

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 5



**Propuesta de interfaz de usuario para el
Ambiente Virtual de Aprendizaje Interactivo
de tecnologías de prótesis oculares**

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

Autor: Yusiel Romero Hernández

Tutor: Lic. Luis Gabriel Viciado Caraballoso

La Habana, Cuba

Junio de 2013

DEDICATORIA

De Yusiel

Dedico este trabajo de diploma a mi mamá por darme todo su apoyo en todo momento, incluso en los momentos más difíciles para ella, gracias por tanto amor, por enseñarme que la perseverancia, la dedicación y el empeño son las armas claves para el logro de nuestros objetivos, por el impulso que me dió siempre que lo necesité, por las palabras de aliento. Por guiarme para ser la persona que soy hoy.

AGRADECIMIENTOS

Quiero agradecer a todas las personas que de una forma u otra han sido partícipes de este logro.

A mi mamá, la persona más importante de mi vida, por sus consejos, su paciencia y apoyo, por su gran esfuerzo por verme un día convertido en ingeniero...!este título es para ti mami!

A Marin, que fue la primera persona que conocí al llegar a la universidad, de ahí en adelante ha sido una persona clave en este logro y más que un amigo ha sido un hermano,

A Yuleisy por sus consejos oportunos en todo momento, a Yindra, por estar siempre presente, especialmente en los momentos más duros de mi carrera, realmente ha sido la hermana que nunca tuve.

A Gretter, por darme el regaño en el momento oportuno y por brindarme tanto apoyo.

A Yoandry, por brindarme tanto apoyo, durante todo el tiempo en la universidad, gracias por tanta paciencia y ser tan buen amigo.

A mi tutor Viciado, por ser un padre más que un profesor.

A Marrero, Humberto, Henry, Toki, Taumara, Mauricio, Eddy, Wicho, Norberto, El Zurdo, Leydis,

Yadira, Darlin, Raúl....a todos GRACIAS

DECLARACIÓN JURADA DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco 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 _____.

Firma del autor

Yusiel Romero Hernández

Firma del tutor

Luis Gabriel Viciado Carabaloso

DATOS DE CONTACTO

Tutor:

Nombre y Apellidos: Luis Gabriel Viciado Carabaloso.

Institución: Universidad de las Ciencias Informáticas.

Título: Licenciado en Educación, especialidad Física.

Categoría Docente: Profesor Auxiliar.

E-mail: viciedo@uci.cu

Se desempeña como Profesor Auxiliar desde el año 2000. Ha realizado tutorías en más de 15 trabajos de diploma en la carrera de Ingeniería Informática y se ha desempeñado en tribunales de eventos científicos y de cambios de categoría docente en la UCI. Tiene más de 20 publicaciones sobre el tema de Laboratorios Virtuales y Ambientes de Aprendizaje Interactivos desde el año 2008.

RESUMEN

El presente trabajo tiene como propósito dotar al Ambiente Virtual de Aprendizaje Interactivo (AVAI) de construcción de prótesis oculares que se desarrolla en el proyecto PROTESIM de una Interfaz Gráfica de Usuario (GUI) fundamentada en los principios básicos para el diseño de las interfaces gráficas, así como de diferentes funcionalidades que presentan algunos juegos serios y ambientes virtuales de aprendizaje.

En este documento se exponen algunos de los conceptos asociados a la interfaz de usuario, además se hace referencia a conceptos asociados a: medio ambiente de aprendizaje, ambiente virtual de aprendizaje interactivo y juegos serios. También se mencionan algunas de las características fundamentales, los principios para el diseño de las mismas, así como el ciclo de desarrollo. Se abordan los conceptos de accesibilidad y usabilidad enmarcado dentro de la Arquitectura de la Información. Se fundamenta la selección de las metodologías, herramientas, tecnologías utilizadas y se presenta la propuesta de solución a través de los artefactos generados en el diseño. Finalmente se valida la solución propuesta mediante el método Delphi.

Palabras clave: Interfaz Gráfica de Usuario (GUI), Principios de diseño.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: Fundamentación teórica y selección de tecnologías	4
1.1 Medio Ambiente de Aprendizaje	4
1.2 Ambiente Virtual de Aprendizaje (AVA)	4
1.3 Tipos de Ambientes Virtuales de Aprendizaje.....	5
1.3.1 Ventajas de los Ambientes Virtuales de Aprendizaje.....	5
1.4 Juegos Serios.....	6
1.5 Ambiente Virtual de Aprendizaje Interactivo (AVAI).....	6
1.6 Creación de un Cuestionario Evaluativo	7
1.7 Tipos de preguntas a utilizar en un cuestionario.....	8
1.8 La interfaz y las dimensiones de la provisión de estímulos sensoriales	9
1.9 Interfaz de usuario.....	9
1.10 Interfaz Gráfica de Usuario	11
1.10.1 Componentes Gráficos	11
1.10.2 Características de las GUI	12
1.10.3 Ventajas del uso de las GUI.....	12
1.10.4 Diseño de Interfaces Gráficas de Usuario	13
1.10.5 Principios para el diseño de interfaces gráficas.....	15
1.11 Motor Gráfico para el desarrollo.....	21
1.11.1 Posibilidades de exportación	22
1.12 Lenguaje de programación.....	22
1.12.1 Lenguaje de programación LUA	23
1.13 Metodologías de desarrollo de software	23
1.13.1 El Proceso Unificado de Desarrollo de Software (RUP)	23
1.14 Lenguaje Unificado de Modelado (UML).....	25
1.15 Herramientas CASE	26
1.15.1 Visual Paradigm for UML Enterprise Edition.....	26
Conclusiones Parciales.....	28
CAPÍTULO 2: Características del Sistema	29
2.1 Elementos tomados en cuenta para la propuesta de solución	29
2.2 Funcionalidades.....	¡Error! Marcador no definido.
2.3 Modelo de Dominio	34

2.3.1 Descripción del modelo de dominio.....	34
2.4 Levantamiento de requisitos del sistema	35
2.4.1 Requisitos funcionales.....	35
2.4.2 Requisitos no Funcionales	35
2.5 Definición del actor del sistema	36
2.6 Casos de uso del sistema	37
2.6.1 Diagrama de casos de uso del sistema.....	37
2.6.2 Descripción de los casos de uso del sistema.....	37
2.7 Diagramas de interacción	43
Conclusiones parciales	47
CAPÍTULO 3: Implementación y Validación del Sistema.....	48
3.1 Pruebas de caja negra	49
3.1.1 Validación CU Autenticar Usuario	50
3.1.2 Validación CU Realizar evaluación teórica.....	53
3.1.3 Validación CU Mostrar resultados de evaluación.....	53
3.1.4 Validación CU Mostrar Videos.....	53
3.1.5 Validación CU Mostrar imágenes	53
3.2 Método Delphi.....	54
3.2.1 Selección de Expertos.....	54
3.2.2 Preguntas realizadas.....	55
3.2.3 Resultados del método Delphi.....	55
Conclusiones parciales	56
CONCLUSIONES	57
RECOMENDACIONES	58
Bibliografía.....	59
Glosario de Términos.....	¡Error! Marcador no definido.

INTRODUCCIÓN

Desde hace más de tres décadas podemos afirmar que estamos en la “Era de la Información y el Conocimiento”, pues resulta imposible mencionar algún sector de la sociedad actual en el que no esté presente el desarrollo tecnológico. En el sector educativo, las Tecnologías de Información y el Conocimiento (TIC) han posibilitado el desarrollo de las metodologías activas para el aprendizaje. Este proceso socio-tecnológico genera escenarios de innovación pedagógica sin precedentes; como los cambios en la gestión y distribución del conocimiento.

La utilización de los ambientes virtuales para el estudio de distintas ramas de la educación es de gran apoyo para el aprendizaje del estudiante, así como una opción viable ante la falta de recursos que impiden llevar a cabo de una forma tradicional los ejercicios que se simulan.

Se han realizado disímiles escenarios virtuales dentro de los que se puede mencionar los laboratorios virtuales de química, específicamente *VlabQ* que permite usar equipos y materiales presentes en un laboratorio de química para simular procesos tales como conservación de la materia, destilación simple y reversibilidad de las reacciones. Otros ejemplos de laboratorios virtuales de química son: *ChemLab*, *LiveChem*. También existen laboratorios de física como *Phet*, e *Interactive Physic* este último permite crear simulaciones como estudios del péndulo, cinemática y dinámica. En nuestra universidad también se han creado ambientes virtuales como es el caso de PROLAVI desarrollado por el Centro de Informática Industrial (CEDIN) perteneciente a la Facultad 5.

Recientemente, se conformó el proyecto PROTESIM, amparado por un convenio firmado entre la Universidad de las Ciencias Informáticas (UCI), la Facultad 5 y el Centro de Rehabilitación de Cara y Prótesis Bucomaxilofacial del Hospital CIMEQ. Este centro se especializa en el desarrollo de tecnologías para la construcción de prótesis bucomaxilofaciales. Los estudiantes de cuarto año de Estomatología y Tecnologías de la Salud que se especializan en este centro, tienen entre los objetivos de su plan de estudio, obtener habilidades en procedimientos básicos durante la elaboración de diferentes tipos de prótesis oculares. Como entrenamiento previo a situaciones reales, es necesario que los estudiantes adquieran un nivel de familiarización con estas tecnologías médicas que repercutan en menos gastos de insumos y un menor número de pérdidas materiales por errores humanos, al entrenarse en un Ambiente Virtual de Aprendizaje en 3D que simula

las situaciones reales sin los problemas antes mencionados y sin consecuencias al repetir las simulaciones tantas veces como sean necesarias.

En el transcurso de las actividades prácticas que ejecutan los estudiantes, se utilizan materiales costosos procedentes del mercado internacional, como puede ser el iris ocular artificial, cuyo precio unitario es de treinta y siete euros. Si se tiene en cuenta que ellos necesitan hacer prácticas de forma continuada para aprender sin errores el procedimiento correcto, se hace evidente el gasto económico que representa para el país, una simple práctica real ejecutada por un grupo promedio de veinticinco estudiantes. De aquí surge la necesidad de entrenar previamente en ambientes simulados a los estudiantes por medio de un Ambiente Virtual de Aprendizaje Interactivo (AVAI) para que pueda evaluarse la preparación previa de estos por parte del profesor, se adquirieran las habilidades necesarias en el ambiente virtual y se ejecuten posteriormente los procedimientos médicos reales previstos con una mayor eficiencia.

El AVAI en construcción aún no cuenta con una Interfaz gráfica de usuario desarrollada, por lo que se hace necesaria su implementación, la cual debe ser funcional e interactiva y que permita a los estudiantes interactuar con la aplicación, permitiendo la necesaria interacción hombre-computadora.

Dada la situación problemática anteriormente expuesta se define el siguiente problema científico: ¿Cómo lograr la interacción de estudiantes con Ambientes Virtuales de Aprendizaje 3D?

El problema planteado está contenido en el **objeto de estudio**: Diseño de interfaces para aplicaciones educativas. Se define, a partir de la situación problemática, como **objetivo general de la investigación**: Desarrollar una interfaz de usuario que permita una adecuada interacción con los estudiantes durante el aprendizaje en un ambiente virtual. Por consiguiente, el **campo de acción** se reduce a: Interfaz de usuario para ambientes virtuales de aprendizaje 3D.

El objetivo general: Desarrollar una interfaz de usuario que permita una adecuada interacción con los estudiantes durante el aprendizaje en un ambiente virtual.

Y para darle cumplimiento al objetivo se proponen las siguientes **tareas de investigación:**

- Análisis de interfaz en videojuegos y juegos serios.
- Análisis de buenas prácticas para el diseño de interfaces.
- Análisis del proceso de creación de una prótesis ocular.
- Captura de requisitos funcionales y no funcionales.

Métodos Teóricos:

- Analítico-Sintético, se aplicará para entender las Interfaces de Usuario (UI) a partir del análisis de las características y de su clasificación.
- Histórico-Lógico, posibilitará conocer los antecedentes y tendencias actuales de las UI a partir del análisis de su evolución y desarrollo actual.

Métodos Empíricos:

- Encuesta a especialistas, permitirá obtener información necesaria para la investigación y su posterior validación, debido a su experiencia; muchas veces no documentada.
- Encuesta a especialistas sobre los estándares de diseños internacionales, que nos permita obtener una guía para el desarrollo del sistema.
- Entrevistas a médicos sobre el procedimiento para creación de una Prótesis Ocular.

Estructura de la Tesis de Grado:

Capítulo 1. Fundamentación Teórica.

Capítulo 2. Características del Sistema.

Capítulo 3. Implementación y Validación.

CAPÍTULO 1: Fundamentación teórica y selección de tecnologías

1.1 Introducción

Para un mayor entendimiento del trabajo realizado se hace necesario el dominio de algunos conceptos tratados durante el desarrollo de la investigación a los cuáles se hace referencia a continuación.

1.2 Medio Ambiente de Aprendizaje

En términos generales, se puede decir que un ambiente de aprendizaje es el lugar donde confluyen estudiantes y docentes para interactuar psicológicamente con relación a ciertos contenidos, utilizando para ello métodos y técnicas previamente establecidos con la intención de adquirir conocimientos, desarrollar habilidades, actitudes y en general, incrementar algún tipo de capacidad o competencia (1).

Elementos esenciales en un ambiente de aprendizaje:

- Una serie de acciones reguladas relativas a ciertos contenidos.
- Un entorno o espacio en donde se llevan a cabo dichas actividades.
- Un proceso de interacción o comunicación entre sujetos.
- Un grupo de herramientas o medios de interacción.

1.3 Ambiente Virtual de Aprendizaje (AVA)

Se entiende por AVA al espacio físico donde las nuevas tecnologías, han rebasado el entorno escolar tradicional, favoreciendo al conocimiento y a la apropiación de contenidos, experiencias y procesos pedagógicos-comunicacionales. Están conformadas por el espacio, el estudiante, el asesor, los contenidos educativos, la evaluación y los medios de información y comunicación.

Los AVA siguen el mismo objetivo principal que sigue la pedagogía, el cual es aprender. Siguiendo sus mismas funciones pedagógicas: actividades de aprendizaje, situaciones de enseñanza, materiales de aprendizaje, apoyo y autorización, evaluación, entre otros. Además, incluyen características más específicas y la diferencia clave es el uso de herramientas de telecomunicación en el proceso enseñanza-aprendizaje (2).

Muchas aplicaciones multimedia, elaboradas con disímiles tecnologías informáticas, pueden considerarse como Ambientes Virtuales de Aprendizajes Interactivos. En nuestro

caso, estamos considerando solo aquellas aplicaciones informáticas que asumen estrategias y experiencias provenientes de la industria de los videojuegos y de la Realidad Virtual, dadas las potencialidades de aplicación en el área del aprendizaje en contextos educativos. Ejemplo de estas aplicaciones, podemos mencionar: simuladores de vuelos y de conducción, entrenadores médicos, programas de visualización científica, instrumentación virtual, los juegos serios y ambientes virtuales, donde la principal forma de interacción ocurre entre escenarios con objetos simulados en 2D o 3D.

1.4 Tipos de Ambientes Virtuales de Aprendizaje

Ambientes Virtuales de Aprendizaje remotos: Requieren de equipos servidores específicos que les den acceso a las máquinas a operar de forma remota, y no pueden ofrecer su funcionalidad ejecutándose de forma local.

Ambientes Virtuales de Aprendizaje Web: Se basan en un software que depende de los recursos de un servidor determinado.

Ambientes Virtuales de Aprendizaje software: Están desarrollados como un programa independiente y para ser ejecutados en los ordenadores, es decir su servicio no requiere de un servidor Web.

1.4.1 Ventajas de los Ambientes Virtuales de Aprendizaje

- Acerca y facilita a un mayor número de alumnos la realización de experiencias, aunque alumno y laboratorio no coincidan en el espacio. El estudiante accede a los equipos del laboratorio, pudiendo experimentar sin riesgo alguno, además, se flexibiliza el horario de prácticas y evita la saturación por el solapamiento con otras asignaturas.
- Reducen el coste del montaje y mantenimiento de los laboratorios físicos, siendo una alternativa barata y eficiente, donde el estudiante simula los fenómenos a estudiar como si los observase en el laboratorio real.
- Es una herramienta de auto aprendizaje, donde el alumno altera las variables de entrada, configuran nuevos experimentos, aprende el manejo de instrumentos, personaliza el experimento, etc. La simulación en el laboratorio virtual, permite obtener una visión más intuitiva de aquellos fenómenos que en su realización manual no aportan suficiente claridad gráfica. El uso de ambientes virtuales de

aprendizaje da lugar a cambios fundamentales en el proceso habitual de enseñanza.

