



ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA
INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

Curso académico 2011/2012
Proyecto de Fin de Carrera

**Histo, un agente que te ayuda
a repasar historia**

Autora: Jennifer Pablos Tenrero
Tutora: Diana Pérez

RESUMEN

Las nuevas Tecnologías de la Información y la Comunicación se han aplicado con éxito a múltiples dominios. En particular, en el dominio de la Educación, la Informática Educativa ha proporcionado grandes beneficios en posibilidad de adaptación, personalización y flexibilidad espacio-temporal.

Los Agentes Conversacionales Pedagógicos son sistemas interactivos que permiten a los estudiantes centrarse en los aspectos en los que tienen mayor dificultad de una forma entretenida y amigable.

En este contexto, se presenta en este Proyecto Fin de Carrera la posibilidad de crear una aplicación de escritorio, un Agente Conversacional Pedagógico llamado Histo cuyo objetivo es ayudar a los alumnos a memorizar y comprender textos de Historia preguntándoles cuestiones de distinta dificultad asociadas a los textos de estudio.

Se considera adecuado este ámbito, dado el elevado número de datos, nombres, fechas y hechos que hay que memorizar, y las posibilidades de interacción que ofrece el ordenador para facilitar el estudio por parte de los estudiantes.

La colaboración de los profesores será también fundamental, ya que serán ellos quienes introduzcan los textos con las preguntas para que los alumnos puedan utilizarlos.

Para llevar a cabo este proyecto, se ha utilizado Java con una base de datos MySQL. Contamos en él con 21 clases de java, entre ellas, habrá algunas con interfaz gráfica, que representarán las ventanas que interactuarán con el usuario. También entre esas clases, tendremos una encargada de realizar la conexión a la base de datos, y una clase, llamada Histo. La cual, mediante un patrón singleton, gestionará todas las instancias a todas las clases sin necesidad de crear objetos, evitando que se puedan perder datos durante una sesión de estudio. Sin embargo, la base de datos, estará compuesta por nueve tablas, en las cuales, se recogen los datos de los usuarios así como de los textos y preguntas insertados, y también los datos de estudio del alumno.

Histo se ha probado con ocho usuarios, que han puntuado con una media de 8.68 el sistema en una escala de 0 a 10. Sus comentarios han sido que es claro y conciso, que es sencillo de usar y que les gustaba obtener corrección en el momento.

AGRADECIMIENTOS

Quiero dar las gracias a todas las personas que han estado a mi lado y han hecho de esto algo posible y real.

Gracias a mi familia, por haber estado siempre ahí, en especial a mi hermana, y a sus amigos, que se prestaron voluntarios para las pruebas, dedicándome su tiempo, a pesar de que no les sobraba.

Gracias a Diana, mi tutora, por su paciencia, su tiempo y su dedicación. Sin su constancia y preocupación no lo habría conseguido.

Gracias a mis amigos, por estar siempre ahí.

Y por último y en especial, a Vanesa, por aguantarme incluso en mis peores días y poner el punto positivo en todo, y a Luis, por tener fe en mí, incluso cuando yo misma la había perdido.

ÍNDICE

1.	INTRODUCCIÓN	1
2.	OBJETIVOS	5
2.1.	DESCRIPCIÓN DEL PROBLEMA.....	5
2.2.	OBJETIVOS	5
2.3.	ESTUDIO DE ALTERNATIVAS	6
2.3.1.	Paradigmas de programación	6
2.3.2.	Lenguajes de programación	7
2.3.3.	Gestores de bases de datos	8
2.3.4.	Toma de decisiones.....	10
2.4.	METODOLOGÍA EMPLEADA.....	11
3.	DESCRIPCIÓN INFORMÁTICA.....	13
3.1.	ESPECIFICACIÓN	13
3.1.1.	Análisis de requisitos.....	13
3.1.2.	Casos de uso	14
3.2.	DISEÑO	15
3.2.1.	Arquitectura a alto nivel.....	15
3.2.2.	Diseño de la base de datos.....	17
3.3.	ALGORITMO	20
4.	PRUEBAS.....	37
4.1.	PRUEBAS UNITARIAS	37
4.2.	PRUEBAS DE CAJA BLANCA	37
4.3.	PRUEBAS DE CAJA NEGRA	39
4.4.	PRUEBAS DE INTEGRACIÓN.....	41
4.5.	PRUEBAS DE VALIDACIÓN	43
4.6.	PRUEBAS DE ACEPTACIÓN	47
5.	CONCLUSIONES Y TRABAJO FUTURO	53
5.1.	CONCLUSIONES	53
5.2.	TRABAJO FUTURO	57
6.	BIBLIOGRAFÍA.....	59
	ANEXO I	61
	ANEXO II	63
	ANEXO III	65
	ANEXO IV.....	67

ANEXO V	69
ANEXO VI	71
ANEXO VII	75

ÍNDICE FIGURAS

Figura 1: Ejemplo Histo	2
Figura 2: Modelo en cascada.....	11
Figura 3: Diagrama de casos de uso	14
Figura 4: Arquitectura a alto nivel.....	16
Figura 5: Modelo E/R.....	18
Figura 6: Modelo relacional	19
Figura 7: Tabla estadísticas_alumno	19
Figura 8: Tabla pregunta	20
Figura 9: Tabla preg_test.....	20
Figura 10: Tabla preg_palabra.....	20
Figura 11: Tabla preg_corta	20
Figura 12: Acceso a Histo	20
Figura 14: Inicio profesor	21
Figura 13: Inicio alumno.....	21
Figura 15: Editor de textos para nuevo texto.....	21
Figura 16: Editor de textos para nuevo texto con datos.....	22
Figura 17: Editor de textos para nuevo texto desde archivo	22
Figura 18: Tipo de pregunta	23
Figura 19: Pregunta Corta	23
Figura 20: Pregunta palabra	24
Figura 21: Pregunta Test	24
Figura 22: Editor de textos para texto ya existente	24
Figura 23: Editor de textos para texto ya existente con datos	25
Figura 24: Histo en modo aleatorio.....	26
Figura 25: Histo en modo normal.....	27
Figura 26: Histo modo normal reanudar texto	27
Figura 27: Histo modo normal mostrando texto reanudado	28
Figura 28: Histo modo normal intentando reanudar texto no existente.....	29
Figura 29: Pregunta corta alumno.....	33
Figura 30: Autocorrección pregunta corta	33
Figura 31: Pregunta corta siguiente	34
Figura 32: Pregunta de tipo palabra alumno	34
Figura 33: Pregunta tipo palabra corregida	35
Figura 34: Pregunta tipo test alumno corregida	35
Figura 35: Pregunta tipo test alumno	35
Figura 36: Diagrama pruebas de caja blanca	38
Figura 37: Pruebas de integración 1.....	41
Figura 38: Pruebas de integración 2.....	41
Figura 39: Pruebas de integración 4.....	41
Figura 40: Pruebas de integración 5.....	42
Figura 41: Pruebas de validación – Inicio profesor	43
Figura 42: Pruebas de validación – Inicio alumno.....	43

Figura 43: Pruebas de validación – Texto nuevo.....	44
Figura 44: Pruebas de validación – Editor de textos	44
Figura 45: Pruebas de validación – Nuevo texto preguntas para corregir.....	45
Figura 46: Pruebas de validación – Editor de textos preguntas para validar.....	45
Figura 47: Pruebas de validación – Tipos de preguntas.....	46
Figura 48: Pruebas de validación – Inicio alumno dos modos	46
Figura 49: Alumnos realizando pruebas en papel.....	48
Figura 50: Alumna realizando pruebas con Histo	49

ÍNDICE TABLAS

Tabla 1: Tabla comparativa paradigmas de programación.....	7
Tabla 2: Tabla comparativa lenguajes de programación	8
Tabla 3: Tabla comparativa gestores de bases de datos.....	9
Tabla 4: Prueba 1 - Acceso de usuario a la aplicación.....	39
Tabla 5: Prueba 2 – Inserción de texto con título ya existente	39
Tabla 6: Prueba 3 – Apertura de archivo no válido desde el explorador.....	40
Tabla 7: Prueba 4 – Texto sin número de preguntas para aprobar	40
Tabla 8: Prueba 5 – Opción no válida para Histo	40
Tabla 9: Resultados pruebas de aceptación en papel.....	48
Tabla 10: Resultados pruebas de aceptación con Histo.....	50
Tabla 11: Resultados encuesta, valoración de 0 a 10	51

1. INTRODUCCIÓN

Este proyecto surge con la idea y objetivo de facilitar la tarea del estudio de historia a alumnos de todas las edades.

El aprendizaje electrónico (también conocido como e-learning) consiste en el aprovechamiento de las nuevas tecnologías para la educación y el estudio. En unas ocasiones, sustituyendo las clases presenciales, y en otras, complementándolas. En este caso, pretendemos complementar la formación adquirida por el alumno en las clases presenciales, ayudándole a repasar lo aprendido, adquirir soltura en la lectura y comprensión de textos de historia y a prepararse de cara a los exámenes.

Aprovechando los avances tecnológicos y las herramientas que tenemos, hemos querido fusionarlo para encontrar una solución a uno de los grandes problemas de los estudiantes: el estudio en casa.

La ausencia de un profesor capaz de guiar al alumno en su estudio en casa, o incluso, el hecho de no poder obtener la solución de los ejercicios realizados de una manera clara y concisa, incrementa la dificultad para realizar la tarea debido a la inseguridad que crea en el alumno la falta de corrección de la respuesta.

En la aplicación a desarrollar en este proyecto, tendrán acceso tanto alumnos como profesores, de esta forma, será el profesor quién introduzca los textos, las preguntas a contestar y los requisitos de dificultad de cada uno de los textos, permitiéndole así, indicar implícitamente a los alumnos qué sucesos o datos son los que tienen mayor importancia en un texto determinado. También añadirá las respuestas para todas las preguntas formuladas. Esto permitirá que la aplicación pueda dar una corrección instantánea a las respuestas aportadas por el alumno, que podrá ver y comprender cuál ha sido el error en caso de haberse equivocado.

Mediante la lectura de textos y la resolución de preguntas, se centrará la atención del alumno sobre los puntos relevantes de la etapa a estudiar, haciendo que la tarea resulte amena, gracias a la continua interacción con el sistema y el lenguaje natural empleado en éste para la comunicación con el alumno.

Tratando que la tarea del profesor sea lo más fácil posible, se le permite crear textos nuevos, y editar los ya existentes. Se le permitirá abrir desde un archivo un documento de texto, que se copiará directamente en el campo, o bien, abrir un documento de Word o de pdf para poder copiar el texto deseado y pegarlo en el campo correspondiente del editor.

Para hacer más dinámica y entretenida la sesión de estudio, las preguntas que se le presentarán al alumno podrán requerir tres tipos de respuestas diferentes: de respuesta corta, que se contestarán con unas cuantas frases sencillas a modo de resumen, de tipo test, en las que se presentarán tres opciones, para que el alumno seleccione la que considere acertada, y por último, con respuesta de una palabra, que se contestarán con una palabra que se adapte a la definición, una fecha, o un nombre.

También se intentará motivar al alumno a través del nivel. El alumno comenzará en un nivel inicial, y solo se le presentarán los textos cuyo nivel (elegido por el profesor) coincida con el del alumno. De esta forma podremos asegurar que el alumno no tendrá que resolver textos de niveles muy distintos del suyo. Esto es importante debido a que si a un alumno que está

empezando, se le presentan textos muy complejos, podría perder la motivación. También puede ocurrir lo contrario, es decir, si a un alumno que ha avanzado y adquirido conocimientos, se le plantean textos muy sencillos, podría aburrirse, y es lo que la aplicación pretende evitar.

Existen otros tipos de aplicaciones orientadas a la ayuda al estudio. Esta aplicación podría haberse realizado como una aplicación web en lugar de aplicación de escritorio. El motivo principal que llevó a la toma de esta decisión tuvo que ver con el hecho de que la aplicación no tenga un rango de edad definido para uso. Esto podría llevar a que alumnos de cortas edades pudieran usarla, con el consiguiente problema de que no tuviesen una conexión a internet en sus casas, o la preocupación por los padres, que en ocasiones se muestran reacios a que sus hijos tengan que navegar por internet. Con la aplicación de escritorio, evitamos este problema haciendo más fácil el acceso al agente.

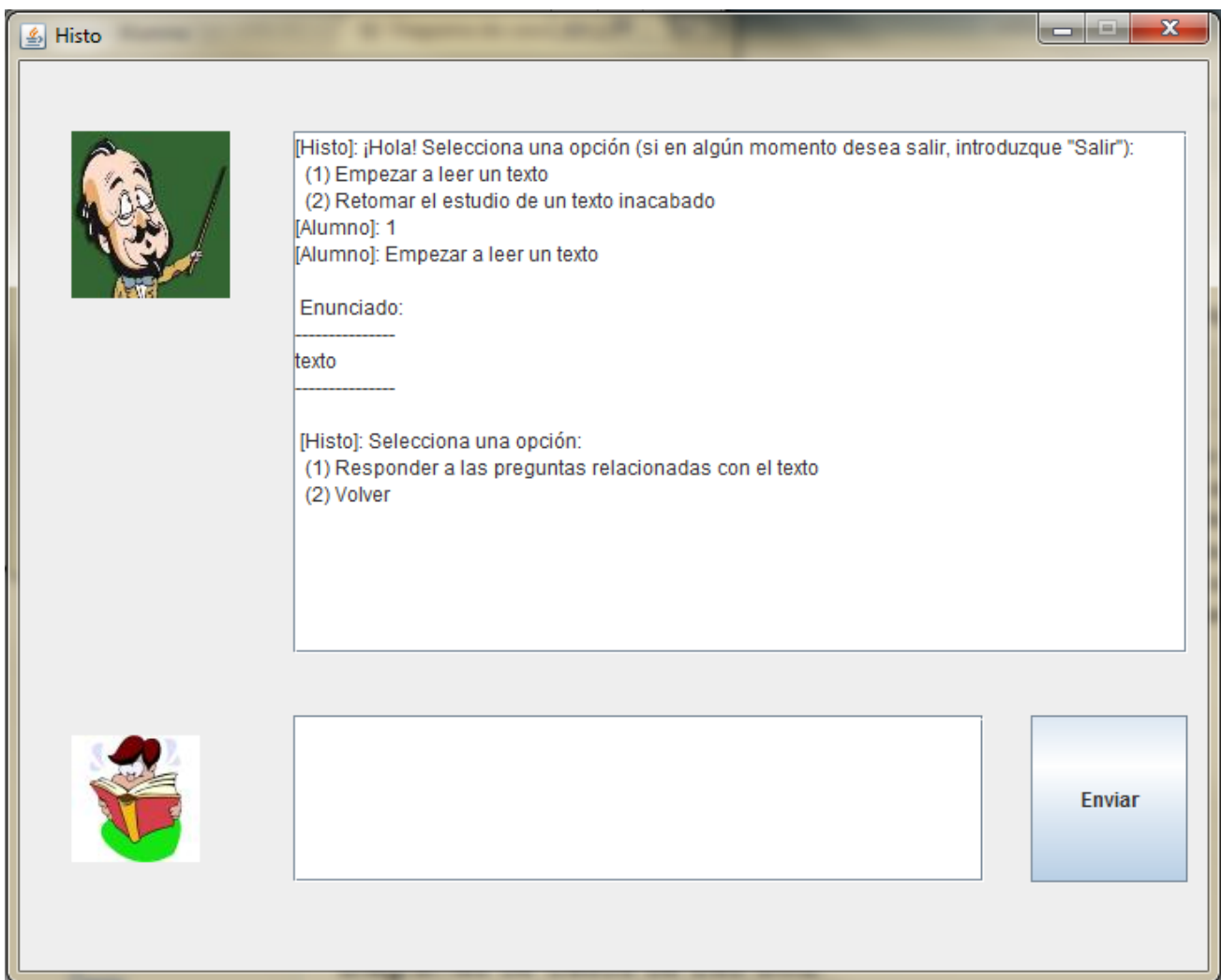


Figura 1: Ejemplo Histo

El presente documento se organiza en los siguientes apartados:

-El capítulo 2, que presenta los objetivos, junto con la descripción del problema, que fue la motivación inicial para llevar a cabo este proyecto. También se presenta la toma de

decisiones que se llevó a cabo en un inicio para escoger las herramientas a usar así como el lenguaje de programación.

-El capítulo 3 describe la aplicación desde los ocho requisitos funcionales, y los tres no funcionales, los casos de uso, la arquitectura a alto nivel, el diseño de la base de datos, así como el algoritmo en que se basa toda la aplicación.

-El capítulo 4 proporciona las pruebas y los resultados obtenidos en ellas. Comenzando por las pruebas unitarias, de caja blanca y caja negra, hasta las pruebas de integración, validación, y por último, de aceptación.

-Finalmente, el capítulo 5 termina el documento con las principales conclusiones, resaltando cómo se han cumplido los objetivos marcados y líneas de trabajo futuro.

2. OBJETIVOS

2.1. DESCRIPCIÓN DEL PROBLEMA

La tarea de estudiar ciertas asignaturas puede resultar tediosa si el estudio se realiza mediante la simple lectura de textos.

Mantener la concentración, en algunas ocasiones puede no ser fácil y aun consiguiéndolo, los resultados obtenidos, no siempre son directamente proporcionales al esfuerzo aplicado para realizar la tarea. La complejidad de las sociedades en algunos períodos de la historia, los cambios políticos continuos, la gran cantidad de fechas y datos de carácter relevante, pueden hacer que la comprensión de una etapa de la historia resulte difícil de entender y más aún de memorizar.

Para solucionar este problema, conviene centrar la atención del alumno sobre los datos o hechos que el profesor considere de mayor importancia. A la par que mantenerle entretenido para que el estudio resulte menos arduo.

La aplicación permitirá que el alumno lea un texto y conteste a las preguntas que el profesor ha insertado para este texto. Las preguntas, de tres tipos diferentes (de respuesta corta, de tipo test y de respuesta de una palabra), serán breves, focalizando los datos importantes para que el alumno los recuerde sin realizar esfuerzo.

Se tendrá presente la evolución del alumno, y los textos que se le plantearán, serán siempre calificados por el profesor de un nivel similar al adquirido por el alumno, para no abrumarle con preguntas muy complicadas, ni aburrirle con conocimientos que ya tiene.

2.2. OBJETIVOS

El objetivo de este proyecto fin de carrera es la creación de una aplicación de escritorio que permita a los alumnos el repaso de los conocimientos adquiridos en la asignatura de historia a través de una interfaz amigable y fácil de utilizar.

El repaso se realizará en base al nivel que el alumno vaya adquiriendo a lo largo del tiempo de uso de la aplicación y a través de los resultados obtenidos en la realización de textos insertados previamente por el profesor.

Para llevar a cabo esta tarea, la aplicación responderá ante dos tipos de usuario diferentes: el profesor, que se encargará de introducir los textos con los que van a trabajar los alumnos, y el alumno, que leerá los textos y contestará las preguntas para completar su aprendizaje.

El profesor se encargará de la introducción de los textos que los alumnos tendrán que leer. La aplicación le irá solicitando los datos necesarios y las preguntas que los alumnos contestarán para afianzar los conocimientos.

Podrá introducir textos nuevos, o modificar los existentes, y escoger entre tres tipos de preguntas diferentes para amenizar la actividad, preguntas de respuesta corta, preguntas de elección múltiple, o preguntas de respuesta de una sola palabra.

Tres ejemplos de preguntas podrían ser:

-¿Por qué en 1700 deciden Inglaterra y Austria presentar al archiduque Carlos como candidato al trono de España?

Respuesta corta: Porque tras la muerte de Carlos II sin descendencia, cuyos reinos serían heredados por Felipe de Anjou, heredero también a la corona de Francia. Ambos países temieron un apoyo inminente de Francia al heredero, que tendría así una gran parte de los territorios europeos bajo su mandato.

-¿En qué año se firma la paz de Utrecht?

Respuesta palabra: 1713

-Felipe de Anjou es aceptado cómo...

Respuesta de tipo test: a. Felipe VI

b. Felipe IV

-> c. Felipe V

Los alumnos podrán repasar historia mediante dos modos de ejecución, aleatorio o mediante menú. En ambos casos, tendrá que leer un texto y contestar a las preguntas que se le plantearán a continuación, cuyas respuestas serán evaluadas. La diferencia entre ambos modos de ejecución, radica en que en el primero, los textos irán apareciendo conforme el alumno vaya finalizando, de manera dinámica, y sin preguntarle qué texto desea continuar, ni tener en cuenta si es un texto iniciado anteriormente o nuevo. Sin embargo, en el segundo modo, el modo normal, el alumno irá indicando qué desea hacer en cada caso, si empezar un texto nuevo, o iniciar uno ya empezado, o poder elegir el texto que desea leer.

Con esto se pretende que el alumno pueda extraer los datos importantes del texto, observando las preguntas que su profesor ha insertado, y pueda memorizar fechas, nombres y datos de importancia elevada con mayor facilidad que con la simple lectura de un texto.

2.3. ESTUDIO DE ALTERNATIVAS

2.3.1. Paradigmas de programación

Opciones

Paradigma imperativo

Constituye una abstracción con respecto a ensamblador, mediante sentencias y variables, que indican a la máquina cómo llevar a cabo una determinada tarea. Son lenguajes imperativos C, Pascal y Fortran.

Paradigma funcional

Basado en el cálculo lambda, es un paradigma que se apoya principalmente en el uso de funciones aritméticas. Haskell, Lisp son lenguajes funcionales.

Paradigma orientado a objetos

Este paradigma se centra en los objetos y en sus interacciones. Sus principales características son la abstracción, la encapsulación, la modularidad y el polimorfismo.

Java, C++, C#, Smalltalk, etc. son lenguajes orientados a objetos.

