

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 6



# Liberación del Visualizador y Editor Molecular del proyecto bioGRATO

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO INFORMÁTICO

## **Autoras:**

Yasenia Rodríguez Garrido

Yarmay Tamayo García

## **Tutores:**

MS.c Aurelio Antelo Collado

Ing. Ileana Martí Pérez

Ing. Martha Nieves Borrero

**JUNIO 2008**

# DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los    días del mes    de del año    .

Yarmay Tamayo García

Yasenia Rodríguez Garrido

Firma del Autor

Firma del Autor

MSc. Aurelio Antelo ColladoIng.

Ing. Ileana Martí Pérez

Firma del Tutor

Firma del Tutor

Ing. Martha Nieves Borrero

Firma del Tutor

# DATOS DEL CONTACTO

## **Tutores:**

### **Aurelio Antelo Collado.**

Ingeniero Industrial.

Master en Matemática Aplicada.

Profesor Asistente con 3 años de experiencia.

Email: aantelo@uci.cu

### **Ileana Martí Pérez**

Ingeniera en Ciencias Informáticas.

Email: imarti@uci.cu

### **Martha Nieves Borrero**

Ingeniera en Ciencias Informáticas.

Email: mnieves@uci.cu

# PENSAMIENTO



*La calidad nunca es un accidente; siempre es el resultado de un esfuerzo de la inteligencia.*

**John Ruskin**

# AGRADECIMIENTO

*Quiero agradecer principalmente a mis **padres** y a mi **hermanita** que me apoyaron todos estos años, por su infinito amor, cariño y comprensión. Por soportar estos años que estuve lejos de ellos, por acompañarme en los buenos y malos momentos. Por ayudarme a que este día llegara, sin ustedes nada fuera posible.*

*A mi tutor **Aurelio**, por haberme tenido la paciencia necesaria, gracias por tu dedicación y apoyo incondicional para que este trabajo fuera posible.*

*A **Mabe**, que más que una amiga ha sido una hermana para mí, por ayudarme y apoyarme durante todos estos años y por poder contar siempre contigo, gracias por tu gran amistad.*

*A mi compañera de tesis **Yarmay**, por su apoyo y amistad.*

*A todos mis **amigos**, en especial a las chicas del apto, a Julio y a los chicos de la tecnología, por estar conmigo en todo este tiempo donde hemos compartido momentos agradables y momentos tristes, que son los que nos hacen crecer y valorar a las personas que nos rodean, gracias por ser mis amigos, los quiero mucho y siempre los llevaré en mi corazón.*

*A toda mi **familia** por el apoyo, amor y confianza que han depositado en mí.*

*A mis tíos **Nélida** y **Pavón** que me han ayudado tanto durante todos estos años.*

*A **Gladys**, **Edel**, **Liam** y **Lianet** porque siempre he podido contar con ellos y han sido para mí como mi propia familia.*

*En fin, a todas aquellas personas que de una forma u otra estuvieron a mi lado y contribuyeron a que este día llegara, muchas gracias.*

**Yasenia**

# AGRADECIMIENTO

*Son muchas las personas que de una forma u otra pasan por nuestras vidas , dejando huellas imborrables, incontables experiencias que quedarán en el recuerdo, pero que persistirán haciendo nuestro mundo más placentero, no quisiera dejar pasar este momento sin antes agradecer a todas las personas que hicieron posible que llegara este día.*

*Primeramente quisiera agradecerle a una persona que me ha brindado todo su cariño, que gracias a su paciencia y esfuerzo me he convertido en la mujer que soy hoy, para tí **mami** todo mi amor.*

*A mi **papá**: Por estar siempre presente, y por ayudarme tanto en todo momento, gracias papi.*

*A mi **hermanita**: A esa personita que es la luz de mis ojos, sin tí no sabría vivir, te quiero.*

*A **Miguel Angel**: Por pasar todo este tiempo junto a mí, por apoyarme en tantos momentos difíciles y hacerme feliz.*

*A **Marianita y Humberto**: Por brindarme todo su apoyo y amor.*

*A mi tutor **Aurelio**: por tu inmensa paciencia y dedicación, sin tí nada hubiese sido posible.*

*A **Anders y Yauseny**: Por escucharme, por ayudarme tanto, por sus consejos y por saber ser verdaderos amigos, gracias por ser mi conciencia.*

*A mi compañera de tesis **Yasenia**: Por ayudarme tanto cuando más lo necesitaba.*

*A todos mis **amigos**: Por posibilitarme pasar parte de mi vida con ustedes y tener un lugarcito en mi corazón, en especial a las chicas sexy del apto, a Julio y los niños lindos de soporte técnico, a todos muchas gracias.*

**Yarmay**

# DEDICATORIA

*Dedico esta tesis con todo mi amor y cariño a las personas más importantes en mi vida:*

*A mis **padres**, que me dieron la vida y han estado conmigo en todo momento brindándome su inmenso amor, dedicación y apoyo, este trabajo es para ustedes.*

*A mi **hermanita**, gracias por estar conmigo y apoyarme siempre, te quiero mucho.*

**Yasenia**

*Quiero dedicar esta tesis con todo mi amor:*

*A mi **mamá**, mi **papá** y a mi **hermanita**, por permitirme estar siempre junto a ustedes.*

*A mi pequeña **Moniquita**, por ser la personita que ha llenado de felicidad mi corazón.*

**Yarmay**

# RESUMEN

La necesidad de lograr un software de alta calidad es primordial para la sociedad actual, pues cada día se hacen mayores las exigencias de los clientes, así como la competitividad de las empresas en el mercado mundial. Con la aplicación de las pruebas de calidad podemos garantizar la calidad de un proyecto en desarrollo, estas pruebas representan una revisión detallada de cada uno de los componentes y se aplican en todas las fases de desarrollo del producto.

En la facultad 6 de la Universidad de las Ciencias Informáticas se ha creado una plataforma para la predicción de actividad biológica en compuestos orgánicos en la que se ha desarrollado un Visualizador y Editor Molecular que debe ser capaz de cargar, editar y visualizar moléculas en diferentes formatos.

Es por eso que en el presente trabajo se hace un estudio de los principales temas de calidad y la documentación referente al proyecto. Se diseñan los casos de prueba y se definen los pasos a seguir para desarrollar las pruebas de calidad. Además se define un plan de pruebas para garantizar la organización de este proceso. Se documentaron las no conformidades encontradas y se realizaron dos iteraciones para poder corregir todos los errores encontrados y comparar los resultados encontrados con los esperados por el software, quedando el producto listo para ser entregado al grupo de calidad de la facultad.

## **PALABRAS CLAVES**

Carga, Pruebas, Instalación, Integración, Caja Negra, Caja Blanca, Casos de Prueba, Calidad de Software

# Índice general

<b>AGRADECIMIENTO</b>	<b>I</b>
<b>DEDICATORIA</b>	<b>III</b>
<b>RESUMEN</b>	<b>IV</b>
<b>ÍNDICE GENERAL</b>	<b>IV</b>
<b>LISTA DE FIGURAS</b>	<b>VIII</b>
<b>INTRODUCCIÓN</b>	<b>1</b>
<b>1. FUNDAMENTACIÓN TEÓRICA</b>	<b>5</b>
1.1. INTRODUCCIÓN . . . . .	5
1.2. CALIDAD DE SOFTWARE. CONCEPTOS Y DEFINICIONES . . . . .	5
1.3. LAS PRUEBAS EN LA CALIDAD DEL SOFTWARE . . . . .	8
1.3.1. ¿QUÉ SON LAS PRUEBAS DE CALIDAD DE SOFTWARE? . . . . .	8
1.4. LAS PRUEBAS EN EL DESARROLLO DEL SOFTWARE . . . . .	8
1.4.1. OBJETIVOS DE LAS PRUEBAS DE CALIDAD . . . . .	8
1.4.2. PRINCIPIOS DE PRUEBAS . . . . .	9
1.4.3. ESTRATEGIAS DE PRUEBAS . . . . .	9
1.5. EL PROCESO DE PRUEBA. ELEMENTOS FUNDAMENTALES . . . . .	10
1.5.1. EL PROCESO DE PRUEBA . . . . .	10
1.5.2. NIVELES DE PRUEBAS . . . . .	11
1.6. MÉTODOS DE PRUEBAS . . . . .	14

1.6.1.	PRUEBA DE CAJA BLANCA . . . . .	14
1.6.1.1.	TÉCNICA DE CAJA BLANCA. PRUEBA DEL CAMINO BÁSICO . . . . .	15
1.6.1.2.	TÉCNICA DE CAJA BLANCA. PRUEBA DE BUCLES . . . . .	16
1.6.2.	PRUEBAS DE CAJA NEGRA . . . . .	17
1.6.2.1.	TÉCNICA DE CAJA NEGRA. PARTICIÓN EQUIVALENTE . . . . .	18
1.6.2.2.	TÉCNICA DE CAJA NEGRA. ANÁLISIS DE VALORES LÍMITE . . . . .	19
1.7.	PLAN DE PRUEBA . . . . .	19
1.8.	HERRAMIENTAS PARA EL DESARROLLO DE LAS PRUEBAS DE SOFTWARE . . . . .	20
1.9.	PROCEDIMIENTO DE PRUEBA . . . . .	21
1.10.	DISEÑO DE CASOS DE PRUEBA . . . . .	22
1.11.	METODOLOGÍA OPENUP . . . . .	23
1.11.1.	ROL DE PROBADOR . . . . .	24
1.12.	CONCLUSIONES . . . . .	26
<b>2.</b>	<b>DISEÑO Y APLICACIÓN DE PRUEBAS DE SOFTWARE</b>	<b>27</b>
2.1.	INTRODUCCIÓN . . . . .	27
2.2.	CARACTERÍSTICAS A PROBAR . . . . .	27
2.3.	DISEÑO DEL PLAN DE PRUEBA . . . . .	28
2.4.	ESTRATEGIA DE PRUEBAS . . . . .	29
2.5.	CONFIGURACIÓN DEL ENTORNO DE PRUEBA . . . . .	29
2.6.	CRONOGRAMA DE PRUEBA . . . . .	30
2.7.	PROCEDIMIENTOS DE PRUEBAS . . . . .	32
2.7.1.	PROCEDIMIENTOS DE PRUEBAS PARA EL VISUALIZADOR . . . . .	32
2.7.1.1.	CASO DE USO: ABRIR FICHERO . . . . .	32
2.7.1.2.	CASO DE USO: GESTIONAR FICHERO . . . . .	33
2.7.1.3.	CASO DE USO: GESTIONAR VISUALIZACIÓN . . . . .	34
2.7.1.4.	CASO DE USO: GESTIONAR MÓDULOS . . . . .	40
2.7.1.5.	CASO DE USO: CALCULAR PROPIEDADES . . . . .	41
2.7.2.	PROCEDIMIENTOS DE PRUEBAS PARA EL EDITOR . . . . .	41
2.7.2.1.	CASO DE USO: CREAR MOLÉCULA . . . . .	41
2.7.2.2.	CASO DE USO: MODIFICAR MOLÉCULA . . . . .	41

2.8. CASOS DE PRUEBAS . . . . .	42
2.8.1. PRIMERA ITERACIÓN . . . . .	42
2.8.1.1. PRUEBAS DE CAJA NEGRA PARA EL VISUALIZADOR . . . . .	42
2.8.2. PRUEBAS DE CAJA NEGRA PARA EL EDITOR MOLECULAR. . . . .	60
2.8.3. PRUEBAS DE CAJA BLANCA PARA EL VISUALIZADOR. . . . .	67
2.8.3.1. MÉTODO OPEN . . . . .	67
2.8.3.2. MÉTODO SAVEPREFERENCES . . . . .	70
2.8.3.3. MÉTODO CHEQUEAMOLPOS . . . . .	71
2.8.3.4. MÉTODO ELIMINAREXTENSIÓN . . . . .	72
2.8.3.5. MÉTODO CHEQUEATABEXIST . . . . .	73
2.8.3.6. MÉTODO UPDATETABLE . . . . .	74
2.8.4. PRUEBA DE INTEGRACIÓN. . . . .	75
2.8.5. PRUEBA DE CARGA. . . . .	76
2.8.6. PRUEBA DE INSTALACIÓN. . . . .	78
2.9. SEGUNDA ITERACIÓN . . . . .	80
2.10. CONCLUSIONES . . . . .	81
<b>3. RESULTADOS DE LAS PRUEBAS . . . . .</b>	<b>82</b>
3.1. INTRODUCCIÓN . . . . .	82
3.2. ANÁLISIS DE LOS RESULTADOS OBTENIDOS EN LAS PRUEBAS DE CAJA NEGRA EN UNA PRIMERA ITERACIÓN. . . . .	82
3.2.1. REGISTRO DE DEFECTOS Y DIFICULTADES DETECTADAS PARA EL VISUALIZADOR . . . . .	82
3.2.2. REGISTRO DE DEFECTOS Y DIFICULTADES DETECTADAS PARA EL EDITOR MOLECULAR: . . . . .	96
3.3. ANÁLISIS DE LOS RESULTADOS OBTENIDOS EN LAS PRUEBAS DE CAJA BLANCA EN UNA PRIMERA ITERACIÓN. . . . .	97
3.4. ANÁLISIS DE LOS RESULTADOS OBTENIDOS EN LAS PRUEBAS DE INTEGRACIÓN EN UNA PRIMERA ITERACIÓN. . . . .	97
3.5. ANÁLISIS DE LOS RESULTADOS OBTENIDOS EN LAS PRUEBAS DE CARGA EN UNA PRIMERA ITERACIÓN. . . . .	98

3.6. ANÁLISIS DE LOS RESULTADOS OBTENIDOS EN LAS PRUEBAS DE INSTALACIÓN EN UNA PRIMERA ITERACIÓN. . . . .	99
3.7. ANÁLISIS DE LOS RESULTADOS OBTENIDOS EN LAS PRUEBAS EN LA SEGUNDA ITERACIÓN. . . . .	99
3.8. CONCLUSIONES . . . . .	99
<b>CONCLUSIONES</b>	<b>101</b>
<b>RECOMENDACIONES</b>	<b>102</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>103</b>
<b>BIBLIOGRAFÍA</b>	<b>104</b>

# Índice de figuras

1.1. Calidad en el ciclo de vida del software . . . . .	6
1.2. Ciclo de Vida de Pruebas (Modelo en V) . . . . .	11
1.3. Grafos de flujo para las instrucciones If, Secuencia, While y Case. . . . .	15
1.4. Representación de Bucles simples, anidados, concatenados y estructurados. . . . .	16
1.5. Técnicas de diseño de casos de Prueba. . . . .	23
1.6. Capas de OpenUP: micro-incrementos, ciclo de vida de la iteración y ciclo de vida del proyecto. . . . .	23
1.7. Representación de la relación entre los artefactos del rol de probador. . . . .	24



# INTRODUCCIÓN

El desarrollo de la industria del software en el mundo ha evolucionado a un ritmo acelerado, sin embargo no se han obtenido los resultados que realmente se esperaban, convirtiéndose así en un tema crítico para la sociedad actual. Esta situación se debe al crecimiento del volumen y complejidad de los productos, los proyectos están excesivamente tarde, se exige mayor calidad y productividad en menos tiempo y hay insuficiente personal calificado. La obtención de un software implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba que permitan uniformar la filosofía de trabajo y que a la vez eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del mismo. Por eso hay que garantizar un conjunto de cualidades que caractericen y determinen su utilidad y existencia. Obteniendo así un software con mayor eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

Las compañías dedicadas al desarrollo de Software, inician una etapa de calidad en sus sistemas de punto de venta, considerado como un elemento crítico. Esta percepción en el Software es el reflejo inherente de los problemas presentados en el área de Informática: plazos y presupuestos incumplidos, insatisfacción del usuario, escasa productividad y la baja calidad en los proyectos. En pocas palabras consiste en tratar de cambiar la idea de un simple proceso no sistemático a un proceso determinado por la planificación y la metodología automatizada.

El proceso de pruebas forma la parte más importante de la gestión de la calidad de software. En su corta historia la prueba de software ha evolucionado mucho: comenzó como un proceso de depuración para pasar a ser un conjunto de actividades dirigidas a la medición del grado de calidad de los productos software. Por supuesto, la preparación y la ejecución de estas actividades requieren una buena coordinación. Para que cada actividad se justifique por sus resultados finales hace falta una buena planificación, así como una evaluación y retroalimentación continuas. Desde los años 70, este tema ha sido motivo de preocupación para

especialistas, ingenieros, investigadores y comercializadores de software, los cuales han realizado gran cantidad de investigaciones al respecto.

En Cuba el desarrollo de una Industria Nacional de Software es una tarea de gran prioridad para el estado cubano debido a la alta perspectiva económica que posee, así como para el aseguramiento de un grupo importante de actividades del país. A pesar de ello, los resultados alcanzados no cubren las expectativas, ya que la productividad es baja, la cantidad real de recursos a consumir (en tiempo principalmente) es impredecible y el trabajo realizado casi nunca tiene la calidad y profesionalidad requerida. Los proyectos se terminan demasiado tarde y los beneficios que pudieran obtenerse al utilizar los mejores métodos e instrumentos en las distintas etapas, no se detectan en este medio indisciplinado y caótico de desarrollo.

Para comenzar a resolver estos problemas no basta con la aplicación de nuevas metodologías y tecnologías para desarrollar y mantener software, es necesario también aprender a administrar procesos de software para que se realicen con calidad. Todo esto requiere el establecimiento de sistemas de calidad y por tanto definir rigurosamente los procesos de software que se llevan a cabo en la empresa.

En la Universidad de las Ciencias Informáticas, específicamente en la Facultad 6, se trabaja en la creación de una plataforma para la predicción de actividad biológica en compuestos orgánicos, en la cual se desarrollan los productos Visualizador y Editor Molecular. Ellos deben ser capaces de cargar y guardar moléculas en distintos formatos y editar las mismas, deben tener una rápida visualización y además ser confiables, cumpliendo con un conjunto de exigencias planteadas por el Centro de Química Farmacéutica, el cual es el cliente potencial de la plataforma.

Por lo antes expuesto el **problema a resolver** consiste en: ¿Cómo verificar los requerimientos de software para el Visualizador y Editor Molecular de la plataforma bioGRATO? Para ello se propone realizar un adecuado diseño de pruebas, que permita conocer los errores sobre los cuales trabajar para así lograr un sistema más estable y eficiente.

Por lo tanto el **objeto de estudio** de la investigación es la calidad de un producto de software, enmarcando el **campo de acción** en pruebas de calidad de software.

Como **objetivo general** de este trabajo se tiene:

Demostrar el cumplimiento de los requerimientos de software planteados para el Visualizador y Editor Molecular de la plataforma bioGRATO.

Del cual se derivan los siguientes **objetivos específicos**:

- Diseñar el plan de pruebas.
- Diseñar las pruebas necesarias para el Visualizador y Editor Molecular.
- Aplicar las pruebas al Visualizador y Editor Molecular.
- Documentar los resultados obtenidos de la aplicación de las pruebas.

Para el cumplimiento de los objetivos propuestos se definieron una serie de **tareas** entre las cuales se encuentran:

- Estudio de aspectos importantes vinculados a la calidad de software.
- Selección de las pruebas de calidad a aplicar.
- Estudio de la documentación necesaria del Visualizador y Editor Molecular para diseñar las pruebas.
- Configuración del entorno para la ejecución de las pruebas.
- Selección de los datos necesarios para hacer las pruebas.
- Ejecución de las pruebas diseñadas.
- Recogida de las no conformidades y las solicitudes de cambios .
- Elaboración del resumen de los resultados de las pruebas .

La tesis está estructurada de la siguiente forma:

## **Capítulo 1. Fundamentación Teórica**

En este capítulo se hace una descripción de los aspectos más importantes vinculados a la calidad de software, así como conceptos y definiciones que ayudan a la comprensión de este tema. Se aborda acerca de las pruebas de calidad haciendo alusión a los principios, objetivos y la estrategia a seguir en el proceso de pruebas, así como la diferentes pruebas que se pueden aplicar para determinar la calidad de un producto software y se definen cuáles son las pruebas que se van aplicar al Visualizador y Editor Molecular. Se da una breve explicación de los métodos de prueba más utilizados y se determina la Metodología a utilizar.

## **Capítulo 2. Diseño y aplicación de pruebas de software**

En este capítulo se hace una breve descripción del producto. Partiendo de la elaboración de un plan de pruebas, donde se recogen determinados aspectos importantes y necesarios para el proceso de pruebas, se diseñan los casos de prueba y se aplican las pruebas ya definidas con anterioridad al Visualizador y Editor Molecular y se recogen las no conformidades encontradas.

