

Universidad de las Ciencias Informáticas

Facultad 7



**Solución informática para la reproducción controlada de no conformidades en las pruebas realizadas a sistemas web**

Trabajo de Diploma para optar por el Título de  
Ingeniero en Ciencias Informáticas

Autor: Alexei Pérez Cepero

Tutor: Ing. Yurién Ricardo Fuentes Guerra

Co-tutora: Ing. Yislenys Suárez Hernández

La Habana, junio de 2011  
“Año 53 de la Revolución”

## **DATOS DE CONTACTO**

### **Tutores:**

#### **Tutor principal:**

Ing. Yurién R. Fuentes Guerra:

Ingeniero en Ciencias Informáticas, graduado en la Universidad de las Ciencias Informáticas (UCI) en el año 2008. Actualmente es Jefe de pruebas en el Grupo de Calidad del Centro de Informática Médica (CESIM) de la UCI. Es Profesor Instructor de la UCI e imparte Ingeniería de software. Correo electrónico: yrfuentes@uci.cu

#### **Co tutora:**

Ing. Yislenys Suárez Hernández:

Ingeniera en Ciencias Informáticas, graduada en el 2009. Estuvo al frente del equipo de Auditorías y Revisiones en el grupo de calidad del CESIM. Actualmente se desempeña como analista en el proyecto SIGICEM del departamento Sistemas de Apoyo a la Salud. Correo electrónico: ysuarezh@uci.cu

## DEDICATORIA

A mi papá por inculcarme sus valores y su rectitud y lograr ser una persona ejemplar ante mis ojos.

A mi mamá por estar conmigo en cada momento de mi vida y ser mi luz, fuente de amor y cariño.

A Yuri por ser hermana, confidente, mejor amiga y madre conmigo, te extraño mi vida deseo verte pronto.

A Mimi, mi querida abuela que aunque no esté conmigo en este momento siempre estará en mi corazón.

A mi Abuela Libia por darme su amor, afecto y cariño, por quererme tanto.

A mi abuelo Pí por ser mi ejemplo a seguir y la persona más bella, sencilla y que más extraña mi corazón, te recuerdo siempre y que tu luz siempre me acompañe yo siempre te llevaré conmigo.

A mi tío Lino que tanto me ha aguantado, y que su apoyo incondicional nunca me faltado, gracias tío por aguantarme y ser más que mi familia.

A mis tíos Tatica y Papo por ser siempre tan preocupados por mí, ser mi familia y estar ahí siempre.

A mi tía Zoe y mis primos Yuleydi y Yudiel por quererme tanto y ser tan preocupados conmigo.

A mis amistades de hoy y espero que siempre Ernesto Orobio, Damián, Ariel, Yurién, Yanicet, Lisandra y Javier Franco a quien estimo mucho.

A mis queridos profes del proyecto, especialmente a Yurién por soportarme siempre y reírse de 'mi tallas'

Y a mí queridísima Aliu, antes un sueño, hoy y por siempre mi novia, mi cosita linda, la mujer que quiero y que deseo y con la que quiero estar siempre, gracias por existir mi amor y por llegar a mi vida en el momento en que más te necesito, mi vida está contigo, gracias por tu apoyo, tu cariño y por quererme.

A mi familia, especialmente a mi papá, a mi mamá, a mi hermana, a mi tío Lino, a mi abuelito Pí y a mi abuela Mimi que están en el cielo y que hicieron tanto para que yo fuera la persona que soy hoy.

A todos en general por darme amor, cariño y comprensión, por ayudarme a crecer.

## **AGRADECIMIENTOS**

*A la Revolución por hacer realidad este sueño. A todos los profesores y compañeros que desde un inicio han ayudado a nuestra formación en las aulas y el proyecto.*

*A la Universidad de las Ciencias Informáticas por ser nuestro hogar durante estos años de formación profesional y personal.*

*A los tutores Yurién Ricardo Fuentes Guerra y Yislenys Suárez Hernández y demás compañeros que han ayudado en el desarrollo del presente trabajo, en especial a Aliu por siempre brindarme su ayuda y navegar a mi lado.*

*A todos mis amigos que han formado parte de mi vida los quiero y les agradezco por existir.*

*Alexei Pérez Cepero*

## **RESUMEN**

En el Centro de Informática Médica existen proyectos de desarrollo de software para los Sistemas de Salud Pública de Cuba y otros países latinoamericanos. En el centro funciona un Grupo de Calidad que mediante sesiones de prueba, avala o no la liberación de software. Durante una sesión de pruebas, al detectar una No Conformidad se procede a su registro. Este proceso no se realiza de manera óptima, pues su ubicación se registra en un texto plano de forma libre. Como consecuencia puede ocurrir pérdida o confusión en el momento en que otro usuario intente reproducir dicha ubicación.

Todo esto trae consigo la necesidad de crear una solución informática para la reproducción controlada de No Conformidades que permita la automatización del componente de su ubicación.

La aplicación se desarrolla sobre tecnologías como el Komodo IDE, Javascript y XUL como lenguajes de programación, Visual Paradigm como herramienta CASE y RUP como metodología de desarrollo. Finalmente se obtuvo una extensión que le permita al usuario grabar de forma dinámica toda la serie de pasos que componen la ubicación de una NC durante las pruebas a un sistema web, así como realizar múltiples funcionalidades relacionadas asociadas al problema. Esto le trae al usuario facilidad de uso y agilidad durante el proceso de registro de NC en una sesión de pruebas.

### **Palabras claves:**

No Conformidad, extensión, proceso de pruebas, pruebas, calidad, sistemas web, ubicación, probador.

# TABLA DE CONTENIDOS

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA</b> .....	<b>6</b>
1.1. CONCEPTOS BÁSICOS ASOCIADOS AL DOMINIO DEL PROBLEMA.....	6
1.2. SISTEMAS AUTOMATIZADOS EXISTENTES.....	7
1.3. ANÁLISIS CRÍTICO.....	14
1.4. HERRAMIENTAS Y TECNOLOGÍAS.....	14
1.5. METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	29
1.6. TECNOLOGÍAS Y HERRAMIENTAS A UTILIZAR.....	34
<b>CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA</b> .....	<b>35</b>
2.1. OBJETO DE ESTUDIO.....	35
2.2. OBJETO DE AUTOMATIZACIÓN.....	35
2.3. PROPUESTA DEL SISTEMA.....	35
2.4. DESCRIPCIÓN DEL MODELO DE DOMINIO.....	36
2.5. REQUERIMIENTOS FUNCIONALES DEL SISTEMA.....	37
2.6. REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA.....	38
2.7. DEFINICIÓN DE LOS CASOS DE USO.....	40
2.8. DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	41
<b>CAPÍTULO 3. DISEÑO DEL SISTEMA</b> .....	<b>46</b>
3.1. ARQUITECTURA DEL SISTEMA.....	46
3.2. MODELO DE DISEÑO.....	48
<b>CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBAS</b> .....	<b>52</b>
4.1. DESPLIEGUE.....	52
4.2. DIAGRAMA DE COMPONENTES.....	52
4.3. PRUEBAS.....	53
<b>CONCLUSIONES</b> .....	<b>58</b>

<b>RECOMENDACIONES.....</b>	<b>59</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>60</b>
<b>BIBLIOGRAFÍA.....</b>	<b>62</b>
<b>ANEXOS .....</b>	<b>65</b>
ANEXO 1.....	65
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>70</b>

---

## **INTRODUCCIÓN**

La evolución de las TIC<sup>1</sup> ha ubicado a los sistemas informáticos actuales en un rol decisivo dentro de las instituciones y empresas. Es un fenómeno tecnológico que ha dotado a la humanidad de medios únicos y de vital importancia para la difusión y expansión del conocimiento, la cultura y la sabiduría. Esta situación se ha visto apoyada por el creciente auge de Internet, así como por el actual desarrollo de los productos informáticos cada vez más actualizados, potentes y con mayor calidad.

El uso de la TIC es considerado como una competencia básica, así como valiosa herramienta que facilita la comunicación, la búsqueda de información, promueve el intercambio cultural y con ello poder tener una completa alfabetización digital. Actualmente la industria del software a nivel mundial es cada vez más competente y culta. Por ello, Cuba en función de su inserción en el mercado internacional, ha destinado cuantiosos recursos, esfuerzos y tecnologías para agilizar y potenciar el desarrollo de las investigaciones en el campo de la informática. De esta forma la industria del software nacional tiene la tarea de liberar productos de una elevada calidad y eficiencia que le permita poder colocarse en los primeros peldaños a nivel mundial en el campo de la informática.

En muchas ocasiones, la calidad del software se ve afectada por deficiencias que afectan su correcto funcionamiento y dañan la imagen de aquella empresa que los produzca. Por ello se hace necesario detectar cada una de las deficiencias o No Conformidades<sup>2</sup> (como se les conoce en el ambiente informático profesional), mediante un riguroso proceso de pruebas por el que todo software debe incursionar antes de su liberación o despliegue.

Uno de los grandes avances de Cuba en la informática y la producción de software es la creación de la Universidad de las Ciencias Informáticas (UCI). Esta surge con la idea de vincular al estudiante en tareas productivas y formar un profesional desde el estudio intensivo y la producción. Diversas facultades crean

---

<sup>1</sup> Tecnologías de la Información y las Comunicaciones

<sup>2</sup> Errores detectados en algún software o sistema informático



software bajo diferentes perfiles de trabajo y logran que la UCI se inserte en el mercado internacional con excelentes resultados.

En la facultad 7, el Centro de Informática Médica (CESIM), asume el reto de la informatización del Sistema de Salud Pública para la cual tiene una serie de productos de software correspondiente a esta línea. Estos productos informáticos crecen en funcionalidad progresivamente por las necesidades de los clientes. Se trata de informatizar el sector de la salud en la sociedad cubana, como parte de un profundo proceso de transformación organizacional que manejan las instituciones médicas debido al auge de la informática.

Como estrategia, para la detección de errores en sus sistemas informáticos, el CESIM cuenta con un grupo de calidad. Una de las principales tareas realizadas dentro del Grupo de Calidad de la Facultad 7, es llevar a cabo el registro de todas las No Conformidades encontradas, durante el proceso de pruebas, a un software determinado. Este proceso se inicia cuando el probador encuentra una No Conformidad y se dispone a registrarla en una plantilla de registro de NC<sup>3</sup> para su posterior análisis y solución por parte del equipo de desarrollo.

Existen diferentes pasos a la hora de registrar la misma, inicialmente se necesita especificar la ubicación exacta de la NC, luego se describe en qué consiste esta, siendo lo más explícito posible, además se incluyen datos importantes como tipo de NC, fecha ocurrencia de la misma y número de iteración de las pruebas; y para finalizar se incluye un anexo (que se pudiera o no prescindir de él) con el objetivo de hacer más fácil de detectar el lugar exacto donde se ubica la NC y como prueba demostrativa y tangible de que realmente existe la misma, en la aplicación que se está probando. Todo este proceso finaliza parcialmente con la generación, por parte de cada uno de los Probadores, de un Registro de NC<sup>4</sup>.

---

<sup>3</sup> No Conformidad

<sup>4</sup> Documento digital que contiene todos los datos para identificar la existencia de una NC.

Actualmente el proceso de registro de NC presenta dificultades, ya que todos los datos del Registro de NC son llenados de forma manual. El principal peso del problema recae en el momento de especificar la ubicación de la NC de manera clara y explícita, pues el probador lo hace mediante un texto plano de forma libre, que en muchas ocasiones crea confusión e incongruencia cuando otro usuario trata de llegar hasta la ubicación exacta de la NC al interactuar con el documento.

A su vez está la disyuntiva de que existen dos tipos principales de ocurrencia de una NC: una es la que se produce en un lugar determinado de la aplicación y otra, cuando siguiendo una secuencia de pasos se logra que se produzca. Este problema trae como consecuencia, una mala planificación y estimación del proceso de pruebas, esfuerzo en vano, es decir planificación sobre la base de datos no reales, mala estimación de la duración del ciclo de pruebas y mala evaluación del trabajo del desarrollador.

Por estas cuestiones mencionadas anteriormente, es que se hace sumamente difícil depender solamente de un texto libre, que provea la información, ya sea de la ubicación o reproducción de la NC. Se hace necesario una forma nueva, un nuevo complemento que auxilie, un nuevo mecanismo; que posibilite una reproducción casi exacta, sin fallas, para que se realice una correcta reproducción controlada de las NC, factor de vital importancia dentro del ciclo de pruebas.

Dada la situación anterior, se tomó como **Problema a resolver**: ¿Cómo lograr reproducir de forma controlada las NC en las pruebas realizadas a sistemas web, para apoyar el proceso de pruebas, de las soluciones desarrolladas en el Centro de Informática Médica?

Este trabajo toma como **objeto de estudio**: El proceso de gestión de las NC en el Centro de Informática Médica. **El campo de acción** se enmarca principalmente en El proceso de reproducción de NC en las pruebas realizadas a sistemas web del CESIM.

Para dar solución al problema antes mencionado se propone como **objetivo general**: Desarrollar una solución informática para la reproducción controlada de No conformidades en las pruebas realizadas a sistemas web.

Para este fin se proponen las siguientes **tareas de la Investigación**:

- ✓ Realizar una evaluación del estado del arte en cuanto a existencia de herramientas de seguimiento de errores (bugtracking) en el ámbito nacional e internacional.
- ✓ Realizar análisis sobre la especificación de los pasos para la reproducción de las NC en las herramientas estudiadas.
- ✓ Definir una estructura para especificar el camino para la reproducción de las NC.
- ✓ Implementar las funciones para reproducir una NC de manera automática.
- ✓ Evaluar posibilidades de modificación del REDMINE.
- ✓ Desarrollar una extensión para el navegador Mozilla Firefox, capaz de reproducir de forma controlada las no conformidades.
- ✓ Realizar pruebas funcionales al sistema.

El presente trabajo cuenta de cuatro capítulos estructurados de la siguiente forma:

### **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

Se aborda el estado del arte del tema a tratar a nivel internacional y nacional, incluye una explicación de las técnicas, tecnologías, metodologías y herramientas empleadas en la investigación para darle solución al problema. Se tratan los conceptos fundamentales para una correcta comprensión del tema.

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

Se define las características del sistema y se enmarca en el objeto de estudio. Se hace un análisis del dominio de la aplicación, se describen los procesos a automatizar para darle solución al problema y se generan documentos confecciona o elabora la documentación correspondiente a esta fase. También se definen requisitos no funcionales y el prototipo de interfaz externa.

### **CAPÍTULO 3: DISEÑO DEL SISTEMA**

En el presente capítulo se describe el diseño del sistema donde se definen los diagramas de clases del diseño por casos de uso, así como la relación existente entre ellos. Posterior a esto se muestra la interacción entre los actores y el sistema mediante los diagramas de secuencia. Además se especifica la seguridad y el diseño de la interfaz de la aplicación.

#### **CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBAS**

Se detallan los aspectos fundamentales que definen la estructura de implementación del sistema. Se analizan los algoritmos más importantes. Se describen las clases que modelan la solución y sus características. Se explican las estructuras de datos empleadas para manejar la información. Se diseñan y ejecutan pruebas unitarias para comprobar los resultados obtenidos. Se analizan los resultados obtenidos hasta el momento y se llega a conclusiones de lo logrado en el trabajo. En dependencia de los resultados se proponen nuevas iteraciones y posibles cambios en el diseño.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En el presente capítulo, se explican los principales conceptos referentes al negocio en cuestión, se realiza un análisis de las diferentes herramientas, tecnologías y metodologías existentes para la detección de errores en general. Por último se hace un análisis de las tendencias, tecnologías y metodologías más usadas en la actualidad y se concluye con la selección de las herramientas, tecnologías y lenguajes a utilizar para el desarrollo de la aplicación.

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

**No Conformidades (NC):** Según la definición de ISO<sup>5</sup>9000: 2000 (3.6.2), una no conformidad es el “incumplimiento de un requisito”. O un error detectado en la documentación durante la revisión.

