



ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA INFORMÁTICA

Curso Académico 2012/2013

Proyecto de Fin de Carrera

**EDU-SOA: UN ENFOQUE ORIENTADO A SERVICIOS PARA LA
CREACIÓN, GESTIÓN Y ENSEÑANZA DE CURSOS WEB**

Autor: Amparo Corbo Crespo

Tutor: Diana Pérez Marín

Resumen

La metodología orientada a servicios (SOA) supone muchos beneficios en campos, como en el financiero, ya que se elimina el uso de aplicaciones monolíticas que requieren operatividad en tiempo real. Esta metodología reduce dependencias y facilita las migraciones. También fomenta la reutilización de código y la eliminación de redundancias, facilita el desarrollo incremental y reduce los riesgos técnicos, además de reducir los costes de mantenimiento de las aplicaciones.

Sin embargo, en el campo educativo no hay muchos estudios de su aplicación y beneficios. Se estima que el uso de SOA para educación también sería beneficioso puesto que se pueden adoptar todas las ventajas vistas en el párrafo anterior.

En este proyecto, se ha desarrollado la herramienta Edu-Reviews siguiendo el concepto de SOA para demostrar sus beneficios en este campo. La metodología utilizada ha sido el ciclo de vida incremental.

Este proyecto se ha estructurado en dos niveles:

Nivel 1. Como desarrollo orientado a servicios aplicado a la educación. Con beneficios a nivel de desarrollo de código, bajo acoplamiento, facilidad de reutilización, facilidad a la hora de utilizar diferentes lenguajes, independencia de los componentes, etc.

Nivel 2. Como producto obteniendo un sistema educativo que se pretende que tenga beneficios tanto para los profesores como para estudiantes.

Se quiere destacar también el esfuerzo que se ha realizado en aprender tanto la filosofía SOA como el uso de las herramientas técnicas relacionadas con SOA como ESB, AJAX, Javascript, CSS, HTML, SQL, Java, Groovy.

El sistema desarrollado ha sido probado por nueve usuarios que lo han valorado con un 9 de media en la escala de 0 (total disconformidad) a 10 (total conformidad), por lo que se puede considerar que el proyecto cumple los objetivos planteados tanto en el nivel de desarrollo con el uso de SOA como en el nivel de producto con EDU-REVIEWS.

ÍNDICE

1. Introducción	1
2. Objetivos	7
2.1 Descripción del problema	7
2.2 Objetivos	7
2.3 Estudio de alternativas.	7
2.3.1 Enterprise Service Bus (ESB)	8
2.3.2 Lenguajes de programación para Servicios Web	11
2.3.3 Gestores de bases de datos.	12
2.3.4 Lenguajes web.....	14
2.3.5 Lenguajes para el intercambio estructurado de información.	16
2.3.6 Servidores Web / HTTP	18
2.3.7 Lenguajes de Programación.	20
2.4 Metodología empleada.....	22
3. DESCRIPCIÓN INFORMÁTICA	23
3.1 Análisis de requisitos	23
3.1.1 Requisitos Funcionales.....	23
3.1.2 Requisitos No Funcionales.....	24
3.1.3 Modelos de Casos de Uso	24
3.2 Diseño	31
3.2.1 Arquitectura de alto nivel.....	31
3.3 Diagrama de entidad relación	32
3.3.1 Modelo entidad-relación	32
3.3.2 Modelo relacional.....	32
3.3.3 Descripción de las tablas.....	33
3.4 Algoritmo	36
3.4.1 Profesor	36

3.4.2 Estudiante	39
3.5 Descripción de los módulos	42
3.5.1 Descripción de los módulos del profesor	43
3.5.2 Descripción de los módulos del estudiante	47
3.5.3 Descripción de los módulos compartidos entre el profesor-estudiante:.....	53
4. Pruebas	57
4.1 Pruebas Unitarias	57
4.1.1 Pruebas de caja blanca.....	58
4.1.2 Pruebas de caja negra	61
4.2 Pruebas de integración.	66
4.3 Pruebas de validación	68
4.4 Pruebas de aceptación	74
5. Conclusiones y Trabajo Futuro	77
5.1 Conclusiones	77
5.2 Trabajo Futuro	79
BIBLIOGRAFÍA	81
ANEXO 1: Manual de usuario	85
ANEXO 2: Manual de instalación.....	91

ÍNDICE DE FIGURAS

Figura 1. Interfaz para el profesor creando una pregunta de rellenar huecos.....	4
Figura 2. Interfaz para el estudiante contestando una pregunta.	5
Figura 3. Tecnologías Edu-Reviews.....	21
Figura 4. Modelo incremental [Piattini et al, 2003].	22
Figura 5. Casos de uso de la herramienta.	25
Figura 6. Casos de uso del profesor.....	26
Figura 7. Casos de uso del estudiante.....	29
Figura 8. Esquema SOA de Edu-Reviews.....	31
Figura 9. Diagrama entidad relación E/R.....	32
Figura 10. Mule Studio ESB.	42
Figura 11. Módulos Profesor.....	43
Figura 12. Módulo inserción lección.....	43
Figura 13. Módulo de borrado de lección.....	44
Figura 14. Módulo de selección de preguntas/respuestas.....	44
Figura 15. Módulo de inserción y modificación de preguntas/respuestas.....	45
Figura 16. Módulo de borrado de preguntas/respuestas.	46
Figura 17. Módulos Estudiante.....	47
Figura 18. Módulo de registro.	48
Figura 19. Módulo de selección de preguntas.	49
Figura 20. Módulo de comprobación de la respuesta introducida.....	50
Figura 21. Módulo de recompensa.	52
Figura 22. Módulo de recarga de las preguntas de una lección.....	53
Figura 23. Módulo de acceso profesor/estudiante.	54
Figura 24. Módulo de selección de lección.	55
Figura 25. Diagrama de caja blanca [Piattini et al, 2003].	57
Figura 26. Diagrama de caja negra [Piattini et al, 2003].....	58
Figura 27. Prueba de caja blanca: algoritmo profesor.	58
Figura 28. Prueba de caja blanca: algoritmo estudiante.	60
Figura 29. Voki. Bienvenida a Edu-Reviews.	67
Figura 30. Voki. Administración y manejo de lecciones.	67
Figura 31. Voki. Administración y manejo de preguntas.....	67
Figura 32. Voki. Elegir lección.	68

Figura 33. Voki. Comprobar respuesta.....	68
Figura 34. Crear nueva cuenta de estudiante.....	69
Figura 35. Autenticación del usuario en Edu-Reviews.	69
Figura 36. Seleccionar, crear o eliminar una lección.	70
Figura 37. Insertar nombre de lección ya existente.....	70
Figura 38. Crear, modificar, eliminar y seleccionar preguntas y respuestas.....	71
Figura 39. Crear pregunta con respuesta vacía.....	71
Figura 40. Crear pregunta con varias respuestas.....	71
Figura 41. Seleccionar lección.	72
Figura 42. Recargar una lección.....	72
Figura 43. Seleccionar preguntas.	73
Figura 44. Comprobar respuesta.....	73
Figura 45. Cuestionario Satisfacción de Edu-Reviews.	74

ÍNDICE DE TABLAS

Tabla 1. Comparativa ESBs.	11
Tabla 2. Comparativa de plataformas de servicios web.	12
Tabla 3. Comparativa gestores de bases de datos.....	14
Tabla 4. Comparación lenguajes Web.....	16
Tabla 5. Lenguajes de intercambio de información.	17
Tabla 6. Comparación de Servidores Web.	20
Tabla 7. Comparación lenguajes de programación.	21
Tabla 8. Tabla sql USER.	33
Tabla 9. Ejemplo tabla USER.....	33
Tabla 10. Tabla sql LESSON.	34
Tabla 11. Ejemplo tabla LESSON.....	34
Tabla 12. Tabla sql QUESTION.	34
Tabla 13. Ejemplo tabla QUESTION.....	35
Tabla 14. Tabla sql ANSWER.	35
Tabla 15. Ejemplo tabla ANSWER.....	35
Tabla 16. Tabla sql USERSESSION.....	35
Tabla 17. Ejemplo tabla USERSESSION.	35
Tabla 18. Tabla sql AWARD.	36
Tabla 19. Ejemplo tabla AWARD.....	36
Tabla 20. Tabla resultado pruebas aceptación.....	75

ÍNDICE DE ACRÓNIMOS

API *Application Programming Interface.*

BBDD *Database.*

CDDL *Common Development and Distribution License.*

CPAL *Common Public Attribution License.*

DHTML *Dynamic HTML.*

ESB *Enterprise Service Bus.*

GPL *General Public License.*

JB1 *Java Business Integration.*

JAX-WS *Java API for XML Web Services.*

JB1 *Java Business Integration.*

PL/SQL *Procedural Language/Structured Query Language.*

RIA *Rich Internet Applications.*

SQL *Structured Query Language.*

SOA *Service-oriented Architecture.*

1. INTRODUCCIÓN

El objetivo de este proyecto es familiarizarse con las Arquitecturas Orientadas a Servicios (SOA). Las arquitecturas SOA [Hurwitz et al. 2009] han obtenido en los últimos años muchos beneficios en dominios empresariales.

SOA es un concepto de arquitectura de software (definido a partir de un conjunto de patrones, principios y prácticas) que define la utilización de servicios para dar soporte a los requisitos del negocio.

SOA permite la creación de sistemas de información altamente escalables que reflejan el negocio de la organización empleando la utilización de servicios (no exclusivamente servicios web) lo que facilita la interacción entre diferentes sistemas internos y/o externos [HTTP 1].

Los Principios de la Orientación a Servicios son los siguientes [Erl, 2007]:

- **Los Servicios deben proporcionar una interfaz o contrato de servicios:** Define las obligaciones entre el proveedor y el consumidor del servicio: nombre del servicio, forma de acceso, funcionalidad, datos de entrada y salida.
- **Los Servicios deben tener acoplamiento débil:** Es decir, que los servicios tienen que ser independientes unos de otros. Siendo, a su vez, más flexibles. Esto facilita el ensamblaje de los servicios y su reutilización para la construcción de servicios más complejos a partir de los simples.
- **Los Servicios no deben tener estado:** No debe guardar ningún tipo de información. En la llamada al servicio, los servicios SOA reciben toda la información necesaria para generar la respuesta.
- **Los Servicios deben permitir la composición:** Todo servicio debe ser construido de forma automática pudiendo ser utilizado posteriormente para construir servicios genéricos de más alto nivel.
- **Los Servicios deben de ser autónomos:** Todo servicio debe tener su propio entorno de ejecución. De forma que se asegura que sea reutilizable desde el

- **Los Servicios deben poder ser descubiertos:** Todo servicio debe poder ser descubierto para que pueda ser utilizado. Se consigue evitar la creación de **servicios con las mismas funcionalidades.**

Este tipo de arquitectura responde a la necesidad de las empresas de adaptarse a los constantes cambios, y por tanto, ser capaces de adaptar sus procesos de negocio.

Las plataformas de e-Learning son espacios virtuales que facilitan el aprendizaje a distancia de diferentes materias. En estas plataformas se oferta a los usuarios una serie de herramientas (foros de discusión, correo electrónico, chats, páginas web) como soporte de los procesos de aprendizaje-enseñanza [HTTP 2].

Los beneficios de las plataformas de educación son los siguientes:

- Flexibilidad en cuanto a la materia de estudio.
- Permite libertad en cuanto al tiempo y al ritmo de aprendizaje.
- Se eliminan las barreras geográficas.
- Permite que la interacción entre profesor/estudiantes se realimente (con las herramientas indicadas anteriormente).

Los posibles beneficios de SOA en educación se traducen en todos los beneficios que nos aporta SOA a nivel técnico: reutilización de servicios (permitiendo acoplar unos servicios con otros para construir servicios más complejos), bajo acoplamiento (mayor independencia)...El uso de una arquitectura orientada a servicios permitiría la reutilización de gran cantidad de material y herramientas educativas que actualmente se encuentran replicadas y no aprovechadas.

En este trabajo, se quiere estudiar el impacto en coste de aprendizaje respecto a los beneficios obtenidos del desarrollo siguiendo una Arquitectura Orientada a Servicios. En particular, se desarrollarán dos aplicaciones: una para el profesor en la que introducirá las lecciones con preguntas de repaso y otra para el estudiante, que podrá repasar las lecciones.

El sistema creado se ha llamado Edu-Reviews. Se ha elegido este nombre porque engloba en sí mismo la finalidad del proyecto: Edu (Educación) – Reviews (Repaso), es decir, se trata de una herramienta de repaso educativo. En este caso, se ha enfocado al repaso de idiomas, concretamente al inglés.

Se ha elegido la temática de idiomas por la situación económica actual. Ya que la orientación de los mercados y de los negocios es el crecer a nivel internacional. Para ello es requisito necesario dominar idiomas.

Edu-Reviews se estructura a partir de la herramienta Mule ESB (*Enterprise Service Bus*). Empleando este tipo de herramienta se orienta nuestro proyecto hacia el uso de arquitecturas orientadas a servicios (SOA) ya que Mule ESB se basa en este concepto. Esto permite crear aplicaciones mediante la generación de servicios que serán consumidos por el resto de aplicaciones [HTTP 3].

Con Mule ESB se pueden integrar los procesos de negocio de forma transparente consiguiendo que estén desacoplados. La forma de comunicarse entre ellos se realiza a través del bus de integración (ESB).

Este bus (ESB) actúa como mediador entre diferentes servicios de negocio y se encarga de conectarlos entre sí mediante mensajes de comunicación estándares. La ventaja de un bus es que únicamente tenemos que definir la manera de integrar un servicio con el bus por lo que nos abstraemos de la configuración del resto de servicios.

Por tanto, este ESB permite la conexión con bases de datos, servicios... que no forman parte estrictamente de nuestra aplicación, es decir, son elementos externos y es posible conectarse a ellos conociendo su interfaz a través de mensajería.

Además de la herramienta Mule ESB, se ha empleado MySQL como sistema gestor para administrar la base de datos. Los lenguajes de programación utilizados son Java, Javascript, CSS, SQL, Groovy y HTML. Para desarrollo web, se ha utilizado la técnica AJAX, y como lenguaje de intercambio JSON.

Edu-Reviews se estructura en dos niveles, por un lado la parte visible por el usuario que corresponde con la interfaz gráfica de una página web y por otro lado, la parte no visible, responsabilidad del ESB, que se ocupa de las solicitudes realizadas desde la interfaz de usuario.

En la parte visible del usuario, se presenta una interfaz gráfica distinta según del tipo de usuario, es decir, profesor o estudiante por sus diferentes actividades con el sistema. El profesor organiza sus preguntas y respuestas en lecciones. El tipo de preguntas son para rellenar con huecos. Por ejemplo, como se muestra en la Figura 1, el profesor añade una pregunta en la que sustituirá cada palabra por un grupo de puntos. En la parte de respuesta escribe las palabras correspondientes a cada uno de esos grupos de puntos.

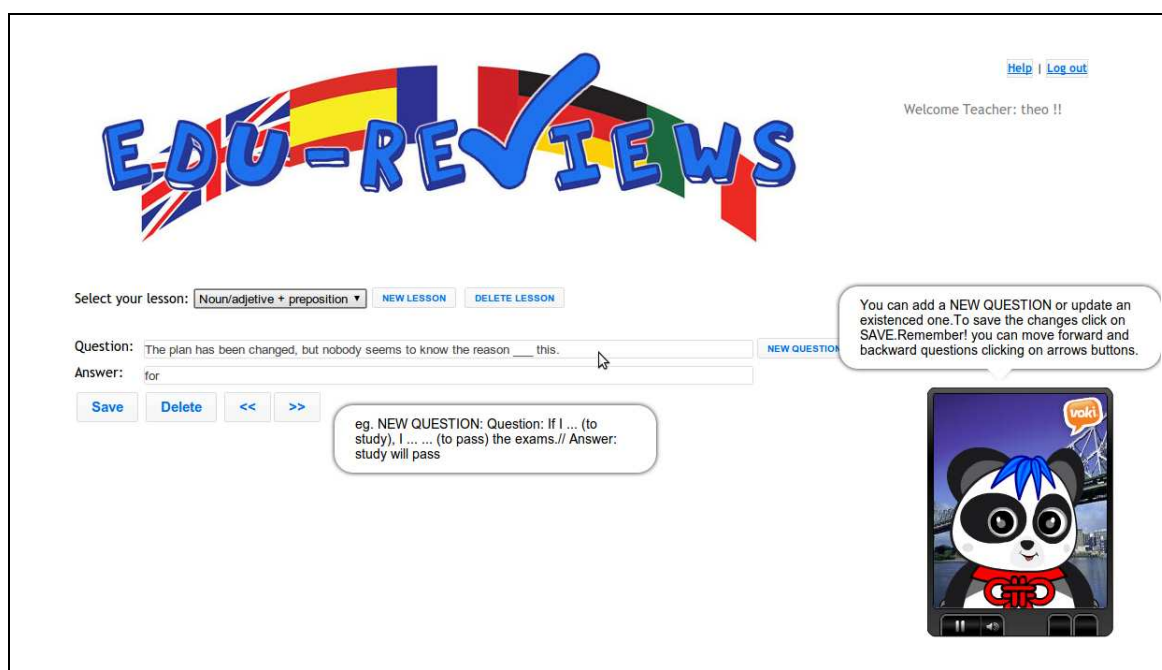


Figura 1. Interfaz para el profesor creando una pregunta de rellenar huecos.

El estudiante, podrá seleccionar las lecciones que estén disponibles e ir contestando las preguntas de cada una de las lecciones que haya insertado el profesor. Como se puede observar en la Figura 2, el estudiante tendrá que responder debajo de la pregunta, en este caso, con dos palabras.

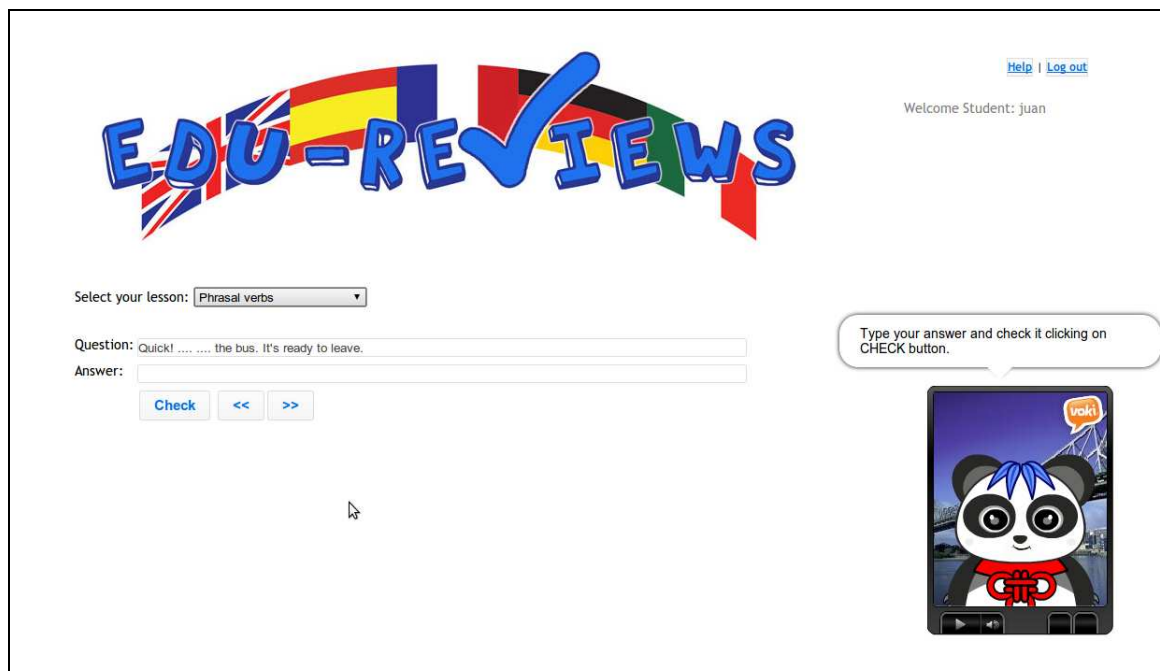


Figura 2. Interfaz para el estudiante contestando una pregunta.

Otro de los objetivos es aplicar esta herramienta al dominio educativo. En este caso, la temática es el aprendizaje y perfeccionamiento de la gramática inglesa, permitiendo extenderse a distintos idiomas, si se deseara por parte del profesor.

Además se emplea un Voki [HTTP 4] que acompañará al profesor y al estudiante mientras estén trabajando y que servirá como elemento de apoyo en la comprensión de las actividades. El voki es una herramienta muy útil para e-Learning (aprendizaje a distancia). Se trata de un widget gratuito que produce un "voki" o personaje (avatar) que se mueve y habla (bien texto que le introducimos escrito, bien grabaciones de audio o incluso nuestra propia voz).

Un ejemplo de cómo utilizar la herramienta Edu-Reviews sería el siguiente: el profesor crea un conjunto de lecciones con sus correspondientes preguntas y respuestas con diferentes niveles y objetivos; y, el estudiante accede a la herramienta para repasar y/o adquirir nuevos conceptos presentes en las diferentes lecciones (introducidas previamente por el profesor).

Edu-Reviews ha sido probada con 9 usuarios de diferentes perfiles, para comprobar si es usable. Los usuarios han valorado la herramienta completando un cuestionario de satisfacción con 8 preguntas. El análisis de estos resultados ha proporcionado un valor

de 9 en la escala 0 a 10 donde 0 significa no usable y 10 significa usable. Es decir, con un 9, los usuarios consideran que la herramienta es eficiente, eficaz y satisfactoria (usable).

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

2. OBJETIVOS

2.1 DESCRIPCIÓN DEL PROBLEMA

La necesidad de este proyecto radica en proporcionar mecanismos de repaso interactivo empleando arquitecturas orientadas a servicios (SOA) en el dominio educativo.

Para ello se ha empleado un tipo de herramienta que proporcione soporte a este tipo de arquitecturas que más adelante describiremos como Enterprise Service Bus (ESB).

Se ha trabajado para realizar una aplicación sencilla e intuitiva tanto para el profesor como para el estudiante, configurable por parte del profesor según el nivel de dificultad a emplear en las preguntas y que además permita al estudiante mejorar su nivel de inglés repasando.

