

**Universidad de las Ciencias Informáticas
Facultad No. 3
Centro de Desarrollo VERTEX**



Título: Módulo de extracción para el Laboratorio Virtual de Química

Trabajo final presentado en opción al título de Ingeniero en Ciencias Informáticas

Autores:

**Raydel Comas Rodríguez
Rydel Comas Rodríguez**

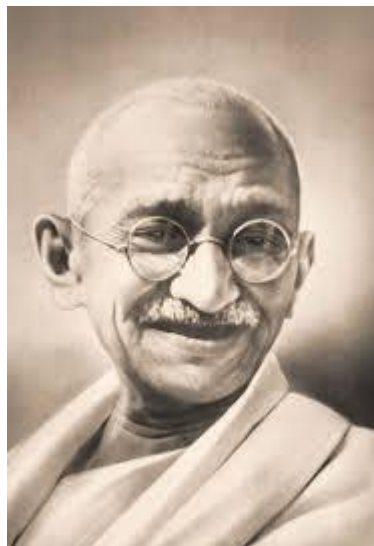
Tutores:

**Ing. Jandy Miguel Gómez
Dr. C. Ramón Carrasco Velar**

Consultante:

Lic. Rayner Batista Téllez

Ciudad de La Habana, junio de 2015



“Nuestra recompensa se encuentra en el esfuerzo, no en el resultado. Un esfuerzo total es una victoria completa”

Mahatma Gandhi

DECLARACIÓN DE AUTORÍA

Declaramos ser los únicos autores de la presente investigación y le concedemos a la Universidad de las Ciencias Informática con carácter exclusivo, los derechos patrimoniales de la misma.

Para que así conste firmo la presente a los ____ días del mes de _____ del año_____.

Raydel Comas Rodríguez
Autor

Rydel Comas Rodríguez
Autor

Ing. Jandy Miguel Gómez
Tutor

Dr. Ramón Carrasco Velar
Tutor

AGRADECIMIENTOS

Agradecemos, a todos los que de una manera u otra han tenido que ver con la realización de este trabajo de diploma. En especial a nuestros padres que han sabido guiarnos del modo correcto a lo largo de nuestras vidas y nos han apoyado en todo. A Malena y a Yoania por ser nuestras confidentes y recorrer junto a nosotros este difícil y largo camino que nos propusimos realizar. A nuestros tutores el ingeniero Jandy Miguel Gómez y el doctor en ciencias Ramón Carrasco Velar por acogernos y ofrecernos todo la ayuda y preparación necesaria para la elaboración de esta tesis. Al ingeniero Manuel Alberto Avila por haber sido el primero en confiar en nosotros y ayudarnos para que la facultad 5 nos permitiera realizar nuestro tema de tesis en el centro Vertex, al cual siempre quisimos pertenecer. También a los profesores Yaima Fiallo Valle, Zobeida Rosa Pérez y todos los que alguna una vez nos impartieron clases y fueron faro y guía en estos cinco años de carrera. Por último pero no menos importante a todos nuestros compañeros, los cuales ha sido una gran familia con la que hemos aprendido a convivir y sobrepasar los obstáculos más difíciles.

DEDICATORIA

Dedico esta tesis a todos mis familiares, en especial a mis padres, mis abuelos y a Malena mi eterna compañera.

Raydel Comas Rodríguez

Dedico esta tesis a mi familia, mis padres, mis abuelos y a mi pareja Yoania pues gracias a ellos soy, lo que he llegado a ser. Este también es el fruto de su esfuerzo y dedicación.

Rydel Comas Rodríguez

RESUMEN

En la actualidad, es imposible imaginar el desarrollo de la humanidad sin la presencia de los ordenadores. Los mismos se han introducido en el actuar diario de las personas como herramientas potentes de las cuales resulta muy difícil prescindir. Junto a este despegue de las ciencias de la computación, se ha venido tratando de hacer más real la interacción hombre - PC, con el fin de lograr la utilización de los sistemas para todo tipo de usuarios. En la presente investigación se realiza un estudio detallado de los principales conceptos relacionados con los laboratorios virtuales. Se exponen la metodología y herramientas que se utilizaron en el desarrollo del módulo de extracción para el Laboratorio Virtual de Química, se mencionan algunas de sus principales características y se realiza una valoración de la solución obtenida.

Palabras claves: laboratorios, módulo, química, virtual.

ÍNDICE

Declaración de autoría	3
AGRADECIMIENTOS	4
DEDICATORIA	5
RESUMEN	6
ÍNDICE	7
INTRODUCCIÓN	8
CAPÍTULO 1. HACIA UN ANÁLISIS TEÓRICO DE LOS LABORATORIOS VIRTUALES	11
1.1 Introducción	11
1.2 Definiciones y conceptos	11
1.3 Antecedentes y evolución de los laboratorios virtuales	13
1.4 La extracción en la química	18
1.4.1 Tipos de extracción	19
1.5 Estudio de metodologías, tecnologías y herramientas	22
1.5.1 Metodología de desarrollo.....	22
1.5.2 Herramientas de desarrollo.....	24
1.5.3 Herramientas de diseño gráfico	28
1.5.4 Herramientas CASE	29
1.5.5 Lenguajes de programación.....	30
1.6 Conclusiones parciales.....	30
CAPÍTULO 2. PROPUESTA DE SOLUCIÓN	32
2.1 Introducción	32
2.2 Descripción de la solución propuesta	32
2.3 Fase de Planificación.....	33
2.3.1 Requisitos no funcionales	33
2.3.2 Historias de Usuario (HU)	33
2.3.3 Plan de Entregas.....	35
2.3.4 Plan de Iteraciones	36
2.4 Fase de Diseño.....	36
2.4.1 Diagrama de clases	36
2.4.2 Tarjetas CRC del sistema.....	37
2.4.3 Arquitectura del sistema.....	39
2.4.4 Patrones del Diseño.....	40
2.5 Conclusiones del capítulo.....	42
CAPÍTULO 3. IMPLEMENTACIÓN, PRUEBAS Y VALIDACIÓN DEL SISTEMA.....	43
3.1 Fase de Desarrollo	43
3.1.1 Estándares de codificación	43
3.1.2 Tareas de implementación	44
3.1.3 Componentes desarrollados en la Implementación	45
3.2 Pruebas de Aceptación.....	47
3.3 Validación del sistema	50
3.4 Conclusiones	51
BIBLIOGRAFÍA	52
ANEXOS	55

INTRODUCCIÓN

La evolución de las ciencias y las tecnologías han contribuido en gran medida al desarrollo de la sociedad actual. Con el surgimiento de los ordenadores o computadoras, se crearon nuevos sistemas capaces de realizar tareas que antes eran muy difíciles o complejas de lograr. El avance y desarrollo en la creación de equipos y el estudio e investigación de las técnicas de gráficos, dieron surgimiento a la realidad virtual (RV). En la actualidad, la misma se encuentra en muchos sistemas que permiten que el usuario experimente en diferentes escenarios virtuales. Los recursos materiales son reemplazados por herramientas computacionales que facilitan el trabajo, de ahí la importancia de su desarrollo.

En el campo de la educación, estas se convierten en un recurso didáctico que permite incluso reproducir laboratorios tradicionales en entornos virtuales los cuales pueden ser usados por los profesores para motivar y atraer la atención de los estudiantes en el proceso de enseñanza-aprendizaje. El uso de los gráficos tridimensionales y el alto grado de interactividad que se obtiene, permiten al estudiante incorporar el conocimiento de forma visual. Cada vez es mayor el número de centros de enseñanza en los que se utilizan aplicaciones con entornos virtuales y ejemplo de ello lo constituye el uso de laboratorios virtuales (LV).

Los laboratorios virtuales apoyan de hecho la gestión de asignaturas técnicas y contribuyen a desarrollar habilidades y destrezas prácticas a través de interacciones, comparaciones, demostraciones e interpretaciones en la computadora. Es importante destacar que el desarrollo de este tipo de herramientas no persigue sustituir, ni las experiencias prácticas, ni la presencia de un profesor en el aula, pero sí resulta una herramienta de auto preparación por parte de los estudiantes con el objetivo de vencer la asignatura a la que esté asociada dicha herramienta, y en ese proceso de enseñanza-aprendizaje adquirir y desarrollar habilidades que después puedan llevar a la práctica.

Actualmente, en la Universidad de las Ciencias Informáticas existe un centro llamado Centro I+D+i Entornos Virtuales Interactivos 3D (VERTEX) con la misión de desarrollar productos y servicios informáticos asociados a los Entornos Virtuales Interactivos 3D, con un alto valor agregado resultado de un ciclo completo de Investigación+Desarrollo+innovación (I+D+i). El mismo ha incursionado desde su surgimiento en las tecnologías relacionadas con los LV obteniendo varios productos de gran aceptación que hoy día son utilizados por la Universidad y por otros Centros educativos.

Algunos ejemplos de estos productos son los LV para:

- Electrónica Analógica

INTRODUCCIÓN

- Instrumentación
- Instrumentación Médica
- Higiene y Seguridad en los Procesos
- Ergonomía y Antropometría
- Medición y análisis de vibraciones
- Circuitos Eléctricos
- Ciencia de Materiales
- Transporte Ferroviario

La química es una ciencia experimental muy costosa. Una situación compleja existente en los laboratorios docentes de las instituciones académicas visitadas, mostró que los laboratorios docentes son antiguos y no cuentan con los recursos necesarios, a lo que se le suma el deterioro por el paso de los años y el intenso uso. Debido a esto, muchos de ellos no prestan todos los servicios necesarios por los cuales fueron creados. Por esto se hace imprescindible el cuidado de la base material que resta y la búsqueda de nuevas soluciones que brinden a los estudiantes la posibilidad de aproximarse al desarrollo de habilidades y cumplir con los objetivos instructivos propios de la disciplina. En la práctica química, se emplean múltiples métodos de extracción inherentes a procesos particulares entre los que se encuentran los métodos de extracción continua líquido-líquido para líquidos más densos y menos densos que el agua, la extracción sólido-líquido por medio de Soxhlet y el equipo de Dean-Stark de extracción de agua. Aunque son de mucha utilidad, no tienen un uso muy amplio en la etapa pre-profesional y por lo tanto, no suelen ser familiares a los estudiantes que deben realizar la práctica correspondiente, pero requieren el estudio tanto de la teoría como de la práctica para un mejor aprendizaje por parte de los mismos. Por todo lo anteriormente descrito, el uso de laboratorios virtuales puede resultar una solución alternativa para aliviar la situación por la que transitan, tanto las instituciones como el proceso académico.

Por todo lo anteriormente descrito, el problema a resolver en el presente trabajo es: **¿Cómo apoyar desde la informática, la enseñanza práctica de los métodos de extracción en química para suplir las distintas necesidades que existen en este aspecto?**

Para lo cual se definen como objeto de estudio **los laboratorios virtuales de química** y como campo de acción **el aprendizaje virtual de métodos de extracción.**

INTRODUCCIÓN

Este trabajo va dirigido especialmente a recrear diferentes procesos de extracción en el laboratorio virtual. De esta forma el objetivo del mismo es: **desarrollar el módulo de extracción para el Laboratorio Virtual de Química.**

Para el cumplimiento del objetivo de esta investigación se proponen los siguientes objetivos específicos:

- Diseñar el marco teórico de la investigación para el estudio de las principales tendencias y conceptos relacionados con los LV.
- Definir los artefactos ingenieriles para apoyar el proceso de desarrollo de la solución.
- Implementar las actividades prácticas y teóricas de extracción que formarán parte de la solución para apoyar el proceso de enseñanza y aprendizaje de esta ciencia.
- Validar y evaluar técnicamente la solución para valorar la satisfacción del cliente.

Para llevar a cabo la investigación se utilizarán los siguientes métodos:

Métodos teóricos

Histórico - Lógico: se empleó para la fundamentación y sistematización de los aspectos teóricos contemplados en el desarrollo de la investigación acerca de los laboratorios virtuales de química, las metodologías a utilizar, los planes de estudios y demás elementos relacionados con el contenido del trabajo.

Analítico - Sintético: durante el proceso de investigación permitió descubrir los elementos esenciales, concepciones y conceptos en torno al objeto de investigación.

Métodos empíricos

Observación: se empleó como método referencial al observar los distintos tipos de Laboratorios Virtuales, que sirvieron como objeto de análisis y comparación para establecer las características y elementos fundamentales que debía cumplir la propuesta que plantea el autor.

