

Capítulo 2: Ingeniería de Software, Análisis y Diseño

En todo desarrollo de sistemas de software es de suma importancia el seguir alguna especificación que permita a los desarrolladores el tener una disciplina que haga que todas las etapas del desarrollo del sistema, desde la pesquisa inicial de requerimientos hasta las pruebas finales del sistema, sean no solo más coherentes sino también más formales.

El desarrollo de software que este proyecto propone, al ser una herramienta que pretende tener aplicación dentro del contexto de un problema real, tiene que seguir un proceso de análisis y diseño que proporcione los cimientos bajo los cuales se va a desarrollar la aplicación conjuntamente. Es por esto que en este capítulo se detallan los procesos de ingeniería de software, análisis, y diseño que se involucran para el desarrollo de una aplicación de software que puede utilizarse como auxiliar al tratamiento del trastorno de lateralidad y ubicación espacial.

El capítulo en sí proporciona una pequeña introducción a lo que es la disciplina de la ingeniería de software, y posteriormente detallará los procesos y principios de análisis y diseño del software que sustentan este proyecto. También se especifican las técnicas de documentación del software que son utilizadas para complementar el desarrollo del sistema que se propone. Aunque el área de estudio y de aplicación de la ingeniería de software abarca también las etapas más complejas de desarrollo y pruebas del software, éstas no se discuten en este capítulo porque se tratarán posteriormente en los capítulos correspondientes.

1.1. Ingeniería de Software

Ingeniería de Software es una disciplina o área de las ciencias de la computación que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelve problemas de todo tipo [PRR98].

Ingeniería de Software no es una disciplina que solo debe seguirse para proyectos de software que se encuentren pensados dentro de ciertas áreas, por el contrario, trata con

áreas muy diversas de las ciencias de la computación, tales como construcción de compiladores, sistemas operativos, o desarrollos en Internet como es muy cercanamente el caso de la aplicación de software de esta propuesta. La Ingeniería de Software abarca todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información aplicables a áreas tales como los negocios, investigación científica, medicina, producción, logística, banca, y – para el caso particular de este estudio – realidad virtual [PRR98].

Un aspecto muy importante de Ingeniería de Software es que proporciona parámetros formales para lo que se conoce como Gestión (o Administración) de Proyectos de Software. Esto se refiere a que Ingeniería de Software proporciona diversas métricas y metodologías que pueden usarse como especificaciones para todo lo referente a la administración del personal involucrado en proyectos de software, ciclos de vida de un proyecto de software, costos de un proyecto, y en si todo el aspecto administrativo que implica el desarrollar software. Por supuesto que estos aspectos no son relevantes para los fines de este proyecto, principalmente porque este proyecto no se desarrolla con fines lucrativos monetariamente hablando.

De acuerdo con Pressman [PRR98], Ingeniería en general es el análisis, diseño, construcción, verificación y gestión de entidades técnicas. En general, todo proceso de ingeniería debe comenzar por contestar las siguientes preguntas: ¿Cuál es el problema a resolver?, ¿Cuáles son las características de la entidad que se utiliza para resolver el problema?, ¿Cómo se realizará la entidad (y la solución)?, ¿Cómo se construirá la entidad?, ¿Cómo va a probarse la entidad?, y ¿Cómo se apoyará la entidad cuando los usuarios finales soliciten correcciones y adaptaciones a la entidad? Para los fines que se desarrolla el software propuesto dentro de este proyecto, podemos contestar estas preguntas en una primera instancia desde un punto de vista global y sin considerar detalles específicos, de tal manera que se pueden establecer los siguientes puntos:

- Desarrollar una aplicación de software que pueda utilizarse como auxiliar en el tratamiento del trastorno de lateralidad y ubicación especial.
- La aplicación de software debe tener características tales que se cumpla con el objetivo del proyecto, es decir, que el software este perfectamente orientado a sus usuarios para que realmente puede ser aplicado al área problema.

- El software se realizará bajo la siguiente premisa:
 - VRML 2.0 para modelar
 - Java para dar comportamiento
 - El Web para presentar
- La aplicación de software deberá ser probada en un intervalo de tiempo adecuado y lo suficientemente amplió como para poder obtener retroalimentación por parte de los usuarios y hacer las correcciones pertinentes. Deberá ser probada en sujetos reales que padezcan del trastorno de lateralidad y ubicación espacial.
- El software deberá estar documentado adecuadamente para facilitar futuros procesos tales como futuras expansiones ó adaptaciones a nuevas exigencias por parte de los usuarios finales.

Existen diferentes modelos de procesos para la Ingeniería de Software. Cada uno de estos modelos pretende de una manera u otra proporcionar lo más posible de orden al complicado proceso de desarrollar software. Para el caso de esta tesis es necesario apegarse lo más posible a uno de estos modelos con el fin de tener una organización de actividades que se planean en base a una serie de etapas lógicas e interconectadas entre si. El modelo de ingeniería de software que esta tesis sigue es el Modelo Lineal Secuencial, que será descrito a continuación.

1.1.1. Modelo Lineal Secuencial

El modelo lineal secuencial, también conocido como modelo en cascada, se basa en un enfoque sistemático y secuencial del desarrollo del software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas, y mantenimiento [PRR98]. La siguiente figura ilustra el modelo lineal secuencial para la ingeniería de software.



Figura 2.1 Modelo Lineal Secuencial

De acuerdo con Pressman, el modelo lineal secuencial contempla seis actividades que deben llevarse a cabo. A continuación se describen estas actividades, y se aterriza cada una de ellas a los fines de este proyecto:

Ingeniería y modelado de Sistemas. El software siempre forma parte de un contexto más grande, que puede ir desde una empresa hasta un sistema. El trabajo comienza estableciendo requisitos de todos los elementos del sistema, y asignando al software algún subgrupo de estos requisitos [PRR98]. En el caso de la herramienta de software que este proyecto propone, queda establecido el hecho de que el software es una aplicación aislada que no se incorpora a un sistema (computacional) más grande, pero si pertenece al contexto de las metodologías de tratamiento contra el trastorno de lateralidad y ubicación espacial, y es por esto que tienen que establecerse requerimientos funcionales y no funcionales que permitan que el software desarrollado pueda ubicarse exitosamente dentro de este contexto. En el capítulo 1, Marco Teórico, ya se establecieron algunos requerimientos generales que un ambiente virtual debe contemplar en sus primeras etapas de desarrollo, más no se han especificado los requerimientos particulares para la aplicación de software propuesta dentro de este proyecto. Estos requerimientos serán considerados posteriormente dentro de este capítulo.

