

1 Introducción

En el mundo actual las nuevas tecnologías avanzan a pasos agigantados, es por eso, que en este trabajo de título se aprovecha de utilizar una que en distintos sectores del mercado como la banca, telefonía, entretención y otros, ha crecido enormemente, esto se refiere a los dispositivos móviles, fuertemente usados para realización de trámites a distancia, o como oficina móvil y muchos sistemas móviles destinados a que sean mucho más personales, de “bolsillo” y no haya dependencia en llegar al hogar a realizar ciertas tareas en el PC de escritorio o desde un equipo portátil de tamaño considerable como los Notebooks, además de la comunicación entre personas como la telefonía celular que deja de lado los equipos fijos, e incluso en el ámbito de la entretención, como con equipos de almacenamiento y reproducción como los mp3 o mp4.

En este caso se aprovechará el uso de equipos móviles como las PDAs o PCs de bolsillo, además de equipos de telefonía celular, en el área de la educación, desarrollando un sistema de autoevaluación, con el cual se busca facilitar la tarea tanto de profesores como de alumnos, especialmente en momentos cercanos a la realización de una evaluación ya que se intenta mejorar la forma de realizar dichas evaluaciones para darle más espacio a profesores para realizar otras tareas o trabajos y a los alumnos para que por si mismos puedan evaluarse y saber cómo se encuentran realmente a nivel de conocimientos, con respecto a los contenidos de un curso, recibiendo de forma inmediata su nota o evaluación si tener que esperar a que pase un periodo de tiempo considerable antes que el profesor revise y les entregue los resultados, como ocurre en la actualidad y que todos sabemos lo complicado que es realizar una prueba por parte del profesor, que tiene que hacer las preguntas para las pruebas o exámenes con respecto al contenido del curso, cada periodo académico, y después tener que corregirlas una por una, lo que ocupa tiempo considerable, especialmente si el profesor imparte más de un curso, y para los alumnos el tener que estar un tiempo determinado respondiendo de forma escrita la evaluación y después esperar a que el profesor entregue los resultados.

Para conocer antecedentes sobre ciertos temas involucrados en este proyecto, se consulta mediante internet y algunos documentos de apoyo usados en ciertos cursos contenidos dentro de la carrera de Ingeniería de Ejecución Informática de la Pontificia Universidad Católica de Valparaíso.

2 Plan de Trabajo

2.1 Etapa 1:

En la etapa 1 de este trabajo de título se realizó el trabajo que se detalla a continuación:

- 09 – 22 de Marzo 2009: Se realiza investigación sobre posibles proyectos anteriormente realizados con respecto al tema de test auto-evaluativos y su funcionamiento según la forma de ser implementados, siendo aplicaciones de escritorio, vía web o si existe alguna aplicación que trabaje sobre dispositivos móviles, como el caso de este proyecto. Esto lo vemos en la sección 3.1 ‘Estado del Arte’
- 23 – 27 de Marzo 2009: Se plantean los objetivos, General y Específicos, de este trabajo de título (sección 3.2 Objetivos).
- 28 Marzo – 03 de Abril 2009: Toma de Requerimientos y Análisis del trabajo de título, realizando diagramas de Casos de Uso, gráficos y narrativos, tanto de primer nivel como expandidos.
- 04 – 08 de Abril 2009: Se plantea la metodología de desarrollo a utilizar para realizar el trabajo de título.
- 09 – 15 de Abril 2009: Se investigan y muestran distintas Herramientas (J2ME y .NET CF) para solucionar el problema del trabajo de título, con los cuales se puede implementar.
- 17 de Abril 2009: Presentación de Avance Etapa 1 del trabajo de título.
- 20 de Abril – 08 de Mayo 2009: Realización de Diseño del trabajo de título, elaboración de Diagramas de Clases, Secuencias y Actividades.
- 11 de Mayo – 5 de Junio 2009: Desarrollo de primeros prototipos, en ambas herramientas de desarrollo para entorno gráfico y solamente en J2ME con funcionalidades agregadas incrementalmente como el ingreso de usuarios a sus perfiles correspondientes (Profesor y Alumno), que serán mejoradas en los incrementos futuros del desarrollo.
- 08 – 18 de Junio 2009: Captura y selección de imágenes de los prototipos realizados en ambas herramientas, para incluirlas en la presentación de esta primera etapa concluida. Además se realiza corrección y ordenamiento del contenido incluido para la revisión final de la etapa 1 de este trabajo de título.
- 26 de Junio 2009: Finalmente, en la revisión final sobre el trabajo de la Etapa 1, se muestra la resolución del trabajo de título, detallando los requerimientos del funcionamiento del test y también imágenes de los primeros prototipos de esta

aplicación, desarrollados en ambas herramientas de desarrollo para ver las diferencias principalmente gráficas que hay entre ellas.

2.2 Etapa 2:

Continuación del trabajo realizado en la Etapa 1 de este trabajo de título, enfocado en los siguientes puntos:

- 29 de Junio – 10 de Julio / 27 de Julio – 07 de Agosto 2009: Continuar con desarrollo del trabajo de título en la fase de análisis y diseño, revisando y corrigiendo algunos detalles en Diagrama de Clases y otros realizados en la Etapa 1, además de mostrar nuevos Diagramas de Secuencias y Actividades, acorde al avance en el desarrollo del trabajo mediante de la metodología escogida.
- 10 – 21 de Agosto 2009: Realizar un Diagrama Relacional para la Base de Datos a la cual se conectará el programa.
- 24 de agosto – 04 de Septiembre 2009: Elaborar un esquema sobre la arquitectura del Sistema, que muestre la distribución de los distintos participantes (dispositivos móviles, Base de Datos, entre otros).
- 07 – 11 de Septiembre: Continuar con la implementación del Sistema corrigiendo y/o mejorando prototipos tras cada incremento.
- 14 – 23 de Septiembre: Realizar pruebas correspondientes a cada incremento realizado, en aspectos como funcionamiento del programa, además de revisión y ordenamiento de la documentación.
- 25 de Septiembre 2009: Presentación de Avance Etapa 2 del trabajo de título.
- 05 – 16 de Octubre 2009: Corrección de Diagramas de Clases y Relacional, según observaciones en presentación de avance de la Etapa 2.
- 19 – 30 de Octubre 2009: Corrección en Software con respecto a observaciones y/o sugerencias sobre opciones, funcionalidades y detalles como mensajes que debiera realizar el programa para una mejor interacción con el(los) usuario(s), realizadas en presentación de avance de la Etapa 2.
- 02 – 05 de Noviembre 2009: Pruebas finales y búsqueda de solución a problemas con la ejecución del programa.
- 06 – 11 de Noviembre 2009: Nuevos intentos de ejecución del programa hasta encontrar el motivo del problema que había con la ejecución de la aplicación; realizar pruebas a los cambios realizados, sin mucho éxito especialmente en la muestra de avisos de interacción con el usuario, entre otras. Se intenta cargar la aplicación en una PDA para la revisión de Software, vía web, cable USB y Bluetooth, sin lograrlo.

- 13 de Noviembre: Entrega Final Informe Etapa 2 de este trabajo de título.
- 13 – 22 de Noviembre 2009: Realizar correcciones (ortografía y escala evaluativa), mejorar y agregar funcionalidades del programa y avisos al usuario, como ver lista del curso por parte del profesor y dar avisos de respuesta correcta/incorrecta al alumno tras contestar cada pregunta del test y montar la aplicación a una PDA y a un teléfono móvil para la última revisión de Software.

3 Análisis de Objetivos y Metodología

3.1 Objetivos:

3.1.1 General:

Desarrollar un Sistema de Autoevaluación para Dispositivos Móviles, en el ámbito de la educación, utilizando Tecnología Java.

3.1.2 Específicos:

- Estudiar sistemas de autoevaluación del aprendizaje existentes.
- Estudiar tecnologías para el desarrollo de aplicaciones para dispositivos móviles.
- Definir la metodología de trabajo y los requerimientos necesarios para el desarrollo.
- Desarrollar pequeños prototipos con las tecnologías disponibles estudiadas, decidir por una de ellas y desarrollar el sistema final de acuerdo a los requerimientos.
- Probar el sistema en un curso dictado en la carrera, definiendo las evaluaciones que realizará el sistema de acuerdo a los contenidos de dicho curso.

3.2 Metodología y Paradigma de Desarrollo

Para este trabajo de título se utiliza la Metodología de Desarrollo Orientada a Objetos, que consiste en identificar distintos “Objetos” participantes en el funcionamiento de un sistema, agrupando en “Clases” aquellos que posean características y comportamientos similares (por ejemplo, se tiene la Clase “Automóvil”, cuyas características estén dadas por el Tipo, Color y Número de Puertas y su comportamiento las acciones de acelerar, frenar y retroceder; una instancia u Objeto de dicha Clase automóvil está dada por un vehículo en particular cuyas características son por ejemplo, Tipo: Todo Terreno, Color: Rojo y Número de Puertas: 4) por lo que, en software, un “objeto” es una parte de código compuesto de variables (que determinan sus características) y métodos (que implementan su comportamiento) y una “clase” es un modelo que define las variables y métodos comunes a un conjunto de objetos similares. Cada Objeto puede interactuar con otro mediante el envío de mensajes que consisten en invocaciones a los métodos de los objetos. Una de las características más cruciales en la Metodología Orientada a Objetos es la Herencia que consiste en que una Clase puede heredar variables y métodos (Súper Clase o Clase Padre) a otras clases (subclases), las cuales, además de poseer atributos y métodos propios, incorpora los atributos y métodos de la Clase Padre.

Otras características asociadas al método Orientado a Objetos son: la **Abstracción**, que consiste en captar las características y comportamientos esenciales de un objeto y así poder agruparlos en clases los objetos semejantes; **Encapsulamiento**, q consiste en unir en la Clase

características (variables) y comportamientos (métodos), para tener todo en una sola entidad, y cuya utilidad es la facilidad para manejar la complejidad, ya que se tienen las clases como cajas negras, donde se conoce su comportamiento pero sin el detalle interno de cómo lo lleva a cabo; y el **Ocultamiento**, que se refiere a la capacidad, como se menciona anteriormente, de ocultar como cajas negras los detalles internos del comportamiento de una clase y solamente exponer los que sean necesarios para el resto del sistema, con esto el uso de la clase se logra restringir, al tener de forma privada ciertos comportamientos de una clase, los que no podrán ser accedidos por otras clases, y controlar el uso de la ellas, dando ciertos mecanismos para modificar sus estados, en los cuáles se validan que se cumplan ciertas condiciones para llevar a cabo dichos cambios en los estados.

Como Paradigma de Desarrollo se utiliza el modelo Incremental, con el cual se realizan pequeñas “cascadas”, ya que los requerimientos del sistema son conocidos por lo que es mejor para llevar un orden en el desarrollo del trabajo con pasos definidos como lo indica el método cascada y con ello poder cubrir cierta cantidad de requerimientos en una mini cascada por cada iteración, con lo cual el programa se desarrolla tomando en cuenta cada vez más requerimientos y se va incrementando cada vez que se realiza una iteración.

Tras cada iteración se obtiene un prototipo desarrollado a través de una pequeña cascada, el cuál cubre ciertos requerimientos, para luego continuar desarrollando, es decir, se realiza una nueva iteración con otra “cascada”, que cubra un nuevo grupo de requerimientos y así ir incrementando el programa a partir del primer prototipo desarrollado.

Para realizar las iteraciones se utiliza el método incremental, ya que al tener un prototipo listo y que cubre de buena forma un grupo de requerimientos, se continúa el desarrollo sobre el mismo, cubriendo nuevos requerimientos a través de una nueva pequeña “cascada”, para tener un nuevo prototipo evolucionado del anterior, es decir, más grande ya que cubre los requerimientos tanto del prototipo de la iteración anterior como de la actual.

Al utilizar la Metodología Orientada a Objetos, se emplea UML para el modelado, a pesar de que no se trabaja con el paradigma de desarrollo de Proceso Unificado o UP, ya que, siendo que esta metodología se basa en UML para el modelado, no implica que UML sea un lenguaje exclusivo para UP, por lo tanto este lenguaje de modelado es independiente al paradigma que se utilice y por esto, se puede usar para modelar el sistema trabajando con cualquiera de ellos, mediante la metodología de desarrollo Orientado a Objetos, como en el caso de este trabajo de título.

3.3 Tecnologías para solucionar el problema:

[Rubio, 08] y [Rubio, 08a] describen las tecnologías señaladas a continuación.

3.3.1 J2ME (Java 2 Mobile Edition):

Es la única opción real que permite un desarrollo multiplataforma para dispositivos móviles. Tiene una estructura altamente modular para adaptarse a las características de cada dispositivo.

La mayoría de los dispositivos móviles actuales soportan la configuración CLDC (Connected Limited Device Configuration), que, junto con la configuración CDC, define el entorno mínimo necesario para la ejecución de aplicaciones java en un gran conjunto de dispositivos móviles. La configuración CLDC se utiliza para dispositivos cuyos recursos, como por ejemplo, de cálculo, memoria, batería y acceso a la red, son limitados. Además incluye una máquina virtual java reducida (Kilo Virtual Machine -KVM-), homóloga a la máquina virtual Java (Java Virtual Machine -JVM-), y algunos paquetes que se encuentran en configuraciones de java normal, como java.lang, java.io y java.util, pero en versión reducida, además del paquete java.microedition.io. Esta configuración sólo incluye las clases básicas.

Los perfiles definen paquetes adicionales para soportar funcionalidades básicas imprescindibles de cada familia de dispositivos y el perfil más común con la configuración CLDC, es MIDP (Mobile Information Device Profile).

MIDP permite el desarrollo de aplicaciones con interfaz de usuario sencilla, adaptada a dispositivos móviles, posee un API especializado para juegos, permite la persistencia básica, conexión por red y sonido. El mayor éxito de MIDP se debe a la implementación de juegos.

La programación MIDP para la configuración CLDC requiere la instalación de Sun Java Wireless Toolkit o NetBeans 6.7, que poseen un emulador de dispositivos móviles.

3.3.2 .Net Compact Framework:

.NET es un conjunto de tecnologías creadas por Microsoft que posibilitan crear programas y comunicarlos entre sí con independencia del lenguaje de programación y la plataforma en que se ejecuten; para esto, la máquina en que se desea ejecutar el programa debe soportar y tener instalado .NET.

El programa puede ser desarrollado en cualquiera de los lenguajes encontrados en el amplio conjunto de .NET, como son, por ejemplo, Visual Basic .NET, C++, C#, entre otros. Con esto el lema “Escribe en el lenguaje que quieras y ejecútalo en la máquina que quieras”. Para esto, se han creado elementos básicos como:

Common Language Runtime (CLR), que se encarga de ejecutar las aplicaciones, crear el entorno de ejecución en la máquina, entre otras; el código fuente es convertido en código intermedio “standard” y el CLR traduce dicho código intermedio al código máquina de la arquitectura concreta para que el procesador ejecute las instrucciones.

Base Class Library (BCL), que es un conjunto de librerías y clases que usan los lenguajes de programación .NET, que ofrece funcionalidades básicas como vectores, arrays, hasta librerías de Base de Datos.

Por sobre los elementos CLR y BCL, residen otros como Windows Form, para aplicaciones de formularios; ASP.NET, que evoluciona de ASP’s comunes, para la creación de páginas webs dinámicas; y Web Services, para invocar de forma remota a servicios de todo tipo a través de la Web.

Aprovechando esta propiedad de .NET, se pueden crear aplicaciones para móviles de manera más fácil, para lo cual se creó .NET Compact Framework, con esto, se puede programar sin ser especialista en dispositivos móviles, de la misma forma como se programan aplicaciones 'normales' ya que el CLR se ocupa del resto. El problema es que el procesador de un móvil es mucho menos potente por lo que es necesario que el CLR sea reducido, es decir, menos potente que el CLR ordinario. Consecuencia de lo anterior, es que se eliminan clases que requieren mayor proceso o que no se pueden implementar en estos dispositivos, por lo que se reduce la cantidad de clases de sobremesa, aunque de igual forma el conjunto funcional de clases en Compact Framework sigue siendo muy amplio y las características básicas propias de .NET siguen intactas.

Para el desarrollo de aplicaciones existen, entre otras, la herramienta de desarrollo Visual Studio .NET, en sus versiones 2003, 2005 y 2008, que poseen distintos tipos de planillas de proyectos, dentro de las cuáles se encuentra una específica para dispositivos móviles, además poseen funciones útiles como emulador de móviles como PocketPC, el despliegue automático a un dispositivo real, etc. Con Visual Studio se puede ejecutar la aplicación en un dispositivo móvil real, mientras se hace la depuración en el mismo PC.

El lado negativo de esta tecnología, es que para utilizar sus herramientas de desarrollo nombradas anteriormente, se necesita obtener una licencia pagada, al contrario de las herramientas de J2ME que son de uso libre.

NOTA: Todos los Casos de Uso presentados en la sección '3.4 Casos de Uso', están basados en el prototipo realizado bajo la tecnología J2ME (ver sección 3.5, Tecnologías para solucionar el problema), mencionada en la sección anterior, aunque ciertos casos de uso, muestran algunas mejoras que se pueden realizar al prototipo, sobre todo los que están relacionados con el usuario 'Profesor'. Dicho prototipo será presentado más adelante.

4 Estado del arte

A nivel educacional y específicamente para el desarrollo de sistemas de autoevaluación, se han realizado distintos proyectos de investigación y desarrollo, aunque la gran diferencia, y que sin duda entrega algún valor agregado al “Sistema de Autoevaluación para Dispositivos Móviles”, es que ellos han sido implementados para trabajar mediante la Web, o mediante un software de escritorio; por lo tanto, lo nuevo del sistema de autoevaluación de este trabajo de título es que está pensado para utilizarlo en un Dispositivo Móvil. Además, los sistemas señalados son del extranjero por lo que sería de gran utilidad en nuestro país realizar este sistema y más aún si se trata de destinarlo a equipos móviles. A continuación, se presentan algunos ejemplos de dichos sistemas:

4.1 Sistema de Autoevaluación Universidad Abierta y a Distancia (UNAD)

Este es un sistema llamado “*Sistema de autoevaluación y retroalimentación inteligente aplicadas a cursos académicos dentro de un proceso de aprendizaje*”, que ha sido realizado por alumnos de la Universidad Abierta y a Distancia (UNAD, con sedes en Colombia y Estados Unidos), el cual está desarrollado para los cursos virtuales de dicha Universidad, a través de su sitio Web, donde los estudiantes inscritos son protagonistas principales de su formación y, los cursos virtuales, una herramienta pedagógica donde está incluido este sistema de autoevaluación con un grado de inteligencia para que los alumnos interactúen de una manera transparente y vayan conociendo su nivel de aprendizaje a través del sistema auto-evaluativo.

La Figura 4.1 [UNAD, 09] muestra una sección del sitio Web de la Universidad Abierta y a Distancia, exclusiva para que los alumnos registrados ingresen, con su nombre de usuario y contraseña, al sistema de autoevaluación.

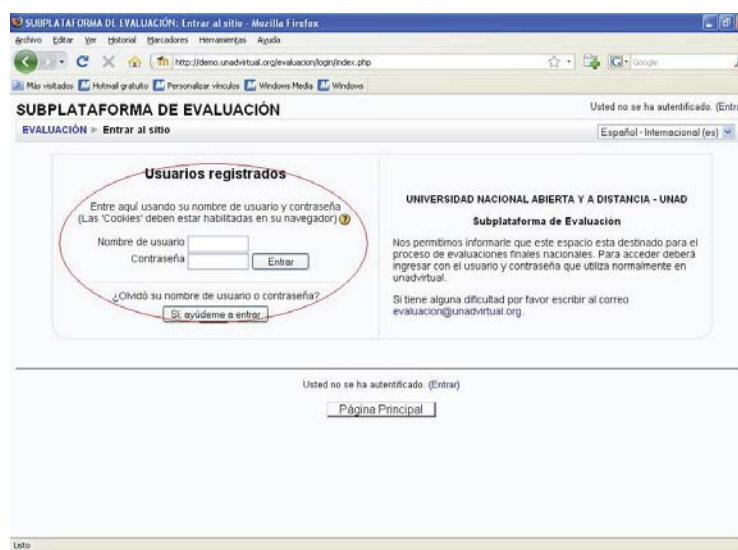


Figura 4.1 Pantalla ingreso a sistema de autoevaluación UNAD

4.2 Sistema de Evaluación “Avaluator”:

También existe un software de escritorio llamado “Avaluator 3.0”, que fue diseñado para evaluar alumnos de distintos niveles. Este sistema es totalmente gratuito, aunque su uso es exclusivo del mundo académico y consta de dos aplicaciones: un editor para formular preguntas de control y un programa que pregunta y evalúa los contenidos formulados por el editor. Además de un conjunto de preguntas correspondientes al ramo de Ciencias Sociales. Su desarrollador es José Fernando Fernández Alcalde [Fernández, 05], Profesor de Informática y Tecnología de la “Escola Cultural” ubicada en Badola (Barcelona), España.