2.2 OBJETIVOS

Los objetivos son los siguientes:

1. Desarrollar una aplicación para el profesor que le permita crear preguntas de rellenar huecos de forma intuitiva y con la guía de un avatar animado.
2. Desarrollar una aplicación para el estudiante que le permita repasar preguntas de rellenar huecos en inglés de forma intuitiva y con la guía de un avatar animado.
3. Realizar el desarrollo según una Arquitectura Orientada a Servicios de una aplicación educativa con tecnología actual como ESB, AJAX, JSON midiendo el coste de aprendizaje respecto a los beneficios obtenidos.

2.3 ESTUDIO DE ALTERNATIVAS.

En este apartado se comparan las distintas tecnologías que se pueden aplicar para el proyecto y se justifican las decisiones tomadas.

Se describirán las herramientas ESB, los lenguajes de programación para servicios web, los gestores de bases de datos, los lenguajes web, los lenguajes para el intercambio estructurado de información, los servidores HTTP/web y los lenguajes de programación.

2.3.1 ENTERPRISE SERVICE BUS (ESB)

Es un modelo de arquitectura de software que proporciona interoperabilidad entre distintos protocolos de comunicación. Actúa de intermediario entre las aplicaciones de negocio (evitando que el consumidor y el proveedor dialoguen directamente). Mediante adaptadores consigue redireccionar y transformar los mensajes entre dichas aplicaciones en la arquitectura orientada a servicios (SOA).

Al actuar como bus de mensajería, elimina la necesidad de sistemas con conectividad punto a punto. Cuando un sistema necesita comunicarse con otro, deposita el mensaje con datos en el bus (ESB) y este bus se encarga de calcular la forma de que llegue a su destino. Cualquier transformación que sea necesaria, se realizará.

Las características básicas que debe proporcionar un ESB son las siguientes [Davis, 2009]:

- Adaptadores/Conectividad de datos.
- Transformación de datos y traducción de mensajes.
- Presencia de adaptadores a múltiples plataformas.
- Enrutamiento inteligente.
- Gestión de servicios.
- Monitorización y bitácora.
- Orquestación y coreografía de procesos de negocio.
- API personalizada.
- Servicios temporales.

Como ejemplos de ESB comerciales están Progress Sonic ESB, Oracle Enterprise Service Bus. Mientras que Apache ServiceMix, Mule, OpenESB son implementaciones software libre (Open Source) [HTTP 5].

Este proyecto se limita a los ESB que son de tipo Open Source [HTTP 6].

APACHE ServiceMix

Desarrollado por Apache Software Foundation. Es uno de los primeros productos ESB compatible con la especificación Java Business Integration (JBI, establecida por Java para la integración de servicios, desde 2005).

Tiene docenas de adaptadores y componentes y proporciona soporte para cualquier protocolo. Permite la creación de flujos de datos complejos utilizando componentes de patrones de integración empresarial. Es un proyecto muy activo con versiones frecuentes [HTTP 7]. Publicado bajo la licencia Apache Software License.

Las principales características son [HTTP 8]:

- Ligero, flexible y fácilmente acoplable.
- Cuenta con soporte Spring integrado.
- Certificado en base a la especificación JBI.
- Utiliza Apache ActiveMQ como sistema de mensajería fiable.
- Mensajería, enrutado y Patrones de Integración con Apache Camel.
- WS-* and RESTful web services con Apache CXF
- Integración flexible entre componentes mediante Apache ServiceMix NMR (Normalized Message Router).
- Servidor OSGi basado en Apache Karaf.
- Motor de WS-BPEL con Apache ODE.

Mule ESB

Desarrollado por MuleSoft. Tiene muchas soluciones de conectividad (componentes y adaptadores). Sus fortalezas las encontramos en la transformación, encaminamiento y seguridad.

Está bien documentado, es un proyecto maduro y totalmente probado. Muchos de los servidores de aplicaciones lo soportan [HTTP 3; Dossot &D'Emic, 2010]. Se publica bajo licencia Common Public Attribution License (CPAL).

Las principales características de Mule ESB son:

- Ligero y altamente escalable.

- Creación y alojamiento de servicios: Se exponen diferentes servicios para ser consumidos por otras aplicaciones de forma que puedan ser reutilizables.
- Mediación y enrutamiento de servicios (basados en reglas).
- Admite multitud de formatos de mensajes: SOAP, REST...
- Permite comunicación síncrona y asíncrona.
- Registro de auditoría y monitorización de la aplicación.
- Diferentes proveedores para la autenticación y autorización.
- Multiplataforma.
- Compatible en múltiples servidores y bases de datos.
- Fácilmente integrable con otros servicios: Facebook, Amazon, Paypal, etc.
- Soporta diferentes lenguajes de programación: Java, Groovy, Javascript, etc.
- Soporta la mayoría de los patrones de integración de aplicaciones empresariales.

Open ESB

Inicialmente diseñado y desarrollado por Sun Microsystems y Seebeyond. Este proyecto actualmente se mantiene y mejora desde la comunidad OpenESB [HTTP 9]. Este ESB se basa en Java. Puede ser utilizado de forma fácil y eficiente tanto para la integración de aplicaciones empresariales (EAI) como para arquitecturas orientadas a servicios (SOA).

Se basa en el estándar JBI (Java Business Integration), XML, XML Schema, WSDL, BPEL que le proporciona la simplicidad, la eficiencia, la durabilidad a largo plazo, permitiendo la reducción de costes a la hora de integrar nuevos componentes.

OpenESB ofrece una completa serie de herramientas para diseñar, desarrollar, probar y desplegar aplicaciones. Este proyecto es fiable, escalable, y permite la reutilización de componentes. Está publicado bajo la licencia CDDL.

Característica	Apache ServiceMix	Mule ESB	Open ESB
Licencia libre	Si	Si	Si
Compatibilidad JBI	Si	No	Si
Documentación	Documentado	Muy documentado	Poco documentado
Multiplataforma	Si	Si	Si

Tabla 1. Comparativa ESBs.

Se decide trabajar con Mule ESB por ser multiplataforma y por la buena documentación de la que dispone.

2.3.2 LENGUAJES DE PROGRAMACIÓN PARA SERVICIOS WEB

Un servicio web se define como cualquier sistema software diseñado para interoperar entre máquinas en la Web independientemente de la plataforma en la que se desarrollen. Este sistema tiene como objetivo ofrecer una serie de servicios al cliente listados en una API. Son ejecutados previa solicitud del cliente en un sistema de hosting remoto [HTTP 10].

Se van a estudiar los posibles lenguajes para implementar los servicios web, se han elegido los siguientes según la plataforma:

Plataforma J2EE (JAVA)

Es una plataforma de programación para desarrollar y ejecutar aplicaciones escritos en Java y pertenece a Sun Microsystems. Tiene varias especificaciones API: JDBC, RMI, JAX-WS...

Para la creación de servicios web se emplea Java API for XML Web Services (JAX-WS). Esta API usa anotaciones y es de tipo Open Source.

Plataforma .NET (Microsoft)

Es una plataforma de desarrollo y ejecución de aplicaciones en lenguajes de programación .NET (como son VisualBasic, C#...). Desarrollada por Microsoft en respuesta al creciente mercado en los negocios de entornos web [HTTP 11].

Característica	.NET	J2EE
Lenguaje/s de programación	C#, Visual Basic, ASP...	Java
Plataforma	Windows	Multiplataforma
Cientes Web	ASP.NET	JSP / Servlets

Tabla 2. Comparativa de plataformas de servicios web.

En la Tabla 2 se muestra la comparativa entre las plataformas de creación de servicios web, en este caso se ha optado por J2EE al ser multiplataforma.

2.3.3 GESTORES DE BASES DE DATOS.

En este apartado, se revisan varios gestores de bases de datos relacionales.

MySQL

Es un sistema de gestión relacional de bases de datos, mutihilo y multiusuario, que funciona bajo licencia GNU GPL [Gilfillan, 2003]. Además, es un sistema multiplataforma, y se caracteriza, por tener una potencial relativamente alto. Al tener licencia libre, esto le coloca en una posición privilegiada para ser usado en aplicaciones de pequeño y mediano tamaño. El lenguaje de acceso a datos es Structured Query Language (SQL).

Las ventajas de este gestor son:

- Software de libre licencia. Dispone de Licencia GPL (General Public License) gratuita y abierta, así como también de licencia comercial.

- Manejable y administrable, en aplicaciones pequeñas y medianas.
- Es multiplataforma. Funciona sobre Linux, Windows, Solaris.
- Proporciona soporte y agrupación de transacciones.

Los inconvenientes de elegir este gestor son:

- Futuro incierto tras la adquisición de MySQL por parte de Oracle.

ORACLE

Este sistema de gestión relacional de bases de datos está orientado a grandes aplicaciones y empresas. Se trata de un software comercial por lo que este producto es bajo pago. Es el principal proveedor y distribuidor de aplicaciones de gestión de base de datos. El lenguaje de acceso a datos es Procedural Language/Structured Query Language (PL/SQL) [Freeman, 2008].

Las ventajas de elegir este gestor son:

- Gran escalabilidad.
- Soporte para cualquier dificultad o problema en el mantenimiento o creación de la base de datos.
- Estabilidad de la plataforma.
- Soporte multiplataforma.

Los inconvenientes de elegir este gestor son:

- Licencia propietaria.
- Alta complejidad de administración.

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 [HTTP 12].

Dirigido por una comunidad de desarrolladores denominada el PGDG (PostgreSQL Global Development Group). El lenguaje de acceso a datos es SQL (Structured Query Language).

Las ventajas de elegir este gestor son:

- Buena escalabilidad.
- Estabilidad de la plataforma.
- Soporte multiplataforma.

Las desventajas de elegir este gestor son:

- Soporte en línea: Hay foros oficiales, pero no hay una ayuda obligatoria.
- Consumo alto de recursos.

Característica	MySQL	Oracle	PostgreSQL
Licencia libre	Si	No	Si
Escalabilidad	Si	Si	Si
Sistema Operativo	Multiplataforma	Multiplataforma	Multiplataforma
Consumo de recursos	Media	Alta	Alta

Tabla 3. Comparativa gestores de bases de datos.

Según la comparativa mostrada en la Tabla 3, se elige el gestor de base de datos relacional MySQL al tratarse de un sistema con licencia libre, multiplataforma y cuyo consumo de recursos es menor comparativamente que PostgreSQL [HTTP 12].

2.3.4 LENGUAJES WEB

Javascript

Es un lenguaje de programación interpretado, en el lado del cliente, muy utilizado debido a su compatibilidad con la mayoría de los navegadores más actuales.

Con Javascript podemos definir interacciones 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 [Del Caso, 2010].

Se emplea JQuery (que es una biblioteca de Javascript) que ofrece un conjunto de funcionalidades que simplifican el uso de Javascript [HTTP 14].

Cascading Style Sheets (CSS)

Lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL [HTTP 15].

Se obtienen mejores resultados que utilizando únicamente páginas en HTML, pudiendo incluir tipos de letra, márgenes, fondos de colores... de forma más estructurada y reutilizable.

Applets en Java

Los applets en Java permiten ejecutar código en los clientes que visualizan una página Web, se transfieren con las páginas Web y el navegador los 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 se pueden abrir directamente ventanas secundarias, controlar Frames, formulario, capas, etc... [HTTP 16].

Asynchronous JavaScript and XML (AJAX)

Es un refinamiento de Dynamic HTML (DHTML). Utiliza todas las herramientas de DHTML (Javascript, CSS, DOM...), sumándole el objeto XMLHttpRequest para

obtener información de manera asíncrona y refrescar solo la parte necesaria de la página web sin perder nada del contexto [HTTP 17].

Positivo

- La mejor tecnología Rich Internet Applications (RIA) hasta el momento.
- Está en su mejor momento para la industria.
- No necesita descargar un complemento.

Negativo

- Incompatibilidades entre navegadores (cada vez menos).
- Desarrollo con JavaScript.

Con todo lo anterior, vemos que hoy en día, una de las mejores posibilidades y más nueva es la utilización de AJAX.

Característica	JavaScript	Css	Applets de Java	AJAX
Aprendizaje	Sencillo	Medio	Complejo	Medio
Orientado a Objetos	Si	No	Si	Si
Multi-plataforma	Si	Si	Si	Si
Código	Organizado	Estructurado	Estructurado	Organizado
Software	Libre	Libre	Libre	Libre

Tabla 4. Comparación lenguajes Web.

Se ha decidido utilizar AJAX junto con CSS, JavaScript al tratarse de lenguajes y tecnologías complementarias.

2.3.5 LENGUAJES PARA EL INTERCAMBIO ESTRUCTURADO DE INFORMACIÓN.

XML (eXtensible Markup Language)

XML es un lenguaje de marcas parecido a HTML. Está diseñado para almacenar datos, no para mostrarlos. Las etiquetas no están predefinidas, se puede definir las propias.

Se trata de un lenguaje autodescriptivo. Está recomendado por W3C [HTTP 15].

Ejemplo de XML:

```
<clientes>
<cliente>
  <nombre> Juan </nombre>
  <apellido> Corbo </apellido>
</cliente>
</clientes>
```

JSON (JavaScript Object Notation)

JSON es un formato de intercambio de datos en texto ligero. Es independiente del lenguaje, se auto-escribe y es fácil de aprender. Utiliza la sintaxis de JavaScript para la descripción de objetos de datos, pero sigue siendo el idioma y la plataforma independiente [HTTP 18].

Ejemplo de JSON:

```
{
"clientes": [
{ "nombre": "Arancha" , "apellido": "Garrido" },
{ "nombre": "Covadonga" , "apellido": "Cháñez" } ]
}
```

Característica	XML	JSON
Formato	Texto plano	Texto plano
Lenguaje auto-descriptible	Si	Si
Jerárquica	Si	Si
Analizar mediante Javascript	Si	Si
Rapidez de lectura/escritura para aplicaciones AJAX	No	Si

Tabla 5. Lenguajes de intercambio de información.

En este caso se emplea JSON como lenguaje de intercambio de datos por su simplicidad ya que facilita el análisis en aplicaciones AJAX.

2.3.6 SERVIDORES WEB / HTTP

Servidor ISS

Este servidor, es de distribución no libre por parte de Microsoft, por lo que para su uso, se requiere de licencias software. Permite la publicación de páginas web y ftp, tanto en local como en red pero para su uso se necesita que corra bajo la plataforma Windows Server, con licencia implícita [Vera, 2008].

Las ventajas de elegir este servidor son:

- Si se tiene una licencia para un sistema operativo Microsoft, (Windows Server) este software ya viene incluido.
- Facilidad de manejo.
- Aconsejable para usuarios con pocos conocimientos que busquen un servidor web, fácil y usable bajo Windows.

Los inconvenientes de elegir este servidor son:

- Solo se puede usar bajo licencia.
- No es compatible con otros sistemas que no sean Microsoft.
- No permite una administración para temas complejos.

Servidor GlassFISH

Es un servidor de aplicaciones de código abierto compatible con Java EE. Desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation (con su versión comercial). Permite servir páginas web, ftp y tecnología Java (implementa la plataforma Java EE, debido a esto, soporta las últimas versiones de tecnologías como: JSP, JSF, Servlets, Ejes, Java API para Servicios Web (JAX-WS), Arquitectura Java para enlaces XML (JAXB), etc....). Además, es de libre distribución bajo una licencia GPL.

La tecnología de este servidor deriva principalmente de tecnología usada por Sun y Apache Tomcat [HTTP 19].

Las ventajas de elegir este servidor son:

- Multiplataforma, por lo que se puede utilizar bajo distintos sistemas operativos.
- No requiere de licencia propietaria, por lo que no es necesario adquirir una licencia para su uso.

Los inconvenientes de elegir este servidor son:

- Menor cantidad de documentación y ejemplos en la red.
- Proyecto reciente con una menor consolidación.

Servidor Web Apache

Es un servidor Web robusto, cuya implementación se realiza de forma colaborativa, con funcionalidades equivalentes a las de los servidores comerciales. Este servidor es de código abierto y está respaldado por una gran comunidad de desarrolladores, denominados Apache Group, que resuelven los posibles fallos y vulnerabilidades de este software.

Su creación es anterior a los dos anteriormente citados, y la experiencia en problemas y vulnerabilidades en la mayoría de los casos está documentada en la web [HTTP 20].

Las ventajas de elegir este servidor son:

- Gran uso en la red, un porcentaje alrededor del 60% de las páginas servidas en la red, son servidas por Apache.
- Software Libre y código abierto. Multiplataforma.
- Configuración fácil y sencilla a través de fichero de configuración.

Jetty

Jetty es un servidor web basado en Java y un contenedor de servlets preparado para ser embebido en nuestras aplicaciones. Se publica como un proyecto de software libre bajo la licencia Apache 2.0 y la licencia pública de Eclipse 1.0.

Jetty está disponible para su distribución y uso comercial. El núcleo del proyecto se encuentra alojado por la Fundación Eclipse a partir de la versión 7 y de las versiones 6 para abajo en Codehaus [HTTP 20].

Las ventajas de elegir este servidor son:

- Sencillo, eficiente, asíncrono, empotrable y pluggable.
- Ligero (apropiado para ofrecer servicios Web) y eficiente.
- Bajo coste de mantenimiento.

	Servidor ISS	GlassFISH	Apache	Jetty
Multiplataforma	Si	Si	Si	Si
Estabilidad	Alta	Alta	Alta	Alta
Licencia	Libre	Libre	Libre	Libre

Tabla 6. Comparación de Servidores Web.

Se elige Jetty porque es ligero, 100% Java y viene integrado con el ESB seleccionado que es Mule.

2.3.7 LENGUAJES DE PROGRAMACIÓN.

Ruby

Lenguaje creado por Yukihiro Matsumoto. Mezcla partes de lenguajes como Perl, Smalltalk, Eiffel, Ada, y Lisp para crear este lenguaje que incorporara tanto la programación funcional como la programación imperativa [HTTP 21].

Características:

- Lenguaje script orientado a objetos.
- Basado en C.
- Utiliza la librería propia de Ruby.
- Se usa en aplicaciones de tiempo real.

Groovy

Lenguaje de programación desarrollado y mantenido por SpringSource y la Comunidad Groovy [HTTP 22].

Características:

- Lenguaje script orientado a objetos.

- Sintaxis similar a Java, compatible con Java.
- Utiliza la biblioteca de Java con algunos añadidos propios.
- La curva de aprendizaje es menor si previamente se conoce Java.

Algunas de las diferencias entre Ruby y Groovy se muestran en la siguiente tabla [HTTP 23]:

	Ruby	Groovy
Orientado a objetos	Si	Si
Basado en el lenguaje	C	Java
Desarrollo aplicaciones en tiempo real	Si	No

Tabla 7. Comparación lenguajes de programación.

En la Tabla 7 se muestran algunas de las diferencias entre ambos lenguajes, finalmente se elige Groovy por su compatibilidad con Java.

A modo de resumen, en esta figura se presentan todas las tecnologías empleadas en la creación del proyecto Edu-Reviews:

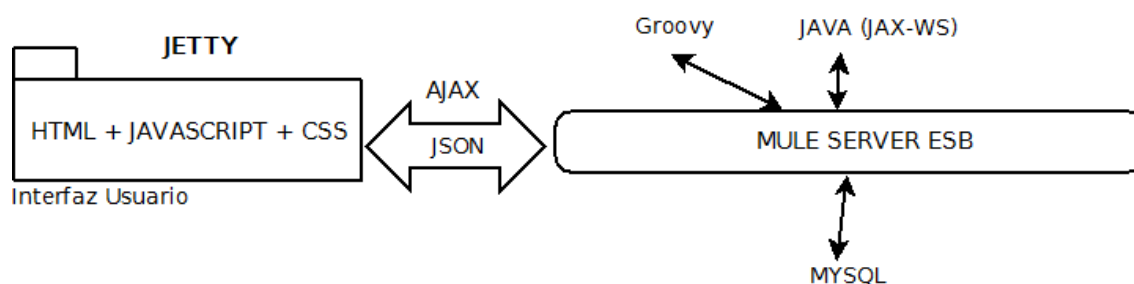


Figura 3. Tecnologías Edu-Reviews

En esta figura se puede observar que la Interfaz de usuario se escribe en HTML, Javascript y CSS y se publica con el servidor HTTP Jetty. La comunicación con el ESB se establece con AJAX mediante mensajería donde los objetos que se pasan están en formato JSON. En la parte del ESB, se han utilizado tecnologías como MySQL, JAVA, Groovy.

2.4 METODOLOGÍA EMPLEADA

La metodología de trabajo utilizada en este proyecto sigue el ciclo de vida software incremental [Piattini, 2003], ya que permite crear el sistema añadiendo nuevas funcionalidades o requisitos (en incrementos).

Cada uno de los incrementos actualiza el sistema. La ventaja de este método es que no es necesario tener todos los requisitos desde un principio. El ciclo de vida de este modelo, como se observa en la Figura 4, se compone de dos partes:

- La primera parte: especificación de requisitos, análisis y diseño preliminar.
- La parte de incrementos: compuesta por fases de diseño detallado, codificación y mantenimiento.

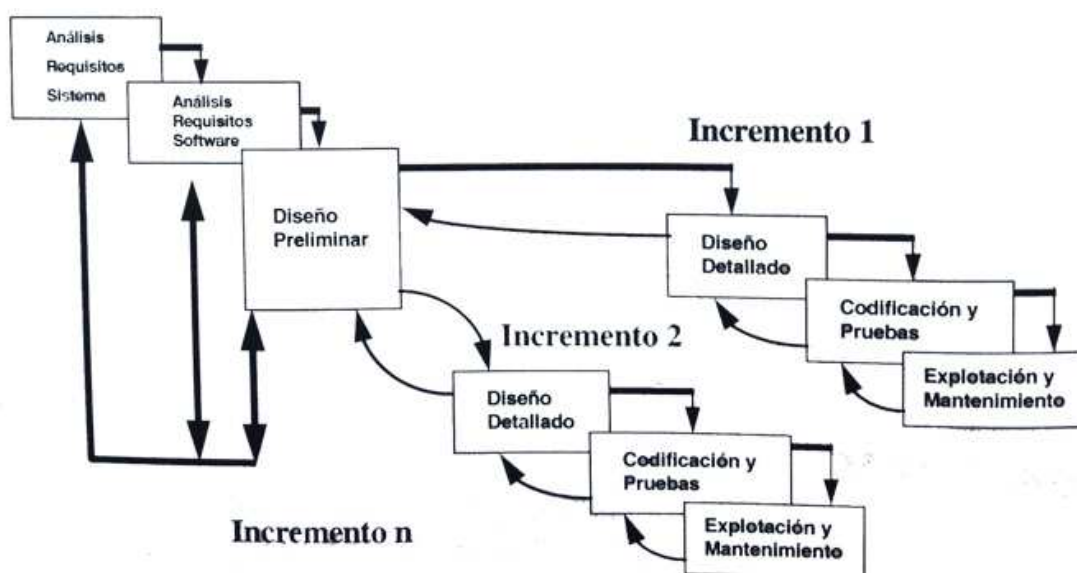


Figura 4. Modelo incremental [Piattini et al, 2003].

