo real en forma de páginas llamadas Node. Las páginas contienen las propiedades básicas de cualquier publicación, como son: título, autor, fecha de creación y otras; las cuales pueden aumentar en la medida en que se integren nuevos módulos a la aplicación.

Un módulo es un archivo que cuenta con uno o más ficheros, el fichero principal es registrado bajo la extensión module e implementa una interface definida por el CMS. Existen dos categorías para los módulos:

- **Módulos de contenido:** Son los que definen un nuevo tipo de contenido, así como las funcionalidades para su creación, edición y posterior publicación.
  - **Módulos funcionales:** Poseen múltiples propósitos en dependencia de las
- En el diseño del sistema expuesto en este capítulo se presentan los módulos de Drupal utilizados y se muestran además las clases persistentes diseñadas, separadas en paquetes, ya que son independientes entre sí en cuanto a funcionalidad. A partir

de este punto se denomina a los módulos, paquetes.

### **Paquete Drupal**

Este paquete contiene la distribución del CMS. A su vez está constituido por cuatro paquetes:

Modules, contiene los módulos que se encargan de las funcionalidades de Drupal.

Includes: donde se encuentran los ficheros de configuración y las clases utilitarias.

Themes: posee los mecanismos de soporte de plantillas y Scripts, que maneja un conjunto de ficheros indispensables para el funcionamiento de la aplicación, fundamentalmente orientados al aspecto visual.

Paquete Script: Contiene los ficheros correspondientes al manejo visual del sistema, CSS y java script.

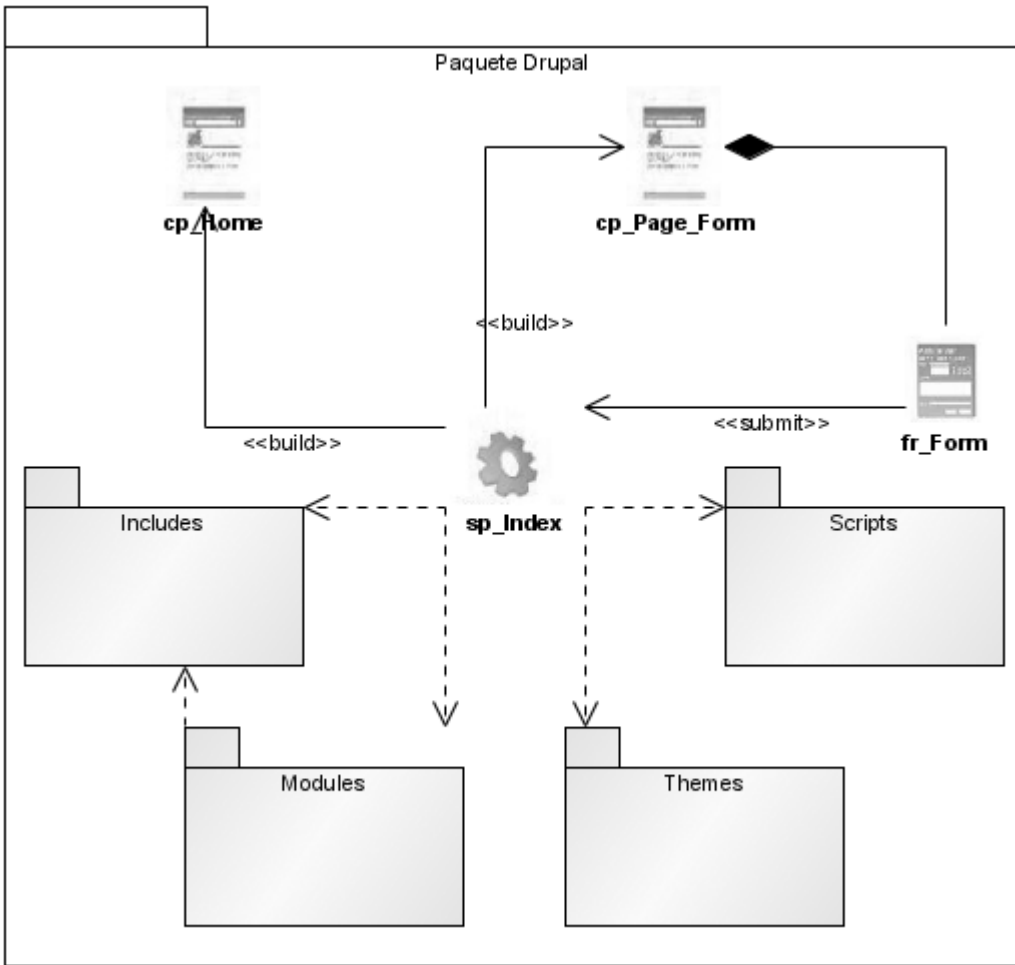


Figure 23 Diagrama de clases de diseño web del paquete Drupal

El paquete Modules contiene las funcionalidades de Drupal y además se le incluyen los paquetes desarrollados para el subsistema de Misión UCI.

