

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

GRADO EN INGENIRÍA INFORMÁTICA

Trabajo Fin de Grado

Desarrollo e implementación de una herramienta para gestionar los
acuerdos de estudios de los estudiantes Erasmus

Autor: Javier Martín Gómez

Tutor/es: Iván González Diego

TRIBUNAL:

Presidente:

Vocal 1º:

Vocal 2º:

FECHA:

Agradecimientos

Índice

AGRADECIMIENTOS	3
ÍNDICE	5
ÍNDICE DE FIGURAS.....	7
ÍNDICE DE TABLAS.....	9
RESUMEN	10
RESUMEN.....	10
ABSTRACT	11
RESUMEN EXTENDIDO	12
INTRODUCCIÓN	15
CONTEXTO.....	15
MOTIVACIÓN	17
OBJETIVOS	18
ANÁLISIS Y ESPECIFICACIÓN DE REQUISITOS	20
DESCRIPCIÓN DEL SISTEMA.....	20
CATÁLOGO DE REQUISITOS	21
CASOS DE USO	24
DISEÑO	31
MODELADO DE DATOS	31
MODELADO DE CLASES.....	32
DIAGRAMAS DE SECUENCIA.....	34
IMPLEMENTACIÓN	45
MEDIOS UTILIZADOS.....	45
ESTRUCTURA DE PAQUETES	45
ACCESO Y OBTENCIÓN DE DATOS.....	46
MANEJO DE ACCIONES DEL USUARIO	47
PRESUPUESTO	50
COSTE DE MANO DE OBRA.....	50
COSTE DE HARDWARE Y MATERIALES.....	51
COSTE DE SOFTWARE.....	51
COSTE TOTAL.....	51
CONCLUSIONES Y TRABAJO FUTURO	53
CONCLUSIONES.....	53
TRABAJO FUTURO	53
MANUAL DE USUARIO (ANEXO).....	56
MANUAL DEL ALUMNO	57
MANUAL DEL PROFESOR.....	61
BIBLIOGRAFÍA	70

Índice de figuras

[illegible]

ESCRIBIR EL TÍTULO DEL CAPÍTULO (NIVEL 1)	4
ESCRIBIR EL TÍTULO DEL CAPÍTULO (NIVEL 1)	1
ESCRIBIR EL TÍTULO DEL CAPÍTULO (NIVEL 1)	4
ESCRIBIR EL TÍTULO DEL CAPÍTULO (NIVEL 1)	1
ESCRIBIR EL TÍTULO DEL CAPÍTULO (NIVEL 1)	4
ESCRIBIR EL TÍTULO DEL CAPÍTULO (NIVEL 1)	1
ESCRIBIR EL TÍTULO DEL CAPÍTULO (NIVEL 1)	4
ESCRIBIR EL TÍTULO DEL CAPÍTULO (NIVEL 1)	1
ESCRIBIR EL TÍTULO DEL CAPÍTULO (NIVEL 1)	4
ESCRIBIR EL TÍTULO DEL CAPÍTULO (NIVEL 1)	1
ESCRIBIR EL TÍTULO DEL CAPÍTULO (NIVEL 1)	4
ESCRIBIR EL TÍTULO DEL CAPÍTULO (NIVEL 1)	1
ESCRIBIR EL TÍTULO DEL CAPÍTULO (NIVEL 1)	4

Índice de tablas

Resumen

Resumen

Palabras clave:

En esta memoria se va a describir el funcionamiento de la herramienta creada para poder gestionar con más facilidad los acuerdos de los estudiantes de la Universidad de Alcalá que van a realizar una estancia Erasmus

La herramienta consiste en una aplicación Java Web donde existen dos tipos de usuarios, con sus funciones propias: los alumnos que van a realizar una estancia Erasmus y los coordinadores (tutores) de estos alumnos.

Toda esta información es almacenada en una base de datos PostgreSQL, la cual se conecta con la aplicación a través del driver JDBC.

Abstract

Keywords:

This report will describe the functioning of the computer program created to be able to more easily manage the agreements of the University of Alcalá students who are going to carry out an Erasmus stay.

The computer program consists of a Java Web application where there are two types of users, with their own functions: the students who are going to carry out an Erasmus stay and the coordinators (tutors) of these students.