En la Figura 4.2, se ilustra el inicio de este programa, donde muestra su nombre y el del autor. Además, un campo para que el alumno, que quiera realizar la evaluación, ingrese su nombre para entrar al sistema y la opción de ver resultados de controles anteriores.



Figura 4.2 Pantalla inicial “Avaluator”

Una vez ingresado al sistema, pasa a la siguiente pantalla (Figura 4.3) en donde aparecen las instrucciones y las opciones de aumentar o disminuir la cantidad de preguntas que le hará el programa y de realizar la evaluación con o sin un máximo de tiempo, haciendo clic en el reloj de arena, el cual también se puede modificar.

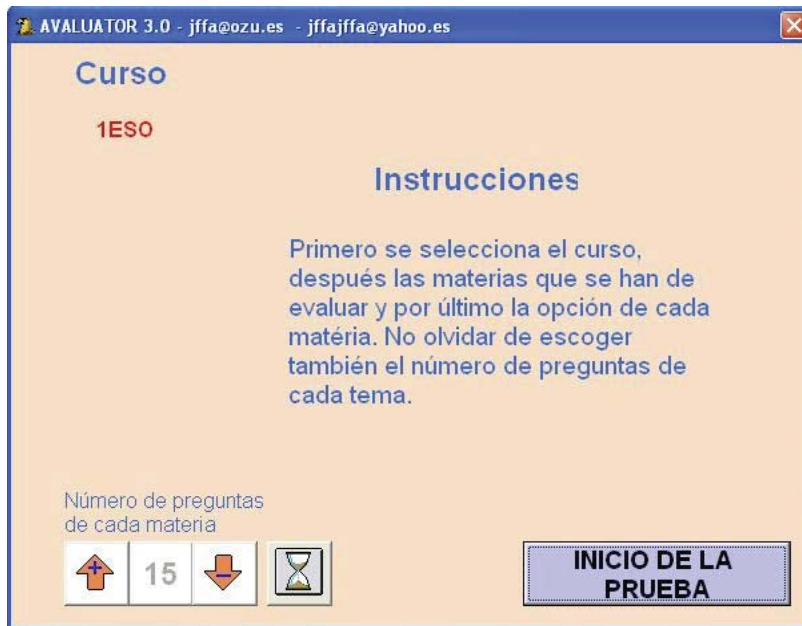


Figura 4.3 Instrucciones de uso

Haciendo clic en uno de los cursos, se ven las materias en donde el alumno debe seleccionar la que le interesa ser evaluado y así poder iniciar la prueba, en este caso tenemos sólo un curso (1ESO) y una materia (Ciencias Sociales), como se muestra en la Figura 4.4. También se ve junto al reloj de arena el campo para editar el tiempo máximo que durará la evaluación, en minutos.



Figura 4.4 Listado de materias de un curso

Como vemos en la Figura 4.5, la pantalla para la evaluación tiene una columna al costado izquierdo, en donde se encuentra la lista de preguntas que tiene que responder el alumno (quien puede pasar de una pregunta a otra seleccionándola en la lista) y a medida que el alumno las responde, nos indica si la respuesta estuvo bien o mal; sobre esta lista hay tres cuadros de distintos colores, los cuales indican la cantidad total de preguntas (color verde), el número de respuestas erróneas (rojo) y respuestas correctas (azul) que lleva el alumno. En la sección principal de la pantalla aparece la pregunta, sobre esta sección hay distintos íconos, uno de ellos el de salida del programa (cuando el alumno elige realizar el test con tiempo, aparece un cronómetro junto al botón de salida, en el caso de la imagen, el test se realiza sin tiempo determinado) y en la parte inferior de la pantalla encontramos un botón de confirmación para saber si la respuesta está bien o mal.

El programa edita y realiza 6 tipos de preguntas y son los siguientes:

1.- Colorear una zona geográfica: Aquí el alumno debe hacer clic en la zona geográfica que la pregunta le pide al alumno, en el caso de la Figura 4.5, se pide colorear Italia.

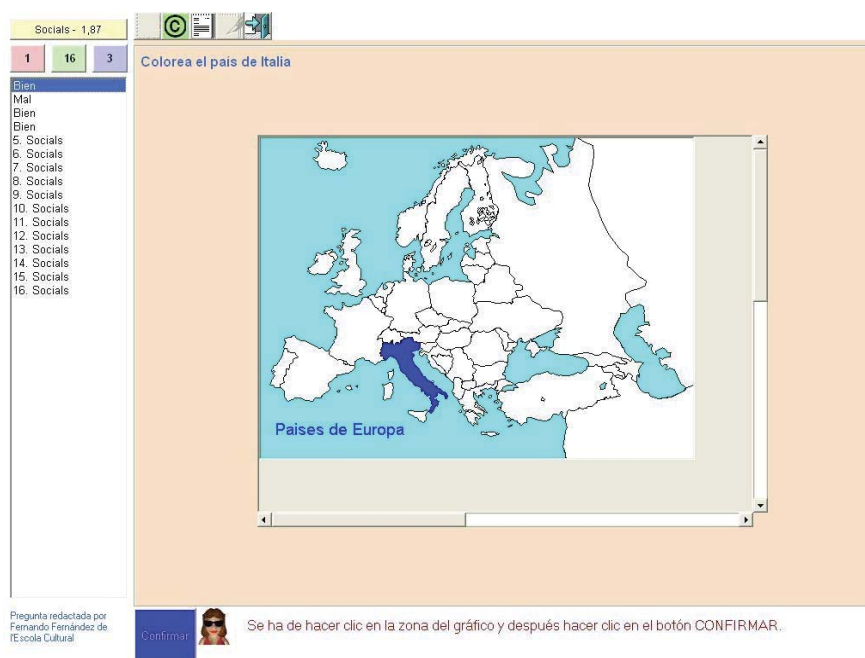


Figura 4.5 Primer tipo de pregunta, colorear una zona geográfica

2.- Preguntas Tipo Test: Muestra la pregunta con distintas alternativas, de las cuales el alumno debe seleccionar la correcta. Las preguntas pueden tener un máximo de 8 alternativas, en el caso de la pregunta que vemos sólo tiene 7 alternativas (Figura 4.6), además podemos ver que el alumno la respondió mal ya que seleccionó la alternativa marcada en color rojo y la respuesta correcta era la indicada en color azul.



Figura 4.6 Segundo tipo de pregunta, tipo test

3.- Comprensión de lectura: Aparece un pequeño texto relacionado con el tema de la materia, del cual se realiza la pregunta y el alumno la contesta indicando la respuesta dentro del mismo texto, como se puede ver en la Figura 4.7.



Figura 4.7 Tercer tipo de pregunta, comprensión de lectura

4.- Pregunta referida a una imagen con respuesta corta: Se presenta al alumno una imagen y la pregunta correspondiente relacionada a ella y el alumno debe responder de forma precisa a dicha pregunta en el campo de respuesta, como se ve en la Figura 4.8.



Figura 4.8 Cuarto tipo de pregunta, referida a una imagen con respuesta corta

5.- Verdadero o Falso: Tal como muestra la Figura 4.9, se presenta una afirmación y el alumno debe indicar si es verdadero o falso lo que se menciona en ella.



Figura 4.9 Quinto tipo de pregunta, verdadero o falso

6.- Con respuesta corta: se presenta la pregunta y el alumno debe responder, de forma precisa, escribiendo en el campo de respuesta (Figura 4.10).



Figura 4.10 Sexto tipo de pregunta, respuesta corta

Además, las preguntas del tipo test (Figura 4.11), por ejemplo, pueden llevar una imagen asociada en forma de apoyo gráfico para ayudar y guiar al alumno con respecto a la pregunta que se le está realizando.



Figura 4.11 Pregunta tipo test con imagen

Al finalizar la evaluación, aparece el resultado de la prueba realizada por el alumno, como un aviso por sobre la última pregunta contestada (Figura 4.12).



Figura 4.12 Muestra de resultados de la prueba en “Avaluator”

Junto con el programa evaluador, se incluye un “*Editor de Preguntas*”, para poder agregar nuevas preguntas al “*Avaluator*”, también nuevas materias y nuevos cursos, añadiendo nuevas carpetas para separar los cursos y/o materias incluidas. En la Figura 4.13 se observa la pantalla para ingresar una nueva pregunta tipo test, donde se ve que se puede agregar el material gráfico o textual de apoyo (botón amarillo, donde se lee “Añadir txt, gif, jpg”). Para agregar preguntas de alguno de los otros tipos, se debe seleccionar uno de los botones coloridos y enumerados, que corresponda al tipo de pregunta que se desea añadir, ubicados en la parte inferior.

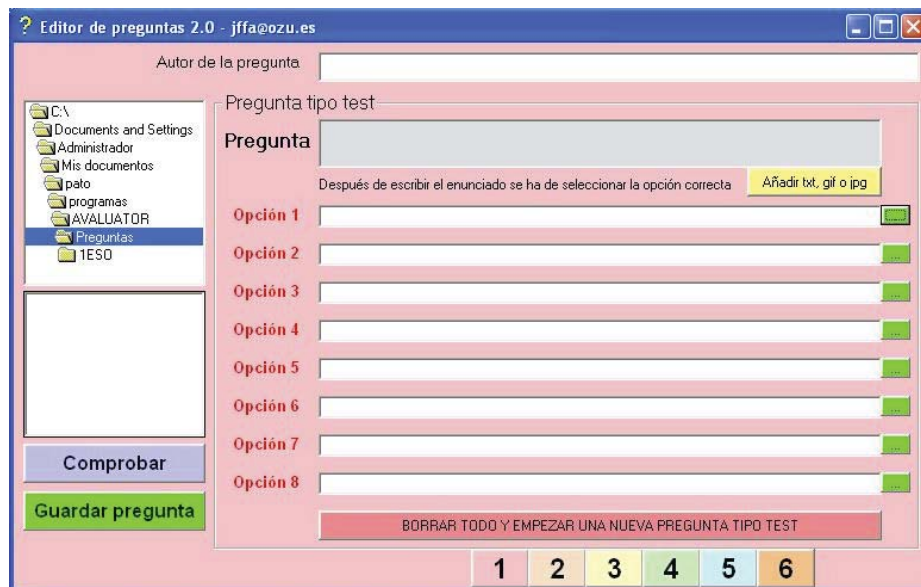


Figura 4.13 Editor de preguntas

4.3 “ALEVIN”:

Para este “Sistema de Autoevaluación” se puede utilizar un algoritmo que tenga ‘inteligencia’ para evaluar las respuestas y seleccionar la siguiente pregunta en base a ellas, este tipo de algoritmo ya existe y se llama “ALEVIN” (“*Algoritmo de Evaluación Inteligente*”), el cuál ha sido desarrollado en lenguaje Java, por Ingenieros de la Universidad de Alcalá (España), motivados por la propuesta de un sistema de autoevaluación sobre móviles, aunque no es abierto y sólo se puede obtener información más detallada sobre él en el libro llamado “ALEVIN – Algoritmo de Evaluación Inteligente”, de la Editorial Universidad de Alcalá de Henares.

Para obtener mayor información con respecto de “ALEVIN”, se intentó reiteradas veces tener comunicación directa, vía correo electrónico, con personal de la Universidad de Alcalá, del área de Ingeniería Informática, específicamente con su secretaria y también directamente con el Director de la Escuela Técnica Superior de Ingeniería Informática, Sr. León González Sotos, quién fue la única persona que respondió a la petición de ayuda por información sobre este tema, aunque no haya sido del todo exitosa ya que la respuesta obtenida fue: “he reenviado su mensaje al profesor Dr. José María Gutiérrez, del departamento de Ciencias de la Computación de esta Universidad (de Alcalá) que creo es la persona adecuada para ayudarlo”¹, pero lamentablemente sin correr la misma suerte de recibir una respuesta como ocurrió con el Director de la Escuela.

Por lo tanto, en este trabajo de título no se utilizará el término de ‘inteligencia’, por lo que el uso de dicho término se limita al “ALEVIN” (“Algoritmo de Evaluación Inteligente”) desarrollado en la Universidad de Alcalá, por ser parte del nombre de ese algoritmo y del cual no hay mayor información.

Este algoritmo se basa en la Teoría de Respuesta al Ítem y los Test Adaptativos Informatizados, que será explicado a continuación:

4.3.1 Test Adaptativos Informatizados (TAI):

En la actualidad, la labor de los profesores a la hora de la calificación de los exámenes y las preguntas escritas se hace difícil pues deben dedicarle mucho tiempo a esta tarea. Y cuando se debe medir la calidad de los estudiantes dado un criterio, el proceso se hace muy lento y trabajoso. Por ejemplo, si se necesita escoger los mejores estudiantes para un proyecto determinado, se tendría que recurrir seguramente al criterio de sus profesores, o a los resultados en las pruebas (que también es un número considerable); ambas variantes consumen mucho tiempo y recursos. Este proceso se hará cada vez más engorroso a medida que aumente el número de estudiantes y proyectos en los que ellos deberán trabajar.

¹ León González Sotos. Director Escuela Técnica Superior de Ingeniería Informática, Universidad de Alcalá, España. Viernes 05, Junio, 2009. Solicitud de mayor información sobre “ALEVIN”, vía correo electrónico (director.etsii@uah.es)

[Martínez, 09] indica que la utilización de los TAIs permitirá agilizar los procesos de confección, aplicación y revisión de los exámenes; evaluar a los estudiantes sin la intervención continua de los profesores; certificarlos académicamente en el conocimiento de una aplicación o técnica determinada y; la elaboración y calificación automática de pruebas para la autoevaluación de los estudiantes.

Las universidades cada día se inclinan más a la educación no presencial y la formación profesional, y a la certificación y evaluación mediante el uso de las nuevas tecnologías.

4.3.2 Teoría de Respuesta al Ítem (TRI):

La *Teoría de Respuesta al Ítem* (TRI), *Teoría del Rasgo Latente* (TRL) o *Teoría de Respuesta al Reactivo* (TRR), es un desarrollo reciente de la *psicometría*, y como tal se refiere también a un modelo matemático propuesto para medir el funcionamiento mental humano.

El origen de la *Teoría de Respuesta al Ítem* se asocia con el trabajo del matemático danés Georg Rasch (1960), y con libro publicado en 1968 por los norteamericanos Lord y Novick, aunque parece cada vez más claro que ideas muy semejantes ya habían sido expuestas antes, incluso en los fundadores del campo (Louis Leon Thurstone, 1912 y Alfred Binet, 1905). Expositores recientes recalcan que esta teoría busca ser Aristotélica, en el sentido de poner más atención a los datos y la manera como los seres humanos responden a instrumentos de medición.

Psicometría:

El término *psicometría* puede referirse a lo siguiente:

- *Psicometría*, una rama de la psicología dedicada, por un lado, a medir rasgos de personalidad, capacidades mentales, nivel de conocimientos y estados de opinión o actitud, y, por otro lado, a la aplicación e investigación de los test y cuestionarios, con el objetivo de generar pautas de construcción y depuración de estas herramientas de medición que garanticen unos mínimos de validez, fiabilidad y consistencia.
- *Método psicométrico*, que se entiende como el conjunto de procedimientos que conducen a la valoración cuantitativa de los fenómenos psicológicos.

5 Análisis del Sistema

5.1 Requerimientos Funcionales

Este sistema requiere ayudar a mejorar la forma de evaluar a los alumnos, automatizando la manera de realizar dichas evaluaciones, dejando de lado las pruebas o exámenes escritos en papel para los que el profesor debe destinar tiempo considerable tanto para diseñarlos como para corregirlos, especialmente si los cursos son numerosos, sabiendo que cada profesor imparte más de un curso por período.

De esta forma, ayudar a los alumnos a que puedan evaluar por sí mismos sus conocimientos, dándose cuenta de forma personal y de inmediato sobre su nivel de aprendizaje con respecto a las materias presentadas en los cursos y que sean evaluadas por este test.

Para esto, el sistema deberá permitir el ingreso de dos tipos de usuario: profesor, que se encarga de la parte del curso, administrar tanto los alumnos, sus notas y la parte evaluativa; y alumno, que ingresa al sistema para realizar el test de autoevaluación, sea en forma de ensayo u obligatorio, y ver un pequeño historial sobre sus últimos resultados. Para ello, tanto profesor como alumno, debe ingresar mediante usuario y contraseña.

Por el lado del profesor, el sistema debe realizar las siguientes funciones, mediante el dispositivo móvil:

- Mostrar por pantalla la lista de alumnos inscritos en la base de datos del sistema, mostrando su nombre y la nota obtenida en el test por cada uno de ellos.
- Ingresar un alumno a la base de datos del sistema, capturando datos como: nombre, apellido(s), Rut y contraseña que le será asignada al alumno (dos veces).
- Eliminar algún alumno de la base de datos, a través del nombre de usuario (Rut), ingresado por el profesor, del alumno que se desea eliminar del sistema.
- Desbloquear o dar el permiso para que un alumno, en caso de que ya haya realizado el test, lo pueda rendir nuevamente.
- Determinar el número máximo de veces que un alumno puede realizar el test obligatorio.

Por parte del alumno, el sistema debe realizar las siguientes funciones:

- Mostrar el test en pantalla, esto de manera pausada, es decir, una pregunta con su encabezado y las 5 alternativas a la vez (la quinta alternativa contiene la opción de “omitir”).
- Tras haber mostrado una pregunta, el alumno debe seleccionar su respuesta entre las 5 alternativas mostradas.

- Una vez seleccionada e ingresada la respuesta, el sistema debe validar dicha respuesta, mediante la alternativa elegida por el alumno y el campo de la alternativa correcta en la base de datos, debe comparar ambos valores.
- Opcionalmente podría mostrar en pantalla un mensaje indicando si se respondió correcta o incorrectamente la pregunta, dependiendo de la comparación hecha entre la respuesta y el valor del campo que indica la respuesta correcta.
- El sistema debe seleccionar aleatoriamente la siguiente pregunta, de acuerdo a lo respondido por el alumno, debe elegir una pregunta de mayor, igual o menor nivel de dificultad, para lo que debe realizarse un algoritmo inteligente, el cuál en base a la validación de las respuestas que haya realizado el alumno, seleccione la siguiente pregunta acorde al nivel del alumno.
- Mostrar la siguiente pregunta con sus 5 alternativas en pantalla, una vez seleccionada por el algoritmo inteligente.
- Una vez que el alumno ha responde un total de 10 preguntas, el sistema debe calcular la nota final de acuerdo al puntaje acumulado durante el test.
- Mostrar en pantalla el resultado obtenido por el alumno.
- En caso de haber realizado el test modo obligatorio, bloquear internamente al alumno para que no pueda realizar el test nuevamente, en caso de haber alcanzado el máximo de test oficiales permitidos por el profesor.
- En caso de que el mismo alumno ingrese al test modo obligatorio después de haberlo realizado el máximo de veces permitidas, se debe mostrar un aviso en pantalla que le indique al usuario que no puede volver a realizar el test en la opción obligatorio (a no ser que el profesor lo haya desbloqueado).

5.2 Casos de uso:

En este apartado se presentan diversos diagramas de casos de uso realizados para el mejor análisis del sistema desarrollado para este trabajo de título, pasando primero por el de alto nivel, luego los diagramas expandidos, que muestran con más detalle cada caso de uso del diagrama de alto nivel, y finalmente los casos de uso se verán aún más detallados en su forma narrativa.

5.2.1 Alto Nivel:

A continuación se detalla en las siguientes tablas cada caso de uso presentado en el Diagrama de Casos de Uso de la Figura 5.1

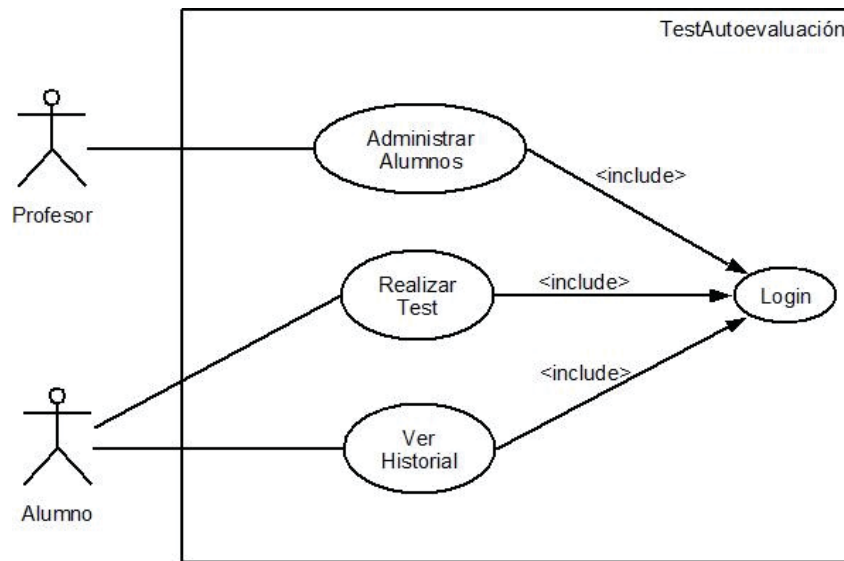


Figura 5.1 Diagrama de Casos de Uso Alto Nivel TestAutoevaluación

Tabla 5.1 Caso de Uso Narrativo Alto Nivel 'Login'

Caso de uso: Login
Participantes: Profesor, Alumno
Tipo: Primario (<include> en los demás)
Descripción: Para ingresar al sistema, cualquiera sea el perfil de usuario, tanto profesor como alumno, debe ingresar su nombre de usuario y contraseña, así, dependiendo el tipo de usuario que sea, podrá acceder a realizar sus tareas correspondientes.