Análisis de los requisitos del software. El proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Dentro del proceso de análisis es fundamental que a través de una colección de requerimientos funcionales y no funcionales, el desarrollador o desarrolladores del software comprendan completamente la naturaleza de los programas que deben construirse para desarrollar la aplicación, la función requerida, comportamiento, rendimiento e interconexión. [PRR98]. En el caso de este proyecto, el proceso de análisis y de obtención de requerimientos se lleva a cabo a través de trabajar conjuntamente con la psicóloga Norma Rodríguez, quien proporciona los parámetros bajo los cuales la aplicación debe desarrollarse para poder de esta manera cumplir con los objetivos de este proyecto. En la sección 2.3 se habla más profundamente de la etapa de análisis.

Diseño. Según Pressman, el diseño del software es realmente un proceso de muchos pasos pero que se clasifican dentro de uno mismo. En general, la actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software [PRR98].

Generación de Código. Esta actividad consiste en traducir el diseño en una forma legible por la máquina. En el caso de la aplicación de software de este proyecto, la generación de código se refiere tanto a la parte de generación de los ambientes virtuales, como a la parte en la cuál se añadirá comportamiento a estos ambientes. El lenguaje de programación VRML 2.0 es un lenguaje de modelado en 3D en el cuál se dibuja por medio de generar código de programación de formato y marcado para especificar las características del objeto u objetos que se van agregando a un mundo o entorno virtual. El comportamiento de las escenas virtuales es decir, su funcionalidad, se puede construir a través de algún otro lenguaje de programación, como clases Java o scripts especificados en JavaScript. Todas estas actividades implican generar código.

Pruebas. Una vez que se ha generado código, comienzan las pruebas del software o sistema que se ha desarrollado. De acuerdo con Pressman, el proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las prueba para la detección de errores [PRR98]. En el caso de una herramienta de software para tratar el trastorno de lateralidad y ubicación espacial, es necesario tener etapas de pruebas tanto para la parte funcional del software, como para la parte aplicativa del mismo. Se requiere poder probar el software con sujetos reales que puedan evaluar el comportamiento del software con el fin de proporcionar retroalimentación a los desarrolladores. Es sumamente importante que durante el proceso de desarrollo no se pierda el contacto con los interesados o solicitantes del desarrollo de software, de esta manera los objetivos de proyecto se mantendrán vigentes y se tendrá una idea clara de los aspectos que tienen que probarse durante el periodo de pruebas.

Mantenimiento. El software indudablemente sufrirá cambios, y habrá que hacer algunas modificaciones a su funcionalidad. Es de suma importancia que el software de

calidad pueda adaptarse con fines de acoplarse a los cambios de su entorno externo [PRR98]. Una de las secciones posteriores de este documento se refiere específicamente a posibles expansiones de este proyecto, y por medio de la documentación apropiada y atinada del software se pueden presentar las vías para el mantenimiento y modificaciones al mismo. En el capítulo 3, Desarrollo e Implementación, se describe paso a paso la manera en que se elaboran las escenas virtuales propuestas por este proyecto, de tal manera que este mismo documento puede utilizarse posteriormente como referencia al cómo deben desarrollarse aplicaciones que sigan la metodología de trabajo que este proyecto propone.

1.2. Ingeniería de Software Educativo (ISE)

Como su nombre lo dice, la Ingeniería de Software Educativo es una rama de la disciplina de la ingeniería de software encargada de apoyar el desarrollo de aplicaciones computacionales que tienen como fin implementar procesos de aprendizaje desde instituciones educativas hasta aplicaciones en el hogar. Si lo que se pretende es lograr aplicaciones de software que califiquen como educativas, es necesario que dentro de las fases de análisis y diseño de las mismas se añadan aspectos didácticos y pedagógicos con el fin de poder garantizar la satisfacción de las necesidades educativas en cuestión. Es de suma importancia involucrar efectivamente a los usuarios, para poder identificar necesidades que debe cubrirse durante la etapa de desarrollo [GORG97].

En el caso particular de este proyecto, la aplicación basada en ambientes virtuales que se propone es una herramienta computacional con fines educativos, es por esto que desde la primera etapa del proceso de Ingeniería de Software que se siguió, se involucró a la psicóloga Norma Rodríguez, quien desde un principio proporcionó una serie de consideraciones didácticas que deben atacarse desde el punto de vista de análisis y diseño de la aplicación para que a través de realidad virtual pueda tratarse el Trastorno de Lateralidad y Ubicación Espacial.

Dentro del proceso de Ingeniería de Software Educativo que se utiliza dentro de este proyecto, es también necesario incorporar la distinción de los diferentes elementos que de acuerdo con López, Escalera, y Ledesma conforman un Ambiente Virtual de Aprendizaje [LOEL02]. La identificación y aterrizaje de cada uno de estos elementos

funge un papel sumamente importante en las etapas de análisis y diseño de la herramienta aplicativa que este proyecto propone. Estos elementos son los siguientes:

- **Usuarios.** Son aquellos que van a aprender a través del Ambiente Virtual de Aprendizaje. En el caso de este proyecto, se refiere a aquellas personas que padecen del trastorno de lateralidad y ubicación espacial, y de personas a quienes se les implanta por primera vez educación acerca de las nociones básicas.
- **Contenido.** Es lo que se va a aprender. Para este proyecto, se establece que lo que se va a aprender es un conocimiento general de las nociones básicas de lateralidad y reconocimiento espacial en personas que presentan deficiencias en estas distinciones.
- **Especialistas.** Se refiere al cómo se va a aprender, cómo se van a materializar todos los contenidos educativos que se utilizarán en proceso de aprendizaje. Generalmente el grupo de especialistas consiste en personas con diferentes especialidades, desde el pedagogo hasta los programadores y diseñadores de los entornos virtuales.
- **Acceso, infraestructura y conectividad.** Este elemento se refiere a la arquitectura general de los ambientes virtuales que van a desarrollarse. Como ya se ha mencionado anteriormente, este proyecto se desarrolla principalmente en VRML 2.0 con el fin de aprovechar todas las ventajas de portabilidad que presenta Internet. La aplicación no será incorporada a un sistema ya existente y más grande, pero si se contemplará a nivel teórico la manera en que este proyecto podría expandirse para aumentar el rango de aplicación de este proyecto. Estas expansiones se discuten en el capítulo 5.