Conclusión

A continuación se muestra una tabla (Tabla 1) a modo de resumen, con los datos más significativos que se han extraído de los tres paradigmas presentados anteriormente:

	Imperativo	Funcional	Orientado a objetos
Efectos laterales	Sí	No	No
Liberación de memoria automática	No	No	Sí
Eficiencia	Alta	Media	Alta

Tabla 1: Tabla comparativa paradigmas de programación

2.3.2. Lenguajes de programación

Opciones

Java

Java es un lenguaje de programación orientado a objetos. Se ofrece bajo la licencia GNU GPL, de forma que prácticamente todo Java es software libre.

Una de las principales ventajas de Java es la independencia de la plataforma, ya que lo que se pretende es que el código ejecutable que se genera después de la compilación, sea apto para diferentes sistemas operativos e incluso sobre diferente hardware.

Además, Java tiene una serie de características que hacen que resulte más sencillo, como por ejemplo la eliminación de herramientas de bajo nivel, eliminando la gestión de punteros, que suele provocar diversos errores, así como la liberación de memoria, de la que se encarga el recolector de basura automáticamente.

Java es un lenguaje adecuado para el manejo de bases de datos, gracias a JDBC (Java DataBase Connectivity) que facilita enormemente la tarea [1].

C++

Es un lenguaje de programación que se extiende de C que permite la manipulación de objetos, aunque no pertenece propiamente al paradigma de orientación a objetos.

El inconveniente de C++ es que la gestión de memoria puede producir errores graves.

Lenguajes web (PHP)

Es un lenguaje de programación interpretado. Su uso está destinado principalmente a la creación de páginas web dinámicas. Es un lenguaje multiplataforma y tiene una gran conectividad con la mayoría de los sistemas gestores de bases de datos actuales.

PHP permite aplicar técnicas de orientación a objetos, pero el modelo no es muy claro.

Conclusión

En el cuadro que se muestra a continuación se resumen las principales características de los lenguajes de programación descritos anteriormente para facilitar la toma de decisiones:

	Java	C++	PHP
Modelo de objetos	Definido	Híbrido (multiparadigma)	Poco definido
Integración con nuestro sistema gestor de bases de datos	Alta	Media	Alta
Creación de GUI para aplicación de escritorio	Sí	Sí	No
Dificultad en accesos a memoria y punteros	Baja	Alta	X

Tabla 2: Tabla comparativa lenguajes de programación

2.3.3. Gestores de bases de datos

Opciones

MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario, que se ofrece bajo la licencia GNU GPL, de forma que todo el mundo puede descargar, instalar y utilizar MySQL libremente sin ninguna aportación económica.

Su licencia libre hace que sea un gestor de bases de datos muy adecuado para aplicaciones de tamaño medio.

MySQL está diseñado para el trabajo tanto en sistemas operativos Windows como en los sistemas UNIX/LINUX [2].

PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional, orientada a objetos y libre, que ha sido publicado bajo la licencia BSD.

Es multiplataforma, por lo que puede utilizarse con cualquier sistema operativo.

A la hora de resolver consultas, PostgreSQL puede resultar bastante lento, especialmente para bases de datos no muy grandes.

ORACLE

Es un sistema de gestión de base de datos relacional. Está considerado como uno de los más completos por su estabilidad, escalabilidad y soporte multiplataforma.

El gran inconveniente de ORACLE es que su licencia se obtiene bajo pago.

Es un sistema realmente útil para gestionar grandes cantidades de datos en aplicaciones de gran tamaño, pero en una aplicación de tamaño mediano o pequeño, su uso puede resultar complejo y lento.

Conclusión

Para facilitar la toma de decisiones, se ha realizado un cuadro a modo de resumen con las principales características de cada gestor:

	MySQL	PostgreSQL	ORACLE
Licencia gratuita	Sí	Sí	No
Multiplataforma	Sí	Sí	Sí
Aprovechamiento de recursos en bases de datos medianas	Alto	Medio	Bajo

Tabla 3: Tabla comparativa gestores de bases de datos

2.3.4. Toma de decisiones

El paradigma escogido es orientado a objetos, por ser el más cómodo y eficiente para el proyecto que nos ocupa, tal como se muestra en la Tabla 1.

Como lenguaje de programación, se ha elegido Java, ya que observando las características expuestas en la Tabla 2, es el que mejor puede cubrir los objetivos marcados para nuestra aplicación.

Para la base de datos, teniendo en cuenta el tamaño de nuestra aplicación y los datos que se van a manejar en la misma, se ha elegido MySQL como el más adecuado a las necesidades que presenta la aplicación. (Tabla 3)

2.4. METODOLOGÍA EMPLEADA

La metodología de desarrollo empleada ha sido el modelo en cascada [3].

Las fases son:

-Análisis de requisitos: en esta fase se definieron los requisitos que debería cumplir la aplicación, es decir, las funcionalidades que debe recoger como mínimo para que se cumplan los objetivos marcados y tenga el funcionamiento deseado. Serán detallados más adelante.

-Diseño: se diseña la base de datos, con el modelo entidad relación, el paso a modelo relacional, y un diseño esquemático y genérico sobre los módulos y clases que contendrá la aplicación.

-Codificación y pruebas: la fase de la codificación ha sido la más duradera de todas. A la vez que se va implementando el código, es necesario ir probando su correcto funcionamiento para resolver todas las incidencias, errores y problemas que puedan surgir a lo largo del desarrollo. Al finalizar la codificación se realizan las pruebas con el código completo, como las de validación, comparando con los requisitos, y las de aceptación, con un grupo de usuarios.

A lo largo de todo el desarrollo se ha vuelto al inicio siempre que se ha encontrado un error o problema. Retomando los requisitos, modificando el diseño o el código siempre que ha sido necesario.

Las etapas se muestran en el siguiente esquema, de acuerdo al orden en que han sido realizadas:

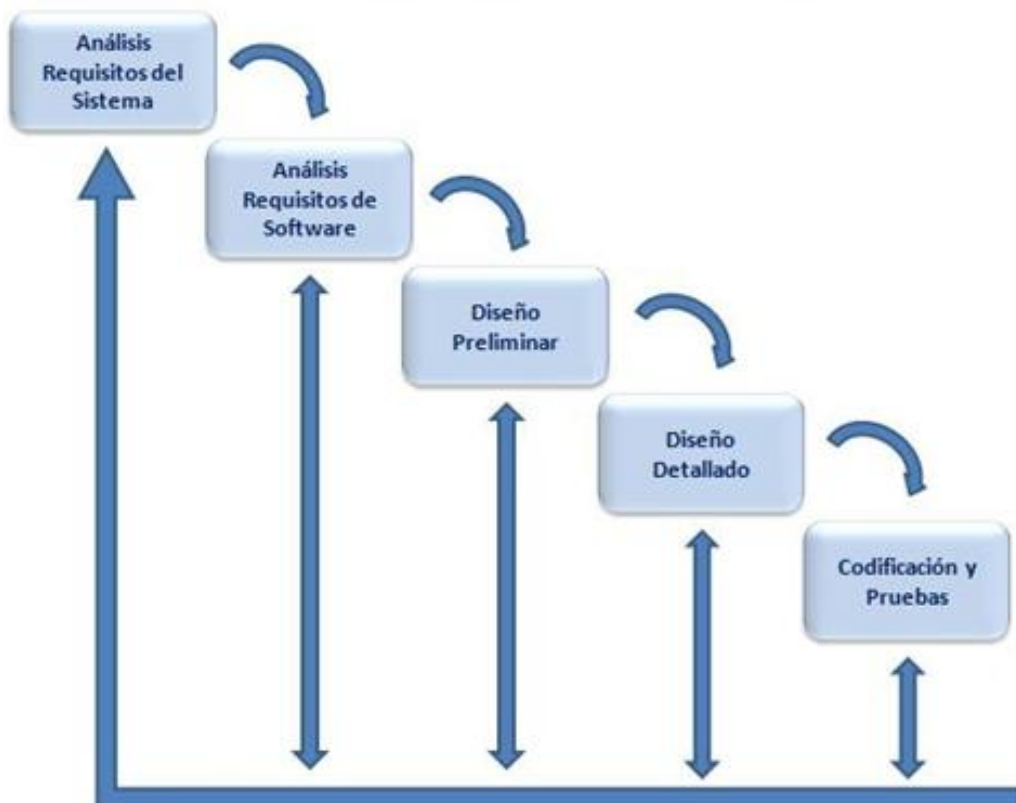


Figura 2: Modelo en cascada

3. DESCRIPCIÓN INFORMÁTICA

3.1. ESPECIFICACIÓN

3.1.1. Análisis de requisitos

REQUISITOS FUNCIONALES

1. En el sistema habrá dos tipos diferentes de usuarios, con diferentes interfaces gráficas, adaptadas a sus opciones. Estos usuarios serán profesores, que podrán introducir textos y preguntas, y alumnos, que podrán consultar estos textos y responder a las preguntas propuestas.
2. Todo texto tendrá un nivel de dificultad, seleccionado por el profesor, para organizar el orden en el que repasará los textos el alumno.
3. Todos los textos tendrán también un número de preguntas, introducido por el profesor, que indicará el número de preguntas del texto, que el alumno tendrá que contestar correctamente, para que se pueda considerar que dicho texto se ha finalizado con éxito.
4. Con cada texto se introducirán mínimo cinco preguntas, que podrán ser de tres tipos diferentes.
 - 4.1. Preguntas de respuesta corta: el profesor introducirá algunas frases como respuesta, y se le mostrarán al alumno para que corrija él mismo su respuesta mediante una comparación.
 - 4.2. Preguntas de elección múltiple: para las que se le facilitará al profesor una interfaz gráfica acorde a las necesidades que presenta este tipo de pregunta.
 - 4.3. Preguntas con respuesta de una palabra, a las que se contestará con una palabra que se adapte a la definición, una fecha, o un nombre propio.
5. El alumno tendrá la posibilidad de empezar la sesión de estudio de dos formas:
 - 5.1. En modo normal, de forma que él mismo irá solicitando si desea empezar un nuevo texto o reanudar uno que haya quedado inacabado en una sesión de estudio anterior.
 - 5.2. En modo aleatorio, de forma que le irán apareciendo textos de su nivel para que los lea, y acto seguido contestar las preguntas. Al terminar un texto, con todas sus preguntas, comenzará un nuevo texto hasta que el alumno decida terminar con la sesión.
6. Todos los textos que se le mostrarán al alumno serán de su mismo nivel, de manera que no tenga que lidiar con textos para los que no está preparado aún, ni realizar textos que puedan resultarle aburridos por estar familiarizado con ellos.
7. Guardado automático tras finalizar cada pregunta, para evitar pérdidas.
8. Cerrar el sistema pulsando el botón cerrar o tecleando "salir".

REQUISITOS NO FUNCIONALES

1. Posibilidad para el alumno de dar las respuestas en lenguaje natural, haciendo más fácil e interesante repasar historia.
2. Rapidez a la hora de mostrar al alumno los textos que se recuperan de la base de datos y para corregir las preguntas realizadas por el alumno.
3. Interfaz amigable y de uso sencillo tanto para el alumno como para el profesor.

3.1.2. Casos de uso

En el diagrama de casos de uso se recoge el comportamiento de la aplicación. Como se muestra en la Figura 3.

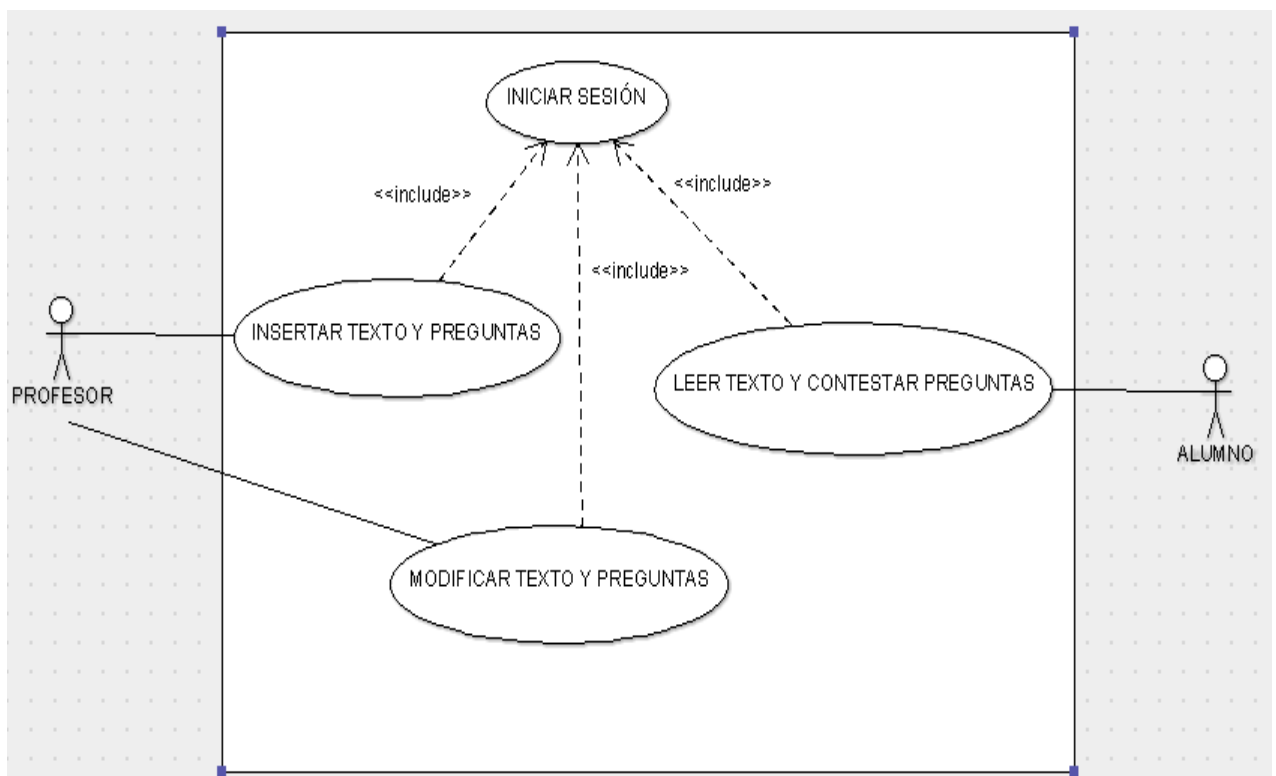


Figura 3: Diagrama de casos de uso

A pesar de tener dos actores para la aplicación, se muestra en un solo diagrama, puesto que ambos deberán iniciar la sesión como usuarios para poder realizar las tareas que deseen.

Para iniciar sesión, deberán introducir un usuario y contraseña, insertado con anterioridad en la base de datos, a través de la pantalla de acceso a histo de la aplicación.

A partir de este momento, solo comparten el módulo de gestión, es decir, el patrón singleton Histo.java y la clase ConexionBD.java que conectará con la base de datos. Las ventanas que se muestran para cada tipo de usuario serán diferentes tanto en cuanto a estética, como en cuanto a implementación.

Para el alumno serán recuperados los textos y preguntas de la base de datos que haya insertado el profesor, pero las ventanas en las que se muestre, serán ligeramente diferentes y adaptadas a las necesidades del usuario.

3.2. DISEÑO

3.2.1. Arquitectura a alto nivel

En la Figura 4 aparecen representados de manera superficial los módulos en los que se encapsulan las distintas clases que componen la aplicación. Como centro en torno al que giran todas las demás clases, encontramos Histo.java, que mediante un patrón singleton gestiona todas las instancias a clases, tanto con interfaz, como sin ella. La clase ConexionBD.java será la encargada de realizar todos los accesos a la base de datos, siendo una clase abstracta, de la que no podrán crearse instancias. La clase Acceso.java, que contendrá la ventana de inicio de sesión, será común para todos los usuarios. A partir de ese momento, encontraremos dos módulos definidos, el del profesor, que contendrá todas las clases que se utilizarán durante la ejecución de la aplicación por parte de un profesor, y el módulo del alumno, que, al igual que el del profesor, contendrá las clases necesarias para que la aplicación funcione para los alumnos.

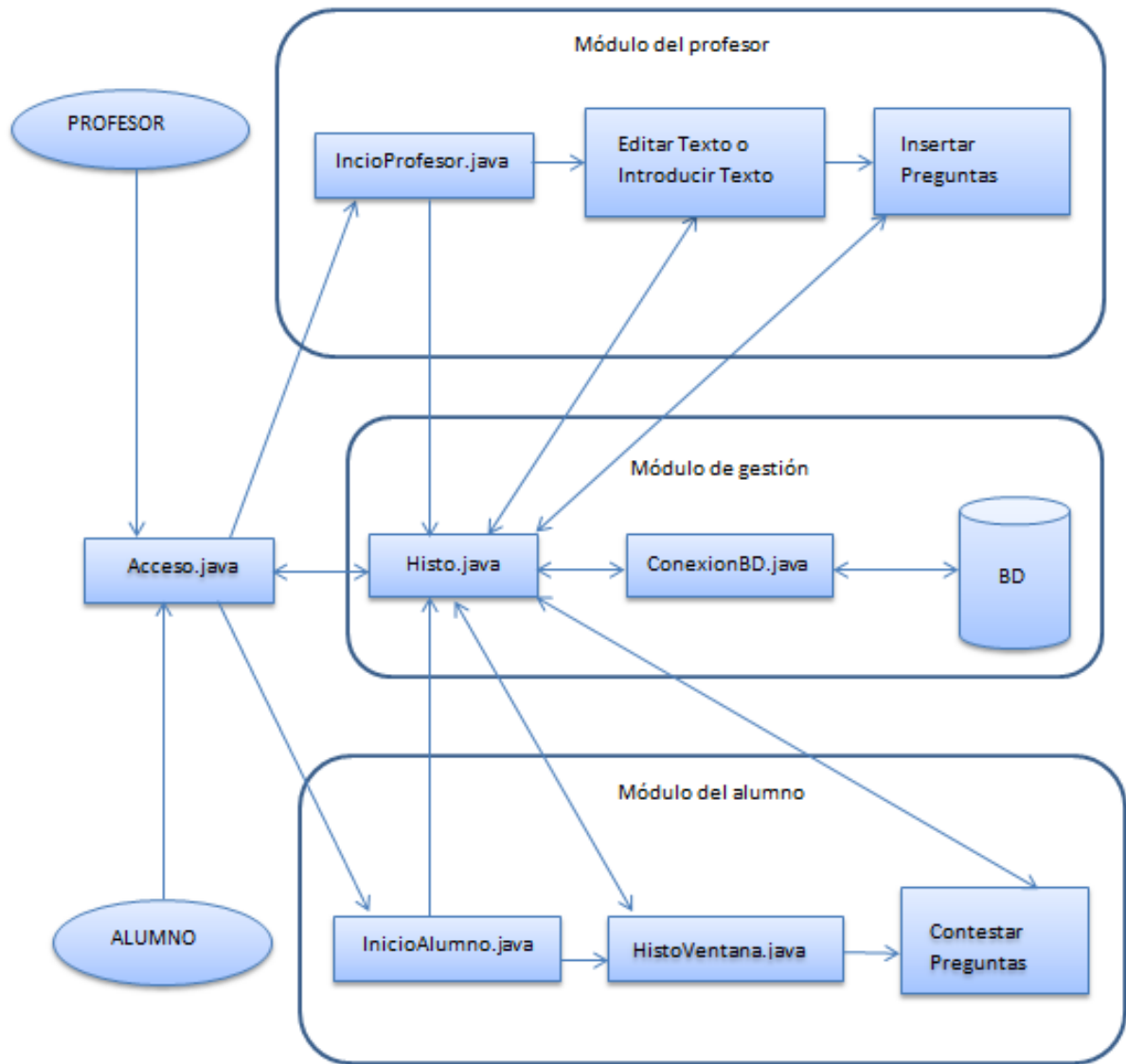


Figura 4: Arquitectura a alto nivel

3.2.2. Diseño de la base de datos

MODELO ENTIDAD RELACIÓN

En el diagrama entidad relación se representan las entidades importantes de un sistema de información junto con sus interrelaciones y atributos.

A lo largo del tiempo que ha durado el desarrollo del proyecto, la base de datos ha sufrido diferentes cambios, con respecto al modelo inicial, debido a las necesidades que han ido surgiendo.

Ha habido que añadir nuevos atributos, ya que poco a poco, se ha ido viendo que hacía falta recoger ciertos datos, y tenía que ser en la base de datos por la persistencia de esos datos.

En la Figura 5, se muestra el diagrama entidad relación de la base de datos de la aplicación, en el que se observan, a grandes rasgos, las entidades que darán lugar a las principales tablas de la base de datos, en la que se apoya toda la aplicación.

También se detallan las cardinalidades de las relaciones. Recogiendo así, datos, como por ejemplo, el hecho de que cada texto, tendrá como mínimo 5 preguntas, y 10 como máximo. Mediante la exclusividad total, se recoge el hecho de que un usuario, será o profesor, o alumno, no podrá ser otra cosa, y solo podrá ser una de las dos, es decir, un alumno no será profesor, y un profesor, no podrá ser alumno.

MODELO RELACIONAL

Basándonos en el esquema del modelo entidad relación que se muestra en la Figura 5, se ha obtenido el esquema relacional de la Figura 6. En él quedan recogidas todas las tablas y los campos de las mismas que componen la base de datos de la aplicación.

Quedan recogidas las claves principales de todas las tablas mediante subrayados, y las claves ajenas, que tendrán una flecha, apuntando hacia la tabla a la que hacen referencia.

Todas las entidades del esquema de la Figura 6, derivan en una tabla en este nuevo esquema. Además, hay dos nuevas tablas, *tiene* y *estadísticas_alumno* que surgen de dos interrelaciones, para evitar las pérdidas de semántica.

En la Figura 7 se muestra un ejemplo de la tabla *estadísticas_alumno* con datos insertados.

En la Figura 8 se muestran tres enunciados sacados de la tabla *pregunta*, que se corresponden con las respuestas de las tablas *preg_test*, *preg_palabra* y *preg_corta*, cuyos ejemplos se muestran en las Figuras 9, 10 y 11 respectivamente.

