



**ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA
INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN**

Curso Académico 2010/2011

Proyecto de Fin de Carrera:

**ADAPTACIÓN AUTOMÁTICA DE DIÁLOGOS DE AGENTES
CONVERSACIONALES PEDAGÓGICOS A LOS RASGOS DE LA
PERSONALIDAD DEL ESTUDIANTE**

Autor: Alberto Redondo Hernández

Tutora: Diana Pérez

AGRADECIMIENTOS

A mi familia que gracias a su apoyo en los buenos y en los malos momentos, se han convertido en uno de los pilares indispensables para seguir este camino.

A mis profesores, sobre todo a mi tutora Diana, por sus consejos, ayuda y confianza que ha prestado en mi durante todo el proyecto. A José, por sus ideas antes situaciones difíciles, sin él no hubiera sido lo mismo.

A mis compañeros, pues han estado ahí cuando los he necesitado.

Y a Julia, por ser una persona tan especial.

Resumen

El uso de las Tecnologías de la Información se ha convertido en una herramienta indispensable en nuestra vida cotidiana. Esto es cierto en múltiples ámbitos y en los que nos hacen más fácil el trabajo del día a día.

En particular, este trabajo se centra en el ámbito educativo (e-learning). Estas son herramientas informáticas que nos ayudan a aprender, de una forma sencilla e intuitiva utilizando todo el potencial de las Tecnologías de la Información.

Si además se tiene en cuenta el comportamiento, personalidad y el estilo de aprendizaje de cada persona, se espera que la información que se transmita, pueda ser comprendida con una mayor facilidad. Los agentes conversacionales son aquellos con los cuales podemos mantener un diálogo directo entre el agente y el interlocutor. Sin embargo, la codificación y la adaptación de agentes conversacionales no es una tarea fácil, ya que cada persona posee una personalidad diferente, aunque en este respecto los psicólogos pueden hacer una estimación, basados en la experiencia del comportamiento durante muchos años.

Con este objetivo, este Proyecto Final de Carrera está orientado a la generación de preguntas adaptadas a cada estudiante, con la ayuda de dos test psicológicos, el test de “Big Five” o de cinco grandes rasgos y el test de “Felder” o test de Estilos de Aprendizaje. Con estos tests se puede hacer una estimación de la forma de ser del estudiante y su comportamiento frente a situaciones de aprendizaje. Con esta estimación se han realizado los patrones para cada tipo de personalidad y para cada Estilo de Aprendizaje. De esta forma se puede adaptar la forma en que se presenta la información, para potenciar los rasgos obtenidos y que cuando el estudiante interactúe con la aplicación se sienta identificado con los aspectos que se le muestran. Este objetivo se ha implementado en una aplicación llamada Flayer mediante el lenguaje de programación Java, que se presenta vía web mediante tecnologías servlet y JSP.

Flayer se ha probado con veinte usuarios obteniendo unos resultados de valores concretos, que nos ayudan a hacernos una idea del funcionamiento que tiene la aplicación y de poder mejorarla. Un 50% de los estudiantes considera que las preguntas generadas automáticamente se han adaptado correctamente y un 40 % de los estudiantes considera que la aplicación es interesante.

INDICE

Capítulo 1. Introducción	1
Capítulo 2. Objetivos	3
2.1. Descripción del problema.....	3
2.2. Estudio de alternativas.....	5
2.2.1. Lenguajes de programación.....	5
2.2.2. Lenguajes Web.....	6
2.2.3. Gestores de bases de datos.....	9
2.2.4. Servidores.....	10
2.2.5. Plataformas de desarrollo.....	10
2.2.6. Sistemas operativos.....	11
2.3. Metodología empleada.....	13
Capítulo 3. Descripción informática	17
3.1. Especificación.....	17
3.1.1. Análisis de requisitos.....	17
3.1.2. Casos de uso.....	18
3.2. Arquitectura de alto nivel.....	27
3.3. Diagrama E/R.....	28
3.4. Algoritmo utilizado.....	35
3.5. Descripción de módulos.....	39
Capítulo 4. Pruebas	47
4.1. Pruebas unitarias.....	47
4.2. Pruebas de interacción.....	52
4.3. Pruebas de validación.....	53
4.4. Pruebas de aceptación.....	56
4.4.1. Informáticos.....	56
4.4.2. No informáticos.....	58
Capítulo 5. Conclusiones y trabajo futuro	59
5.1. Logros alcanzados.....	59
5.2. Trabajos futuros.....	60
Bibliografía	61

Anexo A. Diagramas de casos de uso.....	62
Anexo B. Diagramas de secuencia.....	68
Anexo C. Evolución del diseño centrado en el usuario.....	70
Anexo D. Problemas solventados.....	76
Anexo E. Manual de instalación.....	77
Anexo F. Manual de usuario.....	79

INDICE DE TABLAS

TABLA 1: COMPARATIVA DE LENGUAJES DE PROGRAMACIÓN.....	6
TABLA 2: LENGUAJES WEB	7
TABLA 3: LENGUAJES WEB SERVIDOR.....	8
TABLA 4: GESTORES DE BASES DE DATOS.....	9
TABLA 5: SERVIDORES.....	10
TABLA 6: ENTORNOS INTEGRADOS.....	11
TABLA 7: SISTEMAS OPERATIVOS.....	13
TABLA 8: CAMINOS.....	49
TABLA 9: PRUEBA DE CAJA NEGRA.....	49
TABLA 10:PARTICION EQUIVALENTE.....	50
TABLA 11: RESULTADOS DE LA PRUEBA 1 CON LOS ALUMNOS DE INFORMATICA.....	57
TABLA 12: RESULTADOS DE LA PRUEBA 2 CON ESTUDIANTES NO INFORMATICOS.....	58

INDICE DE FIGURAS

FIGURA 1: PANTALLA PRINCIPAL.....	1
FIGURA 2: DISEÑO CENTRADO EN EL USUARIO.....	15
FIGURA 3: DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN.....	18
FIGURA 4: DIAGRAMA DE CASOS DE USO DEL PROFESOR.....	18
FIGURA 5. DIVISIÓN DEL CASO DE USO GESTIONAR PREGUNTAS.....	19
FIGURA 6. DIVISIÓN DEL CASO DE USO REALIZAR TEST SATISFACCIÓN.....	20
FIGURA 7. DIVISIÓN DEL CASO DE USO VER PREGUNTAS MODIFICADAS.....	21
FIGURA 8. DIAGRAMA DE CASOS DE USO DEL USUARIO ALUMNO.....	22
FIGURA 9. DIVISIÓN DEL CASO DE USO REALIZAR TEST ESTILOS DE APRENDIZAJE.....	22
FIGURA 10. DIVISIÓN DEL CASO DE USO REALIZAR TEST OCEAN.....	23
FIGURA 11. DIVISIÓN DEL CASO DE USO VER RESULTADOS.....	24
FIGURA 12. DIVISIÓN DEL CASO DE USO REALIZAR TEST SATISFACCIÓN.....	25
FIGURA 13. DIVISIÓN DEL CASO DE USO VOTAR PREGUNTAS.....	26
FIGURA 14. ARQUITECTURA DE ALTO NIVEL.....	27
FIGURA 15. DIAGRAMA E/R DE BASE DE DATOS TEST.....	28
FIGURA 16. DIAGRAMA E/R BASE DE DATOS WILLOW.....	29
FIGURA 17. TABLA USUARIO DE BASE DE DATOS TEST.....	31
FIGURA 18. TABLA TEST_ESTILOS DE BASE DE DATOS TEST.....	31
FIGURA 19. TABLA TEST_OCEAN DE BASE DE DATOS TEST.....	31
FIGURA 20. TABLA PREGUNTA_ADAPTADA DE BASE DE DATOS TEST.....	31
FIGURA 21. TABLA PREGUNTA_MODIFICAR DE BASE DE DATOS TEST.....	32
FIGURA 22. TABLA PREGUNTA_COMBINAR DE BASE DE DATOS TEST.....	32
FIGURA 23. TABLA TEST_SATISFACION DE BASE DE DATOS TEST.....	32
FIGURA 24. TABLA ESTILOS_FINAL DE BASE DE DATOS TEST.....	32
FIGURA 25. TABLA PERSONALIDAD_FINAL DE BASE DE DATOS TEST.....	33
FIGURA 26. TABLA PREGUNTAS_COMPLETADAS_FINAL DE BASE DE DATOS TEST.....	33
FIGURA 27. TABLA PREGUNTAS_COMPLETADAS_FINAL_CON ESTA DE BASE DE DATOS TEST.....	33
FIGURA 28. TABLA ESTADISTICAS DE BASE DE DATOS TEST.....	34
FIGURA 29. TABLA USUARIO DE BASE DE DATOS WILLOW.....	34
FIGURA 30. TABLA PREGUNTA DE BASE DE DATOS WILLOW.....	34
FIGURA 31. DIAGRAMA DE LOS MODULOS PROFESOR.....	39
FIGURA 32. DIAGRAMA DE LOS MODULOS ESTUDIANTE.....	39
FIGURA 33. PRUEBA DE CAJA BLANCA.....	48
FIGURA 34. PRUEBA DE CAJA NEGRA.....	50
FIGURA 35. PRUEBA DE CAJA NEGRA 2.....	50
FIGURA 36. DIAGRAMA DE INTEGRACION DE LOS MODULOS.....	51
FIGURA 37. PRUEBA DE VALIDACION 1.....	52
FIGURA 38. PRUEBA DE VALIDACION 2.....	52
FIGURA 39. PRUEBA DE VALIDACION 3.....	52
FIGURA 40. PRUEBA DE VALIDACION 4.....	53
FIGURA 41. PRUEBA DE VALIDACION 5.....	53
FIGURA 42. PRUEBA DE VALIDACION 6.....	54
FIGURA 43. PRUEBA DE VALIDACION 7.....	54
FIGURA 44. PRUEBA DE VALIDACION 8.....	54
FIGURA 45. PRUEBA DE VALIDACION 9.....	55
FIGURA 46. GRAFICA PRUEBA 1.....	56
FIGURA 47. GRAFICA PRUEBA 2.....	57
FIGURA 48. DIAGRAMA DE SECUENCIA DEL ESTUDIANTE PRIMERA INSTANCIA.....	68
FIGURA 49. DIAGRAMA DE SECUENCIA DEL ESTUDIANTE SEGUNDA INSTANCIA.....	68
FIGURA 50. DIAGRAMA DE SECUENCIA DEL PROFESOR.....	69
FIGURA 51. INTERFAZ DE ACCESO INICIAL.....	70
FIGURA 52. INTERFAZ DE ACCESO EVOLUCION 2.....	71
FIGURA 53. INTERFAZ DE ACCESO EVOLUCION 3.....	72
FIGURA 54. INTERFAZ DE ACCESO EVOLUCION 4.....	72
FIGURA 55. INTERFAZ DE TEST INICIAL.....	73
FIGURA 56. INTERFAZ DE TEST EVOLUCION.....	74
FIGURA 57. INTERFAZ DE TEST EVOLUCION 2.....	74
FIGURA 58. INTERFAZ DE TEST EVOLUCION 3.....	75

Capítulo 1: Introducción

Hoy en día existen una gran variedad de aplicaciones informáticas para la educación que nos permiten realizar ejercicios a través de un ordenador. Muchas de ellas se realizan vía Web como por ejemplo tests que forman parte de plataformas e-learning.

Además, en la actualidad, se tiende a que las aplicaciones sean adaptativas, intentando conocer los gustos, prioridades y características del usuario, para que este reciba lo que necesita, y cuando lo necesita según sus propósitos.

Las aplicaciones de autor proporcionan una interfaz para la creación de entornos educativos adaptativos, permitiendo la configuración de entornos y la creación de actividades o tareas en los mismos. Con estas aplicaciones los profesores pueden crear información nueva y adaptada a sus alumnos, potenciando de esta manera el aprendizaje de los mismos, y aprovechándose de las ventajas que nos ofrece las últimas tecnologías para la enseñanza actual.

En este proyecto, el objetivo es generar las preguntas de los tests adaptándose a cada estudiante. Por ello se ha implementado el sistema Web llamado Flayer que se puede ver en la Figura 1.

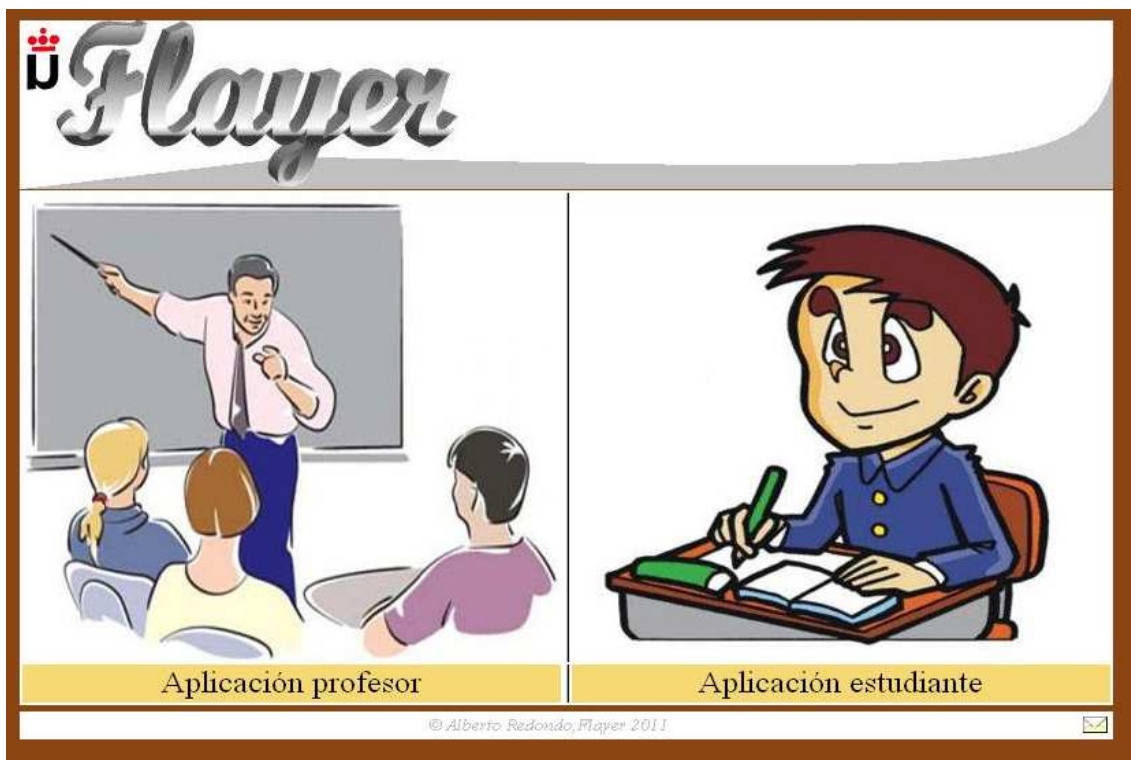


Figura 1. Pantalla principal

Se ha elegido el nombre de Flayer, en la aplicación, como la descripción de tener una ventaja respecto a los demás, ya que se parte de la idea de que si conocemos a las personas la información que se puede transmitir podrá ser asimilada más fácilmente.

Flayer es una aplicación que nos permite hallar la personalidad de un estudiante, así como los estilos de aprendizaje que utiliza normalmente a la hora de mantener diálogos conversacionales. Se espera que, sabiendo estos rasgos un profesor puede transmitir al estudiante preguntas de forma más clara y precisa. Para ello, Flayer transforma las preguntas introducidas por el profesor de forma automática, y que se quieren transmitir al estudiante de una forma natural.

En particular se espera que si el estudiante recibe la información de la pregunta de forma más adecuada a su personalidad y estilo de aprendizaje entonces prestará más atención y contestará mejor. Para conseguir realizar esta adaptación, Flayer integra dos test psicológicos: el test de “Big Five” o de cinco grandes rasgos y el test de “Felder” o test de Estilos de Aprendizaje. Dichos tests se encargan de observar, cómo reacciona el estudiante ante diferentes situaciones. Según estos resultados, se puede estimar el comportamiento, forma de pensar, la capacidad de razonamiento, y el nivel de abstracción del estudiante.

La generación de estos rasgos, resultados del análisis de los test completados por los estudiantes, le permiten a Flayer poder aplicar un mecanismo, para adaptar individualmente y automáticamente las preguntas a sus rasgos. Este mecanismo le permite al profesor, enseñar la unidad didáctica, de una manera efectiva. En la actualidad, sería muy complicado que un profesor, fuese adaptando su forma de presentar la información, a cada estudiante pensando que cada uno tiene una personalidad diferente y que los estilos de aprendizaje comparando los unos con los otros pueden cambiar.

Este trabajo consta de 5 capítulos: en el capítulo 2 se tratan los objetivos de forma específica, en el capítulo 3 se centra en la descripción informática principal del proyecto, en el capítulo 4 describe las pruebas realizadas y el capítulo 5 termina con las conclusiones y el trabajo futuro.

Capítulo 2: Objetivos

2.1 Descripción del problema

La adaptación automática de los sistemas de enseñanza a distancia a los rasgos de la personalidad y estilos de aprendizaje del estudiante es una tarea bastante complicada. Uno de los motivos más relevantes de esta dificultad es el alto número de factores y variables a tener en cuenta en el desarrollo de sistemas de enseñanza adaptativa.

Las personas tenemos rasgos de personalidad distintas y aprendemos de forma distinta. Los sistemas de enseñanza adaptativos han probado sus beneficios en cuanto a los contenidos y la navegación. Sin embargo, poco se ha trabajado en el área de generación de preguntas adaptadas a la personalidad y estilos de aprendizaje de cada estudiante.

En este trabajo, el objetivo es desarrollar una aplicación que a partir del test de “Big Five” o de cinco grandes rasgos y el test de “Felder” o test de Estilos de Aprendizaje, pueda recabar la información necesaria para generar preguntas adaptativas.

En particular, Flayer tiene en cuenta los siguientes **Estilos de Aprendizaje**:

- **ACTIVO**: persona que tiende a retener y comprender mejor la información.
- **PASIVO**: persona que prefiere pensar la información en silencio primero.
- **PERCEPTIVO**: persona que tiende a preferir los hechos de aprendizaje.
- **INTUITIVO**: persona que prefiere descubrir las posibilidades y sus relaciones.
- **VISUAL**: persona que recuerda mejor lo que ve (imágenes, diagramas, películas...).
- **VERBAL**: persona que saca más provecho de las palabras y de las explicaciones.
- **GLOBAL**: persona que tiende a aprender en grandes saltos, sin ver las conexiones
- **SECUENCIAL**: persona que tiende a ganar la comprensión en pasos lineales.

Y los siguientes **tipos de personalidad**:

- **EXTROVERSION**: persona que tiene rasgos como hablador/a, asertivo/a, energético/a, amigable, dominante...
- **INTROVERSION**: persona que tiene rasgos como tranquilo/a, reservado/a, vergonzoso/a, tímido/a...
- **CORDIALIDAD**: persona que es simpática, amable, afectiva, provechosa...

- **ANTIPATIA:** persona que tiene rasgos como crítica, fría, dura de corazón, poco agradecida...
- **RESPONSABILIDAD:** persona que tiene rasgos como organizada, planificadora, eficiente, fiable...
- **DESORGANIZACION:** persona que tiene rasgos como descuidada, frívola, irresponsable, olvidadiza...
- **ESTABILIDAD EMOCIONAL:** persona que tiene rasgos como estable, calmada, contenta...
- **NEUROTICISMO:** persona que tiene rasgos como tensa, ansiosa, nerviosa, preocupada...
- **ABIERTO DE MENTE:** persona con intereses amplios, imaginativa, original, penetrante...
- **CONVENCIONALISMO:** persona que tiene rasgos como ordinaria, de intereses estrechos, simple, superficial...

También es importante tener en cuenta que estos tests siempre realizan una estimación y que aunque suelen tener un porcentaje de acierto alto, son valores aproximados. Así, se supone por ejemplo, que un estudiante con estilo de aprendizaje visual tiende a preferir una pregunta con una imagen respecto a una pregunta exclusivamente textual, pero no se descarta que también un estudiante con estilo de aprendizaje verbal quisiera tener la imagen, por lo que se dará al profesor siempre la posibilidad de modificar la pregunta generada automáticamente dando control total de la aplicación al profesor para quién se espera que sea una herramienta útil e intuitiva.

En detalle, la aplicación a desarrollar en este proyecto deberá cumplir los siguientes objetivos generales:

- Identificar los rasgos de personalidad de estudiantes.
- Clasificar los diferentes estilos de personalidad de los estudiantes.
- Permitir a los profesores introducir preguntas para cualquier materia.
- Transformar automáticamente las preguntas originales de los profesores según la información almacenada de los estudiantes del test de “Big Five” o de cinco grandes rasgos y el test de “Felder” o test de Estilos de Aprendizaje.
- Almacenar las preguntas generadas para que puedan ser modificadas, confirmadas o eliminadas por el profesor y visualizadas por los estudiantes.
- Realizar estadísticas sobre las preguntas generadas automáticamente.

- Consultar la opinión de profesores y estudiantes sobre la aplicación y las preguntas generadas siguiendo un Diseño Centrado en el Usuario.

2.2 Estudio de alternativas

En esta sección se revisan las distintas tecnologías existentes en la actualidad para poder desarrollar la aplicación objetivo de este proyecto, escogiendo las tecnologías más adecuadas en cada caso.

2.2.1 LENGUAJES DE PROGRAMACION

Existen diferentes lenguajes de programación, entre los que se encuentran:

- **C:** es un lenguaje de programación de alto nivel, con el que se pueden escribir programas con fines muy diversos. Una de sus grandes ventajas, respecto a otros lenguajes de programación, es que el código producido por el compilador C, está muy optimizado en tamaño, eso produce una mayor velocidad de ejecución, una desventaja significativa es que C, es independiente de la plataforma solo en código fuente, lo cual, cada plataforma diferente debe proporcionar el compilador adecuado [1].
- **Java:** es un lenguaje de programación desarrollado por Sun Microsystems, en la década de los 90, orientado a objetos. Dicho lenguaje toma sintaxis de C y C++, aunque su modelo de objetos es más simple y elimina herramientas de bajo nivel, ya que inducen a muchos errores, como la manipulación de memoria. Las aplicaciones en Java pueden estar compiladas en bytecode o en código máquina nativo, normalmente se suele utilizar bytecode interpretado o compilado a código nativo para su ejecución, pero también se admite la ejecución directa por hardware del bytecode por un procesador Java [HTTP1].

La Tabla 1 muestra la comparativa entre los lenguajes de programación analizados.

Característica	JAVA	C
Orientado a objetos	√	x
Herramientas de bajo nivel	x	√
Sencillez	√	x
Robustez	√	x
Interpretado	√	x
Portabilidad	√	x

Excepciones	√	Algunas
Representación	Alta	Alta

Tabla 1. Comparativa de lenguajes de programación

Se ha elegido Java como lenguaje de programación por su sencillez, por ser un lenguaje interpretado y por la capacidad de portabilidad que posee.

2.2.2 LENGUAJES WEB

- **Lenguajes web cliente**

Javascript: es un lenguaje de programación, en el lado del cliente, en el cual se pueden hacer pequeños programas dentro del ámbito de una página Web, es muy utilizado debido a su compatibilidad con la mayoría de los navegadores más actuales. Con Javascript podemos definir interactividades con el usuario, así como crear efectos especiales en las páginas Web, el mayor recurso con el que cuenta dicho lenguaje, es el propio navegador, ya que es el que se encarga de interpretar las instrucciones Javascript [HTTP2].