- Los estudiantes aprenden mediante prueba y error, sin miedo a sufrir o provocar un accidente, sin avergonzarse de realizar varias veces la misma práctica, ya que pueden repetirlas sin límite; sin temor a dañar alguna persona, herramienta o equipo. Pueden asistir al laboratorio cuando ellos quieran, y elegir las áreas del laboratorio más significativas para realizar prácticas sobre su trabajo

1.5 Juegos Serios

El ejercicio continuado de un juego desarrolla las habilidades de ejecución concretas que forman parte de los objetivos pedagógicos del mismo. A la vez, debido a su carácter lúdico e interactivo, las posibilidades de reutilización aumentan dado que se pueden aplicar a ilimitados contextos de aprendizaje.

En el desarrollo de los juegos, el error desempeña un papel fundamental. El jugador no pierde ni arriesga nada, pues se encuentra en una situación simulada; y por este motivo se siente menos presionado al reaccionar ante situaciones o ejercicios que suponen un riesgo económico o humano. Siempre puede equivocarse sin sufrir consecuencias personales ni ante terceros. Uno de los principios fundamentales del juego mantiene que para aprender es necesario equivocarse. Por este motivo se pretende que el jugador, a través del método de ensayo-error, aprenda formas y técnicas de actuación eficientes (3).

El estímulo generado en el participante durante el transcurso del juego, propicia procesos de aprendizaje efectivos. Con el objetivo de ganar, el participante tiene la motivación suficiente como para aprender, dejarse llevar y mejorar en el desempeño del juego. Los discentes participantes, por lo tanto, pueden experimentar en un mundo virtual o imaginario el ejercicio de prácticas reales sin tener que enfrentarse a adversidades, ni sufrir consecuencias sobre sus actos. Así, cuando hayan concluido sus estudios, estarán más preparados para enfrentarse a problemas y situaciones reales reaccionando según la manera aprendida en el juego.

1.6 Ambiente Virtual de Aprendizaje Interactivo (AVAI)

Un AVAI es un espacio electrónico, donde se simulan en 2D o 3D situaciones problemáticas provenientes de un diseño de aprendizaje, donde deben cumplirse objetivos instructivos y educativos de un programa de estudio, en que estudiantes y

tutores colaboran en escenarios simulados y cumplen tareas experimentales o de entrenamiento, asumiendo un rol propio de su profesión y donde son evaluados por ello.

El elemento distintivo de los AVAI es que se manipulan los mismos objetos que en la experimentación real y pueden obtenerse los mismos resultados, donde el software brinda la información necesaria para realizar las transformaciones.

1.7 Creación de un Cuestionario Evaluativo

En el proceso de elaboración de un cuestionario se deben contemplar los siguientes puntos:

Definición del constructo o aspecto a medir

Antes de proceder a medir algo debemos tener una idea muy clara de lo que queremos medir. Ello puede requerir la realización de una revisión de la bibliografía y la consulta con expertos en la materia. Sean actitudes, conductas o conocimientos, se debe definir en forma clara y precisa.

Cuando se inicia el proceso de construcción de un cuestionario, se debe tener en cuenta la población a la que va dirigido (4).

Composición de los ítems

Los cuestionarios se componen de una serie de ítems. El ítem es la unidad básica de formación de un instrumento de evaluación, y generalmente consta de una pregunta y de una respuesta cerrada.

La definición de cada ítem ha de ser exhaustiva y mutuamente excluyente. Por otro lado, al formular la pregunta deben tenerse en cuenta factores como la comprensión (es necesario adaptar el lenguaje y el tipo de elección de respuestas al nivel sociocultural de los individuos a quienes va dirigido el cuestionario), así como la aceptabilidad para el sujeto que es preguntado.

Existen una serie de criterios para la redacción de las preguntas, que son los siguientes:

Utilizar preguntas breves y fáciles de comprender.

No emplear palabras que induzcan una reacción estereotipada.

Los ítems tienen definido un sistema de puntuación que pueden ser: simple o ponderado. Se dice que son ítems simples cuando la puntuación directa se obtiene con el sumatorio de respuestas acertadas o de los valores que se hayan dado a cada opción.

Se habla de ítems ponderados cuando el valor de cada opción de respuesta no es la misma o no se otorga el mismo valor a todos los aciertos.

Codificación de las respuestas

En función del número de opciones o tipo de respuestas, éstas pueden ser:

Dicotómicas: Sí/No, Verdadero/Falso.

Policotómicas: Estoy descontento conmigo mismo/ No me valoro/ Me odio/ Estoy satisfecho de mí mismo.

Proceso de validación

Una vez diseñado el borrador definitivo, es decir, una vez delimitada la información, formuladas las preguntas, definido el número de ellas que vamos a incluir en el cuestionario y ordenadas las preguntas, corresponde llevar a cabo la realización de la prueba piloto y la evaluación de las propiedades métricas de la escala.

Prueba piloto

Normalmente, se pasa el borrador del cuestionario a 30-50 personas, siendo aconsejable que se parezcan a los individuos de la muestra.

1.8 Tipos de preguntas a utilizar en un cuestionario

Para obtener la información deseada es necesario recurrir a varios tipos de preguntas:

Según el modo de formularse, las preguntas pueden ser cerradas y abiertas.

Preguntas cerradas:

Las cuestiones cerradas ofrecen al usuario que va a ser evaluado todas las alternativas posibles, o al menos todas aquellas que mejor responden a la situación que deseamos conocer.

Estas preguntas suelen ser de respuestas cortas, ya sean: sí o no, o: verdadero o falso por lo que se debe elegir alguna respuesta poniendo una señal convenida: una cruz, rodear con un círculo, subrayar o marcar.

Preguntas abiertas:

Las preguntas abiertas no ofrecen ninguna categoría para elegir. Sólo contienen la pregunta y no ofrecen ningún tipo de respuesta, dejando ésta a la consideración del usuario que completa el cuestionario.

1.9 La interfaz y las dimensiones de la provisión de estímulos sensoriales

Las nuevas tecnologías, cuando son aplicadas en un proceso de aprendizaje, se manifiestan como la interfaz que despliega información en forma de imágenes, textos, sonidos, etc. Su papel, en este caso, es la provisión de estímulos sensoriales. La provisión de estímulos tiene, como mínimo, dos dimensiones que deben considerarse en el diseño de ambientes virtuales de aprendizaje: la dimensión atencional y la dimensión motivacional.

- La dimensión atencional se refiere a la potencialidad que tiene la interfaz para centrar la atención del aprendiz en los estímulos relevantes. Esta potencialidad puede manifestarse a través de dos formas: al enfatizar los aspectos relevantes de la información o, al inhibir los ruidos e interferencias del entorno.
- La dimensión motivacional se refiere a la potencialidad de la interfaz para estimular o mantener la motivación del estudio hacia el aprendizaje (1).

Teniendo en cuenta la importancia que tiene la interfaz para la creación de ambientes virtuales de aprendizaje, a continuación se hace un profundo análisis de conceptos y buenas prácticas que se deben tener en cuenta para el diseño de interfaces.

1.10 Interfaz de usuario

La interfaz de usuario (IU), es uno de los componentes más importantes de cualquier sistema computacional, pues funciona como el vínculo entre el humano y la máquina. La interfaz de usuario es un conjunto de protocolos y técnicas para el intercambio de información entre una aplicación computacional y el usuario. La IU es responsable de solicitar comandos al usuario, y de desplegar los resultados de la aplicación de una manera comprensible. No es responsable de los cálculos de la aplicación, ni del almacenamiento, recuperación y transmisión de la información.

La IU de un programa es un conjunto de elementos de hardware y software de una computadora que presentan información al usuario y le permiten interactuar con la información. También se puede considerar como partes de las UI la documentación (manuales, ayuda, referencia, tutoriales) que acompaña al hardware y al software (5).

La IU es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el dispositivo, normalmente suelen ser fáciles de entender y fáciles de accionar (6).

Si se analizan conceptos anteriormente tratados sobre el término interfaz de usuario se puede determinar que una interfaz de usuario es la parte de la aplicación o sistema informático que da la primera impresión a la vista del usuario y con la que interactúa directamente en espera de una respuesta del programa. La misma actúa como un sistema comunicador entre el programa y el humano como si fuese un traductor entre ambos, mediante menús, íconos y botones el usuario dirige el funcionamiento de dicho sistema. En todo momento el usuario debe de estar al tanto de lo que está ocurriendo con el programa, por tanto la interfaz le debe de brindar a la persona toda la información, en tiempo real.

Clasificación de interfaces de usuario. Según su construcción, pueden ser de software o de hardware:

Interfaces de hardware: es un conjunto de controles o dispositivos que permiten la interacción hombre-máquina.

Interfaces de software: son programas o parte de ellos, que permiten expresar nuestros deseos al ordenador o visualizar su respuesta.

Atendiendo a como el usuario puede interactuar con una interfaz, se conocen varios tipos de interfaces de usuario:

Interfaces alfanuméricas: solamente presentan texto.

Interfaces táctiles: representan gráficamente un "panel de control" en una pantalla sensible que permite interaccionar con el dedo de forma similar a si se accionara un control físico.

Interfaces gráficas de usuario: permiten comunicarse con el ordenador de una forma muy rápida e intuitiva representando gráficamente los elementos de control y medida.

1.11 Interfaz gráfica de usuario

Las interfaz gráfica de usuario, surge dada la necesidad de hacer más simple el uso de los ordenadores para todo tipo de usuarios y no solo restringir el uso de estos a usuarios avanzados “Las GUI son una forma de representar la información procesada por la computadora de manera visual, concreta y manejable por el usuario final haciéndola más comprensible, más didáctica, más asimilable, más usable.” También existen otras definiciones como: “La interfaz gráfica de usuario es un tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Habitualmente las acciones se realizan mediante manipulación directa para facilitar la interacción del usuario con la computadora. La interfaz gráfica del usuario, dicho de forma más sencilla, es el conjunto de elementos gráficos (ventanas, menús, etc.) que permiten la interacción entre el usuario y una aplicación informática (7).

1.11.1 Componentes gráficos

Algunos de los componentes gráficos comunes en las interfaces gráficas son (8):

Apuntador: Símbolo que aparece en la pantalla y que se desplaza para seleccionar objetos y comandos.

Dispositivo apuntador: Dispositivo, como el ratón, que permite seleccionar objetos de la pantalla.

Escritorio: Es el área de la pantalla donde los íconos son agrupados.

Íconos: Pequeñas imágenes que representan comandos, archivos o ventanas.

Los íconos en el contexto de las interfaces gráficas son signos esquemáticos que representan algún tipo de fichero, carpeta, aplicación, o dispositivos de un sistema informático (9). Los íconos son elementos muy importantes dentro de las GUI ya que son idóneos para transmitir conceptos en menos espacio del que podría describir una palabra a través del lenguaje natural.

Menús. Opciones o comandos agrupados en forma de listas.

Los menús son listas de comandos, atributos, o cualquier tipo de elementos, agrupados de forma estructurada normalmente inscrito dentro de una barra de menús o de un área específica en la interfaz. Usualmente los menús esquematizan una organización de elementos de forma jerárquica por niveles, se representan en forma de listas, tanto de forma horizontal como vertical, a través de los mismos se accede a cada uno de los subelementos de cada elemento del menú. Los menús se pueden clasificar en menús contextuales, menús de navegación, menús jerárquicos y menús de inicio.

Ventanas. Son áreas que dividen la pantalla, y en cada una de ellas puede ejecutarse un programa o abrir un archivo diferente.

1.11.2 Características de las GUI

En la actualidad, una buena interfaz gráfica de usuario debe contar con un conjunto de características (10):

- Posee un dispositivo apuntador (típicamente un ratón).
- Promueve la consistencia de la interfaz entre programas.
- Sigue el paradigma de la interacción objeto-acción.
- Permite la transferencia de información entre programas.
- Se puede manipular en la pantalla directamente los objetos y la información.
- Provee elementos de interfaz estándar como menús y diálogos.
- Existe una muestra visual de la información y los objetos (íconos y ventanas).
- Proporciona respuesta visual a las acciones del usuario.
- Existe información visual de las acciones y modos del usuario/sistema (menús, paletas).
- Proporciona flexibilidad en el uso de dispositivos de entrada (teclado/ratón).

1.11.3 Ventajas del uso de las GUI

La asimilación del contenido por parte del usuario es rápida y clara: el usuario al acceder al sistema es capaz de familiarizarse de forma amena con el mismo, le da sensación de seguridad y realismo, permitiéndole una rápida comprensión (7).

- **Contenidos accesible:** la información que se muestra está disponible para que el usuario acceda a esta en el momento que lo necesite, sin grandes contratiempos.

- Usuarios sin experiencias pueden aprender el uso del sistema rápidamente: el usuario no tiene que ser experto en la materia para su interacción con el sistema.
- Retroalimentación garantizada: le permite al usuario navegar con más facilidad, siempre es posible volver atrás.
- Disponibilidad y portabilidad: se debe poder disponer de la información cuando se necesite, siempre que sea personal autorizado. Así mismo permite ser utilizado en múltiples ordenadores sin que necesiten modificaciones de importancia.

1.11.4 Diseño de Interfaces Gráficas de Usuario

El diseño elegido para un sistema debe ser el apropiado para la comunidad de usuarios a la que va dirigido y las tareas que van a realizar. Un mismo diseño puede ser apropiado para algunos usuarios y poco apropiado para otros, eficiente para algunas tareas e ineficiente para otras (11). Lo significativo es que los usuarios finales se sientan a gusto y seguros, o sea, que sientan que pueden controlar el sistema. El diseño debe invitar a los usuarios a explorar sus opciones y capacidades, permitiéndoles aprender a manejarlo fácilmente. Actualmente, el proceso del desarrollo de una interfaz se concibe como un ciclo que consta de 4 etapas. Mientras se disponga de tiempo, se deben de hacer tantos ciclos de mejoramiento como nos sea posible.

- Diseño.
- Implementación.
- Medición.
- Evaluación.

Diseñar la interfaz gráfica de usuario. Todos los elementos visuales se pueden hacer primero a mano y luego refinar con las herramientas adecuadas. Al diseñar una interfaz gráfica de usuario, es necesario especificar a qué audiencia va a llegar. En esta fase, el modelado del diseño orientado a objetos y los diagramas de entidad-relación pueden ser utilizados para transmitir y comprender cierta funcionalidad en la interfaz. Algunos de los temas necesarios para realizar el modelo del diseño son los lenguajes de programación que se utilizaran para desarrollar la interfaz y la cantidad de espacio físico de almacenamiento que se necesita. (12).

Implementación. Es interesante realizar un prototipo previo, una primera versión del programa que se realice rápidamente y permita visualizar el producto para ser probado

antes de codificarlo definitivamente. El nivel de sofisticación del prototipo debería incrementarse a lo largo del proceso de diseño de interfaces de usuario.

Medición y Evaluación. En esta etapa se prueban las funcionalidades de la interfaz y se tratan de diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo. En algunos casos, puede ser beneficioso solicitar a los usuarios reales que prueben el prototipo. A estas pruebas se les conoce como pruebas de aceptación, su objetivo es comprobar si el cliente está satisfecho con el producto desarrollado y si este producto cumple con sus expectativas, en términos de los errores que genera y de la funcionalidad que suministra.

En la siguiente tabla se expone un resumen de las diferentes actividades que se realizan en cada una de las etapas del desarrollo.

Diseño	<ul style="list-style-type: none"> • Análisis de Requerimientos del producto. • Análisis de las tareas. • Conocimiento del usuario. • Generación de posibles metáforas y análisis de tipo de diálogo. • Revisión de posibilidades para la implementación.
Implementación	<ul style="list-style-type: none"> • Generación de prototipos. • Desarrollo de la aplicación, sitio o sistema.
Medición (Test de usabilidad)	<ul style="list-style-type: none"> • Planificación (desarrollo del plan, definición de las medidas, selección de participantes, formación de observadores, preparación de los materiales). • Test (prueba piloto, tests con usuarios).

