

Universidad de las Ciencias Informáticas
Facultad 5



**Diseño de un software para la estimulación
visual de niños Estrábicos y Ambliopes.**

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

Autora: Gretel García Hidalgo

Tutor: MSc. Pedro Carlos Pérez Martinto

Junio/2009

Autor: Gretel García Hidalgo

Dedicatoria

A mis padres. A mi hermano. A Eduar y Martinto

Gretel

Agradecimientos:

Agradezco ante todo a mi magnifico tutor MSc. Pedro Carlos Pérez Martinto, además de enormemente preocupado, conté siempre con su apoyo durante todo este tiempo, no solo docentemente sino también un gran apoyo moral.

Por último gratificar la ayuda que me han dado mis compañeras de grupo Daily, Lisbety, Lenna, Yanetsi, Lien, Yailín.....y otros que de alguna manera han contribuido a la realización de este trabajo.

RESUMEN

Con el creciente desarrollo de las tecnologías, la educación especial ha tomado en consideración el uso de las mismas para lograr una mejor rehabilitación de niños con ambliopía y estrabismo, así como llevar a cabo un adecuado control. La Educación Especial ha tenido como reto la explotación de estas tecnologías; apareado a esto, han surgido problemas para los cuales se plantea una propuesta de solución con el presente trabajo.

En las instituciones especiales se emplean recursos y métodos que no satisfacen todas las necesidades. Por lo anteriormente planteado es necesario vincular las tecnologías a objetivos más afines al cumplimiento de los objetivos de estos centros educacionales especializados.

Con el objetivo de dar solución al problema presentado, se acordó diseñar un Software que le permitirá al sujeto desarrollar habilidades óculo-manuales, además de lograr una mayor motivación dentro del proceso de tratamiento y un mejoramiento en el control de la información de los resultados obtenidos por cada sujeto.

Contenido

INTRODUCCIÓN.....	8
.....	12
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	12
1.1 ALGUNOS FUNDAMENTOS ANÁTOMO - FISIOLÓGICOS Y ÓPTICOS DE LA VISIÓN.....	13
1.2 CONSIDERACIONES ACERCA DE LA CORRECCIÓN Y/O COMPENSACIÓN DE LA AMBLIOPÍA	13
1.3 TECNOLOGÍA	15
1.4 ANÁLISIS DE INVESTIGACIONES REALIZADAS EN CENTROS DE REHABILITACIÓN VISUAL.....	18
1.5 MÉTODOS E INSTRUMENTOS EMPLEADOS.....	19
POSIBLES HERRAMIENTAS PARA DAR SOLUCIÓN A LA PROPUESTA DE SOFTWARE:.....	19
1.5.2 Plataforma de Desarrollo.....	19
1.6.1 SQL Server	24
1.7 Herramientas CASE (Computer Aided Software Engineering).....	27
<i>Las herramientas CASE son un conjunto de programas y ayudas que brindan ayuda y 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.....</i>	<i>27</i>
1.7.1 Soluciones para la realización del diseño.....	27
1.8 HERRAMIENTA DE MODELACIÓN VISUAL	35
1.8.1 Rational Rose Enterprise Edition	35
1.8.2 Visual Paradigm	35
1.9 SOLUCIÓN PROPUESTA.....	37
CONSIDERACIONES FINALES.....	38
CAPÍTULO 2: DEFINICIÓN DE REQUERIMIENTOS DEL SISTEMA.....	39
2.1 INTRODUCCIÓN:.....	39
2.2 MODELO DEL DOMINIO	41
2.3 DEFINICIÓN DE CONCEPTOS DEL MODELO DEL DOMINIO:.....	41
2.3.1 Glosario de Términos del Dominio:.....	41
<i>Posteriormente se reconocen los actores del sistema a desarrollar y se conciben, a través de la agrupación de los requerimientos funcionales, los posibles resultados de valor que le pueda brindar a sus actores, o lo que es lo mismo, los casos de uso del sistema. Además, se seleccionan los casos de uso arquitectónicamente significativos para hacerles sus descripciones textuales en formato expandido.....</i>	<i>44</i>
<i>Captura de Requerimientos:.....</i>	<i>44</i>
2.4 REQUERIMIENTOS FUNCIONALES:.....	44

<u>2.6 MODELO DE CASOS DE USOS DEL SISTEMA.....</u>	<u>48</u>
<u> 2.6.1 Descripción de los actores:.....</u>	<u>48</u>
<u> 2.6.2 Casos de uso del sistema:.....</u>	<u>51</u>
<u>CONSIDERACIONES FINALES</u>	<u>61</u>
<u>CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA.....</u>	<u>62</u>
<u>3.1 INTRODUCCIÓN</u>	<u>62</u>
<u>3.2 DESCRIPCIÓN DE LA ARQUITECTURA</u>	<u>63</u>
<u>3.3 DIAGRAMAS DE CLASES DE ANÁLISIS.....</u>	<u>64</u>
<u> 3.3.1 CU: Gestionar Usuario.....</u>	<u>64</u>
<u> 3.3.2 CU: Autenticar Usuario</u>	<u>65</u>
<u> 3.3.3 CU: Conformar Laberinto.....</u>	<u>65</u>
<u> 3.3.4 CU: Obtener Objetos.....</u>	<u>66</u>
<u> 3.3.5 CU: Ejecutar Pregunta Nivel.....</u>	<u>66</u>
<u> 3.3.6 CU: Revisar Resultados.....</u>	<u>67</u>
<u>3.4 DIAGRAMAS DE CLASES DE DISEÑO.....</u>	<u>68</u>
<u> 3.4.2 CU: Autenticar Usuario.....</u>	<u>69</u>
<u> 3.4.3 CU: Revisar Resultados.....</u>	<u>70</u>
<u> 3.4.4 CU: Conformar Laberinto.....</u>	<u>71</u>
<u> 3.4.5 CU: Obtener Objetos.....</u>	<u>72</u>
<u> </u>	<u>72</u>
<u> 3.4.6 CU: Ejecutar Pregunta Nivel.....</u>	<u>73</u>
<u>3.5 DIAGRAMAS DE SECUENCIA:.....</u>	<u>75</u>
<u> 3.5.3 CU: Gestionar Usuario: Eliminar Usuario</u>	<u>79</u>
<u> 3.5.4 CU: Autenticar Usuario.....</u>	<u>80</u>
<u> </u>	<u>80</u>
<u> 3.5.5 CU: Ejecutar Pregunta Nivel.....</u>	<u>81</u>
<u> 3.5.6 CU: Revisar Resultados.....</u>	<u>82</u>
<u> 3.5.7 CU: Conformar Laberinto.....</u>	<u>83</u>
<u> 3.5.8 CU: Obtener Objetos.....</u>	<u>84</u>
<u> </u>	<u>84</u>
<u>3.6 DIAGRAMA DE CLASES PERSISTENTES.....</u>	<u>85</u>
<u>CONSIDERACIONES FINALES.....</u>	<u>88</u>
<u>CONCLUSIONES:.....</u>	<u>89</u>
<u>RECOMENDACIONES.....</u>	<u>90</u>
<u>AL CONCLUIR ESTE TRABAJO SE RECOMIENDA LO SIGUIENTE:.....</u>	<u>90</u>
<u>PROFUNDIZAR EN EL ESTUDIO Y CREACIÓN DE SOFTWARE PARA LA ESTIMULACIÓN VISUAL EN NIÑOS ESTRÁBICOS Y/O</u> <u>AMBLOPIES.....</u>	<u>90</u>
<u>REALIZAR LA IMPLEMENTACIÓN DEL SOFTWARE BASADO EN EL ANÁLISIS Y DISEÑO PROPUESTO EN ESTE TRABAJO E</u> <u>IMPLANTARLO PAULATINAMENTE EN TODAS LAS ESCUELAS DE ATENCIÓN A DÉBILES VISUALES SEGÚN LAS CONDICIONES</u> <u>EXISTENTES EN NUESTRO PAÍS.....</u>	<u>90</u>

.....	91
ANEXOS:.....	94
ANEXO 2.....	95
ANEXO 3.....	96
ANEXO 4.....	97
ANEXO 5.....	98
ANEXO 6.....	99
APÉNDICES:.....	100
ÍNDICE DE FIGURAS Y TABLAS.....	102

Dentro del sistema de la Enseñanza Especial en Cuba, existen en la provincia de Ciudad de La Habana tres centros donde se brinda atención al estrabismo y la ambliopía. Dichos centros (dos escuelas y un círculo infantil), cuentan con equipos de terapia y con ejercicios visuales para realizar el tratamiento, tanto en el consultorio oftalmológico que tiene la escuela como durante las actividades docentes.

Algunos de estos equipos presentes en los consultorios oftalmológicos, no poseen los recursos psicológicos necesarios para el logro de una motivación adecuada, además de que existen grandes dificultades para acceder a las nuevas tecnologías e insumos necesarios para la reparación de dichos equipos.

En discursos pronunciados por nuestro Comandante en Jefe Fidel Castro Ruz, ha detallado las diferencias que se han logrado respecto a etapas revolucionarias anteriores y cuanto poseen estas escuelas. Explicó que "...las escuelas para niños ambliopes --que no son los ciegos-- tienen que ser diseñadas especialmente." "Las escuelas para niños ciegos y débiles visuales requieren igualmente instalaciones y equipos muy especiales. Es maravilloso ver después lo que aprenden esos niños y lo que hacen en esas escuelas tan humanas."¹; Sin embargo producto del período especial muchos de esos equipamientos perdieron la posibilidad de que fueran reparados sistemáticamente, algunos caducaron y otros fueron dados de baja.

Desde hace un tiempo, se vienen realizando estudios con el fin de buscar alternativas más apropiadas y que funcionen según los principios de funcionamiento de alguno de estos equipos, unido a la confluencia de la utilización de requerimientos psicológicos necesarios en las etapas evolutivas de estos escolares.

"Las nuevas y favorables condiciones desarrolladas en la actual Revolución Educacional, han permitido la introducción de las Tecnologías de la Informática y las Comunicaciones (TICs)

¹ Castro Ruz, F. Discurso pronunciado por el Presidente de la República de Cuba en la clausura del II Encuentro Mundial de Educación Especial celebrado en el Teatro "Karl Marx", Ciudad de La Habana, 1998.

con vistas de perfeccionar el Proceso Docente Educativo (PDE), que ha llegado a todo el sistema de enseñanza y educación.”²

Bajo estas condiciones es necesario continuar buscando alternativas para explotar los recursos que existen, con vista a desarrollar la creatividad de los maestros y el resto del personal.

Estas nuevas alternativas permiten atenuar las presentes dificultades existentes y cumplir con el objetivo fundamental de los centros médicos pedagógicos para trabajar con niños ambliopes, es decir, restablecer y desarrollar las funciones visuales, evitar el retraso pedagógico y la aparición de trastornos psicológicos a causa de la alteración ocular que constituye el defecto primario.

Los niños débiles visuales poseen características específicas, poca agudeza visual la cual consiste en la capacidad de ver detalles en los objetos, la cual se debe saber que no debe utilizarse para predecir el funcionamiento visual o educacional de un sujeto, otra de las características es la ambliopía, que solo consiste en aquellos que tienen la vista débil o que padecen de debilidad visual, es un déficit funcional de la visión causada por un desarrollo anómalo secundario a una estimulación visual anormal, puede ser de moderada a severa, en dependencia del grado de afectación visual.

Teniendo en cuenta la importancia social que se le concede al desarrollo de las funciones visuales en los niños y valorando los aportes teóricos que han sido planteados en investigaciones anteriores, se considera ensayar el uso de un modelo que pueda ser utilizado con doble objetivo, rehabilitativo y académico, y lógicamente, el niño obtendrá mayor motivación y concentración en las tareas.

A partir de lo expresado surge la siguiente interrogante, constituyendo al **problema científico** de este trabajo: ¿Cómo contribuir a la estimulación visual de niños estrábicos y/o ambliopes sobre la base de los principios de funcionamiento del Localizador Bangester? Siendo el **objeto de investigación** el proceso de rehabilitación de dificultades visuales a partir de la

² Suárez Castro, Y. "Modelo de Software para la Estimulación Visual de los Escolares Ambliopes con Fijación Central". Tesis para optar por el título de Lic. en Educación. ISP EJV, Ciudad de La Habana, 2005.

modelación de un software. Mientras que el **campo de acción** se enmarca en la modelación de un software dirigido a la estimulación de la visión en niños Estrábicos y/o Ambliopes mediante el desarrollo de habilidades óculo-manuales.

El **objetivo** de este trabajo es modelar un software para la estimulación visual de escolares estrábicos y/o ambliopes.

Se plantean a continuación un grupo de **tareas investigativas** que permitirán dar solución a los objetivos planteados:

- Análisis de las concepciones científico teóricas que sustentan la rehabilitación visual mediante el uso de los software educativos con el propósito de fundamentar el sistema propuesto.
- Obtención de los requisitos del software con vistas a perfeccionar el desarrollo del sistema y facilitar el entendimiento entre usuarios y clientes del módulo a diseñar.
- Análisis y discusión de la arquitectura de la aplicación con el objetivo de organizar y aportar elementos que ayuden a la toma de decisiones.
- Estudio de metodologías como base del desarrollo del sistema que permita contribuir al análisis y diseño de la aplicación.

Entre los métodos científicos que posibilitaron cumplimentar los objetivos y tareas investigativas se utilizaron:

- **Método de trabajo con fuentes bibliográficas**, utilizado con el objetivo de estudiar los principios básicos del proceso de ingeniería, específicamente lo relacionado con la ingeniería de software.
- **Método Analítico-Sintético**, se utiliza para el procesamiento de la información recopilada durante la investigación.

- **Método Deductivo-Inductivo**, fue necesario usarlo para que a partir de leyes y principios generales sobre el proceso de determinación de las vías más importantes para la elaboración de modelo previsto.
- **Método de Observación**, utilizado para constatar las características del equipo oftalmológico Localizador Bangester; punto inicial para el conocimiento de sus principios de funcionamiento.

Novedad científica: Estará relacionado con el logro de un modelo, desde la ingeniería de software, con vista a posibilitar la implementación de un método o herramienta capaz de brindar una vía para el desarrollo de habilidades visuales en escolares estrábicos y/o ambliopes, partiendo de los principios de funcionamiento del equipo oftalmológico Localizador Bangester y materializando sus tareas a través del uso de laberintos 3D como tareas rehabilitativas.

Capítulo 1: Fundamentación Teórica

El contenido de este trabajo se estructura en tres capítulos:

El **Capítulo I**. Fundamentación teórica: donde se realiza un estudio del estado del arte. Además, se describen los principales aspectos de las herramientas a utilizar para la implementación del software.

El **Capítulo II**. Características del sistema: donde se expone el objeto de estudio, se realiza un estudio del sistema, se plantea el modelo del dominio, los requisitos y casos de uso del sistema; así como la expansión de los casos de uso.

El **Capítulo III**. Análisis y diseño del sistema: aquí se muestran los diagramas de clases de análisis. En la parte del diseño se exponen los detalles relacionados con el diseño del sistema.

Cada capítulo es iniciado por una breve introducción donde se dan a conocer los temas que se desarrollarán durante el mismo. Finaliza con las consideraciones finales, en las que se plantean los resultados parciales obtenidos. Finalmente se proporciona un apéndice compuesto por un glosario de abreviaturas y un glosario de términos, para apoyar la comprensión del lenguaje técnico utilizado en la investigación y el desarrollo del trabajo realizado.

Capítulo 1: Fundamentación Teórica

El presente capítulo tiene como objetivo exponer los fundamentos teóricos generales que sirven de punto de partida para comprender las bases biológicas de las dificultades visuales Estrabismo y Ambliopía. En dicho capítulo se tratará, de manera breve, algunas consideraciones acerca del proceso correctivo compensatorio, las funciones tradicionales de los ordenadores (las computadoras) y otras que se proponen ser analizadas por el tutor, a partir de investigaciones que se acometen desde otras áreas del saber humano y no son objetivos del presente trabajo.

Además, se describen las metodologías, lenguajes y herramientas existentes para la creación de modelos similares partiendo siempre de los principios de funcionabilidad del equipo oftalmológico referenciado, incluyendo la importancia de la implementación de laberintos como vía fundamental para el desarrollo óculo manual en escolares estrábicos y/o ambliopes; así como la importancia del uso de laberintos en video-juegos para el tratamiento oftalmológico.

1.1 Algunos fundamentos Anátomo - Fisiológicos y Ópticos de la Visión

Desde los primeros días de vida del niño, la visión se relaciona íntimamente con el tacto, oído, olfato y otros tipos de perceptibilidad. Como resultado, se forman sistemas de enlace dinámico - complejos, constituyendo la base sensitiva de toda la actividad psíquica y de la orientación del niño en el espacio.

El órgano visual del niño se desarrolla de forma gradual hasta los 8 años en que aproximadamente termina. Es por eso que el niño aprende a ver, como aprende a caminar y a hablar.

1.2 Consideraciones acerca de la corrección y/o compensación de la ambliopía

El científico conocido como Vigostki se opuso a los intentos de biologizar las concepciones que sobre el desarrollo de los niños con necesidades educativas especiales existían y que en

algunos textos se absolutizaban y llegó a plantear que la deficiencia no era tanto biológica como social.

Enfatizó en que el insuficiente desarrollo que se observa en las personas con algún tipo de discapacidad, se debe esencialmente a la ausencia de una adecuada educación basada en métodos y procedimientos metodológicos especiales que permitan un desarrollo semejante al de los niños de la norma.

También señaló: "... el tacto en el sistema de conducta del ciego, y la visión en el sordo, no desempeñan el mismo rol que en las personas que ven y oyen normalmente; los deberes del tacto y de la visión respecto al organismo y sus funciones son otras..."³; pero no ocurre así en los niños que padecen estrabismo y/o ambliopía, los cuales presentan una disminución de la agudeza visual producto de causas funcionales corregibles.

Estos trastornos, a pesar de corregirse y/o compensarse dentro del propio desarrollo socio-cultural del sujeto, requieren alternativas rehabilitativas o habilitativas que conducen al aspecto biológico, de acuerdo con el desarrollo del propio órgano; dejando por sentado el carácter biológico, psicológico y social del individuo.

La compensación y/o la corrección de la deficiencia, objeto de análisis, se producen por vía indirecta (en las ambliopías) o directa (en los estrabismos), según el caso, ya sea de carácter social, psicológico o clínico; y no se trata de que en el niño ambliope solo se formen mecanismos psicofisiológicos que permitan, a través de esa vía, suprimir la mala representación de un objeto en corteza; sino que dentro del propio proceso de formación, los especialistas utilicen alternativas cada vez más eficientes para la reincorporación saludable del niño a su medio regular, en caso que se desarrolle socio - culturalmente fuera de este. Claro está, todo depende de la propia naturaleza del diagnóstico oftalmológico y del trabajo conjunto familia - escuela especializada.

El análisis del carácter biológico y social del defecto, lleva a Vigostki a plantear que la compensación en su accionar socio - psicológico puede tener también dos resultados: la

³ Vigostki, L.S. Obras Escogidas en V Tomos. Tomo V/ L.S. Vigostki. Editorial Pueblo y Educación. Ciudad de La Habana, p234.

victoria, que logramos en los niños objeto de análisis, no solo psicológicamente mediante la separación necesaria para la instrucción y educación especializada, sino que en la acción clínica juega un papel excepcional en la orientación de cómo padres y maestros pueden apoyar el tratamiento.

La percepción de la realidad es construida más o menos eficientemente por el sujeto (intra-psicológicamente), está influenciada por las características de su órgano visual (utilización o no de mediadores instrumentales y sociales), y por el desarrollo de las habilidades potenciales. De ahí que exaltemos los principios que Vigostki expone y demuestra en su obra.

Parte de numerosas indagaciones de especialistas en el tema han llevado la idea que los modelos de juegos dirigidos al proceso de tratamiento de esta entidad diagnóstica se realice mediante el uso de laberintos donde el sujeto posibilite todo un movimiento espacial a la vez que se ejecuten actividades de recolección de objetos lumínicos.

En dichos laberintos, análogamente con el equipo oftalmológico, se han de realizar actividades o tareas rehabilitativas que contribuirán al desarrollo de dichas habilidades visuales tributivas al proceso de aprendizaje y de una calidad de vida superior, garantizando un tránsito desde el centro escolar especializado hacia su entorno de la enseñanza general primaria.

De esta manera se sistematizan los valores científicos teóricos que desde investigaciones realizadas en 1999 por MSc. Pedro Carlos Pérez Martinto y otros colaboradores, en el Centro de Referencia Latinoamericana para la Educación Especial - CELAEE, y posibilitar la experimentación de nuevas alternativas de tratamiento con el uso de las tecnologías de la informática y las comunicaciones.

1.3 Tecnología

La educación ha sufrido continuos cambios, descubriendo nuevos métodos para la enseñanza y el aprendizaje de sus educandos como son el uso de las TIC.

Pensar en el uso óptimo y utilidad didáctica de los medios audiovisuales, perfeccionar las relaciones pedagógicas en el Proceso Educativo Docente (PDE) resulta cada vez más necesario, debido a que las transformaciones que se han venido presentando en el ámbito de la educación han orientado el crecimiento y la diversidad de opciones educativas para la formación y actualización de los alumnos y maestros, la capacitación para el trabajo, la recreación y la utilización del tiempo libre.