## **Capítulo 3. Resultados de las Pruebas**

En este capítulo se describen los resultados obtenidos después de aplicarse cada una de las pruebas diseñadas al Visualizador y Editor Molecular, haciendo una breve descripción de las no conformidades encontradas. Se hace un análisis general de los resultados que se obtuvieron después de todo el proceso de pruebas.

# Capítulo 1

## FUNDAMENTACIÓN TEÓRICA

### 1.1. Introducción

En este capítulo se abordan conceptos, definiciones y aspectos muy importantes para la comprensión de este trabajo. Se tratan temas relacionados con la calidad y las pruebas de software como son el aseguramiento y control de la calidad, los principios y estrategias de pruebas a seguir en el desarrollo de un software, los niveles, procedimientos y técnicas que se utilizan para el diseño y aplicación de los casos de pruebas, entre otros aspectos de merecida importancia y se plantean las pruebas que serán aplicadas al Visualizador y Editor Molecular. También se hace alusión a la metodología empleada así como los métodos de prueba más utilizados.

### 1.2. Calidad de Software. Conceptos y Definiciones

El software se ha convertido hoy en día en uno de los principales objetivos estratégicos de las organizaciones, debido a que cada vez más, los procesos principales de las organizaciones y su supervivencia, dependen de los sistemas informáticos para su buen funcionamiento. Es por ello que el software de alta calidad es una meta que todos los desarrolladores de software deben tener, pero ¿qué es calidad de software?

Según el estándar de la IEEE[1] la calidad de software "es el grado con el que un sistema, componente o proceso cumple los requerimientos y las necesidades o expectativas del cliente o usuario". Siendo una compleja mezcla de factores que variarán a través de diferentes aplicaciones y según los clientes que las pidan.

Para Pressman[2] es 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’’.



Figura 1.1: Calidad en el ciclo de vida del software

El modelo ISO/IEC 9126 presenta el concepto de calidad y requerimientos de calidad como un paso necesario para la revisión del producto descompuesto en la calidad interna, externa y en uso[3]. En la figura 1, se puede apreciar que las necesidades de calidad del usuario sobre el producto software, contribuyen a especificar los requerimientos de calidad externa y estos a su vez los requerimientos de calidad interna. Es válido señalar que según Cueva[4] los requisitos de software son la base de las medidas de calidad.

Otro concepto que da Cueva[5] y que está relacionado con la calidad del software es el Aseguramiento de la Calidad del Software (Software Quality Assurance (SQA)) que es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) podrá satisfacer los requisitos dados de calidad.

El aseguramiento de la calidad del software se diseña para cada aplicación antes de comenzar a desarrollarla y no después. Según Cueva[6], este está presente en:

- Métodos y herramientas de análisis, diseño, programación y prueba.
- Inspecciones técnicas formales en todos los pasos del proceso de desarrollo del software.
- Estrategias de prueba multiescala.

- Control de la documentación del software y de los cambios realizados.
- Procedimientos para ajustarse a los estándares (y dejar claro cuando se está fuera de ellos).
- Mecanismos de medida (métricas).
- Registro de auditorías y realización de informes.

Y el mismo presenta las siguientes actividades:

- Métricas de software para el control del proyecto.
- Verificación y validación del software a lo largo del ciclo de vida.
- La gestión de la configuración del software.

Otro aspecto importante en la calidad de un producto software es la Gestión de la Calidad del Software que según la ISO 9000[7], es el conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades y se implanta a través de la planificación de la calidad, el control de la calidad, el aseguramiento de la calidad y la mejora de la calidad, en el marco del sistema de calidad.

El Control de Calidad es una actividad que se tiene que llevar a cabo para garantizar la calidad de todo producto o servicio que se ofrece a los consumidores, en las siguientes líneas se presenta este concepto. Aplicado al software, el control de calidad es una serie de inspecciones, revisiones, y pruebas utilizados a lo largo del ciclo de desarrollo para asegurar que cada producto cumple con los requisitos que le han sido asignados, centradas según Cueva[8], en dos objetivos fundamentales:

- Mantener bajo control un proceso.
- Eliminar las causas de los defectos en las diferentes fases del ciclo de vida.

Las actividades del control de calidad pueden ser manuales, completamente automáticas o una combinación de herramientas automáticas e interacción humana. La calidad de un sistema software es algo subjetivo que depende del contexto y del objeto que se pretenda conseguir. Para determinar dicho nivel de calidad se deben efectuar unas pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema.

La fase de pruebas es una de las más costosas del ciclo de vida software. En sentido estricto, deben realizarse pruebas de todos los artefactos generados durante la construcción de un producto, lo que incluye especificaciones de requisitos, casos de uso, diagramas de diversos tipos y, por supuesto, el código fuente y el resto de productos que forman parte de la aplicación. Obviamente, se aplican diferentes técnicas de prueba a cada tipo de producto software.

### **1.3. Las Pruebas en la Calidad del Software**

#### **1.3.1. ¿Qué son las Pruebas de Calidad de Software?**

La prueba de software es un proceso que corre en paralelo al proceso de desarrollo del software, usado para identificar posibles fallos de implementación, calidad, o usabilidad de un software y que se realiza por el convencimiento de que todo sistema debe ser “revisado” con el objetivo de establecer el nivel de calidad requerido. Básicamente es una fase en el desarrollo de software que consiste en probar las aplicaciones construidas. Las mismas se integran dentro de las diferentes fases del ciclo del software dentro de la Ingeniería de software.

Según Acuña[9] podemos concluir que una prueba de calidad de software es una actividad en la cual un sistema o uno de sus componentes se ejecutan en circunstancias previamente especificadas, los resultados se observan, registran y se realiza una evaluación de algún aspecto.

### **1.4. Las Pruebas en el Desarrollo del Software**

#### **1.4.1. Objetivos de las Pruebas de Calidad**

Las pruebas de calidad a un software han evolucionado mucho más allá de simplemente depurar la aplicación y posteriormente migrala a producción para convertirse en un proceso integrado, automatizable y administrable que se realiza desde el momento mismo del inicio de la codificación y se extiende hasta la entrega final al cliente. El objetivo de estas pruebas según Méndez[10] es evaluar la calidad del producto a lo largo de todo el ciclo de vida apoyándose en un conjunto de buenas prácticas, entre las que destacan:

- Verificar que el producto de software trabaja según el diseño.
- Validar que los requerimientos son implementados apropiadamente.

- Encontrar defectos en el software con un mínimo de costo y tiempo.
- Detallar los errores que fueron encontrados.
- Ofrecer un producto altamente seguro y confiable.

### 1.4.2. Principios de Pruebas

Para la aplicación de métodos para el diseño de casos de prueba efectivos, se deben conocer los principios básicos que guían las pruebas del software, Valencia[11] hace referencia a los siguientes:

1. La prueba es el proceso de ejecutar un programa con la intención de encontrar errores (parece un enfoque destructivo. Actitud correcta: una prueba exitosa es aquella que encuentra un error).
2. Es imposible probar completamente cualquier módulo no trivial o cualquier sistema.
3. La prueba implica creatividad y trabajo duro.
4. La prueba puede prevenir posibles errores, cuando se realizan a través de diferentes etapas del ciclo de vida.
5. Es mejor que las pruebas sean realizadas por personas diferentes a quienes hicieron el desarrollo del sistema.

### 1.4.3. Estrategias de Pruebas

En algunos casos indican que en la disciplina de pruebas, generalmente no se implementa de forma organizada y sistemática. Además, algunos autores afirman que el proceso de ejecución de pruebas debe ser considerado durante todo el ciclo de vida de un proyecto, para así obtener un producto de alta calidad. Su éxito dependerá del seguimiento de una estrategia de prueba adecuada.

La estrategia de prueba de software integra un conjunto de actividades que describen los pasos que hay que llevar a cabo en un proceso de prueba, tomando en consideración cuánto esfuerzo y recursos se van a requerir, con el fin de obtener como resultado una correcta construcción del software.

## 1.5. El Proceso de Prueba. Elementos Fundamentales

### 1.5.1. El Proceso de Prueba

El proceso de prueba debe comenzar en la fase misma de la definición de requerimientos y terminar con la finalización de la aplicación de los casos de prueba al sistema ejecutable. Esto es, la prueba de software representa un subproceso que permea prácticamente todo el proceso de desarrollo de software:

Fase de Requerimientos: se verifica si se siguen estándares y si los requerimientos son completos, consistentes, rastreables (traceable), probables (testable), y modulares.

Entradas : estándar de requerimientos; documento(s) de requerimientos.

Salidas: recomendaciones para que el documento de requerimientos presente las características deseables mencionadas arriba.

Fase de Diseño: verificar si las propuestas de diseño satisfacen los requerimientos, si presentan características deseables, y si se siguieron los estándares de diseño. En particular, verificar si la arquitectura del sistema y de la base de datos son apropiadas, si los diseños tienen alta cohesión y bajo acoplamiento, y si la base de datos tiene la normalización adecuada (no redundancia ni problemas de integridad referencial).

Entradas: estándar de diseño; documento(s) de diseño; documento(s) de requerimientos.

Salidas: recomendaciones para que el documento de diseño presente las características deseables mencionadas arriba; documento(s) con refinamientos de los planes y casos de prueba de la fase anterior, más diseños preliminares de casos de prueba de interacción.

Fase de Construcción (programación): verificar si se siguen estándares, si los programas son consistentes con el diseño, y si el código es modular, además de cohesivo y bajo en acoplamiento; eventualmente, verificar el nivel y calidad de rehúso.

Pressman[12] plantea que el proceso de pruebas de software consta de las siguientes etapas:

1. Inspección del análisis (verifica si se cometieron errores o fallas en la etapa de análisis).

2. Inspección del diseño (debe ser completo y eficiente).
3. Inspección del código (observar el entendimiento y facilidad del código).
4. Pruebas unitarias (probar cada método de las clases implementadas por separado).
5. Pruebas de integración (probar todas las clases, verificando que compaginen entre sí).
6. Pruebas de validación de requerimientos (verificar que cumple con todos los requerimientos exigidos por el cliente).
7. Pruebas de sistema (ejecutar el programa para verificar si cumple con los requisitos exigidos).

### 1.5.2. Niveles de Pruebas

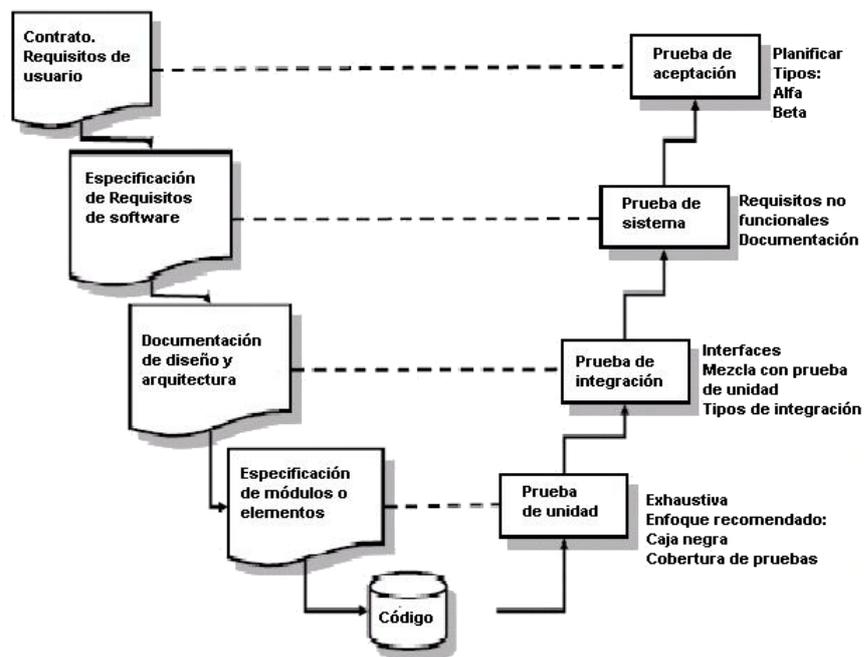


Figura 1.2: Ciclo de Vida de Pruebas (Modelo en V)

Existen diferentes niveles y tipos de pruebas entre los cuales se encuentran:

#### Pruebas de Unidad:

Pueden abarcar desde un módulo, a un grupo de pocos módulos o un programa completo y suelen ser realizadas por personal de desarrollo. Estas pruebas se abordan principalmente con criterios de caja negra y caja blanca.

En el primer caso, la idea de probar una interface sin conocer detalles de su implementación concreta coincide plenamente con uno de los principios básicos de la Orientada a Objeto, la encapsulación. Por otro lado disponiendo del código fuente de los módulos, existe la posibilidad de aplicar técnicas de caja blanca midiendo los grados de cobertura de sentencia, de decisión, de condición, etc., respecto a las pruebas de caja negra anteriormente diseñadas e implementadas.

### **Pruebas de Integración:**

Se comprueba la compatibilidad y funcionalidad de las interfaces entre las distintas partes que componen un sistema, estas partes pueden ser módulos, aplicaciones individuales, aplicaciones cliente/servidor, etc. Este tipo de pruebas es especialmente relevante en aplicaciones distribuidas.

Tipos de Integración:

- Incremental (ascendente y descendente) Se combina el siguiente módulo que se debe probar con el conjunto de módulos que ya están probados.
- No Incremental Se prueba cada módulo por separado, luego se integran todos de una vez y se prueba el programa completo.

### **Pruebas de Validación:**

Las pruebas de validación para el software se enfocan en las acciones visibles por el usuario además de las salidas del sistema que puedan ser reconocidas por él. Dichas acciones y salidas engloban las expectativas razonables del usuario y están definidas en las especificaciones de los requerimientos del software. La derivación de las pruebas de validación está basada en los casos de uso contenidos en el modelo de casos de uso de la fase de diseño.

Existen dos tipos de requerimientos que deben tomarse en cuenta en la planificación de las pruebas de validación. Los primeros son los requerimientos funcionales, tomados a partir del modelo de casos de uso como se mencionó anteriormente. Los segundos son los llamados no-funcionales, entre los cuales podemos mencionar:

- Desempeño, que tiene que ver con el espacio, tiempo de ejecución y otros recursos del sistema que requiera el software para ejecutarse.
- Documentación, la cual debe estar correcta, inteligible, coherente y completa.
- Volumen de datos, el cual se refiere a la cantidad de carga que puede y debe soportar el sistema.
- Demanda pico, que está relacionado con la tensión o stress que debe soportar el sistema en intervalos de tiempo.
- Recuperabilidad, asociado a la capacidad de respaldo del sistema ante cualquier error de ejecución; por ejemplo, si el sistema maneja archivos de datos que modifica constantemente debe constar de módulos que se encarguen de guardar siempre los archivos antes de utilizarlos; otro ejemplo de recuperabilidad podría ser la capacidad del sistema para restablecer las condiciones finales antes del error de ejecución.
- Usabilidad, pues hay que considerar ergonomía del sistema, es decir, facilidad de uso; por ejemplo, se debe tomar en cuenta las condiciones en las cuales el usuario tendrá que trabajar con el sistema para que su jornada no se le haga pesada sino más bien agradable.

### **Pruebas de Sistema:**

El software ya validado se integra con el resto del sistema donde algunos tipos de pruebas a considerar son:

- Rendimiento: determinan los tiempos de respuesta, el espacio que ocupa el módulo en disco o en memoria, el flujo de datos que genera a través de un canal de comunicaciones, etc.
- Resistencia: determinan hasta donde puede soportar el programa determinadas condiciones extremas.
- Robustez: determinan la capacidad del programa para soportar entradas incorrectas. Seguridad: se determinan los niveles de permiso de usuarios, las operaciones de acceso al sistema y acceso a datos.
- Usabilidad: se determina la calidad de la experiencia de un usuario en la forma en la que éste interactúa con el sistema, se considera la facilidad de uso y el grado de satisfacción del usuario.

- Instalación: se determinan las operaciones de arranque y actualización del software.

### **Pruebas de Aceptación:**

Son las que hará el cliente, se determina que el sistema cumple con lo deseado y se obtiene la conformidad del cliente. Tienen como objetivo fundamental comprobar si el producto está listo para ser implantado para el uso operativo en el entorno del usuario. Entre sus características se encuentran:

- Participación del Usuario.
- Está enfocada hacia la prueba de los requisitos de usuario especificados.
- Está considerada como la fase final del proceso para crear una confianza en que el producto es el apropiado para su uso en explotación.

## **1.6. Métodos de Pruebas**

Existen diferentes métodos de pruebas, entre los más importantes se encuentran los de Caja Blanca y Caja Negra.

### **1.6.1. Prueba de Caja Blanca**

La prueba de Caja Blanca usa la estructura de control del diseño procedural para derivar los casos de prueba. Tienen como idea fundamental confeccionar casos de prueba que garanticen que se verifican todos los caminos independientes.

Verificaciones para cada camino independiente:

- Probar sus dos facetas desde el punto de vista lógico, es decir, verdadera y falsa.
- Ejecutar todos los bucles en sus límites operacionales.
- Ejercitar las estructuras internas de datos.

Los errores lógicos y las suposiciones incorrectas son inversamente proporcionales a la probabilidad de que se ejecute un camino del programa. A menudo creemos que un camino lógico tiene pocas posibilidades de ejecutarse cuando, de hecho, se puede ejecutar de forma regular. Los errores tipográficos son aleatorios. “Los errores se esconden en los rincones y se aglomeran en los límites” Beizer[13].

### 1.6.1.1. Técnica de Caja Blanca. Prueba del Camino Básico

La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes. Un camino independiente es aquel que introduce por lo menos una sentencia de procesamiento (o valor de condición) que no estaba considerada. Para obtener un conjunto de caminos independientes se construirá el Grafo de Flujo asociado y se calculará su Complejidad Ciclomática.

#### Grafo de Flujo:

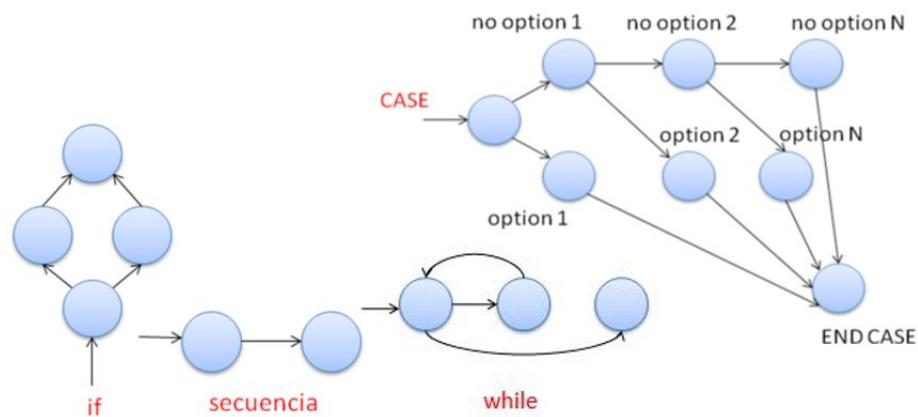


Figura 1.3: Grafos de flujo para las instrucciones If, Secuencia, While y Case.

#### Derivación de casos de prueba:

Pasos para realizar las pruebas:

1. A partir del diseño o del código fuente, dibujar el grafo de flujo asociado.
2. Se calcula la complejidad ciclomática del grafo.
3. Se determina un conjunto básico de caminos independientes.
4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

#### Complejidad Ciclomática:

La Complejidad Ciclomática de un grafo de flujo  $V(G)$  establece el número de caminos independientes.

Puede calcularse de tres formas alternativas:

- El número de regiones del grafo de flujo.
- $V(G) = A - N + 2$ , donde A es el número de aristas y N es el número de nodos.
- $V(G) = P + 1$ , donde P es el número de nodos predicado.

