



Pontificia Universidad Católica de Valparaíso

Facultad de Ingeniería

Escuela de Ingeniería Informática

Ingeniería de Ejecución en Informática

**SISTEMA DE APOYO A LA EDUCACIÓN
MATEMÁTICA PARA NIÑOS INVIDENTES Y CON
BAJA VISIÓN**

Autores:

Juan Ignacio Horta Lucero - Miguel Ángel Quilape Tranamil

Informe final del Proyecto para optar al Título profesional de

Ingeniero de Ejecución en Informática

Profesor guía:

Silvana Roncagliolo de la Horra

DICIEMBRE 2007

*“Dedico este trabajo a mi familia y personas cercanas que me apoyaron
durante el largo proceso de aprendizaje en la Universidad”*

Juan Ignacio Horta Lucero

“Dedico este trabajo a mi familia, mi novia Alba, a la memoria de mi madre y sobrina, así como a todas las personas que Dios puso en mi camino, quienes me apoyaron siempre y compartieron mi sueño.”

Miguel Ángel Quilape Tranamil

Agradecimientos

Agradecemos a las personas de los lugares que visitamos en busca de información, quienes abrieron las puertas a un mundo desconocido hasta entonces.

A la Escuela Santa Lucía que tomó el papel de cliente y apoyó el desarrollo del proyecto y por supuesto, a los profesores de la Universidad quienes nos guiaron en el desarrollo.

Juan Ignacio Horta Lucero

Miguel Ángel Quilape Tranamil

Resumen

El presente proyecto informático, titulado “Sistema de apoyo a la educación matemática para niños invidentes y con baja visión”, tiene como finalidad contribuir a la enseñanza, ejercitación y evaluación de las matemáticas a niños no videntes y de baja visión, pertenecientes a cursos de 1° a 4° año básico. Para conseguir esto se realizan procesos como: Asignar ejercicios de acuerdo al avance particular de cada niño; permitir al profesor modificar la lista de ejercicios disponibles en una actividad y obtener resultados de rendimiento histórico mediante reportes y estadísticas.

Abstract

The current project is called "System to support math education for children with low vision and blind", the goal of this project is contribute to the education, exercise and evaluation of mathematics of this special kids, this project is for children between 1st and 4th elementary levels. To achieve this purpose, we use different techniques, such as: Assign exercises according to the progress of each child; allow the teacher to modify the exercises list available on each activity, and with this we will get the performance results through reports and statistics.

1. Introducción

La Organización Mundial de la Salud (OMS) prevé que en el año 2020 el número de invidentes habrá aumentado de 45 a 75 millones y el número de discapacitados visuales habrá crecido de 135 a 200 millones. Cada año la cifra se eleva en uno o dos millones de personas debido, en parte, al aumento de la esperanza de vida [Resnikoff, 2003].

No obstante esta creciente cifra, aun existen muchas personas desinformadas respecto al tema, que tienden a excluir de la sociedad a las personas ciegas o que padecen de baja visión, esto por pensar que sus limitaciones son muchas más de las que realmente existen. En la vida real, las limitaciones de las personas invidentes se pueden reducir a tres aspectos: desplazamiento, control del ambiente y adquisición de cultura.

El presente proyecto, se enfoca en el tercer aspecto, “la adquisición de cultura”, limitación que se refiere a la forma en que una persona ciega logra captar toda la información visual que genera nuestra sociedad, en este caso, dicha información corresponderá al área de educación matemática.

Para lograr satisfacer esta necesidad de obtener información, es de gran utilidad la interacción de una persona no vidente con un PC normal. Actualmente, existen un número significativo de tecnologías que permiten esto, dentro de las cuales están: los lectores de pantalla, sintetizadores de voz, dispositivos Braille (que convierten un texto en códigos táctiles), reconocedores ópticos de caracteres (OCR) y simples scanners.

Para lograr comprender a la perfección un ámbito como el de la educación de niños invidentes y con baja visión, es que se decidió acudir a la Escuela Santa Lucía.

La enseñanza de estos niños, se realiza cumpliendo los objetivos fundamentales propuestos por el currículo nacional para sus distintos niveles educacionales. Para facilitar alcanzar estos objetivos, es que los colegios ocupan softwares educativos como un instrumento de apoyo y medición de los avances realizados por los alumnos.

En general, los softwares ocupados por niños con discapacidad visual, se caracterizan por la reproducción en audio de todas las actividades que estos proporcionan, debido a esto, es que se comete el error de calificar cualquier programa que cumpla con esta característica como un “software educativo para no videntes”; obviando otras importantes características, tales como la adaptación del niño con los comandos básicos del programa o el tipo de ejercicios más apropiados para ellos, que en este caso sería ocupar el sonido para guiarlos a través de toda la actividad y no como una simple introducción a éstas.

Otro punto a considerar, es la escasez de softwares orientados a niños no videntes, situación que influye considerablemente en el hecho de que aquellos pocos que existen en la actualidad, resulten monótonos para quienes los utilizan diariamente; como consecuencia de esto, se evidencia un escaso interés en el área educacional para la cual estos softwares fueron desarrollados.

La falta de medición, es otra gran falencia que poseen muchos de estos softwares, ya que resulta de suma importancia para este tipo de programas educacionales, lograr almacenar los resultados de cada niño, y de esta manera, conseguir concentrarse en sus debilidades para así corregirlas.

En este contexto, no existen herramientas que se enfoquen a todos estos puntos por si solas, sino que atacan alguno de ellos de forma específica, y sin la posibilidad de integración.

Por este motivo sería de mucha utilidad, desarrollar alguna herramienta que pueda satisfacer las necesidades anteriormente expuestas: un software educativo que permita asignar adecuadamente el tipo de ejercicio, esto de acuerdo a los resultados generados por el niño, modificar la lista de ejercicios disponibles en una actividad y verificar estadísticas referentes a las debilidades y fortalezas de un niño en particular, todo esto enfocado a niños no videntes y con baja visión.

2. Objetivos del Proyecto

2.1. Objetivo General

Contribuir en la enseñanza, ejercitación y evaluación de las matemáticas a niños no videntes y de baja visión, pertenecientes a cursos de 1° a 4° año básico, mediante el desarrollo de un software de interfaz audible y gráfica apropiada, que asigne ejercicios de acuerdo al avance particular de cada niño, obteniendo resultados de rendimiento histórico mediante reportes y estadísticas. Además de permitir al profesor gestionar tales ejercicios y los parámetros asociados.

2.2. Objetivos Específicos

Para poder cumplir con el objetivo general del proyecto se definieron los siguientes objetivos específicos:

- Investigar sobre los lenguajes, herramientas y metodología más adecuados, considerando las características del proyecto en cuestión.
- Diseñar ejercicios interactivos que complementen los aprendizajes establecidos en las bases curriculares.
- Diseñar e implementar un motor de reglas adecuado, el cual decidirá el nivel de ejercicio que realizará cada niño, de acuerdo a su desempeño anterior.
- Validar el software en la Escuela Santa Lucía.

3. Estado del Arte

3.1. La ceguera en la historia

Según los antecedentes históricos recopilados, comienza la educación a los invidentes en el año 1745, en ideas planteadas por Diderot, filósofo y educador francés. Casi 30 años después, el francés Valentín Haüy funda en París una institución para invidentes concebida como un centro educativo. En 1820, en el mismo París, Louis Braille diseñó un método compuesto por 63 signos, resultantes de la combinación de seis puntos en relieve, que permitía simplificar la lectura y hacía posible la escritura, lo que abrió mejores posibilidades a los invidentes de todo el mundo.

A lo largo del siglo XIX surgieron escuelas especiales dirigidas a la educación de los invidentes. En ellas se recluía al niño invidente, separándolo de su familia y alejándolo de la vida cotidiana sin ninguna preparación para afrontar una vida normal e incapacitándoles para insertarse en la comunidad.

En el siglo XX, después de la primera guerra mundial, se fundan en muchos países centros de rehabilitación para personas que quedaron ciegas a causa de la guerra. Dentro de las fundaciones cabe destacar la importante fundación española de la Organización Nacional de Ciegos (ONCE), creada en tiempos de Franco, destinada a resolver el problema de quienes perdieron la vista a causa de la guerra civil.

Actualmente, la ONCE es un organismo único en el mundo. Siendo una de las diez empresas más importantes de España, tiene una destacada participación en los sectores de la prensa, bienes raíces y bancos.

En Chile, en el año 1882, se crea la escuela para sordomudos y en abril de 1920, por un decreto especial, se crea la sección para invidente, primera escuela de invidentes del país; ésta se separaría de la anterior en el año 1951. Para este centro se trajeron desde Alemania técnicas de enseñanza para la cultura educacional del invidente.

En la Universidad de Chile, en el año 1955, a instancias de la fundación norteamericana "Foundation Over Seas Blind", se crea el centro de formación de profesores de educación especial, con personal especializado en educación y rehabilitación de personas ciegas.

La Unesco tiene un importante papel en esta materia, pues propicia en países en vías de desarrollo amplios programas de educación para minusválidos, facilitando cursos de entrenamiento y becas para profesores. Con fondos proporcionados por los países miembros, la Unesco ha financiado proyectos de investigación para dotar a los invidentes del más moderno y eficiente material.

3.1.1. Ley N° 19.284 de Integración Social de las Personas con Discapacidad.

La Ley N° 19.284, publicada en el Diario Oficial el 14 de Enero de 1994 y promulgada el 5 de Enero de ese mismo año, establece las normas que permiten la plena integración de las personas con discapacidad en la sociedad. En su artículo N° 20 del capítulo N° 1, establece que "Las Bibliotecas de acceso público, deberán contar gradualmente con espacio físico, recursos informativos y brindar facilidades destinadas a no videntes".

La mencionada ley estipula que es el Estado el responsable de financiar programas para las personas discapacitadas, y "deberá promover la capacitación laboral creando programas especiales con el fin de permitir e incrementar su inserción en el trabajo. Igualmente, velará para que los programas de capacitación se lleven a cabo de acuerdo a las necesidades de las personas y a los requerimientos y posibilidades del mercado. Además, tendrá que velar por la inserción laboral de las personas con discapacidad a objeto de asegurar su independencia, desarrollo personal, ejercicio del derecho a constituir una familia y a gozar de una vida digna.

Además la ley reglamenta la importación de ayudas técnicas con exenciones arancelarias, a través de un sistema de reintegro del dinero que se cancela en la aduana. A este beneficio pueden acceder únicamente dichas personas o personas jurídicas sin fines de lucro. También, establece la creación del Fondo Nacional de la Discapacidad (FONADIS), de carácter autónomo y que tiene por misión administrar los recursos que le confieran en favor de las personas con discapacidad". Se relaciona con el Estado a través del Ministerio de Planificación y Cooperación (MIDEPLAN).

3.1.2. Fondo Nacional de la Discapacidad (FONADIS).

El Fondo fomenta la capacitación laboral y la integración social a través del financiamiento de proyectos de rehabilitación para personas con discapacidad, y habilita unidades con los equipamientos que permite la Ley 19.284 (art. 56, letra b), que sean relevantes en el proceso de rehabilitación integral. También ofrece cursos entregados por diversas entidades capacitadoras, las que realizan actividades y ayudas técnicas que apoyen a la integración social de más de un millón de personas en Chile.

3.1.3. Consejo Nacional del Libro y la Lectura.

También existe el Fondo Nacional del Fomento del Libro y la Lectura, constituyendo un incentivo esencial para el financiamiento de los proyectos que sean presentados en el ámbito de la discapacidad visual, como incentivo a la lectura y capacitación en el uso de los recursos bibliotecarios.

3.2. Descripción de la organización y situación actual

Hoy en día, en las mayorías de los establecimientos educacionales se está haciendo el esfuerzo de integrar las Tecnologías de Información y Comunicación (TIC) a la educación tradicional, lo que ha dado muy buenos resultados a quienes se atreven a hacer uso de estas herramientas. Dentro de este concepto es que se encuentran los distintos software educativos utilizados en distintos establecimientos educacionales del país y en específico la Escuela Santa Lucía, la que hace uso de distintos software para complementar la educación e integración de los niños invidentes y de baja visión que conforman su alumnado. Por esta razón es que la Escuela Santa Lucía cuenta con dos laboratorios de computación, en donde cada PC cuenta con parlantes, obviamente esenciales, para la interacción de los niños con los distintos software educativos que se utilizan.

Entre los software educativos utilizados se encuentran Abrapalabra, La granja de Theo y Seth, Cantalettras, El toque mágico, Sebran, los cuales se describirán en detalle posteriormente en este documento.

Primeramente, el profesor debe seleccionar uno de los distintos software, de acuerdo al nivel de aprendizaje en que se encuentra el niño en particular y a su condición visual. Luego de eso,

el niño interactúa con el software escogido, el cual no siempre entrega algún resultado significativo o estadístico al profesor, teniendo éste que dedicarse a la observación de cada niño por separado para poder obtener una evaluación en su desempeño.

Como se puede observar, existen bastantes falencias en el sistema actual, siendo la primera, la poca estandarización de actividades dentro de los mismo software, las cuales deberían servir de igual manera a niños invidentes y de baja visión.

Ya cuando el niño se encuentra realizando una actividad no es para nada conveniente que no se obtenga ningún resultado que permita reflejar su desempeño o capacidad frente a la actividad, para lo cual el profesor debe estar evaluando presencialmente a cada niño, desperdiciando tiempo valioso que podría ser mejor utilizado. Además, al no obtenerse datos sobre las actividades desempeñadas, menos se puede hablar de almacenarlos, por lo cual no queda registro de ningún tipo, ya sea de actividad realizada, nivel alcanzado, aciertos, fallos o tiempos. Datos que serían muy importantes para la realización de estadísticas grupales o individuales de los niños.

Otro factor no menos importante y que está implícito, es lo atractivo que puedan resultar las actividades para los niños, no se debe olvidar que a ellos le gustan los desafíos y es lo que más los motiva a aprender, para los cual sería aconsejable que cada niño siguiera una escala de exigencias de acuerdo a su desempeño individual, pudiendo subir el nivel de exigencia a medida que más avanza en su aprendizaje individual. Todo lo anteriormente mencionado se complementa con una interfaz atractiva y estimulante, ya sea visualmente para los niños de baja visión y auditiva para los invidentes.

3.3. Software Educativos para niños invidentes y con baja visión

En general, los software ocupados por niños con discapacidad visual, se caracterizan por la reproducción en audio de todas las actividades que estos proporcionan, debido a esto, es que se comete el error de calificar cualquier programa que cumpla con esta característica como un “software educativo para no videntes”; obviando otras importantes características, tales como la adaptación del niño con los comandos básicos del programa o el tipo de ejercicios más

apropiados para ellos, que en este caso sería ocupar el sonido para guiarlos a través de toda la actividad y no como una simple introducción a estas.

La escasez de software orientados a niños no videntes, influye considerablemente en el hecho de que aquellos pocos que existen en la actualidad, resulten monótonos para quienes los utilizan diariamente; como consecuencia de esto, se evidencia un escaso interés en el área educacional para la cual estos software fueron desarrollados.

La falta de medición, es otra gran falencia que poseen muchos de estos software, ya que resulta de suma importancia para este tipo de programas educacionales, lograr almacenar los resultados de cada niño y proporcionarles a los docentes la información de una forma clara y actualizada.

Además de esto, los software presentan una asignación de ejercicios de forma lineal o aleatorio, lo que no resulta de utilidad para medir los avances de cada niños. En el caso de asignación lineal, al iniciar nuevamente una actividad, el niño deberá comenzar desde el nivel más bajo para así ir incrementando la dificultad; no se guarda el avance del niño.

Entre los software educacionales utilizados por la “Escuela Santa Lucia” destacan los que a continuación se describen.

3.3.1. Abrapalabra: “La magia de aprender”

Software multimedial e interactivo para la enseñanza de la lectura en lengua castellana. Está contenido en 2 CD-ROMs con 160 ejercicios agrupados en 70 unidades de aprendizaje. Los ejercicios que incluye apuntan al desarrollo de habilidades relacionadas con la adquisición del proceso lector, comenzando por el apresto, pasando por la lectura inicial y concluyendo en la comprensión lectora.

Con el fin de motivar al niño, el software presenta dos personajes que lo guiarán durante todo su recorrido; ellos son el Mago Nombrón y el Conejo Cuentín, quienes irán viviendo aventuras en diversos escenarios que servirán para contextualizar cada uno de los ejercicios y unidades de aprendizaje. Una serie de atractivos personajes de fantasía también estarán presentes (animales, princesas, condes, piratas, fantasmas y otros) para motivar al niño a entrar en cada ambiente de trabajo y favorecer así su aprendizaje y retención.

El programa sugiere una modalidad de uso secuencial, debido a la presentación de sus contenidos de acuerdo a su nivel de dificultad. Es así como las 6 primeras unidades incluyen ejercicios de apresto y reconocimiento de vocales; las siguientes 60 unidades tratan el reconocimiento de consonantes, la adquisición del proceso lector inicial y la comprensión lectora, las dos unidades siguientes tratan los grupos consonánticos y una última unidad presenta textos con historias interactivas. Incluye una completa guía para el adulto con información pedagógica y referencias para la instalación y navegación del programa. El software permite guardar registros de los ejercicios ya recorridos por un niño y guardar el estado de avance, de hasta 40 niños en forma simultánea.

Apela a la estimulación visual y auditiva del usuario. Cada ejercicio incluye instrucciones orales, la posibilidad de repetirlo y retroalimentación inmediata. Ofrece un mapa general de los contenidos, desde el cual es posible entrar a cualquier ejercicio (navegación libre, en lugar de secuenciada), abrir la guía para el adulto y juegos imprimibles.

3.3.2. La granja de Theo y Seth

La granja de Theo y Seth, es un software que permite a los aprendices integrar y desarrollar contenidos específicos del currículo de NB1, primero y segundo año básico; referidos a antecesor y sucesor, ordinalidad, adición con y sin reserva y sustracción con y sin canje.

El juego está protagonizado por Theo y Seth, dos extraterrestres que interactúan con el niño y son los que ejecutan las acciones en el juego. El software está compuesto por dos módulos y estos se dividen en distintos niveles de dificultad.

Objetivos La granja de Theo y Seth:

- Asimilar los conceptos de antecesor y sucesor para aplicarlo en situaciones de la vida cotidiana.
- Reforzar el concepto de número.
- Reconocer numerales escritos y representarlos con material concreto, y además en macro tipo o pizarra Braille.

- Generalizar a través de ejercicios, los contenidos matemáticos de adición y sustracción con y sin reserva y canje respectivamente.

Usuarios:

- Niños invidentes o con visión residual entre 6 y 14 años.
- Educadores.
- Padres y apoderados videntes o con baja visión que participen como mediadores.

Contenidos cognitivos:

- Antes – después.
- De cantidad.
- Valor posicional numérico.
- Formación de números hasta de tres dígitos.

Requerimientos Técnicos Mínimos:

- Computador Procesador Pentium III.
- 128MB RAM de memoria.
- 300MB espacio de disco duro.
- Tarjeta de sonido.
- Parlantes.
- Teclado o Mouse.

Interfaz física:

La interacción con el juego es a través de interfaz física; teclado. Las teclas involucradas en la interacción del juego son:

- Flecha derecha e izquierda: Estas se usan para desplazarse.

- Barra espaciadora: Con esta barra el niño acciona las palancas y elige algunas opciones.
- TAB: Se usa para recorrer los módulos y niveles.
- Flecha abajo: Vuelve a la cocina para elegir sub-módulo.
- ESC: sale del juego.
- Teclado numérico: Se utiliza para la formación de numerales y resolución de operatoria.

3.3.3. Cantaletas

Es un sistema multimedial para la enseñanza de la lectura y escritura de niños invidentes cuya característica fundamental es que se sustenta en un modelo de triple mediación: para el niño, para el educador lego y para el educador experto. Para el niño, el sistema media los aprendizajes en una gradiente de complejidad progresiva. Para el educador lego, el sistema brinda permanentes consejos instruccionales que apoyan el proceso de enseñanza y enseña los rudimentos de la lecto-escritura Braille. Para el mediador experto, el sistema ofrece un detallado y exhaustivo registro diagnóstico de los logros y la progresión del aprendizaje de los niños. El sistema tiene una filosofía de inclusión, siendo su diseño atractivo para el trabajo conjunto de invidentes y videntes.

Todas las actividades están diseñadas en forma de juegos y una de sus características importantes es que permite al niño invidente la ejercitación de la escritura Braille a través de las modalidades Cajetín y máquina Perkins. Asimismo, las actividades se enfocan al conocimiento de las letras en tinta y a la ubicación espacial de las letras en el teclado del computador.

Objetivo: Permitir al niño invidente traducir a sus modalidades sensoriales intactas (tacto y oído), la experiencia lecto-escritora del vidente, apoyando la enseñanza de lectura y escritura inicial, a través de actividades lúdicas que permiten la enseñanza del sistema Braille y el uso del teclado.

- Orientado a: Niños no videntes y videntes.
- Edad: Niños entre 4 y 10 años.

- Áreas que abarca: Desde apresto lector hasta lectura de textos breves.

3.3.4. El Toque Mágico

Es un sistema multimedial diseñado para estimular el desarrollo de importantes nociones de apresto escolar en niños invidentes como son: el lenguaje, los números, la orientación espacial y la orientación temporal. El programa presenta atractivas actividades en un ambiente motivador, con interfaz auditiva, imágenes y características interactivas, por lo que también puede ser usado por niños videntes, lo cual lo constituye en un medio de integración entre alumnos con y sin trastornos visuales.

Al igual que el programa Cantaletas, el Toque Mágico se sustenta en un modelo de triple mediación: para el niño, para el educador lego y para el educador experto. Para el niño, el sistema media los aprendizajes en una gradiente de complejidad progresiva. Para el educador lego, el sistema brinda permanentes consejos instruccionales que apoyan el proceso de enseñanza. Para el mediador experto, el sistema ofrece un detallado y exhaustivo registro diagnóstico de los logros y la progresión del aprendizaje de los niños.

El niño necesita del educador para iniciar el recorrido por El Toque Mágico, pero una vez que ha entendido el modo de interacción y se ha ubicado en la actividad, la permanencia del educador es optativa, si se cuenta con una pantalla táctil.

- Objetivo: Apoyar el proceso de apresto escolar, estimulando en el niño invidente el desarrollo de nociones relevantes en la educación preescolar.
- Orientado a: Niños no videntes y videntes.
- Edad: Niños entre 3 y 6 años.
- Áreas que abarca: Lenguaje, Numérica, Orientación Espacial y Orientación.

4. Metodología

4.1. Aprendizaje de las Matemáticas

La adquisición del conocimiento y la comprensión de las matemáticas que requiere la vida cotidiana son una responsabilidad central de la escuela. El desarrollo creciente de la tecnología, la modernización de la industria y las comunicaciones y, en general, la influencia de la economía en la vida social plantea nuevos y grandes desafíos a este sector. Si bien los computadores ejecutan ciertas tareas rutinarias - a menudo mejor que el ser humano - los puestos de trabajo requieren hoy conductas reflexivas a cuyo desarrollo puede contribuir un conocimiento y una práctica adecuada de las matemáticas; esto, sin considerar que una parte significativa del desempeño laboral exige una formación matemática avanzada.

En las escuelas básicas, la enseñanza de las matemáticas está orientada en dos direcciones complementarias e inseparables. Por una parte, ofrecer a todos los estudiantes la opción de ampliar y profundizar los estudios que son propios de este sector, sin perder de vista el papel que las matemáticas desempeñan en la comprensión de aprendizajes propios de otros sectores. En este sentido, adquiere especial relevancia la dimensión formativa del sector, promoviendo el desarrollo del pensamiento lógico, del análisis, de la deducción, de la precisión, de la capacidad de construir y resolver problemas a partir de la realidad y de formular y comprender modelos de tipo matemático.

Por otra parte, la enseñanza debe contribuir a un mejor desempeño de las personas en la vida diaria, a través de la utilización de conceptos y destrezas matemáticas que les permitan reinterpretar la realidad y resolver problemas cotidianos del ámbito familiar, social y laboral, contribuyendo al mismo tiempo a establecer un lenguaje para la comprensión de los fenómenos científicos y tecnológicos.

4.1.1. Objetivos Fundamentales y Contenidos Mínimos Obligatorios

Dentro de lo que implica el aprendizaje de las matemáticas así como en todas las materias, se han establecido una serie de objetivos fundamentales y contenidos mínimos obligatorios que son los considerados adecuados para la buena formación de los niños.

El presente proyecto, está enfocado en el Primer Ciclo Básico, es decir, desde primero a cuarto año básico.

Según la descripción curricular se establece que en este ciclo debe buscar consolidar, sistematizar y ampliar las nociones y prácticas matemáticas que los niños y niñas ya poseen como resultado de la interacción con su medio y del trabajo realizado en el nivel de educación parvularia. También se busca promover el desarrollo de formas de pensamiento que posibiliten a los niños procesar información acerca de la realidad y profundizar así sus conocimientos acerca de ella; desarrollar una actitud positiva hacia el aprendizaje de las matemáticas y apropiarse de formas de razonar matemáticamente; adquirir herramientas que les permitan reconocer, plantear y resolver problemas, desarrollar la confianza y la seguridad en sí mismos, al tomar conciencia de sus capacidades, intuiciones y creatividad.

Los objetivos y contenidos planteados se presentan agrupados en torno a cuatro ejes temáticos: números, operaciones aritméticas, formas y espacio, y resolución de problemas. Esta organización ha sido realizada con propósitos analíticos con el fin de detallar con mayor precisión lo que se espera que niños y niñas hayan aprendido al finalizar el Nivel Básico 1 (NB1) y el Nivel Básico 2 (NB2). Sin embargo, esto no significa que los contenidos correspondientes a cada eje deban ser tratados en forma independiente. Muy por el contrario, la implementación didáctica de estos Objetivos Fundamentales y Contenidos Mínimos Obligatorios requiere de una articulación permanente de los contenidos de los cuatro ejes a fin de promover aprendizajes interrelacionados que correspondan a una visión integrada del quehacer matemático. Es decir, que en las actividades a realizar en el aula se establezcan secuencias de aprendizaje en las que se integren contenidos que son afines o complementarios y que provienen de los diferentes ejes temáticos.

4.1.2. Descripción de los cuatro ejes temáticos

A continuación se detallan los cuatro ejes temáticos que conforman la base curricular para el primer ciclo básico a partir de extractos del Marco Curricular de la Educación Básica, Objetivos Fundamentales y Contenidos Mínimos Obligatorios de la Educación Básica, establecido por el Ministerio de Educación del Gobierno de Chile.

4.1.2.1. Números

Se considera fundamental la asociación entre el aprendizaje de los números en el aula y los múltiples usos que éstos tienen en prácticas sociales en que los alumnos participan. En concordancia con lo anterior, el aprendizaje a nivel oral se considera como punto de partida y, por tanto, precede al escrito. De este modo, se favorece el que los niños avancen en sus razonamientos matemáticos y en su capacidad de establecer relaciones entre los números, al poder prescindir de las exigencias formales propias del lenguaje matemático escrito. Para promover la apropiación de los números naturales como una secuencia linealmente ordenada, se otorga una gran importancia a aprender a contar, en contextos muy diversos. La práctica de esta habilidad contribuye a desarrollar en el niño el sentido de la cantidad, es decir, a ser capaz de relacionar consistentemente un número con el monto de la cantidad que dicho número representa. Con igual propósito se promueve el desarrollo de habilidades tales como estimar, redondear, medir y comparar, aplicables tanto a conjuntos de objetos como a magnitudes.

Si bien las actividades relacionadas con la acción de medir se introducen en el subsector de Comprensión del Medio Natural, Social y Cultural, es en las clases de matemáticas donde se usan las medidas obtenidas, para organizarlas, procesarlas y obtener nuevas informaciones. De esta forma se incentiva la necesaria coordinación entre los diferentes subsectores del nivel.

No cabe duda que el aprendizaje de los números resulta más efectivo y se consolida mejor cuando se sustenta en una comprensión gradual de nuestro sistema de numeración, cuya estructura es bastante compleja. Si este hecho no se considera, se corre el riesgo de generar aprendizajes fragmentados, costosos en tiempo y energía y difíciles de generalizar. En consecuencia, se pone énfasis en la formación y transformación de números de acuerdo a las reglas del sistema de numeración, en el concepto de valor posicional y en el estudio de los números en bloques de tres cifras, acorde a la organización de este mismo sistema (unidades,

decenas y centenas en NB1, y unidades, decenas y centenas de mil en NB2). También se considera importante abordar el aprendizaje de la descomposición aditiva y multiplicativa de los números, ya que ésta proporciona elementos adicionales para el manejo de los números y la comprensión del sistema. Con el propósito de que los niños logren tener una visión más amplia del mundo de los números, en NB2 se incorporan las fracciones, las que complementan a los números naturales por cuanto posibilitan la cuantificación de partes de unidades, tanto en el ámbito de lo discreto (conteo) como de lo continuo (medición).

4.1.2.2. Operaciones aritméticas

La comprensión del sentido de las operaciones aritméticas es un objetivo fundamental en el aprendizaje de las matemáticas. Este aspecto de las operaciones pasa por la comprensión, tanto de las acciones que ellas representan, como de la posibilidad que ellas ofrecen para determinar información numérica desconocida a partir de información numérica conocida. Para ello, es necesario proponer a los alumnos, en forma oral, una gran variedad de relatos relacionados con las diversas acciones asociadas a cada una de las operaciones, y orientarlos para que puedan representar los números involucrados en ellas mediante objetos manipulables o dibujos simples. De este modo, la información desconocida podrá ser obtenida a través de conteos. Posteriormente, podrán incorporar la simbología asociada a cada una de las operaciones.

El aprendizaje del sentido de la adición y de la sustracción está contemplado para NB1, enfatizando el carácter inverso de cada una de estas operaciones con respecto a la otra. En NB2 se considera el aprendizaje del sentido de la multiplicación y de la división (con resto igual a cero) poniendo de relieve, en forma análoga, el carácter inverso entre ellas.

Es fundamental que los alumnos establezcan relaciones entre el estudio de las operaciones aritméticas en el aula y su aplicación en prácticas sociales habituales. Esto les permitirá abordar en la escuela problemas en los que utilizarán dichas operaciones para ampliar y precisar su conocimiento de la realidad. También, les proveerá de herramientas para desenvolverse con mayor autonomía en una realidad social tan rica en información numérica como la actual.