El uso de los medios audiovisuales rebasa el espacio del aula y trascienden el entorno de la escuela; es por ellos que existen variados criterios acerca de las funciones que se le atribuyen a los medios audiovisuales en el contexto del proceso enseñanza-aprendizaje, entre ellas están:

- **Función informativa:** Permite el estudio de la realidad a que se hace referencia y la describe lo más objetivamente posible.
- **Función motivadora:** Pretenden suscitar emociones y afectos, estimular el estudio y la búsqueda de nuevos conocimientos. Influye en la voluntad de los destinatarios, mediante la emotividad.
- **Función lúdica:** Pretende el goce del aprendizaje mediante el juego, fundamentalmente en los primeros grados.
- **Función investigativa:** Contribuye a incentivar la necesidad de encontrar nuevas alternativas para el conocimiento científico.
- **Función controladora:** Posibilita utilizarlos para el control del aprendizaje logrado por los estudiantes durante todo el proceso educativo.
- **Función estimulativa y rehabilitativa de funciones:** Pretende posibilitar que los escolares con determinadas disfunciones encuentren una alternativa más para su desarrollo o estimulación, a la vez que genera, a partir de su modelación, las vías idóneas para el desarrollo de su discapacidad.⁴

Las informaciones que llegan hoy a la escuela por medio de la televisión, video y la computación promueven el aprendizaje diferente en los niños y jóvenes. Es responsabilidad

⁴Pérez Martinto, P.C. "La rehabilitación visual de escolares ambliopes con fijación central", documento de trabajo. UCI. Ciudad de la Habana. 2005.

del maestro aprovechar de forma positiva y eficaz las potencialidades de estas tecnologías en la escuela ya que contribuyen nuevos estímulos para el desarrollo del proceso de enseñanza-aprendizaje. Un profesor creativo, práctico, proactivo considerará los medios audiovisuales e informáticos como sus aliados, útiles para el desarrollo del PDE y de herramienta de inestimable valor para hacer realidad la Revolución en la Educación que estamos comenzando.

Los software educativos pueden contribuir con el trabajo correctivo-compensatorio, es por ello que los beneficios de su utilización en los niños con necesidades educativas especiales se multiplican y constituyen un recurso de grandes posibilidades educativas, enriquecen su aprendizaje, acentúan sus fortalezas y eliminan el sentido del fracaso en los escolares.

Una de las principales causas que se plantean en los "Estudios de Permanencia y Rehabilitación de escolares estrábicos y ambliopes", que inciden en que los escolares permanecieran un promedio de 3 a 4 años en la escuela, es el estado actual de los equipos oftalmológicos no sustituidos, por esta razón no vemos esta tecnología solo como un medio para desarrollar los conocimientos, sino también como un apoyo al proceso de rehabilitación que se realiza en el consultorio oftalmológico.

Para ello los software educativos deben cumplir con determinados aspectos: localización de objetos en el espacio, sus relaciones y formas de los objetos, uso del colorido, la multimedia, posibilidades para desarrollar la creatividad e independencia, posibilidades para vincular el contenido del software con la realidad y con el currículo escolar.

De ahí, que en esta propuesta, se plantea una alternativa para que el maestro aproveche la posibilidad del desarrollo de la informática que existe hoy en la actualidad y la emplee como una vía de estimular el desarrollo de la agudeza visual de niños ambliopes con fijación central.

Lo importante es posibilitar que ese niño compense sus necesidades y desarrolle sus potencialidades. Es por ello que el software se puede incorporar al listado de alternativas destinadas a la rehabilitación visual de estos niños.

1.4 Análisis de investigaciones realizadas en centros de Rehabilitación Visual

La foto-estimulación es uno de los ámbitos en los que se ha hecho uso de los ordenadores en el contexto de la estimulación visual y plantean: “Los monitores de ordenadores son una fuente de luz adaptable a las condiciones circunstanciales y a disímiles tipos de usuarios. Además, el hecho de que las imágenes en pantalla se creen activando secuencialmente sus diferentes partes, puede utilizarse como fuente de estimulación rítmica para los ojos. Esta característica, y sus consecuencias fisiológicas, han convertido a los monitores en el principal instrumento de algunos de los tratamientos más recientes en el ámbito de la ambliopía”.

La computación incluye juegos vinculados al estudio que motivan al niño y facilitan el restablecimiento de las funciones visuales para una mejor rehabilitación y aprendizaje, lo que quedó demostrado, de ahí la importancia de formar en ellos el interés por el estudio, de modo que amplíen sus conocimientos y habilidades. La utilidad de la computación como método alternativo de tratamiento en el sistema corrector pedagógico, para la rehabilitación de estos escolares por el resultado visual y el rendimiento académico alcanzado, fue irrefutable.

A esto se le suma la investigación realizada el curso escolar 2004 – 2005 en la escuela 28 de enero del municipio de Guanabacoa, por las autoras Fernández Zapata, Y. y Reyes García, Y., graduadas de la carrera Educación Especial, las cuales obtuvieron resultados alentadores en la estimulación visual de estos niños a partir del uso de software dirigidos a el aprendizaje de asignaturas básicas y la estimulación, respectivamente. En estos trabajos, las autoras muestran los significativos resultados en el desarrollo de la agudeza visual de ambliopes con fijación central.

Lo que me permite valorar la efectividad del uso de un software específico con el fin de aumentar las posibilidades de tratamiento en la escuela, específicamente, los consultorios oftalmológicos.

No se ha encontrado información acerca de algún software como este, pero sin embargo existen otros que permiten implementar un programa de Estimulación Visual, en especial con los estudiantes de corta edad, porque les enseña que su ambiente es un lugar interesante y les fomenta a desarrollar intereses más allá de sus propios cuerpos. La Estimulación Visual,

ayuda al cerebro del niño a ver mejor y discriminar formas, colores y detalles interiores de los objetos. Ofrecer actividades para que el estudiante tenga la oportunidad de tener experiencias visuales exitosas que proveerán a la vez motivación adicional. Este tipo de actividades en las que el niño interactúa con la computadora, son ideales, ya que fomentan su independencia y una oportunidad para que los estudiantes aprendan a través de la exploración, ensayo y error.

1.5 Métodos e instrumentos empleados.

Posibles herramientas para dar solución a la propuesta de software:

1.5.1 Modelación teórica: desarrollar la estructura de análisis y diseño del software, con el fin de reproducir los principios de funcionamiento y las relaciones de los componentes para así avanzar en la futura implementación de una nueva versión. Este método se utilizará en el proceso de programación del nuevo software en el lenguaje C++.

1.5.2 Plataforma de Desarrollo

Plataforma .NET

La plataforma .NET es una capa de software que se coloca entre el Sistema Operativo (SO) y el programador y que abstrae los detalles internos del SO. Las características fundamentales de esta plataforma son las siguientes:

- ✓ **Portabilidad:** Debido a la abstracción del programador respecto al SO, una aplicación .NET puede ser ejecutada en cualquier SO de cualquier máquina que disponga de una versión de la plataforma. En estos momentos la plataforma .NET solamente está disponible para la familia Windows aunque se está desarrollando una versión para Linux.
- ✓ **Multilinguaje:** Cualquier lenguaje de programación puede adaptarse a la plataforma .NET y ejecutarse en ella debido a que siempre se compila en código intermedio

- ✓ **Interoperabilidad:** La interoperabilidad entre diferentes trozos de código escritos en diferentes lenguajes es total. El código puede ser escrito en cualquier lenguaje compatible con .Net ya que siempre se compila en código intermedio

La plataforma .NET es la encargada de proveer lo que se llama código administrado, es decir, un entorno que provee servicios automáticos al código que se ejecuta. Los servicios son variados:

Plataforma Mono

Mono es una plataforma eficaz y sencilla para el desarrollo de aplicaciones, para portar soluciones ya existentes para Windows al mundo Linux y en general como una excelente herramienta para el desarrollo de programas multiplataforma y servicios Web. Es sin duda una plataforma que será crucial para que las empresas puedan migrar sus aplicaciones al mundo Linux y para que nuevas aplicaciones puedan ser desarrolladas con el menor costo posible.

¿Por qué Plataforma .Net?

Se decidió utilizar plataforma .NET debido a que con la utilización del cargador de clases se pueden cargar las mismas en memoria. Contiene además un compilador que transforma código intermedio de alto nivel, independiente del hardware que lo ejecuta a código de máquina, propio del dispositivo que lo ejecuta. Cuenta a su vez con un recolector de basura que elimina de memoria objetos no utilizados.

1.5.2.1 Características de Visual Studio.



Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas Windows. Soporta varios lenguajes de programación tales como Visual C++,

Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones Web, así como servicios Web en cualquier entorno que soporte la plataforma .NET. Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas Web y dispositivos móviles.

1.5.3 Características referentes a lenguajes de programación

1.5.3.1 Lenguaje de programación C++



El lenguaje de programación C++ es un lenguaje de programación que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos. Las principales características del C++ son las facilidades que proporciona para la programación orientada a objetos y para el uso de plantillas o programación genérica (templates).

C++ está considerado por muchos como el lenguaje más potente, debido a que permite trabajar tanto a alto como a bajo nivel, sin embargo es a su vez uno de los que menos automatismos presenta, lo cual "dificulta" mucho su aprendizaje.

1.5.3.2 Lenguaje de programación C#



El lenguaje de programación *C#*, es actualmente uno de los lenguajes de programación más populares en informática y las comunicaciones. Permite a los programadores abordar el desarrollo de aplicaciones complejas con facilidad y rapidez. Es como si se tomaran todas las cosas buenas de Visual Basic y fueran añadidas a C++, aunque recortando algunas de las tradiciones más ocultas y difíciles de conocer de C y C++. Con *C#* no sólo se pueden escribir programas para la Web, sino que también permite desarrollar aplicaciones de propósito general. Se basa en las lecciones aprendidas de los lenguajes C, C++, Java y Visual Basic. Por ello se trata de un lenguaje que combina todas las cualidades que se pueden esperar de un lenguaje moderno (orientación a objetos, gestión automática de memoria, etc.) a la vez que proporciona un gran rendimiento.

¿Por qué C++ como lenguaje de programación?

Fue seleccionado el lenguaje de programación C++ para la implementación de debido a que es el lenguaje en el que se tiene mayor experiencia por parte del autor, además de ser un lenguaje muy robusto que permite elaborar desde aplicaciones sencillas hasta sistemas operativos y mucho más, todo eso dependiendo del manejo del lenguaje. Existen muchos algoritmos cuyo pseudocódigo se encuentra ya desarrollado en C++, de manera se puede tomar y amoldarlo según el interés.

1.5.4 SceneToolKit (STK):



SceneToolKit (STK) es la primera y una de las herramientas brindadas por el Proyecto de Herramientas de Desarrollo para Sistemas de Realidad Virtual, tiene como objetivo básico agrupar las funcionalidades comunes a cualquier sistema de realidad virtual, de manera que se les facilite el trabajo a los programadores de juegos y simuladores a través de la reutilización de código. Permite no solamente la visualización de los entornos sintéticos sino además la aplicación de leyes físicas y matemáticas, animaciones, entre otras, usando las librerías gráficas OpenGL y DirectX, sobre plataforma Windows y Linux. A la vez, se retroalimenta con las necesidades de los programadores que las utilicen, así como de sus investigaciones.⁵

La Herramienta Básica para Sistemas de Realidad Virtual SceneToolKit permite, entre otras funciones, la carga y visualización de entornos creados en el 3DSMAX, para lo cual han sido

⁵ 2007. SceneToolKit. Sistemas de Realidad Virtual. *Ayuda y Documentación*. UCI, Ciudad de La Habana : s.n., 2007.

creados algunos *plugins* o macros para exportar ficheros legibles por la herramienta. Dicha exportación se hace desde el mismo 3DSMAX.

Para el desarrollo del sistema propuesto se usa el plugin *px3DXExport.mcr* el cual no es más que un macro para la exportación de modelos en formato 3DX.

1.6 Soluciones para la realización de la Base de Datos

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

1.6.1 SQL Server



SQL Server es un Sistema de Gestión de Bases de Datos Relacionales (SGBDR), desarrollado por Microsoft, que permite, como su propio nombre indica, la gestión de un entorno de bases de datos relacional. Sigue una arquitectura cliente/servidor. Es un potente motor de bases de datos de alto rendimiento capaz de soportar millones de registros por tabla y con herramientas de desarrollo integradas como Visual Studio 6.0 o .NET.

Algunas de las ventajas que ofrece utilizar un Sistema de Bases de Datos son las siguientes:

1. Independencia entre datos y tratamientos. El cambio en los programas no influye en la disponibilidad de los datos, así como la modificación de éstos no afecta a la reprogramación de las aplicaciones que los usan.
2. Coherencia de resultados: Debido a que los datos son almacenados una sola vez, se evitan los problemas que puede ocasionar la redundancia.

3. Mejor disponibilidad de datos para los usuarios: Los datos son compartidos por un conjunto de usuarios, que accede a ellos de forma concurrente, siempre que estén autorizados a ello.
4. Mayor valor informativo: El conjunto de los datos almacenados en la base de datos ofrece un mayor valor informativo, que la suma de ellos independientemente.
5. Mayor eficiencia en la recogida, validación e introducción de los datos en el sistema: Al no existir redundancia, los datos se recogen y validan una sola vez, aumentando así la eficiencia.
6. Reducción del espacio de almacenamiento: La desaparición (o disminución) de redundancias, unido a las técnicas de compactación, implica una menor ocupación de almacenamiento secundario.

Dada la investigación realizada finalmente se determinó que no se utilizará como gestor de base de datos para el desarrollo del software propuesto, teniendo en cuenta que se tendrá una pequeña base de datos y no es necesaria la dificultosa instalación.

1.6.2 Lenguaje XML



XML: Extensible Markup Language.

XML («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del Lenguaje de Marcado Generalizado (SGML) y permite definir la gramática de lenguajes específicos (de la misma manera que Lenguaje de Marcas de Hipertexto (HTML) es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular,

sino una manera de definir lenguajes para diferentes necesidades. Se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo, entre otros. XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

¿Por qué XML?

Se decidió utilizar la tecnología XML para el desarrollo de este trabajo debido a que presenta ventajas en cuanto a la comunicación de datos, si la información se transfiere en XML cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella. Otro punto a tener en cuenta es la migración de datos pues sería muy sencillo mover datos de una base de datos a otra.

1.7 Herramientas CASE (Computer Aided Software Engineering)

Las herramientas CASE son un conjunto de programas y ayudas que brindan ayuda y 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.

1.7.1 Soluciones para la realización del diseño.

La ingeniería de software tiene varios modelos de desarrollo en los cuales se puede apoyar para la realización de software, Entre los cuales de destacan:

Modelo en cascada o Clásico (modelo tradicional).

Éste toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución y las representa como fases separadas del proceso.

El modelo en cascada consta de las siguientes fases:

1. Definición de los requisitos.
2. Diseño de software
3. Implementación y pruebas unitarias
4. Integración y pruebas del sistema.
5. Operación y mantenimiento.

Una fase no comienza hasta que termine la fase anterior y generalmente se incluye la corrección de los problemas encontrados en fases previas.

Modelo en espiral (modelo evolutivo)

La idea detrás de este modelo es el desarrollo de una implantación del sistema inicial, exponerla a los comentarios del usuario, refinarla en N versiones hasta que se desarrolle el sistema adecuado. Una ventaja de este modelo es que se obtiene una rápida realimentación del usuario, ya que las actividades de especificación, desarrollo y pruebas se ejecutan en cada iteración.

Desarrollo iterativo e incremental.

Una de las ventajas de este modelo es precisamente que los clientes no esperan hasta el fin del desarrollo para utilizar el sistema, pueden empezar a usarlo desde el primer incremento. Además los clientes pueden aclarar los requisitos que no tengan claros conforme ven las entregas del sistema. También, se disminuye el riesgo de fracaso de todo el proyecto, ya que se puede distribuir en cada incremento.

Existen diferentes metodologías de desarrollo y las más usadas son:**Rational Unified Process.**

El Proceso Unificado Rational (Rational Unified Process) es la metodología para el desarrollo y construcción de software basado íntegramente en UML como soporte a la metodología.

Extreme Programming (XP).

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo.

1.7.1.1 El Proceso Unificado de Desarrollo de Software (RUP) como base en el desarrollo de la solución.

El ciclo de vida de RUP se caracteriza por:

- **Dirigido por casos de uso:** Los casos 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. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso, es decir, cómo se llevan a cabo.

- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.

- **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo y los incrementos, al crecimiento del producto.

A continuación se describen las cuatro fases por las que está compuesta la metodología RUP:

Inicio: Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.

Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.

Construcción: Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene 1 o varios release del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios.

Transición: El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores cada una de estas fases es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

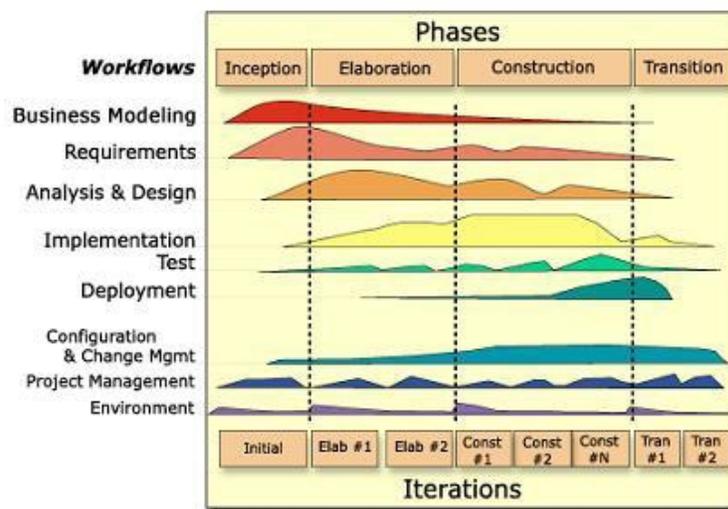


Figura 1. Fases e Iteraciones de la Metodología RUP.

1.7.1.2 Extreme Programming (XP):

Extreme Programming (XP): La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

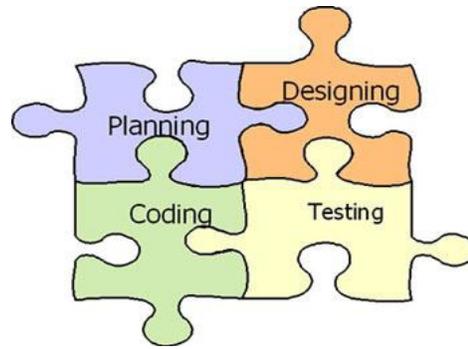


Figura 2. Metodología Extreme Programing

Características de XP, la metodología se basa en:

- **Pruebas Unitarias:** Se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se pueda hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantase a obtener los posibles errores.
- **Refabricación:** Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Por qué RUP?

Se decidió utilizar RUP debido a que este proceso está dividido en 4 fases, las cuales se dividen en iteraciones, es basado en roles, posee mucha documentación, los artefactos son el objetivo principal de cada actividad, muy organizativo. Así puede verse de una forma más detallada el papel que juega cada objeto en el sistema. Por todo lo antes dicho se puede decir que este proceso de desarrollo es el idóneo para llevar a cabo la descripción de este software.

1.7.2 Lenguaje de Modelado



UML

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

¿Por qué UML?

Se decidió utilizar UML para contar con un lenguaje estándar y poder escribir los planos del software. UML es una serie de reglas y recomendaciones para representar modelos. Este lenguaje de modelado permite documentar y especificar los elementos creados mediante un lenguaje común describiendo modelos. Además de que RUP propone como lenguaje de modelado a UML.

1.8 Herramienta de modelación visual

1.8.1 Rational Rose Enterprise Edition



Rational Rose es una herramienta que se utiliza para el desarrollo de un software. Rational Rose Enterprise es el producto más completo de la familia Rational Rose. Todos los productos Rational Rose incluyen soporte UML. Es la mejor elección para el ambiente de modelado. Proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.

1.8.2 Visual Paradigm

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software, la cual dentro de sus características ofrece un entorno de creación de diagramas para UML 2.0. Además de la disponibilidad de múltiples versiones, para cada necesidad y disponibilidad en múltiples plataformas. Ofrece el diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad. Presenta capacidades de ingeniería directa (versión profesional) e inversa.



¿Por qué Rational Rose?

Se decidió utilizar Rational Rose debido a que posee todas las características que los desarrolladores, analistas, y arquitectos exigen: soporte UML incomparable, completo soporte al

equipo, desarrollo basado en componentes con soporte para arquitecturas líderes en la industria, facilidad de uso, integración optimizada, entre otras características que conllevan a que Rational sea reconocido como el líder tecnológico por su rol en el desarrollo del UML.

1.9 Solución propuesta

Asumiendo lo antes expuesto sobre las principales herramientas y procedimientos que se utilizarán y considerando las tecnologías que están a total disposición, se decidió recurrir a el lenguaje C++ y Visual Studio .NET para el desarrollo de una aplicación que admita aprovechar al máximo las ventajas de esta plataforma. Además se decidió implementar el desarrollo del sistema con la herramienta STK por las ventajas que brinda y el diseño se ejecutará con 3DSMAX. Se eligieron estas herramientas por su potencial y por estar disponible en nuestra Universidad.

Los procedimientos de informatización de la sociedad, la migración al Software Libre, la implantación en la UCI de un conjunto de técnicas de código abierto, eventualmente, obligarán a las aplicaciones existentes que han sido desarrolladas y construidas en .NET a migrar hacia plataformas como Mono, la versión de código abierto de .NET. La solución trazada puede ser emigrada a esta plataforma de una forma segura, ya que no se han utilizado ninguna particularidad no estándar de la misma.

Consideraciones finales

A modo general en este capítulo que concluye se definieron los principales conceptos que ayudarán a comprender el desarrollo del software. Se fundamentaron las principales técnicas, tecnologías y tendencias más utilizadas en la actualidad mundial en cuanto a estos temas, propiciando un mayor conocimiento para determinar cuáles son mejores opciones para emplear en el diseño.

Se reflejaron además algunas de las principales características de las herramientas a utilizar en el desempeño y realización de este software.

Capítulo 2: Definición de Requerimientos del Sistema

2.1 Introducción:

Este capítulo define la perspectiva de módulo a construir, dando un primer punto de vista, proporciona una noción de manera práctica de lo que se va a fabricar, teniendo en cuenta los requerimientos y necesidades del cliente, partiendo de las técnicas a utilizar dadas a conocer en el capítulo anterior.