En 1991, Galvis propone una metodología para la Ingeniería de Software Educativo que se asemeja mucho a la metodología que establece el modelo lineal secuencial descrito anteriormente, y que sigue este proyecto. Esta metodología establece mecanismos de análisis, y diseño educativo y comunicacional de validez comprobada [GORG97]. La figura 2.2 ilustra este modelo:

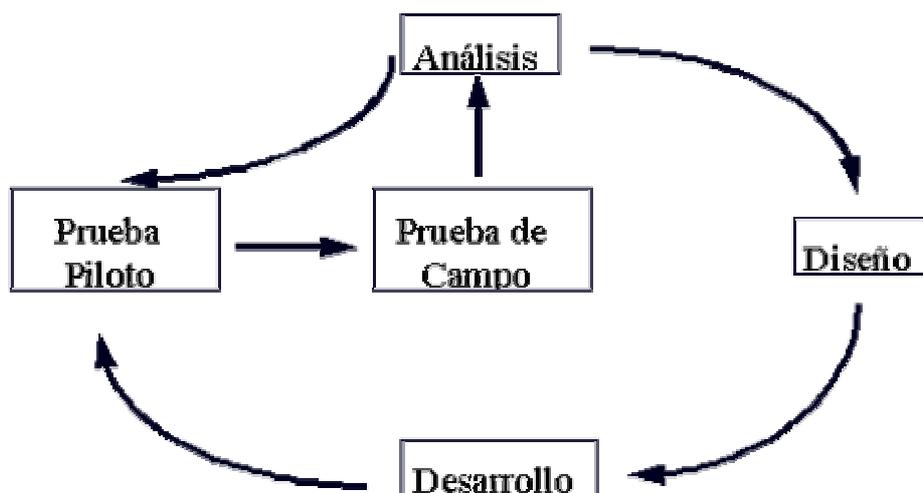


Figura 2.2: Metodología ISE propuesta por Galvis [GORG97].

Como ya se mencionó anteriormente, la Ingeniería de Software Educativo establece actividades que tienen que ser complementadas por medio de la incorporación de los aspectos didácticos y pedagógicos. El como se incorporan estos aspectos al proceso de ingeniería de software que sigue este proyecto se comentará en la secciones siguientes.

1.3.Análisis

Para que el desarrollo de un proyecto de software concluya con éxito, es de suma importancia que antes de empezar a codificar los programas que constituirán la aplicación de software completa, se tenga una completa y plena comprensión de los requisitos del software.

Pressman establece que la tarea del análisis de requisitos es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refina en detalle el ámbito del software, y se crean modelos de los requisitos de datos, flujo de información y control, y del comportamiento operativo. Se analizan soluciones alternativas y se asignan a diferentes elementos del software. El análisis de requisitos permite al desarrollador o desarrolladores especificar la función y el rendimiento del software, indica la interfaz del software con otros elementos del sistema y establece las

restricciones que debe cumplir el software. El análisis de requisitos del software puede dividirse en cinco áreas de esfuerzo, que son:

1. **Reconocimiento del problema.** Reconocer los elementos básicos del problema tal y como los perciben los usuarios finales.
2. **Evaluación y síntesis.** Definir todos los objetos de datos observables externamente, evaluar el flujo y contenido de la información, definir y elaborar todas las funciones del software, entender el comportamiento del software en el contexto de acontecimientos que afectan al sistema.
3. **Modelado.** Crear modelos del sistema con el fin de entender mejor el flujo de datos y control, el tratamiento funcional y el comportamiento operativo y el contenido de la información.
4. **Especificación.** Realizar la especificación formal del software
5. **Revisión.** Un último chequeo general de todo el proceso.

En el caso de una herramienta de software que puede emplearse como auxiliar en el tratamiento de lateralidad y ubicación espacial, el análisis de requerimientos debe llevarse a cabo en base a las necesidades de los terapeutas y de los pacientes. Debido a que el software que este proyecto propone no está basando en algún sistema anterior, la funcionalidad del software debe emular lo más atinadamente posible las actividades de juego con las que se trata este trastorno hoy en día.

Como se mencionó en el capítulo anterior, el tratamiento al trastorno de lateralidad y ubicación espacial en pacientes que tienen de 5 a 14 años de edad se lleva a cabo por medio de actividades de juego que regularmente se presentan al paciente a través de libros de ejercicios y otros impresos en los que se encuentran plasmadas estas actividades. La premisa esencial de este proyecto es entonces, el llevar estas actividades de los libros a entornos virtuales con comportamientos implementados de tal manera que se conserve la eficiencia de las actividades de juego presentadas en papel, pero con las ventajas agregadas que tiene la Terapia de Exposición usando Realidad Virtual.

En el caso de este proyecto el proceso de análisis comenzó con una primera entrevista con la psicóloga Norma Rodríguez. En esta primera entrevista se platicó acerca de lo que es el trastorno de lateralidad y ubicación espacial, como se manifiesta, sus impactos

en los padecientes, y más particularmente de las metodologías de tratamiento que se aplican para ayudar a los pacientes a superar sus problemas consecuentes. En vista de que este proyecto se plantea como un proyecto de aplicación computacional, y no de estudio neurofisiológico, la parte fundamental de la etapa de análisis es la que se realizó alrededor de la metodologías de tratamiento, que es precisamente lo que se pretende modelar dentro de entornos virtuales tridimensionales accesibles a través de Internet.

Los requerimientos de sistema que se definen en la etapa de análisis de un proceso de Ingeniería de Software generalmente se clasifican como requerimientos funcionales y requerimientos no funcionales. Los principales requerimientos funcionales y no funcionales de la herramienta de software que este proyecto propone se definen en las dos secciones siguientes.