- **Visual Basic .Net:** es uno de los lenguajes de programación de alto nivel que pertenecen al paquete .NET, con este lenguaje se puede escribir tanto programas convencionales como dirigidos a Internet. Las aplicaciones podrán mostrar una interfaz gráfica al usuario, o bien una interfaz de texto, como muestran las aplicaciones de consola. .NET tiene fundamentalmente tres componentes; una máquina virtual (CLR: Common Language Runtime), que procesa código escrito en un lenguaje intermedio (MSIL: Microsoft Intermediate Language), una biblioteca de clases (biblioteca .NET) y ASP.NET que proporciona los servicios necesarios para crear aplicaciones Web [2].
- **DHTML:** no sería precisamente considerado lenguaje de programación, mas bien, se trata de una capacidad que disponen los navegadores actuales y por la cual, se tiene un mayor control de las páginas Web. Dentro de DHTML, podría englobarse cualquier página que responde a las actividades del usuario, realiza efectos y funcionalidades en los cuales se pueden mostrar y/o ocultar elementos de la página. Se aprovecha de las nuevas estructuras para visualizar elementos denominada capa, estas, se pueden ocultar, mostrar, desplazar, etc... [HTTP2].
- **CSS:** son unas hojas de estilo en cascada, que nos permiten crear paginas Web, de una manera más exacta. Gracias a esta tecnología tenemos un mayor control,

de los resultados finales de la página, obteniendo mejores resultados que utilizando únicamente páginas en HTML, pudiendo incluir tipos de letra, márgenes, fondos de colores... [HTTP2].

- **Applets en Java:** estos permiten ejecutar código en los clientes que visualizan una página Web, realizados en Java, se transfieren con las páginas Web y el navegador ejecuta en el espacio de la página. Son más difíciles de programar que los scripts en Javascript y requieren unos conocimientos básicos o medios del lenguaje en Java. La principal ventaja de los Applets de Java, consiste en que son menos dependientes del navegador que los scripts en Javascript e independientes del sistema operativo del ordenador donde se ejecutan, una desventaja considerable es la lentitud de procesar y el espacio muy delimitado frente a Javascript. Por ello, no podemos abrir directamente ventanas secundarias, controlar Frames, formulario, capas, etc... [HTTP2].

La Tabla 2 se muestra los diferentes lenguajes Web cliente.

Característica	JavaScript	Visual Basic	Css	Applets de Java
Aprendizaje	Sencillo	Sencillo	Medio	Complejo
Orientado a objetos	√	x	x	√
Sistema Operativo	Multiplataforma	Windows o Mac	Multiplataforma	Multiplataforma
Código	Organizado	Muy Limitado	Estructurado	Estructurado
Software	Libre	Libre	Libre	Libre

Tabla 2. Lenguajes Web cliente

- **Lenguajes web servidor**
 - **CGI, Common Gateway Interface :** este lenguaje se utiliza para la programación de páginas dinámicas de servidor, en la actualidad se encuentra en desuso debido a la dificultad con la que se desarrollan los programas y la gran carga para el servidor a la hora de su ejecución, el lenguaje utilizado para los CGI, es Perl, aunque también se pueden utilizar C, C++ o Visual Basic [HTTP2].
 - **ASP (Active Server Pages):** es utilizado para la creación de páginas dinámicas de servidor, fue desarrollado por Microsoft, se caracteriza por escribirse en la misma página Web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft). Con ASP se pueden realizar diversas aplicaciones, permite acceso a

bases de datos, al sistema de archivos del servidor y a todos los recursos de los que disponga el servidor [HTTP2].

- **PHP, Hipertext Preprocesor:** es un lenguaje de programación para crear sitios Web dinámicos e interactivos. Una de las características clave de PHP es que puede incorporar código PHP en páginas Web HTML, facilitando la creación de contenido dinámico. PHP (Hypertext Preprocessor), da una buena idea de su propósito, que es, procesar información y crear hipertexto (HTML) como resultado. PHP es un lenguaje interpretado, el motor PHP procesa un script PHP cada vez que se ejecuta [3].
- **JSP, Java Server Pages:** es una tecnología utilizada para la creación de páginas Web con programación en Java. Las aplicaciones creadas con JSP, se pueden ejecutar en servidores Web, de múltiples plataformas, las páginas JSP se componen de código HTML/XML, mezclado con etiquetas para programar scripts de servidor con sintaxis en Java, por ello, las JSP se pueden escribir en nuestro editor HTML/XML. El motor de las páginas JSP está basado en los servlets de Java-son programas realizados en Java, para ejecutarse en un servidor. Los archivos que se generan son con extensión .jsp, antes de que sean funcionales, el motor JSP realiza una traducción en un servlet, implementado en un archivo class (Byte codes de Java) [HTTP2].

En la Tabla 3 se muestra los diferentes Lenguajes Web servidor analizados.

Característica	ASP	PHP	JSP
Aprendizaje	Sencillo	Sencillo	Complejo
Orientado a objetos	x	√	√
Sistema Operativo	Windows	Multiplataforma	Multiplataforma
Código	Desorganizado	Poco legible	Estructurado
Software	No Libre	Libre	Libre

Tabla 3. Lenguajes Web Servidor

Como Lenguaje Web se ha elegido JSP, por ser multiplataforma y software libre.

2.2.3 GESTORES DE BASES DE DATOS

- **MySql:** es un sistema de gestión de bases de datos relacional, le permite soportar gran carga gracias a su diseño multihilo. MySQL fue creada por la empresa MySQL AB, de origen sueco, que mantiene su copyright del código fuente del servidor. En la actualidad MySQL es software libre [HTTP3].
- **Postgresql:** es un sistema de gestión de bases de datos objeto-relacional, basado en el proyecto PostGres, desarrollado en la Universidad de Berkeley, ha sido el pionero en conceptos existentes en el sistema objeto-relacional actual, posteriormente se ha incluido en otros sistemas de gestión comerciales. PostGreSQL contiene características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, sin embargo, no es un sistema de gestión de bases de datos puramente orientado a objetos [HTTP4].
- **Microsoft Access:** permite la creación de ficheros de bases de datos relacionales, fue desarrollado por Microsoft, se encuentra dentro del paquete Microsoft Office, una de sus posibilidades es la creación de ficheros con bases de datos, que pueden ser consultados por otros programas. Microsoft Access se encuentra en la categoría de gestión, si hablásemos de sistemas de información. Dentro de las diversas funciones, se encuentra la de manipular los datos en forma de tablas, realizar cálculos complejos con fórmulas y funciones e incluso dibujar gráficas de distintos tipos [HTTP5].

En la Tabla 4 se muestra la comparativa de los gestores de bases de datos.

Característica	MySQL	PostgreSQL	Access
Sistema Operativo	Multiplataforma	Multiplataforma	Microsoft Windows
Unicode	√	√	√
Tabla temporal	√	√	x
Índices Hash	Tablas HEAP solamente	√	x
Vista materializada	x	x	√

Tabla 4. Gestores de Bases de Datos

Se ha elegido MySQL como gestor de bases de datos por ser multiplataforma y por su capacidad unicode.

2.2.4 SERVIDORES

- **Apache:** es un servidor Web robusto, cuya implementación se realiza de forma colaborativa, con funcionalidades equivalentes a las de los servidores comerciales. El proyecto se encuentra controlado y dirigido por un grupo de voluntarios de todo el mundo, los cuales, usan la Web y Internet para comunicarse, se denominan Apache Group [HTTP6].
- **Tomcat:** es un servidor Web debido a que gestiona solicitudes y respuestas http (incluye el servidor Apache), gracias a sus conectores http, también, es servidor de aplicaciones o contenedor de Servlets/JSP. En la actualidad, Tomcat es usado como servidor Web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Funciona en cualquier sistema operativo, que disponga de la maquina virtual Java, ya que fue escrito en Java [HTTP7].
- **GlassFish:** es un servidor de aplicaciones, que implementa la plataforma JavaEE5, debido a esto, soporta las ultimas versiones de tecnologías como: JSP, JSF, Servlets, Ejes, Java API para Servicios Web (JAX-WS), Arquitectura Java para enlaces XML (JAXB), etc....[HTTP8].

En la Tabla 5 se muestra la comparativa de los diferentes servidores analizados

Característica	Apache	Tomcat	GlassFish
Estabilidad	Alta	Alta	Alta
Instalación	Sencilla	Muy Sencilla	Sencilla
Idiomas	Inglés	Inglés	Inglés
Sistema Operativo	Multiplataforma	Multiplataforma	Multiplataforma
Software	Libre	Libre	Libre

Tabla 5. Servidores

Se ha elegido Tomcat como servidor por su alta estabilidad y por ser software libre.

2.2.5 PLATAFORMAS DE DESARROLLO

- **Eclipse:** es un entorno de desarrollo integrado, multiplataforma y de código abierto, utilizado para desarrollar lo que el proyecto denomina, “Aplicaciones de Cliente Enriquecido”, al contrario que las aplicaciones “Cliente-liviano”, basadas en navegadores. En general, dicha plataforma ha sido utilizada para

desarrollar entornos de desarrollo integrados, como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ), sin embargo, se puede utilizar para aplicaciones cliente como BitTorrent, Azureus. Eclipse es una comunidad de usuarios, extendiéndose en áreas de aplicación cubiertas, como puede ser, Eclipse Modeling Project, que cubre casi todas las áreas de Model Driven Engineering. Fue desarrollado por IBM, como el sucesor de su familia de herramientas para VisualAge [HTTP9].

- **NetBeans:** es un entorno de desarrollo integrado, el cual, permite que las aplicaciones sean desarrolladas con un conjunto de componentes denominados módulos. Un modulo es un archivo Java, que contiene clases de Java escritas para interactuar con las APIs de NetBeans y un archivo (manifestfile) que hace que lo identifique como módulo. Las aplicaciones realizadas con módulos, se pueden agrandar añadiendo más módulos, ya que pueden ser independientes los unos de los otros. NetBeans es un proyecto de código abierto, con una gran base de usuarios. Fue desarrollado por Sun Microsystems y continúa siendo su patrocinador principal [HTTP10].

En la Tabla 6 se muestra la comparativa de los entornos integrados analizados.

Característica	Eclipse	NetBeans
Sistema Operativo	Multiplataforma	Multiplataforma
Género	IDE, Java SDK, C/C++	Java SDK
Software	Libre	Libre
Lenguajes de programación	Jsp,Sh,Perl,Php, Sed	Java, Jsp, Haskell, Cpp, Yacc, Sh, Lex, Perl, Objc, Ansic
Idioma Español	√	√

Tabla 6. Entornos Integrados

Se ha elegido NetBeans como entorno integrado, por ser un software libre y por la cantidad de lenguajes de programación que soporta.

2.2.6 SISTEMAS OPERATIVOS

- **Microsoft Windows:** en sus inicios, era un programa funcionando en el sistema operativo MS-DOS. Su primera versión, Windows 1.0, salió a la luz en 1985 y la última versión de gran popularidad fue la 3.x, en 1990. Windows avanzó

cuando se convirtió en un sistema operativo con el nombre de Windows 95, respaldado por una gran campaña publicitaria, dicha versión, se acompañó con grandes mejoras, luego se sucedieron versiones posteriores como Windows 98, ME, XP, Vista, etc. Ha sido cuestionado por su baja seguridad, la competencia desleal y el monopolio de Microsoft, aprovechando su popularidad. Desde la versión que le hizo famoso, Microsoft se ha convertido en la empresa dominante en los sistemas actuales de computadores, muy unido a los procesadores Intel [HTTP11].

- **Linux:** es un término empleado para referirse al núcleo o kernel libre similar a Unix, es usado con herramientas de sistema GNU. Su desarrollo, ha sido un gran ejemplo prominente de software libre, el código completo, puede ser redistribuido, modificado, y utilizado libremente por cualquiera bajo los términos de la GPL (Licencia Pública General de GNU) y otra serie de licencias libres. Aunque Linux (núcleo) es el sistema operativo, la parte fundamental de la interacción entre el núcleo y el usuario, se maneja con las herramientas del proyecto GNU o GNOME, la gran parte de la comunidad de usuarios, prefieren utilizar el término Linux, para referirse a la unión de ambos proyectos. Las distribuciones de Linux, se han adaptado según el cometido que representaban en cada momento y que cumplieran las necesidades de un determinado grupo de usuarios, algunas de ellas son conocidas por su uso en servidores y supercomputadoras, donde tiene la mayor cuota de mercado, con menor cuota es utilizado en las computadoras de escritorio, portátiles, teléfonos móviles, etc... [HTTP12].
- **Mac OS, Macintosh Operating System:** es un sistema operativo, creado por Apple, para la línea de computadoras Macintosh. Es conocido, por haber sido el primer sistema dirigido al público, con una interfaz gráfica que interactúa con el ratón, ventanas, iconos y menús. Apple no dio demasiada importancia a la existencia del sistema operativo, en los primeros años, procurando que la máquina fuera más agradable de cara al usuario, a diferencia de sistemas operativos como MS-DOS. Hill Atkinson, Jef Raskin y Andy Hertzfeld, eran el equipo de desarrollo del Mac Os, cuya primera versión de este, se remonta al 1984, su desarrollo se extiende progresivamente hasta la versión 9 lanzada en 1999. A partir de Mac Os X, el sistema es un derivado de Unix, que ha

mantenido la interfaz gráfica con gran mayoría de elementos de versiones anteriores [HTTP13].

En la Tabla 7 se muestra la comparativa de los diferentes sistemas operativos analizados.

Característica	Windows	Mac	Linux
Software	No Libre	No Libre	Libre
Idioma Español	√	√	√
Interfaz Gráfica de Usuario	Alta	Alta	Media
Seguridad	Baja	Alta	Alta
Estabilidad(Última versión)	Alta	Alta	Alta

Tabla 7 Sistemas Operativos

Se ha elegido Windows como sistema operativo, por su interfaz gráfica y por su facilidad de manejo.

2.3 Metodología empleada

La metodología empleada para el completo desarrollo del proyecto, ha sido el Proceso Unificado de Desarrollo, Unified Process (UP) en inglés. El Proceso Unificado es un proceso de desarrollo de software. Dicho proceso, es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. El UP está equilibrado por ser el producto final de tres décadas de desarrollo y uso práctico. Del mismo modo, se caracteriza por ser interactivo e incremental, dirigido por casos de uso y por estar centrado en la arquitectura [4].

La arquitectura del UP, en un sistema software descrito mediante diferentes vistas del sistema en construcción. El modo en que se organizan cada una de las fases depende de los casos de uso que se consideren necesarios para su descripción gráfica.

Los casos de uso capturan los requisitos funcionales, definiendo los contenidos en cada iteración de cara al usuario. Con esto, se pretende que se hagan casos de uso para cada uno de los pasos, que se presentan en el ciclo de vida del proyecto de software, por este motivo se visionan dirigidos a requisitos importantes para el usuario y los que pudieran ser relevantes.

En este ámbito, el UP está orientado hacia los riesgos, y hace que se definan todos los sean probables en el principio de cada fase. De esta forma, se puede establecer un orden de las fases, según el control de los principales riesgos que se han encontrado.

Uno de los aspectos importantes en el UP, es su característica incremental e interactiva. Desarrollar un producto de software, es una tarea que puede tener una magnitud de varios meses e incluso de años. Por ello, es importante dividir el trabajo en pequeñas partes o mini-proyectos. Los mini-proyectos se componen de interacciones, que son finalizadas con un incremento. Las iteraciones son referidas a pasos en el flujo de trabajo y los incrementos al crecimiento en el producto. Para ser eficaz, las iteraciones deben estar controladas, para ello, es necesario que sean elegidas individual o de forma colectiva y llevadas a cabo de una manera planeada.

Los desarrolladores de software, se basan en dos factores en las iteraciones: el primero, la iteración se trata con un grupo de casos de uso y amplían el uso que se le va a dar al producto, el segundo, se trata con los casos de usos que se consideren más relevantes. En cada iteración, estos desarrolladores de software, especifican e identifican, los casos de uso más importantes y diseñan guiándose por la arquitectura, implementando el diseño de componentes y verificando que estos cumplen los casos de uso. Si una iteración cumple su objetivo, el desarrollo continuo a la siguiente iteración, si ocurre el caso contrario, los desarrolladores de software deben revisar las decisiones que se tomaron y ver la situación desde un enfoque distinto.

Esta metodología llevada a cabo, contiene aspectos del Diseño Centrado en el Usuario (DCU), este diseño contiene un enfoque cuyo proceso está dirigido conteniendo información para las personas que van a hacer uso del producto. El DCU es un proceso cíclico, en el cual el usuario especifica y dirige las decisiones sobre el diseño donde la usabilidad es evaluada de forma iterativa, mejorándose constantemente. Principalmente DCU se basa en cuatro fases diferenciadas:

- **Identificar** los usuarios a los cuales está dirigido el producto, el uso que harán de este y en qué condiciones.
- **Especificar** los objetivos que deberán de cumplirse tanto del proveedor como del usuario.

- **Realizar** soluciones de diseño desde las primeras nociones conceptuales hasta la solución final.
- **Evaluar** las soluciones que se han dado en el diseño satisfaciendo y cumpliendo los requisitos o detectando aspectos incorrectos en la usabilidad.

Todas estas fases, se realizan con un objetivo, lograr la satisfacción de todos los usuarios con la calidad de uso de la aplicación desarrollada, creando interfaces que facilitan la consecución de sus objetivos y adaptando las tecnologías a las expectativas de los usuarios.

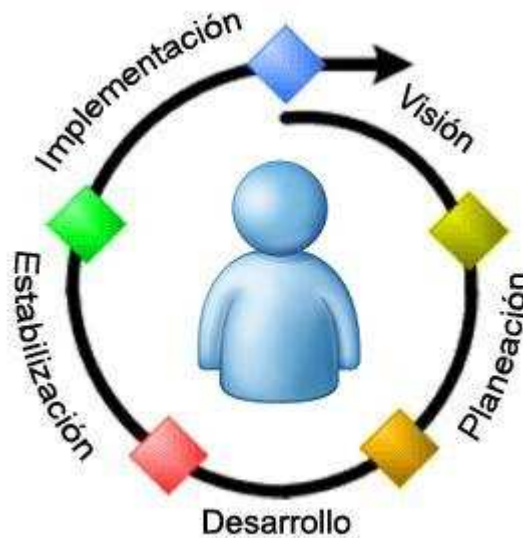


Figura 2. Diseño centrado en el usuario

Capítulo 3: Descripción Informática

3.1 Especificación

3.1.1 Análisis de requisitos

- **Requisitos Funcionales**

RF I. El programa identificará los principales rasgos de la personalidad del estudiante según el test de personalidad Big-Five-OCEAN [HTTP14].

RF II. El programa clasificará al estudiante según su puntuación en el test de Estilos de Aprendizaje Soloman-Felder [HTTP15].

RF III. Las preguntas almacenadas en la base de datos serán mostradas al profesor, para poder ser verificadas, modificadas o combinadas con los rasgos del estudiante.

RF IV. El estudiante deberá completar online los tests de Personalidad y Estilos de Aprendizaje.

RF V. El profesor podrá almacenar, modificar y eliminar las preguntas de la base de datos, preguntas que el programa transformará según la información resultado de los tests.

RF VI. El programa transformará las preguntas de la Base de Datos según la información de los tests de Personalidad y Estilos de Aprendizaje.

RF VII. El programa almacenará las nuevas preguntas en la Base de Datos.

RF VIII. El estudiante podrá revisar las preguntas generadas automáticamente según los distintos patrones y seleccionar aquella que considera que es más adecuada.

- **Requisitos No Funcionales**

RNF I. El programa debe cumplir los principios de interacción persona-ordenador visibilidad, consistencia y ser usable.

RNF II. La velocidad de respuesta debe ser inferior a un minuto en las iteraciones con los tests y de una hora en el caso de la transformación de las preguntas en la base de datos.

3.1.2 Casos de uso

Los diagramas de casos de uso representan las diferentes operaciones que debe realizar el sistema y la manera de actuar con su entorno, más concretamente, los usuarios y otras aplicaciones que interactúan con él. En nuestra aplicación, actúan dos tipos de actores diferentes, cada uno de ellos posee distintos casos de uso, como se puede observar en la Figura 3.

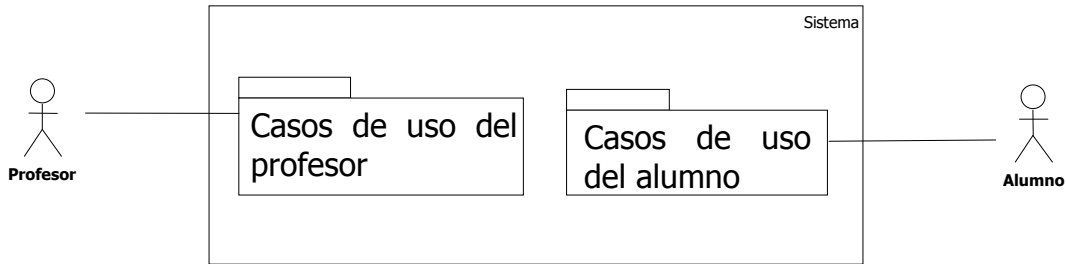


Figura 3. Diagrama de casos de uso de la aplicación

En el siguiente diagrama, podemos observar la descripción de los casos de uso del usuario **profesor**:

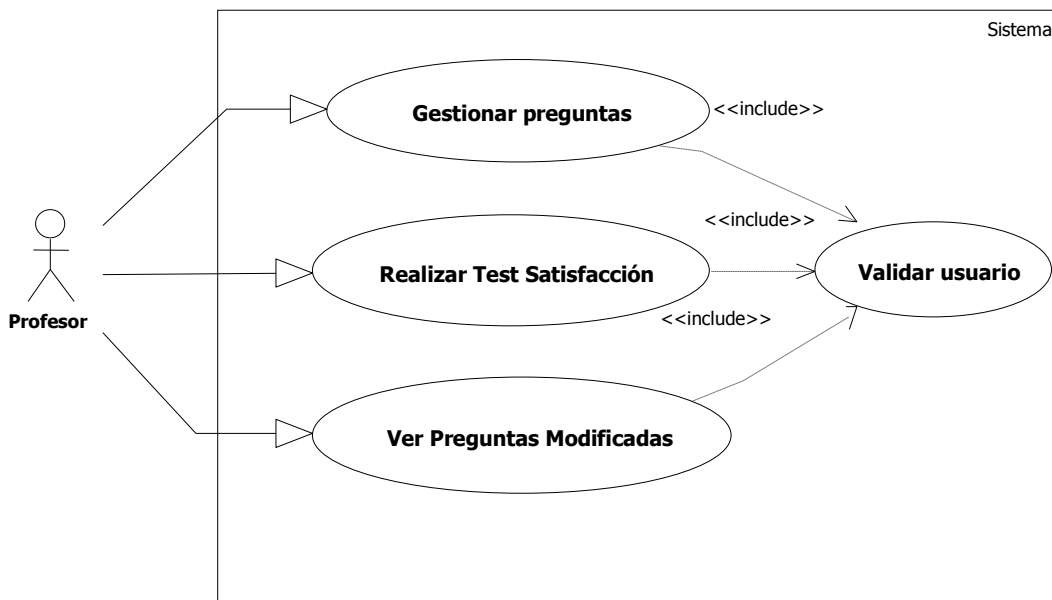


Figura 4. Diagrama de casos de uso del profesor

En los siguientes diagramas se dividen los casos de uso principales del usuario profesor:

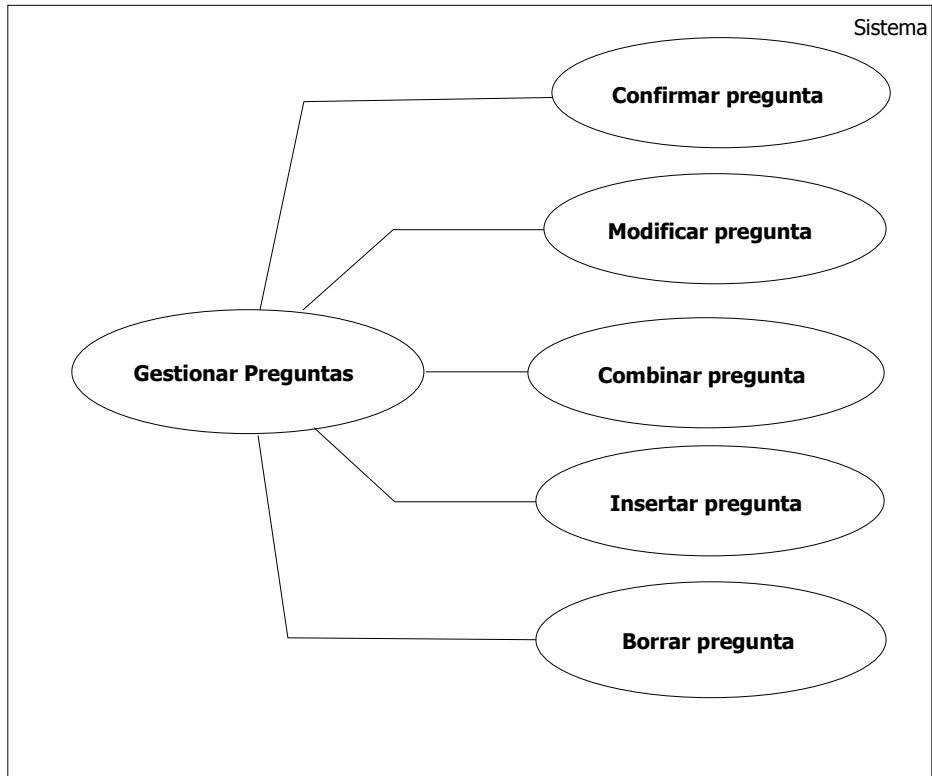


Figura 5. División del caso de uso Gestionar Preguntas

• **Caso de uso “Gestionar Preguntas”:**

Objetivo: Gestionar las preguntas.