All this information is stored in a PostgreSQL database, which is connected to the application through the JDBC driver.

Resumen extendido

El proyecto realizado consiste en la creación de una herramienta para administrar de manera más eficiente los acuerdos de los estudiantes de la Universidad de Alcalá que van a llevar a cabo una estancia Erasmus, permitiendo al alumno más sencillez para gestionar las convalidaciones que desea realizar y al profesor una mayor facilidad de decisión sobre si estas convalidaciones son válidas.

Dicha herramienta consiste en una página web interna de la Universidad de Alcalá, cuyo servidor será propio de la Universidad. La página web se ha desarrollado en Java, utilizando el IDE NetBeans, en el cual se han tenido que instalar los plugins necesarios para la adecuada implementación de la aplicación. Además, para crear dicha web, se ha utilizado la tecnología JSP, permitiendo crear el contenido al que los usuarios dispondrán.

La información necesaria para el correcto funcionamiento de la web se obtiene o se manipula en una base de datos relacional diseñada con PostgreSQL. Para realizar la conexión entre la página web a la base de datos, se utiliza el driver JDBC (Java DataBase Connectivity).

Para poder comunicar al usuario con la aplicación, se ha utilizado el servidor web Apache Tomcat. Como se ha comentado anteriormente, la aplicación se va a alojar en un servidor físico dentro de la Universidad de Alcalá. Este servidor y el usuario se comunican a través del protocolo HTTP y, gracias a Apache, se podrán comunicar de forma segura y eficiente.

Una vez lanzada la aplicación para su función, tendrán acceso a ella los alumnos que hayan sido aceptados para realizar una estancia Erasmus y los tutores de dichos estudiantes. Es evidente que habrá dos tipos de usuarios: los alumnos y los profesores.

Los alumnos podrán dar de alta asignaturas, tanto de la Universidad de Alcalá como de la universidad de destino. Para dar de altas las asignaturas deberán introducir la información necesaria como los créditos, el nombre o el código de dicha asignatura. Con estas asignaturas, podrán dar de alta las convalidaciones que deseen realizar en una estancia determinada. También, podrán consultar el estado de las convalidaciones, es decir, si ha sido aceptada, rechazada, cancelada (decisión propia) o todavía está pendiente de confirmar. Además, podrán buscar las antiguas convalidaciones realizadas de las asignaturas que deseen para facilitar la elección de las asignaturas que compondrán el acuerdo.

Por otro lado, los profesores podrán ver la información de las convalidaciones que los alumnos deseen realizar seleccionando la estancia que ellos deseen y, a su vez, aceptarlas o rechazarlas, además de poder cancelarlas si un alumno rectifica en una convalidación que ha recibido la aceptación del tutor. Igualmente, podrán dar de altas a nuevos alumnos, los cuales han sido adjudicados al coordinador, al igual que otorgarles una estancia. También, podrán dar de alta grados de la Universidad de Alcalá y universidades de destino. Por último, podrá cerrar estancias una vez que el alumno ya haya vuelto de ella o, si el alumno decide renunciar a la estancia, establecerla como renunciada.

En este documento se describe en primer lugar el contexto, la motivación y los objetivos del proyecto. En esta parte se explica por qué es útil la creación de esta herramienta y cuál es el fin que se debe lograr una vez puesta en marcha.

Posteriormente, se lleva a cabo una extensa descripción del sistema y de los requisitos, dividiéndolos en funcionales y no funcionales. Asimismo, se muestran los casos de uso de los diferentes actores que formarán parte del sistema.

También, se describe la arquitectura software del sistema, incluyendo el diseño de la base de datos, estructura del proyecto, procedimientos realizados y diagramas de secuencia que explican las distintas funcionalidades.

Por último, se detalla el presupuesto del proyecto, las conclusiones obtenidas, el posible trabajo futuro y un manual de usuario que describe la actividad que puede llevar a cabo cada usuario utilizando la herramienta.

Introducción

Contexto

El programa Erasmus (European Community Action Scheme for the Mobility of University Students) es aquel que fomenta el intercambio de estudiantes entre universidades europeas, cuya duración puede tener entre 1 semestre a un curso completo como máximo. La Universidad de Alcalá participa en este programa y posee convenios con más de 50 países de todo el mundo.