Entrevista no estructurada: este método permitió establecer una comunicación con las personas de utilidad para la investigación y la recopilación de información especializada que se tendrá en cuenta para el desarrollo propio de la solución.

CAPÍTULO 1. HACIA UN ANÁLISIS TEÓRICO DE LOS LABORATORIOS VIRTUALES

1.1 Introducción

En el presente capítulo se realiza un estudio detallado de los principales conceptos relacionados con los Laboratorios Virtuales. Se analizan sus ventajas y desventajas, se caracterizan sus antecedentes históricos y la evolución de los mismos. También se exponen las principales metodologías y herramientas que se utilizan en la actualidad en el desarrollo de los Laboratorios Virtuales y se realiza una valoración y selección de las mismas.

1.2 Definiciones y conceptos

Ambiente de aprendizaje interactivo

Un ambiente de aprendizaje interactivo es un espacio electrónico, donde se simulan situaciones problemáticas en 2D o 3D de un diseño de aprendizaje en las que deben cumplirse objetivos instructivos y educativos de un programa de estudio. Estudiantes y profesores colaboran en escenarios simulados y cumplen tareas de entrenamiento o experimentales, asumiendo un rol propio de su profesión y donde son evaluados por ello.

Actualmente existen aplicaciones multimedia, elaboradas con tecnologías informáticas que pueden considerarse como ambientes de aprendizajes interactivos. Para este caso, solo se considerarán aquellas aplicaciones informáticas que asumen estrategias y experiencias provenientes de la industria de los videojuegos y de la realidad virtual. Como ejemplos de estas aplicaciones, se pueden mencionar: simuladores de vuelos, simuladores de conducción, entrenadores médicos, programas de visualización científica, instrumentación virtual, los juegos serios y laboratorios virtuales, donde la principal forma de interacción ocurre en escenarios con objetos simulados en 2D o 3D.

Realidad virtual

Todo este desarrollo en el gráfico 3D, ha traído consigo que se comiencen a desarrollar aplicaciones, cuyo objetivo sea el de simular, en un entorno generado por computadoras, un mundo virtual con cierto grado de realismo. Estas aplicaciones se denominan aplicaciones de *realidad virtual* (RV) (2). A pesar de que en la literatura se pueden encontrar muchas definiciones de *realidad virtual* (en inglés: *virtual reality*), seguramente una de las más completas es la que propuso A. Rowell: “La *realidad virtual* es una simulación interactiva por computador desde el punto de vista del participante, en la cual se sustituye o se aumenta la información sensorial que recibe” (1).

Uno de los elementos básicos que caracterizan a un sistema de RV, es la simulación interactiva. La misma consiste en una simulación de la realidad, la cual se recrea en un mundo virtual, que solo existe en la memoria del ordenador. Y que sea interactiva, es lo que la distingue de una animación, principalmente porque en una animación como por ejemplo una película, los espectadores son entes pasivos, porque no pueden alterar el contenido de las imágenes que se proyectan en la pantalla. Esto no se puede hacer, porque las mismas han sido grabadas desde una posición, siguiendo una trayectoria que será inalterable. No sucede así en un sistema de RV, el cual permite al usuario además de interactuar con el medio y sus objetos, moverse por el mundo de manera dinámica, sin movimientos predefinidos.

La realidad virtual utiliza la interacción implícita en contraposición a la interacción clásica. Esta última es cuando un usuario quiere realizar una determinada operación como por ejemplo, moverse de una habitación a otra, en la que es necesario que se comunique de forma explícita con el sistema. Para la realización del procedimiento de interacción, el usuario utiliza un esquema de comunicación determinado por la interfaz de la aplicación. Ya sea una basada en comandos o una basada en GUI (*interfaz gráfica de usuario*). En cualquiera de los dos casos, el usuario debe conocer o recordar cuál sería el botón o comando asociado a esa determinada acción que quiere realizar. Y esta interacción se envía por los dispositivos de interacción clásicos como mouse y teclados.

A pesar de que las interfaces gráficas en los últimos años han mejorado. No dejan de suponer cierto esfuerzo por parte del usuario, que requiere de cierto tiempo de entrenamiento y familiarización con el sistema. En cambio, la realidad virtual con interacción a través de métodos implícitos, lo que busca es que la comunicación entre el usuario y el sistema, se lleve a cabo por métodos que impliquen la utilización de movimientos naturales. Un ejemplo es el control de la cámara virtual. En un sistema de realidad virtual, con este tipo de interacción, el sistema actualizaría la posición de la misma, en función de los movimientos de la cabeza del usuario. Si el usuario quiere ver la parte del mundo virtual que tiene detrás, no tendría que utilizar ningún comando ni pulsar una tecla solamente tendría que realizar el mismo gesto que haría en el mundo real.

Independientemente del tipo de interacción que se utilice, ya sea implícita o explícita, la presencia de dispositivos periféricos para la entrada de datos en cualquier sistema de RV siempre va a existir. Pero la diferencia fundamental es la percepción que tiene el usuario de estos dispositivos.

Laboratorios Virtuales

Dentro las definiciones sobre Laboratorios Virtuales se destacan algunas:

“Son simulaciones de prácticas manipulativas que pueden ser hechas por la/el estudiante lejos de la universidad y el docente” (3).

En el Informe de la reunión de expertos (4) sobre laboratorios virtuales organizada por el Instituto Internacional de Física Teórica y Aplicada (IITAP) Ames, Iowa 10-12 de mayo de 1999 se definió que Laboratorio virtual es:

“Un espacio electrónico de trabajo concebido para la colaboración y la experimentación a distancia con objeto de investigar o realizar otras actividades creativas, y elaborar y difundir resultados mediante tecnologías difundidas de información y comunicación”.

“Las prácticas de laboratorio pueden desarrollarse de manera que el alumno esté en contacto físico y pueda manipular los elementos, dispositivos e instrumental requerido para el experimento (laboratorio real) o utilizando simulaciones interactivas programadas con el empleo de las PC (laboratorio virtual). Ambas formas requieren la auto preparación por parte de los estudiantes, a través de materiales impresos (textos o folletos), o en formato electrónico.” (5)

A partir de las definiciones anteriores, se puede plantear como definición que un laboratorio virtual o práctica virtual, es un sistema informático que recrea al laboratorio técnico o real, creado para facilitar la educación, donde el usuario tiene la posibilidad de desarrollar la práctica fácilmente, bajo ningún costo y sin correr riesgos, con una visión muy realista.

1.3 Antecedentes y evolución de los laboratorios virtuales

Primeros laboratorios virtuales

Según la tesis doctoral “*Modelo de referencia de laboratorios virtuales y Aplicaciones a sistemas de Tele-educación*” (8) que recoge gran parte de la revisión histórica relativa a los laboratorios virtuales y sus características principales, la primera aproximación a los laboratorios virtuales aparece en el año 1984, donde surge el concepto de instrumento virtual (9) y sus características se determinaron de acuerdo a los fundamentos de programación.

En 1992 aparece el término **Laboratorio Virtual** (10), para describir la programación orientada a objetos en el desarrollo de un laboratorio de simulación.

En 1994 se presenta un estudio realizado por la Universidad de Vanderbilt en E.E.U.U. en el que se desarrolla un laboratorio virtual basado en simulación como apoyo a las prácticas tradicionales (11) en el que se concluyó que existe la necesidad de esta herramienta para aprender las habilidades básicas y el manejo de los equipos, con lo cual se optimiza tanto el tiempo de los alumnos como el del personal de laboratorio. Ese mismo año aparece un artículo (12) en el que se define explícitamente un LV como un programa de simulación.

Ya en la conferencia IMTC (IEEE Instrumentation and Measurement Technology Conference) celebrada en junio de 1996, empiezan a presentarse distintos aspectos de lo que es un LV. Desde entonces las referencias de los laboratorios se han incrementado vertiginosamente.

Investigadores de la Universidad de Illinois en 1997 presentan un completo laboratorio de instrumentación electrónica puesto a disposición de los usuarios a través de Internet. Este es el primer LV con control remoto de instrumentación electrónica en funcionamiento (13).

En el 2000 se van sucediendo artículos en conferencias y revistas donde se describen distintos laboratorios virtuales y donde se va entrando en detalle en los distintos métodos a utilizar en el desarrollo de laboratorios virtuales y se comentan posibles soluciones que mejoren o aumenten el rendimiento de éstos. La idea de utilizar la simulación como paso previo al uso de los instrumentos permitirá que se reduzca el tiempo necesario de uso del instrumento real y por tanto, del recurso más costoso.

Laboratorios Virtuales en la actualidad

En la actualidad existen muchos laboratorios virtuales relacionados con la química. Como ejemplo de estos pueden mencionarse:

Crocodile Chemistry: es un Laboratorio Virtual de Química en el que se pueden simular experimentos, representar resultados en gráficos y observar reacciones en 3D. Cuenta con una librería de objetos que permite seleccionar una serie de recipientes como: matraces, probetas y pipetas entre otros. Permite además modificar parámetros de los diferentes componentes, como por ejemplo el tamaño de las partículas, la concentración o la tasa de flujo de un gas etc. (14) También se pueden realizar experimentos con (14) ácidos y bases, con metales, mezclas y reacciones, con la tabla periódica, con compuestos no metálicos y experimentos de electroquímica y químico-físicos.

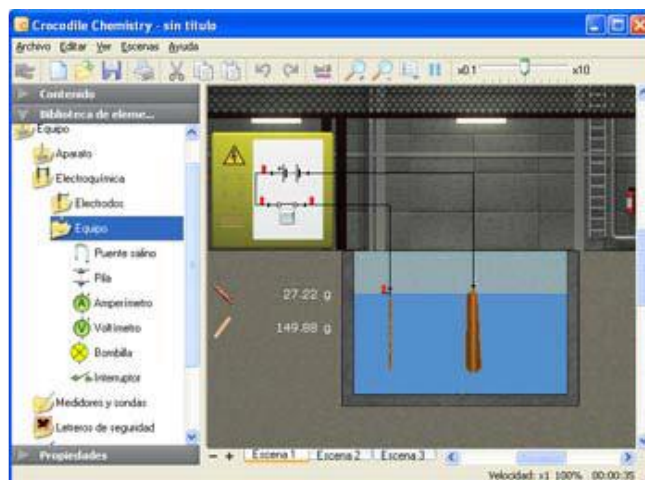


Figura 1. Crocodile Chemistry (14)

ChemLab: el programa Model ChemLab v2.0 para Windows y Mac OS (Model Science) es la simulación interactiva de un laboratorio de química. En él se usan el equipamiento y los procedimientos comunes de laboratorio para simular los pasos involucrados en la realización de los experimentos. La simulación de cada práctica de laboratorio se halla en un módulo separado y constituye una extensión del programa ChemLab, por lo que es posible la realización de muchas y diferentes prácticas, usando una interfase común de laboratorio (15).

El ChemLab consta de dos ventanas, una de texto y otra de laboratorio, divididas por una barra. La ventana de texto sirve para la documentación textual en el ChemLab y está dividida a su vez en 3 ventanas: Introducción, Procedimiento y Observaciones. La primera sólo sirve para leer la introducción a la práctica, la segunda permite leer el procedimiento que debe seguir el usuario para realizar el experimento en la ventana de laboratorio, y la tercera está diseñada para que el usuario anote las observaciones que se le indican en la ventana del procedimiento (15).

La ventana de laboratorio permite visualizar la simulación animada del laboratorio. En ella se agregan los utensilios o equipos a utilizar. Estos objetos y las sustancias que se emplean pueden agregarse y utilizarse usando los comandos del menú principal, la barra de herramientas (Chem toolbar) o el menú contextual del botón derecho del ratón (15).

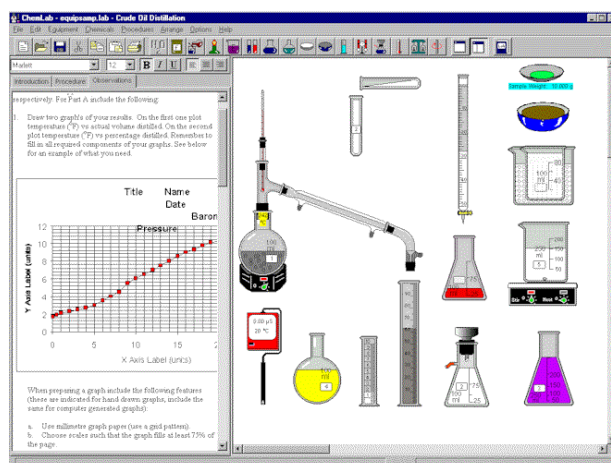


Figura 2. ChemLab (15)

VLabQ: es un simulador interactivo de prácticas de laboratorio de Química. Utiliza equipos y procedimientos estándares para simular los procesos que intervienen en un experimento o práctica (16).