1.3.1. Requerimientos Funcionales

Los requerimientos funcionales son los que se encargan de definir lo que la herramienta de software debe hacer. Definen los alcances del sistema en cuanto a las acciones que debe de realizar, y en cuanto a la transferencia de datos entre todas las diferentes funciones del sistema [KII02].

En el caso de este proyecto, los principales requerimientos funcionales son los siguientes:

1. **Sensibilidad a la presencia de los usuarios.** Este proyecto gira en torno a la posibilidad de poder modelar el tratamiento del trastorno de lateralidad y ubicación espacial dentro de ambientes virtuales. Estos ambientes virtuales deben de tener la posibilidad de reaccionar a las acciones de un usuario que se encuentre utilizando el sistema, dado que se debe tener una cierta interacción y comunicación entre los usuarios del ambiente virtual y el ambiente virtual en si mismo. VRML 2.0 proporciona dentro su especificación diferentes nodos que permiten a un mundo virtual, o a un objeto contenido dentro de un mundo

virtual, el disparar ciertas acciones como respuesta a acciones o eventos que el usuario efectúe. Estas acciones y eventos deben ser capturadas y procesadas por el mundo virtual, para producir respuestas al usuario del entorno virtual. La manera en que se implementa e inyecta esta sensibilidad a los ambientes virtuales se discute a detalle en el capítulo 3 (Desarrollo e Implementación.)

2. **Funcionalidad Global.** Esta herramienta de software debe de tener la facultad de desplegar y presentar diferentes actividades con ejercicios de los que tradicionalmente se utilizan para tratar el trastorno de lateralidad y ubicación espacial. Las actividades tendrán que ser desplegadas bajo petición de un usuario, y tendrán que tener comportamientos específicos y detallados con respecto a las acciones que el usuario tenga que realizar para resolver el problema que se presente a través del ambiente virtual activado. La gran mayoría de la funcionalidad de los ambientes virtuales estará dada a través de las opciones funcionales que ofrece el lenguaje VRML en su versión 2.0, y en aquellos casos donde se requiera funcionalidad más compleja, se completará VRML a través de Java.

3. **Alcance.** Con anterioridad ya se ha mencionado que el objetivo de este trabajo es modelar dentro de ambientes virtuales el tratamiento al trastorno en cuestión. Sin embargo, es importante mencionar que esta herramienta no tiene como objetivo el funcionar como un asistente para diagnosticar este trastorno. El alcance de este proyecto se limita a presentar dentro de mundos virtuales las actividades de juego que se utilizan para tratar el trastorno de lateralidad y ubicación espacial.

Los mencionados anteriormente son los requerimientos funcionales particulares de este proyecto, sin embargo, en el Capítulo 1, (Marco Teórico), se discuten requerimientos generales que deben contemplarse dentro del desarrollo de todo ambiente virtual. En el caso particular de este proyecto, los requerimientos fundamentales son aquellos establecidos por los requerimientos no funcionales. Estos se discuten en la sección correspondiente.

1.3.2. Requerimientos No Funcionales

Los requerimientos no funcionales son aquellos que definen lo que la herramienta de software debe tener en cuanto a apariencia, sensación, operabilidad, y mantenimiento [KII02].

De acuerdo con Galvis, el objetivo de la etapa de análisis cuando se sigue una metodología de ISE (Ingeniería de Software Educativo), es determinar el contexto en el cual se va a crear la aplicación para poder derivar los requerimientos que deberá atender la solución interactiva como complemento a otras soluciones basadas en uso de otros medios, teniendo en claro el rol de cada uno de los medios educativos seleccionados y la viabilidad de usarlos [GORG97]. Tomando en cuenta esta consideración, y trabajando en conjunto con la psicóloga Norma Rodríguez, es como se derivan los requerimientos no funcionales que debe cubrir la herramienta de software propuesta por este estudio.

Galvis establece que para recolectar este tipo de requerimientos hay que por lo menos tomar en cuenta la siguiente información:

- **Características de la población objetivo.** Se refiere a cuestiones como la edad, características físicas y mentales, experiencias previas, expectativas, actitudes, aptitudes, o intereses. En el caso de este estudio, es muy importante considerar que el rango de edad de los pacientes que van a tratarse es de 5 a 14 años de edad. No todos los pacientes saben leer y escribir. Las características físicas son completamente irrelevantes, y se da por sentado que los pacientes manifiestan problemas en cuanto a sus conocimientos de las nociones básicas. Se asume que ningún paciente tiene experiencia con ambientes virtuales, pero a través de la información que se obtuvo por medio de la psicóloga que apoya este estudio, se sabe que algunos de estos pacientes tienen experiencia con computadoras y con videojuegos. Esto último siendo importante dado que si un paciente tiene experiencia con videojuegos, le será más fácil desenvolverse dentro de los ambientes virtuales que este proyecto propone.
- **Conducta de entrada y campo vital.** Es necesario ubicar la herramienta de software dentro de las áreas bajo las cuales se desenvuelve el paciente. Es importante considerar aspectos como el nivel escolar y desarrollo mental. A

través de una investigación preeliminar, para este proyecto se sabe que el nivel escolar es esencialmente de nivel primaria, y que las actividades de juego y tratamiento que pretenden modelarse están diseñadas para pacientes a este nivel educativo.

- **Problema o necesidad a atender.** Como su nombre lo dice, es necesario ubicarse dentro del contexto del problema que pretende atacarse. En el caso de esta tesis el objetivo es modelar el tratamiento al trastorno de lateralidad y ubicación espacial a dentro de ambientes virtuales interactivos.
- **Justificación de los medios interactivos a utilizar.** Galvis establece que el apoyo informático debe ser tomado en cuenta siempre y cuando no exista un mecanismo mejor para resolver el problema. El caso de este proyecto, se tienen varias justificaciones que pueden apreciarse desde distintos puntos de vista. Primero que nada, los desarrollos dentro de las ciencias computacionales se caracterizan por llevar actividades manuales a procesos automatizados, y con esto se facilita el trabajo. En el capítulo 1, se mencionan la grandes ventajas que se han manifestado a través de la utilización de la Terapia de Exposición Usando Realidad Virtual, y son precisamente estas ventajas las que se buscan incorporar al tratamiento del trastorno en cuestión. Otro aspecto importante es que a través de utilizar la plataforma del Web como presentador, el potencial de portabilidad y de crecimiento es mucho mayor que a través de los procesos manuales impresos. Por último, el presentar un tratamiento a través de Realidad Virtual y computadoras, resulta muy llamativo para pacientes con las características definidas para la población objetivo; al ser mas llamativo, la apertura a recibir tratamiento es mayor y con esto se agiliza el tratamiento mismo.