Se emplea esta metodología debido a la novedad de utilizar la arquitectura SOA. Primero se prueba comunicando el ESB con una base de datos, luego se añade un servicio, luego con dos servicios y luego con más servicios, con pruebas en cada paso.

Se construye, paso a paso cada uno de los servicios de los que se compone la herramienta hasta alcanzar, en el último incremento, al producto final en esta versión.

3. DESCRIPCIÓN INFORMÁTICA

3.1 ANÁLISIS DE REQUISITOS

En esta sección se describen los requisitos de Edu-Reviews, tanto funcionales como no funcionales. Estos requisitos se han obtenido a partir de la información facilitada por el cliente (en este caso, la tutora) cubriendo su necesidad de construir un software [Piattini et al, 2003].

3.1.1 REQUISITOS FUNCIONALES

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema. Los requisitos funcionales definen lo que el sistema “debe hacer” y “no debe hacer”. Todas estas características son esenciales, es decir, no sería aceptable si el sistema desarrollado no satisface alguna de ellas.

RF1 – Deben existir dos tipos de usuarios: estudiante y profesor. El estudiante se podrá registrar online, mientras que el profesor tendrá que pedir acceso al administrador del sistema.

RF2 – El usuario (profesor/estudiante) podrá autenticarse en el sistema con su nombre de usuario y su contraseña.

RF3 – El profesor podrá seleccionar, crear y eliminar lecciones.

RF4 – El profesor no podrá crear lecciones con el mismo nombre.

RF5 – El profesor podrá seleccionar, crear, modificar y eliminar preguntas y respuestas.

RF6 – El profesor no podrá crear preguntas iguales.

RF7 – El profesor podrá crear una pregunta cuya respuesta esté vacía.

RF8 – El profesor podrá crear preguntas con varias respuestas.

RF9 – El estudiante podrá seleccionar cada una de las lecciones insertadas por el profesor.

RF10 – El estudiante podrá recargar una lección una vez contestadas todas las preguntas correspondientes a dicha lección.

RF11 – El estudiante podrá seleccionar cualquiera de las preguntas pertenecientes a la lección seleccionada.

RF12 –El estudiante podrá comprobar si la respuesta que ha introducido es correcta.

3.1.2 REQUISITOS NO FUNCIONALES

Los requisitos que se especifican a continuación están relacionados con la usabilidad, la fiabilidad, el rendimiento, el tiempo de respuesta, la capacidad de almacenamiento y las características de la interfaz necesarias para que la herramienta sea aceptable. Describen atributos del sistema o atributos del ambiente del sistema. Son los siguientes:

RNF1 – Edu-Reviews debe ser usable.

RNF2–Edu-Reviews tendrá un interfaz amigable, sencilla e intuitiva.

3.1.3 MODELOS DE CASOS DE USO

A partir del conjunto de requisitos anteriores, se analiza el sistema utilizando los casos de uso. Esta es la forma más simple para modelar los requisitos del sistema desde la perspectiva del usuario.

Los diagramas de casos de uso representan las diferentes operaciones que son realizadas por el sistema y la manera de actuar con su entorno, más concretamente, los usuarios y otras aplicaciones que interactúan con él.

Identificación de actores

Se define como actor a cualquier agente externo que interactúe con la aplicación. Un actor será, por tanto, una entidad externa al sistema que inicia una interacción con el mismo.

En Edu-Reviews hay dos tipos de actores diferentes, cada uno de ellos posee distintos casos de uso:

- **Profesor:** Persona que va a interactuar con la herramienta creando/borrando y modificando las lecciones y las preguntas según niveles de dificultad.
- **Estudiante:** Persona que va a interactuar con la herramienta de forma que podrá repasar y aprender las lecciones establecidas por el profesor.

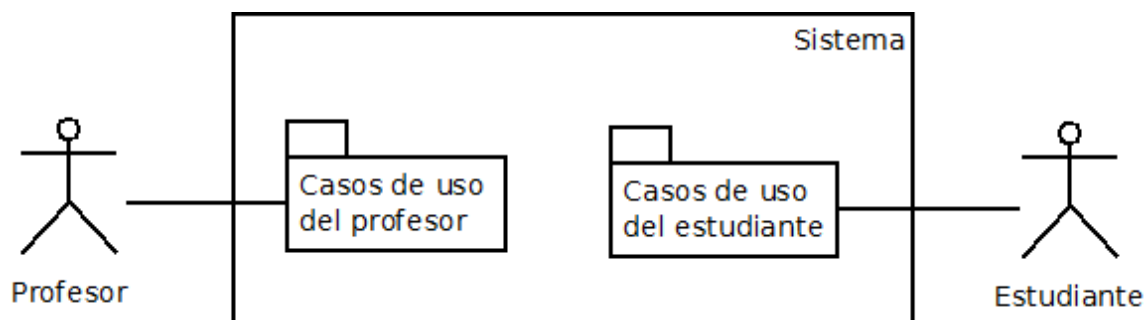


Figura 5. Casos de uso de la herramienta.

En la Figura 6, se observa el diagrama de los casos de uso del usuario profesor:

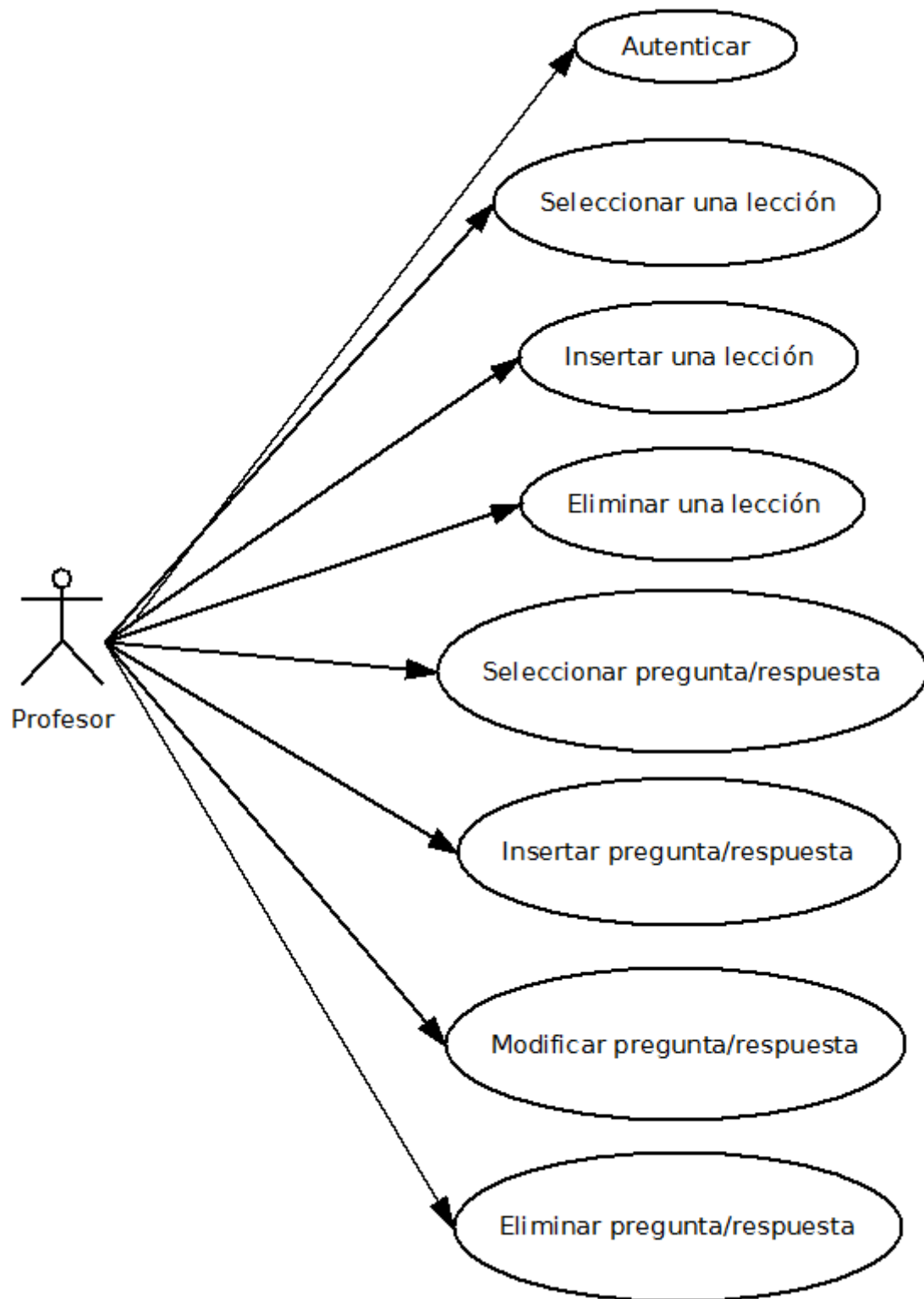


Figura 6. Casos de uso del profesor.

Autenticar	
Descripción	El caso de uso comienza cuando un profesor registrado presiona el botón “Log in”. El caso de uso verificará su identidad solicitando dicha información al servicio <login>. En caso de que los datos sean correctos se dará acceso a la herramienta.
Requisitos que contiene	RF2

Seleccionar lección	
Precondiciones	El profesor debe estar registrado y ha accedido previamente con sus datos correctamente a la herramienta. Se solicita al servicio <lessons> el listado de lecciones del menú desplegable.
Descripción	El caso de uso comienza cuando un profesor selecciona una de las lecciones disponibles. Se solicitará al servicio <questionAnswerList> el listado de todas las preguntas/respuestas asociadas a la lección seleccionada.
Requisitos que contiene	RF3

Insertar una lección	
Precondiciones	El profesor debe estar registrado y ha accedido previamente con sus datos correctamente a la herramienta. Se solicita al servicio <lessons> el listado de las lecciones.
Descripción	El caso de uso comienza cuando el profesor presiona el botón “NEW LESSON”. El caso de uso comprueba que el tema introducido no exista y que no esté vacío, en ese caso solicita al servicio <insertLesson> que lo añada.
Requisitos que contiene	RF3 y RF4

Eliminar una lección	
Precondiciones	El profesor debe estar registrado y ha accedido previamente con sus datos correctamente a la herramienta. Se solicita al servicio <lessons> el listado de las lecciones.
Descripción	El caso de uso comienza cuando el profesor presiona el botón “DELETE LESSON”. El caso de uso comprueba si se ha seleccionado una lección y en ese caso, solicita al servicio <deleteLesson> que lo borre.
Requisitos que contiene	RF3

Insertar pregunta/respuesta	
Precondiciones	El profesor ha seleccionado una lección previamente.
Descripción	El caso de uso comienza cuando el profesor presiona el botón “NEW QUESTION”, introduce la nueva pregunta/respuesta y presiona el botón “SAVE”. El caso de uso comprueba que el tema introducido no exista y que no esté vacío, en ese caso solicita al servicio <saveQuestion> que lo añada.
Requisitos que contiene	RF5

Modificar pregunta/respuesta	
Precondiciones	El profesor ha seleccionado una lección previamente.
Descripción	El caso de uso comienza cuando el profesor modifica una pregunta y/o respuesta existente y presiona el botón “SAVE”. El caso de uso solicita al servicio <saveQuestion> que modifique la pregunta/respuesta actual.
Requisitos que contiene	RF5

Eliminar pregunta/respuesta	
Precondiciones	El profesor ha seleccionado una lección previamente.
Descripción	El caso de uso comienza cuando el profesor presiona el botón "DELETE". El caso de uso solicita al servicio <deleteQuestion> que borre la pregunta/respuesta seleccionada.
Requisitos que contiene	RF5

En la Figura 7, se observa el diagrama de los casos de uso del usuario estudiante:

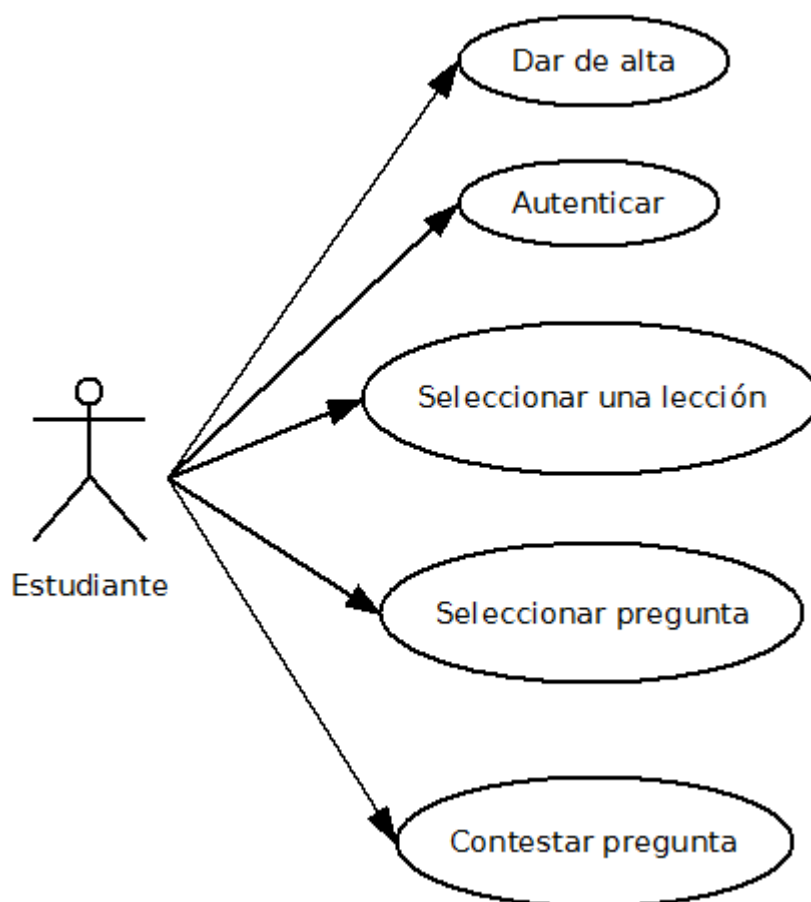


Figura 7. Casos de uso del estudiante.

Darse de Alta	
Descripción	El caso de uso comienza cuando un estudiante presiona el botón "Register". El estudiante utiliza el caso de uso para darse de alta en el sistema. El caso de uso comprobará que el usuario no existe solicitando dicha información al servicio <register> que se ocupa de ello.
Requisitos	RF1

Autenticar	
Descripción	El caso de uso comienza cuando un estudiante registrado presiona el botón “Log in”. El caso de uso verificará su identidad solicitando dicha información al servicio <login>. En caso de que los datos sean correctos se dará acceso a la herramienta.
Requisitos que contiene	RF2

Seleccionar lección	
Precondiciones	El estudiante debe estar registrado y ha accedido previamente con sus datos correctamente. Se solicita al servicio <lessons> el listado de las lecciones que se almacenarán en un menú desplegable.
Descripción	El caso de uso comienza cuando un estudiante selecciona una de las lecciones. Se solicitará al servicio <questionUserList> el listado de preguntas asociadas a la lección seleccionada que no han sido contestadas correctamente por el estudiante.
Requisitos que contiene	RF9

Seleccionar pregunta	
Precondiciones	El estudiante debe haber seleccionado una lección previamente.
Descripción	El caso de uso comienza cuando un estudiante selecciona una pregunta haciendo uso de los botones “<<” y “>>” para desplazarse hacia delante y hacia atrás en el listado circular de preguntas.
Requisitos que contiene	RF11

Contestar pregunta	
Precondiciones	El estudiante debe haber seleccionado una pregunta.
Descripción	El caso de uso comienza cuando un estudiante responde a la pregunta y presiona el botón “Check” que solicita al servicio <checkUserAnswer> si es correcta o no.
Requisitos que contiene	RF12

Recargar lecciones	
Precondiciones	El estudiante debe haber contestado correctamente todas las preguntas de una lección.
Descripción	El caso de uso comienza cuando un estudiante presiona el botón “reload” que solicita al servicio <reloadLesson> volver a cargar las preguntas de la lección.
Requisitos que contiene	RF10

3.2 DISEÑO

En este capítulo se aborda el diseño de la aplicación. Esto se realiza siguiendo un enfoque descendente desde la arquitectura de alto nivel, a un nivel más bajo.

3.2.1 ARQUITECTURA DE ALTO NIVEL

El proyecto Edu-Reviews se basa en una arquitectura orientada a servicios (SOA) y se implementa utilizando un bus (ESB).

La arquitectura distribuida del ESB, permite integrar de forma flexible servicios, aplicaciones, bases de datos, etc... dentro de SOA conectándolos independientemente y permitiendo su visibilidad con el resto de los recursos unidos al ESB.

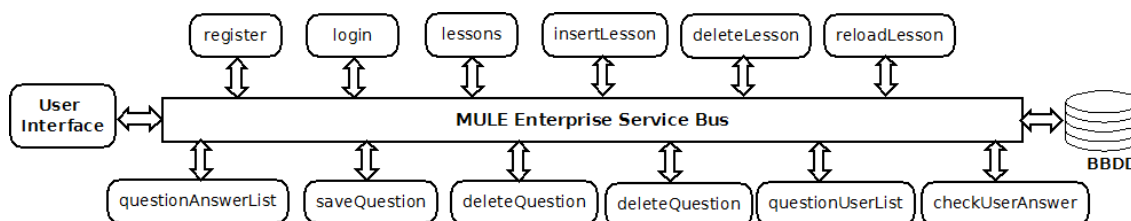


Figura 8. Esquema SOA de Edu-Reviews.

En la Figura 8 se representa el esquema SOA de la herramienta Edu-Reviews, se observa como Mule ESB mantiene comunicados todos los módulos, la base de datos y

la interfaz de usuario. Los módulos se explicarán en detalle en la sección 3.5 Descripción de los módulos.

3.3 DIAGRAMA DE ENTIDAD RELACIÓN

Se muestra el diagrama E/R que se crea según las necesidades identificadas en el análisis:

3.3.1 MODELO ENTIDAD-RELACIÓN

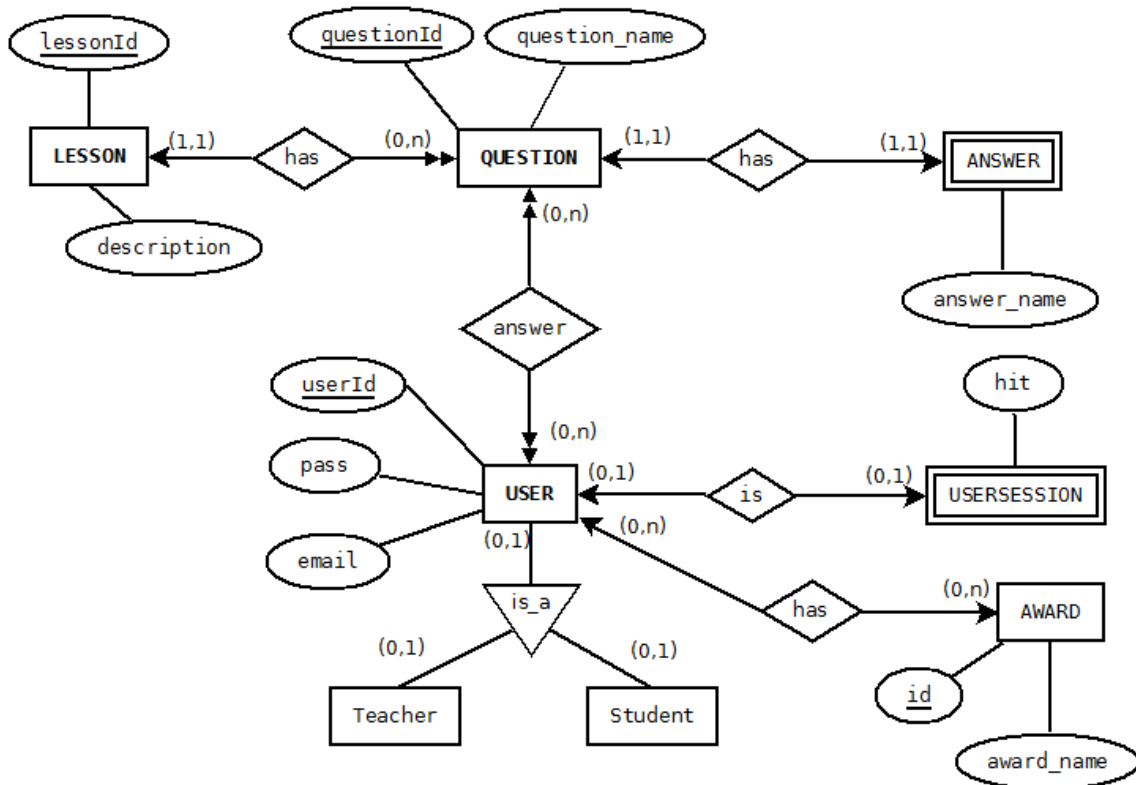


Figura 9. Diagrama entidad relación E/R

3.3.2 MODELO RELACIONAL

Para el modelo de entidad-relación descrito en la Figura 9, el paso a tablas es:

USER (userId, pass, email, isTeacher)

LESSON (lessonId, description)

QUESTION (questionId, question_name, lessonId)

ANSWER (answer_name, questionId)

USERSESSION (hit, userId)

AWARD (id, award_name)

3.3.3 DESCRIPCIÓN DE LAS TABLAS.

USER: tabla donde se almacenan todos los usuarios (estudiantes y profesores) que inician sesión en la aplicación. Consta de los siguientes datos (userId, pass, email, isTeacher), siendo el userId clave primaria. Los campos que la componen son:

Field	Type	Null	Key	Default	Extra
userId	varchar(15)	NO	PRI	NULL	
pass	varchar(15)	NO		NULL	
email	varchar(25)	NO		NULL	
isTeacher	smallint(1)	NO		NULL	

Tabla 8. Tabla sql USER.