Cada simulación o práctica se guarda en un archivo que contiene todos los reactivos y condiciones que se usarán durante el experimento. Se puede guardar en cualquier momento todo el contenido del laboratorio, tanto el equipo como su contenido y las condiciones para continuar con la práctica posteriormente. Una vez cargada una práctica, el simulador muestra diferentes textos que sirven como guía para realizarla (16).

Contiene instrumentos de laboratorio, tales como (16): vasos de precipitados, matraces Erlenmeyer, embudo Büchner, matraz de balón, reactor, buretas, probetas, pipetas y tubo de ensayo entre otros. Además brinda:

Equipo de medición: pHmetros, termómetros, conductímetros y balanzas.

Equipo térmico: mechero, parrilla y baño de hielo. Así como agitador de vidrio, vidrio de reloj, cápsula de porcelana y calorímetro.

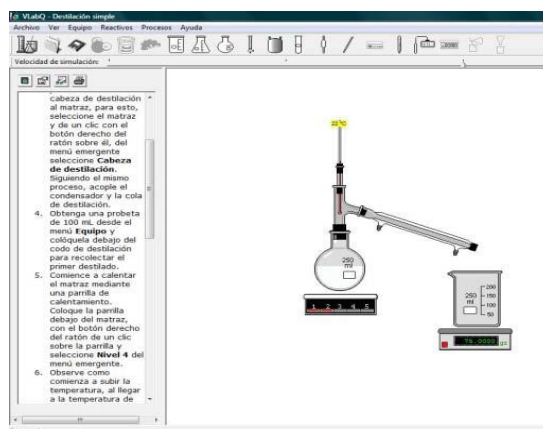


Figura 3. VLabQ (16)

QuimiLab: Es un laboratorio virtual diseñado para acompañar al curso de Química del nivel secundario y universitario. Este tiene una interfaz basada en arrastrar-y-soltar los elementos y ventanas móviles que simulan productos químicos, vidriería e instrumentos. Brinda una serie de equipamiento agrupados por tipo (17):

Vidriería: vasos de precipitados, matraces, tubos de ensayo, buretas, agua de grifo y un calorímetro.

Instrumentación: balanza, plato caliente, medidores de temperatura, presión, de pH y espectrofotómetro.

Productos químicos: está disponible una base de datos de compuestos químicos. Cada compuesto presenta sus propiedades para todas las fases incluyendo densidades, puntos de ebullición y de fusión y capacidades calóricas.

Reacciones: la base de datos de reacciones químicas contiene proporciones estequiométricas de los reaccionantes junto con las condiciones experimentales requeridas, datos cinéticos y constantes de equilibrio.



Figura 4. QuimiLab (17)

En la Tabla 1 se presenta un resumen de diferentes características de los cuatro laboratorios virtuales analizados, aunque se debe hacer énfasis en que todos son privados, no poseen evaluador teórico del contenido y no incluyen prácticas de métodos de extracción entre sus prestaciones.

Además, todos estos laboratorios están desarrollados en 2D lo que quita realismo a las prácticas, pues los usuarios no experimentan la sensación de estar en el interior de un laboratorio. Estos laboratorios tienen un elevado costo en el mercado y requieren de ordenadores con prestaciones que son superiores a las que ofrecen los ordenadores con que contamos en nuestras escuelas.

No es factible acometer la realización de un laboratorio virtual de química en sus múltiples

actividades en el plazo de tiempo que corresponde a un diploma. Además, como la familiarización de los estudiantes con los métodos y equipamiento para las técnicas de extracción es pobre, a los diferentes niveles de la enseñanza de la disciplina, resulta aconsejable la creación del módulo de extracción para apoyar ese aspecto del proceso de enseñanza-aprendizaje de la *química* práctica.

Aspectos	<i>Crocodile Chemistry</i>	<i>ChemLab</i>	<i>VLabQ</i>	<i>QuimiLab</i>
Privativo	SI	SI	SI	SI
Presenta actividades relacionadas con la extracción	NO	SI	NO	NO
Evaluador teórico	NO	NO	NO	NO
Desarrollo en 3D	NO	NO	NO	NO
Requerimientos de Hardware	ALTO	ALTO	ALTO	ALTO
Costo del software	ALTO	ALTO	ALTO	ALTO

Tabla 1. Comparación entre LV de Química analizados.

1.4 La extracción en la química

La química es una disciplina básica que forma parte del diseño curricular de las enseñanzas secundaria, tecnológica y preuniversitaria, así como de un gran número de carreras universitarias. Uno de los contenidos más importantes es el correspondiente a la separación de sustancias y en particular la extracción.

La separación de un compuesto por extracción se basa en la transferencia selectiva del compuesto desde una mezcla sólida o líquida con otros compuestos hacia una fase líquida (normalmente un disolvente orgánico). El éxito de la técnica depende básicamente de la diferencia de solubilidad en el disolvente de extracción entre el compuesto deseado y los otros compuestos presentes en la mezcla inicial. El principal objetivo de la extracción es separar selectivamente el producto de una reacción, o bien eliminar las impurezas que lo acompañan en la mezcla de reacción, gracias a sus diferencias de solubilidad en el disolvente de extracción elegido.

1.4.1 Tipos de extracción

Extracción por el método de Soxhlet

Es un método de extracción continuo que se utiliza para recuperar componentes de interés a partir de materiales sólidos. Consiste en colocar el material a extraer, previamente molido y pesado, en un cartucho de celulosa (dedal) que funciona como filtro. Ese cartucho con el material sólido se introduce en la cámara de extracción, conectada por la parte inferior a un balón que contiene el líquido para la extracción y un refrigerante en la parte superior. El disolvente contenido en el balón se calienta a ebullición, el vapor asciende por el tubo lateral, se condensa en el refrigerante y cae caliente sobre el material. Cuando el líquido condensado alcanza el nivel del sifón, regresa al balón enfriando el disolvente dentro del balón. El proceso se repite hasta conseguir el agotamiento deseado del material.

En la figura 5 se muestra el equipamiento necesario para llevar a cabo este tipo de método en el laboratorio.



Figura 5. Equipo de extracción Soxhlet

Extracción por el método Dean Stark

El aparato Dean-Stark o receptor Dean-Stark (Fig. 6) se utiliza en la química sintética para recoger el agua [1] (u ocasionalmente otro líquido) de un reactor. Se utiliza en combinación con un condensador y un reactor discontinuo para la eliminación continua del agua que se produce durante una reacción química que se realiza a temperatura de reflujo. Fue inventado en 1920 para la determinación del contenido de agua en el petróleo.

Durante la reacción que ocurre en el matraz o balón de reacción, los vapores que emanan el disolvente de reacción y el componente a ser removido viajan matraz arriba hacia el condensador

[5], y luego gotea después de la condensación en la trampa de destilación [9]. Aquí, los líquidos inmiscibles se separan en capas. Cuando la parte superior (menos denso) alcanza el nivel de la parte del brazo que puede fluir de vuelta al reactor retorna a este, mientras que la capa inferior permanece en la trampa. La trampa logra su plena capacidad cuando el nivel inferior alcanza el nivel de la parte del brazo, por lo que desde este punto, la capa inferior comienza a fluir de nuevo hacia el reactor también. Es por lo tanto importante drenar la capa inferior del aparato de Dean-Stark tantas veces como sea necesario.

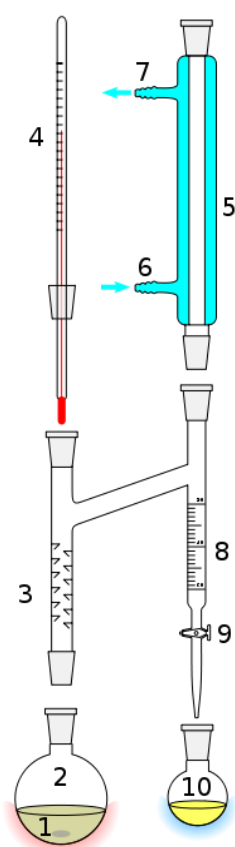


Figura 6. Equipo de extracción Dean-Stark

Extracción líquido-líquido con disolvente más denso o menos denso que el agua

La extracción líquido-líquido es un método muy útil para separar componentes de una mezcla líquida. El éxito de este método depende de la diferencia de solubilidad del compuesto a extraer en dos disolventes diferentes. Cuando se agita un compuesto con dos disolventes inmiscibles, el compuesto se distribuye entre los dos disolventes. A una temperatura determinada, la relación de concentraciones del compuesto en cada disolvente es siempre constante, y esta constante es lo que se denomina coeficiente de distribución o de reparto ($K = \text{concentración en disolvente}_2 / \text{concentración en disolvente}_1$).

Es frecuente obtener mezclas de reacción en disolución o suspensión acuosa (bien porque la reacción se haya llevado a cabo en medio acuoso o bien porque durante el final de reacción se haya añadido una disolución acuosa sobre la mezcla de reacción inicial). En estas situaciones, la extracción del producto de reacción deseado a partir de esta mezcla acuosa se puede conseguir añadiendo un disolvente orgánico adecuado, más o menos denso que el agua, que sea inmisible con ella y capaz de solubilizar la máxima cantidad de producto a extraer pero no las impurezas que lo acompañan en la mezcla de reacción, o viceversa. Después de agitar la mezcla de las dos fases para aumentar la superficie de contacto entre ellas y permitir un equilibrio más rápido del producto a extraer entre las dos fases, se producirá una transferencia del producto deseado desde la fase acuosa inicial hacia la fase orgánica, en una cantidad tanto mayor cuanto mayor sea su coeficiente de reparto entre el disolvente orgánico de extracción elegido y el agua. Unos minutos después de la agitación, las dos fases se separan de nuevo, espontáneamente por decantación, debido a la diferencia de densidades entre ellas, con lo que la fase orgánica que contiene el producto deseado se podrá separar mediante una simple decantación de la fase acuosa conteniendo impurezas. La posición relativa de ambas fases depende de la relación de densidades. Dado que después de esta extracción, la fase acuosa frecuentemente aún contiene cierta cantidad del producto deseado, se suele repetir el proceso de extracción un par de veces más con disolvente orgánico puro.

Una vez finalizada la operación de extracción, se tiene que recuperar el producto extraído a partir de las fases orgánicas reunidas. Para ello, se tiene que secar la fase orgánica resultante con un agente desecante, filtrar la suspensión resultante y finalmente eliminar el disolvente orgánico de la disolución seca conteniendo el producto extraído por destilación o evaporación.



Figura 7. Equipo para extracción líquido-líquido más denso



Figura 8. Equipo para extracción líquido-líquido menos denso

1.5 Estudio de metodologías, tecnologías y herramientas

1.5.1 Metodología de desarrollo

Una metodología de desarrollo de software es un conjunto de procedimientos, métodos y técnicas que ayudan a organizar y dirigir el trabajo con el objetivo de facilitar el desarrollo de un producto. Es una guía de pasos a seguir, que indica qué persona realiza qué actividad en el desarrollo del software. Además indican la información que se debe obtener después de terminada dicha actividad (18).

Existen varias propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo:

Las tradicionales se centran principalmente en el control de los procesos, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir en cada etapa, y las herramientas y notaciones que se usarán.

Las metodologías ágiles dan mayor valor a la persona, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas.

Metodologías Ágiles	Metodologías Tradicionales
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Proceso menos controlado, con pocos	Proceso mucho más controlado, con

principios.	numerosas políticas y normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

Tabla 2. Diferencias entre metodologías ágiles y no ágiles (19).

Debido a que actualmente el enfoque ágil está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se necesita reducir drásticamente el tiempo de desarrollo pero manteniendo la alta calidad (20), se estudiaron tres de las metodologías pertenecientes a este enfoque, las cuales han probado ser muy efectivas en la creación de los LV actuales. Al mismo tiempo son las más utilizadas por el Centro de desarrollo VERTEX de la Universidad al que está asociada la presente investigación (32).

SCRUM - Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle.

Es una metodología ágil que surge como modelo para el desarrollo de productos tecnológicos, se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad. Esta metodología requiere trabajo duro puesto que no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias evolutivas del proyecto, en ocasiones es necesario complementarlo con otras metodologías (21).