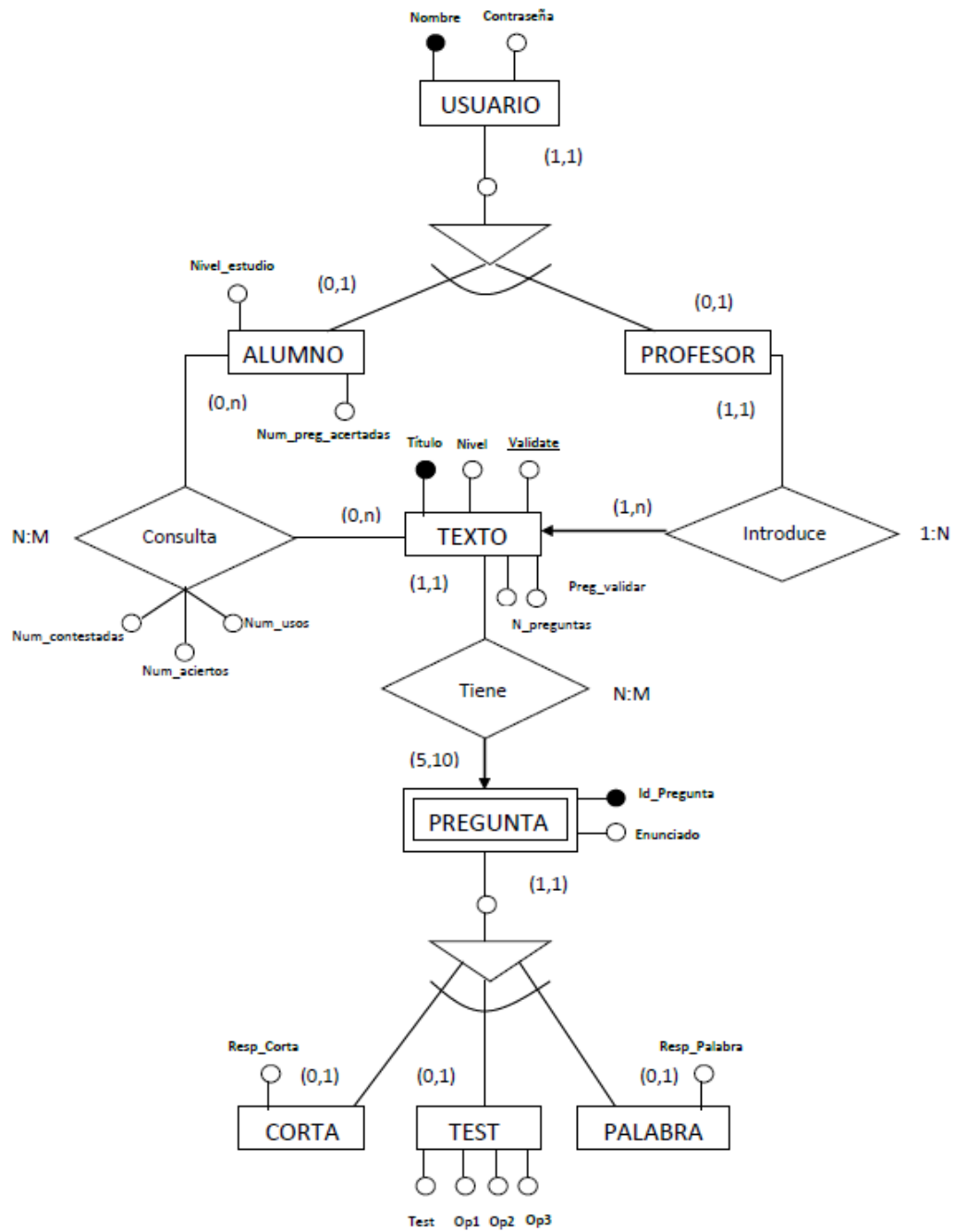


Figura 5: Modelo E/R

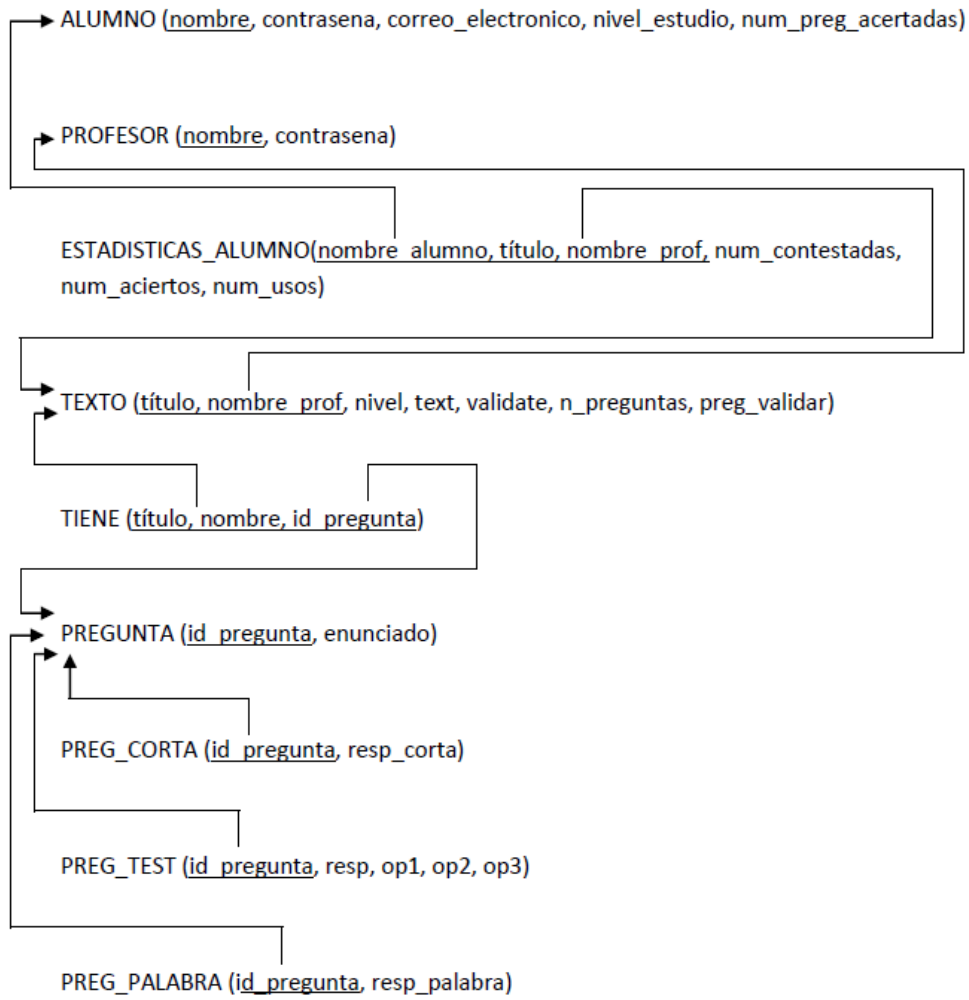


Figura 6: Modelo relacional

NOMBRE_ALUMNO	TÍTULO	NOMBRE_PROF	NUM_CONTESTADAS	NUM_ACIERTOS	NUM_USOS
alumno	Conquista y colonización de américa	profesor1	5	4	1
alumno1	Al-Ándalus, evolución política	profesor1	6	4	1
jennifer	Al-Ándalus, evolución política	profesor1	6	4	1
usuario1	La conquista del Reino Nazarí	jenniferpt	6	3	1

Figura 7: Tabla estadísticas_alumno

ID_PREGUNTA	ENUNCIADO
4	¿A qué homínido pertenecían los restos encontrados en la Sima de los Huesos?
8	Proceso histórico mediante el que los pueblos peninsulares asimilan las formas de vida romanas, dejando elementos tales como el latín y el cristianismo
68	¿Qué supuso la conquista del Reino Nazarí de Granada para los reyes?

Figura 8: Tabla pregunta

ID_PREGUNTA	RESP	OP1	OP2	OP3
4	Homo Eidelbergensis	Homo Eidelbergensis	Homo Ergaster	Homo Antecesor

Figura 9: Tabla preg_test

ID_PREGUNTA	RESP_PALABRA
8	romanización

Figura 10: Tabla preg_palabra

ID_PREGUNTA	RESP_CORTA
68	Fue un instrumento de propaganda política y religiosa.

Figura 11: Tabla preg_corta

3.3. ALGORITMO

A continuación se va a detallar el funcionamiento de Histo, con la ayuda de capturas de pantalla de la aplicación que ilustrarán su funcionamiento.

Histo se divide en dos modos de usuario: alumno y profesor. Las clases que usan en común, son Histo.java, que mediante un singleton se encarga de hacer las transiciones de ventana en ventana conservando la información necesaria, y ConexionBD.java, que contiene todas las conexiones y accesos a la base de datos, tanto para insertar datos, como para recuperarlos.

Tanto alumno como profesor comparten una ventana de inicio, que tendrá como función autenticar al usuario.

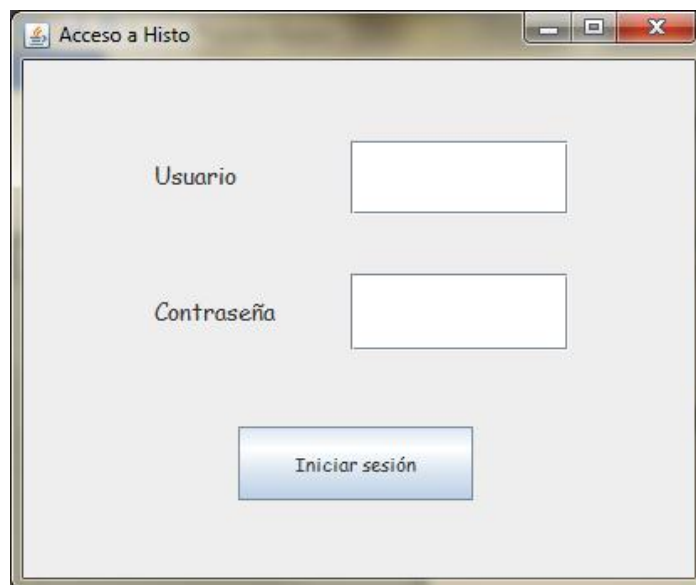


Figura 12: Acceso a Histo

A través del método de Histo.java, iniciarSesion(), se llama al método buscarUsuario() de ConexionBD.java. Se devolverá si el usuario es profesor o alumno, y en función del tipo de usuario que devuelva se presentará la ventana de inicio, que será diferente según el tipo de usuario.



Figura 14: Inicio alumno

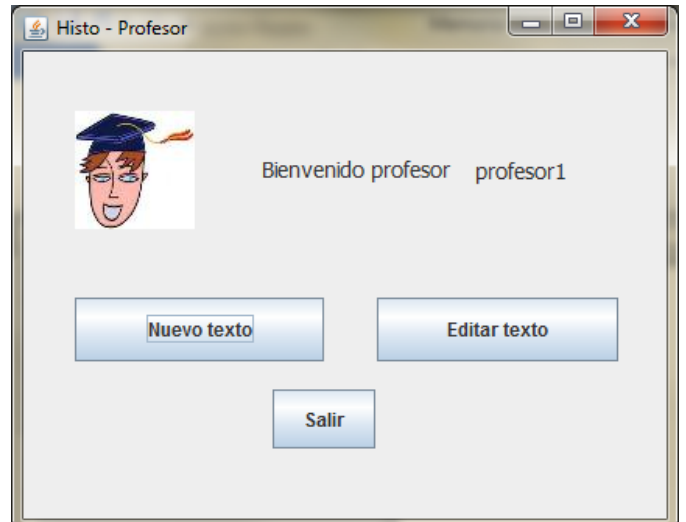


Figura 13: Inicio profesor

PROFESOR:

En el caso de que el usuario sea profesor, se le plantean dos opciones, insertar un texto nuevo o editar un texto ya existente.

La implementación de ambas opciones es muy similar. En el caso de pulsar sobre nuevo texto, se abriría la ventana mostrada en la Figura 15.

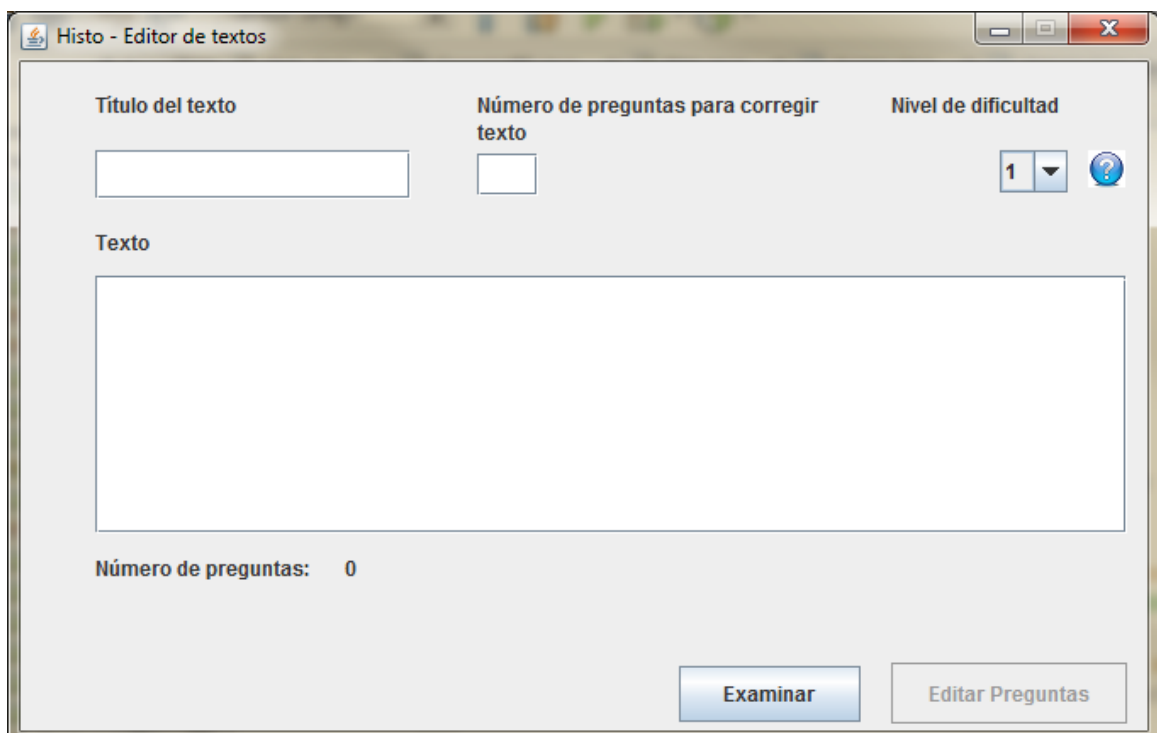


Figura 15: Editor de textos para nuevo texto

Antes de proceder a la edición de las preguntas habrá que insertar obligatoriamente un título, un texto, un nivel de dificultad para el texto y el número de preguntas que el alumno tendrá que acertar para que se considere que ha aprobado el texto. Una vez se haya completado todo, el botón Editar Preguntas se activará.

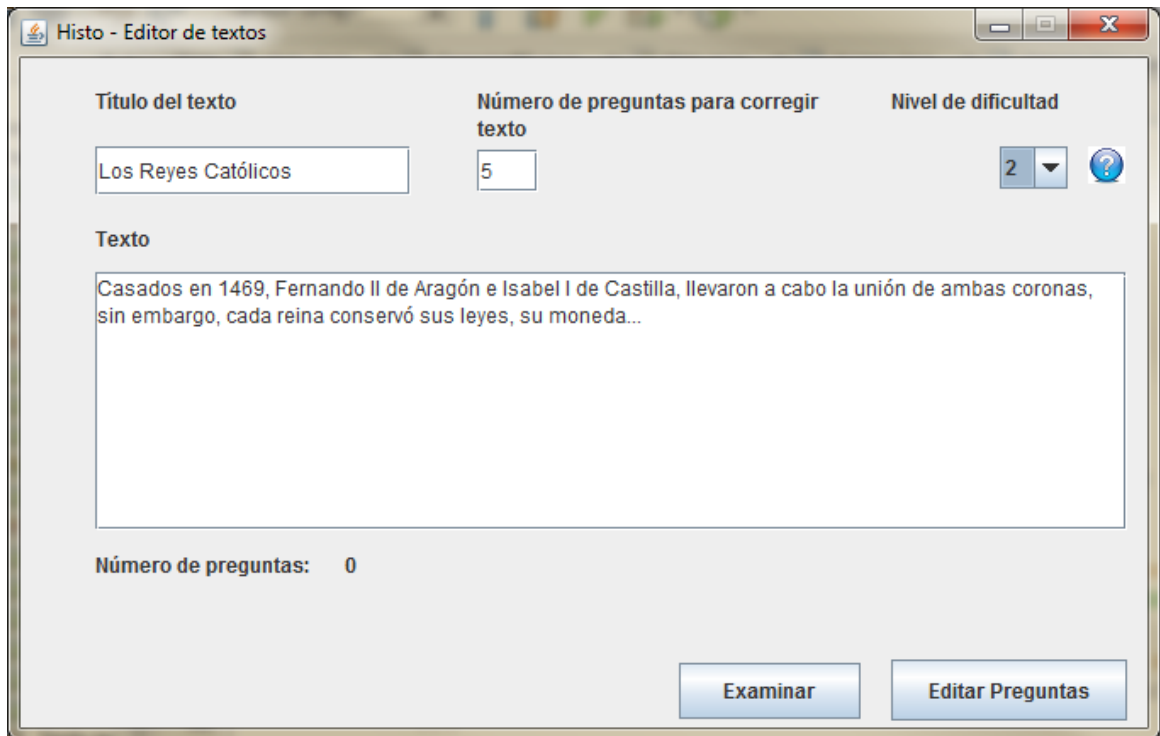


Figura 16: Editor de textos para nuevo texto con datos

Otra alternativa a insertar el texto, será copiarlo desde un .txt, un .doc, un .pdf...para ello, se pincha sobre el botón Examinar, y se abrirá el programa correspondiente permitiendo copiar y pegar.

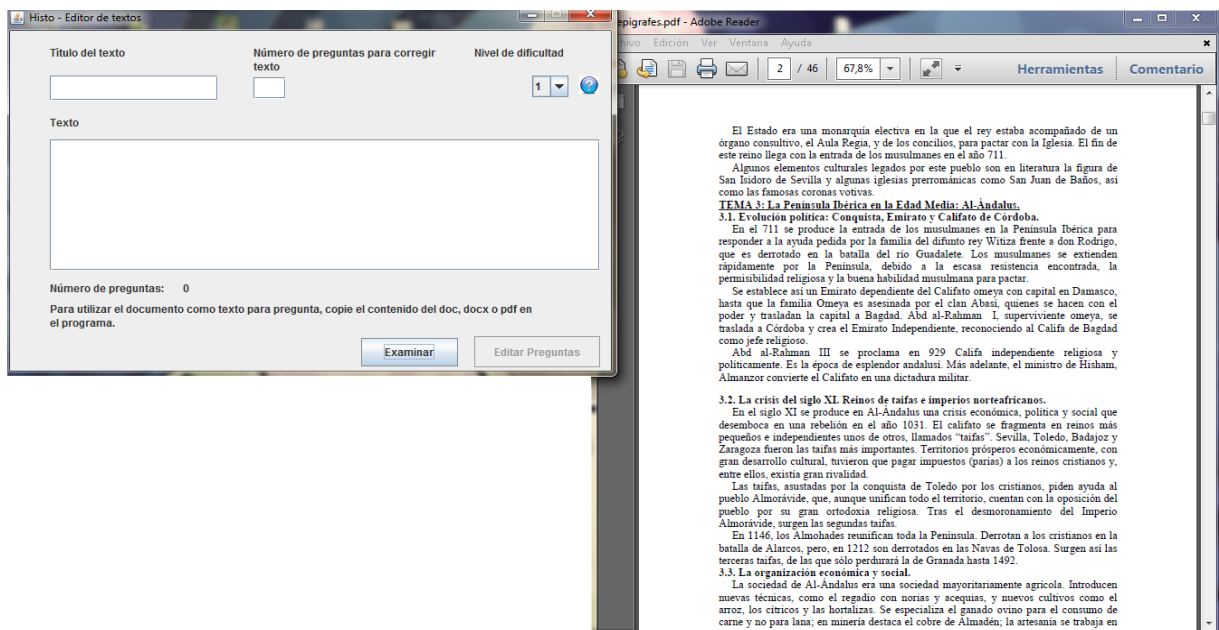
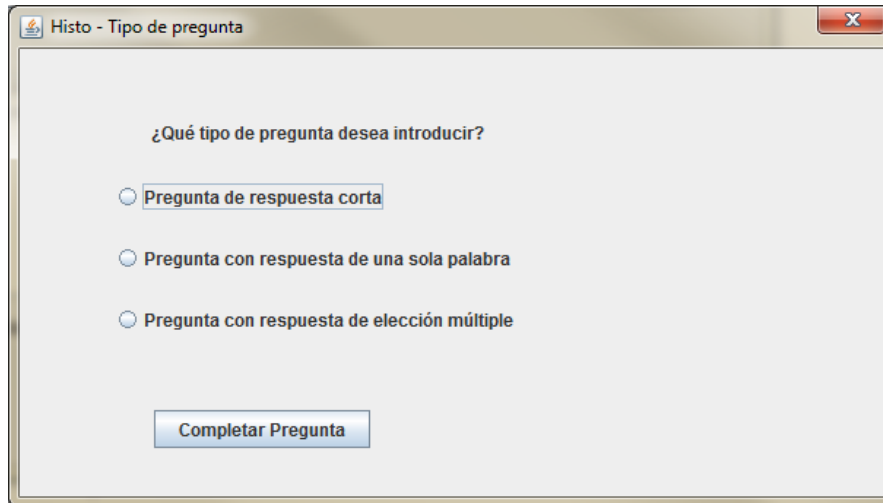


Figura 17: Editor de textos para nuevo texto desde archivo

Al pulsar sobre el botón Editar Preguntas, el texto será almacenado en la base de datos, a través de un método en Histo.java que se encargará de realizar la llamada a ConexionBD.java.