Además de conocer un amplio rango de situaciones que pueden ser representadas mediante las cuatro operaciones aritméticas elementales, consideradas como modelos matemáticos, los

alumnos necesitan disponer de procedimientos de cálculo rápidos y eficaces. Para tal efecto, se propone que el progreso en el aprendizaje de procedimientos de cálculo esté estrechamente ligado con el proceso de aprendizaje de los números, de modo que apoye a este último. Para ello, será necesario planificar cuidadosamente las relaciones entre los números que se incluyan en las prácticas operatorias.

Se asigna un lugar importante al aprendizaje de procedimientos de cálculo mental, que consisten en la memorización de algunas relaciones numéricas y en la capacidad de inferir rápidamente otras relaciones numéricas, mediante la aplicación de propiedades de los números y de las operaciones aritméticas. Es importante destacar que la práctica del cálculo mental constituye una suerte de laboratorio para el aprendizaje experimental de estas propiedades -único posible en el primer ciclo de educación básica- lo que representa una base de sustentación sólida para la construcción de ulteriores conocimientos matemáticos. Al mismo tiempo, la posibilidad de calcular mentalmente en forma eficaz contribuye a desarrollar en los estudiantes sentimientos de confianza en su capacidad de aprendizaje en el dominio de las matemáticas.

El aprendizaje del cálculo escrito se sustenta en los registros informales que hacen los alumnos para incrementar la capacidad de su memoria durante el cálculo mental. Al complejizarse las relaciones entre los números involucrados en un cálculo, la escritura proporciona un apoyo insustituible para consignar las etapas del proceso y retener los resultados parciales. Posteriormente, este registro puede irse haciendo en forma más resumida y, con la intervención del profesor, llegar a adoptar un formato convencional, válido para cualquier par de números y específico de cada una de las operaciones aritméticas que los niños están aprendiendo. Al término de NB2, se espera que los alumnos consideren la aplicación de procedimientos escritos convencionales, para sumar y restar números con diferentes cantidades de cifras, como una extensión lógica de los procedimientos aprendidos para sumar y restar números de dos y tres cifras. Respecto de la multiplicación, se espera que al término de NB1 los alumnos se apropien de un procedimiento convencional y lo apliquen a casos restringidos. En el caso de la división, en cambio, la apropiación de un procedimiento convencional no está considerada como un logro esperable al término del primer ciclo básico.

En NB2 se incorpora la calculadora como herramienta de cálculo, cuando la complejidad de éstos, debido a las relaciones entre los números involucrados o a la extensión del ámbito

numérico, así lo aconsejen. Es importante que los alumnos y alumnas manejen los conceptos de adición, sustracción, multiplicación y división, que sepan en qué circunstancias es necesario recurrir a cada una de estas operaciones, que manejen procedimientos de cálculo mental y escrito, y que puedan hacer una estimación razonable de los resultados que obtendrán; pero no es necesario ni conveniente que dediquen un tiempo excesivo a la realización de largos y tediosos cálculos escritos, que nada aportan al aprendizaje de las matemáticas y que, más bien, contribuyen a que los niños se alejen de esta área del conocimiento.

4.1.2.3. Formas y espacio

Una tarea importante a desarrollar en este eje es la de proporcionar a los niños y niñas un conjunto de experiencias que les permita reconocer la diversidad de formas de los objetos que les rodean, establecer relaciones entre ellas y considerar a las formas geométricas como simplificaciones de las formas que encuentran en el entorno.

El estudio de las formas geométricas en este ciclo está referido principalmente a cuadrados, rectángulos, triángulos y círculos, entre las figuras planas, y a cubos, prismas rectos, pirámides, cilindros y conos, entre los cuerpos geométricos. Los aprendizajes fundamentales radican en la identificación de los elementos que conforman a figuras y cuerpos, el reconocimiento de relaciones de posición y de medida entre estos elementos, la visualización y anticipación de las formas que se pueden obtener por yuxtaposición, separación y cambios de posición de formas básicas y la representación plana de los cuerpos geométricos estudiados.

Las figuras y los cuerpos geométricos que son estudiados son fuente de observación y de experimentación, a partir de objetos que tengan dichas formas u otras próximas a ellas. Para esto es importante que los objetos y materiales didácticos utilizados sean muy variados en tamaños y relaciones entre sus medidas y que los alumnos tengan múltiples oportunidades de construir objetos a partir de consignas específicas.

La descripción y estudio de las formas que nos rodean está en estrecha relación con su ubicación en el espacio. Surge, entonces, la necesidad de complementar dicho estudio con el de las posiciones y trayectorias, contenidos que, además, adquieren gran relevancia en otras áreas del conocimiento. En este sentido, son temas de estudio en este nivel la descripción y

ubicación de posiciones y la interpretación y representación gráfica de trayectorias de objetos y personas, considerando diferentes puntos de observación, referentes y direcciones.

4.1.2.4. Resolución de problemas

La resolución de problemas constituye el núcleo central de la actividad matemática y debe, por tanto, ocupar un lugar relevante en el trabajo del subsector desde los niveles más elementales, tanto como medio para el aprendizaje de los contenidos matemáticos del nivel, como con la finalidad de desarrollar la habilidad de resolución de problemas.

En consecuencia, en este eje se consideran dos aspectos. Uno es el desarrollo de la habilidad para resolver problemas, para el cual son indispensables la apropiación de los aspectos básicos relacionados con las etapas del proceso de resolución y el desarrollo de la confianza en la propia capacidad de formular y resolver problemas. El otro se refiere al tipo de problemas que los niños deben resolver, los que están relacionados con los contenidos de los ejes de números, operaciones aritméticas y formas y espacio.

Existen múltiples tipos de problemas matemáticos, así como múltiples formas de solucionarlos. Sin embargo, todos ellos tienen muchas cosas en común: requieren de un reconocimiento de su existencia, la búsqueda de una estrategia para resolverlos, la puesta a prueba de la estrategia elegida, entre otros. Enseñar a resolver problemas es un proceso largo, que exige enfrentar a alumnos y alumnas a múltiples y variadas situaciones, de modo que se apropien de algunos aspectos fundamentales del proceso, reconozcan su importancia y vayan adquiriendo cada vez más confianza y seguridad. También, para que realicen un proceso de meta cognición que les permita reconocer sus propias estrategias y para que desarrollen una actitud positiva y deseos de enfrentar la resolución de problemas con entusiasmo y perseverancia [Ministerio de Educación, 2002].

4.2. Inteligencia en la asignación de ejercicios mediante un motor de reglas

En cuanto a la asignación de ejercicios para los niños de la escuela, la idea es que el software siga el aprendizaje y desempeño de cada niño para asignar un ejercicio adecuado al nivel que éste ha alcanzado hasta el momento.

Existen variadas técnicas para llevar a cabo esta tarea, entre las cuales se pueden mencionar las redes neuronales artificiales [Kung, 1993], algoritmos genéticos [Jesús Alfonso López, 2000], motores de reglas, sistemas basados en conocimientos, lógica difusa y árboles de decisión, entre otras.

Teniendo en cuenta que la característica de “inteligencia” en la asignación de ejercicios es una parte importante dentro del sistema y ya se han dividido esfuerzos entre muchas más tareas, se ha tomado contacto con el alumno de Ingeniería Civil Informática, Pablo Mellado, quien tomó como proyecto el tema “Agentes Inteligentes para el desarrollo lógico-matemático infantil”, por lo tanto se ha optado por aprovechar su experiencia y así obtener más información sobre su trabajo.

Pablo Mellado, optó por la utilización de motores de reglas, el cual es un sistema que aplica reglas if/then a los datos que posee un determinado sistema. Esta aplicación de reglas se realiza con la finalidad de poder tomar una cierta acción en los datos o en el sistema. Su propósito primario es separar la lógica del ambiente, de la lógica del sistema, con esto la lógica del ambiente puede ser mantenida por separado. Utilizar un motor de reglas permite separar sus sentencias if/then del código de sistema, así luego se pueden manejar por separado. El comportamiento del sistema puede entonces ser modificado sin tener que cambiar el código. Las reglas se almacenan en forma separada con tal de poder modificarlas en cualquier momento que se estime.

Básicamente un motor de reglas está compuesto de tres elementos: un conjunto de reglas, el espacio de trabajo (o el conocimiento que posee), y el procesador de reglas. Las reglas son sentencias de la forma IF-THEN, de tal manera que si se cumplen todas las condiciones del IF se ejecutan todas las acciones del THEN. El espacio de trabajo es donde se guarda el conocimiento, que el motor utilizará para decidir que reglas deben activarse. Dentro de las reglas pueden existir conflictos. Un conflicto es cuando varias reglas distintas pueden activarse para el mismo conjunto de hechos y la aplicación de dichas reglas, pueden tener resultados contradictorios.

Algunas de las estrategias para resolver conflictos son: asignarle una prioridad a cada regla, estrategias FIFO o LIFO, aplicarlas en el orden en que se declararon o aplicarlas en orden aleatorio [Mellado, 2006].

4.2.1. Funcionamiento del Motor de Reglas.

La asignación inteligente de los ejercicios se ha basado en el trabajo de Pablo Mellado, destacando además que cuenta con una base pedagógica que lo respalda, para implementar de buena forma esta asignación mediante un motor de reglas, primero se debe tener un listado de ejercicios lo suficientemente amplio los cuales serán las alternativas o caminos a seguir para los niños según se vaya dando su desempeño.

Para estos ejercicios hay que tener varios parámetros de evaluación, como por ejemplo si respondieron correctamente o no, el tipo de respuesta que se dio, ya que puede ser que lo que respondieron esté mucho más alejado de la respuesta que las otras alternativas y el tiempo que se demoraron en responder. Luego se debe hacer una ponderación con todos estos factores y sacar un puntaje, este puntaje servirá para clasificar el ejercicio, con esto se consigue tener un ranking de ejercicios, los cuales deberán partir con un puntaje a criterio de los profesores de la Escuela Santa Lucía. Como se puede ver, el ranking será dinámico, es decir, podría variar en el tiempo según los niños resuelvan los ejercicios.

Al contar con este ranking de ejercicios, cuando un niño termine alguno, el motor de reglas evaluará si existe necesidad de avanzar a otro nivel más exigente (un ranking mayor). En éste punto es donde es vital conservar el historial del niño separadamente del resto, mediante cuentas de usuario.

Con toda esta lógica implementada, se podrá garantizar una correcta y dinámica asignación en los ejercicios a cada niño por separado, además de asegurar que el ranking se adaptará a las capacidades del grupo de niños que usarán el sistema.

4.3. Atributos Genéricos del Software Basado en sonido para Personas Ciegas.

Constructividad.

Este aspecto implica que el software debe permitir al usuario ejecutar acciones y construir a partir de las herramientas del software. En el caso del software educativo para invidentes, este atributo se logra en la medida que se entregue la información necesaria que permita construir

un esquema mental en el usuario, aspecto que le ayudará a desenvolverse en el ambiente virtual [mTISA, 2003].

Este aspecto es muy importante porque la constructividad debe estar orientada a construir un concepto general de software en los usuarios invidentes, concepto que hasta ahora no existe, así como debe permitir que el usuario interactúe directa y libremente con los elementos que se le presentan y poder decidir respecto al resultado final. Cabe destacar que la constructividad se encuentra directamente relacionada con la etapa de apropiación del software. Mientras más afianzada esté la etapa de apropiación en el usuario, mayor será el nivel de constructividad que desarrolle.

Para favorecer el desarrollo de constructividad en el usuario invidente debe existir o hacer lo siguiente:

- Un guión que contextualice en forma general el software.
- Un dialogo permanente entre el software y el usuario.
- Los diálogos deben ser precisos respecto de los sucesos.
- Las pistas auditivas deben ser estandarizadas y no arbitrarias.
- Estandarizar el uso de periféricos.
- Estandarizar la distribución de elementos.
- Incrementar los contenidos de acuerdo a la relación del usuario con el software y sus necesidades.
- Incorporar diferentes niveles en el juego.
- Entregar progresivamente las posibilidades de construcción.
- Utilizar metáforas relacionadas con la experiencia del alumno.

Navegabilidad.

La posibilidad que tenga el usuario invidente de desenvolverse en el espacio virtual está directamente relacionada con la posibilidad de construir mentalmente el espacio que navega y

basándose en ello, desplazarse. Este desplazamiento debe ser a través de pistas programadas en las funciones del teclado, así como pistas auditivas claramente definidas. La navegación en un software de este tipo debe efectuarse mediante teclado, de esta forma el usuario invidente podrá acceder al software de manera autónoma. Las teclas que se deben utilizar para estos fines deben tender a la estandarización, ya que de esta manera el uso de un nuevo software no implicará un nuevo aprendizaje en lo que se refiere a la interacción.

En consecuencia, el software deberá permitir al usuario moverse por los distintos botones presionando alguna tecla por definir (se sugiere el tabulador), desplegar menús en el caso de que los haya y desplazarse a través de los mismos. A medida que el usuario invidente se posiciona en cada uno de los botones deberá proporcionarse una retroalimentación auditiva que otorgue el nombre del botón o ícono, junto con leer los menús.

Interactividad.

El software es interactivo cuando permite una retroalimentación inmediata con el usuario, aspecto que favorece la interacción cognitiva por la cual el usuario comprende y cambia su entendimiento, lo que ayuda a desarrollar la estructura del pensamiento. La interactividad del usuario con discapacidad visual con el software debe ser a través de pistas audibles, imágenes bien definidas, y con la utilización de teclas y/o comandos estandarizados previamente.

En un usuario invidente la interactividad debe favorecer el desarrollo de la iniciativa y potenciar el auto aprendizaje, a través de un aprendizaje heurístico que permita y estimule el descubrimiento y la aplicación de los contenidos en el software a otras áreas de su educación.

Contenidos.

En un software diseñado para personas con discapacidad visual las instrucciones que se proporcionan deben ser audibles, ya que éste no podrá leer órdenes o enunciados que aparezcan en la interfaz. No obstante estas igualmente pueden existir para las personas videntes y con baja visión que utilicen el recurso. Las instrucciones verbales deben ser claras, precisas y breves, ya que el usuario deberá recordarlas al momento de interactuar con el software. Por esta razón, es recomendable que el usuario tenga también la posibilidad de volver a escuchar estas instrucciones cuando así lo requiera.

Interfaz.

Fuentes: La tipografía a utilizar debe ser de palo seco, evitar en lo posible utilizar tipografías romanas o con sheriff, a menos que sean todas con mayúsculas, descartar completamente el uso de tipos decorativos. Tamaño no menor a los 18 puntos, con un interletraje medio, debe ser de un color que posibilite un buen contraste con el fondo sobre el cual se encuentra. Cabe destacar que los contrastes más favorables para las personas con baja visión son amarillo-azules, negro-blancos.

Imágenes: Deben poseer un buen contraste con los fondos, deben ser sencillas, claras y precisas, de manera que el usuario pueda reconocerlas fácilmente con sus restos visuales. Por lo general, se opta por imágenes más icónicas que fotográficas.

Colores: Fuertes y contrastantes, de preferencia dentro de la gama de los azules y amarillos, ya que estos colores son los que percibe de mejor manera una persona con visión residual, debido a que son diametralmente opuestos tanto en croma como en luminiscencia.

Iconos: Simples, grandes, con un mensaje claro, de buen contraste en el fondo. Deben ser interpretados fácilmente por el usuario, además deben tener asociada una locución o sonido para que la persona ciega pueda saber donde se encuentra posicionando y las posibilidades que este ícono proporciona.

Botones: Deben ser grandes y sencillos, además a ellos debe estar asociada una locución o sonido, para que los usuarios invidentes puedan saber donde se encuentran posicionados y cual es su función.

Audio: En el caso de software para personas ciegas el audio se convierte en el componente más importante del producto, por esta razón éste debe ser de la más alta calidad y contemplar todas aquellas instancias donde se hace imprescindible su existencia. La utilización de este medio se justifica en la presentación del producto, en las diferentes instancias de interacción, en la descripción de los íconos, botones, lectura de menús, musicalización de contexto, etc. Las locuciones que se incluyan en el producto deberán ser realizadas por voces congruentes con el grupo de usuario al cual esta orientado el software, es decir, si el recurso es un producto para niños se sugiere emplear una voz entretenida y motivadora, que posea una buena articulación y que sea grata de escuchar.

5. Desarrollo

5.1. Solución propuesta

Para el problema descrito se ha decidido llevar a cabo el desarrollo de un sistema que permita ejercitar el contenido entregado en el aula, específicamente en el ramo de matemáticas desde 1° a 4° básico, a niños invidentes y de baja visión haciendo uso de una interfaz adecuada para ambos tipos de usuarios.

El sistema permitirá hacer ingreso de los ejercicios, los cuales quedarán almacenados en el sistema, al igual que información relevante ligada al desempeño de los ejercicios, la cual será utilizada en reportes y estadísticas que permitan al profesor hacer evaluaciones de desempeños grupales o individuales.

La asignación de los ejercicios será previo análisis del desempeño particular de cada alumno dentro de la actividad correspondiente, garantizando así que la ejercitación esté de acuerdo al nivel que corresponde al alumno.

Ante los requerimientos captados desde los profesores, tipos de usuarios y participación de éstos en el sistema, se ha tomado la decisión de dividir el sistema en dos subsistemas o aplicaciones separadas, una para el profesor y otra para el alumno. A continuación se detallan las características de ambas.

5.1.1. Aplicación para el profesor

Esta aplicación permitirá al profesor gestionar ejercicios, ver reportes y estadísticas. En la gestión de ejercicios el ingreso de un nuevo ejercicio será eligiendo alguna de las bases o plantillas de ejercicios preestablecidas en el sistema, la cual tendrá distintos parámetros asociados dependiendo del tipo, siendo obligatoria y común entre todas la grabación del enunciado del ejercicio, la que servirá para el caso que el niño sea invidente. Las alternativas de respuesta serán transformadas a sonido por el sistema de forma automática. Además a cada ejercicio se agregarán datos que permitan realizar la evaluación del desempeño, las cuales son

tiempo de respuesta y nivel de dificultad del ejercicio (ranking). Para completar la gestión se podrán modificar y eliminar ejercicios.

En cuanto a los reportes, se podrá generar un listado de alumnos ordenado por ranking, pudiendo ser ordenado de mayor a menor y viceversa. Otra forma de ver el desempeño de forma general es la posibilidad de generar gráficos que representen y agrupen los errores o aciertos por cada actividad. Por último también se podrá revisar un listado histórico de todos los ejercicios realizados por cada niño, mostrando el tiempo estimado, tiempo usado, ranking del ejercicio y ranking del niño para la actividad correspondiente, tipo de respuesta (correcta, incorrecta en menor o mayor grado), fecha y número del ejercicio.

Es esta aplicación además podrá hacer acceso el administrador, el cual podrá realizar la gestión de usuarios y de respaldos.

En la gestión de usuarios deberá hacer ingreso de los datos de identificación típicos y además identificar el tipo de usuario que se ingresa, pudiendo ser administrador, profesor o alumno. Para todos los casos debe ingresar una contraseña de acceso al sistema, la cual junto con el nombre de usuario permitirá la identificación al ingreso del sistema.

La gestión de respaldos está pensada en posibilitar respaldar, la información contenida en la base de datos y los archivos utilizados para los ejercicios, de forma fácil y transparente para el profesor, el cual podrá realizar respaldos, eliminarlos y cargarlos cuando estime conveniente. Los respaldos serán organizados, indicando fecha y hora en que fueron realizados.

5.1.2. Aplicación para el alumno

Esta aplicación tendrá como fin permitir la ejercitación por parte del alumno previa identificación, la cual es vital para almacenar la información de desempeño y realizar la asignación adecuada de los ejercicios.

La interfaz de esta aplicación será de interfaz audible y gráfica especialmente adecuada para niños invidentes y de baja visión. Para cada ejercicio estará en pantalla el enunciado y las cuatro alternativas de respuesta, pudiendo navegar por ellas a través del teclado. Paralelamente las opciones serán dadas en forma sonora, al igual que el enunciado. La navegación será hecha a través del teclado, utilizando teclas comunes en aplicaciones para

personas ciegas o de baja visión. Se podrá repetir el enunciado, recorres las alternativas y dar como respuesta la alternativa en la cual se ha posicionado. Realizado esto, la aplicación informará si la respuesta es correcta o no, en el último caso indicará cual era la respuesta correcta al alumno.

El sistema antes de asignar un ejercicio al niño, revisará el desempeño que este ha tenido para la actividad correspondiente, de acuerdo a eso presentará al niño un ejercicio adecuado en dificultad. Una vez dada la respuesta se analizará esta y el tiempo usado para responder, lo que derivará en la modificación del desempeño del niño y eventualmente el ranking que ha alcanzado hasta el momento. Otra característica será la de evaluar el desempeño del ejercicio, recogiendo datos de desempeño cada vez que un alumno lo responda, lo que permitirá la modificación dinámica del ranking ingresado en primera instancia por el profesor.

5.2. Restricciones del Proyecto

Durante el planteamiento del proyecto han surgido las siguientes restricciones:

- El sistema consta de bases o plantillas de ejercicios adecuadas a los contenidos exigidos sólo para el primer ciclo básico en las bases curriculares.
- La aplicación tiene como fin apoyar los contenidos entregado en el aula por los profesores, en ningún caso se pretende reemplazarlos.
- El stock de ejercicios para cada dificultad y actividad dependerá del profesor a cargo, ya que el sistema sólo asigna entre los que estén disponibles.
- La calidad de grabación es sensible al entorno, por lo tanto dependerá del profesor asegurar la calidad de la grabación de lo enunciados.
- Para la ejercitación es importante la concentración de los niños en resolver el ejercicio, por lo que es esencial el uso de audífonos, disponibles en la Escuela Santa Lucía.
- La periodicidad de los respaldos dependerá de quien este a cargo de realizarlos. Al igual del respaldo físico de estos, para prevenir pérdidas de estos ante probables fallos del servidor que provocaran la perdida total de la información almacenada en él.

- El sistema utiliza librerías de sonido que no se han actualizado para ser utilizadas en Windows Vista, por lo tanto se garantiza el funcionamiento correcto del sistema bajo el sistema operativo Windows Xp, el cual es el utilizado por la Escuela Santa Lucía en sus laboratorios.

5.3. Lenguajes y Herramientas de Desarrollo

Para el desarrollo de las interfaces del sistema se han utilizado los siguientes lenguajes de programación:

- Java, para desarrollar la aplicación del profesor.
- ActionScript, para desarrollar la aplicación del alumno.
- Lenguaje de Consulta Estructurado (SQL), para el acceso y consultas a la base de datos.
- PHP, para generar la comunicación entre la aplicación Flash y la base de datos.
- XML, para guardar las preguntas y respuestas de los ejercicios.

Las herramientas son los programas utilizados como apoyo para la creación del software, estas se nombran a continuación:

- MySQL, como base de datos [MySQL].
- Apache, como servidor Web [Apache].
- Macromedia Flash, para programar en ActionScript.
- NetBeans, para programar en Java [NetBeans].
- Sybase PowerDesigner 12, para generar diagramas representativos del sistema.
- Microsoft Office Word, para escribir la documentación del sistema.
- DBDesigner4, para diseñar la base de datos.

5.4. Recursos de Hardware y Software para puesta en marcha

5.4.1. Hardware

La Escuela Santa Lucía cuenta con un laboratorio equipado con 12 computadores con las características mostradas en la tabla 5.1.

Tabla 5.1 Características de los computadores.

Placa Madre:	Intel DG31PR
Procesador:	Intel Pentium DC-2180 2,0 GHz
RAM:	1GB
Disco Duro:	160 Sata
Internet:	Sí, ADSL 1024/128 KB/s
Gabinete	<i>Codegen 6089</i>
Monitor	<i>Samsung TFT 15,6" 632NW</i>
Audífonos	Logitech ClearChat Comfort USB

5.4.2. Software

Para el correcto funcionamiento del sistema en el interior de la Escuela Santa Lucía, se requiere el software nombrado a continuación:

- MySQL, como base de datos [MySQL].
- Apache, como servidor Web [Apache].
- PHP, para conectar la aplicación del alumno que esta desarrollada en Flash con la base de datos MySQL.

- JVM, para ejecutar la interfaz de administración del profesor.
- PHPMyAdmin, para administrar la base de datos MySQL [PHPMyAdmin].

5.5. Paradigma y Metodología

Se conoce que la construcción de un sistema de software implica la toma de decisiones sobre la arquitectura del sistema. Estas decisiones pueden ser cruciales para el éxito o fracaso del sistema resultante, por lo que se requiere seleccionar un proceso de desarrollo de software con el fin de obtener la calidad del sistema de software deseada y cumpla con los requerimientos establecidos. Por lo tanto, para el logro exitoso de las distintas metas es necesario definir un conjunto de tareas como marco de trabajo, las cuales son aplicables a todos los proyectos del software, con independencia de su tamaño o complejidad. Estas tareas llevarán a distintos productos intermedios que en su conjunto formarán el sistema de software que se esta construyendo.

Acercándose más al sistema es que se debe mencionar que diversos autores han utilizado la ingeniería de software para la elaboración de material multimedia interactivo, logrando de esta manera que el proceso de desarrollo y mantenimiento del software educativo sea una actividad que dependa de pautas establecidas, con modelos conceptuales y herramientas de trabajo, y no del arte de aquellos que tengan la experiencia exclusivamente.

Se conoce además, que para lograr software educativo con las condiciones deseadas, se deben incorporar dentro de las fases de análisis y diseño, aspectos didácticos y pedagógicos, es decir, el diseño instruccional, de manera que faciliten y garanticen la satisfacción de las necesidades educativas del público al que va dirigido el software. Se deben involucrar también a los usuarios, para conseguir identificar necesidades y/o problemas específicos y se puedan establecer mecanismos de resolución adecuados y apoyar cada una de las fases en sólidos principios educativos, comunicativos y computacionales.

Sabiendo que se debe evaluar y optar por la metodología y paradigma más adecuados para el desarrollo del software, teniendo presente las problemáticas propias del tipo de sistema sumado a otros factores no propios del sistema, es que se ha fijado la atención en el modelo de cascada (en algunos textos llamado ciclo de vida clásico o modelo lineal secuencial) y en

el proceso unificado (UP), considerando que la combinación de ambas podría satisfacer a cabalidad las necesidades del sistema.

A continuación se mencionaran brevemente estas dos alternativas para luego explicar la decisión sobre el uso de ambas combinadas con la correspondiente justificación.

5.5.1. Modelo lineal secuencial (modelo cascada)

Este modelo sugiere un enfoque sistemático, secuencial, para el desarrollo del software, el cual consta de 4 actividades bien definidas, las cuales son análisis, diseño, implementación y prueba.

La ilustración 5.1 muestra las actividades del modelo cascada y la forma en que se debe implementar.

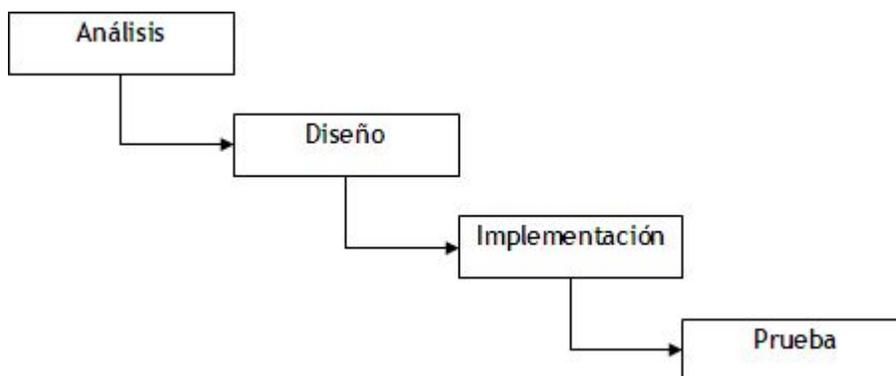


Ilustración 5.1 Modelo Cascada.

Para cada actividad la definición es la siguiente:

Análisis: Es el proceso de recopilar los requisitos que debe satisfacer el software, siendo necesario entender la naturaleza del software a construir, además de comprender el dominio de información del software, la función requerida, comportamiento, rendimiento e interconexión.

Diseño: El diseño del software es realmente un proceso de muchos pasos que se centra en cuatro atributos distintos de programa: estructura de datos, arquitectura de software, representaciones de interfaz y detalle procedimental (algoritmo). El proceso del diseño traduce requisitos en una representación del software donde se puede evaluar su calidad antes de que comience la codificación.

Implementación: En este paso se hace la transformación del diseño del software a código. El éxito y facilidad de esta tarea se debe en gran parte a la calidad y detalle que tenga la etapa de diseño.

Prueba: Ya habiendo codificado el software se comienza con las pruebas del programa. El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales; es decir, realizar las pruebas para la detección de errores y asegurar que la entrada definida produce resultados reales de acuerdo con los resultados requeridos.

El gran problema que enfrenta este modelo es el hecho de que rara vez un proyecto sigue esta secuencia plenamente. Uno de los factores que provoca esto es la dificultad en algunos casos de obtener clara y totalmente todos los requerimientos al principio del proyecto, de haber algún error en la captura de requerimientos sólo se evidenciará cuando esté muy avanzado el proyecto y luego de una larga espera por parte del cliente, siendo catastrófico a estas alturas.

A pesar de presentar estas debilidades este modelo sigue siendo el más utilizado y es un importante cimiento para otros modelos que han incorporado otras características en busca de mejorarlo [Pressman, 2002].

5.5.2. Proceso Unificado (UP)

Un proceso de desarrollo de software es un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Sin embargo, el Proceso Unificado más que un simple proceso; es un marco de trabajo genérico aplicable a las distintas necesidades y variables presentes en los distintos tipos de proyectos que puedan existir.

El Proceso Unificado está basado en componentes, queriendo decir que el software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. Otra característica del Proceso Unificado es el uso del Lenguaje Unificado de Modelado (UML) para la representación de todos los modelos que reflejarán al sistema en cada etapa de su construcción. Pero sin duda lo que distingue al Proceso Unificado son sus tres cualidades claves, el ser dirigido por casos de uso, centrado en la arquitectura y además ser iterativo e incremental.

A continuación se presentan con mayor detalle las características mencionadas con anterioridad.