Al no poder definir las funcionalidades del negocio se define primeramente el modelo de objetos del dominio, con el objetivo de comprender y describir los conceptos más importantes dentro del contexto del sistema. Un modelo de dominio es en realidad un caso especial de un modelo de negocio más completo, es una variante simplificada del negocio.

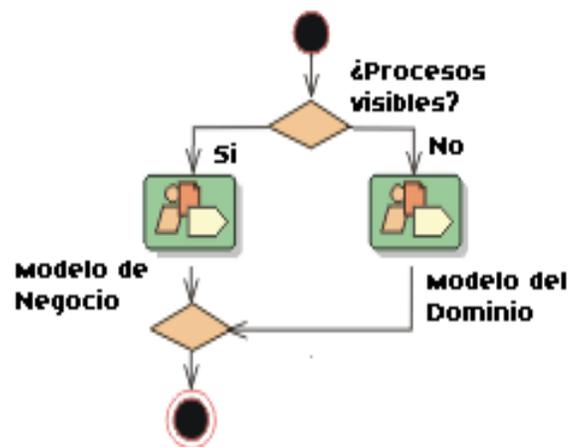


Figura3. Posibles procesos.

Luego se definen los requerimientos del sistema, dando lugar a la creación de actores del sistema, casos de usos y diagrama de casos de uso del sistema y con esto una descripción expandida de cada uno de los casos de uso.

2.2 Modelo del Dominio

Se decidió desarrollar un modelo de dominio, pues no se cuenta con un proceso claro y definido, por esto, se pretende captar los conceptos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema. Esto ayudará a los usuarios, clientes e interesados, a utilizar un vocabulario común para poder entender el contexto en que se basa el sistema.

Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Para identificar todos los conceptos que se emplearán en el diagrama se utilizará un glosario de términos.

2.3 Definición de Conceptos del Modelo del Dominio:

A continuación se presentan un conjunto de conceptos que conforman el glosario de términos del modelo del dominio con el propósito de facilitar un mayor entendimiento de los términos manejados en el diagrama correspondiente.

2.3.1 Glosario de Términos del Dominio:

Laberinto: juego conformado para beneficio del usuario. Cada laberinto posee sus propias características.

Gestor de datos: es el encargado de almacenar la última fecha de acceso del usuario al software y además genera una tabla de récords, la cual va actualizando en la Base de Datos.

Fecha de acceso: es el encargado de actualizar la fecha de acceso de un usuario y además tiene que verificar que no sea igual en un mismo día.

Récords: se encarga de guardar los récords implantados por el usuario en una tabla de récords que se almacenará en la Base de Datos.

Niveles de Dificultad: estructura formada por varios niveles (laberintos diferentes), donde varía el grado de dificultad, a medida que va avanzando el usuario va aumentando el nivel.

Fácil: nivel más simple donde se muestran los laberintos con menos grado de dificultad.

Intermedio: nivel donde se muestran laberintos de mediano grado de dificultad, un poco más complejo que el primer nivel (Fácil).

Difícil: nivel donde se muestran los laberintos con mayor grado de dificultad. Nivel superior.

Motor Gráfico: estructura que muestra la parte visual del producto, contiene objetos, un mapa por cada laberinto. Muestra una interfaz propia en cada nivel y todo en cuanto al entorno del laberinto. Muestra toda la parte gráfica del juego.

Mapa: guía que servirá al usuario para orientarse en el juego, observar en qué posición se encuentra y hacia donde debe dirigirse.

Objetos: elementos que deberá recoger obligatoriamente el usuario a medida que avance el juego, por cada laberinto recogerá una cantidad de objetos determinada.

Interfaz de nivel: estructura gráfica que tendrá cada uno de los laberintos según el nivel en el que se desarrolle y las características propias del laberinto.

Entorno: es el encargado de exponer adecuadamente cada laberinto, con todos los elementos disponibles para el mismo.

2.3.2

Modelo

del

Dominio:

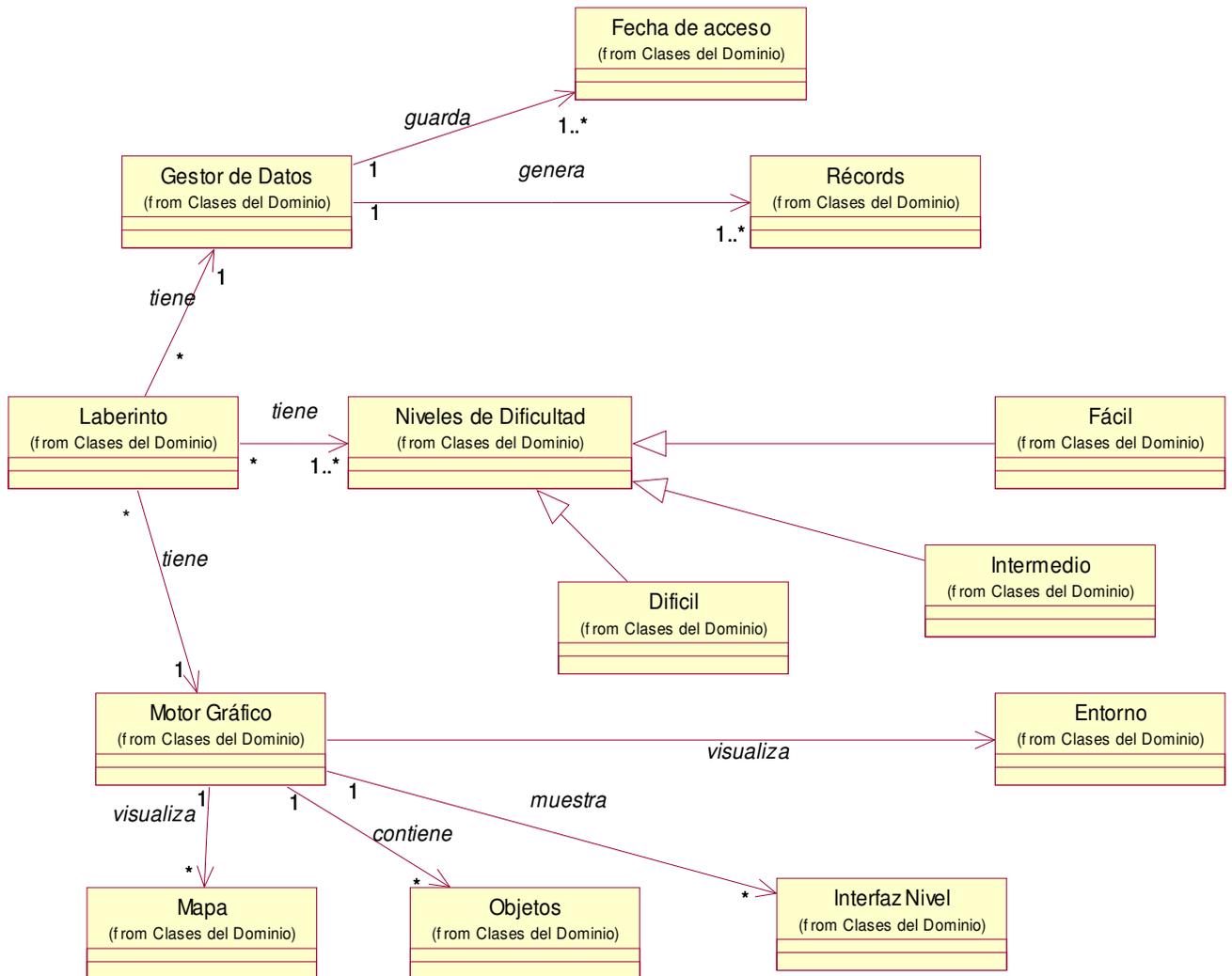


Figura 4. Modelo del dominio

Posteriormente se reconocen los actores del sistema a desarrollar y se conciben, a través de la agrupación de los requerimientos funcionales, los posibles resultados de valor que le pueda brindar a sus actores, o lo que es lo mismo, los casos de uso del sistema. Además, se seleccionan los casos de uso arquitectónicamente significativos para hacerles sus descripciones textuales en formato expandido.

Captura de Requerimientos:

Los requisitos constituyen capacidades o condiciones que el sistema debe cumplir, facilitando el entendimiento entre usuarios y clientes del módulo a diseñar. Los siguientes requisitos establecen las funcionalidades e instrucciones que el módulo debe cumplir en su implementación:

2.4 Requerimientos Funcionales:

RF1 El sistema debe permitir Autenticar Usuario.

RF2 El sistema debe permitir Gestionar Usuario.

RF2.1 El sistema debe permitir insertar datos de usuario.

RF2.3 El sistema debe permitir eliminar usuario.

RF2.4 El sistema debe permitir modificar usuario.

RF3 Mostrar formulario con datos de usuario.

RF4 El sistema debe permitir verificar la existencia del usuario.

RF5 El sistema debe permitir buscar un usuario.

RF6 El sistema debe permitir al maestro Revisar Resultados.

RF6.1 El sistema debe permitir al maestro buscar resultado de estudiante.

RF6.2 El sistema debe mostrar estado del estudiante.

RF7 El sistema debe permitir al maestro enviar estado de estudiante (Avanzado, No avanzado).

RF8 Generar laberintos.

RF9 Mostrar tabla de récords. (Cantidad de objetos recogidos, Nivel).

RF10 El sistema debe permitir guardar automáticamente la última fecha de acceso al juego del estudiante.

RF11 El sistema debe permitir Conformar Laberintos.

RF11.1 Mostrar tiempo de juego.

RF11.2 Mostrar nivel en que se encuentra el estudiante.

RF11.3 Mostrar mapa del laberinto.

RF11.4 Mostrar interfaz de nivel.

RF11.5 Mostrar entorno del laberinto.

RF11.6 El sistema debe permitir ver cantidad de objetos recogidos.

RF11.7 El sistema debe permitir desplazamiento por el laberinto.

RF11.8 Mostrar cantidad de objetos a recoger en cada nivel.

RF12 El sistema debe permitir Obtener Objetos.

RF12.1 El sistema debe permitir generar cierta cantidad de objetos en forma aleatoria.

RF12.2 Determinar la cantidad de objetos recogidos.

RF12.3 El sistema debe permitir recoger objetos.

RF12.4 El sistema debe permitir buscar objetos.

RF12.5 El sistema debe permitir obtener objetos.

RF13 El sistema debe permitir Ejecutar Pregunta de Nivel.

RF13.1 Mostrar pregunta de nivel predeterminada.

RF13.2 El sistema debe permitir responder pregunta de nivel.

RF13.3 Evaluar respuesta pregunta nivel.

RF13.4 Aumentar tiempo de ejecución del juego si la respuesta es correcta.

RF13.5 Cargar el próximo nivel al responder correctamente la pregunta de nivel.

2.5 Requerimientos No Funcionales:

Los siguientes requisitos son propiedades o cualidades que el sistema debe tener, estableciendo de esta forma aspectos que regulan el comportamiento de este.

- o **Usabilidad:** El producto debe estar concebido para que el usuario piense en “qué desea hacer” y no cómo hacerlo, por lo que éste requerimiento debe estar presente en alto grado en el producto final. Cualquier persona con conocimiento mínimo en informática o en el manejo de computadoras podrá utilizar el sistema.
- o **Seguridad:** Autenticación y registro del usuario para permitir o denegar el acceso al sistema. Brindar solo los servicios permitidos a cada usuario.
- o **Rendimiento:** Debe tener alto grado de velocidad de procesamiento o cálculo, tiempo de respuesta y de recuperación y disponibilidad.

- o **Soporte:** En una versión inicial deberá ser compatible con la plataforma Windows, pero debe estar preparado para que con rápidas modificaciones pueda migrar para Linux.
- o **Diseño e implementación:** Se regirá por la filosofía de Programación Orientada a Objetos.
- o **Software:** El software se deberá ejecutar en sistema operativo Windows 98 y superior.

2.6 Modelo de casos de usos del sistema

En esta sección se reconocen el o los posibles actores del sistema a desarrollar y se conciben los casos de uso del sistema, procesos en los cuales se dan resultados de valor para un actor determinado del sistema a elaborar, para esto se debe tener en cuenta los requisitos funcionales, anteriormente expuestos, definiendo así los casos de uso del sistema, que muestran resultados notables a sus actores.

2.6.1 Descripción de los actores:

Los actores son los roles que un usuario o usuarios del sistema llevan a cabo en algún momento del tiempo. También pueden ser otros sistemas con los que el sistema en proceso de modelado tiene interacción. Estos estimulan al sistema con eventos de entrada o la recepción de algún resultado que este produzca.

Actor del sistema

Tabla 1: Actor Del Sistema: Estudiante.

Actores	Justificación
Estudiante	Es el que se beneficiará con la elaboración del producto y sus funcionalidades, tales como jugar, el estudiante pasará de nivel cuando obtenga la cantidad máxima de objetos a recoger en ese nivel. Tiene la posibilidad mediante las preguntas inter-niveles de adquirir conocimientos según el grado de escolaridad y si responden correctamente aumenta el tiempo de desarrollo del juego.

Tabla 2: Actor del Sistema: Maestro.

Actores	Justificación
Maestro	Se encarga de registrar todos los usuarios del sistema. Muestra y analiza los resultados de cada estudiante almacenados en la

Base de datos.

Tabla 3: Actor del Sistema: Usuario.

Actores	Justificación
Usuario	Este actor es una generalización de estudiante y maestro que se define para interactuar con el caso de uso Autenticar Usuario.

2.6.1.1 Casos de uso del sistema

Los casos de uso describen el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, se establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades que debe cumplir el módulo a implementar. Los casos de uso pueden incluir funcionalidades de otros casos de uso manteniendo sus comportamientos.

Tabla 4: Caso de Uso del Sistema: Autenticar Usuario.

CU - 1	Autenticar Usuario
Actor	Usuario
Descripción	Permite autenticar los diferentes tipos de usuarios.
Referencia	RF1

Tabla 5: Caso de Uso del Sistema: Gestionar Usuario.

CU - 2	Gestionar Usuario
Actor	Maestro
Descripción	Permite eliminar, modificar e insertar un usuario.
Referencia	RF2, RF2.1, RF2.2, RF2.3

Tabla 6: Caso de Uso del Sistema: Revisar Resultados.

CU - 3	Revisar Resultados
Actor	Maestro
Descripción	Permite revisar los resultados obtenidos de cada estudiante.
Referencia	RF6, RF6.1, RF6.2

Tabla 7: Caso de Uso del Sistema.

CU - 4	Conformar Laberinto
Actor	Estudiante
Descripción	Permite observar todas las funcionalidades del laberinto.
Referencia	RF11, RF11.1, RF11.2, RF11.3, RF11.4, RF11.5, RF11.6, RF11.7, RF11.8

Tabla 8: Caso de Uso del Sistema: Obtener Objetos.

CU - 5	Obtener Objetos
Actor	Estudiante
Descripción	Permite al estudiante interactuar con diferentes tipos de objetos.
Referencia	RF12, RF12.1, RF12.2, RF12.3, RF12.4, RF12.5

Tabla 9: Caso de Uso del Sistema: Ejecutar Pregunta de Nivel.

CU - 6	Ejecutar Pregunta de Nivel
Actor	Estudiante
Descripción	Permite mostrar, responder y evaluar una pregunta concluido cada nivel.
Referencia	RF13, RF13.1, RF13.2, RF13.3, RF13.4, RF13.5

2.6.2 Casos de uso del sistema:

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema, en este caso, del módulo de seguimiento de terrenos. Se describen aquellos aspectos del sistema que son relevantes al propósito del modelo y a un apropiado nivel de detalle, dando una visión general de las relaciones de las funcionalidades o procesos fundamentales.

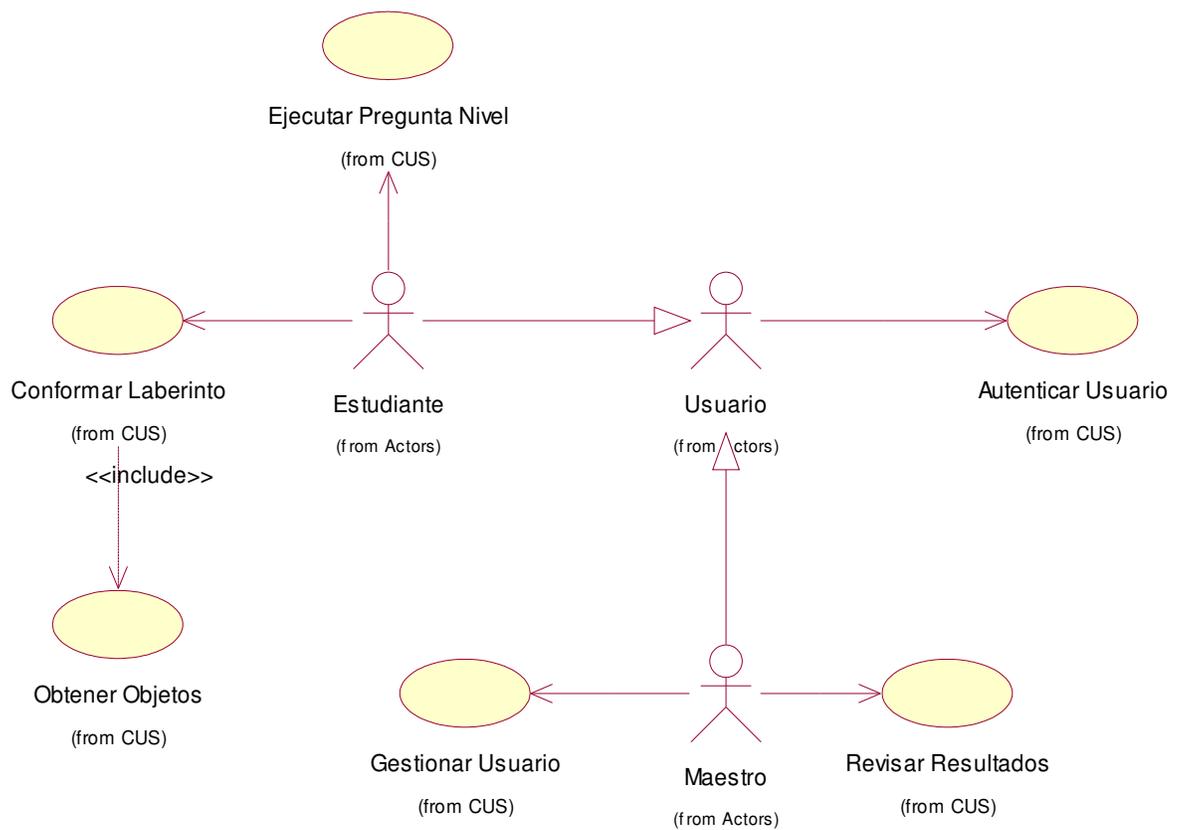


Figura 5. Diagrama de Casos de Uso del Sistema.

2.6.3 Breve descripción del Diagrama de Casos de Uso del Sistema.

El Usuario es el encargado de inicializar el CU Autenticar Usuario, de él heredan el maestro y el estudiante ese CU, debido a que ambos van a autenticarse para acceder a la aplicación.

El Estudiante al ejecutar la aplicación, es quien va a inicializar los CU Conformar Laberinto, en el cual se muestra todo el entorno del laberinto, es donde se va a ejecutar todas las funcionalidades del juego, además de visualizarse todos los elementos definidos en cada laberinto.

Tiene además como un caso incluido el CU Obtener Objetos, que define todo lo referente a los objetos, que se van a recoger durante el juego.

Inicializa también el CU Ejecutar Pregunta Nivel, al terminar cada nivel deberá responder la pregunta de conocimiento la cual le servirá de beneficio propio y aumentará el tiempo de ejecución del juego.

El Maestro inicializa el CU Gestionar Usuario que es donde él introduce los datos de identificación de cada usuario, tanto estudiante como maestro lo registra en el sistema y además inicializa el CU Revisar Resultados que es donde por cada estudiante se darán a conocer los resultados obtenidos.

2.6.4 Especificación de los casos de uso:

Cada caso de uso tiene una descripción que describe la funcionalidad que se construirá en el sistema propuesto, las tablas presentadas a continuación forman parte de este formato expandido, donde se argumentan con mayor profundidad los flujos operacionales de cada caso de uso.

Tabla 10: Descripción Caso de Uso del Sistema: Autenticar Usuario.

Nombre del caso de uso	Autenticar Usuario	
Actores	Usuario	
Propósito	Permitir al usuario acceder al sistema.	
Resumen:	Se inicia cuando un usuario ya sea maestro o estudiante desea interactuar con el sistema, según los privilegios de cada usuario.	
Referencias	RF1	
Precondiciones	El Usuario debe estar registrado en el sistema.	
Curso normal de los eventos:		
Sección General		
Acción del actor	Respuesta del sistema	
1-El Usuario introduce usuario y contraseña.	1.1-El sistema verifica si los datos son válidos: verifica si ya está registrado. 1.2-El sistema muestra la interfaz según el tipo de usuario registrado.	
Cursos Alternos		
	1.1-Si no está registrado el sistema muestra un mensaje "Acceso denegado".	
Pos-condiciones	El Usuario tendrá acceso al sistema.	
Prioridad	Crítico	

Tabla11: Descripción Caso de Uso del Sistema: Gestionar Usuario.