Actores: Usuario profesor.

Descripción: Un profesor puede gestionar las adaptadas de cada estudiante.

Precondición: El profesor tiene que acceder a la aplicación.

Flujo de acciones:

- Confirmar pregunta para seguir con la siguiente.
- Modificar la pregunta.
- Combinar la pregunta con otra pregunta.
- Insertar una pregunta en la base de datos
- Borrar una pregunta de la base de datos

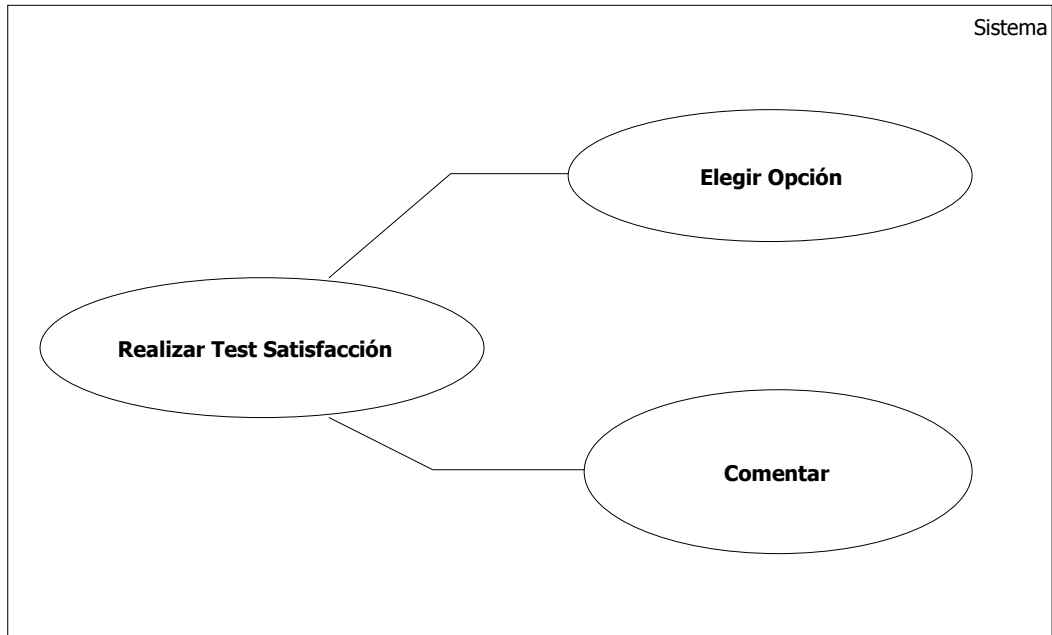


Figura 6. División del caso de uso Realizar test Satisfacción

• **Caso de uso “Realizar test Satisfacción”:**

Objetivo: Realizar el test de satisfacción.

Actores: Usuario profesor.

Descripción: Un profesor puede dar su opinión sobre la aplicación.

Precondición: El profesor tiene que haber gestionado las preguntas.

Flujo de acciones:

- Elegir una opción del test.
- Comentar aspectos que el profesor considere relevantes.

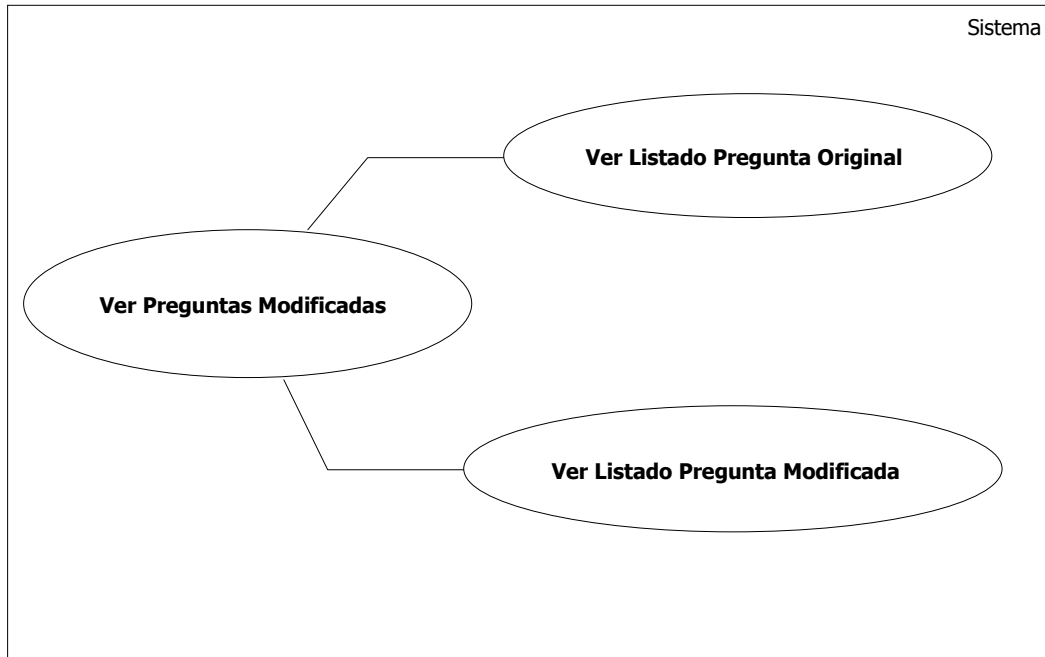


Figura 7. División del caso de uso Ver Preguntas Modificadas

• **Caso de uso “Ver Preguntas Modificadas”:**

Objetivo: Ver las preguntas que se han modificado y confirmado.

Actores: Usuario profesor.

Descripción: Un profesor puede ver el resultado de las modificaciones de las preguntas.

Precondición: El profesor tiene que haber gestionado las preguntas.

Flujo de acciones:

- Ver el listado de las preguntas en su estado original.
- Ver el listado de las preguntas en su estado modificado.

En el siguiente diagrama, podemos observar la descripción de los casos de uso del usuario **alumno**:

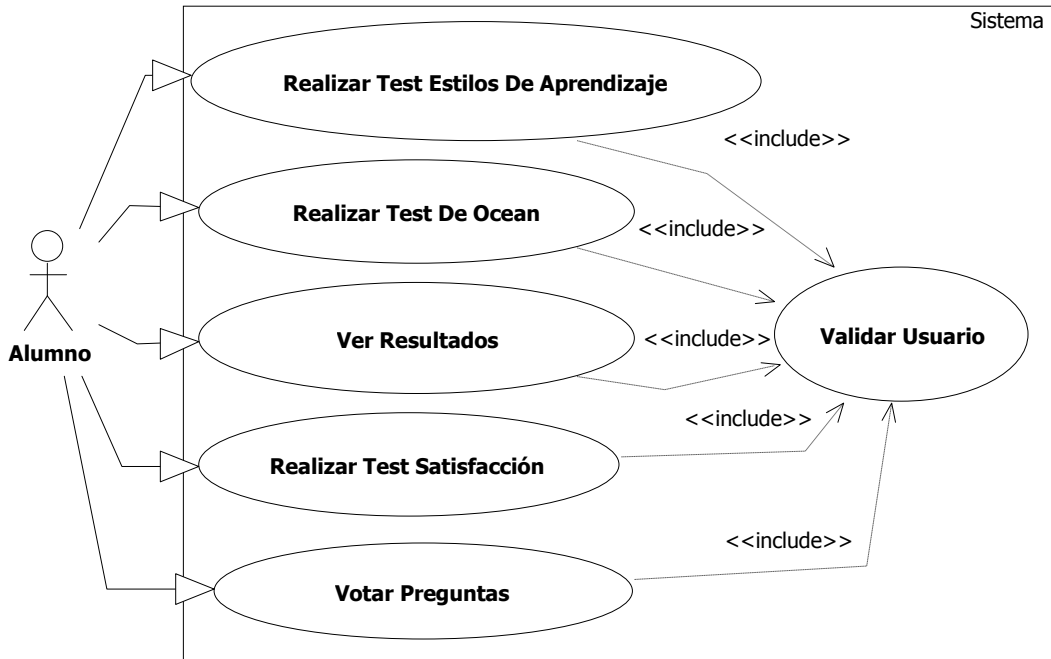


Figura 8. Diagrama de casos de uso del usuario Alumno

En los siguientes diagramas se dividen los casos de uso principales del usuario alumno:

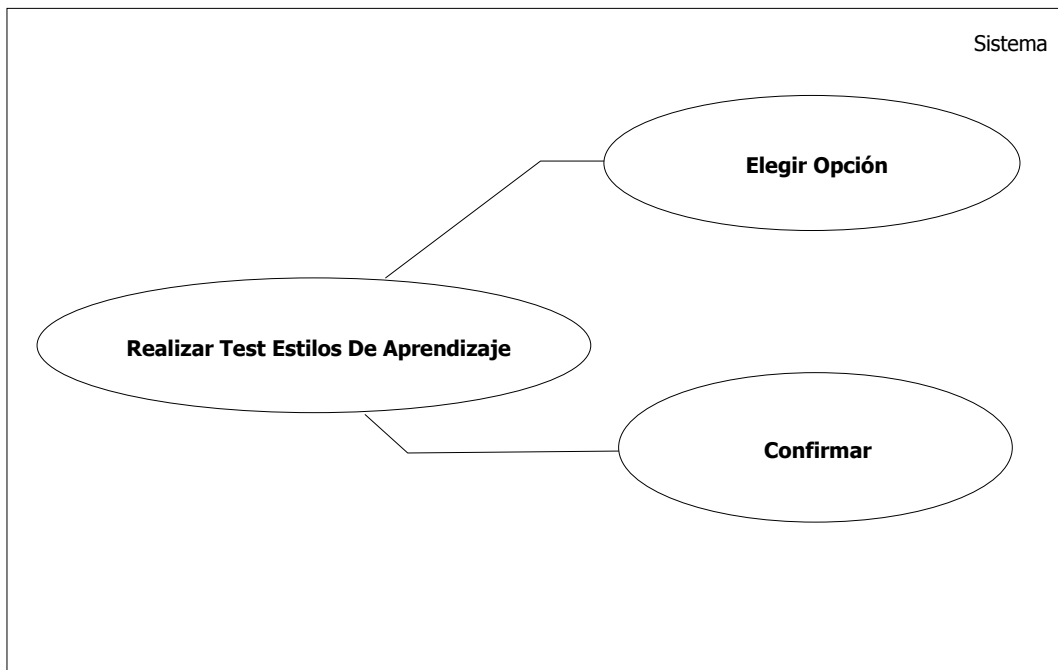


Figura 9. División del caso de uso Realizar test Estilos De Aprendizaje

• **Caso de uso “Realizar test Estilos De Aprendizaje”:**

Objetivo: Responder el test Estilos de Aprendizaje.

Actores: Usuario estudiante.

Descripción: El estudiante elige las opciones del test Estilos de Aprendizaje.

Precondición: El estudiante tiene que acceder a la aplicación.

Flujo de acciones:

- Elegir una opción del test.
- Confirmar las opciones elegidas del test.

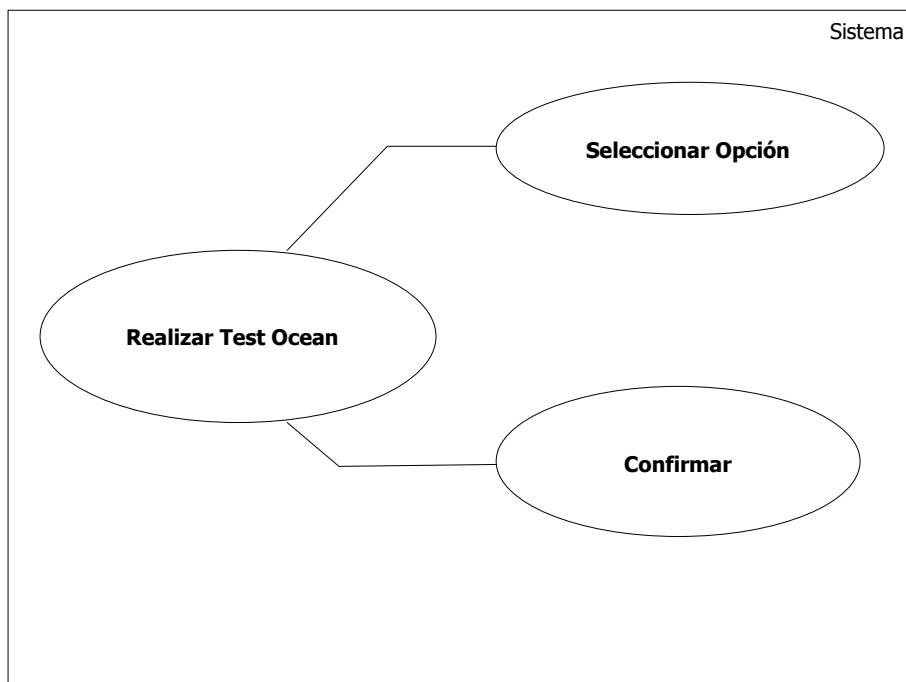


Figura 10. División del caso de uso Realizar test Ocean

• **Caso de uso “Realizar test Estilos Ocean”:**

Objetivo: Responder el test Ocean.

Actores: Usuario estudiante.

Descripción: El estudiante elige las opciones del test Ocean.

Precondición: El estudiante tiene que haber completado el test Estilos de Aprendizaje.

Flujo de acciones:

- Elegir una opción del test.
- Confirmar las opciones elegidas del test.

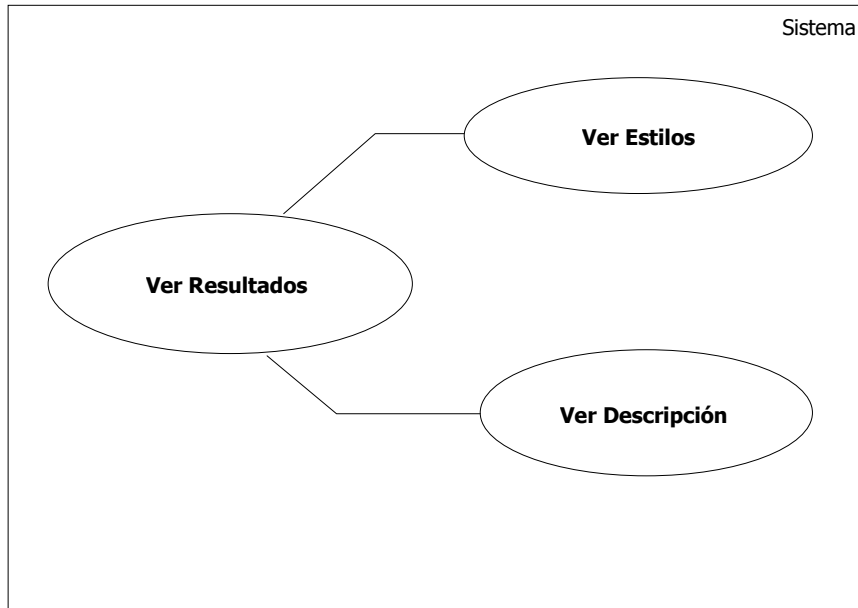


Figura 11. División del caso de uso Ver Resultados

• **Caso de uso “Ver Resultados”:**

Objetivo: Ver los resultados obtenidos a partir de los tests.

Actores: Usuario estudiante.

Descripción: El estudiante ve sus Estilos de Aprendizaje y sus Tipos de Personalidad.

Precondición: El estudiante tiene que haber completado los tests.

Flujo de acciones:

- Ver Estilos.
- Ver una descripción de cada estilo.

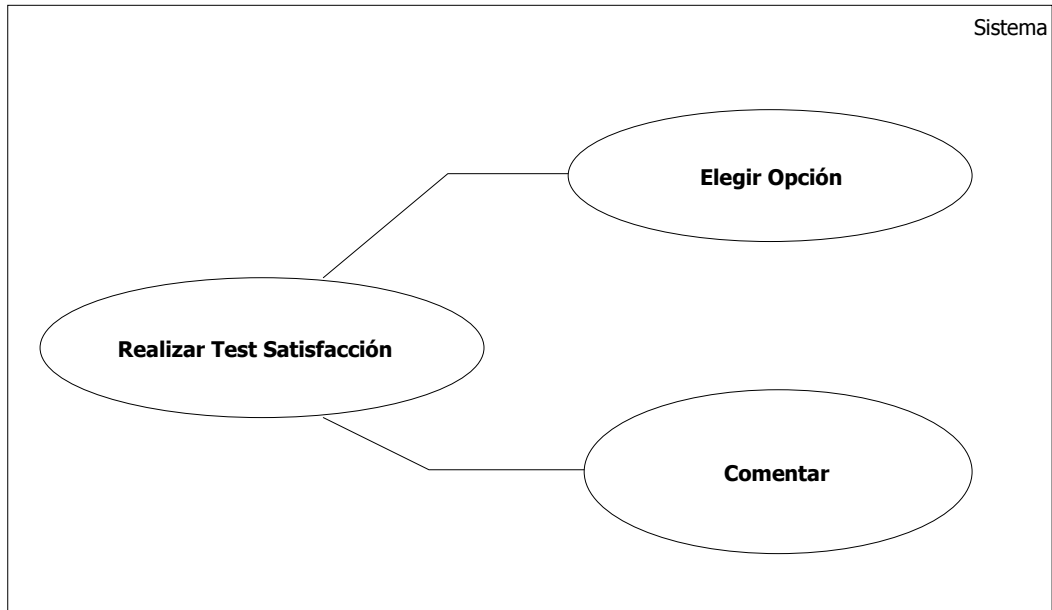


Figura 12. División del caso de uso Realizar test Satisfacción

• **Caso de uso “Realizar test Satisfacción”:**

Objetivo: Realizar el test de satisfacción.

Actores: Usuario estudiante.

Descripción: Un estudiante puede dar su opinión sobre los tests, los resultados obtenidos o aspectos que este considere de importancia.

Precondición: El estudiante tiene que haber visto los resultados de los tests.

Flujo de acciones:

- Elegir una opción del test.
- Comentar aspectos que el estudiante considere relevantes.

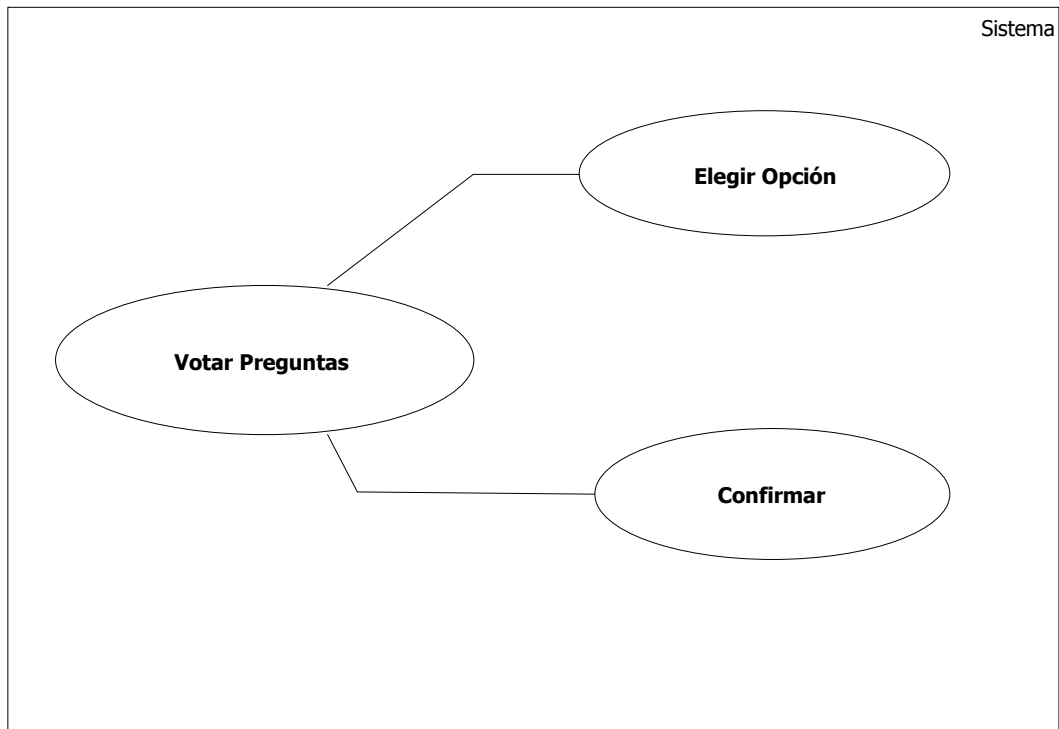


Figura 13. División del caso de uso Votar Preguntas

• **Caso de uso “Votar Preguntas”:**

Objetivo: Votar las preguntas adaptadas.

Actores: Usuario estudiante.

Descripción: Un estudiante puede votar las preguntas que considere que entiende mejor.

Precondición: El estudiante tiene que haber visto los resultados de los tests y haber cerrado sesión.

Flujo de acciones:

- Elegir la pregunta que desea votar.
- Confirmar la votación elegida.