Ejemplo tabla user:

userId	pass	isTeacher	email
rouri88	Rodrigo	0	rodri_algora@hotmail.com
theo	Hola	1	theo@theo.es
juan	juan	0	zjuan@gmail.com

Tabla 9. Ejemplo tabla USER.

LESSON: tabla donde se almacenan todas las lecciones introducidas por el profesor. Contiene una clave primaria para identificar cada una de las lecciones y la descripción de cada lección. Los campos que la componen son:

Field	Type	Null	Key	Default	Extra
lessonId	int(5)	NO	PRI	NULL	auto_increment
description	varchar(80)	YES		NULL	
email		NO		NULL	
isTeacher		NO		NULL	

Tabla 10. Tabla sql LESSON.

Ejemplo tabla lesson:

lessonId	description
2	Noun/adjective + preposition
3	Articles a/an and the
4	Phrasal verbs
8	Conditionals
9	Adverbs

Tabla 11. Ejemplo tabla LESSON.

QUESTION: tabla donde se almacenan todas las preguntas. Consta de un identificador para cada pregunta questionId, question_name que contiene la pregunta, lessonId que contiene la lección correspondiente y una columna por cada uno de los estudiantes que existan en la tabla user (por defecto se inicializa a NULL).

Field	Type	Null	Key	Default	Extra
questionId	int(11)	NO	PRI	NULL	auto_increment
question_name	varchar(200)	NO		NULL	
lessonId	int(5)	NO	MUL	NULL	
user1	tinyint(1)	YES		NULL	

Tabla 12. Tabla sql QUESTION.

Ejemplo tabla question:

questionId	question_name	lessonId	rouri88	juan
22	Where is USB drive I lent you last week?	3	1	NULL
37	Does your little brother ghosts?	4	NULL	1

Tabla 13. Ejemplo tabla QUESTION.

ANSWER: tabla en la que se almacenan todas las respuestas de las preguntas de la tabla QUESTION.

Field	Type	Null	Key	Default	Extra
answer_name	varchar(120)	NO	PRI	NULL	
questionId	int(11)	YES	MUL	NULL	

Tabla 14. Tabla sql ANSWER.

Ejemplo tabla answer:

answer_name	questionId
the	22
believe in	37

Tabla 15. Ejemplo tabla ANSWER.

USERSESSION: tabla en la que se almacena el usuario actual. Si se trata de un estudiante, se anotan los aciertos que realice durante la sesión.

Field	Type	Null	Key	Default	Extra
userId	varchar(15)	NO		NULL	
hit	int(3)	YES		NULL	

Tabla 16. Tabla sql USERSESSION.

Ejemplo table userSession;

userId	hit
juan	7

Tabla 17. Ejemplo tabla USERSESSION.

AWARD: tabla en la que almacena cada uno de los premios que puede recibir el usuario actual dependiendo del número de aciertos que consiga por sesión.

Field	Type	Null	Key	Default	Extra
id	int(3)	NO	PRI	NULL	auto_increment
award_name	varchar(80)	NO		NULL	

Tabla 18. Tabla sql AWARD.

Ejemplo tabla award:

id	award_name
1	A BICHO QUE NO CONOZCAS, NO LE PISES LA COLA.
3	A BUEN AMIGO BUEN ABRIGO.

Tabla 19. Ejemplo tabla AWARD.

3.4 ALGORITMO

3.4.1 PROFESOR

Acceso profesor

- Rellenar los campos de User y Password y pulsar el botón Login.
- En caso de que los datos introducidos no sean correctos ->
 - o Se muestra un mensaje al profesor de que alguno de los datos introducidos es incorrecto.
- En caso contrario ->
 - o Se envían los datos a través de Mule ESB al servicio login que recibe los datos insertados y comprueba si el usuario existe o no:
 - En caso de que el usuario no exista:
 - Se muestra un mensaje que indica al profesor de que el usuario no existe.
 - En caso contrario:

- Se solicita al servicio lessons a través Mule ESB que devuelva la lista de lecciones existentes para que se muestren en la aplicación.

Crear lección

- Se pulsa el botón NEW LESSON.
- Se muestra una ventana donde se pide insertar el nombre de la nueva lección
 - Si el nombre de la lección está vacía ->
 - Se muestra un mensaje que indica que no puede estar vacío.
 - Si el nombre de la lección está repetido ->
 - Se muestra un mensaje al profesor que ya existe una lección con ese nombre.
 - En caso contrario:
 - Se realiza la petición mediante Mule ESB al servicio insertLesson para insertar la nueva lección.
 - El servicio recibe el nombre de la lección nueva y se ocupa de insertar la lección nueva en la base de datos Edureviews.
 - Se añade a la lista de lecciones del desplegable la nueva lección.

Borrar lección

- Se selecciona del desplegable la lección a borrar.
- Se pulsa el botón DELETE LESSON.
 - En caso de que no se seleccione ninguna lección->
 - Se muestra un mensaje que indica que no se ha seleccionado ninguna lección.
 - En caso contrario, se muestra un mensaje que solicita la confirmación de borrado de dicha lección.
 - Botón Aceptar:
 - Se realiza la petición mediante Mule ESB al servicio deleteLesson.

- El servicio recibe el identificador de la lección se ocupa de borrarla de la base de datos Edureviews.
- Botón Cancelar: se cancela dicha operación.

Seleccionar lección

- Se selecciona una lección de la lista del menú desplegable de lecciones.
- Se solicita mediante Mule ESB al servicio questionAnswerList, el conjunto de las preguntas/respuestas asociadas a la lección seleccionada.
 - El servicio recibe el identificador de la lección seleccionada y se ocupa de solicitar a la base de datos Edureviews el conjunto de preguntas y respuestas.
- Se muestra al profesor las preguntas/respuestas asociadas a la lección.

Seleccionar preguntas

Las preguntas y respuestas de cada una de las lecciones son almacenadas en una lista circular.

- Se pulsa el botón >> para mostrarla siguiente pregunta/respuesta.
- Se pulsa el botón << para mostrar la pregunta/respuesta anterior.

Crear pregunta/respuesta

- Se pulsa el botón NEW LESSON y los campos de texto de pregunta/respuesta se inicializan a vacío.
- Se escribe la nueva pregunta en el campo pregunta y la respuesta a dicha pregunta en el campo respuesta.
- Se pulsa el botón SAVE para guardar la pregunta y la respuesta.
 - En caso de que exista la misma pregunta ->
 - Se solicita mediante Mule ESB al servicio saveQuestion que almacene la pregunta y respuesta en la base de datos Edureviews.
 - Se muestra al usuario la pregunta/respuesta una vez almacena.

Modificar pregunta/respuesta

- Se selecciona alguna de las preguntas/respuestas con los botones << y/o>>.
- Se modifica el campo pregunta y/o respuesta.
- Se pulsa el botón SAVE para guardar la pregunta/respuesta.
 - o Se solicita mediante Mule ESB el servicio saveQuestion que recibe la pregunta y respuesta modificadas y las almacena en la base de datos Edureviews.

Borrar pregunta/respuesta

- Se selecciona alguna de las preguntas/respuestas existentes con los botones << y/o>>.
- Se pulsa el botón DELETE para borrar la pregunta/respuesta.
 - o En caso de que no se haya seleccionado ninguna pregunta/respuesta
 - Se muestra un mensaje que indica que no se ha seleccionado ninguna pregunta/respuesta.
 - o En caso contrario, se muestra un mensaje de confirmación de borrado de la pregunta/respuesta.
 - Botón Aceptar: borrar la pregunta/respuesta.
 - Se solicita mediante Mule ESB al servicio deleteQuestion, que recibe el identificador de la pregunta a borrar, eliminar la pregunta y la respuesta correspondiente de la base de datos Edureviews.
 - Botón Cancelar: se cancela dicha operación.

3.4.2 ESTUDIANTE

Acceso estudiante

- Rellenar los campos de User y Password y pulsar el botón LOG IN.
- En caso de que los datos introducidos no sean correctos ->
 - o Se muestra un mensaje al estudiante de que alguno de los datos introducidos es incorrecto.
- En caso contrario ->

- Se envían los datos a través de Mule ESB al servicio login que recibe los datos insertados y comprueba si el usuario existe o no:
 - En caso de que el usuario no exista:
 - Se muestra un mensaje que indica al profesor de que el usuario no existe.
 - En caso contrario:
 - Se solicita al servicio lessons a través Mule ESB que devuelva la lista de lecciones existentes para que se muestren en la aplicación.

Registrar nuevo estudiante

- Se pulsa el botón Register.
- Se abre una ventana formulario en la que se pide al estudiante que introduzca sus datos:
 - En caso de que los datos introducidos sean incorrectos ->
 - Se muestra un mensaje que indica que alguno de los datos introducidos no es correcto.
 - En caso contrario ->
 - Se solicita mediante Mule ESB al servicio register, que recibe los datos de registro, almacenarlos en la tabla user de la base de datos Edureviews.
 - Se muestra un mensaje que indica al estudiante que se ha realizado el registro con éxito y que ya puede loguearse.

Seleccionar lección

- Se selecciona una de las lecciones del menú desplegable de lecciones.
- Se solicita mediante Mule ESB al servicio questionUserList el conjunto de las preguntas asociadas a la lección seleccionada que no han sido contestadas correctamente por el estudiante.
 - El servicio recibe el identificador de la lección seleccionada y devuelve las preguntas de la base de datos Edureviews.
- Se muestra al estudiante las preguntas asociadas a la lección.

Seleccionar pregunta/respuesta

Las preguntas y respuestas de cada una de las lecciones son almacenadas en una lista circular.

- Se pulsa el botón >> para mostrar la siguiente pregunta.
- Se pulsa el botón << para mostrar la pregunta anterior.

Comprobar respuesta

- Se introduce la respuesta de la pregunta en el campo respuesta.
- Se pulsa el botón Check:
 - o Mediante Mule ESB el servicio checkUserAnswer recibe la respuesta:
 - En caso de que la respuesta sea correcta:
 - En caso de alcanzar un determinado número de aciertos ->
 - o Se recompensa al estudiante con un premio, en este caso un refrán.
 - En caso contrario ->
 - o Se muestra un mensaje que anima al estudiante a seguir respondiendo.
 - En caso de que la respuesta sea incorrecta:
 - Se muestra un mensaje que indica al estudiante que la respuesta es incorrecta y se le incluye la respuesta correcta.

Recargar lección

- Una vez contestadas correctamente todas las preguntas por el estudiante correspondientes a una lección, aparece el botón reload.
 - o Pulsar botón reload
 - Mediante Mule ESB se llama al servicio reloadLesson que recibe el identificador de la lección correspondiente y se actualiza para que todas las preguntas de esa lección vuelvan a estar disponibles para el estudiante actual (aparezcan como no contestadas).

3.5 DESCRIPCIÓN DE LOS MÓDULOS

En esta sección, se explican los principales módulos de los que se compone la aplicación Edu-Reviews. En primer lugar, se explican los módulos del profesor, luego los del estudiante y finalmente los comunes a ambos.

Estos módulos se construyen con la aplicación Mule Studio que es la interfaz gráfica del Mule ESB (ver Figura 10). Cada uno de ellos corresponde con un flujo formado por una serie de elementos: transformadores, endpoints, componentes, filtros, flujos de control...por el que discurre el mensaje Mule. El contenido de este mensaje se podrá modificar empleando elementos como transformadores, filtrar empleando filtros o bien controlar su entrada con los flujos de control.

La información referente a cada uno de los elementos que forman cada módulo se puede encontrar en la referencia [HTTP 24].

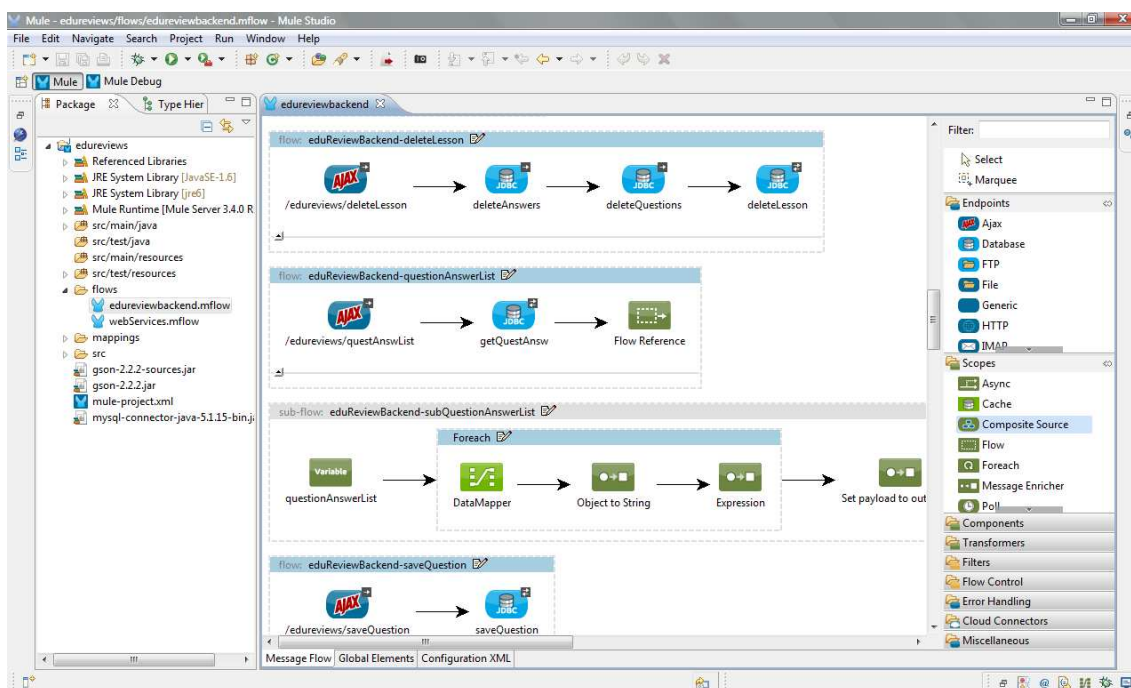


Figura 10. Mule Studio ESB

3.5.1 DESCRIPCIÓN DE LOS MÓDULOS DEL PROFESOR

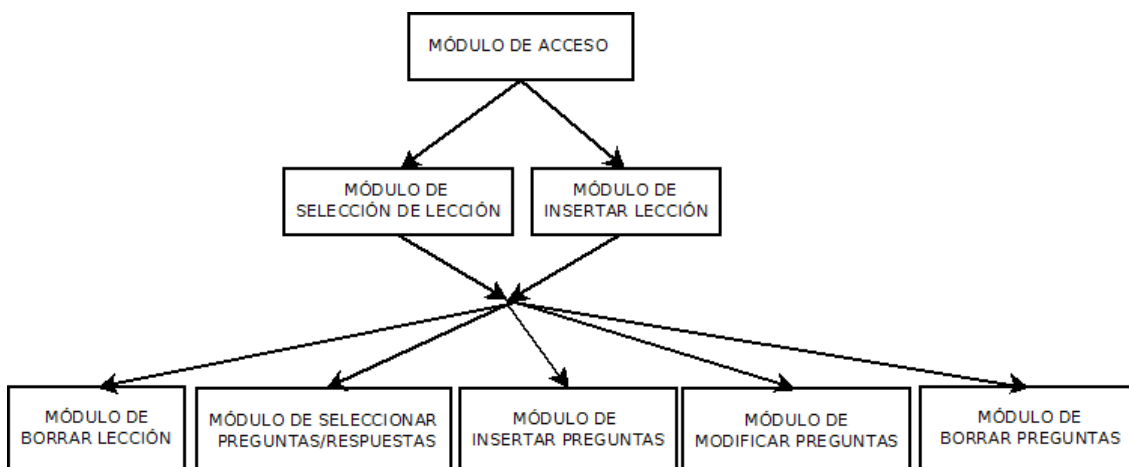


Figura 11. Módulos Profesor.

Módulo de inserción de lección (sólo profesor)

En este módulo, el profesor puede añadir mediante un formulario el nombre de una nueva lección y pulsando el botón insertar, almacenar dicha lección en la base de datos Edureviews.

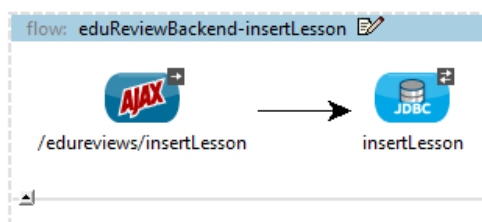


Figura 12. Módulo inserción lección.

En la Figura 12 se puede observar el icono AJAX (representa el nodo que recibe la petición de insertar lección: /edureviews/insertLesson desde la interfaz de usuario). Este nodo recibe un mensaje en formato JSON que contiene el nombre de la lección a insertar. Este mensaje se envía al icono JDBC (que representa la conexión con la base de datos del proyecto Edureviews) donde se realiza un insert en la tabla lessons.

Módulo de borrado de lección (sólo profesor)

En este módulo se codifica la posibilidad de que el profesor seleccione una de las lecciones que quiere borrar. El borrado de la lección incluye también las preguntas y respuestas correspondientes a esa lección de la base de datos Edureviews.

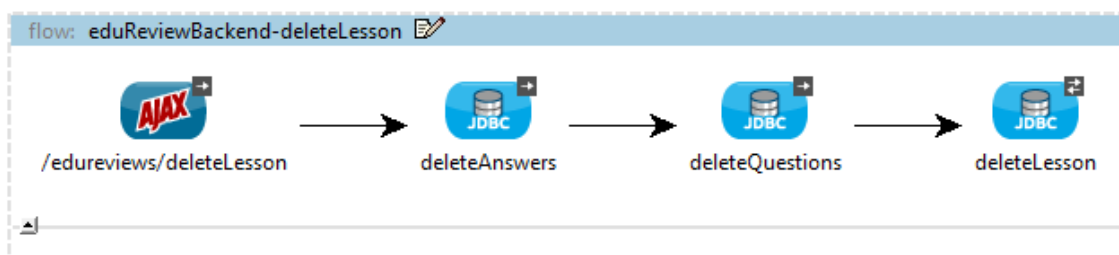


Figura 13. Módulo de borrado de lección.

En la Figura 13 se observa el icono AJAX (representa el nodo que recibe la petición de borrar lección: /edureviews/deleteLesson desde la interfaz de usuario). Este nodo recibe un mensaje en formato JSON que contiene el identificador de la lección a borrar. Este mensaje se envía por orden a tres iconos JDBC. En cada uno de ellos se realiza un delete (primero en la tabla answer, después en la tabla question y por último en la tabla lesson).

Módulo de selección de preguntas/respuestas (sólo profesor)

En este módulo se muestra al profesor el conjunto de todas las preguntas y respuestas asociadas a la lección seleccionada.

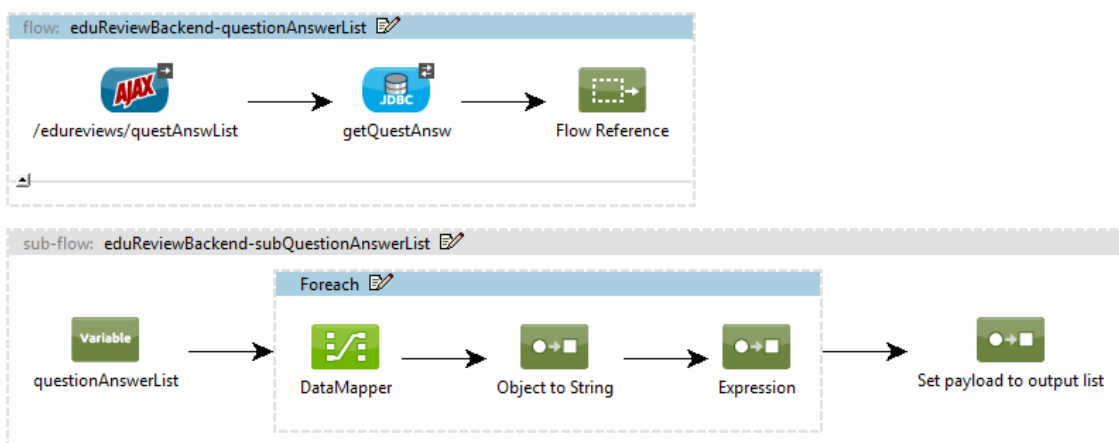


Figura 14. Módulo de selección de preguntas/respuestas.

En la Figura 14 se observa el icono AJAX (representa el nodo que recibe la petición del listado completo de preguntas y respuestas de la lección actual: /edureviews/questAnswList desde la interfaz de usuario). Este nodo recibe un mensaje

en formato JSON con el identificador de la lección actual. El icono JDBC se encarga de obtener dicho listado de base de datos (mediante un select), cuyo contenido se envía a un subflujo. En este subflujo (representado en la parte inferior de la figura como sub-flow), se declara la variable `questionAnswerList` y por cada línea del listado de base de datos (foreach) se transforma (con ayuda del transformador `DataMapper`) en un listado de objetos de tipo `String` (almacenado en la variable `questionAnswerList`) que devuelve al usuario.

Módulo de inserción y modificación de preguntas/respuestas (sólo profesor)

En este módulo, el profesor puede ver todas las preguntas/respuestas existentes en la base de datos según la lección solicitada y puede insertar nuevas preguntas con sus respuestas correspondientes pulsando el botón `NEW QUESTION`, escribiendo la pregunta y la respuesta y posteriormente pulsando sobre el botón `SAVE`.

También puede modificar una pregunta existente y guardarla con el botón `SAVE`.

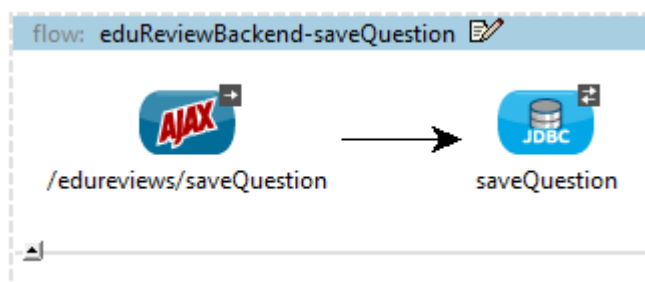


Figura 15. Módulo de inserción y modificación de preguntas/respuestas.