Tabla 5.2 Caso de Uso Narrativo Alto Nivel 'Administrar Alumnos'

Caso de uso: Administrar Alumnos
Participantes: Profesor
Tipo: Primario
Descripción: Si el Profesor ingresa como 'admin' debe registrarse con sus datos personales para luego ingresar con su Rut y contraseña. Una vez que ingresa con su usuario y contraseña, el profesor puede administrar a los alumnos de su curso en el sistema, para lo cual puede ingresar un nuevo alumno, eliminar algún alumno, ver la lista de alumnos con sus notas, además de darle el permiso a algún alumno que requiera realizar el test nuevamente y determinar cuántas veces pueden realizar el test como máximo.

Tabla 5.3 Caso de Uso Narrativo Alto Nivel ‘Realizar Test’

Caso de uso: Realizar Test
Participantes: Alumno
Tipo: Primario
Descripción: Tras haber ingresado al sistema con su nombre de usuario y contraseña, el alumno tiene la opción de rendir el test obligatorio (en caso de que no lo haya realizado el máximo de veces posible) o de ensayo (que no cuenta dentro de la cantidad de test posibles que puede realizar); en caso del test obligatorio, si el alumno ya realizó el test el máximo de veces, tendrá que tener el permiso del profesor para realizarlo de nuevo y que éste haya desbloqueado el sistema para dicho alumno y así pueda realizar el test nuevamente, si el profesor no entrega ese permiso (dentro de las opciones de su perfil), el sistema le avisa al alumno que no puede ingresar al test porque ya lo realizó. Una vez que el alumno ha contestado las preguntas del test, el sistema le mostrará su calificación.

Tabla 5.4 Caso de Uso Narrativo Alto Nivel ‘Ver Historial’

Caso de uso: Ver Historial
Participantes: Alumno
Tipo: Primario
Descripción: Además de realizar el test de modo obligatorio o ensayo, el alumno puede ver el registro de sus puntajes obtenidos en los últimos test obligatorios realizados.

5.2.2 Expandidos:

En el diagrama de casos de uso de la Figura 5.2, se muestran con más detalle las tareas del usuario ‘Profesor’, detallando narrativamente cada caso de uso del diagrama en las tablas presentadas a continuación:

Tabla 5.5 Caso de Uso Narrativo Expandido ‘Listar Alumnos’

Caso de uso	Listar alumnos
Participantes	Profesor
Tipo	Primario
Descripción	Esta opción es para que el profesor pueda ver la lista de alumnos del curso para saber quiénes han rendido el test, sus datos y las notas de todos los alumnos inscritos, hayan o no realizado la autoevaluación.
Referencias Cruzadas	Funciones de registro, ingresar Rut

Tabla 5.5 Caso de Uso Narrativo Expandido 'Listar Alumnos'

Curso Normal de dos Eventos	
Profesor	Sistema
1.- Usuario ingresa con su nombre de usuario (Rut) y contraseña.	2.- Al comprobar que el usuario es un 'profesor', se muestra el menú de las tareas que puede realizar el profesor.
3.- Profesor selecciona opción de Listar Alumnos (o curso).	4.- Sistema lista los alumnos registrados, mostrando su nota correspondiente, en caso los que ya rindieron el test.
Cursos Alternativos	
Paso 2.- La clave es incorrecta. Aviso de clave incorrecta.	
Paso 4.- No hay alumnos. Aviso de lista de curso vacía.	

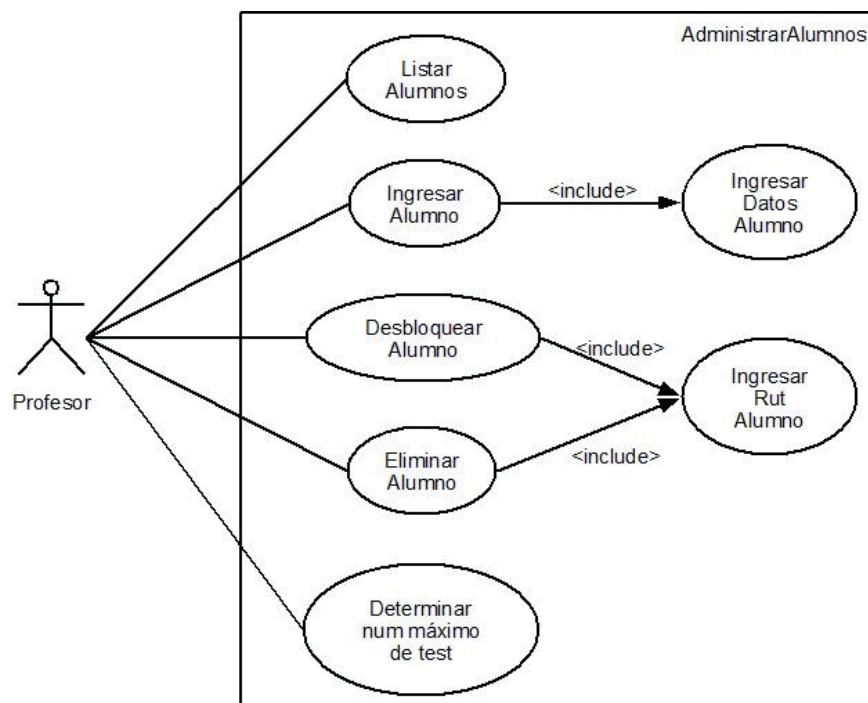


Figura 5.2 Diagrama de Casos de Uso Perfil Profesor

Tabla 5.6 Caso de Uso Narrativo Expandido 'Ingresar Alumno'

Caso de uso	Ingresar Alumno
Participantes	Profesor
Tipo	Primario
Descripción	El profesor tiene la opción de ingresar nuevos alumnos a la base de datos del curso y así estén inscritos en el sistema para poder rendir el test. Para esto debe ingresar los datos del alumno como nombre, apellidos, Rut y la contraseña que le dará al alumno para que pueda ingresar a rendir el test (caso de uso 'Ingresar Datos Alumnos', indicado como <include>).
Referencias Cruzadas	Funciones de registro, Ingresar Datos de Alumno.
Curso Normal de dos Eventos	
Profesor	Sistema
1.- Usuario ingresa con su nombre de usuario (Rut) y contraseña. 3.- Profesor selecciona opción de Ingresar Alumno. 5.- Profesor ingresa datos solicitados	2.- Al comprobar que el usuario es un 'profesor', se muestra el menú de las tareas que puede realizar el profesor. 4.- Sistema solicita datos del alumno, como nombre, apellido(s), Rut y clave que se le asignará. 6.- Sistema guarda los datos ingresados y avisa que el ingreso ha sido satisfactorio.
Cursos Alternativos	
Paso 2.- La clave es incorrecta. Aviso de clave incorrecta. Paso 5.- Algún dato mal ingresado (por ejemplo el Rut), En Paso 6.- dar aviso de datos mal ingresados. Paso 6.- Problemas para guardar los datos. Aviso de error al guardar los datos.	

Tabla 5.7 Caso de Uso Narrativo Expandido 'Desbloquear Alumnos'

Caso de uso	Desbloquear Alumnos
Participantes	Profesor
Tipo	Primario
Descripción	En caso de que el profesor lo estime, puede desbloquear a un alumno, ingresando su Rut, y darle permiso para realizar el test nuevamente, ya que el sistema "bloquea" a los alumnos que realizan el test.
Referencias Cruzadas	Funciones de registro, ingresar Rut.

Tabla 5.7 Caso de Uso Narrativo Expandido 'Desbloquear Alumnos'

Curso Normal de dos Eventos	
Profesor	Sistema
<p>1.- Usuario ingresa con su nombre de usuario (Rut) y contraseña.</p> <p>3.- Profesor selecciona opción de Desbloquear Alumno.</p> <p>5.- Profesor ingresa el Rut del alumno a quien dejará realizar nuevamente el test.</p>	<p>2.- Al comprobar que el usuario es un 'profesor', se muestra el menú de las tareas que puede realizar el profesor.</p> <p>4.- Sistema solicita Rut del alumno.</p> <p>6.- Sistema busca el Rut y desbloquea al alumno asociado. Aviso de Alumno puede volver a realizar el test.</p>
Cursos Alternativos	
<p>Paso 2.- La clave es incorrecta. Aviso de clave incorrecta.</p> <p>Paso 5.- Rut mal ingresado. En Paso 6.- dar aviso de Rut no válido.</p> <p>Paso 6.- No se encuentra el Rut ingresado. Aviso de Alumno no encontrado.</p> <p>Paso 6.- Alumno no había realizado el test. Aviso de alumno no ha sido bloqueado por el sistema, disponible para realizar test.</p>	

Tabla 5.8 Caso de Uso Narrativo Expandido 'Eliminar Alumno'

Caso de uso	Eliminar Alumno
Participantes	Profesor
Tipo	Primario
Descripción	Tal como puede ingresar alumnos, el profesor también puede eliminar alumnos de la base de datos ingresando el Rut, de acuerdo a si el alumno está realmente inscrito en el curso o si el alumno ya no sigue en el curso por retiro o, incluso, si es que quedó registrado en la base de datos de un período académico anterior.
Referencias Cruzadas	Funciones de registro, Ingresar Rut.
Curso Normal de dos Eventos	
Profesor	Sistema
<p>1.- Usuario ingresa con su nombre de usuario (Rut) y contraseña.</p> <p>3.- Profesor selecciona opción de Eliminar Alumno.</p> <p>5.- Profesor ingresa el Rut del alumno que desea eliminar.</p>	<p>2.- Al comprobar que el usuario es un 'profesor', se muestra el menú de las tareas que puede realizar el profesor.</p> <p>4.- Sistema solicita Rut del alumno.</p> <p>6.- Sistema busca el Rut y elimina al alumno asociado. Aviso de Alumno eliminado.</p>

Tabla 5.8 Caso de Uso Narrativo Expandido ‘Eliminar Alumno’

Cursos Alternativos
Paso 2.- La clave es incorrecta. Aviso de clave incorrecta. Paso 5.- Rut mal ingresado. En Paso 6.- dar aviso de Rut no válido. Paso 6.- No se encuentra el Rut ingresado. Aviso de Alumno no encontrado. Paso 6.- Problema al eliminar datos. Aviso de error al eliminar o no se puede eliminar los datos.

Tabla 5.9 Caso de Uso Narrativo Expandido ‘Determinar num máximo de test’

Caso de uso	Determinar num máximo de test	
Participantes	Profesor	
Tipo	Primario	
Descripción	Si bien, por defecto el alumno puede realizar una sola vez el test obligatorio para que el sistema lo deje bloqueado, el profesor tiene esta opción para cambiar el máximo de veces que el alumno pueda realizar el test tipo obligatorio.	
Curso Normal de dos Eventos		
Profesor	Sistema	
1.- Usuario ingresa con su nombre de usuario (Rut) y contraseña.	2.- Al comprobar que el usuario es un ‘profesor’, se muestra el menú de las tareas que puede realizar el profesor.	
3.- Profesor selecciona opción de Eliminar Determinar num máximo de test.	4.- Sistema solicita número para determinar el máximo de test que puede realizar un alumno.	
5.- Profesor ingresa número que estime conveniente.	6.- Almacena número ingresado por Profesor.	
Cursos Alternativos		
Paso 2.- La clave es incorrecta. Aviso de clave incorrecta. Paso 5.- Número mal ingresado. En Paso 6.- dar aviso de número no válido. Paso 6.- Problema al almacenar dato ingresado. Aviso de error no se puede eliminar los datos.		

A continuación, en la Figura 5.3, vemos con más detalle la tarea de Realizar test por parte del usuario ‘Alumno’, detallándose narrativamente en las tablas que siguen:

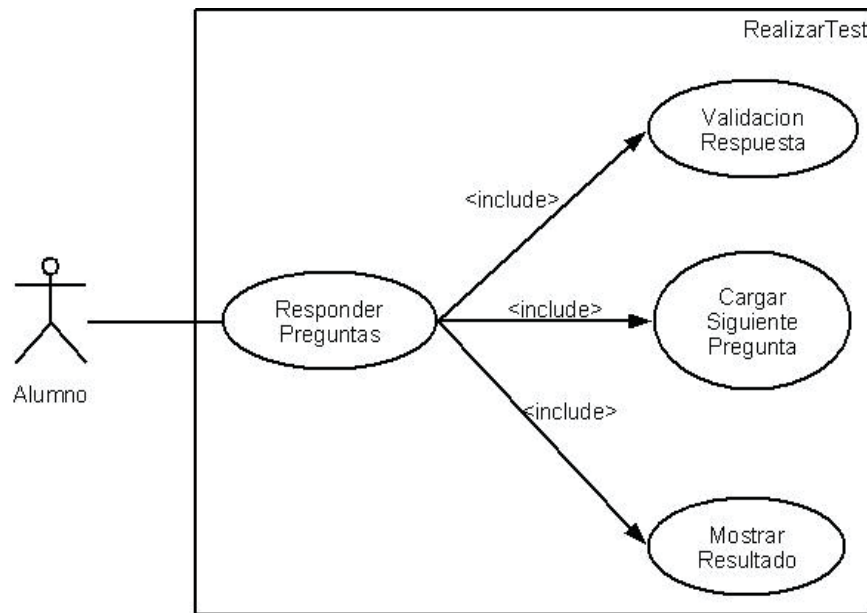


Figura 5.3 Diagrama de Casos de Uso Perfil Alumno

Tabla 5.10 Caso de Uso Narrativo Expandido 'Responder Preguntas'

Caso de uso	Responder Preguntas
Participantes	Alumno
Tipo	Primario
Descripción	Aquí el Alumno está limitado a responder las preguntas del test que le van apareciendo en pantalla, una tras otra, a medida que va contestando, para lo cual el sistema debe validar la respuesta que da el alumno a la pregunta visualizada en pantalla, consultando en la base de datos el campo que indica la alternativa correcta, y en función de la respuesta, entregar un mensaje de respuesta correcta/incorrecta, si es correcta en el mensaje se indica el puntaje obtenido tras responder correctamente la pregunta y si es incorrecta se indica cual es la alternativa de la respuesta correcta a esa pregunta, y luego cargar de manera aleatoria, dentro de las preguntas del nivel que corresponda, la siguiente pregunta que se mostrará al alumno. Una vez realizado el test, es decir, que el alumno ha contestado la última pregunta, el sistema hace la validación de la respuesta, muestra el mensaje de respuesta correcta/incorrecta y luego el resultado obtenido en el test.
Referencias Cruzadas	Funciones de registro, validación de respuestas, selección de preguntas, cálculo de puntuación.

Tabla 5.10 Caso de Uso Narrativo Expandido 'Responder Preguntas'

Curso Normal de dos Eventos	
Profesor	Sistema
<p>1.- Usuario ingresa con su nombre de usuario (Rut) y contraseña.</p> <p>3.- Alumno responde la pregunta que se le muestra en pantalla, seleccionando una de las 4 alternativas.</p>	<p>2.- Al comprobar que el usuario es un 'alumno', comienza el test, mostrando la primera pregunta en pantalla.</p> <p>4.- Sistema evalúa la respuesta seleccionada por el alumno.</p>
	<p>5.- Muestra detalles al alumno sobre la respuesta entregada:</p> <ul style="list-style-type: none"> - Si es correcta, entrega el puntaje obtenido. - Si la respuesta es incorrecta, se muestra la pregunta realizada con su respectiva respuesta correcta. <p>6.- Muestra en pantalla la siguiente pregunta de acuerdo a la respuesta entregada a la pregunta anterior:</p> <ul style="list-style-type: none"> - Si está correcta, seleccionará una pregunta de igual o mayor nivel de dificultad. - Si la respuesta entregada es incorrecta, selecciona una pregunta de igual o inferior nivel a la anterior.
<p>7.- Pasos 3.- a 6.- se repite hasta la pregunta número 9.</p> <p>8.- Para la pregunta número 10, se repiten los pasos 3.-, 4.- y 5.-, continuando el curso en el siguiente paso (9.-)</p>	
	<p>9.- Calcula el puntaje total realizado por el alumno tras responder la décima pregunta y final del test.</p> <p>10.- Muestra en pantalla la nota obtenida por el alumno en el test.</p>
Cursos Alternativos	
<p>Paso 2.- La clave es incorrecta. Aviso de clave incorrecta.</p> <p>Paso 2.- Alumno ya rindió anteriormente el test. Muestra aviso en pantalla diciendo que el alumno ya rindió el test.</p> <p>Paso 2.- Usuario no registrado (es decir, si no está en la lista de alumnos). Muestra aviso de usuario incorrecto.</p> <p>La ocurrencia de alguno de los casos anteriores, impide el ingreso al sistema para realizar el test.</p>	

6 Diseño e Implementación de Prototipo

6.1 Diseño:

Los siguientes diagramas, tanto de Clases como de Secuencia, están diseñados acorde al prototipo presentado más adelante, en la sección 6.2 de este documento.

6.1.1 Diagrama de Clases:

Muestra las clases por las que será desarrollado el programa (Figura 6.1), el cuál consta de un Usuario, de cuya clase heredan otras dos, Profesor y Alumno, que representan los perfiles que puede tomar un usuario del programa. También encontramos las clases Test y Preguntas. Con esto se muestra que un Usuario comienza el programa en la clase más general que es Test, donde debe identificarse para poder ingresar a su perfil correspondiente. Si ingresa con perfil de Profesor, se encarga de administrar a los Alumnos del curso y también las Preguntas. En caso de que ingrese un usuario con perfil Alumno, se mantendrá en el Test, para realizar su evaluación mediante las preguntas que Test selecciona desde el conjunto de Preguntas guardadas en el sistema.

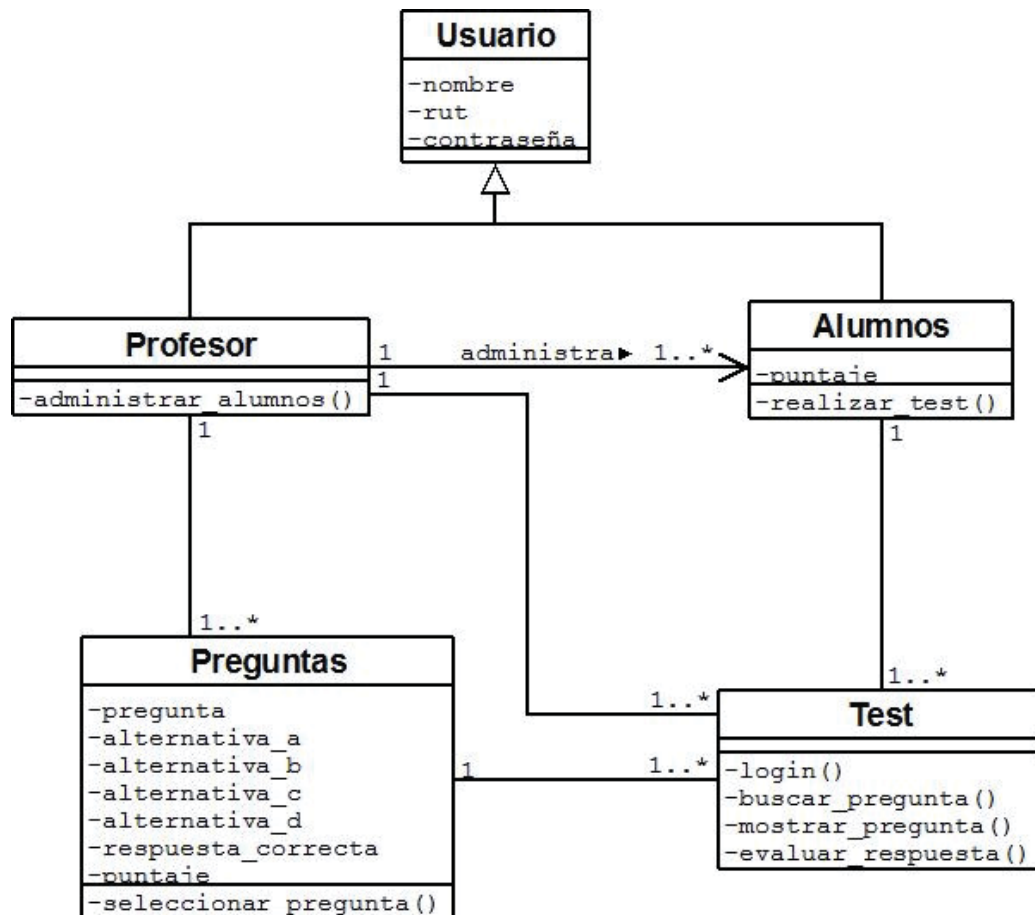


Figura 6.1 Diagrama de Clases para prototipo del sistema

6.1.2 Diagrama de Secuencia:

Este diagrama (Figura 6.2) está relacionado con la parte central y más importante del sistema, es decir, en cómo trabaja de forma secuencial el programa y cómo interactúan las clases participantes, cuando un alumno realiza el test. Lo cual comienza cuando el alumno ingresa al sistema, éste le muestra la primera pregunta y de ahí va respondiendo secuencialmente cada una de las preguntas que el sistema le muestra, una tras otra, a medida que va respondiendo. Una vez respondida la décima pregunta, el sistema guarda el puntaje y lo muestra por pantalla.