### Conjunto básico:

Es un conjunto de caminos independientes

#### 1.6.1.2. Técnica de Caja Blanca. Prueba de Bucles

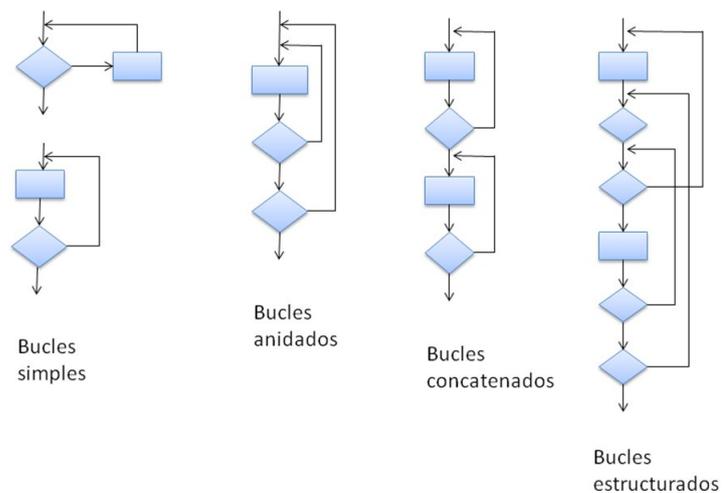


Figura 1.4: Representación de Bucles simples, anidados, concatenados y estructurados.

**Pruebas para Bucles simples** (n es el número máximo de iteraciones permitidos por el bucle)

- Pasar por alto totalmente el bucle.
- Pasar una sola vez por el bucle.
- Pasar dos veces por el bucle.
- Hacer m pasos por el bucle con  $m < n$ .
- Hacer  $n-1$ ,  $n$  y  $n + 1$  pasos por el bucle.

### **Pruebas para Bucles Anidados**

- Comenzar en el bucle más interior estableciendo los demás bucles en sus valores mínimos.
- Realizar las pruebas de bucle simple para el más interior manteniendo los demás en sus valores mínimos.
- Avanzar hacia afuera confeccionando pruebas para el siguiente bucle manteniendo todos los externos en los valores mínimos y los demás bucles anidados en sus valores típicos.
- Continuar el proceso hasta haber comprobado todos los bucles.

### **Pruebas para Bucles concatenados**

- Siempre que los bucles concatenados sean independientes se puede aplicar lo relativo a bucles simples/anidados. En caso de ser dependientes se evaluarán como bucles anidados.

### **Pruebas para Bucles no estructurados**

- Siempre que se usen los mecanismos que aporta la programación estructurada, este tipo de bucles no estarán presentes.

#### **1.6.2. Pruebas de Caja Negra**

Las pruebas de Caja Negra se centran en los requisitos funcionales del software, intentan encontrar errores de los siguientes tipos:

- Funciones incorrectas o inexistentes.
- Errores relativos a las interfaces.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores debidos al rendimiento.
- Errores de inicialización o terminación.

### 1.6.2.1. Técnica de Caja Negra. Partición Equivalente

La partición equivalente es un método que divide el campo de entrada de un programa en clases de datos. Una condición de entrada es un valor numérico específico, un rango de valores, un miembro de un conjunto de valores o lógica. Una clase de equivalencia representa un conjunto de estados válidos y no válidos para una condición de entrada. La prueba de partición equivalente se basa en evaluar las clases de equivalencia para una condición de entrada.

Identificar Clases de Equivalencia:

Se examina cada condición de entrada y se divide en dos o más grupos.

Se identifican dos tipos de clases:

- Clases de equivalencia válidas.
- Clases de equivalencia no válidas.

Si la condición de entrada es un:

- Rango, se define una clase de equivalencia válida y dos no válidas.
- Valor específico, se define una clase de equivalencia válida y dos no válidas.
- Miembro de conjunto, se define una clase de equivalencia válida y otra no válida.
- Lógica, se define una clase válida y otra no válida.
- Asignar un número único a cada clase de equivalencia.
- Escribir casos de prueba que cubran tantas clases válidas no incorporadas como sea posible hasta que se cubran todas las clases de equivalencia válidas.
- Escribir casos de prueba que cubran una sola clase no válida no incorporada hasta que se cubran todas las clases de equivalencia no válidas.

### 1.6.2.2. Técnica de Caja Negra. Análisis de Valores Límite

La técnica de análisis de valores límites selecciona casos de prueba que ejerciten los valores límite y complementa la prueba de partición equivalente. En lugar de realizar la prueba con cualquier elemento de la partición equivalente, se escogen los valores en los bordes de la clase. Se derivan tanto casos de prueba a partir de las condiciones de entrada como con las de salida.

## 1.7. Plan de Prueba

Un plan de pruebas está constituido por un conjunto de pruebas. Cada prueba debe dejar claro qué tipo de propiedades se quieren probar (corrección, robustez, fiabilidad, amigabilidad), dejar claro cómo se mide el resultado, especificar en qué consiste la prueba (hasta el último detalle de cómo se ejecuta) y definir cuál es el resultado que se espera.

Un caso de prueba consta de 3 bloques de información:

1. El propósito de la prueba.
2. Los pasos de ejecución de la prueba.
3. El resultado que se espera.

Respecto al orden de pruebas, una práctica frecuente es la siguiente:

1. Pasar pruebas de caja negra analizando valores límite. Recordando que hay que analizar condiciones límite de entrada y de salida.
2. Identificar clases de equivalencia de datos (entrada y salida) y añadir más pruebas de caja negra para contemplar valores normales (en las clases de equivalencia en que estos sean diferentes de los valores límite; es decir, en rangos amplios de valores).
3. Añadir pruebas basadas en "presunción de error". A partir de la experiencia y el sentido común, se aventuran situaciones que parecen proclives a padecer defectos, y se buscan errores en esos puntos. Son pruebas del tipo "¡Me lo temía!"
4. Medir la cobertura de caja blanca que se ha logrado con las fases previas y añadir más pruebas de caja blanca hasta lograr la cobertura deseada. Normalmente se busca una buena cobertura de ramas.

El propósito del plan de pruebas es explicitar el alcance, enfoque, recursos requeridos, calendario, responsables y la gestión de riesgos de un proceso de pruebas. Para la definición del plan de pruebas, deberemos valorar:

- El alcance de la aplicación.
- La complejidad de sus procesos.
- Plataformas en las que deberemos probar.
- Conocimientos y formación de quienes ejecutarán las pruebas.
- Normativas legales aplicables.

Es muy importante tener en cuenta que:

- Las pruebas deben estar presentes a lo largo de todo el ciclo de vida del desarrollo (la correspondencia entre las fases de desarrollo y tipos de prueba se conoce como el llamado “modelo en V”).
- El coste medio se encuentra entre un 30 % y un 50 % del coste total del proyecto.
- Siempre hay errores.
- Probar exhaustivamente el software es imposible.
- No es recomendable que el programador pruebe sus propios programas.
- Podemos disponer de herramientas.

Finalmente, no deberemos olvidar la importancia de actualización del plan de pruebas para reflejar los cambios que se produzcan en los requisitos y/o proceso de desarrollo de nuestro producto.

## 1.8. Herramientas para el Desarrollo de las Pruebas de Software

**TEST** es una herramienta de automatización de pruebas unitarias y de productos con código estándar, que trabaja sobre clases escritas en la plataforma Microsoft.NET, sin requerir que los desarrolladores realicen un solo caso de prueba o stub. Mejora la fiabilidad, funcionalidad, seguridad, desarrollo y mantenimiento de las aplicaciones .NET. Trabaja además con lenguajes de programación que utilizan el marco Microsoft .NET, incluyendo C# y VB.NET. Esta herramienta puede probar cualquier fichero o paquete que haya sido

construido utilizando el CLR de .NET. Facilita la creación y ejecución de pruebas definidas por el usuario basado en el framework de Nunit.

**JTEST 8.0** es una herramienta que brinda avances tecnológicos en la ejecución de pruebas, ayudando así a verificar de manera automática la funcionabilidad de aplicaciones cada vez más complejas, en empresas con sistemas en permanente cambio (J2EE, servicios de SOA/Web, permite reducir el tiempo de entrega del software así como una disminución del riesgo de generar software defectuoso o con problemas de vulnerabilidad. Jtest ofrece la generación y la ejecución automatizada de los casos de prueba a través de la simulación de un entorno real, posibilita una prueba en tiempo de ejecución que permite la temprana detección de defectos en el código que de otro modo pudieran pasar inadvertidos hasta etapas avanzadas de desarrollo como el aseguramiento de la calidad.

**JUnit** es una herramienta con la que se pueden desarrollar casos de pruebas fácilmente, heredando de sus propias clases base. Además ofrece una serie de interfaces gráficas para visualizar estas pruebas, ejecutarlas, ver sus resultados, seleccionar aquellas que queremos ejecutar, etc. A estas colecciones de clases, junto con sus herramientas se las conoce como "Marcos de pruebas", ya que gracias a ellas, se tiene toda la infraestructura necesaria para desarrollar pruebas unitarias de forma rápida, cómoda, extensible y fiable. JUnit ha tenido tanto éxito que se ha extendido a otros muchos lenguajes de programación, gracias al trabajo desinteresado de muchos programadores. Todos los frameworks heredados de Junit han recibido la denominación xUnit, con la que se indica que se trata de una migración, y se siguen las normas que marcó JUnit. Entre los frameworks xUnit, existen versiones para C/C++ (CUnit y CPPUnit), Delphi (DUnit), PHP (PHPUnit), HTML (HTMLUnit), NUnit (plataforma .NET), VUnit (Visual Basic), etc. El modo de trabajar de todos los frameworks xUnit es parecido entre ellos, aunque cada uno con las peculiaridades de su propio lenguaje.

Esta herramienta será utilizada para comprobar que las pruebas para el Visualizador y Editor Molecular, sean diseñadas correctamente.

## 1.9. Procedimiento de Prueba

Tiene como objetivo especificar los pasos para la ejecución de un conjunto de casos de prueba o más generalmente, los pasos utilizados para analizar un elemento software con el propósito de evaluar un conjunto de características del mismo.

Procedimientos y técnicas generales de pruebas:

1. Use siempre datos de entrada bien definidos para los que se conozcan los resultados correctos que deben obtenerse.
2. Detecte primero los defectos obvios (usando datos de prueba muy simples) y luego sí realice pruebas más complejas.
3. Cuando modifique algo mientras prueba realice un solo cambio cada vez y utilice los mismos datos con los que detectó el defecto.
4. Pruebe el programa para verificar si detecta entradas incorrectas.

## 1.10. Diseño de Casos de Prueba

Técnicas de Diseño de Casos de Prueba:

Utilizamos técnicas para conseguir una confianza aceptable en que se detectarán los defectos existentes:

- Equilibrio entre los recursos empleados y la confiabilidad de los casos de prueba.
- Elegir los casos de prueba que puedan representar a los de más.
- La elección no debe ser al azar.

Existen tres enfoques para el diseño de casos de prueba:

- El enfoque estructural o de caja blanca.
- El enfoque funcional o de caja negra.
- El enfoque aleatorio que consiste en utilizar modelos (en muchas ocasiones estadísticos) que representen las posibles entradas al programa para crear a partir de ellos los casos de prueba.

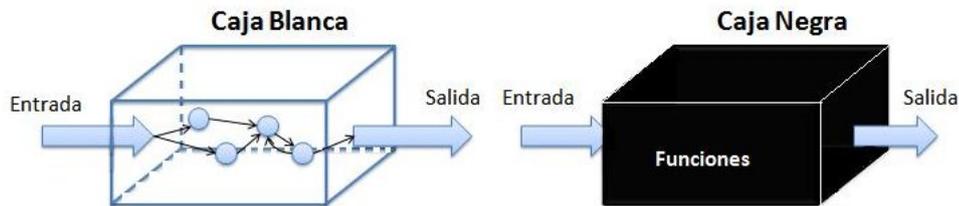


Figura 1.5: Técnicas de diseño de casos de Prueba.

### 1.11. Metodología OpenUP

OpenUP[14] es un proceso unificado que aplica acercamientos iterativos e incrementales dentro de un ciclo de vida estructurado. OpenUP abraza una filosofía pragmática, ágil enfocada en la naturaleza de colaboración del desarrollo del software. Es una herramienta que se puede ampliar a una gran variedad de tipos del proyecto.

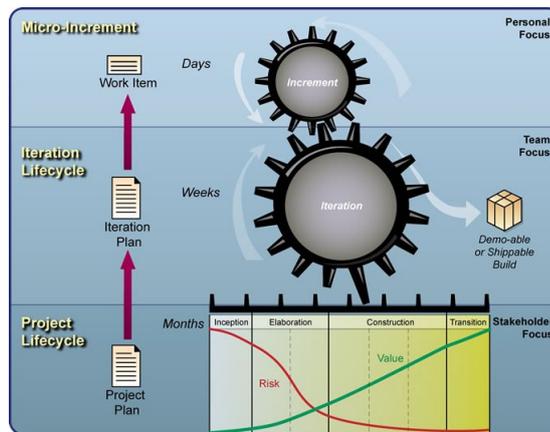


Figura 1.6: Capas de OpenUP: micro-incrementos, ciclo de vida de la iteración y ciclo de vida del proyecto.

El esfuerzo personal en un proyecto de OpenUP se organiza en micro-incrementos. Éstos representan unidades cortas de trabajo que producen un paso constante y medible del progreso del proyecto. El proceso aplica la colaboración intensiva como sistema de desarrollo incremental. Estos micro-incrementos proporcionan un lazo de regeneración extremadamente corto que conduzca a decisiones adaptables dentro de cada iteración. El ciclo de vida de un proyecto en OpenUP está estructurado en cuatro fases: Inicio, Elaboración, Construcción y Transición.

El ciclo de vida del proyecto establece los interesados y con los miembros del equipo de visibilidad y puntos de decisión a lo largo del proyecto. Esto permite una supervisión efectiva, y le permite tomar decisiones de

"ir o no ir a" en los momentos apropiados. Un plan de proyecto define el ciclo de vida y el resultado final es una aplicación liberada.

### 1.11.1. Rol de Probador



Figura 1.7: Representación de la relación entre los artefactos del rol de probador.

En esta metodología el probador es el responsable de las actividades base para el desarrollo de las pruebas. Esas actividades incluyen identificar, definir, ejecutar y realizar las pruebas necesarias, así como el registro de los resultados de las pruebas y el análisis de estos resultados.

El probador es el principal responsable de las siguientes tareas:

- Identificar las pruebas que deben realizarse.
- Identificar el enfoque más apropiado para la aplicación de la prueba.
- Aplicar las pruebas individuales.
- Establecer y ejecutar las pruebas.
- Registrar los resultados y verificar que los ensayos se han llevado a cabo.
- Analizar y recuperarse de los errores de ejecución.
- Comunicar los resultados de las pruebas para el equipo.

#### Artefacto: Caso de Prueba

Este artefacto es la especificación de un conjunto de prueba de los insumos, las condiciones de ejecución, y los resultados esperados, identificados con la finalidad de obtener una evaluación de algún aspecto particular de un escenario. En si un caso de prueba especifica las condiciones que deben ser validados para permitir la evaluación de algunos aspectos particulares del sistema bajo prueba. Es más formal que una prueba de concepto y por lo general toma la forma de una especificación. En ambientes menos formales, los casos de prueba se pueden crear mediante la identificación de un número de identificación exclusivo, nombre, datos de prueba asociados, y los resultados esperados. Los casos de prueba se pueden obtener de muchas fuentes, pero por lo general incluyen un subconjunto de los requisitos y otros tipos de atributos de calidad.

El propósito de este artefacto es:

- Disponer de un medio de prueba de captura de insumos, condiciones, y los resultados esperados para un sistema.
- Identificar sistemáticamente los aspectos de los programas informáticos para poner a prueba.
- Especificar si un resultado esperado se ha alcanzado sobre la base de un sistema de verificación de requisito, un conjunto de requisitos, o escenario.

### **Artefacto: Prueba de Log**

Este artefacto recoge la producción en bruto capturado durante una única ejecución de una o más pruebas para un solo ciclo de pruebas. Proporciona detalles, generalmente basados en el tiempo récord, que sirve a la vez como la comprobación de que un conjunto de pruebas ya fueron ejecutadas y proporciona información relacionada con el éxito de dichas pruebas. El objetivo es la creación de un registro preciso, que permite un diagnóstico posterior a la ejecución de los resultados de las pruebas. Estos datos se analizan posteriormente para ayudar a determinar los resultados de algunos aspectos de la prueba de esfuerzo.

El propósito de este artefacto es:

- Comprobar que un conjunto de pruebas ya ha sido ejecutado.
- Proporcionar información relacionada con el éxito de dichas pruebas.

### **Artefacto: Prueba de Secuencia de Comandos**

Este artefacto contiene paso a paso las instrucciones para realizar una prueba, lo que permite su ejecución. Estos pueden adoptar la forma de instrucciones documentadas que se ejecutan manualmente legibles por ordenadores o instrucciones que permiten la ejecución de pruebas automatizadas.

## **1.12. Conclusiones**

Después de haber realizado un estudio detallado de la documentación recopilada acerca de la calidad del software, podemos decir que para lograr un software de alta calidad y confiabilidad, necesitamos que el producto esté libre de errores, por lo que se ha decidido aplicar las siguientes pruebas por ser las más adecuadas para este software: pruebas de unidad, de integración, de validación y de sistema, aplicando así técnicas de Caja Negra y Caja Blanca y se decidió utilizar la Metodología Open UP.

## Capítulo 2

# DISEÑO Y APLICACIÓN DE PRUEBAS DE SOFTWARE

### 2.1. Introducción

En el presente capítulo se expone el diseño y la aplicación de las pruebas al Visualizador y al Editor Molecular del proyecto bioGRATO que se está desarrollando para el Centro de Química Farmacéutica, especificando cada uno de los elementos de las pruebas definidas. En el mismo se precisará un plan de pruebas como base para todo el proceso de pruebas que se llevará a cabo, así como la estrategia de pruebas a seguir y la configuración del entorno de pruebas.

### 2.2. Características a probar

Las características específicas a probar pueden variar, en dependencia del tipo de software en cuestión y de lo que se pretenda evaluar con la prueba que se va a realizar. Hay características que no se pueden olvidar, como por ejemplo si la aplicación cumple con las funcionalidades requeridas desde el comienzo, si no tiene errores a la hora de ejecutarlo, si es comprensible para el usuario, si se puede usar correctamente con facilidad y si cuenta con una documentación detallada y que corresponda con la aplicación. También se pueden medir aspectos generales que tienen que ver con el código del programa y permiten conocer si dicho código está bien estructurado y es reutilizable. Por ello es importante comprobar que la aplicación sea fácil de mantener para garantizar que exista por mucho tiempo.

Las aplicaciones que se están desarrollando serán sometidas a un proceso de pruebas en el que se verificarán fundamentalmente los atributos de fiabilidad, funcionamiento y usabilidad, para lograr con la comprobación de estos aspectos que las mismas funcionen de una manera eficaz.

### 2.3. Diseño del Plan de Prueba

Especificaciones de software:

Las aplicaciones desarrolladas son desktop y para que las mismas funcionen el usuario debe tener como requerimientos mínimos de software:

- Sistema Operativo Linux o Windows 95 o superior.
- Máquina Virtual de Java 1.6

Especificaciones de hardware:

- Pentium III o superior.
- 256 de RAM o superior.
- 40 GB de HD o superior.
- Tarjeta de red.

Es objetivo primordial verificar en el proceso de pruebas, como parte del plan de pruebas, que los requerimientos funcionales del sistema a desarrollar sean alcanzados, pues dichos requerimientos serán la base de los casos de uso del sistema y constituirán las principales funcionalidades con las que debe contar el sistema. Los requerimientos funcionales del Visualizador y Editor Molecular se encuentran en el documento especificación de requisitos del expediente del proyecto.

En el plan de pruebas, además de los aspectos a los que ya se ha hecho referencia, se definen otros que se consideran muy importantes como la estrategia de pruebas a seguir, la configuración del entorno de diseño de pruebas, los datos necesarios para hacer las pruebas, los procedimientos de pruebas y por último los casos de pruebas. A todos estos aspectos se hará referencia en los siguientes epígrafes de este capítulo, quedando así conformado todo el plan de pruebas.

## 2.4. Estrategia de Pruebas

Una estrategia de prueba integra las técnicas de diseño de casos de prueba, en una serie de pasos bien planificados, que llevan a una construcción correcta del software. Además una estrategia debe incluir pruebas de bajo nivel que verifiquen que todos los pequeños segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requisitos del cliente. Una estrategia debe proporcionar además una guía al profesional y un conjunto de hitos para el jefe de proyecto. Debido a que los pasos de la estrategia de prueba se dan a la vez cuando aumenta la presión de los plazos fijados, se debe poder medir el progreso y los problemas deben aparecer lo antes posible.

La estrategia de pruebas desarrollada para el Visualizador y Editor Molecular, tiene los siguientes objetivos:

- Identificar los tipos de prueba a utilizar.
- Definir las técnicas de pruebas que se van a utilizar a lo largo del proceso.
- Definir el entorno en que se van a desarrollar las pruebas.
- Definir los casos de pruebas que se deriven de la aplicación de las técnicas de pruebas.