En la Figura 15 se observa el icono AJAX (representa el nodo que recibe la petición de guardar pregunta: `/edureviews/saveQuestion` desde la interfaz de usuario). Este nodo recibe un mensaje en formato JSON que contiene la pregunta, la respuesta y el identificador de la lección a la que pertenecen. Este mensaje se envía al icono JDBC (que representa la conexión con la base de datos del proyecto `Edureviews`) donde se realiza un insert en las tablas `question` y `answer`.

Módulo de borrado de preguntas/respuestas (sólo profesor)

En este módulo, el profesor puede ver todas las preguntas/respuestas existentes en la base de datos según la lección solicitada y puede borrar la pregunta/respuesta actual pulsando sobre el botón DELETE. Se pedirá confirmación de dicha operación.

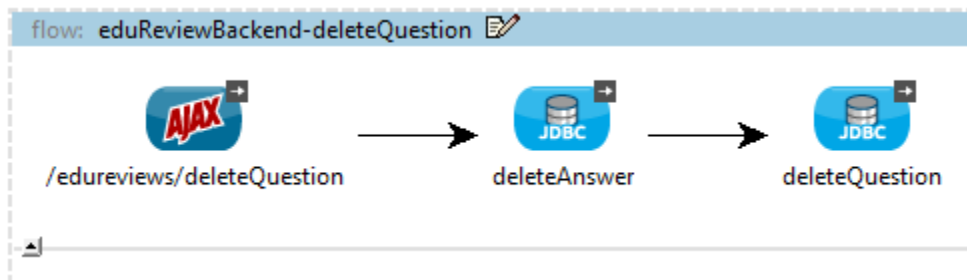


Figura 16. Módulo de borrado de preguntas/respuestas.

En la Figura 16 se observa el icono AJAX (representa el nodo que recibe la petición de borrar pregunta/respuesta: /edureviews/deleteQuestion desde la interfaz de usuario). Este nodo recibe un mensaje en formato JSON que contiene el identificador de la pregunta a borrar. Este mensaje se envía por orden a dos iconos JDBC. En cada uno de ellos se realiza, un delete (primero sobre la tabla answer y después sobre la tabla question, correspondientemente).

3.5.2 DESCRIPCIÓN DE LOS MÓDULOS DEL ESTUDIANTE

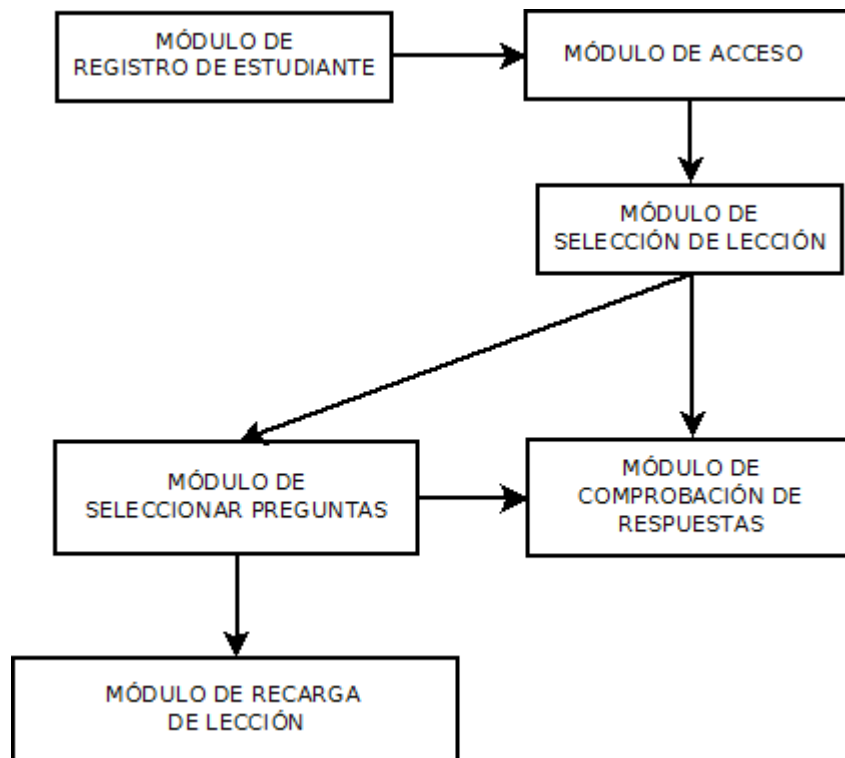


Figura 17. Módulos Estudiante.

Módulo de registro (sólo estudiante)

En este módulo el estudiante introduce sus datos en un formulario y se almacenan en la base de datos Edureviews. En caso de que el nombre de usuario ya exista en la base de datos se informa al usuario. En caso contrario, se almacena el usuario en base de datos Edureviews y se inicializa el listado de las preguntas respondidas del usuario.

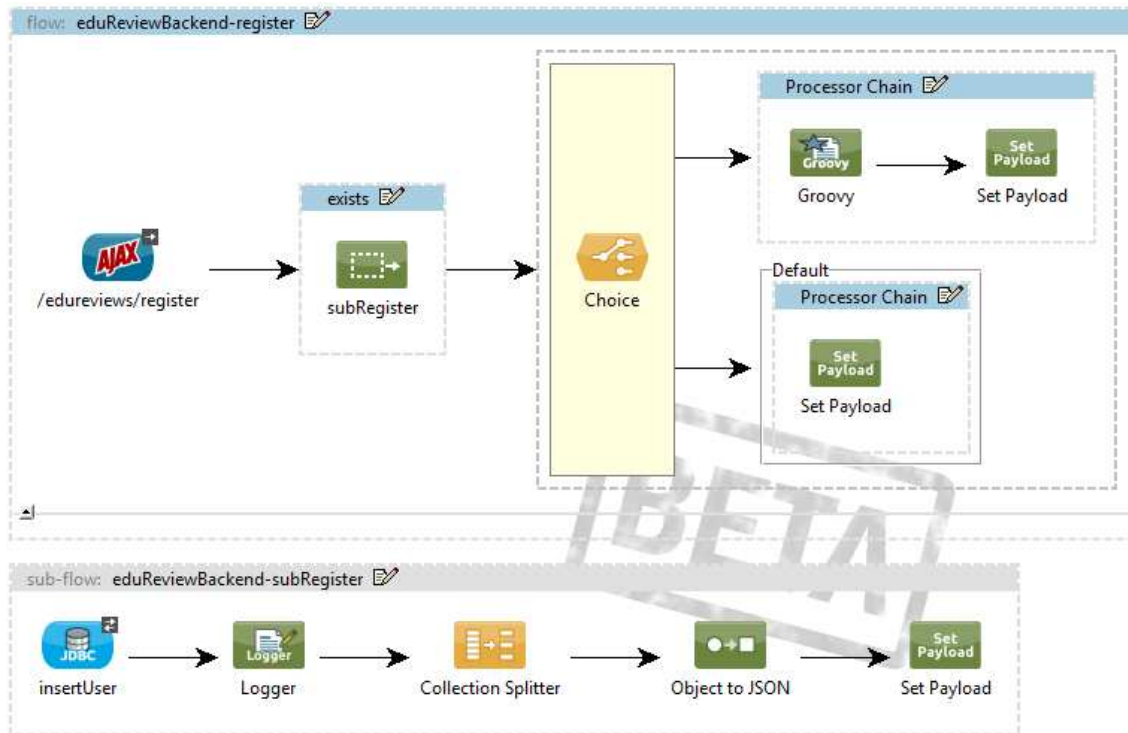


Figura 18. Módulo de registro.

En la Figura 18 se observa el icono AJAX (representa el nodo que recibe la petición de registro de estudiante: /edureviews/register desde la interfaz de usuario). Este nodo recibe un mensaje en formato JSON que contiene todos los datos solicitados al usuario para efectuar el registro. En primer lugar se comprueba si el usuario existe (para ello se consulta en base de datos si el nombre de usuario es el mismo). En caso de que no exista se realiza una inserción en base de datos en la tabla user y empleando groovy se realiza un Alter en la tabla question para añadir la columna correspondiente al usuario. Y por último, se muestra un mensaje en la interfaz del usuario indicándole que el registro ha sido correcto.

En caso de que el usuario exista, se muestra un mensaje en la interfaz del usuario avisándole de que el nombre de usuario ya existe.

Módulo de selección de preguntas (sólo estudiante)

En este módulo, el estudiante podrá ver todas las preguntas existentes en la base de datos según la lección solicitada. Sólo podrá ver aquellas preguntas que no hayan sido contestadas correctamente.

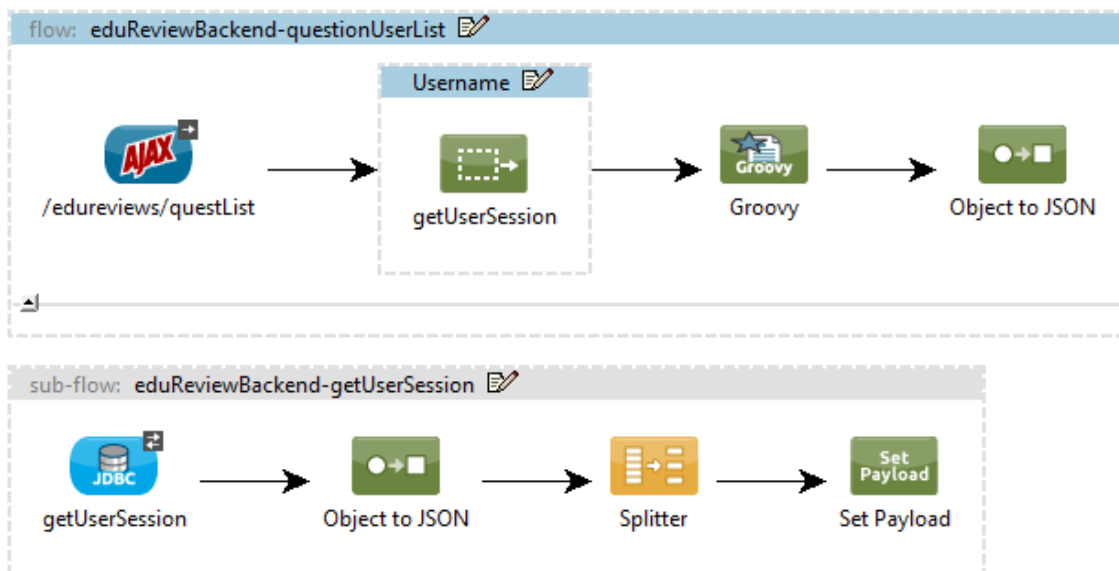


Figura 19. Módulo de selección de preguntas.

En la Figura 19 se observa el icono AJAX (representa el nodo que recibe la petición del listado de preguntas del estudiante de la lección actual: /edureviews/questList desde la interfaz de usuario). Este nodo recibe un mensaje en formato JSON que contiene el identificador de la lección actual. Desde el icono getUserSession se solicita a la tabla UserSession de la base de datos el nombre de usuario actual. Con el identificador del usuario y de la lección, desde el icono Groovy, se hace un select de la tabla question y se obtiene el listado de las preguntas que no han sido contestadas por el estudiante. Dicho listado se devuelve en formato JSON a la interfaz de usuario.

Módulo de comprobación de la respuesta introducida (sólo estudiante)

En este módulo, el estudiante, tras haber escrito la respuesta de la pregunta pulsa el botón CHECK que se encarga de comparar la respuesta introducida por el estudiante con la respuesta correcta. En caso de que la respuesta sea correcta se informa al estudiante y se le anima a seguir repasando o, si ha alcanzado el número de aciertos suficientes, recibirá un premio. En caso de que la respuesta sea incorrecta, se informa al estudiante de la solución correcta.

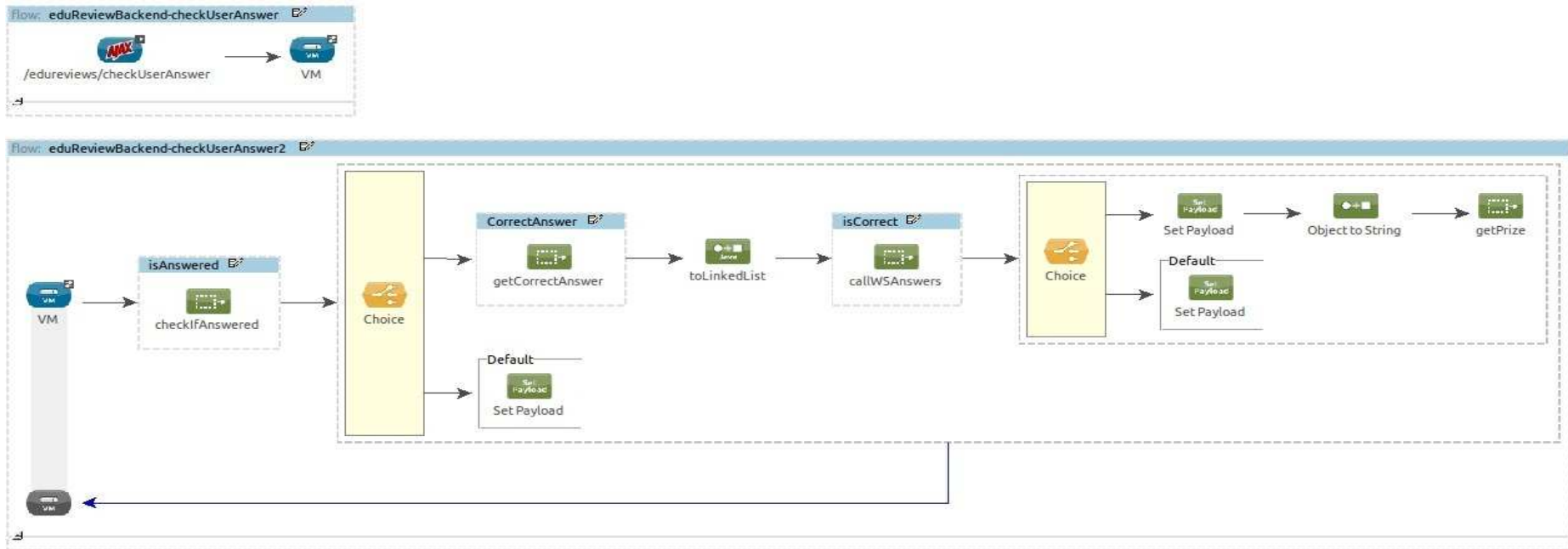


Figura 20. Módulo de comprobación de la respuesta introducida.

En la Figura 20 se observa el icono AJAX (representa el nodo que recibe la petición de comprobar la respuesta: /edureviews/checkUserAnswer desde la interfaz de usuario). Este nodo recibe un mensaje en formato JSON que contiene la respuesta dada por el estudiante y el identificador de la pregunta.

En el icono checkifAnswered, se comprueba si la pregunta ha sido contestada previamente. Si ha sido contestada se informa al usuario mediante un mensaje. Si no ha sido contestada, se solicita a la base de datos la respuesta correcta. Ambas respuestas, la correcta y la dada por el usuario, se envían al servicio web Answers donde se chequea si son iguales. En caso de que la respuesta sea correcta se llama al módulo de recompensa. En caso contrario, se indica al usuario de que la respuesta no es correcta.

Módulo de recompensa (sólo estudiante)

En este módulo, tras haber contestado correctamente durante la sesión un número de respuestas correctas el estudiante recibirá un premio, en este caso un dicho o refrán para animarle a seguir repasando.

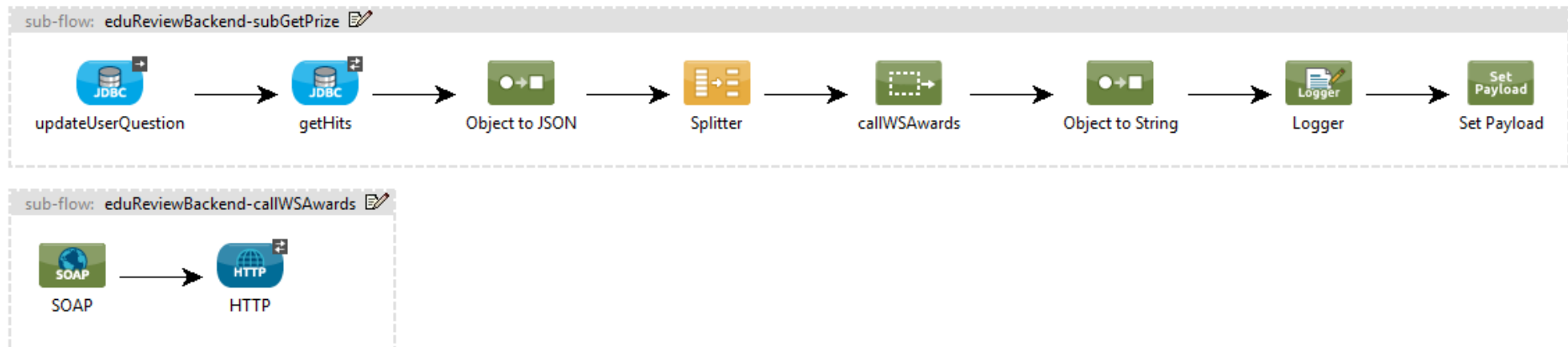


Figura 21. Módulo de recompensa.

En el icono JDBC updateUserQuestion, se anota la pregunta como contestada correctamente en la tabla question, en la columna del estudiante actual. En el icono JDBC getHits se hace un select de la tabla userSession para obtener el número de aciertos que lleva el estudiante y con este resultado se envía al servicio web Awards que se ocupa de identificar si existe o no premio para el estudiante según el número de aciertos.

Módulo de recarga de las preguntas de una lección (sólo estudiante)

En este módulo, tras haber contestado correctamente todas las preguntas de una lección, el estudiante puede volver a cargarlas para repasar de nuevo.

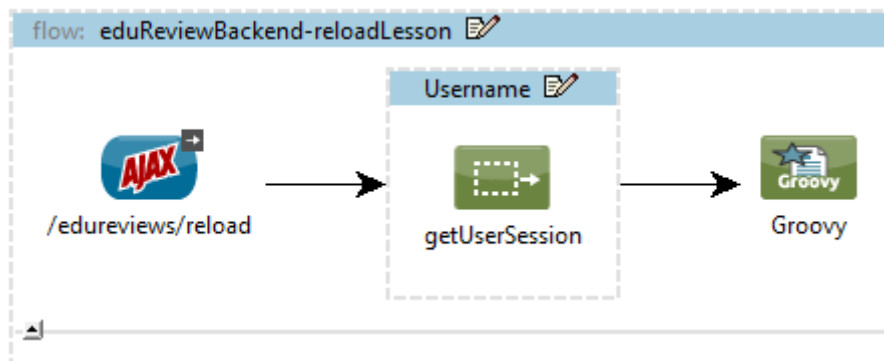


Figura 22. Módulo de recarga de las preguntas de una lección.

En la Figura 22 se observa el icono AJAX (representa el nodo que recibe la petición de recargar el listado de preguntas del estudiante de la lección actual: /edureviews/reload desde la interfaz de usuario). Este nodo recibe un mensaje en formato JSON que contiene el identificador de la lección actual. Desde el icono getUserSession se solicita a la tabla UserSession de la base de datos el nombre de usuario actual. Con el identificador del usuario y de la lección, desde el icono Groovy, se hace un update en la tabla question de la base de datos Edureviews inicializando todas las preguntas de la lección como no contestadas.

3.5.3 DESCRIPCIÓN DE LOS MÓDULOS COMPARTIDOS ENTRE EL PROFESOR-ESTUDIANTE:

Módulo de acceso profesor/estudiante

En este módulo el profesor/estudiante introduce sus datos de identificación (nombre de usuario y contraseña) para validar sus datos con los que se encuentran en la base de datos Edureviews.

En caso de que los datos del usuario sean incorrectos, se avisa al usuario. En caso contrario se inicializan los datos de la sesión (nombre de usuario y número de preguntas correctas contestadas en la sesión).

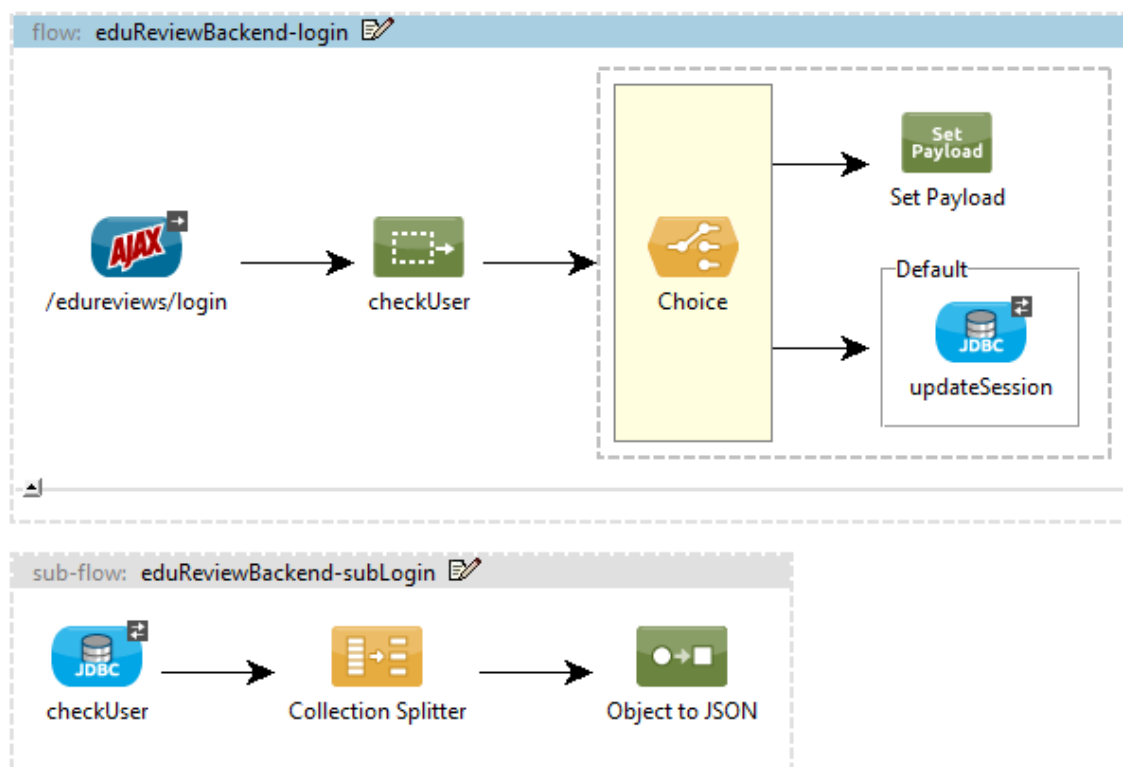


Figura 23. Módulo de acceso profesor/estudiante.