Este programa tiene multitud de ventajas para los estudiantes, desde aprender nuevos idiomas hasta conocer otras culturas. Antes de iniciar la estancia, es necesario seleccionar las asignaturas que se desean cursar en la universidad de destino, las cuales serán utilizadas para ser convalidadas por asignaturas de la Universidad de Alcalá siempre que se cumplan una serie de requisitos. Para ello, se crea el documento Learning Agreement.

El Learning Agreement es el documento más importante que tiene que rellenar un alumno que ha recibido la confirmación de admisión de una universidad de destino para realizar una estancia Erasmus. Consiste en un acuerdo de aprendizaje, donde cada estudiante introducirá información acerca de las asignaturas que cursará en la universidad de destino y sus correspondientes asignaturas de la universidad de origen, las cuales serán solicitadas para su correcta convalidación.

Antes de poder completar el Learning Agreement, el alumno enviará a su tutor académico las asignaturas que desea convalidar, decidiendo este último si las convalidaciones son posibles y, si se rechazan, buscar otra posible convalidación. Cuando el tutor y el alumno se ponen de acuerdo, el alumno ya puede empezar a completar el Learning Agreement.


Una vez que el alumno complete todos los campos necesarios, su universidad de destino y su universidad de origen recibirán una copia de él.

Para poder disfrutar del Erasmus, el estudiante deberá obtener este acuerdo firmado por el coordinador Erasmus de su nueva universidad, acordando el plan de estudios que se recoge en el Learning Agreement.

Además, el Learning Agreement se puede modificar durante la estancia. El alumno puede modificar acuerdos ya aceptados por las razones que él considere, siempre y cuando lo delibere con el coordinador académico de la universidad de destino.

Finalmente, una vez que concluye el contrato de estudios, el alumno deberá entregarlo con todas las firmas necesarias en la Oficina de Relaciones Internacionales.

En la Fig.1 se muestra una parte del Learning Agreement que el alumno debe completar.


Erasmus+

Learning Agreement

Student Mobility for Studies

Higher Education:
Learning Agreement form
Student's name
Academic Year 20.../20...

Student	Last name(s)	First name(s)	Date of birth	Nationality ¹	Sex [M/F]	Study cycle ²	Field of education ³

Sending Institution	Name	Faculty/Department	Erasmus code ⁴ (if applicable)	Address	Country	Contact person name ⁵ ; email; phone

Receiving Institution	Name	Faculty/Department	Erasmus code (if applicable)	Address	Country	Contact person name; email; phone

Before the mobility

Table A Before the mobility	Study Programme at the Receiving Institution			
	Planned period of the mobility: from [month/year] 09/2018 to [month/year] 12/2018			
	Component ⁶ code (if any)	Component title at the Receiving Institution (as indicated in the course catalogue ⁷)	Semester [e.g. autumn/spring; term]	Number of ECTS credits (or equivalent) ⁸ to be awarded by the Receiving Institution upon successful completion
	Total:			
Web link to the course catalogue at the Receiving Institution describing the learning outcomes: https://weboodi.uef.fi/weboodi/				

The level of language competence ⁹ in ENGLISH [indicate here the main language of instruction] that the student already has or agrees to acquire by the start of the study period is: A1 <input type="checkbox"/> A2 <input type="checkbox"/> B1 <input checked="" type="checkbox"/> B2 <input type="checkbox"/> C1 <input type="checkbox"/> C2 <input type="checkbox"/> Native speaker <input type="checkbox"/>

Table B Before the mobility	Recognition at the Sending Institution			
	Component code (if any)	Component title at the Sending Institution (as indicated in the course catalogue)	Semester [e.g. autumn/spring; term]	Number of ECTS credits (or equivalent) to be recognised by the Sending Institution
	Total:			
	Provisions applying if the student does not complete successfully some educational components: https://www.uah.es/es/estudios/Grado-en-Sistemas-de-Informacion/			

Fig.1 Parte del Learning Agreement

Motivación

A priori, rellenar todo el contenido necesario en el Learning Agreement puede parecer algo simple y sencillo, pero no es así. Completar todo el documento puede llegar a ser algo verdaderamente costoso para el estudiante, incluso incrementaría su dificultad si recibe el rechazo de algún acuerdo.