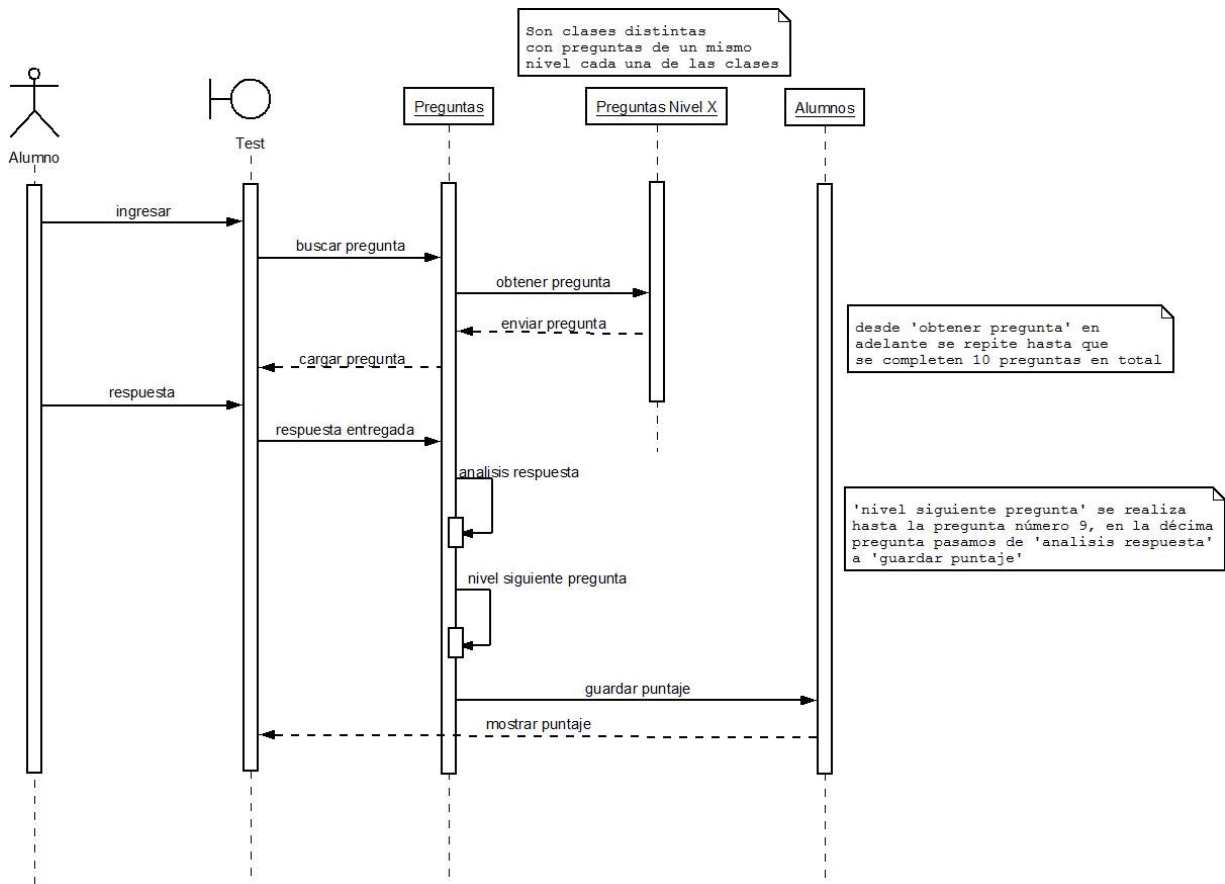


Figura 6.2 Diagrama de Secuencia perfil Alumno para prototipo del sistema

6.1.3 Diagrama de Actividades:

Aquí, en la Figura 6.3, se muestra la interacción entre el alumno y el sistema, cuando dicho usuario realiza el test, y las tareas o actividades efectuadas por cada uno de los participantes.

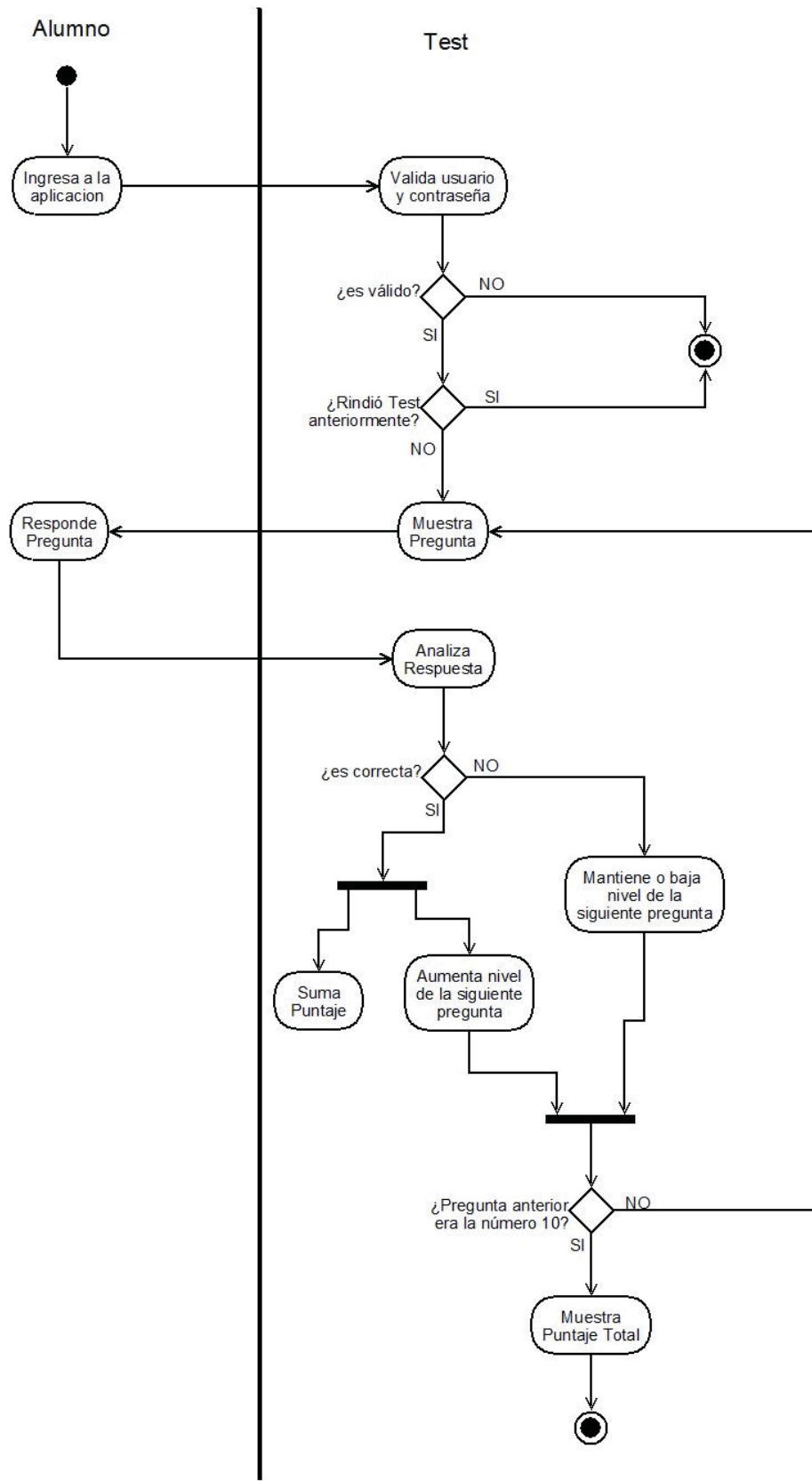


Figura 6.3 Diagrama de Actividades perfil Alumno para prototipo del sistema

6.2 Resolución:

6.2.1 Test

El Test de Autoevaluación, consiste en realizar a los alumnos un total de 10 preguntas de selección múltiple, con 4 alternativas de respuesta (más una quinta para omitir), las cuales se muestran una a una por pantalla, seleccionadas de un universo de 100 preguntas predefinidas, divididas en cinco (5) grupos de distintos niveles de dificultad, y dependiendo de la respuesta marcada por el alumno se elige la siguiente pregunta que se realizará, por ejemplo, si la respuesta entregada por el alumno es correcta, se le indica que ha contestado correctamente y el nivel de dificultad de la pregunta siguiente será más elevado, en caso contrario, se le indicará que su respuesta ha sido incorrecta, dándole a conocer la respuesta correcta para que tenga conciencia de su equivocación y sepa sus falencias tras cada pregunta, y se le presentará una nueva pregunta pero con el mismo o menor nivel de dificultad que el de la pregunta anterior; en caso de que seleccione la opción ‘Omitir’, se le indica la respuesta correcta a la pregunta omitida y la siguiente pregunta tendrá el mismo nivel que la anterior. Cada pregunta aparecerá en la pantalla con sus 4 alternativas más una quinta opción de “omitir” y serán seleccionadas al azar desde un conjunto amplio de preguntas, como se ha mencionado anteriormente, con el fin de que no se repita la misma pregunta a un alumno que se encuentre realizando el test, además de evitar que se repita la misma serie de preguntas cada vez que se ejecute la aplicación.

Una vez que el alumno termina completamente de responder el test, es decir, tras responder las 10 preguntas, se obtiene el puntaje total que el usuario consiguió dependiendo de las respuestas entregadas a cada pregunta, si fueron correctas o incorrectas.

OBSERVACIÓN: Las preguntas que se incluyen en este programa corresponden a materia del curso de Laboratorio de Programación, en carrera de Ingeniería de Ejecución Informática, de la Pontificia Universidad Católica de Valparaíso, cuyo curso trata sobre la programación en Lenguaje C.

El almacenamiento tanto de dichas preguntas, como de los datos de los usuarios (profesor y alumnos), será en una base de datos central, para poder acceder a ella desde distintos dispositivos móviles, teniendo en cuenta que cada alumno y profesor tendrá en su poder un equipo móvil y así poder utilizar la aplicación a nivel de curso(s).

6.2.2 Prototipo J2ME

En esta sección se presenta gráficamente la resolución, a nivel de prototipo, de este trabajo de título, desarrollado con la tecnología J2ME, con imágenes del sistema obtenidas por el emulador de un dispositivo móvil, incluido en la herramienta Wireless Toolkit.

En la Figura 6.4, se puede ver el emulador mencionado anteriormente, cuya pantalla muestra el inicio del programa de evaluación.



Figura 6.4 Pantalla de Presentación “Sistema de Autoevaluación Sobre Móviles”

Como se muestra en la Figura 6.5, luego de la pantalla de inicio, se tiene la sección donde el usuario se debe identificar, con su nombre de usuario (en este caso el Rut) y su contraseña.



Figura 6.5 Ingreso de Usuario “Sistema de Autoevaluación Sobre Móviles”

Si se ingresa con el perfil de Profesor, se ven las opciones que el profesor puede realizar en este sistema, de las cuales debe seleccionar la que desea ejecutar (Figura 6.6).

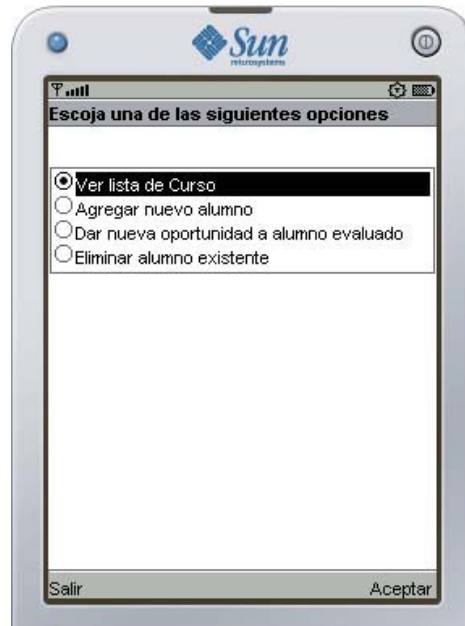


Figura 6.6 Pantalla principal Perfil Profesor

En caso de seleccionar, por ejemplo, la opción de ‘Agregar nuevo Alumno’, pasa a la siguiente pantalla (Figura 6.7), donde debe ingresar, al igual que en la pantalla de registro de nuevo administrador, el nombre, apellido, Rut y la contraseña que le asignará al alumno. Una vez ingresados los datos y pulsar ‘Siguiente’, el programa mostrará un aviso de ‘Alumno ingresado exitosamente’.



Figura 6.7 Ingreso Alumno Nuevo

Por otro lado, al ingresar al programa como Alumno, se le muestran dos opciones para rendir el test, si quiere realizarlo tipo ensayo o tipo oficial (Figura 6.8), en este último caso, su puntuación queda disponible al profesor; lo anterior siempre que el alumno no haya dado el test oficial anteriormente o esté ‘desbloqueado’ para darlo nuevamente, además de la opción de ver sus últimos puntajes obtenidos.

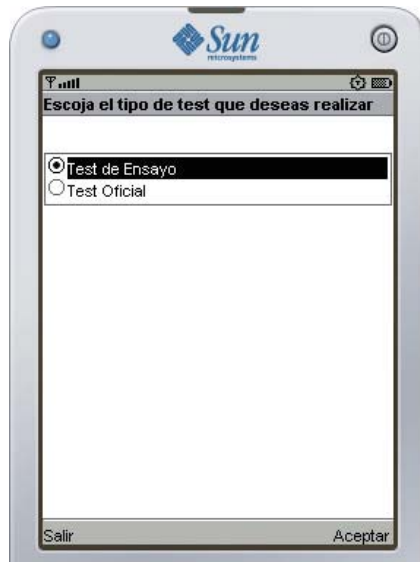


Figura 6.8 Pantalla principal Perfil Alumno

Una vez que el alumno selecciona el tipo de test que desea realizar, aparece inmediatamente la primera pregunta con sus 4 alternativas, y una quinta con la opción de omitir (Figura 6.9), de las cuales el alumno debe seleccionar la correcta o simplemente escoger “Omitir”, y pulsar ‘Siguiente’.

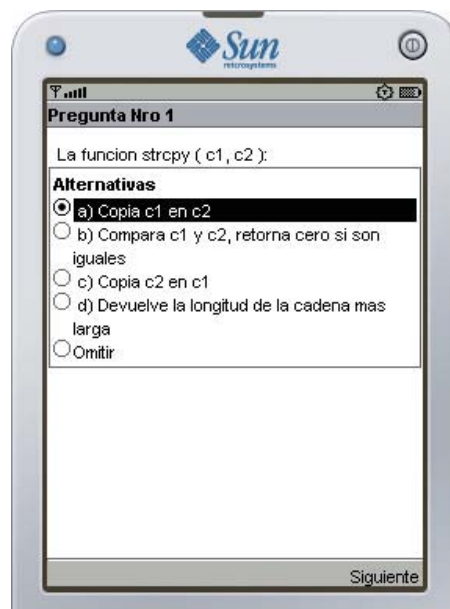


Figura 6.9 Presentación de las Preguntas realizadas por el sistema

Una vez terminado el test, es decir, después de responder las 10 preguntas, el programa muestra el resultado obtenido por el alumno, como se muestra en la siguiente imagen (Figura 6.10).



Figura 6.10 Presentación de resultado obtenido tras realizar test

6.2.3 Prototipo .NET Compact Framework

Al igual que el prototipo en J2ME, se desarrolló uno en .NET utilizando la herramienta Visual Studio 2008 (el cual tiene el mismo funcionamiento que el prototipo Java) y a continuación se muestran algunas imágenes de dicho prototipo en .NET Compact Framework.

En el caso de la Figura 6.11, muestra la sección donde el usuario se debe identificar, con su nombre de usuario (en este caso el Rut) y su contraseña.



Figura 6.11 Ingreso de Usuario “Sistema de Autoevaluación Sobre Móviles” .Net

Al ingresar con el perfil de Profesor, se pueden ver las opciones que el profesor puede realizar en este sistema, de las cuales debe seleccionar la que desea ejecutar (Figura 6.12).



Figura 6.12 Pantalla principal Perfil Profesor .Net

En el caso de ingresar al programa como Alumno, y siempre que éste no haya dado el test anteriormente o esté 'desbloqueado' para darlo nuevamente, la aplicación muestra de inmediato la primera pregunta del test con sus 4 alternativas, de las cuales el alumno debe seleccionar la correcta y pulsar 'Siguiente', como se puede ver en la siguiente interfaz gráfica de la pantalla que muestra la pregunta y sus alternativas (Figura 6.13).



Figura 6.13 Presentación de la Preguntas realizadas por el sistema .Net

Cabe mencionar que la diferencia que hay entre ambos prototipos es que visualmente el de Java, puede ejecutarse en blanco y negro o a color dependiendo de la máquina virtual que tenga incorporado el dispositivo móvil, en el caso del emulador de NetBeans utilizado para las imágenes anteriormente vistas, el programa se visualiza en colores; por otro lado, en el caso del prototipo en .NET vemos que es colorido, por la facilidad que entrega Visual Studio para añadir colores además de pequeñas imágenes y por la forma de trabajo en dicha herramienta, que es como si se desarrollara una aplicación de escritorio con los colores comunes entregada por la herramienta y que se pueden ver en diversos programas desarrollados con esta tecnología.

Otro punto tal vez menos relevante, pero no menos importante al tratarse de desarrollar un buen programa, es que existe un problema de cierta forma notorio con la no visualización de un caracter, lo que dificulta tomar la decisión por una de las 4 alternativas, ya que esto ocurre en algunas alternativas de ciertas preguntas incluidas en el sistema. Específicamente es el caracter '&', el cual no se visualiza cuando se antepone a una palabra (en este caso cuando en lenguaje de programación C se utiliza '&' ante una variable), mejor veamos un ejemplo:

En el caso de que se muestre una sentencia lógica en lenguaje C como:

```
while(a && b){  
    ...
```

Se puede ver en pantalla de nuestro programa el caracter '&'.
Pero cuando tenemos otra sentencia como:

```
&variable...;
```

Al tener dicha sentencia como parte de una alternativa en alguna pregunta, no se puede ver en pantalla de nuestro programa de autoevaluación dicho caracter, sino que se ve el primer caracter de la palabra que le sigue, subrayado.

En lugar de que se vea de esta forma:

a) **&**variable...;

Se visualiza de la siguiente manera:

a) variable...;

Lo anterior es un detalle a corregir para que quien realice el test mediante el sistema desarrollado en .NET no tenga problemas o confusiones al momento de seleccionar la respuesta que crea correcta en caso de que le aparezca una pregunta con alternativas que tengan dicha sentencia.

Si durante el desarrollo de este proyecto no se logra corregir dicho detalle, solo será desarrollado en Java, pero de todas maneras intentando corregir lo que ocurre en .Net para tener también un sistema en esa tecnología.

7 Diseño e Implementación de Sistema Final

7.1 Diseño

Se continúa con el diseño del proyecto, incluyendo nuevos diagramas a los vistos anteriormente en la sección 6.1.

7.1.1 Diagrama de Clases

El siguiente diagrama de clases (Figura 7.1) es el definitivo para el desarrollo de este trabajo de título porque muestra un cambio con respecto al visto anteriormente, ya que se agrega la clase 'Curso' para que el programa funcione con alumnos y preguntas de distintos cursos y no solamente con una lista de alumnos como se muestra en el diagrama de clases y en los prototipos presentados anteriormente. En este diagrama vemos que tanto la clase Profesor como Alumnos interactúan con el programa mediante la clase test, siendo ésta la que da ingreso al profesor a un curso en particular para ver sus opciones de administrar el curso y en caso de que un alumno entre al programa, la clase test ingresa a las preguntas, a través de Curso, de forma transparente al usuario, mostrándole las preguntas del test que debe responder el alumno.