En la Figura 23 se observa el icono AJAX (representa el nodo que recibe la petición del loguearse en la herramienta Edu-Reviews: /edureviews/login desde la interfaz de usuario). Este nodo recibe un mensaje en formato JSON que contiene el nombre de usuario y la contraseña del usuario. En el icono checkUser se solicita a la tabla user de la base de datos, si existe el nombre de usuario. En caso de que no exista se avisa al usuario mediante un mensaje. En caso contrario, se actualiza el nombre de usuario actual en la tabla userSession.

Módulo de selección de lección

En este módulo se muestra al profesor/estudiante las lecciones existentes, a partir de las que están almacenadas en la base de datos Edureviews.

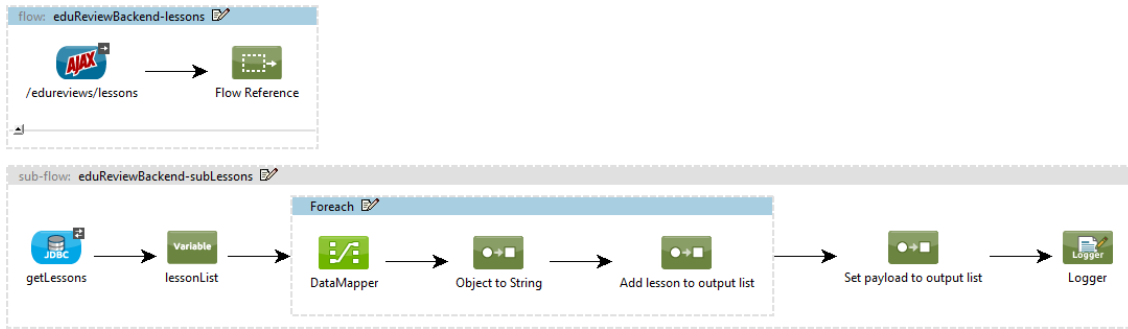


Figura 24. Módulo de selección de lección.

En la Figura 24 se observa el icono AJAX (representa el nodo que recibe la petición del listado lecciones: /edureviews/lessons desde la interfaz de usuario). Se llama al sub-flujo subLessons (representado en la parte inferior de la figura). Desde este sub-flujo, en el icono JDBC se hace un select del listado de las lecciones de base de datos que, tras sufrir un conjunto de transformaciones, se convierte a formato String. Dicho listado se devuelve a la interfaz de usuario.

4. PRUEBAS

Se implementa la siguiente estrategia de pruebas:

- Pruebas unitarias.
- Pruebas de integración.
- Pruebas de validación.
- Pruebas de aceptación.

La estructura de las pruebas realizadas sigue un enfoque ascendente desde las pruebas unitarias de cada módulo del sistema hasta valorar si la aplicación cumple con los requisitos no funcionales, tales como que sea manejable y fácil de usar, hasta la comprobación del nivel de satisfacción alcanzado por los usuarios.

4.1 PRUEBAS UNITARIAS

Con este tipo de pruebas se comprueba el diseño y el comportamiento de cada uno de los módulos del sistema una vez construidos. Se distingue entre pruebas de Caja Blanca y Caja Negra:

- El enfoque estructural o de caja blanca (código interno), verifica que todas las bifurcaciones y posibilidades son correctas en un módulo.

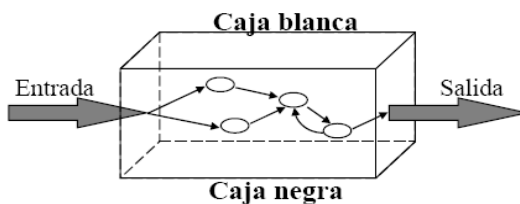


Figura 25. Diagrama de caja blanca [Piattini et al, 2003].

- El enfoque funcional o de caja negra (requisitos funcionales) comprueba que la salida esperada coincide con la salida resultado de cada módulo.



Figura 26. Diagrama de caja negra [Piattini et al, 2003].

4.1.1 PRUEBAS DE CAJA BLANCA

Algoritmo profesor

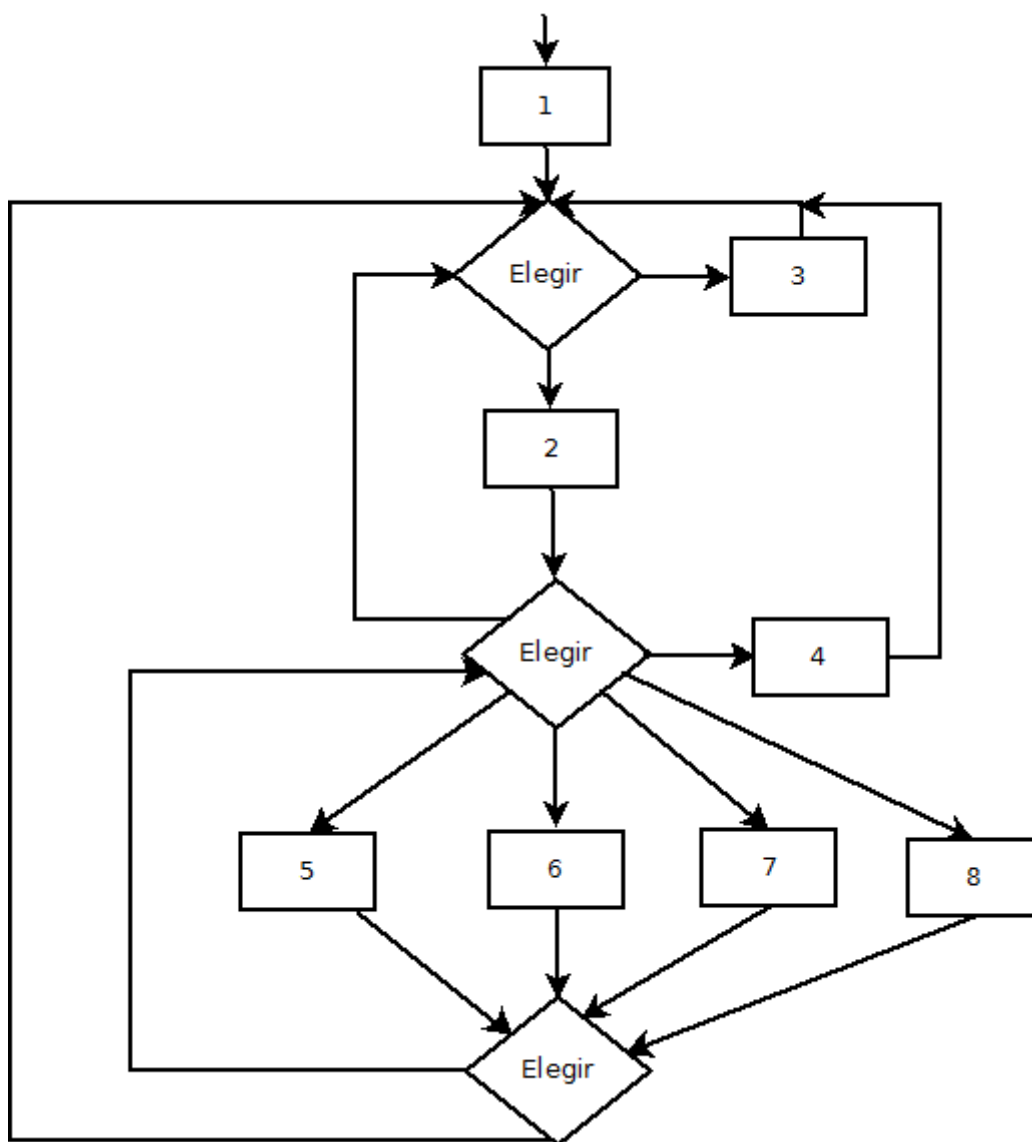


Figura 27. Prueba de caja blanca: algoritmo profesor.

Bloque 1

Acceder con los datos del profesor.

Bloque 2

Seleccionar una lección (menú desplegable).

Bloque 3

Crear una nueva lección (botón NEW LESSON).

Bloque 4

Borrar la lección seleccionada (botón DELETE LESSON).

Bloque 5

Seleccionar pregunta/respuesta (botones << y >>).

Bloque 6

Crear una nueva pregunta/respuesta (botón NEW QUESTION)

Bloque 7

Guardar la pregunta/respuesta (botón SAVE)

Bloque 8

Borrar pregunta/respuesta (botón DELETE)

Algoritmo estudiante

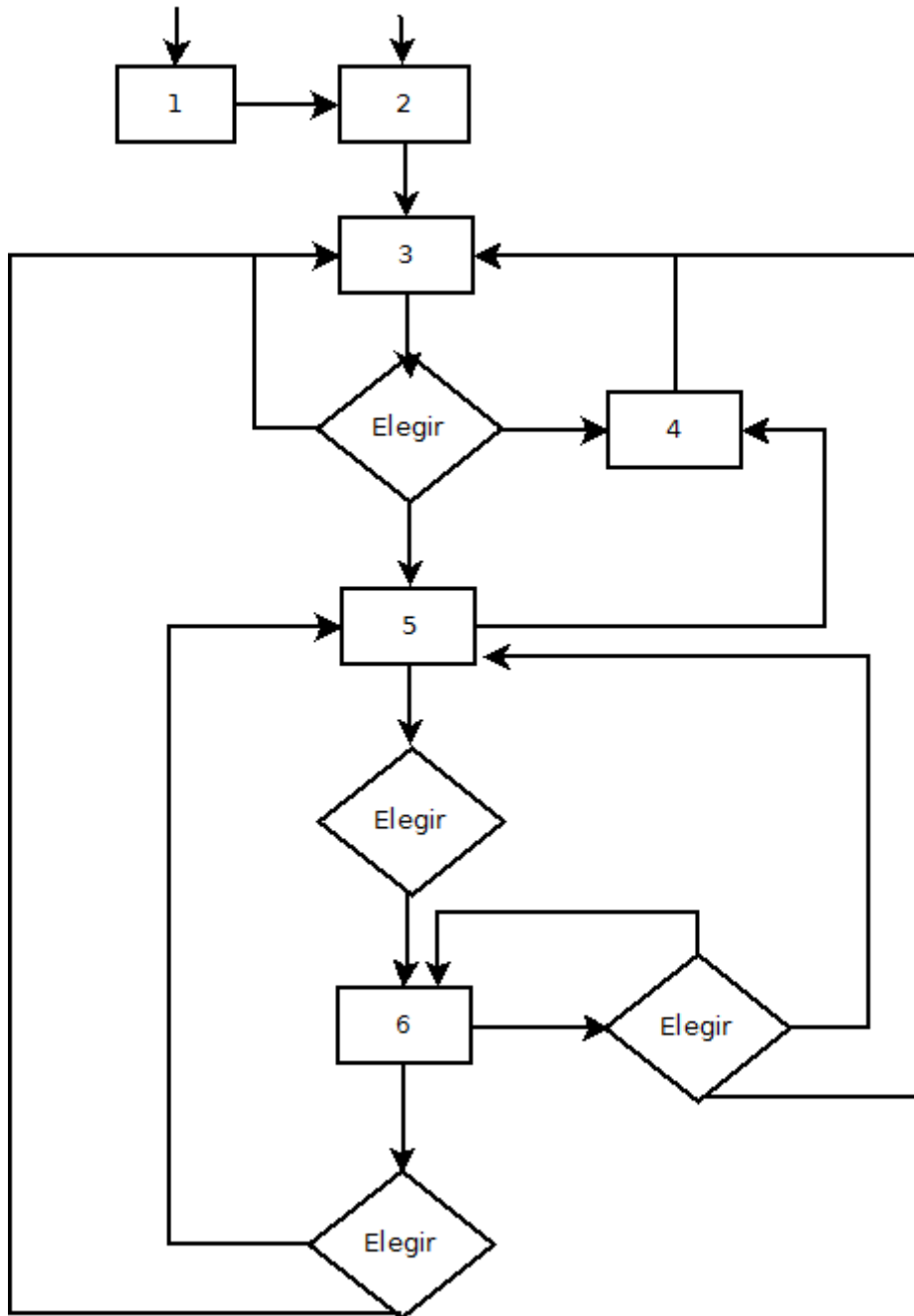


Figura 28. Prueba de caja blanca: algoritmo estudiante.

Bloque 1

Registrar datos estudiante nuevo.

Bloque 2

Acceder con los datos del estudiante.

Bloque 3

Seleccionar una lección (menú desplegable).

Bloque 4

Recargar una lección (botón reload).

Bloque 5

Selección de una pregunta (botones << y >>).

Bloque 6

Contestar una pregunta (botón check).

Como se puede observar tanto en la Figura 27 como en la Figura 28, se cubren todas las ramas del grafo comprobando que no hay caminos aislados o bucles de los que no podemos salir, y validando de esta forma el correcto funcionamiento de cada uno de los algoritmos (profesor y estudiante).

4.1.2 PRUEBAS DE CAJA NEGRA

En este apartado se desarrollan casos de prueba que se basan únicamente en el comportamiento de entrada/salida. Se prueba la herramienta con una serie de valores límite para comprobar si cumple o no con los requisitos funcionales.

Estos valores hacen llevar al límite a la herramienta para que pueda dar lugar a fallos y su correspondiente corrección.

Prueba 1: Registrar estudiante

Entrada	Datos de estudiante
Descripción	Se insertan los datos del estudiante para registrarlo en la herramienta: nombre de usuario, contraseña y dirección de correo electrónico.
Salida esperada	Mensaje informando al estudiante de que está registrado.
Salida real	Mensaje informando al estudiante de que está registrado.

Entrada	Datos de estudiante incorrectos
Descripción	Se insertan los datos del estudiante para registrarlo en la herramienta: Username: 1234 Password: **** Email: 1223@uno.es
Salida esperada	Mensaje informando al estudiante de que el nombre de usuario debe estar formado por letras, números, _ y comenzar con una letra.
Salida real	Mensaje informando al estudiante de que el nombre de usuario es incorrecto (debe estar formado por letras, números, _ y comenzar con una letra).

Entrada	Nombre de usuario repetido.
Descripción	Se insertan los datos del estudiante para registrarlo en la herramienta, pero el nombre del usuario existe ya.
Salida esperada	Mensaje informando al estudiante de que el nombre de usuario ya existe.
Salida real	Mensaje informando al estudiante de que el nombre de usuario ya existe.

Prueba 2: Autenticar estudiante/profesor

Entrada	Nombre usuario y contraseña correcta.
Descripción	Se inserta el nombre de usuario y contraseña correcta.
Salida esperada	Se permite acceso a la aplicación.
Salida real	Se permite acceso a la aplicación.

Entrada	Nombre usuario y contraseña incorrecta.
Descripción	Se inserta el nombre de usuario y/o contraseña incorrecta.
Salida esperada	Mensaje informando al usuario de que el nombre de usuario y/o contraseña son incorrectas.
Salida real	Mensaje informando al usuario de que el nombre de usuario y/o contraseña son incorrectas.

Prueba 3: Insertar lección (profesor)

Entrada	Insertar nueva lección.
Descripción	Se inserta una nueva lección.
Salida esperada	Se actualiza la lista de lecciones incluyendo la nueva lección.
Salida real	Se actualiza la lista de lecciones incluyendo la nueva lección.

Entrada	Insertar lección en blanco.
Descripción	Se deja en blanco el nombre de la lección y se pulsa Insertar.
Salida esperada	Mensaje informando de que el nombre de la lección no puede estar en blanco y por tanto, no se inserta.
Salida real	Mensaje informando de que el nombre de la lección no puede estar en blanco y por tanto, no se inserta.

Entrada	Insertar lección repetida
Descripción	Se introduce el nombre de una lección que ya existe en la lista.
Salida esperada	Mensaje informando de que el nombre de la lección existe y por tanto, no se inserta en la lista.
Salida real	Mensaje informando de que el nombre de la lección existe y por tanto, no se inserta en la lista.

Prueba 4: Borrar lección (profesor)

Entrada	Borrar lección existente.
Descripción	Se borra una lección existente.
Salida esperada	Mensaje pide confirmación de dicha operación. Si se confirma, se actualiza la lista de lecciones eliminando la lección.
Salida real	Mensaje pide confirmación de dicha operación. Si se confirma, se actualiza la lista de lecciones eliminando la lección.

Entrada	Borrar lección no seleccionada.
Descripción	No se ha marcado ninguna lección para borrar.
Salida esperada	Mensaje informando de que no hay lección seleccionada para eliminar.
Salida real	Mensaje informando de que no hay lección seleccionada para eliminar.

Prueba5: Insertar pregunta

Entrada	Insertar pregunta y respuesta.
Descripción	Se inserta una pregunta y una respuesta correcta.
Salida esperada	Se actualiza el listado de preguntas y respuestas mostrando la pregunta/respuesta insertada en primer lugar.
Salida real	Se actualiza el listado de preguntas y respuestas, pero no aparece en primer lugar.

→ Se corrige para que el profesor vea en primer lugar la última pregunta/respuesta insertada.

Entrada	Insertar respuesta en blanco a una pregunta.
Descripción	Se inserta una pregunta con una respuesta en blanco.
Salida esperada	Se actualiza el listado de preguntas y respuestas.
Salida real	Se actualiza el listado de preguntas y respuestas.

Entrada	Insertar pregunta en blanco
Descripción	Se inserta una pregunta en blanco.
Salida esperada	Mensaje informando de que la pregunta no puede estar en blanco.
Salida real	Mensaje informando de que la pregunta no puede estar en blanco.

Entrada	Insertar pregunta repetida.
Descripción	Se inserta una pregunta repetida.
Salida esperada	Mensaje informando de que no ha habido cambios porque considera que se trata de la misma pregunta.
Salida real	Al insertar de nuevo una pregunta existente en la lista, se selecciona del listado de preguntas dicha pregunta (con su respuesta).

→ No se considera un fallo, sino una funcionalidad distinta a la esperada, pero igual de válida.

Prueba 6: Borrar pregunta

Entrada	Borrar pregunta y respuesta
Descripción	Se selecciona una pregunta/respuesta para ser borrada.
Salida esperada	Mensaje pidiendo confirmación de dicha operación. Si se confirma, se borra la pregunta y la respuesta del listado.
Salida real	Mensaje pidiendo confirmación de dicha operación. Si se confirma, se borra la pregunta y la respuesta del listado.

Entrada	Borrar una pregunta y respuesta vacías.
Descripción	Se crea una nueva pregunta.
Salida esperada	Mensaje informando que no existen elementos que borrar.
Salida real	Mensaje informando que no existen elementos que borrar.

Prueba 7: Responder una pregunta (estudiante)

Entrada	Insertar respuesta correcta a una pregunta.
Descripción	Se inserta la respuesta correcta a una pregunta.
Salida esperada	Mensaje informando de que la respuesta es correcta.
Salida real	Mensaje informando de que la respuesta es correcta.

Entrada	Responder de nuevo a una pregunta contestada.
Descripción	Se escribe la respuesta a una pregunta.
Salida esperada	Mensaje informando de que la pregunta ya ha sido contestada.
Salida real	Mensaje informando de que la pregunta ya ha sido contestada.

Entrada	Responder incorrectamente a una pregunta.
Descripción	Se inserta la respuesta incorrecta a una pregunta.
Salida esperada	Mensaje informando de que la respuesta es incorrecta e indicando la solución correcta.
Salida real	Mensaje informando de que la respuesta es incorrecta e indicando la solución correcta.

4.2 PRUEBAS DE INTEGRACIÓN.

Estas pruebas permiten, partiendo de un módulo el cual se sabe que funciona correctamente por sí mismo, añadir otros módulos y comprobar que no se producen fallos. Se verifica la interacción entre componentes del sistema software.

Herramienta Profesor

Se realizan los siguientes pasos para integrar cada uno de los módulos hasta llegar a la Figura 11:

1. Integración de la base de datos con ESB.
2. Integración del módulo de acceso.
3. Integración del módulo de registro.
4. Integración del módulo de selección de lección.
5. Integración del módulo de añadir lección.
6. Integración del módulo de borrar lección.
7. Integración del módulo de seleccionar preguntas y respuestas.
8. Integración del módulo de guardar preguntas y respuestas.
9. Integración del módulo de modificar preguntas y/o respuestas.
10. Integración del módulo de borrar preguntas y respuestas.

Una vez integrados todos estos módulos, se continúa con la integración de la herramienta del estudiante hasta alcanzar la Figura 17.

Herramienta Estudiante

Se realizan los siguientes pasos para integrar cada uno de los módulos:

1. Integración de la base de datos con ESB (común profesor).
2. Se prueba el módulo de acceso (común profesor).
3. Se prueba el módulo de selección de lección (común profesor).
4. Integración del módulo de seleccionar preguntas del estudiante.
5. Integración del módulo de responder preguntas.
6. Integración del módulo de recargar preguntas.

Al integrar estos módulos se obtiene la herramienta Edu-Reviews.

Por último, a esta herramienta se le incorpora un nuevo módulo, el agente voki. Este agente se ha diseñado para que interactúe con la herramienta. Su función es tanto ayudar al usuario como intentar aumentar su satisfacción en la interacción con el sistema.

1. Voki da la bienvenida e invita al usuario a registrarse.



Figura 29. Voki. Bienvenida a Edu-Reviews.

2. Profesor:
 - a. Accede a la herramienta. El voki le indica cómo puede crear y eliminar las lecciones.

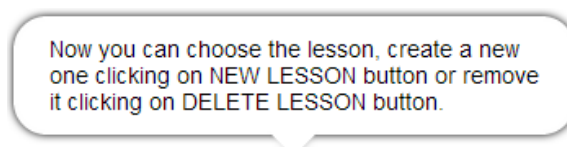


Figura 30. Voki. Administración y manejo de lecciones.

- b. Accede a una lección. El voki le indica cómo puede crear, eliminar, modificar las preguntas y las respuestas de las lecciones.