Dirigido por casos de uso: la motivación para la creación de un software es el satisfacer necesidades de distintos tipos de usuarios, pudiendo ser personas o sistemas externos, definiendo así como usuario a quien tenga interacción con el sistema que se construye. Por lo mismo para capturar eficazmente los requisitos para el software es necesario preguntarse quienes interactuarán con el sistema, para obtener que resultado de él. Esta interacción provocará una secuencia de acciones con un resultado final importante para aquel usuario. Esto es lo que se llama un caso de uso y es nada menos que la representación de un requerimiento funcional; siendo el modelo general de casos de uso, la representación de todos estos requerimientos. A pesar de ser cierto que los casos de uso guían el proceso, estos se van creando en conjunto con la arquitectura, ya que los casos de uso guían la arquitectura y la arquitectura influye en la elección de los casos de uso, madurando ambas cosas a medida que avanza el desarrollo del sistema.

Centrado en la arquitectura: la arquitectura del sistema es nada menos que la preocupación de tomar en cuenta las diversas vistas que pudiera tener el sistema de software para su construcción, entre estas vistas podrían estar contenidas una gran parte de los requerimientos no funcionales. A través de esta arquitectura se le da una forma preliminar al sistema, obtenida de no más del 10% de los casos de uso de mayor relevancia, detallando éstos para así descubrir más acerca de la arquitectura, lo que a su vez permitirá madurar más casos de uso.

Iterativo e incremental: como es sabido la construcción de un sistema puede fácilmente tomar bastante tiempo, por eso es práctico dividir el trabajo en mini proyectos, definidos en número y metas para cada uno. Así cada mini proyecto es una iteración sobre una fase en la construcción del software que contribuye al crecimiento del mismo. Esta división en mini proyectos sólo es efectiva y conveniente si el número de iteraciones es controlado y se establecen bien los incrementos necesarios para cada iteración. Cabe destacar que cada iteración hace referencia a las mismas tareas del modelo lineal o cascada, demostrando que este modelo es importante y marca los cimientos en este proceso al igual que en otros modelos que lo incorporan [Jacobson, 2000].

En la ilustración 5.2 se puede observar un diagrama explicativo del Proceso Unificado, en el cual se muestran las fases que este tiene.

Ilustración 5.2

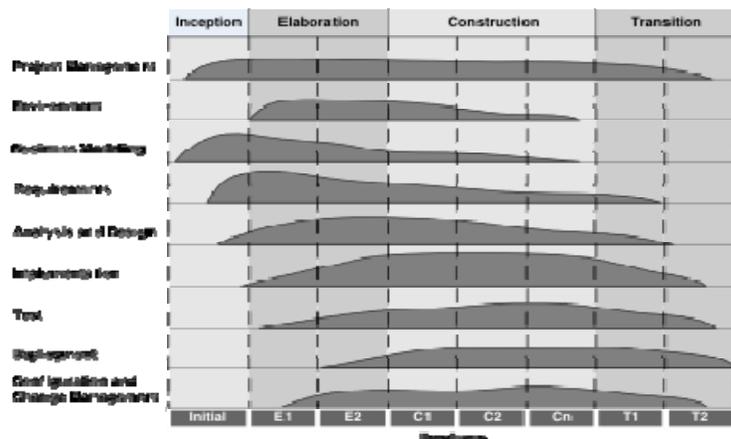


Ilustración 5.2 Proceso Unificado.

5.5.3. Combinación entre Modelo Cascada y Proceso Unificado

Ante la tarea de optar por una metodología para el desarrollo del sistema, el primer paso fue evaluar la conveniencia en distintos aspectos, que se deberían tener presente antes de tomar una decisión. Tales aspectos fueron varios, los cuales se trataran con mayor detalle a continuación.

Para el uso de cascada la motivación fue:

- Se dedicó bastante tiempo al inicio del proyecto a la investigación sobre el tema, además de hacer visitas a distintos lugares que ayudaran a comprender mejor el mundo de los invidentes.
- Las funcionalidades para el profesor no son desconocidas ni demasiadas, como para no poder capturar todos los requisitos al principio de forma eficaz.
- La Escuela Santa Lucía, facilitó software que usaban actualmente indicando las virtudes y defectos en ellos.

- Se consiguió el apoyo de la Escuela Santa Lucía para el desarrollo del sistema, teniendo de esta forma un cliente de quien capturar los requerimientos.
- Por estar la escuela en Santiago, no convenía por costo y tiempo, viajar muy seguido para mostrar avances, prefiriendo hacer un trabajo exhaustivo de investigación y captura de requerimientos solamente al principio del Proyecto.

Sin embargo, también existían razones para optar por el Proceso Unificado, las cuales son:

- Al tratarse de niños de baja visión e invidentes, los requerimientos sobre la interfaz, su gráfica y usabilidad, estaban fuera de lo que se tenía como experiencia anterior, siendo demasiado riesgoso no iterar al construir las para lograr asegurar y satisfacer a cabalidad los requerimientos.
- Por haber sido aprendido y practicado anteriormente en la Universidad, ya se contaba con experiencia en el uso del Proceso Unificado, siendo una ventaja su uso.
- El hecho de que el Proceso Unificado proporcione una cantidad de modelos, usando UML, que permiten mostrar claramente el sistema a clientes y desarrolladores.
- Define cada flujo de trabajo y su producto, siendo los distintos modelos.

5.5.4. Conclusiones

Para ambas opciones existían distintas motivaciones, ante lo cual decidió utilizar la combinación de ambas, creando un modelo híbrido. Entonces la construcción de la aplicación del profesor sería según el modelo de cascada puro, por no representar mayor riesgo, pero en el caso de la aplicación del niño el mismo modelo cascada sufriría cambios en las etapas de diseño e implementación, agregando iteraciones incrementales propias del Proceso Unificado, de modo de minimizar el riesgo de no satisfacer los requerimientos ligados a esta aplicación. Además conservando el uso de UML y los modelos propios que define el Proceso Unificado para cada flujo de trabajo como producto. Demás esta decir que la metodología será orientada a objetos dado que el Proceso Unificado implica esto. Todas estas decisiones están fundamentadas en los puntos favorables para cada opción que se enumeraron anteriormente.

5.6. Estudio de Factibilidad

En la realización de un proyecto informático es necesario determinar la viabilidad de éste según los recursos disponibles para su realización. Además se debe establecer la rentabilidad del proyecto a futuro para decidir si se justifica realizar la inversión. Para el presente proyecto se ha realizado el siguiente estudio de factibilidad.

5.6.1. Factibilidad Técnica

La factibilidad técnica consiste en determinar si el sistema puede ser diseñado, construido y utilizado con los conocimientos técnicos, tecnología existente y a la vez accesible.

Diseño y construcción: Para el diseño y construcción se han utilizado algunas herramientas de código libre, para las cuales no hay restricciones de uso, encontrándose además accesibles vía Internet. Para el caso de lenguajes o herramientas que si requieren de licencias, estas las provee la Escuela de Ingeniería Informática. Esta última también provee de hardware necesario, sumando además el disponible por quienes participaron del proyecto.

Utilización: La Escuela Santa Lucía dispone del hardware necesario para el funcionamiento del sistema de forma correcta y eficiente.

5.6.2. Factibilidad Operacional

Para llevar a cabo la operación del sistema es necesaria una instrucción a los profesores de la Escuela Santa Lucía y a los alumnos. Dado que los profesores a cargo están debidamente capacitados en el área informática y los alumnos ya se encuentran familiarizados en el uso del computador y software similar, es que el aprendizaje para el uso del sistema de forma correcta y eficiente será breve.

5.6.3. Factibilidad Legal

La factibilidad legal indica si el proyecto cumple con las normativas y leyes vigentes para la realización de un proyecto informático, observando si incurre en infracciones, violaciones u otras faltas que pudiesen imposibilitar la puesta en marcha del sistema.

Para este proyecto no existen trabas legales que impidan el buen desempeño y funcionamiento del sistema, puesto que no incurre en infracciones a las leyes vigentes, específicamente:

- Ley N° 19.223 la cual tipifica figuras penales relativas a la informática, en artículos 1, 2, 3, 4 pues el software no daña, no altera, etc. Ningún sistema de información.
- Ley N° 17.336 en relación a la propiedad intelectual, específicamente el artículo 41 que dice relación con la realización de copias o adaptaciones del proyecto computacional, pues en el proyecto no se realizan copias de código fuente, interfaces, etc. de algún otro software que pudiese tratarse de la misma materia.

Para el desarrollo la Escuela de Ingeniería Informática provee de las licencias necesarias en el caso de los lenguajes o herramientas que lo requieran. Para la puesta en marcha no hay licencias comprometidas.

5.6.4. Factibilidad Económica

En el aspecto económico, para el desarrollo del sistema no se tuvo que incurrir en gastos, ya que las licencias estaban disponibles o bien eran gratuitas. El detalle se observa a continuación.

Tabla 5.2 Software y lenguajes utilizados.

Lenguaje o Herramienta	Disponibilidad	Costo
Base de datos MySql	Uso gratuito para fines educacionales.	No hay
Flash y ActionScript	Licencia de uso para la Escuela de Ingeniería Informática.	No hay
Java y NetBeans	Gratuita.	No hay
PHP	Gratuita	No

		hay
Servidor Web Apache	Gratuita	No hay

En lo relativo a hardware necesario, fue provisto por la Escuela de Ingeniería Informática, sumado también el disponible por parte de quienes participaron en el proyecto.

Otro tipo de gasto ligado al desarrollo es el costo de la mano de obra, el cual queda nulo ya que el proyecto fue desarrollado como requisito para titulación de los participantes.

Para la puesta en marcha la Escuela Santa Lucía cuenta con un laboratorio que cumple con todos los requerimientos de hardware para el sistema, por lo cual no se necesitó adquirir ningún hardware adicional.

Por último hay que considerar los beneficios económicos del proyecto. En este caso los beneficios son más bien intangibles, ya que el sistema permitirá automatizar y agilizar un proceso que el profesor realizaba de forma personalizada a cada alumno, o bien utilizaba software que carecía de ciertas características que le son más útiles para evaluar los conocimientos de los alumnos. Ahora el profesor podrá evaluar el aprendizaje de los alumnos y la rapidez de resolución, lo que podrá hacerse simultáneo a una gran cantidad de alumnos, para finalmente obtener resultados generales o individuales para su análisis.

En resumen, el proyecto implica incurrir en gastos durante ninguna de sus etapas, obteniendo finalmente los beneficios intangibles mencionados anteriormente, concluyendo que es factible de realizar desde el punto de vista económico.

5.6.5. Conclusión

Puesto que los estudios de factibilidad técnica, operacional, legal y económica arrojan resultados positivos, se concluye que el proyecto es factible de ser desarrollado.

5.7. Riesgos del Proyecto

La gestión de riesgos es una de las tareas mas importantes dentro del desarrollo de un proyecto, ya que anticiparse a ellos evitaría tener complicaciones en la programación del proyecto o en la calidad del software que se está desarrollando. De forma simple se puede concebir un riesgo como la probabilidad de que una circunstancia adversa ocurra. Los riesgos son una amenaza para el proyecto, para el software que está desarrollando y para la organización.

Categoría de Riesgos:

- Riesgos del Proyecto.
- Riesgos del Producto.
- Riesgos del Negocio.

El Proceso de Gestión de riesgos comprende las siguientes etapas:

- Identificación de Riesgos.
- Análisis de Riesgos.
- Planificación de Riesgos.
- Supervisión de Riesgos.

5.7.1. Identificación de Riesgos

En la tabla a continuación, se muestran los tipos de riesgos más comunes en el desarrollo de un software, junto con su clasificación y una descripción detallada de cada uno.

Tabla 5.3 Riesgos Generales.

Riesgo	Tipo Riesgo	Descripción
Retardo en asimilar	Proyecto y	Los desarrolladores se demoran en aprender las

nuevas tecnologías.	Producto	tecnologías utilizadas para desarrollar el sistema.
Cambio Organizacional Escuela Santa Lucia.	Proyecto y Producto	Cambio organizacional de la escuela, directora o profesor encargado del proyecto se retiran.
No disponibilidad de hardware.	Proyecto	Falta de hardware necesario para que los desarrolladores implementen el sistema.
Cambios en los Requerimientos.	Proyecto y Producto	Dependiendo de la evolución del proyecto irán apareciendo otros requerimientos.
Retrasos en la entrega de los prototipos.	Proyecto y Producto	Los prototipos del proyecto no estarán listos antes de la fecha de entrega.
Subestimación de tamaño.	Proyecto y Producto	El tamaño del sistema ha sido subestimado, lo que genera una sobrecarga de trabajo.
Software resulta difícil de utilizar.	Producto	Los niños invidentes y con baja visión tienen problemas para entender el funcionamiento del software.
Cambio de Tecnología.	Negocio	La tecnología en la cual se construye el sistema ha sido sustituida por una nueva tecnología.

5.7.2. Análisis del Riesgo

El análisis de riesgo, corresponde a la calificación de cada uno, definiendo la probabilidad que ocurra en el transcurso del proyecto y los efectos que generarían si ocurriesen, a continuación se detallan con mayor profundidad.

Tabla 5.4 Calificación de los riesgos.

Riesgo	Probabilidad	Efectos
Retardo en asimilar nuevas tecnologías.	Baja	Tolerable

Cambio Organizacional Escuela Santa Lucia.	Baja	Serio
No disponibilidad de hardware.	Baja	Serio
Cambios en los Requerimientos	Moderada	Serio
Retrasos en la entrega de los prototipos.	Moderada	Serio
Subestimación de tamaño.	Alta	Serio
Software resulta difícil de utilizar.	Moderada	Serio
Cambio de Tecnología.	Moderada	Tolerable

5.7.3. Planificación del Riesgo

La planificación de riesgos, corresponde a las estrategias adoptadas para contrarrestarlos en caso que llegasen a ocurrir, lo que también podría llamarse planes de contingencia. En la tabla siguiente se muestra una definición del riesgo y la estrategia usada para contrarrestarlos.

Tabla 5.5 Estrategias para contrarrestar los riesgos.

Riesgo	Estrategia
Retardo en asimilar nuevas tecnologías.	Se deben reasignar las tareas de acuerdo a los conocimientos de los desarrolladores para aprender una tecnología nueva.
Cambio Organizacional Escuela Santa Lucia.	Se debe de preparar un documento indicando los beneficios que otorga el proyecto para la escuela.
No disponibilidad de hardware.	Reestructurar la planificación de manera de que el hardware que va estar no disponible, no sea esencial en algún periodo determinado.
Cambios en los Requerimientos.	Rastrear la información para valorar el impacto de los requerimientos, maximizar la información oculta en ellos.

Retrasos en la entrega de los prototipos.	Reorganizar las labores dentro del grupo de trabajo, y potenciar las virtudes del grupo laboral.
Subestimación de tamaño.	Aumentar la asignación de h/h a partes del proyecto que han sido subestimadas.
Software resulta difícil de utilizar.	Realizar adaptación del prototipo para que sea más sencillo de utilizar por los niños.
Cambio de Tecnología.	Averiguar las posibilidades que ofrecen los distintos equipos que se encuentran en el mercado para la implementación del proyecto.

5.7.4. Supervisión del Riesgo

La supervisión del riesgo, corresponde a los indicadores que muestran una posible tendencia que estos ocurran en un proyecto. A continuación se muestran los indicadores potenciales para detectar la posible aparición de algunos de los riesgos definidos para el proyecto.

Tabla 5.6 Indicadores potenciales de riesgos.

Riesgo	Indicadores Potenciales
Retardo en asimilar nuevas tecnologías.	El desarrollo de ciertos módulos del sistema demora más de lo estimado.
Cambio Organizacional Escuela Santa Lucia.	Falta de acciones por parte de la escuela Santa Lucia.
No disponibilidad de hardware.	Entrega retrasada del hardware, problemas tecnológicos detectados.
Cambios en los Requerimientos.	Demasiadas peticiones de cambios y surgimientos de nuevas necesidades por parte del cliente.
Retrasos en la entrega de los	Fracaso en el cumplimiento de alguno de los tiempos

prototipos.	acordados.
Subestimación de tamaño.	Aparecen nuevas funcionalidades que dan mayor complejidad al sistema.
Software resulta difícil de utilizar.	Se registran muchas respuestas incorrectas cuando los niños desarrollan una actividad utilizando un prototipo.
Cambio de Tecnología.	Nueva tecnología sale al mercado reemplazando la ya existente.

5.8. Análisis de Requerimientos

La ingeniería de requerimientos y su clasificación se presenta a continuación.

Funcionales.

- Permitir diferenciación de tipos de usuario a través de cuentas.
- Interfaces distintas y adecuadas tanto para niños como para profesores.
- Generación de estadísticas individuales o grupales en cuanto al rendimiento de los niños en los ejercicios.
- Gestión de usuarios niños y profesores.
- Asignación adecuada de ejercicios para cada niño, previa evaluación de su desempeño anterior, mediante un motor de reglas.
- Modificación dinámica del nivel de dificultad asignado a cada ejercicio, dependiendo de cómo fue abordado tal ejercicio por lo niños.
- Ingreso de nuevos ejercicios por parte de los profesores.

No funcionales.

- Considerar las restricciones propias de la Escuela en el diseño e implementación de la solución.
- Se debe procurar la facilidad de cambios al sistema, eventualmente por nuevo requerimientos.
- Se debe garantizar la facilidad de mantenimiento del sistema mediante la modularización de sus partes.
- El sistema debe estar siempre en óptimas condiciones de operación sin importar la cantidad de usuarios que lo estén usando, manteniendo el ritmo ágil y dinámico en la realización de los ejercicios.
- Robustez y estabilidad del sistema, sobre todo para la interfaz para los niños.
- Manual de usuario.
- Interfaz amigable, simple e intuitiva que permite una rápida adaptación de los usuarios al sistema, tanto por parte de los profesores como de los niños.

5.9. Diseño de ejercicios

A continuación se detallan las bases de los ejercicios diseñados, para esto se han separado los ejercicios de acuerdo a los cuatro ejes temáticos propuestos en las bases curriculares.

Números.

Presentación de secuencias de números, preguntando si corresponde al tramo que se pide y según la suma a efectuar para formarlo. Eventualmente podrían ser secuencias de fracciones o decimales.

¿Cual es la secuencia correcta entre 9 y 14 sumando 1 cada vez?

9,	10,	12,	13,	14	
10,	11,	12,	13		
1,	10,	11,	12,	13,	14
10, 11, 12, 13, 14, 15					

Presentación de un número preguntando cual de los números dados como alternativa es menor o mayor al presentado. Números naturales, fracciones y decimales.

¿Cual de estos números es menor a 16?

17

19

22

14

Presentación de un valor y varias alternativas de su descomposición, preguntando cual de ellas es la correcta.

¿Cual de las siguientes descomposiciones es correcta para el número 12?

2 + 9

14 - 4

6 + 6

Operaciones.

Sumas, restas, multiplicaciones y divisiones (simples o combinadas) de números con una incógnita ya sea en alguna de las cantidades afectadas por la operación o preguntando por el resultado, se dan opciones de la cual una es la correcta.

1 + X = 3, ¿Que valor tiene X?

2

3

4

5

Operación incógnita con los valores operados y el resultado visible, preguntando sobre que operación fue aplicada.

12 X 2 = 6 ¿Qué operación se aplicó?

Suma

Resta

División

Multiplicación

Formas y espacios.

Preguntas acorde al aprendizaje exigido en su nivel. Un ejemplo para 1° básico podría ser preguntar por objetos de su entorno cercano y a que figura geométrica básica se asemeja.

¿El balón de fútbol a cual de estas figuras se asemeja más?

Cuadrado

Rectángulo

Triángulo

Círculo

Resolución de problemas.

Preguntas acorde al aprendizaje exigido en su nivel. Podría quizá hacerse preguntas sobre problemas de la vida diaria, a los cuales se presentarán alternativas de respuesta, preguntando por la correcta.

Temas: operaciones con cantidades, operaciones con dinero, cálculo de horas.

Si Juan tiene 6 manzanas y está con 11 amigos, ¿cuantas manzanas le faltan para que cada amigo y él tengan una?

5

6

4

7

5.10. Diseño del Motor de Reglas

Para hacer que el sistema pueda decidir de forma autónoma la asignación de la dificultad del ejercicio siguiente previa evaluación de variables obtenidas a partir del desempeño de los niños en las actividades, es que se debe crear una base de conocimiento, la cual será analizada y actualizada por un motor de reglas.

La idea del diseño de un motor de reglas es de justamente, y valga la redundancia, la de definir en exactitud las reglas a la que serán sujetas las variables ligadas a la realización de las actividades por parte de los niños, llamadas variables de entrada, obteniendo como resultado la decisión de avanzar, retroceder o mantener la dificultad de los ejercicios, además de obtener un puntaje que se sumará o restará al valor de dificultad asignado al ejercicio desarrollado.

Como primer paso, para acercarse más a cómo funcionará el motor de reglas, se seleccionaron las variables, las que se describen a continuación.

NOTA: para definir límites en la calificación de dificultad de los ejercicios se establece que la dificultad vaya de 1 a 50, siendo la dificultad 1 la más fácil y la dificultad 50 la más difícil.

Variables de entrada.

Uso del tiempo: Valor que indicará el tiempo que demoró el niño en responder el ejercicio en comparación al tiempo que el profesor estimó adecuado.

Uso del tiempo = tiempo usado / tiempo estimado

Tipo respuesta: Indica si el niño ha resuelto correctamente el ejercicio o no. En el caso de una respuesta errónea esta variable indicará que tan cerca se estuvo de la respuesta correcta.

Tabla 5.7 Tipo de Respuesta.

Tipo de Respuesta	Valor
Correcta	1
Incorrecta menor grado	2
Incorrecta mayor grado	3

Desempeño histórico niño: Representará el desempeño histórico del niño en la dificultad actual. Este valor se considera necesario para poder llevar a cabo de mejor manera la decisión de asignar mayor, menor o igual dificultad al niño.

Tabla 5.8 Desempeño histórico niño.

Desempeño histórico niño	Valor
Excelente	2
Bueno	1
Regular	0
Malo	-1
Pésimo	-2

Desempeño histórico en ejercicio: Representa la estimación de dificultad histórica del ejercicio de acuerdo al desempeño que muestran los niños al desarrollarlo.

Tabla 5.9 Desempeño histórico ejercicio.

Dificultad ejercicio	Valor
Muy fácil	2
Fácil	1
Adecuada	0
Difícil	-1
Muy difícil	-2

Variables de salida.

Desempeño histórico en ejercicio: Corresponde a la misma variable de entrada, sólo que será actualizada por el motor de reglas de acuerdo al desempeño del niño en el ejercicio recientemente desarrollado.

Nuevo Ranking ejercicio: Representará el nuevo ranking de dificultad asociado al ejercicio, el que pudiera ser menor, igual o mayor al que tenía al comienzo. Vale destacar que si éste es distinto al que tenía (sufre modificación) el desempeño histórico en ejercicio es reseteado, es decir, queda en desempeño adecuado (0).

Desempeño histórico niño: Corresponde a la misma variable de entrada, sólo que será actualizada por el motor de reglas de acuerdo al desempeño del niño en el ejercicio recientemente desarrollado.

Nuevo Ranking niño: Representará el nuevo ranking de dificultad asociado a la actividad y el niño, el que pudiera ser menor, igual o mayor al que tenía al comienzo. Vale destacar que si éste es distinto al que tenía (sufre modificación) el desempeño histórico niño es reseteado, es decir, queda en grado regular (0). La ilustración 5.3 muestra las entradas y salidas del motor de reglas.

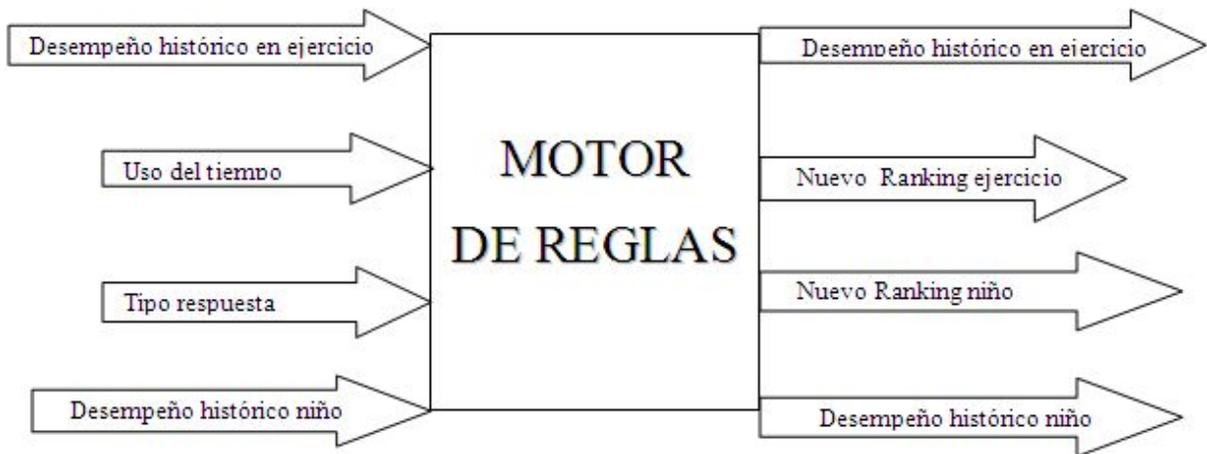


Ilustración 5.3 Variables de entrada y salida del Motor de Reglas.

Ya habiendo definido y explicado las variables a continuación en la Ilustración 5.4 se presenta un diagrama de flujo de cómo tomará las decisiones el motor de reglas.

Ilustración 5.4 Funcionamiento del Motor de Reglas.

(Para ver presione Ilustración 5.4)

Como se puede ver, se han diseñado las reglas de modo de solucionar lo siguiente:

- Se hará el seguimiento del desempeño de los niños con cada ejercicio, asegurando tener mucho más de una sola evaluación para modificar la dificultad asignada al ejercicio en cuestión.
- Igualmente habrá un seguimiento del desempeño del niño en la dificultad actual, permitiendo una evaluación más adecuada al momento de decidir si aumentar, disminuir o mantener la dificultad asignada para él.
- Las respuestas al ser correctas, incorrectas en menor y mayor grado, permiten una mejor evaluación de la respuesta, dando indicios más claros sobre el real desempeño del niño, así en caso de error y tomando en cuenta el tiempo de respuesta, no siempre se castiga el desempeño del niño en la actividad, calificándolo no de forma extrema (pésima o excelente) sino de rendimiento regular.

5.11. Elementos de la arquitectura lógica y física

La arquitectura física del sistema a implementarse en la Escuela Santa Lucía se muestra a continuación en la ilustración 5.5.

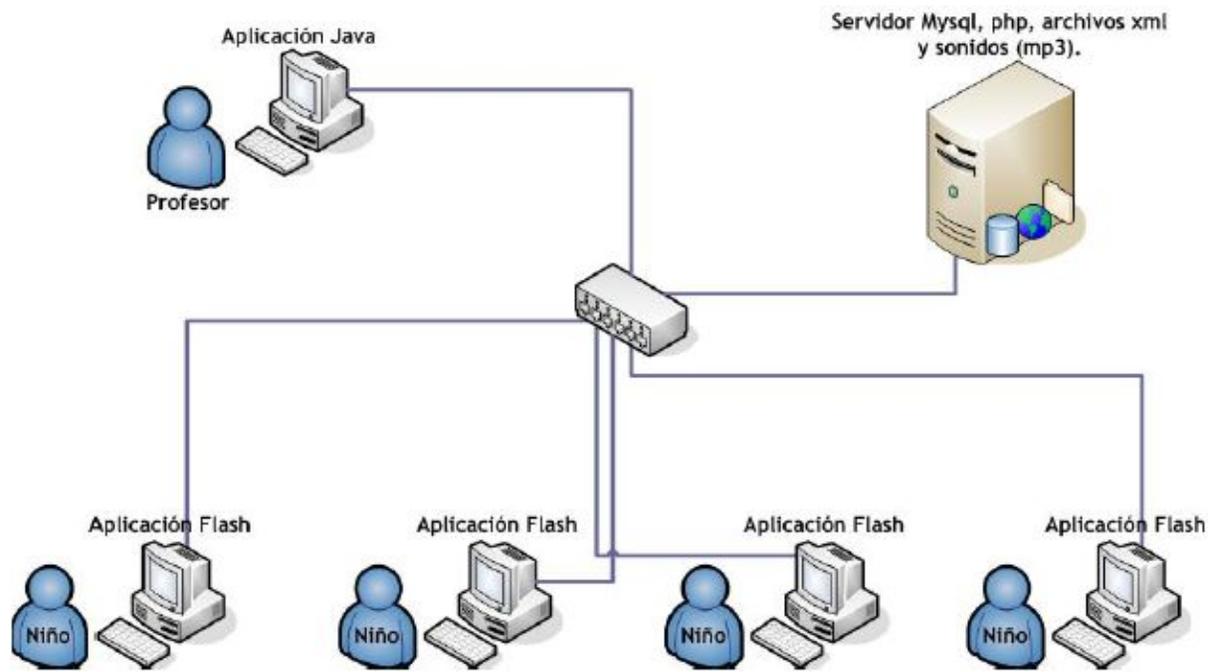


Ilustración 5.5 Arquitectura física del sistema.

5.11.1. Actores del sistema

A través de las visitas a la Escuela Santa Lucia, es que se logró determinar tres roles fundamentales, que interactuaran directamente con el software educacional a desarrollar.

Alumno: El alumno es la persona que interactúa con la interfaz de actividades del software, realiza una serie de ejercicios, los cuales son asignados de forma inteligente por el programa, esto se realiza dependiendo de las respuestas dadas por el alumno.

Profesor: Es la persona que interactúa con la interfaz de la administración. El es quien puede modificar la lista de ejercicios pertenecientes a una actividad en particular, ingresando un nuevo ejercicio, eliminando o actualizando uno existente. Además, el profesor puede revisar estadísticas, como por ejemplo, los alumnos con mejor y peor ranking y el historial de ejercicios realizados por un alumno en particular.

Administrador: El Administrador interactúa con la interfaz de administración del sistema. Su labor consiste en gestionar a los usuarios, ya sea creando, modificándolos o eliminándolos del Sistema.