**Registro de No Conformidades (RNC):** Documento donde se registran todas las no conformidades que se le detectan al producto. (1)

**Calidad del software:** La calidad del software es definida como la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente.(2)

**Prueba de software:** Las pruebas de software, en inglés *testing* son los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador.(3)

**Proceso de pruebas:** El proceso de pruebas consiste en demostrar que los artefactos generados están libres de fallos". El proceso de pruebas consiste, básicamente, en la realización de una serie de pruebas al código, para observar con qué frecuencia podría fallar ese código, encontrar esos errores y solucionarlos.

---

<sup>5</sup> Organización Internacional para la Estandarización (International Organization for Standardization)

Estas pruebas hay que realizarlas al comienzo del ciclo de vida del software ya que, cuanto más avanza el ciclo de vida, más costoso sería solucionar el fallo encontrado.(4)

### 1.2. Sistemas automatizados existentes.

Se desee o no, en todo proyecto hay errores, estos deben ser detectados y solucionados debido a que atentan contra la calidad del producto. Generalmente el control de los errores en los proyectos se ha realizado mediante emails, documentos Word, hojas Excel, etc. En estos momentos se han realizado importantes mejoras en el proceso de seguimiento de errores, mostrándose estas mejoras en la creación de varias aplicaciones a nivel mundial que facilitan y agilizan la gestión de errores durante el proceso de aseguramiento de la calidad del software.

#### 1.2.1. *Ámbito internacional.*

##### **Bugzilla**

Con seguridad es el más conocido de todos los sistemas. Creado por Mozilla, permite tener un seguimiento de errores y cambios, comunicación entre los miembros del proyecto, enviar y revisar parches y manejar la garantía de calidad. Además, entre otras características, permite llevar un control de tiempo, distintas formas de autenticación, gráficos y multilinguaje.<sup>6</sup>, servidor web Apache, el cual se recomienda, módulos Perl específicos y Mail Transfer Agent.<sup>7</sup> Requiere Perl<sup>8</sup>, MySQL o PostgreSQL.

Características:

- ✓ Estructura de base de datos optimizada para un mayor rendimiento y escalabilidad.
- ✓ Excelente seguridad para proteger la confidencialidad.
- ✓ Avanzada herramienta de consultas que pueden recordar sus búsquedas.
- ✓ Capacidades de correo electrónico integrado.

---

<sup>6</sup> Gestores de Base de Datos

<sup>7</sup> Agente de Transferencia de Correo

<sup>8</sup> Lenguaje de programación web (Practical Extraction and Report Language)

- ✓ Editar los perfiles de usuario y preferencias de correo electrónico completa.
- ✓ Amplio sistema de permisos.
- ✓ Probado bajo el fuego de Mozilla como sistema de seguimiento de fallos.

### Beneficios:

- ✓ Mejorar la comunicación.
- ✓ Aumentar la calidad de los productos.
- ✓ Mejorar la satisfacción del cliente.
- ✓ Garantizar la rendición de cuentas.
- ✓ Aumentar la productividad.
- ✓ Bugzilla puede adaptarse a múltiples situaciones.(5)

### **BUGS - the Bug Genie**

Desarrollado por Ruffdogs, fácil de usar, "BUGS"<sup>9</sup> - Bug Genie es una empresa de código abierto del sistema de seguimiento en cuestión, el cual aumenta su proceso de desarrollo, ofreciendo una herramienta avanzada para la gestión de informes de errores, peticiones y comentarios de los usuarios de sus productos, seguro, con historial sobre los errores, mensajería privada, temas y multilinguaje, entre otras características.

### Beneficios

- ✓ Mejor gestión de usuarios con control de acceso detallado.
- ✓ La mejora de funcionalidad de búsqueda con apoyo de la agrupación.
- ✓ Mejora de la funcionalidad de mensajes con carpetas de búsqueda.
- ✓ Mejora de la presentación de informes en cuestión, asistente automatizado con duplicado de búsqueda.

---

<sup>9</sup> Se refiere a los fallos encontrados en un software.

- ✓ Artículos y noticias.
- ✓ La integración de Subversion (SVN)<sup>10</sup>.
- ✓ Calendario de eventos y reuniones.
- ✓ Carteleras mundiales y equipo.
- ✓ Generación automática de hoja de ruta.
- ✓ Apoyo a la cuestión sub-tareas.
- ✓ Gran apoyo lingüístico - BUGS 2 viene parcialmente traducido en varios idiomas. (6)

### GNATS

Proyecto GNU<sup>11</sup>, soporta distintas bases de datos, además existe una comunidad que desarrolla nuevas extensiones. GNU GNATS es un conjunto de herramientas para el seguimiento de los errores denunciados por los usuarios a un sitio central. Permite la gestión y el informe de problemas de comunicación con los usuarios a través de diversos medios. GNATS almacena toda la información sobre los informes de problemas en sus bases de datos y proporciona herramientas para la consulta, edición y mantenimiento de las bases de datos.

Gracias a su arquitectura, GNATS no está vinculada a una única interfaz de usuario - puede ser utilizado a través de línea de comandos, e-mail, Emacs<sup>12</sup>, o un dominio de red, por lo general se usa con una interfaz web. A esto se suma el hecho de que toda la configuración se puede almacenar en archivos de texto, permitiendo un fácil uso y proporcionando una buena flexibilidad.(7)

---

<sup>10</sup> Sistema de Control de Versiones.

<sup>11</sup> Proyecto iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre: GNU No es Unix (GNU isNot Unix)

<sup>12</sup> Editor de texto con varias funciones, muy popular entre los programadores y usuarios técnicos



### CodeTrack

La característica principal es que no necesita ni base de datos ni servidor de correo (algo que puede ser una ventaja dependiendo de los recursos que se puedan obtener en un proyecto). La información la guarda en archivos de tipo XML<sup>13</sup> y tan sólo es necesario tener instalado Apache y PHP<sup>14</sup>. De fácil instalación, funciona en navegadores antiguos y seguro, con características que hacen que sea una aplicación sencilla y útil para crear, revisar y gestionar los informes de error y respuestas de desarrolladores para múltiples proyectos de una interfaz común. Especialmente adecuado para entornos de intranet y extranet, CodeTrack incluye una fuerte autenticación, y permite el control de acceso personalizado a los proyectos individuales.

Se reportan beneficios del CodeTrack según el sitio Sourceforge.net tales como:

- ✓ CodeTrack es rápida: búsqueda de miles de errores en cuestión de segundos.
- ✓ Instalación Trivial- Se instala en menos de cinco minutos.
- ✓ Realiza un seguimiento de las cuestiones a través de múltiples proyectos.
- ✓ Completa la historia y la auditoría de error por defecto.
- ✓ Fuerte autenticación fuera de la caja.
- ✓ Soporte completo para navegadores más antiguos (como Netscape 4.x).
- ✓ No hay base de datos necesarios para el backend<sup>15</sup>.
- ✓ No se necesita servidor de correo.
- ✓ No hay necesidad de ocultar las bibliotecas compartidas para extender su sistema a través de los directorios.
- ✓ Permite restringir el acceso según el usuario o papel, por sistema o por proyecto.
- ✓ La aplicación entera es un solo programa PHP.(8)

---

<sup>13</sup> Lenguaje de Marcas Ampliable (eXtensible Markup Language)

<sup>14</sup> Acrónimo recursivo: PHP Hypertext Pre-processor (inicialmente Personal Home Page Tools)

<sup>15</sup> Término usado para definir el final de un proceso

### **Mantis**

Desarrollado en PHP, siendo necesario MySQL y un servidor web (Apache a ser posible), este bug tracker<sup>16</sup> tiene entre sus características el ser fácil de instalar, permitir varios proyectos, varios lenguajes, emails, búsquedas y filtros.

MantisBT es un servicio gratuito basado en la web. Está escrito en el lenguaje de scripting PHP, y trabaja con MySQL, MS SQL<sup>17</sup>, bases de datos PostgreSQL y un servidor de web. MantisBT se ha instalado en Windows, Linux, Mac OS, OS / 2, y otros. Casi cualquier navegador debería ser capaz de funcionar como un cliente. Es liberado bajo los términos de la GNU (GPL)<sup>18</sup>. (9)

### **Bug-a-Boo**

Solo para Linux, permite temas y está disponible en inglés y alemán. Bug-A-Boo es una web basada en el sistema de reporte de errores para Linux. Es liberado bajo los términos de la GNU (GPL). Funciona en cualquier servidor web, proporciona funcionalidad CGI<sup>19</sup>. Bug-A-Boo no necesita ningún servidor de base de datos, pero trae sus propias tablas locales, se accede con Tdbengine. Por lo tanto, es fácil de configurar y un verdadero monstruo de rendimiento. Cuenta con múltiples idiomas, tres diferentes niveles de usuario, diseños y temas, las notificaciones por correo electrónico y los perfiles de visualización para todos los usuarios. (10)

---

<sup>16</sup> Término usado para definir algunas herramientas usadas en la gestión de errores

<sup>17</sup> Siglas para definir el gestor de base de datos Microsoft SQL

<sup>18</sup> General PublicLicense

<sup>19</sup> Interfaz de entrada común (Common Gateway Interface)

### Zentrack

También diseñado para ayuda de helpdesk<sup>20</sup>, con una interfaz bastante configurable, permite controlar el flujo de trabajo y el manejo de los datos, control de los tiempos y de las funcionalidades para los usuarios y lista de contactos.

Se reportan beneficios del Zentrack según el sitio Zentrack.net tales como:

- ✓ Campos personalizados - agrega sus propias fechas, números, cadenas.
- ✓ Comportamientos - dependencias complejas y listas.
- ✓ Realiza un seguimiento de horas, las estimaciones, y los plazos.
- ✓ Interfaces y adaptación.
- ✓ Pasarela de correo electrónico para crear y responder a las entradas.
- ✓ Recordatorios de artículos atrasados.
- ✓ Usabilidad y la interfaz.
- ✓ Aspecto personalizable.
- ✓ Control de acceso en la pantalla por pantalla, campo por campo base.
- ✓ Atajos de teclado.
- ✓ Diseño compacto.(11)

### Trac

Se integra con SVN y permite navegar por el código, abrir tickets y hacer una buena gestión del proyecto. Escrito en Python, admite las bases de datos SQLite, PostgreSQL y MySQL. Trac utiliza un enfoque minimalista basado en la web de software de gestión de proyectos. Su misión es ayudar a los desarrolladores a escribir software de gran mantenimiento durante el camino que acompaña la construcción

---

<sup>20</sup> Término para definir un sistema de ayuda brindado por las organizaciones a sus usuarios

del software. Trac debe imponer lo menos posible sus prácticas en un equipo establecido en proceso de desarrollo y políticas. Proporciona un interfaz para Subversion.(12)

### 1.2.2. *Ámbito Nacional.*

Actualmente en la UCI, al igual que en el resto de las instituciones que desarrollan software en el país, se realizan acciones para incidir en la mejora del proceso de detección y corrección de errores de los productos informáticos. Para ello se comienzan a utilizar algunas herramientas de las antes mencionadas. A su vez se han llevado a cabo algunas investigaciones dirigidas fundamentalmente a buscar otras soluciones que se adapten a las características particulares de los procesos que se llevan a cabo en la Universidad.

### **Diseño de una aplicación para el Seguimiento de Errores de los productos software de la Facultad 7**

El objetivo del trabajo se enmarca en diseñar un sistema informático para la automatización del seguimiento de errores, que favorezca el proceso de control de calidad de los productos de software desarrollados en la Facultad 7. Lo cual facilite el trabajo de los miembros del Proyecto Calidad de la Facultad 7 en cuanto a la detección y seguimiento de errores, tanto del software como de la documentación generada durante el proceso de desarrollo, además de brindar facilidades en la elaboración de los documentos legales que se generan en el proceso de revisión. Culminó con la propuesta de desarrollo de un sistema informático, el cual se enmarca en la implementación del proceso correspondiente al Proceso de Pruebas a la Documentación. (13)

### **Sistema para la Gestión de las No Conformidades durante las Pruebas de Calidad**

La investigación tiene como objetivos la modelación e implementación de un sistema que posibilite obtener reportes a partir de la información contenida en los Registros de No Conformidades, conocer en qué estado se encuentran los indicadores de calidad que se miden en el proyecto y gestionar el acceso a la información de los usuarios que harán uso de la misma. Los principales requerimientos del sistema son los siguientes:

- ✓ Gestión de Usuarios y sus roles en el Sistema.
- ✓ Gestión de Reportes basados en los Registros de No Conformidades.
- ✓ Gestión de No Conformidades.(14)

### 1.3. Análisis crítico

Una vez realizado el estudio de las herramientas expuestas se llega a la conclusión de que muchas de ellas son propietarias y los costes de soporte técnico y mantenimiento son muy elevados. Además algunas son de licencia gratis y código libre, pero ninguna de ellas responde a las necesidades que se requiere para darle solución a dicho problema. Para darle solución al mismo es necesario desarrollar una solución informática que sea capaz de suplir dichas necesidades, siendo factible realizar un estudio de las herramientas y tecnologías para su elaboración.

### 1.4. Herramientas y tecnologías

#### 1.4.1. Lenguaje de programación

En el mundo del desarrollo del software muchos programadores usan lenguajes de programación de alto nivel y orientados a objetos; la decisión del uso de uno u otro para el desarrollo de un software determinado está en las librerías que estos utilizan, en las cuales radica la verdadera riqueza del lenguaje. La elección final de uno u otro lenguaje también dependerá del posible conocimiento que se tenga de la sintaxis del mismo y de su adaptación al medio para el que se desee programar, ya que es diferente programar para red local, para Windows, Mac o Linux. A continuación se presentarán algunas características, ventajas y desventajas del lenguaje de programación seleccionado para trabajar.

#### **El lenguaje Java**

Este es un lenguaje desarrollado por la compañía Sun Microsystem en los años noventa. Está inspirado en C++ y se proyectó con la finalidad de obtener un producto de pequeñas dimensiones, simple y portátil sobre diferentes plataformas y sistemas operativos ya sea a nivel de código fuente como a nivel de código binario.

Los compiladores de Java generan archivos de código binario especial (Java bytecode) que es un lenguaje de máquina a bajo nivel y puede incluso ser interpretado directamente por un procesador.

Las aplicaciones desarrolladas en Java se deben ejecutar en la Máquina Virtual de Java, en inglés Java Virtual Machine (JVM). Este es un programa que se puede ejecutar sobre varios sistemas operativos y su función es servir de puente o mediador que puede entender tanto el Java bytecode como el sistema sobre el

que se quiere ejecutar el mismo. Permitiendo así la portabilidad de programas escritos en este lenguaje. JVM posee instrucciones para los diferentes tipos de tareas:

- ✓ Carga y almacenamiento.
- ✓ Operaciones aritméticas.
- ✓ Conversiones de tipos.
- ✓ Creación y manipulación de objetos.
- ✓ Gestión de pilas con métodos como *Push* y *Pop*.
- ✓ Transferencias de control de programa (*branching*).
- ✓ Invocación y retorno a métodos.
- ✓ Lanzamiento de excepciones.

### Características de Java

- ✓ Ligero.
- ✓ Multiplataforma.
- ✓ Código libre.
- ✓ Simple y poderoso.
- ✓ Seguro.
- ✓ Orientado por Objetos.
- ✓ Robusto.
- ✓ Interactivo.
- ✓ Independiente de arquitectura de hardware.
- ✓ Interpretado y rápido.
- ✓ Fácil de aprender.
- ✓ Herramientas poderosas: threads, excepciones, APIs, RMI, etc.

Ventajas de Java:

- ✓ Es una fuente abierta, así que los usuarios no tienen que luchar con los impuestos sobre patente pesados cada año.
- ✓ Independiente de la plataforma.
- ✓ El poder de Java API sea alcanzada fácilmente por los reveladores.
- ✓ Java realiza la colección de basura de las ayudas, así que la gerencia de memoria es automática
- ✓ Java asigna siempre objetos en el apilado.
- ✓ Usando JAVA se puede desarrollar aplicaciones web dinámicas.
- ✓ Permite que usted cree programas modulares y códigos reutilizables.