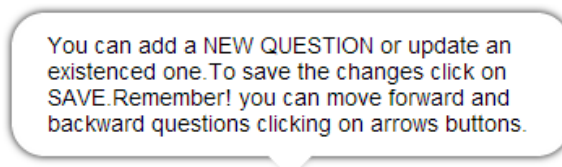
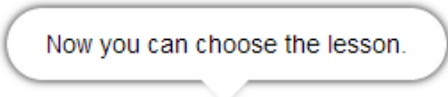


Figura 31. Voki. Administración y manejo de preguntas.

3. Estudiante:

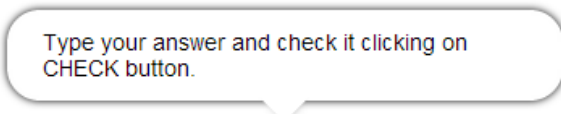
- a. Accede a la herramienta Edu-Reviews. El voki le indica que debe seleccionar una lección para continuar.



Now you can choose the lesson.

Figura 32. Voki. Elegir lección.

- b. Accede a una lección y el voki le indica cómo corregir la respuesta de la pregunta seleccionada.



Type your answer and check it clicking on CHECK button.

Figura 33. Voki. Comprobar respuesta.

4.3 PRUEBAS DE VALIDACIÓN

En este apartado se verifica si el sistema cumple todos los requisitos del cliente que se pedían en la fase análisis y permite que los usuarios del sistema den el visto bueno definitivo. Se trata de determinar si se está construyendo el producto correcto.

Se mostrará mediante capturas de pantalla como se han ido cumpliendo estos requisitos.

1. RF1 – Deben existir dos tipos de usuarios: estudiante y profesor. El estudiante se podrá registrar online, mientras que el profesor tendrá que pedir acceso al administrador del sistema.

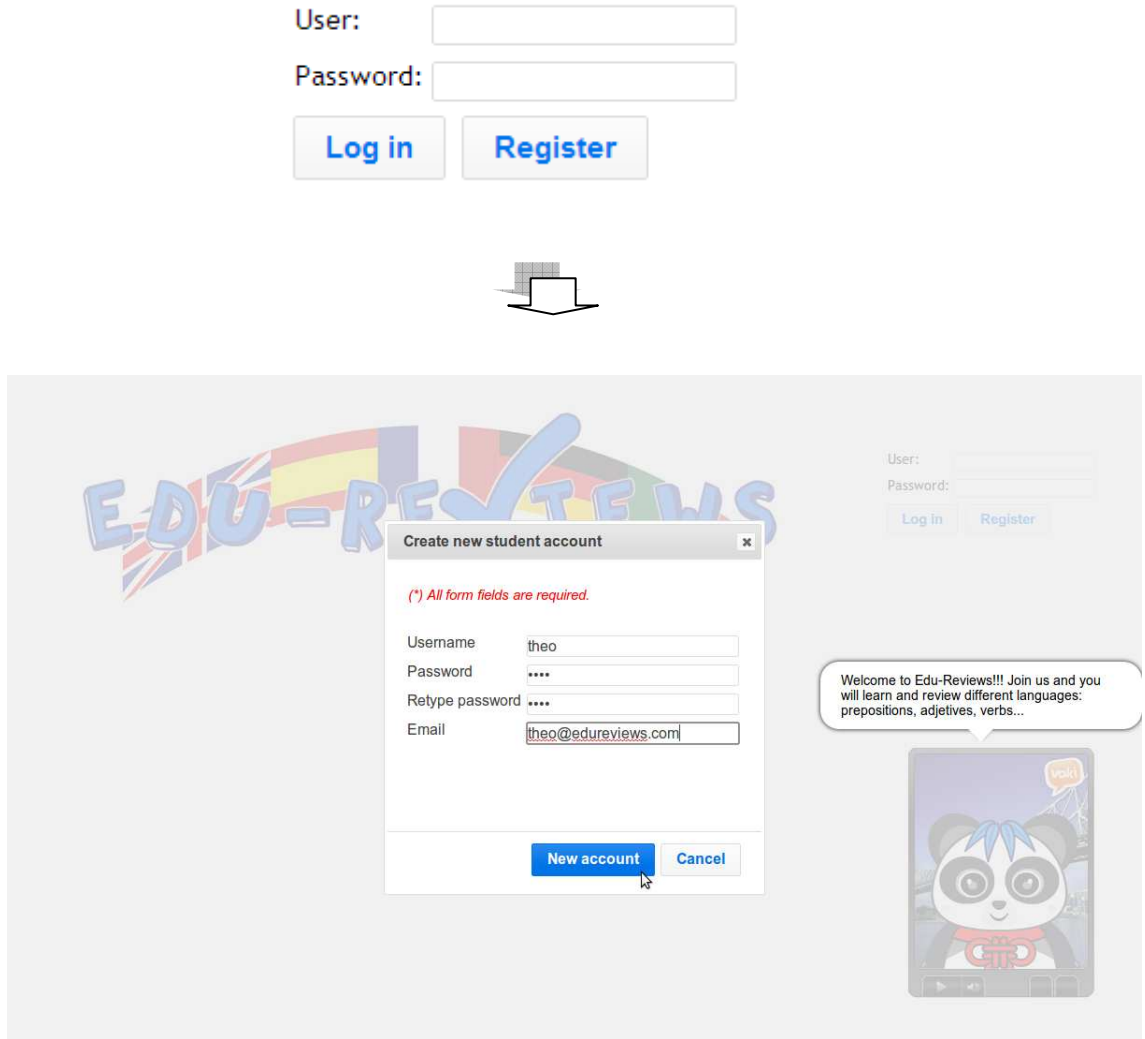


Figura 34. Crear nueva cuenta de estudiante.

Como se observa en la Figura 34, el estudiante pulsa sobre el botón Register y le aparece una ventana de formulario para el registro. Una vez que se han rellenado todos los datos, pulsar New account para generar la nueva cuenta.

2. RF2 – El usuario (profesor/estudiante) podrá autenticarse en el sistema con su nombre de usuario y su contraseña.

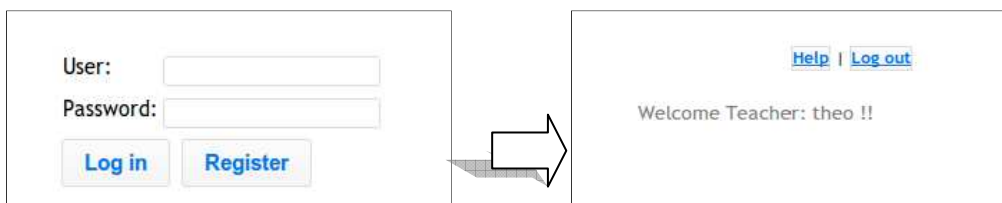


Figura 35. Autenticación del usuario en Edu-Reviews.

Como se observa en la Figura 35, el usuario introduce sus datos (User/password) y pulsa sobre el botón Log in. Al ser correctos los datos aparece un mensaje de bienvenida.

3. RF3 – El profesor podrá seleccionar, crear y eliminar lecciones.



Figura 36. Seleccionar, crear o eliminar una lección.

Como se observa en la Figura 36, el profesor podrá seleccionar una lección del menú desplegable, o bien pulsar el botón NEW LESSON para crear una nueva lección, o bien pulsar el botón DELETE LESSON para borrar la lección seleccionada.

4. RF4 – El profesor no podrá crear lecciones con el mismo nombre.

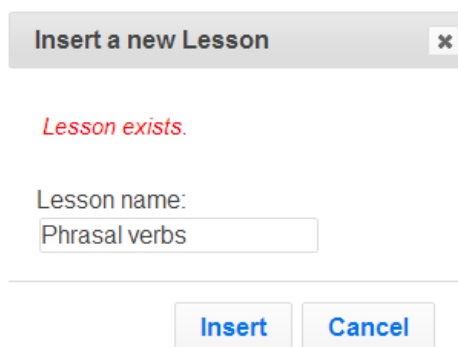


Figura 37. Insertar nombre de lección ya existente.

Como se observa en la Figura 37, el profesor trata de usar un nombre de lección existente. Al pulsar el botón Insert, se le indica mediante un mensaje que la lección existe.

5. RF5 – El profesor podrá seleccionar, crear, modificar y eliminar preguntas y respuestas.



Question: Quick! the bus. It's ready to leave. NEW QUESTION

Answer: Get on

Save Delete << >>

Figura 38. Crear, modificar, eliminar y seleccionar preguntas y respuestas.

Como se observa en la Figura 38, se puede crear una nueva pregunta (botón NEW QUESTION), modificar una pregunta existente (guardándola posteriormente con el botón Save), eliminar la pregunta/respuesta actual (Delete) y seleccionar cualquiera de las preguntas de la lección actual (con los botones << y >>).

6. RF6 – El profesor no podrá crear preguntas iguales.

En este caso, al escribir el nombre de una pregunta que ya existe en el listado de preguntas, se selecciona por defecto dicha pregunta.

7. RF7 – El profesor podrá crear una pregunta cuya respuesta esté vacía.



Question: My sister often stays at Uncle Tim's in Detroit. NEW QUESTION

Answer:

Save Delete << >>

Figura 39. Crear pregunta con respuesta vacía.

Como se observa en la Figura 39, el profesor crea una pregunta con respuesta vacía y la guarda con el botón Save.

8. RF8 – El profesor podrá crear preguntas con varias respuestas



Question: Quick! the bus. It's ready to leave. NEW QUESTION

Answer: Get on

Save Delete << >>

Figura 40. Crear pregunta con varias respuestas.

Como se aprecia en la Figura 40, el profesor crea una pregunta con dos huecos que corresponden a las dos palabras de la respuesta.

9. RF9 - El estudiante podrá seleccionar cada una de las lecciones insertadas por el profesor.

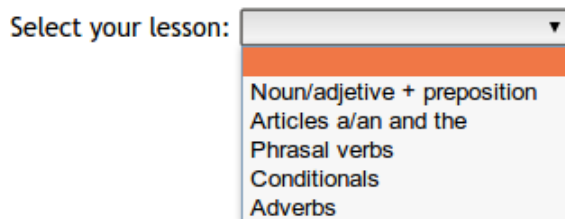


Figura 41. Seleccionar lección.

Como se observa en la Figura 41, el estudiante pulsa sobre la flecha del menú para desplegarlo y observar el listado de lecciones.

10. RF10 – El estudiante podrá recargar una lección una vez contestadas todas las preguntas correspondientes a dicha lección.

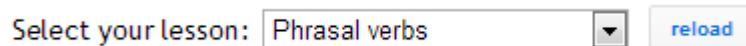
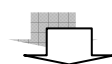
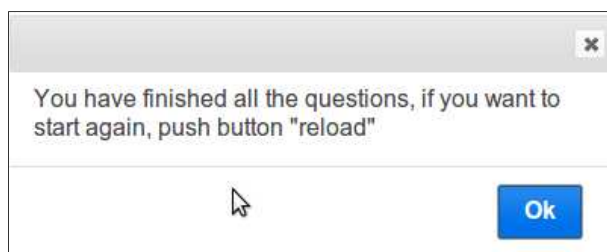


Figura 42. Recargar una lección.

Como se aprecia en la Figura 42, una vez respondidas todas las preguntas de una lección, aparecerá un mensaje invitando a pulsar el botón reload para recargar el listado de preguntas de la lección.

11. RF11 – El estudiante podrá seleccionar cualquiera de las preguntas pertenecientes a la lección seleccionada.

Select your lesson:

Question:

Answer:

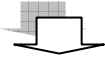

Figura 43. Seleccionar preguntas.

Como se observa en la Figura 43, el estudiante, empleando los botones << y >> podrá recorrer hacia delante o hacia atrás la lista de las preguntas de la lección seleccionada.

12. RF12 –El estudiante podrá comprobar si la respuesta que ha introducido es correcta.

Question:

Answer:



This is not the correct answer. The correct answer is
--> for

Figura 44. Comprobar respuesta.

Como se puede apreciar en la Figura 44, si el estudiante escribe en el campo Answer la respuesta y esta es incorrecta, aparece un mensaje indicando que esa respuesta es incorrecta junto con la respuesta correcta.

4.4 PRUEBAS DE ACEPTACIÓN

En este apartado se describen las pruebas realizadas por usuarios para la aceptación del sistema con el fin de evaluar el grado de satisfacción alcanzado por los usuarios (profesores / estudiantes) y la completitud de los objetivos marcados.

Los perfiles elegidos han sido al azar, tanto de perfiles técnicos como de perfiles no técnicos. La Figura 45 muestra las preguntas realizadas a los usuarios de Edu-Reviews. La información de las primeras cuatro preguntas se recoge en la Tabla 20 (valoración de la facilidad de manejo, rapidez, diseño y opinión general) en la escala de 0 (total disconformidad) a 10 (total conformidad).

Cuestionario Satisfacción de Edu-Reviews

Responda a las siguientes preguntas con su valoración del 0 (total disconformidad) al 10 (total conformidad).

1. ¿Le ha gustado la aplicación?
2. ¿Cree que la aplicación es útil?
3. ¿Le ha resultado sencillo interactuar con la aplicación?
4. ¿Ha necesitado ayuda en algún momento?

Responda brevemente.

5. ¿Qué le parece la herramienta?
6. ¿Qué encuentra más fácil de usar?
7. ¿Qué encuentra más difícil de usar?
8. ¿Qué cambiaría?

Figura 45. Cuestionario Satisfacción de Edu-Reviews.

Este cuestionario era facilitado de forma individual a cada usuario al término de interactuar una media de 15 minutos con la herramienta. Se puede observar cómo la valoración de los usuarios es positiva, tanto en facilidad como en manejo, independientemente del perfil de los usuarios.

También se ha probado Edu-Reviews en varios navegadores como Firefox y Google Chrome.

Rango edad	Perfil usuario	¿Le ha gustado la aplicación?	¿Cree que la aplicación es útil?	¿Le ha resultado sencillo interactuar con la aplicación?	¿Ha necesitado ayuda en algún momento?	Tiempo en finalizar las 3 tareas.
35-40	No técnico	9	9	10	2	20'
30-35	Técnico agrícola	10	10	10	1	15'
30-35	Técnico informático	8	9	10	0	15'
30-35	Técnico industrial	8	8	8	2	20'
30-35	Técnico telecomunicaciones	7	9	8	3	15'
30-25	Técnico informático	10	9	9	3	20'
20-25	No técnico	9	9	9	2	10'
35-40	Técnico telecomunicaciones	9	9	8	2	15'
30-35	Técnico informático	9	10	9	2	20'
MEDIA		8.8	9.1	9	1.8	17'

Tabla 20. Tabla resultado pruebas aceptación.

Estos resultados nos sirven para analizar los requisitos no funcionales que establecimos en la fase inicial del proyecto:

RNF1 – Edu-Reviews debe ser usable.

RNF2–Edu-Reviews tendrá un interfaz amigable, sencilla e intuitiva.

Estos requisitos se cumplen al estudiar los resultados obtenidos. El 100% de los usuarios, a la pregunta de si les gusta la aplicación marcan un valor superior a 5, y en promedio el valor es de 8.8. Respecto a la utilidad de la aplicación, el 100% coincide en

que es muy útil con un valor medio de 9.1. Respecto a la sencillez de la aplicación, el 100% de los usuarios proporcionan un valor superior a 5, y en promedio el valor es de 9. Respecto a la pregunta de si ha necesitado ayuda en algún momento, el 100% de los usuarios marcan un valor inferior a 5, por lo que con un promedio del 1.8 se puede inferir que la herramienta es bastante intuitiva.

Respecto al tiempo, en promedio los usuarios han completado 17 minutos en el que los usuarios, dependiendo de si se trataban de profesores o alumnos, realizaban distintas tareas. Mientras que a los profesores se les indicaba loguearse, crear una lección y nuevas preguntas, los alumnos debían loguearse, seleccionar una lección y responder preguntas.

5. CONCLUSIONES Y TRABAJO FUTURO

5.1 CONCLUSIONES

Los objetivos marcados al inicio se han cumplido mediante la implementación de la herramienta Edu-Reviews, en particular el primer objetivo “Desarrollar una aplicación para el profesor que le permita crear preguntas de rellenar huecos de forma intuitiva y con la guía de un avatar animado” se cumple porque la herramienta permite el acceso y registro de profesores. Los profesores pueden crear, borrar lecciones y crear, borrar, modificar preguntas en cada lección. Estas operaciones son guiadas mediante un voki.

El segundo objetivo “Desarrollar una aplicación para el estudiante que le permita repasar preguntas de rellenar huecos en inglés de forma intuitiva y con la guía de un avatar animado” se cumple porque la herramienta permite el acceso y el registro de estudiantes. Los estudiantes pueden seleccionar distintas lecciones y contestar a las preguntas. Estas operaciones son guiadas mediante un voki.

El tercer objetivo “Realizar el desarrollo según una Arquitectura Orientada a Servicios de una aplicación educativa con tecnología actual como ESB, AJAX, JSON midiendo el coste de aprendizaje respecto a los beneficios obtenidos” se cumple con esta herramienta y con el análisis realizado posteriormente de su uso y de sus beneficios.

Las pruebas realizadas han verificado el correcto funcionamiento del sistema, y también se han validado los requisitos expuestos en el análisis como se describe a continuación.

En la fase de análisis, se establecieron los siguientes requisitos funcionales:

RF1 – Deben existir dos tipos de usuarios: estudiante y profesor. El estudiante se podrá registrar online, mientras que el profesor tendrá que pedir acceso al administrador del sistema.

RF2 – El usuario (profesor/estudiante) podrá autenticarse en el sistema con su nombre de usuario y su contraseña.

RF3 – El profesor podrá seleccionar, crear y eliminar lecciones.

RF4 – El profesor no podrá crear lecciones con el mismo nombre.

RF5 – El profesor podrá seleccionar, crear, modificar y eliminar preguntas y respuestas.

RF6 – El profesor no podrá crear preguntas iguales.

RF7 – El profesor podrá crear una pregunta cuya respuesta esté vacía.

RF8 – El profesor podrá crear preguntas con varias respuestas.

RF9 – El estudiante podrá seleccionar cada una de las lecciones insertadas por el profesor.

RF10 – El estudiante podrá recargar una lección una vez contestadas todas las preguntas correspondientes a dicha lección.

RF11 – El estudiante podrá seleccionar cualquiera de las preguntas pertenecientes a la lección seleccionada.

RF12 –El estudiante podrá comprobar si la respuesta que ha introducido es correcta.

Todos los requisitos antes enunciados se han cumplido como demostramos en la parte de validación de pruebas del apartado anterior de la memoria.

También se establecieron los siguientes requisitos no funcionales:

RNF1 – Edu-Reviews debe ser usable.

RNF2–Edu-Reviews tendrá un interfaz amigable, sencilla e intuitiva.

Estos requisitos no funcionales también se han cumplido satisfactoriamente como se ha validado en las pruebas realizadas.

Desde mi punto de vista, este tipo de arquitecturas SOA supone un gran coste en cuanto al aprendizaje y la definición de lo que se tiene que hacer al principio. Una vez que se tiene claro qué se tiene que hacer y cómo, se convierte en una forma cómoda y simple de trabajar. Es interesante para desarrollos a largo plazo, donde se prevee incorporar nuevas funcionalidades en el futuro. En cambio, si se trata de sistemas con un enfoque a corto plazo, la utilización de esta arquitectura puede ser muy costosa.

En comparación con las arquitecturas que he manejado hasta el momento, como el sistema distribuido en ADA y conectado por sockets vía TCP, el hecho de añadir cualquier nuevo componente (escrito, por ejemplo, en Java) y su conexión con éste supone un gran coste en cuanto al tiempo de desarrollo y líneas de código. Para una arquitectura de tipo SOA, el único requisito necesario sería conocer las interfaces de ambos componentes para establecer dicha comunicación empleando un ESB. Por lo que este trabajo se simplifica enormemente.

5.2 TRABAJO FUTURO

Se podrían incluir como trabajo futuro los siguientes puntos:

- Permitir al profesor añadir explicación teórica de cada lección.
- Añadir algún apartado en el que se ofrezcan opciones de respuesta a la pregunta (preguntas tipo test) y otros tipos de respuesta.
- En el logo de la aplicación aparecen banderas de distintos países, la idea es internacionalizarlo para varios idiomas, por ejemplo mediante el uso de un botón que permita seleccionar el idioma a repasar.

BIBLIOGRAFÍA

Referencias

[Davis, 2009] Davis, J. (2009), “SOA OPEN SOURCE”. Ed. Anaya MULTIMEDIA.

[Del Caso, 2010] Del Caso, A., (2010), “Javascript Guía Práctica”, Ed. Anaya MULTIMEDIA.

[Dossot & D’Emic, 2010] Dossot, D., D’Emic J. (2010) “Mule in Action”, Ed Manning Publications Co.

[Erl, 2007] Erl, T. (2007), “SOA Principles of Service Design”, Ed. Prentice Hall/Pearson PTR.

[Freeman, 2008] Freeman, R. G., (2008), “Oracle database 11g new features”, Ed. McGraw-Hill.

[Gilfillan, 2003] Gilfillan, I. (2003), “MySQL”, Ed. Anaya MULTIMEDIA.

[Hurwitz et al. 2009] Hurwitz, J., Bloor, R., Kaufman, M., Halper, F. (2009), “Service Oriented Architecture for Dummies”, Wiley Publishing Inc.

[Piattini et al, 2003] Piattini, M., Calvo-Manzano, J., Cervera, J., Fenandez, L. (2003) “Análisis y diseño de aplicaciones informáticas de gestión, una perspectiva de ingeniería del software”, Ed. RA-MA.

[Vera, 2008] Vera, J. A., (2008), “El gran libro de Windows Server 2008”, Ed Marcombo.