## 2.5. Configuración del Entorno de Prueba

La configuración del entorno donde se vayan a ejecutar las diferentes pruebas que se realizan a un software es un aspecto muy importante dentro del proceso de pruebas, pues si no se analizan bien los recursos de software y hardware que necesita el producto que se está construyendo, a la hora de probar dicho producto se prescindirá de los elementos necesarios para la ejecución de un proceso de pruebas exitoso.

Por ello se tuvo en cuenta a la hora de llevar a cabo todo el proceso de pruebas al Visualizador y Editor Molecular, algunos requerimientos de hardware y de software que hicieron posible un mejor desarrollo de las pruebas, en aras de que las mismas lograran minimizar los errores en la aplicación en desarrollo. Algunos de los requerimientos que se consideraron necesarios para las pruebas son los referenciados a continuación:

Requerimientos de software:

- Una PC con Windows 95 o superior.

- Una PC con Linux (Ubuntu)
- Máquina Virtual de Java 1.6

Requerimientos de hardware:

- PC con Microprocesador Pentium IV a 3.0 GHz o superior
- Con 160 GB de Disco Duro.
- 512 MB de memoria RAM o superior

## 2.6. Cronograma de Prueba

Tabla 2.1: Cronograma de pruebas

Fecha de Inicio	Fecha de Fin	Actividades	Personal Implicado
4/02/08	4/02/08	Aceptación y firma del Plan de Prueba	Probadores Especialistas Funcionales
6/02/08	12/02/08	Verificación de las condiciones previas para el inicio de las pruebas Inicio de las pruebas	Probadores Especialistas Funcionales Analistas
<b>Primera Iteración al Editor</b>			
12/02/08	18/02/08	Ejecución de las pruebas unitarias a través del método de Caja Negra	Probadores
20/02/08	20/02/08	Análisis de las no conformidades y las solicitudes de cambio	Probadores Analista Desarrolladores Líder del Proyecto
<b>Primera Iteración al Visualizador</b>			
22/02/08	1/03/08	Ejecución de las pruebas unitaria a través del método de Caja Negra	Probadores

<b>Fecha de Inicio</b>	<b>Fecha de Fin</b>	<b>Actividades</b>	<b>Personal Implicado</b>
3/03/08	21/03/08	Ejecución de las pruebas unitaria a través del método de Caja Blanca	Probadores
25/03/08	29/03/08	Ejecución de las pruebas de Integración	Probadores
31/03/08	2/04/08	Ejecución de las pruebas de Carga	Probadores
4/04/08	4/04/08	Ejecución de las pruebas de Instalación	Probadores
8/04/08	8/04/08	Análisis de las no conformidades y las solicitudes de cambio	Probadores Analista Desarrolladores Líder del Proyecto
<b>Segunda Iteración al Editor</b>			
8/04/08	8/04/08	Entrega por el equipo de desarrollo de las respuestas a las no conformidades	Probadores Desarrolladores Líder del Proyecto
10/04/08	12/04/08	Ejecución de las pruebas unitarias a través del método de Caja Negra	Probadores
12/04/08	12/04/08	Análisis de las no conformidades y las solicitudes de cambio	Probadores Analista Desarrolladores Líder del Proyecto
<b>Segunda Iteración al Visualizador</b>			
5/05/08	5/05/08	Entrega por el equipo de desarrollo de las respuestas a las no conformidades	Probadores Desarrolladores Líder del Proyecto
7/05/08	15/05/08	Ejecución de las pruebas unitarias a través del método de Caja Negra	Probadores
17/05/08	23/05/08	Ejecución de las pruebas unitarias a través del método de Caja Blanca	Probadores
27/05/08	27/05/08	Ejecución de las pruebas de Integración	Probadores
29/05/08	31/05/08	Ejecución de las pruebas de Carga	Probadores

Fecha de Inicio	Fecha de Fin	Actividades	Personal Implicado
2/06/08	2/06/08	Ejecución de las pruebas de Instalación	Probadores
4/06/08	4/06/08	Análisis de las no conformidades y las solicitudes de cambio	Probadores Analista Desarrolladores Líder del Proyecto

## 2.7. Procedimientos de Pruebas

### 2.7.1. Procedimientos de pruebas para el Visualizador

Para el Visualizador que se está implementando se han definido varios procedimientos de pruebas en dependencia de los diferentes casos de uso que conforman la aplicación, estos procedimientos serán la base para el diseño de casos de pruebas y guiarán la confección de los mismos, puesto que describen cómo se puede ejecutar un caso de prueba de manera general. Es importante señalar que en este documento se hará referencia a los casos de usos más importantes, el resto de los mismos se encuentran en el expediente del proyecto.

#### 2.7.1.1. Caso de Uso: Abrir Fichero

Tabla 2.2: Procedimiento “Abrir Fichero”

No	Prueba	Acciones a realizar	Resultados esperados
1	Abriendo Fichero	Seleccione en el menú Archivo la opción Abrir. Aparecerá una ventana de diálogo donde se selecciona la extensión y el fichero molecular a cargar. Luego se presiona el botón Abrir.	Se verifica si el fichero ya está abierto. Adiciona el mismo a la lista de ficheros abiertos y lo visualiza en el área de visualización.
2	Abriendo Fichero	Seleccione en el menú Archivo la opción Abrir. Aparecerá una ventana de diálogo donde se selecciona la extensión y los ficheros moleculares a cargar. Luego se presiona el botón Abrir.	Se verifica si los ficheros ya están abiertos. Adiciona los ficheros a la lista de ficheros abiertos.

No	Prueba	Acciones a realizar	Resultados esperados
3	Abriendo Fichero de Preferencia de Visualización	Seleccione en el menú Archivo la opción Abrir Preferencias de Visualización. Aparecerá una ventana de diálogo donde se seleccionará el fichero de preferencia a cargar. Luego se presiona el botón Abrir.	Cambia la visualización de la molécula de acuerdo a las cargadas en el fichero de preferencias.
4	Abriendo Fichero Reciente	Seleccione en el menú Archivo la opción Abrir Reciente. Se desplegará una lista de los ficheros que fueron abiertos con anterioridad donde se seleccionará el fichero que se desea volver a cargar.	Se verifica si el fichero ya está abierto. Adiciona el mismo a la lista de ficheros abiertos y lo visualiza en el área de visualización.

### 2.7.1.2. Caso de Uso: Gestionar Fichero

Tabla 2.3: Procedimiento “Gestionar Fichero”

No	Prueba	Acciones a realizar	Resultados esperados
1	Cerrando fichero	Seleccione en el menú Archivo la opción Cerrar.	Se cierra el fichero que se esté visualizando sin quitarlo de la lista de ficheros abiertos.
2	Cerrando todos los ficheros	Seleccione en el menú Archivo la opción Cerrar Todo.	Se cierran todos los ficheros visualizados sin quitarlos de la lista de ficheros abiertos.
3	Eliminando fichero	Seleccione en el menú Archivo la opción Eliminar.	Se cierra el fichero que se esté visualizando y lo elimina de la lista de ficheros abiertos.
4	Eliminando todos los ficheros	Seleccione en el menú Archivo la opción Eliminar Todo.	Se cierran todos los ficheros visualizados y los elimina de la lista de ficheros abiertos

No	Prueba	Acciones a realizar	Resultados esperados
5	Guardando fichero	Seleccione en el menú Archivo la opción Guardar Como. Aparecerá una ventana de diálogo donde se selecciona el lugar donde se quiere guardar el fichero y la extensión con que será guardado. Luego se presiona el botón Guardar.	Se verifica si la extensión es distinta de mol, utiliza el servicio de conversión de formato para convertir a la extensión especificada y se guarda el fichero en el destino indicado.
6	Guardando fichero de Preferencia de visualización	Seleccione en el menú Archivo la opción Guardar Preferencias de Visualización. Aparecerá una ventana de diálogo que permite seleccionar el lugar y escribir el nombre con que se quiere guardar la Preferencia de Visualización. Luego se presiona el botón Guardar.	Se guarda el fichero de Preferencia en el lugar indicado con el nombre especificado.
7	Exportando a PDF	Seleccione en el menú Archivo la opción Exportar a PDF. Aparecerá una ventana de diálogo que permite seleccionar el lugar donde se quiere exportar el fichero. Luego se presiona el botón Exportar.	Se guarda el fichero en el lugar indicado con el nombre y extensión especificada.
8	Exportando a POVRAY	Seleccione en el menú Archivo la opción Exportar a POVRAY. Aparecerá una ventana de diálogo que permite seleccionar el lugar donde se quiere guardar el fichero y el nombre del mismo y se completan los datos solicitados. Luego se presiona el botón Ejecutar.	Se exporta el fichero en el lugar indicado con el nombre y extensión especificada.

### 2.7.1.3. Caso de Uso: Gestionar Visualización

Tabla 2.4: Procedimiento “Gestionar Visualización”

No	Prueba	Acciones a realizar	Resultados esperados
----	--------	---------------------	----------------------

No	Prueba	Acciones a realizar	Resultados esperados
1	Visualizando moléculas como Bolas y Cilindros	Se selecciona en el menú Visualizar la opción Bolas y Cilindros.	Se visualiza la molécula como Bolas y Cilindros.
2	Visualizando moléculas como Alambres	Se selecciona en el menú Visualizar la opción Alambres.	Se visualiza la molécula como Alambres.
3	Visualizando moléculas como Radios de Van Der Waals	Se selecciona en el menú Visualizar la opción Radios de Van Der Waals.	Se visualiza la molécula como Radios de Van Der Waals.
4	Creando Superficies De Puntos	Se selecciona la opción De Puntos en la opción Superficies del menú Visualizar.	Se verifica que este no sea el tipo de superficie que se está visualizando en ese momento y visualiza la molécula con superficie De Puntos.
5	Creando Superficies de Van Der Waals	Se selecciona la opción Van Der Waals en la opción Superficies del menú Visualizar.	Se verifica que este no sea el tipo de superficie que se está visualizando en ese momento y visualiza la molécula con superficie Van Der Waals.
6	Creando Superficie Molecular	Se selecciona la opción Molecular en la opción Superficies del menú Visualizar.	El sistema verifica que este no sea el tipo de superficie que se está visualizando en ese momento y visualiza la molécula con superficie Molecular.
7	Creando Superficie Disolvente	Se selecciona la opción Disolvente en la opción Superficies del menú Visualizar.	Se verifica que este no sea el tipo de superficie que se está visualizando en ese momento y visualiza la molécula con superficie Disolvente.

No	Prueba	Acciones a realizar	Resultados esperados
8	Creando Superficie Accesible al Disolvente	Se selecciona la opción Accesible al Disolvente en la opción Superficies del menú Visualizar.	Se verifica que este no sea el tipo de superficie que se está visualizando en ese momento y visualiza la molécula con superficie Accesible al Disolvente.
9	Creando Superficies de Capa	Se selecciona la opción Carga en la opción Superficies del menú Visualizar.	Se verifica que este no sea el tipo de superficie que se está visualizando en ese momento y visualiza la molécula con superficie de Carga.
10	Creando Superficies Opacas	Se selecciona la opción Superficies Opacas en la opción Superficies del menú Visualizar.	Se verifica que este no sea el tipo de superficie que se está visualizando en ese momento y visualiza la molécula con Superficies Opacas.
11	Creando Superficies Traslucidas	Se selecciona la opción Superficies Traslucidas en la opción Superficies del menú Visualizar.	Se verifica que este no sea el tipo de superficie que se está visualizando en ese momento y visualiza la molécula con Superficies Traslucidas.
12	Ocultando Superficie	Se selecciona la opción Ocultar en la opción Superficies del menú Visualizar.	Se oculta la superficie que se esté visualizando en ese momento.
13	Creando Vista Frontal	Se selecciona la opción Frontal en la opción Vistas del menú Visualizar.	Se verifica que esta no sea la vista que se está visualizando en ese momento y visualiza la molécula desde la vista Frontal.
14	Generando Vista Superior	Se selecciona la opción Superior en la opción Vistas del menú Visualizar.	Se verifica que esta no sea la vista que se está visualizando en ese momento y visualiza la molécula desde la vista Superior
15	Generando Vista Izquierda	Se selecciona la opción Izquierda en la opción Vistas del menú Visualizar	Se verifica que esta no sea la vista que se está visualizando en ese momento y visualiza la molécula desde la vista Izquierda.

No	Prueba	Acciones a realizar	Resultados esperados
16	Generando Vista Derecha	Se selecciona la opción Derecha en la opción Vistas del menú Visualizar.	Se verifica que esta no sea la vista que se está visualizando en ese momento y visualiza la molécula desde la vista Derecha.
17	Generando Vista Estereográfica Gafas Rojo + Cian	Se selecciona la opción Estereográficas y dentro de esta se escoge la opción Gafas Rojo + Cian en la opción Vistas del menú Visualizar.	Se verifica que esta no sea la vista que se está visualizando en ese momento y visualiza la molécula con vista Estereográfica Rojo + Cian.
18	Generando Vista Estereográfica Gafas Rojo + Azul	Se selecciona la opción Estereográficas y dentro de esta se escoge la opción Gafas Rojo + Azul en la opción Vistas del menú Visualizar.	Se verifica que esta no sea la vista que se está visualizando en ese momento y visualiza la molécula con vista Estereográfica Rojo + Azul.
19	Generando Vista Estereográfica Gafas Rojo + Verde	Se selecciona la opción Estereográficas y dentro de esta se escoge la opción Gafas Rojo + Verde en la opción Vistas del menú Visualizar.	Se verifica que esta no sea la vista que se está visualizando en ese momento y visualiza la molécula con vista Estereográfica Rojo + Verde.
20	Generando Vista Estereográfica Visión Vizca(Cross-Eyed)	Se selecciona la opción Estereográficas y dentro de esta se escoge la opción Visión Vizca(Cross-Eyed) en la opción Vistas del menú Visualizar.	Se verifica que esta no sea la vista que se está visualizando en ese momento y visualiza la molécula con vista Estereográfica Visión Vizca(Cross-Eyed)
21	Generando Vista Estereográfica Visión Paralela (Wall-Eyed)	Se selecciona la opción Estereográficas y dentro de esta se escoge la opción Visión Paralela (Wall-Eyed) en la opción Vistas del menú Visualizar.	Se verifica que esta no sea la vista que se está visualizando en ese momento y visualiza la molécula con vista Estereográfica Visión Paralela (Wall-Eyed).
22	Restaurando Vista Estereográfica	Se selecciona la opción Estereográficas y dentro de esta se escoge la opción Restaurar en la opción Vistas del menú Visualizar.	Se restaura la vista Estereográfica que se esté visualizando en ese momento.
23	Creando Representaciones de Backbone	Se selecciona la opción Backbone en la opción Representaciones del menú Visualizar.	Se verifica que esta no sea la representación que se está visualizando en ese momento y luego la crea.

No	Prueba	Acciones a realizar	Resultados esperados
24	Creando Representaciones Cartoon	Se selecciona la opción Cartoon en la opción Representaciones del menú Visualizar.	Se verifica que esta no sea la representación que se está visualizando en ese momento y luego la crea.
25	Creando Representaciones Cohetes Cartoon	Se selecciona la opción Cohetes Cartoon en la opción Representaciones del menú Visualizar.	Se verifica que esta no sea la representación que se está visualizando en ese momento y luego la crea.
26	Creando Representaciones de Cintas	Se selecciona la opción Cintas en la opción Representaciones del menú Visualizar.	Se verifica que esta no sea la representación que se está visualizando en ese momento y luego la crea.
27	Creando Representaciones Cohetes	Se selecciona la opción Cohetes en la opción Representaciones del menú Visualizar.	Se verifica que esta no sea la representación que se está visualizando en ese momento y luego la crea.
28	Creando Representaciones de Hilos	Se selecciona la opción Hilos en la opción Representaciones del menú Visualizar.	Se verifica que esta no sea la representación que se está visualizando en ese momento y luego la crea.
29	Creando Representaciones de Trazas	Se selecciona la opción Trazas en la opción Representaciones del menú Visualizar.	Se verifica que esta no sea la representación que se está visualizando en ese momento y luego la crea.
30	Ninguna Representación	Se selecciona la opción Ninguna en la opción Representaciones del menú Visualizar.	Se quita la representación que esté visualizada.
31	Cambiando Visualización de Átomo a 15% de Van Der Waals	Se selecciona la opción 15% de Van Der Waals en la opción Átomo del menú Visualizar.	Se verifica que esta opción no se esté visualizando en ese momento y luego cambia la visualización de los átomos.
32	Cambiando Visualización de Átomo a 20% de Van Der Waals	Se selecciona la opción 20% de Van Der Waals en la opción Átomo del menú Visualizar.	Se verifica que esta opción no se esté visualizando en ese momento y luego cambia la visualización de los átomos.

No	Prueba	Acciones a realizar	Resultados esperados
33	Cambiando Visualización de Átomo a 25 % de Van Der Waals	Se selecciona la opción 25 % de Van Der Waals en la opción Átomo del menú Visualizar.	Se verifica que esta opción no se esté visualizando en ese momento y luego cambia la visualización de los átomos.
34	Ocultando Visualización de Átomos	Se selecciona la opción Ocultar en la opción Átomo del menú Visualizar.	Se oculta el tipo de visualización que se esté visualizando.
35	Cambiando Visualización de Enlace a Modelo de Alambre	Se selecciona la opción Modelo de Alambres en la opción Enlace del menú Visualizar.	Se verifica que esta opción no se esté visualizando en ese momento y luego cambia la visualización del enlace.
36	Cambiando Visualización de Enlace de Radios a 0.10 Ansgtroms	Se selecciona la opción Radios y dentro de esta la opción 0.10 Ansgtroms en la opción Enlace del menú Visualizar.	Se verifica que esta opción no se esté visualizando en ese momento y luego cambia la visualización del enlace.
37	Cambiando Visualización de Enlace de Radios a 0.15 Ansgtroms	Se selecciona la opción Radios y dentro de esta la opción 0.15 Ansgtroms en la opción Enlace del menú Visualizar.	Se verifica que esta opción no se esté visualizando en ese momento y luego cambia la visualización del enlace.
38	Cambiando Visualización de Enlace de Radios a 0.20 Ansgtroms	Se selecciona la opción Radios y dentro de esta la opción 0.20 Ansgtroms en la opción Enlace del menú Visualizar.	Se verifica que esta opción no se esté visualizando en ese momento y luego cambia la visualización del enlace.
39	Ocultando Visualización de Enlace	Se selecciona la opción Ocultar en la opción Enlace del menú Visualizar.	Se oculta el tipo de visualización que se esté visualizando.
40	Cambiando Visualización de Etiqueta a Nombre	Se selecciona la opción Nombre en la opción Etiqueta del menú Visualizar.	Se verifica que esta opción no se esté visualizando en ese momento y luego visualiza los nombres de las etiquetas.

No	Prueba	Acciones a realizar	Resultados esperados
41	Cambiando Visualización de Etiqueta a Símbolos	Se selecciona la opción Símbolos en la opción Etiqueta del menú Visualizar.	Se verifica que esta opción no se esté visualizando en ese momento y luego visualiza los símbolos.
42	Cambiando Visualización de Etiqueta a Números	Se selecciona la opción Números en la opción Etiqueta del menú Visualizar.	Se verifica que esta opción no se esté visualizando en ese momento y luego visualiza los números.
43	Ocultando Visualización Etiquetas	Se selecciona la opción Ocultar en la opción Etiqueta del menú Visualizar.	Se oculta el tipo de visualización que se esté visualizando.

#### 2.7.1.4. Caso de Uso: Gestionar Módulos

Tabla 2.5: Procedimiento “Gestionar Módulos”

No	Prueba	Acciones a realizar	Resultados esperados
1	Adicionando Módulos	Se selecciona en el menú Preferencias la opción Adicionar Módulos. Aparecerá una ventana de diálogo donde se puede seleccionar la carpeta donde se encuentran los Módulos que se quieren adicionar. Se verifica que los Módulos presentes en la carpeta seleccionada son del tipo que utiliza la plataforma y además comprueba que no estén incluidos ya en la plataforma y se muestran en la ventana de diálogo. Luego se selecciona el Módulo que se desea adicionar y se presiona el botón Importar.	Se verifica que este módulo no esté ya cargado y que sea del tipo que contiene la plataforma y lo adiciona en la opción Módulos del menú Preferencias.