Evaluación	<ul style="list-style-type: none"> • Conclusión (análisis de los datos, elaboración del informe, resultados y recomendaciones). • Comparación contra estándares (internos y/o externos), versiones anteriores del mismo producto y productos competidores. • Verificación de las diferencias. • Generación de nuevas metas.
------------	---

Tabla 1 Resumen de las actividades de cada una de las etapas del desarrollo.

1.11.5 Principios para el diseño de interfaces gráficas

Una buena interfaz es aquella que le permite al usuario corregir los errores que pueda cometer, dándole una sensación de control, donde los usuarios ven rápidamente cómo alcanzar sus metas y comprenden cómo realizar su trabajo. Para el diseño de las interfaces GUI existen un conjunto de principios, a tener en cuenta para que dicha interfaz tenga un diseño adecuado y a su vez una buena aceptación por parte de los usuarios.

Anticipación: Las aplicaciones deben intentar anticiparse a las necesidades y deseos del usuario. No esperes que el usuario busque o recuerde información o herramientas. Muestra al usuario toda la información y herramientas necesarias para cada etapa en su trabajo (13).

Autonomía: No existe autonomía en ausencia de control; y el control no se puede tener sin información suficiente. Comunicar el estado es fundamental para que el usuario responda apropiadamente con la información disponible. El sistema siempre debe mantener informado al usuario acerca de lo que está pasando, a través de una retroalimentación apropiada dentro de un plazo razonable. La interfaz debe ser capaz de posibilitarle al usuario una idea aproximada del estado del sistema, de una forma sutil. Este principio es muy importante, ya que para lograr una interfaz gráfica de usuario apropiada y de buena tolerancia por parte de los usuarios a quienes va dirigida dicha interfaz, el usuario final debe sentirse sumergido en un ambiente flexible, donde sea capaz de aprender rápidamente el uso de la aplicación (13).

Percepción del color: Básicamente trata la idea de que se deben utilizar además de los colores convencionales en las IU, otros mecanismos secundarios para proveer la información a aquellos usuarios con problemas en la visualización de colores. Para nadie es un secreto que el color es uno de los elementos más complejos a la hora de intentar diseñar una interfaz gráfica correcta, además de ser una herramienta potente de comunicación si se usa de forma adecuada (14). La simplicidad es importante en el diseño de interfaces a color. Existe una simplicidad inherente en el color, la cual debería ser usada cuando se desarrolla el diseño. Los cuatro colores fisiológicamente primarios son el rojo, el verde y el azul. Estos colores son fáciles de aprender y recordar. Vinculando significados prácticos e intuitivos a estos colores simples.

Cuando se diseña una pantalla, el diseñador de la interfaz enriquece el desarrollo del usuario con un modelo mental efectivo (15). La claridad es también una pauta importante para usar color. Experimentos han mostrado que el tiempo de búsqueda para encontrar una pieza de información es disminuido si el color de esta pieza es conocido por anticipado, y si el color sólo se aplica a esa pieza. Los colores de interfaz estandarizados deberán de ser establecidos y usados a través del desarrollo de la GUI. El uso de colores claros y simples puede facilitar a los usuarios la búsqueda de piezas de información de forma rápidamente y eficientemente. El color ha probado ser superior al blanco y negro por la efectividad en el tiempo de proceso de información y por el rendimiento de memoria. La estética y lo atractivo de la interfaz son inherentemente aumentados por el uso del color.

Ley de Fitt: Esta es la ley más básica y conocida de las leyes del diseño de interfaces de usuario. Esta ley dice que cuanto más grande y más cercano al puntero del ratón es un objeto, más sencillo es el hacer *click* sobre él. Esta afirmación es de sentido común, pero muchas veces es ignorado completamente en el diseño de interfaces (16). Utiliza objetos grandes para las funciones importantes. Este principio exhorta a que los controles más utilizados en la aplicación, deben de ser los más grandes y de mayor distinción. Plantea, además, que nunca se deben colocar los controles a un píxel de distancia del borde de la pantalla.

Interfaces Explorables: De manera resumida y de una forma más clara, podemos decir que la GUI debe permitirle al usuario realizar la operación inversa de cualquiera acción que pueda efectuar dentro de la interfaz y que pueda implicar algún riesgo en el trabajo

que se esté realizando, ya que los usuarios a menudo cometen errores cuando utilizan el sistema, es por ello que la interfaz debe darle la posibilidad al usuario de explorar el sistema sin temor a cometer errores. Para dar cumplimiento a este principio es recomendable la anexión de un comando para que el usuario pueda deshacer todas las operaciones que realice, esto sustituiría la necesidad de contar con diálogos de confirmación para cada acción que se realice en el sistema. No se debe de encerrar al usuario en un único camino, aunque se le debe de guiar por el mejor de los caminos, ya que esto le facilita a los nuevos usuarios o aquellos que solo quieren terminar de desarrollar alguna actividad dentro del sistema, hacerlo de una forma rápida y con el mínimo de esfuerzo, aunque siempre hay que dar posibilidades abiertas a los usuarios que deseen explorar el sistema (5).

Legibilidad: En este principio deben tenerse en cuenta algunos aspectos como son: la consistencia y la tipografía. Los comandos y menús deben de tener el mismo formato. La consistencia es un factor importante dentro de la legibilidad ya que las interfaces consistentes reducen el tiempo de aprendizaje del usuario. El tamaño de las fuentes debe ser lo suficientemente grande como para poder ser leído en monitores estándares y es importante hacer clara la presentación visual de objetos. Se deberá usar un número reducido de tipos de letra que deben ser legibles, claros y singulares. Usar colores de gran contraste para el texto y el fondo (17). No debe mezclarse más de dos o tres tamaños de fuente y estilos (subrayado, negrita, cursiva) en una ventana, debido a que esto se verá poco profesional y distraerá al usuario de la información que se transmite.

Consistencia: A menudo, algunos objetos en la interfaz se consideran visibles, porque no parecen controles y es casi imposible que el usuario descubra que se puede interactuar con esos objetos. Por lo tanto, si se decide utilizarlos, deberá aparecer muy claro su significado. La interfaz de usuario se mantiene en concordancia con el ambiente de trabajo (18).

Arquitectura de la Información: La Arquitectura de Información(AI) tiene como propósito la búsqueda de los mejores diseños para la presentación de la información y su posterior comprensión, así como la usabilidad, que estudia el conjunto de características del diseño y funcionamiento de una interfaz de usuario, para obtener una correcta operación y comprensión de los contenidos, son disciplinas cuya actividad está dirigida a lograr la máxima satisfacción del usuario durante el proceso de interacción con los productos de

información (19). Con respecto a la AI, Wurman se refirió a ella como: "... La ocupación profesional emergente del siglo XXI, dedicada a las necesidades de la era, enfocada a la claridad, entendimiento humano y la ciencia de organización de la información...". Edward Tufte, por su parte, la define como "... el diseño de la presentación de la información para facilitar su entendimiento...". La arquitectura de información se ocupa del diseño estructural de los sistemas de información, su problema central es la organización, recuperación y presentación de información mediante el diseño de ambientes intuitivos (20).

Usabilidad: El grado de usabilidad de un sistema interactivo es un aspecto relacionado con la interfaz de usuario y es inversamente proporcional al tiempo que malgastan los usuarios de dicho sistema intentando averiguar el alcance de qué hace o dónde está una determinada funcionalidad. Esta característica hace, por tanto, referencia a la rapidez y facilidad con que las personas llevan a cabo sus tareas a través del uso del producto con el que están trabajando. Un producto se considera fácil de aprender a usar en términos del tiempo que toma el usuario para llevar a cabo su objetivo, el número de pasos que tiene que realizar para ello y el éxito que tiene en predecir la acción apropiada para llevar a cabo. Para desarrollar productos usables hay que entender los objetivos del usuario, hay que conocer los trabajos y tareas del usuario que el producto automatiza o modifica (21).

La Organización Internacional para la Estandarización ofrece dos definiciones de usabilidad.

La usabilidad se refiere a la capacidad de un software de ser comprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso. (ISO/IEC 9126).

La usabilidad es la eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios en un contexto de uso específico. (ISO/IEC 9241).

En la definición podemos observar que la usabilidad se compone de dos tipos de atributos:

Accesibilidad: Un concepto íntimamente ligado al de usabilidad es el de accesibilidad. Se refiere a la facilidad de uso, sino a la posibilidad de acceso. En concreto a que el diseño, como prerequisite imprescindible para ser usable, posibilite el acceso a todos sus usuarios, sin excluir a aquellos con limitaciones individuales, discapacidades, dominio del

idioma o limitaciones derivadas del contexto de acceso (hardware). La accesibilidad es la cualidad de accesible, un adjetivo que se refiere a aquello que es de fácil acceso, trato o comprensión. El concepto se utiliza para nombrar al grado en que todas las personas, más allá de sus capacidades físicas o técnicas, pueden utilizar un cierto objeto o acceder a un servicio (22). Los seres humanos son diferentes entre sí y lo ideal sería que todas las interfaces de usuario pudieran acomodarse a esas diferencias, de tal modo que cualquier persona fuera capaz de utilizarlas sin problemas, sin que nadie se vea limitado en el uso de algo por causa de esas diferencias personales. Es necesario evitar diseñar solamente atendiendo a características de grupos de población específicos, imponiendo barreras innecesarias que podrían ser evitadas prestando más atención a las limitaciones de estos. El lenguaje de la interfaz debe ser claro, sencillo y directo, adaptado al nivel comprensivo del usuario.

Atributos cuantificables de forma objetiva: como son la eficacia o número de errores cometidos por el usuario durante la realización de una tarea, y eficiencia o tiempo empleado por el usuario para la consecución de una tarea.

Atributos cuantificables de forma subjetiva: como es la satisfacción de uso, medible a través de la interrogación al usuario, y que tiene una estrecha relación con el concepto de Usabilidad Percibida.

Facilidad de aprendizaje: Cuán fácil es aprender la funcionalidad básica del sistema y ser capaz de realizar correctamente la tarea que desea realizar el usuario. La facilidad de aprendizaje se mide normalmente por el tiempo empleado en el sistema hasta ser capaz de realizar ciertas tareas en menos de un tiempo dado, comparado por el tiempo empleado habitualmente por los usuarios expertos. Este atributo es muy importante para usuarios nóveles. Son los usuarios y no los diseñadores o los desarrolladores, los que determinan cuándo un producto es fácil de usar.

Eficiencia: El número de transacciones por unidad de tiempo que el usuario puede realizar usando el sistema. Este atributo busca la máxima velocidad de realización de tareas del usuario. Cuanto mayor es la usabilidad de un sistema, más rápido es el usuario al utilizarlo, y el trabajo se realiza con mayor rapidez. Nótese que eficiencia del software en cuanto a su velocidad de proceso no implica necesariamente eficiencia del usuario.

Recuerdo en el tiempo: Para usuarios intermitentes (que no utilizan el sistema regularmente) es vital ser capaces de usar el sistema sin tener que aprender cómo funciona partiendo de cero cada vez. Este atributo refleja el recuerdo acerca de cómo funciona el sistema que mantiene el usuario, cuando vuelve a utilizarlo tras un período de no utilización.

Tasa de errores: Este atributo contribuye de forma negativa a la usabilidad de un sistema. Se refiere al número de errores cometidos por el usuario mientras realiza una determinada tarea. Un buen nivel de usabilidad implica una tasa de errores baja. Los errores reducen la eficiencia y satisfacción del usuario, y pueden verse como un fracaso en la transmisión al usuario del modo de hacer las cosas con el sistema.

Satisfacción: Este es el atributo más subjetivo. Muestra la impresión subjetiva que el usuario obtiene del sistema.

1.12 Interfaz gráfica en videojuegos

La interfaz gráfica de usuario en los videojuegos es un elemento de diseño muy importante, ya que su descuido puede traducirse en una frustración para el jugador y dejar de lado un juego bien desarrollado o al contrario esta puede sumergir al jugador aún más en la ambientación, potenciar la identidad del juego e incluso facilitar los procesos de aprendizaje.

La interfaz gráfica puede tener diferentes plataformas, diferentes tipos de interacción, aspectos estéticos y visuales., HUD, tipos de controles, esquema de controles, íconos, botones, contadores y menús.

Elementos visuales como los HUDs y los contadores juegan un papel fundamental, pues muestran al jugador información que le ayuda a tomar decisiones en las diferentes situaciones de juego (23) .

1.13 Motor Gráfico para el desarrollo

Como motor gráfico se utilizó Shiva3D, este es un motor con un editor gráfico diseñado para crear fácilmente aplicaciones y videojuegos para la web, consolas y dispositivos móviles en tiempo real. Este editor es de tipo WYSIWYG, acrónimo de: "lo que ves es lo que obtienes".

Además contiene un editor que permite importar los recursos de 3DMax, Blender, Maya y MakeHuman, tales como texturas, sonidos y películas. Así como configurar los materiales, partículas, animación, HUDs, terreno y mar de la escena (24).

Shiva3D puede producir juegos en 3D y simulaciones gráficas para Windows, Mac, Linux, iPhone, IPAD, Android, Palm OS, Wii y WebOS, independientes o integrados en la mayoría de los navegadores web. El motor del juego utiliza OpenGL y DirectX, además es compatible con los estándares como NVIDIA PhysX, Mod y ARToolKit.

Características más notables:

- Motor de juego multiplataforma.
- Ofrece funciones de programación, renderizado, animación y efectos especiales.
- Herramienta unificada de edición, compila código fuente generado por el editor en una aplicación ejecutable.
- Utiliza lenguaje Lua.

- Motor gráfico con optimización de iluminación dinámica y sombras.

1.13.1 Posibilidades de exportación

Entre las posibilidades que brindan algunos motores gráficos, se encuentra la presencia de una aplicación encargada de exportar la solución implementada. En el caso del motor gráfico ShiVa3D, se cuenta con la herramienta *Shiva3D Authoring Tools* que ofrece posibilidades de exportar a varios sistemas operativos, dispositivos Wii, dispositivos móviles y la Web. Es muy importante tomar en cuenta estas funcionalidades brindadas por el motor propuesto, ya que permite de una manera muy sencilla y sin necesidad de re-implementar, modificar su proyecto de acuerdo a los requerimientos de hardware y software, e incluso ofrecer exportado a Web, donde solo se necesitaría del uso de un navegador que bien puede venir con su sistema operativo.

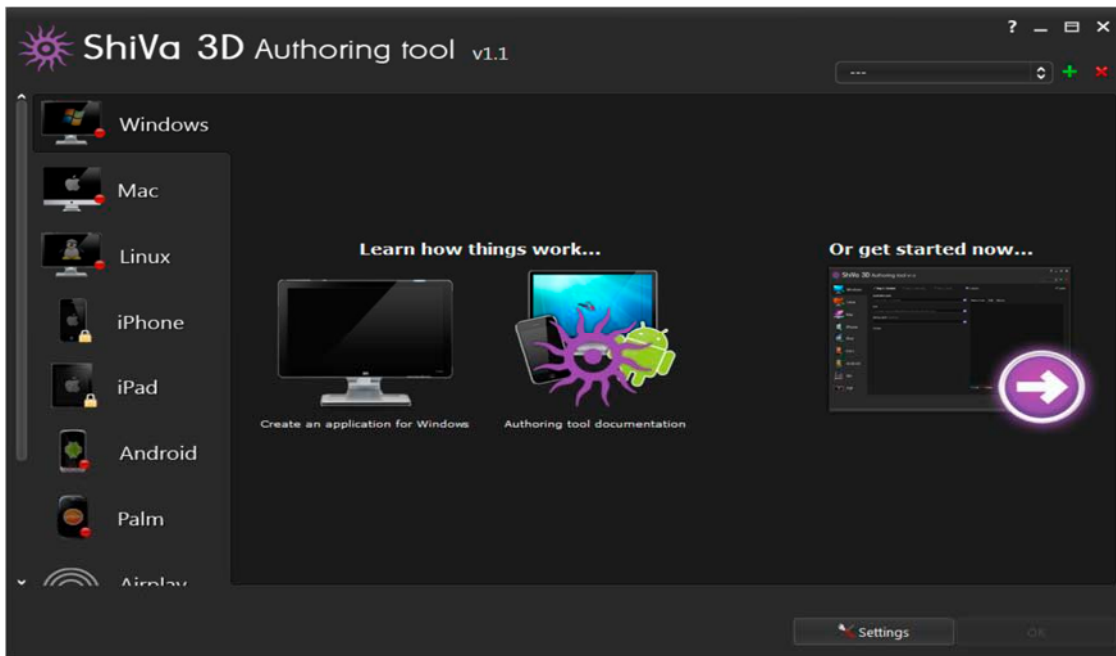


Imagen 1 Facilidades de Exportación de ShiVa3D.