5.11.2. Diagrama General de Casos de Uso

En la ilustración 5.6 se puede observar el diagrama general de casos de uso, este cuenta con 3 actores principales: Profesor, Administrador y Alumno. Además se especifican 16 casos de uso.

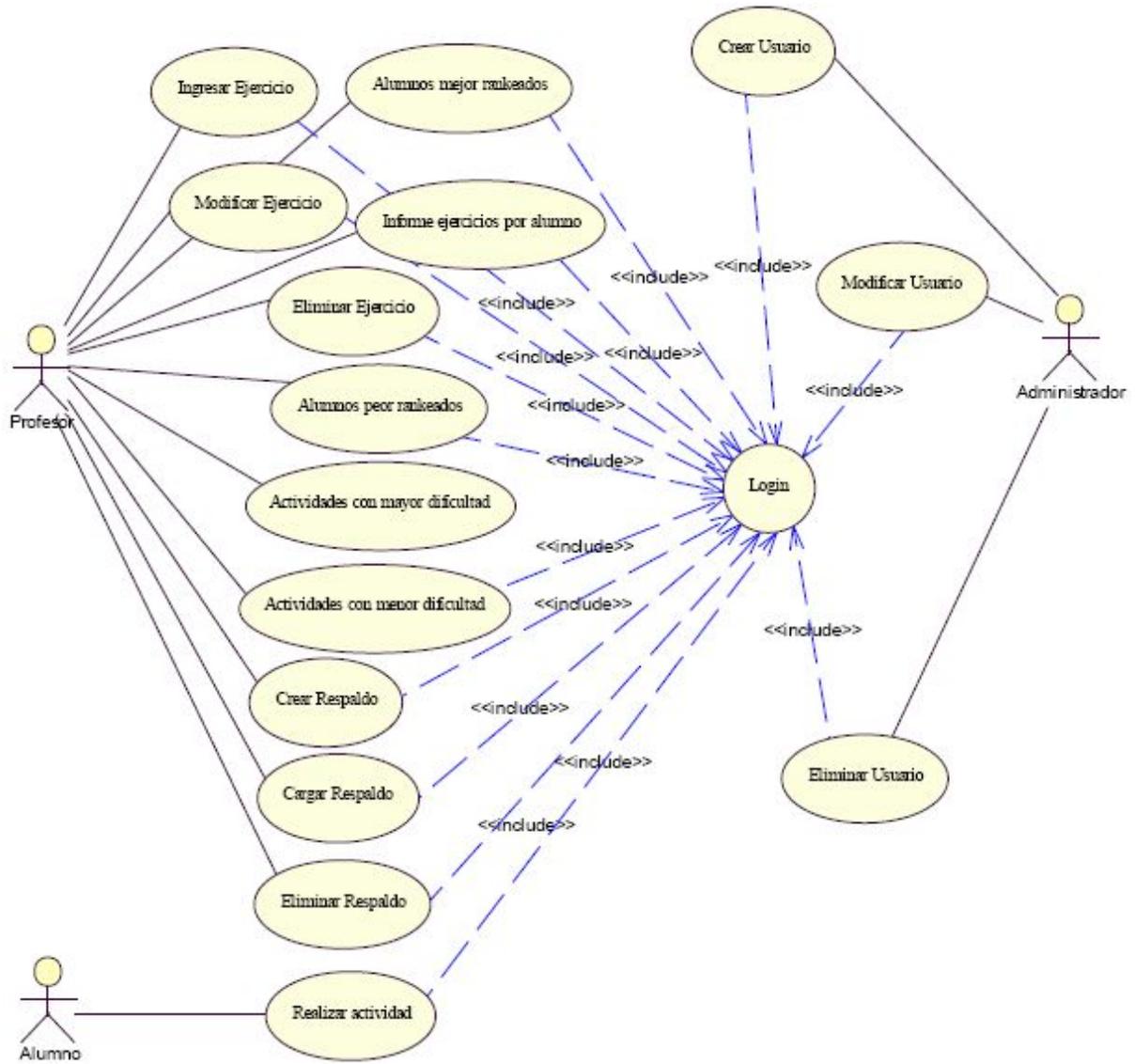


Ilustración 5.6 Diagrama general de C-U.

5.11.3. Modelo de Clases

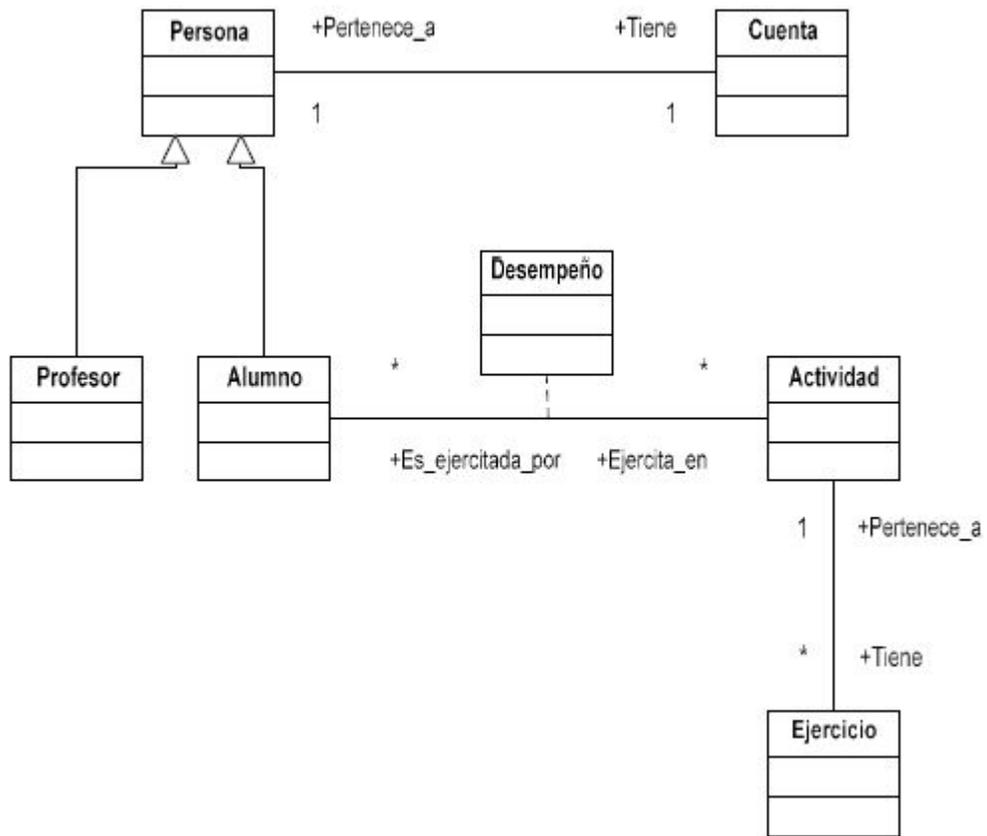


Ilustración 5.7 Modelo de Clases.

5.11.4. Modelo Relacional

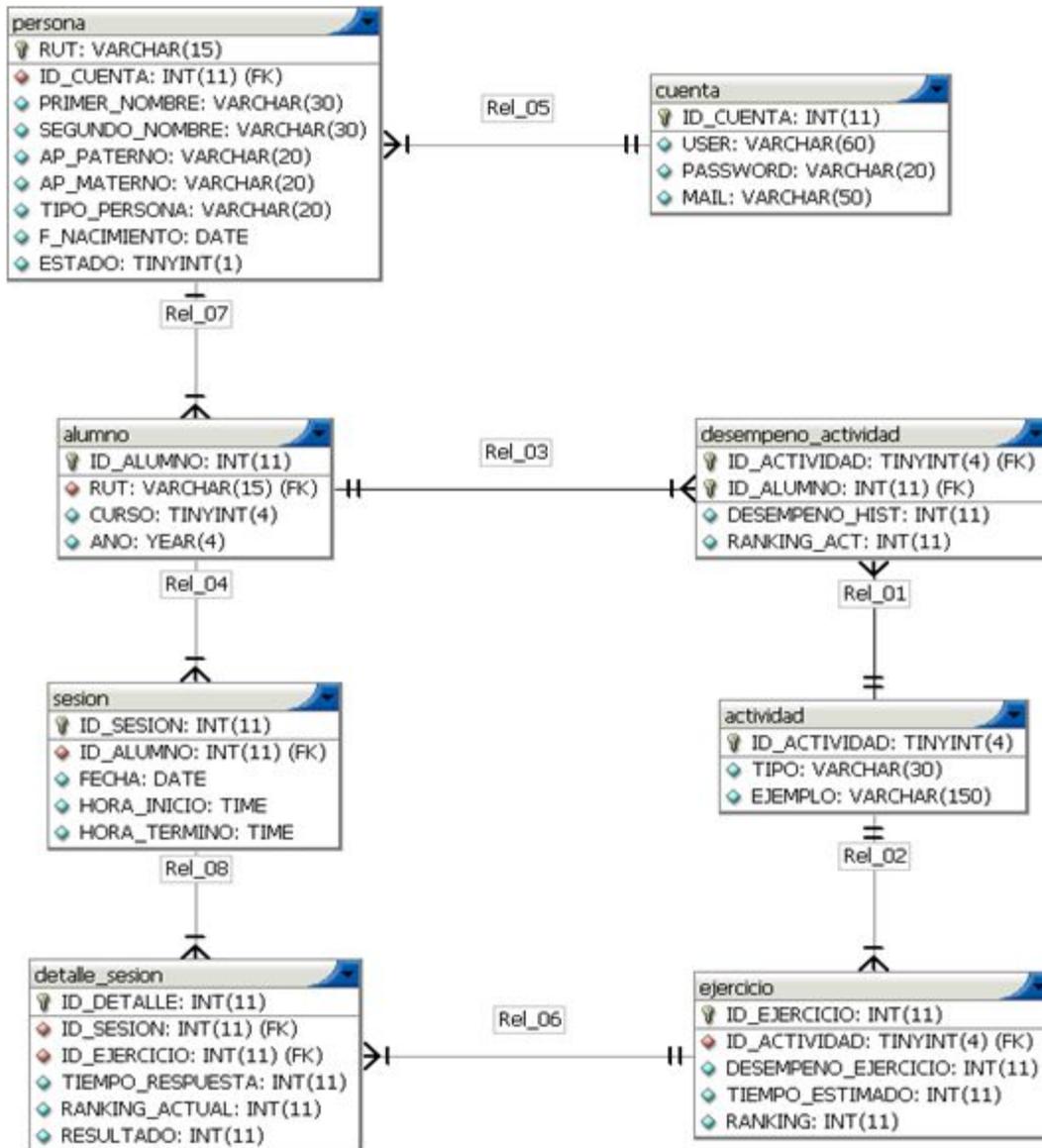


Ilustración 5.8 Modelo Relacional

5.12. Implementación del Sistema

Como ya se explicó anteriormente de forma detallada, para el desarrollo del sistema se decidió utilizar el Proceso Unificado para hacer uso de los artefactos que propone, además de la notación UML. Sin embargo, las dos aplicaciones serían desarrolladas siguiendo modelos distintos, la aplicación del profesor sería desarrollada siguiendo el modelo cascada o tradicional y la aplicación del alumno siguiendo el modelo iterativo e incremental, este último

perteneciente al Proceso Unificado. Esta decisión esta debidamente argumentada anteriormente.

A continuación se detalla la implementación de ambas aplicaciones por separado, según el modelo correspondiente.

5.12.1. Implementación de la aplicación del profesor

Esta aplicación fue desarrollada siguiendo el modelo cascada, el que define cuatro actividades a cumplir para el desarrollo de un sistema, las cuales son análisis, diseño, implementación y prueba. Las actividades de análisis y diseño ya se han detallado anteriormente, las cuales dieron como resultado diversos artefactos, los cuales representan la estructura de datos, arquitectura de software, representaciones de interfaz y detalle procedimental. El paso siguiente fue la implementación de la aplicación guiándose en los productos de las actividades anteriores.

A continuación se muestran las interfaces de la aplicación ya implementada.



Ilustración 5.9 Identificación para acceso a la aplicación.

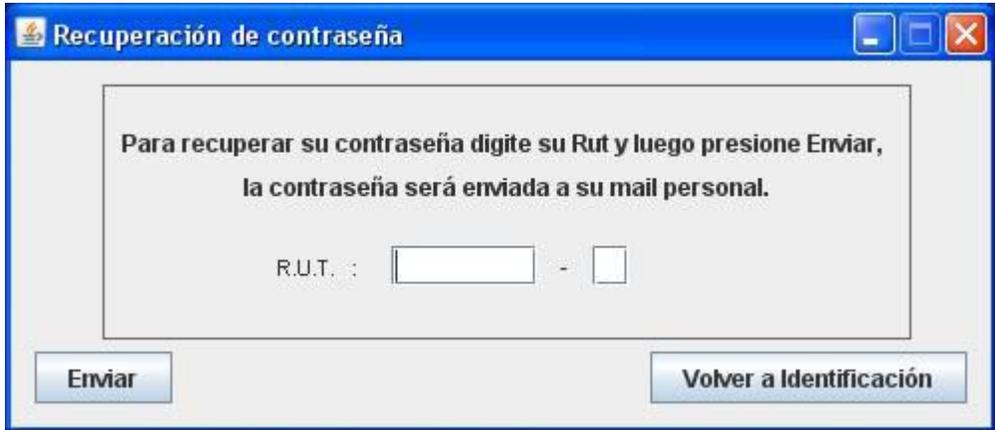


Ilustración 5.10 Recuperación de contraseña.

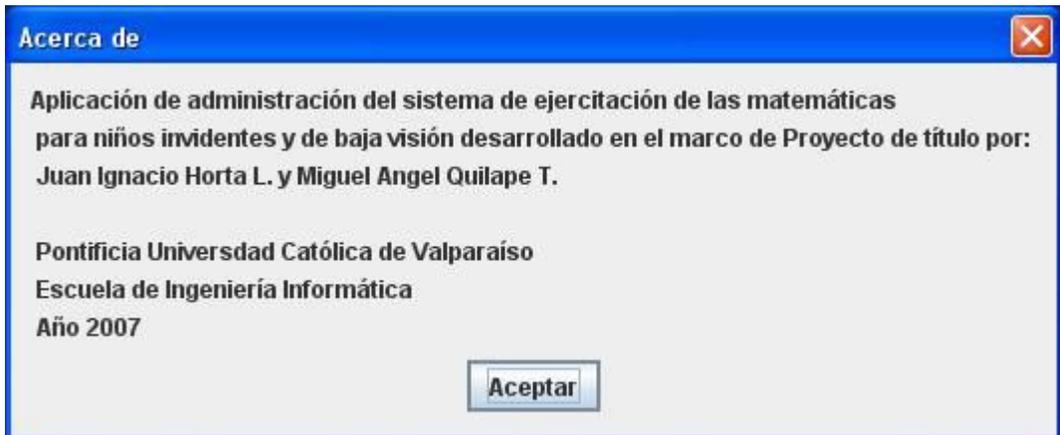


Ilustración 5.11 Ventana de información acerca del sistema.



Ilustración 5.12 Menú para usuario tipo Administrador.

Detalle del menú Administrador.

Menú General

- Salir

Gestión Usuarios

- Nuevo
- Editar información
- Eliminar

RespalDOS

- Gestionar

Ayuda

- Guía Rápida Alumno (despliegue del manual del alumno en pdf)
- Guía Rápida Profesor (despliegue del manual del profesor en pdf)

- Acerca de...

Crear Usuario

Datos Personales

Ingreso rut -

Primer Nombre

Segundo Nombre

Apellido Paterno

Apellido Materno

Fecha de Nacimiento 1939 Enero 1

Tipo Usuario

Confirmar Validar Datos

Cuenta Profesor/Administrador

user/rut -

Contraseña

Repita Contraseña

E-Mail

Datos Alumno

user

password

Curso

Año

Crear Usuario Volver al Menú Cancelar

Ilustración 5.13 Ingreso de nuevo usuario.

Modificar Usuarios

Seleccione Persona

15852177-6 ▼

Datos Personales

Primer Nombre: Juan

Segundo Nombre: Ignacio

Apellido Paterno: Horta

Apellido Materno: Lucero

Confirmar Validar Datos

Cuenta Profesor/Administrador

user/rut:

Contraseña:

Repita Contraseña:

E-Mail:

Modificar Volver al Menú Cancelar

Datos Alumno

user:

password:

Curso: Primero Básico ▼

Año: 1998 ▼

Ilustración 5.14 Modificación de datos de usuarios.

Eliminar Persona

Seleccione Persona

TIPO DE CUENTA	R.U.T	PERSONA
Administrador	15852177-6	Juan Ignacio Horta Lucero
Profesor	6031575-2	Jose Pedro Fuenzalida Chapa
Alumno	20270512-k	alonso ignacio salinas horta
Alumno	21108786-2	antonia ignacia salinas horta

Eliminar Volver al Menú

Ilustración 5.15 Eliminación de usuarios.

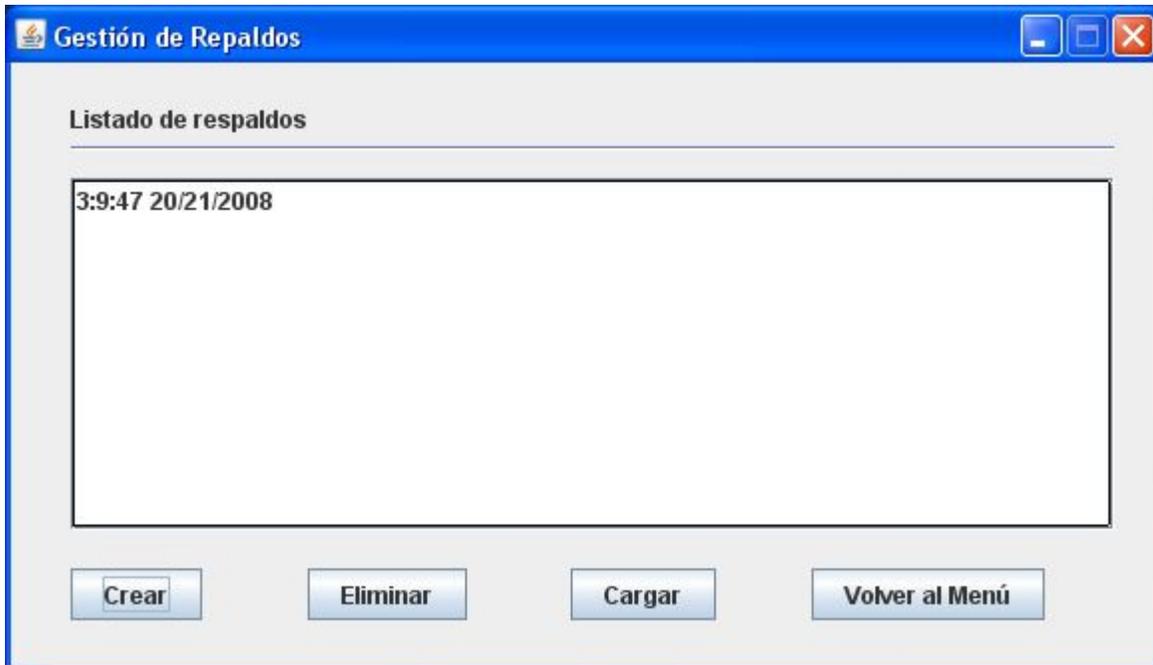


Ilustración 5.16 Creación, eliminación y carga de respaldos.



Ilustración 5.17 Menú para usuario tipo Profesor.

Detalle del menú Profesor.

Menú General

- Salir

Gestión Ejercicios

- Nuevo
- Editar
- Eliminar

Reportes

- Ejercicios por alumno
- Rankings de mayor a menor
- Rankings de menor a mayor
- Actividades con mayor cantidad de errores
- Actividades con menor cantidad de errores

Ayuda

- Guía Rápida Alumno (despliegue del manual del alumno en pdf)
- Guía Rápida Profesor (despliegue del manual del profesor en pdf)
- Acerca de...

Ingreso de nuevo Ejercicio

Ejemplo del ejercicio
 ¿Cual es la secuencia correcta entre 9 y 14 sumando 1 cada vez?

Seleccione el tipo de actividad
 1 Numeros

Grabar el audio
 Ejercicio.mp3
 Grabar
 Escuchar

Escriba el ejercicio

A) Cuadrado Suma Correcta Incorrecta Leve Incorrecta Grave
 B) Triángulo Resta Correcta Incorrecta Leve Incorrecta Grave
 C) Círculo Multiplicación Correcta Incorrecta Leve Incorrecta Grave
 D) Rectángulo División Correcta Incorrecta Leve Incorrecta Grave

Dificultad asignada 1 **Tiempo estimado** Segundos

Crear Ejercicio
 Cancelar
 Volver al Menú

Ilustración 5.18 Ingreso de nuevo ejercicio.

Modificar Ejercicios

Actividad 1 Numeros **Ejercicio** 1
 2
 3
 4
 5

Seleccione un Ejercicio... Cargar

Grabar el audio
 Ejercicio.mp3
 Grabar
 Escuchar

Pregunta :
 La pregunta del ejercicio ?

A) Cuadrado Suma Correcta Incorrecta Leve Incorrecta Grave
 B) Triángulo Resta Correcta Incorrecta Leve Incorrecta Grave
 C) Círculo Multiplicación Correcta Incorrecta Leve Incorrecta Grave
 D) Rectángulo División Correcta Incorrecta Leve Incorrecta Grave

Dificultad asignada 1 **Tiempo estimado** Segundos

Modificar
 Cancelar
 Volver al Menú

Ilustración 5.19 Modificación de ejercicios.

Eliminar Ejercicio

Actividad: 1 Numeros Ejercicio: 1, 2, 3, 4, 5

Seleccione un Ejercicio... Cargar

Pregunta :
La pregunta del ejercicio ?

Opciones	Tipo Respuesta	Tiempo Estimado	Ranking
A)	<input type="text"/>	<input type="text"/>	<input type="text"/>
B)	<input type="text"/>		
C)	<input type="text"/>		
D)	<input type="text"/>		

Eliminar Volver al menú Cancelar

Ilustración 5.20 Eliminación de ejercicios.

Ejercicios por Alumno

Buscar Alumno

Todos los Alumnos
 Nombre
 Apellido
 Curso

Alumnos

Todos los informes

Seleccionar informe de un mes
 Seleccionar informe entre fechas

Detalle de ejercicio realizado por actividad

Ejercicio	Fecha	Dificultad	T. Estimado	T. Usado	Respuesta

Ilustración 5.21 Listado de ejercicios por alumno.

Alumnos con mejores resultados

Alumnos más altos

1. antonia ignacia salinas horta/ Curso 1 año 2008
2. alonso ignacio salinas horta/ Curso 2 año 2008

Ilustración 5.22 Alumnos de mejores Rankings.

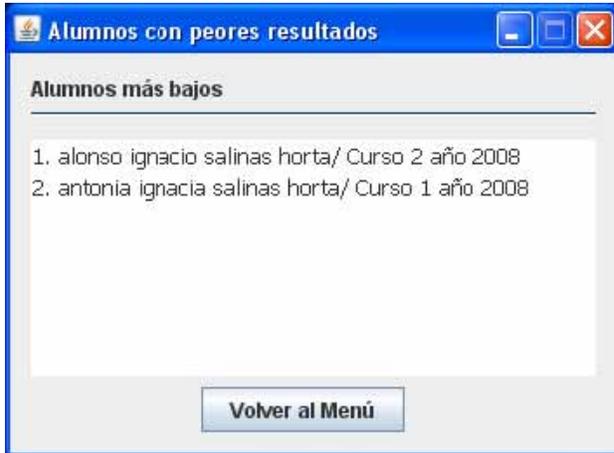


Ilustración 5.23 Alumnos con peores Rankings.



Ilustración 5.24 Actividades con mayor cantidad de errores.

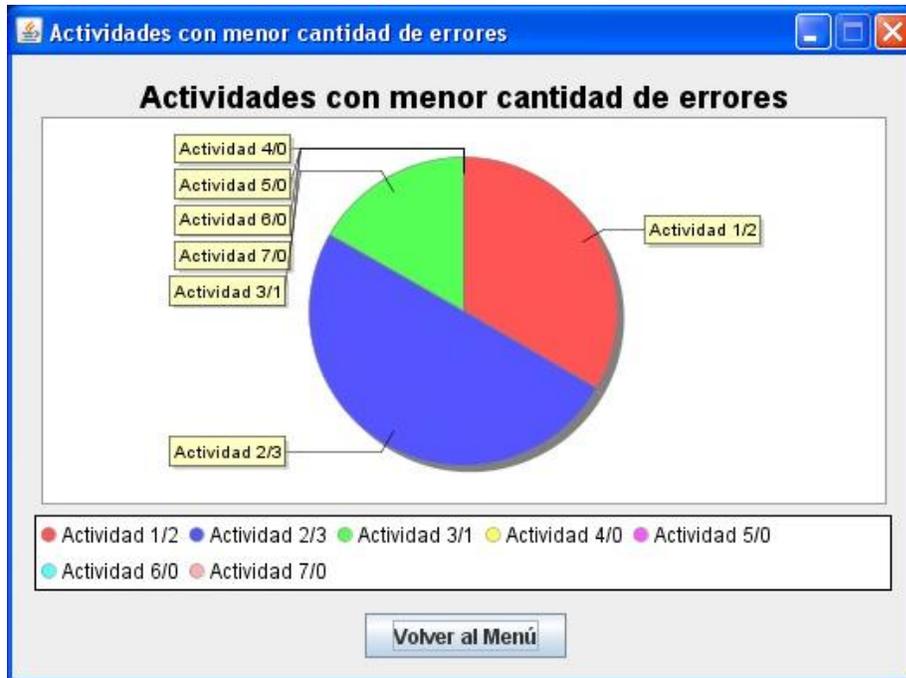


Ilustración 5.25 Actividades con menor cantidad de errores.

5.12.2. Implementación de la aplicación del alumno

Para la implementación del alumno se utilizó el modelo iterativo e incremental, el utilizado por el Proceso Unificado, debido a las características especiales que debían considerarse para esta aplicación. Para esta aplicación el análisis de requerimientos ya está detallado anteriormente, siendo la mayoría requerimientos no funcionales. Para la aplicación del alumno se decidió desarrollarla en Flash y ActionScript. A continuación se mencionan y detallan las iteraciones.

5.12.2.1. Iteración “Comunicación entre flash y Base de Datos”

Análisis.

Se investigó acerca de las formas de conexión entre una aplicación del alumno y MySQL, teniendo en cuenta siempre la optimización en el acceso a la base de datos y la modularidad. Para este caso se decidió hacer uso de la tecnología XML como alternativa al almacenamiento de información y del lenguaje PHP como intermediario en la comunicación de la base de datos y la aplicación.

Diseño.

Se diseñó una versión de archivos XML y PHP para probar la conexión.

Implementación.

Se codificaron los XML y PHP usados para la conexión, además de crear la aplicación para estos fines.

Prueba.

Se probó la conexión, pasando valores desde la base de datos a la aplicación y viceversa, arrojando resultados positivos en ambos casos.

5.12.2.2. Iteración “Estructura lógica de la aplicación”

Análisis.

Se analizó el grupo de datos a almacenar y la conveniencia de conservarlos en archivos XML o en la base de datos, pensando en la modularidad y optimización de los recursos.

Diseño.

Se estableció que los datos propios de cada ejercicio serían almacenados en archivos XML, evitando acceso excesivo a la base de datos. Dichos archivos estarán en un servidor para ser cargados de forma dinámica por la base de datos. El único audio a grabar será el enunciado del ejercicio, también cargado desde el servidor, los otros sonidos estarán incluidos en la aplicación ya que son de uso común a todos los ejercicios y sería un desperdicio de recursos cargarlos desde el servidor a cada uso. El diagrama a continuación aclara esta idea.

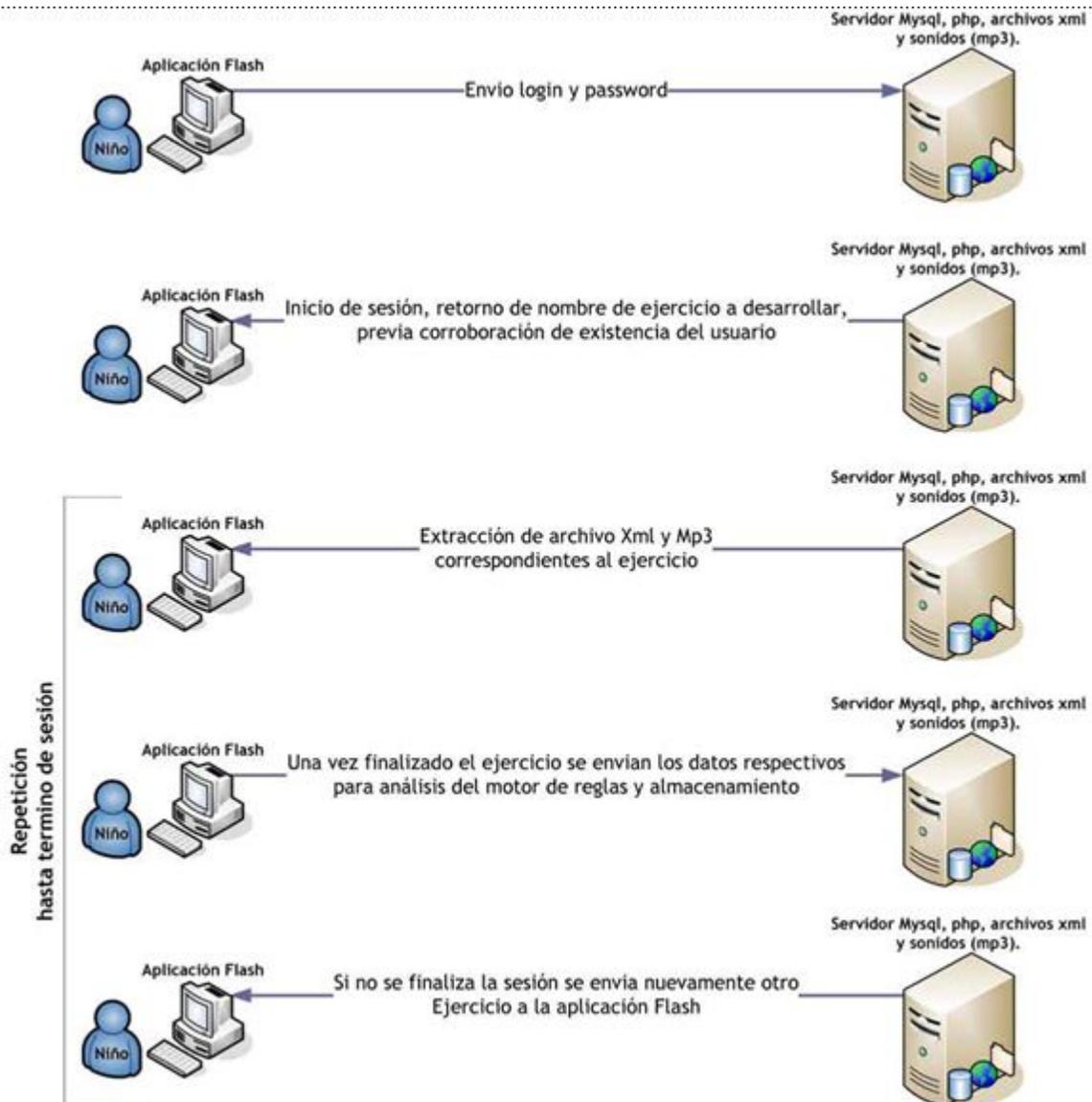


Ilustración 5.26 Comunicación entre servidor y aplicación del alumno.