Para el correcto funcionamiento del subsistema Misión UCI se utilizan varios paquetes de Drupal. Los fundamentales se enumeran a continuación y son mostrados en el diagrama.

CCK: paquete que permite la creación y personalización de campos.

Views: Muestra y lista el contenido. Este paquete consiste en consultas que se realizan según la información brindada por el desarrollador.

Search: Como su nombre indica este paquete implementa funcionalidades para ayudar en la búsqueda.

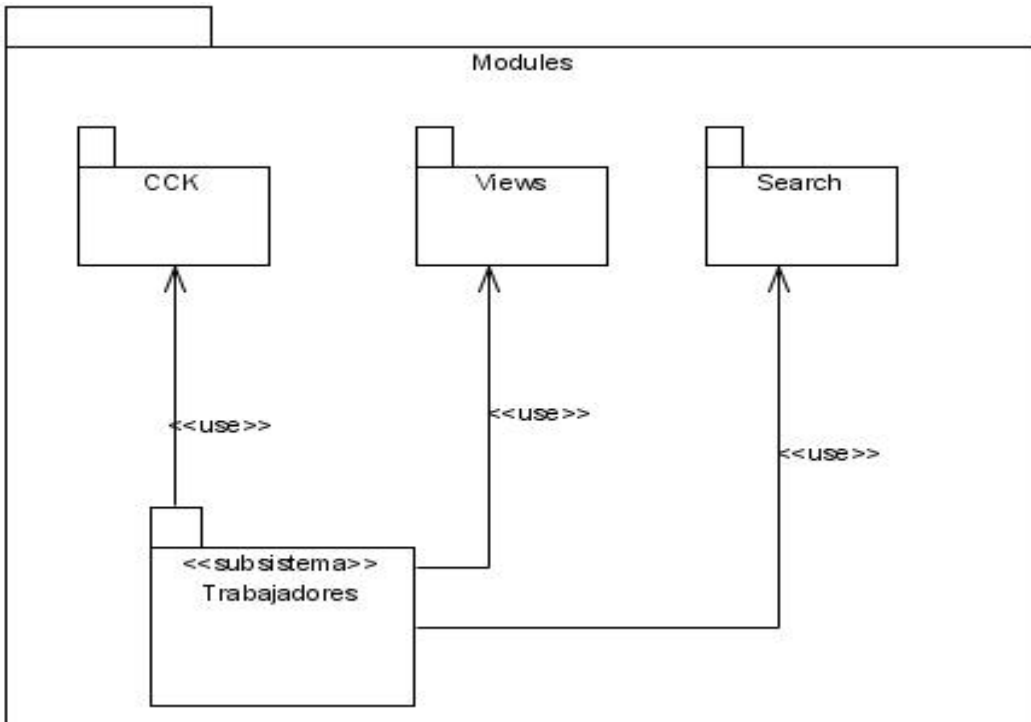


Figure 24 Diagrama de clases del diseño Paquete Modules

El paquete desarrollado contiene las clases persistentes, con sus atributos, necesarios para poder implementar las funcionalidades del sistema. El paquete Trabajadores define los datos. Está compuesto por dos clases persistentes y sus relaciones.