El estudiante debe llevar a cabo una exhausta investigación, la cual trata de explorar la web de la universidad de destino para ver el temario de cada asignatura, hablar con otros estudiantes que han realizado la estancia allí o preguntar a su coordinador, quien es posible que no tenga soluciones.

En cuanto a la labor del tutor de la universidad de origen, puede ser incluso más costosa que la del alumno. Habitualmente, antes de recibir el Learning Agreement oficial, el profesor recibe por parte del alumno las propuestas de asignaturas tanto de origen como de destino. Estas se pueden recibir de diversas maneras: desde un documento Word bien estructurado, hasta un mensaje por correo electrónico. Sin embargo, lo más tedioso para el profesor consiste en recorrerse todas las webs, informaciones y guías docentes de cada asignatura de la universidad de destino y comparar su temario con las asignaturas de la universidad de origen. A esto hay que tener en cuenta que regularmente el profesor no recibe toda la información necesaria por parte del alumno, omitiéndose información relevante como número de créditos, página web de la asignatura de la universidad de destino, etcétera. Por lo tanto, esta herramienta es creada para evitar todos los inconvenientes comentados.

Respecto al alumno, dispondrá de una interfaz manejable y sencilla de entender para introducir las convalidaciones que él desee, además de poder observar las convalidaciones de cursos anteriores de las asignaturas va a solicitar convalidar.

En cuanto al profesor, tendrá la posibilidad de ver toda la información necesaria para decidir si una convalidación es aceptada o rechazada. Esta información consiste en los créditos de ambas asignaturas, un link que redirige a la web de cada asignatura, nombre, etcétera.

En conclusión, la motivación por la cual se crea esta herramienta es la de resolver todos los problemas que tienen tanto alumno como profesor para decidir cuáles serán las convalidaciones que se incluirán en el Learning Agreement definitivo.

Objetivos

El objetivo principal de la implementación de esta herramienta es facilitar a los alumnos Erasmus la elección de asignaturas que desean convalidar y que su tutor académico pueda decidir si acepta la convalidación con más rapidez. Gracias a la elaboración de esta herramienta, se agilizará el proceso previo a la realización del Learning Agreement.

Además, se proponen los siguientes objetivos secundarios que complementará al objetivo principal:

- Desarrollar una aplicación web a la que se pueda acceder desde cualquier navegador.
- Desplegar una interfaz gráfica intuitiva para el usuario que sea simple de manejar.
- Diseñar un sistema gracias al estudio de herramientas como Java, NetBeans, JSP, PostgreSQL o HTML, además de seguir técnicas de diseño de programación orientada a objetos.
- Estudiar el funcionamiento de algunos patrones de diseño como el patrón DAO o el patrón Modelo Vista Controlador.
- Elaborar un manual de usuario sobre el funcionamiento de la herramienta.
- Redactar esta memoria, la cual será muy importante debido a posibles mejoras futuras de la herramienta, así como posibles estudios de los datos almacenados.

Análisis y Especificación de requisitos

Descripción del sistema

La aplicación web desarrollada incluye las funcionalidades necesarias para que los usuarios, tanto profesores como alumnos puedan realizar las operaciones que ellos deseen. La herramienta debe distinguir si el usuario que inicia sesión es alumno o profesor. Una vez que lo distingue, accede a la interfaz correspondiente.

En el caso del alumno, muestra una pantalla con diferentes opciones, de las cuales él podrá elegir una. Después, se abrirá una nueva página con la interfaz de la opción que él ha elegido. Estas opciones pueden ser insertar nueva asignatura de origen o de destino, realizar una nueva convalidación, buscar convalidaciones antiguas, ver el estado de las convalidaciones actuales o cambiar la contraseña.

Respecto al profesor, esto se amplía. Al iniciar sesión un usuario profesor, aparecerá una página con varios aspectos: por un lado, enlaces para realizar diversas operaciones en las que se incluyen dar de alta a alumnos, profesores, estancias de alumnos, asignaturas universidades y grados, cerrar estancias o renunciar a ellas (con petición previa del alumno) y cambiar la contraseña. Por otro lado, un desplegable de opciones con las estancias de los alumnos que tiene asignados, tanto actuales como pasadas. Al seleccionar una opción del desplegable se completa la tabla de la parte inferior con la información de las convalidaciones que componen esa estancia. Un poco más abajo, se encuentran las opciones de modificar asignatura (aceptar o rechazar la convalidación) y cancelar asignatura para las convalidaciones aceptadas que el alumno desea cancelar. En la parte derecha aparece información del alumno seleccionado.