3.2 Arquitectura de alto nivel

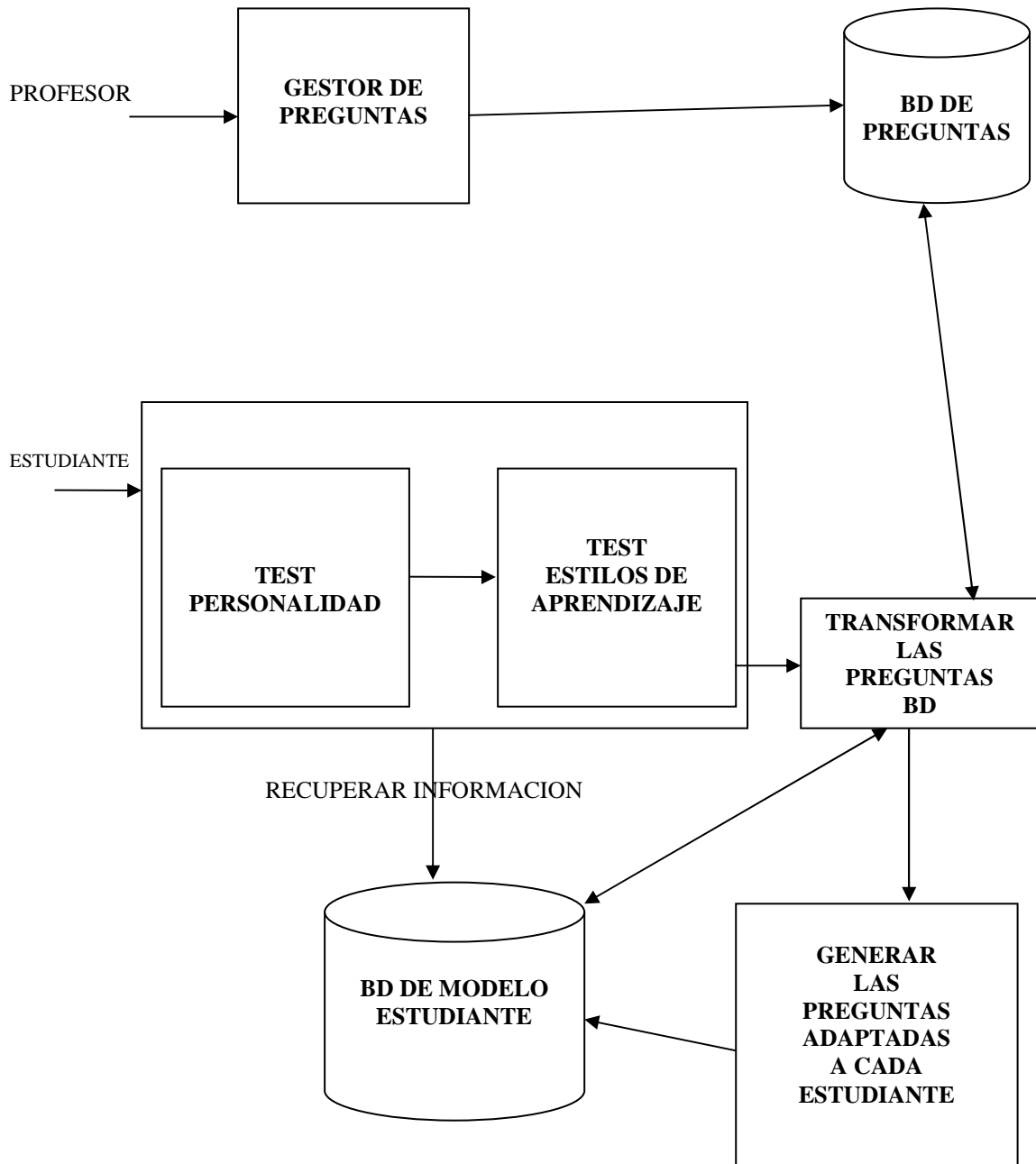


Figura 14. Arquitectura de alto nivel

Como se puede observar en la Figura 14, en Flayer los dos roles principales son el del profesor y el del estudiante. En particular:

1. Los profesores **gestionan** las preguntas: entran en la aplicación para introducir, modificar, confirmar o eliminar las preguntas de su asignatura. Estas son las preguntas que serán adaptadas a cada estilo de aprendizaje y tipo de personalidad para cada estudiante.
2. Los estudiantes **reciben** las preguntas: entran en la aplicación donde realizarán el test de “Big Five” o de cinco grandes rasgos y el test de “Felder” o test de Estilos de Aprendizaje. La información extraída del análisis de dichos tests se almacenará en la base de datos, al igual que las preguntas adaptadas según los Estilos de Aprendizaje y tipo de personalidad para cada estudiante.
3. **Comprobar** nivel satisfacción: esta comprobación se basa en las preguntas obtenidas automáticamente en la aplicación, dándole al estudiante la posibilidad de que vote las preguntas que entiende mejor, dichas estadísticas se almacenan en la base de datos para su posterior análisis.

Se pueden distinguir dos bases de datos: la base de datos de modelo de estudiante donde se almacena las preguntas que genera la aplicación (Willow) y la base de datos de preguntas es aquella donde se almacena las preguntas originales (test),

3.3 Diagrama E/R

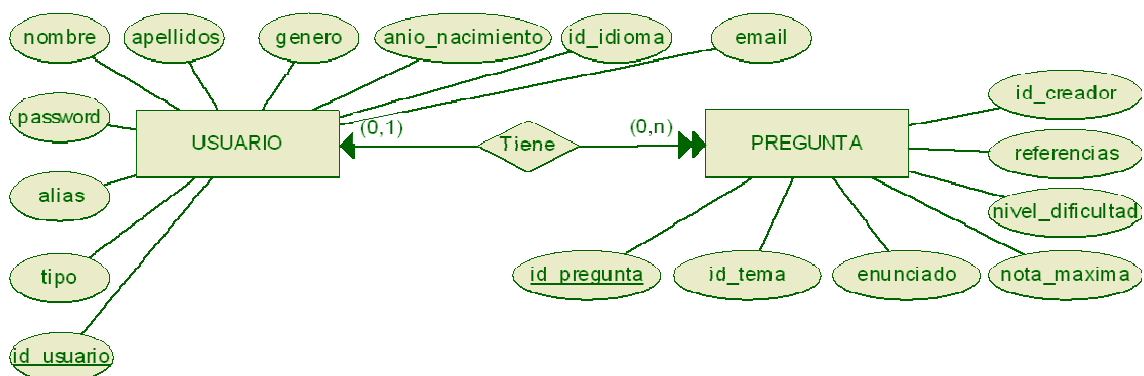


Figura 15. Diagrama E/R base de datos Willow

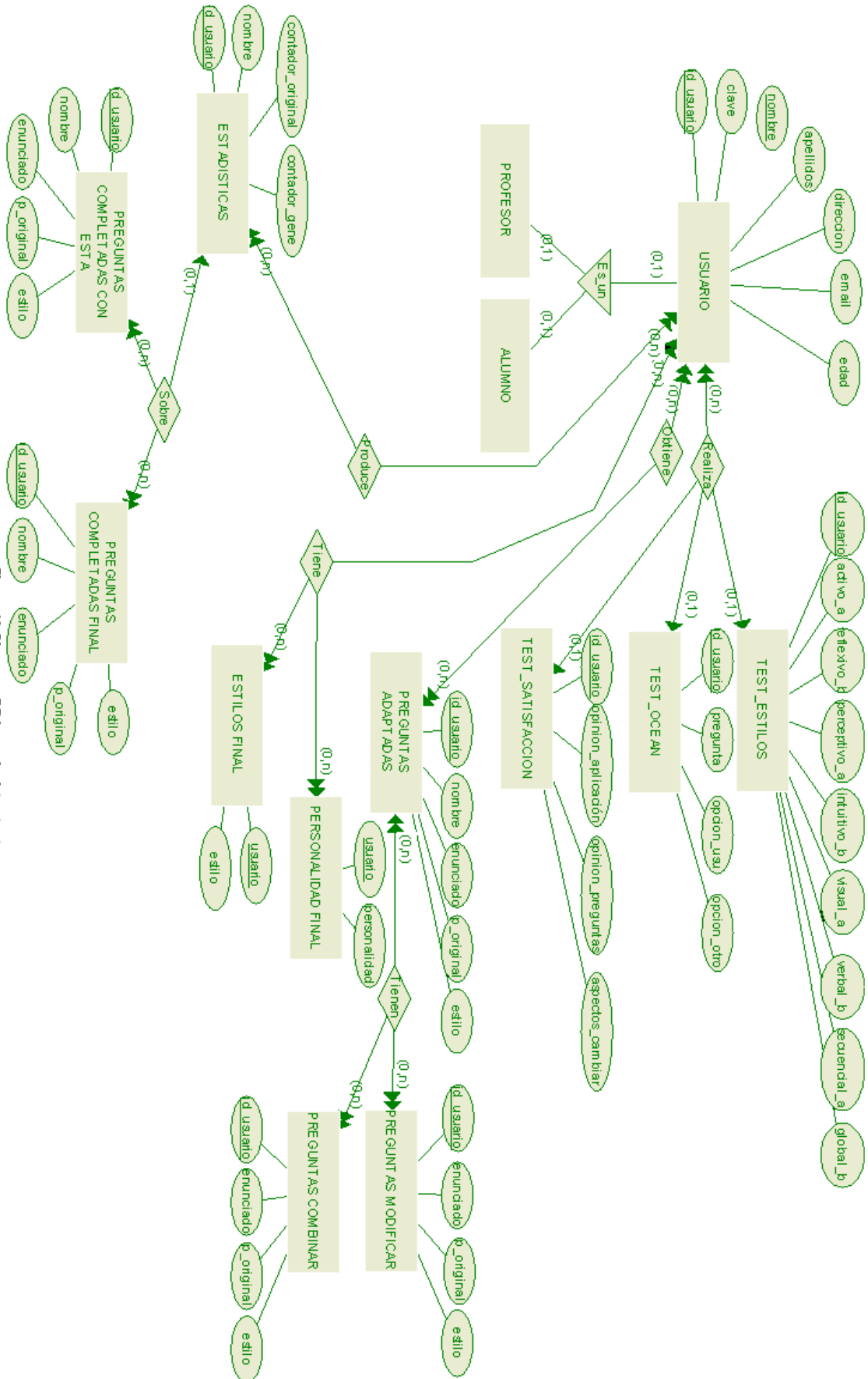
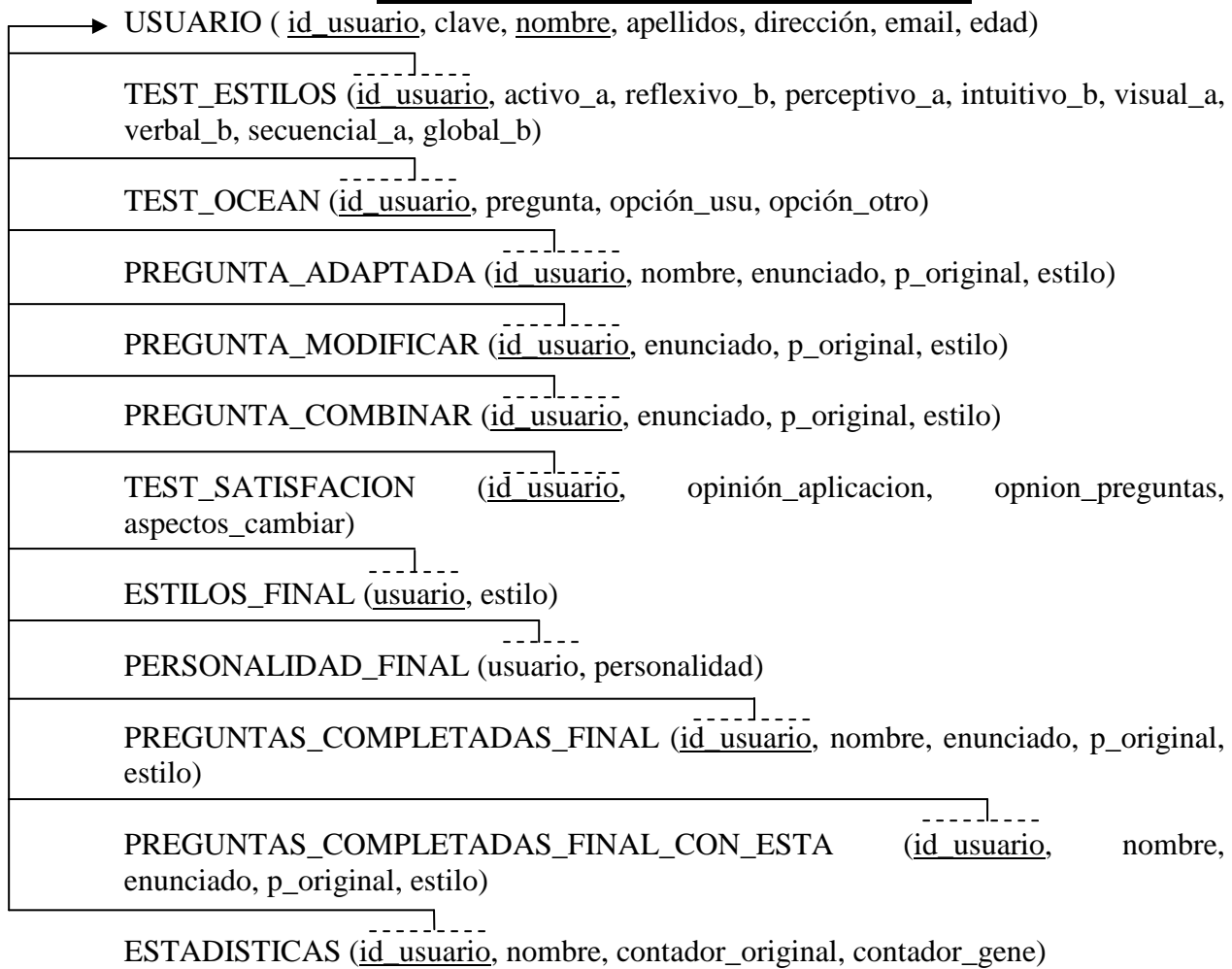
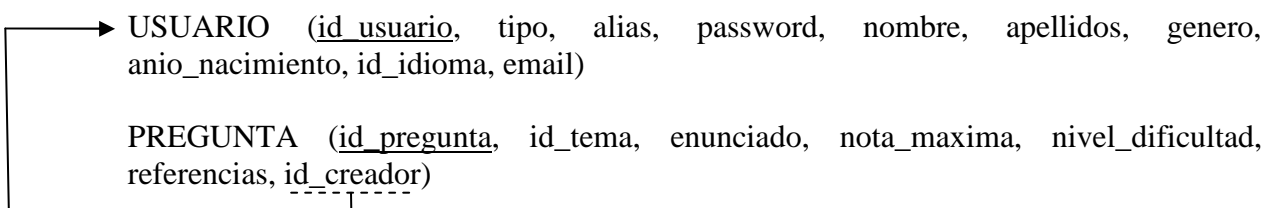


Figura 16. Diagrama ER base de datos test

Modelo Relacional (Base de datos test)



Modelo Relacional (Base de datos estudiante)



La base de datos que se explica a continuación, almacena la información de las preguntas generadas por la aplicación, así como información relacionada con el estudiante.

Seguidamente se describe cada una de las tablas utilizadas en base de datos **test**:

USUARIO: tabla donde se almacenan todos los usuario que inician sesión en la aplicación, constará de todos sus datos personales (id_usuario, clave, nombre, apellidos, dirección, email, edad), siendo id_usuario clave primaria, campos que la componen:

id_usuario	clave	nombre	apellidos	direccion	email	edad
64	pedro123	pedro	NULL	NULL	NULL	NULL

Figura 17. Tabla Usuario de base de datos test

TEST_ESTILOS: tabla donde se almacenan los resultados mientras el usuario realiza el Test de Estilos de Aprendizaje, constará de las elecciones que vaya eligiendo durante el test (id_usuario, activo_a, reflexivo_b, perceptivo_a, intuitivo_b, visual_a, verbal_b, secuencial_a, global_b), campos que la componen:

id_usuario	activo_a	reflexivo_b	perceptivo_a	intuitivo_b	visual_a	verbal_b	secuencial_a	global_b
pedro	1	0	1	0	1	0	1	0

Figura 18. Tabla Test_estilos de base de datos test

TEST_OCEAN: tabla donde se almacenan los resultados mientras el usuario realiza el test de Ocean, constará de las elecciones que vaya eligiendo durante el test (id_usuario, pregunta, opción_usu, opción_otro), siendo opción_usu la opción que pulsa el usuario y opción_otro la opción que pulsaría un acompañante si estuviese con el estudiante, campos que la componen:

id_usuario	pregunta	opcion_usu	opcion_otro
0	1	1	0
0	2	4	0
0	3	4	0
0	4	2	0

Figura 19. Tabla test_ocean de base de datos test

PREGUNTA_ADAPTADA: tabla donde se almacenan las preguntas generadas por la aplicación para cada usuario según su tipo de personalidad y sus estilos de aprendizaje, constará de las preguntas original, adaptada y estilo aplicado (id_usuario, nombre, enunciado, p_original, estilo), siendo id_usuario clave primaria, campos que la componen:

id_usuario	nombre	enunciado	p_original	estilo
3434	pedro	Qué dirías tú, para todo el mundo, sobre ¿Los atributos h...	¿Los atributos heredados se suelen usar en a...	extroversión
3435	pedro	Intentar cambiar el contexto de la pregunta, pulse el botó...	¿Los atributos heredados se suelen usar en a...	abierto de mente
3433	pedro	A ver si te sabes esto, ¿Los atributos heredados se suele...	¿Los atributos heredados se suelen usar en a...	antipatía
3432	pedro	Si depende de tí, que tu amigo apruebe, Cómo responde...	¿Los atributos heredados se suelen usar en a...	responsabilidad
3431	pedro	Si te plantearan la siguiente pregunta, sin obligación de r...	¿Los atributos heredados se suelen usar en a...	neuroticismo

Figura 20. Tabla Pregunta_adaptada de base de datos test

PREGUNTA_MODIFICAR: tabla donde se almacenan las preguntas que se van a modificar, constará de las preguntas original, adaptada y estilo aplicado (id_usuario, enunciado, p_original, estilo), siendo id_usuario clave primaria, campos que la componen:

id_usuario	nombre	enunciado	p_original	estilo
1	pedro	¿Qué dirías tú, para todo el mundo, sobre ¿Los atrib...	¿Los atributos heredados se suelen usar en análisis asc...	extroversion

Figura 21. Tabla Pregunta_modificar de base de datos test

PREGUNTA_COMBINAR: tabla donde se almacenan las preguntas que se van a combinar, constará de las preguntas original, adaptada y estilo aplicado (id_usuario, enunciado, p_original, estilo), siendo id_usuario clave primaria, campos que la componen:

id_usuario	nombre	enunciado	p_original	estilo
1	pedro	Si depende de tí, que tu amigo apruebe...	¿Los atributos heredados se suelen usar...	responsabilidad

Figura 22. Tabla Pregunta_combinar de base de datos test

TEST_SATISFACION: tabla donde se almacenan los resultados obtenidos del test de Satisfacción realizado al usuario, constará de dos preguntas sobre la aplicación y un campo de descripción (id_usuario, opinión_aplicacion, opinion_preguntas, aspectos_cambiar), siendo id_usuario clave primaria, campos que la componen:

id_usuario	opinion_aplicacion	opinion_preguntas	aspectos_cambiar
10	Mala	En algunas ocasiones	Quitad los chistes que son muy malos
11	Interesante	En algunas ocasiones	En el primer test, una opción intermedia. Los chistes, muy buenos.
12	Interesante	En algunas ocasiones	Una descripción final más personalizada. Tal vez pondría más preguntas por pantalla.
13	Buena	Sí	Pondría mejores chistes, porque hay algunos malisimos. Eso si el de la bici es muy bueno xD
14	Interesante	Sí	Más preguntas acerca de un mismo tema ya que en ocasiones es necesario especificar má...
15	Buena	En algunas ocasiones	

Figura 23. Tabla test_satisfacion de base de datos test

ESTILOS_FINAL: tabla donde se almacenan los estilos finales de cada usuario, una vez que ha realizado el test de Estilos de Aprendizaje, constará de los estilos obtenidos (usuario, estilo), campos que la componen:

usuario	estilo
alberto	reflexivo
alberto	intuitivo
alberto	verbal
alberto	global

Figura 24. Tabla Estilos_final de base de datos test

PERSONALIDAD_FINAL: tabla donde se almacenan el tipo de personalidad de cada usuario, una vez que ha realizado el test de Ocean, constará de los tipos de personalidad obtenidos (usuario, personalidad), campos que la componen:

usuario	personalidad
alberto	extroversión
alberto	cordialidad
alberto	desorganización
alberto	neuroticismo
alberto	abierto de mente

Figura 25. Tabla Personalidad_final de base de datos test

PREGUNTAS_COMPLETADAS_FINAL: tabla donde se almacenan las preguntas completadas de cada usuario, una vez han sido confirmadas por el profesor, constará de las preguntas original, adaptada y el estilo aplicado (id_usuario, nombre, enunciado, p_original, estilo), siendo id_usuario clave primaria, campos que la componen:

id_usuario	nombre	enunciado	p_original	estilo
390	sergio	En el contexto de los lenguajes de programación, ¿Los atrib...	¿Los atributos heredados se suelen usar en análisis as...	abierto de mente
391	sergio	Si te plantearan la siguiente pregunta, sin obligación de res...	¿Los atributos heredados se suelen usar en análisis as...	neuroticismo
392	sergio	A ver si te sabes esto, ¿Los atributos heredados se suelen...	¿Los atributos heredados se suelen usar en análisis as...	antipatía
393	sergio	Si depende de tí, que tu amigo apruebe, Cómo responden...	¿Los atributos heredados se suelen usar en análisis as...	responsabilidad
394	sergio	En el contexto de los lenguajes de programación, ¿Los atrib...	¿Los atributos sintetizados se suelen usar en análisis a...	abierto de mente
395	sergio	Qué dirías tú, para todo el mundo, sobre ¿Los atributos heré...	¿Los atributos heredados se suelen usar en análisis as...	extroversión
396	sergio	Si te plantearan la siguiente pregunta, sin obligación de res...	¿Los atributos sintetizados se suelen usar en análisis a...	neuroticismo
397	sergio	Seguro que no me puedes responder ¿Los atributos sintetiz...	¿Los atributos sintetizados se suelen usar en análisis a...	antipatía

Figura 26. Tabla Preguntas_completadas_final de base de datos test

PREGUNTAS_COMPLETADAS_FINAL_CON_ESTA: tabla donde se almacenan las preguntas completadas de cada usuario, cuando ya las han votado, constará de las preguntas original, adaptada y el estilo aplicado (id_usuario, nombre, enunciado, p_original, estilo), siendo id_usuario clave primaria, campos que la componen:

id_us...	nombre	enunciado	p_original	estilo
1	pruebas1	¿Cómo ves un compilador?	¿Qué es un compilador?	perceptivo
2	pruebas1	Háblame de compilador	¿Qué es un compilador?	activo
3	pruebas1	Cómo concibesUna expresión regular sir...	Una expresión regular sirve para:¶ a) Re...	perceptivo
4	pruebas1	¿Qué tipo de análisis se apoya en el uso...	Una gramática en formato BNF sirve par...	activo
5	pruebas1	¿En qué tipo de análisis ves que es útil ú...	Una gramática en formato BNF sirve par...	perceptivo
6	pruebas1	¿Qué es el código máquina?	¿Qué es el código máquina?	activo
7	pruebas1	Visualizar código máquina, ¿qué pinta tie...	¿Qué es el código máquina?	perceptivo

Figura 27. Tabla Preguntas_completadas_final_con_esta de base de datos test

ESTADISTICAS: tabla donde se almacenan los resultados obtenidos de las votaciones de los usuario, sobre las preguntas que mejor entienden, constará de la votación de la pregunta original o de la pregunta generada (id_usuario, nombre, contador_original, contador_gene), campos que la componen:

id_usuario	nombre	contador_original	contador_gene
3	bea	2	6

Figura 28. Tabla Estadísticas de base de datos test

La siguiente base de datos que se explica a continuación, almacena la información de las preguntas originales, se ha utilizado la estructura de la base de datos de Willow aunque se hubiera podido integrar con cualquier otra base de datos.