Los puntos anteriores conducen a este proyecto a elaborar una lista de requerimientos no funcionales dentro los cuales destacan los siguientes:

1. **Clasificación de las actividades.** Como ya se mencionó anteriormente, el rango de edad de los pacientes que serán tratados a través de esta herramienta es un poco amplio, por lo que sería absurdo asumir o pensar que una actividad es apropiada para todos los pacientes. La psicóloga Norma Rodríguez especificó que las actividades que conforman el tratamiento a este trastorno se encuentran clasificadas dentro de niveles de dificultad que van de acuerdo a la edad de los

pacientes. A los pacientes más chicos, por ejemplo, no se les pueden introducir conceptos o etiquetas como “derecha” o “izquierda” si aún no saben leer, pero si se les puede hacer distinguir los lados sin necesidad de poner nombres. Este requerimiento se refiere a que esta clasificación de actividades debe respetarse dentro de la herramienta de software aplicativo.

2. **Instrucciones claras.** Es necesario proporcionar a los usuarios instrucciones que describan el objetivo que debe cumplirse dentro de cada una de las actividades que se encuentren modeladas dentro de las escenas virtuales. En vista de que no todos los pacientes saben leer, desplegar las instrucciones escritas en pantalla no es suficiente. Es por esto que la herramienta de software debe de ser capaz de proporcionar las instrucciones de manera clara y precisa, y que todos los usuarios puedan entender cuales son los objetivos que tienen que ser alcanzados. Se hace necesario entonces el uso de archivos de sonidos.
3. **Operabilidad.** En el capítulo 1 se menciona lo importante que es identificar al usuario apropiadamente. Se estableció que este proyecto identifica a sus usuarios como terapeutas y pacientes que no tienen experiencia alguna con Realidad Virtual. Es por esto que los usuarios tienen que poder interactuar con las escenas virtuales sin la necesidad de controles complejos que pueden quitarle el atractivo a esta metodología de tratamiento. Podría decirse que los usuarios finales son los pacientes, ya que en un momento dado el terapeuta puede o no estar presente.
4. **Conservar características de las metodologías tradicionales.** El trastorno de lateralidad y ubicación espacial se trata a través de actividades de juego que tienen ciertas características que las hacen apropiadas para ser utilizadas como metodologías de tratamiento a este padecimiento neurofisiológico. El hecho de que estas actividades vayan a ser presentadas a través de entornos tridimensionales, exige que estas características se conserven.

En la sección Diseño, se discute como la lista de requerimientos funcionales y no funcionales se traduce en una aplicación de software basada en entornos virtuales interactivos.

1.4.Diseño

Anteriormente se mencionó que la etapa de diseño es cuando se traducen los requerimientos funcionales y no funcionales en una representación de software. El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería. De acuerdo con Pressman, el objetivo del diseño es producir un modelo o representación de una entidad que se va a construir posteriormente [PRR98].

De acuerdo con McGlaughlin [MCG91], hay tres características que sirven como parámetros generales para la evaluación de un buen diseño. Estos parámetros son los siguientes:

1. El diseño debe implementar todos los requisitos explícitos obtenidos en la etapa de análisis.
2. El diseño debe ser una guía que puedan leer y entender los que construyen el código y los que prueban y mantienen el software.
3. El diseño debe proporcionar una idea completa de lo que es el software.

En la sección siguiente se establecen tipos diferentes de diseño que la etapa de diseño del proceso de ingeniería de software produce.

1.4.1. Diseño del Software

El diseño del software desarrolla un modelo de instrumentación o implantación basado en los modelos conceptuales desarrollados durante el análisis del sistema. Implica diseñar la decisión sobre la distribución de datos y procesos [MAJO97].

El diseño es la primera de las tres actividades técnicas que implica un proceso de ingeniería de software; estas etapas son diseño, codificación (en el caso de este proyecto Desarrollo e Implementación) y pruebas. Generalmente la fase de diseño produce un diseño de datos, un diseño arquitectónico, un diseño de interfaz, y un diseño procedimental [PRR98].

El diseño de datos esencialmente se encarga de transformar el modelo de dominio de la información creado durante el análisis [PRR98]. En el caso particular de este proyecto el diseño de datos no juega un papel determinante dado que la herramienta de software propuesta, de la manera en que será físicamente desarrollada e implementada, no requiere de estructuras de datos complejas, ni de un esquema de base de datos por ejemplo.

En el diseño arquitectónico se definen las relaciones entre los principales elementos estructurales del programa [PRR98]. Para una herramienta de software basada en el desarrollo e implementación de ambientes virtuales éste es un aspecto fundamental dado que en esta representación del diseño se establece la estructura modular del software que se desarrolla. Dado que este proyecto pretende proponer una metodología de tratamiento al trastorno de lateralidad y ubicación espacial a través de Realidad Virtual, la codificación y generación de ambientes y entornos virtuales es esencial. Cuando se utiliza VRML 2.0 es necesario codificar cada una de las instrucciones que crearán un objeto determinado con sus propias características y atributos. Si se pretendiera codificar por completo toda una escena virtual dentro de un mismo archivo, el archivo crecería superlativamente y su manipulación, adaptación, y mantenimiento se volverían tareas bastante complejas e incómodas. Afortunadamente, a través del nodo *Inline* de la especificación 2.0 de VRML puede darse un alto nivel de modularidad a los mundos virtuales dado que cada objeto puede describirse o codificarse por separado, para posteriormente ser referenciado dentro de la escena virtual contenedora. El nodo *Inline* se detalla en el capítulo siguiente.