Nombre del caso de uso	Gestionar Usuario	
Actores	Maestro	
Propósito	Controlar y acceder a los datos de cada usuario.	
Resumen: El caso de uso se inicia cuando el maestro necesita registrar un usuario para darle determinados permisos, además de poder eliminar y modificar un usuario existente.		
Referencias	RF2, RF2.1, RF2.2, RF2.3	
Precondiciones	El Maestro debe estar registrado en el sistema.	
Curso normal de los eventos:		
Sección General		
Acción del actor		Respuesta del sistema
1- El Maestro selecciona la opción "Gestionar Usuario"		1.1- El sistema muestra una ventana con las siguientes opciones: "Registrar usuario", "Modificar usuario" y "Eliminar Usuario"
2-El Maestro selecciona una de las tres opciones que muestra el sistema.		2.1-Si selecciona la opción entrar ver la sección Registrar usuario ver sección "Registrar usuario" 2.2-Si selecciona la opción Modificar usuario ver sección "Modificar usuario" 2.3-Si selecciona la opción Eliminar usuario ver sección "Eliminar usuario"
Sección "Insertar usuario"		
1-El Maestro selecciona el tipo de usuario a registrar, ya sea Estudiante o Maestro		1.1-El sistema muestra el formulario según el tipo de usuario seleccionado.
2-El Maestro introduce los datos de Identificación del usuario.		2.1--El sistema verifica si los datos son válidos: verifica si no está registrado. 2.2- El sistema archiva la información en la Base de datos y muestra un mensaje: "El usuario ha sido registrado"

	correctamente”.
Sección “Modificar usuario”	
1-El Maestro selecciona la opción “Modificar usuario”	1.1-El sistema verifica si ya se ha registrado algún usuario en la base de datos. 1.2-El sistema muestra un listado de usuarios registrados.
2-El Maestro selecciona un usuario	2.1-El sistema muestra el formulario donde aparecen los datos registrados del usuario
3-El Maestro modifica los datos del usuario seleccionado.	3.1-Verifica que no se hayan dejado espacios en blanco. 3.2-Si están correctos el sistema muestra un mensaje “Sus datos han sido guardados correctamente”
Sección “Eliminar Usuario”	
1-El Maestro selecciona la opción “Eliminar usuario”	1.1-El sistema verifica que exista algún usuario registrado en la base de datos. 1.2-El sistema muestra listado de usuarios registrados en el sistema
2-El Maestro selecciona un usuario.	1.1-El sistema muestra un mensaje de confirmación:”Está seguro que desea el eliminar ese usuario?”
2.1 El Maestro selecciona una de las dos opciones: “Si” o “No”	2.1-Si selecciona la opción “Si” el sistema elimina al usuario seleccionado.
Cursos Alternos	
Sección “Registrar usuario”	
	2.1-Si está registrado el sistema muestra un mensaje “El usuario ya existe”.
Sección “Modificar usuario”	
	1.1-Si no hay ningún usuario registrado mostrar mensaje: “No hay ningún usuario registrado.”

	3.1-Si no están correctos, el sistema muestra un mensaje: "Debe completar el formulario"
Sección "Eliminar usuario"	
	1.1-Si no hay ningún usuario registrado mostrar mensaje: "No hay ningún usuario registrado." 2.1-Si selecciona "No" el sistema conserva al usuario seleccionado.
Pos-condiciones	Los datos del usuario quedarán registrados si se decide insertar un usuario, excluidos al utilizar la opción "Eliminar Usuario" o actualizados en caso de "Modificar Usuario".
Prioridad	Crítico

Tabla12: Descripción Caso de Uso del Sistema: Revisar Resultados.

Nombre del caso de uso	Revisar Resultados
Actores	Maestro
Propósito	Mostrar y analizar los resultados de cada estudiante.
Resumen:	Se inicia cuando el maestro necesita valorar a un determinado estudiante para llegar a una conclusión del estado del estudiante, si ha avanzado o no.
Referencias	RF6, RF6.1, RF6.2
Precondiciones	El Maestro debe estar registrado en el sistema.
Acción del actor	Respuesta del sistema
1-El Maestro selecciona la opción "Revisar Resultados".	1.1-El sistema verifica que exista algún usuario registrado en la base de datos. 1.2-El sistema muestra un listado de todos los estudiantes inscritos y en cada uno de los nombres la opción de Ver Resultados.
2-El Maestro selecciona la opción "Ver Resultados" de un determinado estudiante.	2.1-El sistema muestra el resultado de ese estudiante (Resultado: cantidad de objetos recogidos y el nivel en el que se acabó el tiempo). 2.2-El sistema muestra una lista desplegable con los estados del estudiante.
3-El Maestro después de valorar el resultado del estudiante selecciona el estado en que se encuentra (Avanzado o No Avanzado)	3.1-El sistema muestra un mensaje de correcto envío de datos: "Sus datos han sido guardados correctamente".
Cursos alternos	
	1.1-Si no hay ningún usuario registrado mostrar mensaje: "No hay ningún usuario registrado."
Pos-condiciones	El estado del estudiante quedará actualizado.
Prioridad	Secundario

Tabla13: Descripción Caso de Uso del Sistema: Conformar Laberinto.

Tabla14: Descripción Caso de Uso del Sistema: Obtener Objetos.

Nombre del caso de uso	Conformar Laberinto
Actores	Estudiante
Propósito	Mostrar todas las funcionalidades del juego.
Resumen: El caso de uso se inicia cuando es necesario mostrarle al estudiante la interfaz y funcionalidades que se llevarán a cabo durante el juego.	
Referencias	RF11, RF11.1, RF11.2, RF11.3, RF11.4, RF11.5, RF11.6, RF11.7, RF11.8
Precondiciones	El estudiante debe ejecutar el programa.
Acción del actor	Respuesta del sistema
1-El Estudiante comienza a moverse en el juego	1.1-El sistema genera objetos de forma aleatoria.
2-El Estudiante comienza a recoger objetos.	2.1-El sistema muestra la cantidad de objetos que ha recogido. 2.2-El sistema verifica si ha recogido la cantidad máxima de objetos de ese nivel. 2.3-Si es verdadero ir al Caso de Uso Responder Pregunta de Nivel.
3-El Estudiante culmina el nivel.	3.1-El sistema almacena automáticamente sus datos en la Base de Datos hasta que se agote el tiempo de ejecución del juego. La Base de Datos la actualiza, guardando solamente los datos del último juego.
Cursos Alternos	
	2.4-Si no está correcto el sistema muestra un mensaje: "Usted ha perdido" y almacena automáticamente sus datos en la Base de Datos hasta que se agote el tiempo de ejecución del juego. La actualiza guardando solamente los datos del último juego.
Pos-condiciones	Los resultados del estudiante quedarán actualizados.
Prioridad	Secundario

Nombre del caso de uso	Obtener Objetos	
Actores	Estudiante	
Propósito	Interactuar con los objetos que se muestran en el laberinto.	
Resumen:	Este caso de uso se inicia cuando es necesario mostrarle al estudiante cada uno de los objetos que deberá obtener durante el juego.	
Referencias	RF12, RF12.1, RF12.2, RF12.3, RF12.4, RF12.5	
Precondiciones	Se debe generar el laberinto.	
Acción del actor	Respuesta del sistema	
1-El Estudiante comienza a moverse en el laberinto.	1.1-El sistema genera un objeto y muestra la cantidad de objetos a recoger.	
2-El Estudiante obtiene de uno en uno los objetos visualizados.	2.1-El sistema comienza a sumar la cantidad de objetos recogidos	
Pos-condiciones	Cuando el estudiante obtenga la cantidad total de objetos a recoger en ese laberinto culminará el nivel.	
Prioridad	Secundario	

Tabla15: Descripción Caso de Uso del Sistema: Ejecutar Pregunta de Nivel.

Nombre del caso de uso	Ejecutar Pregunta Nivel
Actores	Estudiante
Propósito	Mostrar, responder y evaluar una pregunta al culminar cada nivel.
Resumen: El estudiante al terminar el nivel deberá responder la pregunta nivel, si responde correctamente aumenta el tiempo de ejecución del programa, de lo contrario, continua el próximo nivel con el tiempo restante.	
Referencias	RF13, RF13.1, RF13.2, RF13.3, RF13.4, RF13.5
Precondiciones	El Estudiante debe culminar el nivel.
Acción del actor	Respuesta del sistema
1-El Estudiante culmina el nivel	1.1-El sistema muestra la pregunta nivel según grado escolar del estudiante y al mismo tiempo detiene el tiempo de ejecución del juego.
2-El Estudiante responde pregunta nivel.	2.1-El sistema verifica si es correcta la respuesta de la pregunta nivel 2.2-Si es correcta el sistema agrega al tiempo de ejecución un tiempo adicional carga el próximo nivel de dificultad activa nuevamente el tiempo de ejecución del juego.
3-El Estudiante continúa el juego en el nuevo nivel de dificultad.	
Cursos Alternos	
	2.3-Si no es correcta el sistema carga el próximo nivel de dificultad y activa nuevamente el tiempo de ejecución del juego.
Pos-condiciones	El Estudiante pasará al próximo nivel y se actualizará el tiempo de ejecución del juego.
Prioridad	Secundario

Consideraciones finales

En el presente capítulo se especificaron las soluciones técnicas que se le dan a este proyecto, partiendo de aquí se diseñará una estructura de clases en correspondencia con las técnicas citadas. Se establecieron además los requisitos funcionales y no funcionales, propiciando el paso a la siguiente etapa, donde se realizaron las descripciones detalladas de los casos de uso a implementar, apoyados en el modelo de los casos de uso del sistema, permitiendo obtener los resultados esperados por el cliente.

Capítulo 3 Análisis y Diseño Del Sistema

3.1 Introducción

En el actual y último capítulo se concibe específicamente el análisis y diseño del sistema, el cual está reflejado por una serie de diagramas que le permitirán a un futuro programador continuar con los flujos de trabajo de implementación y prueba de la aplicación.

A continuación se muestran los diagramas de clases de análisis del sistema propuesto, con el objetivo de dar una idea a las clases de diseño a realizar posteriormente, donde podemos apreciar además sus responsabilidades y relaciones.

Durante el ciclo de desarrollo iterativo es posible pasar a la fase de diseño, una vez terminados estos diagramas de análisis. Durante este paso se logra una solución lógica que se funda en el paradigma orientado a objetos. Los diagramas de diseño de clases resumen la definición de las clases (e interfaces) implementables en el software.

A continuación se muestran los diagramas de secuencia que a su vez describen la realización de cada uno de los casos de uso. Finalmente se muestra el diagrama de las clases persistentes del cual se genera el diseño de la base de datos del sistema.

3.2 Descripción de la Arquitectura

El objetivo de la Arquitectura del Software es aportar elementos que ayuden a la toma de decisiones y, al mismo tiempo, proporcionar conceptos y un lenguaje común que permitan la comunicación entre los equipos que participen en un proyecto.

La Arquitectura del Software construye abstracciones, materializándolas en forma de diagramas comentados. Para realizar la propuesta de análisis y diseño del sistema que le de cumplimiento a la situación problemática se utilizará la Arquitectura Tres Capas. La misma define cómo organizar el modelo de diseño en capas, donde los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores.

Es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma tal que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Las tres capas están compuestas por:

- *Capa de Presentación:* Esta capa contiene las clases interfaces de la aplicación con las que va a interactuar el usuario, estas son las encargadas de gestionar todas las peticiones que el mismo realice.
- *Capa de Negocio:* Esta capa contiene toda la lógica del negocio, en ella se encuentran las clases controladoras, en este caso se realizó una clase controladora por cada caso de uso identificado en el sistema.
- *Capa de Datos:* En esta capa se guardan todos los datos necesarios para la aplicación, es donde se encuentra la base de datos que esta contiene una clase por cada entidad.

Arquitectura Tres Capas.

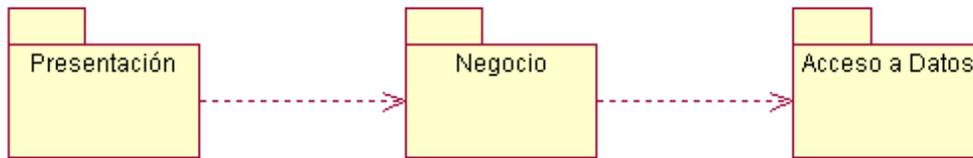


Figura 6. Arquitectura Tres Capas.

3.3 Diagramas de Clases de Análisis.

A continuación se muestran los diagramas de clases de Análisis propuestos que constituyen una concepción más fácil de las clases de diseño del sistema.

3.3.1 CU: Gestionar Usuario

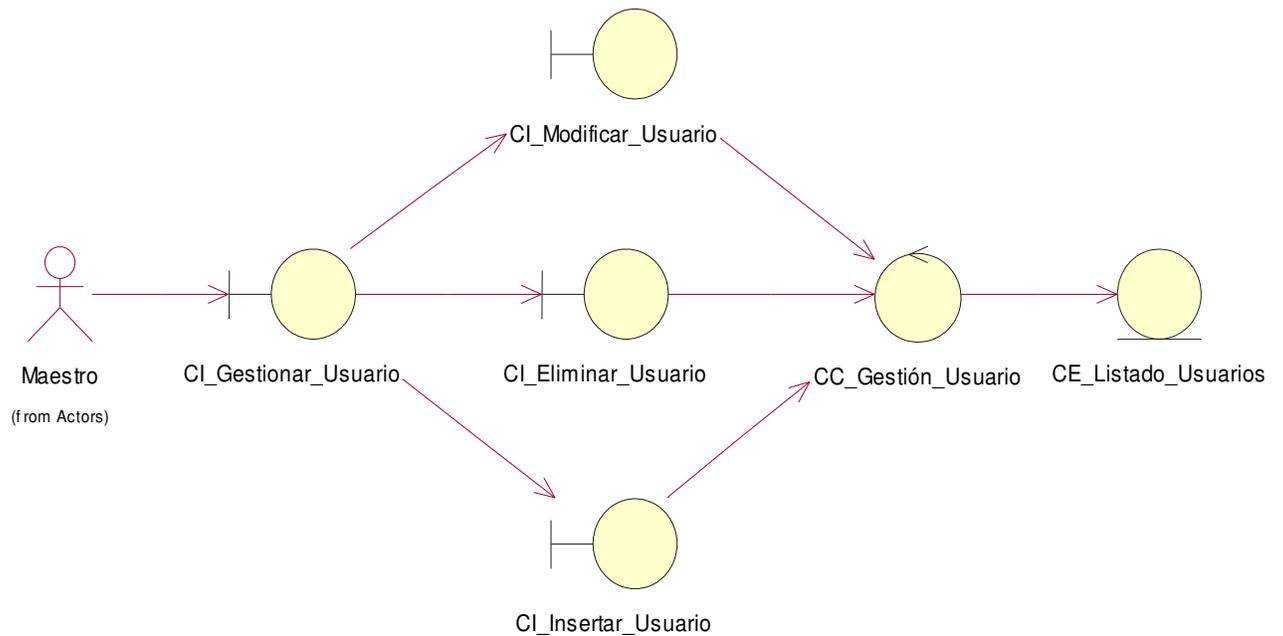


Figura 7. Clase de Análisis: Gestionar Usuario.

3.3.2 CU: Autenticar Usuario

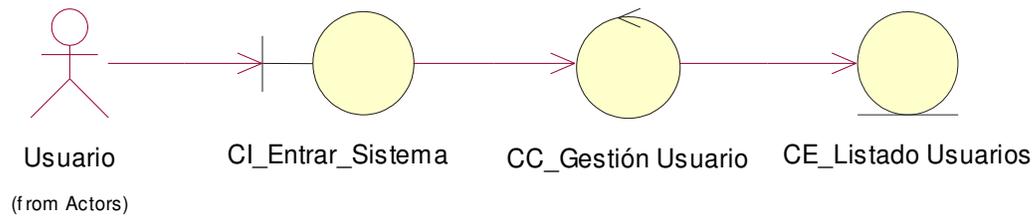


Figura 8. Clase de Análisis: Autenticar Usuario.

3.3.3 CU: Conformar Laberinto

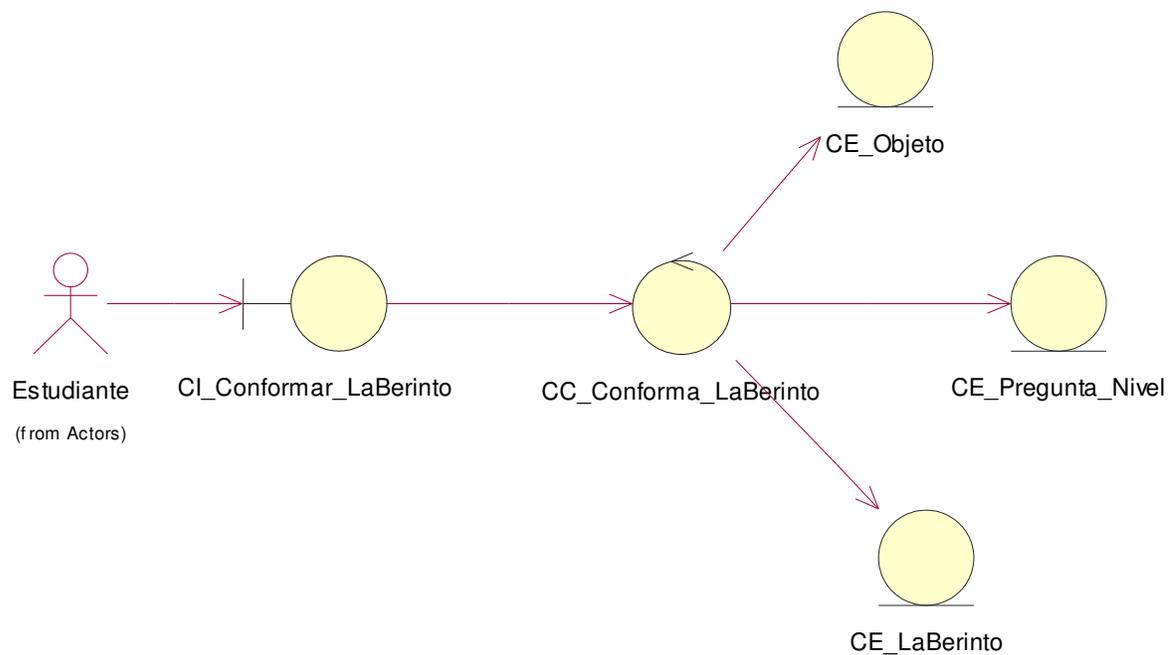


Figura 9. Clase de Análisis: Conformar Laberinto.

3.3.4 CU: Obtener Objetos

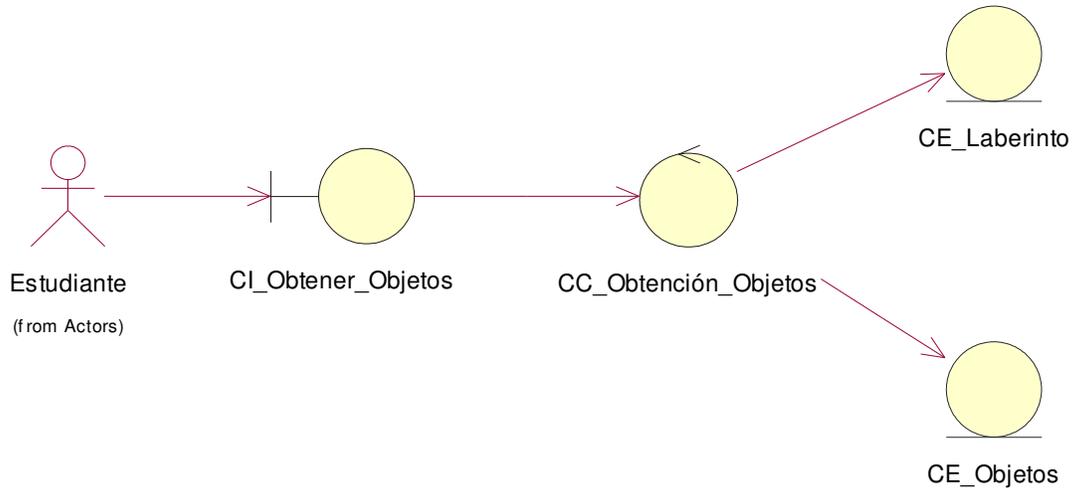


Figura 10. Clase de Análisis: Obtener Objetos.

3.3.5 CU: Ejecutar Pregunta Nivel

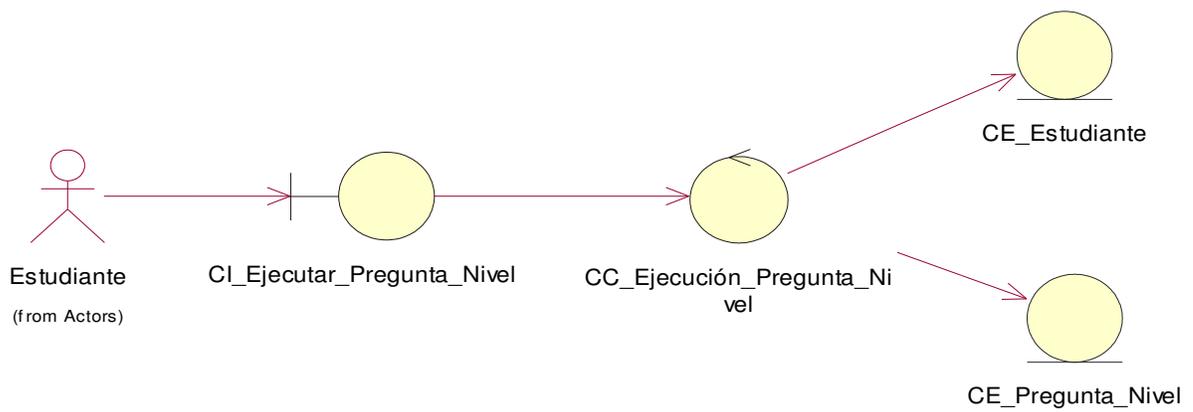


Figura 11. Clase de Análisis: Ejecutar Pregunta Nivel.