No	Prueba	Acciones a realizar	Resultados esperados
2	Eliminando Módulos	Se selecciona en el menú Preferencias la opción Eliminar Módulos. Aparecerá una ventana de diálogo donde se selecciona la carpeta que contiene los ficheros con los Módulos que se pueden eliminar, se selecciona el que se desea eliminar y se presiona el botón Eliminar.	Se elimina el fichero que contiene el Módulo de la carpeta ext y además elimina el nombre del Módulo de la opción Módulos del menú Preferencias.

### 2.7.1.5. Caso de Uso: Calcular Propiedades

Tabla 2.6: Procedimiento “Calcular Propiedades”

No	Prueba	Acciones a realizar	Resultados esperados
1	Calculando propiedades	Se selecciona en el menú Preferencias la opción Calcular Propiedades.	Se muestra en el área inferior de la interfaz datos de los átomos y los enlaces, la masa molecular y el por ciento de descomposición.
2	Convirtiendo a 3D	Se selecciona en el menú Preferencias la opción Convertir a 3D.	La molécula es enviada al servicio de conversión en 3D, se recibe convertida del servicio y se visualiza en el área de visualización.

### 2.7.2. Procedimientos de pruebas para el Editor

#### 2.7.2.1. Caso de Uso: Crear Molécula

Tabla 2.7: Procedimiento “Crear Molécula”

No	Prueba	Acciones a realizar	Resultados esperados
1	Creando molécula	Se selecciona la opción Nuevo en la barra de herramientas	Se crean la molécula según las opciones escogidas.

#### 2.7.2.2. Caso de Uso: Modificar Molécula

Tabla 2.8: Procedimiento “Modificar Molécula”

No	Prueba	Acciones a realizar	Resultados esperados
1	Modificando molécula	Se selecciona la opción Abrir en la barra de herramientas y se muestra un explorador donde se localizan los ficheros que se desean abrir.	Se edita la molécula según las opciones escogidas.

## 2.8. Casos de Pruebas

Un caso de prueba no es más que un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados que se diseñan con el propósito de cumplir un objetivo en particular o una función esperada. Los casos de pruebas deben verificar si el producto que se está realizando satisface los requerimientos del usuario final tal y como se describe en la especificación de los requerimientos y los casos de uso. Además, un caso de prueba debe garantizar que el producto se comporta como se describe en las especificaciones funcionales del diseño.

En fin, el diseño de un buen caso de prueba será el eslabón fundamental del proceso de prueba, pues mediante los mismos se encontrarán la mayor cantidad de defectos posibles y de esta forma se garantizará que el software en construcción, cuando llegue a su fase final, esté libre de errores.

Los casos de prueba diseñados para el Visualizador y Editor Molecular, en una primera iteración, se detallan a continuación.

### 2.8.1. Primera Iteración

#### 2.8.1.1. Pruebas de Caja Negra para el Visualizador

En las pruebas de caja negra, diseñadas para el Visualizador, se utilizó la técnica de partición equivalente siendo esta la más efectiva para trabajar con la interfaz del software y desmostrando así la efectividad en la funcionalidad del mismo. El diseño de estas pruebas se realizó a partir de la descripción de los casos de uso, archivadas en el expediente del proyecto.

En el documento se presenta de forma completa los casos de pruebas diseñados para el caso de uso Abrir Ficheros, en los restantes casos de uso presentaremos los casos de pruebas que arrojaron resultados insatisfactorios. Aunque hay que aclarar que todos aparecen de forma íntegra en el expediente del proyecto.

## Casos de prueba del caso de uso “Abrir Fichero”

### 1. Descripción General:

Este caso de uso tiene como propósito cargar ficheros para su posterior uso en la aplicación.

### 2. Al Caso de Uso Abrir Ficheros se le realizaron las siguientes pruebas:

Pruebas de funcionalidad mediante la técnica de Partición Equivalente.

### 3. CPR 1: Abrir

#### 3.1 Descripción de la Funcionalidad:

Mediante esta funcionalidad el sistema da la posibilidad de abrir una o varios ficheros moleculares.

#### 3.2 Flujo Central:

1. El especialista selecciona la opción Abrir del menú “Archivo”.

1.1 El sistema muestra una ventana de diálogo donde permite seleccionar la extensión y el o los ficheros, ya sea molécula o proteína, que desee abrir.

2. El especialista selecciona la extensión y la o las moléculas o proteínas que desea abrir y luego presiona el botón Abrir.

2.1 El sistema verifica si las extensiones son distintas de mol o pdb, de ser así le solicita el servicio de conversión de formato que convierta el o los ficheros a mol.

2.2 Copia el o los mismos en la carpeta tmp.

2.3 Si existen ficheros ya cargados y son varios ficheros a cargar, el sistema muestra una ventana de diálogo con la información siguiente: “Desea Adicionar o Reemplazar las Existentes”.

3. El especialista selecciona Adicionar.

3.1 El sistema adiciona los ficheros al final del árbol de ficheros cargados.

3.2 Adiciona los 10 últimos ficheros en la lista de ficheros recientes.

### 3.3 Flujo Alternativo de los eventos:

2.1 Si el servicio de conversión no está disponible muestra el siguiente error “No está ejecutándose el servicio de conversión”.

2.3 El sistema adiciona el fichero al final el árbol de ficheros cargados.

2.4 Adiciona el fichero en la lista de ficheros recientes.

2.5 Visualiza el fichero en el área de visualización.

3. El especialista selecciona Remplazar.

3.1 El sistema elimina los ficheros existentes en el árbol de ficheros cargados y adiciona los nuevos ficheros en el árbol de ficheros cargados.

### 3.4 Condiciones de Ejecución:

Existencia de los ficheros.

Tabla 2.9: Casos de prueba para la funcionalidad “Abrir”

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
Se selecciona en la menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 6.mol		El sistema verifica que la molécula no está cargada, adiciona la molécula a la lista de ficheros cargados y activa las opciones para molécula de los menú Seleccionar y Visualizar y visualiza la molécula en el área de visualización	Satisfactorio	
	Se selecciona en la menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 6.mol	El sistema verifica que la molécula está cargada y muestra el siguiente mensaje "Fichero ya visualizado"	Insatisfactorio	Al presionar el botón Abrir se verifica que no se muestra dicho mensaje de error
Se selecciona en la menú Archivo la opción Abrir y en la ventana de diálogo se seleccionan las moléculas siguientes: 143.mol 250.mol 1330.mol 1483.mol		El sistema verifica que las moléculas no están cargadas y si hay alguna otra molécula cargada muestra el siguiente mensaje: "Desea Adicionar o Reemplazar las existentes?", luego adiciona o reemplaza según la opción ejecutada por el usuario y adiciona las moléculas a la lista de ficheros cargados	Satisfactorio	

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	Escribir en la ventana de diálogo Abrir el nombre de la molécula: molécula.mol	El sistema muestra un mensaje de error "El fichero seleccionado no existe"	Insatisfactorio	El sistema no muestra el mensaje de error

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la proteína siguiente: 114D.pdb		El sistema verifica que la proteína no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para proteína de los menú Seleccionar y Visualizar y visualiza la proteína en el área de visualización	Satisfactorio	
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la proteína siguiente: 114D.pdb	El sistema verifica que la proteína ya está cargada y muestra el siguiente mensaje "Fichero ya visualizado"	Insatisfactorio	Al presionar el botón Abrir se verifica que no se muestra dicho mensaje de error
Se seleccionan en la ventana de diálogo Abrir las proteínas siguientes: 1CRN.pdb 1D66.pdb 1D68.pdb 1EBL.pdb		El sistema verifica que las proteínas no están cargadas y si hay alguna otra cargada muestra el siguiente mensaje: "Desea Adicionar o Reemplazar las existentes?", luego adiciona o reemplaza según la opción ejecutada por el usuario y adiciona las proteínas a la lista de ficheros cargados	Satisfactorio	

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 3177.xyz		El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y visualiza la molécula en el área de visualización	Satisfactorio	
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 1330. unixyz	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados y activa las opciones para proteína de los menú Seleccionar y Visualizar y visualiza la proteína en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 250.res	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 1483.qcout	El sistema verifica que la molécula no está cargada, la adiciona a la lista de moléculas cargadas, activa las opciones para molécula de los menú Seleccionar y Visualizar y visualiza la molécula en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 1701.ent	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargadas, activa las opciones para molécula de los menú Seleccionar y Visualizar y visualiza la molécula en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 890.nwo	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 593. mopout	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 1649. mopcrt	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 6.mdl	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 2821.sd		El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Satisfactorio	
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 1483.sdf	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 1821.jout	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 890.hin		El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Satisfactorio	
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 1330. qm1gp	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 250. mm1gp	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 6.gpr	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 3177. gamout	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 929.gam	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 1913.scf	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 593.cml	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 299.cif	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	Se selecciona en el menú Archivo la opción Abrir y en la ventana de diálogo se selecciona la molécula siguiente: 1913. mol2	El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización	Insatisfactorio	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza

## CPR 2: Abrir Reciente

Tabla 2.10: Caso de prueba para la funcionalidad “Abrir Reciente”

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
Se selecciona en el menú Archivo la opción Abrir Reciente y en la lista de ficheros abiertos recientemente la molécula siguiente: 1330.mol		El sistema verifica que la molécula no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización.	Satisfactorio	

Se selecciona en el menú Archivo la opción Abrir reciente y en la lista de ficheros abiertos recientemente la molécula siguiente: 250.mol		El sistema verifica que la proteína no está cargada, la adiciona a la lista de ficheros cargados, activa las opciones para molécula de los menú Seleccionar y Visualizar y la visualiza en el área de visualización.	Satisfactorio	
	Se selecciona en el menú Archivo la opción Abrir Reciente.	El sistema verifica que no se ha abierto ningún fichero recientemente y no muestra activa la opción Abrir Reciente del menú Archivo.	Insatisfactorio	Al ir a la opción Abrir Reciente esta se encuentra activa cuando no se ha abierto ningún fichero recientemente.

### CPR 3: Abrir Preferencias de Visualización

Tabla 2.11: Caso de prueba para la funcionalidad “Abrir Preferencias de Visualización”

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
Se selecciona en la ventana de diálogo Abrir Preferencias de Visualización el fichero de Preferencia de Visualización: preferencial		El sistema verifica que es un fichero de Preferencia de Visualización y además no esté cargado, luego lo carga y aplica las Preferencias de Visualización en el área de visualización	Satisfactorio	
	Se selecciona en la ventana de diálogo Abrir Preferencias de Visualización el fichero de Preferencia de Visualización: preferencia 2.doc	El sistema verifica que no es un fichero de Preferencia de Visualización y muestra un mensaje de error: "Error en tipo de fichero"	Insatisfactorio	El sistema no muestra el mensaje de error
	Se escribe en la ventana de diálogo Abrir Preferencias de Visualización el nombre del fichero: ejemplo	El sistema verifica que no existe este fichero y muestra un mensaje de error: "El fichero no existe"	Insatisfactorio	El sistema no muestra el mensaje de error

### Casos de prueba del caso de uso "Gestionar Fichero"

#### CPR 2: Eliminar Fichero

Tabla 2.13: Casos de prueba para la funcionalidad "Eliminar Fichero"

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	Se selecciona en el menú Archivo la opción: Eliminar Todo	El sistema elimina todos los ficheros visualizados eliminándolos de la lista de ficheros abiertos	Insatisfactorio	Si no está marcado un fichero molecular el sistema no elimina todos los ficheros

### CPR 3: Guardar Fichero

Tabla 2.14: Caso de prueba para la funcionalidad “Guardar Fichero”

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	En la ventana de diálogo Guardar Como se escoge el lugar donde se desea guardar el fichero molecular y el nombre con el que desea guardarlo	El sistema verifica que no haya un fichero guardado con el mismo nombre en el lugar donde se especifique y guarda el fichero en este lugar especificado	Insatisfactorio	No implementado

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	En la ventana de diálogo Guardar Preferencias de Visualización se especifica el lugar donde se desea guardar el fichero de Preferencia de Visualización y el nombre con el que se desea guardar	El sistema verifica que no haya un fichero de Preferencia de Visualización guardado con el mismo nombre en el lugar donde se especifique y da la posibilidad de escoger el tipo de fichero que se desea guardar	Insatisfactorio	El sistema no permite especificar el tipo de fichero que se desea guardar

### Casos de prueba del caso de uso “Gestionar Visualización”

#### CPR 4: Crear Representaciones

Tabla 2.21: Caso de prueba para la funcionalidad “Crear Representaciones”

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	En la opción Representaciones del menú Visualizar se selecciona la opción: Cohetes	El sistema verifica que esta no sea la representación que se está visualizando en ese momento y luego la crea	Insatisfactorio	El sistema no visualiza nada

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	En la opción Representaciones del menú Visualizar se selecciona la opción: Ninguna	El sistema quita la representación que esté visualizada	Insatisfactorio	El sistema no visualiza nada

### 2.8.2. Pruebas de Caja Negra para el Editor Molecular.

En las pruebas de caja negra, diseñadas para el Editor, se utilizó la técnica de partición equivalente siendo esta la más efectiva para trabajar con la interfaz del software y demostrando así la efectividad en la funcionalidad del mismo. El diseño de estas pruebas se realizó a partir de la descripción de los casos de uso, archivadas en el expediente del proyecto.

#### Casos de prueba del caso de uso “Crear Molécula”

#### Casos de prueba del caso de uso “Modificar Molécula”

##### CPR 1: Modificar Molécula

Tabla 2.26: Caso de prueba para la funcionalidad “Modificar Molécula”

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	Se selecciona en la barra de herramientas la opción Abrir	El sistema abre una ventana de diálogo donde el especialista podrá seleccionar la molécula que desea modificar y luego visualiza el fichero seleccionado en el área de trabajo en 2D	Insatisfactorio	El Editor no visualiza bien la molécula en 2D
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Tres		El sistema verifica si tiene indicado una arista en el área de trabajo y tomando como base la arista seleccionada dibuja el ciclo de tres	Satisfactorio	
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Tres		El sistema verifica si tiene indicado un vértice en el área de trabajo, dibuja un enlace y seguido al enlace el ciclo de tres	Satisfactorio	
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Cuatro		El sistema verifica si tiene indicado una arista en el área de trabajo y tomando como base la arista seleccionada dibuja el ciclo de cuatro	Satisfactorio	
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Cuatro		El sistema verifica si tiene indicado un vértice en el área de trabajo, dibuja un enlace y seguido al enlace el ciclo de cuatro	Satisfactorio	

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Cinco		El sistema verifica si tiene indicado una arista en el área de trabajo y tomando como base la arista seleccionada dibuja el ciclo de cinco	Satisfactorio	
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Cinco		El sistema verifica si tiene indicado un vértice en el área de trabajo, dibuja un enlace y seguido al enlace el ciclo de cinco	Satisfactorio	
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Seis		El sistema verifica si tiene indicado una arista en el área de trabajo y tomando como base la arista seleccionada dibuja el ciclo de seis	Satisfactorio	
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Seis		El sistema verifica si tiene indicado un vértice en el área de trabajo, dibuja un enlace y seguido al enlace el ciclo de seis	Satisfactorio	
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Benceno		El sistema verifica si tiene indicado una arista en el área de trabajo y tomando como base la arista seleccionada dibuja el ciclo de benceno	Satisfactorio	

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Benceno		El sistema verifica si tiene indicado un vértice en el área de trabajo, dibuja un enlace y seguido al enlace el ciclo de benceno	Satisfactorio	
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Siete		El sistema verifica si tiene indicado una arista en el área de trabajo y tomando como base la arista seleccionada dibuja el ciclo de siete	Satisfactorio	
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Siete		El sistema verifica si tiene indicado un vértice en el área de trabajo, dibuja un enlace y seguido al enlace el ciclo de siete	Satisfactorio	
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Ocho		El sistema verifica si tiene indicado una arista en el área de trabajo y tomando como base la arista seleccionada dibuja el ciclo de ocho	Satisfactorio	
El especialista selecciona, en la opción Plantilla de la barra de herramientas, la opción Ocho		El sistema verifica si tiene indicado un vértice en el área de trabajo, dibuja un enlace y seguido al enlace el ciclo de ocho	Satisfactorio	

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El especialista selecciona, en la opción Enlace de la barra de herramientas, la opción Simple		El sistema verifica si tiene indicado una arista en el área de trabajo y añade un enlace simple como común a la arista indicada	Satisfactorio	
El especialista selecciona, en la opción Enlace de la barra de herramientas, la opción Simple		El sistema verifica si tiene indicado un vértice en el área de trabajo y añade el enlace simple a partir del vértice seleccionado	Satisfactorio	
El especialista selecciona, en la opción Enlace de la barra de herramientas, la opción Doble		El sistema verifica si tiene indicado una arista en el área de trabajo y añade un enlace doble como común a la arista indicada	Satisfactorio	
El especialista selecciona, en la opción Enlace de la barra de herramientas, la opción Doble		El sistema verifica si tiene indicado un vértice en el área de trabajo y añade el enlace doble a partir del vértice seleccionado	Satisfactorio	
El especialista selecciona, en la opción Enlace de la barra de herramientas, la opción Triple		El sistema verifica si tiene indicado una arista en el área de trabajo y añade un enlace triple como común a la arista indicada	Satisfactorio	

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El especialista selecciona, en la opción Enlace de la barra de herramientas, la opción Triple		El sistema verifica si tiene indicado un vértice en el área de trabajo y añade el enlace triple a partir del vértice seleccionado	Satisfactorio	
El especialista selecciona, en la opción Enlace de la barra de herramientas, la opción Cadena		El sistema verifica si tiene indicado un vértice en el área de trabajo y añade el enlace cadena a partir del vértice seleccionado	Satisfactorio	
El especialista selecciona, en la barra de herramientas, la opción Nuevo		El sistema abre una hoja nueva en el área de trabajo donde el especialista podrá crear la molécula con las características que desee	Satisfactorio	
El especialista selecciona, en la barra de herramientas, la opción Guardar Molécula		El sistema abre una ventana de diálogo donde el especialista podrá seleccionar el lugar y nombre del fichero donde va a almacenar la molécula, luego guarda el fichero en 2D con el nombre y extensión especificados y en el destino indicado	Satisfactorio	

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El especialista selecciona, en la barra de herramientas, la opción Deshacer		El sistema deshace la última operación realizada	Satisfactorio	
El especialista selecciona, en la barra de herramientas, la opción Eliminar Enlace		El sistema elimina el enlace que el especialista seleccione	Satisfactorio	
El especialista selecciona, en la barra de herramientas, la opción Limpiar el área de trabajo		El sistema borra todo lo creado por el especialista y pone en blanco el área de trabajo	Satisfactorio	
El especialista selecciona, en la barra de herramientas, la opción Cambiar átomos		El sistema muestra la tabla periódica dando la posibilidad al especialista de escoger cualquier componente y seleccionar en la molécula el átomo por el cual lo cambiará	Satisfactorio	
El especialista selecciona, en la barra de herramientas, la opción Aplicar Escala Zoom In		El sistema acerca la molécula tanto como desee el especialista	Satisfactorio	

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
El especialista selecciona, en la barra de herramientas, la opción Aplicar Escala Zoom Out		El sistema aleja la molécula tanto como desee el especialista	Satisfactorio	

### 2.8.3. Pruebas de Caja Blanca para el visualizador.

Para las pruebas de caja blanca se usa la técnica del camino básico ya que es la más efectiva para obtener todas las variantes en las que se puede ejecutar un procedimiento. Se le realizaron a los métodos que son críticos para la arquitectura ya que son los más importantes para el cumplimiento de las funcionalidades del Visualizador. A continuación se describen los casos de prueba diseñados para estos métodos:

#### 2.8.3.1. Método Open

En el expediente del proyecto, se puede encontrar el código fuente, el grafo de flujo así como la complejidad ciclomática.

##### Camino 1: 1-2-3-2-4-5-18

**Caso de prueba:** Cancelar abrir un fichero.

**Entrada:** Fichero molecular (250.mol).

**Resultado Esperado:** Cierra la ventana de diálogo Abrir.

**Condiciones:** Existencia del fichero.

##### Camino 2: 1-2-3-2-4-5-6-7-8-9-10-9-12-13-15-7-16-17-19

**Caso de prueba:** Abrir cinco ficheros.

**Entrada:** Ficheros moleculares (250.mol, 6.mol, 3177.mol, 299.mol, 593.mol).

**Resultado Esperado:** Abre los ficheros y los adiciona a la lista de ficheros cargados.

**Condiciones:** Existencia de los ficheros moleculares y que al menos haya un fichero abierto recientemente.

**Camino 3:** 1-2-3-2-4-5-6-7-8-9-10-9-12-13-15-7-16-17-18-19

**Caso de prueba:** Abrir un fichero.

**Entrada:** Fichero molecular (3177.mol).

**Resultado Esperado:** Abre el fichero, lo adiciona a la lista de ficheros cargados y lo visualiza en el área de trabajo.