Entre sus principales características se encuentra que es un lenguaje de propósito general y orientado a objetos. Su sintaxis ha sido trabajada mejorando la de C++ logrando mayor sencillez y legibilidad. Presenta mayor robustez al simplificar la gestión de memoria y eliminar las complejidades del manejo explícito de punteros. Presenta capacidades avanzadas de ejecución multi-hilo y proporciona mecanismos de carga dinámica de clases en tiempo de ejecución. Se puede compilar y ejecutar en cualquier plataforma de sistema operativo por ejemplo en Windows, Solaris o Linux gracias a su máquina virtual. Su desarrollo ha sido rápido y exitoso debido a la gran cantidad de grandes empresas colaboradoras que han dado su aporte para enriquecerlo.

Es un lenguaje gratuito ya que Sun Microsystems distribuye gratuitamente su producto base, el J2SE<sup>21</sup>. Este es un paquete que incluye herramientas para generar programas Java con depurador, compilador y herramienta para la documentación de código. Además posee la JVM<sup>22</sup>, la jerarquía de clases (API de java) con su código fuente y la documentación necesaria.

---

<sup>21</sup> En Inglés Java 2 Standard Edition

<sup>22</sup> En inglés Java Virtual Machine (Máquina virtual de java)

La opinión de la mayoría de los programadores es que se torna más potente en el desarrollo de aplicaciones Web y ha sido mayormente orientado en ese sentido por sus promotores.

La ejecución de programas escritos en Java suele comportarse más lenta que la de aplicaciones de otro lenguaje haciendo un uso voraz de recursos como memoria y procesador. Esto se hace más notorio si la ejecución se basa en cálculos matemáticos complejos o si la aplicación presenta un diseño cargado de componentes visuales.(15)

### **El lenguaje C++**

El lenguaje C++ se comenzó a desarrollar en 1980 por Bjarne Stroustrup de la AT&T primero con el nombre de "C with classes" debido a que su objetivo era mejorar algunas características del C pero manteniendo su basamento en el mismo. Luego, en 1983 adquiere el nombre de "C++" debido a su esencia de C y con mejoras como su operador de incremento numeral (++). Este se estandarizó hacia 1989 por sendos comités de la ANSI y la ISO y se enriqueció con la aparición de la STL desarrollada por Alexander Stepanov y publicada en 1994.

Es un lenguaje de alto nivel pero manteniendo la posibilidad de realizar operaciones a nivel de máquina. Es imperativo, orientado a objetos y de propósito general al que se le han añadido nuevos tipos de datos, clases, plantillas, mecanismo de excepciones, sistema de espacio de nombres, funciones *inline*, sobrecarga de operadores, referencias para el manejo de punteros, manejo de memoria persistente y algunas utilidades adicionales de su librería estándar. Simplifica y enriquece el uso de comentarios, la declaración de enumeradores, la declaración de variables y la declaración de estructuras.

Es versátil, flexible, conciso y muy eficiente. Por muchos años fue el preferido en el desarrollo de aplicaciones. Se ha utilizado para implementar el núcleo de sistemas como Windows y Java.(16)

### **El lenguaje XML**

Es un lenguaje de etiquetado extensible muy simple, pero estricto, que juega un papel fundamental en el intercambio de una gran variedad de datos. Sirve para estructurar, almacenar e intercambiar información. Es la base de los servicios Web, el contenido almacenado en un documento XML se puede transferir fácilmente a través de la red. Los servicios Web XML actúan de forma independiente y además permiten que las aplicaciones compartan información e invoquen funciones de otras aplicaciones independientemente del



sistema operativo o la plataforma en que se ejecutan y los dispositivos utilizados para obtener acceso a ellos.

### Características de XML

XML impone una sintaxis más rígida para las marcas, que permite su proceso de forma más eficiente. En XML, las marcas de término no pueden ser omitidas (a diferencia de la marca P en HTML, por ejemplo). Marcas sin contenido, como IMG o BR en HTML, terminan con un /> para indicar que allí acaban. XML también distingue entre minúsculas y mayúsculas. También, cualquier valor de un atributo en una marca debe ir entre comillas (es decir, no se pueden omitir). Esto significa que interpretar XML sin conocer el conjunto válido de marcas es mucho más sencillo. En particular, definir entonces el tipo del documento (lo que en SGML y XML se llama DTD de *document type declaration*) no es obligatorio.

En este caso, las marcas se obtienen a medida que se interpreta el documento. XML permite definir lenguajes de marcas para cualquier fin y tiene capacidades de validación de datos. La figura adjunta muestra un ejemplo de un documento estructurado con XML usando un conjunto de marcas definido para correo electrónico y sin usar un DTD. El no usar un DTD tiene también sus desventajas, ya que en ese caso la falta de información dificulta tener software automático que apoye la generación de documentos (por ejemplo, un editor de XML similar a los editores de HTML).

### Usos del XML

Actualmente XML está siendo usado para muchos fines diferentes. La lista a continuación son sólo algunos de los más importantes:

- ✓ XSL: el eXtensible Style sheet Lenguaje es la contraparte de XML a CSS (Cascading Style Sheets), el lenguaje de estilo que permite independizar cómo se ve un documento, de su contenido y estructuración semántica.
- ✓ XLL: el eXtensible Link Language permite definir distintos tipos de enlaces (links) entre documentos, ya sean externos o internos.
- ✓ RDF: el ResourceDescription Framework es un estándar de la Web para proveer interoperabilidad entre aplicaciones y descripciones de metadatos.

- ✓ MathML: dos conjuntos de marcas definidas usando XML para presentar fórmulas y expresar la semántica de expresiones matemáticas.
- ✓ SMIL: el Synchronized Multimedia Integration Languages un lenguaje para sincronizar presentaciones multimediales en la Web, donde la posición y tiempo de activación de distintos objetos puede ser especificada.(17)

### El lenguaje XUL

Es el lenguaje XML para interfaces de usuario de Mozilla. Te permite crear potentes aplicaciones multiplataforma que pueden ejecutarse con conexión a Internet o sin ella. Estas aplicaciones son fácilmente personalizables con texto alternativo, gráficos, y diseños por lo que pueden ser fácilmente instalados o traducidos para diversos mercados. Los desarrolladores web que estén familiarizados con HTML Dinámico (DHTML) pueden aprender XUL fácilmente y empezar a crear aplicaciones ya.

¿Qué es XUL y por qué fue creado?

XUL fue creado para facilitar y acelerar el desarrollo del navegador Mozilla. Es un lenguaje XML, por lo tanto todas las características del XML están también en XUL.

La mayoría de las aplicaciones necesitan ser desarrolladas usando características de una plataforma específica, convirtiendo el desarrollo de aplicaciones multiplataforma en una tarea costosa y que consume mucho tiempo. Esto podrá no importarles a algunas personas, pero el usuario podría querer usar la aplicación en otros dispositivos, como equipos de bolsillo. En el pasado se desarrollaron muchas soluciones multiplataforma. Por ejemplo, Java tiene como un punto fuerte de venta la soportabilidad. XUL es un lenguaje diseñado específicamente para construir interfaces soportables.

Desarrollar una aplicación toma mucho tiempo incluso si es sólo para una plataforma. El tiempo requerido para compilar y depurar también puede ser muy largo. Con XUL, una interfaz puede ser implementada y modificada de forma fácil y rápida. XUL tiene todas las ventajas de otros lenguajes XML. Por ejemplo, XHTML u otros lenguajes XML como MathML o SVG se pueden usar junto a él. Además, el texto que se muestra con XUL es fácilmente localizable, lo que significa que puede ser traducido a otros idiomas con poco esfuerzo. Se pueden aplicar hojas de estilo para modificar la apariencia de la interfaz de usuario (muy parecido a las funcionalidades de aspectos o temas presentes en Winamp o algunos gestores de ventanas).

¿Qué tipo de interfaz se puede hacer con XUL?

XUL proporciona la habilidad de crear la mayoría de los elementos encontrados en las interfaces gráficas modernas. Es tan general que este puede ser aplicado a las necesidades específicas de ciertos dispositivos y tan poderoso que los desarrolladores pueden crear sofisticadas interfaces con éste.

Algunos elementos que pueden ser creados son:

- ✓ Controles de entrada tales como cuadros de texto y cajas de chequeo
- ✓ Barra de herramientas con botones u otros contenidos
- ✓ Menús en barras de menú o menú emergente
- ✓ Pestañas de diálogo
- ✓ Árbol de información jerárquica o tabulada
- ✓ Teclas de accesos directo

El contenido mostrado puede ser creado desde el contenido de un archivo XUL o con datos de una fuente de datos. En Mozilla, tales fuentes de datos son utilizadas para los mensajes de una cuenta de correo, los marca-páginas y los resultados de búsqueda. El contenido de los menús, árboles y otros elementos pueden ser llenados con estos datos, o con sus propios datos suministrados en un archivo.

Existen varios casos en los cuales aplicaciones XUL son creadas:

- ✓ Extensión de Firefox: una extensión suma funcionalidades al navegador mismo, a menudo barra de herramientas suplementaria, menús contextuales o una interfaz de personalización del navegador. Para conseguir eso una funcionalidad XUL llamada overlay usada. Permite hacer fusionar una interfaz existente como el navegador Firefox con una interfaz de usuario de la extensión. Unas extensiones pueden igualmente ser aplicadas a otros productos de Mozilla tal como Thunderbird.
- ✓ Aplicación XULRunner autónoma: XULRunner es una versión empaquetada de una plataforma Mozilla permitiendo crear aplicaciones XUL autónomas. No se necesita ningún navegador para arrancar éstas aplicaciones ya que contienen sus propios archivos ejecutables.
- ✓ Paquete de XUL: a medio camino entre las dos otras se encuentran aplicaciones creadas de la misma manera que una extensión pero funcionando como una aplicación independiente en una ventana separada. Estos paquetes se emplean cuando no se desea una imponente aplicación completa XULRunner pero tampoco implica la instalación de un navegador Mozilla.

Aplicación XUL remota: se puede también colocar solamente código XUL en un servidor web y abrirlo con un navegador, como cualquier página web. Sin embargo este método es limitado, por razones de seguridad sobre qué acciones se permiten, como la apertura de otras ventanas.(18)

### El lenguaje JavaScript

JavaScript es un lenguaje interpretado que permite incluir macros en páginas Web. Estas macros se ejecutan en el ordenador del visitante de las páginas, y no en el servidor. JavaScript es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

JavaScript proporciona los medios para:

- ✓ Controlar las ventanas del navegador y el contenido que muestran.
- ✓ Programar páginas dinámicas simples.
- ✓ Evitar depender del servidor Web para cálculos sencillos.
- ✓ Capturar los eventos generados por el usuario y responder a ellos sin salir a Internet.
- ✓ Simular el comportamiento de las macros CGI cuando no es posible usarlas.
- ✓ Comprobar los datos que el usuario introduce en un formulario antes de enviarlos.

Comunicarse con el usuario mediante diversos métodos.

La característica de JavaScript que más simplifica la programación es que, aunque el lenguaje soporta cuatro tipos de datos, no es necesario declarar el tipo de las variables, argumentos de funciones ni valores de retorno de las funciones. El tipo de las variables cambia implícitamente cuando es necesario, lo que dificulta el desarrollo de programas complejos, pero ayuda a programar con rapidez macros sencillas. En esto, JavaScript se separa totalmente de lenguajes como C, C++ o Java.

JavaScript ha sido inventado por Netscape, que comenzó a ofrecerlo como parte de su Navigator v.2.0. El nombre original de JavaScript fue LiveScript, pero se modificó en el último momento, aparentemente para aprovechar el tirón de JAVA. Al ser código interpretado, JavaScript es más lento que Java, pero en la práctica no suele ser un factor de importancia.(19)

### *1.4.2. Selección de un lenguaje de programación*

Finalmente se ha comprobado que es más factible usar el lenguaje JavaScript, ya que el lenguaje C++ es muy potente y estable pero complicado para algunos propósitos. C# es muy similar al Java y su sintaxis es amigable pero eventualmente es menos eficiente en la ejecución de extensiones para Mozilla. Además, JavaScript aporta características novedosas que simplifican grandemente las labores de implementación y con la combinación de este con el XUL para la creación de la interfaz de usuario de la extensión, que se desea crear, se puede apreciar que son los lenguajes idóneos.

Ha alcanzado gran auge y se utiliza para desarrollar gran cantidad de aplicaciones desde mediano a gran tamaño, obviando esfuerzo de programadores en actividades de rutina. Está debidamente estandarizado. Se usa en varios departamentos productivos en la facultad 7 y en la UCI. Es gratis tanto el JavaScript como la plataforma que lo soporta. Resulta, por tanto, el adecuado para las aspiraciones y necesidades de este trabajo.

### *1.4.3. Entorno de desarrollo integrado*

#### **NetBeans**

NetBeans IDE es un entorno de desarrollo – una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java – pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

También está disponible NetBeans Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones. El NetBeans IDE 6.8 contiene las herramientas para que los desarrolladores de software puedan crear

aplicaciones desktop, Enterprise, web, y aplicaciones móviles, con el lenguaje Java, así como también C/C++, PHP, Java Script, Groove, and Ruby.

Lo nuevo que trae NetBeans IDE 6.8, el primero que soporta Java EE 6 platform.

Características:

- ✓ Mejoras en el editor de código.
- ✓ Soporte para Ruby, JRuby, y Ruby on Rails.
- ✓ Instalación y actualización más simple.
- ✓ Enlazar datos con el Swing GUI.
- ✓ Profiling integrado, profiling “points”.
- ✓ Características visuales para el desarrollo web.
- ✓ Creador gráfico de juegos para celulares.
- ✓ Mejoras para SOA y UML.
- ✓ Soporte para PHP.

### **Microsoft Visual Studio:**

Es un IDE para sistemas Windows desarrollado por Microsoft Corporation. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio 2005 permite a los desarrolladores crear sitios, aplicaciones y servicios web y aplicaciones de escritorio en cualquier entorno que soporte la plataforma. Sobre el mismo se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

Los programadores de Windows encuentran los nuevos formularios *Windows Forms* intuitivos y muy eficaces. Estos son compatibles con cualquier lenguaje basado en .NET. Con la herencia visual, los programadores pueden simplificar enormemente la creación de aplicaciones basadas en Windows, centralizando en formularios primarios la lógica común y la interfaz de usuario para toda la solución. Utilizando delimitadores y acoplamiento, los programadores pueden generar formularios redimensionables

automáticamente, mientras el editor de menús permite crear menús de manera visual directamente desde el *Diseñador de Windows Forms*.

Visual Studio .NET continúa siendo el punto de referencia para la productividad de los programadores. Con un único entorno de programación (IDE) integrado para todos los lenguajes, las organizaciones de programación pueden aprovechar las ventajas de un cuadro de herramientas, un depurador y una ventana de tareas comunes, reduciendo enormemente la curva de aprendizaje del programador y garantizado que siempre puedan elegir el lenguaje más apropiado para sus tareas y conocimientos. (20)

Con la función para completar instrucciones de *IntelliSense* y la comprobación automática de errores de sintaxis, Visual Studio .NET informa a los programadores cuándo el código es incorrecto y proporciona el dominio inmediato de las jerarquías de clases y las API. Con el Explorador de soluciones, los programadores pueden reutilizar fácilmente código a través de diferentes proyectos e incluso, generar soluciones multilenguaje que satisfagan con mayor eficacia sus necesidades empresariales.

Gracias al entorno totalmente extensible, se puede disfrutar de las ventajas de una activa comunidad de componentes y complementos de otros fabricantes, con componentes y controles que contribuyen a personalizar y ampliar el entorno de acuerdo con las necesidades.