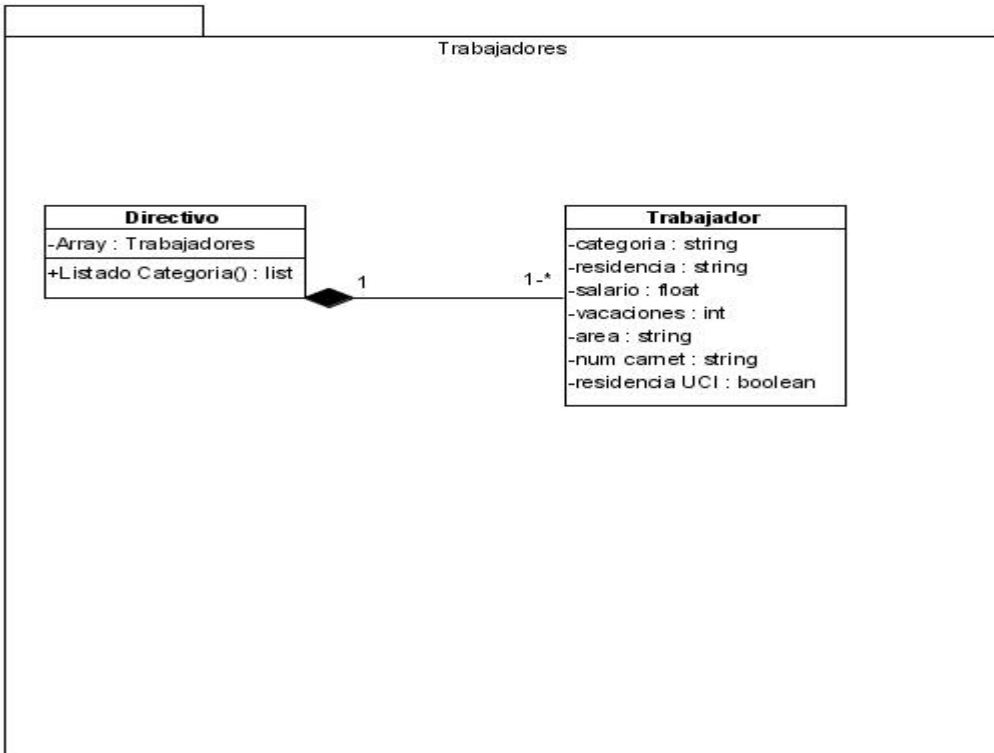


Figure 25 Paquete Trabajadores

## Patrones de diseño de Drupal

### Patrones creacionales

#### Singleton

El patrón de diseño **singleton** (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

El patrón *singleton* se implementa creando en nuestra clase un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con atributos como protegido o privado). La instrumentación del patrón puede ser delicada en programas con múltiples hilos de ejecución. Si dos hilos de ejecución intentan crear la instancia al mismo tiempo y esta no existe todavía, sólo uno de ellos debe lograr crear el objeto. La solución clásica para este problema es utilizar exclusión mutua en el método de creación de la clase que implementa el patrón. Las situaciones más habituales de aplicación de este

patrón son aquellas en las que dicha clase controla el acceso a un recurso físico único (como puede ser el ratón o un archivo abierto en modo exclusivo) o cuando cierto tipo de datos debe estar disponible para todos los demás objetos de la aplicación.

El patrón *singleton* provee una única instancia global gracias a que:

- La propia clase es responsable de crear la única instancia.
- Permite el acceso global a dicha instancia mediante un método de clase.
- Declara el constructor de clase como privado para que no sea instanciable directamente.

## Patrones Estructurales

### Decorator (patrón de diseño)

El patrón **Decorator** responde a la necesidad de añadir dinámicamente funcionalidad a un Objeto. Esto nos permite no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera.

## Aplicabilidad

- Añadir objetos individuales de forma dinámica y transparente
- Responsabilidades de un objeto pueden ser retiradas
- Cuando la extensión mediante la herencia no es viable
- Hay una necesidad de extender la funcionalidad de una clase, pero no hay razones para extenderlo a través de la herencia.
- Hay la necesidad de extender dinámicamente la funcionalidad de un objeto y quizás quitar la funcionalidad extendida.

## Patrones de Comportamiento

### Observer (patrón de diseño)

El patrón Observador también conocido como "spider" define una dependencia del tipo uno-a-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros dependientes. El objetivo de este patrón es desacoplar la clase de los objetos clientes del objeto, aumentando la modularidad del lenguaje, así como evitar bucles de actualización (espera activa o polling).

Este patrón también se conoce como el patrón de publicación-inscripción o

modelo-patrón. Estos nombres sugieren las ideas básicas del patrón, que son bien sencillas: el objeto de datos, llamémoslo "Objeto" a partir de ahora, contiene atributos mediante los cuales cualquier objeto observador o vista se puede suscribir a él pasándole una referencia a sí mismo. El Sujeto mantiene así una lista de las referencias a sus observadores. Los observadores a su vez están obligados a implementar unos métodos determinados mediante los cuales el Sujeto es capaz de notificar a sus observadores "suscritos" los cambios que sufre para que todos ellos tengan la oportunidad de refrescar el contenido representado. De manera que cuando se produce un cambio en el Sujeto, ejecutado, por ejemplo, por alguno de los observadores, el objeto de datos puede recorrer la lista de observadores avisando a cada uno.