**Condiciones:** Existencia del fichero molecular y que al menos haya un fichero abierto recientemente.

**Camino 4:** 1-2-3-2-4-5-6-7-8-9-10-9-12-13-14-15-7-16-17-19

**Caso de prueba:** Abrir 4 ficheros.

**Entrada:** Ficheros moleculares (3177.mol, 1913.mol, 1821.mol, 929.mol).

**Resultado Esperado:** Abre los ficheros y los adiciona a la lista de ficheros cargados.

**Condiciones:** Existencia de los ficheros moleculares y que existan 10 ficheros abiertos recientemente.

**Camino 5:** 1-2-3-2-4-5-6-7-8-9-10-9-12-13-14-15-7-16-17-18-19

**Caso de prueba:** Abrir un fichero.

**Entrada:** Fichero molecular (1913.mol).

**Resultado Esperado:** Abre el fichero, lo adiciona a la lista de ficheros cargados y lo visualiza en el área de trabajo.

**Condiciones:** Existencia del fichero molecular y que existan 10 ficheros abiertos recientemente.

**Camino 6:** 1-2-3-2-4-5-6-7-8-9-10-11-9-12-7-16-17-19

**Caso de prueba:** Abrir 3 ficheros.

**Entrada:** Ficheros moleculares (1913.mol, 299.mol, 890.mol).

**Resultado Esperado:** Abre los ficheros y los adiciona a la lista de ficheros cargados.

**Condiciones:** Existencia de los ficheros moleculares y que existan 10 ficheros abiertos recientemente.

**Camino 7:** 1-2-3-2-4-5-6-7-8-9-10-11-9-12-7-16-17-18-19

**Caso de prueba:** Abrir un fichero.

**Entrada:** Fichero molecular (299.mol).

**Resultado Esperado:** Abre el fichero, lo adiciona a la lista de ficheros cargados y lo visualiza en el área de trabajo.

**Condiciones:** Existencia del fichero molecular y que existan 10 ficheros abiertos recientemente.

**Camino 8:** 1-2-3-2-4-5-6-7-8-9-12-13-14-15-7-16-17-19

**Caso de prueba:** Abrir 4 ficheros.

**Entrada:** Fichero molecular (6.mol, 1913.mol, 299.mol, 890.mol).

**Resultado Esperado:** Abre los ficheros y los adiciona a la lista de ficheros cargados.

**Condiciones:** Existencia de los ficheros moleculares y que no exista ningún fichero abierto recientemente.

**Camino 9:** 1-2-3-2-4-5-6-7-8-9-12-13-14-15-7-16-17-18-19

**Caso de prueba:** Abrir un fichero.

**Entrada:** Fichero molecular (890.mol).

**Resultado Esperado:** Abre el fichero, lo adiciona a la lista de ficheros cargados y lo visualiza en el área de trabajo.

**Condiciones:** Existencia del fichero molecular y que no exista ningún fichero abierto recientemente.

### 2.8.3.2. Método savePreferences

En el expediente del proyecto, se puede encontrar el código fuente, el grafo de flujo así como la complejidad ciclomática.

**Camino 1:** 1-2-4-5-4-6-7-10

**Caso de prueba:** Salvar preferencias.

**Entrada:** Tema identificado por defecto.

**Resultado Esperado:** Se salva la preferencia.

**Condiciones:** Existencia del tema identificado por defecto.

**Camino 2:** 1-2-4-5-4-6-7-8-9-10

**Caso de prueba:** Salvar preferencias.

**Entrada:** Tema identificado por defecto.

**Resultado Esperado:** Se levanta una excepción.

**Condiciones:** Existencia del tema identificado por defecto, pero no debe existir permiso de escritura en el directorio.

**Camino 3:** 1-2-3-4-5-4-6-7-10

**Caso de prueba:** Salvar preferencias.

**Entrada:** No existe tema identificado por defecto.

**Resultado Esperado:** No se levanta excepción y se salva la preferencia.

**Condiciones:** No debe existir ningún tema identificado por defecto.

**Camino 4:** 1-2-3-4-5-4-6-7-8-9-10

**Caso de prueba:** Salvar preferencias.

**Entrada:** No existe tema identificado por defecto.

**Resultado Esperado:** Se levanta una excepción y no se salva la preferencia.

**Condiciones:** No debe existir ningún tema identificado por defecto.

### 2.8.3.3. Método chequeaMolPos

En el expediente del proyecto, se puede encontrar el código fuente, el grafo de flujo así como la complejidad ciclomática.

**Camino 1:** 1-2-5-6

**Caso de prueba:** Posición elemento.

**Entrada:** Un elemento.

**Resultado Esperado:** Posición no encontrada.

**Condiciones:** No debe haber ningún elemento en la lista.

**Camino 2:** 1-2-3-2-5-6

**Caso de prueba:** Posición elemento.

**Entrada:** Un elemento.

**Resultado Esperado:** Posición no encontrada.

**Condiciones:** Existencia de al menos un elemento en la lista.

**Camino 3:** 1-2-3-4-5-6

**Caso de prueba:** Posición elemento.

**Entrada:** Un elemento.

**Resultado Esperado:** Posición encontrada.

**Condiciones:** Existencia de al menos un elemento en la lista.

#### 2.8.3.4. Método EliminarExtensión

En el expediente del proyecto, se puede encontrar el código fuente, el grafo de flujo así como la complejidad ciclomática.

**Camino 1:** 1-2-4-5

**Caso de prueba:** Eliminar extensión.

**Entrada:** Ningún elemento.

**Resultado Esperado:** No se elimina el elemento.

**Condiciones:** No debe haber ningún elemento en la lista.

**Camino 2:** 1-2-3-2-4-5

**Caso de prueba:** Eliminar extensión.

**Entrada:** Un elemento.

**Resultado Esperado:** Se elimina el elemento.

**Condiciones:** Existencia de al menos un elemento en la lista.

### 2.8.3.5. Método `ChequeaTabExist`

En el expediente del proyecto, se puede encontrar el código fuente, el grafo de flujo así como la complejidad ciclomática.

**Camino 1:** 1-2-5-6

**Caso de prueba:** Posición elemento en tabla.

**Entrada:** Un elemento.

**Resultado Esperado:** Posición no encontrada.

**Condiciones:** La tabla debe estar vacía.

**Camino 2:** 1-2-3-2-5-6

**Caso de prueba:** Posición elemento en tabla.

**Entrada:** Un elemento.

**Resultado Esperado:** Posición no encontrada.

**Condiciones:** Existencia de al menos un elemento en la tabla, no debe existir el elemento entrado en la tabla.

**Camino 3:** 1-2-3-4-5-6

**Caso de prueba:** Posición elemento en tabla.

**Entrada:** Un elemento.

**Resultado Esperado:** Posición encontrada.

**Condiciones:** Existencia de al menos un elemento en la tabla, existencia del elemento entrado en la tabla.

### 2.8.3.6. Método UpdateTable

En el expediente del proyecto, se puede encontrar el código fuente, el grafo de flujo así como la complejidad ciclomática.

**Camino 1:** 1-2-9-10

**Caso de prueba:** Actualizar tabla de extensiones.

**Entrada:** Lista de extensiones sin elementos.

**Resultado Esperado:** Crea la tabla vacía.

**Condiciones:** No de existir elementos en lista de extensiones.

**Camino 2:** 1-2-3-4-5-8-2-9-10

**Caso de prueba:** Actualizar tabla de extensiones.

**Entrada:** Un elemento de la lista de extensiones.

**Resultado Esperado:** Crear Tabla.

**Condiciones:** Existencia de al menos un elemento en lista de extensiones .

**Camino 3:** 1-2-3-4-5-6-7-8-2-9-10

**Caso de prueba:** Actualizar tabla de extensiones.

**Entrada:** Elemento con defecto.

**Resultado Esperado:** Trata excepción y crea la tabla.

**Condiciones:** Existencia de al menos un elemento en lista de extensiones.

#### 2.8.4. Prueba de Integración.

Tabla 2.27: Casos de prueba para las pruebas de Integración

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	Teniendo seleccionada una molécula en el Visualizador, se selecciona en el menú Preferencias la opción Editor Molecular	El sistema abre el Editor Molecular donde se muestra la molécula dibujada en 2D y donde el especialista podrá hacerle modificaciones de acuerdo a las opciones que ofrece el Editor, luego de guardados los cambios se visualiza en el Visualizador la molécula editada en 3D	Insatisfactorio	El Editor Molecular no visualiza bien la molécula en 2D

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	Teniendo seleccionada una molécula en el Visualizador, se selecciona en el menú Preferencias la opción Editor Molecular	El sistema abre el Editor Molecular donde se muestra la molécula dibujada en 2D visualizada en el centro del área de trabajo y donde el especialista podrá hacerle modificaciones de acuerdo a las opciones que ofrece el Editor, luego de guardados los cambios se visualiza en el Visualizador la molécula editada en 3D	Insatisfactorio	El Editor no visualiza, en el centro del área de trabajo, la molécula
Sin tener seleccionada ninguna molécula en el Visualizador, se selecciona en el menú Preferencias la opción Editor Molecular		El sistema abre el Editor Molecular donde el especialista podrá crear una molécula de acuerdo a las opciones que este ofrece y al guardar los cambios realizados se visualiza en el Visualizador la molécula creada en 3D	Satisfactorio	

### 2.8.5. Prueba de Carga.

Para el diseño de esta prueba se partió de los requisitos no funcionales del sistema los cuales plantean que un especialista puede visualizar a lo sumo 25 moléculas y pueden estar trabajando 5 especialistas simultáneamente.

Tabla 2.28: Casos de prueba para las pruebas de Carga

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
En una PC se abre el Visualizador y se seleccionan 10 moléculas a cargar con la extensión .cml		El sistema muestra en la lista de ficheros abiertos las 10 moléculas cargadas	Satisfactorio	
En una PC se abre el Visualizador y se seleccionan 20 moléculas a cargar con la extensión .alc		El sistema muestra en la lista de ficheros abiertos las 20 moléculas cargadas	Satisfactorio	
En una PC se abre el Visualizador y se seleccionan 30 moléculas a cargar con la extensión .mdl		El sistema muestra en la lista de ficheros abiertos las 30 moléculas cargadas	Satisfactorio	
En una PC se abre el Visualizador y se seleccionan 40 moléculas a cargar con la extensión .hin		El sistema muestra en la lista de ficheros abiertos las 40 moléculas cargadas	Satisfactorio	

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
	En 5 PC se abre el Visualizador y se seleccionan 40 moléculas a cargar con la extensión .cml	El sistema, en las 5 PC, muestra en la lista de ficheros abiertos las 40 moléculas cargadas	Insatisfactorio	El sistema muestra, en una sola PC, las moléculas cargadas en la lista de ficheros abiertos y en las demás máquinas muestra el mensaje de error "Servicio no disponible"

### 2.8.6. Prueba de Instalación.

Tabla 2.29: Casos de prueba para las pruebas de Instalación

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
Se ejecuta el instalador de la aplicación bioGRATOIn-staller.jar dando doble clic sobre dicha aplicación		Se muestra una interfaz donde se puede seleccionar el idioma que se debe mostrar durante el proceso de instalación	Satisfactorio	
Se selecciona el idioma y se presiona el botón OK		Se muestra una pantalla donde aparecen los datos fundamentales del equipo de desarrollo	Satisfactorio	

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
Se presiona el botón Siguiente en la parte inferior derecha		Aparece una nueva pantalla donde se debe leer detenidamente la información sobre el software que aparece en la misma	Satisfactorio	
Se presiona el botón Siguiente en la parte inferior derecha		Se muestra una nueva pantalla donde se elige el directorio de instalación de la aplicación	Satisfactorio	
Se presiona el botón Siguiente		Si en la ruta de instalación seleccionada existe ya una carpeta con el nombre bioGRATO el instalador muestra el mensaje "El directorio ya existe!" Confirma su deseo de realizar la instalación ahí y sobrescribir los posibles ficheros existentes?, si no existe ya la carpeta entonces muestra el mensaje "Se creará el directorio destino"	Satisfactorio	
Se presiona el botón Aceptar		Aparece una ventana donde se pueden seleccionar todos los paquetes que se quieren instalar	Satisfactorio	

Clases válidas	Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
Se selecciona el único paquete que existe en esta primera versión y se presiona el botón Siguiente		El sistema comienza a instalarse en la computadora	Satisfactorio	
Se presiona el botón Siguiente		Se muestra una pantalla donde se puede seleccionar, si se desea, crear accesos directos de la aplicación	Satisfactorio	
Se presiona el botón Siguiente		Se muestra una pantalla con información de que el proceso de instalación ha sido satisfactorio	Satisfactorio	
Se presiona el botón Hecho		La aplicación queda instalada en la máquina para su posterior ejecución	Satisfactorio	
Se presiona en cualquier paso de la instalación el botón Salir		El proceso de instalación es abortado	Satisfactorio	

## 2.9. Segunda Iteración

Después de la ejecución de las pruebas en la primera iteración, se recogieron las no conformidades detectadas y fueron entregadas al equipo de desarrollo para darle respuesta a cada una de ellas. Luego se realizó una segunda iteración aplicando las mismas pruebas diseñadas en la primera iteración, donde se comprobó que los defectos encontrados en la misma habían sido solucionados correctamente y que el producto está libre de errores y listo para entregar al grupo de calidad de la facultad. El diseño de los casos de prueba para esta segunda iteración se encuentra archivado en el expediente de proyecto.

## 2.10. Conclusiones

Con el fin de lograr un producto de alta calidad, en este capítulo se explican las características que tendrá el plan de pruebas diseñado para el Visualizador y Editor Molecular y la configuración que tendrá el entorno que será empleado en la etapa de realización de pruebas. Se desarrolló un plan de pruebas donde se recogen procedimientos y estrategias a seguir, siendo estos favorables para la realización de las pruebas con éxito, agilizando este proceso y garantizando una buena disciplina de trabajo. Se diseñaron los casos de prueba que son la base para el desarrollo de las mismas y se realizaron varias iteraciones donde se ejecutaron las pruebas necesarias y se recogieron las no conformidades encontradas, las cuales fueron entregadas al equipo de desarrollo con el fin de eliminar los errores detectados y dándole cumplimiento a las funcionalidades del producto.

## Capítulo 3

# RESULTADOS DE LAS PRUEBAS

### 3.1. Introducción

Este capítulo hará un análisis de los resultados obtenidos después de diseñar y aplicar las pruebas al Visualizador y Editor Molecular, exponiendo detalladamente cada uno de los defectos y dificultades detectadas en cada una de las iteraciones realizadas. Para ello se detallan las clases de equivalencia diseñadas y las no conformidades detectadas en cada una de las pruebas que fueron aplicadas.

### 3.2. Análisis de los resultados obtenidos en las pruebas de Caja Negra en una primera iteración.

Las pruebas aplicadas fueron diseñadas por dos probadoras del equipo de desarrollo. Se utilizó una PC con Microprocesador Pentium IV, Sistema Operativo Linux, con al menos 512 MB de memoria RAM y 160 GB de Disco Duro. Para cada caso de uso se confeccionó un caso de prueba compuesto por clases de equivalencia con datos tanto válidos como no válidos y obteniendo los siguientes resultados en cada una de las pruebas ejecutadas:

#### 3.2.1. Registro de defectos y dificultades detectadas para el Visualizador

*CU: Abrir Fichero*

Tabla 3.1: Registro de defectos y dificultades detectadas para el CU Abrir Fichero

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Interfaz	1	El sistema no muestra el mensaje de error: "Fichero ya visualizado"	Cuando se selecciona, en la ventana de diálogo Abrir, un fichero ya cargado	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Implementar para la segunda iteración
Interfaz	2	El sistema no muestra el mensaje de error "El fichero seleccionado no existe"	Cuando se escribe en la ventana de diálogo Abrir el nombre de un fichero que no existe: molécula.mol	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Implementar para la segunda iteración
Interfaz	3	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 1330.unixyz	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	4	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 250.res	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Implementar para la segunda iteración

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Interfaz	5	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 1483.qcout	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	6	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 1701.ent	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	7	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 890.nwo	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	8	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 593.mopout	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	9	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 1649.mopcrt	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Interfaz	10	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 6.mdl	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	11	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 1483.sdf	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	12	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 1821.jout	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	13	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 1330.qm1pg	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	14	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 250.mm1gp	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Interfaz	15	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 6.gpr	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	16	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 3177.gamout	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	17	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 929.gam	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	18	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 1913.scf	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	19	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 593.cml	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Interfaz	20	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 299.cif	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	21	Adiciona la molécula a la lista de ficheros abiertos pero no la visualiza	Cuando se selecciona en la ventana de diálogo el fichero 1913.mol2	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	22	El sistema muestra activa la opción Abrir Reciente cuando no se ha abierto ningún fichero recientemente	Cuando se va a la opción Abrir Reciente del menú Archivo	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	23	El sistema no muestra el mensaje de error: "Error en tipo de fichero"	Cuando se va a la ventana de diálogo Abrir Preferencias de Visualización	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Implementar para la segunda iteración

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Interfaz	24	El sistema no muestra el mensaje de error: "El fichero no existe"	Cuando se escribe en la ventana de diálogo Abrir Preferencias de Visualización el nombre de un fichero que no existe	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Implementar para la segunda iteración

En el Anexo 1 se muestra de forma visual la no conformidad número 1, la cual plantea que el sistema no muestra el mensaje de error: "Fichero ya visualizado" cuando el especialista selecciona abrir una molécula que ya está visualizada.

En el Anexo 2 se muestra de forma visual la no conformidad número 2, la cual plantea que el sistema no muestra el mensaje de error "El fichero seleccionado no existe" cuando el especialista escribe el nombre de una molécula o proteína que no existe.

En el Anexo 3 se muestra de forma visual la no conformidad número 23, la cual plantea que el sistema no muestra el mensaje de error: "Error en tipo de fichero" cuando el especialista escribe el nombre de un fichero de preferencias de visualización que no existe.

### ***CP: Gestionar Fichero***

Tabla 3.2: Registro de defectos y dificultades detectadas para el CU Gestionar Fichero

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
----------	----	----------------	-------------------------	-------------------------------------	-------------	---------------

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Interfaz	1	Si no está marcado un fichero molecular el sistema no elimina todos los ficheros	Cuando se selecciona en el menú Archivo la opción Eliminar Todo	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Implementar para la segunda iteración
Interfaz	2	No se encuentra la opción Guardar Como implementada	Cuando se va al menú Archivo	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Interfaz	3	El sistema solo tiene especificado en el tipo de archivo que se desea guardar Todos los archivos y no tiene de tipo fichero de Preferencia de Visualización	Cuando se va a la ventana de diálogo Guardar Preferencia de Visualización	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Implementar para la segunda iteración

En el Anexo 4 se muestra de forma visual la no conformidad número 3, la cual plantea que el sistema solo tiene especificado en el tipo de archivo que se desea guardar Todos los archivos y no tiene de tipo fichero de Preferencia de Visualización.