Referencias Web

HTTP 1 [http:// es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios](http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios)

HTTP 2 <http://www.e-abclearning.com/queesunaplataformadeelearning>

HTTP 3 <http://www.mulesoft.org/>

HTTP 4 <http://www.voki.com/>

HTTP 5 http://en.wikipedia.org/wiki/Enterprise_service_bus

HTTP 6 <http://architects.dzone.com/news/top-open-source-esbs>

HTTP 7 http://en.wikipedia.org/wiki/Apache_ServiceMix

HTTP 8 <http://servicemix.apache.org/>

HTTP 9 <http://www.open-esb.net/>

HTTP 10 <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>

HTTP 11 <http://www.desarrolloweb.com/articulos/1680.php>

HTTP 12 http://www.postgresql.org/es/sobre_postgresql

HTTP 13 http://www.wikivs.com/wiki/MySQL_vs_PostgreSQL

HTTP 14 <http://jqueryui.com/>

HTTP 15 <http://www.w3schools.com>

HTTP 16 <http://www.desarrolloweb.com/>

HTTP 17 <http://www.elrincondeajax.com/manual-ajax/>

HTTP 18 <http://www.json.org/json-es.html>

HTTP 19 <http://glassfish.java.net>

HTTP 20 http://es.wikipedia.org/wiki/Servidor_HTTP_Apache

HTTP 21 <http://www.ruby-lang.org/es/about/>

HTTP 22 <http://groovy.codehaus.org/>

HTTP 23 <http://silvanolte.com/blog/2012/12/11/the-differences-between-groovy-and-ruby/>

HTTP 24 <http://www.mulesoft.org/documentation/display/current/Message+Processors>

ANEXO 1: MANUAL DE USUARIO

El acceso del usuario (profesor o estudiante) a la herramienta Edu-Reviews se realiza vía web, pero no se ha podido instalar en un servidor. Las pruebas se han realizado de forma local utilizando un portátil.


PROFESOR

Registro:

- Para el registro de profesores, ponerse en contacto con admin@edureviews.com.

Acceso a la herramienta:

- Introduzca el nombre de usuario y la contraseña y pulse el botón Log in. Se muestra la siguiente pantalla:



The screenshot shows the Edu-Reviews interface. At the top, the logo 'EDU-REVIEWS' is displayed in a stylized, colorful font. Below the logo, there is a dropdown menu labeled 'Select your lesson:' with a downward arrow. To the right of the dropdown are two buttons: 'NEW LESSON' and 'DELETE LESSON'. In the top right corner, there are links for 'Help' and 'Log out'. Below the main interface, a speech bubble contains the text: 'Now you can choose the lesson, create a new one clicking on NEW LESSON button or remove it clicking on DELETE LESSON button.' Below the speech bubble is a small image of a panda character on a smartphone screen.

Operaciones sobre las lecciones:

- Seleccionar lección:
 - o Despliegue el menú de lecciones y pulse sobre el nombre de una lección.
- Añadir lección:
 - o Pulse el botón NEW LESSON. Aparece la siguiente ventana:

Insert a new Lesson✕

(*) All form fields are required.

Lesson name:

InsertCancel

- Introduzca el nombre de la nueva lección y pulse sobre el botón Insert.

- Borrar lección:

- Seleccione la lección del menú desplegable de lecciones.
- Pulse el botón DELETE LESSON para borrar la lección. Aparece la siguiente ventana:

Delete lesson✕

Are you sure you want to delete this lesson?

AcceptCancel

- Pulse Accept para borrar la lección o Cancel para cancelar la operación de borrado.



[Help](#) | [Log out](#)

Welcome Teacher: theo !!

Select your lesson: NEW LESSON DELETE LESSON

Question: NEW QUESTION

Answer:

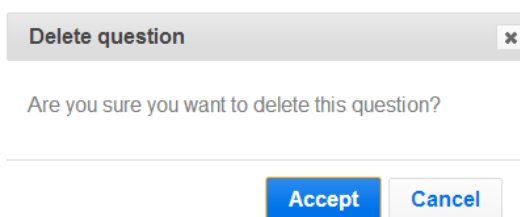
SaveDelete<<>>

You can add a NEW QUESTION or update an existed one. To save the changes click on SAVE. Remember! you can move forward and backward questions clicking on arrows buttons.



Operaciones sobre las preguntas:

- Seleccionar pregunta:
 - o Pulse sobre los botones >> y << para mostrar la pregunta siguiente o anterior, respectivamente.
- Añadir pregunta:
 - o Pulse sobre el botón NEW QUESTION.
 - o Escriba la pregunta en el campo Question.
 - o Escriba la respuesta en el campo Answer.
 - o Pulse el botón Save para guardar la nueva pregunta/respuesta.
- Borrar pregunta:
 - o Seleccione una pregunta del listado de preguntas.
 - o Pulse sobre el botón Delete. Se abrirá la siguiente ventana de confirmación:



- o Pulse Accept para borrar la pregunta o Cancel para cancelar la operación de borrado.
- Modificar pregunta:
 - o Seleccione una pregunta del listado de preguntas.
 - o Modifique en el campo Question la pregunta y/o en el campo Answer la respuesta.
 - o Pulse Save para guardar los cambios.

ESTUDIANTE

Registro de un nuevo estudiante:

- Pulse sobre el botón Register
- Se abre una ventana de registro, insertar los datos y pulsar New account.

Create new student account ✕

() All form fields are required.*

Username

Password

Retype password

Email

- Se abre una ventana informando de que el registro ha sido realizado con éxito.

Acceso a la herramienta:

- Introduzca el nombre de usuario y la contraseña y pulse el botón Log in.

Operaciones sobre las lecciones:



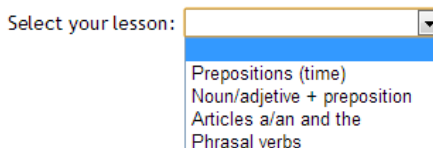
Select your lesson:

[Help](#) | [Log out](#)
Welcome Student: juan

Now you can choose the lesson.



- Seleccionar lección
 - o Despliegue el menú de lecciones y pulse sobre el nombre de una lección.



Operaciones sobre las preguntas (correspondientes a una lección):

A screenshot of the 'EDU-REVIEWS' interface. The title 'EDU-REVIEWS' is in large, colorful, stylized letters. Below it, there's a dropdown menu for 'Select your lesson:' with 'Noun/adjective + preposition' selected. A question field contains the text 'Don't ask me to decide. I'm not very good ___ making decisions.' The answer field contains 'at'. There are 'Check', '<<', and '>>' buttons. To the right, there's a 'Welcome Student: juan' message and a 'vaki' character on a smartphone screen. A speech bubble says 'Type your answer and check it clicking on CHECK button.'

- Seleccionar pregunta:
 - o Pulse sobre los botones >> y << para mostrar la pregunta siguiente o anterior, respectivamente.
- Responder pregunta:
 - o Seleccione una pregunta.
 - o Introduzca la respuesta en el campo Answer.
 - o Pulse sobre el botón Check para comprobar si es correcta.
 - o En caso de que sea correcta, se muestra una ventana animando al usuario a seguir.
 - o En caso de que sea incorrecta, se muestra una ventana (indicando al usuario que la respuesta no es correcta, y la respuesta correcta).

ANEXO 2: MANUAL DE INSTALACIÓN

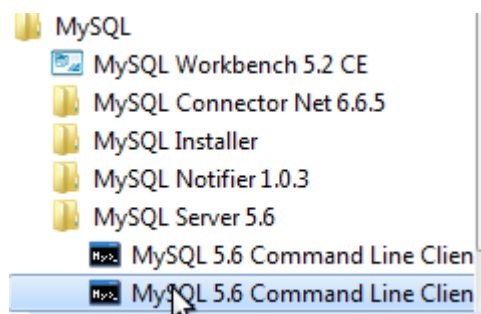
Pasos a realizar para la instalación de la aplicación Edu-Reviews:

1. Instale **Mysql Server 5.6.10** en la máquina.

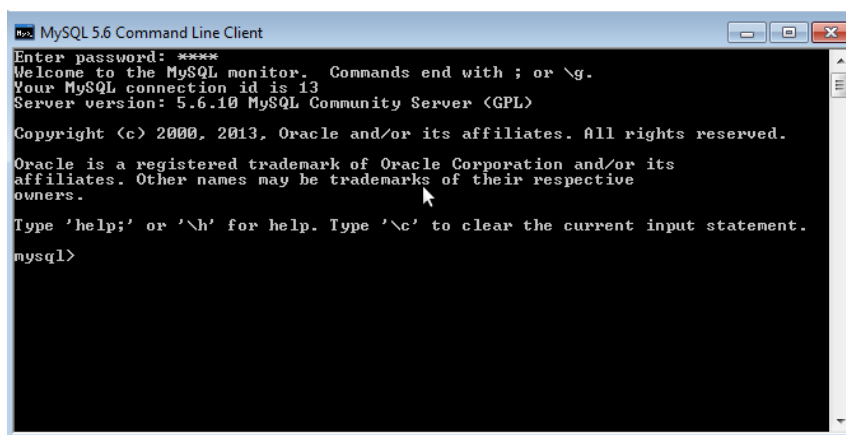
- **WINDOWS:** *mysql-installer-community-5.6.10.1.msi*

Se crea el usuario “root” con la contraseña 1122

Una vez instalado MySQL Server, se arranca la consola MySQL, seleccionando MySQL 5.6 Command Line Client:



Introduzca la password y pulse intro:



Cargue los tres scripts de la aplicación EduReviews con el comando:

```
mysql> source catalog.sql
```

```
mysql> source data.sql
```

```
mysql> source data_award.sql
```

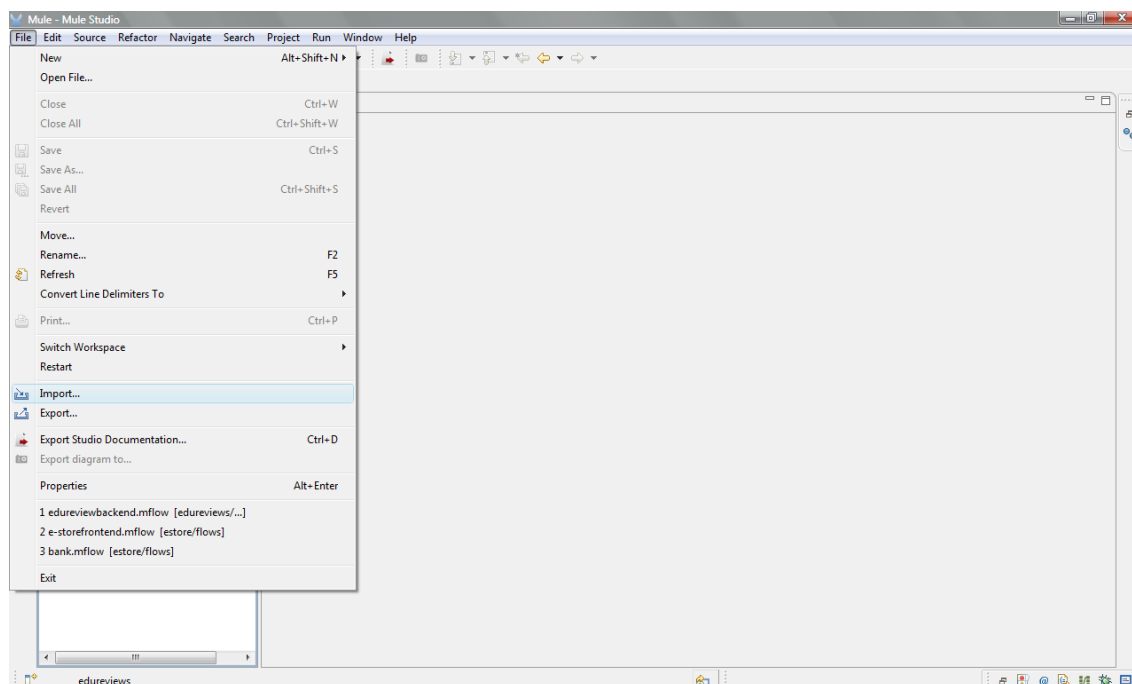
2. Instale **JRE 1.6**.
3. Instale la versión de **MuleStudio** correspondiente al SO (en este caso usaremos Windows).

La página de descarga de dicha aplicación es <http://www.mulesoft.org/download-mule-esb-community-edition#>

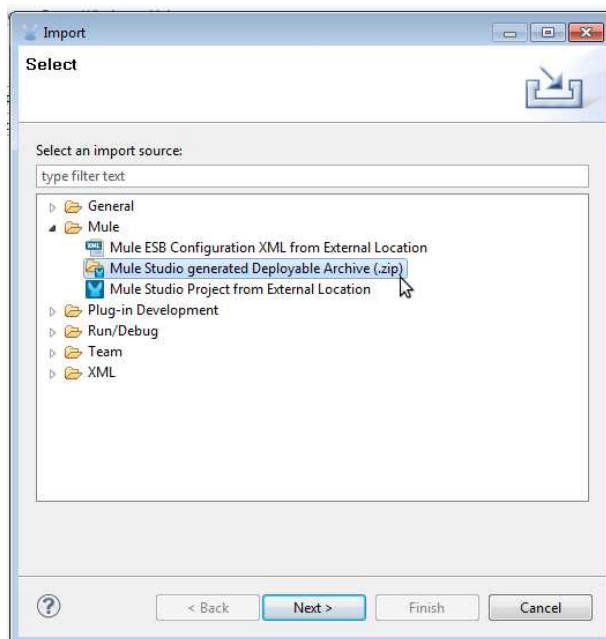
- **LINUX:** *MuleStudio-for-linux-32bit-1.3.2-201212121942.tar.gz*
- **WINDOWS:** *MuleStudio-for-win-32bit-3.4.0-201302201319.zip*

Una vez instalado **MuleStudio**:

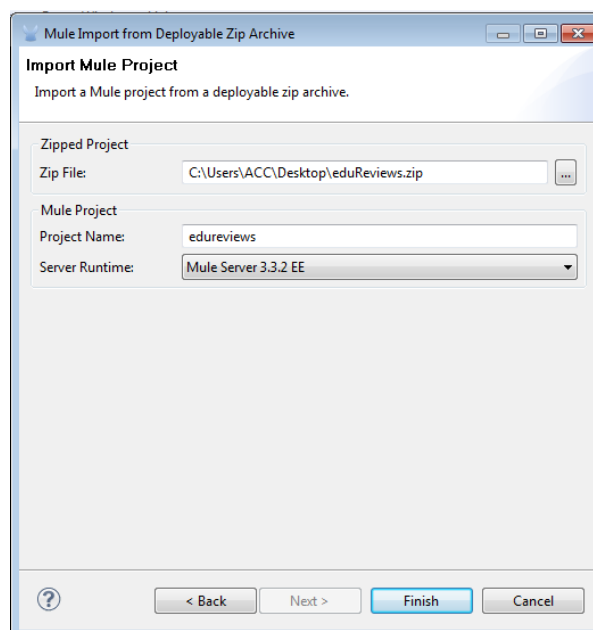
- Seleccione **File > Import**



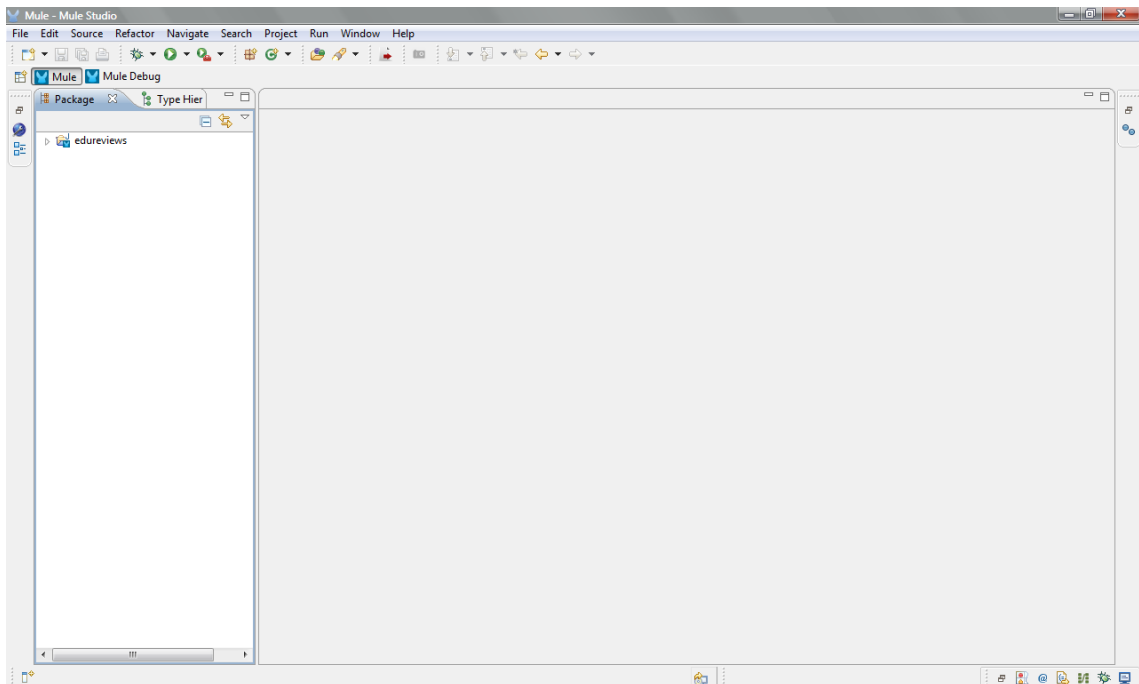
- Seleccione la carpeta Mule, y **Mule Studio generated Deployable** y pulsar Next.



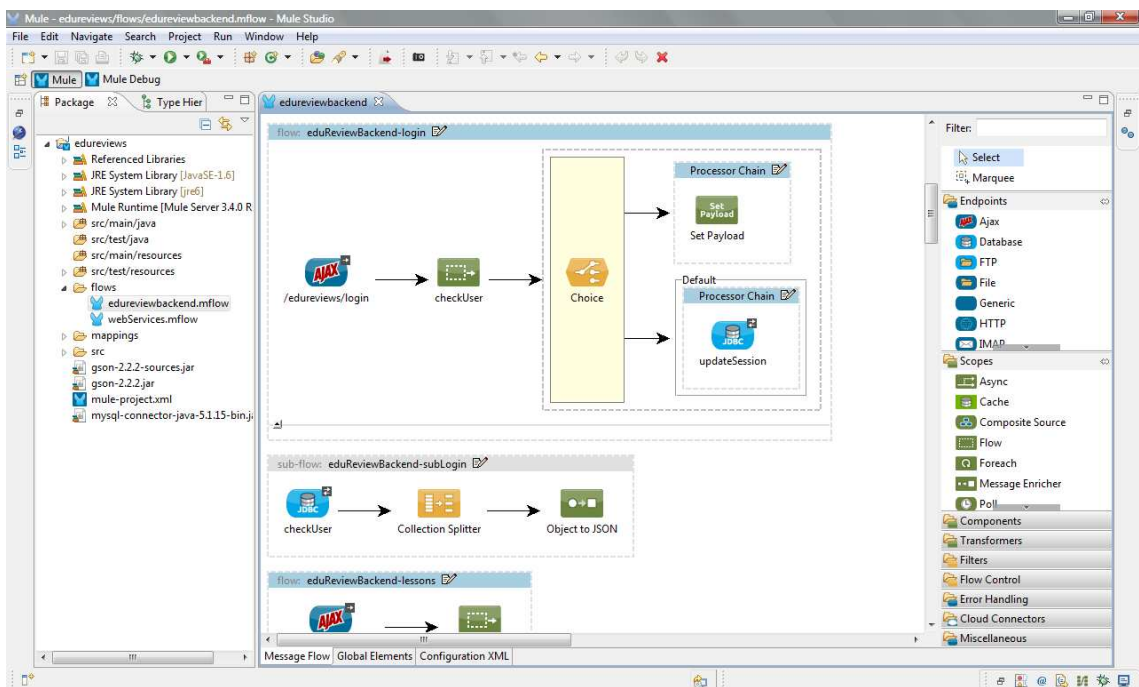
- En Zip File escriba la ruta donde se encuentra el fichero **edureviews.zip**
En Project Name: Edureviews y pulse el botón Finish.



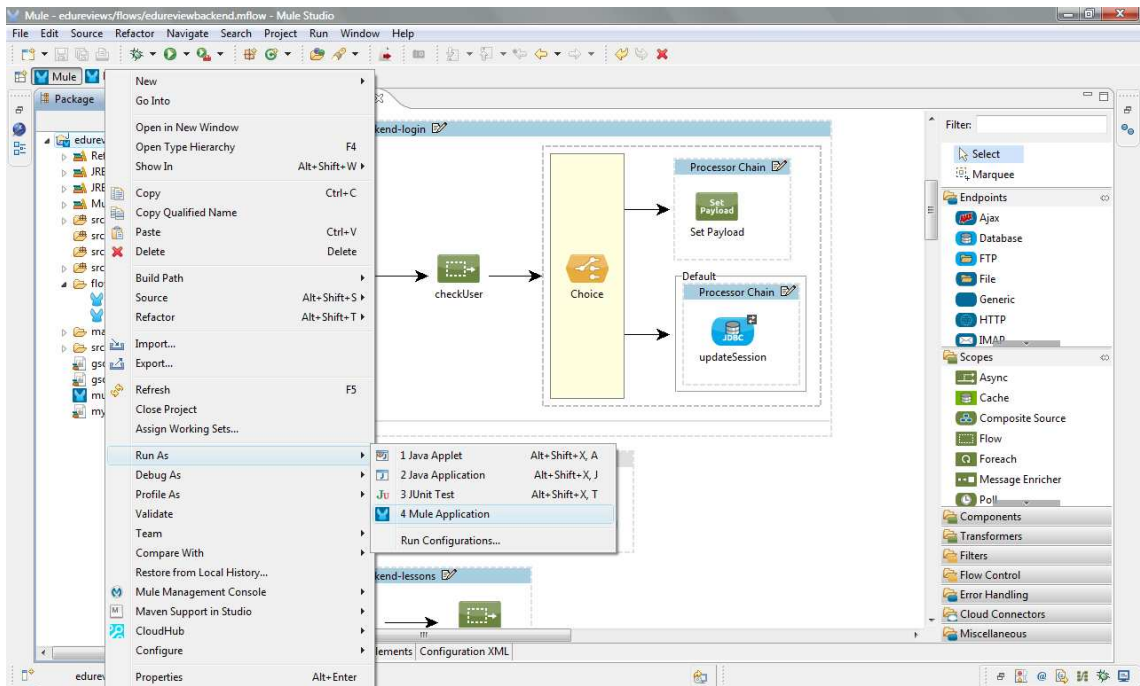
- Una vez finalizada la carga se observa la siguiente ventana:



4. Despliegue la carpeta edureviews:



5. Para arrancar el proyecto, pulse el botón derecho sobre la carpeta edureviews, seleccionar **Run As> Mule Application**



6. Espere hasta que en la ventana Console aparezca **Started app 'edureviews'**.