Este patrón suele observarse en los marcos de interfaces gráficas orientados a objetos, en los que la forma de capturar los eventos es suscribir 'listeners' a los objetos que pueden disparar eventos.

## **Arquitectura Candidata**

### **Arquitectura de Capas**

Después de un análisis profundo de varias arquitecturas se llegó a la conclusión de que la más acertada para desarrollar el sistema es la arquitectura de capas. Esta arquitectura define como organizar el modelo de diseño a través de capas que pueden estar físicamente distribuidas o no lo cual quiere decir que los componentes de una capa solo pueden hacer referencia a componentes que se encuentran en la capa inmediata inferior. Este patrón es de gran utilidad porque simplifica la comprensión y la organización de sistemas complejos reduciendo las dependencias, de manera que las capas inferiores no son conscientes de los detalles de las capas superiores. Este patrón es de gran ayuda en la identificación de los elementos que pueden ser reutilizados. El sistema tendría tres capas, una de capa de presentación que estaría conformada por las clases de diseño de Drupal y las páginas que conforman el sitio web, una capa de aplicación que estaría conformada por las clases que integran el paquete Trabajadores las cuales se encargan de realizar los procesos del negocio y una capa de acceso a datos conformada por los servicios web que serán utilizados por las clases del paquete Trabajadores.



## **Conclusiones**

En este capítulo hemos profundizado en las características de la propuesta del sistema y se modeló detalladamente la solución propuesta, a través del modelo de diseño, se representó el modelo lógico así como el modelo físico.



## **Conclusiones**

La realización de este trabajo responde a la necesidad de buscar una alternativa que permita el acceso a la información relacionada con los trabajadores de la UCI sin poner en riesgo la misma, especialmente, teniendo en cuenta que el acceso de los mismos representa una labor compleja, que requiere de un lento y cuidadoso tratamiento. Con el objetivo de lograr una solución lo más óptima posible se estudió como se llevaban a cabo los procesos, cuales eran las necesidades de los clientes, se hizo un levantamiento de requisitos y se analizaron las herramientas, técnicas, tecnologías, metodologías más factibles para el desarrollo, además se buscaron referencias en el ámbito internacional, nacional y dentro de la propia universidad, de sistemas que manejaran de alguna manera el tema.

Luego de todo este estudio previo se llegó a la conclusión de que el sistema sería implementado en PHP, CMS Drupal, Servidor Web Apache, el diseño de interfaz en Quanta Plus y la metodología de desarrollo RUP.

A partir de este momento se modeló el negocio propuesto, se detectaron actores y trabajadores, además de identificar cuales serían las actividades a automatizar.

Se definieron los requerimientos funcionales y no funcionales, se representó el diagrama de Casos de Uso del Sistema y se describieron para una mejor comprensión y entendimiento de sus funcionalidades. Se precisaron las clases de análisis y se diseñó el sistema, a través de diagramas de clases Web, entre otros.

Luego de todo este proceso se concluye que el Módulo para la gestión de información de los trabajadores significa una mejora sin duda para facilitar el trabajo con la información recopilada, así como que se han cumplido los objetivos propuestos. Este trabajo tiene como aporte principal optimizar la búsqueda de información evitándole perder tiempo útil a los directivos y trabajadores de la UCI.

## **Recomendaciones**

A partir de la presente investigación y teniendo en cuenta las ideas que surgieron durante el desarrollo del análisis y diseño del módulo para la gestión de información de los trabajadores se considera que la dirección del proyecto de Informatización debe tomar en cuenta las siguientes recomendaciones:

- Implementar el Sistema de Gestión de Trabajadores teniendo en cuenta el análisis y diseño propuesto en esta investigación.
- Continuar la investigación iniciada con el presente trabajo en lo relativo a la gestión de información de los trabajadores para garantizar mejoras en las próximas fases de desarrollo.
- Mantener sobre el sistema un estricto cumplimiento del proceso de mantenimiento y actualización periódica, logrando así que se mantenga la fiabilidad y funcionamiento óptimo del sistema y de la información que se gestiona a través de él.
- Realizar una investigación que ayude a la creación de otros módulos que se puedan añadir al sistema que permitan un mejor funcionamiento y mayor eficacia.