***CU: Gestionar Selección***

Tabla 3.3: Registro de defectos y dificultades detectadas para el CU Gestionar Selección

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Ortográfico	1	Hay un error ortográfico, la palabra Proteína está escrita sin tilde	Cuando se va a la opción Proteína del menú Seleccionar	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Arreglar para la segunda iteración
Ortográfico	2	Hay un error ortográfico, la palabra Proteína está escrita sin tilde	Cuando se va a la opción Toda la Proteína dentro de la opción Proteína del menú Seleccionar	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Arreglar para la segunda iteración
Ortográfico	3	Hay un error ortográfico, la palabra Aminoácidos está escrita sin tilde	Cuando se va a la opción Aminoácidos dentro de la opción Proteína del menú Seleccionar	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Arreglar para la segunda iteración
Ortográfico	4	Hay un error ortográfico, la palabra Selección está escrita sin tilde	Cuando se va a la opción Invertir Selección del menú Seleccionar	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Arreglar para la segunda iteración

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Ortográfico	5	Hay un error ortográfico, la palabra Hidrógenos está escrita sin tilde	Cuando se va a la opción Hidrógenos del menú Seleccionar	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Arreglar para la segunda iteración
Ortográfico	6	Hay un error ortográfico, la palabra Nitrógeno está escrita sin tilde	Cuando se va a la opción Nitrógeno del menú Seleccionar	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Arreglar para la segunda iteración
Ortográfico	7	Hay un error ortográfico, la palabra Oxígeno está escrita sin tilde	Cuando se va a la opción Oxígeno del menú Seleccionar	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Arreglar para la segunda iteración
Ortográfico	8	Hay un error ortográfico, la palabra Fósforo está escrita sin tilde	Cuando se va a la opción Fósforo del menú Seleccionar	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Arreglar para la segunda iteración
Ortográfico	9	Hay un error ortográfico, la palabra Halógenos está escrita sin tilde	Cuando se va a la opción Halógenos del menú Seleccionar	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Arreglar para la segunda iteración

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Ortográfico	10	Hay un error ortográfico, la palabra Flúor está mal escrita y no está acentuada	Cuando se va a la opción Flúor dentro de la opción Halógenos del menú Seleccionar	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizado	No significativo	Arreglar para la segunda iteración
Ortográfico	11	Hay un error ortográfico, la palabra Selección está escrita sin tilde	Cuando se va a la opción Invertir Selección del menú Seleccionar	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	No significativo	Arreglar para la segunda iteración

**CU: Gestionar Visualización**

Tabla 3.4: Registro de defectos y dificultades detectadas para el CU Gestionar Visualización

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Interfaz	1	El sistema no visualiza nada	Cuando se escoge dentro de la opción Representaciones del menú Visualizar la opción Cohetes	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Interfaz	2	El sistema no visualiza nada	Cuando se escoge dentro de la opción Representaciones del menú Visualizar la opción Ninguna	Etapa de Diseño y Aplicación de Pruebas al módulo Visualizador	Significativo	Implementar para la segunda iteración
Ortográfico	3	Cuando se va a la opción Estereográfica dentro de la opción Vistas en el menú Visualizar	Hay un error ortográfico, la palabra Estereográfica está escrita sin tilde	Etapa de Implementación y pruebas del módulo Visualizador	No significativo	Arreglar para la segunda iteración
Ortográfico	4	Cuando se va a la opción Visión Bizca(Cross-Eyed) que se encuentra dentro de la opción Estereográfica dentro Vistas en el menú Visualizar	Hay un error ortográfico, la palabra Visión está escrita sin tilde	Etapa de Implementación y pruebas del módulo Visualizador	No significativo	Arreglar para la segunda iteración

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Ortográfico	5	Cuando se va a la opción Visión Paralela(Wall-Eyed) que se encuentra dentro de la opción Estereográfica dentro Vistas en el menú Visualizar	Hay un error ortográfico, la palabra Visión está escrita sin tilde	Etapa de Implementación y pruebas del módulo Visualizador	No significativo	Arreglar para la segunda iteración

En el Anexo 5 se muestra de forma visual la no conformidad número 1 y 2, las cuales plantean que el sistema no visualiza nada cuando el especialista escoge dentro de la opción Representaciones del menú Visualizar la opción Cohetes ó Ninguna.

***CU: Cambiar Preferencia de Visualización***

Tabla 3.5: Registro de defectos y dificultades detectadas para el CU Cambiar Preferencias

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Ortográfico	1	Cuando se va a la opción Hidrógenos	Hay un error ortográfico, la palabra Hidrógenos está escrita sin tilde	Etapa de Implementación y prueba del módulo Visualizador	No significativo	Arreglar para la segunda iteración

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Ortográfico	2	Cuando se va a la opción Alineación de etiquetas	Hay un error ortográfico, la palabra Alineación está escrita sin tilde	Etapa de Implementación y prueba del módulo Visualizador	No significativo	Arreglar para la segunda iteración
Ortográfico	3	Cuando se va a la opción Mostrar Halos de Selección	Hay un error ortográfico, la palabra Selección está escrita sin tilde	Etapa de Implementación y prueba del módulo Visualizador	No significativo	Arreglar para la segunda iteración
Ortográfico	4	Cuando se va a la opción Mostrar solo la selección	Hay un error ortográfico, la palabra selección está escrita sin tilde	Etapa de Implementación y prueba del módulo Visualizador	No significativo	Arreglar para la segunda iteración
Ortográfico	5	Cuando se va a la opción Nanómetros IE-09	Hay un error ortográfico, la palabra Nanómetros está escrita sin tilde	Etapa de Implementación y prueba del módulo Visualizador	No significativo	Arreglar para la segunda iteración

**CU: Gestionar Módulos**

Tabla 3.6: Registro de defectos y dificultades detectadas para el CU Gestionar Módulos

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Ortográfico	1	Cuando se va a la opción Módulos del menú Preferencias	Hay un error ortográfico, la palabra Módulos está escrita sin tilde	Etapa de Implementación y prueba de los módulos Visualizador y Editor	No significativo	Arreglar para la segunda iteración
Ortográfico	2	Cuando se va a la opción Adicionar Módulos de la opción Módulos del menú Preferencias	Hay un error ortográfico, la palabra Módulos está escrita sin tilde	Etapa de Implementación y prueba de los módulos Visualizador y Editor	No significativo	Arreglar para la segunda iteración
Ortográfico	3	Cuando se va a la opción Eliminar Módulos de la opción Módulos del menú Preferencias	Hay un error ortográfico, la palabra Módulos está escrita sin tilde	Etapa de Implementación y prueba de los módulos Visualizador y Editor	No significativo	Arreglar para la segunda iteración

### 3.2.2. Registro de defectos y dificultades detectadas para el Editor Molecular:

*CU: Modificar Molécula*

Tabla 3.7: Registro de defectos y dificultades detectadas para el CU Modificar Moléculas

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
----------	----	----------------	-------------------------	-------------------------------------	-------------	---------------

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Interfaz	1	El sistema dibuja mal la molécula abierta en 2D	Cuando se va a la opción Abrir en la barra de herramientas	Etapa de Diseño y Aplicación de pruebas del módulo Editor Molecular	Significativo	Arreglar para la segunda iteración

### 3.3. Análisis de los resultados obtenidos en las pruebas de Caja Blanca en una primera iteración.

La aplicación de las pruebas de caja blanca a los métodos del Visualizador no arrojaron ninguna no conformidad.

### 3.4. Análisis de los resultados obtenidos en las pruebas de Integración en una primera iteración.

Estas pruebas se ejecutaron en una PC con Microprocesador Pentium IV, Sistema Operativo Linux, con al menos 512 MB de memoria RAM y 160 GB de Disco Duro. Se confeccionaron los casos de prueba compuestos por clases de equivalencia con datos tanto válidos como no válidos y obteniendo los siguientes resultados:

Tabla 3.7: Registro de defectos y dificultades detectadas en las pruebas de Integración

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
----------	----	----------------	-------------------------	-------------------------------------	-------------	---------------

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Interfaz	1	El Editor Molecular dibuja mal en 2D la molécula abierta	Cuando se integra el Editor Molecular con una molécula visualizada en 3D en el visualizador	Etapa de Diseño y Aplicación de Pruebas a los módulos Visualizador y Editor Molecular	Significativo	Arreglar para la segunda iteración
Interfaz	2	El Editor Molecular no visualiza, en el centro del área de trabajo, la molécula	Cuando se integra el Editor Molecular con una molécula visualizada en 3D en el visualizador	Etapa de Diseño y Aplicación de Pruebas a los módulos Visualizador y Editor Molecular	Significativo	Arreglar para la segunda iteración

### 3.5. Análisis de los resultados obtenidos en las pruebas de Carga en una primera iteración.

Para esta prueba se utilizaron 5 PC con Microprocesador Pentium IV, Sistema Operativo Linux, con al menos 512 MB de memoria RAM y 160 GB de Disco Duro. Se confeccionaron casos de prueba compuestos por clases de equivalencia con datos tanto válidos como no válidos y obteniendo los siguientes resultados:

Tabla 3.8: Registro de defectos y dificultades detectadas en las pruebas de Carga

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección de Código del CP	Importancia	Recomendación
----------	----	----------------	-------------------------	--	-------------	---------------

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección de Código del CP	Importancia	Recomendación
Interfaz	1	El sistema carga las 40 moléculas en una sola PC y muestra en las otras máquinas el mensaje de error "Servicio no disponible"	Al cargar 40 moléculas en 5 PC	Etapa de Diseño y Aplicación de Pruebas del módulo Visualizador	Significativo	Arreglar para la segunda iteración

### 3.6. Análisis de los resultados obtenidos en las pruebas de Instalación en una primera iteración.

La aplicación de las pruebas de instalación al Visualizador y Editor Molecular no arrojaron ninguna no conformidad.

### 3.7. Análisis de los resultados obtenidos en las pruebas en la segunda iteración.

Al finalizar la primera iteración las dificultades encontradas fueron entregadas al equipo de desarrollo del proyecto para su posterior solución. Luego se realizó una segunda iteración de pruebas donde se pudo comprobar que las mismas fueron solucionadas, en su totalidad, por lo que de forma general se puede decir que el software Visualizador y Editor Molecular desarrollado en la plataforma bioGRATO, cumple con todas las funcionalidades especificadas.

### 3.8. Conclusiones

Después de un profundo análisis de los resultados obtenidos al aplicar cada una de las pruebas diseñadas en la primera iteración para el Visualizador y Editor Molecular, podemos resumir que se registraron y documentaron los 55 errores encontrados, donde 26 de ellos fueron significativos y 29 no significativos. Los 26 significativos

tienen gran importancia pues pueden influir en el cumplimiento de las funcionalidades del software y pueden poner en duda la eficiencia y confiabilidad del producto. Los 29 restantes, en su mayoría de interfaz, no fueron de gran relevancia pues no contribuyeron a fallas en la aplicación y la funcionalidad no sufrió afectaciones. Los errores ortográficos no provocan afectaciones en la ejecución del software pero si van contra la calidad y la estética del producto. Es importante señalar que en las pruebas de caja blanca no se detectaron errores cumpliendo así con lo esperado.

En la segunda iteración se pudo comprobar que todos los errores fueron solucionados por lo que se puede afirmar que el software está listo para ser liberado por el grupo de calidad de la facultad, contando con toda la documentación necesaria.

# CONCLUSIONES

Luego de aplicar todo el proceso de pruebas al Visualizador y Editor Molecular de la proyecto bioGRATO y demostrar la importancia que tiene la ejecución de las mismas en el desarrollo del software para lograr una adecuada detección de errores y valorar el beneficio que aportan al software en construcción las fallas encontradas, se puede concluir que:

1. Se demostró que el Visualizador y Editor Molecular cumple con los requisitos de software especificados.
2. Se realizó un plan de prueba, en el que se incluye todo lo necesario con el fin de guiar todo un proceso de pruebas.
3. Se diseñaron y aplicaron, al Visualizador y Editor Molecular, pruebas de Caja Negra, Caja Blanca, Integración, Carga y de Instalación en dos iteraciones. Detectándose 55 no conformidades en la primera iteración y ninguna en la segunda.
4. Se documentaron todos los errores detectados y se entregaron al grupo de desarrollo del proyecto las solicitudes de cambio.

# RECOMENDACIONES

Para lograr la liberación del producto, que es el objetivo final del proyecto, se recomienda:

- Realizar las pruebas de Aceptación.
- Entregar el Visualizador y Editor Molecular al grupo de calidad de la facultad para su liberación.

## Referencias

- [1] IEEE, Standard Glossary of Software Engineering Terminology. Software Engineering Standards Collection, 1990.
- [2] Pressman, Roger S. Ingeniería del software. Un enfoque práctico. Cuarta. s.l. : McGraw Hill, 1998.
- [3] Normalización Técnica Andina - Comunidad Andina -CAN- ISO 9126
- [4] Juan Manuel. Cueva, [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF).
- [5] Iden a la Referencia 4.
- [6] Iden a la Referencia 4.
- [7] ISO9000. Aplicación de las normas ISO 9000 a la enseñanza y la formación
- [8] Iden a la Referencia 4.
- [9] Acuña.S, <http://kybele.escet.urjc.es/docencia/IS4/2007-2008/Material/Pruebas%20de%20Software.pdf>
- [10] Aplicación de un Método para Especificar Casos de Prueba de Software en la Administración Pública ( Mendez )<http://www.sistedes.es/TJISBD/Vol-1/No-4/articles/pris-07-mendez-amecp.pdf>
- [11] Valencia, [http://eisc.univalle.edu.co/materias/Material\\_Desarrollo\\_Software/PRUEBASoftware.pdf](http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/PRUEBASoftware.pdf)
- [12] Iden a la Referencia 2.
- [13] Beizer, [http://www.aleshores.net/IMG/pdf/MTP\\_tema\\_9.pdf](http://www.aleshores.net/IMG/pdf/MTP_tema_9.pdf)
- [14] OpenUP, <http://www.openup.es/>

# Bibliografía

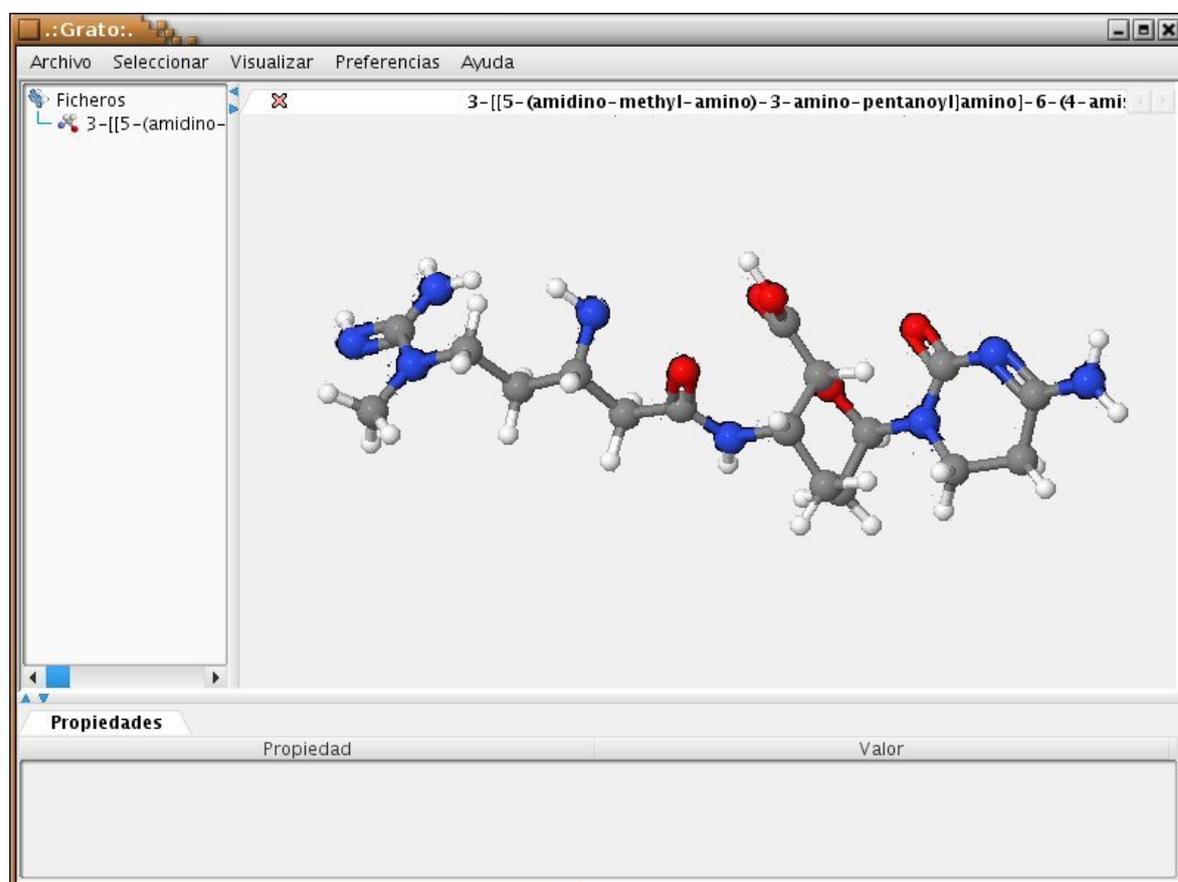
- [1] Acunna.S,[http://kybele.escet.urjc.es/docencia/IS4/2007-2008/Material/Pruebas %20de %20Software.pdf](http://kybele.escet.urjc.es/docencia/IS4/2007-2008/Material/Pruebas%20de%20Software.pdf)
- [2] Beizer, [http://www.aleshores.net/IMG/pdf/MTP\\_tema\\_9.pdf](http://www.aleshores.net/IMG/pdf/MTP_tema_9.pdf)
- [3] Baeza\_Yates, R. A. and others. Computing in Chile: The Jaguar of the Pacific Rim? Communications of the ACM. Vol. 38, No 9, 1995.
- [4] Calidad de software. Cueva Lovelle, Juan Manuel. Pampa, España : s.n., 1999.
- [5] Calidad de Software,Pruebas del Software,[En Linea], [Disponible en: [http://kybele.escet.urjc.es/documentos/ISG/ %5BISG-2006-07 %5DPuebasSoftware.pdf](http://kybele.escet.urjc.es/documentos/ISG/%5BISG-2006-07%5DPuebasSoftware.pdf)
- [6] Calidad en ingeniería del software, Calidad en ingeniería del software[En Linea] [Disponible en: [dmi.uib.es/~bbuades/calidad/index.htm](http://dmi.uib.es/~bbuades/calidad/index.htm)
- [7] Coherencia y calidad en la edicion miultilingue,Control de la Calidad del Software[En Linea] [Disponible en: [http://www.sdl.com/es/services/localisation\\_services/software\\_qa.asp](http://www.sdl.com/es/services/localisation_services/software_qa.asp)
- [8] Calidad y software. com ,Pruebas unitarias,[En Linea], [Disponible en: [http://www.calidadysoftware.com/testing/pruebas\\_unitarias1.html](http://www.calidadysoftware.com/testing/pruebas_unitarias1.html)
- [9] Calidad del Software,Calidad ISO 9000[En Linea][Disponible en: <http://www.adrformacion.com/cursos/calidad/calidad.html>
- [10] CARRILLO, L. V. L. Caracterización de la Prueba de Software: Clasificación y Técnicas, 2005.
- [11] Diseno y ejecucion de las pruebas pruebas, El proceso de prueba[En Linea] [ Disponible en: [http://148.202.148.5/cursos/cc321/fundamentos/unidad6/tema6\\_1.html](http://148.202.148.5/cursos/cc321/fundamentos/unidad6/tema6_1.html)

- [12] Díaz.J.F. Metodos de pruebas de programas[En Linea] [Disponible en: [http://www.galeon.com/neoprogramadores/met\\_test.htm](http://www.galeon.com/neoprogramadores/met_test.htm)
- [13] El Contexto de la Prueba de Software: Probar para Incrementar la Calidad 2005
- [14] Fernández Carrasco, Oscar M., García León, Delba y Beltrán Benavides, Alfa. Calidad del software,Un enfoque actual sobre la calidad del software. 1995.[Disponible en : [http://bvs.sld.cu/revistas/aci/vol3\\_3\\_95/aci05395.htm](http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm)
- [15] Fundamentos de la Prueba de Software: Conceptos, Justificación y Alcance, 2005.
- [16] González, C. Un plan de Pruebas Exitoso.[En Linea] 2002 [Disponible en:<http://www.americaxxi.cl/modules.php?name=News&file=article&sid=20>.
- [17] Gustavo A. Valencia R.[http://www.greensqa.com/archivos/Servicio %20Pruebas %20de %20Software.pdf](http://www.greensqa.com/archivos/Servicio%20Pruebas%20de%20Software.pdf)
- [18] GONZÁLES, J. "Las normas de la Calidad del Software". España, Addison-Wesley, 2002. p.
- [19] [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF\(15/10/2006\)](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF(15/10/2006))
- [20] IEEE, Standard Glossary of Software Engineering Terminology. Software Engineering Standars Collection, 1990.
- [21] ISO 9000:2005 . Sistemas de Gestión de la calidad. Principios Fundamentos y Vocabulario. Secretaría Central de ISO, Traducción certificada.[2005]
- [22] ISO 9000:2000, "Sistema de Gestión de la Calidad - Principios y Vocabulario", 2000.
- [23] ISO9000. Aplicación de las normas ISO 9000 a la enseñanza y la formación
- [24] [ITI, 2007] "Testeo de funcionalidad", 2007. [Disponible en: [http://squac.iti.upv.es/servicios/testeo/testeo\\_de\\_](http://squac.iti.upv.es/servicios/testeo/testeo_de_)
- [25] Juan Manuel. Cueva, [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF).
- [26] Lovelle, J. M. C. Calidad del Software, Calidad del Software Grupo GIDIS Universidad Nacional de la Pampa, [En Linea]1999[Disponible en: [www.uniovi.es](http://www.uniovi.es).
- [27] Mañas, J.A.Calidad del Software ,Calidad de las pruebas.[En Linea] [Disponible en: <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/calidad.htm>.