Catálogo de requisitos

Tras explicar la descripción del sistema, llegamos al turno de listar los requisitos necesarios para el correcto comportamiento del sistema.

Requisitos funcionales

Tabla.1 Lista de requisitos funcionales

IDENTIFICADOR	DESCRIPCIÓN	PRIORIDAD
RF-01	El sistema debe solicitar al usuario que inicie sesión para acceder a las funcionalidades	Muy alta
RF-02	El sistema comprobará que el usuario y la contraseña coinciden con lo registrado en la base de datos	Muy alta
RF-03	El sistema proporcionará distintas vistas según el usuario que haya iniciado sesión	Muy alta
RF-04	El sistema registrará toda la información que se desee añadir en la base de datos	Muy alta
RF-05	Ambos usuarios podrán agregar asignaturas tanto de origen como de destino a ambos usuarios	Muy alta
RF-06	Ambos usuarios podrán cambiar su contraseña	Alta
RF-07	El alumno podrá añadir una nueva convalidación a la estancia que él considere	Muy alta
RF-08	El alumno podrá buscar antiguas convalidaciones de la asignatura de origen que él desee introduciendo el código de dicha asignatura	Muy alta
RF-09	El alumno podrá contemplar el estado de las convalidaciones que ha solicitado de la estancia que él desee	Muy alta
RF-10	El coordinador podrá dar de alta a un nuevo alumno	Muy alta
RF-11	El coordinador podrá dar de alta una nueva estancia para un alumno registrado	Muy alta
RF-12	El coordinador podrá dar de alta a un nuevo coordinador	Alta
RF-13	El coordinador podrá dar de alta nuevas universidades de destino	Muy alta

IDENTIFICADOR	DESCRIPCIÓN	PRIORIDAD
RF-14	El coordinador podrá dar de alta nuevos grados de la UAH	Muy alta
RF-15	El coordinador podrá establecer como renunciada a una estancia de un alumno	Alta
RF-16	El coordinador podrá cerrar una estancia de un alumno que ya ha terminado su estancia Erasmus	Muy alta
RF-17	El coordinador podrá contemplar la información de las convalidaciones de la estancia de un alumno que él desee	Muy alta
RF-18	El coordinador podrá aceptar o rechazar una convalidación según él crea oportuno	Muy alta
RF-19	El coordinador podrá realizar comentarios acerca de por qué acepta o rechaza una convalidación	Alta
RF-20	El coordinador podrá cancelar una convalidación a petición del alumno	Muy alta
RF-21	El coordinador podrá realizar comentarios acerca de por qué cancela una convalidación	Alta

Requisitos no funcionales

Tabla.2 Lista de requisitos no funcionales

IDENTIFICADOR	DESCRIPCIÓN	PRIORIDAD
RNF-01	Los nombres y contraseñas de usuarios serán únicos	Muy alta
RNF-02	Se almacenará toda la información necesaria para el correcto funcionamiento del sistema en una base de datos PostgreSQL	Muy alta
RNF-03	Se desarrollará el proyecto utilizando el lenguaje de programación Java	Muy alta
RNF-04	Se utilizará la tecnología JSP para la creación de la página web	Muy alta
RNF-05	El proyecto se desarrolla en el entorno de desarrollo NetBeans IDE	Alta
RNF-06	La aplicación se desarrolla siguiendo unos patrones software	Alta
RNF-07	El sistema deberá ser capaz de operar adecuadamente con varios usuarios de manera concurrente	Muy alta
RNF-08	El sistema estará disponible a cualquier hora del día	Alta
RNF-09	El sistema será accesible desde dentro del servidor de la Universidad de Alcalá	Muy alta
RNF-10	El sistema será accesible a través de un ordenador de forma online	Muy alta
RNF-11	El sistema proporcionará mensajes de confirmación o error orientativos a los usuarios	Alta
RNF-12	El sistema será escalable, por lo que permitirá la implementación de nuevas funcionalidades	Alta
RNF-13	Se entregará un manual de usuario que detallará el uso de la aplicación	Muy alta

Casos de uso

Una vez establecida la especificación de requisitos, pasamos al estudio de los casos de uso del sistema, con el fin de admirar los diferentes puntos de vista del sistema.