Como metodología ágil:

1. Es un modo de desarrollo de carácter adaptable más que predictivo.
2. Orientado a las personas más que a los procesos.
3. Emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones.

Además, emplea las siguientes prácticas de la gestión ágil:

1. Revisión de las iteraciones
2. Desarrollo incremental
3. Desarrollo evolutivo
4. Auto-organización

5. Colaboración

Crystal Methodologies es un conjunto de metodologías para el desarrollo de software centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. El desarrollo de software se considera un juego operativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas (19).

Extreme Programming XP es una metodología ágil centrada en potenciar las relaciones interpersonales, promoviendo el trabajo en equipo y preocupándose por el aprendizaje de los desarrolladores. Se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

Esta metodología intenta reducir la complejidad del software (SW) por medio de un trabajo orientado directamente al objetivo. Además, presenta un diseño evolutivo que hace que se le de poca importancia al análisis como fase independiente, puesto que se trabaja exclusivamente en función de las necesidades del momento (19).

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc (19).

Después de un estudio de las principales tendencias en metodologías ágiles de desarrollo se decide utilizar la metodología XP para la realización de las prácticas de extracción del Laboratorio Virtual de Química, ya que muchas de sus características son aplicables al contexto de realización del proyecto. Principalmente, se cuenta con un equipo de desarrollo pequeño (en este caso, dos personas), frecuentemente los integrantes del proyecto deben reunirse con el cliente para determinar el estado de la solución, debido a que los requisitos surgidos una vez empezado el proyecto pueden sufrir cambios. Además no es necesario generar muchos artefactos que se generan utilizando otro tipo de metodología porque el negocio no requiere de grandes volúmenes de información y el tiempo de desarrollo es relativamente corto.

1.5.2 Herramientas de desarrollo

Analizando las tendencias actuales en la elaboración de materiales didácticos, entornos virtuales y

LV en 3D, destaca la presencia de dos herramientas muy utilizadas por el Centro VERTEX, las cuales presentan facilidad de licencia y han demostrado ser efectivas en la creación de LV.

UDK

Unreal Engine es un motor de juego de PC y consolas creados por la compañía Epic Games. Implementado inicialmente en el shooter en primera persona Unreal en 1998, siendo la base de juegos como Unreal Tournament, Deus Ex, Turok, Tom Clancy's Rainbow Six: Vegas, America's Army, Red Steel, Gears of War, BioShock, BioShock 2, BioShock Infinite, Star Wars Republic Commando, Batman: Arkham Asylum o Mass Effect. También se ha utilizado en otros géneros como el rol y juegos de perspectiva en tercera persona. Unreal Engine también ofrece varias herramientas adicionales de gran ayuda para diseñadores y artistas. Actualmente cuenta con cuatro versiones las cuales fueron mejorando y añadiendo nuevas características necesarias para mejorar la experiencia de los usuarios y los videojuegos (24).

UDK 1

- Renderizado, detección de colisiones, IA, visibilidad, opciones para redes y manipulación de archivos de sistema.

UDK 2

- Reescritura completa del código del núcleo y del motor de renderizado, además de integrar el nuevo UnrealEd 3. También incluyó el SDK de Karma physics.
- Soporte para la PlayStation 2, GameCube y Xbox.

UDK 3

- HDRR, normal mapping, y sombras dinámicas.
- Renderizado para mayor número de objetos simultáneos, físicas más realistas para efectos de agua, físicas de texturas corporales, mayor destructibilidad para los entornos, IA mejorada y efectos mejorados en luces y sombras con rutinas avanzadas para los shaders.

UDK 4

- Características y soporte para la séptima generación de consolas de videojuegos.

La última versión de este motor es el Unreal Engine 4, está diseñado para la tecnología:

DirectX 9 (para las plataformas Windows XP/Vista/7 de 32/64-bit y Xbox 360)

DirectX 10 (para plataformas Windows Vista/7 32/64-bit)

OpenGL (para plataformas Linux, Mac OS X de 32/64-bit y PlayStation 3).

Unity 3D

Unity 3D para PC y Mac viene empaquetado como una herramienta para crear juegos,

aplicaciones interactivas, visualizaciones y animaciones en 3D y tiempo real.

Unity puede publicar contenido para múltiples plataformas como PC, Mac, Nintendo Wii y iPhone. También puede publicar juegos basados en web usando el plugin Unity web player. Como base técnica, esta es posiblemente la mejor por debajo de los 100.000€.

El editor de Unity es el centro de la línea de producción, ofreciendo un completo editor visual para crear juegos. El contenido del juego es construido desde el editor y el gameplay se programa usando un lenguaje de scripts. Esto significa que los desarrolladores no necesitan ser unos expertos en C++ para crear juegos con Unity, ya que las mecánicas de juego son compiladas usando una versión de JavaScript, C# o Boo, un dialecto de Python.

Los juegos creados en Unity son estructurados en escenas como el motor Gamebryo. En Unity una escena puede ser cualquier parte del juego, desde el menú de inicio como un nivel o área de tu juego; la elección es tuya ya que una escena es un lienzo en blanco sobre el que dibujar cada parte del juego usando las herramientas de Unity.

Esta herramienta también incluye un editor de terrenos, desde donde puedes crear un terreno (como una hoja en blanco), sobre la que los artistas podrán esculpir la geometría del terreno usando herramientas visuales, pintar o texturizar, cubrir de hierba o colocar árboles y otros elementos de terreno importados desde aplicaciones 3D como 3DS Max o Maya (23).

Motores gráficos	Unity 3D	UDK
Plataformas	Android iOs Mac Windows Web XBOX LA Wii PS Network	PC (Microsoft Windows, GNU/Linux) Apple Macintosh (Mac OS, Mac OS X) Dreamcast, Gamecube, Wii, Xbox, Xbox 360, PlayStation 2, PlayStation 3, Xbox One, PlayStation 4)
	Dólares Indie, Gratuita Unity Pro, \$1500 Android Básica, \$400 iOs Básica, \$400	Euros Shiva PLE, Gratuita (solo para aprendizaje) Shiva Editor Básic, \$169 Shiva Editor Advanced,

Licencias	Android Pro, \$1500, requiere Unity Pro iOS Pro, \$1500, requiere Unity Pro	\$1499
Scripting	C# Javascript Boo	Está escrito en Unreal Script (Lenguaje propio similar Java o C# modificado).
Extensiones o plugins	C# C++	Lua C++
Importación de Assets	Modelos 3D animados (formatos nativos). Blender 3DStudio Max Maya Cinema 4D Texturas PNG JPG TGA BMP Sonidos OGG MP3 WAV Videos AVI MPG MP4 OGG	Modelos 3D animados • Collada DAE Texturas • PNG • JPG • TGA • BMP Sonidos • OGG • MP3 • WAV Videos • AVI • MPG

Tabla 3. Comparación entre Unity 3D y UDK.

Después de realizada esta comparación se decide usar Unity 3D como herramienta de desarrollo porque este hace que el proceso de creación de la aplicación sea más ágil, dándole al desarrollador un set de pasos lógicos para construir cualquier panorama concebible y proporcionándole además las herramientas necesarias para explotar todas sus funcionalidades. Por otro parte establece el uso del concepto Game Object (objetos activos de la escena), que

permite crear los componentes de la aplicación y tratarlos cada uno como elementos aparte, con propiedades y características distintas.

Los componentes a su vez tienen variables, esencialmente por las cuales serán controlados. Haciendo objetos individuales dentro de la aplicación e iniciando funcionalidad en ellos, se logra expandir esta de una manera progresiva lógica.

1.5.3 Herramientas de diseño gráfico

Las herramientas de diseño gráfico permiten realizar la modelación y el manejo de objetos y entornos virtuales, brindan un conjunto de funciones que facilitan el trabajo con escenarios virtuales y algunas posibilitan la creación de visitas virtuales.

Blender

Es una poderosa herramienta para modelado 3D digital, tiene un modo de funcionamiento y una interfaz que son distintas a las de otros programas 3D. Reúne también amplias funciones para animación, composición, video y juegos. Blender 3D tiene otras características que son ventajosas. La principal es que se trata de Software Libre y además gratuito, condicionando esto, que se pueda descargar e instalar sin ninguna restricción y costo. Otra ventaja es que está disponible en todas las plataformas, permitiendo instalarlo en Windows, Mac OSX o GNU-Linux.

Blender es un programa de modelado y animación con las siguientes capacidades: animación 3D, creación de videojuegos 3D en tiempo real, generación de ambientes y paisajes 3D, generación de efectos especiales 3D, sombreado 3D, texturizado interactivo 3D, modelado 3D y simulación física (26).

Autodesk 3ds Max

Proporciona potentes herramientas integradas de modelado, animación, renderización y composición en 3D que multiplican rápidamente la productividad de los artistas y diseñadores. Ofrece herramientas y experiencias específicas a los desarrolladores de juegos, realizadores de efectos visuales y diseñadores gráficos.

El premiado software Autodesk 3ds Max es una de las herramientas preferidas por los profesionales líderes en desarrollo de juegos, televisión, cine y composición digital que quieren una solución 3D completa con resultados inmediatos. Además proporciona opciones de renderización integradas, incluida renderización en red ilimitada con tecnología mental ray (25).

Partiendo del estudio anteriormente realizado se resume que aunque Blender ofrece características relevantes para desarrollar el diseño gráfico, Autodesk 3ds Max, es una potente herramienta que se ajusta excelentemente a las expectativas de crear un ambiente 3D que servirá para la integración con el motor gráfico Unity 4.0. Además de esta poseer total compatibilidad con muchas de las herramientas de renderizado y tratamiento de imágenes que se usarán.

1.5.4 Herramientas CASE

Las herramientas CASE representan un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. Brindan ayuda a los analistas, ingenieros de software y desarrolladores, permitiendo el modelado de los sistemas mediante diferentes diagramas y generación de código a partir de estos y viceversa. El uso de estas herramientas aumenta la calidad del software desarrollado, debido a que una funcionalidad se basa en la comprobación automática de errores. Además permiten la reutilización de componentes de software como librerías, acelera el proceso de desarrollo del software y permite un desarrollo gradual e iterativo (27).

Visual Paradigm

Entre las herramientas CASE más utilizadas se encuentran Visual Paradigm. Su mayor éxito consiste en ser multiplataforma. Utiliza UML como lenguaje de modelado ofreciendo soluciones de software que permiten a las organizaciones desarrollar las aplicaciones de calidad más rápido, bien y más barato.

La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido una herramienta UML profesional que soporta el ciclo de vida completo del proceso de desarrollo del software: análisis y diseño orientados a objeto, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable, a través de la utilización de un enfoque Orientado a Objetos. Esta herramienta permite aumentar la calidad del software, a través de la mejora de la productividad en el desarrollo y mantenimiento del software. Entre otros aspectos a destacar se tienen que Visual Paradigm permite la reutilización del software, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería del Software. Se encuentra disponible en múltiples plataformas (Windows, Linux).El mismo posibilita la generación de bases

de datos transformación de diagramas de Entidad-Relación en tablas de base de datos. Admite la exportación e importación de ficheros XMI y facilita generar informes (28).

Partiendo del estudio anteriormente realizado se resume que Visual Paradigm ofrece características más relevantes para desarrollar el análisis y diseño del LV, ya que se ajusta a las características que persigue la Universidad de Ciencias Informáticas: es multiplataforma (Windows y Linux). Aporta a los desarrolladores de software una plataforma para construir aplicaciones más baratas y de mejor calidad. Además la universidad cuenta con la licencia de esta herramienta para su utilización. Es importante destacar también que ofrece versiones libres como es el caso de Visual Paradigm UML 6.4 Community Edition.

1.5.5 Lenguajes de programación

El motor de juego Unity3D brinda a los programadores la posibilidad de contar con tres lenguajes de programación para el desarrollo de aplicaciones. Estos lenguajes son (30):

- ✓ **C# (C Sharp):** basado en la plataforma .NET la cual puede integrarse con Mono. Es ideal para mejor rendimiento en los juegos.
- ✓ **Javascript:** se puede empezar usando este lenguaje para aprender a hacer scripts rápidos.
- ✓ **Boo:** no es más que una implementación de Python.

Los lenguajes de programación mencionados anteriormente brindan al programador potencialidades similares al trabajar con Unity3D, es decir no importa cual se elija, cualquiera servirá para exportar aplicación a las plataformas seleccionadas (30).

Para desarrollar la aplicación se seleccionó el lenguaje de programación C#, ya que el equipo de desarrollo cuenta con experiencia desarrollando en este lenguaje y tiene un mayor dominio de este que del resto.