Hay que mencionar que también se tomó la decisión de que al momento de que el alumno use la aplicación, esta irá presentando un ejercicio de cada actividad, formando un ciclo.

Implementación.

Se codificó en ActionScript (Flash) la carga de XML. Además de la carga de sonidos desde él. Además se crearon nuevos archivos PHP para habilitar la identificación al inicio.

Prueba.

Se probaron archivos XML para cargar y la aplicación extrajo correctamente la información contenida. También se probó la carga de archivos de sonido desde el servidor, funcionando a la perfección.

5.12.2.3. Iteración “Interfaz gráfica”

Análisis.

Se reunió información sobre las características gráficas que debiera tener la aplicación, en cuanto a colores, formas, etc. Además se observaron los software provistos por la Escuela Santa Lucía, buscando reunir las mejores características.

Diseño.

Se definieron las características gráficas que estarían presentes en la interfaz de la aplicación.

Implementación.

De acuerdo a lo diseñado se creó la interfaz gráfica en la aplicación.

Prueba.

La única forma de probar el diseño gráfico de la interfaz fue contra las indicaciones del material usado y la consulta al profesor encargado de la Escuela Santa Lucía, quien aprobó el diseño gráfico realizado.

5.12.2.4. Iteración “Integración de archivos XML”

Análisis.

Ya tomada la decisión de almacenar la información de los ejercicios, se analizó la problemática de darles un formato específico para cada tipo de actividad, dado que cambia el tipo de información de los ejercicios por actividad.

Diseño.

Se decidió la estructura de los archivos XML para cada tipo de actividad.

Implementación.

Se agregó a la aplicación del profesor la capacidad de generar los archivos XML con las diversas estructuras. Paralelamente se codificó en la aplicación del alumno la funcionalidad de extraer información de los archivos XML para cada tipo de actividad.

Prueba.

Se cargaron en la aplicación del alumno todos los tipos de archivos XML, resultando exitosa la prueba en todos los casos.

5.12.2.5. Iteración “Interfaz audible”

Análisis.

Para completar la aplicación sólo faltaba agregar la interfaz audible. El primer paso fue tomar la decisión de cómo obtener los sonidos para las respuestas, cumpliendo con los requerimientos de que fuera una voz amistosa y clara.

Diseño.

Se optó por utilizar una voz grabada desde una persona en lugar de una generada por algún software, teniendo que diseñar el algoritmo que construyera el sonido completo de la alternativa o respuesta de forma dinámica a partir de archivos de sonido grabados por separado. Esta lista de sonidos también fue creada durante esta actividad para que fuera grabada por la persona a cargo.

Implementación.

La lista de sonidos fue grabada por la persona a cargo y se codificó el algoritmo mencionado, integrándolo posteriormente a la aplicación. Además se codificó la carga del enunciado desde el servidor.

Prueba.

Se cargaron ejercicios de de las distintas actividades y la interfaz audible funcionó según lo esperado. La voz y la interfaz fueron aprobadas por el profesor de la Escuela Santa Lucía.

5.12.2.6. Iteración “Navegación por teclado”

Análisis.

A las interfaces gráfica y audible es necesario agregar la navegación por teclado, para que los alumnos invidentes y de baja visión naveguen por las alternativas de respuesta.

Diseño.

Se recogió información sobre software similar y sobre que teclas utilizan, decidiendo así cuales son las idóneas e intuitivas para los alumnos.

Implementación.

Se codificó el uso de teclas para la navegación por las alternativas, repetición del sonido de enunciado del ejercicio, escoger opción como respuesta, salir de la aplicación y confirmar o cancelar salida de la aplicación.

Prueba.

Se probó la totalidad de comandos por teclado implementados al igual que combinaciones que pudieran llevar a algún mal funcionamiento, siendo superadas estas pruebas.

5.12.2.7. Iteración “Motor de reglas”

Análisis.

Ya definidas las variables a usar por el motor de reglas y también el funcionamiento solo faltaba agregarlo a la aplicación y finalizar las transacciones a la base de datos por medio de los archivos PHP.

Diseño.

El diseño del motor de reglas estaba preparado con anterioridad.

Implementación.

Se codificaron las reglas descritas en el diseño del motor de reglas.

Prueba.

Se realizaron ejercicios en la aplicación dando respuestas acertadas y erradas, corroborando finalmente en la base de datos la correcta actualización de los datos de desempeño.

5.12.2.8. Interfaces de la aplicación del alumno

Al iniciar la aplicación el alumno se encontrará con la siguiente pantalla, en la cual debe ingresar el usuario y password, siendo el nombre de usuario los dos nombres seguidos sin espacio y el password son los dos apellidos de la misma forma.

La imagen muestra una pantalla de inicio de sesión con un fondo naranja. En el centro, hay un campo de texto blanco con el label "Usuario" encima. Debajo de este campo, hay otro campo de texto blanco con el label "Password" encima. En la parte inferior, hay un botón amarillo con un borde negro y el texto "Ingresar" en negro.

Ilustración 5.27 Identificación para acceso a la aplicación.

Una vez logueado se comienza con el desarrollo de los distintos tipos de ejercicios, durante la carga de cada ejercicio se mostrara el siguiente mensaje.

CARGANDO EJERCICIO

Ilustración 5.28 Carga de un ejercicio.

SECUENCIA DE NUMEROS
¿Cual es la secuencia correcta entre 1 y 6
sumando 1 cada vez?

2 4 5 6

 2 3 4 5

3 4 5 6

5 4 3 2

Ilustración 5.29 Ejercicio para el alumno.

La tecla de escape llevará a una pantalla de confirmación de salida, mostrada a continuación.



DESEAS SALIR?

Ilustración 5.30 Salir de la aplicación.

5.13. Pruebas

Las pruebas son de gran importancia en la garantía de la calidad del software. Los objetivos principales de realizar una prueba son:

- Detectar un error.
- Tener un buen caso de prueba.
- Descubrir un error no descubierto antes.

Los métodos de prueba del software tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo.

Principios de la prueba:

- Hacer un seguimiento de las pruebas hasta los requisitos del cliente.
- Plantear y diseñar las pruebas antes de generar ningún código.
- El 80% de todos los errores se centran en solo en el 20% de los módulos.
- Empezar las pruebas en módulos individuales y avanzar hasta probar el sistema entero.
- No son posibles las pruebas exhaustivas.

- Deben realizarse por un equipo independiente al equipo de desarrollo.

Un software fácil de probar tiene las siguientes características:

- Operatividad.
- Objetividad.
- Controlabilidad.
- Capacidad de descomposición.
- Simplicidad.
- Estabilidad.
- Facilidad de comprensión.

Atributos de una buena prueba:

- Más alta probabilidad de encontrar un error.
- No debe ser redundante.
- No debería ser ni demasiado sencilla ni demasiado compleja.

Enfoques de Pruebas.

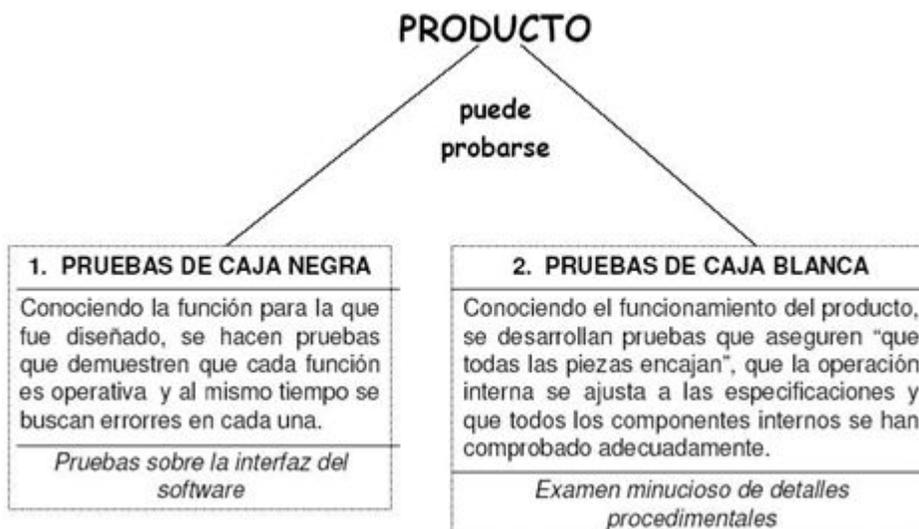


Ilustración 5.31 Enfoques de Pruebas.

Prueba de Caja Negra.

Los datos de prueba se escogerán atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.

Criterios mínimos que guiarán al escoger los datos de prueba:

- Valores Fáciles: El programa se depurará con datos de fácil comprobabilidad.
- Valores típicos realistas: se ensayará un programa con datos seleccionados para que representen como se aplicará. Los Datos han de ser sencillos, de modo que los resultados sean verificables en forma manual.
- Valores extremos.
- Valores ilegales: Cuando en un programa entra basura, su salida habrá de ser un mensaje de error adecuado. Es preferible que el programa ofrezca indicación de errores en la entrada y que realice cálculos que sigan siendo factibles luego de desechar la entrada equivocada.

El método de la caja negra se centra en los requisitos fundamentales del software y permite obtener entradas que prueben todos los requisitos funcionales del programa.

Con este equipo de pruebas se intenta encontrar:

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

Con la aplicación de esa técnica se obtiene un conjunto de pruebas que:

- Reduce el número de casos de pruebas y nos dicen algo sobre la presencia o ausencia de errores.

Partición equivalente.

Una partición equivalente es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos. El diseño de casos de prueba para la partición equivalente se basa en la evaluación de las clases de equivalencia.

Análisis de valores límite.

Nos lleva a elegir las pruebas que nos ejecuten los valores límite, con esta técnica se complementa la partición equivalente.

Prueba de comparación.

Esta técnica consiste en la comparación de salidas de un mismo software pero de sus diferentes versiones.

Prueba de Caja Blanca.

Establece la observación de que un programa difícilmente puede considerarse como probado por completo si su código contiene partes que nunca han sido ejecutadas. Este método analiza la estructura lógica del programa y, para cada alternativa que puede presentarse, los datos de prueba ideados conducirán a ella. Se procura escoger los que verifiquen cada posibilidad en las proposiciones case, las cláusulas de cada proposición **if** y la condición de terminación de cada ciclo.

En un programa extenso, este método es impráctico, pero en un modulo pequeño constituye un excelente medio de prueba y depuración.

En una prueba que se vale del método de caja blanca, se tornan patentes las ventajas de un diseño de programa modular.

Un buen criterio de prueba para proyectos extensos consiste en aplicar los métodos de caja blanca a cada módulo pequeño conforme se escriba; luego se usan esos datos en las secciones más amplias del programa una vez terminadas.

Prueba de Caja de Pandora.

Consiste en absterse de realizar pruebas de depurar bastante bien un proyecto; se deja al cliente que lo ensaye y acepte. El resultado es una bomba de tiempo.

Pruebas de Integración.

Las pruebas de integración se llevan a cabo durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto. Estas pruebas se pueden plantear desde un punto de vista estructural o funcional.

Las pruebas estructurales de integración son similares a las pruebas de caja blanca; pero trabajan a un nivel conceptual superior. En lugar de referirse a sentencias del lenguaje, se refieren a llamadas entre módulos. Se trata pues de identificar todos los posibles esquemas de llamadas y ejercitarlos para lograr una buena cobertura de segmentos o de ramas.

Las pruebas funcionales de integración son similares a las pruebas de caja negra. Aquí se trata de encontrar fallos en la respuesta de un módulo cuando su operación depende de los servicios prestados por otro módulo. A medida que se acerca al sistema total, estas pruebas se van basando más y más en la especificación de requisitos del usuario.

Las pruebas finales de integración cubren todo el sistema y pretenden cubrir plenamente la especificación de requisitos del usuario. Además, a estas alturas ya suele estar disponible el manual de usuario, que también se utiliza para realizar pruebas hasta lograr una cobertura aceptable.

Pruebas de Aceptación.

Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo, pues sería impresentable de cara al cliente; sino una vez pasada todas las pruebas de integración por parte del desarrollador.

La experiencia muestra que aún después del más cuidadoso proceso de pruebas por parte del desarrollador, quedan una serie de errores que sólo aparecen cuando el cliente se pone a usarlo.

Conclusión.

Las pruebas del software se presentan en el Anexo F, se ha escogido desarrollar pruebas de caja negra por tratarse del más indicado de acuerdo al tamaño del proyecto.

5.14. Validación en Escuela Santa Lucía

Como lo indica la Ingeniería en Software, la validación proporciona una seguridad final de que el software satisface todos los requisitos funcionales, de comportamiento y de rendimiento, es decir, si se han satisfecho realmente los requerimientos del cliente [Pressman, 2002]. Dado que este proyecto presentaba diversas complejidades en los requerimientos, se estableció como objetivo específico efectuar la validación en la Escuela Santa Lucía, ya que los profesores y alumnos de este establecimiento son los usuarios finales y por ende los más apropiados para llevar a cabo esta tarea.

Para la validación se hizo uso de las pruebas alfa y beta. La prueba alfa se realiza con un cliente quien opera de forma natural el sistema, en el ambiente de desarrollo, bajo la observación del desarrollador, quien registra los errores y problemas de uso. La prueba beta es llevada a cabo por los usuarios finales del sistema en el lugar donde será utilizado normalmente, no controlado por el desarrollador, quien tampoco esta presente para observar la interacción del sistema y los usuarios. En el caso de la prueba beta, la responsabilidad de registrar errores, ya sean reales o imaginarios, recae en el cliente, quién informa periódicamente al desarrollador los resultados de la prueba.

Debido a las dificultades en el contacto con el cliente de forma personal, se decidió realizar la prueba alfa con niños de visión normal, para probar los contenidos de los ejercicios y si se asimilaba intuitivamente el funcionamiento de la aplicación. Es decir, se probó el contenido, dejando para la prueba beta la validación de las interfaces audible y gráfica.

Como ya se describió anteriormente la prueba beta consiste en recopilar información desde el cliente al usar el sistema en el ambiente definitivo que tendrá este. Como el sistema tenía que cubrir necesidades de pedagogía, interfaces audible y gráfica y las propias para cualquier sistema, se investigó acerca de experiencias anteriores en el desarrollo de sistemas de similares características que se han realizado. Si bien existen ya trabajos realizados, queda un largo camino aún, ya que el uso de la informática en la educación implica la unión de muchos aspectos, partiendo por la ingeniería y la pedagogía, sumado a ellos alguna necesidad

adicional dependiendo del tipo de usuarios final, en el caso presente de niños invidentes y de baja visión.

En general los instrumentos más utilizados son los cuestionarios de valoración, donde las respuestas a estos cuestionarios son valoradas entre 0 y 5, por ejemplo, siendo el resultado el grado de conformidad del usuario con las afirmaciones propuestas. Los instrumentos de evaluación, en forma de planillas se deben confeccionar con inclusión no sólo de preguntas del tipo cerradas, sino también de preguntas abiertas, y casillas de verificación, permitiendo al usuario final la descripción de aspectos problemáticos y particulares del programa que no hayan sido tenidos en cuenta durante la confección del instrumento.

Se deberá tener en cuenta al redactar los cuestionarios, la utilización de un vocabulario adecuado, sin ambigüedades y claro para los destinatarios previstos en cada caso en particular. En la mayor parte de los cuestionarios relevados se consideran algunos aspectos claves o sobresalientes: como el logro de los objetivos, los aspectos técnicos, el desarrollo de contenidos, actividades y la documentación.

5.14.1. Cuestionario de valoración

Como la validación corresponde a determinar si los requerimientos fueron satisfechos a cabalidad, para el diseño del cuestionario de validación se han tomado en cuenta estos y además se han añadido puntos relevantes mencionados en diversos trabajos relacionados al tema educativo. Entonces, para el diseño del cuestionario de valoración se han considerado:

- Requerimientos del sistema
- Modelo para entornos interactivos basados en sonido para aprender [mTISA, 2003].
- Ingeniería en software educativo [Lage].
- Metodología de diseño, desarrollo y evaluación de software educativo [Cataldi, 2000].
- Un modelo de evaluación de software educativo para la enseñanza de la matemática [Hernández, 2007].
- Escala de evaluación de material multimedia para personas con Síndrome de Down [Ortega], solo algunos conceptos interesantes y aplicables al caso.

- Escala de evaluación de software educativo [Evaluación Software Educativo].

Como se puede observar son variadas las fuentes pero también son varios los tipos de requerimientos. A continuación se presenta la tabla de valoración construida con los resultados obtenidos.

Tabla 5.10 Cuestionario de valoración.

Aspecto a valorar	Muy bueno	Bueno	Normal	Malo	Muy malo
Manual de uso					
Número de Páginas	X				
Calidad del lenguaje		X			
Precisión de las instrucciones		X			
Autosuficiencia del texto		X			
Apoyo gráfico	X				
Accesibilidad de la información	X				
Ejemplificaciones			X		
Diseño educativo					
Permite la experimentación	X				
Información ante errores	X				
Guía el aprendizaje		X			
Manejo fácil	X				
Capacidad de adicción	X				

Motiva/Estimula		X			
Concentra la atención sobre la tarea			X		
Facilita el aprendizaje		X			
Individualiza el aprendizaje (ofrece diferentes niveles)	X				
Ingreso, eliminación y eliminación de ejercicios	X				
Información de uso de cada usuario		X			
Las actividades son adecuadas a los contenidos		X			
Estructura adecuada para la presentación de los ejercicios	X				
Adecuada definición de actividades		X			
Diseño Funcional					
Facilidad de uso		X			
Accesibilidad		X			
Presentación de datos	X				
Calidad de los reportes	X				
Gestión de ejercicios	X				
Gestión de usuarios	X				
Gestión de respaldos	X				

Interfaz Gráfica					
La estructura visual es adecuada	X				
Los colores son adecuados para la baja visión		X			
Distinción entre los elementos de la interfaz		X			
Se resaltan los elementos importantes o en uso		X			
El tipo de letra es legible para la baja visión	X				
El tamaño de la letra es adecuado	X				
La distribución de los elementos es adecuada	X				
La distancia entre elementos no produce agobio	X				
Interfaz audible					
El sonido es claro	X				
La pronunciación es fácilmente identificable (del sonido integrado)		X			
La voz es amigable y familiar (del sonido integrado)		X			
El lenguaje utilizado es el adecuado	X				
Los enunciados y opciones se repiten siempre que el alumno lo requiera	X				
Se informa del estado de la aplicación		X			
Se comunica resultado en la ejecución de ejercicio	X				

Se dan señales necesarias para navegar a través de las opciones	X				
Manejo por teclado					
Se usan teclas ya conocidas y comunes a otros software utilizado por los niños	X				
Las teclas son de uso intuitivo	X				
El alumno no debe recordar un gran número de teclas	X				
La aplicación puede ser manejada totalmente por el teclado	X				
Las teclas no usadas (al presionarse por accidente) no interfieren el funcionamiento de la aplicación	X				

5.14.2. Conclusiones

Como se puede observar los resultados son satisfactorios, ya que el nivel más bajo alcanzado y solo en pocas ocasiones, es el de Normal. Lo que nos indica que los requerimientos han sido satisfechos casi en su totalidad. Este resultado es bastante bueno por las complicaciones que representaban las diversas características que debía tener el sistema.

Sin embargo, se tiene muy claro que aún falta camino por recorrer en futuros requerimientos, ya que es tema de la Escuela Santa Lucía el definir la participación que tendrá el sistema en la educación y a partir de eso formular las necesidades nuevas que debiera abarcar el sistema.

6. Conclusiones

Seguramente si se comparan las herramientas y lenguajes, en número y características, a las de otro proyecto informático, se concluirá que este proyecto es bastante especial. Definitivamente lo es, ya que involucró el aprendizaje y uso de tecnologías poco habituales en el desarrollo de sistemas, como lo es Flash por ejemplo, el cual tiene como fin la creación de animaciones, pero cuenta con un gran potencial para el uso informático, ya que posee un lenguaje de programación muy similar a Java Script, llamado ActionScript. Este lenguaje posee las mismas características de lenguajes tradicionales, adicionando el manejo de comportamientos del contenido de una animación. Otra tecnología utilizada y quizás poco común es XML, la cual facilitó el almacenamiento de cierta información del sistema y a la vez disminuir la carga sobre la base de datos sin sacrificar disponibilidad, agregando modularidad.

Siguiendo con el análisis, también se hizo uso de lenguajes mas tradicionales como Java, que fue utilizado para la construcción de la aplicación del profesor, la que explotó las variadas capacidades del lenguaje, como el envío de mails, grabación de sonidos y escritura/lectura de archivos XML

La base de datos utilizada fue MySQL, la cual es gratuita para la mayoría de sus usos pero no por ello menos eficiente en su desempeño. En este punto se debe mencionar PHP, que participó como intermediario entre Flash y la base de datos. Por último se utilizó también el servidor web Apache, el cual pesar de ser gratuito tiene muy buenas características.

Cada una de estas herramientas y lenguajes fue escogida porque representaba ventajas o simplemente eran las más idóneas. Pero también esta variedad representa un riesgo a la culminación exitosa del proyecto, ya que hay que dedicar tiempo al aprendizaje, lo que divide los esfuerzos en varias tareas.

Siguiendo en la línea de los riesgos se debe mencionar además la necesidad de impregnarse de lo que es la pedagogía en educación básica. Un tema totalmente opuesto a la formación de

un ingeniero, por lo cual se tuvo que dedicar bastante tiempo para interiorizarse de conceptos y contenidos de las bases curriculares dadas por el Ministerio de Educación.

Para finalizar con el tema de los riesgos, también se debe mencionar las características especiales con las que debe cumplir un sistema que interactúe con personas ciegas o de baja visión, como lo son la facilidad de uso, navegabilidad, robustez, etc., implicando que se tenga que trabajar en diseño de interfaces audibles y gráfica, colores, estilos de letra y control por teclado.

Por lo tanto podemos decir que los riesgos para este proyecto estaban representados por el aprendizaje de: lenguajes o herramientas desconocidas, aspectos pedagógicos para el diseño de los ejercicios y diseño de interfaces audibles y gráficas para personas ciegas y de baja visión. Estos riesgos podrían llevar a que no se cumpla con los requerimientos iniciales o se vea afectada la calidad del sistema. Para no caer en las situaciones anteriores es necesario seleccionar un proceso de desarrollo de software.

Para este proyecto se ha decidido hacer uso del Proceso Unificado y el modelo cascado o clásico. Mas específicamente la idea fue la de desarrollar la aplicación para el alumno, la que reviste mayor riesgo, siguiendo el modelo iterativo incremental y la aplicación del profesor siguiendo el modelo cascada. Usando para ambas aplicaciones los productos o artefactos que propone el Proceso Unificado.

Toda esta planificación, definición de actividades y productos asociados, no son suficientes para garantizar el éxito del proyecto, también nos dimos cuenta de lo importante de una buena comunicación con el cliente, de la diferencia que provoca demostrar disposición a escuchar y proponer. No se pretende decir que todo fue perfecto y sin problemas, porque los hubieron, pero lo verdaderamente importante es que pudieron ser superados.

Pasando al análisis del software en sí, se puede afirmar que se ha dado una correcta solución al problema planteado, e incluso ante posibles nuevos requerimientos, dado que siempre se tuvo en mente la escalabilidad del sistema, se podría fácilmente agregar funcionalidades y expandir el alcance de su funcionamiento, pensando incluso en hacer uso de una red mas amplia como lo es Internet, para llegar a niños que no estén en posibilidad de asistir a la Escuela Santa Lucía.

Sin embargo, el sistema aún puede ser mejorado para incrementar su utilidad y beneficios, queda pendiente la tarea de hacerlo evolucionar, tarea que ojala en algún momento apoye alguna institución.

7. Referencias

[Apache]	www.apache.org
[Cataldi, 2000]	Cataldi, Zulma. Tesis: "Metodología de diseño, desarrollo y evaluación de software educativo"
[DuBois, 2000]	DuBois, Paul. Libro: MySQL (2000).
[Evaluación Software Educativo]	http://www.unex.es/didactica/Tecnologia_Educativa/info05A.htm/
[Hernández, 2007]	Hernandez, Victor. "Un modelo de evaluación de software educativo para la enseñanza de Matemática"
[Jacobson, 2000]	Jacobson, Ivar y Booch, Grady y Rumbaugh, James Pearson Educación. Madrid. España, año 2000, Proceso Unificado de Desarrollo de Software.
[Jesús Alfonso López, 2000]	Tripod, "Home page sobre Algoritmos Genéticos", 2000, http://members.tripod.com/jesus_alfonso_lopez/AgIntro.html
[Kung, 1993]	"DIGITAL NEURAL NETWORKS", S. Y. Kung, 1993 by PTR Prentice Hall, Inc.
[Lage]	Lage, "Ingeniería de Software Educativo"
[Mellado, 2006]	Mellado, Pablo. Proyecto: "Agentes Inteligentes para el desarrollo lógico-matemático infantil" (2006).
[Ministerio de Educación, 2002]	Ministerio de Educación Gobierno de Chile, Objetivos fundamentales y contenidos mínimos obligatorios de la educación

	básica (actualización 2002).
[MySQL]	http://www.MySql.com/
[mTISA, 2003]	Sánchez Jaime, Baloian Nelson y Flores Héctor, “Modelo para Entornos Interactivos basados en sonido para aprender”, 8° Taller internacional de software educativo, Universidad de Chile.
[NetBeans]	www.netbeans.org/
[Ortega]	Ortega, Juana M ^a . “Escala de evaluación de material multimedia para personas con Síndrome de Down”
[Pressman, 2002]	Pressman, Roger S. Libro: Ingeniería del software: un enfoque práctico. (5a. ed.), McGraw Hill. Madrid. España, año 2002
[Resnikoff, 2003]	Dr. Serge Resnikoff, Coordinador del Programa de Prevención de la Ceguera y la Sordera, Manejo de las Enfermedades No Transmisibles, OMS.

8. Glosario de Términos

Apache: Servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

Ceguera: Agudeza visual inferior a 3/60 (o equivalente) en el mejor ojo con la mejor corrección posible, o restricción del campo visual de cada ojo a 10 grados del punto de fijación.

Jaws for Windows: Software lector de pantallas para invidentes.

MySQL: Sistema de gestión de base de datos relacional, multihilo y multiusuario.

PHPMyAdmin: es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas webs, utilizando Internet.

UNESCO: Organización de las Naciones Unidas para la educación, la ciencia y la cultura.

9. Lista de Abreviaturas o Siglas

BDR: Base de Datos Relacional.

FONADIS: Fondo Nacional de la Discapacidad.

IP: Internet Protocol.

JIT: Just in Time.

JVM: Java Virtual Machine.

MIDEPLAN: Ministerio de Planificación y Cooperación.

NB: Nivel Básico.

OCR: Reconocimiento óptico de caracteres.

OMS: Organización Mundial de la Salud.

ONCE: Organización Nacional de Ciegos Españoles.

TCP: Transmission Control Protocol.

TIC: Tecnología de Información y Comunicación.

UML: Lenguaje Unificado de Modelado.

UP: Proceso Unificado.

XML: eXtensible Markup Language.

Anexo A.

Especificación de Casos de Uso.

Tabla A.1 Caso de uso crear usuario.

Caso de Uso	Crear Usuario.
Objetivo	Este caso de uso tiene como objetivo la creación de un nuevo usuario en el Sistema, especificando el tipo de usuario, el cual determinará sus privilegios.
Actor Primario	Administrador.
Precondiciones	El Administrador ha sido identificado y autenticado en el sistema.
Poscondiciones	El nuevo usuario es agregado al sistema con sus respectivos permisos.
Escenario Principal (Flujo Básico)	<ol style="list-style-type: none"> 1. El Administrador solicita creación de nuevo usuario. 2. El Administrador comienza la creación del nuevo usuario. 3. El Administrador selecciona el tipo de cuenta e ingresa el Rut del usuario y los datos requeridos por el Sistema. 4. El Sistema da aviso de la correcta creación del usuario. <p>El Administrador puede volver al paso 2 hasta indicar término.</p>
Escenarios Alternativos (extensiones)	<p>3a.- Rut del usuario ya existe.</p> <ol style="list-style-type: none"> 1.- El Sistema da aviso de existencia del mismo Rut. 2.- El Administrador debe ingresar nuevo Rut de usuario o verificar que no exista ya el usuario en el Sistema. <p>4a.- Falla en el ingreso de password.</p> <ol style="list-style-type: none"> 1.- El Administrador debe ingresar nuevamente la password.

Frecuencia de Ocurrencia	Baja.
---------------------------------	-------

Tabla A.2 Caso de uso eliminar usuario.