1.14 Lenguaje de programación

Un lenguaje de programación es un dialecto artificial trazado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden utilizarse para crear sistemas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Constituyen además una técnica de comunicación mediante el cual el programador le entrega las instrucciones al computador. Pueden clasificarse teniendo en

cuenta distintos criterios: lenguajes interpretados, lenguajes compilados, lenguajes de script entre otros. Dentro de los lenguajes de programación destacan C++, C#, Java, Lua, entre otros (25).

1.14.1 Lenguaje de programación LUA

Lua es un lenguaje de programación imperativo y estructurado, potente, ligero y muy rápido, está considerado como uno de los lenguajes de más reputación por su rendimiento. Varios indicadores muestran Lua como el lenguaje más rápido de interpretar (26).

Este es un lenguaje fácil de aprender, la sintaxis es simple y clara. Uno puede leer y entender pequeños scripts sin saber nada del lenguaje. Por lo tanto, es posible utilizarlo como el idioma de los archivos de configuración que puede ser editado por personas que no saben nada acerca de programación. Permite una gran libertad y flexibilidad al programar, además rápidamente puede compilar y ejecutar secuencias de comandos con miles de líneas de código. La máquina virtual de Lua es de un poco más de 160 KB, y agregando el compilador sólo otros 200 KB. Este es un lenguaje de extensión, suficientemente compacto para usarse en diferentes plataformas. Su semántica puede ser extendida y modificada redefiniendo funciones de las estructuras de datos utilizada, además ofrece soporte para funciones de orden superior, colector de basura.

1.15 Metodologías de desarrollo de software

Un proceso de software detallado y completo suele denominarse “Metodología”. Las metodologías se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental, entre otros). Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, entre otros (27).

1.15.1 El Proceso Unificado de Desarrollo de Software (RUP)

RUP es una metodología de software que permite el desarrollo de aplicaciones a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad.

El proceso de desarrollo constituye un marco metodológico que define en términos de metas estratégicas, objetivos, actividades y artefactos requeridos en cada fase de

desarrollo. Esto permite enfocar esfuerzo de los recursos humanos en términos de habilidades, competencias y capacidades a asumir roles específicos con responsabilidades bien definidas. RUP es una de las metodologías robustas o pesadas, que presenta entre sus características ser un proceso de desarrollo orientado a objetos, utiliza el Lenguaje Unificado de Modelado (UML) como lenguaje de representación visual. Este proceso unificado define “Quién”, “Cómo”, “Cuándo” y “Qué” debe hacerse en el proyecto. Tiene tres características fundamentales: es iterativo e incremental, centrado en la arquitectura y dirigido por casos de usos.

Dirigido por casos de uso: Los casos de uso representan los requisitos de software capturados durante el flujo de trabajo de requisitos, la planificación del proyecto se hace en términos de casos de uso, los desarrolladores crean realizaciones de casos de uso en términos de clases y subsistemas, los componentes se incorporan en los incrementos y cada uno realiza un conjunto de casos de uso, y por último se verifica que el sistema implementa los casos de uso correctos para el usuario.

Centrado en la arquitectura: La arquitectura representa la forma del futuro sistema en términos de vistas arquitectónicas, sobre la cual el equipo de desarrollo y usuarios deben estar de acuerdo, ya que estas describen los elementos del modelo más importantes para su desarrollo, la arquitectura va madurando en las interacciones comenzando con los casos de uso relevantes desde el punto de vista arquitectónico.

Iterativo e incremental: “El Proceso Unificado propone que cada fase se desarrolle en iteraciones, ya que el incremento en la complejidad de los sistemas actuales hace que sea factible dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta un incremento. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros”. El período de vida del software esta particionado en ciclos, cada ciclo consta de cuatro fases: concepción o inicio, elaboración, construcción y transición y cada vez que termina un ciclo se produce una versión del sistema. Es ideal para proyectos cuyos requisitos no son variables y para grandes equipos de desarrollo. Sin embargo puede adaptarse a diferentes condiciones. En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo, los 6 primeros son flujos de ingeniería y los tres últimos de apoyo.

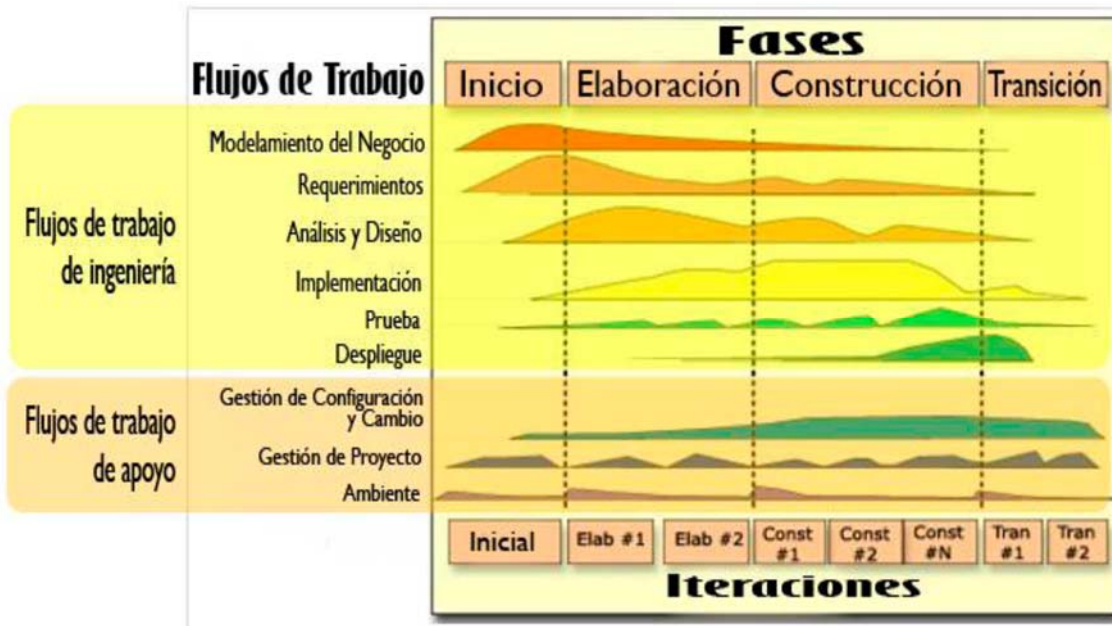


Imagen 2 RUP (Fases, flujos de trabajo e iteraciones).

1.16 Lenguaje Unificado de Modelado (UML)

UML es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos. Este contiene diagramas que permiten la modelación de los diferentes componentes del sistema. Es utilizado en varias metodologías aunque presenta una estrecha relación con la que será utilizada en este trabajo de diploma: RUP, ya que este hace uso de todos los diagramas propuestos por este lenguaje. Con UML se fusiona la notación de estas técnicas para formar una herramienta compartida entre todos los ingenieros software que trabajan en el desarrollo orientado a objetos, de igual forma, los clientes, desarrolladores y otras personas involucradas o interesadas, pueden comprender el funcionamiento y la estructura de un sistema determinado. UML agrupa los diagramas en tres tipos diferentes (28).

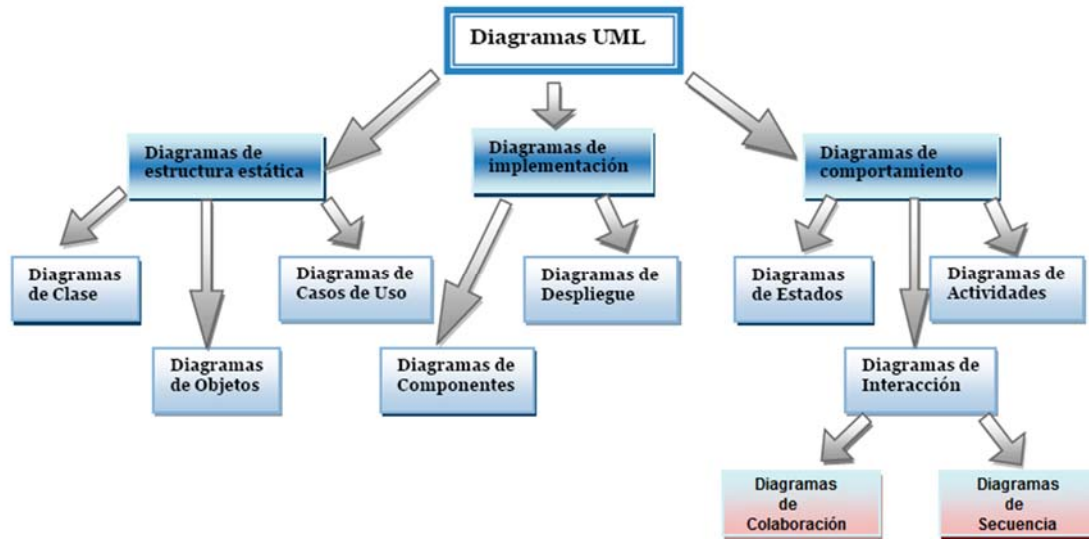


Imagen 3. Diagramas UML.

1.17 Herramientas CASE

Las herramientas CASE son un conjunto de programas y ayudas que brindan asistencia técnica a analistas, ingenieros de software y desarrolladores para el análisis de requisitos, modelado visual y documentación durante parte o todo el ciclo de vida de un proyecto de software. La selección de esta herramienta está estrechamente relacionada con la metodología de desarrollo de software y el lenguaje de modelado a utilizar.

1.17.1 Visual Paradigm for UML Enterprise Edition

El Visual Paradigm para UML es una potente plataforma, la cual está diseñada para una amplia gama de usuarios, incluidos los analistas de sistemas o ingenieros de software. Permite representar todo tipo de diagramas UML para las distintas fases como la captura de requisitos, análisis, diseño e implementación. Presenta una serie de ventajas tales como la generación del código fuente en Java, C#, C a partir de diagramas de clases. Además permite aplicar ingeniería inversa en los lenguajes Java, C++, CORBA IDL, PHP, XML, Ada y Python. Algunas de las características que presenta son:

- Navegación intuitiva entre el modelo visual y el código.
- Modela todos los diagramas de UML.
- Validación de modelos en tiempo real.
- Presenta recursos centrado en la interfaz para mejorar la usabilidad.

- Diagrama de diseño automático.
- Permite exportar diagramas como imagen en el formato JPG, PNG y SVG.
- Presenta sub-diagramas de apoyo para todos los modelos UML.
- Diseño centrado en caso de uso y enfocado al negocio que le permite generar un software con mayor calidad.

Una de las ventajas distintivas que presenta esta herramienta CASE es que apoya la sincronización del código Java, lo que le permite la generación de código en este lenguaje de programación a partir de modelos. Esta herramienta CASE es una de las que soporta el análisis textual, una técnica que se utiliza para la captura de requisitos. Una característica fundamental que presenta esta herramienta es la disponibilidad de múltiples plataformas: es soportada tanto en el Sistema Operativo Windows como en el GNU/Linux.

1.18 Conclusiones parciales

En el presente capítulo se realiza un recorrido por los diferentes conceptos asociados a los AVA, AVAI y juegos serios.

Se abordan conceptos de temas referentes a la arquitectura de la información, accesibilidad y usabilidad, los cuales se pretenden incluir en el diseño de la interfaz, atendiendo a la tipografía que se utilizará y a la distribución de la información. Se presentó un estudio de algunos de los principios para el diseño de las interfaces gráficas de usuario, ya que se procura orientar la propuesta al cumplimiento de estos principios, dígase interfaces explorables, ley de Fitt y percepción del color. Se identificó la metodología, las herramientas, el motor gráfico y el lenguaje de programación. Específicamente la metodología de desarrollo RUP, la herramienta CASE Visual Paradigm for UML, lenguaje de programación Lua y como motor gráfico Shiva3D.

CAPÍTULO 2: Características del Sistema

2.1 Introducción

En este capítulo se dará una breve panorámica de la solución que se propone para el problema que encierra este trabajo de diploma, además se hace una selección de los elementos investigados en el capítulo anterior para aplicarlos a la solución propuesta.

Se realiza el proceso de análisis y diseño, el cuál va a ser representado mediante artefactos tales como los diagramas de secuencia y diagramas de casos de uso.

2.2 Elementos tomados en cuenta para la propuesta de solución

La investigación tiene como propósito principal lograr el desarrollo de una interfaz de usuario, que sea adaptable para el AVAI. Una interfaz de usuario es la parte del sistema con que el usuario interactúa y la que le brinda el acceso y el control a todas las funcionalidades e información que brinda la aplicación informática, por lo que el diseño tiene que estar orientado al tipo de usuario que va a interactuar con la misma, en este caso, por lo general son estudiantes. Las etiquetas son utilizadas para mostrar información al usuario de manera uniforme y el texto que contiene siempre comienza con letra inicial mayúscula.

Al analizar los distintos tipos de interfaz de usuario que existen, así como las buenas prácticas para el desarrollo de las mismas, se seleccionó la interfaz gráfica de usuario debido a que en la misma está plasmada el uso de metáforas, que son de gran utilidad para transmitir con pocas palabras al usuario, lo que este debe de hacer en cada momento.

El tamaño de los gráficos (metáforas) e íconos mostrados en la misma son fáciles de visualizar por el usuario, estos son usados de manera consistente en todas las escenas, dándole uniformidad al sistema y contribuyendo a que se agilice el proceso de aprendizaje de los estudiantes.

Los cuestionarios que se muestran en la aplicación fueron creados por expertos en el tema y teniendo en cuenta elementos claves como la composición de los ítems o preguntas y la codificación de las respuestas.

Al analizar las preguntas que se formulan se decidió seleccionar preguntas cerradas y respuestas dicotómicas, debido que las respuestas que brindará el estudiante serán de Verdadero (V) o Falso (F).

Un aspecto relevante tenido en cuenta en el diseño de la interfaz de usuario fue la opinión del cliente, pues la opinión del usuario que va a utilizar el sistema es lo más importante para su éxito. Esto fue fundamental en elementos como, los colores e imágenes.

Propuesta de solución:



Imagen 4 Menú principal



Imagen 5 Entrada de datos.



Imagen 6 Selección de Roles.