1.6 Conclusiones parciales

En este capítulo se expusieron los problemas fundamentales que hacen necesario contar con las prácticas de extracción para el Laboratorio Virtual de Química, las cuales deben responder al programa de estudio de nuestro país. Se definió además, el concepto de LV que se utilizará en la presente tesis. Así mismo, se efectuó un análisis de algunas herramientas y tecnologías para la realización de la propuesta, seleccionándose Unity3D como herramienta de desarrollo, 3D

StudioMax como herramienta de diseño gráfico, y se definió XP como metodología para guiar todo el proceso. Por último, se optó por C# como lenguaje de programación.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

2.1 Introducción

El siguiente capítulo tiene como objetivo fundamental presentar la descripción de la solución y la documentación generada durante todo el proceso de desarrollo del proyecto el cual ha sido guiado por la metodología **Extreme Programming XP**.

2.2 Descripción de la solución propuesta

Teniendo en cuenta los objetivos que persigue el presente trabajo de diploma y en aras de utilizar la información recopilada en el capítulo anterior, se propone como solución desarrollar un módulo para el LV de Química que posibilite el desarrollo de las actividades tanto teóricas como prácticas relacionadas con los métodos de extracción continua líquido-líquido para líquidos más densos y menos densos que el agua, de la extracción sólido-líquido por medio de Soxhlet y el equipo de Dean-Stark.

Este módulo está dividido en cuatro sub-módulos, cada uno de los cuales se compone de dos actividades, una práctica y otra teórica:

- Sub-módulo “Extracción Soxhlet”
- Sub-módulo “Extracción Dean-Stark”
- Sub-módulo “Extracción líquido-líquido más denso”
- Sub-módulo “Extracción líquido-líquido menos denso”

El desarrollo de estos sub-módulos enmarca un conjunto de requerimientos que facilitan en gran medida la interacción del usuario con el laboratorio:

- Desarrollar una interfaz que le permita al usuario seleccionar el tipo de actividad de extracción que desea realizar.
- Leer los archivos XML con las diferentes preguntas que se le presentan al usuario como parte de la actividad teórica.
- Evaluar las respuestas entradas por el usuario.
- Diseñar un entorno realista de un *laboratorio de química* para efectuar la actividad práctica.
- Permitir al usuario seleccionar los objetos necesarios de la escena para ensamblar los instrumentos de extracción.

- Evaluar el ensamblado del instrumental.
- Mostrar calificación final en base a los resultados obtenidos en las actividades realizadas.

2.3 Fase de Planificación

La metodología XP se centra en la satisfacción del cliente, trata de dar al usuario final el software que él necesita en el momento que lo requiere. Permite aprovechar al máximo las ventajas del trabajo en equipo, crea un elevado nivel de colaboración y comunicación.

2.3.1 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe poseer. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

La finalidad de este trabajo de diploma es lograr una interfaz de navegación sencilla para que el usuario o estudiante que utilice la aplicación pueda maniobrar la misma de forma fácil, además de lograr una calidad gráfica aceptable para alcanzar un alto grado de realismo en la recreación del Laboratorio de Química. A continuación se exponen algunos de los principales requisitos a tener en cuenta en la realización de la aplicación:

Usabilidad: el producto final podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y conocimientos sobre los métodos de extracción en química.

Soporte: la aplicación contará antes de su puesta en marcha con un período de pruebas para detectar posibles errores y se brindará el servicio de configuración inicial.

Hardware: es necesario para el funcionamiento del sistema, un microprocesador *Intel Pentium IV* a 2,6 GHz, 512 MB RAM, 32 bits de profundidad de color y 128 MB de memoria de video integrado o 64 MB de video dedicado, compatible con *Open GL* o *DirectX 3D*.

Software: el producto final será multiplataforma debido a que podrá ser ejecutado en Windows y Linux.

Portabilidad: el sistema puede ser instalado en máquinas diferentes sin que se altere su funcionamiento.

Restriciones de diseño e implementación: cualquier implementación sobre la base del código aplicado en este sistema debe ser desarrollado en C#, Java Script o Boo (un dialecto del lenguaje Python), sobre Unity 3D, usando Programación Orientada a Objetos.

2.3.2 Historias de Usuario (HU)

Las Historias de Usuarios (HU) proporcionarán detalles sobre la estimación del riesgo y cuánto tiempo será empleado en su implementación. El cliente es el encargado de asignarle una prioridad

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

a cada HU y es el equipo de desarrollo el encargado de asignarle un costo, este se traduce en el tiempo que llevará el desarrollo de las mismas. Es importante destacar, que las HU nuevas pueden describirse en cualquier momento, con esto se comprueba la flexibilidad de la metodología.

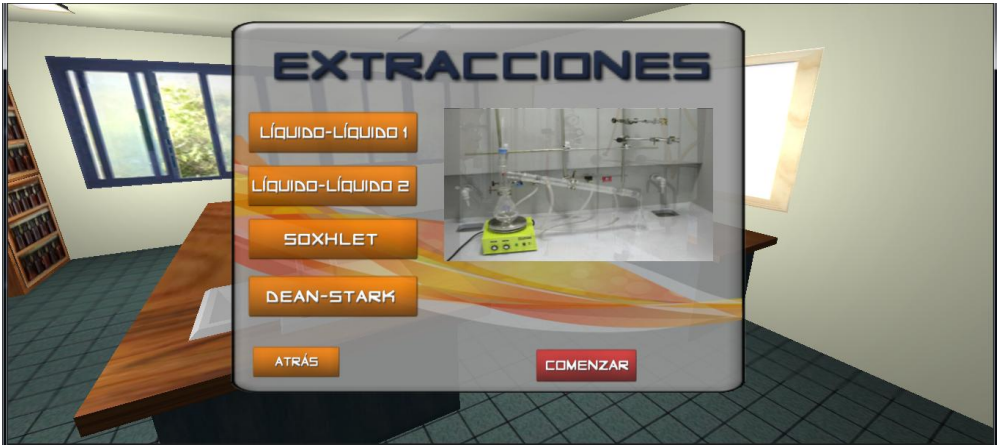
Número: 1	Nombre: Seleccionar actividad
Programador: Raydel	Iteración Asignada: 1
Prioridad en Negocio: Alta	Tiempo Estimado: 2 semanas
Riesgo en Desarrollo: Alto	Tiempo Real: 2 semanas
Descripción: El sistema ofrece la posibilidad de seleccionar el tipo de actividad de extracción que se desea realizar (Soxhlet, Dean-Stark, Líquido a Líquido (más denso), Líquido a Líquido (menos denso)).	
Observaciones:	
Prototipo de interfaz:	
	

Tabla 4. HU Seleccionar actividad

Número: 2	Nombre: Cargar actividad seleccionada
Programador: Raydel	Iteración Asignada: 1
Prioridad en Negocio: Alta	Tiempo Estimado: 2 semanas

Riesgo en Desarrollo: Alto	Tiempo Real: 2 semanas
Descripción: El sistema carga la escena correspondiente al tipo de actividad de extracción que se seleccionó (Soxhlet, Dean-Stark, Líquido-Líquido (más denso), Líquido-Líquido (menos denso)).	
Observaciones:	
Prototipo de interfaz:	
	

Tabla 5. HU Cargar actividad seleccionada

Las restantes HU quedarán reflejadas en el Expediente de Proyecto correspondiente a la investigación.

2.3.3 Plan de Entregas

El plan de entregas tiene el objetivo fijar el período de tiempo que se puede tardar en la implementación de cada una de las HU. En la primera iteración se intentará establecer una arquitectura del sistema la cuál puede ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura.

Entregas	Historias de Usuarios	Número
Entrega 1	Seleccionar actividad	1
	Cargar actividad seleccionada	2
	Retornar al menú principal	3
Entrega 2	Navegar por la escena	4
	Interactuar con los modelos en la escena	5
Entrega 3	Cargar el fichero de preguntas teóricas	6

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

	Aplicar el cuestionario teórico	7
	Emitir una nota parcial sobre la actividad teórica	8
Entrega 4	Seleccionar equipamiento en la escena	9
	Ensamblar elementos seleccionados	10
	Emitir nota parcial sobre actividad práctica	11
Entrega 5	Emitir nota total de la práctica seleccionada	12

Tabla 6. Plan de Entregas

2.3.4 Plan de Iteraciones

En la siguiente tabla se muestran las iteraciones definidas para el desarrollo del sistema y las entregas del producto que son realizadas en cada una de estas.

Iteración 1 3ra Semana de Enero	Iteración 2 3ra Semana de Marzo	Iteración 3 3ra Semana de Mayo
Entrega 1	Entrega 2	Entrega 4
	Entrega 3	Entrega 5

Tabla 7. Plan de Iteraciones

2.4 Fase de Diseño

El diseño del sistema crea una estructura para la organización de la lógica del sistema y posibilita que sea escalable con cambios en un solo lugar. Los diseños deben poseer el mayor grado de sencillez posible, dividiéndolos en varias partes en caso de tornarse complejos.

2.4.1 Diagrama de clases

En la figura a continuación se muestra el Diagrama de Clases correspondiente al sistema, reflejando los elementos que conforman la solución.

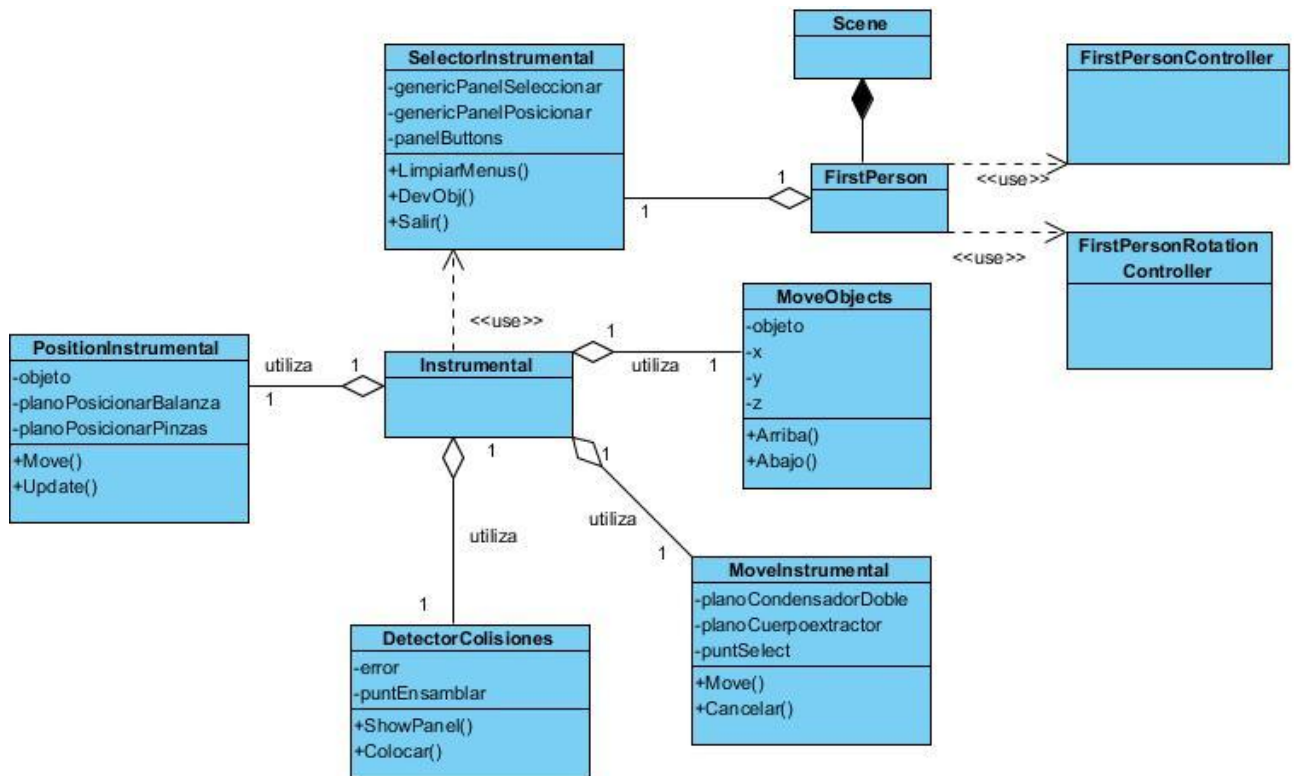


Figura 9. Diagrama de Clases

2.4.2 Tarjetas CRC del sistema

“SelectorInstrumentalExt.cs”

Descripción: Entidad encargada de seleccionar los elementos de la escena.