Los asistentes para aplicaciones, las plantillas de proyecto y los ejemplos de código fuente de Visual Studio .NET permiten crear aplicaciones con rapidez para Windows, la Web y dispositivos con una inversión inicial mínima. La ayuda dinámica y Microsoft Developer Network (MSDN®) proporcionan asistencia basada en la tarea y el lenguaje de programación actuales, garantizado que los programadores sepan aprovechar las oportunidades de la plataforma Microsoft .NET en el lenguaje que hayan elegido. Las macros de Visual Studio, similares a las macros de Microsoft Visual Basic para Aplicaciones en Office, permiten la automatización de tareas rutinarias dentro del IDE, mejorando aún más la productividad global de los programadores de Visual Studio.

**La versión Visual Studio 2005 se puede adquirir en las siguientes ediciones:**

- ✓ Visual Studio 2005 Team System: Para grandes equipos de desarrollo.
- ✓ Visual Studio 2005 Professional: Para programadores independientes o pequeños grupos de programadores con múltiples propósitos.

- ✓ Visual Studio 2005 Standard y Visual Studio 2005 Express: Ediciones más simples para usuarios que se inician y estudiantes principalmente sin grandes propósitos.

Los precios de compra de las mismas oscilan entre los \$300 dólares y los \$1 400 dólares desde las ediciones Express y Standard hasta la TeamSystem en dependencia del distribuidor, los elementos que incluya el paquete y otros términos a tener en cuenta.

### **Komodo IDE 4.2**

Es un IDE potente y multiplataforma en varios idiomas de principio a fin, para el desarrollo de aplicaciones web dinámicas. Komodo IDE hace que la creación de sólidas aplicaciones Web rápido y fácil, con un rico conjunto de características para el lado del cliente Ajax, tecnologías como CSS, HTML, JavaScript y XML, junto con la dinámica de soporte avanzado para idiomas como Perl, PHP, Python, Ruby y Tcl. Es un entorno de desarrollo para programadores de una amplia gama de lenguajes: PHP, Perl, Python, Ruby, Javascript, HTML.

Aquí están algunas características dominantes de “Komodo IDE”:

#### **Redactor multilingüe:**

- ✓ Tecnologías: CSS, HTML, JavaScript y XML.
- ✓ Idiomas del lado del Servidor: Perl, PHP, pitón, rubí y Tcl.
- ✓ Idiomas de plantillas de Web: RHTML, HTML y Django.

#### **Auto completamiento:**

- ✓ Escribir el código más rápidamente.
- ✓ CSS, HTML, JavaScript, Perl, PHP, pitón, rubí, Tcl, XML y XSLT.
- ✓ Terminación Esquema-basada de XML/HTML.
- ✓ Ayuda del archivo de la Múltiple-lengua, tal como CSS y terminación del JavaScript en el HTML.
- ✓ Ayuda para agregar bibliotecas de tercera persona.
- ✓ Diferenciación de la versión del intérprete de la información de biblioteca incorporada y estándar.



### Ayuda multilingüe del archivo:

- ✓ Coloración correcta de la sintaxis de archivos multilingües y de las plantillas.
- ✓ Idiomas definidos por el usuario o UDL, usados para proporcionar la ayuda para RHTML, Plantilla-Juego de herramientas, HTML y Django). (21)

#### 1.4.4. Selección de un entorno de desarrollo

La herramienta a desarrollar es una aplicación típica de escritorio, es una extensión para el navegador Mozilla Firefox, no necesita conexiones a bases de datos. Su función se basa en procesar información entrada por el usuario y realizar tareas orientadas a las pruebas de software, así como y generar un archivo XML con los resultados.

De la investigación realizada se llegó a la conclusión de que para estos propósitos era factible utilizar Komodo IDE 4.2.0 porque brinda las posibilidades que se requieren para desarrollar un programa de este tipo. Su descarga desde Internet es gratuita y se instala de forma rápida, y con moderados requerimientos de hardware para su correcto funcionamiento. Es libre y bastante amigable en cuanto a ambiente de trabajo y es perfectamente compatible con lenguajes de programación como Javascript, XUL y XML, elementos claves para el desarrollo de la extensión.

De esta forma, una vez analizadas las ventajas y desventajas de todas las posibles combinaciones de lenguaje – plataforma – IDE. Se decidió realizar la implementación mediante el conjunto formado por los lenguajes JavaScript, XML y XUL y Komodo como IDE de desarrollo.

#### 1.4.5. Herramienta Case

### Visual Paradigm 6.4

Poderosa herramienta CASE que al igual que el Rational Rose utiliza UML para el modelado, es la herramienta por excelencia para ser utilizada en un ambiente de software libre. Permite crear tipos diferentes de diagramas en un ambiente totalmente visual.

Posibilita la representación gráfica de los diagramas permitiendo ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clase, actividad, estado, entre otros. Además, identifica requisitos y comunica información, se centra en cómo los componentes del sistema

interactúan entre ellos, sin entrar en detalles excesivos, además, permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico.

Tiene disponible distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community. Facilita licencias especiales para fines académicos.

Entre sus características más significativas se puede resaltar que su licencia es gratuita, posee varios idiomas, sus ediciones son compatibles y es de fácil uso para la creación de aplicaciones web.(22)

### **Enterprise Architect 7.1**

Enterprise Architect (EA) es una herramienta CASE<sup>23</sup> para el diseño y construcción de sistemas de software. EA soporta la especificación de UML 2.0, que describe un lenguaje visual por el cual se pueden definir mapas o modelos de un proyecto.

EA es una herramienta progresiva que cubre todos los aspectos del ciclo de desarrollo, proporcionando una trazabilidad completa desde la fase inicial del diseño a través del despliegue y mantenimiento. También provee soporte para pruebas, mantenimiento y control de cambio.

*Algunas de las características claves de Enterprise Architect son:*

- ✓ Generar código para el software que está construyendo.
- ✓ Crear elementos del modelo UML para un amplio alcance de objetivos.
- ✓ Ubicar esos elementos en diagramas y paquetes.
- ✓ Crear conectores entre elementos.
- ✓ Documentar los elementos que ha creado.
- ✓ Realizar ingeniería reversa del código existente en varios lenguajes.

---

<sup>23</sup> Ingeniería de software asistida por computadoras (Computer Aided Software Engineering).

- ✓ Importación/Exportación XMI 2.1.<sup>24</sup>
- ✓ Modelado basado en el Equipo
- ✓ Archivos compatibles o modelos basados del repositorio.
- ✓ Control de versiones con cualquier herramienta SCC.<sup>25</sup>
- ✓ Seguridad incorporada y administración de permisos.

Usando EA, puede realizar ingeniería directa y reversa de código C++, C#, Delphi, Java, Python, PHP, VB.NET y clases de Visual Basic, sincronizar códigos y elementos del modelo, diseñar y generar elementos de base de datos. La documentación de alta calidad puede ser rápidamente exportada desde sus modelos en industria estándar .formato RTF<sup>26</sup> e importar a Word para una personalización y presentación final.

Enterprise Architect sustenta todos los diagramas y modelos UML. Puede modelar procesos de negocio, sitios web, interfaces de usuario, redes, configuraciones de hardware, mensajes y más. Estimar el tamaño de su proyecto en esfuerzo de trabajo en horas. Capturar y trazar requisitos, recursos, planes de prueba, solicitudes de cambio y defectos. Desde los conceptos iniciales hasta el mantenimiento y soporte, Enterprise Architect tiene las características que precisa para diseñar y administrar su desarrollo e implementación.

(23)

#### 1.4.6. Lenguaje de Modelado

### UML 2.0

UML<sup>27</sup>, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (*Object Management Group*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos

---

<sup>24</sup> Importación/Exportación.

<sup>25</sup> Herramienta utilizada para el control de versiones.

<sup>26</sup> Formato de texto enriquecido (Rich Text Format).

<sup>27</sup> Unified Modeling Language

concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir.(24)

### 1.5. Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Pueden ser comparadas con un plan de contingencias en el que se va indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además quienes deben participar en el desarrollo de las actividades y qué papel deben de tener. Detallan además la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

#### 1.5.1. RUP<sup>28</sup>

Constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos, basado íntegramente en Lenguaje Unificado de Modelado (UML) como soporte a la metodología. El RUP es un producto de Rational (IBM<sup>29</sup>). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Define cuatro fases: Inicio, Elaboración, Construcción y Transición. Y nueve flujos de trabajo, seis de Ingeniería (Modelado del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue) y tres de apoyo (Gestión de la Configuración, Gestión de Proyecto y Ambiente). RUP pretende implementar las mejores prácticas actuales en Ingeniería de Software:

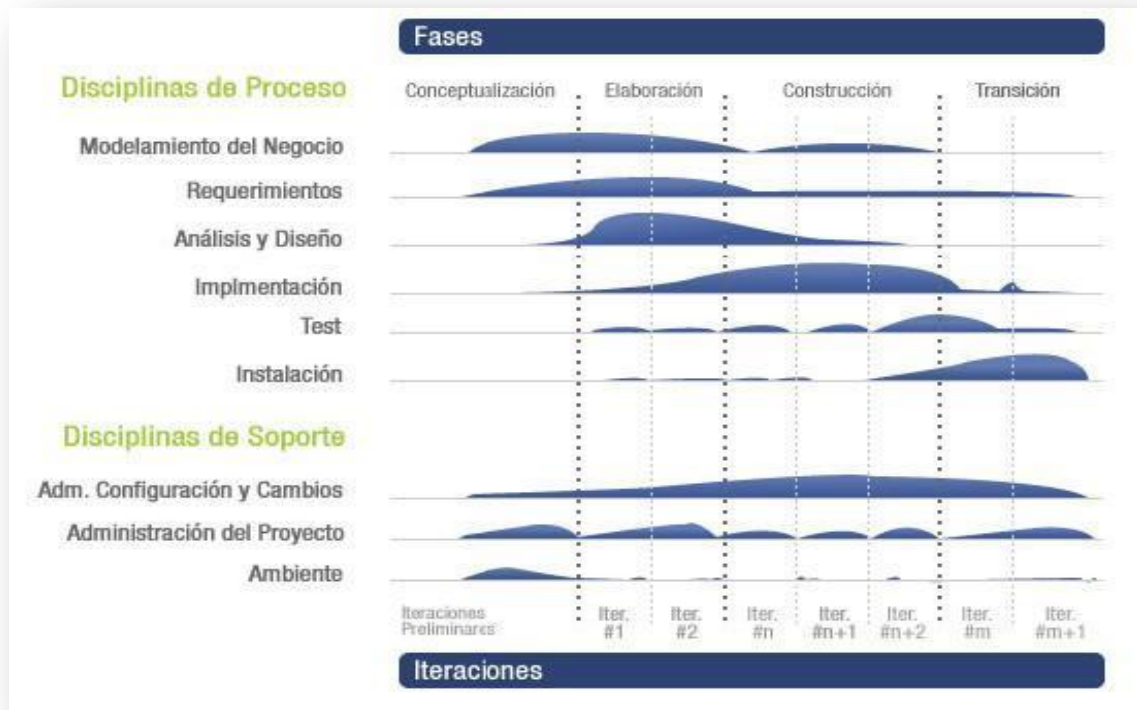
- ✓ Desarrollo iterativo del software.
- ✓ Administración de requerimientos.
- ✓ Uso de arquitecturas basadas en componentes.

---

<sup>28</sup> Proceso Unificado Racional (Rational Unified Process).

<sup>29</sup>International Business Machines.

- ✓ Modelación visual del software.
- ✓ Verificación de la calidad del software.
- ✓ Control de cambios.



**Ilustración 1: Fases e Iteraciones de la Metodología RUP.**

### 1.5.2. XP (Programación Extrema)

La Programación Extrema (XP), mejor conocida por su nombre en inglés Extreme Programming (XP), es una de las llamadas Metodologías Ágiles de desarrollo de software más exitosa de los tiempos recientes. Esta se basa en la simplicidad, la comunicación y la reutilización del código desarrollado. Es utilizada en proyectos con equipos de desarrollo pequeños y con plazos de entrega corto, de manera que se pueda llevar a cabo una programación rápida o extrema. Una particularidad que tiene, es que el usuario final debe ser miembro del equipo.

Esta metodología tiene las siguientes características.

**Pruebas unitarias:** Estas se basan en la realización de pruebas a los principales procesos, de manera que se adelante en algo hacia el futuro. Se pueden hacer pruebas de las fallas que pudieran ocurrir, pudiendo adelantarse en ocasiones a posibles errores.

**Refactorización del código:** Se basa en la reutilización del código, para lo cual se crean modelos estándares, siendo más flexible el cambio.

**Programación en parejas:** Se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera, es revisada y discutida mientras se escribe, siendo más importante que la posible pérdida de productividad inmediata.

La programación extrema (XP) se basa en varios valores fundamentales:

- ✓ La comunicación: Entre los usuarios y los desarrolladores. XP hace casi imposible la falta de comunicación.
- ✓ La simplicidad: Al desarrollar y codificar los módulos del sistema. Es mejor hacer hoy algo simple, que hacerlo complicado y probablemente no usarlo mañana.

La retroalimentación: Concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales. XP ofrece una mayor oportunidad de dirigir el esfuerzo eficientemente.(25)

Cuando usar XP:

- ✓ Cuando los clientes no tienen ideas claras de los requerimientos y los van cambiando.
- ✓ Para proyectos de riesgo: fecha fija de entrega, algo nunca hecho por el grupo, algo nunca hecho por la comunidad de desarrolladores.
- ✓ No es apto para proyectos con mucho personal (entre 2 y 10 programadores).
- ✓ Integra gerentes y clientes a la formulación de preguntas, la negociación de cronogramas y alcance, la creación de las pruebas.
- ✓ Automatiza las pruebas; es posible en casi todos los dominios. Es lícito repensar el diseño para facilitar el ensayo.

- ✓ El objetivo es entregar el software tal cual se necesita y en el momento en que se necesita. Incidentalmente, los proyectos XP muestran mayor productividad.

Entre las principales características que distinguen la metodología XP de otras se encuentran:

Es **Iterativa**: El proyecto se divide en versiones y estas a su vez en ciclos. En cada ciclo se desarrollan historias de usuarios que añaden incrementalmente funcionalidad al sistema.

Es **Compleja**: XP propone un conjunto de técnicas (denominadas prácticas) para ser utilizadas durante el desarrollo. Cada técnica por separado aporta escaso valor al proyecto, algunas son complejas y de difícil implantación.

Es **informal**: XP concede un elevado grado de libertad en la ejecución de los procesos. En lugar de establecer procesos formales de trabajo, se limita a señalar los principios y valores que deben guiar la actuación del equipo. Es característica también la ausencia de documentación.

Esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. Por tanto, se debe responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final del ciclo de la programación.(26)

### 1.5.3. SCRUM

El Scrum es un modelo de referencia, que define un conjunto de prácticas y roles, que pueden tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. El resultado final en esta metodología se consigue de forma iterativa e incremental. Al comienzo de cada iteración (sprint) se determina que partes se van a construir, tomando como criterios la prioridad para el negocio, y la cantidad de trabajo que se podrá abordar durante la iteración.

La metodología ágil SCRUM inicialmente documentada por dos japoneses (Takeuchi y Nonaka) en 1986 y con aportes de especialistas (Sutherland y Schwaber) a lo largo de la última década, tiene como objetivo hacer que los procesos que influyen en el desarrollo de tecnologías se incrementen con rapidez y flexibilidad, siempre tomando en cuenta los tiempos y reglas de negocio que dan motivo al desarrollo del proyecto.

Está especialmente indicada para proyectos con un rápido cambio de requisitos. SCRUM entiende el desarrollo de software como algo muy complejo, sujeto a multitud de variaciones e incertidumbres, sobre todo al comienzo. También trata de que todos los involucrados en el proyecto conozcan perfectamente en

qué punto del desarrollo se encuentra el proyecto y qué falta por hacer. Es ideal para proyectos con un constante cambio de requerimientos.

*Los actores de SCRUM son:*

- ✓ ProductOwner.
- ✓ Scrum Master.
- ✓ ScrumTeam.
- ✓ Usuario o Clientes.