Seguidamente, se describe cada una de las tablas utilizadas en base de datos **Willow**:

USUARIO: tabla donde se almacenan los datos de los usuario, constará de sus datos personales (id_usuario, tipo, alias, password, nombre, apellidos, genero, anio_nacimiento, id_idioma, email), siendo id_usuario clave primaria, campos que la componen:

id_usuario	tipo	alias	password	nombre	apellidos	genero	anio_n...	id_idioma	email
1	estudiante	alberto	alberto123	Alberto	Redondo	hombre	1980	0	a.redondo.h@alumnos.urjc.es
3	estudiante	paco	paco123	Paco	Perez	hombre	1980	0	paco_dd@urjc.es

Figura 29. Tabla Usuario de base de datos Willow

PREGUNTA: tabla donde se almacenan las preguntas originales introducidas por el profesor, constara del enunciado y algunos aspectos relevantes (id_pregunta, id_tema, enunciado, nota_maxima, nivel_dificultad, referencias, id_creador), siendo id_pregunta clave primaria, campos que la componen:

id_pregunta	id_tema	enunciado	nota_maxima	nivel_dificultad	referencias	id_creador
1730	61	¿Qué es un compilador?	1	2	12	1
1729	61	Una expresión regular sirve para: a) Re...	1	2	12	1
1728	61	Una gramática en formato BNF sirve par...	1	2	12	1
1731	61	¿Qué es el código máquina?	1	2	12	1
1732	61	¿Qué es un intérprete?	1	2	12	1
1733	61	¿cuál es la principal diferencia entre un b...	1	2	12	1
1734	61	¿Cuáles son los dos grandes component...	1	2	12	1

Figura 30. Tabla Pregunta de base de datos Willow

3.4 Algoritmo utilizado

En esta sección se describe algoritmo principal de la aplicación para generar las preguntas adaptada a cada Estilo de Aprendizaje y tipo de personalidad:

El algoritmo, como se muestra a continuación, aplica una serie de patrones de forma aleatoria para cada patrón se tienen varias preguntas, esto hace que no se parezca la pregunta anterior a la misma pregunta actual. Por ejemplo: si el patrón es Perceptivo y la pregunta original es “¿Qué es una manzana?”, la aplicación elegiría una de las tres siguientes frases:

- Qué insinúas sobre, ¿Qué es una manzana?
- Cómo ves, ¿Qué es una manzana?
- Como concibes, ¿Qué es una manzana?

Para la creación de los patrones se ha hecho una análisis de cada Estilo de Aprendizaje y tipo de personalidad y se ha creado un patrón para cada uno de ellos exceptuando el Estilo de Personalidad Secuencial y el Estilo de Personalidad Global, en los cuales la aplicación muestra de forma ordenada las preguntas y de forma esquema general respectivamente en el módulo de estadísticas.

Los patrones se aplican a la pregunta original que fue introducida por el profesor, para cada tipo de personalidad y cada estilo de aprendizaje del estudiante una vez obtenidos los resultados del test de “Big Five” o de cinco grandes rasgos y el test de “Felder” o test de Estilos de Aprendizaje. Básicamente el algoritmo consiste en:

- Para Estilos de Aprendizaje:

MIENTRAS haya preguntas originales{

P=pregunta_original;

MIENTRAS haya estilos_aprendizaje{

E=coger_estilo;

SI e=ACTIVO {

X=patrón en plantilla

SI patrón entonces

Pregunta_nueva=**“Hablame de X”**

SI no patrón entonces

Pregunta_nueva=**“Deberia acoertar la frase”**

}

```
SI e=PASIVO {
    Numero=numero_aleatorio();
    Switch(numero){
        num1: Pregunta_nueva="Piensa, " + pregunta_original.
        num2: Pregunta_nueva="Reflexiona sobre, " + pregunta_original.
        num3: Pregunta_nueva="Tomate un tiempo y dime, " + pregunta
                _original.
        Num4: Pregunta_nueva="Mira, "+ pregunta_original
    }
}
```

```
SI e=PERCEPTIVO{
    Numero=numero_aleatorio();
    Switch(numero){
        num1: Pregunta_nueva=" Que insinúas sobre, " + pregunta_original.
        num2: Pregunta_nueva="Cómo ves, " + pregunta_original.
        num3: Pregunta_nueva="Cómo concibes, " + pregunta_original.
    }
}
```

```
SI e=INTUITIVO{
    Numero=numero_aleatorio();
    Switch(numero){
        num1: Pregunta_nueva="Evidente que, " + pregunta_original.
        num2: Pregunta_nueva="Claro que, " + pregunta_original.
        num3: Pregunta_nueva="Qué es cierto, " + pregunta_original.
    }
}
```

```
SI e=VISUAL{
    num1:Pregunta_nueva="Imagina, por un momento, " + pregunta_original.
}
```

```
SI e=VERBAL{
    Numero=numero_aleatorio();
    Switch(numero){
        num1: Pregunta_nueva="Escríbenos, " + pregunta_original.
        num2: Pregunta_nueva="Comenta, " + pregunta_original.
    }
}
```

```
        num3: Pregunta_nueva="Explica," + pregunta_original.
    }
}
SI e=SECUENCIAL {
    posteriormente mostrar preguntas ordenadas
}
SI e=GLOBAL {
    posteriormente mostrar preguntas esquemática general
}
Almacenar pregunta_nueva
} //final del mientras haya estilos
o Para los siguientes tipos de personalidad, se han aplicado estos patrones:
MIENTRAS haya tipos de personalidad{
    T_p=coger_tipo_personalidad;
    SI t_p=EXTROVERSION {
        Pregunta_nueva="Qué dirías tú, para todo el mundo sobre," +
                                pregunta_original.
    }
    SI t_p=INTROVERSION {
        Pregunta_nueva="Para ti," + pregunta_original.
    }
    SI t_p=CORDIALIDAD {
        Pregunta_nueva="Cómo creería usted," + pregunta_original.
    }
    SI t_p=ANTIPATIA{
        Numero=numero_aleatorio();
        Switch(numero){
            num1: Pregunta_nueva="A ver si te sabes esto," + pregunta_original.
            num2: Pregunta_nueva="Seguro que no me puedes responder" +
                                pregunta_original.
        }
    }
}
SI t_p=RESPONSABILIDAD{
    Numero=numero_aleatorio();
```

```
Switch(numero){
    num1: Pregunta_nueva="Si tuvieses que explicárselo a un amigo, Cómo
        responderías a " + pregunta_original.
    num2: Pregunta_nueva="Si depende de ti que tu amigo apruebe, Cómo
        le responderías a esta pregunta" + pregunta_original.
}
}
SI t_p=DESORGANIZACION{
    X=coger_palabra_clave_pregunta;
    Numero=numero_aleatorio();
    Switch(numero){
        num1: Pregunta_nueva="X, ¿Qué es?"
        num2: Pregunta_nueva="X, ¿Cuáles son?"
        num3: Pregunta_nueva="X, ¿Qué tipo es?"
    }
}
SI t_p=ESTABILIDAD EMOCIONAL{
    Pregunta_nueva="Intentar cambiar el contexto de la pregunta";
}
SI t_p=NEUROTICISMO{
    Pregunta_nueva="Si te plantearan la siguiente pregunta, sin obligación de
        responderla, como responderías a " + pregunta original.
}
SI t_p=ABIERTO DE MENTE{
    Pregunta_nueva="Intentar cambiar el contexto de la pregunta";
}
SI t_p=CONVENCIONALISMO{
    Numero=numero_aleatorio();
    Switch(numero){
        num1: Pregunta_nueva="Según conciben los españoles informáticos" +
            pregunta original.
        num2: Pregunta_nueva="Según las normas educativas españolas" +
            pregunta original.
    }
}
```



```
}  
Almacenar_pregunta_nueva;  
} //final del mientras haya tipos de personalidad  
} //final del mientras haya preguntas originales
```

3.5 Descripción de los módulos

En esta sección, se explican los principales módulos de los que se compone Flayer:

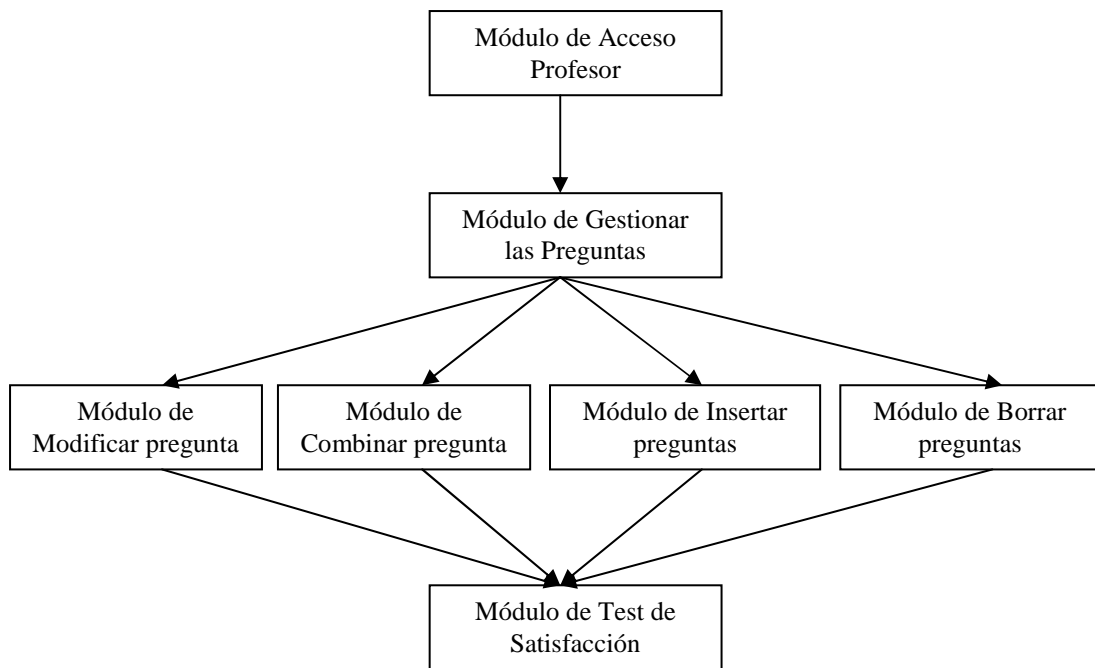


Figura 31. Diagrama de los módulos profesor

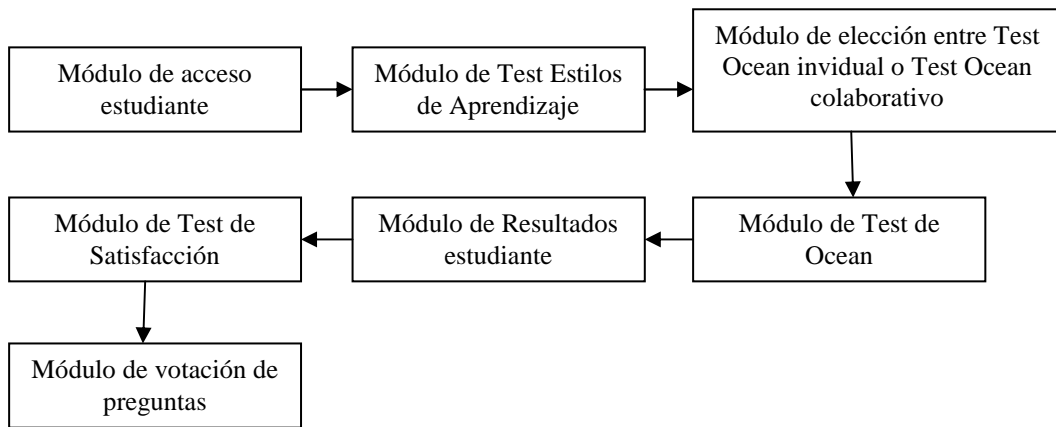


Figura 32. Diagrama de los módulos estudiante

- **Módulo de acceso profesor:**

En este módulo, denominado **profesor.jsp** el profesor introduce su usuario y contraseña, para validar sus datos con los que se encuentran en la base de datos, el esquema general seria:

```
Comprobamos_existencia;
Comprobamos si hay preguntas que modificar;
SI datos_correctos{
    Redireccionamos_a pregunta_generada_2.jsp (gestionador de preguntas)
}SI NO{
    Mostrar mensaje explicando el suceso;
}
```

- **Módulo de acceso estudiante:**

En este módulo, denominado **estudiante.jsp**, el estudiante introduce su usuario y contraseña, para validar sus datos con los que se encuentran en la base de datos, el esquema general seria:

```
Comprobamos_datos_estudiante;
SI datos_correctos {
    SI NO ha hecho el test_estilos{
        Redireccionamos a test_estilos.jsp
    }
    SI ha hecho test_estilos Y NO test_ocean {
        Redireccionamos a testEstilosFinalizado.jsp
    }
    Si ha hecho los test{
        Redireccionamos a resultadosEstudianteFinal.jsp
    }
}SI NO{
    Mostrar mensaje del suceso
}
```

- **Módulo de Test Estilos de Aprendizaje**

En este módulo, se muestran las páginas del test de Estilos de Aprendizaje, con un máximo de 4 preguntas por página, para facilitar la interacción con el estudiante. El esquema general de **test_estilos.jsp..test_estilos14.jsp** sería:

```
SI estudiante ha pulsado siguiente{
    Almacenar_opciones_elegidas_en_la_base_de_datos;
    Redireccionar a la siguiente hoja de test; (test_estilos.jsp2..test_estilos14.jsp)
}
```

- **Módulo de elección entre test Ocean individual o test Ocean colaborativo:**

En este módulo, se da la opción al estudiante de que elija si desea realizar el siguiente test de una forma individual, es decir, solamente él interactuando con la aplicación o con un acompañante, que de esta manera habría dos usuarios, él que daría la opinión de si mismo y el acompañante que daría una segunda opinión, de esta forma se intenta conseguir que la información sea lo más objetiva posible. El esquema general de **testEstilosFinalizado.jsp** sería:

```
Si estudiante pulsa individual{
    Redireccionamos a test_ocean_indi.jsp
}SI estudiante pulsa de forma colaborativa{
    Redireccionamos a test_ocean.jsp
}SI estudiante pulsa cerrar sesión {
    borrarDatosSesionEstudiante;
    Redireccionar a estudiante.jsp
}
```

- **Módulo de test de Ocean:**

En este módulo, se muestran las páginas del test de Ocean, con un máximo de 4 preguntas por página, para que la interacción con el estudiante sea más intuitiva. Se sigue la misma estructura que en el test de Estilos de Aprendizaje para mantener la consistencia interna de la aplicación según los principios de la Interacción Persona-Ordenador. La parte que diferencia el test de Ocean del test de Estilos, es que dicho test al terminarse se muestran los resultados obtenidos al estudiante. El esquema general sería:

```
SI ultima pagina del test y pulsa Finalidar{
    Guardar_datos_estudiante_en_el_test;
    Redireccionamos a resultadosEstudiante.jsp
}
```

- **Módulo de Resultados estudiante:**

En este módulo, se le muestra al usuario los resultados del test de Estilos de Aprendizaje y el test de Ocean, primero se le muestra el estilo de aprendizaje o tipo de personalidad y después se le da una breve descripción. En este módulo, el usuario puede elegir si salir de Flayer, que de ese modo se guardarían los resultados obtenidos en los dos test o ir al módulo de test de Satisfacción, en el cual nos daría su opinión sobre la aplicación. El esquema general de **resultadosEstudiante.jsp** sería:

```
SI estudiante pulsa Realizar test Satisfaccion {
    Redireccionamos a testEstudiante.jsp
}SI NO {
    Redireccionamos a Estudiante.jsp
}
```

- **Módulo de test de Satisfacción:**

En este módulo, se le pide al usuario que complete un test de Satisfacción, en el cual se le realizan dos preguntas sobre Flayer y se deja un recuadro para que el usuario pueda dar su opinión personal sobre aspectos que considere relevantes. También se ha introducido, una descripción en la cual se le avisa que su opinión es de gran ayuda para el mantenimiento y mejora de Flayer. El esquema general de **testEstudiante.jsp** y **testProfesor.jsp** sería el siguiente:

```
SI el usuario(profesor o alumno) pulsa cerrar sesion{
    SI usuario=profesor{
        Guardar_datos_profesor_en_base_de_datos;
        Redireccionar a profesor.jsp
    }SI NO{
        Guardar_datos_estudiante_en_base_de_datos;
        Redireccionar a estudiante.jsp
    }
}
```

```
    }  
}
```

- **Módulo de Gestionar las preguntas:**

En este módulo, el profesor puede gestionar las preguntas en su totalidad, se ha puesto un índice con los patrones que puede combinar, con una breve descripción de cada uno. También, se muestran dos contadores, uno para que el profesor sepa en cada momento las preguntas que están sin modificar y otro para que sepa las preguntas que ya ha modificado o confirmado. Del mismo modo, el profesor puede elegir la opción de Modificar, Siguiente, Insertar Pregunta, Borrar Pregunta o Salir. Se ha incluido una sección de Ayuda, por si el profesor duda en el funcionamiento de este modulo, poder acceder a una pequeña descripción de cómo usar con Flayer. El esquema general de

pregunta_generada_2.jsp:

```
    Contar_preguntas_que_falta_por_modificar();  
    Contar_preguntas_que_estan_modificadas();  
    SI profesor pulsa confirmar{  
        Guardar_pregunta_confirmada_en_base_de_datos;  
        Redireccionar a pregunta_generada_2.jsp con nueva pregunta  
    }SI profesor pulsa modificar {  
        Guardar_pregunta_modificar_en_base_de_datos;  
        Redireccionamos a modificar_pregunta.jsp  
    }SI profesor pulsa un patrón y pulsa modificar {  
        Guardar_pregunta_combinar_en_base_de_datos;  
        Redireccionamos a combinar_pregunta.jsp  
    }SI profesor pulsa Insertar {  
        Redireccionamos a insertarPregunta.jsp  
    }SI profesor pulsa BorrarPregunta {  
        Redireccionamos a BorrarPregunta.jsp  
    }SI profesor pulsa Salir{  
        Redireccionamos a profesor.jsp  
    }SI profesor pulsa varios patrones y modificar{  
        MostrarMensajeError();  
    }  
}
```

- **Módulo de Modificar pregunta:**

En este módulo, se le muestra al profesor la pregunta original y la pregunta que desea modificar, con el estilo de la pregunta, así el profesor tiene toda la información necesaria para modificarla y no tiene que recordar esa información, se ha implementado para que el profesor no pueda dejar la pregunta sin modificar, mostrándole un mensaje avisándole de que no ha modificado la pregunta. El esquema general de **modificar_pregunta.jsp** sería el siguiente:

```
SI profesor pulsa confirmar y ha introducido la pregunta{
    Redireccionar a pregunta_generada_2.jsp
}SI NO{
    MostrarMensajeError();
}
```

- **Módulo de Combinar pregunta:**

En este módulo, se le muestra al profesor la pregunta original, la pregunta generada por la aplicación con el estilo de esta, la pregunta que ha elegido del índice con el estilo de esta, así el profesor tiene toda la información necesaria para combinar las preguntas y dicha información no tiene que recordarla del gestor de preguntas, igualmente como se hizo en el módulo de modificar las preguntas, también se ha implementado para que el profesor no pueda dejar las preguntas sin combinar, avisándole con mensaje de que las preguntas no se han combinado. El esquema general de **combinar_pregunta.jsp** sigue el mismo esquema general que el módulo de modificar las preguntas.

- **Módulo de Insertar preguntas:**

En este módulo, el profesor puede ver las preguntas originales que se encuentran en la base de datos, y puede añadir las que considere necesarias, insertándolas en el recuadro de texto y pulsando el botón de Insertar, en ese momento la aplicación actualiza la información y se muestra las preguntas actuales que se encuentran en la base de datos. Si hubiese estudiantes que hubieran terminado los test, se aplicarían los patrones a estas nuevas preguntas y así todos los estudiantes dispondrían de estas nuevas preguntas insertadas. El esquema general de **InsertarPregunta.jsp** sería el siguiente:

```
SI profesor pulsa insertar y ha escrito la pregunta{
    Insertar_pregunta_en_base_de_datos();
    Redireccionar a InsertarPregunta.jsp;
    Actualizar información();
}SI profesor pulsa Cerrar Sesión{
    Redireccionar a pregunta_generada_2.jsp
}SI NO {
    Mostar_mensaje_error();
}
```

- **Módulo de Borrar preguntas:**

En este módulo, el profesor puede borrar las preguntas originales que considere, simplemente tiene que elegir la pregunta que desea borrar y pulsar el botón de Borrar, en ese momento la aplicación actualiza la información y muestra las preguntas que se encuentran actuales en la base de datos. El esquema general de **BorrarPregunta.jsp** sería el siguiente:

```
SI profesor pulsa Borrar y ha seleccionado una pregunta{
    Borrar_pregunta_de_base_de_datos();
    Redireccionar a BorrarPregunta.jsp;
    ActualizarInformacion();
}SI profesor pulsa Salir { }
    Redireccionar a pregunta_generada_2.jsp;
}
```

- **Módulo de votación de preguntas:**

En este módulo, el estudiante puede votar las preguntas que entiende mejor, se muestra la información por pregunta original como cambia según el estilo/patrón que se ha aplicado y el estudiante tiene que dar su opinión sobre estas, se ha incluido una barra de progreso para que el estudiante tenga una estimación de la cantidad de preguntas que le queda aún por responder. El esquema general de **resultadosEstudianteFinal.jsp** es:

```
SI estudiante pulsa Siguiente y elige una pregunta {
    Guardar_informacion_en_base_de_datos;
```

```
Redireccionar a resultadosEstudianteFinal.jsp;  
Actualizar_informacion;  
}SI estudiante pulsa Salir {  
    Guardar_informacion_en_base_de_datos;  
    Redireccionar a estudiante.jsp  
}
```


Capítulo 4: Pruebas

Para comprobar el correcto funcionamiento de la aplicación se han realizado distintos tipos de pruebas que se describen a continuación.

4.1 Pruebas unitarias

Con este tipo de pruebas se comprueba el correcto funcionamiento de cada módulo de forma independiente. Se distingue entre pruebas de **Caja Blanca** que son aquellas que verifican que todas las bifurcaciones y posibilidades son correctas en un módulo y de **Caja Negra** que son aquellas que comprueban que la salida esperada coincide con la salida resultado de cada módulo.