## Referencias Bibliográficas

[1]HSM. *HSM Management*, 2008. [Disponible en: <http://www.gestion.com>]

[2]NETWORK., F. *BUSCAR EN EL DICcionario INFORMÁTICO*, 2000. [Disponible en: <http://www.lawebdelprogramador.com/diccionario/buscar.php?letra=&cadena=Informaci%F3n&x=13&y=5>]

[3]NETWORK., F. *BUSCAR EN EL DICcionario INFORMÁTICO*, 2000. [Disponible en: <http://www.lawebdelprogramador.com/diccionario/buscar.php?letra=&cadena=Informaci%F3n&x=13&y=5>]

[4]ACIMED *Revista Cubana de los profesionales de la Información y la comunicación en salud*, (septiembre-octubre, 2002), 10.

[5]NETWORK., F. *BUSCAR EN EL DICcionario INFORMÁTICO*, 2000. [Disponible en: <http://www.lawebdelprogramador.com/diccionario/buscar.php?letra=&cadena=Informaci%F3n&x=13&y=5>]

[6]INFOMED. *Biblioteca medica nacional* 2007. [Disponible en: <http://www.sld.cu/sitios/bmn/temas.php>]

[7]GONZALES, M. Y. S. D. Y. L. P. U. *Sistemas de gestión de contenidos: En busca de una plataforma ideal*, 2006.

[8]CECAM Sistemas de Gestión Universitaria, 2007.

[9]10, U. F. *curso optativo de XML*, 2007/2008.

[10]SERVER, S. S. *SOAP Client*, 2005. [Disponible en: <http://soapclient.com/>]

[11]SERVER, S. S. *SOAP Client*, 2005. [Disponible en: <http://soapclient.com/>]



[12]OASIS. *The web page you are looking for is no longer available*, 2006. [Disponible en: <http://uddi.xml.org/uddi-org>]

[13]GONZALES, M. Y. S. D. Y. L. P. U. *Sistemas de gestión de contenidos: En busca de una plataforma ideal*, 2006.

[14]MÁRQUEZ, P. *CMS Drupal*, CINCE, 2008. [Disponible en: <http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=547&mode=thread&order=0&thold=0>]

[15]LAFFOON, E. *Quanta Plus*, 2005. [Disponible en: <http://quanta.kdewebdev.org/>]

[16]MANAGER, F. D. *Visual Paradigm para UML*, 2007. [Disponible en: [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(Iglesia\\_Anglicana\)\\_%5Bcuenta\\_de\\_Linux\\_14716\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Iglesia_Anglicana)_%5Bcuenta_de_Linux_14716_p/)]

[17] Jacobson, Ivar; Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. Editorial Félix Varela, La Habana, 2004.

## **Bibliografía**

ACHOUR, M.; F. BETZ, *et al. Manual de PHP*, 2008.

ALFONSO, X. C. G. Y. J. M. Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto, 2004.

GALVEZ, I. J. FUNDAMENTOS DE LA METODOLOGIA RUP RATIONAL UNIFIED PROCESS, 2007.

JACOBSON, I. B., GRADY Y RUMBAUGH, JAMES. *El Proceso Unificado de Desarrollo de Software*. Habana, Félix Varela, 2004. p.

PARADIMG, I. V. *Visual Paradigm for UML.*, 2008. [Disponible en: <http://www.visual-paradigm.com/product/vpuml/>]

PRESSMAN, R. *Ingeniería de Software*. 1998. p.

## Glosario de términos

**Herramienta CASE:** Acrónimo de Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador, 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.

**RUP:** El Proceso Unificado de Rational (RUP, en inglés Rational Unified Process) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

**Software libre:** Es la denominación del software que brinda libertad a los usuarios sobre un producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

**UML:** (Unified Modeling Language) Lenguaje Unificado de Modelado. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