*Las acciones fundamentales de SCRUM son:*

- ✓ ProductBacklog.
- ✓ Sprint Backlog.
- ✓ DailyScrum Meeting.

**Trabajar bajo la metodología SCRUM ofrece las siguientes:**

*Ventajas:*

- ✓ Ofrece la posibilidad de ajustar la funcionalidad en base a la necesidad de negocio del cliente.
- ✓ Brinda la visualización del proyecto día a día.

*Desventajas:*

- ✓ No genera toda la evidencia o documentación de otras metodologías.
- ✓ No es apto para todos los proyectos.
- ✓ Es necesario complementarlo con otros procesos (XP).(27)

#### *1.5.4. Selección de la metodología a utilizar*

Se realizó un estudio de las metodologías más usadas para el desarrollo del software y de acuerdo con las características y complejidad de este sistema, se decide trabajar con RUP. Esto implica que se necesita una gran cantidad de documentación, que permita a los desarrolladores entender con profundidad las



funcionalidades que deben implementar. RUP genera una documentación detallada de todo el proceso de software desde sus inicios, a diferencia de metodologías ágiles como XP y SCRUM.

RUP resulta más adaptable para los proyectos a largo plazo que necesiten un desarrollo iterativo capaz de mantener un buen control sobre los cambios, además detecta la mayor cantidad de riesgos en las primeras fases y se basa en las mejores prácticas de la Ingeniería de Software.

### 1.6. Tecnologías y herramientas a utilizar

La programación orientada a objetos ha ido evolucionando y cada vez son más las posibilidades que esta variante brinda en el desarrollo de productos informáticos. El surgimiento del lenguaje Javascript y el IDE Komodo, minimizan grandemente el trabajo a la hora de crear extensiones para Mozilla, así como la interpretación de lenguajes como XML y XUL tan necesarios para crear la interfaz de la extensión que se desea implementar, así como darle funcionalidad a la misma.

Al programar en JavaScript usando Komodo IDE, se contrastan estos grandes avances que finalmente repercuten en la optimización de tiempo, costo y esfuerzo así como en el incremento de la calidad en el desarrollo de software. La reutilización de componentes previamente programados, es una de las principales tendencias en los procesos ingenieriles para fines informáticos, hoy. Esta permite ahorrar tiempo, esfuerzo y costos, así como minimizar errores en la programación de elementos que ya existen y están probados. De ahí la importancia de saber seleccionar las herramientas claves utilizadas en este trabajo para la creación de una extensión para Mozilla.

En este capítulo, luego del estudio realizado se arriba a la conclusión de que el sistema estará guiado por la metodología de desarrollo RUP. Por las potencialidades ofrecidas se escoge como herramienta de modelado Visual Paradigm 6.4. En la implementación del sistema, se utilizará el Komodo IDE 4.2.0, para la programación en lenguajes como XUL y XML asociados al lenguaje Javascript, pues se trata de una aplicación de escritorio, exactamente una extensión integrable con el Mozilla Firefox.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

En este capítulo se realiza la descripción del sistema a desarrollar, para lograr una mayor claridad y comprensión por parte del equipo de desarrollo. Los puntos que se abordarán son los siguientes: descripción del modelo de dominio, especificación de los requisitos funcionales y no funcionales, representación de los actores y casos de uso a través de un diagrama de casos de uso del sistema, así como la descripción textual de ellos.

### 2.1. Objeto de estudio

#### 2.1.1. Situación Problemática

En el Grupo de Calidad del Centro de Informática Médica (CESIM) y en la Universidad de las Ciencias Informáticas se utiliza la herramienta REDMINE para la Gestión de proyectos y del proceso de pruebas, llevando el control y registro de las No Conformidades encontradas y almacenadas por los probadores en dicha aplicación web. La cual desempeña diversas tareas relacionadas con la organización de la documentación que existe en los proyectos de la universidad.

Actualmente durante el proceso de registro de una no conformidad, la ubicación de la misma se registra mediante un texto plano de forma libre creando diferentes dificultades en el momento de acceder a dicha ubicación de la no conformidad encontrada.

Con el uso de la herramienta que se propone en este trabajo se garantiza la automatización de los procesos y la solución real a los problemas que existen.

### 2.2. Objeto de Automatización

Resulta de vital importancia que REDMINE garantice, que las no conformidades encontradas y registradas se pueda acceder a su ubicación sin pérdida ni confusiones, para que el proceso de pruebas no se vea afectado, por lo que se desea obtener el siguiente proceso: estandarizar el proceso de registro de la ubicación, de cada no conformidad encontrada.

### 2.3. Propuesta del Sistema

Se propone la implementación de una extensión de Mozilla Firefox que genere un archivo XML, con la información referente al registro de la ubicación de cada NC. La extensión deberá ser capaz de estandarizar

la ubicación de una No Conformidad encontrada, al capturar cada clic dado por el usuario en un escenario determinado cada vez que encuentre una No Conformidad, registrando de esta forma la traza de la ubicación de la misma. El sistema deberá proporcionar toda una serie de funcionalidades asociadas al proceso de grabación, ya sea editarla, pausarla, etc.

### 2.4. Descripción del Modelo de Dominio

El modelo de dominio es una representación visual estática del entorno real objeto del proyecto. Es un diagrama con los objetos que existen (reales) relacionados con el proyecto que se va a desarrollar y las relaciones que hay entre ellos. Este modelo ayuda a comprender los conceptos que utilizan los usuarios, es decir, los conceptos con los que trabajan y con los que deberá trabajar la aplicación.

El modelo de dominio se realiza si no se logran determinar los procesos de negocio con fronteras bien establecidas donde se logren ver claramente, quienes son las personas que realizan cada proceso de negocio, quienes son los beneficiados con cada uno de estos procesos, pero además quienes son las personas que desarrollan las actividades en cada uno de estos procesos. (28)

Para un mejor entendimiento del Diagrama de Modelo de Dominio conformado, en este punto se proporciona un marco conceptual con las definiciones identificadas, las cuales son:

**Probador:** Persona encargada de detectar las NC, grabar y reproducir su ubicación.

**Redmine:** Gestor y planificador de proyectos con interfaz web, orientado a la coordinación de tareas, comunicación de participantes, y que puede especializarse en proyectos de desarrollo gracias a herramientas como la integración en un repositorio de código.

**No conformidades:** Se refiere al “No cumplimiento de un requisito determinado”, a su vez es el error, bug o mal funcionamiento que pueda existir en un software.

**Anexo:** Imagen o dato adicional del área en que se produce el error, que ayuda a identificar la NC registrada.

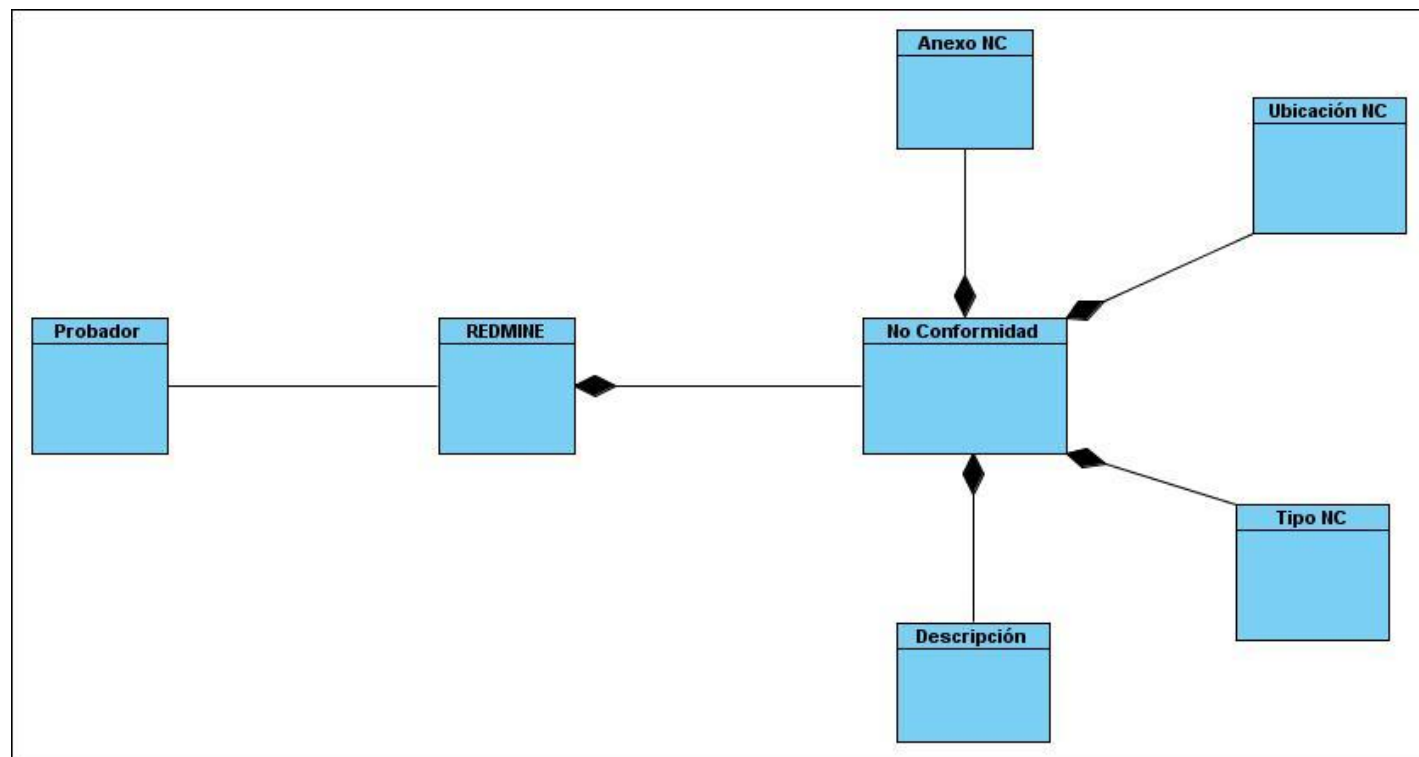
**Tipo:** Se refiere a la clasificación o tipo de NC encontrada.

**Ubicación:** Puede ser un link, URL o descripción manual que posibilite determinar donde se encuentra una NC.

**Descripción:** Descripción textual y detallada del tipo de NC encontrada.

**Título:** Título que se le asigna a la NC al ser registrada.

El modelo de dominio se describe mediante diagramas de UML, especialmente mediante diagramas de clases. En la figura (2) se representa el modelo de dominio realizado.



**Ilustración 2: Diagrama del modelo del Dominio.**

### 2.5. Requerimientos Funcionales del Sistema

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

La solución informática cuenta con las siguientes funcionalidades:

*Nota: Se usa el prefijo **RF** en su nomenclatura.*

**RF1** Iniciar grabación

**RF2** Detener grabación

*RF3 Pausar grabación*

*RF4 Reproducir ubicación*

*RF5 Realizar ciclos de reproducción*

*RF6 Editar grabación*

*RF7 Guardar archivo*

*RF8 Abrir archivo*

*RF9 Configurar opciones de grabación*

*RF10 Limpiar archivo de grabación*

## 2.6. Requerimientos No Funcionales del Sistema

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener, forman una parte significativa de la especificación. Tienen la característica de marcar la diferencia, pues una vez comprobado que el producto cumple con lo requerido por el cliente, las propiedades no funcionales; como cuán usable, seguro, conveniente y agradable, distinguen a un producto bien aceptado de uno con poca aceptación.

Para un correcto funcionamiento el sistema debe de tener como características principales para su funcionamiento los siguientes requisitos no funcionales:

**Usabilidad:** El sistema de ser diseñado de forma tal, que los usuarios que harán uso del mismo obtengan los conocimientos necesarios en el menor tiempo posible para la explotación de sus funcionalidades.

**RNF1**---La interfaz de la aplicación es de un flujo sencillo, típico de herramientas de escritorio.

**RNF2**---Será útil para reproducir las NC sin pérdida alguna por parte del usuario.

**Soporte:** Debe presentar un diseño que sea capaz de la realización de pruebas que permitan la explotación de la aplicación de forma eficiente. Se incluye la posibilidad de brindar asistencia técnica que permita la solución de problemas en tiempo real de ejecución que garantice la solución de fallas que pueda presentar. Debe ser garantizada su adaptabilidad y compatibilidad con los distintos Sistemas Operativos.

**RNF3**---Contará con una interfaz de fácil exploración, y de sencillos controles para su funcionamiento.

**Portabilidad:** Estos requerimientos especifican los atributos que debe presentar el software para facilitar su traslado a otras plataformas u entornos.

**RNF4**---La aplicación podrá ser usada en sistemas operativos Windows o Linux.

**Seguridad:** Es el requerimiento más complejo y difícil de garantizar, a su vez permitirá garantizar que la aplicación será utilizada correctamente por cada usuario según sus niveles permitidos.

- ✓ **Confidencialidad:** Se debe tratar el manejo de permisos de forma que solamente se acceda a la información autorizada de acuerdo a los niveles de permisos que debe tener cada usuario del sistema. De esta forma se garantiza que la información no sea expuesta a personal indebido.
- ✓ **Integridad:** Se debe de tratar el manejo de la información de forma tal, que la información no sea modificada por personal ajeno, evitando de esta forma alteraciones en los resultados planteados en la documentación.
- ✓ **Disponibilidad:** Se deberá garantizar el acceso pleno de cada usuario con facultades para el uso de la aplicación las 24 horas del día.

### **Apariencia o Interfaz Externa**

Requerimiento que describe la apariencia del producto. Es válido destacar que no se trata del diseño de la interfaz en detalle, sino que especifican cómo se pretende que sea la interfaz externa del producto.

**RNF5**---Una herramienta sencilla, fácil de usar y presenta las funcionalidades bastante explícitas.

**RNF6**---Tiene colores relajantes y no presenta sobrecarga de imágenes.

**Software:** Para el uso de la aplicación se utilizará como sistema operativo Windows XP y como plataforma de instalación Mozilla Firefox en sus versiones desde la 3.5 hasta la 3.6.\*

**Hardware:** Estos requerimientos especifican las características lógicas para cada interfaz entre el producto y los componentes de hardware del sistema.

**RNF7**---Rendimiento mínimo del hardware.

- ✓ Procesador 2.8 GHz.
- ✓ 256 MB de memoria RAM.

## 2.7. Definición de los Casos de Uso

Los casos de uso son el componente clave del modelado. Su propósito es ilustrar como un sistema permite a un actor cumplir una meta, ilustrando todos los posibles caminos apropiados que ellos pueden tomar para cumplirla, así como las situaciones que podrían hacerlo fallar.