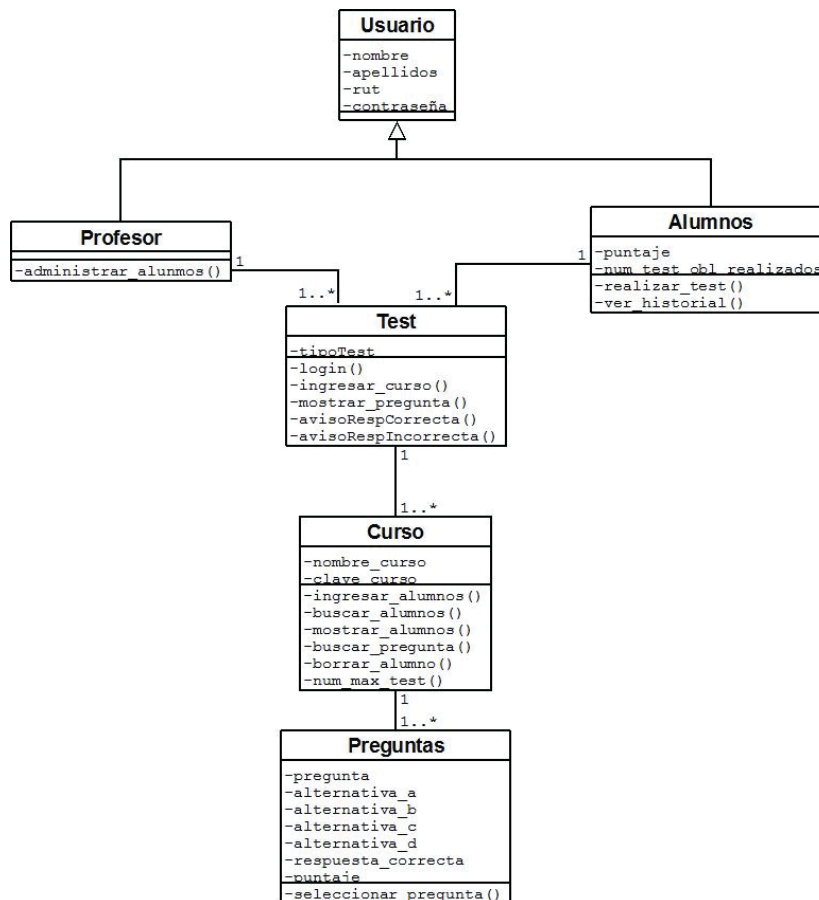


Figura 7.1 Diagrama de Clases Sistema de Autoevaluación

7.1.2 Diagramas de Secuencia

- a) El siguiente diagrama de secuencias (Figura 7.2), enfocado a la interacción de las clases con el alumno cuando ingresa a realizar el test, muestra una pequeña variación con respecto al diagrama de este tipo mostrado anteriormente, ya que al agregar la clase 'Curso' en el diagrama de clases, este diagrama también debió sufrir esa modificación. Al haber dos opciones de realizar el test (ensayo y obligatorio), la tarea en que el sistema bloquea al alumno para que no vuelva a realizar el test ocurre solamente cuando el alumno entra a realizar el test en la opción de obligatorio y cuando el número de test obligatorios que ha realizado sea igual al máximo de test obligatorios permitidos por el profesor, que por defecto es uno (1).

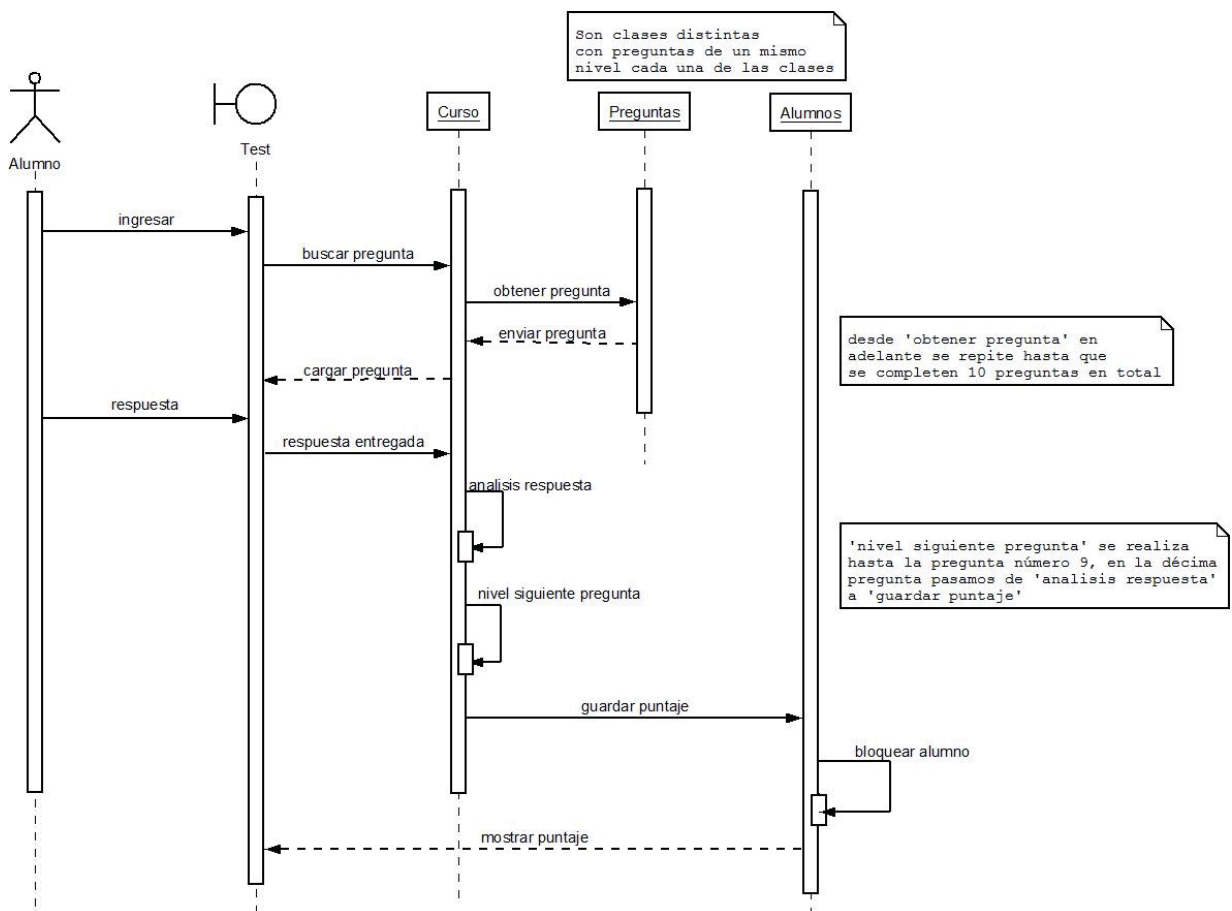


Figura 7.2 Secuencia Realizar Test

- b) El diagrama de secuencia que continúa, muestra la interacción de las clases cuando ingresa el usuario 'Profesor', específicamente a la opción de ver la lista de alumnos del curso donde se muestran datos del alumno como el nombre, apellido(s) y sus notas correspondientes en caso de que hayan realizado el test (Figura 7.3).

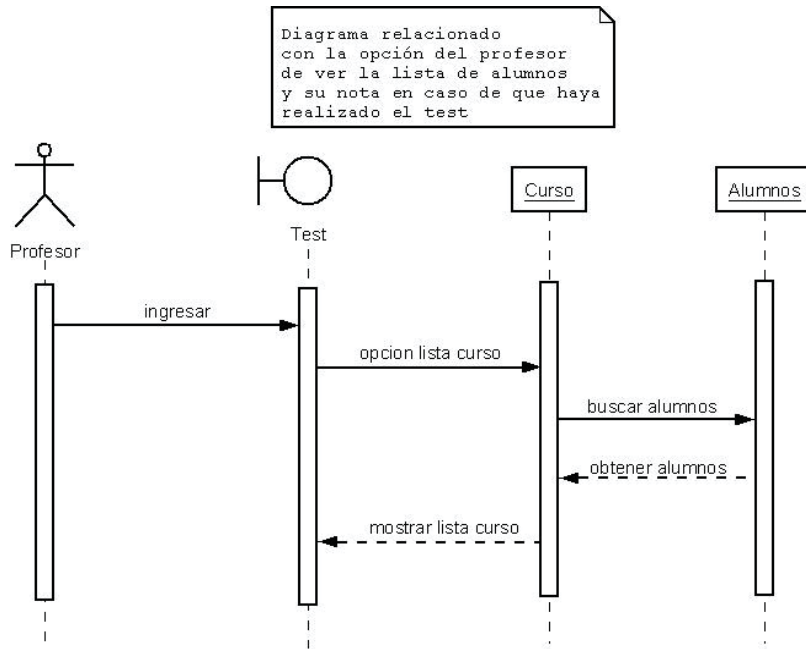


Figura 7.3 Secuencia Listar Alumnos

- c) En la Figura 7.4 se muestra el diagrama de secuencia relacionado con el ingreso del usuario “Profesor” a su opción de Ingresar un alumno nuevo a la base de datos del sistema, donde el profesor debe registrar los datos del alumno como su Nombre, Apellido(s), Rut y la Contraseña que le asignará.

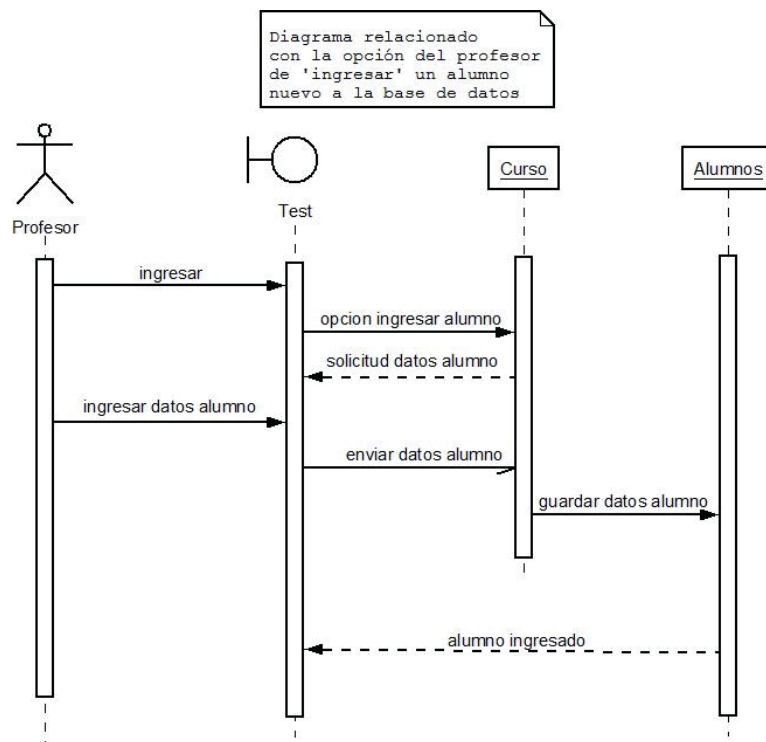


Figura 7.4 Secuencia Ingresar Alumno

- d) A continuación, se presenta el diagrama de secuencia correspondiente a la opción en que el profesor solicita “desbloquear” o dar permiso a un alumno para que pueda realizar el test obligatorio nuevamente (Figura 7.5), ingresando el Rut del alumno a quién se le dará esa oportunidad de repetir el test, previamente ingresando al curso correspondiente, para que luego el alumno sea buscado mediante el Rut ingresado por el profesor.

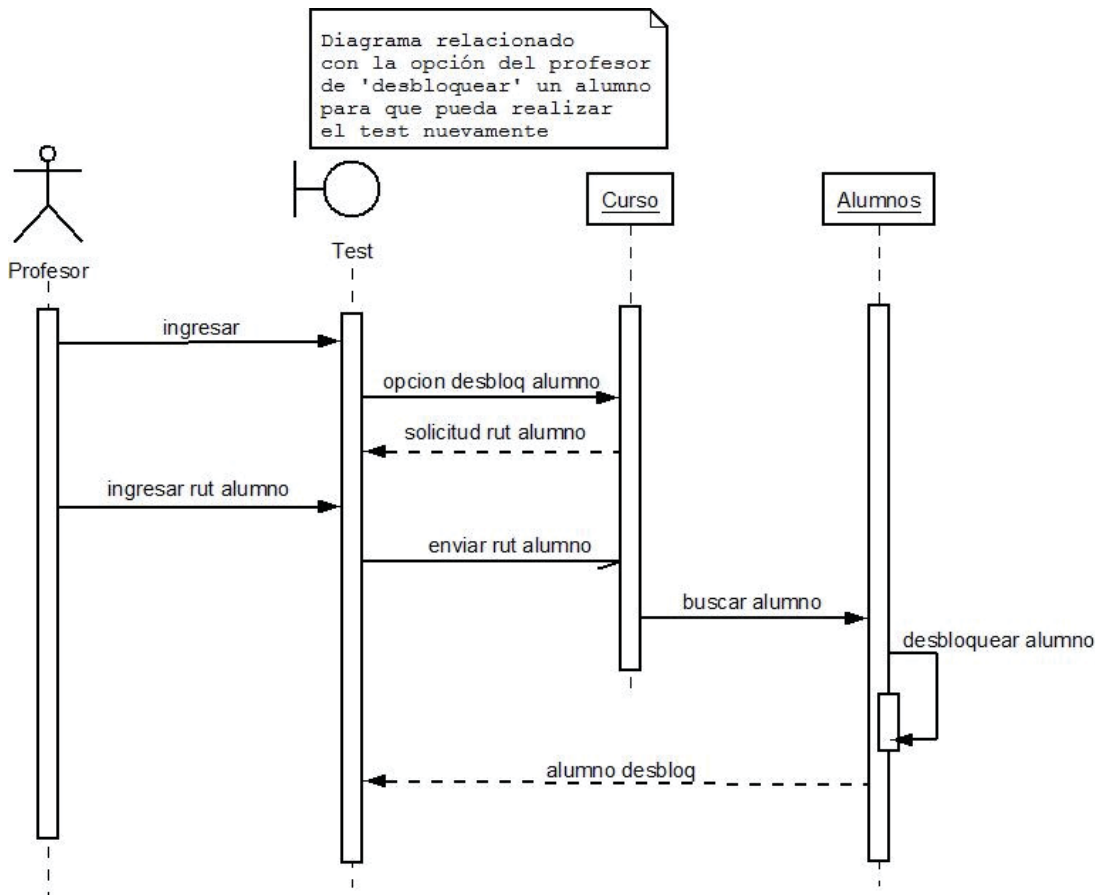


Figura 7.5 Secuencia Desbloquear Alumnos

- e) El siguiente diagrama de secuencia (Figura 7.6), está relacionado con la opción de Eliminar un alumno de la base de datos, correspondiente al perfil del Profesor, quién debe ingresar el Rut del alumno a través del cual será buscado y borrado de la base de datos.
- f) En la siguiente Figura 7.7 se muestra el diagrama de secuencia relacionado con la opción del profesor de Determinar el número máximo de test obligatorios que pueden realizar los alumnos.

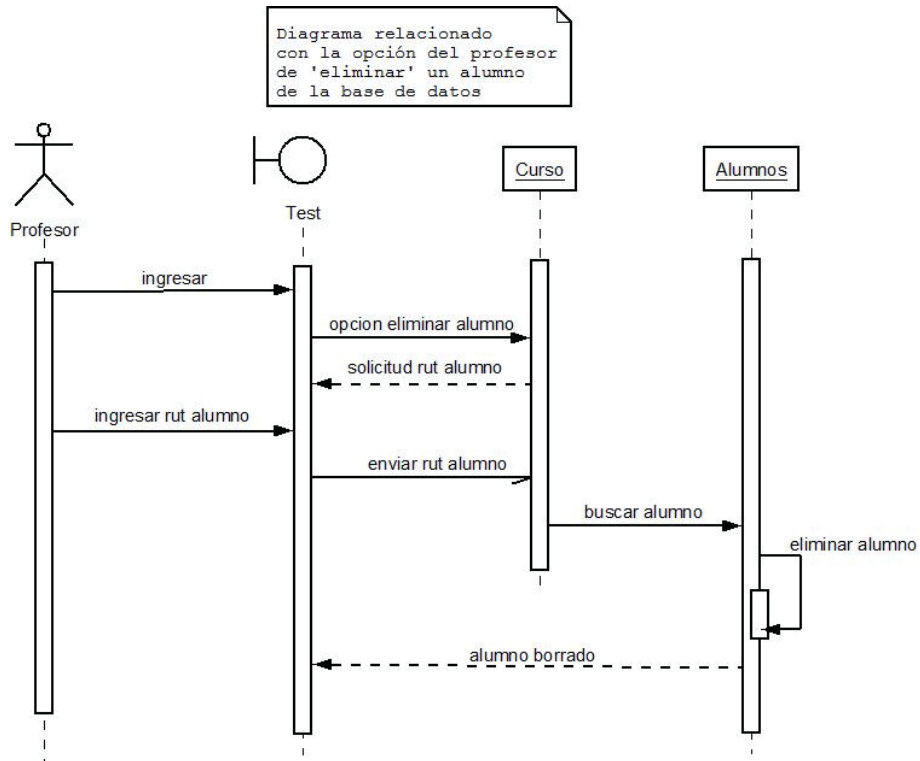


Figura 7.6 Secuencia Eliminar Alumnos

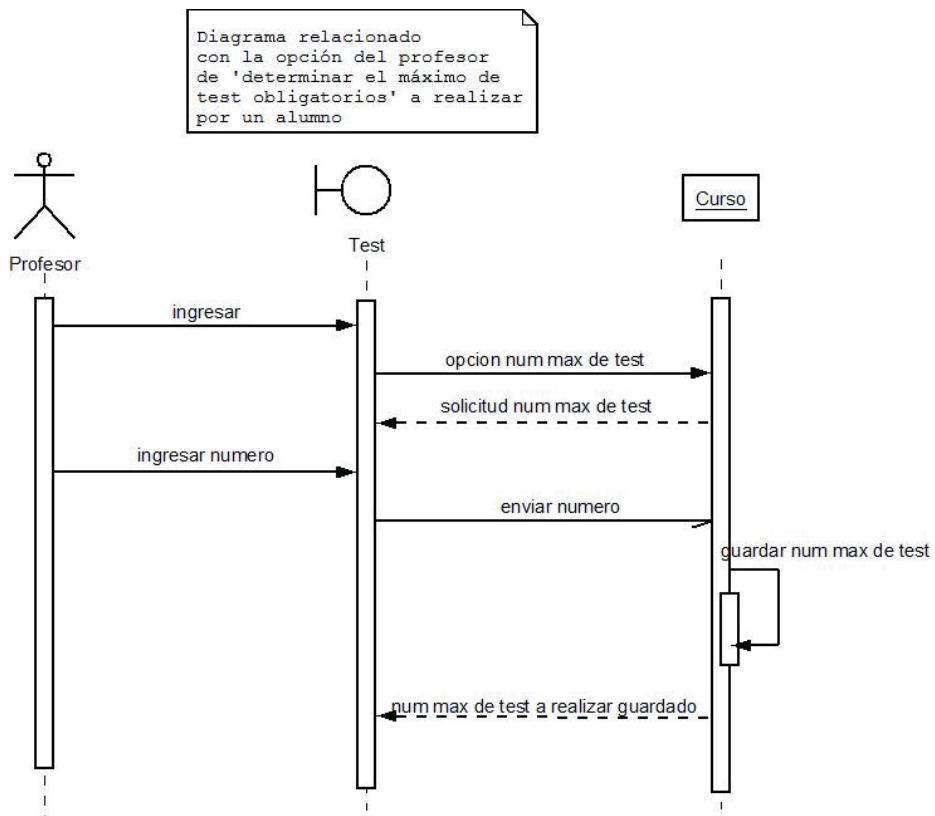


Figura 7.7 Secuencia Determinar Número Máximo de Test Obligatorios

7.1.3 Diagramas de Actividades

- a) El siguiente diagrama de actividades (Figura 7.8) muestra una pequeña variación al diagrama de este tipo mostrado en la sección 6.1.3, a partir de la variación en el diagrama de clases. Recordemos que en este diagrama se muestra la interacción de las clases con el alumno cuando ingresa a realizar el test.