En la Figura 18 se muestra la pantalla que aparece tras pulsar el botón. En ella se ofrece la posibilidad de elegir entre tres tipos de preguntas diferentes.



¿Qué tipo de pregunta desea introducir?

Pregunta de respuesta corta

Pregunta con respuesta de una sola palabra

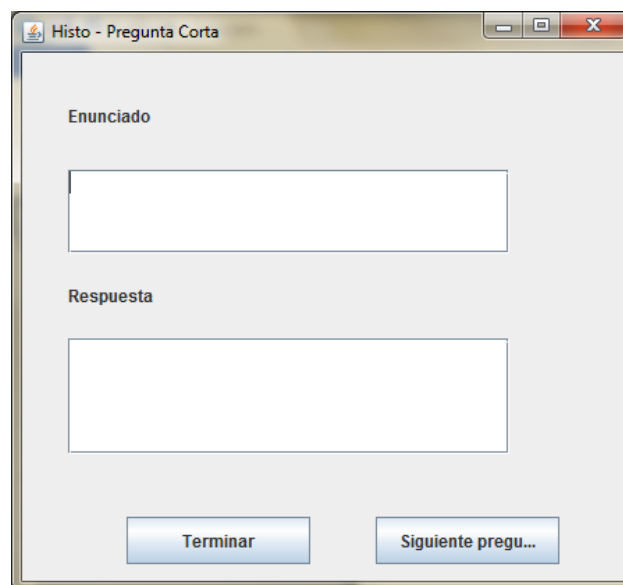
Pregunta con respuesta de elección múltiple

Completar Pregunta

Figura 18: Tipo de pregunta

Tras seleccionar el tipo aparecerá una ventana adaptada a los datos que esa pregunta requiere para ser insertada en la base de datos. Una vez más, Histo.java se encargará de crear la instancia de la clase que corresponda según la selección realizada por el profesor.

Las Figuras 19, 20 y 21 muestran las pantallas de los tres tipos de pregunta posibles.



Enunciado

Respuesta

Terminar Siguiente pregu...

Figura 19: Pregunta Corta

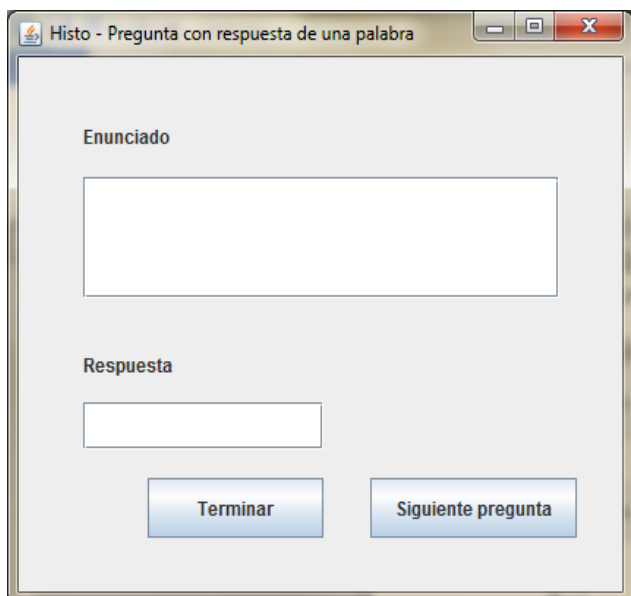


Figura 20: Pregunta palabra

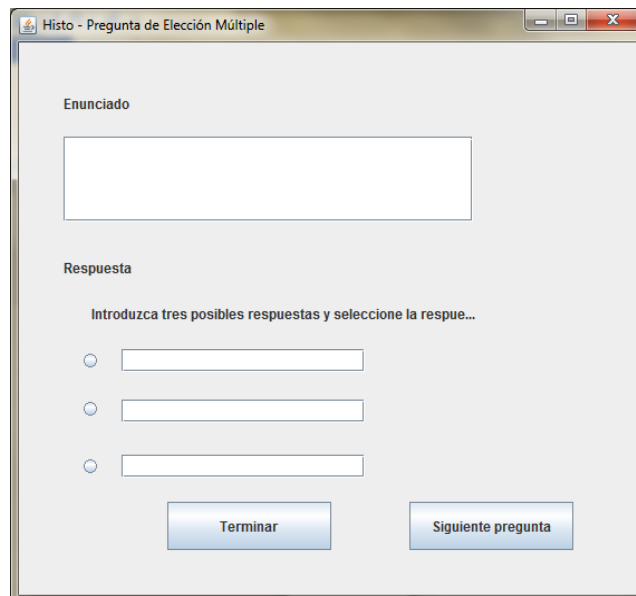


Figura 21: Pregunta Test

Al pulsar sobre el botón siguiente pregunta, vuelve a aparecer la ventana de selección de tipo, no sin antes guardar la pregunta en la base de datos y actualizar el número de preguntas del texto relacionado con dichas preguntas.

En el caso de haber elegido al inicio la opción de editar un texto ya existente, se muestra la ventana que refleja la Figura 22 a continuación.

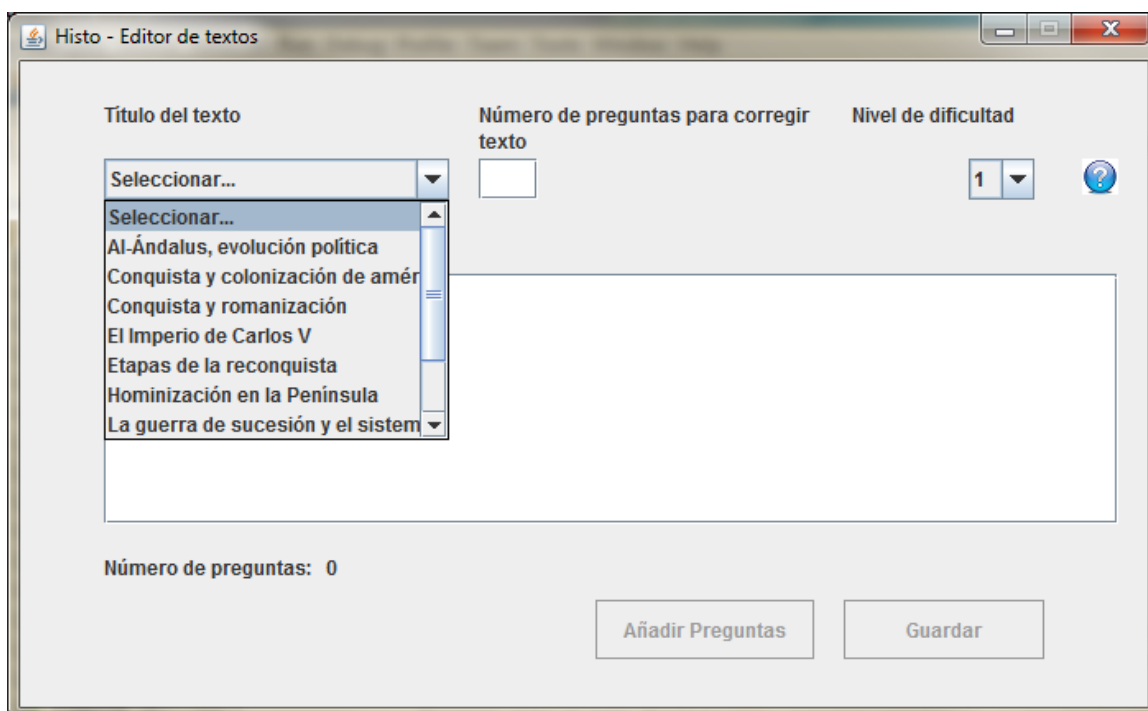


Figura 22: Editor de textos para texto ya existente

En el combo se mostrarán todos los textos introducidos por el profesor que está usando la aplicación. Al seleccionar el texto, se activarán los botones de guardar y añadir preguntas.

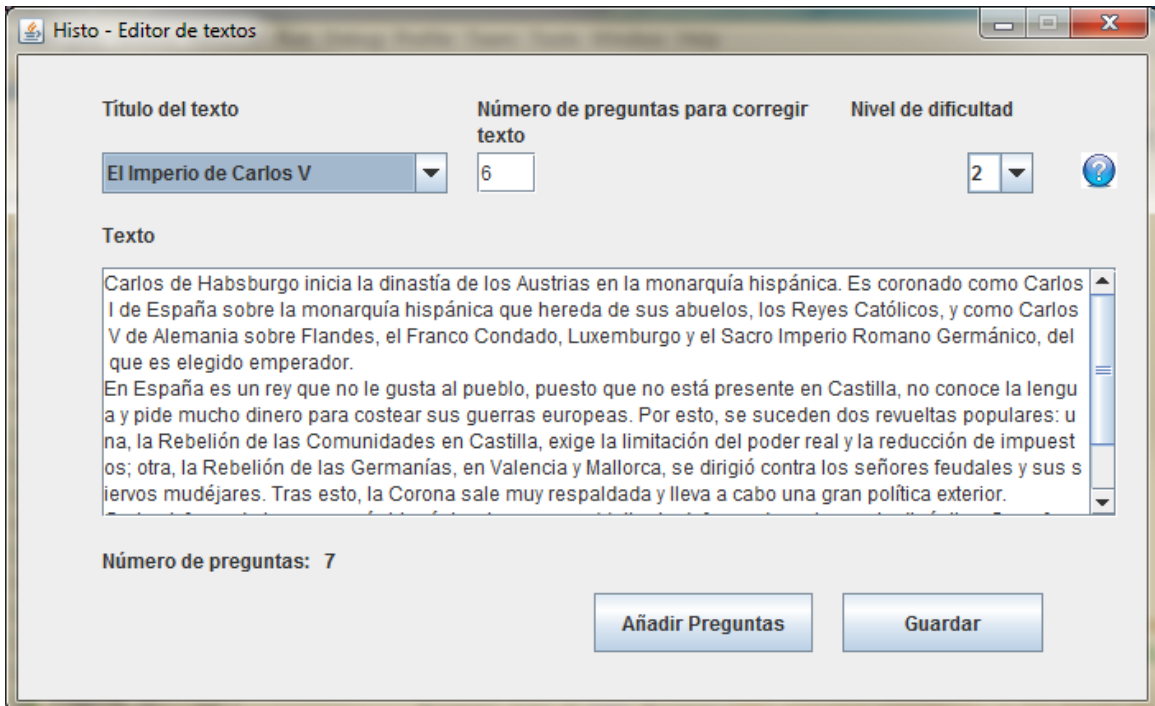


Figura 23: Editor de textos para texto ya existente con datos

El botón guardar, gracias al método getInstance() del singleton Histo.java, podrá llamar al método insertarTexto() de ConexionBD.java, que realizará un update sobre la tabla texto de la base de datos, actualizando y guardando los cambios que el usuario haya introducido.

El botón añadir preguntas funcionará de la misma manera que editar preguntas en el caso del texto de nueva inserción.

ALUMNO:

Como se ve en la Figura 14, en la que aparece la ventana de inicio del alumno, éste tendrá dos opciones para empezar su repaso de historia. Un modo aleatorio, y un modo normal. La diferencia entre ambos modos, será que, en el caso del modo normal se podrá elegir entre empezar un texto nuevo, o reanudar uno ya empezado, pudiendo seleccionar este último a partir de una palabra clave contenida en el título. Sin embargo, en el modo aleatorio, irán saliendo textos con sus preguntas, uno detrás de otro, sin posibilidad de escoger el que venga a continuación.

Al iniciar el modo aleatorio, se abre HistoVentana.java, pero su comportamiento será ligeramente distinto a cuando se abre en modo normal. HistoVentana.java, mediante fases que determinan en qué punto de la ejecución se encuentra la aplicación, realizará la petición de un texto a Histo.java, que se encargará de llamar al método escogerTexto() de ConexionBD.java.

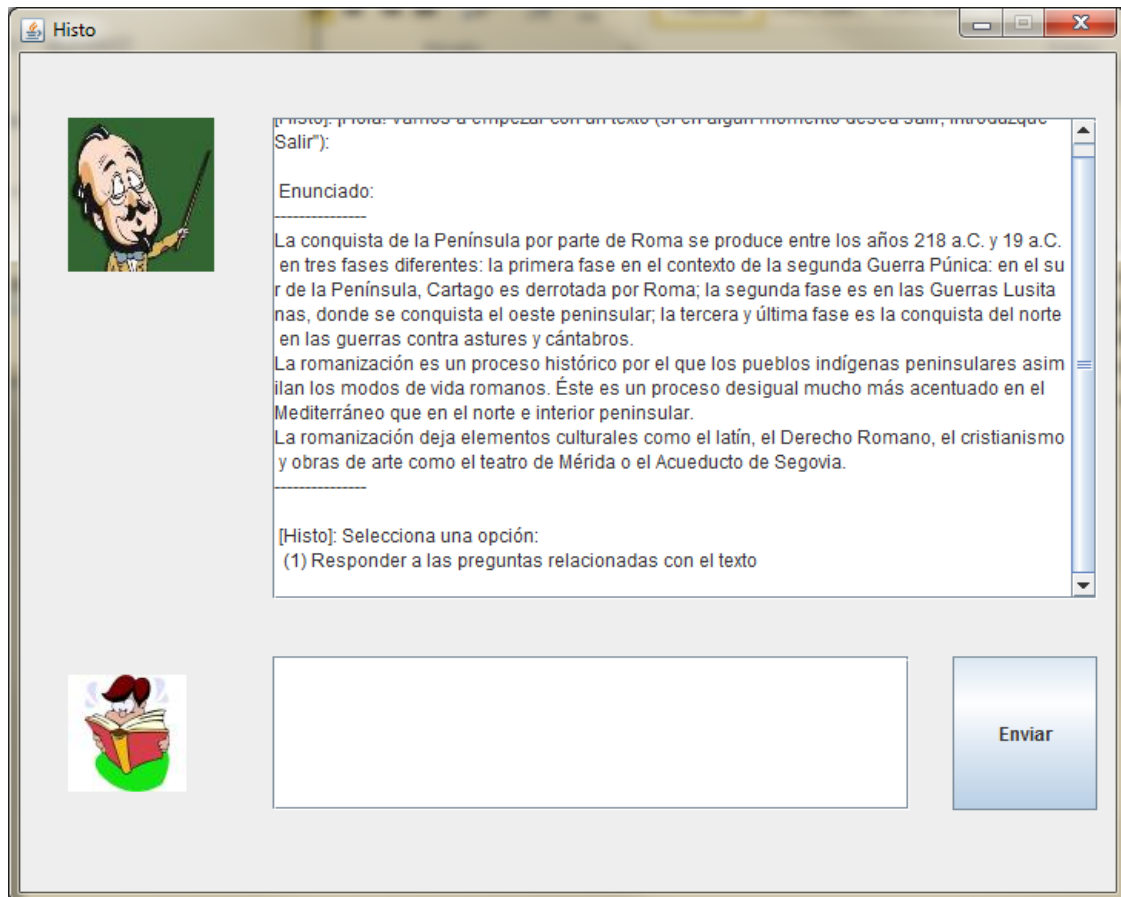


Figura 24: Histo en modo aleatorio

Este método realiza una select sobre la base de datos, donde se recuperan todos los datos necesarios como el título, el nivel de dificultad, etc. Se realiza una combinación entre la tabla texto y la tabla estadísticas_alumno. Y se impondrán unas condiciones para la recuperación, que el nivel de texto y el del alumno sea el mismo, y que el texto esté validado, es decir, que tenga más de cinco preguntas. Ordenados por número de usos y título, devolverá el primero de los que cumpla todas las condiciones. Se actualiza el número de usos para evitar que este texto salga otra vez de forma muy seguida.

HistoVentana.java mostrará el texto y acto seguido solicita al alumno que confirme cuando desee empezar a contestar preguntas.

En el caso del modo normal, tenemos dos opciones.

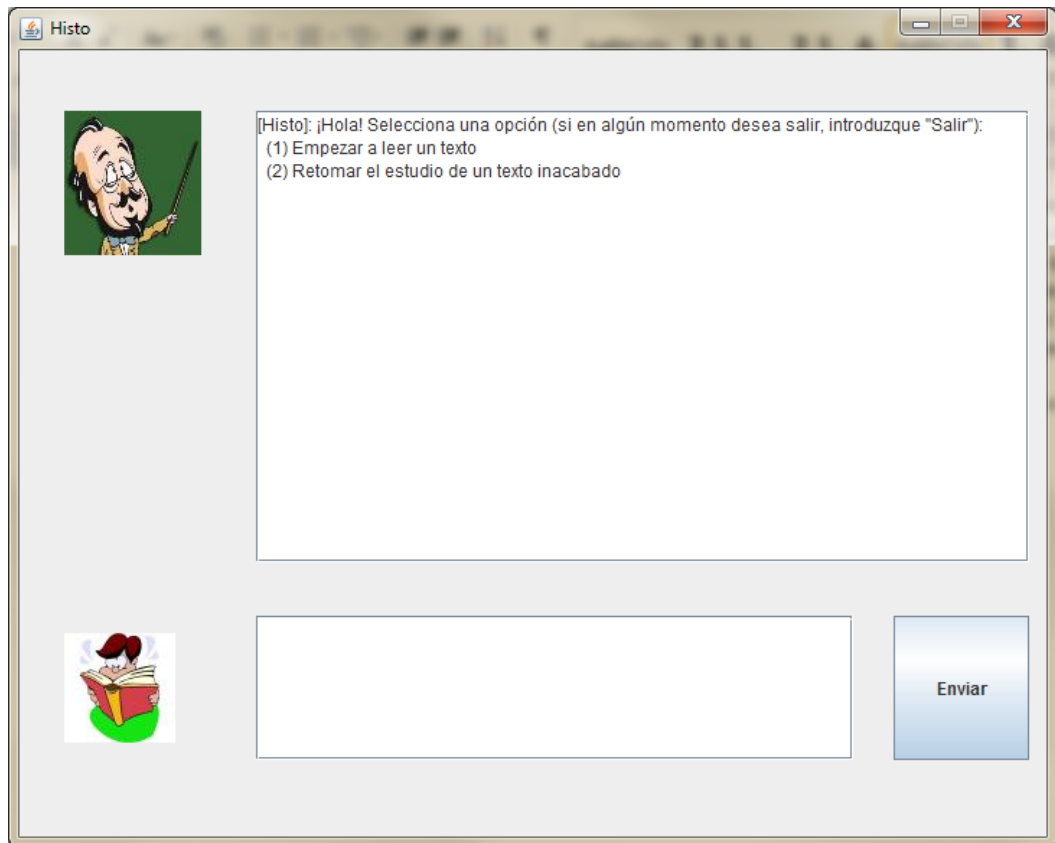


Figura 25: Histo en modo normal

Si seleccionamos la opción 1, el algoritmo para recuperar el texto será igual que el anterior. Sin embargo, en el caso de seleccionar la opción 2, tenemos la posibilidad de elegir qué texto queremos reanudar.

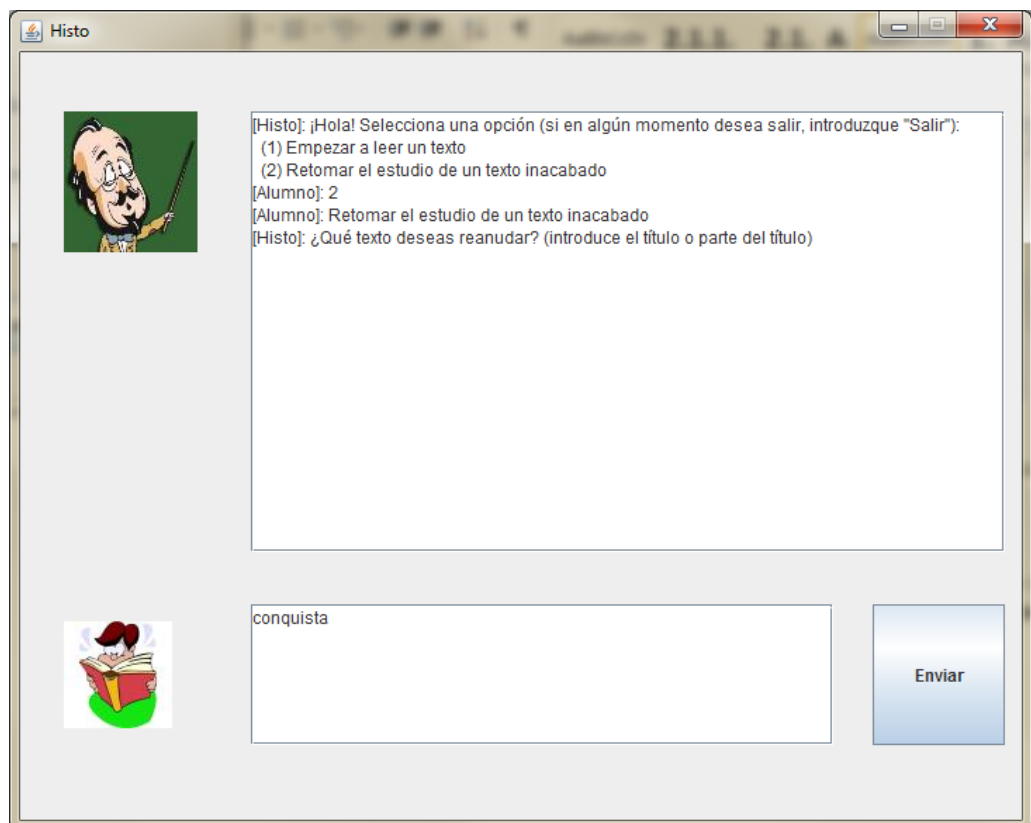


Figura 26: Histo modo normal reanudar texto

Si escribimos una palabra contenida en el título de alguno de los textos, no muestra ese texto.