**CUS 1** *Realizar grabación.*

**CUS 2** *Reproducir grabación*

**CUS 3** *Realizar ciclos de reproducción*

**CUS 4** *Editar grabación*

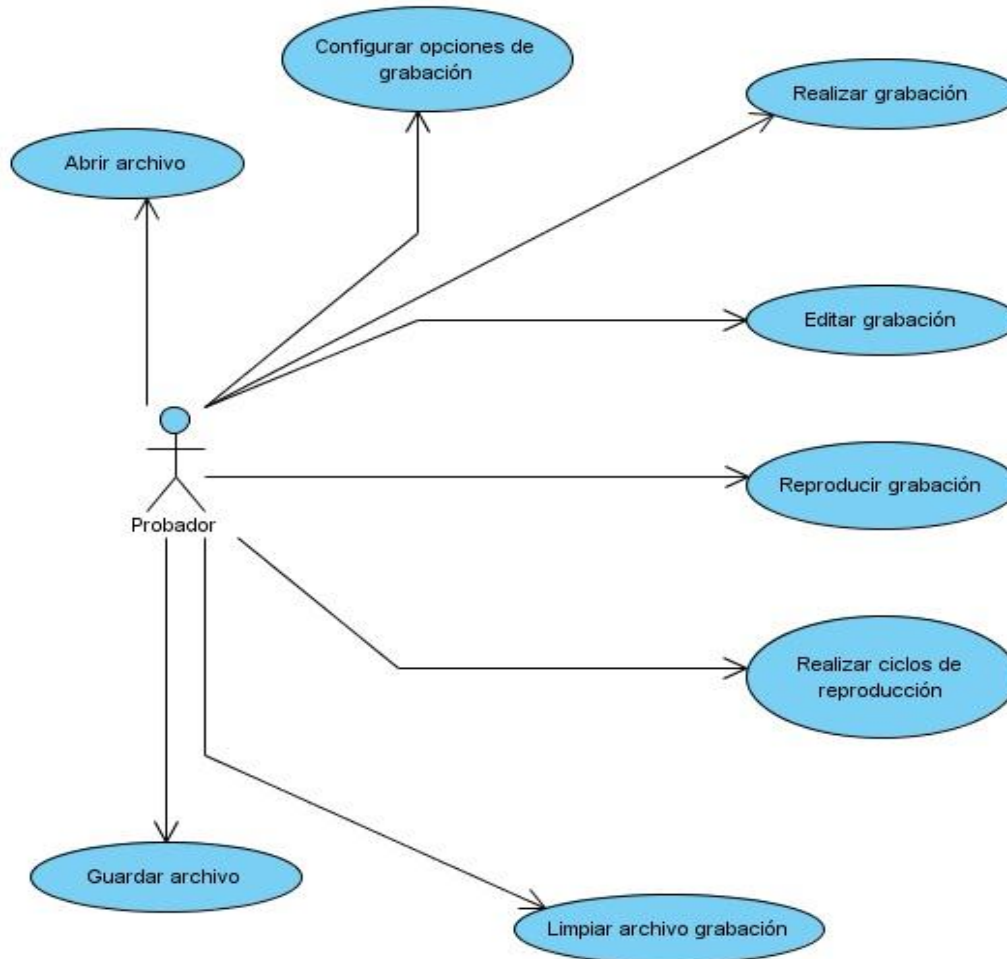
**CUS 5** *Guardar archivo*

**CUS 6** *Abrir archivo*

**CUS 7** *Configurar opciones de grabación*

**CUS 8** *Limpiar archivo de grabación*

2.8. Diagrama de Casos de Uso del Sistema



**Ilustración 3: Diagrama de Casos de Uso del Sistema.**

2.8.1. Definición de los actores

Actores	Justificación
Probador	Es el encargado de detectar las NC, grabar y reproducir su ubicación.

**Tabla 1: Descripción del Actor del Sistema**



### 2.8.2. Listado de los casos de uso

De acuerdo a las funcionalidades que se le agregarán a la solución informática a desarrollar quedan conformados los siguientes Casos de Uso: Realizar grabación, Reproducir grabación y Realizar ciclos de reproducción, los demás se encuentran especificados en el [Anexo 1](#)

#### Caso de Uso: Realizar grabación

Caso de uso	
<b>CUS 1</b>	Realizar grabación
<b>Propósito</b>	Grabar la ubicación de una NC en las pruebas realizadas a sistemas web.
<b>Actores:</b> Probador	
<b>Resumen:</b> El caso de uso inicia cuando el probador accede a la opción Grabar ubicación, el sistema muestra las opciones correspondientes para que el probador grabe la secuencia de pasos que componen la ubicación de una NC. El caso de uso culmina cuando el probador logra grabar la ubicación de la NC satisfactoriamente.	
<b>Referencias</b>	RF1,RF2,RF3
FlujoNormal de Eventos	
Acción del actor	Respuesta del sistema
1. El probador accede al panel inferior del navegador Mozilla y selecciona la opción "Grabar" (fácilmente reconocible por un botón rojo redondo). Luego comienza a navegar por la aplicación web que se esté probando dando click en cada vista hasta conformar el camino. La grabación puede ser pausada en cualquier instante del proceso (Ver flujo alterno "Pausar").	2. El sistema va registrando cada click dado o dato llenado durante todo el proceso, conformando así una traza de la navegación por la aplicación web.

<a href="#">Ver anexo 1</a>	
3. El probador accede a la opción “Detener grabación” (fácilmente reconocible por un botón negro cuadrado) una vez concluido el proceso o detectada la NC o error.	4. El sistema detiene la grabación y la registra internamente para que posteriormente pueda ser editada, exportada o reproducida.
<b>El Caso de Uso termina</b>	

**Tabla 2: Descripción del CU: Realizar grabación**

**Caso de Uso:** Reproducir grabación

Caso de uso	
<b>CUS 2</b>	Reproducir grabación
<b>Propósito</b>	Reproducir satisfactoriamente una ubicación previamente grabada.
<b>Actores:</b> Probador	
<b>Resumen:</b> El caso de uso inicia cuando el probador accede a la opción Reproducir ubicación, el sistema realiza las funciones correspondientes para reproducir la grabación. El caso de uso culmina cuando el probador logra reproducir una grabación previamente echa satisfactoriamente.	
<b>Referencias</b>	RF2, RF3, RF4
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El probador accede al panel inferior del navegador Mozilla y selecciona la opción “Reproducir” fácilmente reconocible por un botón verde en forma triangular. La reproducción puede ser pausada en cualquier instante del proceso (Ver flujo alterno “Pausar”). <a href="#">Ver Anexo 1</a>	2. El sistema va reproduciendo cada click dado o dato llenado durante todo el proceso, conformando así una traza de la navegación por la aplicación web en forma de video o macro de repetición.

El Caso de Uso termina	
------------------------	--

**Tabla 3: Descripción del CU: Reproducir grabación.**

**Caso de Uso:** Realizar ciclos de reproducción

Caso de uso	
<b>CUS 3</b>	Realizar ciclos de reproducción
<b>Propósito</b>	Reproducir varias veces continuas de forma automática una reproducción.
<b>Actores:</b> Probador	
<b>Resumen:</b> El caso de uso se inicializa cuando el probador decide reproducir una grabación previamente echa más de 1 vez de forma automática. El caso de uso culmina cuando la cantidad de ciclos especificados se reproducen satisfactoriamente.	
<b>Referencias</b>	RF1, RF2, RF3, RF5
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El probador accede al panel inferior del navegador Mozilla y selecciona la opción "Realizar ciclos de reproducción" (fácilmente reconocible por una flecha verde curva).	2. El sistema muestra una ventana para especificar la cantidad de ciclos que se desea reproducir.
3. El probador especifica la cantidad de ciclos a reproducir y selecciona la opción Aceptar. Luego selecciona la opción "Reproducir" (Ver descripción del caso de uso Reproducir grabación).	

---

El Caso de Uso termina	
------------------------	--

**Tabla 4: Descripción del CU: Realizar ciclos de reproducción**

En este capítulo se definieron las características de la aplicación a construir, las cuales se especifican en términos de requisitos funcionales y no funcionales, que fueron modelados gráficamente mediante un Diagrama de Casos de Uso del Sistema. Se identificaron y describieron los actores que interactuarán con las funcionalidades previstas para cada uno de estos casos de usos. Finalmente, se concluye que, con el desarrollo de este capítulo quedaron sentadas las bases que darán paso al diseño y construcción de la herramienta para la reproducción controlada de NC.

## CAPÍTULO 3. DISEÑO DEL SISTEMA

En este capítulo se realiza la descripción del diseño de la aplicación haciendo uso de los distintos diagramas afines para dicha actividad. Se realiza el análisis de cada uno de los casos de usos y se describe mediante el uso de estereotipos web la relación entre las distintas páginas que conforman la aplicación que se pretende desarrollar. Se confecciona además el diagrama de clases persistentes para la generación de la base de datos, se definen los principios de diseño y los patrones y estándares a utilizar.

### 3.1. Arquitectura del sistema

La arquitectura de software se centra tanto en los elementos estructurales significativos del sistema, como subsistemas, clases, componentes, nodos y las colaboraciones que tienen lugar entre estos elementos a través de las interfaces. Los casos de uso dirigen la arquitectura para hacer que el sistema proporcione la funcionalidad y uso deseado, alcanzando a la vez objetivos de rendimientos razonables. Una arquitectura debe ser completa, pero también debe ser suficientemente flexible como para incorporar nuevas funciones, y debe soportar la reutilización del software existente.

La arquitectura abarca decisiones importantes sobre:

- ✓ La organización del sistema.
- ✓ Los elementos que componen el sistema y sus interfaces, unido a su comportamiento.
- ✓ La composición de los elementos estructurales y del comportamiento en subsistemas progresivamente más grandes.
- ✓ Estilos de arquitectura que guían esta organización: elementos y sus interfaces, sus colaboraciones y su composición.(29)

#### 3.1.1. *Arquitectura de software basada en componentes.*

Una arquitectura basada en componentes describe una aproximación de ingeniería de software al diseño y desarrollo de un sistema. Esta arquitectura se enfoca en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas. Esto provee un nivel de abstracción mayor que los principios de orientación por objetos y no se enfoca en asuntos específicos de los

objetos como los protocolos de comunicación y la forma como se comparte el estado. El estilo de arquitectura basado en componentes tiene las siguientes características:

- ✓ Es un estilo de diseño para aplicaciones compuestas de componentes individuales.
- ✓ Pone énfasis en la descomposición del sistema en componentes lógicos o funcionales que tienen interfaces bien definidas.
- ✓ Define una aproximación de diseño que usa componentes discretos, los que se comunican a través de interfaces que contienen métodos, eventos y propiedades.(30)

### 3.1.2. Desarrollo de software basado en componentes.

El desarrollo de software basado en componentes se ha convertido actualmente en uno de los mecanismos más efectivos para la construcción de grandes sistemas y aplicaciones de software, trata de fundamentar las bases para el diseño y desarrollo de aplicaciones distribuidas basadas en componentes de software reutilizable.

El desarrollo de software basado en componentes se basa en la reutilización de los componentes, estos se diseñan y desarrollan con el objetivo de utilizarse en otras aplicaciones, al utilizar componentes ya probados previamente reduce el tiempo de desarrollo, se mejora la fiabilidad del producto y aumenta la competitividad en coste. Puede verse como una extensión natural de la programación orientada a objetos dentro del ámbito de los sistemas abiertos y distribuidos. Entre sus ventajas se tienen:

Ventajas:

- ✓ Reutilización del software: Permite alcanzar un mayor nivel de reutilización de software.
- ✓ Simplifica las pruebas: Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados.
- ✓ Simplifica el mantenimiento del sistema: Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.
- ✓ Mayor calidad: Dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo. (31)

## 3.2. Modelo de diseño

Modelo de diseño Un Modelo de 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. Este artefacto constituye la entrada fundamental utilizada para el correcto desarrollo de la implementación.(32)

### 3.2.1. Patrones de diseño

Para llevar a cabo un buen diseño se han definido una serie de patrones. Un patrón de diseño de software es una solución reusable y comprobada a un problema común en el desarrollo de software. **Los patrones GRASP**<sup>30</sup> describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Dentro de este grupo se identifican 5 patrones fundamentales experto, creador, alta cohesión, bajo acoplamiento y el controlador.

### 3.2.2. Bajo Acoplamiento

El acoplamiento es la medida de la fuerza con que una clase está conectada a otras clases, a las que conoce y recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras. Una clase con alto (o fuerte) acoplamiento recurre a muchas otras, lo que trae consigo que los cambios una clases ocasionen cambios en otras, sean más difíciles de entender y de reutilizar porque se requiere la presencia de otras clases de las que dependen.

---

<sup>30</sup>Patrones Generales de Software para Asignación de Responsabilidades (General Responsibility Assignment Software Patterns)

### 3.2.3. Alta Cohesión

La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Una baja cohesión hace muchas cosas no afines o realiza trabajo excesivo.(33)

### 3.2.4. Diagrama de clases del diseño.

En el diagrama de clases de diseño se muestran los atributos y métodos de cada clase y se representa de una forma sencilla la colaboración y las responsabilidades de las distintas clases que forman el sistema. El diagrama de clases de diseño representa la parte estática del sistema. Contiene las clases del diseño y sus relaciones.

Se presentará el diagrama de clases del diseño integrador de los casos de uso: Realizar grabación, Reproducir grabación, Realizar ciclos de reproducción, Editar grabación, Guardar, Abrir archivo, Configurar opciones de grabación y Limpiar archivo de grabaciones.



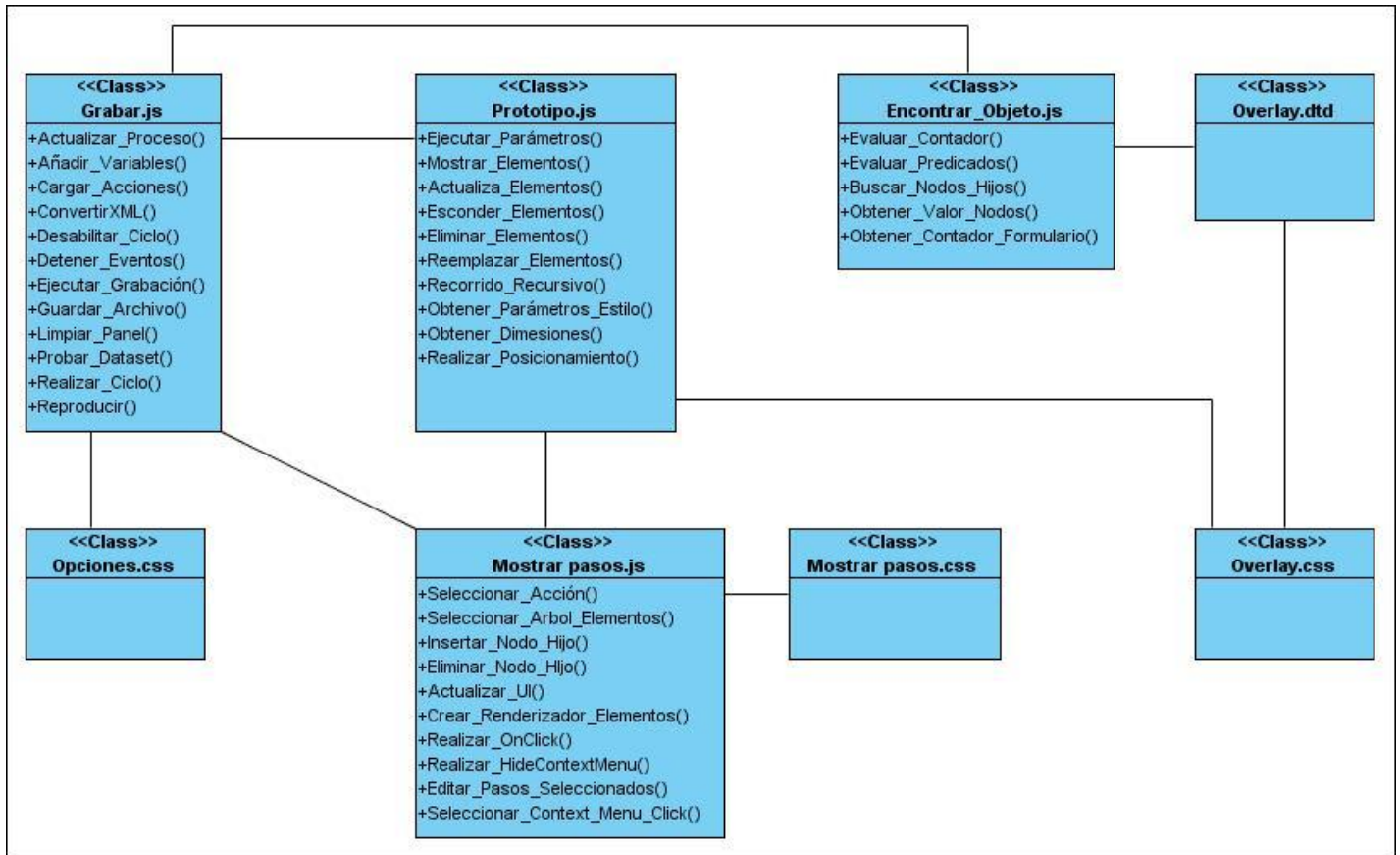
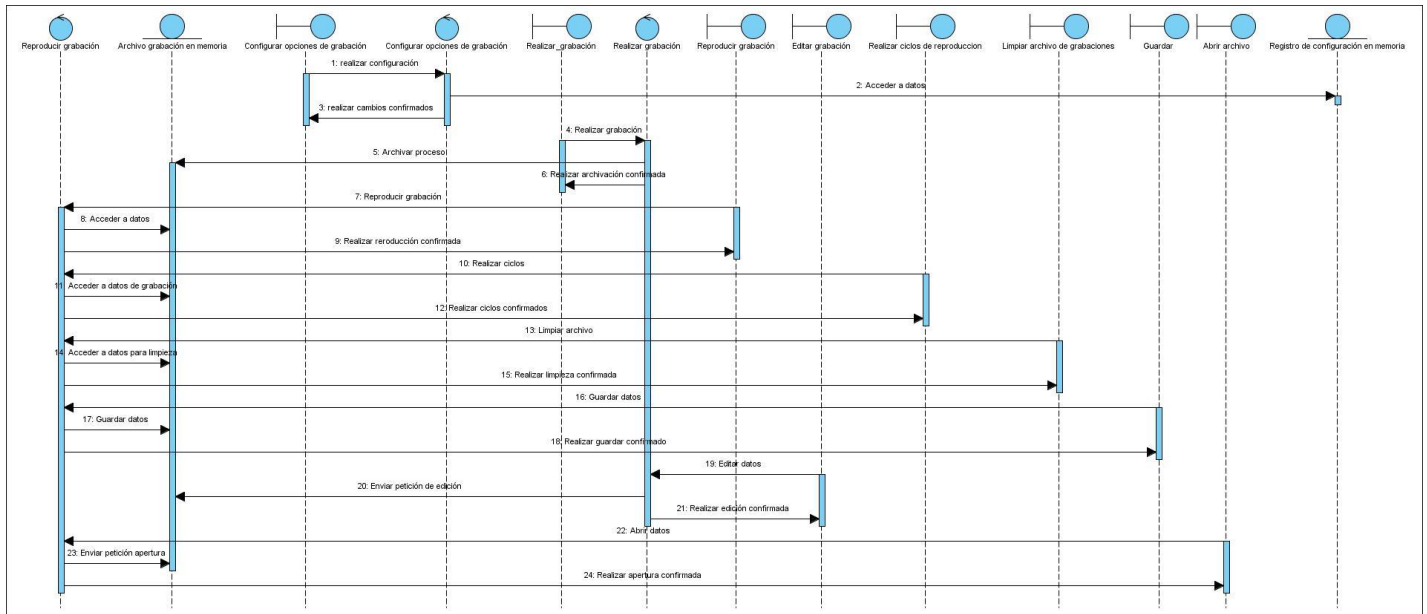