- Pruebas de **Caja Blanca**

1. Comprobamos_datos_estudiante;
2. SI datos_correctos {
3. SI NO ha hecho el test_estilos{
4. Redireccionamos a **test_estilos.jsp**
5. }
6. SI ha hecho test_estilos Y NO test_ocean {
7. Redireccionamos a **testEstilosFinalizado.jsp**
8. }
9. Si ha hecho los tests{
10. Redireccionamos a **resultadosEstudianteFinal.jsp**
11. }
12. }SI NO{
13. Mostrar mensaje del suceso
14. }

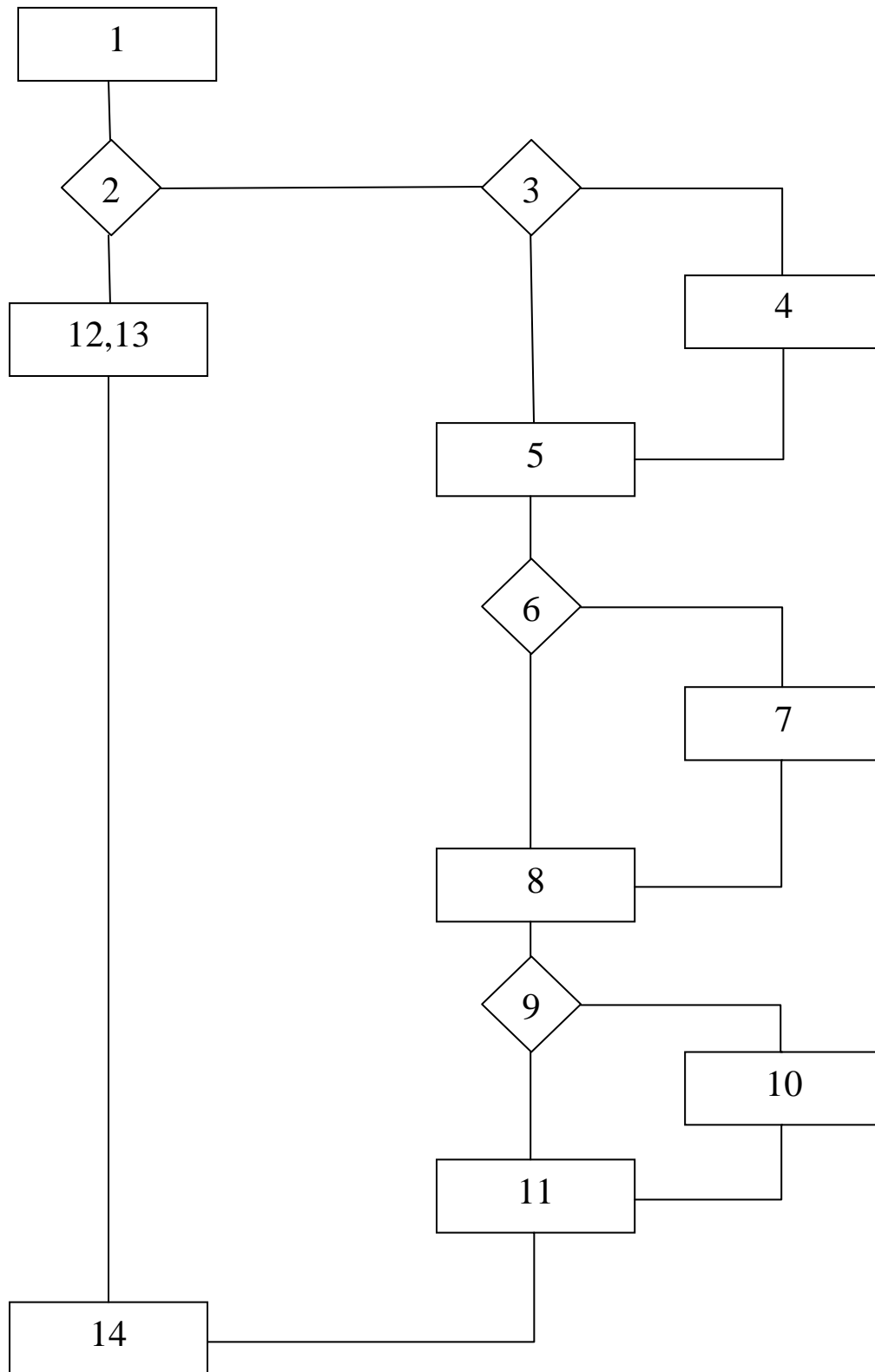


Figura 33. Prueba de Caja Blanca

Nº de caminos = Nº regiones = Aristas – Nodos – 2

Nº de caminos = 16-13+2=5

- **Camino 1:** 1, 2, 12, 13, 14.
- **Camino 2:** 1, 2, 3, 5, 6, 8, 9, 11, 14.
- **Camino 3:** 1, 2, 3, 4, 5, 6, 8, 9, 11, 14.
- **Camino 4:** 1, 2, 3, 5, 6, 7, 8, 9, 11, 14.
- **Camino 5:** 1, 2, 3, 5, 6, 8, 9, 10, 11, 14.

CAMINO	Datos_correctos	Test_Estilos	Test_Ocean	Ha hecho los tests
Camino 1	FALSO	FALSO	FALSO	FALSO
Camino 2	VERDADERO	FALSO	FALSO	FALSO
Camino 3	VERDADERO	VERDADERO	FALSO	FALSO
Camino 4	VERDADERO	FALSO	VERDADERO	FALSO
Camino 5	VERDADERO	FALSO	FALSO	VERDADERO

Tabla 8. Caminos

Como se puede observar en la tabla 8, en el camino 1 los datos no son correctos por lo que no se comprueba si ha hecho los tests, en el camino 2 los datos si son correctos pero no ha hecho ningún test, en el camino 3 los datos son correctos y ha hecho el test de estilos, en el camino 4 los datos son correctos y ha hecho el test de ocean y en el camino 5 los datos son correctos y ha realizado los dos test.

- Prueba de **Caja Negra 1:**

Prueba Caja Negra	
Prueba	Concatenar dos cadenas
Propósito	Verificar que la aplicación toma correctamente las entradas para su correcta salida
Pre-requisitos	Tener abierta la aplicación
Datos a utilizar	Entradas: 2 (cadena1,cadena2) Salidas: 1 (cadena 1+cadena 2)
Acciones	Ingresa una cadena
Resultado final	Presentará: cadena 1+cadena 2

Tabla 9. Prueba de Caja Negra

La cadena 1 que la aplicación va a concatenar es “*Qué dirias tú, para todo el mundo, sobre*”.

Y la cadena 2 que va a ser concatenada se muestra en la Figura 34:

id_pregunta	id_tema	enunciado	nota_maxima	nivel_dificultad	referencias	id_creador
1730	61	¿Qué es un compilador?	1	2	12	1

Figura 34. Prueba de Caja Negra

El resultado de la prueba se muestra en la figura 35:



Figura 35. Prueba de Caja Negra 2

Como se puede observar en el resultado de la prueba se ha concatenado correctamente la cadena 1 con la cadena 2, obteniendo una salida correcta.

- Prueba de **Caja Negra 2** basado en partición equivalente:

Un usuario intenta acceder a la aplicación con un nombre válido “alberto” y una contraseña válida “alberto123”.

	VALIDA	NO VALIDA
Usuario	7 caracteres todos letras	Más de 7 caracteres Menos de 7 caracteres Distinta de letras
Contraseña	10 caracteres	Más de 10 caracteres Menos de 10 caracteres

Tabla 10. Partición equivalente

1. Entrada: usuario “alberto” contraseña “alberto123” Salida: permitir acceso (usuario y contraseña válidos se encuentran registrados en la base de datos).
2. Entrada: usuario “alberto2” contraseña “alberto123” Salida: no permitir acceso (usuario no permitido por contener caracteres que no son letras y más de 7).
3. Entrada: usuario “albert” contraseña “alberto123” Salida: no permitir acceso (usuario contiene menos de 7 caracteres).

4. Entrada: usuario “alb5er” contraseña “alberto123” Salida: no permitir acceso (usuario contiene menos de 7 caracteres y algunos no son letras).
 5. Entrada: usuario “ ” contraseña “alberto123” Salida: no permitir acceso (usuario se encuentra vacío).
 6. Entrada: usuario “1234567” contraseña “alberto123” Salida: no permitir acceso (usuario no válido por no contener ni una sola letra, aunque contiene 7 caracteres).
 7. Entrada: usuario “alberto” contraseña “ ” Salida: no permitir acceso (usuario válido pero contraseña no es válida).
- Prueba de **Caja Negra 3** basada en los valores límite:
 1. Entrada: usuario “alberto” contraseña “22222” Salida: no permitir acceso, intentar otra vez: usuario “alberto” contraseña “22222” Salida: no permitir acceso, intentar otra vez: usuario “alberto” contraseña “22222” Salida: no permitir acceso. // **intenta 3 veces** (usuario válido pero contraseña no es válida).
 2. Entrada: usuario “alberto” contraseña “22521” Salida: no permitir acceso, intentar otra vez: usuario “alberto” contraseña “51421” Salida: no permitir acceso, intentar otra vez: usuario “alberto” contraseña “alberto123” Salida: permitir acceso. // **intenta 3 veces** (usuario válido pero contraseña no es válida).
 3. Entrada: usuario “alberte” contraseña “alberto443” Salida: no permitir acceso, intentar otra vez: usuario “alberte” contraseña “alberto443” Salida: no permitir acceso, intentar otra vez usuario “alberte” contraseña “alberto443” Salida: no permitir acceso, intentar otra vez usuario “alberte” contraseña “alberto443” Salida: permitir acceso. // **intenta 4 veces** (usuario y contraseña no son válidos).
 4. Entrada: usuario “alberto” contraseña “alberto333” Salida: no permitir acceso, intentar otra vez: usuario “alberto” contraseña “alberto123” Salida: permitir acceso. // **intenta 2 veces** (usuario “alberto” válido y contraseña “alberto123” válida).

4.2 Pruebas de integración

Este tipo de pruebas contienen a los módulos de la aplicación y se prueba el sistema como un conjunto. En este caso se plantean las pruebas de integración desde el punto de vista estructural, a continuación se muestra como se han integrado todos los módulos:

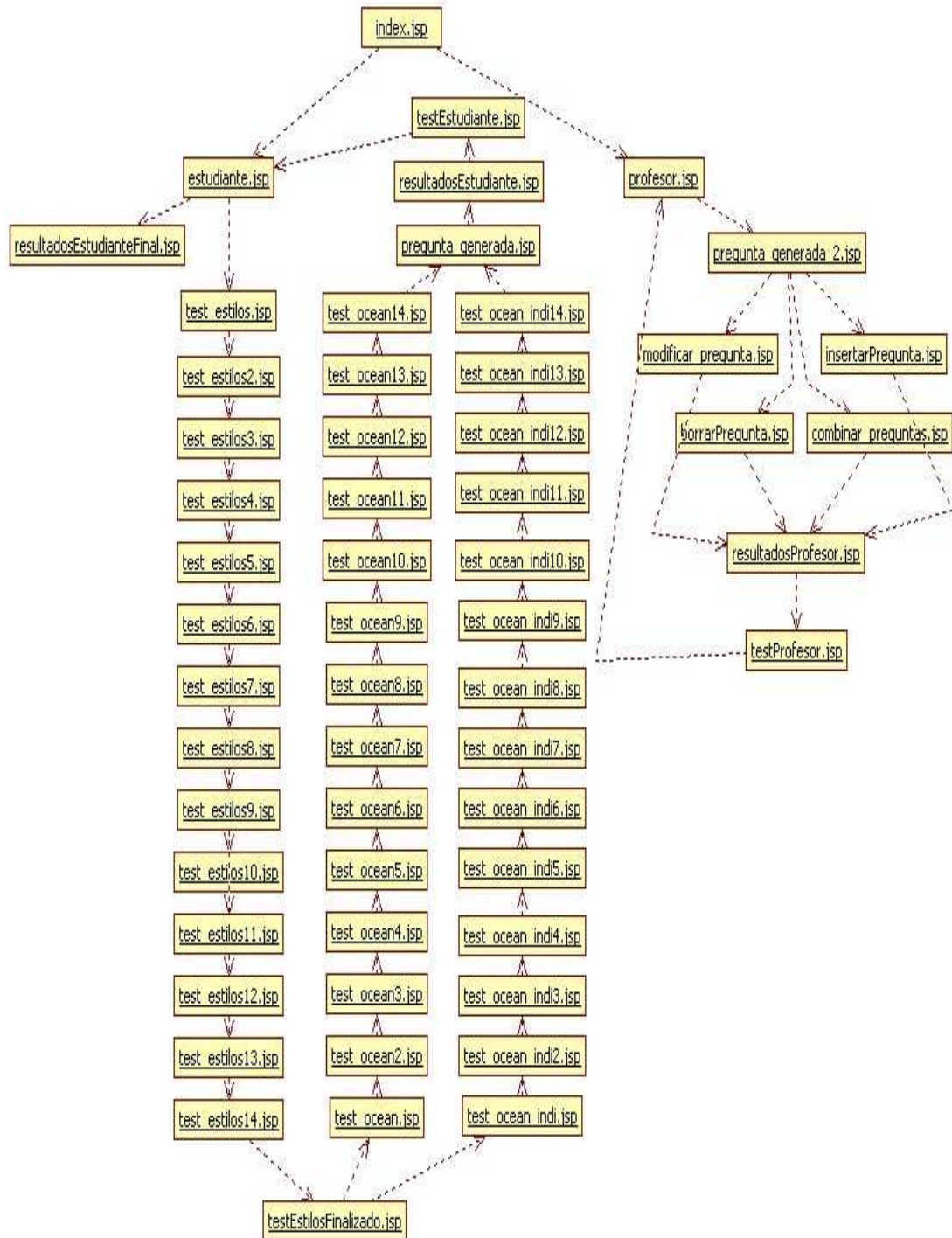


Figura 36. Diagrama de integración de los módulos

4.3 Pruebas de validación

Con este tipo de pruebas podemos ver que la aplicación cumple con los requisitos que se plantearon en la especificación, pruebas que se han realizado:

- Prueba para ver que la aplicación identifica los principales rasgos de la personalidad del estudiante.

<i>Test Ocean</i>	<i>Descripción</i>
<i>extroversion</i>	<i>tienes rasgos como hablador/a, asertivo/a, energético/a, amigable, dominante...</i>
<i>antipatia</i>	<i>eres una persona critica/ona, fria, dura de corazón, poco agradecida...</i>
<i>responsabilidad</i>	<i>eres una persona organizada, planificadora, eficiente, fiable...</i>
<i>neuroticismo</i>	<i>eres una persona tensa, ansiosa, nerviosa, preocupada...</i>
<i>abierto de mente</i>	<i>eres una persona con intereses amplios, imaginativa, original, penetrante...</i>

Figura 37. Prueba validación 1

- Prueba para ver que la aplicación clasifica según los diferentes estilos de aprendizaje del estudiante.

alberto , estos son los resultados obtenidos:

<i>Test Estilos Aprendizaje</i>	<i>Descripción</i>
<i>activo</i>	<i>tiendes a retener y comprender mejor la información</i>
<i>perceptivo</i>	<i>tiendes a preferir los hechos de aprendizaje</i>
<i>visual</i>	<i>recuerdas mejor lo que ves (imágenes, diagramas, películas...)</i>
<i>secuencial</i>	<i>tiendes a ganar la comprensión en pasos lineales</i>

Figura 38. Prueba validación 2

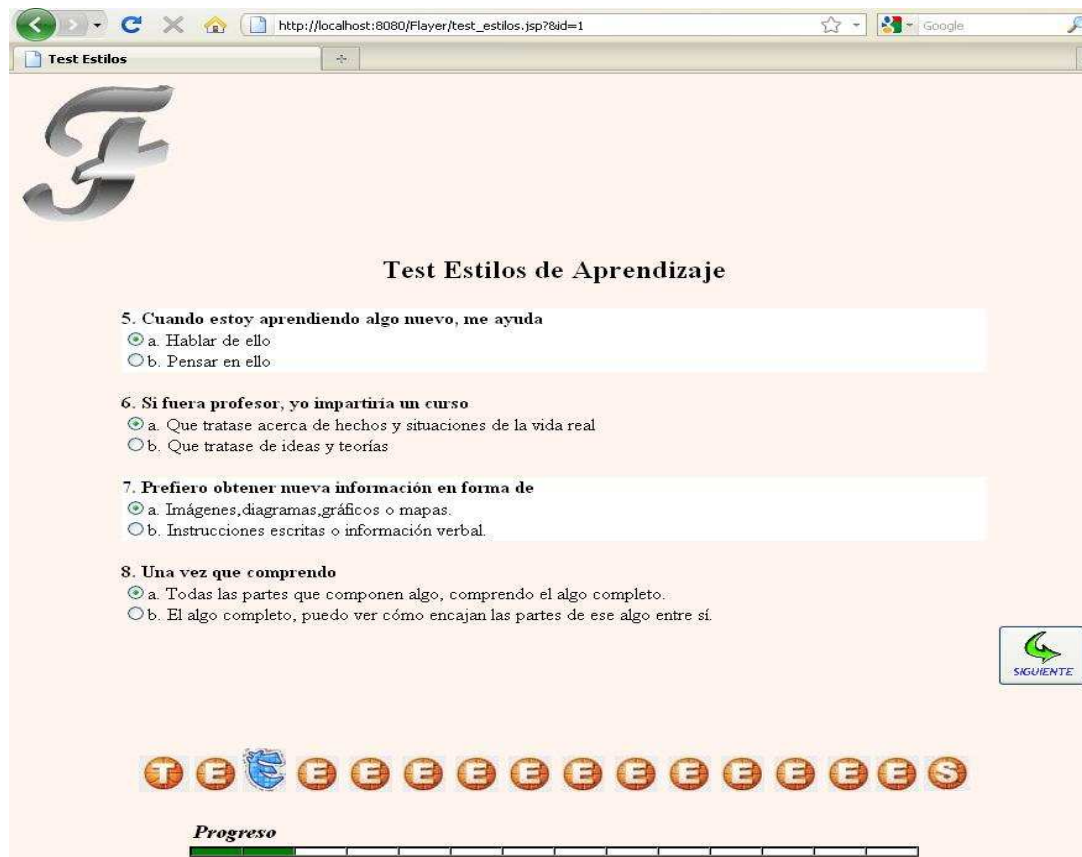
- Prueba para ver que la aplicación muestra las preguntas generadas al profesor.

Tabla resumen

<i>Pregunta Original</i>	<i>Pregunta Final</i>
¿Qué significa que una operación sea atómica?	¿Qué significa que una operación sea atómica?
¿Qué significa que una operación sea atómica?	Cómo ves, ¿Qué significa que una operación sea atómica?
¿Qué es un proceso zombi?	¿Qué es un proceso zombi?
¿Qué es un proceso zombi?	Cómo ves, ¿Qué es un proceso zombi?
¿Cuáles son los distintos tipos de capacidades que puede desarrollar un niño a lo largo de su vida?	Háblame de capacidades
¿Cuáles son los distintos tipos de capacidades que puede desarrollar un niño a lo largo de su vida?	Cómo concibes, ¿Cuáles son los distintos tipos de capacidades que puede desarrollar un niño a lo largo de su vida?
¿Qué significa que una operación sea atómica?	¿Qué dirías tú, para todo el mundo, sobre ¿Qué significa que una operación sea atómica?
¿Qué significa que una operación sea atómica?	A ver si te sabes esto, ¿Qué significa que una operación sea atómica?
¿Qué significa que una operación sea atómica?	Si tuvieras que explicárselo a un amigo como responderías a, ¿Qué significa que una operación sea atómica?
¿Qué significa que una operación sea atómica?	Si te plantearan la siguiente pregunta, sin obligación de responderla, como responderías a, ¿Qué significa que una operación sea atómica?
¿Qué significa que una operación sea atómica?	¿Qué significa que una operación sea atómica?
¿Qué es un proceso zombi?	¿Qué es un proceso zombi?
¿Qué es un proceso zombi?	¿Qué es un proceso zombi?
¿Qué es un proceso zombi?	¿Qué es un proceso zombi?
¿Qué es un proceso zombi?	¿Qué es un proceso zombi?
¿Qué es un proceso zombi?	¿Qué es un proceso zombi?
¿Cuáles son los distintos tipos de capacidades que puede desarrollar un niño a lo largo de su vida?	¿Qué dirías tú, para todo el mundo, sobre ¿Cuáles son los distintos tipos de capacidades que puede desarrollar un niño a lo largo de su vida?
¿Cuáles son los distintos tipos de capacidades que puede desarrollar un niño a lo largo de su vida?	A ver si te sabes esto, ¿Cuáles son los distintos tipos de capacidades que puede desarrollar un niño a lo largo de su vida?
¿Cuáles son los distintos tipos de capacidades que puede desarrollar un niño a lo largo de su vida?	Si tuvieras que explicárselo a un amigo como responderías a, ¿Cuáles son los distintos tipos de capacidades que puede desarrollar un niño a lo largo de su vida?

Figura 39. Prueba validación 3

- Prueba para comprobar que el estudiante realiza los test online.



• Figura 40. Prueba validación 4

- Prueba para ver que el profesor puede modificar, borrar, insertar, confirmar, modificar y combinar las preguntas generadas por la aplicación.

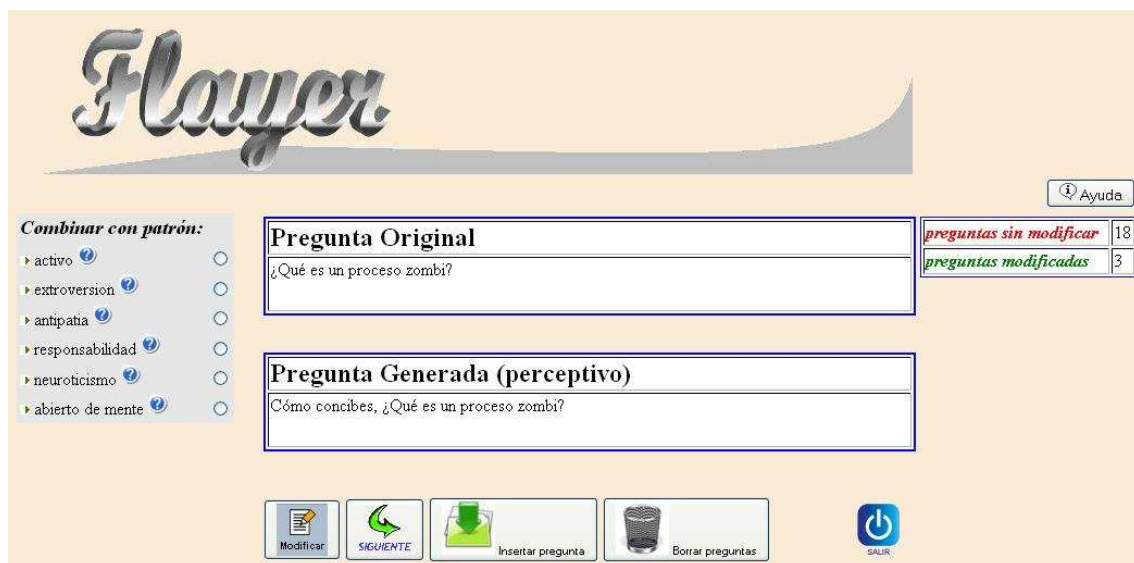


Figura 41. Prueba validación 5

- Prueba para comprobar que la aplicación transforma las preguntas según los resultados obtenidos en los tests.

Pregunta Generada (perceptivo)
Cómo concibes, ¿Qué es un proceso zombi?