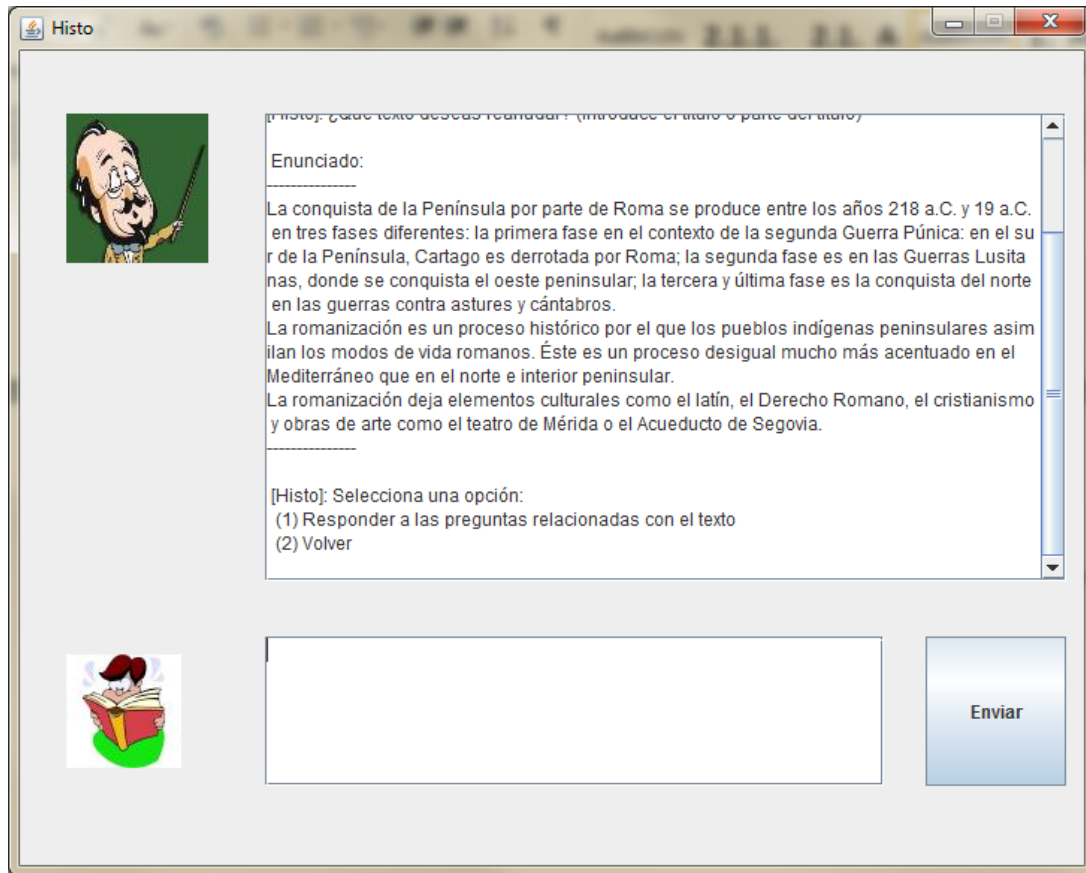


Figura 27: Histo modo normal mostrando texto reanudado

En caso de que con la palabra introducida no se encuentre ningún texto, se muestra la lista de títulos de textos que están empezados.

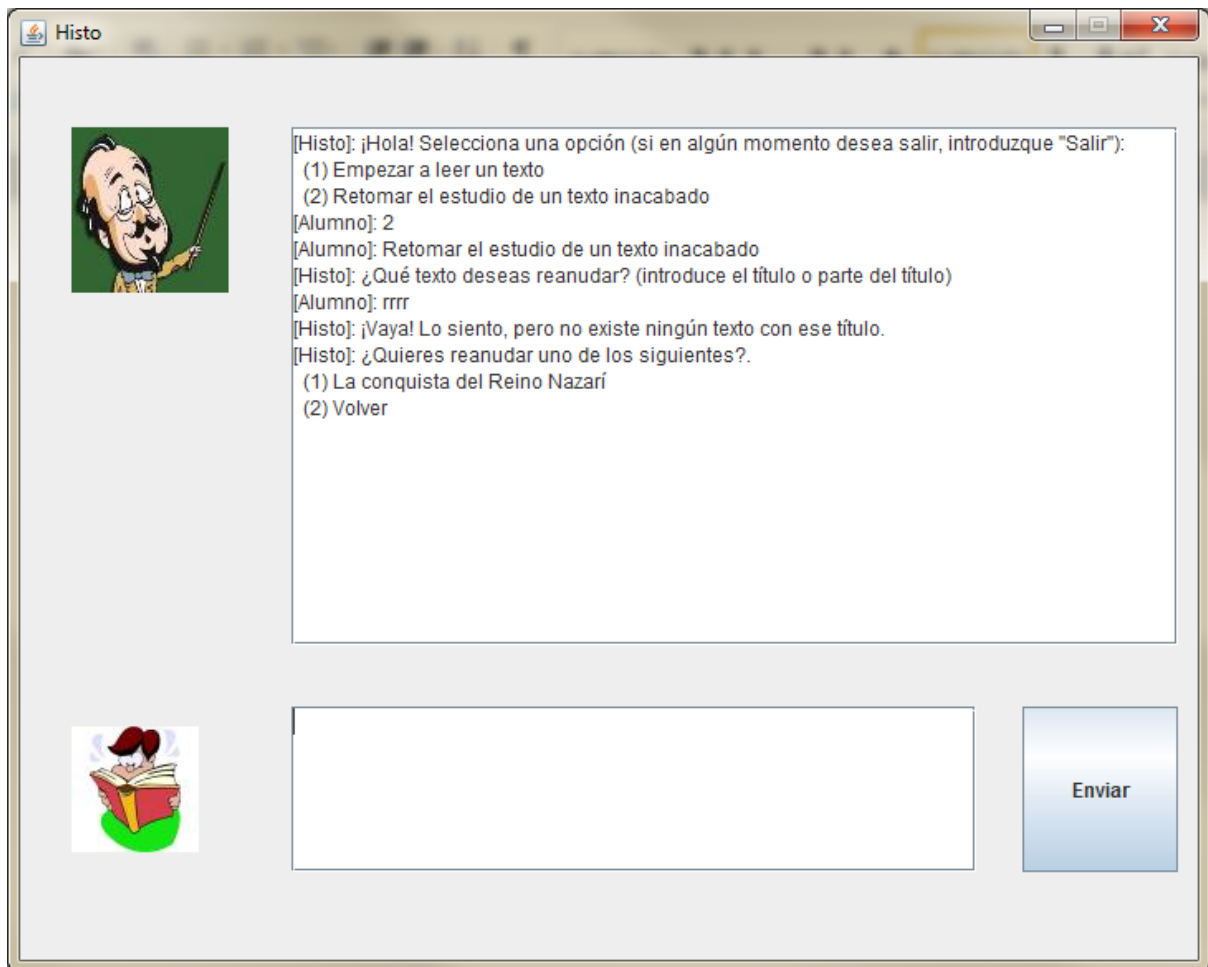


Figura 28: Histo modo normal intentando reanudar texto no existente

ALGORITMO HISTO VENTANA

En este se controla la fase en la que se encuentra el alumno, determinando cual o cuales son las opciones que puede seleccionar y las tareas que puede realizar. Las diferentes fases son las siguientes:

- FASE 0: Comienzo. Opciones:
 - Empezar
 - Retomar
- FASE 1: Empezar un texto. Opciones:
 - Se muestra el texto
 - Responder a las preguntas
- FASE 2: Retomar: Opciones:
 - Introduce parte del título. Si coincide:
 - Mostrar coincidencias y seleccionar una
 - Si no
 - Mostrar todas las posibilidades
- FASE 3: Retomar un texto seleccionado: Opciones:
 - Se muestra el texto
 - Responder a las preguntas
- FASE 4: Reinicio.
 - Igual que el comienzo

El texto que se muestra en la ventana de Histo al iniciar el alumno es el siguiente:
 [Histo]: ¡Hola! Selecciona una opción (si en algún momento desea salir, introduzque "Salir")
 (1) Empezar a leer un texto
 (2) Retomar el estudio de un texto inacabado

```

funcion gestionFaseHisto( Cadena textoIntroducido )
si fase es igual a 0 entonces
  si textoIntroducido es igual a 1 entonces
    escribirLinea("[Alumno]: " + textoIntroducido );
    escribirLinea("[Alumno]: Empezar a leer un texto " );
    fase = 1;
    mostrarTexto();
  sino si textoIntroducido es igual a 2 entonces
    escribirLinea("[Alumno]: " + textoIntroducido );
    escribirLinea("[Alumno]: Retomar el estudio de un texto
      inacabado");
    escribirLinea("[Histo]: ¿Qué texto deseas reanudar? (introduce el
      título o parte del título) " );
    fase = 2;
  sino entonces
    escribirLinea("[Alumno]: " + textoIntroducido );
    escribirLinea("[Histo]: Esa opción no es válida. Introduce una de
      las siguientes opciones ");
    escribirLinea(" (1) Empezar a leer un texto ");
    escribirLinea(" (2) Retomar el estudio de un texto inacabado ");
  fin si
sino si fase es igual a 1 entonces
  si textoIntroducido es igual a 1 entonces
    escribirLinea("[Alumno]: " + textoIntroducido );
    escribirLinea("[Alumno]: Responder a las preguntas relacionadas
      con el texto " );
    empezarPreguntas();
  sino si textoIntroducido es igual a 2 entonces
    escribirLinea("[Alumno]: " + textoIntroducido );
    escribirLinea("[Alumno]: Retomar el estudio de un texto inacabado
      " );
    escribirLinea("[Histo]: ¿Qué texto deseas reanudar? (introduce el
      título o parte del título) " );
    fase = 2;
  sino entonces
    escribirLinea("[Alumno]: " + textoIntroducido );
    escribirLinea("[Histo]: Esa opción no es válida. Introduce una de
      las siguientes opciones ");
    escribirLinea(" (1) Empezar a leer un texto ");
    escribirLinea(" (2) Retomar el estudio de un texto inacabado ");
  fin si
sino si fase es igual a 2 entonces
  listaTitulos = obtenerTitulosParecidos( textoIntroducido );
  si NO listaTitulos.esVacia() entonces
    si listaTitulos.tamaño() es igual a 1 entonces
      fase = 1;
      mostrarTexto( listaTitulos.obtener(0) );
    sino entonces

```

```

    escribirLinea("[Alumno]: " + textoIntroducido );
    escribirLinea("[Histo]: Hay " + listaTitulos.tamaño() + " textos
        similares al introducido. ");
    escribirLinea("[Histo]: ¿Cuál deseas reanudar?. ");
para Entero i desde 0 hasta listaTitulos.tamaño() incrementando
    por 1 hacer
        escribirLinea(" (" + (i + 1) + ") " + listaTitulos.obtener(i)
            );
    fin para
    escribirLinea(" (" + (listaTitulos.tamaño() + 1) + ") Volver
        ");
    fase = 3;
fin si
sino
    Lista listaTitulos = obtenerTextosParaReanudar;
    escribirLinea("[Alumno]: " + textoIntroducido );
    si NO listaTitulos.esVacia() entonces
        escribirLinea("[Histo]: ¡Vaya! Lo siento, pero no existe ningún
            texto con ese título. ");
        escribirLinea("[Histo]: ¿Quieres reanudar uno de los
            siguientes?. ");
        para Entero i desde 0 hasta listaTitulos.tamaño() incrementando
        por 1 hacer
            escribirLinea(" (" + (i + 1) + ") " + listaTitulos.obtener(i)
                );
        fin para
        escribirLinea(" (" + (listaTitulos.size() + 1) + ") Volver ");
        fase = 3;
        escribirLinea("[Histo]: ¡Vaya! Lo siento, pero no existe ningún
            texto para reanudar. ");
        escribirLinea(" (1) Empezar a leer un texto nuevo ");
        fase = 4;
    fin si
fin si
sino si fase es igual a 3 entonces
    escribirLinea("[Alumno]: " + introducido );
    si textoIntroducido es mayor que 0 AND numTitulo es menor o igual
    que listaTitulos.size() entonces
        mostrarTexto( listaTitulos.obtener( textoIntroducido - 1 ) );
    sino entonces
        escribirLinea("[Histo]: Esa opción no es válida. Introduce una de
            las siguientes opciones ");
        para Entero i desde 0 hasta listaTitulos.tamaño() incrementando por
        1 hacer
            escribirLinea(" (" + (i + 1) + ") " + listaTitulos.get(i) );
        fin para
        escribirLinea(" (" + (listaTitulos.size() + 1) + ") Volver ");
    sino si fase es igual a 4 entonces
        si textoIntroducido es igual a 1 entonces
            escribirLinea("[Alumno]: " + textoIntroducido );
            escribirLinea("[Alumno]: Empezar a leer un text ");
            fase = 1;
            mostrarTexto();

```

```

sino entonces
    escribirLinea("[Alumno]: " + textoIntroducido );
    escribirLinea("[Histo]: Esa opción no es válida. Introduce una de
        las siguientes opciones ");
    escribirLinea(" (1) Empezar a leer un texto ");
fin si
fin si
fin función

```

ALGORITMO OBTENER SIGUIENTE TEXTO

```

función escogerSiguienteTexto
siguienteTexto = ejecutarSentencia(
    " SELECT"
    + "      T.titulo,  T.text,  T.nombre_prof,  T.nivel,  T.validate,
      T.n_preguntas,  T.preg_validar,  E.num_usos,  E.titulo  AS
      tituloEstadisticas"
    + " FROM "
    + "      texto T"
    + "      LEFT JOIN estadisticas_alumno E ON ( E.titulo = T.titulo AND
      E.nombre_prof = T.nombre_prof AND ( E.nombre_alumno = " +
      nombreAlumno + " OR E.nombre_alumno IS NULL ) )"
    + " WHERE "
    + "      T.nivel = " + nivelAlumno
    + "      AND T.validate = 1 "
    + " ORDER BY num_usos, titulo "
    + " LIMIT 1 ");
);
si siguienteTexto.noUsado() entonces
    insertar(
        " INSERT INTO estadisticas_alumno "
        + "      ( nombre_alumno , titulo , nombre_prof , num_usos )"
        + " VALUES"
        + "      ( " + nombreAlumno + " , " + tituloTexto + " , " +
      nombreProfesor + " , 0 );
    );
fin si
fin función

```

Tras haber leído el texto, se podrá empezar con las preguntas.

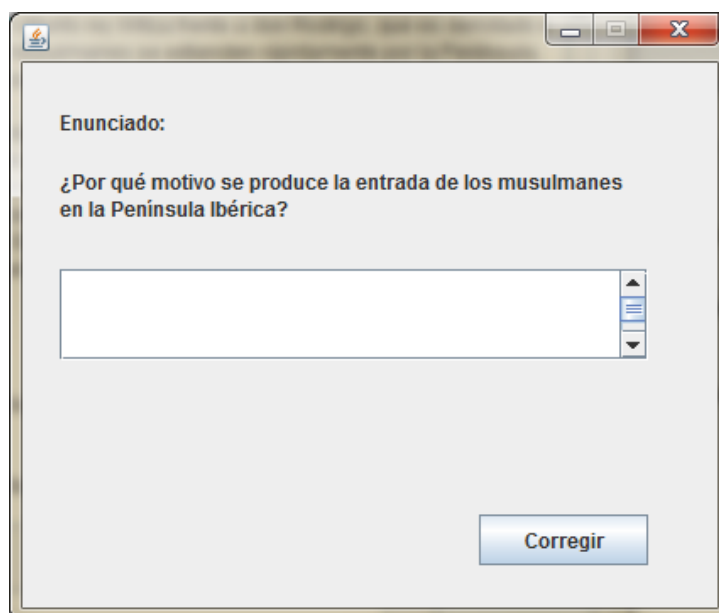
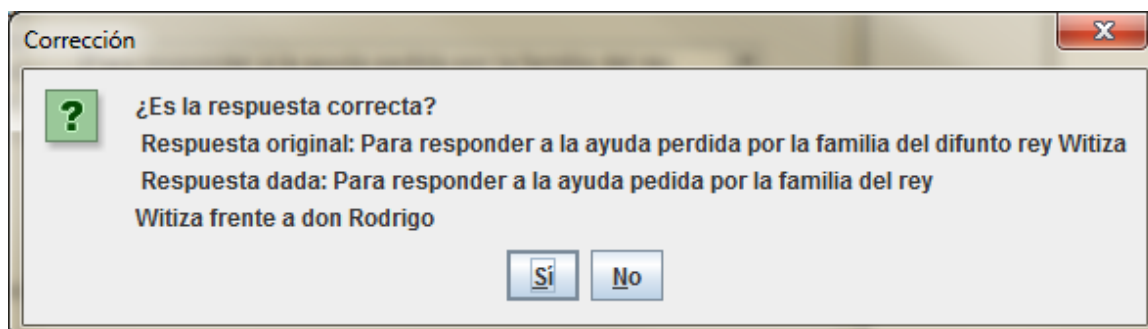


Figura 29: Pregunta corta alumno

Tras introducir una respuestas y pulsar sobre corregir, en el caso de las preguntas cortas, aparece una ventana nueva, con nuestra respuesta, y la correcta, para que sean comparadas, y el propio alumno será el que decida si su respuesta ha sido acertada o no.



Se contabilizarán las preguntas acertadas, que serán necesarias para decidir si el alumno ha aprobado el texto o no.

Tras corregir la pregunta, el texto del botón corregir cambiará, para poder pasar a la siguiente pregunta.

Figura 30: Autocorrección pregunta corta

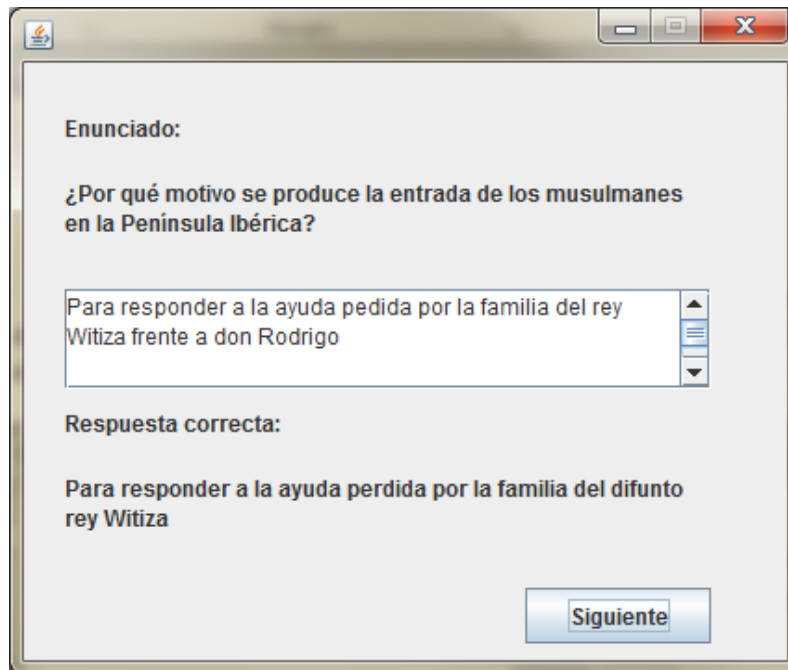


Figura 31: Pregunta corta siguiente

En el caso de las preguntas de tipo palabra, bastará con insertar la palabra, nombre o fecha que se pida.

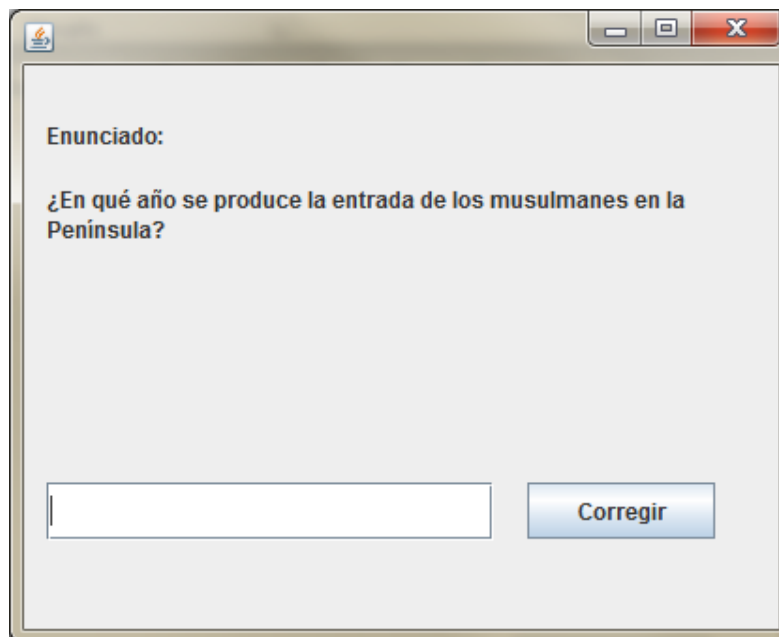


Figura 32: Pregunta de tipo palabra alumno

En este caso si se lleva a cabo la corrección automática de la respuesta introducida por el alumno, contabilizando de nuevo, si ha sido un acierto.

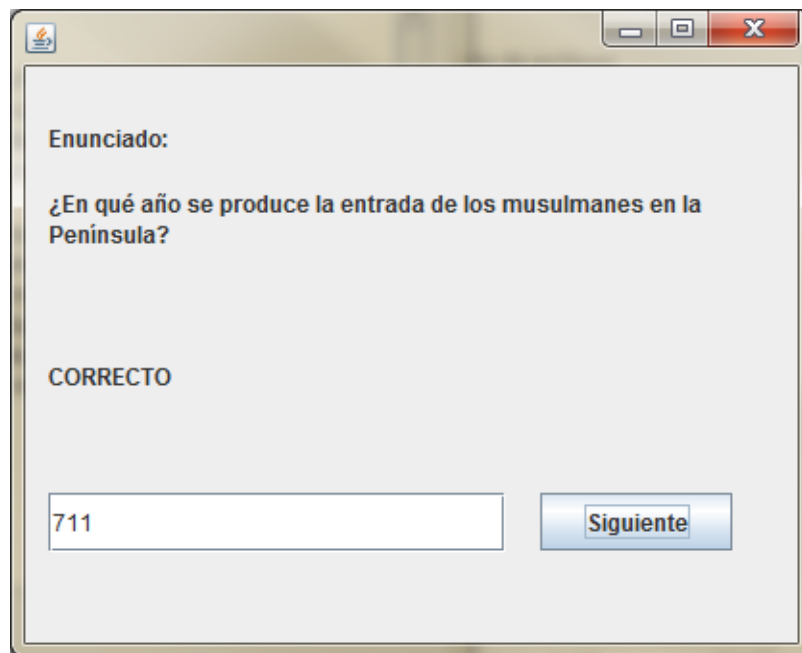


Figura 33: Pregunta tipo palabra corregida

Lo mismo ocurrirá con las preguntas tipo test, que también se corregirán de manera automática.