Ilustración. 4: Diagrama de clases del diseño

3.2.5. Diagrama de Interacción (Secuencia)

Se presentará el diagrama de interacción integrador de los Caso de Uso Realizar grabación, Reproducir grabación, Realizar ciclos de reproducción, Editar grabación, Guardar, Abrir archivo, Configurar opciones de grabación y Limpiar archivo de grabaciones.



**Ilustración. 5: Diagrama de Interacción (Secuencia)**

En el capítulo se detalló la fase del diseño de la aplicación, donde se definen las clases y modelos que la conforman. Se realizó un estudio de las diferentes arquitecturas y se definió el uso de la arquitectura basada en componentes y el uso del patrón GRASP (bajo acoplamiento y alta cohesión) para el diseño del sistema y se realizó una descripción de las clases del diseño de la aplicación. Además se detallan los tipos de datos o componentes que las conforman.

---

## **CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBAS**

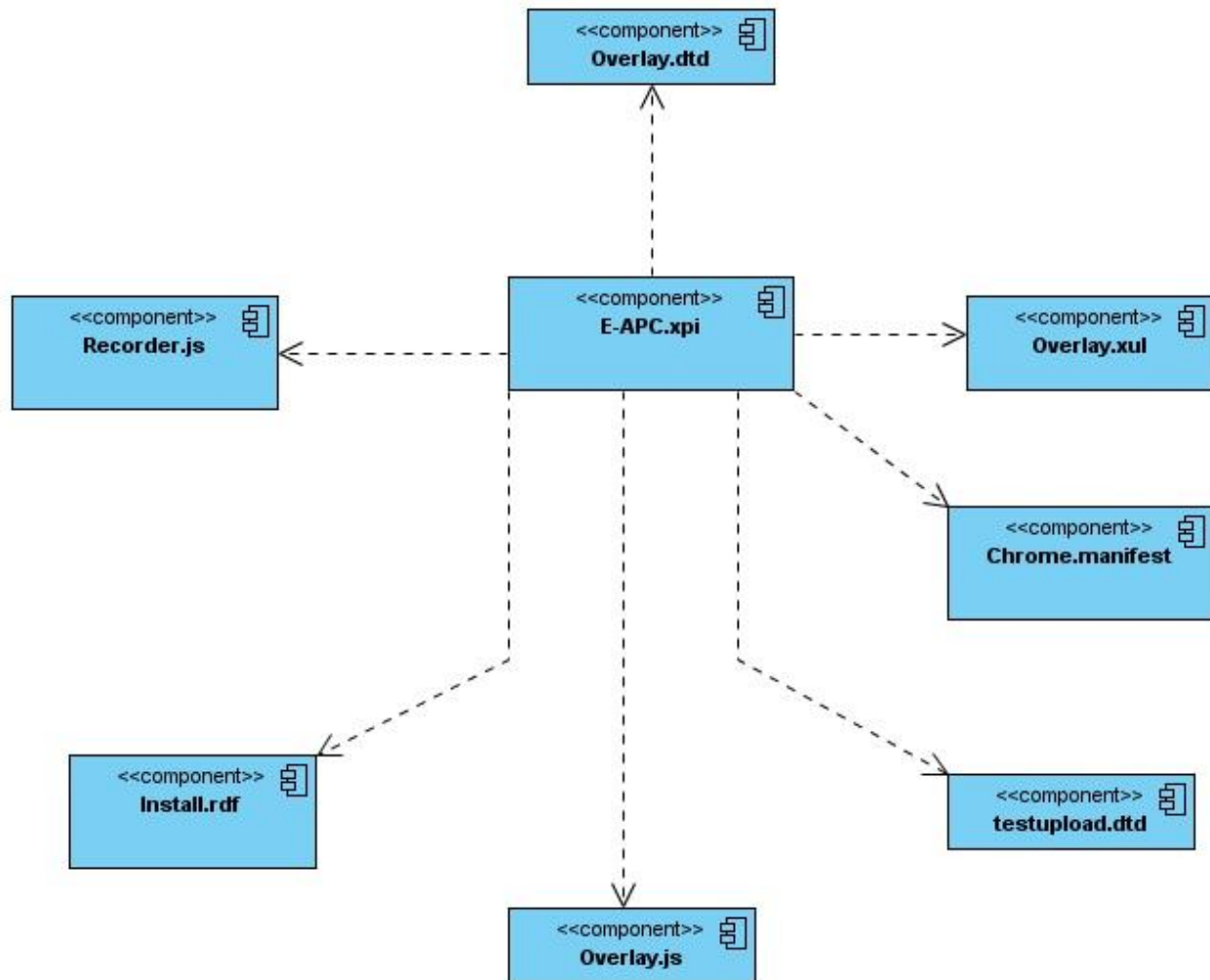
El propósito fundamental de este capítulo es definir cómo desarrollar la arquitectura definida durante el diseño y el sistema como un todo. Se implementan las clases y subsistemas definidos en el capítulo anterior en términos de componentes. Se realizarán los diagramas de componentes y de despliegue, y finalmente desarrollar pruebas de caja negra a la herramienta para garantizar un buen funcionamiento de la misma.

### **4.1. Despliegue**

El diagrama de despliegue define la arquitectura física del sistema por medio de nodos interconectados. Se modela la topología del hardware sobre el que se ejecuta el sistema. Para la puesta en práctica de la herramienta, solo se necesitan computadoras clientes donde correrá la aplicación en cada una.

### **4.2. Diagrama de componentes**

Un componente representa una parte física del sistema, los mismos pueden ser una librería, un ejecutable, una tabla etc. El diagrama de componentes permite conocer a los desarrolladores la estructura física que tiene el sistema y como se relacionan sus partes. A continuación se muestra cómo quedará la estructura de la implementación de la herramienta para la reproducción controlada de NC.



**Ilustración 6: Diagrama de componentes**

### 4.3. Pruebas

La prueba es el proceso de ejecución de un programa con la intención de descubrir un error. Se puede ir realizando pruebas desde la fase de inicio del software hasta la fase de construcción, siendo esta última fase donde tiene mayor volumen el flujo de trabajo de prueba.

El proceso de realización de pruebas a un software, está precedido por el diseño de los casos de prueba, que se definen según las funcionalidades descritas en los casos de uso. Existen casos de pruebas para los diferentes métodos: Caja Negra y Caja Blanca. En el proceso de pruebas en cuestión, se hace uso de los casos de prueba de Caja Negra, de ahí el motivo por el que se diseña un caso de prueba por caso de uso del sistema.

El siguiente es un ejemplo donde se detalla el caso de uso Realizar grabación.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Iniciar grabación	EC1.1: Iniciar grabación	En este escenario el usuario selecciona la opción de Grabar.
	EC1.2: Detener grabación	En este escenario el usuario tiene la posibilidad de Detener una grabación en curso.
	EC1.3: Pausar grabación	En este escenario el usuario tiene la posibilidad de Pausar una grabación en curso.

Este es un ejemplo de la matriz de datos del caso de uso Realizar grabación, de la SC1 Iniciar grabación.

ID del escenario	Escenario	Respuesta del sistema	Resultado de la prueba	Flujo central
EC:1.1	Iniciar grabación	El sistema va registrando cada clic dado o dato llenado durante todo el proceso, conformando así una traza de la navegación por la aplicación web.	Satisfactorio	<ol style="list-style-type: none"> <li>1. El actor selecciona la opción Grabar.</li> <li>2. El sistema muestra registra todas las acciones del actor en las vistas del</li> </ol>

				sistema web.
EC1.2	Detener grabación	El sistema detiene la grabación y la registra internamente para que posteriormente pueda ser editada, exportada o reproducida.	Satisfactorio	2.1 El actor selecciona la opción Detener. 2.2 El sistema detiene la grabación y la registra internamente.
EC1.3	Pausar grabación	El sistema pone en pausa la acción que se está realizando en ese momento.	Satisfactorio	1.1 El actor selecciona la opción Pausar. 1.2 El sistema pone en pausa la grabación en curso.

El siguiente es un ejemplo donde se detalla el caso de uso Editar grabación.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC2: Editar grabación	EC2.1: Editar grabación	En este escenario el usuario selecciona la opción de Editar grabación.

Este es un ejemplo de la matriz de datos del caso de uso Editar grabación, de la SC2Editar grabación.

ID del escenario	Escenario	Respuesta del sistema	Resultado de la prueba	Flujo central
EC:2.1	Editar grabación	El sistema muestra un panel con las opciones de edición: subir, bajar, cortar o pegar cada línea de registro que compone las acciones realizadas en cada grabación.	Satisfactorio	<ol style="list-style-type: none"> <li>1. El actor selecciona la opción Editar.</li> <li>2. El sistema muestra un panel con las opciones de edición</li> </ol>

El siguiente es un ejemplo donde se detalla el caso de uso Guardar.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC3: Guardar	EC3.1: Guardar	En este escenario el usuario selecciona la opción de Guardar grabación.

Este es un ejemplo de la matriz de datos del caso de uso Guardar, de la SC3Editar grabación.

ID del escenario	Escenario	Respuesta del sistema	Resultado de la prueba	Flujo central
EC:3.1	Guardar	El sistema muestra un panel con las opciones de guardado, permitiéndole al probador seleccionar dónde desea guardar el archivo, en la PC o en otro medio externo de almacenamiento.	Satisfactorio	<ol style="list-style-type: none"> <li>1. El actor selecciona la opción Guardar.</li> <li>2. El sistema muestra un panel con las opciones de guardado.</li> </ol>

En este capítulo, se obtuvo el diagrama de componentes y quedó conformado el modelo de implementación del sistema. El sistema no requiere de un diagrama de despliegue, porque para la puesta en marcha de la herramienta, solo se necesitan computadoras clientes donde correrá la aplicación. Además se desarrollaron los diseños de casos de pruebas, así como las pruebas funcionales al sistema.



---

## **CONCLUSIONES**

Una vez concluida la investigación, se le ha dado cumplimiento a los objetivos planteados, obteniéndose los resultados que a continuación se mencionan:

Las soluciones informáticas para la gestión de errores y registro de NC, no responden a las necesidades existentes de reproducir de forma controlada una No Conformidad.

Se seleccionaron las tecnologías, metodologías y herramientas más adecuadas para la implementación de la herramienta. Se definió una arquitectura de diseño e implementación para la elaboración de la aplicación.

Se obtuvo como resultado final una herramienta capaz de reproducir de forma controlada y automatizada las NC en pruebas que se realizan a los sistemas web.

---

## **RECOMENDACIONES**

Para futuras investigaciones, se recomienda:

- ✓ Someter el sistema a una fase de pruebas de caja blanca, usando las herramientas destinadas para este proceso.
- ✓ Implementar una versión para otros navegadores como Netscape, Opera o Google Chrome, partiendo de la base de la investigación y la implementación realizadas aquí expuestas.
- ✓ Implementar una funcionalidad de búsqueda y detección de las grabaciones que presenten la misma ubicación es decir con la misma URL interna registrada al concluir el proceso.
- ✓ Lograr compatibilizar la extensión con versiones del navegador Mozilla Firefox superiores a las 3.5.\*

## REFERENCIAS BIBLIOGRÁFICAS

1. *“Documentación de una no conformidad”*. 2005.
2. **Pressman, R.** *“Enfoque practico de la Ingeniería de Software”*. 1998.
3. Calisoft. *Calisoft*. [En línea] [http://calisoft.uci.cu/index.php?view=article&catid=10:art&id=17:proceso-de-pruebas&option=com\\_content&Itemid=24](http://calisoft.uci.cu/index.php?view=article&catid=10:art&id=17:proceso-de-pruebas&option=com_content&Itemid=24).
4. Prueba de software. *Rincón del vago*. [En línea] <http://rincondelvago.com/prueba-de-software.html>.
5. Bugzilla. *Bugzilla*. [En línea] 2005. <http://www.Bugzilla.org>.
6. **Moody, David**. The Bug Genie. [En línea] <http://www.thebuggenie.com>...
7. gnats [En línea] <http://alumnus.caltech.edu/~dank/gnats.html>.
8. kennwhite-sourceforg. [En línea] <http://kennwhite.sourceforge.net/codetrack>.
9. Mantis. [En línea] <http://www.mantisbt.org>.
10. bugaboo. [En línea] [www.bugaboo.com/learn/bugaboo](http://www.bugaboo.com/learn/bugaboo).
11. sourceforge. [En línea] <http://sourceforge.net/projects/zentrack/>.
12. Edgewall. [En línea] <http://trac.edgewall.org>.
13. **Rojas Morrell, Raúl**. 2007. Diseño de una aplicación para el Seguimiento de Errores de los productos software de la Facultad 7. Trabajo de diploma para optar por el título de ingeniero en ciencias informáticas.
14. **Molina, Yenisel**. Sistema para la Gestión de las No Conformidades durante las Pruebas de Calidad. Trabajo de diploma para optar por el título de ingeniero en ciencias informáticas.
15. CodeGear. [En línea] Borland Software Corporation. CodeGear from Bor <http://www.codegear.com/products/bds2006>.
16. **Mueller, John**. *ANAYA MULTIMEDIA-ANAYA INTERACTIVA*. 2002.
17. **Ayala Montero, Ramón - Dierking, Howard**. *XML Iniciación y MSDN Magazine*. 2001.
18. Xul [En línea] <http://www.xul.fr/xul-tools.html>.