3.3.6 CU: Revisar Resultados

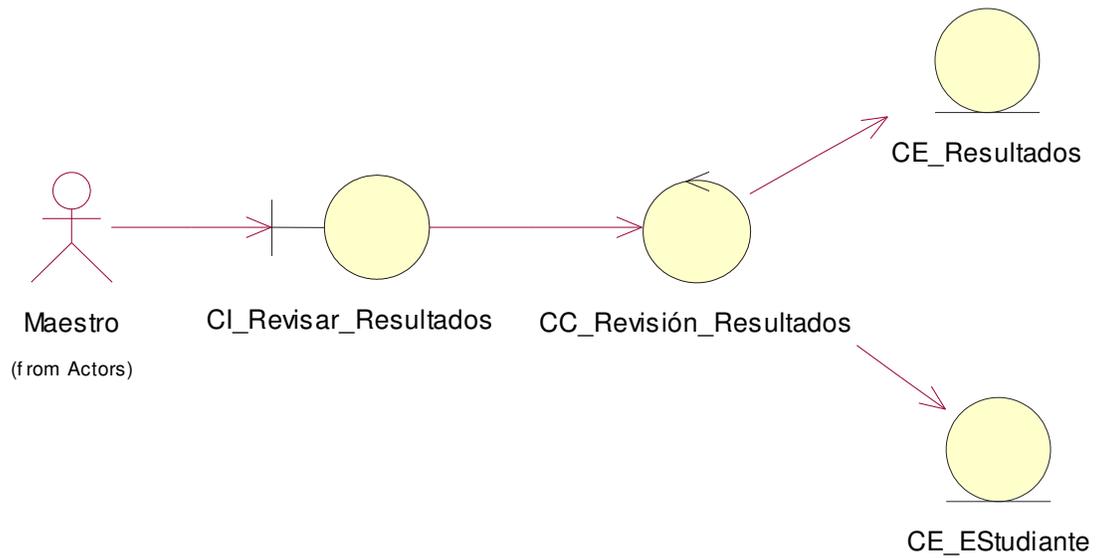


Figura 12. Clase de Análisis: Revisar Resultados.

3.4 Diagramas de Clases de Diseño

En la fase de diseño se modela el sistema de manera que soporte todos los requisitos, tanto funcionales como no funcionales.

3.4.1 CU: Gestionar Usuario

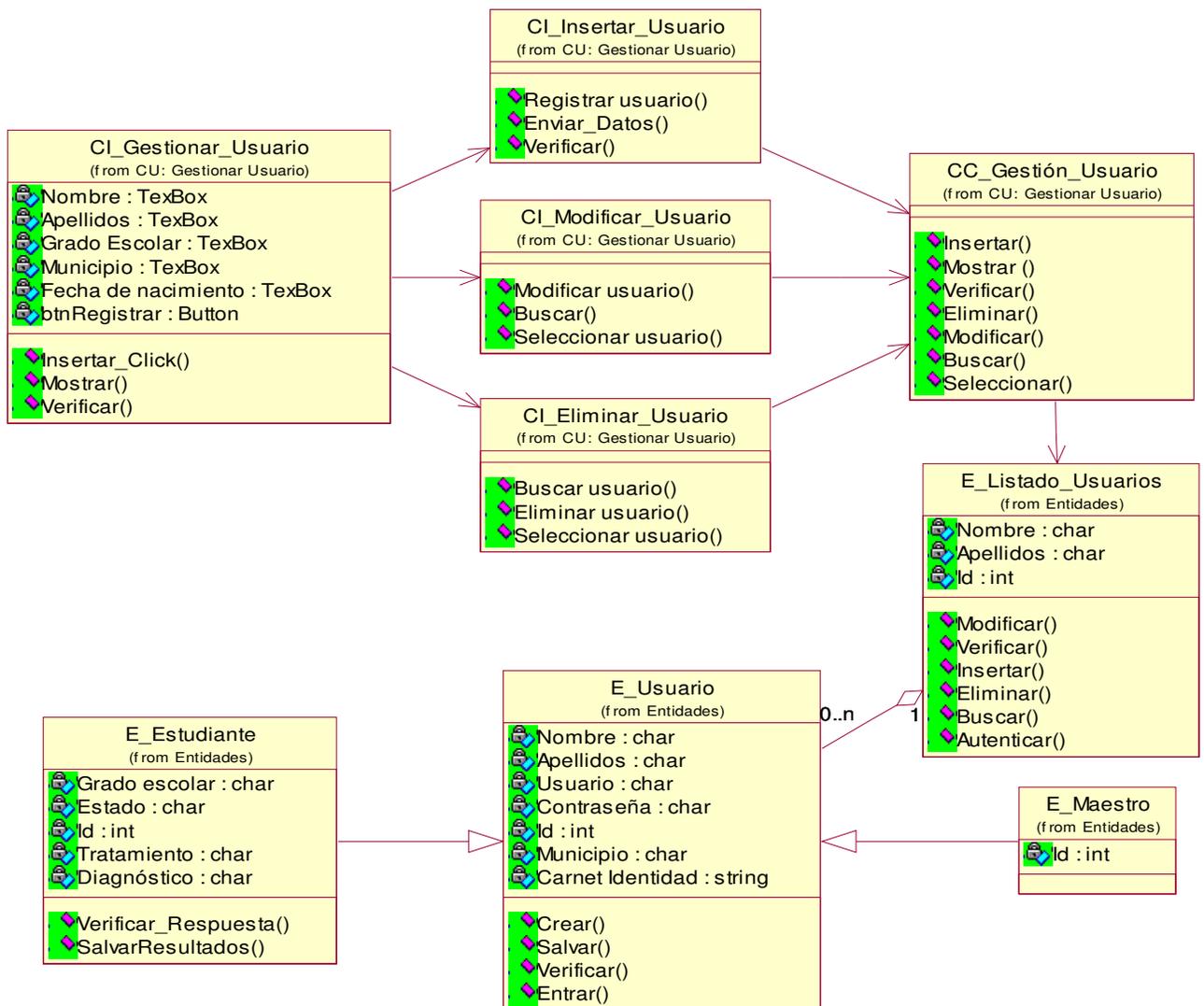


Figura 13. Clase de Diseño: Gestionar Usuario.

3.4.2 CU: Autenticar Usuario

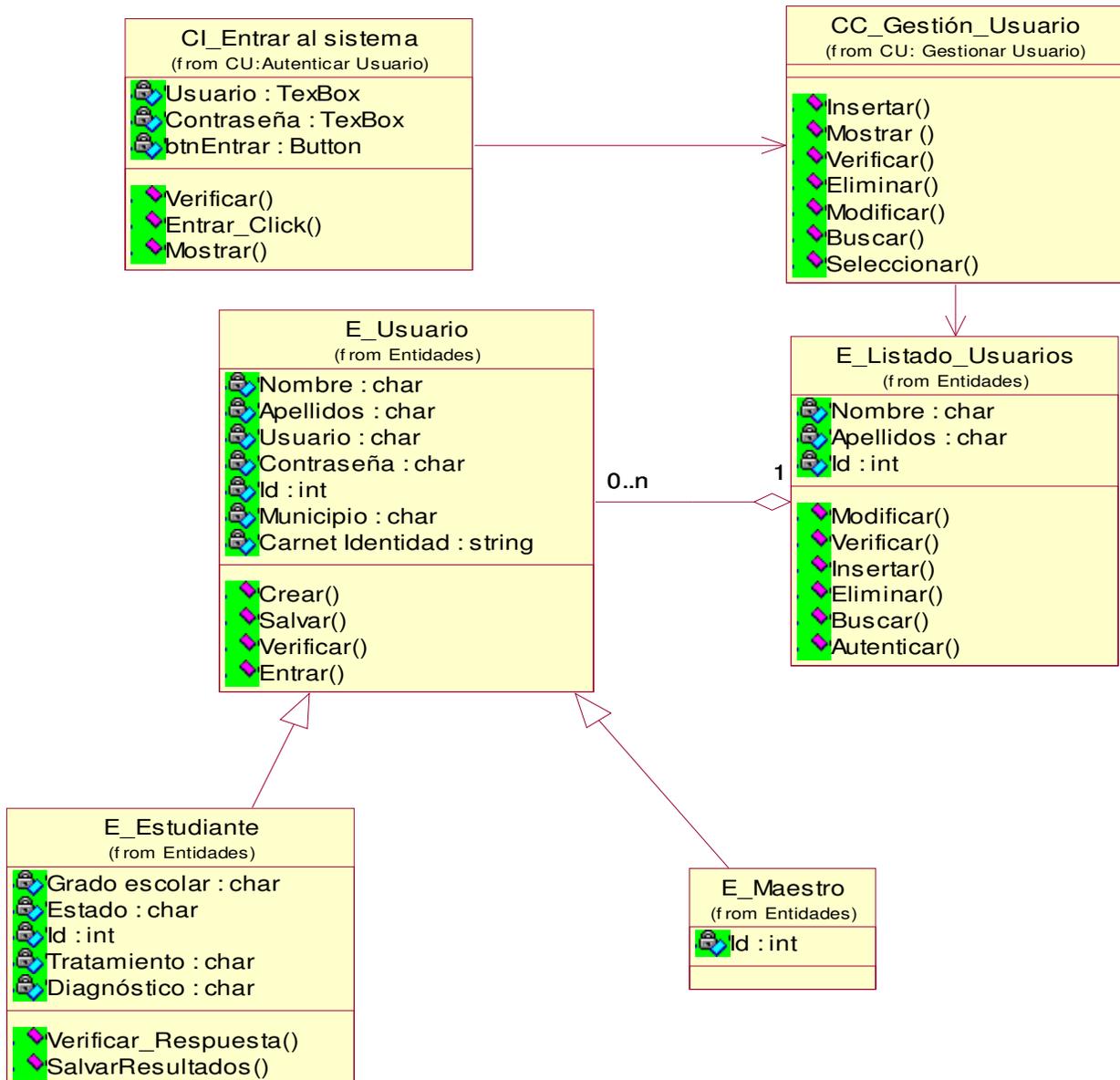


Figura 14. Clase de Diseño: Autenticar Usuario.

3.4.3 CU: Revisar Resultados

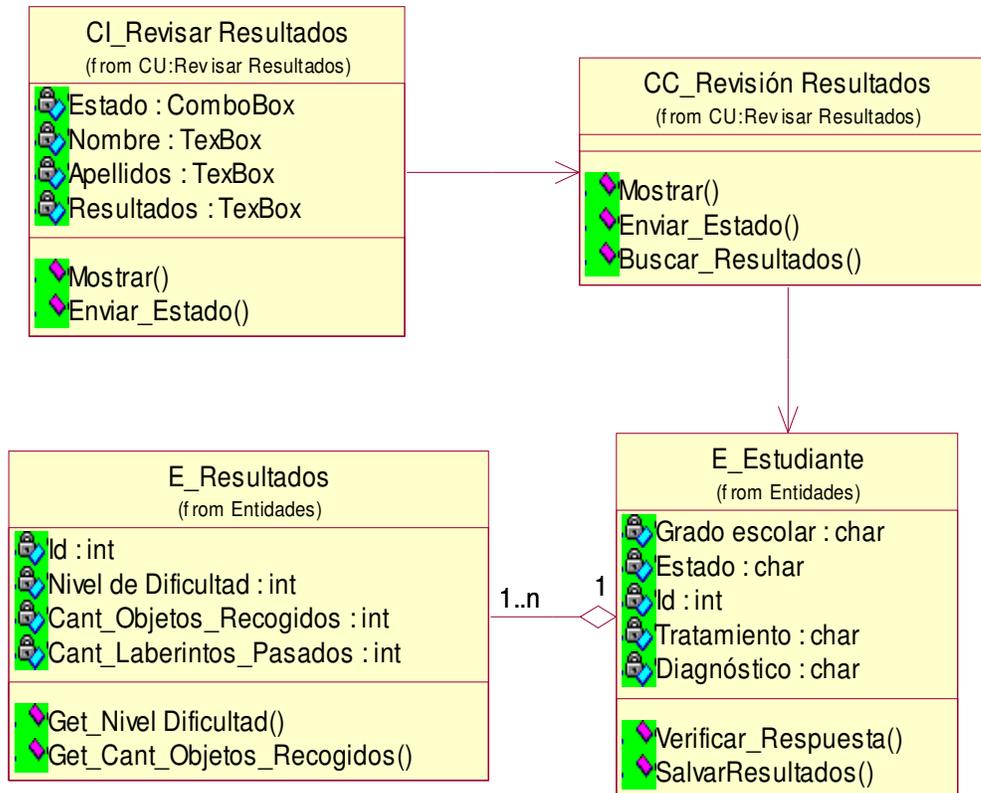


Figura 15. Clase de Diseño: Revisar Resultados.

3.4.4 CU: Conformar Laberinto

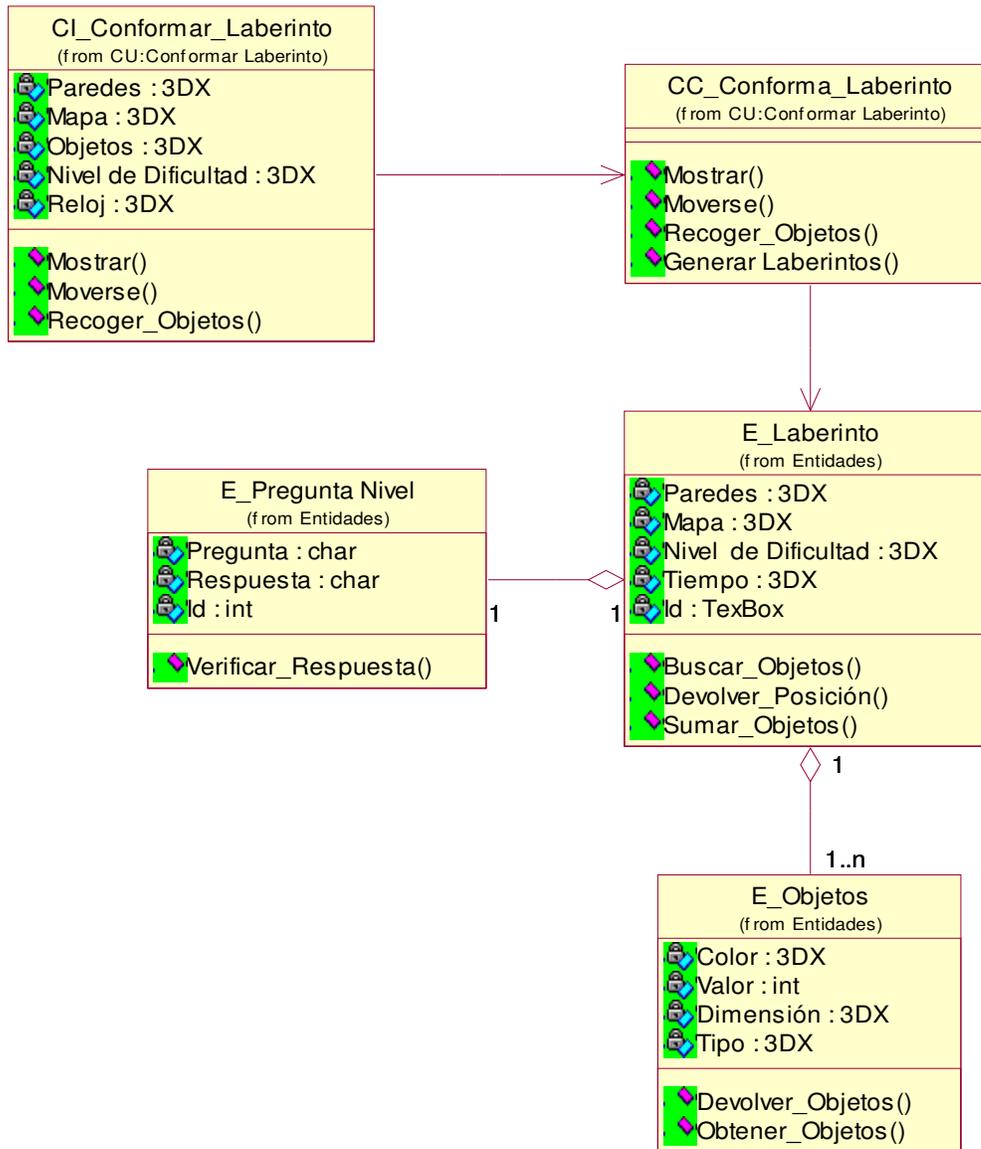


Figura 16. Clase de Diseño: Conformar Laberinto.

3.4.5 CU: Obtener Objetos

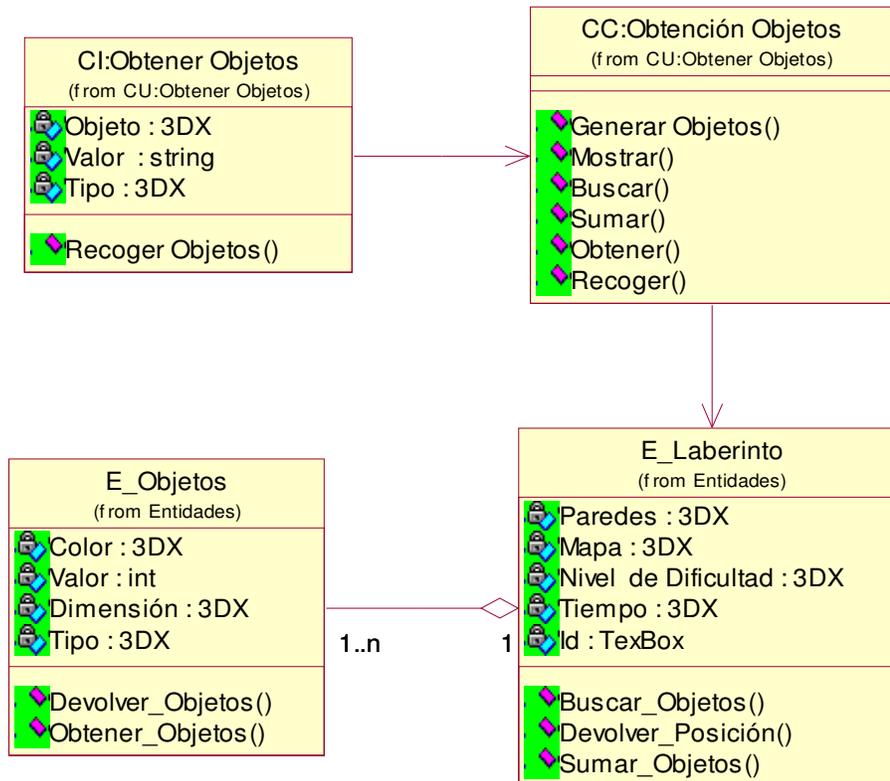


Figura 17. Clase de Diseño: Obtener Objetos.

3.4.6 CU: Ejecutar Pregunta Nivel

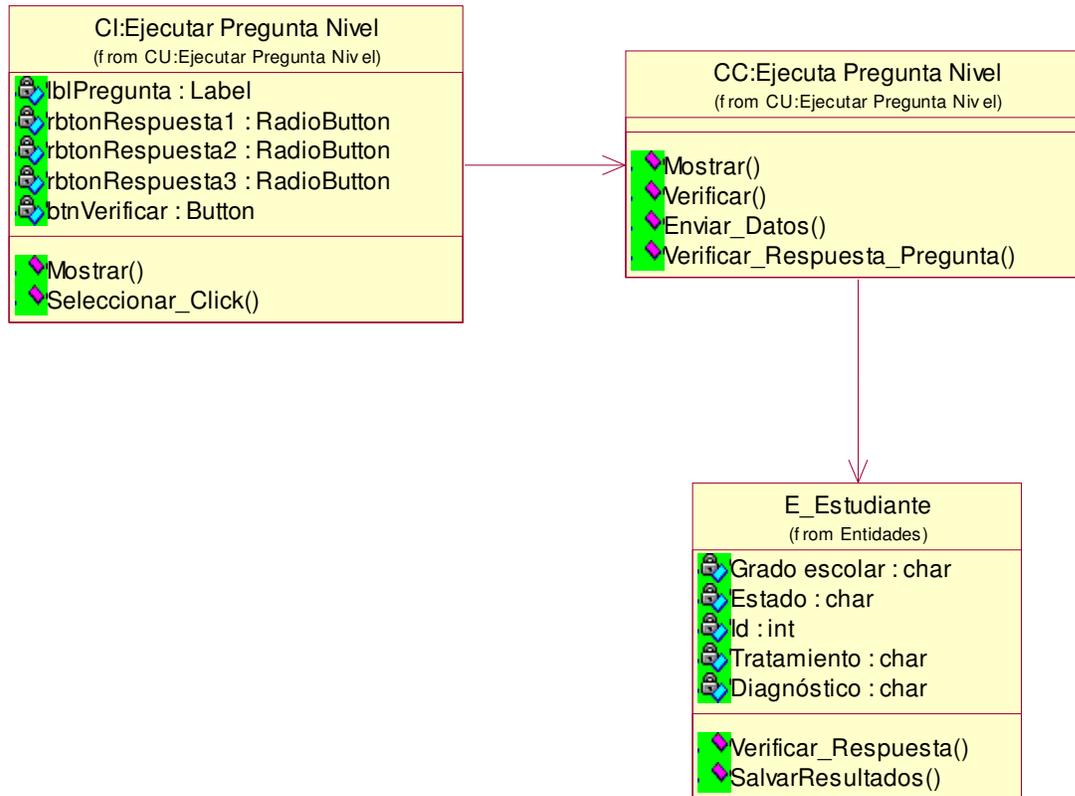


Figura 18. Clase de Diseño: Ejecutar Pregunta Nivel.

3.5 Diagramas de Secuencia:

Los diagramas de interacción se dividen en dos tipos de diagramas de UML, los diagramas de secuencia y los diagramas de colaboración. Para modelar los aspectos dinámicos de este sistema se utilizará diagramas de secuencia por cada uno de los escenarios de cada caso de uso, destacando la secuencia temporal de los mensajes.