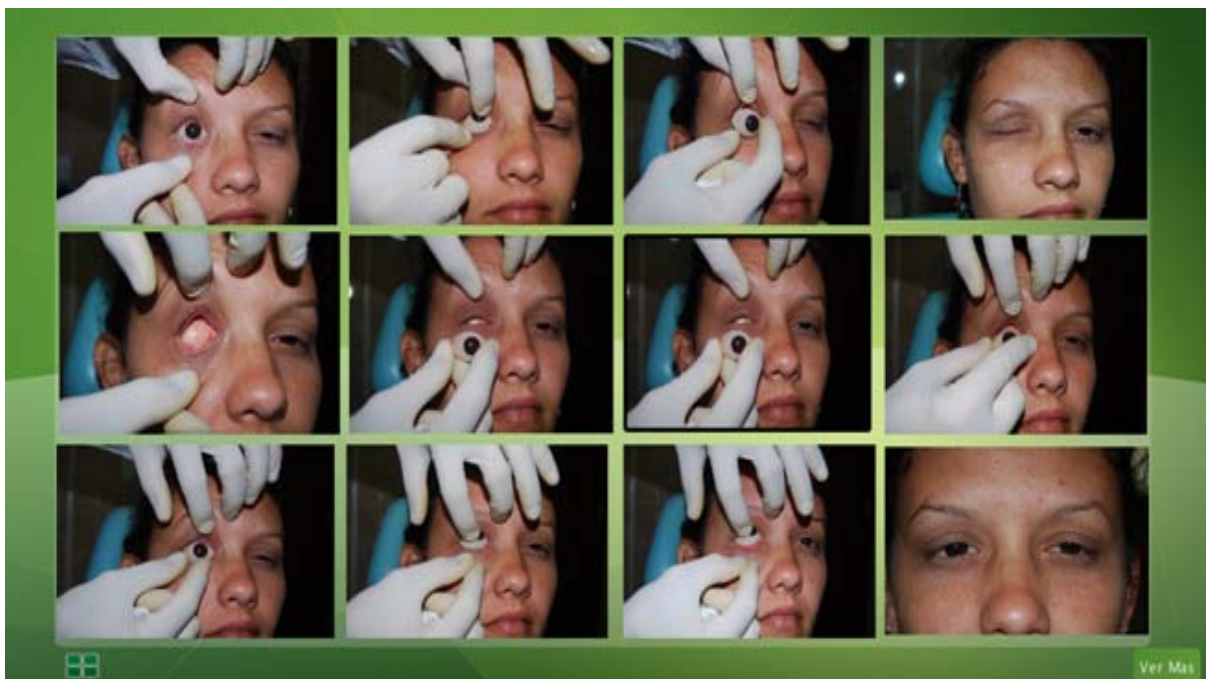


Imagen 7 Imágenes de Ayuda.

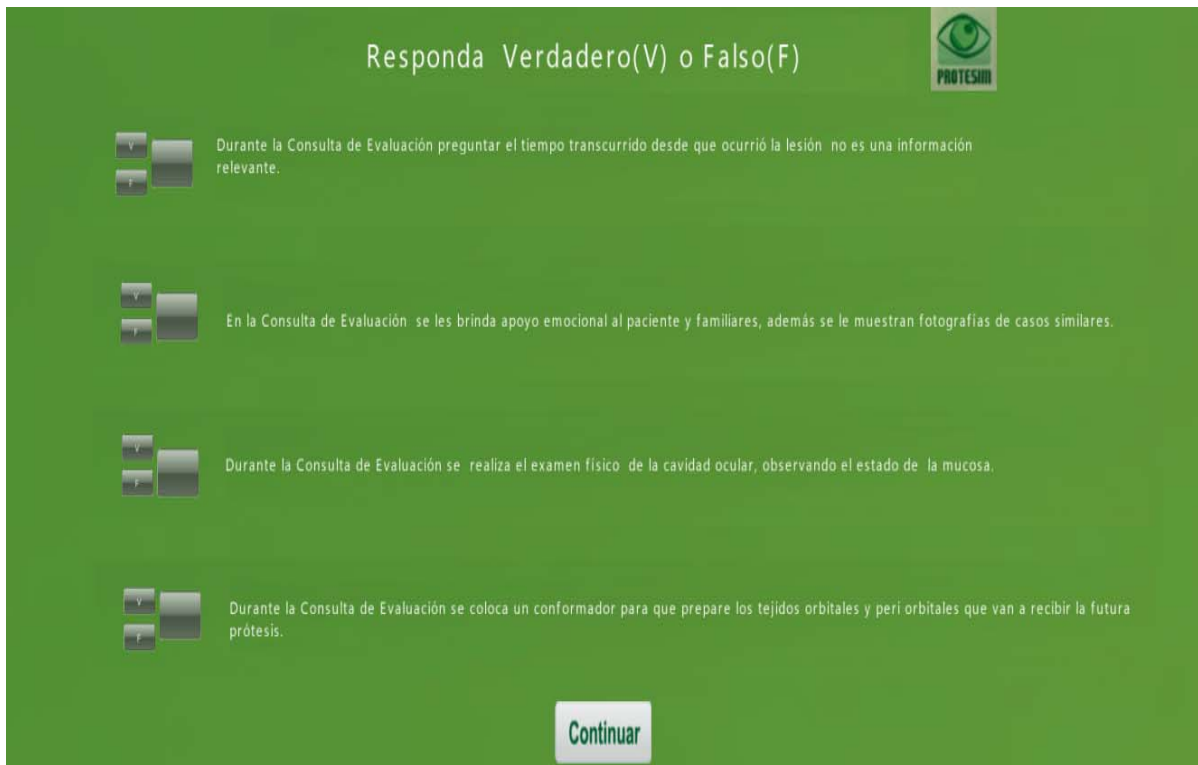


Imagen 8 Cuestionario evaluativo.

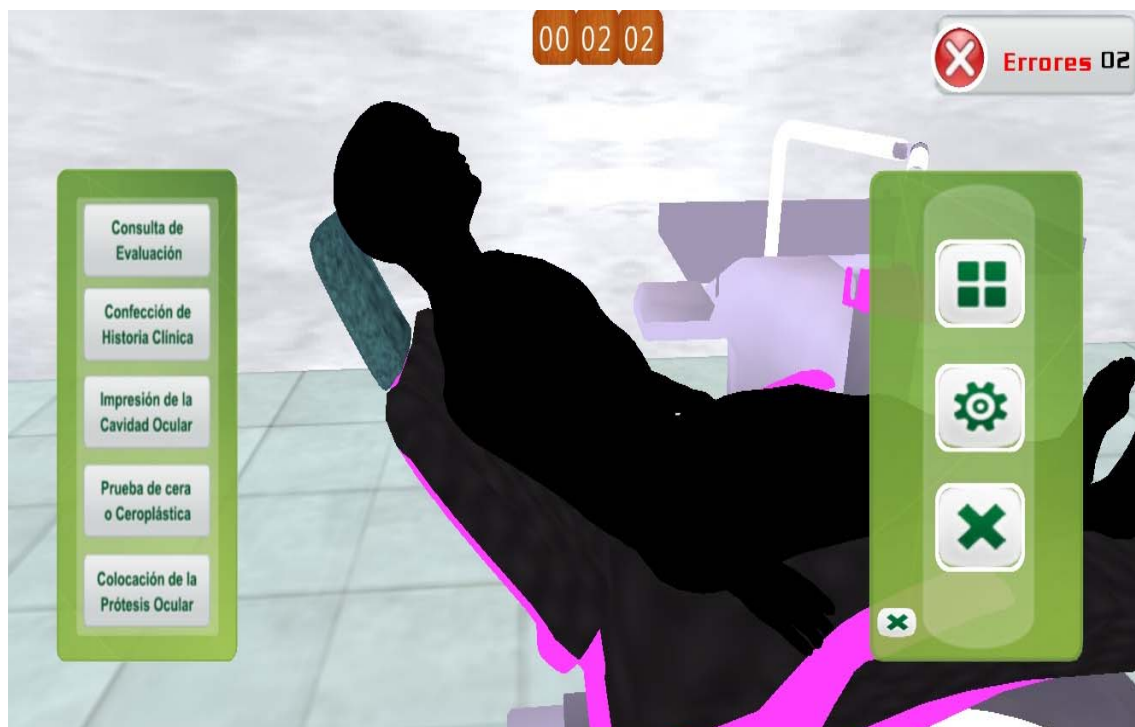


Imagen 9 Escena de trabajo.

2.3 Funcionalidades

La GUI propuesta, cuenta con un menú principal que funciona como un punto de partida y desde el cual el usuario puede navegar con facilidad hacia toda la aplicación.

Desde el menú principal, el estudiante podrá acceder a imágenes y videos.

Al seleccionar la opción de comenzar la práctica de simulación, el usuario deberá insertar su usuario y contraseña con el que se identifica, posteriormente deberá seleccionar el rol en el cuál se desempeñará.

Al seleccionar el rol Clínico el usuario accederá a diferentes cuestionarios con preguntas relacionadas con los diferentes procedimientos por los que debe transitar durante la práctica de simulación. Si el estudiante responde correctamente el 75% de las preguntas por cada proceder, entonces podrá acceder a la práctica de simulación y posteriormente ver sus resultados. Si las respuestas correctas son menores del 75% entonces se le mostrará un panel en el cuál visualizará la cantidad de errores cometidos en cada proceder, además se le brindarán opciones para salir de la aplicación y de volver al menú inicial.

Mientras el usuario se desempeña en el rol Clínico, si es presionada la tecla “ESC” se mostrarán paneles de los laterales, del lateral izquierdo un panel en el que se muestran los diferentes procedimientos del rol Clínico que estarán bloqueados o desbloqueados. En dependencia si el proceder fue realizado o no y del lateral derecho un panel en el que se le brinda al usuario un conjunto de acciones tales como, salir al menú principal, salir de la aplicación y una ayuda al usuario en la que se le muestra al usuario las teclas que debe presionar en caso de tener alguna duda, además se mostrará el tiempo que transcurre durante la práctica de simulación y los errores que va cometiendo. La forma de mostrar estos elementos fundamentales para el avance del estudiante en la práctica de simulaciones es semejante a como se muestra en muchos videojuegos actuales, y aunque no se está construyendo un videojuego, si se tienen en cuenta estos elementos para lograr una mayor aceptación por parte de los usuarios finales, que en este caso serán estudiantes.

Al finalizar la práctica de simulación se mostrará una pantalla en la que se visualizarán los errores cometidos por cada proceder, además de una puntuación proveniente de las respuestas del cuestionario evaluativo.

2.4 Modelo de Dominio

El negocio se representa mediante un Modelo de Dominio, ya que es fácil de interpretar el funcionamiento del sistema y la simplicidad del entorno donde está enmarcado. Se utiliza el modelo de dominio ya que es un subconjunto del modelo del negocio y se utiliza cuando las fronteras de negocio no están bien definidas. El modelo de dominio es una representación visual de las clases conceptuales del mundo real. Se debe concebir como un diccionario visual de abstracciones que será utilizado en fases posteriores.

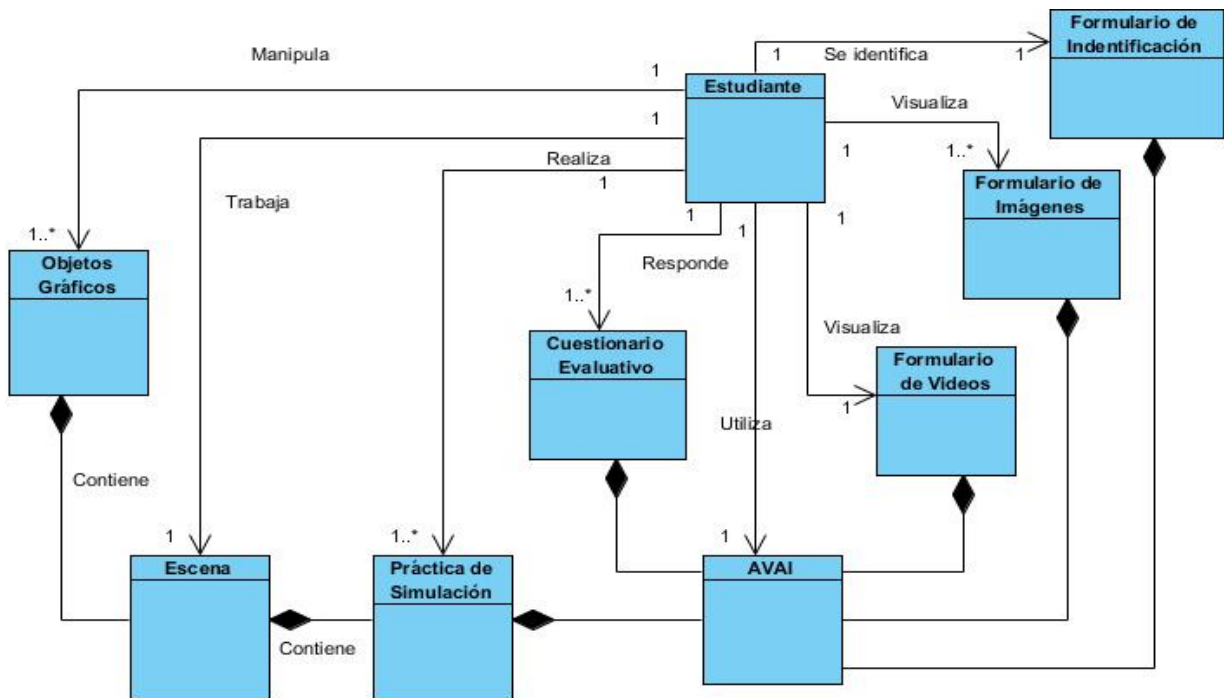


Imagen 10 Diagrama modelo de dominio.

2.4.1 Descripción del modelo de dominio

Estudiante: Es el usuario que está capacitado para realizar la práctica de simulación.

Práctica de simulación: Constituye el ambiente de trabajo con el que va a interactuar el usuario.

Formulario de videos: Podrá ser visualizado por los estudiantes.

Formulario de imágenes: Podrá ser visualizado por los estudiantes.

Cuestionario evaluativo: Realiza la evaluación de los estudiantes.

Escena: Es el lugar donde se van a ubicar los objetos para interactuar con ellos, y realizar la práctica de simulación.

Objetos gráficos: Podrán ser utilizados por los estudiantes durante la práctica de laboratorio.

Formulario de identificación: Validará la entrada de los estudiantes a la aplicación.

2.5 Levantamiento de requisitos del sistema

Este flujo de trabajo es de vital importancia para el desarrollo de una aplicación informática, pues es aquí donde se deja claro, lo que se quiere hacer, y lo que el cliente desea ver en el sistema que se va a desarrollar. El mismo tiene como misión convertir el problema expresado, en términos de dominio del negocio a soluciones descritas en lenguaje de dominio de la tecnología de información. Consiste en un conjunto de actividades de Ingeniería de software que se ocupa de recoger las necesidades de los clientes y los usuarios, sobre un sistema dado y traducirlas a especificaciones técnicas del sistema. Los requisitos se clasifican en requisitos funcionales y no funcionales, y constituyen el contrato que se debe de cumplir. La interfaz de usuario del ambiente virtual de aprendizaje en construcción debe brindar la posibilidad de que el sistema cumpla con las expectativas, y los requisitos funcionales que se mencionarán a continuación.

2.5.1 Requisitos funcionales

RF1. Autenticar usuario.

RF2.Mostrar imágenes.

RF3. Mostrar videos.

RF4.Realizar evaluación teórica.

RF5.Mostrar resultados de la evaluación.

2.5.2 Requisitos no Funcionales

Los requisitos no funcionales son aquellos que especifican los criterios que pueden usarse para juzgar la operación de un sistema, no su comportamiento específico, por lo tanto, se puede definir requisito no funcional aquel requisito que no describe una funcionalidad dentro del sistema, sino cualidades con los que debe cumplir el mismo. Los requisitos no funcionales, se clasifican en varios grupos.

1. Apariencia e interfaz externa

- Se debe proporcionar un ambiente agradable y fácil para el usuario.

2. Requisitos no funcionales de Restricciones en el Diseño e Implementación

- La aplicación se desarrollará utilizando ShiVa3D como motor gráfico.
- Se utilizará LUA como lenguaje de desarrollo.
- En el trabajo con gráficos tridimensionales se utilizará la herramienta Blender.

3. Hardware

Requerimientos mínimos:

- Intel Pentium IV 2Ghz o AMD ATLON XP 2600+.
- 512 megabytes (MB) de RAM.
- Tarjeta gráfica con 64MB mínimo de RAM.
- Resolución 1440x900.

Requerimientos recomendados:

- Intel Core Dúo 1.8Ghz o AMD Athlon 64 X2 3600+.
- 1024 megabytes (MB) de RAM.
- Tarjeta gráfica con 256MB mínimo de RAM (NVidia Geforce 6600 o ATI X600)
- Resolución 1680x1050.
- Dispositivos de audio.

4. Software

La aplicación permitirá ejecutarse en cualquier versión de los sistemas operativos MS Windows, así como en MAC OS.

2.6 Definición del actor del sistema

Los actores del sistema son agentes externos, roles que los usuarios o dispositivos juegan cuando interactúan con el software. En este caso se tiene un actor del sistema el cual es el estudiante.