El diseño de interfaz describe cómo se comunica el software consigo mismo, con los sistemas que operan con él, y con los operadores que lo emplean [PRR98]. En el caso de la herramienta de software propuesta por este estudio la interfaz del software consigo mismo se lleva a cabo de 2 maneras:

- Nodos de VRML 2.0 se comunican con otros nodos ¹
- Nodos que se comunican con Scripts de comportamiento descritos en Java o en JavaScript.

¹ Este proceso se conoce como ROUTING. Los detalles serán discutidos en el capítulo 3 (Desarrollo e Implementación).

VRML es un lenguaje de modelado diseñado específicamente para integrarse a la plataforma de Internet. Es por este que para los fines de este proyecto se antoja lógico el desarrollar la interfaz con los operadores del software a través de HTML, VRML, JavaScript, o cualquier otra tecnología que puede incorporarse a las especificaciones de esta plataforma.

De acuerdo con Pressman, el diseño procedural transforma elementos estructurales de la arquitectura del programa en una descripción procedural de los componentes del software [PRR98].

1.4.2. Arquitectura del Software

El diseño de la arquitectura del software se refiere a la estructura global del software y las maneras en que esa estructura proporciona integridad conceptual a un sistema [SHA95]. De acuerdo con Pressman, en su forma más simple, la arquitectura es la estructura jerárquica de los módulos del programa, la manera de interactuar de estos componentes, y la estructura de los datos usados por estos módulos [PRR98].

La arquitectura del software que este proyecto propone como una herramienta aplicativa dentro de un contexto real, está pensada de acuerdo a las propiedades que Shaw y Garlan [SHA95] describen como aspectos que deben especificarse como partes de un buen diseño arquitectónico. Estos tres aspectos son: Propiedades estructurales, Propiedades extra-funcionales, y Familias de Sistemas Relacionados. En las siguientes sub-secciones se elaborará con respecto a cada uno de estos tres aspectos respectivamente.

1.4.2.1. Propiedades Estructurales

Este es el aspecto de la representación de software que define los componentes de un sistema, y la manera en que se empaquetan estos componentes e interactúan unos con los otros. Este proyecto propone ambientes virtuales integrados para formar una aplicación conjunta que puede presentarse apropiadamente a los usuarios. La

especificación de la arquitectura del software tiene que respetar fuertemente el concepto de modularidad del software, ya que es necesario poder identificar los componentes individuales que al unirse entre si forman un ambiente virtual.

La manera en que este proyecto de software se encuentra estructurado tiene que ser especificada desde su componente más básico hasta los usuarios finales. De esta manera, se obtiene la tan necesaria modularidad que según Myers [MYE78] es un atributo del software que permite a un programa ser manejable intelectualmente. Un programa grande compuesto de un solo módulo no puede ser entendido fácilmente por un lector. El número de caminos de control, número de variables y la complejidad global harían se comprensión casi imposible. Como ya se mencionó anteriormente, VRML permite modularidad a través de ofrecer la posibilidad de crear los componentes de una escena virtual por separado.

La figura 2.3, un modelo entidad-relación, ilustra la cohesión general que existe entre los diferentes elementos de la herramienta de software que este estudio implementa

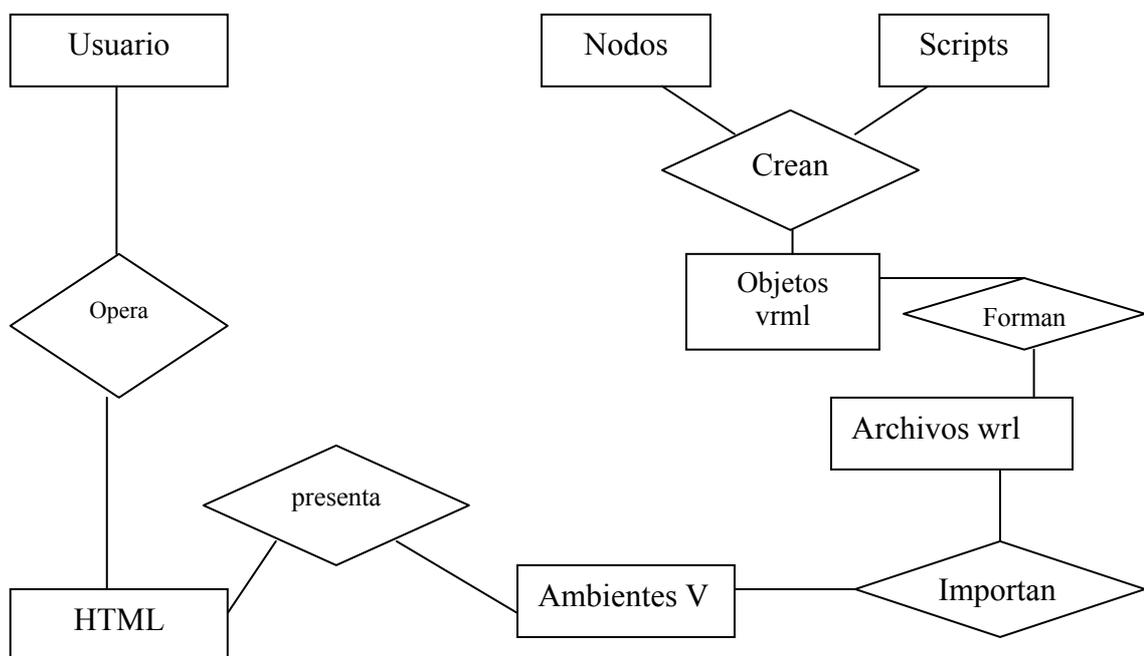


Figura 2.3 Modelo Entidad Relación para la arquitectura del sistema

El modelo entidad relación se explica de la siguiente manera. Los dos componentes más básicos de la herramienta de software que este proyecto propone son los nodos del lenguaje VRML 2.0 y los Scripts de comportamiento en Java y JavaScript. Los nodos y los Scripts crean y definen objetos de VRML que tiene sus propios atributos y características. Un conjunto o agrupación de objetos VRML forman archivos **.wrl* que en base a los objetos que los conforman pueden describir elementos más complejos de una escena virtual. Los ambientes virtuales importan una serie de archivos **.wrl* que una vez integrados forman el ambiente virtual, que se incrusta en un documento HTML que es lo que da la interfaz con el usuario. De esta manera, se tiene una modularidad total con respecto a la arquitectura del software, ya que para modificar un objeto VRML se redefine ya sea el nodo(s) o el script(s) que lo describe. A su vez, para realizar modificaciones sobre un archivo **.wrl* pueden manipularse individualmente los objetos que lo conforman, y de esta manera se puede dar mantenimiento a un ambiente virtual sin necesidad de manipular una cantidad excesiva de líneas de código.