3.5.1

CU:

Gestionar

Usuario:

Insertar

Usuario

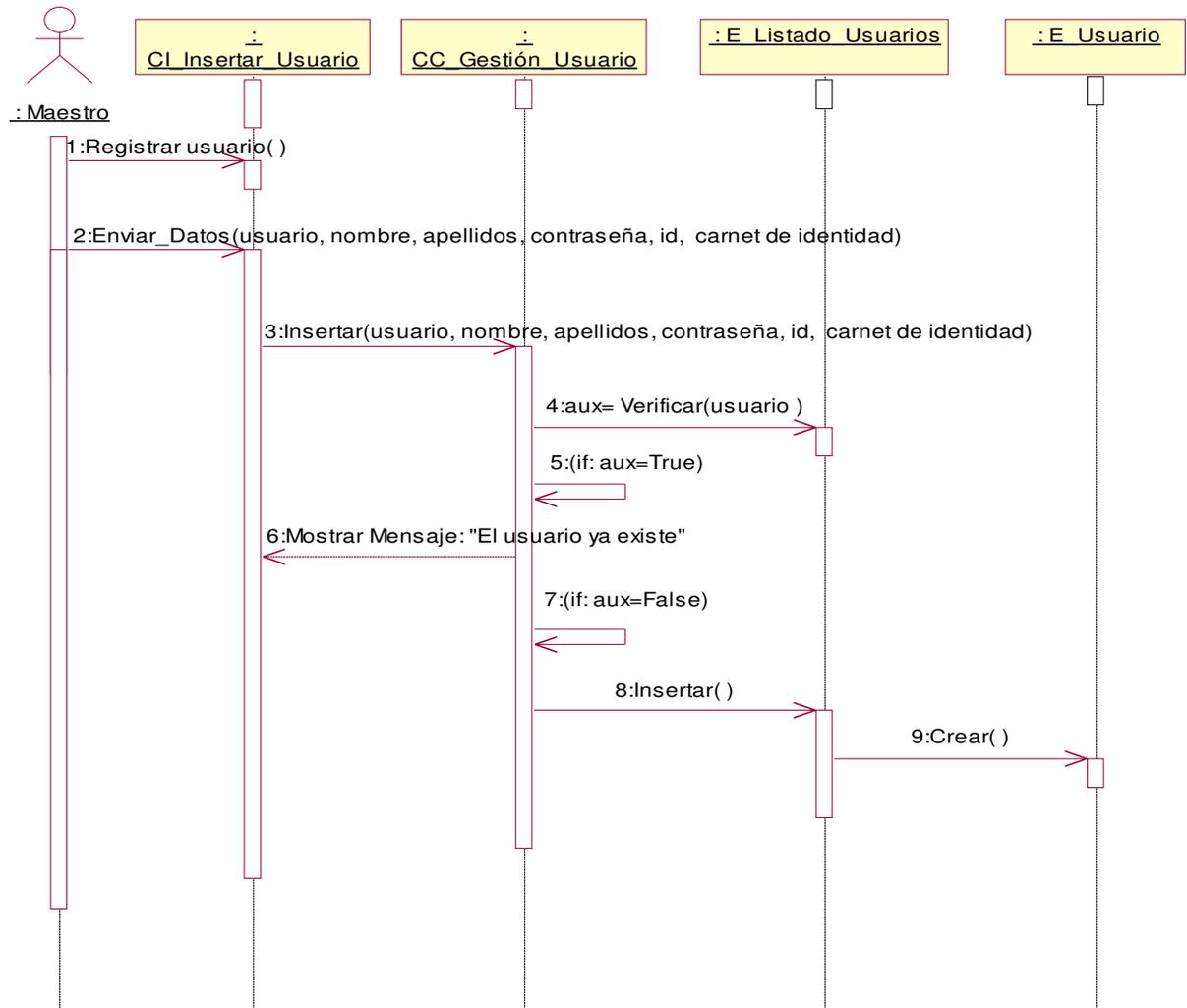


Figura 19. Clase de Secuencia: Gestionar Usuario: Insertar Usuario.

3.5.2 CU: Gestionar Usuario: Modificar Usuario

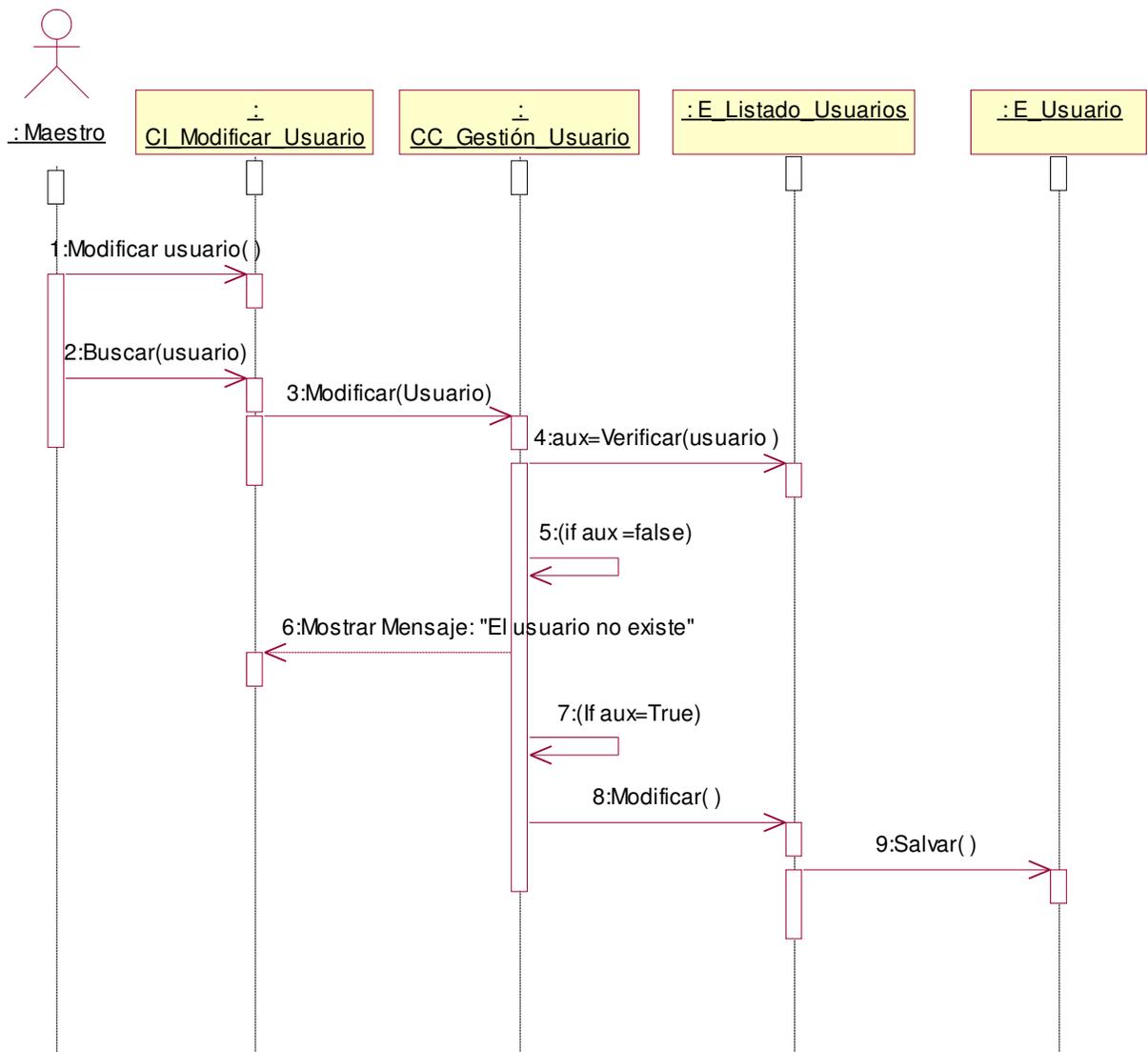


Figura 20. Clase de Secuencia: Gestionar Usuario: Modificar Usuario.

3.5.3 CU: Gestionar Usuario: Eliminar Usuario

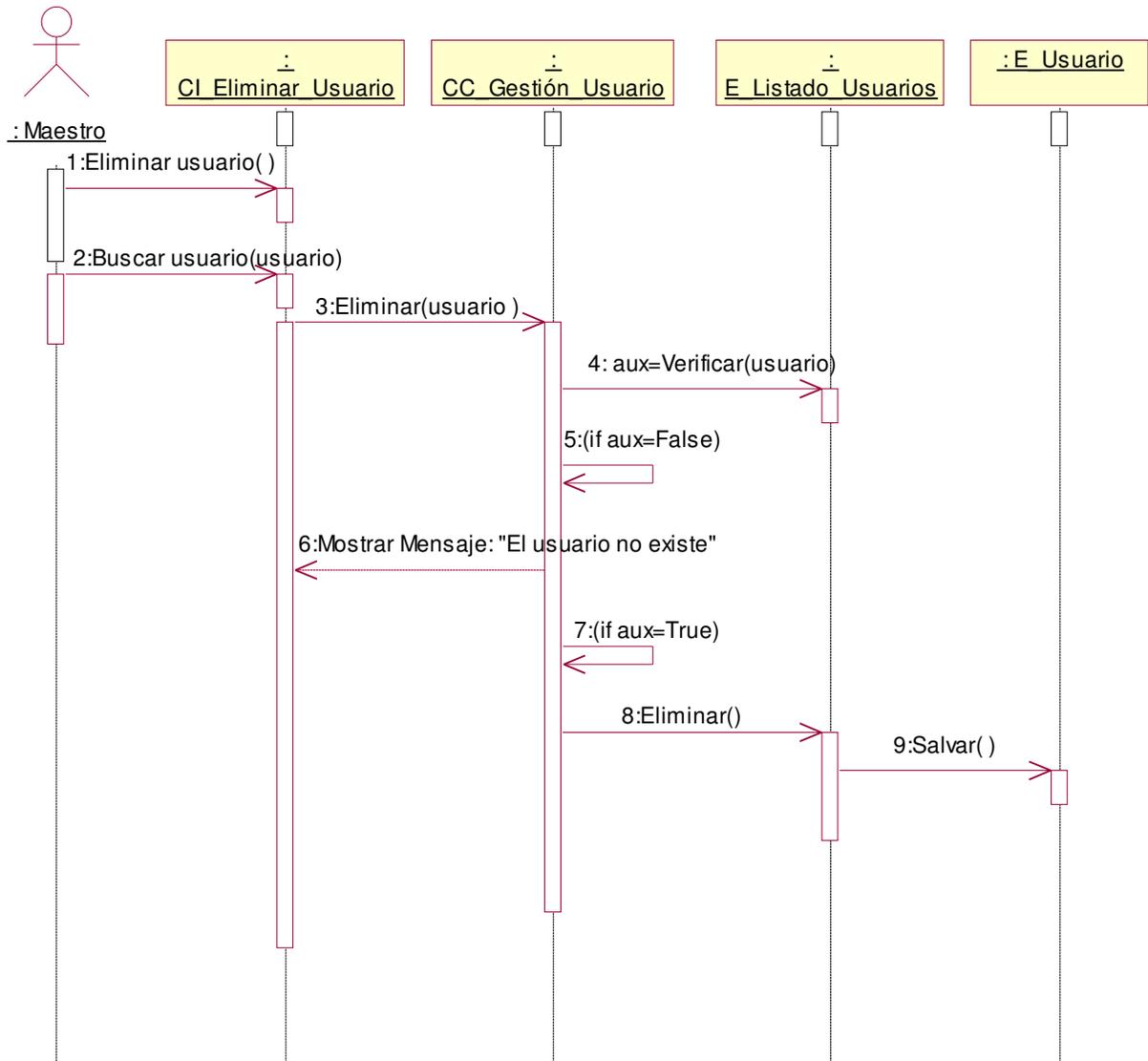


Figura 21. Clase de Secuencia: Gestionar Usuario: Eliminar Usuario.

3.5.4 CU: Autenticar Usuario

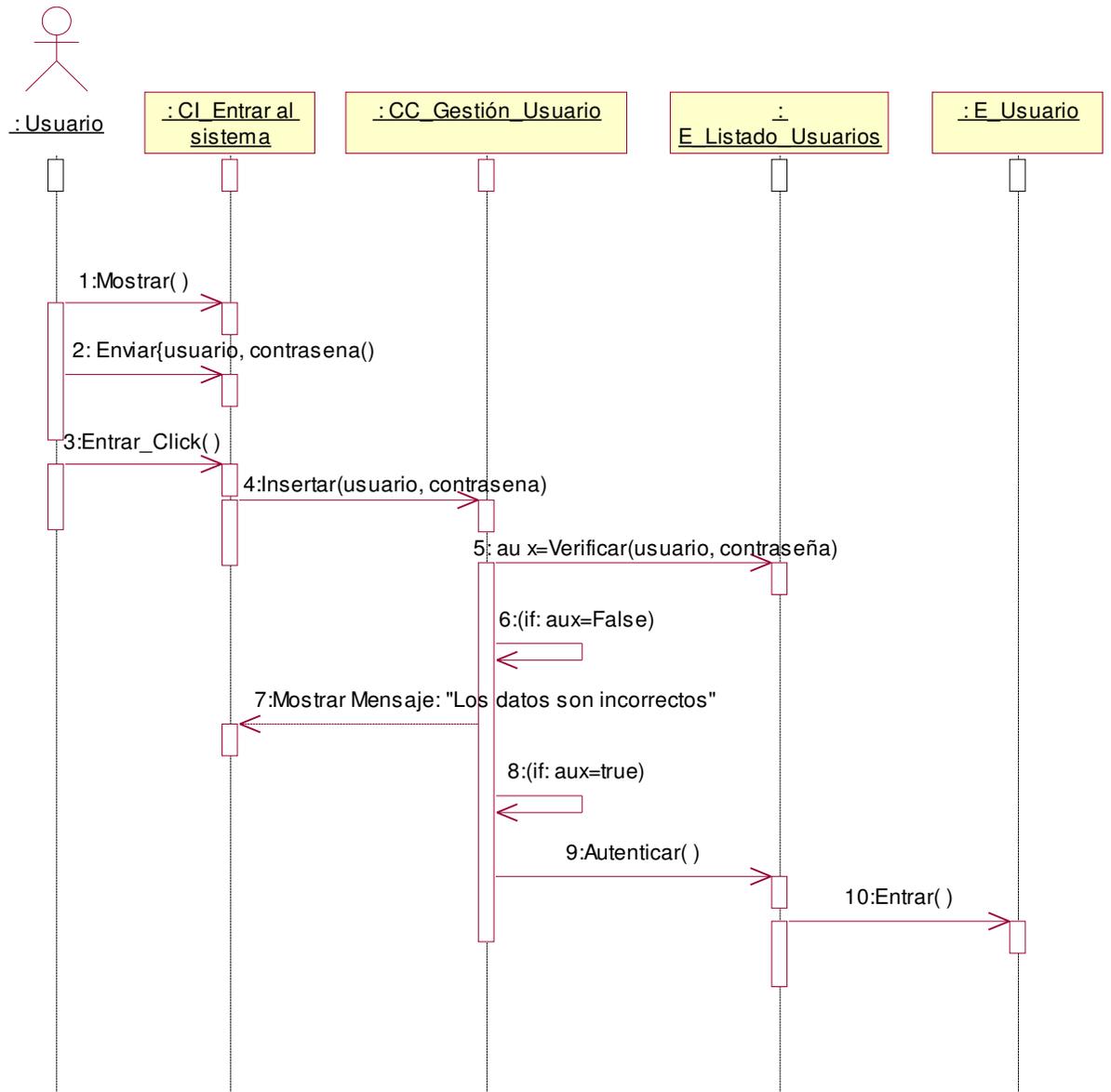


Figura 22. Clase de Secuencia: Autenticar Usuario.

3.5.5 CU: Ejecutar Pregunta Nivel

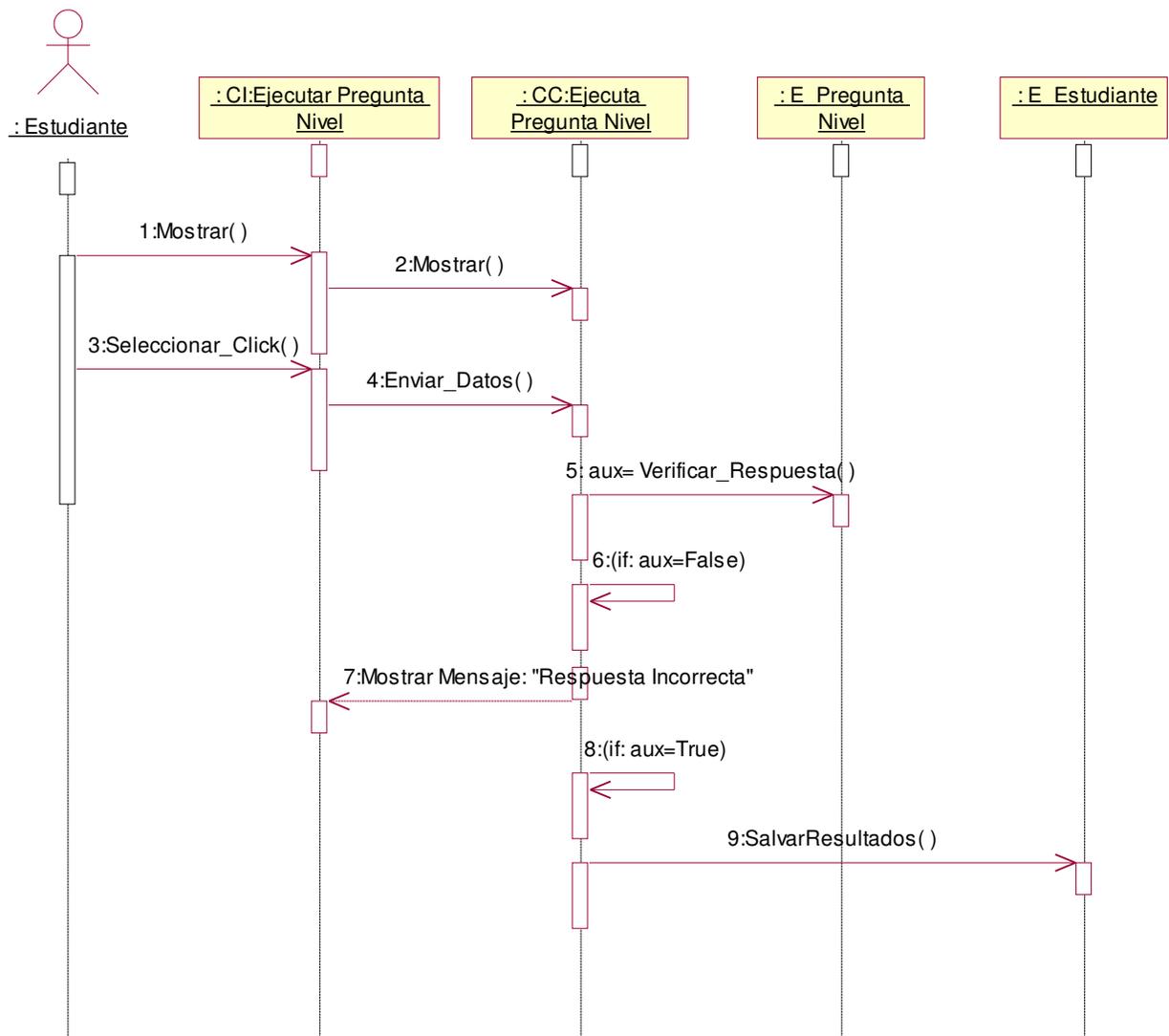


Figura 23. Clase de Secuencia: Ejecutar Pregunta Nivel.

3.5.6 CU: Revisar Resultados

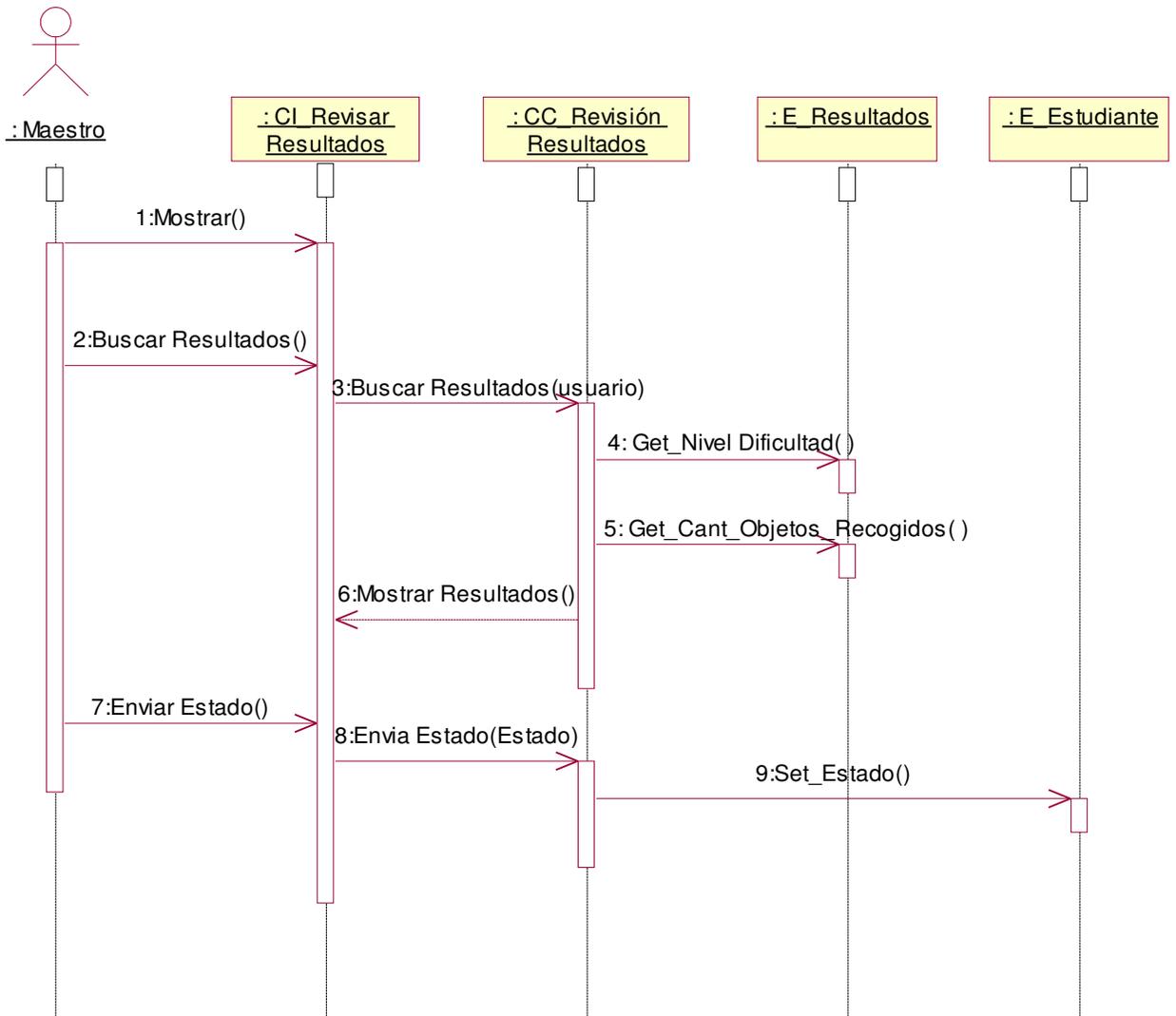


Figura 24. Clase de Secuencia: Revisar Resultados.