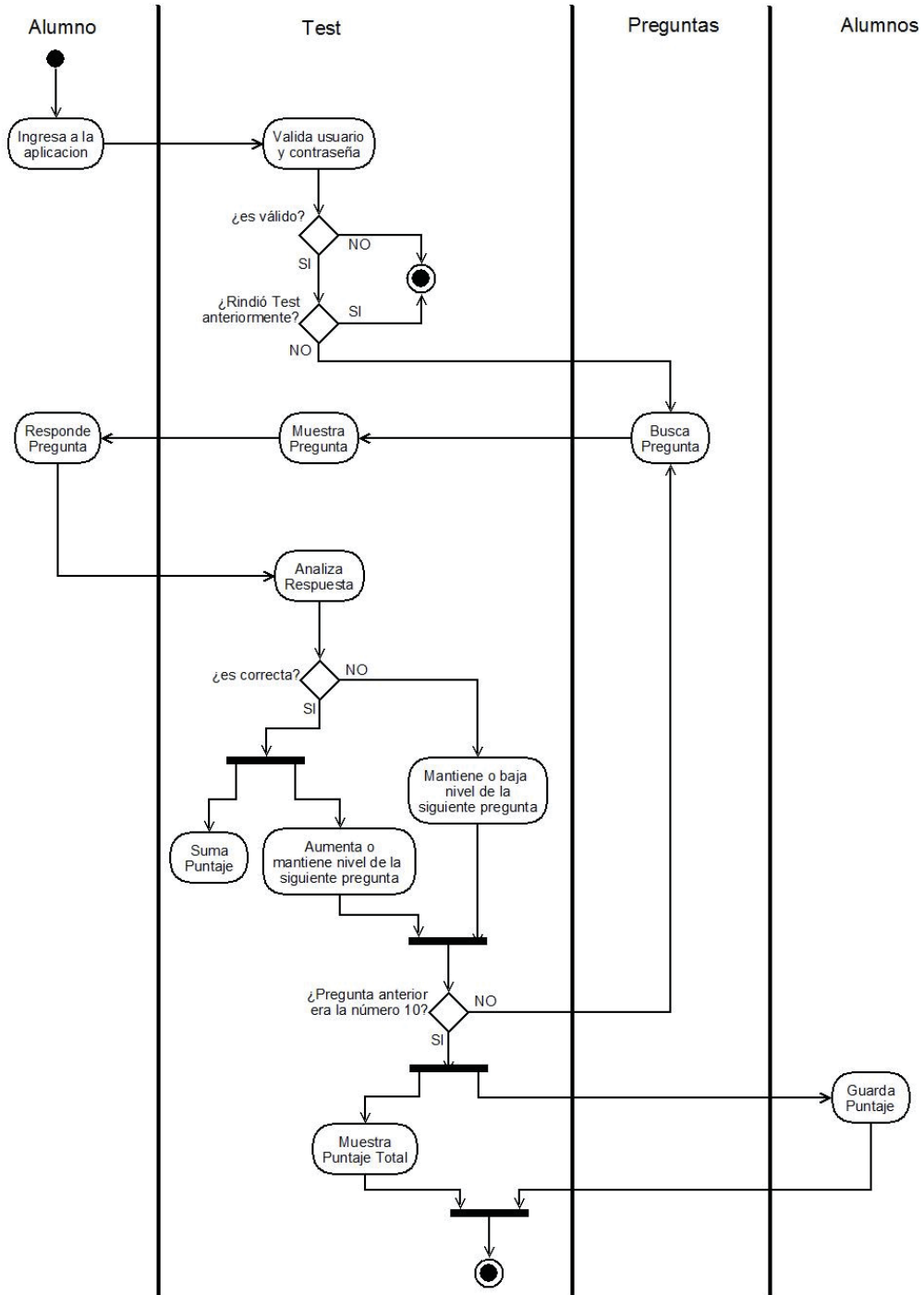


Figura 7.8 Actividades Realizar Test

- b) Este diagrama (Figura 7.9) muestra las actividades realizadas por el profesor, el programa y las clases participantes en la opción del profesor correspondiente a ver la lista de alumnos de un curso y sus notas en caso de haber realizado el test.

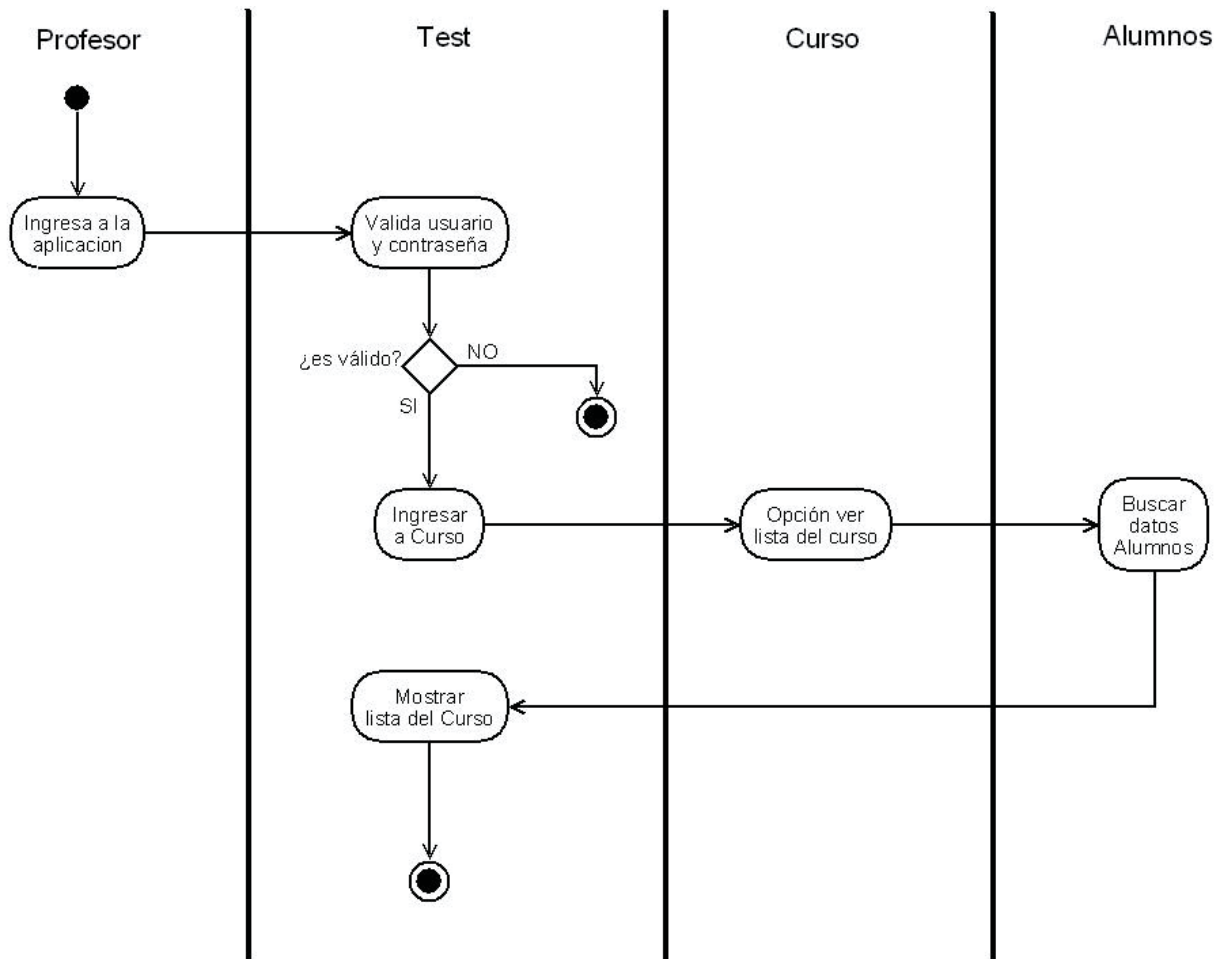


Figura 7.9 Actividades Listar de Alumnos

- c) En el diagrama de actividades de la Figura 7.10, se muestra las tareas realizadas por los distintos participantes dentro de la opción del profesor de “desbloquear” o dar permiso a un alumno para realizar nuevamente el test.
- d) En el diagrama de actividades presentado a continuación en la Figura 7.11, vemos las tareas que realizan las distintas clases que participan en la opción de Eliminar Alumno de la Base de Datos, del perfil del profesor.
- e) En el diagrama de actividades de la Figura 7.12, se muestran las tareas realizadas por las clases participantes en la opción del profesor de determinar el número máximo de test obligatorios que pueden realizar los alumnos.

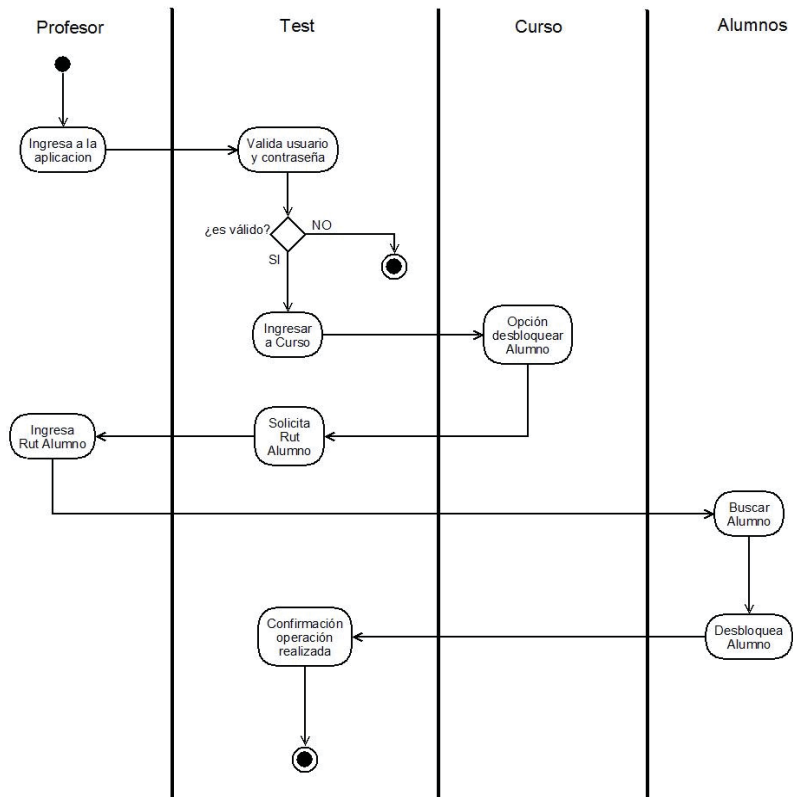


Figura 7.10 Actividades Desbloquear Alumnos

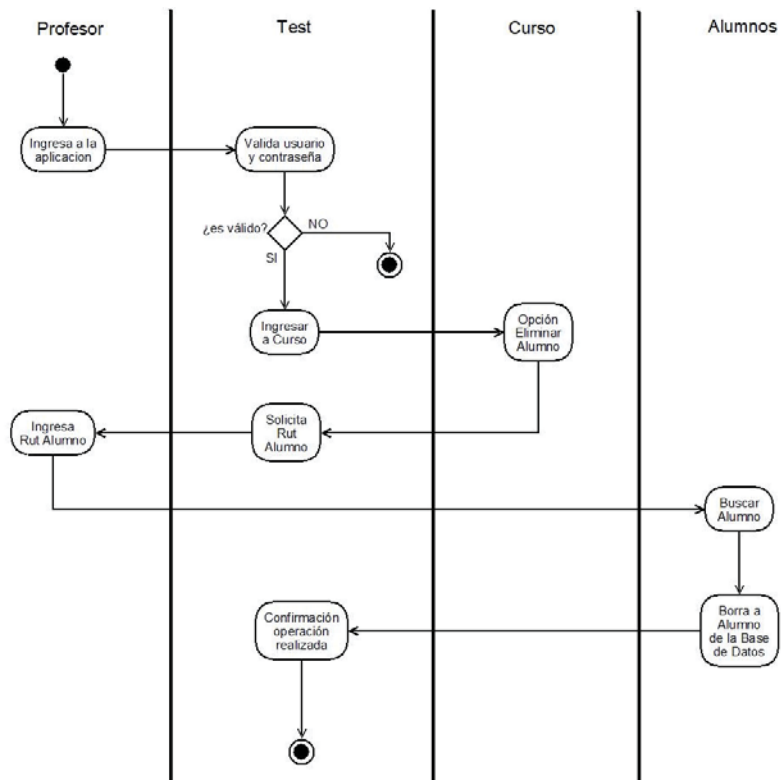


Figura 7.11 Actividades Eliminar Alumnos

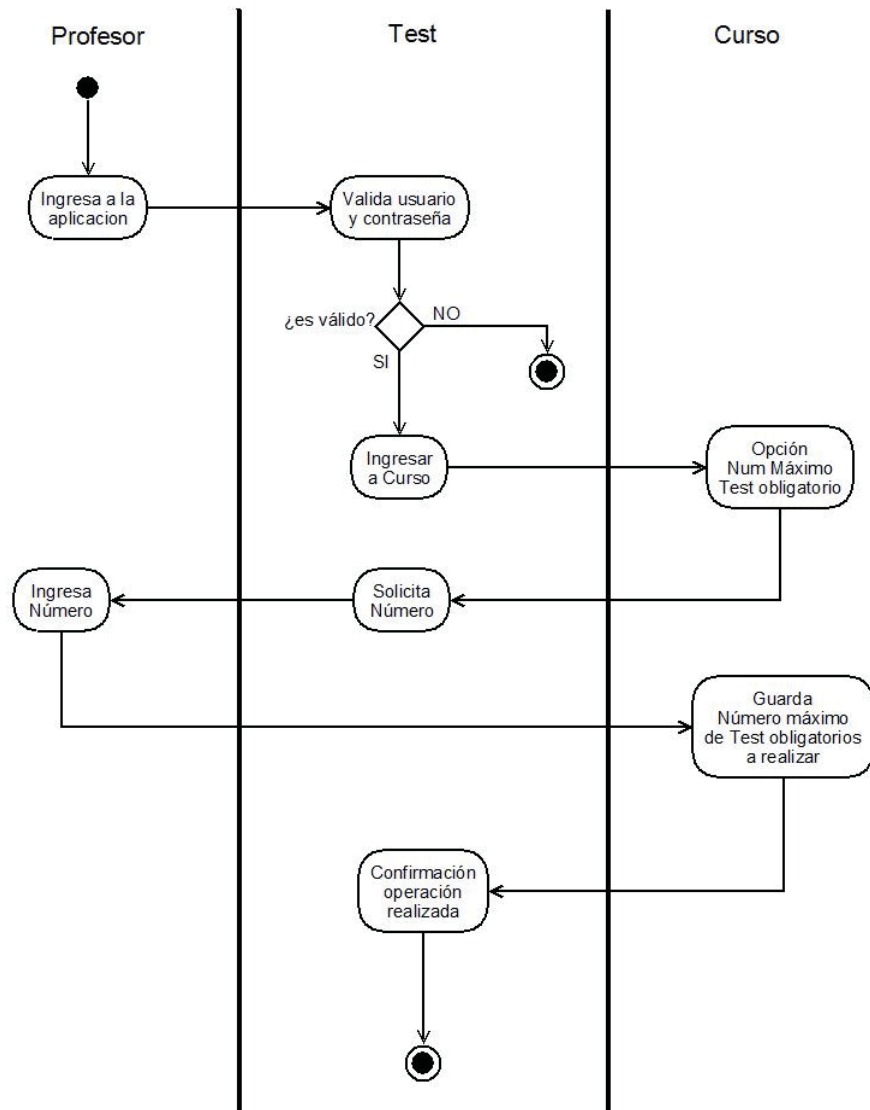


Figura 7.12 Actividades Determinar Número Máximo de Test Obligatorios

[Bátiz, 09] y [Cabrera, 08] presentan diagramas tomados como ejemplo para realizar los diversos diagramas de casos de uso, clases y secuencias, mostrados en este documento.

7.1.4 Diagrama Relacional

El siguiente diagrama relacional, muestra cómo será implementada la Base de Datos del programa y las relaciones incluidas dentro de ella con sus atributos correspondientes (como datos de los usuarios y las preguntas del test, entre otros) que se almacenarán en dicha Base de Datos (Figura 7.13).

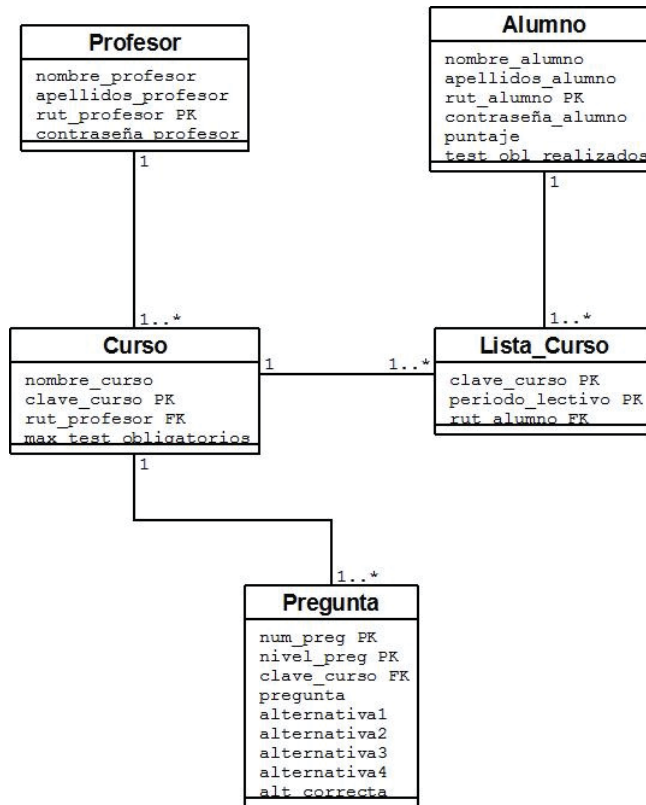


Figura 7.13 Diagrama Relacional

7.1.5 Arquitectura Sistema de Autoevaluación

En el siguiente diagrama presentado en la Figura 7.14, se muestra un esquema de cómo queda físicamente implementado este proyecto, con una base de datos con las preguntas y datos de usuarios, en un servidor de manera centralizada, siendo accedido por dispositivos móviles de forma inalámbrica a través de Internet (o dentro de una red local donde se encuentre la base de datos dentro de un servidor central de dicha red).

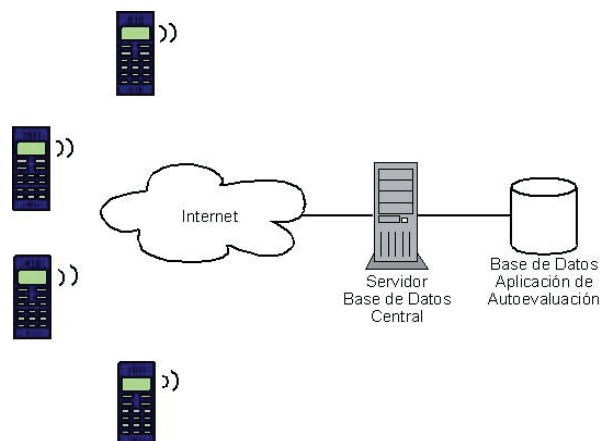


Figura 7.14 Arquitectura del Sistema

7.2 Resumen Desarrollo del Sistema

Al desarrollar este “Sistema de Autoevaluación para Dispositivos Móviles” a través de una Metodología Incremental, se realizan diversas iteraciones, tras las cuales se obtienen prototipos con Incrementos en el programa, los que serán detallados a continuación.

7.2.1 Primer Incremento

Durante la primera iteración en el desarrollo del sistema, se obtiene el primer incremento del programa donde se encuentran bien definidos los perfiles de usuarios del sistema, Profesor y Alumno, así como también la forma del ingreso de los usuarios.

En el perfil Profesor, se tiene el menú principal a distintas funciones que puede realizar, como la de Listar Alumnos, Ingresar, Eliminar y Desbloquear Alumno, donde todas se encuentran en funcionamiento.

Por el lado del perfil Alumno, se encuentra en funcionamiento la realización del test en sí, es decir, la forma de cómo el alumno puede ver y responder las preguntas y con esto la respuesta del sistema para evaluar, seleccionar y mostrar la siguiente pregunta, además de la finalización del test y la presentación de los resultados obtenidos por el alumno, por lo tanto la parte interna y principal del sistema se encuentra, en esta iteración, funcionando en buenas condiciones.

7.2.2 Segundo Incremento

Una vez con el primer prototipo o incremento en buen funcionamiento, se corrigen algunos detalles en el sistema, con respecto a las distintas funciones de ambos tipos de usuarios.

En el usuario Alumno, se realiza un menú con dos opciones de realización del test: una opción de realizarlo de forma obligatoria, en la cual, tras realizar el test, el puntaje obtenido queda registrado en la base de datos y el alumno queda bloqueado para que no pueda ingresar a esta opción nuevamente; y una segunda opción para realizar el test en forma de ensayo, donde el sistema muestra el resultado final, pero no bloquea al alumno, por lo que puede ingresar a realizar el test en esta opción cuantas veces lo desee. Además se agrega dentro de la realización del test, independiente de la opción que el alumno elija para realizar el test, la función que el sistema muestre, cada vez que el alumno responde una pregunta, un aviso sobre la respuesta indicando si es correcta o incorrecta y, en caso de ser correcta, mostrar el puntaje obtenido, o si es incorrecta, mostrar la pregunta realizada y su respuesta debidamente correcta.

Por el lado del Profesor, se modifica la función de Ingresar Alumno, separando el campo donde ingresa el nombre del alumno en dos campos, uno para el nombre y otro para el apellido, además de agregar otro campo para que repita la contraseña que le asigna al alumno. También se agrega otra opción en el menú del profesor, para que pueda modificar el número máximo de test que puede realizar el alumno cuando accede a la opción de ‘test obligatorio’ y no sea bloqueado por el sistema tras el primer test que realiza sino que una vez que realiza el

test, en la opción “obligatoria”, el total de veces indicado por el profesor en esta nueva función.