Figura 42. Prueba validación 6

- Prueba para ver que el sistema almacena las nuevas preguntas en la base de datos.



id_usuario	nombre	enunciado	p_original	estilo
3434	pedro	¿Qué dirías tú, para todo el mundo, sobre ¿Los atributos h...	¿Los atributos heredados se suelen usar en a...	extroversión
3435	pedro	Intentar cambiar el contexto de la pregunta, pulse el botó...	¿Los atributos heredados se suelen usar en a...	abierto de mente
3433	pedro	A ver si te sabes esto, ¿Los atributos heredados se suele...	¿Los atributos heredados se suelen usar en a...	antipatía
3432	pedro	Si depende de tí, que tu amigo apruebe, Cómo responde...	¿Los atributos heredados se suelen usar en a...	responsabilidad
3431	pedro	Si te plantearan la siguiente pregunta, sin obligación de r...	¿Los atributos heredados se suelen usar en a...	neuroticismo

Figura 43. Prueba validación 7

- Prueba para observar que el estudiante puede revisar las preguntas que se generaron para sus diferentes estilos de aprendizaje y tipos de personalidad.

Pulsa aquella pregunta que entiendes mejor

Cómo concibes, ¿Qué significa que una operación sea atómica?	<input checked="" type="radio"/>
¿Qué significa que una operación sea atómica?	<input type="radio"/>

Progreso

© Alberto Redondo, Mayer 2011

Figura 44. Prueba validación 8

- Prueba para comprobar que el programa cumple con los principios de Interacción Persona-Ordenador y que sea usable.



Figura 45. Prueba validación 9

- Prueba con mediciones de reloj, para comprobar que la aplicación tarda menos de un minuto en las interacciones en los tests y menos de una hora en la transformación de las preguntas en la base de datos.

4.4 Pruebas de aceptación

Este tipo de pruebas son realizadas por los usuarios sobre el sistema al completo, con el fin de ayudar a corregir errores que no se hayan detectado y poder mejorar la aplicación en aspectos relacionados de cara a la Interacción Persona-Ordenador. Se han realizado dos pruebas; la primera con diez estudiantes de Informática y la segunda con diez usuarios no informáticos. De ambas pruebas se han extraído los resultados almacenados en la base de datos, con la aplicación ejecutándose en el enlace público <http://flyer.no-ip.info/Flyer>

4.4.1 Informáticos

Diez estudiantes de Informática de la URJC realizaron el test de “Big Five” o de cinco grandes rasgos y el test de “Felder” o test de Estilos de Aprendizaje y después votaron las preguntas generadas por la aplicación.

A cada estudiante se le asignó un usuario y una contraseña. La duración estimada de la prueba fue de dos horas. Los resultados se muestran en la Tabla 10.

Usuario	Opinión aplicación	¿Se han adaptado las preguntas?
1	Mala	En algunas ocasiones
2	Interesante	En algunas ocasiones
3	Interesante	En algunas ocasiones
4	Buena	Sí
5	Interesante	Sí
6	Buena	En algunas ocasiones
7	Normal	Sí
8	Interesante	En algunas ocasiones
9	Mala	Sí
10	Normal	Sí

Tabla 11. Resultados de la prueba con los alumnos de Informática

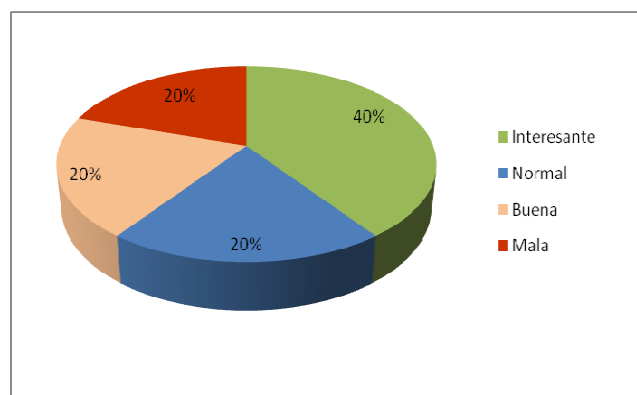


Figura 46. Gráfica con los resultados de la prueba 1 con los alumnos de Informática

Las opiniones principales recogidas en el test de satisfacción son las siguientes:

- *“Se consigue una descripción final más personalizada. Tal vez pondría más preguntas por pantalla. Me ha parecido buena la idea de poner chistes entre algunas preguntas para que no se hiciese tan pesado, pero los chistes podrían ser mejorables, aunque el de la bici me ha gustado”.*
- *“Me parece bien que haya más preguntas acerca de un mismo tema ya que en ocasiones es necesario especificar más acerca de algo.”*

Algunos estudiantes confundieron la “F” mayúscula de la URL de la página de Flayer y otros tuvieron dificultad en encontrar el botón de siguiente de la aplicación en la página de los tests.

4.4.2 No informáticos

Diez estudiantes no informáticos realizaron el test de “Big Five” o de cinco grandes rasgos y el test de “Felder” o test de Estilos de Aprendizaje. Se realizó en diferentes días y asignando un usuario y una contraseña a cada uno de los usuarios.

Los resultados se muestran en la Tabla 11.

Usuario	Opinión aplicación	¿Se han adaptado las preguntas?
1	Buena	Sí
2	Muy Interesante	Sí
3	Buena	Sí
4	Buena	Sí
5	Muy Interesante	En algunas ocasiones
6	Buena	Sí
7	Buena	En algunas ocasiones
8	Muy Interesante	Sí
9	Buena	Sí
10	Interesante	Sí

Tabla 12. Resultados de la prueba 2 con estudiantes no informáticos

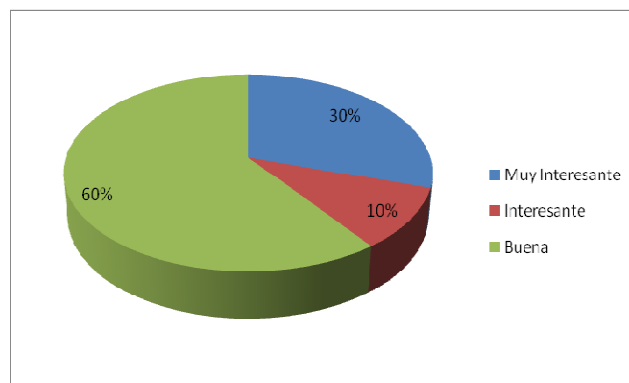


Figura 47. Gráfica de resultados de la prueba 2 con estudiantes no informáticos

Los diez usuarios (100%) votaron por la pregunta adaptada, sus opiniones principales recogidas en el test de satisfacción son las siguientes:

- “Me parece una aplicación muy interesante, ya que ha conseguido acertar en mi personalidad y describirme como soy.”
- “A mí me ha parecido muy buena, aunque yo mejoraría los chistes.”

Capítulo 5: Conclusiones y trabajo futuro

5.1 Logros alcanzados

Se han cumplido los objetivos establecidos de crear una aplicación capaz de generar automáticamente las preguntas, y con valoración positiva por parte tanto de usuarios informáticos y no informáticos demostrando su usabilidad.

Se comenta la continuación cómo se han cumplido los requisitos establecidos:

- Se ha conseguido que la aplicación identifique los principales rasgos de personalidad de un estudiante mediante el test Big-Five-OCEAN.
- Se ha conseguido que la aplicación clasifique al estudiante según sus estilos de aprendizaje mediante el test de Estilos de Aprendizaje.
- Se ha conseguido que el programa una vez ha adaptado las preguntas para cada estudiante, se las mostrase al profesor para que éste pudiese confirmarlas, modificarlas o combinarlas con los diferentes rasgos de personalidad del estudiante.
- Se ha conseguido que el estudiante complete online los tests de Personalidad y Estilos de Aprendizaje mediante el link <http://flyer.no-ip.info/Flayer>.
- Se ha conseguido que el profesor pueda insertar, modificar y borrar las preguntas de la base de datos mediante los módulos correspondientes en la aplicación.
- Se ha conseguido que la aplicación adapte las preguntas automáticamente mediante los resultados obtenidos en los tests de Personalidad y de Estilos de Aprendizaje.
- Se ha conseguido que la aplicación almacene las preguntas adaptadas para cada estudiante en la base de datos.
- Se ha conseguido que el estudiante revise las preguntas generadas automáticamente y pueda elegir la que considere que mejor entiende.
- Se ha conseguido que la velocidad de respuesta sea inferior a un minuto en las interacciones con los test y menor de una hora en la transformación de las preguntas en la base de datos.
- Se ha conseguido que la aplicación funcione de forma concurrente, por ello permite que los estudiantes interactúen con ella todos a la vez y no tengan que hacerlo de forma secuencial.

5.2 Trabajos futuros

Se podrían incluir estas ampliaciones como trabajo futuro:

- Poner todas las letras de la URL en minúscula para no obtener confusiones en los usuarios.
- El botón de siguiente colocarlo en un lugar centrado de la página para conseguir que sea más visible.
- Permitir que las preguntas se importen automáticamente de plataformas e-learning como Moodle o b-learning como las Will Tools.
- Añadir más posibilidades de modificación y confirmación de las preguntas para el profesor.

Bibliografía

Referencias

- [1] Fco. Javier Ceballos. (2007). C/C++, Curso de Programación (3ª Edición). Ra-Ma.
- [2] Fco. Javier Ceballos. (2005). Microsoft Visual Basic .NET Lenguaje y aplicaciones, Ra-Ma.
- [3] Matt Doyle. (2010). Fundamentos PHP Práctico. Anaya.
- [4] Ivar Jacobson, Grady Booch, James Rumbaugh (2000). El Proceso Unificado de Desarrollo de Software.

Referencias Web

- [HTTP1] <http://www.blogterrier.com.ar/2009/05/lenguajes-de-programacion-orientada.html> , último acceso 4 Enero 2011.
- [HTTP2] <http://www.desarrolloweb.com/> , último acceso 4 Enero 2011.
- [HTTP3] http://danielpecos.com/docs/mysql_postgres/x57.html, último acceso 4 Enero 2011.
- [HTTP4] http://danielpecos.com/docs/mysql_postgres/x15.html, , último acceso 4 Enero 2011.
- [HTTP5] http://es.wikipedia.org/wiki/Microsoft_Access , último acceso 4 Enero 2011.
- [HTTP6] http://www.xolucionesinformaticas.com/index.php?option=com_content&view=article&id=77&Itemid=85 , último acceso 4 Enero 2011.
- [HTTP7] <http://es.wikipedia.org/wiki/Tomcat>, último acceso 4 Enero 2011.
- [HTTP8] http://blogs.sun.com/AlanVargas/entry/qu%C3%A9_es_glassfish, último acceso 4 Enero 2011.
- [HTTP9] <http://www.calameo.com/books/0000446953a6e3deb1aaa> , último acceso 4 Enero 2011.
- [HTTP10] www.dosideas.com/wiki/NetBeans , último acceso 4 Enero 2011
- [HTTP11] www.alegsa.com.ar/Dic/windows.php , último acceso 4 Enero 2011.
- [HTTP12] www.webquest.es/node/1781 , último acceso 4 Enero 2011.
- [HTTP13] http://es.wikipedia.org/wiki/Mac_OS , último acceso 4 Enero 2011.
- [HTTP14] <http://es.outofservice.com/bigfive/?srclang=en>, último acceso 4 Enero 2011.
- [HTTP15] <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>, último acceso 4 Enero 2011.

Anexo A: Diagramas de casos de uso

En esta parte se muestra los diagramas de casos de uso, de una forma más descriptiva:

- **CASO DE USO VALIDAR USUARIO (Usuario y Profesor):**

Actor	Sistema
	1.El sistema muestra el interfaz al usuario
2.El usuario introduce nombre y clave de acceso	
3.El usuario clikea en el botón de aceptar	
	4. El sistema valida su nombre y su clave de acceso y da paso al siguiente interfaz.
Caminos alternativos: -Paso 2 y 3: El usuario clikea en el botón borrar -Paso 2 y 3: El usuario cierra la ventana. -Paso 2: El usuario introduce nombre y clave de acceso y no pulsa ningún botón. -Paso 4: El sistema no valida el nombre y su clave de acceso, con motivo de no estar dado de alta en la base de datos.	

- **CASO DE USO GESTIONAR PREGUNTAS (Profesor):**

Actor	Sistema
	1. El sistema muestra el interfaz de gestionar las preguntas.
2. El usuario pulsa Modificar.	
	3. El sistema muestra el interfaz de modificar la pregunta.
4. El usuario introduce la pregunta modificada y pulsa Confirmar.	
	5. El sistema muestra el interfaz de gestionar las preguntas.
6. El usuario elige un patrón y pulsa	

Generación automática de preguntas adaptadas

Modificar.	
	7.El sistema muestra el interfaz de combinar preguntas
8. El usuario introduce la pregunta combinada y pulsa Confirmar	
	9.El sistema muestra el interfaz de gestionar las preguntas.
10. El usuario pulsa Siguiente	
	11. El sistema muestra la siguiente pregunta.
12. El usuario pulsa Insertar Pregunta	
	13. El sistema muestra el interfaz de insertar preguntas.
14. El usuario escribe la pregunta y pulsa Insertar y Volver.	
	15. El sistema muestra el interfaz de gestionar las preguntas.
16. El usuario pulsa Borrar.	
	17. El sistema muestra el interfaz de borrar preguntas.
18. El usuario elige una pregunta y pulsa el botón de Borrar y Volver.	
	19. El sistema muestra el interfaz de gestionar las preguntas.
Caminos alternativos: -Pasó 2, 4, 6, 8, 10, 12, 14, 16,18: el usuario cierra la ventana. -Paso 4, 8: el usuario introduce la pregunta y no pulsa confirmar. -Paso 16: el usuario elige una pregunta y no pulsa el botón de Borrar.	

• **CASO DE USO REALIZAR TEST SATISFACCION (Profesor):**

Actor	Sistema
	1. El sistema muestra el interfaz del test de Satisfacción.
2. El usuario elige una opción, introduce una descripción y pulsa Cerrar Sesión.	
	3. El sistema muestra el interfaz de validar usuario del Profesor.
Caminos alternativos: Paso 2: El usuario cierra la ventana. Paso 2: El usuario elige una opción y no pulsa el botón de Cerrar Sesión. Paso 2: El usuario no elige una opción y pulsa el botón de Cerrar Sesión.	

• **CASO DE USO REALIZAR TEST SATISFACCION (Estudiante):**

Actor	Sistema
	1. El sistema muestra el interfaz del test de Satisfacción.
2. El usuario elige una opción, introduce una descripción y pulsa Cerrar Sesión.	
	3. El sistema muestra el interfaz de validar usuario del estudiante.
Caminos alternativos: Paso 2: El usuario cierra la ventana. Paso 2: El usuario elige una opción y no pulsa el botón de Cerrar Sesión. Paso 2: El usuario no elige una opción y pulsa el botón de Cerrar Sesión.	

• **CASO DE USO VER PREGUNTAS MODIFICADAS:**

Actor	Sistema
	1. El sistema muestra el interfaz de gestionar las preguntas.
2. El usuario pulsa el botón de Siguiente	
	3. El sistema muestra el interfaz de ver preguntas modificadas.
4. El usuario pulsa el botón de Cerrar Sesión.	
	5. El sistema muestra el interfaz de validar usuario del Profesor.
Caminos alternativos: Paso 2 y 4: El usuario cierra la ventana. Paso 4: El usuario pulsa Realizar test Satisfacción. Paso 4: El usuario no pulsa el botón de Cerrar Sesión.	

• **CASO DE USO REALIZAR TEST ESTILOS DE APRENDIZAJE:**

Actor	Sistema
	1. El sistema muestra el interfaz del test de Estilos De Aprendizaje.
2. El usuario elige una opción y pulsa el botón de Siguiente.	
	3. El sistema muestra el interfaz de siguiente página del test de Estilos de Aprendizaje.
4. El usuario pulsa el botón de Finalizar.	
	5. El sistema muestra el interfaz de test De Estilos de Aprendizaje Finalizado.

Caminos alternativos:

-Paso 2 y 4: El usuario cierra la ventana.

-Paso 2: el usuario no elige una opción y pulsa Siguiente

• **CASO DE USO REALIZAR TEST OCEAN:**

Actor	Sistema
	1. El sistema muestra el interfaz del test de Ocean.
2. El usuario elige una opción y pulsa el botón de Siguiente.	
	3. El sistema muestra el interfaz de siguiente página test de Ocean.
4. El usuario pulsa el botón de Finalizar.	
	5. El sistema muestra el interfaz de Resultados de los test.
Caminos alternativos:	
-Paso 2 y 4: El usuario cierra la ventana.	
-Paso 2: el usuario no elige una opción y pulsa Siguiente	

• **CASO DE USO VER RESULTADOS:**

Actor	Sistema
	1. El sistema muestra el interfaz de test de Ocean.
2. El usuario pulsa la opción de Finalizar.	
	3. El sistema muestra el interfaz de los Resultados obtenidos.
4. El usuario pulsa Cerrar Sesión.	
	5. El sistema muestra el interfaz de validar usuario del Estudiante.
Caminos alternativos:	

-Paso 2 y 4: El usuario cierra la ventana.
-Paso 4: El usuario no pulsa Cerrar Sesión.

• **CASO DE USO VOTAR PREGUNTAS:**

Actor	Sistema
	1. El sistema muestra el interfaz de votar las preguntas.
2. El usuario elige una pregunta y pulsa el botón de Siguiente.	
	3. El sistema muestra el interfaz de votar las preguntas, con la siguiente pregunta.
4. El usuario pulsa Salir.	
	5. El sistema muestra el interfaz de validar usuario de Estudiante.
Caminos alternativos: -Paso 2 y 4: El usuario cierra la ventana. -Paso 2: El usuario no elige una pregunta y pulsa el botón de Siguiente. -Paso 2: El usuario elige una pregunta y no pulsa el botón de Siguiente.	

Anexo B: Diagramas de secuencia

A continuación se muestra un diagrama de secuencia, que describe el recorrido del estudiante en primera instancia:

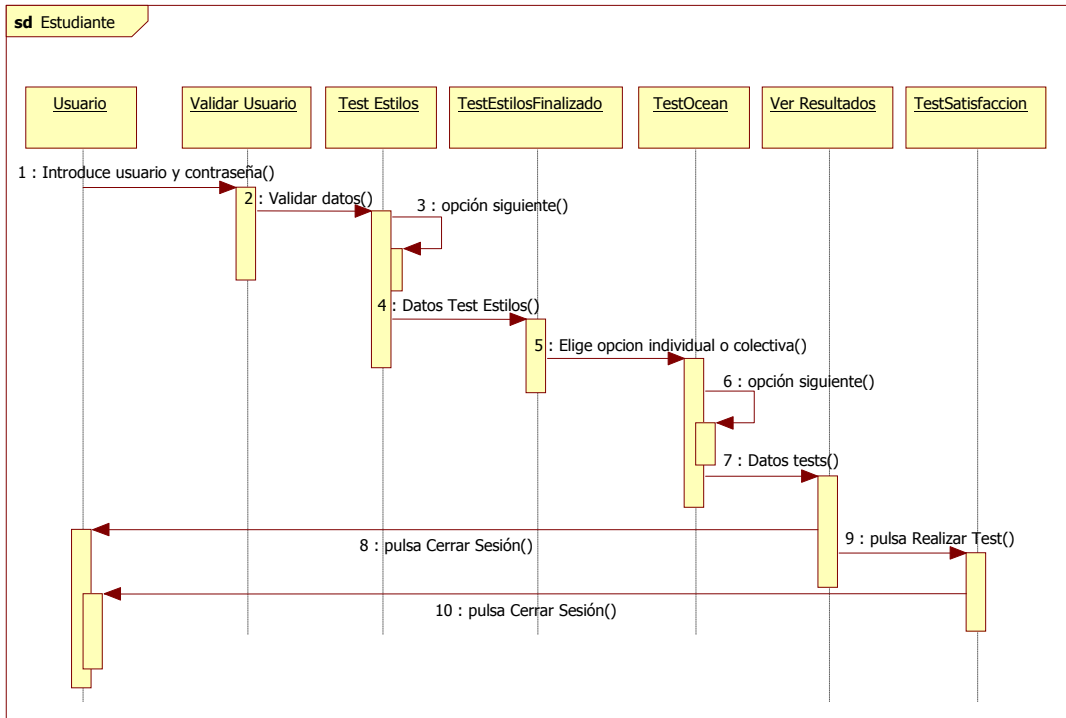


Figura 48. Diagrama de secuencia del estudiante primera instancia

En el siguiente diagrama, se muestra el recorrido del estudiante en segunda instancia:

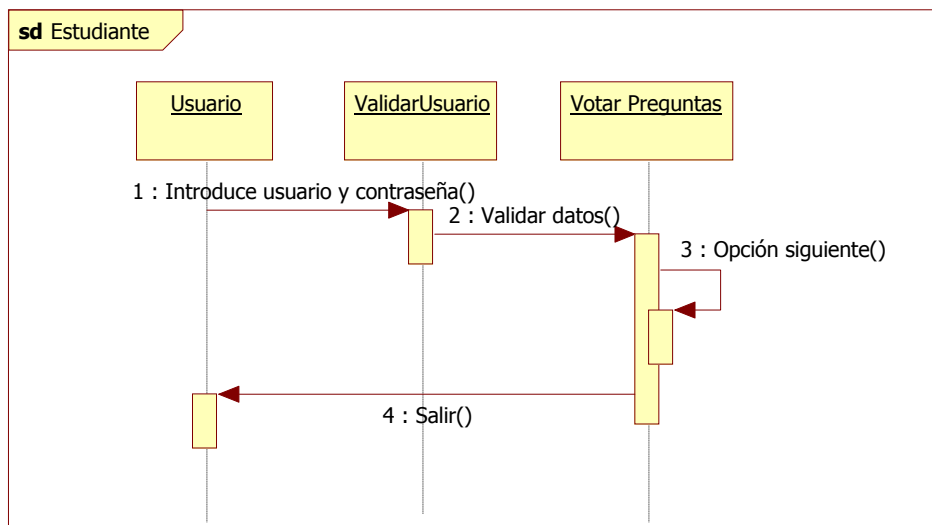


Figura 49. Diagrama de secuencia del estudiante segunda instancia

Seguidamente se muestra el diagrama de iteración del usuario Profesor, que describe el recorrido del Profesor en el sistema:

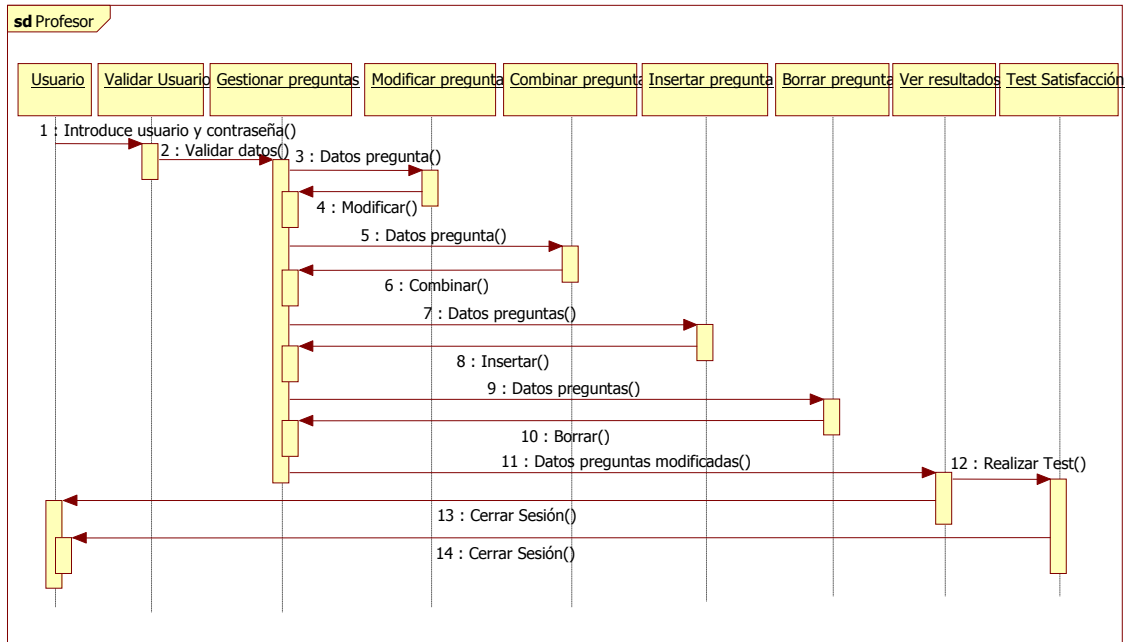


Figura 50. Diagrama de secuencia del profesor

Anexo C: Evolución del diseño centrado en el usuario

La evolución del diseño centrado en el usuario, ha sido una parte muy importante del proyecto, ya que se ha intentado facilitar la interacción Persona-Ordenador. Veamos unos ejemplos de la evolución de los interfaces.