**UML:** Unified Modeling Language en español lenguaje unificado de modelado.

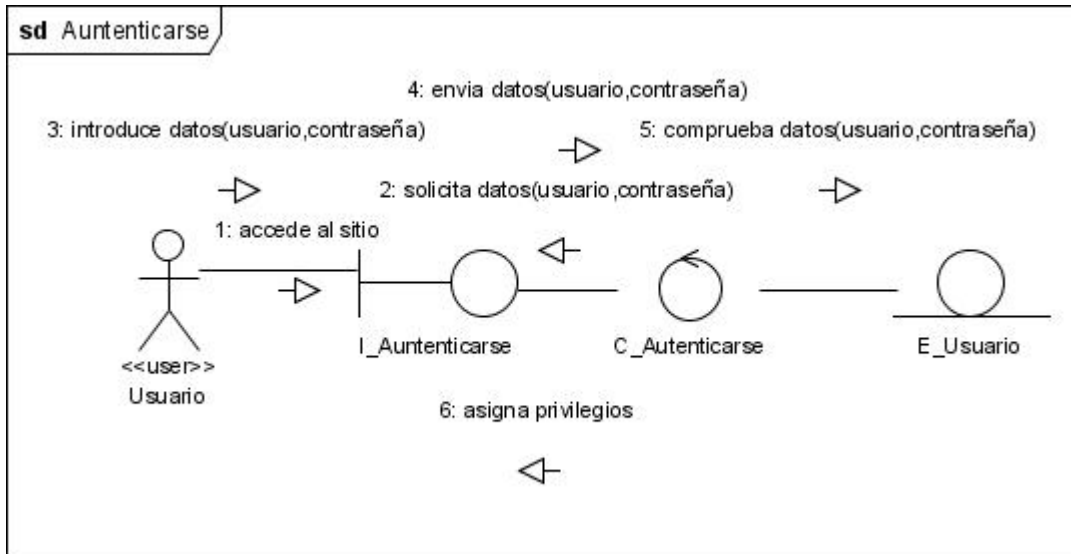
**XML:** significa extensible markup language, o lenguaje de anotación extensible. XML no es sólo un lenguaje, es una forma de especificar lenguajes, de ahí lo de extensible. Todo lenguaje que se exprese de una forma determinada puede ser XML.

**HTTP:** HyperText Transfer Protocol o Protocolo de transferencia de hipertexto es el método más común de intercambio de información en la Word Wide Web. Es el sistema mediante el cual se transfiere información entre los servidores y los clientes.