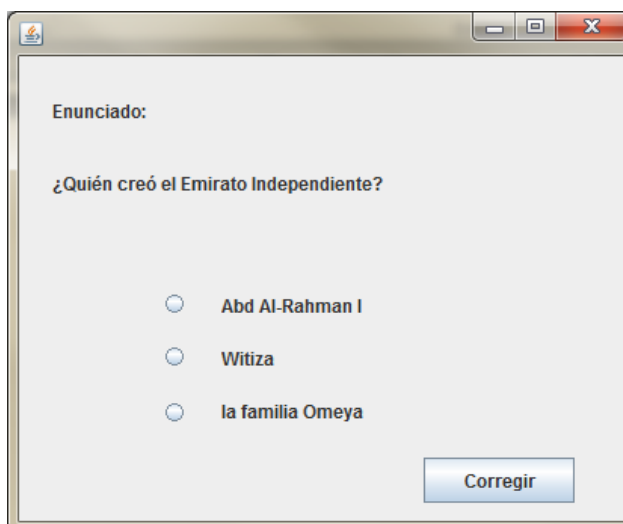


Figura 35: Pregunta tipo test alumno

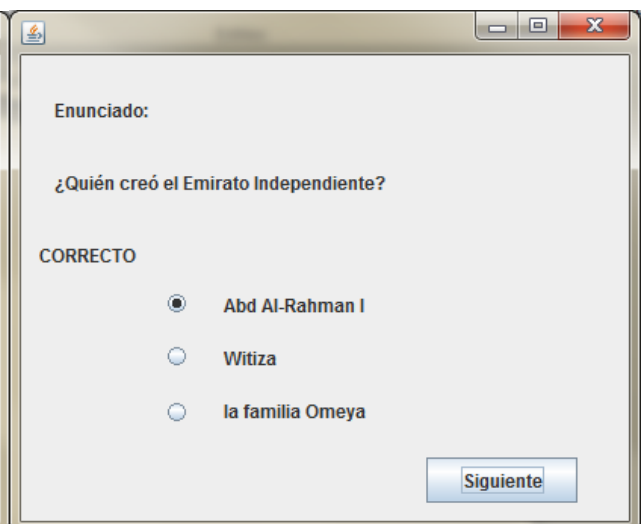


Figura 34: Pregunta tipo test alumno corregida

ALGORITMO PARA SUBIR DE NIVEL

Este código se ejecuta cada vez que el alumno finaliza un texto con todas sus preguntas:

```
función gestionarNivel(Entero nivel, Cadena nombreAlumno)
devuelve Entero hacer

    Entero numTextosNivel = obtenerNumTextosNivel(nivel);
    Lista listaTextosFinalizados = obtenerListaTextosFinalizados(nombreALumno);
    Entero numTextosAprobados = 0;

    si (numTextosNivel es igual listaTextosFinalizados.tamaño()) entonces

        para cada texto de listaTextosFinalizados hacer
            si (texto.obtenerNumPreguntasAcertadas() es mayor o
                igual texto.obtenerNumPreguntasNecesarias()) entonces
                numTextosAprobados = numTextosAprobados + 1
            fin para

        si (numTextosAprobados es mayor o igual numTextosNivel * 0.5)
            entonces
                si nivel es igual 4 entonces
                    devuelve 3; // Significa que subiría de nivel, pero que
                    ya se encuentra en el nivel máximo.
                sino
                    devuelve 1; // Significa que sube de nivel.
                fin si
            sino
                si nivel es igual 1 entonces
                    devuelve 4; // Significa que bajaría de nivel, pero que
                    ya se encuentra en el nivel mínimo.
                sino
                    devuelve 2; // Significa que baja de nivel.
                fin si
            fin si
        sino
            devuelve 0; // Significa que no hay subida ni bajada de nivel
        fin si

fin función
```

Dependiendo del valor devuelto por la función, se mostrará un mensaje u otro en histoVentana y se procederá posteriormente a mostrar los textos del nivel correspondiente.

4. PRUEBAS

4.1. PRUEBAS UNITARIAS

Consiste en realizar las pruebas del código por módulos, comprobando que cada uno de ellos cumple con la funcionalidad que le es propia.

4.2. PRUEBAS DE CAJA BLANCA

En este tipo de pruebas de software se realiza sobre las funciones internas de cada módulo. Se recorren todos los caminos de ejecución posibles. Para realizar las pruebas de caja blanca, escogeremos el módulo del alumno, y para ello, a continuación, se desglosarán sus opciones en bloques, que de forma genérica, representarán las funciones a realizar según las elecciones llevadas a cabo por el alumno.

Bloque 1

El alumno introduce en la pantalla de acceso su nombre de usuario y su contraseña, que solo le permitirán el acceso a la aplicación en caso de que ambos sean correctos y concuerden con los datos almacenados en la base de datos. Si pulsa cerrar, saldrá de la aplicación.

Bloque 2

El usuario tendrá dos opciones, que serán modo aleatorio, y modo normal, y se corresponderán con los bloques 3 y 6 respectivamente.

Bloque 3

En la ventana de histo, le aparecerá el primer texto, que tendrá que leer antes de empezar a contestar las preguntas. Tendrá tres opciones, contestar las preguntas (que se corresponderá con el bloque 4), volver (bloque 2) o salir (bloque 1), opción con la que terminará el modo aleatorio.

Bloque 4

Aparecerá el enunciado de una pregunta, tras contestarla, podrá pulsar sobre corregir.

Bloque 5

Tras haber sido corregida la pregunta, pulsará siguiente. Si hay más preguntas volverá al bloque 4, en caso contrario, volvería al bloque 3.

Bloque 6

El usuario ha escogido el modo normal. En este caso tendrá dos opciones, empezar un nuevo texto (volviendo al bloque 3) o reanudar uno ya empezado (bloque 7).

Bloque 7

Para reanudar un texto que ya haya empezado tendrá que introducir una palabra que esté contenida en el título del texto. Se presentan tres opciones, que no encuentre ningún texto con esa palabra en el título (bloque 8), que encuentre un texto con esa palabra en el título, en cuyo caso se dirigiría de nuevo al bloque 3, y por último que encuentre varios textos con esa palabra, que también se dirigiría al bloque 9.

Bloque 8

Mostrará todos los títulos de textos que hayan sido empezados por el alumno, pero que se hayan dejado inacabados, en caso de haber dejado alguno empezado. El alumno elegirá uno de ellos, que le llevará al bloque 3 para mostrar el texto elegido. Si no hubiese textos para reanudar, mostrará la opción de empezar un texto nuevo (bloque 3).

Bloque 9

Mostrará todos los textos cuyo título contenga la palabra introducida por el alumno. Tendrá que elegir uno de ellos, volviendo al bloque 3, que sacará el texto seleccionado, y no uno al azar de la lista.

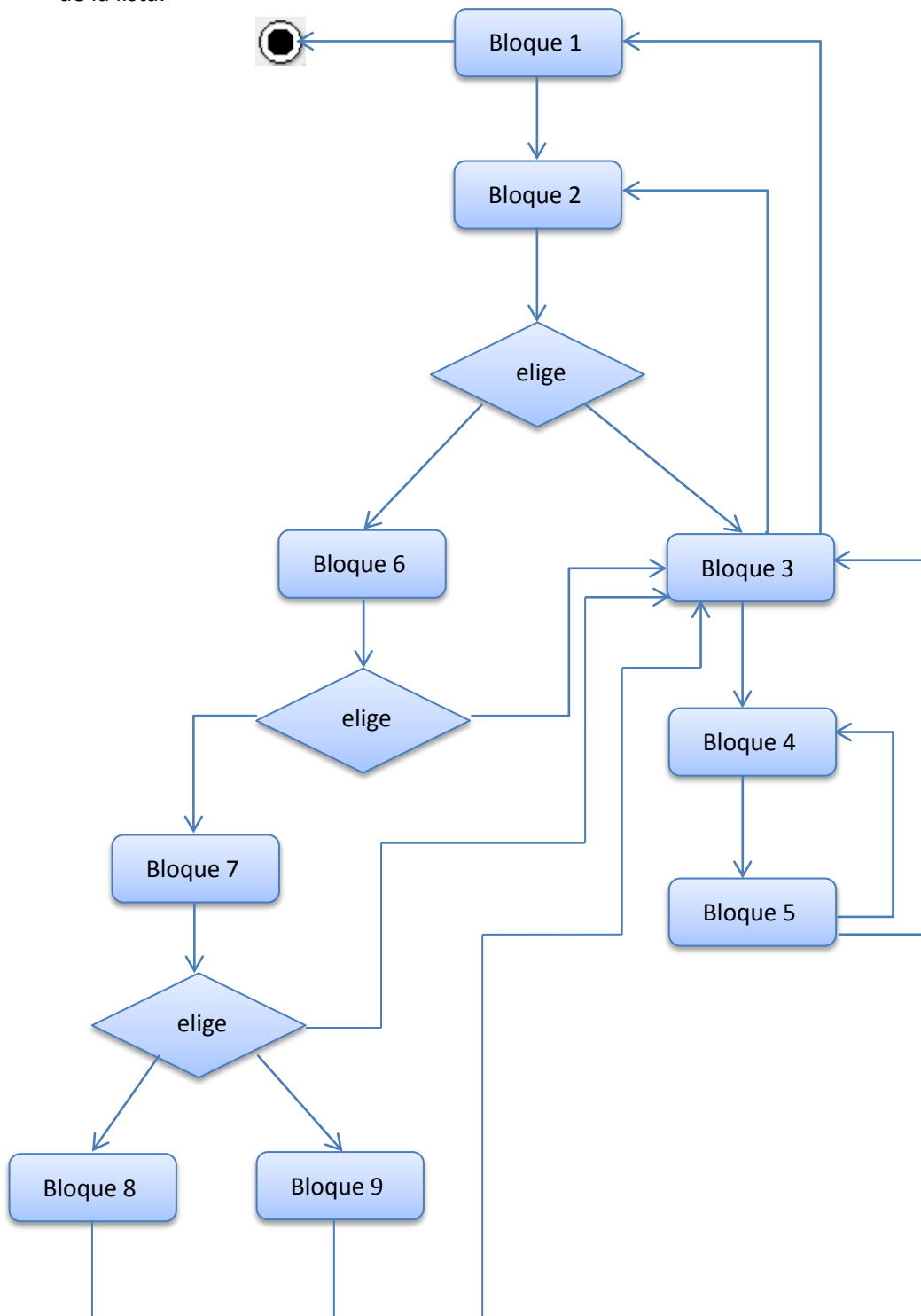


Figura 36: Diagrama pruebas de caja blanca

4.3. PRUEBAS DE CAJA NEGRA

En un tipo de pruebas que mediante su interfaz gráfica, se basará en las entradas recibidas, y comparando las salidas obtenidas con las deseadas, determinará si se cumplen los objetivos marcados, concordando con su funcionalidad. Utilizaremos una serie de casos límite para comprobar que cumple con los requisitos.

Prueba 1 – Acceso de usuario a la aplicación

Clases java	Entrada	Descripción	Salida esperada	Salida obtenida
Acceso.java ConexionBD.java Histo.java	Nombre o contraseña erróneos	Se intentará validar un usuario cuyo nombre o contraseña no concordará con lo insertado en la base de datos o no existirá.	Aparece un mensaje en la ventana de inicio: "Error de autenticación"	Aparece un mensaje en la ventana de inicio: "Error de autenticación"

Tabla 4: Prueba 1 - Acceso de usuario a la aplicación

Prueba 2 – Inserción de texto con título ya existente

Clases java	Entrada	Descripción	Salida esperada	Salida obtenida
NuevoTexto.java ConexionBD.java Histo.java	Se introduce un título ya existente en la base de datos	Al pulsar sobre el botón editar preguntas, se intenta guardar el texto con un título que ya está insertado en la base de datos	Mensaje de error en el label que informa de que con ese título ya existe un texto en la base de datos	java.sql.SQLException: Cannot delete or update a parent row: a foreign key constraint fails (`histo`.`tiene`, CONSTRAINT `fk_tiene_texto` FOREIGN KEY (`titulo`, `nombre`) REFERENCES `texto` (`titulo`, `nombre_prof`))

Tabla 5: Prueba 2 – Inserción de texto con título ya existente

El error se produce porque para comparar el título insertado en el campo de texto, se recuperan todos los textos insertados, cuyo autor es el profesor que va a insertar el nuevo texto. En este caso, el texto existía, pero había sido insertado por otro profesor, por lo que habilitaba el botón Editar Preguntas, pero luego se producía un error al tratar de insertarlo.

Se corrige comprobando cuándo es necesario recuperar todos los textos, y cuándo solo se recuperan los insertados por éste profesor.

Prueba 3 – Apertura de archivo no válido desde el explorador

Clases java	Entrada	Descripción	Salida esperada	Salida obtenida
ExploradorArchivos.java	Se selecciona un archivo no válido (.sql)	La aplicación solo permite el uso de .txt, y la apertura de .doc, .pdf...	Mensaje de error indicando que el archivo no puede abrirse	Mensaje de error indicando que el archivo no puede abrirse

Tabla 6: Prueba 3 – Apertura de archivo no válido desde el explorador

Prueba 4 – Texto sin número de preguntas para aprobar

Clases java	Entrada	Descripción	Salida esperada	Salida obtenida
NuevoTexto.java EditorTexto.java ConexioBD.java Histo.java	Se pone a 0 o no se rellena el campo de preguntas para aprobar	Este campo contiene el número de preguntas que el alumno tendrá que acertar para que se considere que ha aprobado el texto	Mensaje de error en el label indicando que hay que introducir un valor en el campo y botón Editar Preguntas deshabilitado	Se habilita el botón cuando se le introduce un 0 a pesar de no ser un valor válido

Tabla 7: Prueba 4 – Texto sin número de preguntas para aprobar

Tras corregirlo, incluyendo una comprobación para que el campo sea distinto de 0, aparece el mensaje de error en el label y queda deshabilitado el botón Editar Preguntas.

Prueba 5 – Opción no válida para Histo

Salida obtenida	Clases java	Entrada	Descripción	Salida esperada
Aviso de que esa opción no es válida y repetición del menú	HistoVentana.java ConexionBD.java Histo.java	Opción no válida	Se introduce una opción no contemplada por la aplicación	Aviso de que esa opción no es válida y repetición del menú

Tabla 8: Prueba 5 – Opción no válida para Histo

4.4. PRUEBAS DE INTEGRACIÓN

A lo largo del desarrollo del proyecto, se han ido realizando las pruebas unitarias, que comprueban el correcto funcionamiento de cada módulo. También durante este proceso se han ido realizando pruebas de integración, combinando los distintos módulos para asegurar que funcionan correctamente con todas sus dependencias.

En primer lugar, se desarrolló Acceso.java, que es la clase java cuya interfaz gráfica pide un usuario y contraseña que darán acceso a la aplicación, y después, el módulo de gestión.

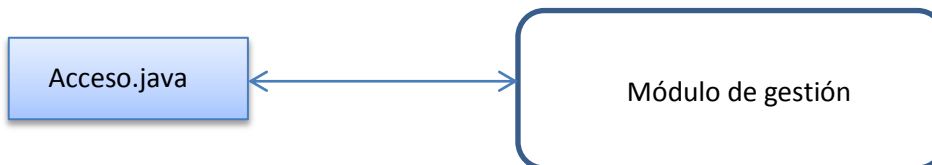


Figura 37: Pruebas de integración 1

Se comprueba que dado un usuario y una contraseña, las clases del módulo de gestión, detalladas en la Figura 4, realizan la búsqueda en la base de datos y devuelven si el usuario es profesor o alumno, o si no ha sido encontrado.

A continuación, una vez creado el módulo del profesor, habrá que comprobar que al combinarse con el módulo de gestión, se obtienen los resultados esperados, como por ejemplo, que inserte correctamente los textos en la base de datos, que recupere los títulos, que pase de la ventana del editor a las de preguntas según se va solicitando, etc.



Figura 38: Pruebas de integración 2

Lo mismo ocurre con el módulo del alumno. Habrá que probar que recupera los datos de la base de datos, y actualiza los cambios, y número de preguntas que va acertando a lo largo de la sesión. También se encarga el módulo de gestión, de ir mostrando las ventanas correspondientes, y de corregir las respuestas del alumno y contabilizar los aciertos.



Figura 39: Pruebas de integración 4

Una vez comprobado que los módulos del alumno y el profesor funcionan correctamente con el módulo de gestión, habría que probar que funcionan todos juntos, es decir, que los datos que inserta el profesor en la base de datos son los que se recuperan para el alumno.

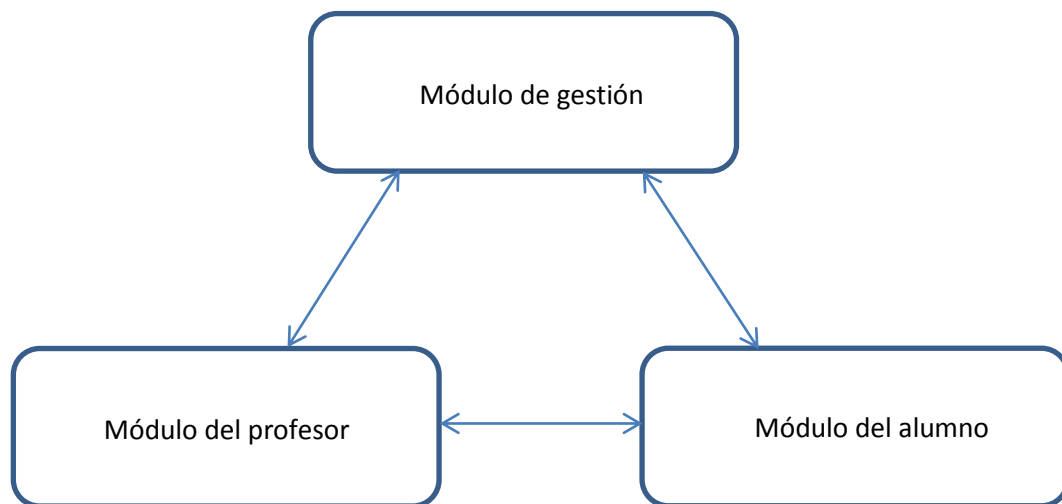


Figura 40: Pruebas de integración 5

4.5. PRUEBAS DE VALIDACIÓN

Las pruebas de validación se realizan para comprobar que se cumplen los requisitos que se definieron al inicio del proyecto (apartado 3.1.1).

1. *En el sistema habrá dos tipos diferentes de usuarios, con diferentes interfaces gráficas, adaptadas a sus opciones*

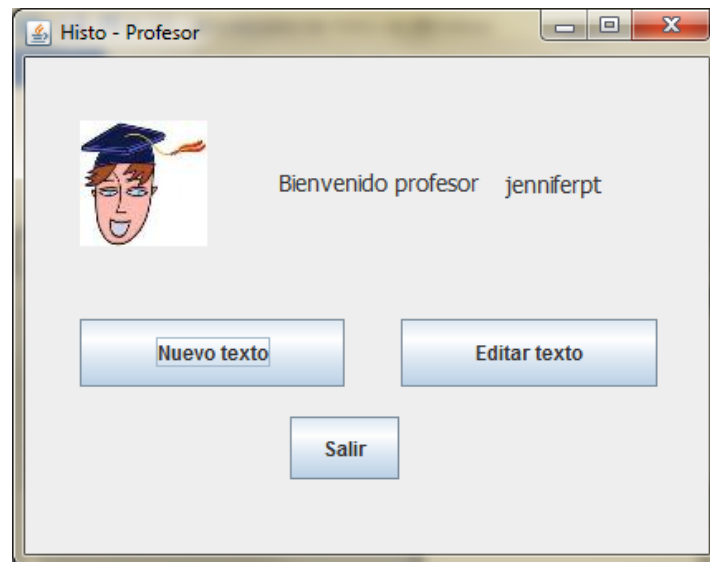


Figura 41: Pruebas de validación – Inicio profesor



Figura 42: Pruebas de validación – Inicio alumno

Las Figuras 41 y 42 se corresponden con el inicio del profesor y el del alumno respectivamente. Mostrando que las opciones para uno y otro usuario son totalmente diferentes.