Caso de Uso	Eliminar Usuario.
Objetivo	Este caso de uso tiene como objetivo la eliminación de un usuario en el Sistema.
Actor Primario	Administrador.
Precondiciones	El Administrador ha sido identificado y autenticado en el sistema. El usuario a eliminar debe existir en el Sistema.
Poscondiciones	El usuario ha sido eliminado del Sistema.
Escenario Principal (Flujo Básico)	<ol style="list-style-type: none"> 1. El Administrador solicita eliminar un usuario. 2. El Administrador comienza la eliminación de un usuario. 3. El Administrador selecciona el usuario a eliminar de una lista de usuarios del Sistema. 4. El Administrador solicita al Sistema la eliminación del usuario. 5. El Sistema pide confirmación de eliminación. 6. El Administrador ingresa confirmación y el Sistema procede. <p>El Administrador puede volver al paso 2 hasta indicar término.</p>
Escenarios Alternativos	5a.- El Administrador cancela la operación.

(extensiones)	1.- El Sistema no realiza la operación iniciada.
Frecuencia de Ocurrencia	Baja.

Tabla A.3 Caso de uso modificar usuario.

Caso de Uso	Modificar Usuario.
Objetivo	Este caso de uso tiene como objetivo la modificación de alguno o todos los datos de algún usuario del Sistema.
Actor Primario	Administrador.
Precondiciones	El Administrador ha sido identificado y autenticado en el sistema. El usuario a actualizar debe existir en el Sistema.
Poscondiciones	Registro del usuario actualizado.
Escenario Principal (Flujo Básico)	<ol style="list-style-type: none"> 1. El Administrador solicita la modificación de un usuario. 2. El Administrador comienza la modificación de un usuario. 3. El Administrador selecciona el tipo de usuario. 4. El Administrador selecciona desde una lista de usuarios del Sistema. 5. El Sistema muestra el detalle del usuario. 6. El Administrador modifica la información del usuario. 7. El Administrador indica término de modificación.

	<p>8. El Sistema pide confirmación de modificación.</p> <p>9. El Administrador confirma la modificación.</p> <p>10. El Sistema modifica y actualiza la información del usuario.</p> <p>El Administrador puede volver al paso 2 hasta indicar término.</p>
<p>Escenarios Alternativos (extensiones)</p>	<p>7a.- El Sistema notifica la falta de algunos datos.</p> <p>1.- El Administrador ingresa todos los datos requeridos por el Sistema.</p> <p>8a.- El Administrador cancela la operación.</p> <p>1.- El Sistema no realiza la operación iniciada.</p>
<p>Frecuencia de Ocurrencia</p>	<p>Baja.</p>

Tabla A.4 Caso de uso crear ejercicio.

<p>Caso de Uso</p>	<p>Crear Ejercicio.</p>
<p>Objetivo</p>	<p>Este caso de uso tiene como objetivo la creación de un nuevo ejercicio en el Sistema, especificando la actividad y el ranking que se le asignará.</p>
<p>Actor Primario</p>	<p>Profesor.</p>
<p>Precondiciones</p>	<p>El Profesor ha sido identificado y autenticado en el sistema.</p>
<p>Poscondiciones</p>	<p>El nuevo ejercicio es agregado al sistema.</p>

Escenario Principal (Flujo Básico)	<ol style="list-style-type: none"> 1. El Profesor solicita creación de un nuevo ejercicio. 2. El Profesor comienza la creación del nuevo ejercicio. 3. El Profesor selecciona el tipo de actividad e ingresa los datos requeridos por el Sistema. 4. El Sistema da aviso de la correcta creación del ejercicio. <p>El Profesor puede volver al paso 2 hasta indicar término.</p>
Escenarios Alternativos (extensiones)	<p>3a.- Falla al ingresar los datos.</p> <p>1.- El Profesor debe ingresar nuevamente los datos pedidos por el Sistema.</p>
Frecuencia de Ocurrencia	Baja.

Tabla A.5 Caso de uso eliminar ejercicio.

Caso de Uso	Eliminar Ejercicio.
Objetivo	Este caso de uso tiene como objetivo la eliminación de un ejercicio del Sistema.
Actor Primario	Profesor.
Precondiciones	<p>El Profesor ha sido identificado y autenticado en el sistema.</p> <p>El ejercicio a eliminar debe existir en el Sistema.</p>
Poscondiciones	El ejercicio ha sido eliminado del Sistema.
Escenario Principal	1. El Profesor solicita eliminar un ejercicio.

(Flujo Básico)	<p>2. El Profesor comienza la eliminación de un ejercicio.</p> <p>3. El Profesor selecciona el ejercicio a eliminar de una lista de ejercicios del Sistema.</p> <p>4. El Profesor solicita al Sistema la eliminación del ejercicio.</p> <p>5. El Sistema pide confirmación de eliminación.</p> <p>6. El Profesor ingresa confirmación y el Sistema procede.</p> <p>El Profesor puede volver al paso 2 hasta indicar término.</p>
Escenarios Alternativos (extensiones)	<p>5a.- El Profesor cancela la operación.</p> <p>1.- El Sistema no realiza la operación iniciada.</p>
Frecuencia de Ocurrencia	Baja.

Tabla A.6 Caso de uso modificar ejercicio.

Caso de Uso	Modificar Ejercicio.
Objetivo	Este caso de uso tiene como objetivo la modificación de algún ejercicio.
Actor Primario	Profesor.
Precondiciones	<p>El Profesor ha sido identificado y autenticado en el sistema.</p> <p>El ejercicio a actualizar debe existir en el Sistema.</p>
Poscondiciones	Ejercicio actualizado.

<p>Escenario Principal</p> <p>(Flujo Básico)</p>	<ol style="list-style-type: none"> 1. El Profesor solicita la modificación de un ejercicio. 2. El Profesor comienza la modificación de un ejercicio. 3. El Profesor selecciona una actividad. 4. El Profesor selecciona desde una lista de ejercicio. 5. El Sistema muestra el detalle del ejercicio. 6. El Profesor modifica la información del ejercicio. 7. El Profesor indica término de modificación. 8. El Sistema pide confirmación de modificación. 9. El Profesor confirma la modificación. 10. El Sistema modifica y actualiza la información del ejercicio. <p>El Profesor puede volver al paso 2 hasta indicar término.</p>
<p>Escenarios Alternativos</p> <p>(extensiones)</p>	<p>6a.- El Sistema notifica la falta de algunos datos.</p> <p>1.- El Profesor ingresa todos los datos requeridos por el Sistema.</p> <p>8a.- El Profesor cancela la operación.</p> <p>1.- El Sistema no realiza la operación iniciada.</p>
<p>Frecuencia de Ocurrencia</p>	<p>Baja.</p>

Tabla A.7 Caso de uso realizar actividad.

Caso de Uso	Realizar Actividad.
Objetivo	Este caso de uso tiene como objetivo la realización de una actividad.
Actor Primario	Alumno.
Precondiciones	El Alumno ha sido identificado y autenticado en el sistema.
Poscondiciones	Se ha cargado la actividad satisfactoriamente y guardado los datos de la actividad realizada.
Escenario Principal (Flujo Básico)	<ol style="list-style-type: none"> 1. El Sistema selecciona un ejercicio de acuerdo al ranking actual del alumno. 2. El Sistema carga el ejercicio seleccionado. 3. El Alumno procede a realizar el ejercicio. 4. El Sistema almacena los datos. 5. El Alumno solicita terminar. 6. El Sistema pide confirmación. <p>El Alumno confirma y el Sistema procede.</p>
Escenarios Alternativos (extensiones)	<p>2a.- No hay ejercicios disponibles.</p> <p>1.- El Sistema muestra mensaje de error y vuelve al paso 1.</p> <p>4a.- Error al cargar el ejercicio seleccionado.</p> <p>1.- El Sistema envía mensaje correspondiente y vuelve al paso 3.</p>
Frecuencia de	Alta.

Ocurrencia	
-------------------	--

Tabla A.8 Caso de uso alumnos mejores rankeados.

Caso de Uso	Alumnos mejor rankeados.
Objetivo	Este caso de uso tiene como objetivo mostrar los alumnos con mejor ranking.
Actor Primario	Profesor.
Precondiciones	El Profesor ha sido identificado y autenticado en el sistema.
Poscondiciones	Se han mostrado los alumnos.
Escenario Principal (Flujo Básico)	<ol style="list-style-type: none"> 1. El Profesor solicita el informe de alumnos con mejor ranking. 2. El Sistema muestra la información. <p>El Profesor puede volver al paso 1 hasta indicar término.</p>
Escenarios Alternativos (extensiones)	<p>1a.- Error de conexión.</p> <ol style="list-style-type: none"> 1. - Se muestra mensaje correspondiente y se pide realizar la operación en otro momento.
Frecuencia de Ocurrencia	Media-Baja.

Tabla A.9 Caso de uso alumnos peor rankeados.

--	--

Caso de Uso	Alumnos peor rankeados.
Objetivo	Este caso de uso tiene como objetivo mostrar los alumnos con peor ranking.
Actor Primario	Profesor.
Precondiciones	El Profesor ha sido identificado y autenticado en el sistema.
Poscondiciones	Se han mostrado los alumnos.
Escenario Principal (Flujo Básico)	1. El Profesor solicita el informe de alumnos con peor ranking. 2. El Sistema muestra la información. El Profesor puede volver al paso 1 hasta indicar término.
Escenarios Alternativos (extensiones)	1a.- Error de conexión. 1. - Se muestra mensaje correspondiente y se pide realizar la operación en otro momento.
Frecuencia de Ocurrencia	Media-Baja.

Tabla A.10 Caso de uso informe ejercicios por alumno.

Caso de Uso	Informe Ejercicios por alumno.
Objetivo	Este caso de uso tiene como objetivo mostrar los ejercicios realizador por un Alumno.
Actor Primario	Profesor.

Precondiciones	El Profesor ha sido identificado y autenticado en el sistema.
Poscondiciones	Se ha mostrado la información.
Escenario Principal (Flujo Básico)	<ol style="list-style-type: none"> 1. El Profesor solicita el informe de ejercicios por alumno. 2. El Sistema muestra una lista de los alumnos. 3. El Profesor selecciona un alumno. 4. El Sistema muestra la información. <p>El Profesor puede volver al paso 1 hasta indicar término.</p>
Escenarios Alternativos (extensiones)	<p>1a.- Error de conexión.</p> <ol style="list-style-type: none"> 1. - Se muestra mensaje correspondiente y se pide realizar la operación en otro momento.
Frecuencia de Ocurrencia	Alta.

Tabla A.11 Caso de uso login.

Caso de Uso	Login.
Objetivo	Realizar el login del usuario en el sistema.
Actor Primario	Profesor, Administrador y Alumno.
Precondiciones	
Poscondiciones	El usuario ha sido identificado y autenticado en el sistema.

Escenario Principal (Flujo Básico)	1. El usuario ingresa su nombre de usuario y contraseña El Sistema identifica al usuario.
Escenarios Alternativos (extensiones)	1a.- Datos inválidos. 1.- Se muestra el mensaje correspondiente y se pide que ingrese nuevos datos. 1b.- Error de conexión. 1. - Se muestra mensaje correspondiente y se pide realizar la operación en otro momento.
Frecuencia de Ocurrencia	Alta.

Tabla A.12 Caso de uso actividades con mayor dificultad.

Caso de Uso	Actividades con mayor dificultad.
Objetivo	Este caso de uso tiene como objetivo mostrar un grafico con el porcentaje de actividades con mayor cantidad de respuestas erróneas ingresadas por los alumnos.
Actor Primario	Profesor.
Precondiciones	El Profesor ha sido identificado y autenticado en el sistema.
Poscondiciones	Se ha mostrado el grafico correspondiente.
Escenario Principal	1. El Profesor solicita el Actividades con mayor dificultad.

(Flujo Básico)	2. El Sistema muestra la información. El Profesor puede volver al paso 1 hasta indicar término.
Escenarios Alternativos (extensiones)	1a.- Error de conexión. 1. - Se muestra mensaje correspondiente y se pide realizar la operación en otro momento.
Frecuencia de Ocurrencia	Media-Baja.

Tabla A.13 Caso de uso actividades con menor dificultad.

Caso de Uso	Actividades con menor dificultad.
Objetivo	Este caso de uso tiene como objetivo mostrar un grafico con el porcentaje de actividades con menor cantidad de respuestas erróneas ingresadas por los alumnos.
Actor Primario	Profesor.
Precondiciones	El Profesor ha sido identificado y autenticado en el sistema.
Poscondiciones	Se ha mostrado el grafico correspondiente.
Escenario Principal (Flujo Básico)	1. El Profesor solicita el Actividades con menor dificultad. 2. El Sistema muestra la información. El Profesor puede volver al paso 1 hasta indicar término.
Escenarios Alternativos	1a.- Error de conexión. 1. - Se muestra mensaje correspondiente y se pide realizar la

(extensiones)	operación en otro momento.
Frecuencia de Ocurrencia	Media-Baja.

Tabla A.14 Caso de uso crear respaldo.

Caso de Uso	Crear Respaldo.
Objetivo	Este caso de uso tiene como objetivo la creación de un respaldo del sistema.
Actor Primario	Profesor.
Precondiciones	El Profesor ha sido identificado y autenticado en el sistema.
Poscondiciones	Se ha creado un respaldo de la base de datos y de los ejercicios.
Escenario Principal (Flujo Básico)	<ol style="list-style-type: none"> 1. El Profesor solicita el crear un respaldo. 2. El Profesor comienza la creación de un respaldo. 3. El Sistema pide confirmación. 4. El Profesor confirma y el Sistema procede. <p>El Profesor puede volver al paso 2 hasta indicar término.</p>
Escenarios Alternativos (extensiones)	<p>1a.- Error de conexión.</p> <ol style="list-style-type: none"> 1. - Se muestra mensaje correspondiente y se pide realizar la operación en otro momento.
Frecuencia de	Media-Baja.

Ocurrencia	
-------------------	--

Tabla A.15 Caso de uso eliminar respaldo.

Caso de Uso	Eliminar Respaldo.
Objetivo	Este caso de uso tiene como objetivo la eliminación de un respaldo del sistema.
Actor Primario	Profesor.
Precondiciones	El Profesor ha sido identificado y autenticado en el sistema. Existe un respaldo del sistema.
Poscondiciones	Se ha eliminado el respaldo del sistema.
Escenario Principal (Flujo Básico)	1. El Profesor solicita eliminar un respaldo. 2. El Profesor comienza la eliminación de un respaldo. 3. El Sistema pide confirmación. 4. El Profesor confirma y el Sistema procede. El Profesor puede volver al paso 2 hasta indicar término.
Escenarios Alternativos (extensiones)	1a.- Error de conexión. 1. - Se muestra mensaje correspondiente y se pide realizar la operación en otro momento.
Frecuencia de Ocurrencia	Media-Baja.

Tabla A.16 Caso de uso cargar respaldo.

Caso de Uso	Cargar Respaldo.
Objetivo	Este caso de uso tiene como objetivo crear un respaldo existente del sistema.
Actor Primario	Profesor.
Precondiciones	El Profesor ha sido identificado y autenticado en el sistema. Existe un respaldo del sistema.
Poscondiciones	Se ha creado un respaldo de la base de datos y de los ejercicios.
Escenario Principal (Flujo Básico)	1. El Profesor solicita el cargar un respaldo. 2. El Profesor comienza la carga de un respaldo. 3. El Sistema pide confirmación. 4. El Profesor confirma y el Sistema procede. El Profesor puede volver al paso 2 hasta indicar término.
Escenarios Alternativos (extensiones)	1a.- Error de conexión. 1. - Se muestra mensaje correspondiente y se pide realizar la operación en otro momento.
Frecuencia de Ocurrencia	Media-Baja.

Anexo B

Diagramas de Secuencia.

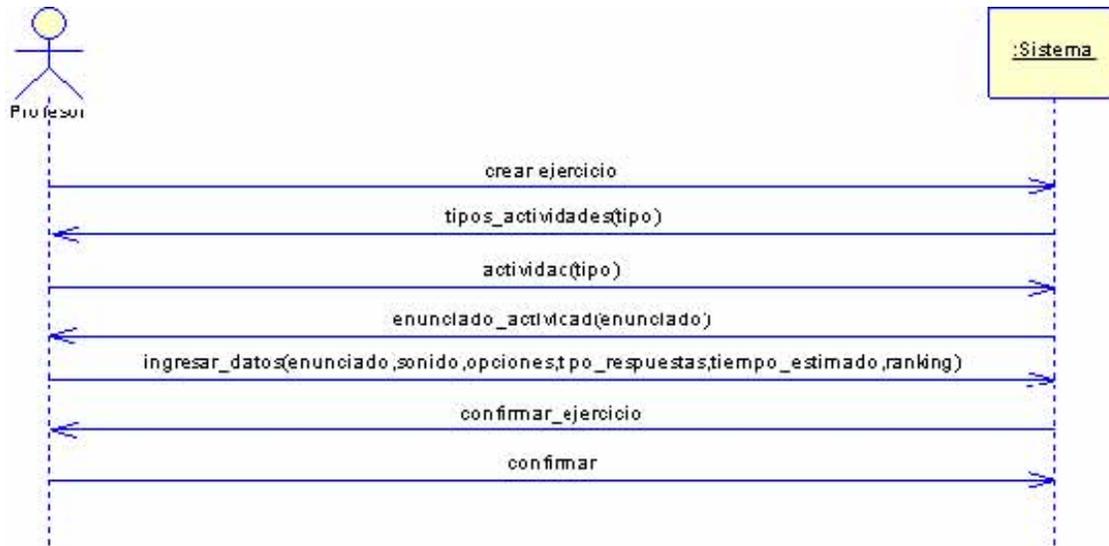


Ilustración B.1 Diagrama de Secuencia crear ejercicio.

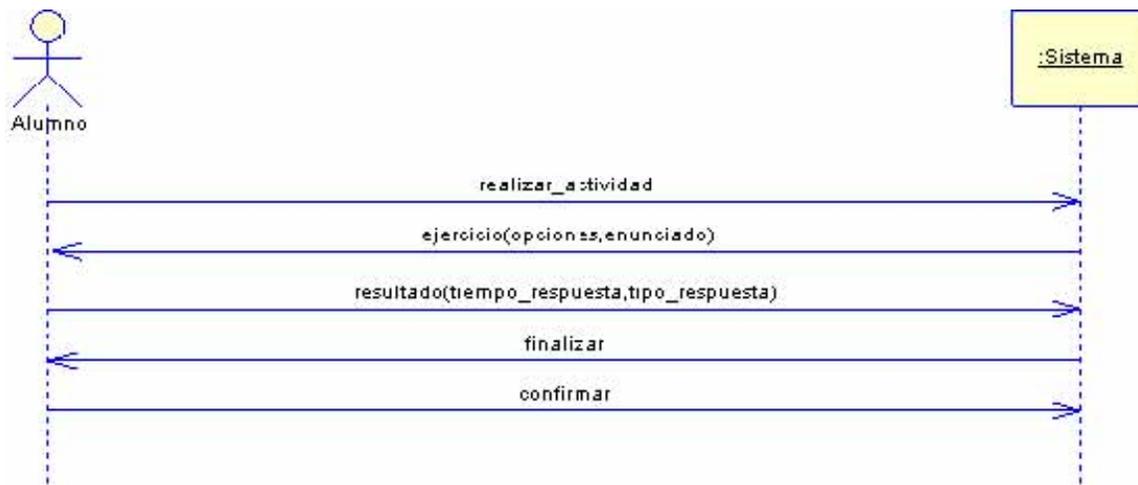


Ilustración B.2 Diagrama de Secuencia realizar actividad.

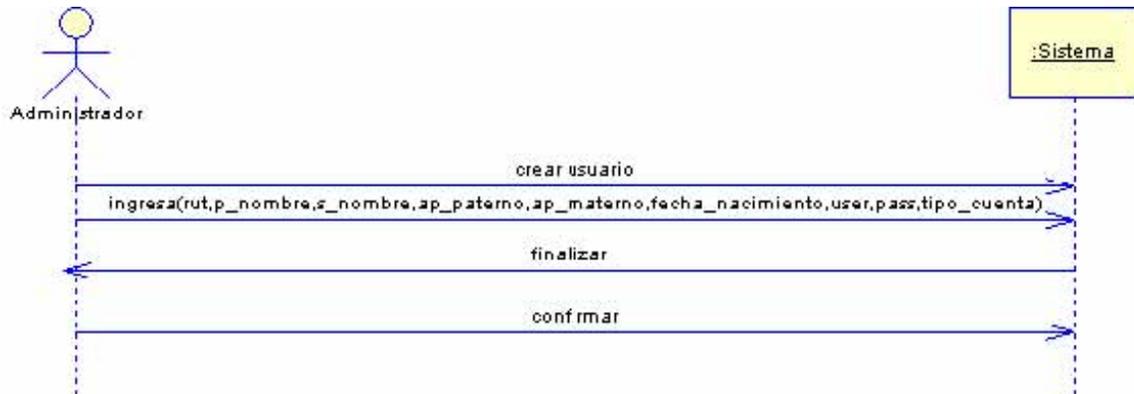


Ilustración B.3 Diagrama de Secuencia crear usuario.

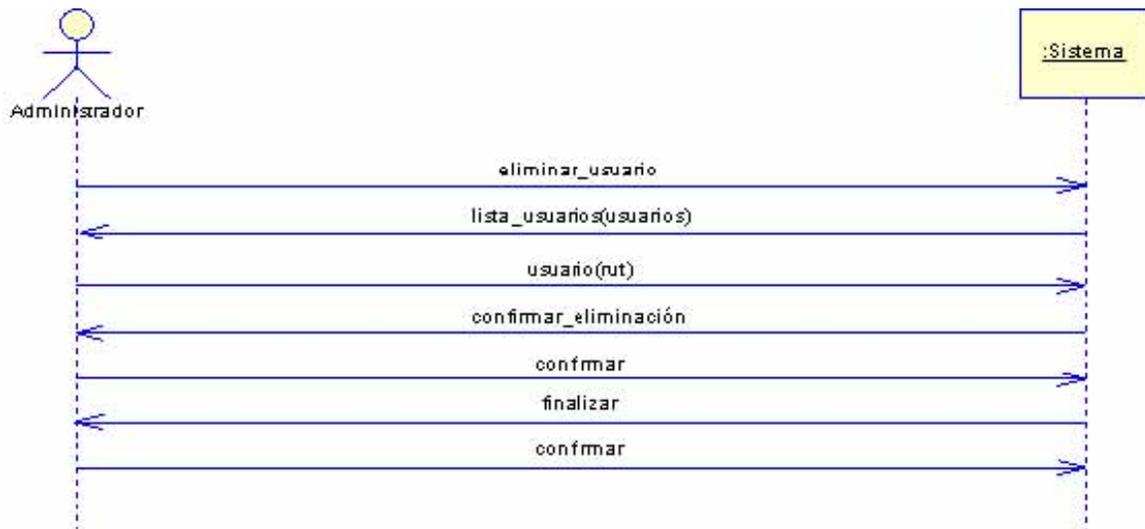


Ilustración B.4 Diagrama de Secuencia eliminar usuario.

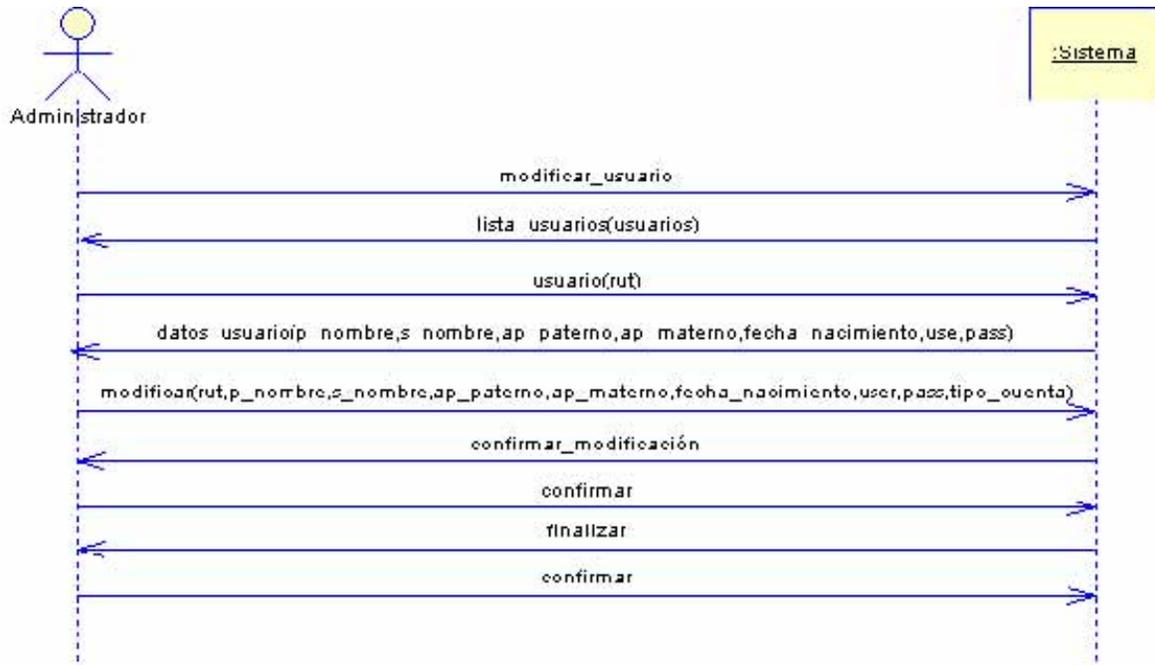


Ilustración B.5 Diagrama de Secuencia modificar usuario.

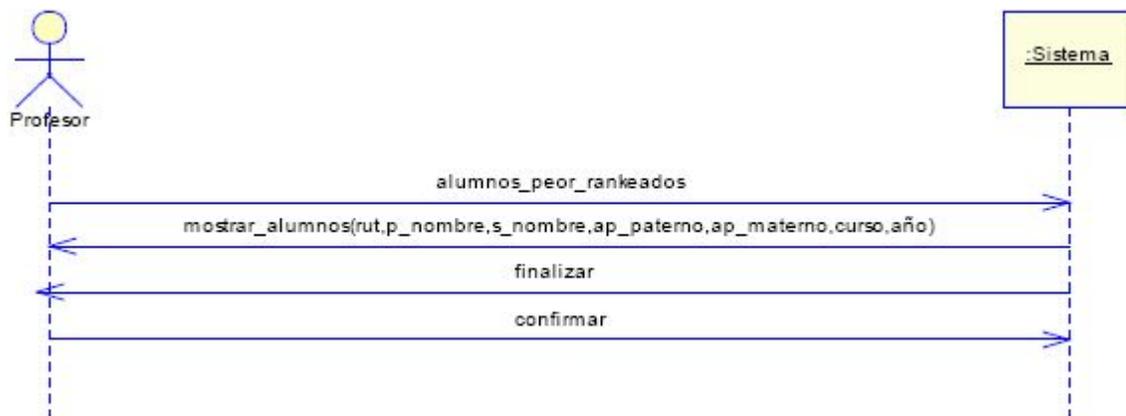


Ilustración B.6 Diagrama de Secuencia alumnos peor rankeados.

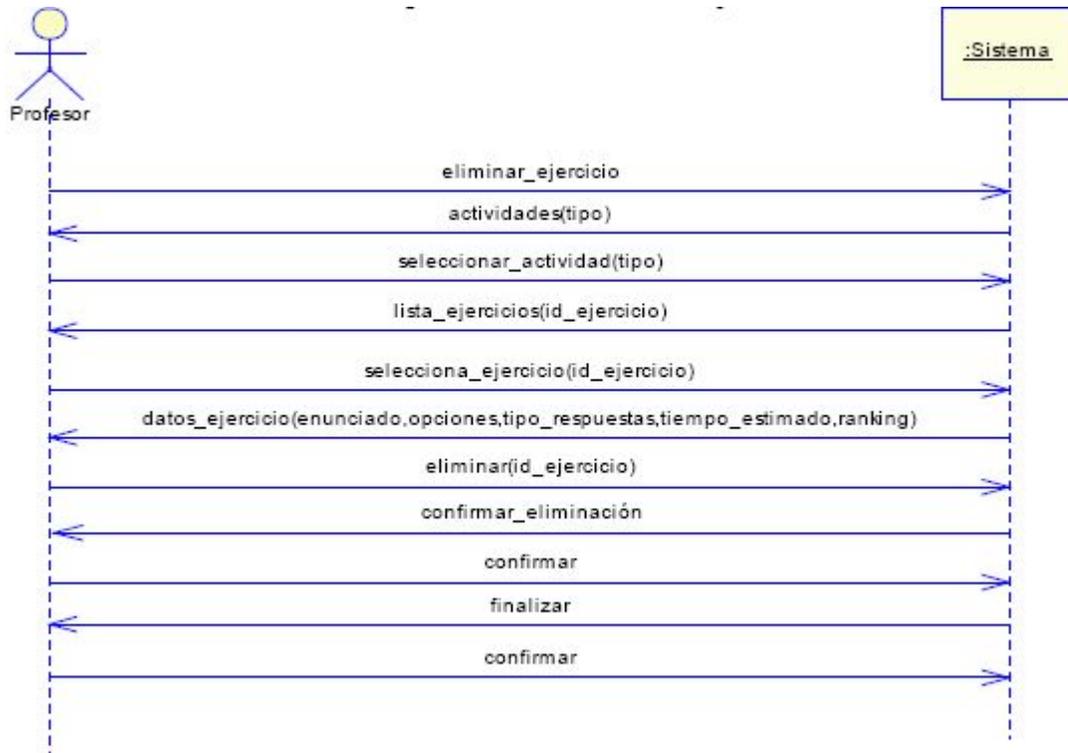


Ilustración B.7 Diagrama de Secuencia eliminar ejercicio.



Ilustración B.8 Diagrama de Secuencia alumnos mejor rankeados.

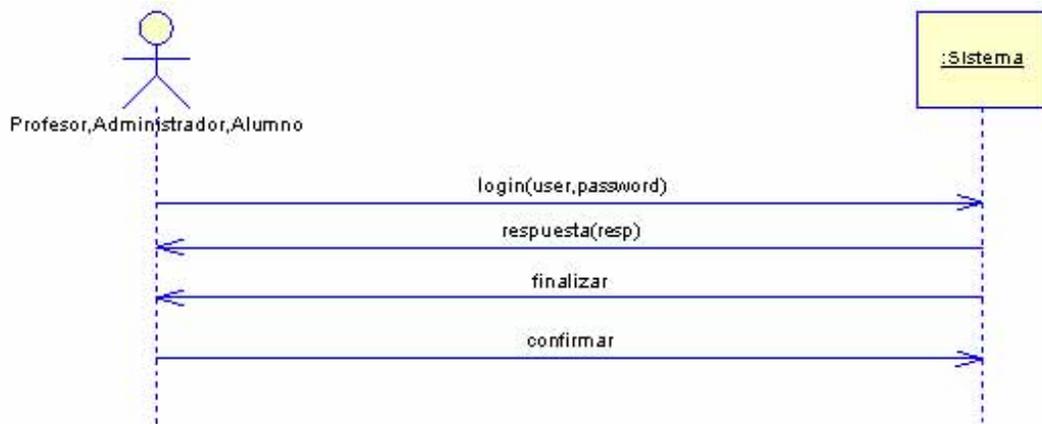


Ilustración B.9 Diagrama de Secuencia login.