7.2.3 Tercer Incremento

En este incremento, encontramos una nueva función para el perfil Alumno, donde se agrega una nueva opción (a partir de la posibilidad de que el profesor puede indicar el número máximo de test obligatorios pueda realizar el alumno), en la cual el Alumno tiene a su disposición el historial de test realizados, donde podrá ver los puntajes obtenidos en los últimos test que ha realizado (de la opción ‘test obligatorio’). Por otro lado, dentro del test se agrega una quinta alternativa a las preguntas con la opción de omitir, en este caso en el aviso se indica la pregunta omitida y su respuesta correcta, además de que la siguiente pregunta sea del mismo nivel que la anterior.

7.2.4 Cuarto Incremento

Para mayor seguridad, se agrega la función de registro del profesor, quien al ingresar por primera vez al sistema lo hace como ‘admin’ e inmediatamente se le solicita ingresar sus datos para quedar registrado en el sistema, una vez realizado esto, puede ingresar sin problemas como usuario registrado, con Rut y Contraseña, a su perfil.

7.3 Presentación del Sistema Final

A continuación se presenta gráficamente la resolución final del sistema desarrollado para este trabajo de título, mediante la tecnología J2ME, con imágenes del sistema obtenidas por el emulador de un dispositivo móvil, incluido en la herramienta de desarrollo NetBeans.

En la Figura 7.15, se puede ver la pantalla que da el inicio del programa de evaluación.

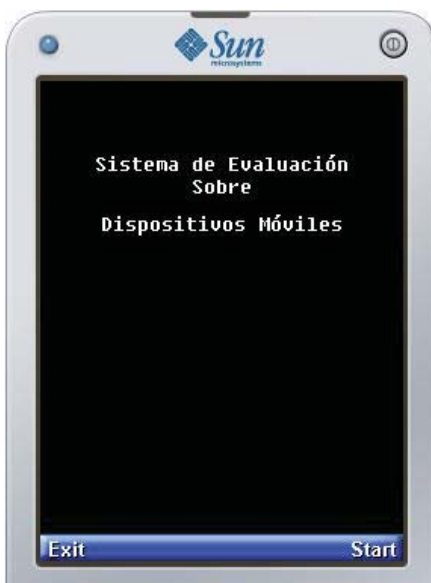


Figura 7.15 Presentación “Sistema de Autoevaluación Sobre Móviles”

Como se muestra en la Figura 7.16, luego de la pantalla de inicio, se tiene la sección de ingreso de usuario.

The screenshot shows a mobile application interface for user login. The title bar is blue with the Sun logo and the text 'Ingreso de Usuario'. The main content area is light blue and contains the instruction 'Por favor ingresa tu Rut y Contraseña'. There are two text input fields: one for 'Rut' and one for 'Contraseña'. The 'Contraseña' field has a 'Qwerty' keyboard icon to its right. At the bottom, there are two buttons: 'Salir' and 'Menu'.

Figura 7.16 Ingreso de Usuario

Si ingresa un nuevo Administrador o Profesor no registrado en el sistema, como 'admin' en lugar de ingresar su Rut, entra a una pantalla de registro de nuevo administrador (Figura 7.17).

The screenshot shows a mobile application interface for registering a new administrator. The title bar is blue with the Sun logo and the text 'Registro Nuevo Administrador'. The main content area is light blue and contains the instruction 'Por favor ingrese sus datos'. There are five text input fields: 'Nombre', 'Apellido', 'Rut', 'Contraseña', and 'Rep. Contraseña'. The 'Apellido' field has a 'Qwerty' keyboard icon to its right. At the bottom, there are two buttons: 'Salir' and 'Menu'.

Figura 7.17 Registro de Nuevo Administrador

Al ingresar con el perfil de Profesor (una vez registrado), se ve el menú con las opciones que puede realizar en este sistema (Figura 7.18).

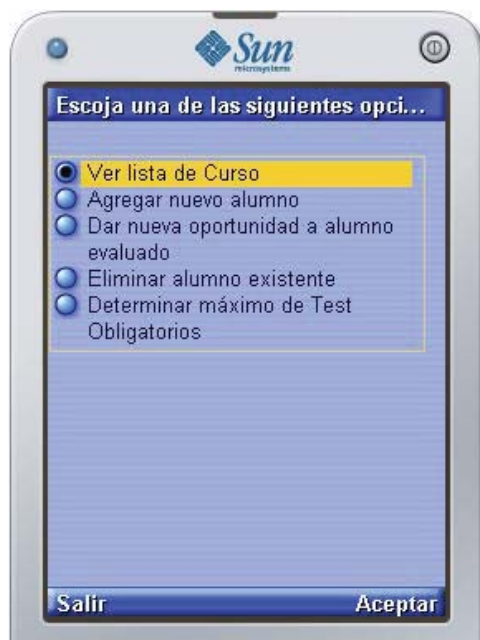


Figura 7.18 Menú Perfil Profesor

En caso de seleccionar la opción de ‘Agregar nuevo Alumno’, pasa a la siguiente pantalla (Figura 7.19), donde debe ingresar, al igual que en la pantalla de registro de nuevo administrador, el nombre, apellido, Rut y la contraseña que le asignará al alumno.

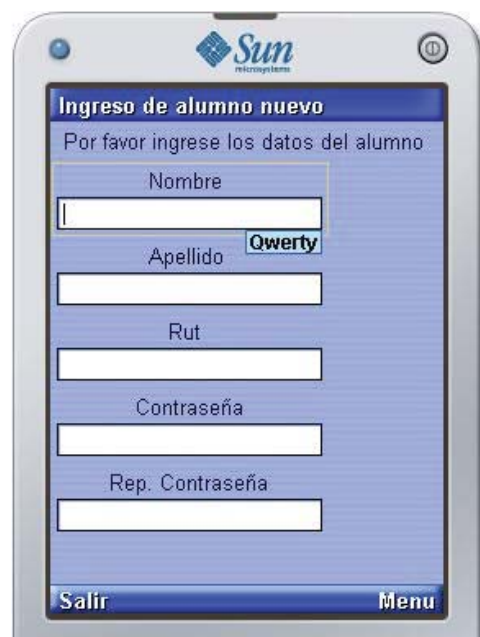


Figura 7.19 Ingresar Alumno Nuevo

Por otro lado, al ingresar al programa como Alumno, se muestran las opciones para este usuario (Figura 7.20), dos opciones para realizar el test (ensayo y oficial) y otra para ver el historial de puntajes obtenidos al realizar el test en su opción oficial.

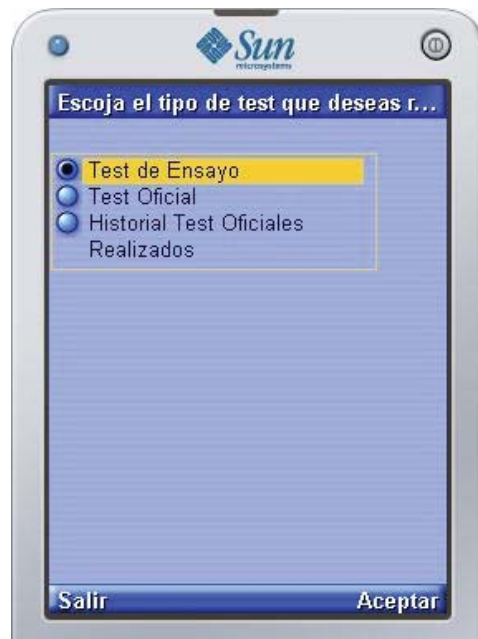


Figura 7.20 Menú Perfil Alumno

Tras seleccionar el tipo de test a realizar, aparece inmediatamente la primera pregunta con sus 4 alternativas, y una quinta con la opción de omitir (Figura 7.21).

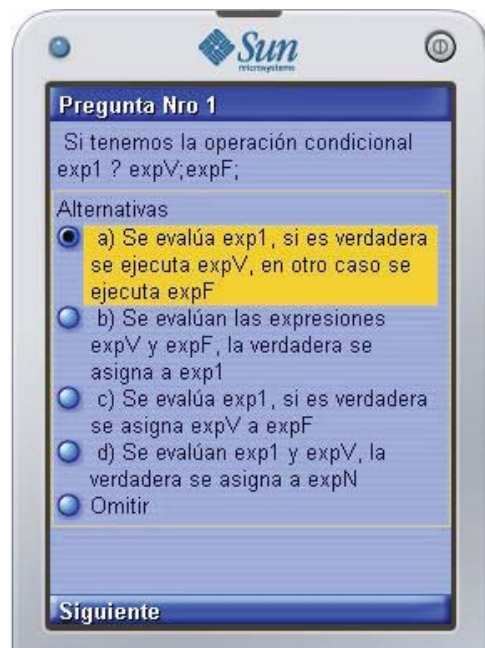


Figura 7.21 Presentación Preguntas del Test

Una vez terminado el test, tras la décima 10 pregunta, el programa muestra el resultado obtenido por el alumno (Figura 7.22).

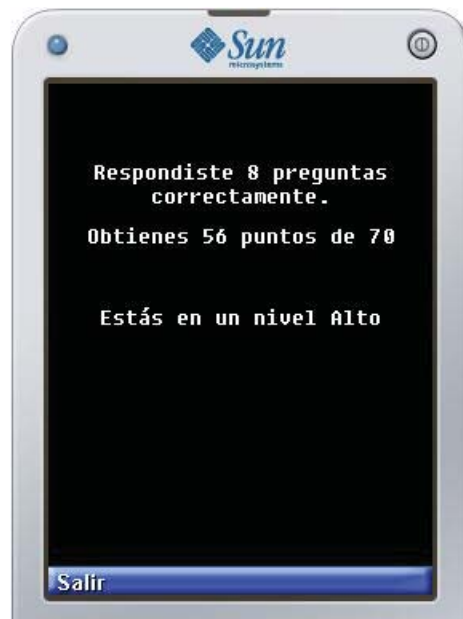


Figura 7.22 Presentación de Puntaje Obtenido

8 Diseño y Aplicación de Pruebas

8.1 Pruebas de Caja Negra

Este tipo de pruebas se concentra sobre la interfaz del software, es decir se comprueban las funciones específicas del sistema y se llaman de “Caja Negra” ya que, a partir de una determinada entrada a cierta función del sistema, se evalúa si es correcta la salida entregada por esa función acorde a lo esperado, sin importar la forma en que genera dicha salida sino que ésta sea de acuerdo al tipo de salida que se requiere, por ejemplo si se tiene una función que pida ingresar un número x y se necesita que entregue dicho número multiplicado por 5, este tipo de prueba sirve para comprobar que efectivamente la salida de esa función sea la correcta sin importar si internamente realizó la operación como sumas consecutivas ($5 + 5 + \dots x$ veces) o si fue realizada efectivamente como una multiplicación ($x * 5$).

A continuación se detalla el diseño y aplicación de estas pruebas realizadas en las distintas iteraciones del sistema.

8.1.1 Diseño de Pruebas de Caja Negra

A continuación se detallan los casos de prueba realizadas en las distintas iteraciones del sistema.

8.1.1.1 Casos de Prueba Primera Iteración

Para la primera iteración de desarrollo de este programa se perciben los siguientes casos de prueba:

- Ingreso de usuarios al sistema: por el lado del ‘Profesor’, debe ingresar con nombre usuario ‘admin’ y se espera como resultado que el sistema muestre el menú de opciones que puede realizar el profesor, y por el lado del ‘Alumno’, debe ingresar con su Rut y Contraseña y se espera como resultado el ingreso del alumno a realizar el test, específicamente que se muestre la primera pregunta. En caso de no que el Rut no esté registrado o si la contraseña es incorrecta, el sistema como respuesta debe mostrar un aviso según el caso que corresponda.
- Usuario ‘Profesor’: en esta etapa del desarrollo se tienen 4 opciones. Para ingresar correctamente a cada una de ellas tan solo se debe seleccionar una y presionar el botón “Aceptar”.
- Opción “Ver lista de curso”: Como resultado de seleccionar e ingresar a esta opción, se espera que el sistema muestre la lista de alumnos con su nota correspondiente, en caso de que haya realizado el test. Al presionar “Salir”, se espera volver al menú del profesor.
- Opción “Agregar nuevo alumno”: Al elegir esta opción, el sistema debe mostrar al “Profesor” un formulario donde se pida ingresar Nombre y Apellido (en un solo

campo), Rut del alumno a ingresar, además de la contraseña que se le asignará. Si dicho alumno no se encuentra registrado, se espera como resultado que el programa de un aviso de registro exitoso y en caso contrario que indique que el usuario ya está registrado con anterioridad en la base de datos. En caso de ingresar contraseñas distintas, el aviso debe indicar que las contraseñas son incorrectas y no se puede ingresar el nuevo usuario.

- Opción “Dar nueva oportunidad a alumno evaluado”: Si el usuario elige esta opción, el sistema debe mostrar un formulario donde se pida el ingreso del Rut de un alumno, en caso de que el alumno ya ha realizado el test, el sistema lo “desbloquea” para que pueda realizarlo nuevamente, en caso de que el Rut no esté registrado o si es de un alumno que no ha realizado el test, y por lo tanto no está bloqueado, el sistema debe indicarlo en un aviso correspondiente al caso.
- Opción “Eliminar alumno existente”: Al seleccionar esta opción, el usuario debe en pantalla un formulario, igual que en el caso anterior, donde se pida el Rut de un alumno y como resultado el sistema lo quita de la base de datos y da un aviso de eliminación exitosa o de alumno no encontrado en caso de que el Rut ingresado no haya sido registrado anteriormente.
- Realizar test de evaluación: Si el usuario que ingresa es un alumno, el sistema le muestra la primera pregunta del test con sus 4 alternativas.
- Responder pregunta: Cuando el usuario selecciona una alternativa como respuesta a la pregunta que se le realiza, el sistema debe mostrar la pantalla correspondiente dependiendo del número de la pregunta recientemente contestada.
- Pregunta 1 a 9: En caso de que la pregunta respondida por el usuario está dentro de este rango, el sistema debe mostrar la siguiente pregunta.
- Pregunta número 10: Si la pregunta respondida por el usuario es la décima, por ser la última pregunta, el sistema debe mostrar el puntaje total obtenido por el usuario.

8.1.1.2 Casos de Prueba Segunda Iteración

Luego de algunas modificaciones con respecto a la primera iteración, se encuentran nuevos casos de prueba y algunos de los anteriores que también se modifican:

- Ingreso de usuarios al sistema: por el lado del ‘Profesor’, se mantiene igual como se menciona anteriormente, al ingresar con nombre usuario ‘admin’ se espera como resultado que el sistema muestre el menú de opciones que puede realizar el profesor, y por el lado del ‘Alumno’, debe ingresar con su Rut y Contraseña y se espera como resultado el ingreso del alumno a una pantalla con las opciones para dicho usuario.
- Usuario ‘Profesor’: en esta etapa el sistema ya debe mostrar 5 opciones para este usuario.

- Opción “Agregar nuevo alumno”: Al elegir esta opción, el sistema debe mostrar al “Profesor” un formulario donde se pida ingresar Nombre y Apellido (en campos separados), Rut del alumno a ingresar, además de 2 veces la contraseña que se le asignará. En caso de ingresar contraseñas distintas, el aviso debe indicar que las contraseñas son incorrectas y no se puede ingresar el nuevo usuario, lo mismo si ya está registrado de antes.
- Modificar máximo de test “obligatorios”: Esta es la quinta opción del profesor. Al seleccionarla, se muestra un campo donde se debe ingresar un número para establecer el número máximo de test “obligatorio” puede realizar cada alumno antes que el sistema lo bloquee, como resultado el programa debe entregar un aviso que indique que el número máximo de test se ha modificado.
- Usuario ‘Alumno’: en esta etapa se tienen 2 opciones para realizar el test, “obligatorio” y “de ensayo” de los cuales debe seleccionar el que desea realizar, en ambos casos la respuesta del sistema debe ser que muestre la primera pregunta del test.
- Aviso de respuesta: Luego de seleccionar la respuesta a la pregunta, el sistema debe mostrar, en caso de que la respuesta sea correcta, el puntaje que el alumno lleva acumulado y si la respuesta es incorrecta, debe mostrar la pregunta realizada y su respuesta debidamente correcta.
- Pregunta 1 a 9: En caso de que la pregunta respondida por el usuario está dentro de este rango, tras ver el aviso de respuesta correcta/incorrecta, el sistema debe mostrar la siguiente pregunta.
- Pregunta número 10: Si la pregunta respondida por el usuario es la décima, tras el aviso de respuesta correcta/incorrecta, por ser la última pregunta, el sistema debe mostrar el puntaje total obtenido por el usuario.

8.1.1.3 Casos de Prueba Tercera Iteración

Tras una pequeña modificación en el sistema, se observa algunos cambios en un determinado caso de prueba, además de uno nuevo, y se mencionan a continuación:

- Usuario ‘Alumno’: en esta etapa se tienen 3 opciones para este usuario, 2 de ellas para realizar el test, “obligatorio” y “de ensayo”, tras seleccionar una de ellas, la respuesta del sistema debe ser que muestre la primera pregunta del test, y la tercera opción que se debe mostrar al ingresar como usuario ‘Alumno’, es de “historial” de test obligatorios realizados.
- Realizar test de evaluación: Si el usuario que ingresa a cualquiera de las dos opciones que tiene para realizar el test, el sistema le muestra la primera pregunta del test, ahora con sus 4 alternativas y una nueva opción de ‘Omitir’.

- Aviso de respuesta: Luego de seleccionar la respuesta a la pregunta, el sistema debe mostrar, en caso de que la respuesta sea correcta, el puntaje que el alumno lleva acumulado y si la respuesta es incorrecta o si se selecciona la opción de ‘Omitir’, debe mostrar la pregunta realizada y su respuesta debidamente correcta.
- Opción “Historial”: Como resultado de ingresar a esta opción, se espera que el sistema muestre la lista de resultados obtenidos por el alumno en el (los) último(s) test obligatorios realizado(s).

8.1.1.4 Casos de Prueba Cuarta Iteración

En esta última iteración se encuentra un nuevo caso de prueba a partir de las últimas modificaciones realizadas:

- Ingreso de usuario ‘Profesor’, al ingresar con nombre usuario ‘admin’ se espera como resultado que el sistema muestre una pantalla de registro de usuario y en caso de ingresar con su Rut y contraseña, se debe mostrar el menú con sus opciones de profesor.
- Registro de usuario ‘Profesor’: Tras ingresar como ‘admin’, se muestra un formulario idéntico al que se muestra cuando el usuario ‘profesor’ ingresa a la opción de “Agregar nuevo alumno”, por lo que tras ingresar los datos correspondientes, el sistema debe dar aviso de registro (exitoso, si se realiza bien el ingreso de los datos, incorrecto si ya esté registrado de antes o contraseñas incorrectas si se ingresan contraseñas distintas). Como resultado luego de dicho aviso, el sistema debe regresar a la pantalla de ingreso de usuario.

Los demás casos de prueba faltantes en alguna iteración con respecto a la anterior es porque no sufrieron modificaciones por lo que se mantiene(n) tal como se menciona(n) en la(s) iteración(es) anterior(es).

8.1.2 Aplicación de Pruebas de Caja Negra

En esta sección se detalla sobre la aplicación de las pruebas de caja negra, específicamente pruebas de equivalencia, que consisten en repartir en clases de equivalencia las diversas entradas al programa, por ejemplo, en el sistema de este trabajo de título, una clase de equivalencia identificada es la ingresar el Rut de un usuario, otra la de ingresar contraseña o password (en ambas clases de equivalencia, sea en el ingreso de usuarios al sistema y en registros de usuarios a la base de datos) y otra clase de equivalencia la de responder una pregunta en el test (seleccionar una alternativa). Al realizar estas pruebas, se utiliza una entrada válida, con la cual el sistema debe responder lo que se espera de él, y una inválida, para ejercitar la capacidad del sistema sobre el manejo de excepciones.