3.5.7 CU: Conformar Laberinto

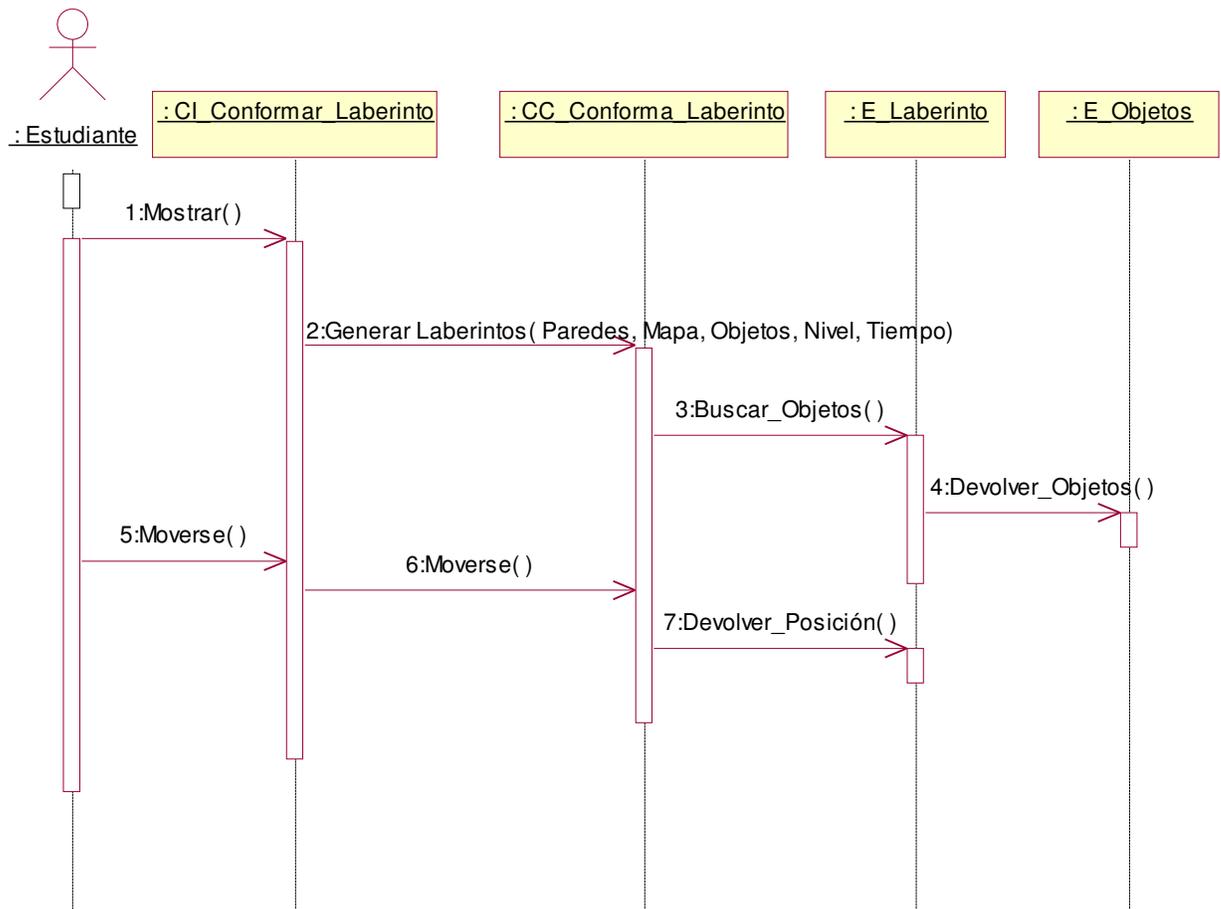


Figura 25. Clase de Secuencia: Conformar Laberinto.

3.5.8 CU: Obtener Objetos

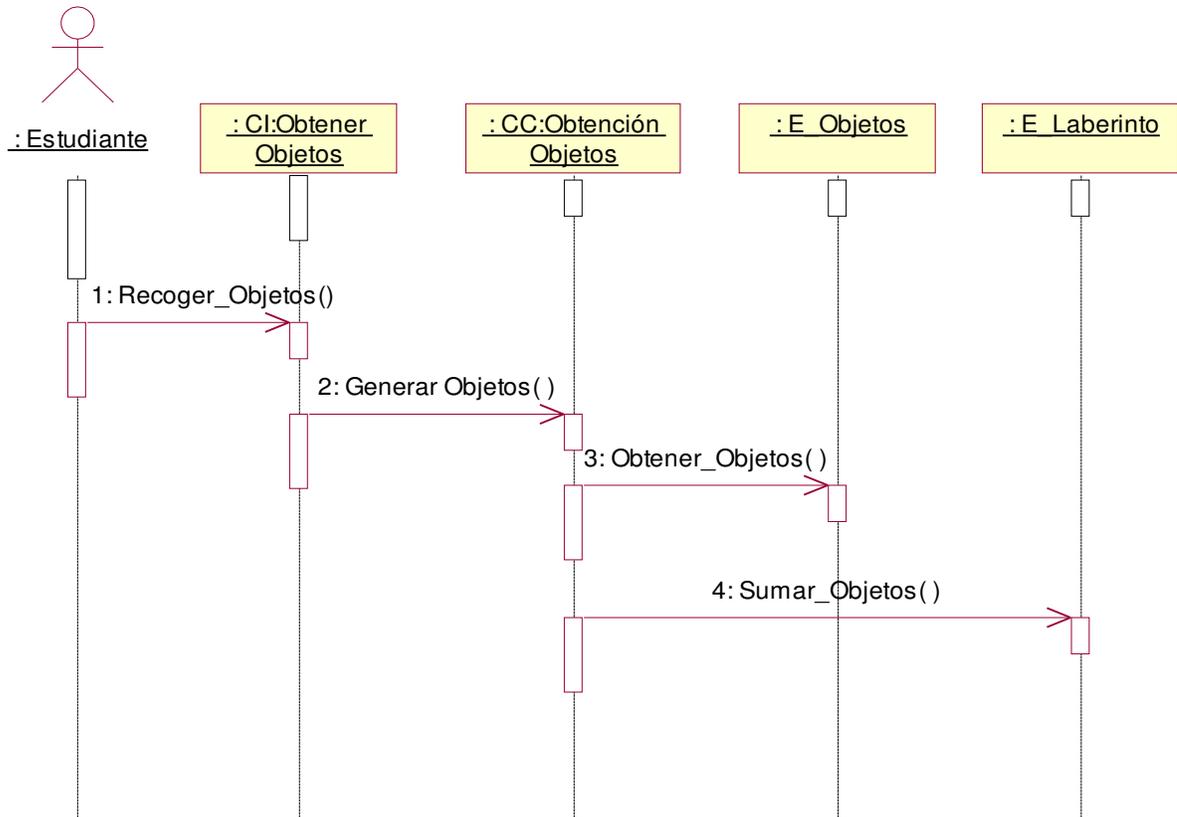


Figura 26. Clase de Secuencia: Obtener Objetos.

3.6 Diagrama de Clases Persistentes

A continuación se muestra el diagrama de Clases Persistentes compuesto por las clases de una aplicación, las cuales implementan las entidades del problema de negocio. Estas clases modelan la información y el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto del mundo real o un suceso.

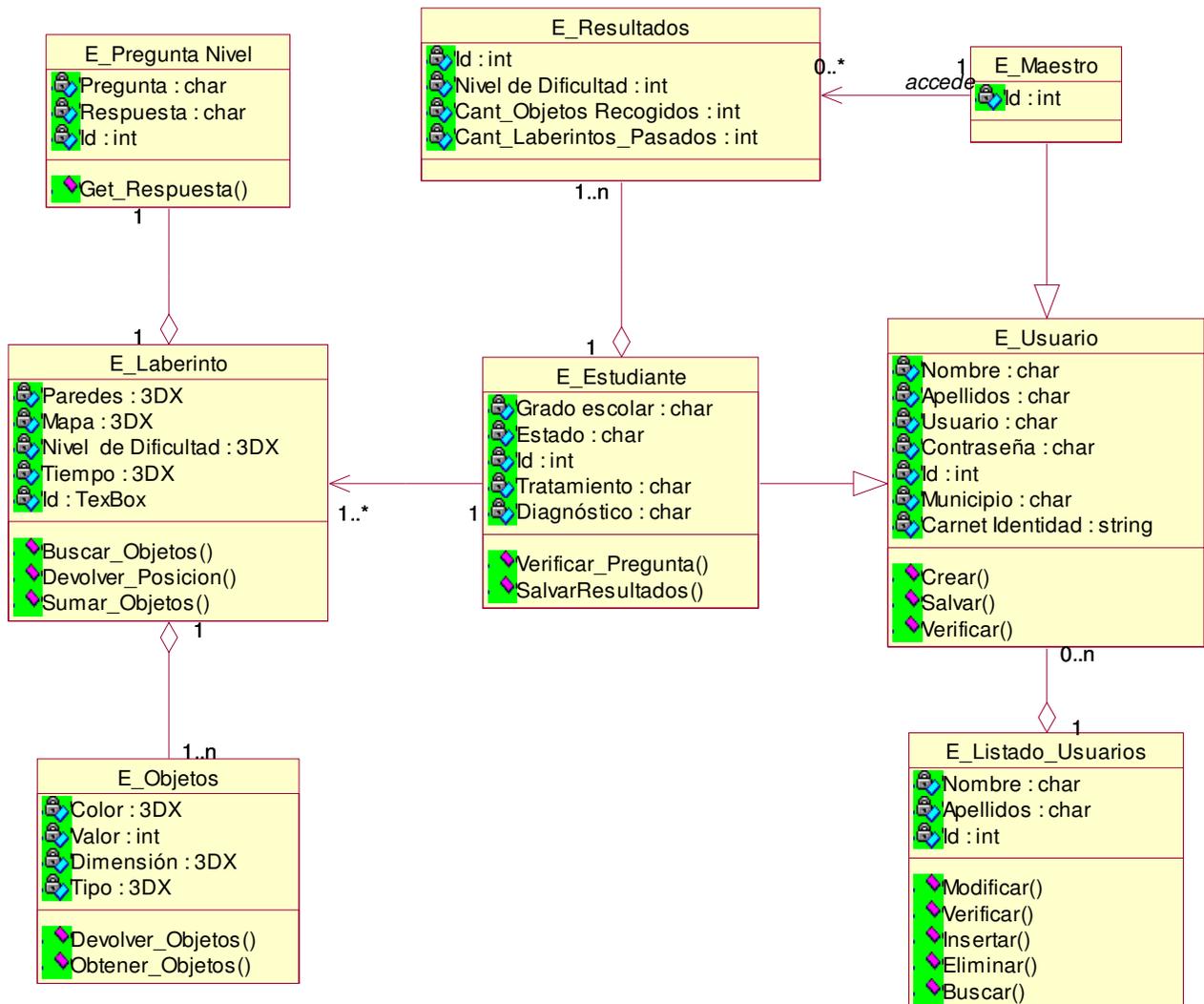


Figura 27. Diagrama de Clases Persistentes.

3.7 Modelo de Datos

A continuación se muestra el Modelo de datos, el cual describe la representación lógica y física de los datos persistentes usados por la aplicación. Este modelo es usado para describir la representación lógica y física de la información persistente manejada por el sistema.

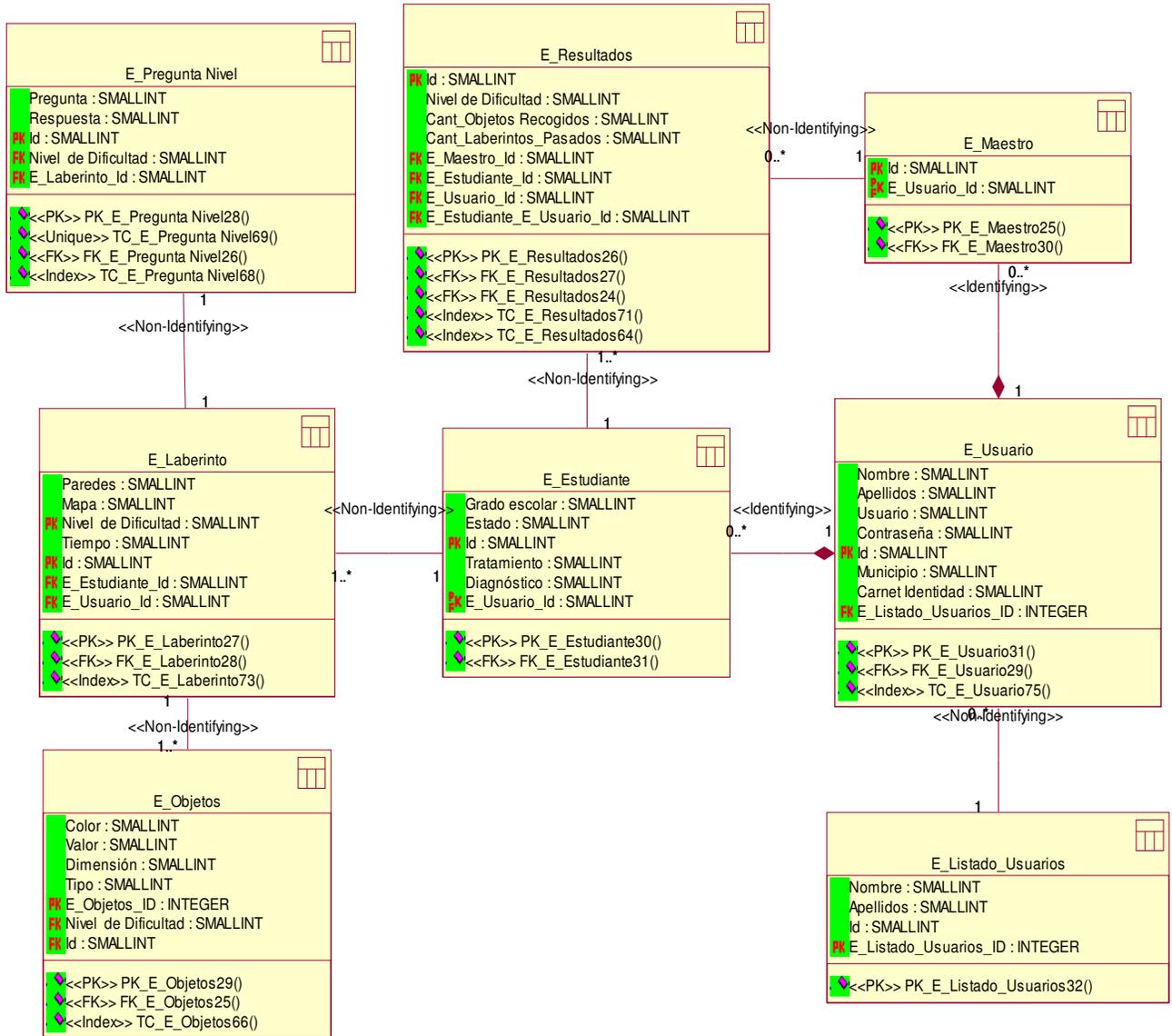


Figura 28. Modelo de Datos.

Consideraciones finales

Al concluir este capítulo se tiene un total diseño del sistema a desarrollar y la secuencia de pasos traducida a mensajes entre clases de los primeros casos de usos a desarrollar, por lo que queda lista la propuesta de diseño para pasar a la implementación de este proyecto.

Conclusiones:

Las teorías estudiadas sobre el proceso de rehabilitación de escolares estrábicos y ambliopes permitieron una mejor comprensión de los requisitos a tener en cuenta para el análisis y diseño del software a implementar en el futuro.

La estructuración del modelo, a partir de los requisitos tanto de la caracterización del escolar como de la tecnología presente en las escuelas para dichos niños, contribuirá a que el proceso de implementación conste con una coherencia importante a la hora de ser utilizados.

Recomendaciones

Al concluir este trabajo se recomienda lo siguiente:

- Profundizar en el estudio y creación de software para la estimulación visual en niños estrábicos y/o ambliopes.
- Utilizar Patrones de diseño con el objetivo de mostrar de manera más concreta la solución basada en el contexto del problema.
- Realizar la implementación del software basado en el análisis y diseño propuesto en este trabajo e implantarlo paulatinamente en todas las escuelas de atención a débiles visuales según las condiciones existentes en nuestro país.

Bibliografía

1. Ávila Saint, N. y coautor. "Un reto a la minusvalía desde la escuela especial para deficientes visuales en Santiago de Cuba": En libro de Resúmenes del II Encuentro Mundial de la Educación Especial. La Habana. 1998.
2. Barraga, N. "Textos Reunidos". Cap.1."Libro sobre Baja Visión". ONCE (Organización Nacional de Ciegos Españoles). Madrid. 1997.
3. Bell Rodríguez, R. "Educación Especial. Razones, visión actual y desafíos". Editorial Pueblo y Educación. Ciudad Habana. 1998.
4. Hernández Sampier, R. "Metodología de la Investigación". Libro en 2 tomos. Editorial Feliz Varela. La Habana, 2004
5. Martin Ford, D. "Los Software Educativos en el proceso de corrección y/o compensación del estrabismo y la ambliopía". Trabajo de Diploma. 2003. Ciudad Habana.
6. Pérez Rodríguez, G. y coautores. "Metodología de la Investigación Educativa". Editorial Pueblo y Educación. Ciudad Habana. 1998.
7. Pérez Martinto P.C. 1999. "Propuesta de Software Educativo para la Estimulación Visual de los escolares Ambliopes con Fijación Central". Tesis de Maestría. Ciudad de La Habana: Centro de Referencia Latinoamericana para la Educación Especial - CELAEE - IPLAC, 1999.
8. _____ "Propuesta de Software Educativo para la Estimulación Visual de los escolares Ambliopes con Fijación Central". Tesis de Maestría. 1999. Ciudad Habana.
9. Velásquez, B. y coautor. "Análisis de la agudeza visual con diferentes técnicas en pacientes ambliopes". México. 2003.
10. Vigostki, LS. "Obras Escogidas en V Tomos. Tomo V- LS. Vigostki. Editorial Pueblo y Educación. Ciudad Habana.

Libros Digitales

11. Sánchez, María A. Mendoza. "Metodologías De Desarrollo De Software". [Consultado en noviembre 2008]. Disponible en:
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
12. Seco, José Antonio González. 2002. "El Lenguaje de Programación C#". [Consultado en noviembre 2008]. Disponible en:
<http://www.programacion.com/tutorial/csharp/3/>.
13. INNOVA, Grupo Soluciones. Rational Rose Enterprise. [Consultado en diciembre 2008]. Disponible en:
<http://www.rational.com.ar/herramientas/roseenterprise.html>.
14. Netveloper. 2002. SQL Server 2000. [Consultado en enero 2009]. Disponible en: http://www.netveloper.com/contenido2.aspx?IDC=64_0.
15. Guerrero, Fernando Tala. "Lenguaje C++. Guía para Programadores". [Consultado en enero 2009]. Disponible en:
<http://www.mailxmail.com/curso/informatica/cplusplus2>.
16. Unai Extremo Baigorri, Borja Sotomayor Basili. La plataforma .NET: ¿El futuro de la Web? [Consultado en febrero 2009]. Disponible en:
<http://people.cs.uchicago.edu/~borja/pubs/revistaeside2002.pdf>.

Sitios Web

17. La Plataforma .NET. Disponible en:
<http://ange.nireblog.com>.
18. XML lenguaje de marcado. Disponible en:
http://www.tejedoresdelweb.com/w/XML_lenguaje_de_marcado.
19. Introducción a la plataforma .NET y Mono. Disponible en:
<http://www.eslomas.com/index.php/archives/2005/05/11/introduccion-plataforma-net-y-mono/>.
20. Programación extrema. Disponible en:

http://es.wikipedia.org/wiki/Programaci%C3%B3n_Extrema.

21. Why Visual Paradigm for UML? Disponible en:

<http://www.visual-paradigm.com/product/vpuml/>.

22. Published Software Architecture Definitions. Disponible en:

<http://www.sei.cmu.edu/architecture/definitions.htm>!

Anexos:

Anexo 1.



Figura 29. Anexo 1:

Anexo 2.