Este fue un diseño inicial del interfaz de acceso del estudiante y del profesor, como podemos observar en la Figura 51:

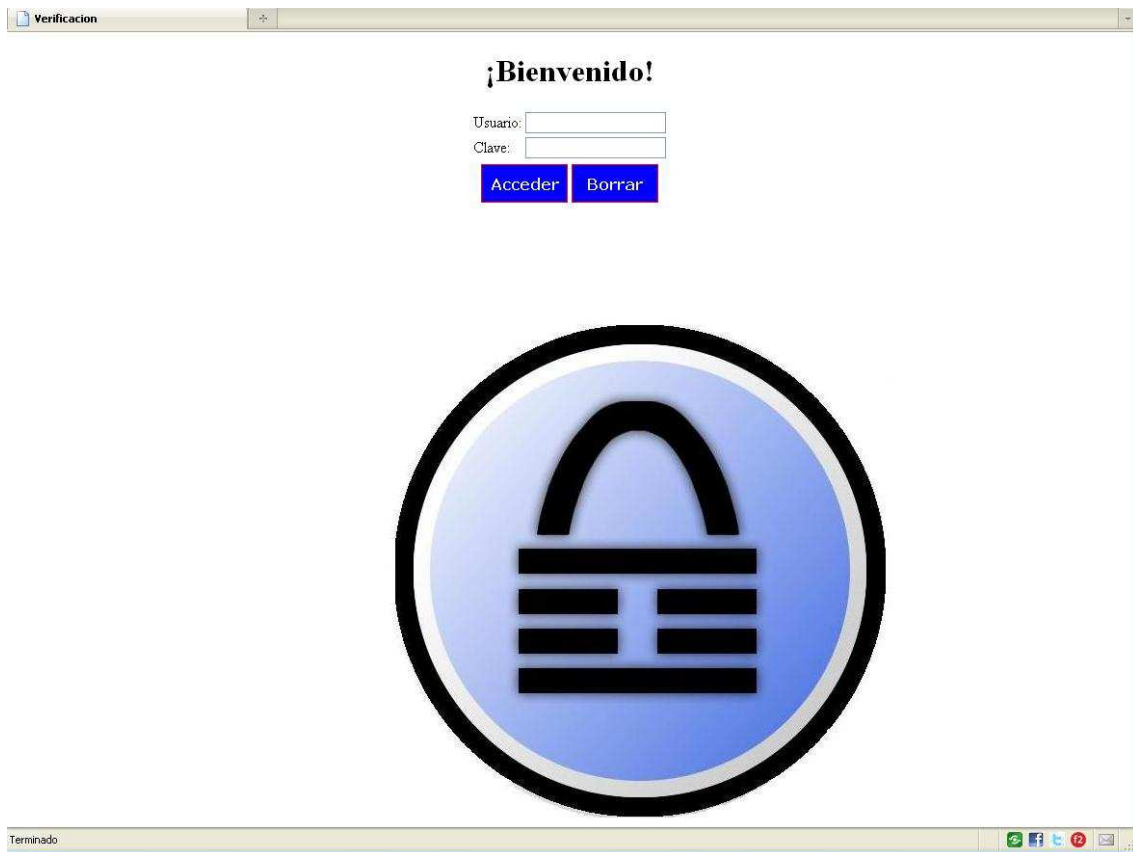


Figura 51. Interfaz de acceso inicial

En este interfaz se observó que el logo era muy grande, por lo tanto se mejoró en el siguiente modelo, como se puede ver en la Figura 52:



Figura 52. Interfaz de acceso evolución 2

Este interfaz mejoró respecto del anterior, porque unificaba el acceso del estudiante y del profesor, debido a ello se pensó en separar el acceso de un estudiante y de un profesor, de esta manera el acceso a la aplicación sería independiente, para ello se creó dos interfaces, incluyendo el logo de la aplicación, un diseño con fondo oscuro donde la información importante se encuentra en fondo blanco, intentando que el usuario la pueda leer con facilidad, del mismo modo, se incluyó un botón de ayuda e información para contactar con el administrador:

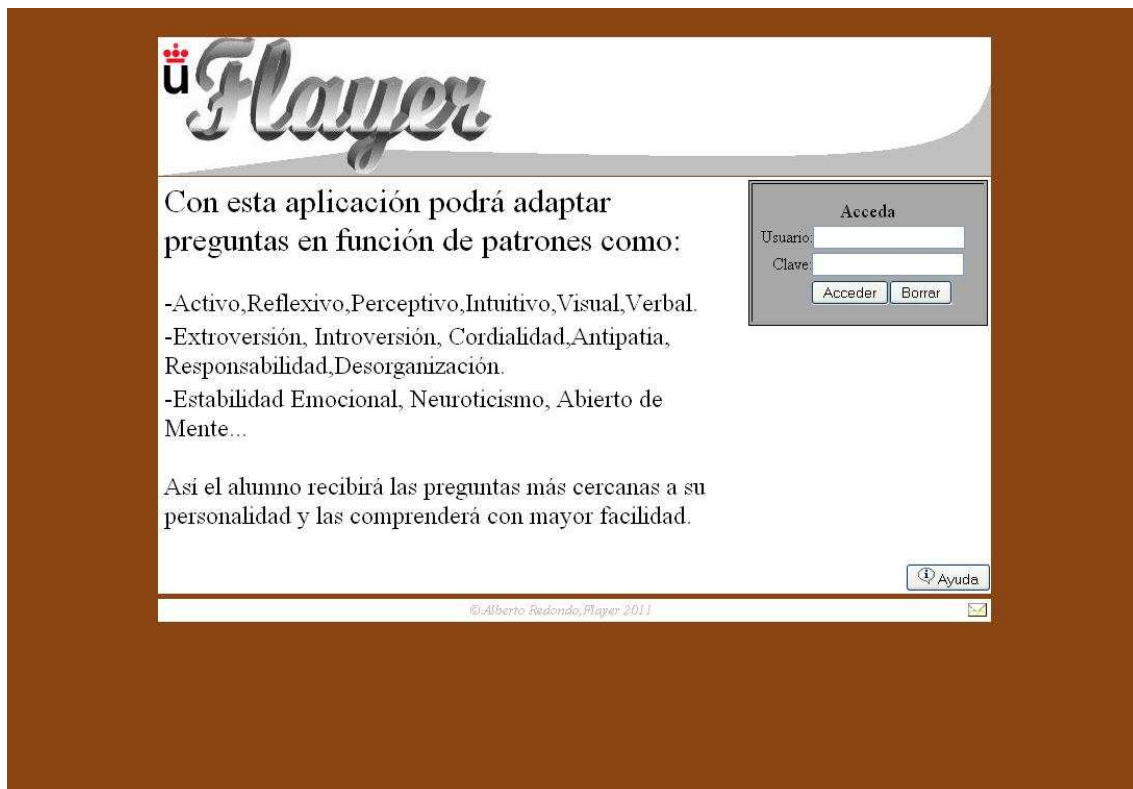


Figura 53. Interfaz de acceso evolución 3

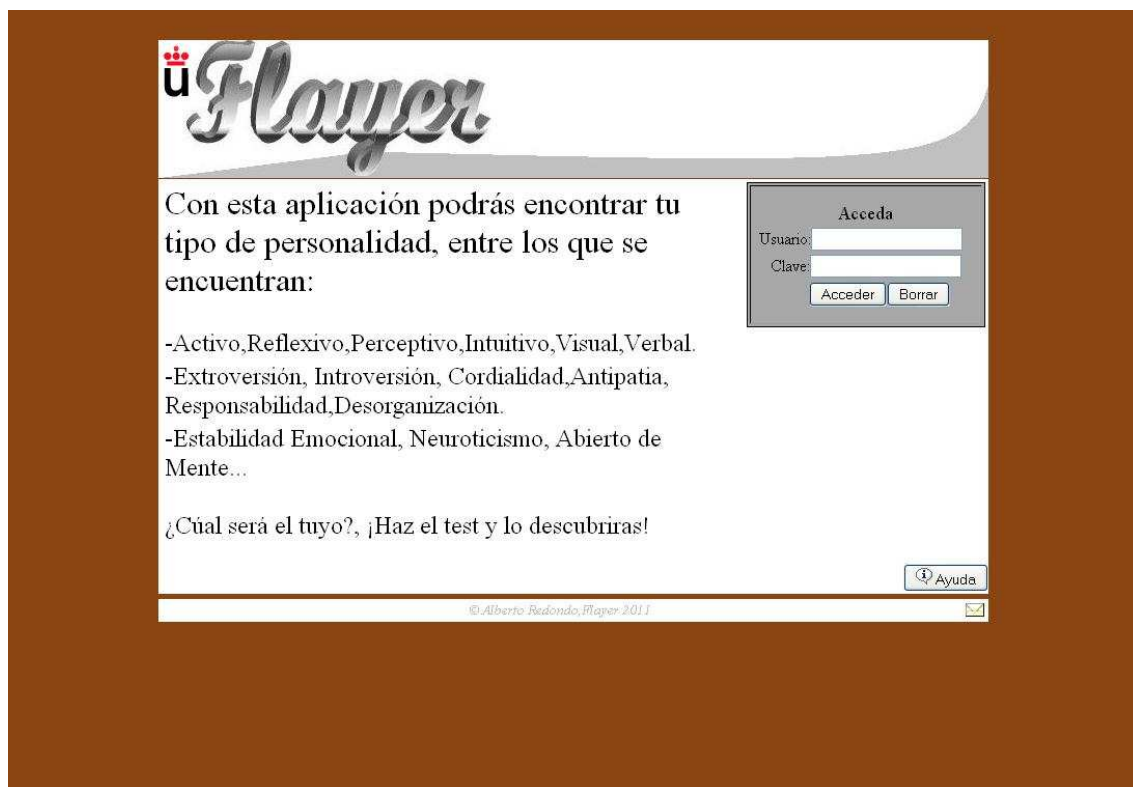


Figura 54. Interfaz de acceso evolución 4

Otro ejemplo de la evolución del diseño centrado en el usuario, lo podemos encontrar en las páginas de los test, en la Figura 55 se muestra un diseño inicial del interfaz:



Test 1

Test Estilos de Aprendizaje

1. Entiendo mejor algo cuando

- a. lo intento o lo pruebo
- b. pienso en ello

2. Yo me considero más bien

- a. Realista
- b. Innovador

3. Cuando pienso en lo que hice ayer, lo más frecuente es que obtenga

- a. Una imagen
- b. Palabras

4. Tiendo a

- a. Comprender detalles de un tema o materia pero puedo tener una idea difusa de su estructura global
- b. Comprender la estructura global pero tener una idea difusa de los detalles

Siguiente

Figura 55. Interfaz de test inicial

En los siguientes interfaces se muestra la evolución que tuvo los test de cara a mejorar la interacción Persona-Ordenador:



Figura 56. Interfaz de test evolución

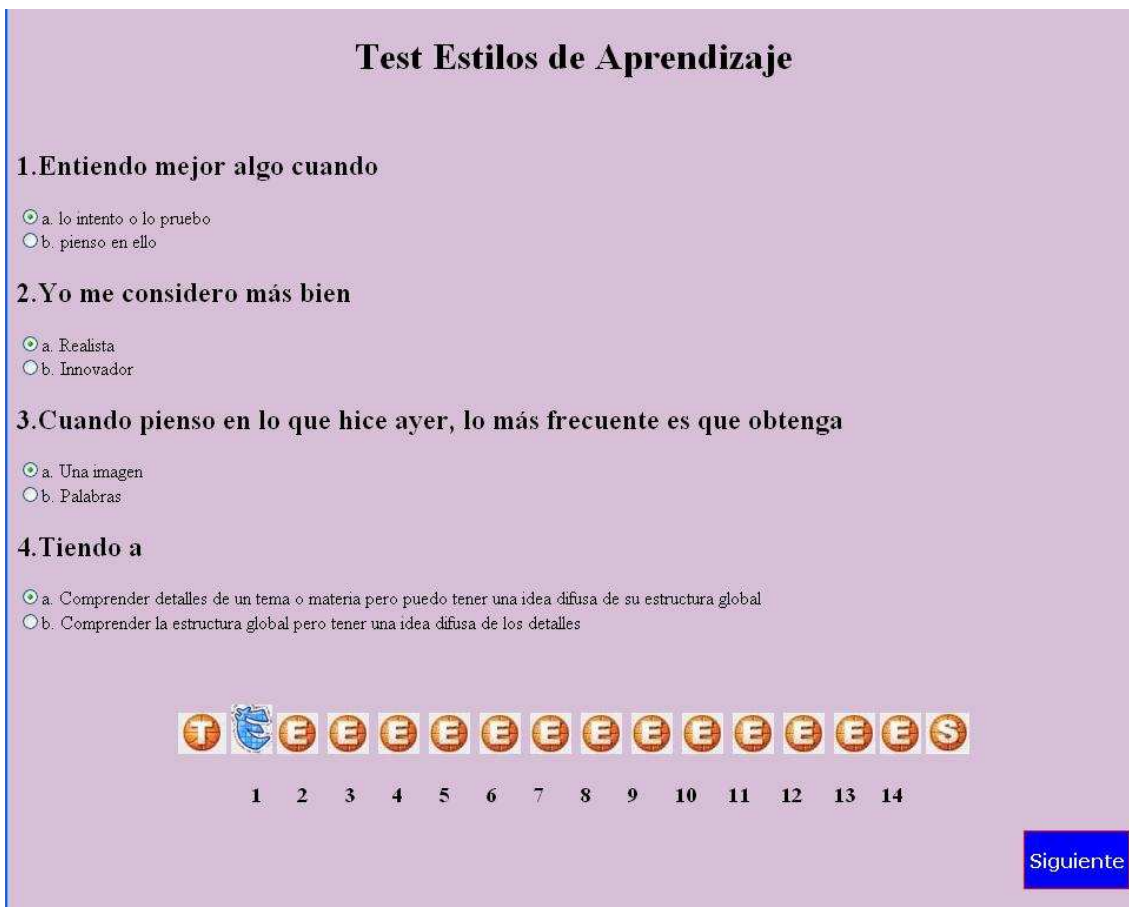



Figura 57. Interfaz de test evolución 2



Test Estilos de Aprendizaje

13. En las clases que he recibido

- a. Normalmente he conseguido saber mucho de los estudiantes.
- b. Raramente he conseguido saber mucho de los estudiantes.

14. Cuando leo literatura realista, prefiero


- a. Algo que me enseñe nuevos hechos o me cuente cómo hacer algo.
- b. Algo que me de nuevas ideas en las que pensar.

15. Me gustan los profesores

- a. Que ponen muchos diagramas en la pizarra.
- b. Que dedican mucho tiempo a explicar.

16. Cuando analizo una historia o una novela

- a. Pienso en los sucesos e intento unirlos para descubrir las tramas ocultas.
- b. Sólo sé cuáles son las tramas ocultas una vez que he terminado de leer y tengo que volver atrás para encontrar los sucesos que las demuestran.



T E E E E E E E E E E E E E E S

Progreso

© Alberto Redondo, Player 2011

Figura 58. Interfaz de test evolución 3

Como se observa en los ejemplos anteriores, se intenta que los interfaces tengan consistencia a la hora de cambiar de un interfaz a otro y del mismo modo que usuario tenga sensación de progreso.

Anexo D: Problemas solventados

Principalmente en el desarrollo de este proyecto han surgido tres grandes problemas:

- El primer problema apareció cuando no se mostraban las tildes en las páginas .jsp, el problema se solucionó al revisar la codificación de la base de datos. En particular, cuando se instala MySQL si no se especifica una codificación de la base de datos se configura por defecto LATIN1 (ISO-8859-1), incompatible con UTF-8, que es una codificación distinta en algunos navegadores web. Por lo tanto salen por pantalla signos equivalentes entre codificaciones y por lo consiguiente no salen ni las “ñ”, ni las tildes. Para solucionar este problema se cambió la configuración de la base de datos a UTF-8.
- El segundo problema apareció en los test de satisfacción ya que la información que se recogía de las opiniones de los usuarios estaba codificada en código ASCII, y al almacenarla en la base de datos MySQL, aparecían caracteres equivalentes y no se entendía la información. Para solucionar dicho problema se realizó un procedimiento de conversión de caracteres para UTF-8.
- El tercer problema apareció cuando se quiso hacer la aplicación concurrente de forma que todos los usuarios pudieran realizar los test a la vez y no de forma secuencial. Para solucionar este problema, se guardó la variable de sesión en la URL de esta manera, se sabe que usuario está en cada momento y en cada página .jsp.

Anexo E: Manual de instalación

Para el correcto funcionamiento de la aplicación se necesita tener instalado dos programas el gestor de la base de datos MySQL y el servidor donde corre la aplicación en este caso Tomcat, a continuación se especifica la instalación de cada uno de ellos:

- **Instalación y configuración de MySql:**

Paso 1. Bajarse el programa en la página <http://dev.mysql.com/get/Downloads>.

Paso 2. Ejecutar el archivo **setup.exe**.

Paso 3. Nos sale un menú para elegir el tipo de instalación, elegir **custom**, pulsamos Next.

Paso 4. Nos sale la pantalla de Custom Setup, elegir Next.

Paso 5. Nos aparecerá Wizard Complete, marcamos la casilla “Configure the MySql Server now y pulsamos Finish”

Paso 6. Nos aparecerá la ventana de configuración del servidor, marcamos “Detailed Configuration” y pulsamos Next.

Paso 7. En la siguiente ventana nos aparecerá la configuración de instancia, elegimos Developer Machine y pulsamos Next.

Paso 8. En la siguiente ventana pulsamos Multifunctional Database y pulsamos Next.

Paso 9. En la siguiente ventana pulsamos Manual Setting , marcamos Concurrent conexions, elegimos 800 y pulsamos Next.

Paso 10. En la siguiente ventana marcamos, Install As Windows Service y Include Bin Directory in Windows PATH y pulsamos Next.

Paso 11. En la siguiente ventana marcamos Modify Security Settings y elegimos un usuario y una contraseña y pulsamos Next.

- **Instalación y configuración de TomCat:**

Paso 1. Bajarse el programa en la página <http://tomcat.apache.org/download-60.cgi>.

Paso 2. Seguir la instalación dejando las opciones por defecto.

Paso 3. Establecer un usuario y una contraseña de acceso al servidor.

Paso 4. La ruta donde se encuentra será C:\Archivos de programa\Apache Software Foundation\Tomcat 6.0, creamos una carpeta llamada Flayer e introducimos los archivos del proyecto que se encuentran en la carpeta web de Flayer del cd.

Paso 5. Introducimos la carpeta que hemos creado con los archivos en el directorio Webapps de Tomcat.

Una vez instalados estos dos programas necesitamos bajarnos el conector entre MySQL y TomCat, no lo podemos bajar de <http://dev.mysql.com/downloads/connector/j/>, después tendríamos que introducirlo en la carpeta de bibliotecas de Tomcat, con ello ya podemos ejecutar la aplicación escribiendo en nuestro navegador <http://localhost:8080/Flayer>.

También la aplicación se ha desarrollado con el servicio No-ip, mediante el cual permite identificar el ordenador, con un nombre de dominio fácil de recordar. Una de sus principales ventajas consiste en que si el router se apaga, al volver a encenderse el servicio No-ip actualiza la ip automáticamente y así se puede seguir accediendo al dominio <http://flyer.no-ip.info/Flayer>.

Anexo F: Manual de usuario

Ventana de inicio de la aplicación: el usuario elige entre estudiante y profesor, contiene los siguientes botones:

- **Estudiante:** Ir a página de acceso de estudiante.
- **Profesor:** Ir a página de acceso de profesor.
- **Icono carta:** Contactar con el administrador de la aplicación.

Ventana acceso estudiante y Ventana de acceso profesor: el usuario se valida para entrar a la aplicación, contiene los siguientes botones:

- **Acceder:** Valida el usuario y la contraseña introducidos.
- **Borrar:** Borra el usuario y la contraseña introducidos.
- **Ayuda:** Ir a página de ayuda.

Ventana de test de Estilos de Aprendizaje y de test Ocean: el usuario puede responder a las preguntas del test, contiene los siguientes botones:

- **Siguiente:** Pasa a la siguiente página de test.
- **Elección:** Marca la opción de la pregunta.

Ventana elección de test Ocean individual o colaborativo: donde el usuario elige el tipo de test que va a realizar, contiene los siguientes botones:

- **Continuar:** Pasa a la siguiente pantalla, según la opción elegida.
- **Elección:** Marca la opción del tipo de test que se quiere realizar.
- **Salir:** Cerrar sesión en la aplicación.

Ventana de resultados del estudiante: el usuario obtiene los resultados del test, contiene los siguientes botones:

- **Cuéntanos que te ha parecido:** Pasa a la pantalla de test de Satisfacción.
- **Cerrar Sesión:** cierra sesión en la aplicación.

Ventana de test de Satisfacción: el usuario puede responder las preguntas del test, contiene los siguientes botones:

- **Elección:** Marca la opción elegida.
- **Cerrar Sesión:** cierra sesión en la aplicación.

Ventana de votación de preguntas: el usuario puede elegir la pregunta a votar, contiene los siguientes botones:

- **Elección:** Marca la opción de la pregunta
- **Siguiente:** Pasa a la siguiente pregunta.
- **Salir:** Cierra sesión en la aplicación.

Ventana de gestionar las preguntas: el usuario puede gestionar las preguntas generadas por la aplicación, contiene los siguientes botones:

- **Modificar:** Pasa a la página de modificar preguntas si no se ha marcado un patrón por el contrario, pasa a la página de combinar preguntas.
- **Siguiente:** Confirma la pregunta actual.
- **Insertar pregunta:** Pasa a la página de insertar preguntas.
- **Borrar pregunta:** Pasa la página de borrar preguntas.
- **Salir:** Cierra sesión en la aplicación.
- **Ayuda:** Pasa a la página de ayuda.

Ventana de modificar preguntas: el usuario puede modificar la pregunta elegida, contiene los siguientes botones:

- **Modificar:** Confirma la modificación elegida.

Ventana de combinar preguntas: el usuario puede combinar las preguntas elegidas, contiene los siguientes botones:

- **Combinar:** Confirma la combinación elegida.

Ventana de insertar preguntas: el usuario puede insertar preguntas en la base de datos, contiene los siguientes botones:

- **Insertar:** Inserta la pregunta en la base de datos.

- **Elección:** Marca la opción elegida.
- **Volver:** Vuelve a la página de gestionar las preguntas.

Ventana de borrar preguntas: el usuario puede borrar preguntas de la base de datos, contiene los siguientes botones:

- **Borrar:** Borra la pregunta de la base de datos.
- **Elección:** Marca la opción elegida.
- **Volver:** Vuelve a la página de gestionar las preguntas.