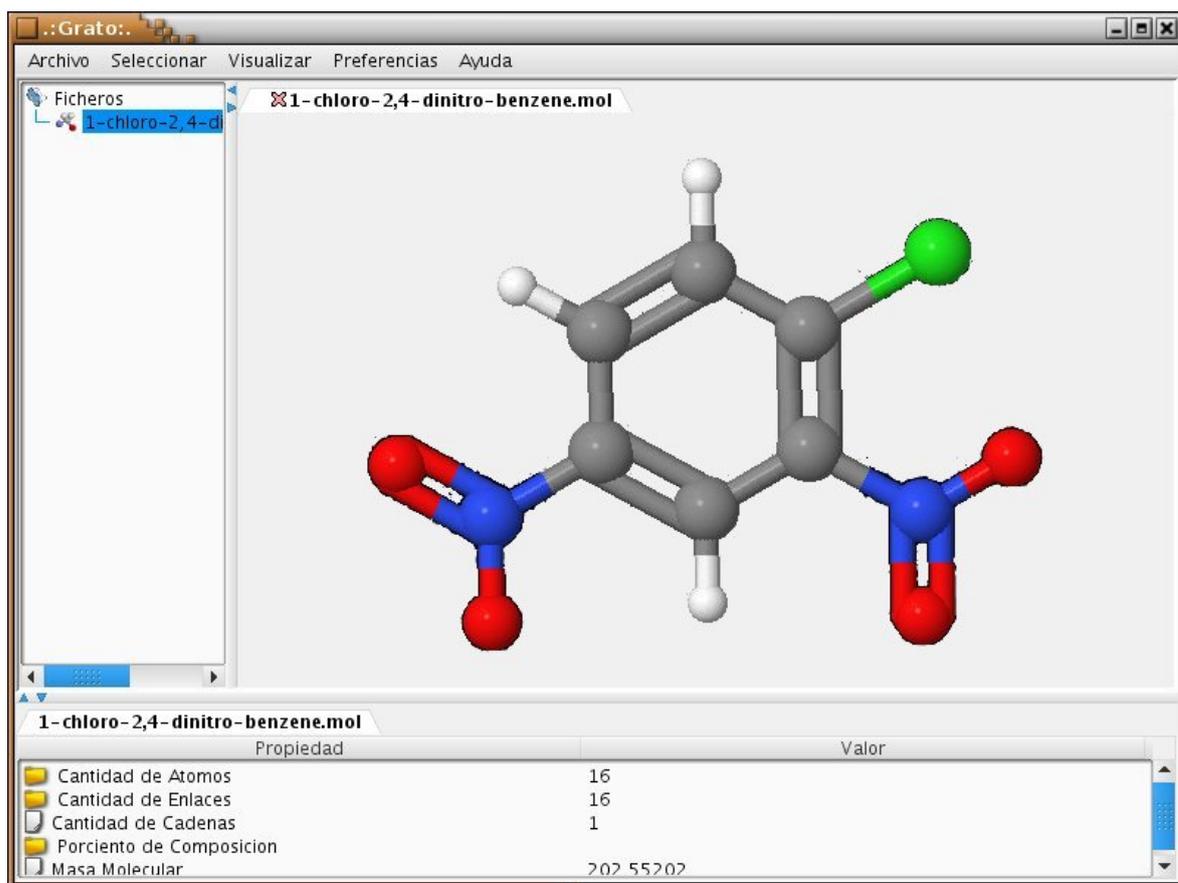
- [28] Mendez ,<http://www.sistedes.es/TJISBD/Vol-1/No-4/articulos/pris-07-mendez-amecp.pdf>
- [29] Metodología, Open UP, <http://epf.eclipse.org/wikis/openup/index.htm>
- [30] Monografías, Modelo para Pruebas de Software y auditoría en Entorno Microsoft.Net [Disponible en: [http://www.monografias.com/trabajos20/pruebas-de-software/pruebas-software.shtml#teoroico\(9/4/2007\)](http://www.monografias.com/trabajos20/pruebas-de-software/pruebas-software.shtml#teoroico(9/4/2007))]
- [31] Mañas, J.A. Calidad del Software, Pruebas de Programas. [En Línea] 1994 [Disponible en: <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm>]
- [32] MEYER, B. Capítulo 1 Calidad del Software en: Construcción de Software Orientado a Objetos. Segunda Edición. Madrid, 1999a.p. Bibliografía IEEE, Standard Glossary of Software Engineering Terminology. Software Engineering Standards Collection, 1990.
- [33] Normalización Técnica Andina - Comunidad Andina -CAN- ISO 9126
- [34] Pressman, Roger S. Ingeniería del software. Un enfoque práctico. Cuarta. s.l. : McGraw Hill, 1998.
- [35] Patricio, Letelier. Pruebas del Software. Valencia : Departamento de Sistemas Informáticos y Computación.
- [36] Peña, J.M.F., Pruebas de integración para componentes de software. 2002, Mexico.
- [37] Pruebas de software, [En Línea], [Disponible en: <http://lml.ls.fi.upm.es/ftp/ed2/0203/Apuntes/pruebas.ppt>]
- [38] Pruebas del software, [En Línea], [Disponible en: <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema09.pdf>]
- [39] Piattini, Mario, García Félix O, "Calidad en el desarrollo y mantenimiento del software", RA-MA Editorial, Madrid, 2003, 310 p., ISBN 970-15-899-8
- [40] QAGROUPS.QAGroups, Empresa de pruebas de software. [En Línea] 2006. [Disponible en : <http://www.qagroup.com.mx/>]
- [41] Rodríguez, D. E. Introducción a la Calidad del Software, Septiembre, 2003.
- [42] Software de control remoto para aplicaciones del escritorio GNOME, Prueba de software Terminado [En Línea] [Disponible en: [http://gbtcr.chileforge.cl/info\\_web/node139.html](http://gbtcr.chileforge.cl/info_web/node139.html)]

- [43] SCALONE, L. F. "Estudio Comparativo de los Modelos y Estándares de Calidad del Software". Buenos Aires, Universidad Tecnológica Nacional, 2006. p.
- [44] Valencia, [http://eisc.univalle.edu.co/materias/Material\\_Desarrollo\\_Software/PRUEBASoftware.pdf](http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/PRUEBASoftware.pdf)
- [45] Valladares, T. R. Junit.
- [46] Wikipedia, Caso de prueba. [En Línea] Enero 2007 [Disponible en: [http://es.wikipedia.org/wiki/Caso\\_de\\_prueba](http://es.wikipedia.org/wiki/Caso_de_prueba)].

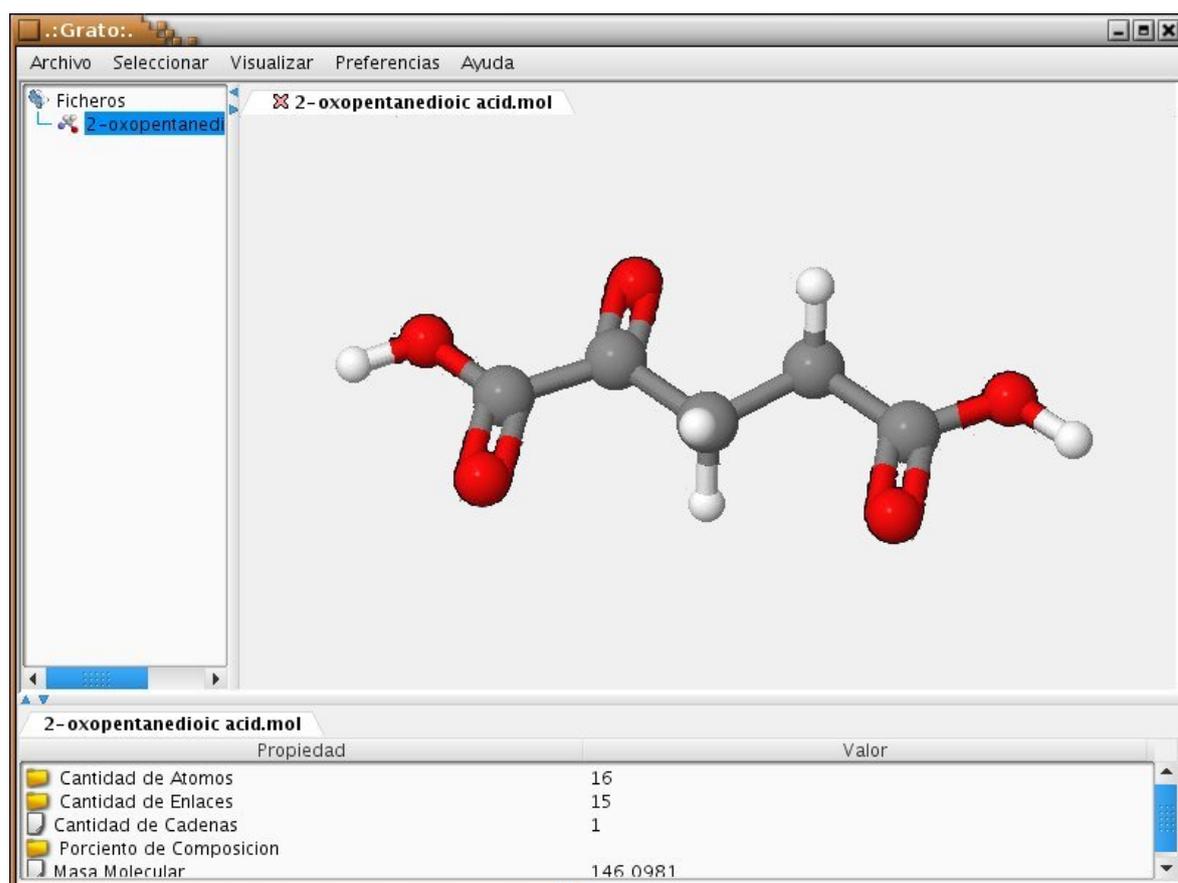
## Anexo 1: Error en la Funcionalidad Abrir



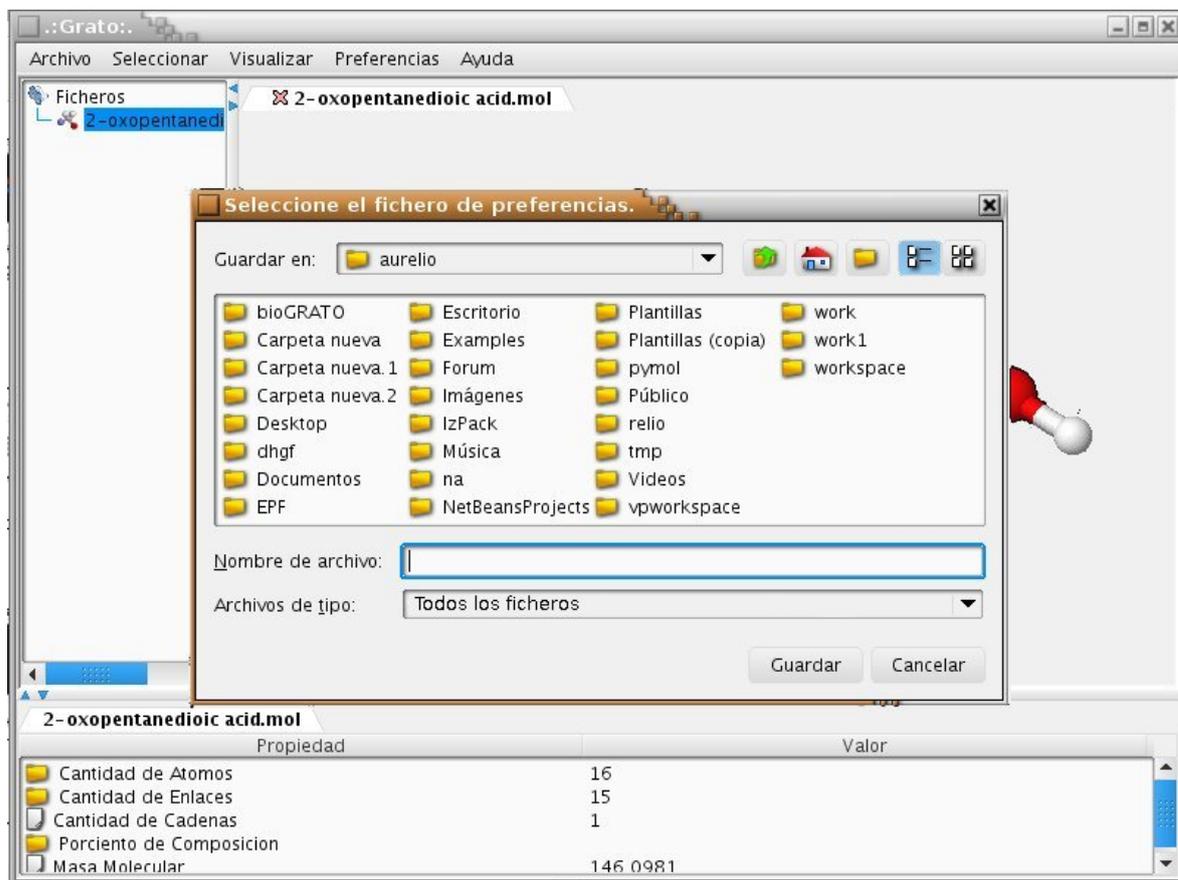
## Anexo 2: Error en la Funcionalidad Abrir



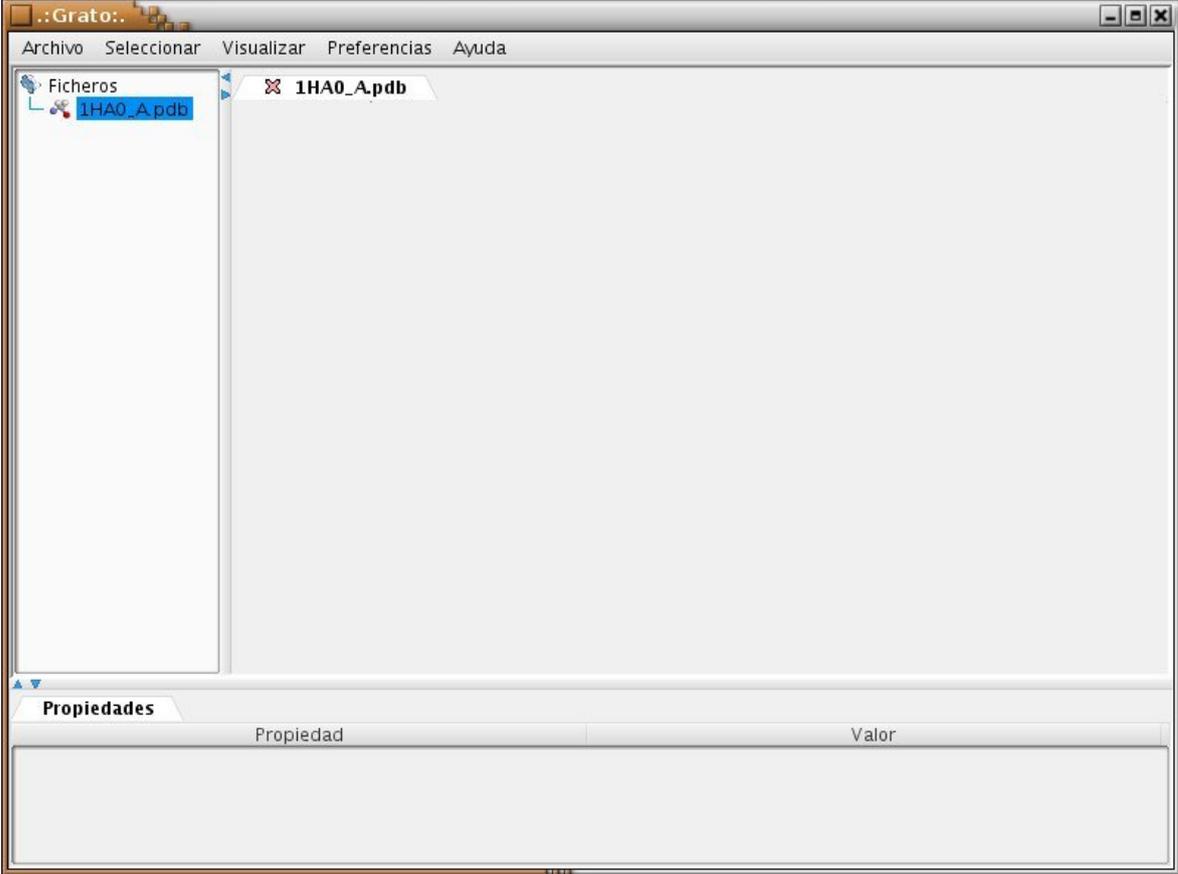
### Anexo 3: Error en la Funcionalidad Abrir Preferencias de Visualización



## Anexo 4: Error en la Funcionalidad Guardar Preferencias de Visualización



# Anexo 5: Error en la Funcionalidad Crear Representaciones



# Glosario de Términos

**Calidad:** Calidad de software. Satisfacción de las necesidades de los usuarios.

**Caso de uso:** Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas.

**Código:** Se refiere a las instrucciones contenidas en un programa, y entendibles por el ordenador. Las aplicaciones normales pueden tener miles de líneas de código que es necesario mantener y actualizar.

**Error:** Acción humana que produce una falta.

**Herramienta:** Subprograma o módulo encargado de funciones específicas y afines entre sí para realizar una tarea. Una aplicación o programa puede contar con múltiples herramientas a su disposición. Por ejemplo, el corrector ortográfico puede ser una herramienta en una aplicación para redactar documentos, pero no es una aplicación en sí misma.

**IEEE:** (Institute of Electrical and Electronics Engineers). Asociación de profesionales norteamericanos que aporta criterios de estandarización de dispositivos eléctricos y electrónicos.

**Interfaz:** Una colección de operaciones que se usan para especificar el servicio de una clase o de un componente. Circuito electrónico que gobierna la conexión entre el hombre y el hardware, y los ayuda a intercambiar información de manera confiable.

**ISO:** (International Organization for Standardization). ISO establece o fija estándares internacionales. Se ocupa de todos los campos, excepto la electricidad y la electrónica, que se rigen por la International Electrotechnical Commission (IEC).

**Procedimiento:** Dentro de una aplicación, se denomina procedimiento al conjunto de instrucciones, controles, etc. que hacen posible la resolución de una cuestión específica. La impresión es un procedimiento, como lo es la incorporación de una imagen a un texto predeterminado, etc.

**Proceso:** Un proceso es una instancia de un programa. Actualmente los sistemas multitarea soportan la ejecución de múltiples procesos, dando la apariencia de que pueden correr simultáneamente (de forma concurrente). De hecho, sólo un proceso puede estar siendo ejecutado al mismo tiempo por el CPU (excepto los CPU con múltiples procesadores). Los procesos son creados, destruidos y comunicados entre sí por el sistema operativo. En Windows se pueden ver los procesos en ejecución desde el Administrador de Tareas.

**Seguridad:** Seguridad es la cualidad de seguro, es decir, de estar libre y exentos de todo daño, peligro o riesgo.

**Software:** Término genérico que designa al conjunto de programas de distintos tipos (sistema operativo y aplicaciones diversas) que hacen posible operar con el ordenador.

**Técnica:** Conjunto de saberes prácticos o procedimientos para obtener un resultado. Requiere de destreza manual e intelectual, y generalmente con el uso de herramientas. Las técnicas se transmiten de generación en generación. La técnica nace en la imaginación y luego se llevan a la concreción, siempre de forma empírica. En cambio la tecnología surge de forma científica, reflexiva y con ayuda de las técnicas.

**Usabilidad:** La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y de ser atractivo para el usuario, en condiciones específicas de uso. La usabilidad también hacer referencia al grado de facilidad con que una aplicación, sitio Web, periférico o sistema, se adapta a sus usuarios. La usabilidad puede estar relacionada incluso a la Ergonomía, y a la potabilidad de un dispositivo.

**Usuario:** Persona que utiliza el software.

**Cliente:** Persona que utiliza con frecuencia los servicios de un profesional o empresa.