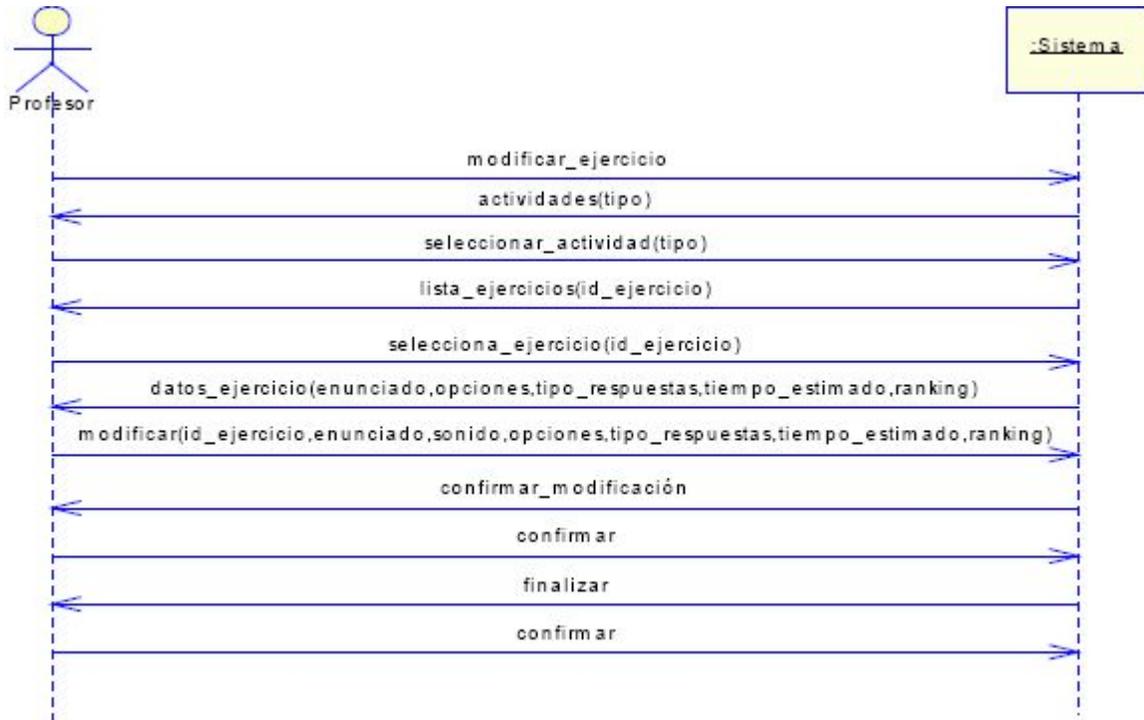


Ilustración B.10 Diagrama de Secuencia modificar ejercicio.

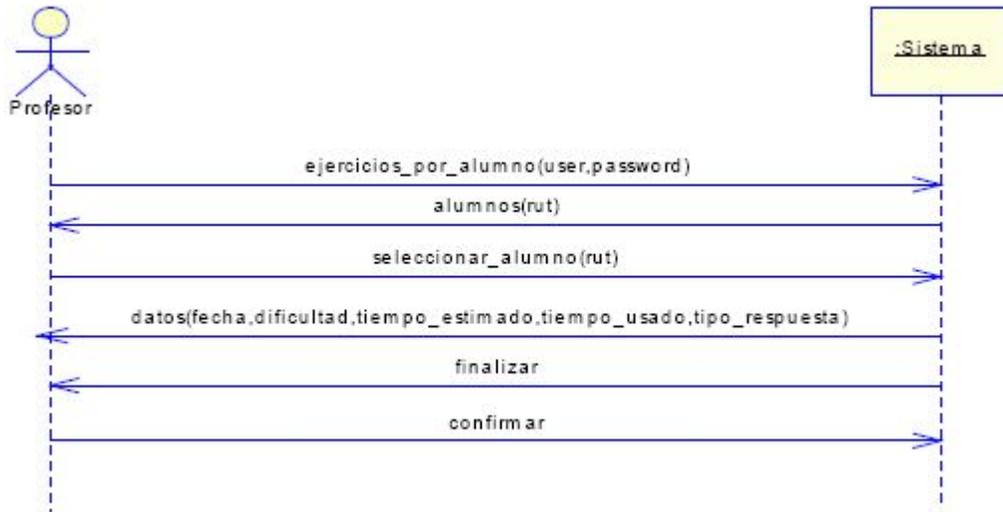


Ilustración B.11 Diagrama de Secuencia informe ejercicios por alumno.

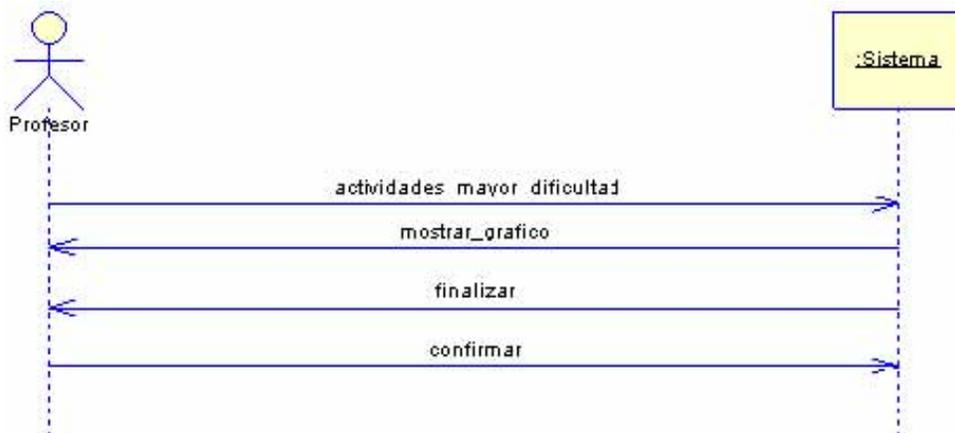


Ilustración B.12 Diagrama de Secuencia actividades con mayor dificultad.

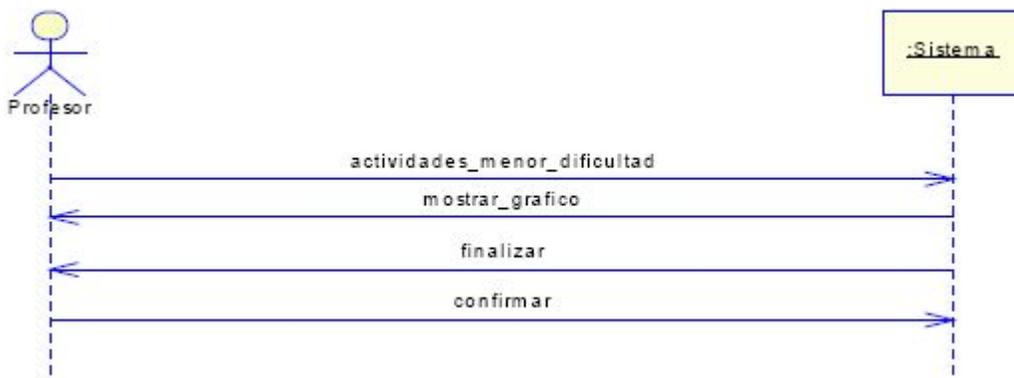


Ilustración B.13 Diagrama de Secuencia actividades con menor dificultad.

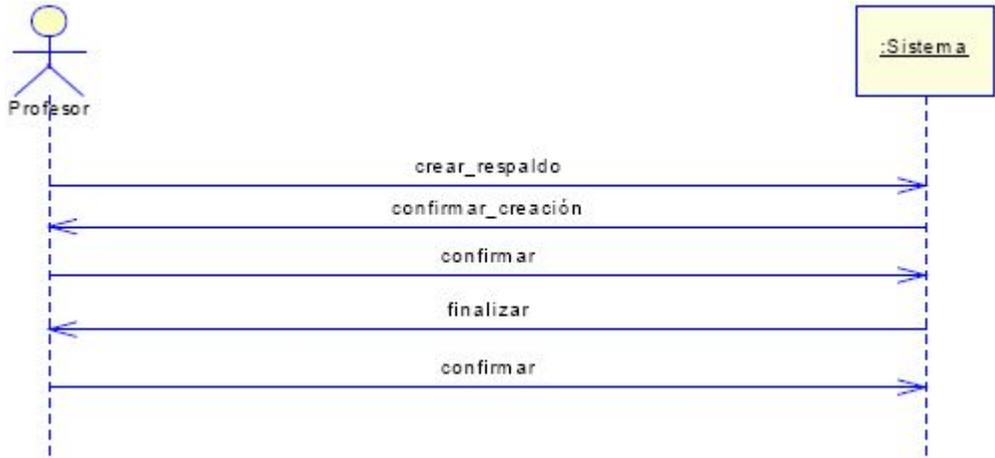


Ilustración B.14 Diagrama de Secuencia crear respaldo.

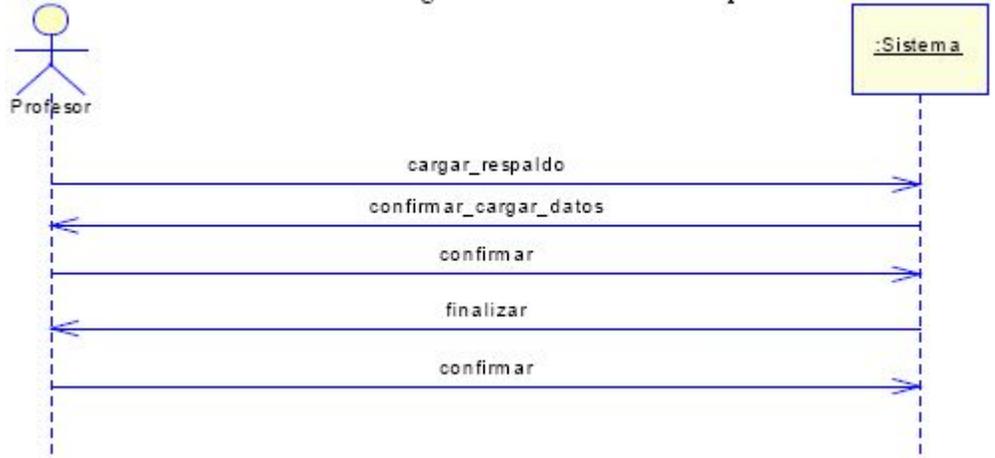


Ilustración B.15 Diagrama de Secuencia cargar respaldo.

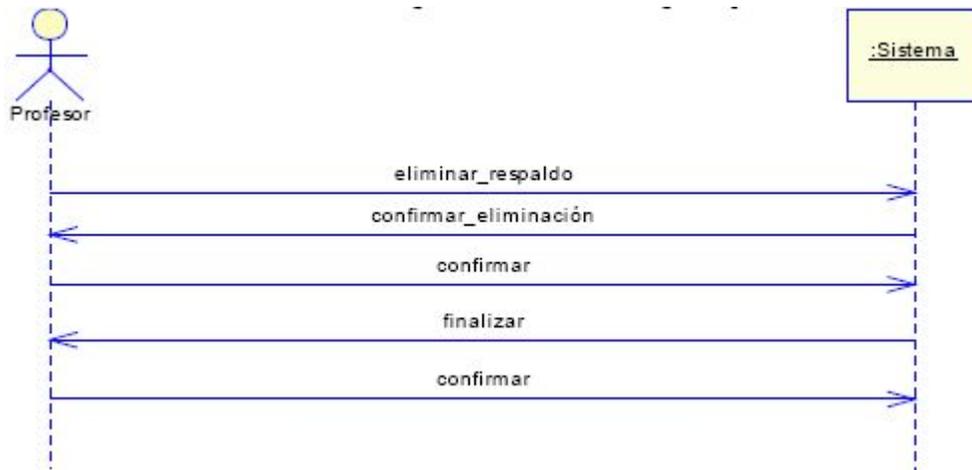


Ilustración B.16 Diagrama de Secuencia eliminar respaldo.

Anexo C

Reseña general de la enfermedad.

Ceguera y baja visión.

La ceguera es la pérdida del sentido de la vista. La ceguera puede ser total o parcial; existen varios tipos de ceguera parcial dependiendo del grado y tipo de pérdida de visión, como la baja visión, el escotoma, la ceguera parcial (de un ojo) o el daltonismo.

El término ceguera ha tenido un significado por demás tenebroso, peyorativo y excluyente. La etimología misma de la palabra “ciego” nos remite invariablemente al significado de negrura, tinieblas, turbio, oculto, imperfecto, cerrado. En lo concerniente a lo moral e intelectual, la ceguera adquiere matices aún más denigrantes: perturbado, oscuro, irracional, irreflexivo, que actúa sin discernimiento, que carece de prudencia, y por supuesto está relacionada con la ignorancia y la falta de conciencia.

Además del valor agregado por parte de los pueblos eslavos, que le dieron a la palabra “ceguera” el significado de “mendicidad”, la carencia del sentido de la vista, primordial en los seres humanos ya que por medio de los ojos se recibe el 80 por ciento de la información generada en el entorno social, representa una verdadera tragedia, una irremediable desgracia, algo indigno y vergonzoso

Por otro lado, se considera que una persona tiene baja visión cuando padece una limitación visual que le dificulta o impide la realización de las tareas de la vida cotidiana.

Clasificación y aspectos etiológicos de la ceguera.

Las causas genéticas, no solo son importantes en los grupos de edades infantiles, también lo son en adultos jóvenes, mientras que en las edades avanzadas, toman más relevancia las enfermedades degenerativas retinianas o maculares, muchas de las cuales son claramente hereditarias pero de aparición tardía.

La ceguera es una de las más impactantes causas de trastornos de la niñez y la que más incapacidad produce. Recientemente los patrones causales de la ceguera han cambiado, de manera que a medida que la medicina progresa y controla las enfermedades prevenibles, se hacen más evidentes las causas genéticas y/o hereditarias. Por consiguiente, la proporción de enfermedades hereditarias que producen la ceguera en la infancia ha aumentando lenta y paulatinamente en los últimos años. Esto es evidente cuando se analiza que hace aproximadamente 40 años, las causas nutricionales e infecciosas eran primordiales.

Hoy en día, al mejorar la calidad de vida y de atención médica preventiva, la genética obtiene lugares más altos como etiología del problema. Los trastornos retinianos por ejemplo, han sido siempre un grave problema etiológico, debido en buena parte a su complejidad, con lo que se ha dificultado su clasificación y categorización. Es innegable que existen muchas enfermedades distintas que producen similares alteraciones pigmentarias de la retina. En los últimos años se ha logrado definir un poco mejor esta clasificación de enfermedades y se ha establecido su diferenciación fenotípica, genotípica, etiológica y fisio-patológica.

Es importante que el paciente, sus familiares, los profesores de invidentes y los médicos generales o de otras especialidades que enfocan primariamente un problema de esta naturaleza, sean plenamente conocedores de la patogénesis del problema en las enfermedades hereditarias que producen ceguera.

Importancia en el estudio genético de las cegueras.

El consenso general de estudios etiológicos de la ceguera tales como los realizados por Fraser, Schappert, Jan y Warburg, es que las tres principales causas de la ceguera en la infancia son:

- Degeneraciones tapeto-retinianas.
- Cataratas congénitas.

- Malformaciones de tipo microftalmos, anoftalmos y Coloboma.

Otras causas menos comunes son el albinismo, anirida, glaucoma infantil y por último, la acromatopsia. Con respecto al “glaucoma infantil” se sabe que es causado por malformaciones de la cámara anterior del ojo, de ahí que su estudio sea complejo y no haya unificación en los criterios de clasificación como causa genética o en la determinación de su frecuencia global. Indudablemente es una causa de ceguera bastante común, y su frecuencia en los diferentes estudios, varía entre el 1 y el 16%. Sin embargo algunos autores insisten en que su frecuencia como etiología de ceguera ha ido en descenso; precisamente debido a la intervención quirúrgica temprana y oportuna.

Las “malformaciones congénitas”, a su vez representan una buena proporción de cegueras infantiles. Su frecuencia es bastante variable entre los diferentes estudios publicados, siendo cercana al 5% en Inglaterra, al 4% en Canadá, al 9% en Alemania y al 13% en Japón. Esta diferencia estriba obviamente en la variabilidad de la población estudiada, los criterios de selección del estado visual de la muestra tomada y de la metodología de evaluación. En Colombia parece estar cerca de un 5 a 7%, sin embargo es algo prematuro dar conclusiones al respecto

Lo que si debe tenerse muy en cuenta al analizar estas malformaciones, es si están presentes en forma aislada o son parte de patologías más complejas. Esta diferenciación es definitiva para determinar la presencia de síndromes genéticos, o de otras enfermedades con compromiso sistémico debidas a agentes externos, como medicamentos, infecciones, etc.

Por otra parte, las “cataratas congénitas” ocupan un lugar importante como etiología de ceguera en la niñez. Lo reportado por Tabarra y colaboradores en 1962, muestra una frecuencia del 33% en niños invidentes de Arabia Saudita. Por su parte, Fraser, Jan y Fijiki encontraron una frecuencia en el orden de 13 al 19% en sus respectivos estudios. La incidencia reportada en Dinamarca es solo del 5%, pero su estudio se basó en una población de solo 150 niños invidentes. Sin embargo, lo reportado en Canadá muestra cifras similares, por lo cual tiende a creerse que hoy en día la frecuencia de cataratas congénitas es realmente más baja que los años anteriores, posiblemente debido a tratamientos quirúrgicos oportunos. Esta teoría de la disminución de la frecuencia tendría explicaciones primero, podría ser la disminución relativa de la frecuencia de infección como la rubéola, por ejemplo. Segundo,

porque la cirugía temprana de la catarata congénita hace que el niño afectado no llegue a presentar niveles tan bajos de visión, por lo que no tendría necesidad de asistir a escuelas de invidentes, bajando así su frecuencia relativa entre la población ciega infantil institucionalizada.

Las alteraciones “tapeto - retinianas” son del 10 al 33% de las causas de ceguera infantil. Lo importante de tener en cuenta en este caso es que la mayoría de las alteraciones genéticas de la retina tienen un inicio hacia los 15 o 35 años de edad, por lo que esta etiología puede pasar desapercibida si se toman grupos de edades muy tempranas. Otros estudios, parecen demostrar frecuencias más altas. Lindstedt en Suecia, por ejemplo, encontró que dentro de todas las causas genéticas las distrofias retinianas representan el 75% mientras Green y Jonson en 1983 llegaron a probar que éstas corresponden al 53% de todas las causas genéticas de ceguera en la muestra analizada por ellos en una región de Canadá. Con respecto a estas retinopatías, debemos saber que muchas de ellas no están solas, sino asociadas a otras alteraciones conformando síndromes específicos, lo que hace primordial una muy extensa evaluación global de cada caso estudiado.

Por otra parte, la atrofia óptica parece ser muy frecuente en casos de ceguera y retardo mental, por lo que algunos autores coinciden en afirmar que posiblemente esa clase de atrofia sea parte de síndromes genéticos hereditarios, o sea debida a anomalías del desarrollo por causas exógenas prevenibles.

¿Qué es la baja visión?

Se considera que una persona tiene baja visión cuando padece una limitación visual que le dificulta o impide la realización de las tareas de la vida cotidiana.

La baja visión puede ser causada por distintas patologías, accidentes o malformaciones congénitas. En todos los casos se trata de una razón crónica e irreversible, por lo que la rehabilitación de la baja visión entra en juego cuando su oftalmólogo no puede hacer nada más por mejorar la visión.

La baja visión afecta particularmente a las personas mayores, ya que varias de las patologías que la causan son enfermedades degenerativas de la visión.

Patologías más comunes de la baja visión.

Degeneración Macular: La degeneración macular es más frecuente en personas mayores y por eso algunas veces se habla de degeneración macular asociada a la edad (DMAE). Las consecuencias son la pérdida de la visión central, visión distorsionada, manchas y líneas torcidas.

Retinopatía Diabética: Aproximadamente el 80% de los diabéticos que tienen esta enfermedad experimentan un hinchazón y goteo de los vasos sanguíneos de la retina, que puede causar imágenes borrosas en el campo visual central. La mayoría de los casos evolucionan hacia un estado donde el crecimiento anormal de los vasos sanguíneos puede romper y manchar de sangre el humor vítreo. Generalmente, un resto de visión permanece en buenas condiciones.

Cataratas: La visión se nubla y produce una pérdida de visión de los detalles. El campo de visión no resulta afectado, aunque el deslumbramiento, la distorsión y las imágenes dobles pueden causar molestias.

Retinosis Pigmentaria: Es una enfermedad hereditaria que afecta a la visión debido a una distrofia de la retina. Se caracteriza por la ceguera nocturna y la visión lateral o “visión de túnel”. También puede afectar a la visión central.

Glaucoma: El glaucoma es una enfermedad del ojo que se define como una neuropatía óptica o una enfermedad del nervio óptico. Una presión intraocular alta es un importante factor de riesgo para desarrollar glaucoma.

Miopía Magna: Una persona con miopía tiene dificultades para enfocar bien los objetos distantes, lo que puede conducir también a dolores de cabeza, bizqueo y tensión en los ojos.

Anexo D

Características de MySQL.

Velocidad.

MySQL es rápido. Los desarrolladores sostienen que MySQL es posiblemente la base de datos más rápida que se pueda encontrar. En la página de MySQL existe una comparación de rendimientos de algunas bases de datos y MySQL.

Facilidad de uso.

MySQL es un sistema de base de datos de alto rendimiento pero relativamente simple y es mucho menos complejo de configurar y administrar que sistemas mas grandes.

Coste.

MySQL es gratuito para la mayoría de usos internos.

Capacidad de gestión de lenguajes de consulta.

MySQL comprende SQL (Structured Query Language, Lenguaje de Consulta Estructurado), el lenguaje elegido para todos los sistemas de base de datos modernos. También puede acceder a MySQL empleando aplicaciones que admitan ODBC (Open Database Connectivity, Conectividad de Bases de Datos), un protocolo de comunicación de bases de datos desarrollado por Microsoft.

Capacidad.

Pueden conectarse muchos clientes simultáneamente al servidor: Los clientes pueden utilizar varias bases de datos simultáneamente. Puede acceder de forma interactiva a MySQL empleando diferentes interfaces que permiten introducir consultas y visualizar los resultados: cliente de línea de comando, navegador web o clientes de sistema X Windows. Además, está disponible una amplia variedad de interfaces de programación para lenguajes como C, Perl, Java, PHP y Pitón. Por lo tanto, tiene la posibilidad de elegir entre usar un software cliente pre-empaquetado o escribir sus propias aplicaciones a medida.

Conectividad y seguridad.

MySQL está completamente preparado para el trabajo en red y las bases de datos pueden ser accedidas desde cualquier lugar en Internet, por lo que puede compartir sus datos con cualquiera, en cualquier parte. Pero MySQL dispone de control de acceso, de forma que aquellos que no deberían ver sus datos, no los vean.

Portabilidad.

MySQL se ejecuta en muchas variantes de UNIX, así como en otros sistemas no-UNIX, como Windows y OS/2. MySQL se ejecuta en hardware que va desde PC hasta servidores de alta capacidad.

Distribución abierta.

MySQL es fácil de obtener; se puede descargar una versión gratuita desde internet. Si no entiende como funciona alguna cosa o tiene curiosidad por un algoritmo, puede conseguir el código fuente e investigarlo. Si no le gusta cómo funcionan las cosas, puede cambiarlas.

¿Qué hay del mantenimiento?

- MySQL incluye un extenso Manual de referencia (450 páginas y sigue creciendo).
- Es posible disponer de contratos de mantenimiento con los desarrolladores de MySQL, para aquellos que prefieran un acuerdo formal.
- Existe una lista de correo a la que se puede suscribir. La lista tiene muchos participantes de los que sacar provecho, entre los que se incluye los desarrolladores de MySQL [DuBois, 2000].

Anexo E

Características de Java.

Sencillo.

- Fue diseñado para facilitar las tareas del programador profesional y utilizar eficientemente el lenguaje.
- Elimina las dificultades que presentan sus lenguajes de procedencia (punteros, herencia múltiple, goto, estructuras, necesidad de liberar memoria).

Orientado a Objetos.

- Modela el mundo real, es decir, cualquier cosa del mundo real puede ser modelada como un objeto. Java se centra en la creación y manipulación de objetos.

- Soporta las características más importantes de las tecnologías Orientadas a Objetos (OO): herencia, encapsulamiento y polimorfismo.

Distribuido.

- Java ha sido diseñado para facilitar la construcción de aplicaciones distribuidas mediante una colección de clases para uso en aplicaciones en red. Maneja protocolos TCP/IP.

Robusto.

- Consta con una tipificación fuerte, lo cual es verificado en tiempo de compilación.
- Proporciona manejo de errores en tiempo de ejecución (gestión de memoria y condiciones excepcionales mal manejadas).
- Todo esto lo hace un lenguaje fiable.

Seguro.

- Java permite la construcción de programas libres de virus.
- Al descargar un applet Java y ejecutarlo en un PC local no existe la posibilidad de que se dañe el sistema, ya que java implementa diversos mecanismos de seguridad para proteger los sistemas.

Portable.

- Utilizando Java, los desarrolladores necesitan escribir una única versión para ejecutarse en todas las plataformas que tengan implementada una JVM.

Alto desempeño.

- Java proporciona de compiladores JIT (Just in Time) que permiten que programas Java independientes de la plataforma se ejecuten con casi el mismo rendimiento en tiempo de ejecución que los lenguajes convencionales compilados.

Multihilo.

- Java tiene la capacidad de ejecutar varias tareas simultáneamente a través de la creación de multihilos. Los hilos sincronizados son muy útiles en la creación de aplicaciones distribuidas y en red.

¿Por qué utilizar Java?

Para el desarrollo del proyecto se ha decidido implementar la interfaz de administración del profesor utilizando el lenguaje de programación Java. Entre las razones más importantes está la portabilidad del software, si en un futuro este debiera migrar a un equipo con UNIX, no se tendría que realizar ningún tipo actualización del código.

Anexo F

Pruebas de Software.

Pruebas de Caja negra interfaz profesor.

Cuenta: interfaz de logueo, el usuario debe proporcionar un RUT, un dígito verificador y una contraseña valida en el sistema.

Tabla F.1 Casos de Prueba interfaz cuenta.

Condición de entrada	Clases válidas	Clases inválidas
RUT	- $1 \leq \text{RUT} \leq 99999999$	- $\text{RUT} < 1$ - $\text{RUT} > 99999999$ - RUT no es número - RUT es vacío
dígito	- $0 \leq \text{dígito} \leq 9$ - $\text{dígito} = k$	- $\text{dígito} < 0$ - $\text{dígito} > 9$ - $\text{dígito} \neq k$ - dígito es vacío

contraseña	- menor a veinte caracteres - mayor a cero caracteres	- mayor a veinte caracteres - contraseña es vacío
------------	--	--

Pruebas de caja negra, interfaz Cuenta:

1. Ingresa valor RUT igual a -1, retorna mensaje de error.
2. Ingresa valor RUT igual a 999999999, retorna mensaje de error.
3. Ingresa valor RUT igual a 'a', retorna mensaje de error.
4. Ingresa valor RUT igual a ', retorna mensaje de error.
5. Ingresa valor RUT igual a 15852177, NO retorna mensaje de error.
6. Ingresa valor dígito igual a -1, retorna mensaje de error.
7. Ingresa valor dígito igual a 10, retorna mensaje de error.
8. Ingresa valor dígito igual a 's', retorna mensaje de error.
9. Ingresa valor dígito igual a ', retorna mensaje de error.
10. Ingresa valor dígito igual a 'k', NO retorna mensaje de error.
11. Ingresa valor contraseña igual a '123456aswergffgrew128', retorna mensaje de error.
12. Ingresa valor contraseña igual a ', retorna mensaje de error.
13. Ingresa valor contraseña igual a 'superior123', NO retorna mensaje de error.

Recuperación de contraseña: interfaz para enviar la contraseña del usuario al mail, el usuario debe proporcionar un RUT y un dígito verificador valido en el sistema.

Tabla F.2 Casos de Prueba interfaz recuperación de contraseña.

Condición de entrada	Clases válidas	Clases inválidas

RUT	- $1 \leq \text{RUT} \leq 999999999$	- RUT < 1 - RUT > 999999999 - RUT no es número - RUT es vacío
dígito	- $0 \leq \text{dígito} \leq 9$ - dígito = k	- dígito < 0 - dígito > 9 - dígito != k - dígito es vacío

Pruebas de caja negra, interfaz Recuperación de contraseña:

1. Ingresar valor RUT igual a -1, retorna mensaje de error.
2. Ingresar valor RUT igual a 999999999, retorna mensaje de error.
3. Ingresar valor RUT igual a 'a', retorna mensaje de error.
4. Ingresar valor RUT igual a ',', retorna mensaje de error.
5. Ingresar valor RUT igual a 15852177, NO retorna mensaje de error.
6. Ingresar valor dígito igual a -1, retorna mensaje de error.
7. Ingresar valor dígito igual a 10, retorna mensaje de error.
8. Ingresar valor dígito igual a 's', retorna mensaje de error.
9. Ingresar valor dígito igual a ',', retorna mensaje de error.
10. Ingresar valor dígito igual a 'k', NO retorna mensaje de error.

Crear Usuario: interfaz perteneciente al administrador cuya función es crear una cuenta, el usuario debe proporcionar un RUT, un dígito verificador, primer nombre, segundo nombre, apellido paterno, apellido materno, contraseña, confirmar contraseña y mail.

Tabla F.3 Casos de Prueba interfaz crear usuario.