2. Todo texto tendrá un nivel de dificultad, seleccionado por el profesor, para organizar el orden en el que repasará los textos el alumno

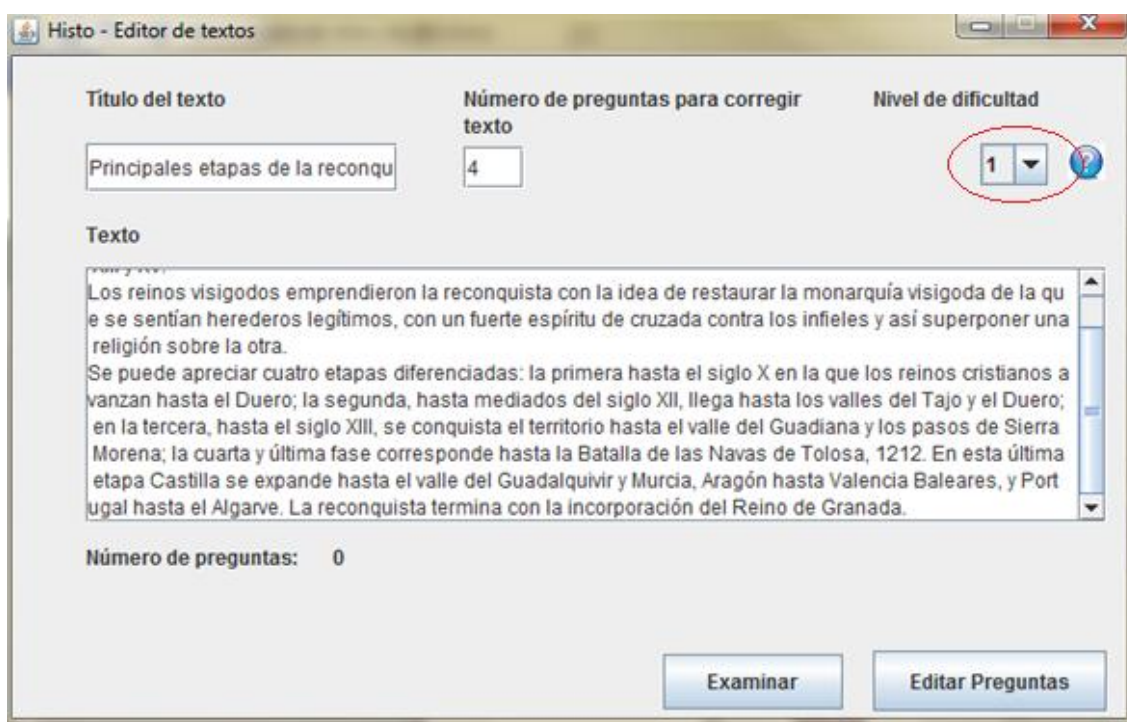


Figura 43: Pruebas de validación – Texto nuevo

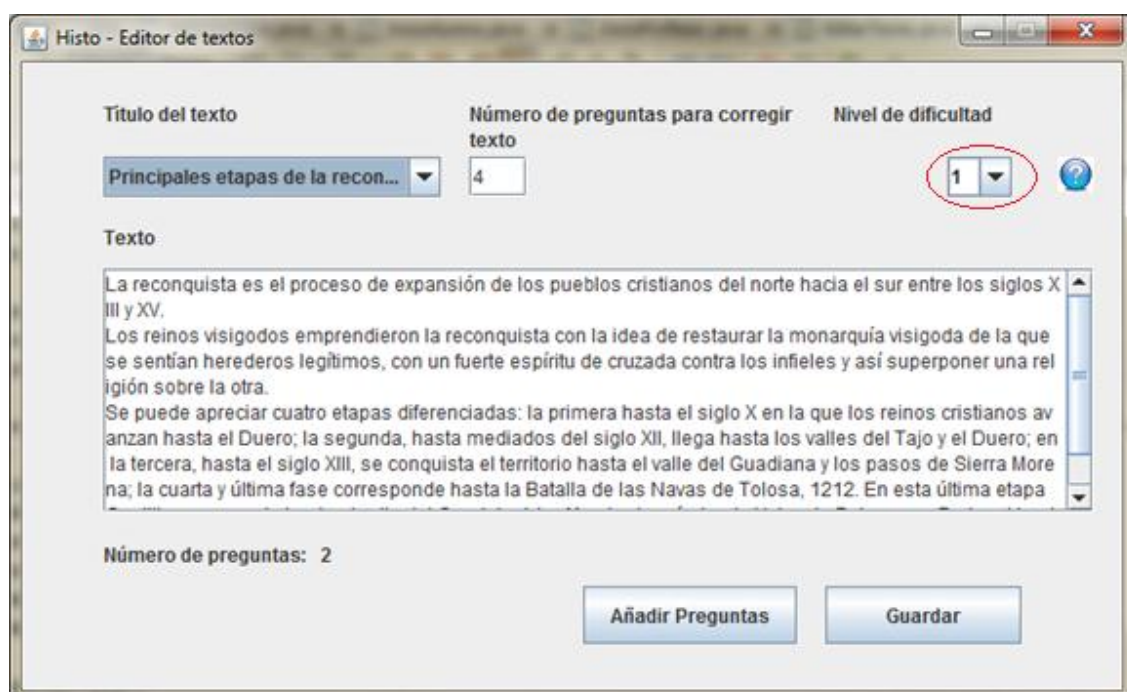


Figura 44: Pruebas de validación – Editor de textos

En las Figuras 43 y 44 se observa como, tanto en los nuevos textos, como en los editados, el texto siempre se guarda con un entero que representará el nivel. Habrá 4 niveles, 1 será el más fácil, y 4 el más difícil. Gracias al combo, se garantiza que el texto siempre tendrá un nivel.

3. Todos los textos tendrán también un número de preguntas, introducido por el profesor, que indicará el número de preguntas del texto, que el alumno tendrá que contestar correctamente, para que se pueda considerar que dicho texto se ha finalizado con éxito

Histo - Editor de textos

Título del texto: Principales etapas de la reconqu...

Número de preguntas para corregir texto: 4

Nivel de dificultad: 1

Texto:

Los reinos visigodos emprendieron la reconquista con la idea de restaurar la monarquía visigoda de la que se sentían herederos legítimos, con un fuerte espíritu de cruzada contra los infieles y así superponer una religión sobre la otra.

Se puede apreciar cuatro etapas diferenciadas: la primera hasta el siglo X en la que los reinos cristianos avanzan hasta el Duero; la segunda, hasta mediados del siglo XII, llega hasta los valles del Tajo y el Duero; en la tercera, hasta el siglo XIII, se conquista el territorio hasta el valle del Guadiana y los pasos de Sierra Morena; la cuarta y última fase corresponde hasta la Batalla de las Navas de Tolosa, 1212. En esta última etapa Castilla se expande hasta el valle del Guadalquivir y Murcia, Aragón hasta Valencia Baleares, y Portugal hasta el Algarve. La reconquista termina con la incorporación del Reino de Granada.

Número de preguntas: 0

Examinar Editar Preguntas

Figura 45: Pruebas de validación – Nuevo texto preguntas para corregir

Histo - Editor de textos

Título del texto: Principales etapas de la recon...

Número de preguntas para corregir texto: 4

Nivel de dificultad: 1

Texto:

La reconquista es el proceso de expansión de los pueblos cristianos del norte hacia el sur entre los siglos X III y XV.

Los reinos visigodos emprendieron la reconquista con la idea de restaurar la monarquía visigoda de la que se sentían herederos legítimos, con un fuerte espíritu de cruzada contra los infieles y así superponer una religión sobre la otra.

Se puede apreciar cuatro etapas diferenciadas: la primera hasta el siglo X en la que los reinos cristianos avanzan hasta el Duero; la segunda, hasta mediados del siglo XII, llega hasta los valles del Tajo y el Duero; en la tercera, hasta el siglo XIII, se conquista el territorio hasta el valle del Guadiana y los pasos de Sierra Morena; la cuarta y última fase corresponde hasta la Batalla de las Navas de Tolosa, 1212. En esta última etapa

Número de preguntas: 2

Añadir Preguntas Guardar

Figura 46: Pruebas de validación – Editor de textos preguntas para validar

Como puede observarse en las Figuras 45 y 46 siempre se requiere un número de preguntas, que será el mínimo de preguntas que el alumno tendrá que acertar para que el texto se considere aprobado.

4. Con cada texto se introducirán mínimo cinco preguntas, que podrán ser de tres tipos diferentes: pregunta de respuesta corta, pregunta tipo test, o pregunta de respuesta de palabra

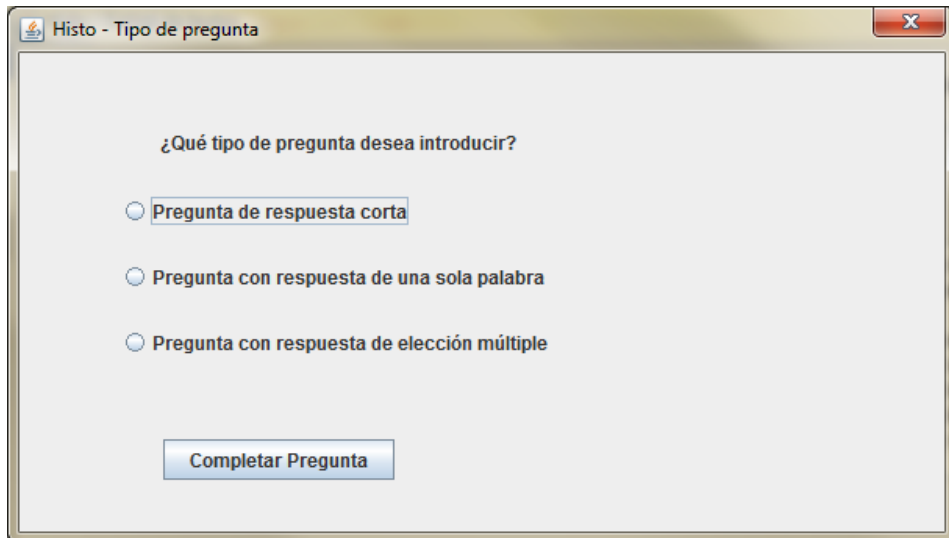


Figura 47: Pruebas de validación – Tipos de preguntas

En la Figura 47 se muestran los tres tipos de preguntas que puede escoger el profesor para insertar.

5. El alumno tendrá la posibilidad de empezar la sesión de estudio de dos formas: en modo normal o en modo aleatorio.



Figura 48: Pruebas de validación – Inicio alumno dos modos

Como se puede ver en la Figura 48, se le ofrecen dos modos de ejecución al alumno, mediante dos botones. Modo normal y modo aleatorio.

6. *Todos los textos que se le mostrarán al alumno serán de su mismo nivel, de manera que no se le presenten textos para los que no está preparado aún, ni realizar textos que puedan resultarle aburridos por estar familiarizado con ellos.*

Una de las condiciones en la selección de la lista de textos es que el nivel del texto coincida con el nivel de estudio del alumno. De este modo se garantiza que el alumno estará siempre realizando textos acordes a su nivel. En caso de que no hubiese textos disponibles para el nivel actual del alumno, aparecería un mensaje mencionando que aún no hay textos para ese nivel.

7. *Guardado automático tras finalizar cada pregunta, para evitar pérdidas.*

Los datos se van actualizando en la tabla estadísticas_alumno cada vez que una pregunta es corregida, evitando así pérdidas de información, y garantizando, que aunque la aplicación se detenga sin haber finalizado un texto, éste, quedará señalado como texto iniciado para que pueda ser realizado escogiendo la opción de reanudar texto.

8. *Cerrar el sistema pulsando el botón cerrar o tecleando “salir”.*

Siempre que se pulse el botón cerrar de la ventana histo, o se teclee “salir”, se volverá a la pantalla de inicio, para poder cambiar de modo. Si se pulsa el botón cerrar de nuevo en esta ventana, la aplicación finalizará su ejecución y para volver a usarla habrá que introducir de nuevo un usuario y una contraseña.

4.6. PRUEBAS DE ACEPTACIÓN

Este tipo de prueba se realiza con usuarios que puedan valorar la aplicación.

Para ello, se ha escogido a un grupo de ocho alumnos de 2º de bachillerato de tecnología y ciencias, que han tenido en este curso la asignatura de historia como asignatura obligatoria.

Para la realización de las pruebas se utilizan como textos los epígrafes cortos obtenidos de sus propios libros y apuntes que están utilizando para prepararse para las pruebas de acceso universitario. Fueron introducidos cuatro textos de nivel 1, cuatro de nivel 2, tres para el nivel 3, y otros tres para el nivel 4. De cada uno de ellos se introdujeron seis o siete preguntas, para garantizar que el mínimo de preguntas por texto quedaba cubierto. Éstas preguntas fueron extraídas y redactadas tras la lectura de cada uno de los textos. Los textos empleados para las pruebas se adjuntas en los anexos I, II, III y IV.

En primer lugar se les pidió que realizaran una serie de textos en papel. Exactamente cuatro textos por alumno, uno por cada nivel de dificultad. Estos textos estaban incluidos en la base de datos, por lo que podrían volver a salir durante el uso de la aplicación.

La Figura 49 muestra a los alumnos realizando las pruebas en papel, mientras tanto, uno de ellos iba pasando por el ordenador para probar Histo.



Figura 49: Alumnos realizando pruebas en papel

Mediante la Tabla 9 observaremos los resultados obtenidos en las pruebas realizadas en papel. Cada alumno ha realizado un texto de cada nivel, y en la Tabla 9 se mostrarán, para cada texto, las preguntas acertadas y si el texto está aprobado o no. Para aprobar un texto, habrá que acertar un mínimo de preguntas, que se le solicita al profesor cuando introduce el texto. Varía en función del nivel de dificultad. En la Tabla 9, el número de preguntas que contiene el texto se muestra al lado de preguntas, entre paréntesis, al igual que el número mínimo de preguntas que hay que contestar correctamente para aprobar el texto, que se muestra al lado de "texto" entre paréntesis.

USUARIOS	NIVEL 1		NIVEL 2		NIVEL 3		NIVEL 4	
	Aciertos (7)	Texto (5)	Aciertos (7)	Texto (5)	Aciertos (6)	Texto (6)	Aciertos (6)	Texto (6)
ALUMNO1	6	APROBADO	4	SUSPENSO	6	APROBADO	4	SUSPENSO
ALUMNO2	5	APROBADO	4	SUSPENSO	4	SUSPENSO	6	APROBADO
ALUMNO3	6	APROBADO	7	APROBADO	6	APROBADO	6	APROBADO
ALUMNO4	5	APROBADO	5	APROBADO	6	APROBADO	5	SUSPENSO
ALUMNO5	7	APROBADO	4	SUSPENSO	5	SUSPENSO	6	APROBADO
ALUMNO6	6	APROBADO	5	APROBADO	6	APROBADO	5	SUSPENSO
ALUMNO7	5	APROBADO	5	APROBADO	6	APROBADO	6	APROBADO
ALUMNO8	5	APROBADO	5	APROBADO	5	SUSPENSO	6	APROBADO
PORCENTAJE APROBADOS	100%		62.5%		62.5%		62.5%	

Tabla 9: Resultados pruebas de aceptación en papel

Se observa en la Tabla 9 que el texto de nivel 1 les ha resultado sencillo, y ha sido aprobado por todos, mientras que en los de los niveles sucesivos, se pueden ver resultados algo más dispares por la dificultad.

Una vez realizados los textos en papel, se les sitúa en el ordenador para realizar la prueba con la aplicación. El tiempo de uso será de unos diez minutos, tiempo suficiente para la lectura de un texto y la realización de las preguntas.

A cada estudiante se le proporciona un usuario y una contraseña (alumnoX, alumnoX) para que puedan realizar la prueba. En algunos casos, el nivel de inicio será el 1, mientras que en otros, el nivel de estudio del alumno se habrá incrementado para poder probar varios niveles de dificultad.

Al inicio de cada tiempo de uso para los alumnos, se les da una breve explicación del funcionamiento de la aplicación.

En general, durante el uso de la aplicación, no surgieron muchas dudas acerca del uso, por lo que podemos deducir que les resultó sencillo o fácil de entender.

La Figura 50 muestra una alumna realizando las pruebas en Histo en un ordenador portátil con la aplicación instalada. La sala estaba en silencio para evitar distracciones y ayudar a la concentración de los alumnos. La alumna está leyendo un texto sobre el cuál tendrá que responder preguntas cuyos resultados garantizarán la comprensión del texto.



Figura 50: Alumna realizando pruebas con Histo

Al finalizar la realización de las pruebas, los datos de la base de datos son extraídos para analizar los resultados de los textos realizados por los alumnos.

Los alumnos 5, 6, 7 y 8 realizaron la prueba en conjunto, ya que el nivel de dificultad que se le puso a ese usuario fue el más elevado de todos.

En la Tabla 10, se muestran los resultados que se han extraído de la base de datos. Se muestra el nivel del texto realizado por los alumnos, el número de preguntas que ha contestado el alumno, y el número de preguntas que tiene el texto insertadas (en formato preguntas contestadas/número total de preguntas), el número de preguntas que ha acertado el alumno, y si el texto ha sido aprobado teniendo en cuenta el número de preguntas mínimo para aprobar que se insertó al introducir el texto.

USUARIO	NIVEL DEL TEXTO	PREGUNTAS CONTESTADAS	PREGUNTAS ACERTADAS	TEXTO APROBADO
ALUMNO1	1	6/6	6	SÍ
ALUMNO2	1	6/6	6	SÍ
	2	7/7	6	SÍ
ALUMNO3	2	6/6	5	SÍ
ALUMNO4	3	6/6	4	NO
GRUPO	4	6/6	3	NO

Tabla 10: Resultados pruebas de aceptación con Histo

En primer lugar, se observa como a medida que el nivel de dificultad aumenta, los resultados se ven afectados. También es influyente el hecho de tener que realizar un texto de nivel 3 o 4 sin haber pasado antes por los niveles inferiores a modo de entrenamiento.

La experiencia de los usuarios en grupo no fue satisfactoria, no siempre se ponían de acuerdo con las respuestas y comentaron que el estudio debe ser individual.

Habría que comentar que hubo una alumna que no sabía cómo escoger la opción del texto, no sabía si tenía que pinchar sobre el texto, o introducir la palabra, y una vez introducida, no se le había ocurrido pulsar "intro" o el botón enviar.

Otro punto interesante, fue que a algún alumno le costó ver la diferencia entre las respuestas de una palabra y las de respuesta corta, pero tras una breve aclaración, lograron contestar correctamente.

Al finalizar el tiempo de uso de la aplicación, se les realiza un breve cuestionario con preguntas acerca de la aplicación.

USUARIO	GENERAL	RAPIDEZ	FACILIDAD DE USO	LO QUE MÁS LE HA GUSTADO	LO QUE MENOS LE HA GUSTADO
ALUMNO1	8	10	9	Interactividad con el sistema	Nada
ALUMNO2	9-10	9	10	Saber siempre dónde pulsar	Nada
ALUMNO3	8	9	8	Las preguntas salen de una en una	Falta de variedad en los textos
ALUMNO4	10	10	9-10	Claro y conciso	Nada
GRUPO	8-9	8	9	Corrige en el momento	Estética y diseño
MEDIA	8.68	8.75	9.06	-	-

Tabla 11: Resultados encuesta, valoración de 0 a 10

En la Tabla 11 se muestran los resultados obtenidos de la encuesta realizada a los alumnos (anexo VII). Se les pidió que valorasen de 0 a 10 la aplicación de manera general, en cuanto a rapidez y en cuanto a facilidad de uso. En la última fila de la Tabla 11, se realiza una media de los valores antes mencionados.

De las tres medias obtenidas, la más alta ha sido en facilidad de uso, que ha sido evaluada con un 9 de media.

Las dos últimas columnas de la Tabla 11, hacen referencia a lo que más les ha gustado a los alumnos, y lo que menos de la aplicación. Cada uno ha aportado un comentario a modo resumen sobre los aspectos más positivos y negativos de la aplicación.

Se puede observar en los comentarios, que los alumnos valoran positivamente la obtención de una corrección o respuesta en el momento y la facilidad de uso y sencillez de la aplicación.

5. CONCLUSIONES Y TRABAJO FUTURO

5.1. CONCLUSIONES

Una vez finalizada la aplicación, podremos decidir si ésta cumple con los requisitos que se expusieron en un inicio, durante las primeras fases del proyecto. Para ello, vamos a ver cómo se cumplen todos los requisitos uno a uno, tanto funcionales como no funcionales.

REQUISITOS FUNCIONALES

ReqF1. En el sistema habrá dos tipos diferentes de usuarios, con diferentes interfaces gráficas, adaptadas a sus opciones. Estos usuarios serán profesores, que podrán introducir textos y preguntas (Figuras 16, 19, 20 y 21), y alumnos, que podrán consultar estos textos y responder a las preguntas propuestas (Figuras 24, 29, 32 y 35).

Histo tendrá acceso diferenciado para profesores y alumnos. Una vez se ha autenticado el usuario en el sistema, su ventana de inicio será de profesor o de alumno en función del perfil del usuario autenticado. Las opciones para ambos perfiles están diferenciadas. En la ventana de inicio del profesor, las opciones que se ofrecen son editar un texto o introducir un texto nuevo, mientras que para el alumno, las opciones serán dos modos de ejecución diferentes, aleatorio y normal. (Figuras 41 y 42).

ReqF2. Todo texto tendrá un nivel de dificultad, seleccionado por el profesor, para organizar el orden en el que repasará los textos el alumno.

El valor del nivel se le presenta al profesor mediante un combo con opciones (niveles del 1 al 4), de forma que al insertar el texto en la base de datos, el nivel siempre tendrá un valor asignado.

ReqF3. Todos los textos tendrán también un número de preguntas, introducido por el profesor, que indicará el número de preguntas del texto, que el alumno tendrá que contestar correctamente, para que se pueda considerar que dicho texto se ha finalizado con éxito.

Antes de insertar o actualizar un texto en la base de datos, se comprobará que todos los datos necesarios contienen un valor distinto de null, para evitar que se pueda guardar un texto para el cual no se han rellenado todos los campos.

ReqF4. Con cada texto se introducirán mínimo cinco preguntas, que podrán ser de tres tipos diferentes.

Cada vez que una pregunta sea insertada se actualizará un contador, si éste no ha llegado a 5 cuando se ha finalizado la inserción del texto, quedará como no validado, de forma que no se podrá recuperar para los alumnos. Mediante la opción de editar texto, el profesor podrá

retomar la inserción del texto y añadir las preguntas restantes. Antes de iniciar una pregunta, el profesor tendrá que decidir qué tipo de pregunta desea insertar, de entre tres tipos diferenciados por la respuesta, que podrá ser corta, de elección múltiple o de una palabra.

ReqF5. El alumno tendrá la posibilidad de empezar la sesión de estudio de dos formas: modo normal y modo aleatorio.

Para ello, la conversación con Histo, que se mostrará en HistoVentana.java, dependerá del modo en el que se haya empezado la sesión de estudio. El modo será elegido por el alumno al iniciar la sesión, que tendrá que pulsar sobre uno de los dos botones.