Actor del Sistema	Justificación
-------------------	---------------

Estudiante	Persona con privilegios para interactuar con la aplicación. Haciendo uso de todas las funcionalidades con que cuenta la misma con fines educativos.
-------------------	---

2.7 Casos de uso del sistema

Los casos de uso representan la manera en que los usuarios interactúan con el sistema. Son artefactos narrativos que describen el comportamiento de una aplicación teniendo en cuenta el punto de vista del actor.

2.7.1 Diagrama de casos de uso del sistema

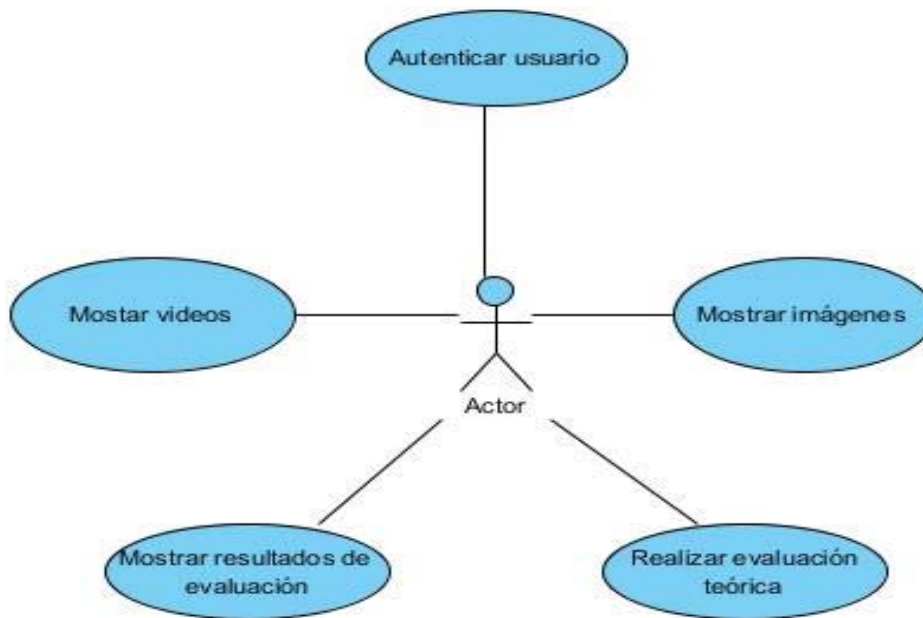


Imagen 10 Diagrama de casos de uso del sistema.

2.7.2 Descripción de los casos de uso del sistema

Caso de Uso	Mostrar resultados de evaluación.
Objetivo	Mostrar al estudiante el resultado obtenido al finalizar el cuestionario evaluativo.
Pre-condiciones	El estudiante debe estar registrado.

Post-condiciones	El estudiante podrá visualizar el resultado del cuestionario evaluativo.	
Actores	Estudiante	
Resumen	El caso de uso se inicia cuando el estudiante responde los cuestionarios, se mostrará el usuario del estudiante que realizó el cuestionario y la cantidad de errores cometidos por cada proceder.	
Complejidad	Alta	
Requisitos de Referencia	RF5	
Prioridad	Alta	
Flujo de eventos: Mostrar resultados de evaluación		
Sección :”Mostrar resultados de evaluación”		
	Actor	Sistema
	Selecciona la opción realizar cuestionario evaluativo.	1.1. Muestra una pantalla con los siguientes campos: <ul style="list-style-type: none"> • Nombre. • Apellidos. • Cantidad de errores cometidos por cada rol.

Caso de Uso	Mostrar Imágenes	
Objetivo	Mostrar imágenes de ayuda al estudiante.	
Pre-condiciones	El estudiante debe estar en el panel de inicio.	
Post-condiciones	El estudiante visualizará imágenes de ayuda.	
Actores	Estudiante.	
Resumen	El caso de uso se inicia cuando el usuario se encuentra en el menú principal de la aplicación y selecciona el icono de “Imágenes” y visualiza las imágenes de apoyo.	
Complejidad	Baja.	
Requisitos de Referencia	RF2.	
Prioridad	Alta.	
Flujo de eventos: Mostrar Imágenes		
Sección :”Mostrar Imágenes”		
	Actor	Sistema
	Selecciona la opción realizar ver imágenes.	1.1. Mostrará formulario en la que el usuario visualizará imágenes.

Caso de Uso	Mostrar videos.	
Objetivo	Mostrar videos de ayuda al estudiante.	
Pre-condiciones	El estudiante debe estar en el panel de inicio.	
Post-condiciones	El estudiante visualizará videos de ayuda.	
Actores	Estudiante.	
Resumen	El caso de uso se inicia cuando el usuario se encuentra en el menú principal de la aplicación y selecciona el icono de "videos" y visualiza los videos de apoyo.	
Complejidad	Baja.	
Requisitos de Referencia	RF3.	
Prioridad	Alta.	
Flujo de eventos: Mostrar videos		
Sección : "Mostrar videos"		
	Actor	Sistema
	Selecciona ;la opción ver videos.	1.1. Mostrará formulario donde el estudiante visualizará videos.

Caso de Uso	Realizar evaluación de teórica.	
Objetivo	Verificar preparación del estudiante.	
Actores	Estudiante.	
Resumen	El caso de uso se inicia cuando el usuario se encuentra en el menú principal de la aplicación y selecciona el icono de “Comenzar evaluación”, el usuario se registra y posteriormente deberá seleccionar el rol en el cual se desempeñará durante la práctica de simulación (Clínico, Asistente, Técnico de Laboratorio), después de seleccionado el rol, deberá responder varias preguntas antes de acceder a la práctica de simulación del rol seleccionado.	
Pre-condiciones	El estudiante deberá estar autenticado	
Post-condiciones	El estudiante accederá a realizar un cuestionario evaluativo.	
Complejidad	Alta.	
Requisitos de referencia	RF4.	
Prioridad	Crítico.	
Flujo básico: Realizar evaluación teórica.		
Sección: “Realizar evaluación de teórica.”		
	Actor	Sistema
	Seleccionará el icono de comenzar la evaluación.	1.1 Mostrará formulario en la que el usuario deberá identificarse con su usuario y contraseña.
	1.2 Introduce en el correspondiente usuario y contraseña. Posteriormente presiona el botón “continuar”.	1.3 Mostrará un formulario donde el estudiante deberá seleccionar el rol.
	1.4 Selecciona el rol que realizará.	1.5 Mostrará un cuestionario evaluativo.

	1.6 Realiza el cuestionario evaluativo y obtiene más del 75% de la puntuación.	1.7 Brinda acceso al entrenamiento.
Flujos Alternos		
	Actor	Sistema
	Realiza el cuestionario evaluativo y obtiene menos del 75% de la puntuación.	1.1 Mostrará formulario en el que se mostrarán la cantidad de errores cometidos por cada proceder.

Caso de Uso	Autenticar usuario.
Objetivo	Permitir al usuario autenticarse en la aplicación.
Pre-condiciones	El estudiante debe estar en el panel de inicio.
Post-condiciones	El estudiante podrá acceder a seleccionar el rol en el cual se desempeñará.
Actores	Estudiante.
Resumen	El caso de uso se inicia cuando el actor ejecuta la aplicación se mostrará el menú principal de la aplicación donde el usuario debe presionar el icono de “Realizar evaluación teórica”, donde la aplicación deberá mostrar una escena donde el usuario inserte su usuario y contraseña. Finaliza cuando el usuario procede a seleccionar el rol en el cual se desempeñará en la práctica de simulación.
Complejidad	Alta.
Requisitos de Referencia	RF1.
Prioridad	Alta.

Flujo de eventos: Autenticar usuario		
Sección :” Autenticar usuario”		
	Actor	Sistema
	Selecciona el icono de “Realizar evaluación teórica”.	1.1 La aplicación muestra un dialogo donde el usuario ingresa los datos correspondientes (nombre, apellidos, profesor).

2.8 Diagramas de interacción

Los diagramas de interacción describen el comportamiento de los objetos dentro de un caso de uso o sección.

Un diagrama de secuencia muestra las interacciones entre objetos según un punto de vista temporal, por lo que representa dicha interacción en base a la cronología de los envíos de mensajes.

En este se pueden apreciar líneas verticales que representan los objetos de un escenario y los mensajes entre dichos objetos son flechas que los conectan. Los períodos de actividad de estos objetos están representados por los rectángulos a lo largo de las líneas verticales. En estos diagramas los actores son encargados de generar varios eventos.

Diagrama de Secuencia CU Autenticar Usuario

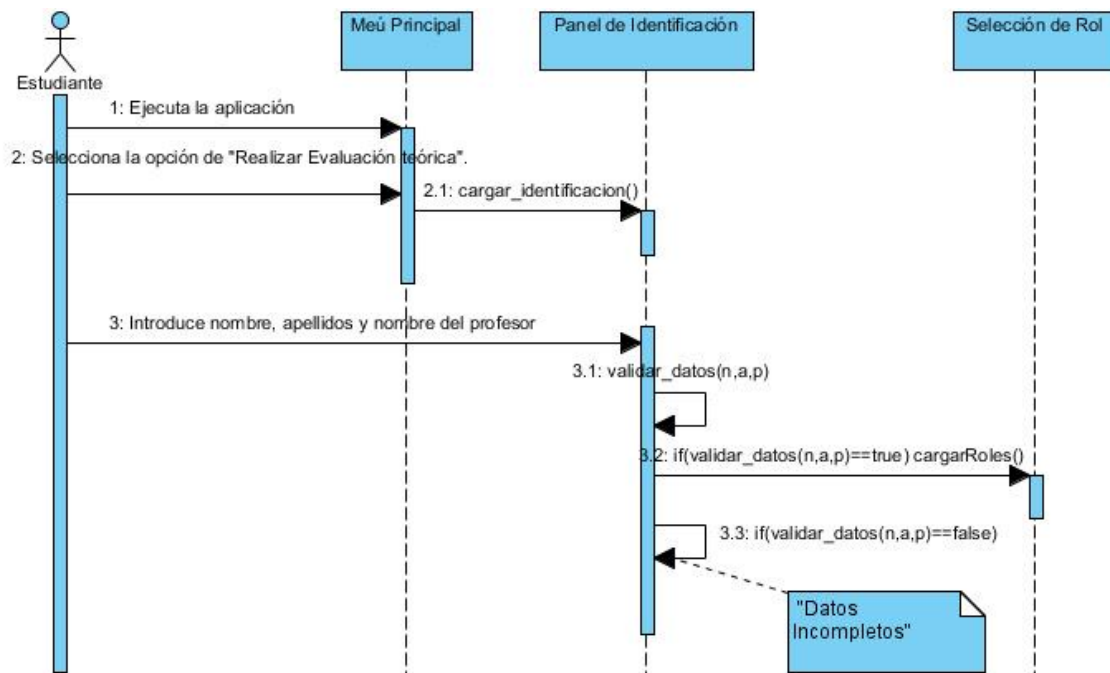


Imagen 11 Diagrama de Secuencia CU Autenticar Usuario.

Diagrama de Secuencia CU Mostrar Imágenes

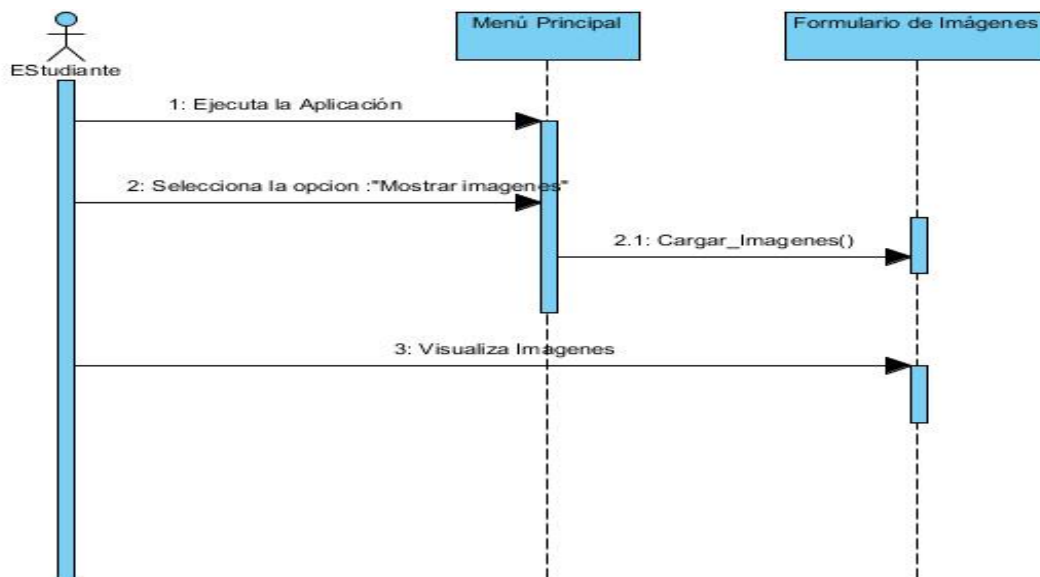


Imagen 12 Diagrama de Secuencia CU Mostrar imágenes.

Diagrama de secuencia CU Realizar evaluación teórica.

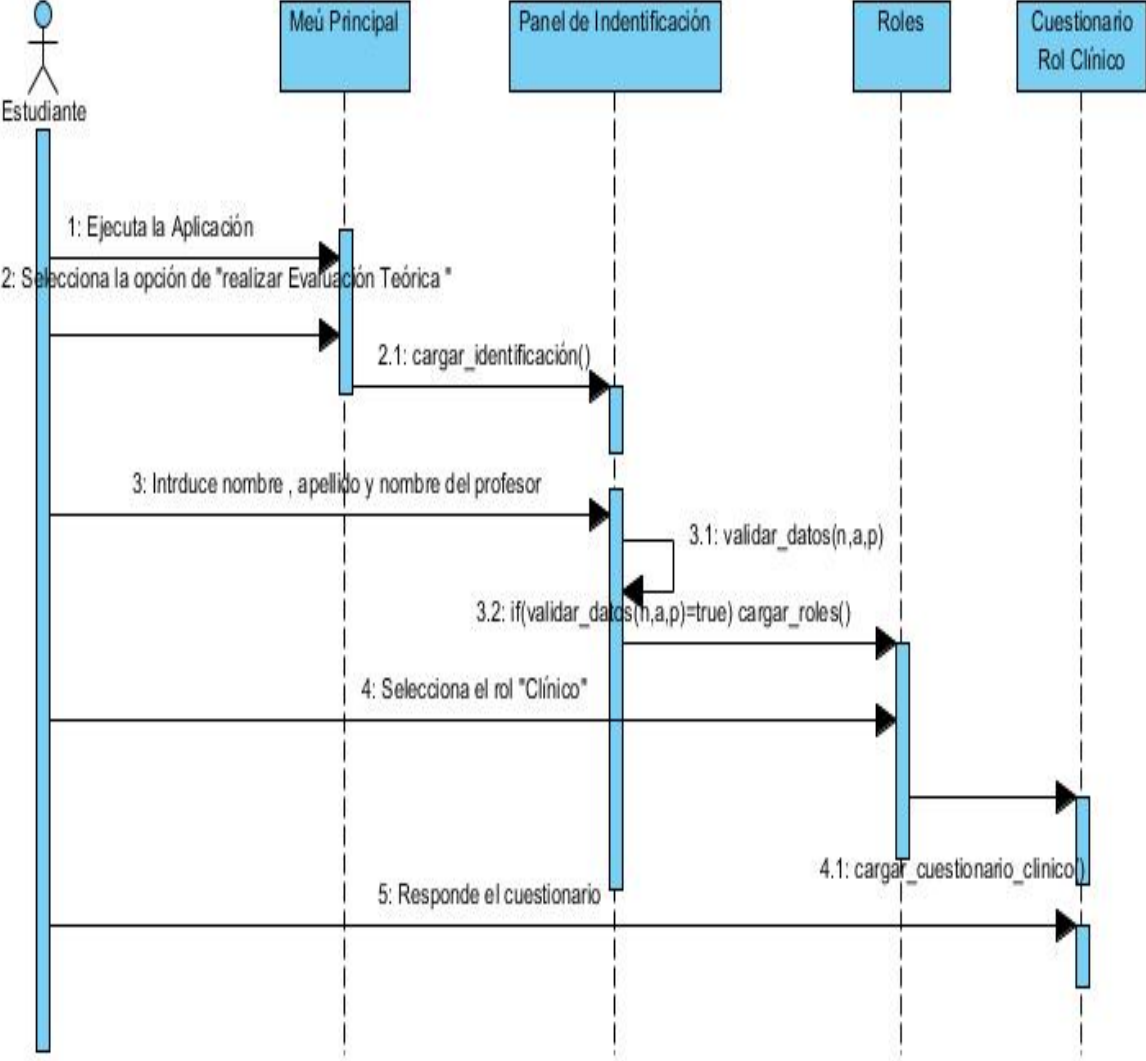


Imagen 13 Diagrama de Secuencia CU Realizar evaluación teórica.