19. Curso de javascript [En línea] [http://webdelprogramador.com/lenguajes\\_script/curso de javascript](http://webdelprogramador.com/lenguajes_script/curso_de_javascript).
20. MSDN Microsoft Developer Network. [En línea] 2008. Microsoft Corporation . MSDN Microsoft Developer Network . Visual Studio Developer C <http://msdn.microsoft.com/en-gb/vstudio/bb265237.aspx>.
21. sourceforgeide [En línea] [http://sourceforgeide.com/codigo web/komodo ide](http://sourceforgeide.com/codigo_web/komodo_ide).
22. freedownloadmanager.[En línea][http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_p ara\\_UML\\_\(M%C3%8D\)\\_14720](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720).
23. Enterprise-Architect. [En línea] [http://ingenierioscomputacionipla.wordpress.com/2008/09/12/enterprise- architect-71-modelado-uml/](http://ingenierioscomputacionipla.wordpress.com/2008/09/12/enterprise-architect-71-modelado-uml/).
24. vico. [En línea] <http://www.vico.org/FormMentorOutsourcingUML.pdf>.
25. kmels. [En línea] <http://kmels.net/files/2009/uvg/cc2003/Resources/Contenidos/XP/xp.pdf>..
26. Programacion Extrema [En línea] [http://migueljaque.com/index.php/metodologias/xp/29-xp/63- programacionextrema](http://migueljaque.com/index.php/metodologias/xp/29-xp/63-programacionextrema).
27. Arturoweb. [En línea] <http://arturoweb.wordpress.com/2008/02/14/scrum-metodologia-agil-de-desarrollo>..
28. Ejemplos\_clase. [En línea] [http://iie.fing.edu.uy/ense/assign/desasoft/practico/hoja8/ejemplos\\_clase2.pdf](http://iie.fing.edu.uy/ense/assign/desasoft/practico/hoja8/ejemplos_clase2.pdf)..
29. **Jacobson, Ivar**. 2000. *El Proceso Unificado*.
30. **Pelaez, Juan Carlos**. Blog de Juan Peláez en Geeks.ms. [En línea] <http://geeks.ms/blogs/jkpelaez/archive/2009/04/18/arquitectura-basada-en-componentes.aspx>.
31. **Fuentes, Lidia. Troya, José y Vallecillo, Antonio**. ETSI Informática. [En línea] <http://www.lcc.uma.es/~av/Docencia/Doctorado/tema1.pdf>.
32. UML y patrones. [En línea] <http://bibliodoc.uci.cu/pdf/reg00061.pdf>.
33. Patrones-de-Diseno-Recopilacion. <http://es.scribd.com/doc/17767790/Patrones-de-Diseno-Recopilacion>
34. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James**. *El Proceso Unificado de Desarrollo de Software*. La Habana : s.n., 2000.

## **BIBLIOGRAFÍA**

1. Arquitectura-Basada-en-Componentes.<http://es.scribd.com/doc/14704374/Arquitectura-Basada-en-Componentes>
2. Rodolfo Quispe-Otazu. [www.rodolfoquispe.org/.../que-es-la-calidad-de-software.php](http://www.rodolfoquispe.org/.../que-es-la-calidad-de-software.php)
3. Calidad de software.[www.uniboyaca.edu.co/.../index.php](http://www.uniboyaca.edu.co/.../index.php)
4. Calidad-software. [www.gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://www.gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF)
5. Extensiones-Mozilla. [http://www.mozilla-hispano.org/.../Instalar\\_extensiones](http://www.mozilla-hispano.org/.../Instalar_extensiones)
6. Mozilla. [www.eumed.net/grumetes/.../firefoxextensiones.htm](http://www.eumed.net/grumetes/.../firefoxextensiones.htm)
7. Documentación de una no conformidad. 2005.
8. **Pressman, R.** “*Enfoque practico de la Ingeniería de Software*”. 1998.
9. Calisoft. *Calisoft*. [En línea] [http://calisoft.uci.cu/index.php?view=article&catid=10:art&id=17:proceso-de-pruebas&option=com\\_content&Itemid=24](http://calisoft.uci.cu/index.php?view=article&catid=10:art&id=17:proceso-de-pruebas&option=com_content&Itemid=24).
10. Prueba de software. *Rincón del vago*. [En línea] <http://rincondelvago.com/prueba-de-software.html>.
11. Bugzilla. *Bugzilla*. [En línea] 2005. <http://www.Bugzilla.org>.
12. **Moody, David.** The Bug Genie. [En línea] <http://www.thebuggenie.com>...
13. gnats [En línea] <http://alumnus.caltech.edu/~dank/gnats.html>.
14. kennwhite-sourceforge. [En línea] <http://kennwhite.sourceforge.net/codetrack>.
15. Mantis. [En línea] <http://www.mantisbt.org>.
16. bugaboo. [En línea] [www.bugaboo.com/learn/bugaboo](http://www.bugaboo.com/learn/bugaboo).
17. sourceforge. [En línea] <http://sourceforge.net/projects/zentrack/>.
18. Edgewall. [En línea] <http://trac.edgewall.org>.
19. **Rojas Morrell, Raúl.** 2007. Diseño de una aplicación para el Seguimiento de Errores de los productos software de la Facultad 7. Trabajo de diploma para optar por el título de ingeniero en ciencias informáticas.

20. **Molina, Yenisel.** Sistema para la Gestión de las No Conformidades durante las Pruebas de Calidad. Trabajo de diploma para optar por el título de ingeniero en ciencias informáticas.
21. CodeGear. [En línea] Borland Software Corporation. CodeGear from Bor  
<http://www.codegear.com/products/bds2006>.
22. **Mueller, John.** ANAYA MULTIMEDIA-ANAYA INTERACTIVA. 2002.
23. **Ayala Montero, Ramón - Dierking, Howard.** XML Iniciación y MSDN Magazine. 2001.
24. Xul [En línea] <http://www.xul.fr/xul-tools.html>.
25. Curso de javascript [En línea] [http://webdelprogramador.com/lenguajes\\_script/curso de javascript](http://webdelprogramador.com/lenguajes_script/curso_de_javascript).
26. MSDN Microsoft Developer Network. [En línea] 2008. Microsoft Corporation . MSDN Microsoft Developer Network . Visual Studio Developer C <http://msdn.microsoft.com/en-gb/vstudio/bb265237.aspx>.
27. sourceforgeide [En línea] [http://sourceforgeide.com/codigo web/komodo ide](http://sourceforgeide.com/codigo_web/komodo_ide).
28. freedownloadmanager.[En línea][http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(M%C3%8D\)\\_14720](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720).
29. Enterprise-Architect. [En línea] <http://ingenierioscomputacionipla.wordpress.com/2008/09/12/enterprise-architect-71-modelado-uml/>.
30. vico. [En línea] <http://www.vico.org/FormMentorOutsourcingUML.pdf>.
31. kmels. [En línea] <http://kmels.net/files/2009/uvg/cc2003/Resources/Contenidos/XP/xp.pdf>.
32. Programacion Extrema [En línea] <http://migueljaque.com/index.php/metodologias/xp/29-xp/63-programacionextrema>.
33. Arturoweb. [En línea] <http://arturoweb.wordpress.com/2008/02/14/scrum-metodologia-agil-de-desarrollo>.
34. Ejemplos\_clase. [En línea] [http://iie.fing.edu.uy/ense/asign/desasoft/practico/hoja8/ejemplos\\_clase2.pdf](http://iie.fing.edu.uy/ense/asign/desasoft/practico/hoja8/ejemplos_clase2.pdf).
35. **Jacobson, Ivar.** 2000. *El Proceso Unificado*.

- 
36. **Pelaez, Juan Carlos.** Blog de Juan Peláez en Geeks.ms. [En línea]  
<http://geeks.ms/blogs/jkpelaez/archive/2009/04/18/arquitectura-basada-en-componentes.aspx>.
  37. **Fuentes, Lidia. Troya, José y Vallecillo, Antonio.** ETSI Informática. [En línea]  
<http://www.lcc.uma.es/~av/Docencia/Doctorado/tema1.pdf>.
  38. UML y patrones. [En línea] <http://bibliodoc.uci.cu/pdf/reg00061.pdf>.
  39. Patrones-de-Diseño-Recopilación. <http://es.scribd.com/doc/17767790/Patrones-de-Diseño-Recopilación>
  40. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. La Habana : s.n., 2000.
  41. Managinf. <http://www.managinf.com/arquitectura.pdf>
  42. Introducción a la arquitectura. <http://download.microsoft.com/download/.../20051122-ARC-BA.ppt>
  43. The JavaScript resource. <http://www.javascript.com>

**ANEXOS**

Anexo 1

Flujo alterno: Pausar

<b>Flujo Alterno</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El probador accede al panel inferior del navegador Mozilla y selecciona la opción "Pausa" (fácilmente reconocible por dos líneas gruesas negras verticales).	2. El sistema pone en pausa la acción que se está realizando en ese momento.
3. El probador accede nuevamente al panel inferior del navegador Mozilla y selecciona la opción "Pausa"	4. El sistema continúa realizando la acción.
<b>El Caso de Uso termina</b>	

**Caso de uso:** Editar grabación

<b>Caso de uso</b>	
<b>CUS 4</b>	Editar grabación
<b>Propósito</b>	Editar una grabación previamente realizada.
<b>Actores:</b> Probador	
<b>Resumen:</b> El caso de uso se inicia cuando el probador decide editar una grabación. El caso de uso finaliza cuando el sistema archiva la grabación con los cambios de la edición.	
<b>Referencias</b>	RF6
<b>FlujoNormal de Eventos</b>	



Acción del actor	Respuesta del sistema
1. El probador accede al panel inferior del navegador Mozilla y selecciona la opción "Editar grabación" (fácilmente reconocible por una hoja blanca y un lápiz azul encima).	2. El sistema muestra un panel con las opciones de edición: subir, bajar, cortar o pegar cada línea de registro que compone las acciones realizadas en cada grabación.
3. El probador edita la grabación, seleccionando las opciones deseadas.	4. El sistema archiva la grabación con las modificaciones hechas por el probador.
<b>El Caso de Uso termina</b>	

**Caso de uso:** Guardar

Caso de uso	
<b>CUS 5</b>	Guardar
<b>Propósito</b>	Guardar una grabación.
<b>Actores:</b> Probador	
<b>Resumen:</b> El caso de uso se inicia cuando el probador decide guardar una grabación. El caso de uso finaliza cuando el sistema guarda la grabación en un archivo XML.	
<b>Referencias</b>	RF7
FlujoNormal de Eventos	
Acción del actor	Respuesta del sistema
1. El probador accede al panel inferior del navegador Mozilla y selecciona la opción "Guardar" (fácilmente reconocible por una carpeta con una flecha roja).	2. El sistema muestra un panel con las opciones de guardado, permitiéndole al probador seleccionar dónde desea guardar el archivo, en la PC o en otro medio externo de

	almacenamiento.
3. El probador selecciona la ubicación donde desea guardar la grabación y selecciona la opción guardar.	4. El sistema archiva la grabación con la extensión XML en la ubicación especificada por el probador.
<b>El Caso de Uso termina</b>	

**Caso de uso:** Abrir archivo

<b>Caso de uso</b>	
<b>CUS 6</b>	Abrir archivo
<b>Propósito</b>	Abrir un archivo previamente guardado.
<b>Actores:</b> Probador	
<b>Resumen:</b> El caso de uso se inicia cuando el probador decide abrir un archivo de una grabación. El caso de uso finaliza cuando el sistema carga el archivo de grabación y le permite al probador trabajar con dicho archivo, en funcionalidades como la edición y reproducción del mismo.	
<b>Referencias</b>	RF8
<b>FlujoNormal de Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El probador accede al panel inferior del navegador Mozilla y selecciona la opción "Abrir archivo" (fácilmente reconocible por una carpeta con una flecha verde).	2. El sistema muestra un panel para seleccionar el medio externo donde se encuentra el archivo que se desea abrir o cargar en la extensión.
3. El probador selecciona la ubicación del	4. El sistema cargar el archivo con la extensión

archivo.	XML de la ubicación especificada por el probador.
<b>El Caso de Uso termina</b>	

**Caso de uso:** Configurar opciones de grabación

<b>Caso de uso</b>	
<b>CUS 7</b>	Configurar opciones de grabación
<b>Propósito</b>	Configurar opciones de grabación
<b>Actores:</b> Probador	
<b>Resumen:</b> El caso de uso se inicia cuando el probador decide Configurar las opciones de las grabaciones que va a realizar. El caso de uso finaliza cuando el sistema guarda la configuración de las opciones de grabación realizadas por el probador.	
<b>Referencias</b>	RF9
<b>FlujoNormal de Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El probador accede al panel inferior del navegador Mozilla y selecciona la opción "Preferencias" (fácilmente reconocible por una panelito pequeño con unos botones).	2. El sistema muestra un panel para seleccionar las opciones de configuración correspondientes.
3. El probador configura las grabaciones seleccionando las opciones de su preferencia.	4. El sistema guarda la configuración de las opciones de grabación realizadas por el probador.
<b>El Caso de Uso termina</b>	

**Caso de uso:** Limpiar archivo de grabaciones

<b>Caso de uso</b>	
<b>CUS 8</b>	Limpiar archivo de grabaciones
<b>Propósito</b>	Limpiar las grabaciones que se encuentran archivadas en memoria dentro de la extensión.
<b>Actores:</b> Probador	
<b>Resumen:</b> El caso de uso se inicia cuando el probador decide limpiar las grabaciones que se encuentran archivadas en memoria dentro de la extensión. El caso de uso finaliza cuando el sistema limpia las grabaciones realizadas previamente por el probador, residentes en memoria.	
<b>Referencias</b>	RF10
<b>FlujoNormal de Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El probador accede al panel inferior del navegador Mozilla y selecciona la opción "Limpiar todo" (fácilmente reconocible por una hoja y una flecha curva azul).	2. El sistema muestra un panel con un mensaje de alerta para verificar si realmente se desea limpiar el archivo de grabaciones.
3. El probador selecciona la opción Aceptar.	4. El sistema limpia las grabaciones realizadas previamente por el probador, residentes en memoria.
<b>El Caso de Uso termina</b>	

---

## GLOSARIO DE TÉRMINOS

**Calidad del software:** La calidad del software es definida como la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente.

**Diagrama de Clases:** Representación de los conceptos de importancia en el área de la aplicación, así como de las relaciones entre estos.

**Extensible Markup Language (XML):** Un lenguaje de marcado extensible que puede usarse para almacenar datos en un formato estructurado, basado en texto y definido por el usuario.

**J' Pruebas:** Persona encargada de asignar las tareas de pruebas a los probadores, así como de supervisar el trabajo en una sesión de pruebas.

**No Conformidades (NC):** Según la definición de ISO9000: 2000 (3.6.2), una no conformidad es el "incumplimiento de un requisito". O un error detectado en la documentación durante la revisión.

**Probador:** Persona encargada de registrar y detectar NC.

**Proceso de pruebas:** El proceso de pruebas consiste en demostrar que los artefactos generados están libres de fallos". El proceso de pruebas consiste, básicamente, en la realización de una serie de pruebas al código que tengamos para observar con qué frecuencia podría fallar ese código, encontrar esos errores y solucionarlos. Estas pruebas hay que realizarlas al comienzo del ciclo de vida del software ya que, cuando avanza en el ciclo de vida, más costoso sería solucionar el fallo encontrado.

**Prueba de software:** Las pruebas de software, en inglés *testing* son los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador.

**Redmine:** Gestor y planificador de proyectos con interfaz web, orientado a la coordinación de tareas, comunicación de participantes, y que puede especializarse en proyectos de desarrollo gracias a herramientas como la integración en un repositorio de código.

**Registro de No Conformidades (RNC):** Documento donde se registran todas las no conformidades que se le detectan al producto.

**Ubicación:** Puede ser un link, URL o descripción manual que posibilite encontrar donde se encuentra una NC.

**XUL:** Lenguaje basado en XML para la interfaz de usuario, es la aplicación de XML a la descripción de la interfaz de usuario en el navegador Mozilla.