Condición de entrada	Clases válidas	Clases inválidas
RUT	- $1 \leq \text{RUT} \leq 999999999$	<ul style="list-style-type: none"> - RUT < 1 - RUT > 999999999 - RUT no es número - RUT es vacío
dígito	<ul style="list-style-type: none"> - $0 \leq \text{dígito} \leq 9$ - dígito = k 	<ul style="list-style-type: none"> - dígito < 0 - dígito > 9 - dígito != k - dígito es vacío
primer nombre	- veinte letras \leq primer nombre \leq un letra	<ul style="list-style-type: none"> - primer nombre > veinte letras - primer nombre es vacío - primer nombre es entero
segundo nombre	- veinte letras \leq segundo nombre \leq una letra	<ul style="list-style-type: none"> - segundo nombre > veinte letras - segundo nombre es

		<p>vacío</p> <p>- segundo nombre es entero</p>
apellido paterno	- veinte letras<=apellido paterno<= una letra	<p>- apellido paterno> veinte letras</p> <p>- apellido paterno es vacío</p> <p>- apellido paterno es entero</p>
apellido materno	- veinte letras<=apellido materno<= una letra	<p>- apellido materno> veinte letras</p> <p>- apellido materno es vacío</p> <p>- apellido materno es entero</p>
<p>Contraseña</p> <p>(en el caso de ser una cuenta para administrador o profesor)</p>	- veinte caracteres<=contraseña<= un carácter	<p>- contraseña> veinte caracteres</p> <p>- contraseña es vacío</p>
<p>confirmar contraseña</p> <p>(en el caso de ser una cuenta para administrador o profesor)</p>	<p>- veinte caracteres<=confirmar contraseña<= un carácter</p> <p>- confirmar contraseña=contraseña</p>	<p>- confirmar contraseña> veinte caracteres</p> <p>- confirmar contraseña es vacío</p>

		- confirmar contraseña != contraseña
e-mail (en el caso de ser una cuenta para administrador o profesor)	- cincuenta caracteres<=e-mail<= un carácter	- e-mail > cincuenta caracteres - e-mail es vacío

Pruebas de caja negra, interfaz Crear Usuario:

1. Ingresar valor RUT igual a -1, retorna mensaje de error.
2. Ingresar valor RUT igual a 999999999, retorna mensaje de error.
3. Ingresar valor RUT igual a 'a', retorna mensaje de error.
4. Ingresar valor RUT igual a ', retorna mensaje de error.
5. Ingresar valor RUT igual a 15852177, NO retorna mensaje de error.
6. Ingresar valor dígito igual a -1, retorna mensaje de error.
7. Ingresar valor dígito igual a 10, retorna mensaje de error.
8. Ingresar valor dígito igual a 's', retorna mensaje de error.
9. Ingresar valor dígito igual a ", retorna mensaje de error.
10. Ingresar valor dígito igual a 'k', NO retorna mensaje de error.
11. Ingresar valor primer nombre igual a 'Robertooooooooooooooooooooo ', retorna mensaje de error.
12. Ingresar valor primer nombre igual a ', retorna mensaje de error.

13. Ingresar valor primer nombre igual a 'as12', retorna mensaje de error.
14. Ingresar valor primer nombre igual a 'Juan', NO retorna mensaje de error.
15. Ingresar valor segundo nombre igual a 'Pedroooooooooooooooooooooo ', retorna mensaje de error.
16. Ingresar valor segundo nombre igual a '', retorna mensaje de error.
17. Ingresar valor segundo nombre igual a 'Raul12', retorna mensaje de error.
18. Ingresar valor segundo nombre igual a 'Miguel', NO retorna mensaje de error.
19. Ingresar valor apellido paterno igual a 'Sotoooooooooooooooooooooo ', retorna mensaje de error.
20. Ingresar valor apellido paterno igual a '', retorna mensaje de error.
21. Ingresar valor apellido paterno igual a 'as12', retorna mensaje de error.
22. Ingresar valor apellido paterno igual a 'Lucero', NO retorna mensaje de error.
23. Ingresar valor apellido materno igual a 'Pereeeeeeeeeeeeeeeeeez ', retorna mensaje de error.
24. Ingresar valor apellido materno igual a '', retorna mensaje de error.
25. Ingresar valor apellido materno igual a '4564hfg12', retorna mensaje de error.
26. Ingresar valor apellido materno igual a 'Gómez', NO retorna mensaje de error.
27. Ingresar valor contraseña igual a '123456789123aswedfgwed', retorna mensaje de error.
28. Ingresar valor contraseña igual a '', retorna mensaje de error.
29. Ingresar valor contraseña igual a '1234asd', NO retorna mensaje de error.
30. Ingresar valor confirmar contraseña igual a '12345678912375678asdf', retorna mensaje de error.

31. Ingresa valor confirmar contraseña igual a ‘’, retorna mensaje de error.
32. Ingresa valor contraseña igual a ‘123’ y valor confirmar contraseña igual a ‘12asd’, retorna mensaje de error.
33. Ingresa valor contraseña igual a ‘123’ y valor confirmar contraseña igual a ‘123’, NO retorna mensaje de error.
34. Ingresa valor e-mail igual a ‘123456789987654455asdwerthyf sseertvaserfsa ser cccxa’, retorna mensaje de error.
35. Ingresa valor e-mail igual a ‘’, retorna mensaje de error.
36. Ingresa valor e-mail igual a ‘jhorta@gmail.com’, NO retorna mensaje de error.

Modificar Usuario: interfaz perteneciente al administrador cuya función es modificar los datos de una cuenta, el usuario debe proporcionar primer nombre, segundo nombre, apellido paterno, apellido materno, contraseña, confirmar contraseña y mail.

Tabla F.4 Casos de Prueba interfaz modificar usuario.

Condición de entrada	Clases válidas	Clases inválidas
primer nombre	- treinta letras<=primer nombre<= un letra	- primer nombre> treinta letras - primer nombre es vacío - primer nombre es entero
segundo nombre	- treinta letras<=segundo nombre<= una letra	- segundo nombre> treinta letras - segundo nombre es vacío

		- segundo nombre es entero
apellido paterno	- veinte letras<=apellido paterno<= una letra	- apellido paterno> veinte letras - apellido paterno es vacío - apellido paterno es entero
apellido materno	- veinte letras<=apellido materno<= una letra	- apellido materno> veinte letras - apellido materno es vacío - apellido materno es entero
contraseña (en el caso de ser una cuenta para administrador o profesor)	- veinte caracteres<=contraseña<= un carácter	- contraseña> veinte caracteres - contraseña es vacío
confirmar contraseña(en el caso de ser una cuenta para administrador o profesor)	- veinte caracteres<=confirmar contraseña<= un carácter - confirmar contraseña=contraseña	- confirmar contraseña> veinte caracteres - confirmar contraseña es vacío - confirmar contraseña !=

		contraseña
e-mail(en el caso de ser una cuenta para administrador o profesor)	- cincuenta caracteres<=e-mail<= un carácter	- e-mail > cincuenta caracteres - e-mail es vacío

Pruebas de caja negra, interfaz Modificar Usuario:

1. Ingresar valor primer nombre igual a 'Robertoooooooooooooooooooooo ', retorna mensaje de error.
2. Ingresar valor primer nombre igual a '', retorna mensaje de error.
3. Ingresar valor primer nombre igual a 'as12', retorna mensaje de error.
4. Ingresar valor primer nombre igual a 'Juan', NO retorna mensaje de error.
5. Ingresar valor segundo nombre igual a 'Pedroooooooooooooooooooooo ', retorna mensaje de error.
6. Ingresar valor segundo nombre igual a '', retorna mensaje de error.
7. Ingresar valor segundo nombre igual a 'Raul12', retorna mensaje de error.
8. Ingresar valor segundo nombre igual a 'Miguel', NO retorna mensaje de error.
9. Ingresar valor apellido paterno igual a 'Sotoooooooooooooooooooooo ', retorna mensaje de error.
10. Ingresar valor apellido paterno igual a '', retorna mensaje de error.
11. Ingresar valor apellido paterno igual a 'as12', retorna mensaje de error.
12. Ingresar valor apellido paterno igual a 'Lucero', NO retorna mensaje de error.

13. Ingresar valor apellido materno igual a 'Pereeeeeeeeeeeeeeeeeez ', retorna mensaje de error.
14. Ingresar valor apellido materno igual a '', retorna mensaje de error.
15. Ingresar valor apellido materno igual a '4564hfg12', retorna mensaje de error.
16. Ingresar valor apellido materno igual a 'Gómez', NO retorna mensaje de error.
17. Ingresar valor contraseña igual a '123456789123aswedfgwed', retorna mensaje de error.
18. Ingresar valor contraseña igual a '', retorna mensaje de error.
19. Ingresar valor contraseña igual a '1234asd', NO retorna mensaje de error.
20. Ingresar valor confirmar contraseña igual a '12345678912375678asdf', retorna mensaje de error.
21. Ingresar valor confirmar contraseña igual a '', retorna mensaje de error.
22. Ingresar valor contraseña igual a '123' y valor confirmar contraseña igual a '12asd', retorna mensaje de error.
23. Ingresar valor contraseña igual a '123' y valor confirmar contraseña igual a '123', NO retorna mensaje de error.
24. Ingresar valor e-mail igual a '123456789987654455asdwertyf sseertvaserfsa ser cccxa', retorna mensaje de error.
25. Ingresar valor e-mail igual a '', retorna mensaje de error.
26. Ingresar valor e-mail igual a 'jhorta@gmail.com', NO retorna mensaje de error.

Ingresar ejercicios: interfaz perteneciente al profesor cuya función es ingresar un nuevo ejercicio en la aplicación, el usuario debe proporcionar un encabezado escrito, encabezado audible, respuesta A, respuesta B, respuesta C, respuesta D, tipo de respuesta A, tipo de respuesta B, tipo de respuesta C, tipo de respuesta D y el tiempo estimado.

Tabla F.5 Casos de Prueba interfaz ingresar ejercicios.

Condición de entrada	Clases válidas	Clases inválidas
encabezado escrito	- encabezado escrito > cero carácter	- encabezado escrito es vacío
encabezado audible	- encabezado audible debe ser grabado	- encabezado audible no fue grabado
respuesta A	<ul style="list-style-type: none"> - $1 \leq \text{respuesta A} \leq 1000$ - respuesta A != respuesta B - respuesta A != respuesta C - respuesta A != respuesta D 	<ul style="list-style-type: none"> - respuesta A < 1 - respuesta A > 1000 - respuesta A es vacío - respuesta A es letra - respuesta A=B - respuesta A=C - respuesta A=D
respuesta B	<ul style="list-style-type: none"> - $1 \leq \text{respuesta B} \leq 1000$ - respuesta B != respuesta A - respuesta B != respuesta C - respuesta B != respuesta D 	<ul style="list-style-type: none"> - respuesta B < 1 - respuesta B > 1000 - respuesta B es vacío - respuesta B es letra - respuesta B=A - respuesta B=C - respuesta B=D
respuesta C	- $1 \leq \text{respuesta C} \leq 1000$	- respuesta C < 1

	<ul style="list-style-type: none"> - respuesta C != respuesta A - respuesta C != respuesta B - respuesta C != respuesta D 	<ul style="list-style-type: none"> - respuesta C >1000 - respuesta C es vacío - respuesta C es letra - respuesta C=A - respuesta C=B - respuesta C=D
respuesta D	<ul style="list-style-type: none"> - $1 \leq \text{respuesta D} \leq 1000$ - respuesta D != respuesta A - respuesta D != respuesta B - respuesta D != respuesta C 	<ul style="list-style-type: none"> - respuesta D <1 - respuesta D >1000 - respuesta D es vacío - respuesta D es letra - respuesta D=A - respuesta D=B - respuesta D=C
tipo de respuesta A	<ul style="list-style-type: none"> - correcta - incorrecta leve - incorrecta grave 	<ul style="list-style-type: none"> - tipo de respuesta A es vacío
tipo de respuesta B	<ul style="list-style-type: none"> - correcta - incorrecta leve - incorrecta grave 	<ul style="list-style-type: none"> - tipo de respuesta B es vacío
tipo de respuesta C	<ul style="list-style-type: none"> - correcta - incorrecta leve 	<ul style="list-style-type: none"> - tipo de respuesta C es vacío

	- incorrecta grave	
tipo de respuesta D	- correcta - incorrecta leve - incorrecta grave	- tipo de respuesta D es vacío
tiempo estimado	$1 \leq \text{tiempo estimado} \leq 360$	- tiempo estimado <1 - tiempo estimado >360 - tiempo estimado es vacío - tiempo estimado es una letra

Pruebas de caja negra, interfaz Ingresar Ejercicio:

1. Ingresar valor encabezado escrito igual a ‘’, retorna mensaje de error.
2. Ingresar valor encabezado escrito igual a ‘1+X=3, ¿Cuál es el valor de ‘X?’’, NO retorna mensaje de error.
3. No graba encabezado audible, retorna mensaje de error.
4. Graba encabezado audible, NO retorna mensaje de error.
5. Ingresar respuesta A igual a ‘0’, retorna mensaje de error.
6. Ingresar respuesta A igual a ‘1001’, retorna mensaje de error.
7. Ingresar respuesta A igual a ‘’, retorna mensaje de error.
8. Ingresar respuesta A igual a ‘ad5’, retorna mensaje de error.
9. Ingresar respuesta A igual a respuesta B, retorna mensaje de error.

10. Ingresa respuesta A igual a respuesta C, retorna mensaje de error.
11. Ingresa respuesta A igual a respuesta D, retorna mensaje de error.
12. Ingresa respuesta A igual a '45', NO retorna mensaje de error.
13. Ingresa respuesta A distinto a respuesta B, respuesta C y respuesta D, No retorna mensaje de error.
14. Ingresa respuesta B igual a '0', retorna mensaje de error.
15. Ingresa respuesta B igual a '1001', retorna mensaje de error.
16. Ingresa respuesta B igual a "", retorna mensaje de error.
17. Ingresa respuesta B igual a 'ad5', retorna mensaje de error.
18. Ingresa respuesta B igual a respuesta A, retorna mensaje de error.
19. Ingresa respuesta B igual a respuesta C, retorna mensaje de error.
20. Ingresa respuesta B igual a respuesta D, retorna mensaje de error.
21. Ingresa respuesta B igual a '45', NO retorna mensaje de error.
22. Ingresa respuesta B distinto a respuesta A, respuesta C y respuesta D, No retorna mensaje de error.
23. Ingresa respuesta C igual a '0', retorna mensaje de error.
24. Ingresa respuesta C igual a '1001', retorna mensaje de error.
25. Ingresa respuesta C igual a "", retorna mensaje de error.
26. Ingresa respuesta C igual a 'ad5', retorna mensaje de error.
27. Ingresa respuesta C igual a respuesta A, retorna mensaje de error.
28. Ingresa respuesta C igual a respuesta B, retorna mensaje de error.
29. Ingresa respuesta C igual a respuesta D, retorna mensaje de error.

30. Ingresa respuesta C igual a '45', NO retorna mensaje de error.
31. Ingresa respuesta C distinto a respuesta A, respuesta B y respuesta D, No retorna mensaje de error.
32. Ingresa respuesta D igual a '0', retorna mensaje de error.
33. Ingresa respuesta D igual a '1001', retorna mensaje de error.
34. Ingresa respuesta D igual a '', retorna mensaje de error.
35. Ingresa respuesta D igual a 'ad5', retorna mensaje de error.
36. Ingresa respuesta D igual a respuesta A, retorna mensaje de error.
37. Ingresa respuesta D igual a respuesta B, retorna mensaje de error.
38. Ingresa respuesta D igual a respuesta C, retorna mensaje de error.
39. Ingresa respuesta D igual a '45', NO retorna mensaje de error.
40. Ingresa respuesta D distinto a respuesta A, respuesta B y respuesta C, No retorna mensaje de error.
41. Ingresa tipo respuesta A igual a '', retorna mensaje de error.
42. Ingresa tipo respuesta A igual a 'correcta', NO retorna mensaje de error.
43. Ingresa tipo respuesta A igual a 'incorrecta leve', No retorna mensaje de error.
44. Ingresa tipo respuesta A igual a 'incorrecta grave', NO retorna mensaje de error.
45. Ingresa tipo respuesta B igual a '', retorna mensaje de error.
46. Ingresa tipo respuesta B igual a 'correcta', NO retorna mensaje de error.
47. Ingresa tipo respuesta B igual a 'incorrecta leve', No retorna mensaje de error.
48. Ingresa tipo respuesta B igual a 'incorrecta grave', NO retorna mensaje de error.
49. Ingresa tipo respuesta C igual a '', retorna mensaje de error.

50. Ingresa tipo respuesta C igual a 'correcta', NO retorna mensaje de error.
51. Ingresa tipo respuesta C igual a 'incorrecta leve', No retorna mensaje de error.
52. Ingresa tipo respuesta C igual a 'incorrecta grave', NO retorna mensaje de error.
53. Ingresa tipo respuesta D igual a '', retorna mensaje de error.
54. Ingresa tipo respuesta D igual a 'correcta', NO retorna mensaje de error.
55. Ingresa tipo respuesta D igual a 'incorrecta leve', No retorna mensaje de error.
56. Ingresa tipo respuesta D igual a 'incorrecta grave', NO retorna mensaje de error.
57. Ingresa tiempo estimado igual a '0', retorna mensaje de error.
58. Ingresa tiempo estimado igual a '361', retorna mensaje de error.
59. Ingresa tiempo estimado igual a '', retorna mensaje de error.
60. Ingresa tiempo estimado igual a '1adf', retorna mensaje de error.
61. Ingresa tiempo estimado igual a '45', NO retorna mensaje de error.

Modificar ejercicios: interfaz perteneciente al profesor cuya función es modificar un ejercicio existente en la aplicación, el usuario debe proporcionar una respuesta A, respuesta B, respuesta C, respuesta D, tipo de respuesta A, tipo de respuesta B, tipo de respuesta C, tipo de respuesta D y el tiempo estimado.

Tabla F.6 Casos de Prueba interfaz modificar ejercicios.

Condición de entrada	Clases válidas	Clases inválidas
respuesta A	<ul style="list-style-type: none"> - $1 \leq \text{respuesta A} \leq 1000$ - $\text{respuesta A} \neq \text{respuesta B}$ - $\text{respuesta A} \neq \text{respuesta C}$ 	<ul style="list-style-type: none"> - $\text{respuesta A} < 1$ - $\text{respuesta A} > 1000$ - respuesta A es vacío

	<ul style="list-style-type: none"> - respuesta A != respuesta D 	<ul style="list-style-type: none"> - respuesta A es letra - respuesta A=B - respuesta A=C - respuesta A=D
respuesta B	<ul style="list-style-type: none"> - $1 \leq \text{respuesta B} \leq 1000$ - respuesta B != respuesta A - respuesta B != respuesta C - respuesta B != respuesta D 	<ul style="list-style-type: none"> - respuesta B < 1 - respuesta B > 1000 - respuesta B es vacío - respuesta B es letra - respuesta B=A - respuesta B=C - respuesta B=D
respuesta C	<ul style="list-style-type: none"> - $1 \leq \text{respuesta C} \leq 1000$ - respuesta C != respuesta A - respuesta C != respuesta B - respuesta C != respuesta D 	<ul style="list-style-type: none"> - respuesta C < 1 - respuesta C > 1000 - respuesta C es vacío - respuesta C es letra - respuesta C=A - respuesta C=B - respuesta C=D
respuesta D	<ul style="list-style-type: none"> - $1 \leq \text{respuesta D} \leq 1000$ - respuesta D != respuesta A 	<ul style="list-style-type: none"> - respuesta D < 1 - respuesta D > 1000

	<ul style="list-style-type: none"> - respuesta D != respuesta B - respuesta D != respuesta C 	<ul style="list-style-type: none"> - respuesta D es vacío - respuesta D es letra - respuesta D=A - respuesta D=B - respuesta D=C
tipo de respuesta A	<ul style="list-style-type: none"> - correcta - incorrecta leve - incorrecta grave 	<ul style="list-style-type: none"> - tipo de respuesta A es vacío
tipo de respuesta B	<ul style="list-style-type: none"> - correcta - incorrecta leve - incorrecta grave 	<ul style="list-style-type: none"> - tipo de respuesta B es vacío
tipo de respuesta C	<ul style="list-style-type: none"> - correcta - incorrecta leve - incorrecta grave 	<ul style="list-style-type: none"> - tipo de respuesta C es vacío
tipo de respuesta D	<ul style="list-style-type: none"> - correcta - incorrecta leve - incorrecta grave 	<ul style="list-style-type: none"> - tipo de respuesta D es vacío
tiempo estimado	<ul style="list-style-type: none"> 1<=tiempo estimado<=360 	<ul style="list-style-type: none"> - tiempo estimado <1 - tiempo estimado >360 - tiempo estimado es vacío - tiempo estimado es

		una letra
--	--	-----------

Pruebas de caja negra, interfaz Modificar Ejercicios:

1. Ingresar respuesta A igual a '0', retorna mensaje de error.
2. Ingresar respuesta A igual a '1001', retorna mensaje de error.
3. Ingresar respuesta A igual a '', retorna mensaje de error.
4. Ingresar respuesta A igual a 'ad5', retorna mensaje de error.
5. Ingresar respuesta A igual a respuesta B, retorna mensaje de error.
6. Ingresar respuesta A igual a respuesta C, retorna mensaje de error.
7. Ingresar respuesta A igual a respuesta D, retorna mensaje de error.
8. Ingresar respuesta A igual a '45', NO retorna mensaje de error.
9. Ingresar respuesta A distinto a respuesta B, respuesta C y respuesta D, No retorna mensaje de error.
10. Ingresar respuesta B igual a '0', retorna mensaje de error.
11. Ingresar respuesta B igual a '1001', retorna mensaje de error.
12. Ingresar respuesta B igual a '', retorna mensaje de error.
13. Ingresar respuesta B igual a 'ad5', retorna mensaje de error.
14. Ingresar respuesta B igual a respuesta A, retorna mensaje de error.
15. Ingresar respuesta B igual a respuesta C, retorna mensaje de error.
16. Ingresar respuesta B igual a respuesta D, retorna mensaje de error.
17. Ingresar respuesta B igual a '45', NO retorna mensaje de error.

18. Ingresa respuesta B distinto a respuesta A, respuesta C y respuesta D, No retorna mensaje de error.
19. Ingresa respuesta C igual a '0', retorna mensaje de error.
20. Ingresa respuesta C igual a '1001', retorna mensaje de error.
21. Ingresa respuesta C igual a "", retorna mensaje de error.
22. Ingresa respuesta C igual a 'ad5', retorna mensaje de error.
23. Ingresa respuesta C igual a respuesta A, retorna mensaje de error.
24. Ingresa respuesta C igual a respuesta B, retorna mensaje de error.
25. Ingresa respuesta C igual a respuesta D, retorna mensaje de error.
26. Ingresa respuesta C igual a '45', NO retorna mensaje de error.
27. Ingresa respuesta C distinto a respuesta A, respuesta B y respuesta D, No retorna mensaje de error.
28. Ingresa respuesta D igual a '0', retorna mensaje de error.
29. Ingresa respuesta D igual a '1001', retorna mensaje de error.
30. Ingresa respuesta D igual a "", retorna mensaje de error.
31. Ingresa respuesta D igual a 'ad5', retorna mensaje de error.
32. Ingresa respuesta D igual a respuesta A, retorna mensaje de error.
33. Ingresa respuesta D igual a respuesta B, retorna mensaje de error.
34. Ingresa respuesta D igual a respuesta C, retorna mensaje de error.
35. Ingresa respuesta D igual a '45', NO retorna mensaje de error.
36. Ingresa respuesta D distinto a respuesta A, respuesta B y respuesta C, No retorna mensaje de error.

37. Ingresa tipo respuesta A igual a ‘’, retorna mensaje de error.
38. Ingresa tipo respuesta A igual a ‘correcta’, NO retorna mensaje de error.
39. Ingresa tipo respuesta A igual a ‘incorrecta leve’, No retorna mensaje de error.
40. Ingresa tipo respuesta A igual a ‘incorrecta grave’, NO retorna mensaje de error.
41. Ingresa tipo respuesta B igual a ‘’, retorna mensaje de error.
42. Ingresa tipo respuesta B igual a ‘correcta’, NO retorna mensaje de error.
43. Ingresa tipo respuesta B igual a ‘incorrecta leve’, No retorna mensaje de error.
44. Ingresa tipo respuesta B igual a ‘incorrecta grave’, NO retorna mensaje de error.
45. Ingresa tipo respuesta C igual a ‘’, retorna mensaje de error.
46. Ingresa tipo respuesta C igual a ‘correcta’, NO retorna mensaje de error.
47. Ingresa tipo respuesta C igual a ‘incorrecta leve’, No retorna mensaje de error.
48. Ingresa tipo respuesta C igual a ‘incorrecta grave’, NO retorna mensaje de error.
49. Ingresa tipo respuesta D igual a ‘’, retorna mensaje de error.
50. Ingresa tipo respuesta D igual a ‘correcta’, NO retorna mensaje de error.
51. Ingresa tipo respuesta D igual a ‘incorrecta leve’, No retorna mensaje de error.
52. Ingresa tipo respuesta D igual a ‘incorrecta grave’, NO retorna mensaje de error.
53. Ingresa tiempo estimado igual a ‘0’, retorna mensaje de error.
54. Ingresa tiempo estimado igual a ‘361’, retorna mensaje de error.
55. Ingresa tiempo estimado igual a ‘’, retorna mensaje de error.
56. Ingresa tiempo estimado igual a ‘1adf’, retorna mensaje de error.
57. Ingresa tiempo estimado igual a ‘45’, NO retorna mensaje de error.

Pruebas de Caja negra interfaz alumno.

Ingresar a la aplicación: interfaz de logueo, el usuario debe proporcionar un nombre de usuario y una contraseña valida en el sistema.

Tabla F.7 Casos de Prueba ingresar a la aplicación.

Condición de entrada	Clases válidas	Clases inválidas
nombre de usuario	- nombre de usuario no puede ser vacío.	- nombre de usuario es vacío
contraseña	- contraseña no puede ser vacío.	- contraseña es vacío

Pruebas de caja negra, interfaz Cuenta:

1. Ingresa valor nombre de usuario igual a ‘’, retorna mensaje de error.
2. Ingresa valor nombre de usuario igual a ‘juanignacio’, NO retorna mensaje de error.
3. Ingresa valor contraseña igual a ‘’, retorna mensaje de error.
4. Ingresa valor contraseña igual a ‘perezgomez’, NO retorna mensaje de error.

Pruebas de Integración del sistema.

Tabla F.8 Pruebas de Integración, crear usuario alumno.

Prueba	Crear Usuario Alumno
Descripción	Creación de un nuevo usuario en el Sistema
Entrada	Rut: 21108786 dígito: 2 primer nombre: Antonia

	segundo nombre: Ignacia apellido paterno: salinas apellido materno: Horta fecha de nacimiento: 2002-12-13 tipo de usuario: Alumno user: antoniaignacia password: salinashorta curso: 1 año: 2008
Criterios de Aceptación	Guardar alumno en la base de datos
Resultados	Guardo en la tabla 'Persona': (21108786-2, 4, Antonia, Ignacia, salinas, Horta, Alumno, 2002-12-13, 1) Guardo en la tabla 'Cuenta': (4, antoniaignacia, salinashorta, "') Guardo en tabla 'Alumno': (1, 21108786-2, 1, 2008) Guardo en tabla 'Desempeño_Actividad': (1,3,0,1),(2,3,0,1);(3,3,0,1);(4,3,0,1);(5,3,0,1);(6,3,0,1);(7,3,0,1)

Tabla F.9 Pruebas de Integración, crear nuevo ejercicio.

Prueba	Crear Nuevo Ejercicio
Descripción	Creación de un nuevo ejercicio en el Sistema.
Entrada	tipo de actividad: 5

	<p>encabezado escrito: $32+11=43$ ¿Qué operación se efectuó?</p> <p>encabezado audible: grabar un audio con el encabezado escrito</p> <p>respuesta A: Suma</p> <p>respuesta B: Resta</p> <p>respuesta C: Multiplicación</p> <p>respuesta D: División</p> <p>tipo respuesta A: correcta</p> <p>tipo respuesta B: incorrecta grave</p> <p>tipo respuesta C: incorrecta grave</p> <p>tipo respuesta D: incorrecta grave</p> <p>ranking: 1</p> <p>tiempo estimado: 5</p>
Criterios de Aceptación	<p>Registrar audio</p> <p>Registrar XML</p> <p>Guardar ejercicio en la base de datos</p>
Resultados	<p>Guardo en la tabla 'Ejercicio': (23, 5, 0, 5, 1)</p> <p>Genero un archivo llamado 'nº ejercicio'.mp3 con la grabación del enunciado</p> <p>Genero un archivo llamado 'nº ejercicio'.XML con los siguientes datos:</p> <pre><?XML version="1.0" encoding="UTF-8"?></pre> <pre><Ejercicio></pre>

	<pre> <actividad>5</actividad> <ejercicio>23</ejercicio> <encabezado>32+11=43 ¿ Qué operación se efectuó ?</encabezado> <opciones> <a1>Suma</a1> <respA>corr</respA> <b1>Resta</b1> <respB>IG</respB> <c1>Multiplicación</c1> <respC>IG</respC> <d1>División</d1> <respD>IG</respD> </opciones> </Ejercicio> </pre>
--	--

Tabla F.10 Pruebas de Integración, crear realizar ejercicio.

Prueba	Realizar un ejercicio
Descripción	El usuario se logeará en el sistema, realizara un ejercicio y comprobara si se registraron los datos.
Entrada	usuario: carlitosandres contraseña: lechugatomate

	<p>respuesta ejercicio</p> <p>tiempo respuesta</p>
<p>Criterios de Aceptación</p>	<p>Guardar la sesión de ejercicio.</p> <p>Guardar el desempeño de la actividad</p> <p>Asignar un nuevo ejercicio de acuerdo al nivel del alumno.</p>
<p>Resultados</p>	<p>Guardo en la tabla 'Sesión': (34,3,2008-06-17,16:08:28,16:08:39)</p> <p>Guardo en la tabla 'Desempeño_Actividad': (1,3,1,1)</p> <p>Se asigno un nuevo ejercicio.</p>