Diagrama de secuencia CU Mostrar resultados de evaluación.

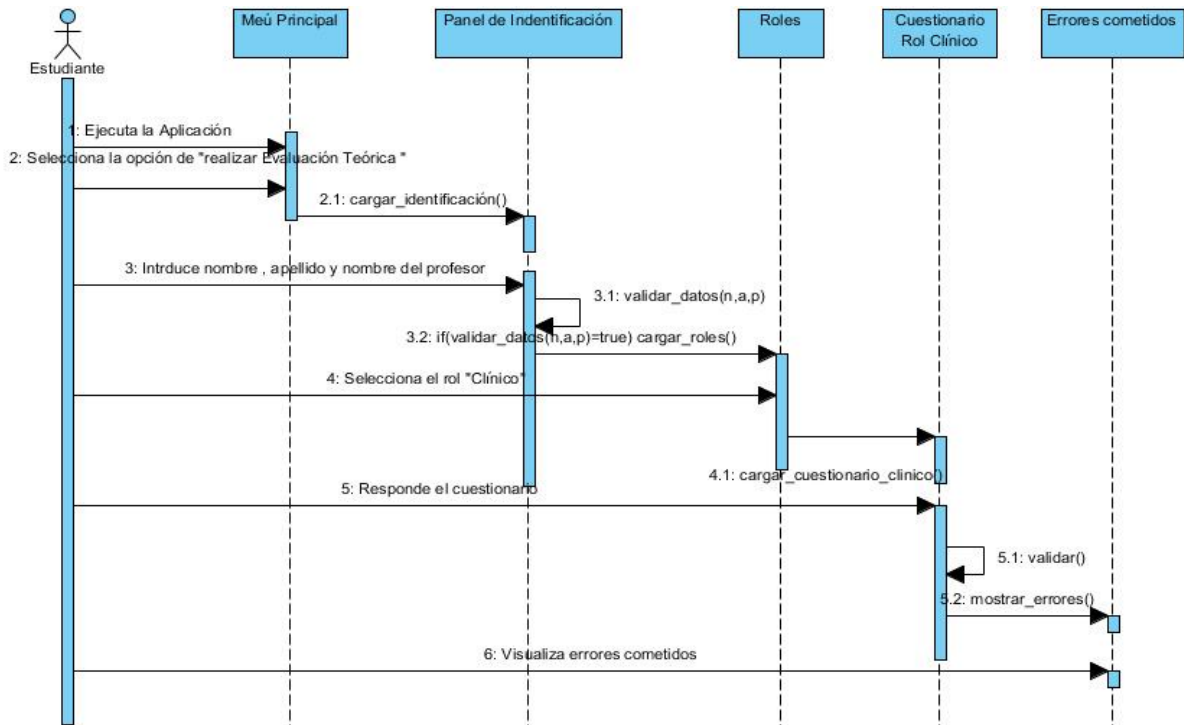


Imagen 15 Diagrama de secuencia CU Mostrar resultados de evaluación.

Diagrama de secuencia CU Mostrar videos

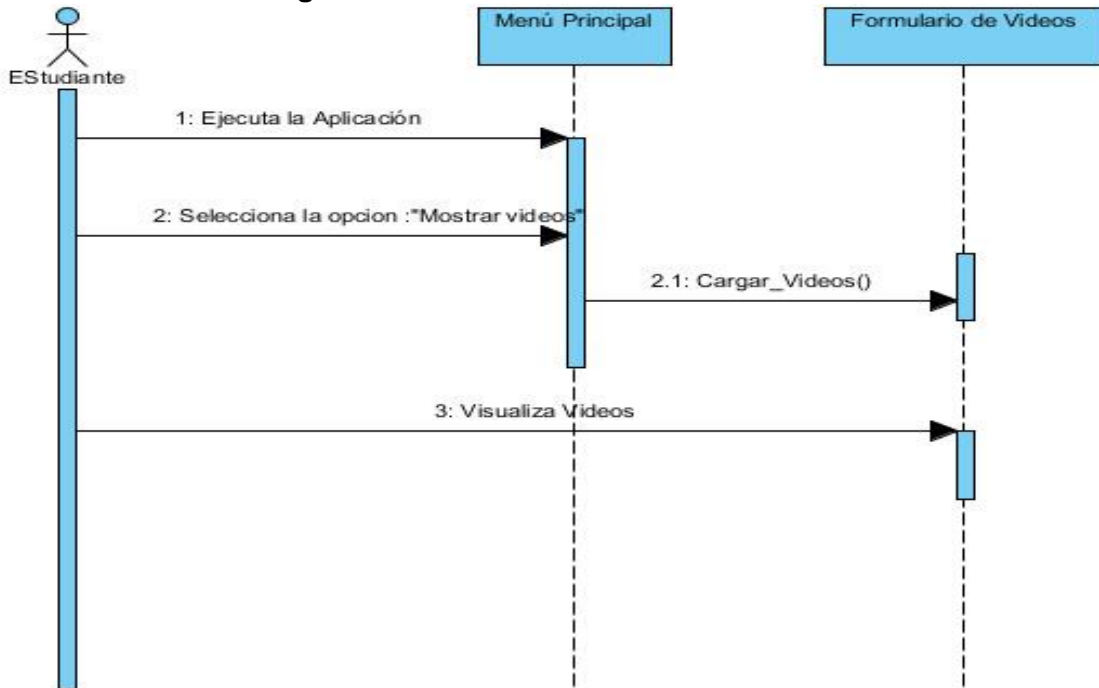


Imagen 16 Diagrama de secuencia CU Mostrar videos.

2.9 Conclusiones parciales

En el capítulo se da una descripción del modelo de dominio, para comprender mejor el funcionamiento del sistema. Se realizó la selección de los actores que intervienen en el sistema, y se detectaron los requisitos funcionales y no funcionales para el AVAI, luego agrupados en casos de usos y descritos para una mejor comprensión de las funcionalidades con que debe de contar la interfaz desarrollada.

Posteriormente se realizaron los diagramas de secuencia para cada caso de uso del sistema en los cuales se refleja de manera detallada el funcionamiento de cada uno de ellos.

CAPÍTULO 3: Implementación y Validación del Sistema

3.1 Introducción

En el presente capítulo se muestran fragmentos de código de algunas de las funcionalidades que brinda el sistema, se realizarán pruebas de caja negra que son descritas teniendo en cuenta la sección, los escenarios y la descripción de la funcionalidad, además se realiza una validación de la interfaz propuesta, mediante el Método Delphi.

```
-----  
function Validando_Datos.onValidando_Identificacion ( )  
-----  
    local hUser = application.getCurrentUser ( )  
    local nombre= hud.getComponent ( hUser, "Entrada_Datos.nombre" )  
    local apellidos= hud.getComponent ( hUser, "Entrada_Datos.apellidos" )  
    local profesor = hud.getComponent ( hUser, "Entrada_Datos.Profesor" )  
    local n = hud.getEditText (usuario)  
    local a = hud.getEditText (contrasena)  
    local p = hud.getEditText (profesor)  
    if( u="" ) or ( c="" ) or ( p="" )  
    then hud.callAction (hUser, "Entrada_Datos.Mostrar_Cartel" )  
    else  
    hud.callAction (hUser, "Entrada_Datos.Cambiar_Panel" )  
    end  
end  
end
```

Imagen 17 Implementación CU Autenticar.


```

var1 = math.roundToNearestInteger ( math.random ( 1, 4 ) )
if( var1==1)
then hud.callAction (this.getUser ( ),"Prueba_de_Cera.Posicion1_Pregunta1" )
elseif(var1==2)
then hud.callAction (this.getUser ( ),"Prueba_de_Cera.Posicion2_Pregunta1" )
elseif (var1==3)
then hud.callAction (this.getUser ( ),"Prueba_de_Cera.Posicion3_Pregunta1" )
else
hud.callAction (this.getUser ( ),"Prueba_de_Cera.Posicion4_Pregunta1" )
end

-----Random2-----

var2 =math.roundToNearestInteger ( math.random ( 1, 4 ) )
if( var1==var2)
then
while((var1==var2 ))
do
var2 =math.roundToNearestInteger ( math.random ( 1, 4 ) )
end
end

if( var2==1)
then hud.callAction (this.getUser ( ),"Prueba_de_Cera.Posicion1_Pregunta2")
elseif(var2==2)
then hud.callAction (this.getUser ( ),"Prueba_de_Cera.Posicion2_Pregunta2")
elseif (var2==3)
then hud.callAction (this.getUser ( ),"Prueba_de_Cera.Posicion3_Pregunta2")
else hud.callAction (this.getUser ( ),"Prueba_de_Cera.Posicion4_Pregunta2")
end

```

Imagen 18 Implementación de preguntas aleatorias.

3.2 Pruebas de caja negra

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa.

Las pruebas de caja negra pretenden encontrar:

- Funciones incorrectas o ausentes
- Errores en la interfaz
- Errores en estructuras de datos o en accesos a bases de datos externas
- Errores de rendimiento
- Errores de inicialización y de terminación

3.2.1 Validación CU Autenticar Usuario

Nombre de la sección	Escenarios de la selección	Descripción de la funcionalidad
SC 1: Autenticar	EC 1.1: Autenticar.	El usuario introducirá los datos solicitados por la aplicación.
	EC .2: El usuario deja un campo sin llenar.	El usuario introduce los datos y deja un campo sin llenar, al presionar el botón Aceptar se le mostrará el error de "Campos Incompletos".
	EC .3: El usuario introduce números o caracteres extraños.	El usuario introduce los datos, pero introduce números o caracteres extraños, al presionar el botón Aceptar se le mostrará el error de "Campos Incorrectos".

Tabla #2 Validación de caso de uso Autenticar.

Descripción de las variables:

Nombre: Campo de texto, no admite datos numéricos o nulos.

Apellidos: Campo de texto, no admite datos numéricos o nulos

Profesor: Campo de texto, no admite datos numéricos o nulos

El estudiante debe de introducir: nombre, apellidos y el nombre del profesor que lo atiende.



Imagen 19 Identificación del estudiante.



Imagen 20 Error de campos vacíos.



Imagen 21 Error de caracteres extraños.



Imagen 22 Error de caracteres numéricos.

3.2.2 Validación CU Realizar evaluación teórica

Nombre de la sección	Escenarios de la selección	Descripción de la funcionalidad
SC 1: Realizar evaluación teórica.	EC 1.1: Cuestionario.	El usuario responderá el cuestionario.
	EC 1.2: El usuario no alcanza el 75%.	El usuario no alcanza más del 75 % de la puntuación, por lo que no podrá acceder a la práctica de simulación.

Tabla #3 Validación CU Realizar evaluación teórica.

3.2.3 Validación CU Mostrar resultados de evaluación

Nombre de la sección.	Escenarios de la selección.	Descripción de la funcionalidad.
SC 1: Mostrar resultados de evaluación.	SC 1: Mostrar resultados de evaluación.	El usuario visualizará el resultado de la evaluación.

Tabla #4 Validación CU Mostrar resultados de evaluación.

3.2.4 Validación CU Mostrar Videos

Nombre de la sección.	Escenarios de la selección.	Descripción de la funcionalidad.
SC 1: Mostrar videos.	SC 1: Mostrar videos.	El usuario visualizará videos de ayuda.

Tabla #5 Validación CU Mostrar Videos.

3.2.5 Validación CU Mostrar imágenes

Nombre de la sección.	Escenarios de la selección.	Descripción de la funcionalidad.
SC 1: Mostrar imágenes.	SC 1: Mostrar imágenes.	El usuario visualizará imágenes de ayuda.

Tabla #6 Validación CU Mostrar imágenes.

3.3 Método Delphi

El método Delphi es un método de prospectiva basado en consultas a expertos en el que los temas planteados, en este caso en forma de cuestionario, son remitidos a un conjunto de jueces llamado panel de expertos que son los que darán su opinión acerca del asunto que se está tratando y entre los cuales se hará necesario garantizar el anonimato (29).

Suelen distinguirse tres etapas o fases fundamentales en la aplicación del método, según la variada bibliografía consultada al respecto, que son:

Fase preliminar: Se delimita el contexto, los objetivos, el diseño, los elementos básicos del trabajo y la selección de los expertos.

Fase exploratoria: Elaboración y aplicación de los cuestionarios según sucesivas vueltas, de tal forma que con las respuestas más comunes de la primera se confecciona la siguiente.

Fase final: Análisis estadísticos y presentación de la información.

3.3.1 Selección de Expertos

Un experto es una persona o un grupo de personas que poseen un alto grado de conocimientos sobre el tema de estudio, ya sea porque se han dedicado a la investigación y estudios de aspectos relacionados con el tema, o bien porque en su experiencia profesional, el tema ha constituido parte importante de su trabajo.

De acuerdo con lo anteriormente expresado, se definen como características fundamentales para los expertos que conforman este grupo de evaluación:

Conocimientos acerca de los contenidos que sustentan la propuesta de solución (30).

- Software educativo.
- Diseño.

Graduado de Nivel Superior.

Prestigio en el colectivo de trabajo.

Capacidad de análisis y pensamiento lógico.

El conocimiento sobre estos temas permitirá que las opiniones brindadas sean confiables y válidas para el objetivo propuesto. Se tendrán en cuenta otras cualidades como la

honestidad, sinceridad y responsabilidad, haciendo más confiables las opiniones brindadas. La búsqueda de los expertos se realizará en el contexto de la Universidad de las Ciencias Informáticas (UCI) y la Universidad de Pinar del Río (UPR) Hermanos Saiz.

3.3.2 Preguntas realizadas

Pregunta 1: ¿Son adecuados los colores utilizados en la interfaz?

Pregunta 2: ¿Considera usted que la tipografía que se utiliza es correcta?

Pregunta 3: ¿Considera usted que los íconos utilizados en la interfaz están en correspondencia con las funcionalidades que representan?

Pregunta 4: ¿Considera usted que la interfaz brinde a los usuarios la posibilidad de acceder con facilidad a sus funcionalidades?

Pregunta 5: ¿Considera usted que el diseño de la interfaz es lo suficientemente sencillo y al mismo tiempo robusto?

Pregunta 6: ¿Considera usted que los elementos gráficos y textuales que conforman la interfaz son claros y de fácil identificación?

3.3.3 Resultados del método Delphi

Después aplicar el método los resultados obtenidos arrojaron que el 80% de las preguntas fueron catalogadas de Muy Adecuadas mientras que el 20% de estas fueron catalogas de Bastante Adecuadas, como se indica en la siguiente gráfica.

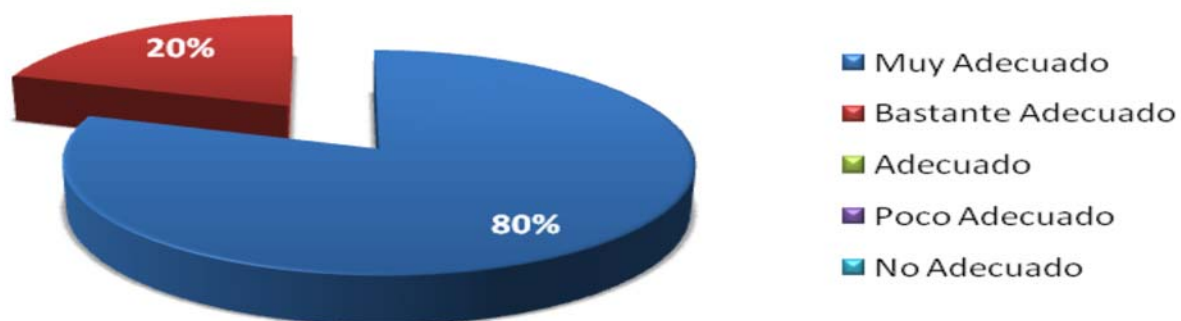


Imagen 23 Gráfica.