Laberinto Bangester

Archivo Ver Niveles Ayuda

- Registrar Nuevo Usuario
- Eliminar Usuario
- Salir

rio

Nombre	Eduardo
Primer Apellido	Millán
Segundo Apellido	Nuñez
CI	00091745678
Sexo	M
Edad	8
Grado Escolar	3ro
Municipio	10 de Octubre
Estado	No Avanzado
Diagnóstico	Estrábico
Tratamiento	Estimulación visual

Cancelar Registrar



Anexo 3.

Laberinto Bangester

Archivo Ver Niveles Ayuda

Datos de Usuarios

CI del Estudiante
00073129051

Nombre: Javier
Primer Apellido: Lorie
Segundo Apellido: Guerra
Edad: 8
Sexo: M
Grado Escolar: 3ro
Municipio: Playa
Diagnóstico: Ambliopía Ojo Izq
Estado: Avanzado
Tratamiento: Estimulación visual



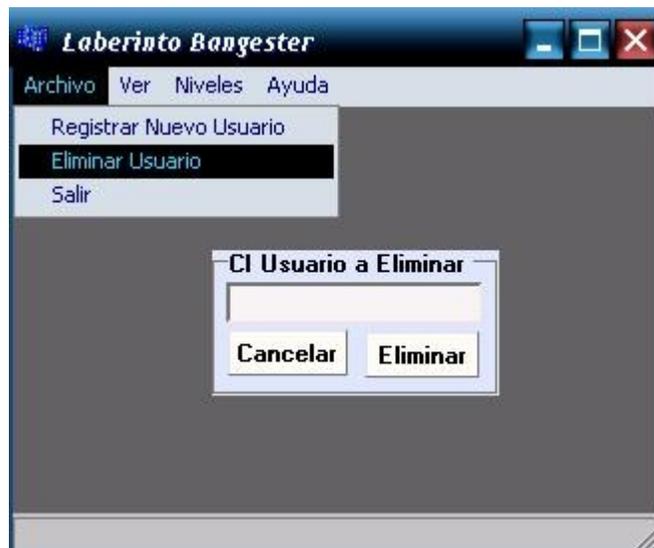
Recorrido del Estudiante



Nivel	Tiempo (min)
Nivel 1	14
Nivel 2	30
Nivel 3	31

Anexo 4.



Anexo 5.

Anexo 6.

	Nombre	Nivel	Cant. de Objetos
1	Eduardo Millan Nuñez	2	25
2	Gretel Garcia Hidalgo	3	40
3	Javier Lorie Guerra	2	24
4	Frank Duvernal Hernandez	1	10
5	Antonio Luis Perea	2	23
6	Leidys Barbara Hernandez	3	32
7	Leopoldina Hidalgo Perez	1	8
8	Dalia Maria Millan	1	8
9	Loraima Maning Aldana	1	5
10	Antonio Guerrero Prieto	2	26
11	Jose Vilato Frometa	3	35
12	Carlos Vallejo Vigoa	1	10

Apéndices:**Glosario de Términos:****A:**

Agudeza visual: Capacidad de ver detalles en los objetos. La agudeza visual no debería utilizarse para predecir el funcionamiento visual o educacional de un sujeto, puesto que un sujeto con bajo nivel de agudeza visual puede funcionar eficientemente ante una tarea que requiera su uso.

Ambliopía: Déficit funcional de la visión causada por un desarrollo anómalo secundario a una estimulación visual anormal, puede ser de moderada a severa, en dependencia del grado de afectación visual.

D:

Diagramas: Dibujo en el que se muestran las relaciones entre las diferentes partes de un conjunto o sistema.

G:

Gestión: Operación para resolver un problema determinado

I:

Interfaces: Superficie de contacto. Conexión entre dos sistemas independientes.

Iterativo: Indica repetición o reiteración

P:

Persistentes: Clases persistentes, de persistir, mantenerse constante.

R:

Requerimientos: Requisitos necesarios para dar solución a un problema determinado

S:

Soporte: De soportar. Apoyo o sostén.

Sistema: Conjunto de reglas o principios sobre una materia racionalmente enlazados entre sí.

Software: Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

Glosario de Abreviaturas:

XML: XML («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

TIC: Tecnología de la Informática y las Comunicaciones.

PDE: Proceso Educativo Docente.

UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad.

Índice de figuras y tablas

Índice de Figuras

INTRODUCCIÓN.....	8
.....	12
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	12
1.1 ALGUNOS FUNDAMENTOS ANÁTOMO - FISIOLÓGICOS Y ÓPTICOS DE LA VISIÓN.....	13
1.2 CONSIDERACIONES ACERCA DE LA CORRECCIÓN Y/O COMPENSACIÓN DE LA AMBLIOPIA	13
1.3 TECNOLOGÍA	15
1.4 ANÁLISIS DE INVESTIGACIONES REALIZADAS EN CENTROS DE REHABILITACIÓN VISUAL.....	18
1.5 MÉTODOS E INSTRUMENTOS EMPLEADOS.....	19
POSIBLES HERRAMIENTAS PARA DAR SOLUCIÓN A LA PROPUESTA DE SOFTWARE:.....	19
1.5.2 Plataforma de Desarrollo.....	19
1.5.2.1 Características de Visual Studio.....	20
1.5.3.1 Lenguaje de programación C++.....	21
1.6.1 SQL Server	24
1.7 Herramientas CASE (Computer Aided Software Engineering).....	27
<i>Las herramientas CASE son un conjunto de programas y ayudas que brindan ayuda y 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.</i>	
.....	27
1.7.1 Soluciones para la realización del diseño.....	27
1.7.1.1 El Proceso Unificado de Desarrollo de Software (RUP) como base en el desarrollo de la solución..	29
Figura 1. Fases e Iteraciones de la Metodología RUP.....	31
Figura 2. Metodología Extreme Programming.....	32
1.7.2 Lenguaje de Modelado.....	33
1.8 HERRAMIENTA DE MODELACIÓN VISUAL	35
1.8.1 Rational Rose Enterprise Edition	35
1.8.2 Visual Paradigm	35
1.9 SOLUCIÓN PROPUESTA.....	37
CONSIDERACIONES FINALES.....	38
CAPÍTULO 2: DEFINICIÓN DE REQUERIMIENTOS DEL SISTEMA.....	39
2.1 INTRODUCCIÓN:.....	39
Figura3. Posibles procesos.....	39
2.2 MODELO DEL DOMINIO	41
2.3 DEFINICIÓN DE CONCEPTOS DEL MODELO DEL DOMINIO:.....	41
2.3.1 Glosario de Términos del Dominio:.....	41
<i>Posteriormente se reconocen los actores del sistema a desarrollar y se conciben, a través de la agrupación de los requerimientos funcionales, los posibles resultados de valor que le pueda brindar a sus actores, o lo que es lo mismo, los casos de uso del sistema. Además, se seleccionan los casos de uso arquitectónicamente significativos para hacerles sus descripciones textuales en formato expandido.</i>	
.....	44
Captura de Requerimientos:.....	44
2.4 REQUERIMIENTOS FUNCIONALES:.....	44

2.6 MODELO DE CASOS DE USOS DEL SISTEMA.....	48
2.6.1 Descripción de los actores:.....	48
2.6.1.1 Casos de uso del sistema.....	49
2.6.2 Casos de uso del sistema:.....	51
Figura 5. Diagrama de Casos de Uso del Sistema.....	51
CONSIDERACIONES FINALES	61
CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA.....	62
3.1 INTRODUCCIÓN	62
3.2 DESCRIPCIÓN DE LA ARQUITECTURA	63
3.3 DIAGRAMAS DE CLASES DE ANÁLISIS.....	64
3.3.1 CU: <i>Gestionar Usuario</i>	64
Figura 7. Clase de Análisis: Gestionar Usuario.....	64
3.3.2 CU: <i>Autenticar Usuario</i>	65
Figura 8. Clase de Análisis: Autenticar Usuario.....	65
3.3.3 CU: <i>Conformar Laberinto</i>	65
Figura 9. Clase de Análisis: Conformar Laberinto.....	65
3.3.4 CU: <i>Obtener Objetos</i>	66
Figura 10. Clase de Análisis: Obtener Objetos.....	66
3.3.5 CU: <i>Ejecutar Pregunta Nivel</i>	66
Figura 11. Clase de Análisis: Ejecutar Pregunta Nivel.....	67
3.3.6 CU: <i>Revisar Resultados</i>	67
Figura 12. Clase de Análisis: Revisar Resultados.....	67
3.4 DIAGRAMAS DE CLASES DE DISEÑO.....	68
Figura 13. Clase de Diseño: Gestionar Usuario.....	68
3.4.2 CU: <i>Autenticar Usuario</i>	69
Figura 14. Clase de Diseño: Autenticar Usuario.....	69
3.4.3 CU: <i>Revisar Resultados</i>	70
Figura 15. Clase de Diseño: Revisar Resultados.....	70
3.4.4 CU: <i>Conformar Laberinto</i>	71
Figura 16. Clase de Diseño: Conformar Laberinto.....	71
3.4.5 CU: <i>Obtener Objetos</i>	72
Figura 17. Clase de Diseño: Obtener Objetos.....	72
3.4.6 CU: <i>Ejecutar Pregunta Nivel</i>	73
Figura 18. Clase de Diseño: Ejecutar Pregunta Nivel.....	73
3.5 DIAGRAMAS DE SECUENCIA:.....	75
Figura 19. Clase de Secuencia: Gestionar Usuario: Insertar Usuario.....	77
Figura 20. Clase de Secuencia: Gestionar Usuario: Modificar Usuario.....	78
3.5.3 CU: <i>Gestionar Usuario: Eliminar Usuario</i>	79
Figura 21. Clase de Secuencia: Gestionar Usuario: Eliminar Usuario.....	79
3.5.4 CU: <i>Autenticar Usuario</i>	80
Figura 22. Clase de Secuencia: Autenticar Usuario.....	80
3.5.5 CU: <i>Ejecutar Pregunta Nivel</i>	81
Figura 23. Clase de Secuencia: Ejecutar Pregunta Nivel.....	81
3.5.6 CU: <i>Revisar Resultados</i>	82
Figura 24. Clase de Secuencia: Revisar Resultados.....	82
3.5.7 CU: <i>Conformar Laberinto</i>	83
Figura 25. Clase de Secuencia: Conformar Laberinto.....	83
3.5.8 CU: <i>Obtener Objetos</i>	84

.....	84
Figura 26. Clase de Secuencia: Obtener Objetos.....	84
3.6 DIAGRAMA DE CLASES PERSISTENTES.....	85
Figura 27. Diagrama de Clases Persistentes.....	85
Figura 28. Modelo de Datos.....	88
CONSIDERACIONES FINALES.....	88
CONCLUSIONES:.....	89
RECOMENDACIONES.....	90
AL CONCLUIR ESTE TRABAJO SE RECOMIENDA LO SIGUIENTE:.....	90
PROFUNDIZAR EN EL ESTUDIO Y CREACIÓN DE SOFTWARE PARA LA ESTIMULACIÓN VISUAL EN NIÑOS ESTRÁBICOS Y/O AMBLOPES.	90
REALIZAR LA IMPLEMENTACIÓN DEL SOFTWARE BASADO EN EL ANÁLISIS Y DISEÑO PROPUESTO EN ESTE TRABAJO E IMPLANTARLO PAULATINAMENTE EN TODAS LAS ESCUELAS DE ATENCIÓN A DÉBILES VISUALES SEGÚN LAS CONDICIONES EXISTENTES EN NUESTRO PAÍS.....	90
.....	91
ANEXOS:.....	94
ANEXO 2.....	95
ANEXO 3.....	96
ANEXO 4.....	97
ANEXO 5.....	98
ANEXO 6.....	99
APÉNDICES:.....	100
ÍNDICE DE FIGURAS Y TABLAS.....	102

Índice de Tablas

<u>INTRODUCCIÓN.....</u>	<u>8</u>
.....	<u>12</u>
<u>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....</u>	<u>12</u>
<u>1.1 ALGUNOS FUNDAMENTOS ANÁTOMO - FISIOLÓGICOS Y ÓPTICOS DE LA VISIÓN.....</u>	<u>13</u>
<u>1.2 CONSIDERACIONES ACERCA DE LA CORRECCIÓN Y/O COMPENSACIÓN DE LA AMBLOPIA</u>	<u>13</u>
<u>1.3 TECNOLOGÍA</u>	<u>15</u>
<u>1.4 ANÁLISIS DE INVESTIGACIONES REALIZADAS EN CENTROS DE REHABILITACIÓN VISUAL.....</u>	<u>18</u>
<u>1.5 MÉTODOS E INSTRUMENTOS EMPLEADOS.....</u>	<u>19</u>
<u>POSIBLES HERRAMIENTAS PARA DAR SOLUCIÓN A LA PROPUESTA DE SOFTWARE:.....</u>	<u>19</u>
<u>1.5.2 Plataforma de Desarrollo.....</u>	<u>19</u>
<u>1.5.2.1 Características de Visual Studio.....</u>	<u>20</u>
<u>1.5.3.1 Lenguaje de programación C++.....</u>	<u>21</u>
<u>1.6.1 SQL Server</u>	<u>24</u>

<u>1.7 Herramientas CASE (Computer Aided Software Engineering).....</u>	<u>27</u>
<u>Las herramientas CASE son un conjunto de programas y ayudas que brindan ayuda y 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.....</u>	<u>27</u>
<u>1.7.1 Soluciones para la realización del diseño.....</u>	<u>27</u>
<u>1.7.1.1 El Proceso Unificado de Desarrollo de Software (RUP) como base en el desarrollo de la solución.....</u>	<u>29</u>
<u> Figura 1. Fases e Iteraciones de la Metodología RUP.....</u>	<u>31</u>
<u> Figura 2. Metodología Extreme Programming.....</u>	<u>32</u>
<u>1.7.2 Lenguaje de Modelado.....</u>	<u>33</u>
<u>1.8 HERRAMIENTA DE MODELACIÓN VISUAL</u>	<u>35</u>
<u>1.8.1 Rational Rose Enterprise Edition</u>	<u>35</u>
<u>1.8.2 Visual Paradigm</u>	<u>35</u>
<u>1.9 SOLUCIÓN PROPUESTA.....</u>	<u>37</u>
<u>CONSIDERACIONES FINALES.....</u>	<u>38</u>
<u>CAPÍTULO 2: DEFINICIÓN DE REQUERIMIENTOS DEL SISTEMA.....</u>	<u>39</u>
<u>2.1 INTRODUCCIÓN:.....</u>	<u>39</u>
<u> Figura3. Posibles procesos.....</u>	<u>39</u>
<u>2.2 MODELO DEL DOMINIO</u>	<u>41</u>
<u>2.3 DEFINICIÓN DE CONCEPTOS DEL MODELO DEL DOMINIO:.....</u>	<u>41</u>
<u>2.3.1 Glosario de Términos del Dominio:.....</u>	<u>41</u>
<u> Posteriormente se reconocen los actores del sistema a desarrollar y se conciben, a través de la agrupación de los requerimientos funcionales, los posibles resultados de valor que le pueda brindar a sus actores, o lo que es lo mismo, los casos de uso del sistema. Además, se seleccionan los casos de uso arquitectónicamente significativos para hacerles sus descripciones textuales en formato expandido.....</u>	<u>44</u>
<u> Captura de Requerimientos:.....</u>	<u>44</u>
<u>2.4 REQUERIMIENTOS FUNCIONALES:.....</u>	<u>44</u>
<u>2.6 MODELO DE CASOS DE USOS DEL SISTEMA.....</u>	<u>48</u>
<u>2.6.1 Descripción de los actores:.....</u>	<u>48</u>
<u> Tabla 1: Actor Del Sistema: Estudiante.....</u>	<u>48</u>
<u> Tabla 2: Actor del Sistema: Maestro.....</u>	<u>48</u>
<u> Tabla 3: Actor del Sistema: Usuario.....</u>	<u>49</u>
<u>2.6.1.1 Casos de uso del sistema.....</u>	<u>49</u>
<u> Tabla 4: Caso de Uso del Sistema: Autenticar Usuario.....</u>	<u>49</u>
<u> Tabla 5: Caso de Uso del Sistema: Gestionar Usuario.....</u>	<u>49</u>
<u> Tabla 6: Caso de Uso del Sistema: Revisar Resultados.....</u>	<u>50</u>
<u> Tabla 7: Caso de Uso del Sistema.....</u>	<u>50</u>
<u> Tabla 8: Caso de Uso del Sistema: Obtener Objetos.....</u>	<u>50</u>
<u> Tabla 9: Caso de Uso del Sistema: Ejecutar Pregunta de Nivel.....</u>	<u>50</u>
<u>2.6.2 Casos de uso del sistema:.....</u>	<u>51</u>
<u> Figura 5. Diagrama de Casos de Uso del Sistema.....</u>	<u>51</u>

Tabla 10: Descripción Caso de Uso del Sistema: Autenticar Usuario.....	53
Tabla11: Descripción Caso de Uso del Sistema: Gestionar Usuario.....	54
Tabla12: Descripción Caso de Uso del Sistema: Revisar Resultados.....	57
Tabla13: Descripción Caso de Uso del Sistema: Conformar Laberinto.....	57
Tabla14: Descripción Caso de Uso del Sistema: Obtener Objetos.....	58
Tabla15: Descripción Caso de Uso del Sistema: Ejecutar Pregunta de Nivel.....	60
CONSIDERACIONES FINALES	61

CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA.....	62
3.1 INTRODUCCIÓN	62
3.2 DESCRIPCIÓN DE LA ARQUITECTURA	63
3.3 DIAGRAMAS DE CLASES DE ANÁLISIS.....	64
3.3.1 CU: Gestionar Usuario.....	64
Figura 7. Clase de Análisis: Gestionar Usuario.....	64
3.3.2 CU: Autenticar Usuario	65
Figura 8. Clase de Análisis: Autenticar Usuario.....	65
3.3.3 CU: Conformar Laberinto.....	65
Figura 9. Clase de Análisis: Conformar Laberinto.....	65
3.3.4 CU: Obtener Objetos.....	66
Figura 10. Clase de Análisis: Obtener Objetos.....	66
3.3.5 CU: Ejecutar Pregunta Nivel.....	66
Figura 11. Clase de Análisis: Ejecutar Pregunta Nivel.....	67
3.3.6 CU: Revisar Resultados.....	67
Figura 12. Clase de Análisis: Revisar Resultados.....	67
3.4 DIAGRAMAS DE CLASES DE DISEÑO.....	68
Figura 13. Clase de Diseño: Gestionar Usuario.....	68
3.4.2 CU: Autenticar Usuario.....	69
Figura 14. Clase de Diseño: Autenticar Usuario.....	69
3.4.3 CU: Revisar Resultados.....	70
Figura 15. Clase de Diseño: Revisar Resultados.....	70
3.4.4 CU: Conformar Laberinto.....	71
Figura 16. Clase de Diseño: Conformar Laberinto.....	71
3.4.5 CU: Obtener Objetos.....	72
Figura 17. Clase de Diseño: Obtener Objetos.....	72
3.4.6 CU: Ejecutar Pregunta Nivel.....	73
Figura 18. Clase de Diseño: Ejecutar Pregunta Nivel.....	73
3.5 DIAGRAMAS DE SECUENCIA:.....	75
Figura 19. Clase de Secuencia: Gestionar Usuario: Insertar Usuario.....	77
Figura 20. Clase de Secuencia: Gestionar Usuario: Modificar Usuario.....	78
3.5.3 CU: Gestionar Usuario: Eliminar Usuario	79
Figura 21. Clase de Secuencia: Gestionar Usuario: Eliminar Usuario.....	79
3.5.4 CU: Autenticar Usuario.....	80

.....	80
<u>Figura 22. Clase de Secuencia: Autenticar Usuario.....</u>	80
<u>3.5.5 CU: Ejecutar Pregunta Nivel.....</u>	81
<u>Figura 23. Clase de Secuencia: Ejecutar Pregunta Nivel.....</u>	81
<u>3.5.6 CU: Revisar Resultados.....</u>	82
<u>Figura 24. Clase de Secuencia: Revisar Resultados.....</u>	82
<u>3.5.7 CU: Conformar Laberinto.....</u>	83
<u>Figura 25. Clase de Secuencia: Conformar Laberinto.....</u>	83
<u>3.5.8 CU: Obtener Objetos.....</u>	84
.....	84
<u>Figura 26. Clase de Secuencia: Obtener Objetos.....</u>	84
<u>3.6 DIAGRAMA DE CLASES PERSISTENTES.....</u>	85
<u>Figura 27. Diagrama de Clases Persistentes.....</u>	85
<u>Figura 28. Modelo de Datos.....</u>	88
<u>CONSIDERACIONES FINALES.....</u>	88
<u>CONCLUSIONES:.....</u>	89
<u>RECOMENDACIONES.....</u>	90
<u>AL CONCLUIR ESTE TRABAJO SE RECOMIENDA LO SIGUIENTE:.....</u>	90
<u>PROFUNDIZAR EN EL ESTUDIO Y CREACIÓN DE SOFTWARE PARA LA ESTIMULACIÓN VISUAL EN NIÑOS ESTRÁBICOS Y/O</u>	
<u>AMBLIOPES.....</u>	90
<u>REALIZAR LA IMPLEMENTACIÓN DEL SOFTWARE BASADO EN EL ANÁLISIS Y DISEÑO PROPUESTO EN ESTE TRABAJO E</u>	
<u>IMPLANTARLO PAULATINAMENTE EN TODAS LAS ESCUELAS DE ATENCIÓN A DÉBILES VISUALES SEGÚN LAS CONDICIONES</u>	
<u>EXISTENTES EN NUESTRO PAÍS.....</u>	90
.....	91
<u>ANEXOS:.....</u>	94
<u>ANEXO 2.....</u>	95
<u>ANEXO 3.....</u>	96
<u>ANEXO 4.....</u>	97
<u>ANEXO 5.....</u>	98
<u>ANEXO 6.....</u>	99
<u>APÉNDICES:.....</u>	100
<u>ÍNDICE DE FIGURAS Y TABLAS.....</u>	102