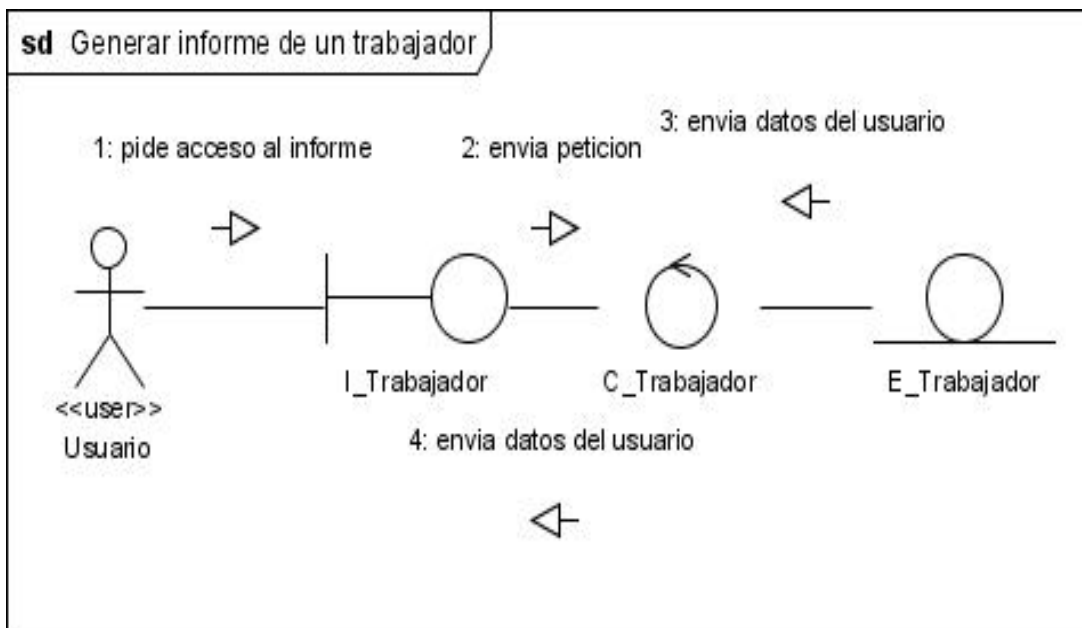
## ANEXOS

### Diagramas de Comunicación

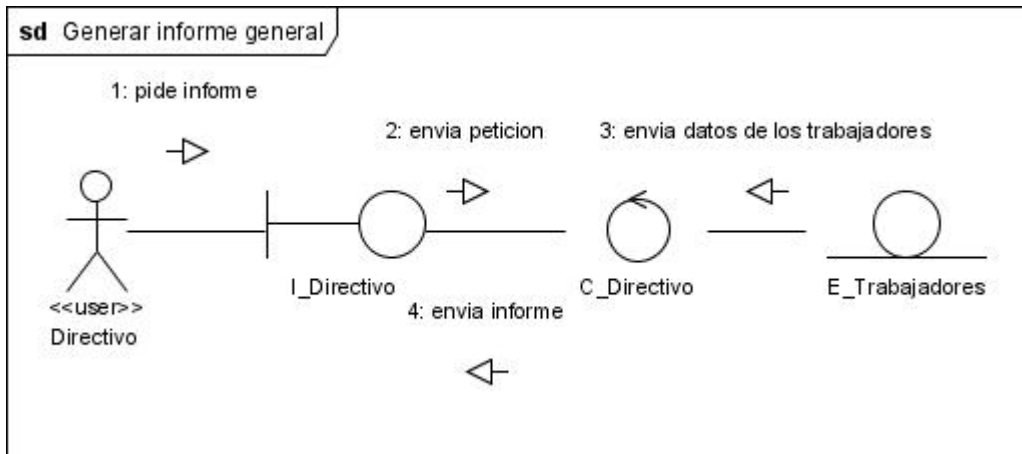
#### Autenticarse



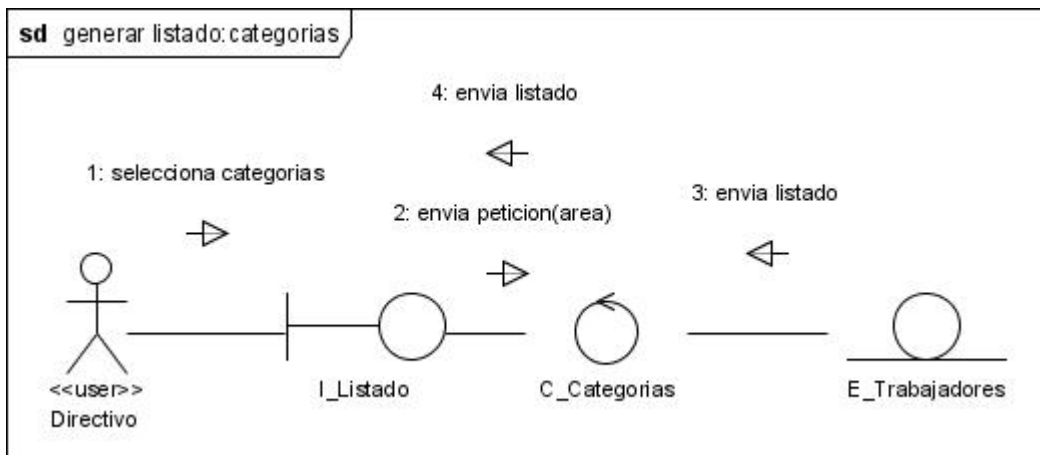
#### Generar informe de un trabajador



### Generar informe general



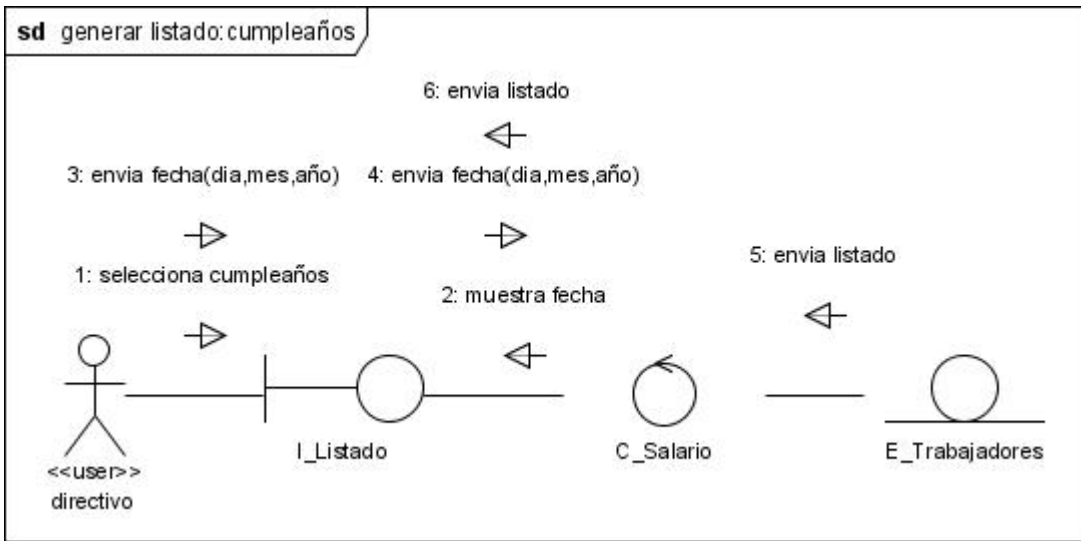