1.4.2.2. Propiedades Extra-funcionales

Esta especificación se refiere al cómo consigue la arquitectura del diseño los requisitos de rendimiento, capacidad, fiabilidad, seguridad, adaptabilidad, y otras características de la herramienta de software [SHA95]. En las secciones correspondientes a los requerimientos funcionales y a los no funcionales se establecieron los principales requerimientos funcionales y no funcionales que se consideran para el desarrollo de la herramienta de software, a continuación se presenta el registro de cada uno de estos requerimientos, y una explicación acerca de cómo el diseño de la herramienta de software propuesta incorpora dentro de sus características esta lista de requerimientos.

1. Sensibilidad a la presencia de los usuarios

- **ID del requerimiento** – rf01
- **Descripción** – Es necesario que los ambientes virtuales propuestos dentro de este proyecto tengan a posibilidad de captar acciones producidas por los usuarios para generar respuestas. Con esto las escenas virtuales se vuelven interactivas.
- **Tipo** – Requerimiento funcional

- **Caso de Uso** – Esta funcionalidad debe estar presente en todos los ambientes virtuales generados dentro de este proyecto, ya que todos los ambientes deben ser interactivos.
- **Fuente** – Psicóloga Norma Rodríguez, y Maestra Carolina Castañeda.
- **Criterio de Evaluación** – El criterio de evaluación para este requerimiento no es algo cuantificable. La evaluación de este requerimiento se hace en base a si la interacción que tienen los ambientes virtuales contribuye el cumplimiento de los objetivos aplicativos del software.

Implementación – Este requerimiento se implementa dentro del sistema por medio de la inclusión de sensores de contacto, proximidad, y tiempo que sean capaces de capturar un evento producido por un usuario para procesarlo y convertirlo en una respuesta que se envía al usuario. Cuando los sensores disponibles dentro de VRML 2.0 no son suficientes, se utiliza Java o JavaScript para proporcionar esta funcionalidad adicional.

2. Funcionalidad Global

- **ID del requerimiento** – rf02
- **Descripción** – Esta herramienta de software debe de tener la facultad de desplegar y presentar diferentes actividades con ejercicios de los que tradicionalmente se utilizan para tratar el trastorno en cuestión.
- **Tipo** – Requerimiento Funcional
- **Caso de Uso** – Este requerimiento constituye en sí mismo todo lo referente a los casos de uso del software propuesto, ya que es a través de la presentación de diferentes escenas virtuales que se pretende interactuar con los usuarios.
- **Fuente** – Psicóloga Norma Rodríguez
- **Criterio de Evaluación** – Presentar diversas escenas virtuales interactivas con funcionalidad apropiada al tratamiento de este trastorno.

Implementación – Este requerimiento se implementa a través del modelado de diferentes escenas virtuales con funcionalidades apropiadas y correspondientes.

3. Clasificación de las actividades

- **ID del requerimiento** – rnf01
- **Descripción** – Considerando que el nivel de afectación que presentan los pacientes en base al trastorno, y dado que el rango de edad de la población objetivo es de 5 a 14 años de edad, es necesario tener dentro del software una clasificación de actividades, para que de esta manera haya actividades apropiadas para cada paciente.
- **Tipo** – Requerimiento no funcional.
- **Caso de Uso** – Esta clasificación debe conservarse en todo los casos de uso de esta de la herramienta propuesta por este estudio.
- **Fuente** – Psicóloga Norma Rodríguez
- **Criterio de Evaluación** – Se evaluará en base a si la clasificación de actividades que presenta dentro de los entornos virtuales cumple o no con la clasificación establecida por el tratamiento tradicional al tratamiento de este trastorno.

Implementación – Dentro del ambiente virtual que funcionará como contenedor de todas las demás escenas virtuales, se establecen zonas en la cuales se encuentran los enlaces a las diferentes actividades correspondientes a cada nivel de dificultad. De esta manera, un paciente determinado tendrá la posibilidad de desplazarse hacia la zona donde se encuentren las actividades que le corresponden.

4. Instrucciones Claras

- **ID del requerimiento** – rnf02
- **Descripción** – Como los usuarios de la herramienta de software son principalmente niños, es de suma importancia que se les presenten instrucciones claras y precisas, tanto de manera escrita como en voz acerca de que es lo que tienen que hacer dentro de cada ambiente virtual de actividad.
- **Tipo** – Requerimiento no funcional
- **Caso de Uso** – Cuando los usuarios soliciten instrucciones acerca de lo que se tiene que hacer.
- **Fuente** – Psicóloga Norma Rodríguez

- **Criterio de Evaluación** – Se evaluará en base a la facilidad o dificultad con la que los pacientes entienden las instrucciones que les proporciona el entorno virtual.

Implementación – Existe la opción de presentar las instrucciones al paciente a través de texto. Sin embargo, esto no es suficiente ya que algunos de los pacientes más jóvenes no saben leer. Es por esto que también existe la opción de presentar las instrucciones a través de mensajes de voz.

5. Operabilidad Sencilla

- **ID del requerimiento** – rnf03
- **Descripción** – En vista de que se ubica a los usuarios de este software como terapeutas y pacientes que no tienen experiencia previa con la manipulación e interacción de ambientes virtuales, la operabilidad de los mismos tiene que ser lo mas sencilla posible.
- **Tipo** – Requerimiento no funcional
- **Caso de Uso** – La operabilidad tiene que mantenerse lo más simple posible en todos los casos de uso
- **Fuente** – Psicóloga Norma Rodríguez, Maestra Carolina Castañeda.
- **Criterio de Evaluación** – En base a que tan bien pueden los pacientes operar el sistema.