Atributos:

Nombre:	Descripción:
label1	Nombre del objeto seleccionado que aparece en el panel posicionar.
label	Nombre del objeto seleccionado que aparece en el panel seleccionar.
ray	Almacena la posición del mouse en la pantalla.
hit	Devuelve que objeto es tocado por el mouse.
objeto	Almacena el objeto de la escena seleccionado en cada momento.
state	Almacena el estado del objeto que puede ser seleccionado o no seleccionado (True o false).
genericPanelSeleccionar	Contiene la configuración del panel seleccionar

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

	instrumental.
genericPanelPosicionar	Contiene la configuración del panel posicionar instrumental.
panelButtons	Contiene la configuración del panel mover instrumental.
buttonColocar	Permite colocar el elemento una vez posicionado correctamente en la mesa de ensamblado.
Responsabilidades:	Colaboración:
Start ()	FirstPerson
Update ()	Instrumental
Salir()	
LimpiarMenus()	
DevObj()	
ShowPanel()	
ClearPanel()	

“PositionInstrumentalExt.cs”

Descripción: Entidad encargada de realizar el posicionamiento de los objetos desde la mesa de armado a su posición antes del ensamblaje.

Atributos:

Nombre:	Descripción:
objeto	Almacena el objeto de la escena seleccionado en cada momento.
planoPosicionarBalanza	Almacena la posición inicial para ubicar la balanza.
planoPosicionarPinzas	Almacena la posición inicial para ubicar las pinzas.
planoPosicionarObjetos	Almacena la posición inicial para ubicar los otros objetos de la escena.
planoPosicionarDoble	Almacena la posición inicial para ubicar el condensador doble.
label	Nombre del objeto que aparece en el panel posicionar.
state	Almacena el estado del objeto que puede ser

	seleccionado o no seleccionado (True o false).
Responsabilidades:	Colaboración:
Update ()	Instrumental
Move()	

Las restantes tarjetas CRC quedarán reflejadas en el Anexo 1 del presente trabajo de diploma.

2.4.3 Arquitectura del sistema

Los patrones de arquitectura son aquellos que expresan un esquema organizativo estructural fundamental para sistemas de software.

La recreación del LV de Química se desarrolla sobre el motor gráfico de *Unity 3D* y posee una arquitectura en 3 capas que se describe a continuación:

- **Capa de Presentación:** es la capa con la que interactúa el usuario y está formada por las interfaces, las cuales le permiten al usuario navegar por el sistema.

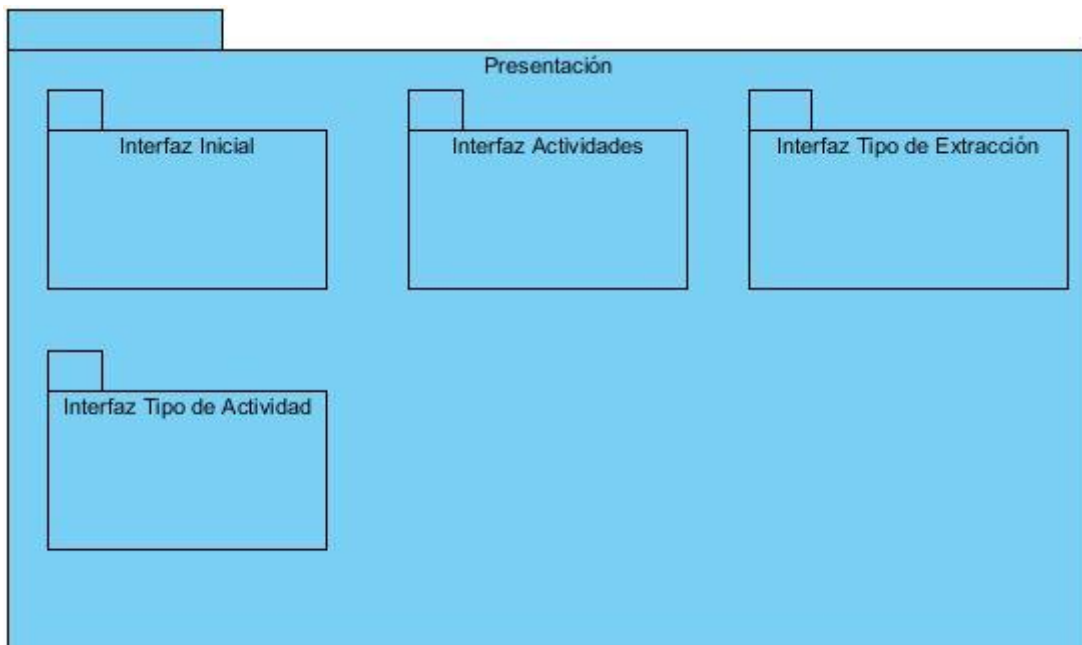


Figura 10. Capa de presentación

- **Capa de Negocio:** está formada por las entidades, que representan objetos que van a ser manejados o utilizados por la aplicación.

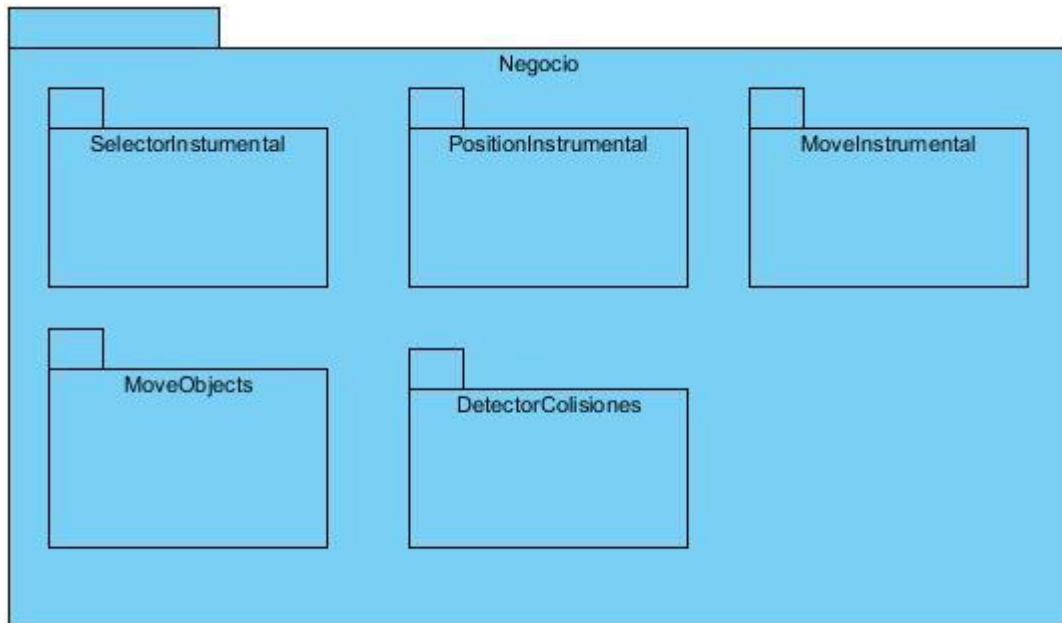


Figura 11. Lógica del negocio

- **Capa de Soporte:** contiene las librerías para la física, los gráficos, el sonido, y el almacenamiento de los datos brindados por el motor gráfico de *Unity 3D*.

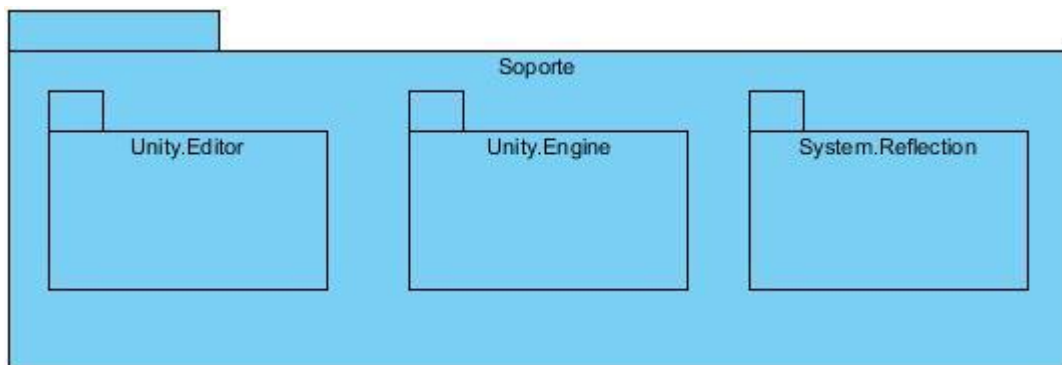


Figura 12. Librerías de soporte

2.4.4 Patrones del Diseño

Los Patrones de diseño son soluciones comunes a problemas de diseño de software orientado a objetos y que además poseen ciertas características de efectividad para resolver ese problema. Son reusables ya que pueden ser aplicados en otros diseños o problemas (31).

Los Patrones de Asignación de Responsabilidades (GRASP) son un sistema orientado a objetos, los cuales envían mensajes a otros objetos para llevar a cabo las operaciones requeridas (31).

Experto: asigna la responsabilidad al experto en información, que es la clase que cuenta con la información necesaria para cumplir dicha responsabilidad. El empleo adecuado de este patrón proporciona muchos beneficios, uno de ellos es que se conserva el encapsulamiento debido a que los objetos se valen de su propia información para hacer lo que se les pide; esto soporta un bajo acoplamiento y permite tener sistemas más robustos. Además brinda soporte a una alta cohesión, ya que al distribuir el comportamiento entre las clases que cuentan con la información requerida alienta que las definiciones de dichas clases sean más sencillas y cohesivas.

En la implementación del LV este patrón se aplica en la entidad “SelectorInstrumental” ya que es la encargada de realizar todas las inicializaciones de los paneles seleccionar y posicionar instrumental.

```
294         if(objeto.tag == "embudo separador-md" )
295         {
296             if(objeto.transform.position.x >0f)
297             {
298                 label.text="embudo separador";
299
300                 genericPanelSeleccionar.GetComponent<TweenAlpha>().Play();
301                 genericPanelSeleccionar.GetComponent<TweenPosition>().Play();
302
303                 state = true;
304             }
305         }
306
307         if(objeto.transform.position.x <0f && objeto.transform.position.z>23f)
308         {
309             label1.text="embudo separador";
310
311             genericPanelPosicionar.GetComponent<TweenAlpha>().Play();
312             genericPanelPosicionar.GetComponent<TweenPosition>().Play();
313
314             state = true;
315         }
316     }
317 }
```

Figura 13. Fragmento de código donde se emplea el patrón Experto

Eric Gamma, Richard Helm, Ralph Johnson y John Vlissides publicaron el primer catálogo de patrones en el ámbito del software en 1994, por lo que estos patrones son conocidos como *patrones GOF* (Gang of Four), que incluyen 23 patrones de diseño (31).

Singleton: garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella. El patrón de diseño singleton (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

En la implementación del LV este patrón se aplica en las variables y paneles que se declaran de tipo estáticos y se usan como una única instancia en todo el sistema. Por ejemplo las variables

booleanas que identifican si un objeto esta ya colocado y la variable error que muestra si se efectuó un error durante el ensamblado.