8.1.2.1 Aplicación en Casos de Prueba Primera Iteración

Para los distintos casos de prueba percibidos para la primera iteración, las pruebas de caja negra realizadas se detallan a continuación:

- En el caso de prueba sobre ingreso de usuario al sistema (login), las pruebas aplicadas consiste en ingresar como entrada típica, un nombre de usuario válido con su contraseña correcta para ejercitar el caso común en el cual el usuario ingresa sin problemas, asegurándose de que para cada usuario ingrese a su perfil correspondiente, y por otro lado, como entrada inválida, dos opciones: una contraseña inválida para el nombre de usuario válido y un usuario no registrado en el sistema o inválido (con cualquier contraseña, ya que al no estar registrado de nombre de usuario la respuesta entregada debe ser esa de que no existe en la base de datos), para probar que se muestren los avisos correspondientes a cada sentencia de manejo de excepciones.
- Al ingresar al perfil Profesor, comprobar que efectivamente se muestran las 4 opciones que tiene en esta iteración y al seleccionar una que realmente se ingrese a la que se está seleccionando.
- Al seleccionar la opción de lista de curso, se debe probar que realmente se listen los alumnos que estén registrados en la base de datos, en caso de que no hayan, que se muestre la lista vacía y luego al salir de esta opción, que regrese al menú principal del perfil.
- En la opción de agregar nuevo alumno, las pruebas se enfocan principalmente en el ingreso del nombre completo del alumno a registrar en un solo campo y el ingreso del Rut del alumno, probando con uno válido, es decir, que no esté en la base de datos para que el sistema lo registre, ya que los usuarios se validan por el Rut, y uno que invalide la acción de agregar un alumno nuevo como es el caso de ingresar un Rut ya registrado anteriormente. Por otro lado, como en esta opción se solicita ingresar una contraseña, como entrada válida se ingresa una cadena de caracteres, y como inválida se deja el campo en blanco.
- Para el caso de prueba de la opción de dar nueva oportunidad a un alumno, como se solicita ingresar el Rut del alumno favorecido, una entrada válida para esta función es ingresar el Rut de un alumno que esté registrado en el sistema y que además haya realizado el test y con eso esté bloqueado por la aplicación. Como entrada inválida, se ingresa el Rut de un alumno que no esté bloqueado y también un Rut que no se encuentre registrado en la base de datos.
- Al igual que en la aplicación de pruebas del caso anterior, en la opción de eliminar un alumno de la base de datos también se pide ingresar el Rut del alumno afectado, por lo que en esta oportunidad, como entrada válida se ingresa el Rut propiamente tal del alumno que se desea quitar del sistema y como entrara inválida se ingresa un Rut que no se encuentre registrado con anterioridad.

- Tras el ingreso de un usuario alumno, se debe confirmar que se ingrese a su perfil correspondiente, esto es que el sistema le muestre la primera pregunta del test con sus 4 alternativas.
- En el caso de responder una pregunta, se tiene como entrada la selección de una de las 4 alternativas, se toma como entrada válida la selección de la alternativa correcta y como inválida la selección de una alternativa incorrecta, ya que dependiendo del caso es la respuesta entregada por el sistema, con lo cual además se debe comprobar que el paso siguiente sea el correcto a partir del número de la pregunta anteriormente contestada.
- Si dicha pregunta respondida en el caso anterior está entre la número 1 y 9, se tiene que verificar que la siguiente pantalla se muestre la próxima pregunta, cuyo nivel sea mayor o menor que la anterior dependiendo de la alternativa seleccionada.
- En el caso de que haya sido la décima pregunta, se comprueba de que el sistema muestre efectivamente el final del test con el puntaje total obtenido por el alumno.

8.1.2.2 Aplicación en Casos de Prueba Segunda Iteración

A continuación se detallan las pruebas de caja negra realizadas para los casos de prueba de la segunda iteración:

- En el caso de prueba sobre ingreso de usuario al sistema (login), se realiza exactamente las mismas pruebas realizadas en la primera iteración, utilizando las mismas entradas válidas e inválidas para comprobar si ingresa al perfil deseado o si se muestran los avisos correspondientes a cada entrada inválida.
- En esta iteración se debe comprobar que al ingresar al perfil del profesor, se muestren 5 opciones en el menú de este usuario ya que se agregó una nueva opción con respecto a la primera iteración.
- En la opción de agregar nuevo alumno, a partir de los cambios sufridos es dicha opción, las pruebas realizadas consisten en ingresar el nombre del alumno por separado (nombre en un campo y apellido en otro), además de las pruebas de ingreso del Rut, con entrada válida uno que no esté registrado anteriormente en la base de datos para que el sistema lo registre, y un Rut ya registrado anteriormente que invalide la acción de agregar un alumno nuevo y comprobar que se entregue el aviso correspondiente. Por otro lado, como en esta iteración se agrega ingresar la contraseña que se asigna al alumno dos veces para el correcto registro del usuario, como entrada válida se ingresa dos veces la misma contraseña, y como inválida se ingresan contraseñas distintas.
- Al seleccionar por parte del profesor la opción de modificar el número máximo de test “obligatorio” que pueda realizar el alumno, se ingresa como entrada válida un número mayor a 0 (cero) y como inválida uno negativo o simplemente dejar el campo en blanco.

- Cuando el usuario ingresa como alumno, se debe comprobar en esta iteración que el sistema muestre el menú para este usuario de dos opciones para realizar el test y luego de seleccionar una de dichas opciones, se muestre la primera pregunta que deba responder el alumno.
- En esta iteración, al responder una pregunta se prueba como entrada válida la selección de la alternativa correcta y como inválida la selección de cualquier otra alternativa y comprobar que el sistema entregue el aviso correspondiente a cada caso.
- Se debe comprobar que, en el caso de que la pregunta contestada sea desde la número 1 hasta la 9, luego del aviso de respuesta correcta/incorrecta, el sistema muestre la pregunta siguiente del test.
- En caso de que la pregunta respondida sea la número 10, se necesita comprobar que luego del aviso de respuesta correcta/incorrecta el sistema muestre el fin del test, con el puntaje obtenido por el alumno.

8.1.2.3 Aplicación en Casos de Prueba Tercera Iteración

Tras sufrir pequeñas modificaciones el sistema y los casos de prueba en comparación a la iteración anterior, las pruebas realizadas para dichos cambios son:

- Al ingresar como alumno, en esta etapa se debe comprobar que en el menú para este usuario se muestren tres opciones, las mismas dos para realizar el test y la tercera para ver el historial de test tipo obligatorio realizados.
- Cuando se ingresa a una de las opciones para realizar el test, se debe comprobar que el sistema muestre la primera pregunta, pero en este caso, además de las 4 alternativas habituales, se pueda ver una quinta alternativa con la opción de “omitir”.
- Una vez que el alumno responda una pregunta, se debe comprobar que el aviso entregado por el sistema sea el correspondiente indicando si la respuesta es correcta o incorrecta, agregando un nuevo aviso que indique que la pregunta fue omitida en caso de que se seleccione la quinta opción.
- Si se ingresa a la opción “Historial”, se debe comprobar que el sistema muestre la fecha, hora y puntaje obtenido en el (los) último(s) test obligatorio(s) realizado(s) por el alumno (en caso de no haber realizado alguna vez el test tipo obligatorio se muestra la pantalla en blanco).

Al aplicar las pruebas correspondientes en la opción de listar curso del usuario ‘Profesor’, en esta iteración, se descubre que al seleccionar dicha opción, la primera vez se muestra el resultado esperado, pero al seleccionarla nuevamente por segunda vez, no se obtiene el mismo resultado, por lo que se decide realizar pruebas de Caja Blanca por ser más minuciosas, las que se detallan más adelante, en la sección 7.2 de este capítulo.

8.1.2.4 Aplicación en Casos de Prueba Cuarta Iteración

En esta última iteración, se realizan nuevas aplicaciones de prueba, para los casos de prueba nuevos en esta iteración y también para los demás que no sufrieron modificaciones de la(s) iteración(es) anterior(es):

- Al ingresar un usuario “profesor” al sistema, se debe comprobar que cuando ingresa con nombre de usuario ‘admin’, se muestre un formulario de registro y si ingresa a través de su Rut y contraseña (como usuario ya registrado en el sistema), se muestre el menú que le corresponde.
- Una vez ingresado con nombre de usuario ‘admin’, el usuario se encuentra con un formulario de registro, en el cuál se realizan las mismas pruebas realizadas en la opción “Agregar nuevo alumno”, ya que los campos a llenar son los mismos, por esto se debe ingresar el nombre y apellido del usuario en los campos correspondientes, y se realizan las pruebas de Rut adecuados, siendo entrada válida el registro del usuario cuyo Rut no se encuentre con anterioridad en la base de datos, y como entrada inválida el Rut de un usuario que ya haya sido registrado en el sistema. También se tiene el ingreso de la contraseña con un campo adicional para repetir dicha contraseña, probando con la misma cadena de caracteres como entrada válida y distintas cadenas de caracteres (o dejando uno o ambos campos en blanco) como entrada inválida.

Como se menciona anteriormente, las pruebas no detalladas en algunas iteraciones con respecto a la(s) anterior(es), es porque no fueron modificadas al pasar a la siguiente iteración, por lo que se aplican de la misma manera mencionada y detallada en la iteración donde se realizan por primera vez.

8.2 Pruebas de Caja Blanca

Las Pruebas de Caja Blanca se basan en realizar un examen minucioso de los detalles procedimentales, es decir se comprueban los caminos lógicos del software.

Al realizar estas pruebas principalmente se comprueba seguir un correcto procedimiento a través de las clases para llevar a cabo la ejecución adecuada de cada una de las opciones que debe realizar la aplicación.

8.2.1 Diseño de Pruebas de Caja Blanca

Al encontrarse un error al aplicar pruebas de Caja Negra en la tercera iteración del desarrollo de este “Sistema de Autoevaluación”, en la opción “Listar Alumnos” del perfil Profesor, se realizan pruebas de rutas o trayectoria, que constituyen una técnica de Pruebas de Caja Blanca que identifica fallas en la implementación del (los) componente(s) y que consisten en ejercitar, al menos una vez, todas las rutas posibles del código, logrando con esto ejecutar todas las instrucciones (sean de control, lógicas o condicionantes) involucradas en realizar esta función de listar alumnos, encargadas desde la búsqueda de los alumnos en la

base de datos hasta mostrar la lista en pantalla, con el fin de detectar y posteriormente corregir el (los) defecto(s) en el código causante(s) de dicho error.

Para esto, se revisa cada detalle, línea a línea del código, viendo las distintas alternativas que pueden tomar todas las variables involucradas en esta tarea, sea contadores o variables de datos como nombre o Rut (entre otras), el tipo de datos q almacena cada uno de ellos y, por ejemplo, si los contadores se inicializan, limpian (o vuelven a su valor inicial), aumentan o disminuyen su valor de forma oportuna de acuerdo a lo que se necesite en los distintos lugares del código y también si realmente están ubicados correctamente dentro del mismo, además de revisar el paso de los parámetros de un módulo a otro y ver todo lo que ocurra con los datos que se transfieren de un lado a otro.

8.2.2 Aplicación de Pruebas de Caja Blanca

Para aplicar estas pruebas de caja blanca, específicamente las de rutas o trayectoria anteriormente mencionada, éstas se concentran en todas las instrucciones participantes en la tarea del programa de listar curso, ya que ahí es dónde se desea encontrar el (los) defecto(s) que provocan el error detectado al ejecutar dicha tarea durante las pruebas de caja negra, por lo que se decide apoyarse en estas últimas pruebas para una mejor observación de ciertas variables críticas como por ejemplo un contador, para lo cual se adiciona temporalmente la instrucción de imprimir el valor de dicha variable en pantalla y así llevar un mejor control de ella y monitorear su comportamiento. En un principio se revisa el código línea a línea, instrucción por instrucción, tantas veces sea necesario dependiendo de la cantidad de caminos lógicos que puedan tomar ciertos datos o variables, detectando como crítico un contador encargado de incrementar su valor a medida que se vaya rescatando un alumno desde la base de datos e imprimiendo en pantalla, por lo que provisoriamente se da la instrucción, como ya se ha mencionado antes, de imprimir el valor de este contador por cada vez que se rescata y muestra el nombre de un alumno; con esto, acertadamente se logra descubrir que a la segunda vez que se quiere mostrar la lista del curso, ese contador no vuelve a su valor inicial sino que continua incrementándose incorrectamente, por ejemplo si hay 5 alumnos en la base de datos, la primera vez que se ejecuta la opción de listar alumnos el contador, iniciado en 0 (cero), efectivamente termina en valor 5, pero a la segunda vez que se quiere ejecutar dicha opción, no regresa a su valor inicial 0 (cero) antes de comenzar a rescatar los alumnos de la base de datos e imprimirlos en pantalla, por lo que continúa incrementando su valor (de 5 que tenía a 6, luego a 7 y así sucesivamente) hasta tomar el valor 10 en el caso del ejemplo mencionado, con esto la aplicación pasaba al estado de error indicando que no se encontraban alumnos en la base de datos, ya que, siguiendo con el ejemplo, se encuentran registrados 5 alumnos que, a la primera ejecución se muestran correctamente pero a la segunda ejecución se intenta mostrar los alumnos número 6 hasta el número 10 los que no se logran encontrar al buscarlos en la base de datos por ser solamente 5 los registrados en ese momento. Con esto se vuelve a revisar con más atención los lugares del código donde está el contador causante del problema, se ve nuevamente su comportamiento y se detecta que es necesario inicializar en 0 (cero) inmediatamente antes al comienzo del ciclo en donde se realiza la tarea de rescatar cada alumno registrado en la base de datos.

8.3 Resumen de Resultados Obtenidos

Al aplicar las pruebas definidas para la Primera Iteración, cada caso de prueba, los resultados obtenidos son los esperados tanto con las entradas válidas como con las inválidas, por ejemplo, al ingresar al sistema con un usuario registrado en la base de datos del programa, es dirigido a su perfil correspondiente y cuando se ingresa con un usuario no registrado o con clave equivocada, el sistema entrega el aviso correspondiente esperado, por otro lado, al realizar el test, se obtiene efectivamente que el test realiza 10 preguntas estimadas para este sistema, con el cambio de nivel entre una y otra tras evaluar una respuesta.

En el caso de las pruebas realizadas en la Segunda Iteración, se obtienen nuevamente los resultados exitosos al aplicar las pruebas sobre funciones ya probadas en la iteración anterior y en el caso de funciones nuevas o en aquellas que sufrieron cambios, las pruebas correspondientes también lograron ser exitosas, sea en el ingreso de entradas válidas, con las cuáles se espera que el programa continúe su curso normal, como en el caso de ingreso al sistema donde se espera que el usuario entre a su perfil correspondiente, y en las inválidas, con las que se espera que el sistema entregue avisos tras ingresar un mal Rut al eliminar un alumno o tras registrar un alumno a la base de datos en caso de que no se concrete dicha operación.

Para la Tercera Iteración, las pruebas aplicadas ya sea en los casos de prueba idénticos a los encontrados en iteraciones anteriores y los que se debieron modificar o agregar en esta iteración, se obtiene que los resultados son positivos tanto con las entradas válidas como inválidas, con las primeras se obtiene los resultados acorde con lo esperado, sea ingreso al perfil de usuario correcto, registro de usuario en la base de datos, evaluación de respuestas al realizar el test y entrega de sus avisos de respuesta correcta/incorrecta u omitida adecuados al caso, etc., a excepción de la opción del profesor de ‘Listar Curso’, donde se descubre que al ingresar por primera vez a dicha opción el resultado es el esperado pero al ingresar por segunda vez, el sistema no realiza lo esperado y entrega un error, para lo cual se aplican pruebas más rigurosas con las cuáles se obtiene éxito al encontrar un pequeño defecto no detectado con anterioridad y luego, una vez que se corrige, al realizar nuevamente las pruebas con las cuales se había obtenido el resultado no esperado, finalmente se logra conseguir el resultado exitoso.

Por último, tras aplicar las pruebas para los casos intactos de las iteraciones anteriores se obtienen resultados positivos, al igual que en la aplicación de pruebas en los casos que sufrieron alguna modificación, ya que al ingresar al sistema como un usuario ‘Profesor’, el programa hace la diferencia al ingresar como ‘admin’, en donde pasa al registro de usuario’, o con Rut y contraseña, con lo que el usuario ingresa a su menú correspondiente, y en casi de probar con un usuario no registrado o con una contraseña incorrecta, el sistema efectivamente entrega el aviso correspondiente a cada caso.

9 Conclusiones

Si bien para solucionar parte del problema ya se ha desarrollado un algoritmo de evaluación “inteligente” fuera del país, al no haber mayores antecedentes sobre él, o que su información sea reservada, este sistema de autoevaluación inteligente sobre dispositivos móviles es una buena alternativa dentro de nuestras fronteras, para aprovechar la tecnología, en este caso para el área de la educación, sobre todo si se trata de los dispositivos móviles, que prestan gran ayuda en distintas áreas y en la abarcada por este proyecto sin duda no será la excepción. Además, como vimos se pueden utilizar distintas alternativas para solucionar el problema, por ejemplo de lenguajes de programación, que aunque difieran en algunos detalles, sirven para llegar al mismo objetivo, en este caso, el desarrollo de un sistema móvil para el área educacional, para lo cual finalmente se utilizó para su desarrollo la tecnología J2ME, cuyo lenguaje de programación que se utiliza es Java, ya que las herramientas de esta tecnología, tanto programas de desarrollo en lenguaje Java como paquetes de archivos, para que el programa pueda destinarse a dispositivos móviles y emular su funcionamiento al trabajar en PC, se encuentran a libre disposición de los desarrolladores, no así las herramientas de la tecnología .NET que tiene sus costos de licencias, por otro lado también dichas herramientas son de mucha utilidad para desarrollar por tener ayudas en pantalla a medida que se avanza en el desarrollo y la posibilidad poder emular el funcionamiento del programa en el PC con la ayuda del emulador, es un gran alivio para ahorro de tiempo en caso de necesitar realizar pruebas ya que no es necesario pasar el programa cada vez que se quiera realizar una prueba y mucho más cuando éstas son para revisión de pequeños detalles.

Además, cabe indicar que este “Sistema de Autoevaluación sobre Dispositivos Móviles, a partir del primer semestre del año 2010, se encuentra a disposición, de forma opcional, de los alumnos del curso “Laboratorio de Programación” de la carrera de Ingeniería de Ejecución Informática de la Pontificia Universidad Católica de Valparaíso, para que prueben su funcionamiento en sus respectivos teléfonos celulares que sean aptos para la ejecución de programas Java (por ejemplo la mayoría de los teléfonos celulares tienen hoy en día esa capacidad ya que los juegos que incluyen son desarrollados en Java) y que a partir del segundo semestre del mismo año estará a disposición como parte del programa de dicho curso, como una herramienta más de ayuda a los alumnos con respecto a los contenidos ahí vistos.

Por último, se puede mencionar para algún trabajo futuro el mejoramiento algorítmico de este sistema, por ejemplo en la parte de analizar las respuestas, selección de la siguiente pregunta, método de puntuación de cada una de las preguntas y cálculo de puntaje final, etc. También puede señalarse realizar mejoras en el acceso, almacenamiento y manejo de datos, además de la comunicación de los dispositivos con la base de datos, especialmente si la aplicación se utiliza en teléfonos celulares, para que el envío y recepción de los datos no tenga costo alguno para el usuario. Por otro lado, se puede realizar este mismo sistema utilizando la tecnología de .Net (presentada en la sección donde se muestran las tecnologías para solucionar el problema de este trabajo de título), aprovechando como base el pequeño prototipo realizado en dicha tecnología y mostrado en este documento.

10 Referencias

[Booch *et al.*, 99] Booch, Grady, Rumbaugh, Jim, Jacobson, Ivan. “*El Lenguaje Unificado de Modelo*”. Editorial Addison Wesley. 1999.

[Bátiz, 09] Bátiz Paredes, Juan de Dios. “*Desarrollo Orientado a Objetos con UML*”, Programación C.E.C.yT. – Instituto Politécnico Nacional, IPN, México. <http://www.scribd.com/doc/2458870/Desarrollo-Orientado-a-Objetos-con-UML-librobookespanolspanish>. Última revisión Noviembre 2009.

[Cabrera, 08] Cabrera, Guillermo. Documento Modelo del Proceso de Software, Curso Ingeniería de Software. Profesor Escuela de Ingeniería Informática, PUCV, 2º semestre, 2008.

[Fernández, 05] Fernández Alcalde, José Fernando. “*Avaluator 3.0*” (Manual de Usuario). Desarrollador, Profesor de Informática y Tecnología de la “Escuela Cultural”, Badolà-Barcelona, España. 2005.

[Guerrero, 09] Guerrero, Luis. Documento presentación “Análisis y Diseño Orientado a Objetos”. 2007. Taller de UML, Departamento de Ciencias de la Computación, Universidad de Chile. <http://www.dcc.uchile.cl/~luguerre/cc61j/recursos/aydoo.ppt>. Última revisión Noviembre 2009.

[Larman, 99] Larman, Craig. “*UML y Patrones*”. Editorial Prentice Hall. 1999.

[Martínez, 09] Martínez Espinosa, Yulkeidi. “*Introducción de los Test Adaptativos Informatizados en las Universidades Cubanas*”. 2005. <http://www.monografias.com/trabajos23/tests-adaptativos/tests-adaptativos.shtml>. Última revisión Junio 2009.

[Rubio, 08] Rubio L., José Miguel. Documento J2ME-Seccion2, Curso Taller de Programación de Dispositivos Móviles, profesor Escuela de Ingeniería Informática, PUCV, 2º semestre, 2008.

[Rubio, 08a] Rubio L., José Miguel. Documento NET_CF, Curso Taller de Programación de Dispositivos Móviles, profesor Escuela de Ingeniería Informática, PUCV, 2º semestre, 2008.

[UNAD, 09] Sitio Web Universidad Abierta y a Distancia, UNAD. <http://www.unad.edu.co/>. Última revisión Junio 2009.