Implementación – La operabilidad del sistema se hará a través del mouse o ratón y de las flechas de izquierda, derecha, arriba, y abajo del teclado. Como la presentación se hará a través de documentos HTML, el mouse es el dispositivo ideal de navegación. Los VRML browsers tienen controles adicionales, pero estos son un tanto complejos y es por eso que serán deshabilitados para que los usuarios puedan desplazarse dentro de los ambientes virtuales utilizando el mouse y las flechas del teclado.

6. Conservar Características de las Metodologías Tradicionales

- **ID del requerimiento** – rnf04

- **Descripción** – Es necesario que los ambientes virtuales que se utilicen para tratar el trastorno en cuestión conserven los atributos que las actividades tradicionales tienen, ya que son estos atributos los que hacen que las actividades sean o no apropiadas.
- **Tipo** – Requerimiento no funcional
- **Caso de Uso** – Todos los ambientes virtuales de actividad tienen que conservar estas características
- **Criterio de Evaluación** – La psicóloga Norma Rodríguez debe aprobar o rechazar las actividades contenidas de los entornos virtuales

Implementación - Las actividades de juego que se modelan dentro de los ambientes virtuales estarán basadas en los libros de texto que actualmente se utilizan para tratar el trastorno de lateralidad y ubicación espacial, de esta manera se tiene certeza de que los ambientes virtuales modelan un tratamiento útil y con resultados comprobados.

En el capítulo 3, Desarrollo en Implementación, se elabora más a detalle con respecto a la implementación de estos requerimientos.

1.4.2.3. Familias de Sistemas Relacionados

Este aspecto se refiere a que el diseño debería tener la capacidad de utilizar bloques de construcción arquitectónica reutilizados [SHA95]. La herramienta de software que este estudio pretende implementar no está basada en algún sistema interior, ni busca ser incorporada dentro de un sistema más grande. Es por esto que dentro del diseño no pueden contemplarse elementos provenientes de otros sistemas precedentes.

1.4.3. Diseño de la Interfaz

Diseñar la interfaz de usuario es un proceso que empieza con la creación de diferentes modelos de función del sistema. De acuerdo con Rubbin [RUB88], durante este proceso se crean los siguientes modelos:

- Un modelo del diseño del sistema, que incorpora representaciones de datos, arquitectónicos, de interfaces y procedimentales del software. Anteriormente se habló acerca de la estructura arquitectónica que el software de este proyecto implementa.
- Un modelo de usuario que muestra el perfil de los usuarios finales del sistema. En la etapa de análisis de este proyecto se ubica a la población objetivo que en su totalidad esta constituida por niños. Es por esto que los entornos virtuales que se presenten a los usuarios deben ser agradables para niños, y sumamente manipulables sin la necesidad de controles complejos.

Normalmente el proceso del diseño de la interfaz del software es un proceso que se lleva a cabo en colaboración con los usuarios finales del mismo. Este estudio hasta el momento ha establecido que la población objetivo no tiene experiencia previa en cuanto a ambientes virtuales se refiere. Por ende, es válido afirmar que la percepción del sistema por parte de los usuarios finales es muy limitada. El proceso de diseño de la interfaz se basa entonces en tomar en cuenta cuidadosamente las características de la población objetivo, y en crear modelos preliminares que posteriormente son ajustados y modificados en base a las recomendaciones y observaciones que hagan los usuarios finales. Durante el proceso de desarrollo se mantienen reuniones periódicas con un terapeuta, al cual se le van mostrando los avances con el fin de obtener retroalimentación.

Por las mismas herramientas computacionales que se incorporan para desarrollar una herramienta de software con los alcances establecidos por este proyecto, hay algunos elementos de la interfaz con el usuario que no solo son los más convenientes sino también los más apropiados. Las consideraciones técnicas que tienen que ser tomadas en cuenta para desarrollar la interfaz son las siguientes

- VRML 2.0 tiene que ser desplegado por un VRML browser
- Un VRML browser es un plug-in que se integra a un Internet browser regular.
- El VRML browser tiene comandos especiales para manipular las escenas virtuales

- Una escena virtual desarrollada en VRML 2.0 es, esencialmente, un documento presentable en Internet que esta descrito en 3 dimensiones.

En base a estas consideraciones, y tomando en cuenta que uno de los atributos principales de la herramienta de software que este estudio propone es el hecho de que incorpora la portabilidad que la plataforma que Internet ofrece, lo más práctico y conveniente es diseñar una interfaz de usuario a través de un navegador de Internet.

Otro aspecto que es sumamente importante es el hecho de que el VRML browser por si solo tiene controles avanzados para navegar dentro de una escena virtual en 3 dimensiones. Considerando que la población objetivo tiene problemas de ubicación y reconocimiento del espacio, sería una contradicción enorme el pensar que los controles del VRML browser no presentarían problemas de uso a la población objetivo. Es por esta razón que los controles del VRML browser tienen que ser desactivados, para poder introducir como instrumento de navegación el mouse y las flechas (izquierda, derecha, arriba, y abajo) del teclado de la computadora.

En términos generales puede decirse que la interfaz de este proyecto está dada por páginas web que tengan incrustadas las escenas virtuales desarrolladas. Los detalles de cómo se implementa y construye la interfaz están dados en el capítulo siguiente.

1.5.Documentación

La documentación de este proyecto de software se presenta en base a los método orientados a objetos propuestos por Martin y Odell en su publicación “Métodos Orientados a Objetos: Consideraciones prácticas” [MAJO97]. Se utilizarán las siguientes técnicas para documentar los componentes más relevantes de la herramienta de software:

- Diagramas de eventos – Para ilustrar la manera en que un usuario del software interactúa con los entornos virtuales
- Diagramas de contexto – Para ubicar el campo de acción que abarcará el software

- Tarjetas CRC – utilizada para representar todas las clases dentro de un diseño. Este proyecto en sí no está apegado estrictamente al concepto de programación o diseño orientado a objetos, sin embargo si se utilizan algunas clases de Java para dar comportamiento a algunos objetos descritos en VRML 2.0.

Referencias a metodologías de como documentar este proyecto, y alguna documentación relevante puede consultarse en el apéndice B incluido en este documento.