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class DetectorColisionesExt : MonoBehaviour {
5
6     public Transform buttonColocar;
7     public Transform PanelError;
8     public Transform PanelComenzarSimulacion;
9
10    public int puntEnsamblar;
11
12    GameObject objeto;
13
14    static bool pinza1;
15    static bool pinza2;
16    static bool pinza3;
17    static bool colocadoapa;
18    static bool colocadocon;
19    static bool error;
20
21    int cont=0;
```

Figura 14. Fragmento de código donde se emplea el patrón Singleton

2.5 Conclusiones del capítulo

En este capítulo se describe el diseño del sistema y se presentan los artefactos más significativos generados durante la etapa de implementación de la aplicación. Se hace además especial énfasis en las historias de usuarios, los protipos de interfaz, las tarjetas CRC, así como el plan de iteraciones realizado durante el desarrollo de la solución. Por último se presenta la arquitectura del sistema y algunos de los patrones de diseño utilizados.

CAPÍTULO 3. IMPLEMENTACIÓN, PRUEBAS Y VALIDACIÓN DEL SISTEMA

3.1 Fase de Desarrollo

A la hora de codificar no se debe olvidar que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas, por esto siempre deben estar presentes. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que esta hará y también tendrá que estar presente cuando se realicen las pruebas que verifiquen que la historia implementada cumple la funcionalidad especificada. En esta fase de la codificación, los clientes y los desarrolladores del proyecto deben estar en comunicación para que los desarrolladores puedan codificar todo lo que se requiere para el proyecto. En esta fase está incluida toda la codificación o programación realizada por los desarrolladores del proyecto.

3.1.1 Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, se establece un estándar de codificación para asegurar que todos los programadores del proyecto trabajen de forma coordinada. Cuando al proyecto de software se le incorpore código fuente previo, o bien cuando se realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación deberá establecer cómo operar con la base de código existente (33). Dentro de los estándares de codificación utilizados en la implementación del LV se encuentran:

Espacio dentro del código: es muy importante que a la hora de escribir código, se dejen los espacios de separación entre los elementos para poder leer con soltura. La razón es obvia. El código sin espacios es completamente ilegible. Cuando se tiene un signo y un operador binario se colocan espacios a ambos lados de ellos y en caso de tener un signo unario los espacios son colocados a uno de sus lados.

Comentario de código: todas las funciones tienen que estar documentadas explicando que realiza cada una de ellas en forma de comentarios (34).

Notación PascalCasing: los nombres de las variables deben ser descriptivos y concisos, se recomienda no utilizar grandes frases ni pequeñas abreviaciones, ya que es mejor conocer lo que

hace una variable con solo leer su nombre; esto es aplicable a variables, clases, funciones y sus argumentos. Los nombres de las funciones pueden contener solamente caracteres alfanuméricos, deben comenzar siempre en letra mayúscula y cuando tienen más de una palabra, la primera letra de cada una de ellas debe capitalizarse (Upper Camel Case).

Indentación en el código (Tabs o espacios): son aquellos tabs o espacios que se ponen delante de las líneas de código, los cuales se han utilizado en este trabajo para lograr una estructura más organizada y legible que permita un mejor entendimiento del código a otros programadores que ne cesiten consultarlo en el futuro. Todo el código desarrollado tendrá una indentación de 8 espacios teniendo en cuenta los niveles de cada sentencia dentro de las funciones.

El siguiente fragmento de código muestra cómo se ponen de manifiesto los estándares de codificación en la implementación del LV.

```
156 // Metodo utilizado para recargar las escenas
157
158 public void RecargarEscena() {
159
160     if (Application.loadedLevelName == "Extraccionsoxhlet") {
161         Application.LoadLevel ("Extraccionsoxhlet");
162     }
163
164     if (Application.loadedLevelName == "Extracciondeanstark") {
165         Application.LoadLevel ("Extracciondeanstark");
166     }
167
168     if (Application.loadedLevelName == "Extraccionmenosdenso") {
169         Application.LoadLevel ("Extraccionmenosdenso");
170     }
171
172     if (Application.loadedLevelName == "Extraccionmasdenso") {
173         Application.LoadLevel ("Extraccionmasdenso");
174     }
175 }
176
177
```

Figura 15. Fragmento de código donde se emplean los estándares de codificación

3.1.2 Tareas de implementación

Tareas de implementación	
Número: 1	Número de la Historia: 1
Nombre de la Tarea: Diseño del panel para la actividad de extracción	
Tipo de Tarea: Desarrollo	Tiempo Estimado: 3 días

Fecha de Inicio: 12 de enero	Fecha de Fin: 15 de enero
Programador responsable: Raydel Comas Rodríguez	
Descripción: Se realiza el diseño del panel para seleccionar el tipo de actividad de extracción que se desea realizar.	

Tareas de implementación	
Número: 2	Número de la Historia: 2 y 3
Nombre de la Tarea: Implementación de la lógica del seleccionar actividad	
Tipo de Tarea: Desarrollo	Tiempo Estimado: 9 días
Fecha de Inicio: 15 de enero	Fecha de Fin: 28 de enero
Programador responsable: Raydel Comas Rodríguez	
Descripción: Se realiza la implementación de la lógica de interacción con el panel de seleccionar actividad.	

Las restantes tareas de implementación quedarán reflejadas en el Anexo 2 de la presente investigación.

3.1.3 Componentes desarrollados en la Implementación

Para la correcta implementación del LV de Química fue necesario desarrollar una serie de componentes los cuáles están divididos por las 3 etapas fundamentales concebidas en la aplicación:

- Teórica
- Práctica
- Evaluación

En cada una de estas etapas los componentes que se crearon fueron del tipo:

- **Escenas:** representan los diferentes niveles creados para realizar cada una de las actividades.
- **Modelos 3D:** representan modelos del instrumental relacionado con cada uno de los métodos de extracción.
- **Paneles:** son aquellos que informan lo que va sucediendo a medida que se interactúa con los objetos de una escena.
- **Scripts:** el Scripting es una parte esencial ya que define el comportamiento del usuario y los objetos dentro del LV.

Etapa teórica

La siguiente figura muestra la escena donde se realizan las actividades teóricas del LV.

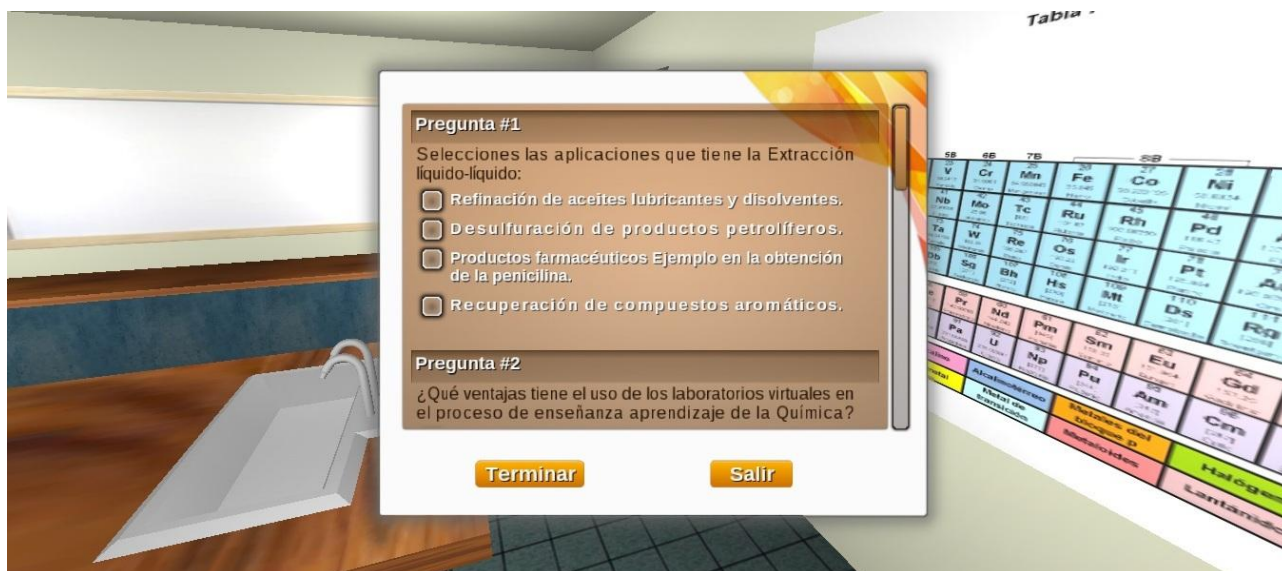


Figura 16. Prototipo de interfaz de escena teórica

Etapa práctica

En esta figura se pueden apreciar algunos de los modelos creados para poder realizar el proceso de ensamblado dentro del LV.



Figura 17. Modelos de instrumental para la práctica de extracción

Esta parte del código se utiliza para lograr el movimiento de los objetos durante el proceso de ensamblado.