### Generar listado: categorías



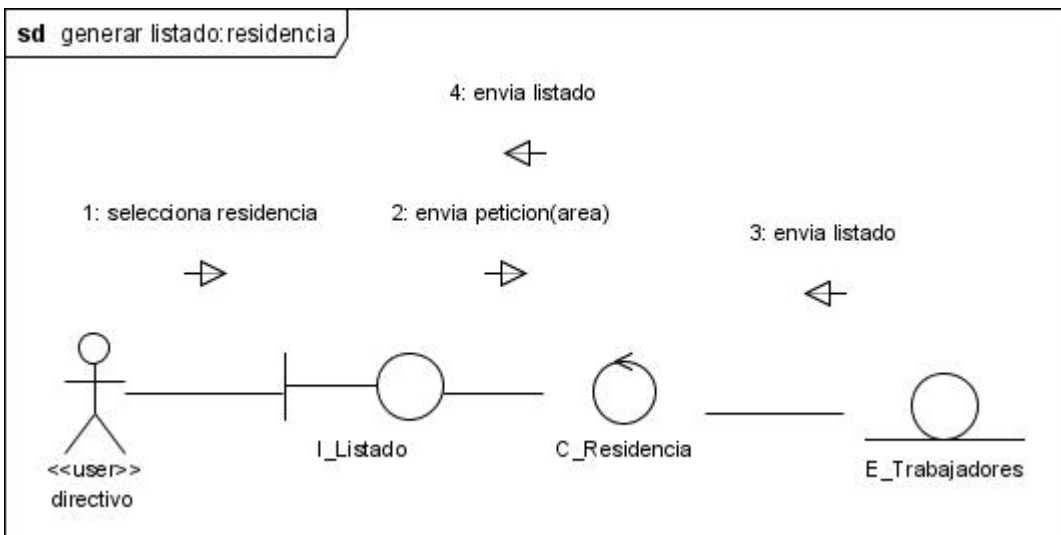
### Generar listado: cumpleaños



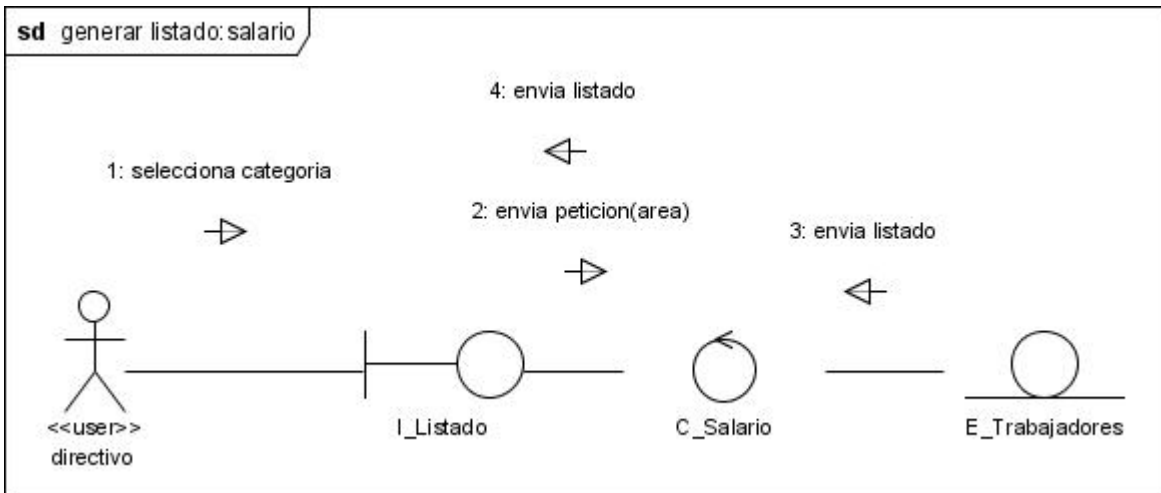
-----ANEXOS-----



**Generar listado: residencia**



**Generar listado: salario**



### Generar listado: vacaciones