ReqF6. Todos los textos que se le mostrarán al alumno serán de su mismo nivel, de manera que no tenga que lidiar con textos para los que no está preparado aún, ni realizar textos que puedan resultarle aburridos por estar familiarizado con ellos.

Esto se ha conseguido teniendo presente durante toda la sesión el nivel de estudio del alumno. Cada vez que el alumno solicita un texto nuevo, se pone como condición, que el texto que se recupere de la base de datos tenga un nivel que coincida con el nivel de estudio del alumno.

ReqF7. Guardado automático tras finalizar cada pregunta, para evitar pérdidas.

Siempre que se pasa a una nueva ventana, tanto para el profesor al insertar preguntas como para el alumno al contestarlas, se le pide que pulse el botón siguiente, que realizará las inserciones y actualizaciones pertinentes en la base de datos.

ReqF8. Cerrar el sistema pulsando el botón cerrar o tecleando "salir".

Histo reconoce para el alumno la cadena "salir". Siempre que el alumno introduzca texto, se evaluará la cadena introducida. Si no se reconoce, se lo dirá al alumno, si la reconoce, llevará a cabo la acción correspondiente. La cadena "salir" es una de las contempladas por el sistema, que cerrará la ventana HistoVentana.java, volviendo a la ventana de inicio del alumno.

Todos los requisitos funcionales se cumplen en la aplicación. En las pruebas de validación explicadas con anterioridad, se confirma el cumplimiento de estos requisitos con capturas de pantalla.

REQUISITOS NO FUNCIONALES

ReqNF1. Posibilidad para el alumno de dar las respuestas en lenguaje natural, haciendo más fácil e interesante repasar historia.

ReqNF2. Rapidez a la hora de mostrar al alumno los textos que se recuperan de la base de datos y para corregir las preguntas realizadas por el alumno.

ReqNF3. Interfaz amigable y de uso sencillo tanto para el alumno como para el profesor.

En cuanto a los requisitos no funcionales, han sido también logrados con éxito.

La posibilidad de las respuestas en lenguaje natural queda cubierta con el hecho de que en las preguntas de respuesta corta, el alumno introducirá su respuesta, para posteriormente compararla con la respuesta correcta. Esto le otorga una cierta libertad a la hora de expresarse y redactar, que no implicará que su respuesta sea incorrecta.

La rapidez de la recuperación de los datos se ha conseguido dividiendo mucho las select que acceden a la base de datos, impidiendo que para recuperar un texto, se produzcan muchos accesos, incrementando el tiempo de respuesta y haciendo que se produzcan demoras innecesarias. Es importante tener en cuenta que si el nivel de contenidos de la base de datos aumenta en un nivel muy importante, sí podría verse ralentizada ligeramente la aplicación.

La aplicación intenta guiar tanto al alumno como al profesor en su uso de la aplicación para que resulte sencillo e intuitivo. Con el alumno, en todo momento, va solicitando el paso siguiente a realizar, para evitar que el alumno se bloquee o no sepa cuáles son sus opciones para continuar.

5.2. TRABAJO FUTURO

Respecto al trabajo futuro, hay que contemplar que existe un amplio abanico de posibilidades en cuanto al modo de repaso de una asignatura.

- Ampliación de los tipos de preguntas. Que podrían focalizarse dependiendo del curso en el que se encuentren los alumnos.

- Corrección de las preguntas de respuesta corta. Sería muy interesante que se pudiese corregir, al menos en cierta medida, ese tipo de respuestas.

- Incluir temas, o incluso imágenes que el alumno pueda emplear para hacer comentarios de fotos, ya que en algunos cursos de educación secundaria, así como en bachillerato, la realización de este tipo de ejercicios es obligatoria.

6. BIBLIOGRAFÍA

REFERENCIAS

[1] Holzer, Steven. Anaya (2000). La Biblia de Java 2

[2] Gilfillan, Ian. Anaya. La Biblia de MySQL

[3] Apuntes de Ingeniería del Software (Kybele)

LIBROS Y APUNTES

- de Miguel, Adoración; Piattini, Mario; Marcos, Esperanza. Ra-Ma (2000). Diseño de Bases de Datos Relacionales

- Eckel, Bruce. Addison Wesley (2002). Piensa en Java

- Vélez Serrano, José F., Técnicas Avanzadas de Diseño de Software

TUTORIALES WEB

- <http://www.webtaller.com/construccion/lenguajes/java/lecciones/conexiones-db-java.php>

- http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=tutorial_basico_jdbc

ANEXO I

TEXTO NIVEL 1

Etapas de la reconquista

La reconquista es el proceso de expansión de los pueblos cristianos del norte hacia el sur entre los siglos XIII y XV.

Los reinos visigodos emprendieron la reconquista con la idea de restaurar la monarquía visigoda de la que se sentían herederos legítimos, con un fuerte espíritu de cruzada contra los infieles y así superponer una religión sobre la otra.

Se puede apreciar cuatro etapas diferenciadas: la primera hasta el siglo X en la que los reinos cristianos avanzan hasta el Duero; la segunda, hasta mediados del siglo XII, llega hasta los valles del Tajo y el Duero; en la tercera, hasta el siglo XIII, se conquista el territorio hasta el valle del Guadiana y los pasos de Sierra Morena; la cuarta y última fase corresponde hasta la Batalla de las Navas de Tolosa, 1212. En esta última etapa Castilla se expande hasta el valle del Guadalquivir y Murcia, Aragón hasta Valencia Baleares, y Portugal hasta el Algarve. La reconquista termina con la incorporación del Reino de Granada.

PREGUNTAS

1.(Corta)¿Cómo se llevó a cabo la reconquista de los pueblos cristianos?

2.(Test)¿Cuántas etapas se aprecian en la reconquista?

- a. 4
- b. 3
- c. 5

3.(Test)¿Con la incorporación de qué reino se finaliza la reconquista?

- a. el Algarve
- b. Valencia
- c. el Reino de Granada

4.(Palabra)¿Hasta qué siglo llega la primera etapa de la reconquista?

5.(Corta)¿A qué batalla corresponde la última etapa de la reconquista?

6.(Test)La reconquista se llevó a cabo entre los siglos...

- a. XII y XV
- b. XIII y XV
- c. XIII y XVI

7.(Palabra)La batalla de las Navas de Tolosa fue en el año...

ANEXO II

TEXTO NIVEL 2

El Imperio de Carlos V

Carlos de Habsburgo inicia la dinastía de los Austrias en la monarquía hispánica. Es coronado como Carlos I de España sobre la monarquía hispánica que hereda de sus abuelos, los Reyes Católicos, y como Carlos V de Alemania sobre Flandes, el Franco Condado, Luxemburgo y el Sacro Imperio Romano Germánico, del que es elegido emperador.

En España es un rey que no le gusta al pueblo, puesto que no está presente en Castilla, no conoce la lengua y pide mucho dinero para costear sus guerras europeas. Por esto, se suceden dos revueltas populares: una, la Rebelión de las Comunidades en Castilla, exige la limitación del poder real y la reducción de impuestos; otra, la Rebelión de las Germanías, en Valencia y Mallorca, se dirigió contra los señores feudales y sus siervos mudéjares. Tras esto, la Corona sale muy respaldada y lleva a cabo una gran política exterior.

Carlos I, fuera de la monarquía hispánica, tuvo como objetivo la defensa de su herencia dinástica. Se enfrentó a Francia, el Imperio Otomano y a los protestantes alemanes. Cuando abdica, el Imperio Alemán se lo cede a su hermano Fernando y el resto a su otro hijo, Felipe II.

PREGUNTAS

1.(Palabra)En España, Carlos de Habsburgo es coronado como...

2.(Corta)¿Dónde es coronado Carlos de Habsburgo como Carlos V?

3.(Test)¿Qué revueltas populares se suceden?

- a. Comunidades y Germanías
- b. Valencia y Mallorca
- c. Valencia y Comunidades

4.(Palabra)¿A quién cede Carlos V el Imperio Alemán tras abdicar?

5.(Corta)¿Qué le cede Carlos I a su hijo, Felipe II, tras abdicar?

6.(Test)¿De dónde fue Carlos V proclamado emperador?

- a. Sacro Imperio Romano Germanico
- b. Alemania
- c. Franco Condado

7.(Corta)¿A qué países se enfrentó para defender la herencia dinástica?

ANEXO III

TEXTO NIVEL 3

Los Austrias del siglo XVII

En el siglo XVII, en España, los reyes no gobernaron personalmente el gobierno, sino que delegaron en los validos, ministros de confianza del rey que gobernaban en su nombre. Los tres reyes de este siglo son Felipe III, quien delega en el duque de Lerma y en el duque de Uceda; Felipe IV, con el conde duque de Olivares y don Luis de Haro; y Carlos II por el que gobernaron el duque de Medinaceli y el conde de Oropesa, en parte por la incapacidad del monarca debido a problemas de salud.

Este sistema trajo consigo un aumento de la corrupción en el puesto de los validos que aprovecharon su poder para conseguir beneficios, títulos y cargos para ellos, familiares y amigos, iniciándose rivalidades e intrigas en la corte.

Como resultado de estas tensiones políticas, así como de la crisis económica y social, la monarquía hispánica tendrá que afrontar una serie de conflictos internos. Felipe III, al expulsar a los moriscos, se enfrenta a una bajada considerable de la mano de obra en Valencia y Aragón; Felipe IV, en 1640, a rebeliones en Cataluña y Portugal; y Carlos III, en un clima de inestabilidad política, a revueltas antiseñoriales y antifiscales en Cataluña y a las segundas germanías en Valencia.

PREGUNTAS

- 1.(Palabra)Un ministro de confianza del rey, que gobierna en su nombre es un...
- 2.(Test)Los validos de Felipe III son...
 - a. conde duque de Olivares y duque de Lerma
 - b. duque de Lerma y duque de Medinaceli
 - c. duque de Lerma y duque de Ucesa
- 3.(Corta)¿Por qué se enfrentó Felipe III a una bajada considerable de la mano de obra en Valencia y Aragón?
- 4.(Palabra)¿En qué año se enfrentó Felipe IV a las rebeliones de Cataluña y Portugal?
- 5.(Test)¿Qué rey se enfrentó a un clima de inestabilidad política con revueltas antiseñoriales?
 - a. Felipe III
 - b. Carlos III
 - c. Felipe IV
- 6.(Test)¿Qué rey fue relevado en sus funciones como gobernante por el duque de Medinaceli y el conde de Oropesa?
 - a. Carlos II
 - b. Felipe III
 - c. Felipe IV

ANEXO IV

TEXTO NIVEL 4

La guerra de sucesión y el sistema de Utrecht

Carlos II muere en 1700 sin descendencia y deja los reinos españoles a Felipe de Anjou, nieto de Luis XIV, rey de Francia. Al temer Inglaterra y Austria que Francia se una a España, presentan a otro candidato, el archiduque Carlos, que llega a ser coronado como Carlos III.

Se forma así, la Gran Alianza de La Haya con Inglaterra, Holanda y Portugal que apoyan a Austria, contra el bando francoespañol formado por Francia, Castilla y Navarra. Así la guerra se sucede tanto en Europa como en España, donde se convierte en una guerra civil.

Dentro de España, Castilla y Navarra apoyan a Felipe de Anjou, mientras Cataluña, Aragón y Valencia apoyan a Carlos quien en 1711 heredará por sorpresa el trono de Austria por la muerte de su hermano. De este modo, Inglaterra, Holanda, Portugal y Saboya cambian de parecer y aceptan a Felipe como Felipe V de España. La guerra se extenderá por España hasta que en 1715 es tomada la isla de Mallorca.

En 1713 se firma la Paz de Utrecht, donde se le otorga a Inglaterra ventajas militares, mercantiles como el asiento de negros y el navío de permiso; y ventajas territoriales, pues consigue Menorca y Gibraltar en el Mediterráneo y Terranova en el Atlántico. En España, la gran perdedora, se asientan los Borbones, se acerca a Francia y se convierte en enemigo directo de Inglaterra y Austria.

PREGUNTAS

1.(Palabra)¿Quién hereda los reinos españoles tras morir Carlos II sin descendencia?

2.(Test)La Gran Alianza de la Haya estaba compuesta por...

- a. Inglaterra, Holanda, Portugal y Francia
- b. Inglaterra, Austria, Castilla y Holanda
- c. Inglaterra, Holanda, Portugal y Austria

3.(Palabra)¿En qué año hereda Carlos el trono de Austria?

4.(Palabra)La Paz de Utrecht se firma en el año...

5.(Test)¿A qué país se le otorgan beneficios con la Paz de Utrecht?

- a. Francia
- b. Inglaterra
- c. Austria

6.(Test)Dentro de España, ¿quién apoya a Felipe de Anjou?

- a. Aragón y Valencia
- b. Aragón, Castilla y Navarra
- c. Castilla y Navarra

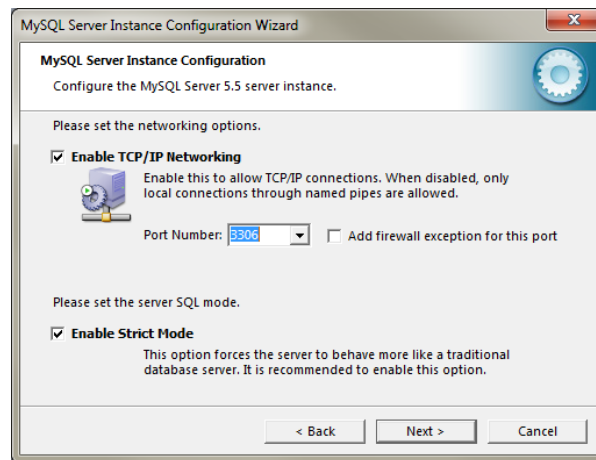
ANEXO V

Manual de instalación

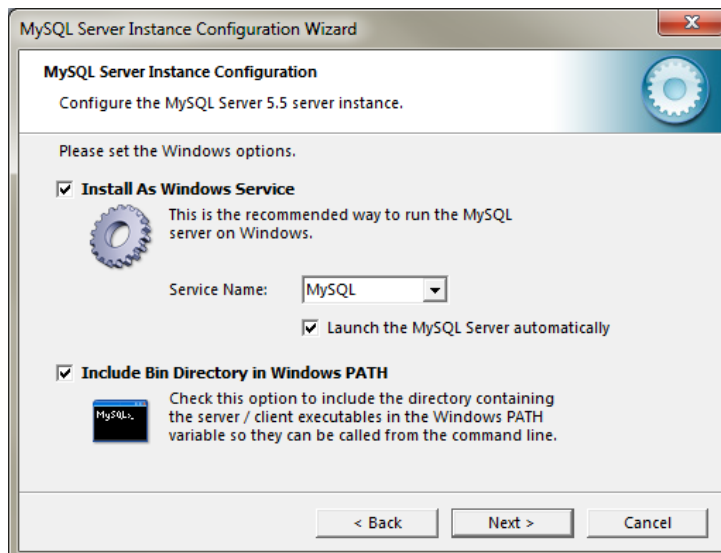
Para poder usar Histo, será necesario tener instalado MySQL. En caso de no estar instalado, habrá que instalarlo siguiendo los pasos detallados a continuación.

Paso 1. En primer lugar, descargaremos MySQL 5.5 del sitio web: <http://dev.mysql.com/downloads/>

Paso 2. Instalaremos MySQL. Iremos pulsando siguiente (next) en todas las pantallas.



Cuando aparezca la pantalla de la imagen anterior, se dejará el puerto 3306, que es el que sale por defecto.



Es importante seleccionar el check de "Include Bin Directory in Windows PATH".



En esta pantalla, habrá que introducir una contraseña. Es muy importante recordarla puesto que será solicitada más adelante.

Paso 3. Se ejecutará el archivo CreateBDHisto.bat, que solicitará un usuario y una contraseña. El usuario será "root" y la contraseña, la que se introdujo en la instalación.

Paso 4. Se ejecuta Histo.bat, que iniciará la aplicación. Es importante asegurarse de que la carpeta lib con el driver o controlador (mysql-connector-java-5.1.20-bin), se encuentra junto con los ejecutables .bat.

ANEXO VI

MANUAL DE USUARIO

PROFESOR

1. *Acceso.* El usuario deberá insertar su usuario y contraseña en los campos correspondientes y pulsar sobre el botón Acceder le aparecerá la ventana de inicio. En esta, deberá escoger si va a crear un nuevo texto, o va a modificar uno ya existente.
2. *Nuevo texto.* Sobre la ventana del nuevo texto deberán ser completados todos los campos.
 - 2.1. *Título.* Identificará al texto y permitirá al alumno realizar búsquedas.
 - 2.2. *Preguntas para validar.* Este número representa el número de preguntas que tendrá que contestar el alumno correctamente para que el texto se considere como aprobado.
 - 2.3. *Nivel.* Habrá que seleccionar en el combo el nivel de 1 a 4, siendo 1 el más fácil, y 4 el más difícil.
 - 2.4. *Texto.* El texto sobre el cuál se realizarán las preguntas.
3. *Selección de tipo de pregunta.* Se podrá elegir entre 3 tipos de preguntas: pregunta de tipo test, pregunta de respuesta de una palabra y pregunta de respuesta corta. Habrá que seleccionar el radiobutton del tipo de pregunta que se quiera completar.
4. *Preguntas.*
 - 4.1. *Pregunta corta.* Habrá que completar el enunciado y la respuesta correcta que se le mostrará al alumno para la autocorrección.
 - 4.2. *Pregunta test.* Se rellena el enunciado, y además las tres opciones, una será correcta, y las otras dos no. Se marca el radiobutton situado a la izquierda de la respuesta correcta, para que quede guardada y se utiliza para la corrección.
 - 4.3. *Pregunta palabra.* Se añade el enunciado y la respuesta, que estará formada por una sola palabra, o un nombre propio con apellido.

Se completarán todas las preguntas que se quiera, hasta un máximo de 10, en el que no se permitirá insertar más. Si se introducen menos de 5, el texto no se validará, y no será mostrado al alumno en sus sesiones de estudio.
5. *Editar texto.* En la pantalla de inicio seleccionamos la opción de editar texto, y aparecerá en esta ocasión, la ventana del editor, pero en vez de un campo para insertar el título, tendremos un combo, en el que aparecerán todos los títulos de textos que el usuario haya insertado con

anterioridad. Para las modificaciones, bastará con cambiar los datos que se quiera, y pulsar en guardar. También se podrán añadir preguntas nuevas, pulsando en añadir preguntas.

ALUMNO

1. *Acceso.* Para acceder a la aplicación, el usuario tendrá que insertar en los campos su usuario y su contraseña. Tras pulsar el botón acceder, aparecerá una ventana de inicio, con dos botones: modo normal y modo aleatorio.
2. *Modo normal.* En la ventana Histo, aparecerán dos opciones, empezar un texto nuevo, o reanudar uno ya empezado. Para seleccionar una de las dos opciones, bastará con introducir el número que el corresponde a la opción en el cuadro de texto situado en la parte inferior, y pulsar “Enter” o el botón Enviar.
3. *Empezar nuevo texto.* Tras escoger la opción de empezar un texto nuevo, aparecerá en la ventana un texto. Al terminar de leer, habrá que seleccionar la opción de contestar preguntas, igual que antes, introduciendo el número, y pulsando la tecla enter o el botón Enviar.
4. *Preguntas.* Irán apareciendo en nuevas ventanas, adaptadas al tipo de pregunta.
 - 4.1. *Pregunta de respuesta corta.* Se contestará de manera breve. Tras haber insertado la respuesta en el campo de texto, se pulsa corregir. Aparecerá una ventana de aviso, con la respuesta dada, y la respuesta correcta, tras compararlas, el usuario decidirá si es correcta o no, pulsando sí o no.
 - 4.2. *Pregunta tipo test.* En este caso, en la ventana se plantean tres opciones con radio buttons a la izquierda para seleccionar la que se considere correcta. Al pulsar corregir, la aplicación indicará si la respuesta ha sido correcta o no.
 - 4.3. *Pregunta de respuesta de una palabra.* En el campo de texto de la respuesta, se contestará con una sola palabra, o el caso correspondiente, una fecha, un número, o el nombre de una persona, que aunque podría incluir nombre y apellidos (dos palabras) se consideraría como una sola. Al pulsar sobre corregir, se muestra en la ventana si la respuesta ha sido correcta, y en caso contrario, la respuesta insertada por el profesor.
5. *Fin de texto.* Al finalizar un texto con todas sus preguntas, la aplicación evaluará si el alumno está en situación de subir de nivel o bajar, o en su defecto, quedarse en el nivel en el que está. En caso de haber dejado el texto sin acabar, es decir, sin haber contestado todas las preguntas propuestas por el profesor, el texto podrá ser retomado con la opción reanudar texto.
6. *Reanudar un texto ya empezado.* El usuario tendrá que insertar una palabra contenida en el título del texto que desee reanudar. Tras insertar la palabra, pulsará la tecla enter o el botón enviar. Si el texto no ha sido encontrado, o se han encontrado varios que contengan esa palabra, se mostrarán todos los títulos, con un número a la izquierda, y para seleccionar

uno de ellos, habrá que teclear el número correspondiente, y pulsar la tecla enter o el botón enviar. Si la aplicación solo encuentra un texto que coincida con la palabra insertada por el usuario, mostrará el texto directamente.

7. *Modo aleatorio*. La única diferencia con el modo normal es que no habrá posibilidad de reanudar textos y seleccionarlos, irán saliendo de manera automática al finalizar un texto.
8. *Salir*. Para finalizar la aplicación bastará con escribir “salir” y pulsar la tecla enter o el botón enviar.

ANEXO VII

CUESTIONARIO DE SATISFACCIÓN

Nombre:

Nivel de manejo con el ordenador:

Valore las siguientes cualidades de 0 a 10:

Opinión general ->

Rapidez ->

Facilidad de uso ->

Conteste a las siguientes preguntas:

¿Qué es lo que más le ha gustado de la aplicación?

¿Qué es lo que menos le ha gustado de la aplicación?