3.4 Conclusiones parciales

En el presente capítulo se mostraron algunos fragmentos de código implementados para el funcionamiento de la interfaz, además se abordó la validación de dicha interfaz, la cual se realizó mediante pruebas de caja negra y por último se aplicó el método de Delphi para obtener las impresiones de expertos en el tema. De esta manera se validó la funcionalidad del software a los distintos objetivos perseguidos. Las pruebas realizadas arrojaron resultados satisfactorios.

CONCLUSIONES

Con la realización de este trabajo de diploma, y en cumplimiento del objetivo inicialmente planteado se obtuvo una interfaz gráfica de usuario que se adapta al AVAI de construcción de prótesis oculares para el proyecto PROTESIM. La interfaz desarrollada incluye dentro de sus características, funcionalidades de algunos ambientes virtuales de aprendizaje y juegos serios. La misma cumple con los principios básicos para el diseño de las interfaces de usuario.

La interfaz es capaz de proporcionar una respuesta visual a las acciones del usuario, brindando un conjunto de funcionalidades que se utilizarán para interactuar con la aplicación de manera sencilla. Además, la misma cuenta con gran aceptación por parte de los expertos y los médicos especialistas.

Se crearon cuestionarios evaluativos para verificar la preparación previa de los estudiantes, además se logró mostrar la información relevante para los estudiantes de forma similar a la utilizada en videojuegos.

.

RECOMENDACIONES

Se proponen las siguientes recomendaciones:

- Comenzar el desarrollo de ambientes virtuales de aprendizaje para la construcción de otras prótesis faciales teniendo en cuenta las múltiples ventajas que brinda el motor gráfico Shiva3D.
- Continuar con el estudio del motor gráfico Shiva3D con el objetivo de crear nuevos y mejores ambientes virtuales de aprendizaje para contribuir a mejorar la calidad de la educación en nuestro país.
- Extender a archivos de texto y audio la información de apoyo que se muestra a los estudiantes en la interfaz.
- Continuar con el estudio del editor de Huds que contiene ShiVa3D para mejorar las futuras versiones de interfaz de usuario.

Citas Bibliográficas

1. Batista, Miguel Ángel Herrera. Consideraciones para el diseño didáctico de ambientes virtuales de aprendizaje.
<http://www.rieoei.org/deloslectores/1326Herrera.pdf>.
2. Avila, Patricia. Ambientes Virtuales de Aprendizaje, una nueva experiencia. 1999.
3. Gómez, Maria Sánchez. Buenas Prácticas en la creación de Serious Games.
<http://spdece07.ehu.es/actas/Sanchez.pdf>.
4. Muñoz, Tomás García. El cuestionario como instrumento de Evaluación.
http://www.univsantana.com/sociologia/EI_Cuestionario.pdf.
5. Mercovich, E. G. Ponencia sobre Diseño de Interfaces y Usabilidad: cómo hacer productos más útiles, eficientes y seductores. Recuperado el 12 de marzo de 2011, de <http://www.gaiasur.com.ar/infoteca/siggraph99/disenio-de-interfaces-y-usabilidad.html> : s.n., 2000.
6. La intersección entre factores humanos, diseño gráfico, interacción y comunicación. Buenos Aires. Argentina. 2009.
7. Ibáñez, Juan Carlos Pantoja. Arquitectura de la Información: Introducción al proceso de desarrollo en el Diseño de Interfaces de Usuario. [En Línea] [Citado: 30 de septiembre del 2009] <http://www2.udec.cl/ai/proceso.pdf>.
8. <http://www.scribd.com/doc/2522263/Consideraciones-para-el-diseno-de-interfaces-de-usuario-en-sistem>, [4] Eugenio Jacobo Hernández. Consideraciones para el diseño de interfaces de usuario en sistemas operativos. [En Línea] [Citado: 10 de diciembre del 2009].
9. Expósito., Carlos Marrero. Interfaz gráfica de usuario: Aproximación semiótica y cognitiva.
http://www.chr5.com/investigacion/investiga_igu/igu_aproximacion_semio-cognitiva_by_chr5.pdf.
10. Mora, Marcos. Pautas para el diseño y la evaluación de interfaces de usuario.
m/pdf/2004_pautas-iula.pdf .
11. Gómez, Leopoldo Sebastián. Diseño de Interfaces de Usuario Principios, Prototipos y Heurísticas para Evaluación. [Citado: 15 de diciembre del 2009]
12. El diseño.
<ingenieria.lm.uasnet.mx/sitio/servicios/pagper/maestros/.../05Diseno.doc>
13. Barros, Rodríguez. DISEÑO DE INTERFACES Y CONDICIONES DE USABILIDAD.
<www.fba.unlp.edu.ar/news/SCYTEC/PDF/RODRIGUEZ%20BARROS.pdf>

14. Roe, Benjamin. Diseño de Interfaces de Usuario Usables: Una Guía Rápida para Desarrolladores de Software Libre y de Código Abierto. . <http://mundogeek.net/traducciones/interfaces-usuario-usables/gui.html> [Citado: 15 de diciembre del 2009].
15. Nielsen, Jakob. Desining Web Usability.[En Línea] [Citado: 17 de enero del 2010]. <http://www.scribd.com/doc/6700249/Jakob-Nielsen-Desining-Web-Usability-Espanhol>.
16. Ricardo Baeza-Yates, Cuauhtémoc Rivera Loaiza y Javier Velasco Martín. Arquitectura de la información y usabilidad en la web. <http://www.mantruc.com/files/art-epi-2004.pdf> [Citado: 17 de enero del 2010].
17. Arquitectura de información y usabilidad: nociones básicas para los profesionales de la información .
18. Tognazzini, Bruce. Principios de diseño de interacción. <http://usabilidadcss.wordpress.com/tag/principios/>[Febrero 2012].
19. Montes., Lic. Antonio. Arquitectura de información y usabilidad: nociones básicas para los profesionales de la información. http://bvs.sld.cu/revistas/aci/vol12_6_04/aci04604.htm [17 de enero del 2010].
20. Ricardo Baeza-Yates, Cuauhtémoc Rivera Loaiza y Javier Velasco Martín. Arquitectura de la información y usabilidad . <http://www.mantruc.com/files/art-epi-2004.pdf> [17 de enero del 2010].
21. Manchón, Eduardo. ¿Qué es la usabilidad? Definición de Usabilidad. . <http://www.ingenieriasimple.com/usabilidad/QueEsUsabilidad.pdf> [17 de enero del 2010].
22. Montes, Lic. Antonio. Arquitectura de información y usabilidad: nociones básicas para los profesionales de la información . [17 de enero del 2010].
23. <http://www.dpad.cl/10-ingeniosas-interfaces-graficas-de-videojuegos/>.
24. Stonetrip. 2003-2010. Shiva3D 3D Game Engine with Development Tools. Shiva3D Game Engine with Development Tools. . <http://www.stonetrip.com> [2003-2010] .
25. Santos, Miguel Ángel Sanz. Conceptos Básicos de Lenguajes. http://ggyma.geo.ucm.es/docencia/documentos/informatica/Informatica_5.pdf.
26. Rio, UPC. 1999. The programming Language Lua. The programming Language Lua. [En línea].
27. Figueroa, R. G., C. J. Solís, et al. "Metodologías Tradicionales vs. Metodologías Ágiles." <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>.

28. “El Lenguaje Unificado de Modelado”.
www.disca.upv.es/enheror/pdf/ActaUML.PDF.

29. María de Lourdes Bravo Estévez, Universidad de Cienfuegos. Cuba. José Joaquín Arrieta Gallastegui, Universidad de Oviedo. España. El método Delphi. Su implementación en una estrategia didáctica para la enseñanza de las demostraciones geométricas. [En Línea].

30. Eneko Astigarraga. EL MÉTODO DELPHI. [En Línea] [Citado: 20 de mayo del 2010]
http://www.unalmed.edu.co/~poboyca/documentos/documentos1/documentos-Juan%20Diego/Plnaifi_Cuencas_Pregrado/Sept_29/Metodo_delphi.pdf.

Bibliografía consultada

1. Benjamin Roe. Diseño de Interfaces de Usuario Usables: Una Guía Rápida para Desarrolladores de Software Libre y de Código Abierto.

<http://mundogeek.net/traducciones/interfaces-usuario-usables/gui.html>

2. Joel Spolsky. Diseño de Interfaz de Usuario para Programadores Capítulo 6: Diseñando para gente que tiene cosas mejores que hacer con su vida. [En Línea] [Citado: 30 de septiembre del 2009]

http://willydev.net/insitecreation/v1.0/descargas/WillyDev_InterfazdeUsuarioparaDesarrolladores6.pdf

3. Carlos Marrero Expósito. Interfaz Gráfica de Usuario. Aproximación semiótica y cognitiva.

http://www.chr5.com/investigacion/investiga_igu/igu_aproximacion_semiocognitiva_by_chr5.pdf

4. Lizbeth Luna González. El diseño de Interfaz Gráfica de Usuario para publicaciones digitales. http://www.revista.unam.mx/vol.5/num7/art44/ago_art44.pdf

5. Heidi Ruiz Villamil, Pedro Cobo Parra. Interfaz gráfico de usuario para el diseño de absorbentes microperforados múltiples. [En Línea] [Citado: 8 de octubre del 2009]

http://www.sea-acustica.es/Buenos_Aires_2008/a-079.pdf

6. Eugenio Jacobo Hernández. Consideraciones para el diseño de interfaces de usuario en sistemas operativos. [En Línea] [Citado: 8 de octubre del 2009]

<http://www.scribd.com/doc/2522263/Consideraciones-para-el-diseno-de-interfaces-de-usuario-en-sistemas-operativos>

7. Jaime Badia, Marta González, Ramon Salat. Usabilidad en el diseño de una interfaz de usuario para un entorno industrial. [En Línea] [Citado: 15 de octubre del 2009] <http://www.aipo.es/articulos/3/69.pdf>

8. Alejandro Baez, Cristian Castañeda, Diego Castañeda. Metodología para el diseño y desarrollo de interfaces de usuario. [En Línea] [Citado: 15 de octubre del 2009]

[http://pegasus.javeriana.edu.co/~fwj2ee/descargas/metodologia\(v0.1\).pdf](http://pegasus.javeriana.edu.co/~fwj2ee/descargas/metodologia(v0.1).pdf)

9. Noe Sierra Romero, Sergio V. Chapa Vergara. Diseño de Interfaces Visuales. [En Línea] [Citado: 15 de octubre del 2009]

<http://www.cs.cinvestav.mx/CursoVis/prinvisual.html>

10. Yusef Hassan, Francisco J. Martín Fernández, Ghzala Iazza. Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información.

<http://www.hipertext.net/web/pag206.htm#Usabilidad%20y%20accesibilidad>

11. Antonio Montes de Oca. Arquitectura de información y usabilidad: nociones básicas para los profesionales de la información. [En Línea] [Citado: 23 de octubre del 2009]

http://bvs.sld.cu/revistas/aci/vol12_6_04/aci04604.htm

12. Jesús Tramullas Saz. Arquitectura de la información: más que diseño, hacia la findability. [En Línea] [Citado: 8 de noviembre del 2009]

<http://www.sedic.es/clip39.pdf>

13. Marta Gómez Reyes. Arquitectura de la Información: Algo más que un Concepto.

<http://www.bibliociencias.cu/gsd/collect/eventos/index/assoc/HASH01fb.dir/doc.pdf>

14. Jesús Bustamante. La arquitectura de la información del siglo XX al XXI. [En Línea] [Citado: 8 de noviembre del 2009]

<http://www.elprofesionaldelainformacion.com/contenidos/2003/marzo/4.pdf>

15. Catherine Roy. Accesibilidad. [En Línea] [Citado: 15 de noviembre del 2009]

<http://vecam.org/article611.html>

16. David Moner Cano, Jordi Sabaté Alsina. USABILIDAD. [En Línea] [Citado: 15 de noviembre del 2009]

http://www.unal.edu.co/documentos/servicio_web/usabilidad_para_web.pdf

Glosario de Términos

Arquitectura de la Información: Disciplina y arte encargada del estudio, análisis, organización, disposición y estructuración de la información en espacios de información, y de la selección y presentación de los datos en los sistemas de información interactivos y no interactivos.

Android: Sistema operativo basado en Linux diseñado originalmente para dispositivos móviles, tales como teléfonos inteligentes, actualmente se encuentra en desarrollo para usarse en netbooks y PCs.

ChemLab: Programa de simulación de un laboratorio de química. Se utilizan el equipamiento y los procedimientos más comunes para simular los pasos necesarios que se efectúan en los experimentos de laboratorio. Cada tipo de simulación se encuentra situado en su propio módulo de simulación, así se pueden usar distintos equipos de laboratorio con una única interfaz.

DirectX: Es una colección de API desarrolladas para facilitar las complejas tareas relacionadas con multimedia, especialmente programación de juegos y vídeo, en la plataforma Microsoft Windows.

IPAD: Dispositivo electrónico tipo Tablet PC desarrollado por Apple Inc. Anunciado el 27 de enero de 2010, el 2 de marzo de 2011 apareció la segunda generación. Se sitúa en una categoría entre un "teléfono inteligente" (Smartphone) y una computadora portátil, enfocado más al acceso que a la creación de contenido.

IPhone: Contracción de *intelligent phonees* una familia de teléfonos inteligentes multimedia con conexión a Internet, pantalla táctil capacitiva y una interfaz de software minimalista diseñados por la compañía Apple Inc.

Interfaz de Usuario: Medio que el usuario utiliza para comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo, normalmente suelen ser fáciles de entender y fáciles de accionar.

Interactive Physics: Programa educativo en el que se hace fácil observar, descubrir, y explorar el mundo físico con simulaciones emocionantes.

Multiplataforma: Término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.

Palm OS: Es un sistema operativo que fue hecho por *Palm Source*, Inc. para computadores de mano fabricados por varios licenciarios.

Phet: Simulador facilita el aprendizaje de la física.

Prótesis Ocular: Modalidad de prótesis facial que va a reparar las pérdidas parciales o totales y deformaciones del globo ocular.

Prótesis Bucomaxilofacial: Sustitución protésica de parte o de la totalidad del maxilar superior, nariz o pómulo. Se aplica cuando la reparación quirúrgica aislada es insuficiente.

Caso de uso: Reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos.

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

Requisitos: Conjunto de características que debe tener un producto o servicio para satisfacer las necesidades y expectativas del cliente.

Rol: Papel, cometido o función que tiene o desempeña que interpreta un actor.

Servidor: Un servidor es un ordenador de gran potencia, que se encarga de "prestar un servicio" a otros ordenadores que se conectan a él.

Software: Término genérico utilizado en informática para designar programas o fragmentos de programas.

NVIDIA PhysX: PhysX es un motor propietario de capa de software intermedia o "middleware" y un kit de desarrollo diseñados para llevar a cabo cálculos físicos muy complejos. Conocido anteriormente como la SDK de NovodeX, fue originalmente diseñado por AGEIA y tras la adquisición de AGEIA, es actualmente desarrollado por Nvidia e integrado en sus chips gráficos.

UML: Unified Modeling Language. Lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.

VLabQ: Simulador interactivo de prácticas de laboratorio de Química que usa equipos y procedimientos estándares para simular los procesos que intervienen en un experimento o práctica.

Web: Red de documentos HTML intercomunicados y distribuidos entre servidores web.

Web OS: Es un sistema operativo multitarea para sistemas embebidos basado en Linux, desarrollado por Palm, Inc.

Wii: Es la sexta videoconsola de sobremesa producida por Nintendo. Fue desarrollada en colaboración con IBM y ATI. Es la sucesora directa de la Nintendo Game Cube y compite actualmente contra la Playstation 3 de Sony y la Xbox 360 de Microsoft como parte de las videoconsolas de séptima generación.