```
41     public void Arriba() {
42
43         GameObject.Find("Main Camera").GetComponent<SelectorInstrumentalExt>().LimpiarMenus();
44         objeto = GameObject.Find("Main Camera").GetComponent<SelectorInstrumentalExt>().DevObj();
45
46         x = objeto.transform.position.x;
47         y = objeto.transform.position.y;
48         z = objeto.transform.position.z;
49         objeto.transform.position = new Vector3(x, y + mov, z);
50
51         GameObject.Find("Main Camera").GetComponent<SelectorInstrumentalExt>().state = true;
52     }
53     public void Abajo() {
54
55         GameObject.Find("Main Camera").GetComponent<SelectorInstrumentalExt>().LimpiarMenus();
56         objeto = GameObject.Find("Main Camera").GetComponent<SelectorInstrumentalExt>().DevObj();
57
58         x = objeto.transform.position.x;
59         y = objeto.transform.position.y;
60         z = objeto.transform.position.z;
61         objeto.transform.position = new Vector3(x, y - mov, z);
62
63         GameObject.Find("Main Camera").GetComponent<SelectorInstrumentalExt>().state = true;
64     }
```

Figura 18. Fragmento de código utilizado para mover los objetos

Etapa evaluativa

El siguiente panel muestra el resultado obtenido por el usuario en la actividad práctica.

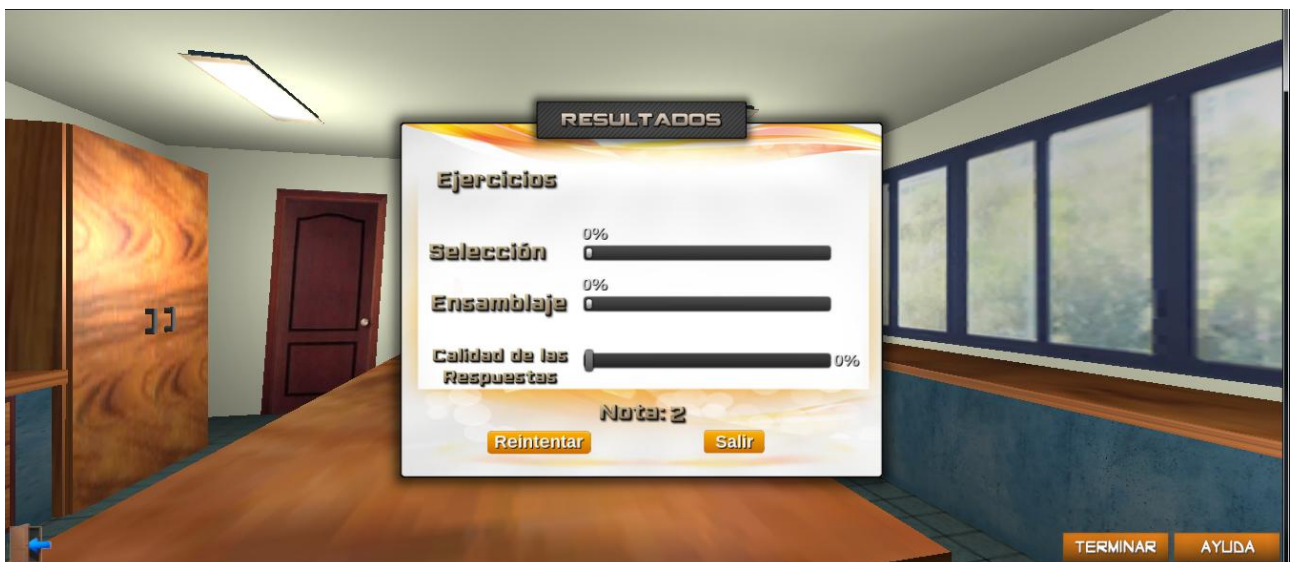


Figura 19. Panel de evaluación práctica

3.2 Pruebas de Aceptación

Uno de los pilares de Extreme Programming (XP) es el proceso de pruebas. XP alienta a probar constantemente, tanto como sea posible. Esto permite aumentar la calidad de los sistemas

reduciendo el número de fallos no detectados y disminuyendo el tiempo entre la aparición de un error y su detección. También permite evitar efectos colaterales no deseados a la hora de realizar modificaciones (20).

Las Pruebas de Aceptación propuestas por XP son realizadas por el cliente en compañía de al menos un integrante del equipo de desarrollo y se orientan a las funcionalidades del sistema. Son diseñadas a partir de las HU y tiene como objetivo verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

Pruebas de Aceptación	
Casos de Prueba: 1	Número de Historia: 1
Nombre de Prueba: Seleccionar actividad	
Descripción: Selecciona el tipo de actividad de extracción que desea realizar.	
Condiciones de ejecución: El usuario debe haber seleccionado la actividad extracción en el panel Actividades.	
Entradas/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. En el Panel Inicial se selecciona el botón Comenzar y se muestra el panel Actividades. 2. En el panel Actividades se selecciona el botón Extracción y se muestra una imagen relacionada con el ejercicio, luego se pulsa el botón Comenzar, de esta forma se muestra el panel Extracciones (Menos denso, Más denso, Soxhlet, Dean-Stark). 3. En el panel Extracciones se escoge la actividad de extracción que se desea realizar y se pulsa Comenzar. 	
Resultado esperado: El sistema muestra la modalidad práctica o teórica.	
Evaluación de la prueba: Satisfactorio	

Pruebas de Aceptación	
Casos de Prueba: 2	Número de Historia: 5
Nombre de Prueba: Interactuar con los modelos de la escena	
Descripción: Permite que el usuario pueda interactuar con el instrumental de química relacionado con los ejercicios en la escena.	
Condiciones de ejecución: El usuario debe haber seleccionado el tipo de actividad extracción que desea realizar (Menos denso, Más denso, Soxhlet, Dean-Stark) y la modalidad práctica.	
Entradas/Pasos de ejecución:	

1. En el Panel Inicial se selecciona el botón Comenzar y se muestra el panel Actividades.
 2. En el panel Actividades se selecciona el botón Extracción y se muestra una imagen relacionada con el ejercicio, luego se pulsa el botón Comenzar, de esta forma se muestra el panel Extracciones (Menos denso, Más denso, Soxhlet, Dean-Stark).
 3. En el panel Extracciones se escoge la actividad de extracción que se desea realizar y se pulsa Comenzar.
 4. En el panel Tipo de Actividad se selecciona la modalidad Práctica y se pulsa Comenzar.
- El sistema muestra la escena donde se llevará a cabo la actividad práctica. Se selecciona algún modelo del instrumental de la escena.

Resultado esperado: El sistema muestra un panel con las acciones que se pueden realizar con el modelo las cuales son tomar o cancelar.

Evaluación de la prueba: Satisfactorio

Pruebas de Aceptación	
Casos de Prueba: 3	Número de Historia: 6
Nombre de Prueba: Cargar el fichero de preguntas teóricas	
Descripción: Permite que el usuario pueda realizar las preguntas teóricas de la actividad seleccionada (Menos denso, Más denso, Soxhlet, Dean-Stark).	
Condiciones de ejecución: El usuario debe haber seleccionado la actividad de extracción que desea realizar y luego el tipo de actividad teórica.	
Entradas/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. En el Panel Inicial se selecciona el botón Comenzar y se muestra el panel Actividades. 2. En el panel Actividades se selecciona el botón Extracción y luego el de Comenzar, de esta forma se muestra el panel Extracciones (Menos denso, Más denso, Soxhlet, Dean-Stark). 3. En el panel Extracciones se escoge la actividad de extracción que se desea realizar y se pulsa Comenzar 4. Luego en el panel Tipo de Actividad se selecciona la modalidad Teórico. 	
Resultado esperado: El sistema muestra un panel con las preguntas teóricas relacionadas al tipo de método de extracción seleccionado.	
Evaluación de la prueba: Satisfactorio	

Pruebas de Aceptación	
Casos de Prueba: 4	Número de Historia: 11

Nombre de Prueba: Emitir nota parcial sobre actividad práctica
Descripción: Permite que el usuario pueda ver su evaluación en la actividad práctica.
Condiciones de ejecución: El usuario debe haber seleccionado el botón Terminar que aparece en la escena práctica y pulsando sí en la advertencia que pregunta si se desea terminar el ejercicio en ese momento.
Entradas/Pasos de ejecución: <ol style="list-style-type: none">1. En el Panel Inicial se selecciona el botón Comenzar y se muestra el panel Actividades.2. En el panel Actividades se selecciona el botón Extracción y se muestra una imagen relacionada con el ejercicio, luego se pulsa el botón Comenzar, de esta forma se muestra el panel Extracciones (Menos denso, Más denso, Soxhlet, Dean-Stark).3. En el panel Extracciones se escoge la actividad de extracción que se desea realizar y se pulsa Comenzar.4. En el panel Tipo de Actividad se selecciona la modalidad Práctica y se pulsa Comenzar.5. En la escena práctica se selecciona el botón Terminar y se muestra un panel que pregunta si se desea terminar el ejercicio en ese momento.6. Se selecciona sí y aparece un panel con la evaluación obtenida en la actividad práctica.
Resultado esperado: El sistema muestra en un panel la evaluación obtenida en la actividad práctica.
Evaluación de la prueba: Satisfactorio

En el Anexo 3 del presente trabajo de diploma se muestra el acta de aceptación otorgado por parte de los clientes una vez realizadas las pruebas al sistema.

3.3 Validación del sistema

Para validar el sistema se efectuó el cuestionario mostrado en el Anexo 4 a profesores, estudiantes y especialistas en química. Algunos de los parámetros que se tuvieron en cuenta para la confección del mismo fueron: la calidad visual del LV, complejidad de las preguntas teóricas, navegabilidad en la solución, el realismo de la escena práctica y la complejidad en el ensamblado del equipamiento.

El siguiente gráfico evidencia los resultados alcanzados en una muestra de 30 participantes:



Figura 20. Gráfico con resultados del cuestionario

3.4 Conclusiones

Después de la investigación realizada sobre los problemas que presentan los estudiantes de química al momento de realizar las prácticas de extracción, se acordó desarrollar el módulo de extracción para el laboratorio virtual de química. En primer lugar se estudiaron cuáles eran las principales tendencias y tecnologías que actualmente existen referentes a los laboratorios virtuales y en específico los LV de química. Luego se realizó el análisis, diseño e implementación de la propuesta del módulo de extracción para el LV de Química, siguiendo el ciclo de vida que propone XP como metodología de desarrollo y se obtuvo como resultado una solución la cual permite que los estudiantes desarrollen habilidades teórico-prácticas que se adquieren al realizar las prácticas de laboratorios reales sobre los métodos de extracción. Para concebir la solución se hace necesario, tener conocimientos sobre el uso de la herramienta de desarrollo Unity 3d y conocer el lenguaje de programación C#. Además dominar el modelado, texturizado y creación de animaciones de objetos industriales en la herramienta de diseño gráfico 3D MAX, así como conocer el lenguaje UML y la herramienta Visual Paradigm para modelar cada uno de los artefactos ingenieriles necesarios. Por último se realizaron las pruebas necesarias para el buen funcionamiento del sistema y se aplicó un cuestionario para la validación del mismo.

BIBLIOGRAFÍA

1. Medina, Leonardo Antonio Nieblas Palau y Yasmany Cubela. Interfaz Visual para la configuración de Entornos Virtuales desarrollados con la Herramienta Scene Tool Kit. La Habana: s.n., 2008.
2. Barcelona, DLSI Facultad informática de. FIB. [En línea] www.lsi.upc.edu/~virtual/SGL/guions/ArquitecturaRV.pdf.
3. MONGE-NÁGERA, J. La evolución de los laboratorios virtuales durante una experiencia de seis años con estudiantes a distancia., 1999.
4. VARY, J. P. *Informe de la reunión de expertos sobre laboratorios virtuales*, Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura, 1999.
5. LUCERO, I.-M., S. - SAMPALLO, G. - AGUIRRE, M. S. -CONCARI. Trabajos de Laboratorio de Física en ambiente real y virtual. Universidad Nacional del Nordeste, Dpto. de Física, 2000.
6. López García, Marta y Morcillo Ortega, Juan Gabriel. Las TIC en la enseñanza de la Biología en la educación secundaria: los laboratorios virtuales. 2007.
7. L. Rosado, J. R. Herreros. Nuevas aportaciones didácticas de los laboratorios virtuales y remotos en la enseñanza de la Física. 2005. <http://www.uv.es/eees/archivo/286.pdf>.
8. Rodrigo, Vicent M. Modelo de referencia de laboratorios virtuales y Aplicaciones a sistemas de Tele-educación, 2003.
9. Goldberg, Harold. What is Virtual Instrumentation? IEEE Instrumentation & Measurement Magazine, 2000.
10. Haden, G.L. An object Oriented Modelling and Simulation Component for Virtual Engineering Laboratories. Proceedings of 22nd annual conference Frontiers in Education, 1992.
11. Mosterman, P.J. Donaldt, M.A.M. Campbell, J. Olin. Burow, C. Bouw, R. Brodersen A.J. Bourne J.R. Virtual Engineering Laboratories: Design and Experiments. Journal of Engineering Education, 1994.
12. Goldberg, K., et al. The Mercury project – Robotic tele-excavation. Beyond the Web: Excavating the real world via mosaic. International WWW Conference. Chicago. 1994.
13. Werges, Stephan C. Naylor, David L. A Networked Instructional Instrumentation Facility. Annual Meeting of the American Society for Engineering Education, Milwaukee, 1997.
14. Yenka. [En línea]. http://www.crocodile-clips.com/es/Crocodile_Chemistry.
15. Model science software Model ChemLab. [En línea]. <http://www.modelscience.com/products.html?ref=home&link=chemlab>
16. Sibeas Soft. VLabQ. [En línea]. <http://www.sibeas.com/prog.php?id=11>
17. StudyRoom Labs. Aulas Virtuales y Laboratorios Multimedia. [En línea].

BIBLIOGRAFÍA

- http://www.studyroomlabs.com/Software/Simulacion_laboratorio_quimica.htm
18. Tellez, Linda Luna. Scribd. [En línea] <http://www.scribd.com>.
 19. Tellez, Linda Luna. Scribd. [En línea] 28 de Abril de 2012. <http://es.scribd.com>.
 20. Penadés, Patricio Letelier y M^a Carmen. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Valencia: Universidad Politécnica de Valencia, 2009.
 21. PALACIOS, Juan. El modelo Scrum. http://eva.uci.cu/file.php/161/Documentos/Materiales_complementarios/UD_1_Procesos/Metodologias/SCRUM/EI_modelo_SCRUM.pdf.
 22. GamerDic. Diccionario online de términos sobre videojuegos y cultura gamer. [En línea]. <http://www.gamerdic.es/termino/motor-de-juego>
 23. COLLADO, David. Empezando en Unity 3D. Webmaster, Unity-Spain.
 24. Unreal Engine. Licencias para UDK. [En línea]. <http://www.unrealengine.com/udk/licensing/licensing-faqs/>
 25. Autodesk. Productos Autodesk 3ds Max. [En línea] <http://www.autodesk.es/adsk/servlet/pc/index?siteID=455755&id=14626995>.
 26. Moya, Rafael. Guía para Modelar arquitectura en Blender. [En línea]. <https://docs.google.com/viewer?a=v&pid=explorer&chrome=true&srcid=0BwC9xl05D8nBMWQwOTVjYjctMjBhMi00MTM0LTljZGQtNTQ1MTk1YWQ2OTM5&hl=en>.
 27. Zarzuela, Jorge Ferrer. Metodologías Ágiles. [En línea]. 2003. <http://libresoft.dat.escet.urjc.es/html/downloads/ferrer-20030312.pdf>.
 28. Pressman, Roger S. Ingeniería de Software, un enfoque práctico. Quinta edición. s.l.: McGraw-Hill Companies, 2002. ISBN: 8448132149.
 29. Visconti, Marcello y Astudillo, Hernán. Fundamentos de la Ingeniería de Software. 2007.
 30. ¡Aprende a desarrollar SoftwareUnity3D! Motor de videojuegos compatible con múltiples plataformas. [En línea]. <http://developeando.net/unity3d/>.
 31. Larman, Craig. UML y patrones. Introducción al análisis y diseño orientado a objetos. [ed.] Pablo Eduardo Roig Velázquez. [trad.] Luz Maria Henhndez Rodriguez. Primera. México Dawn Speth White, 1999. pág. 419. ISBN/970-17-0261-1.
 32. Caballero, Aidé Peralta. Arceo, Frida Díaz Barriga. Diseño instruccional de ambientes virtuales de aprendizaje desde una perspectiva constructivista. [En línea]. <http://www.gabinetecomunicacionyeducacion.com/files/adjuntos/Dise%C3%B1o%20Instruccional%20de%20ambientes%20virtuales%20de%20aprendizaje%20desde%20una%20perspectiva%20constructivista.pdf>.
 33. Developer Network. Revisiones de código y estándares de codificación. [En línea]. <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
 34. Marín, David. Estándares de codificación. [En línea]. 2003.

BIBLIOGRAFÍA

<http://www.aspl.es/fact/files/aspl-fact/estandares-node2.html>.