Habr  dos tipos de actores: el coordinador y el alumno. Cada uno realizar  unas funciones distintas que se ven reflejadas en los siguientes diagramas.

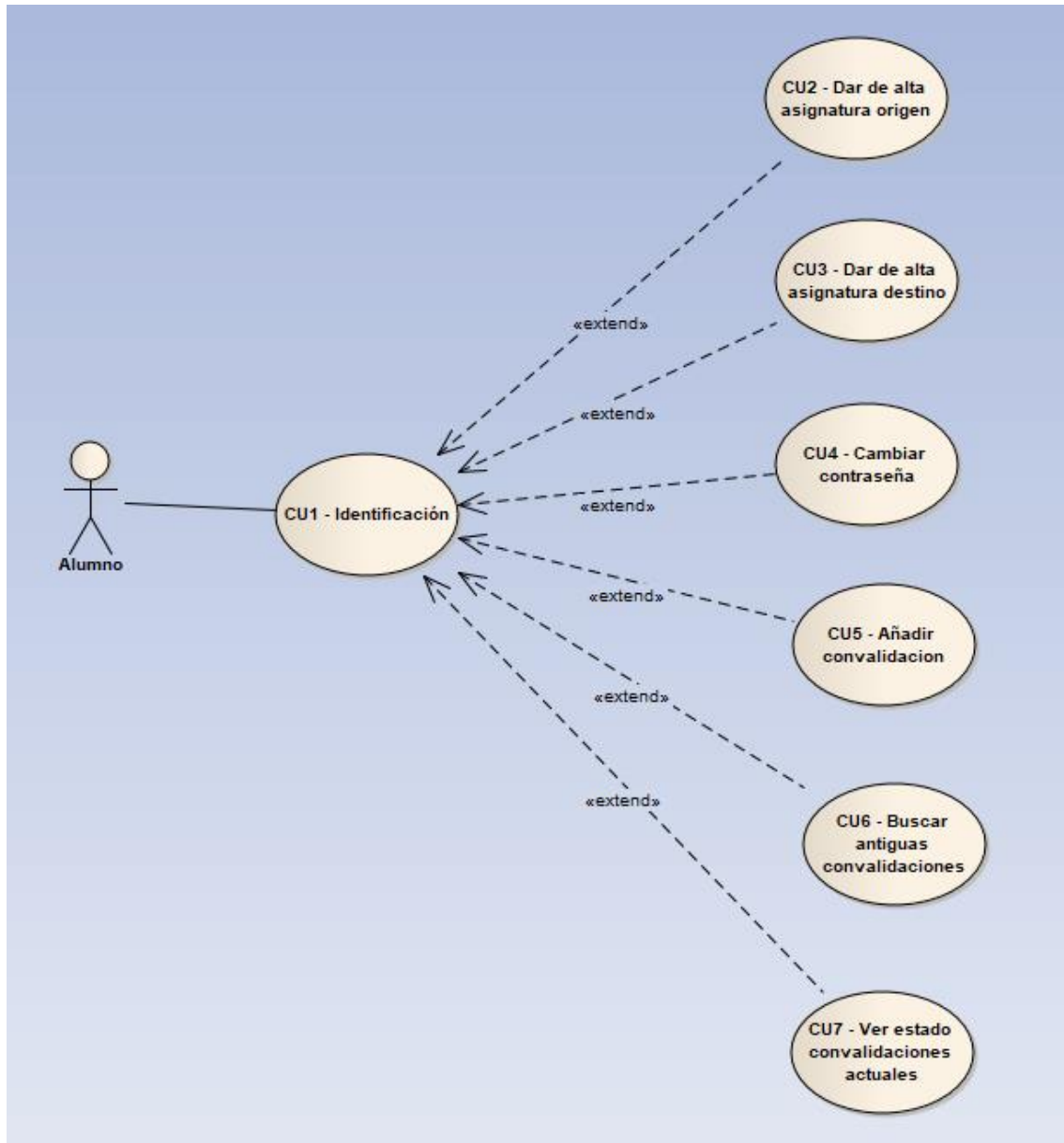


Fig.2 Diagrama de casos de uso del actor alumno

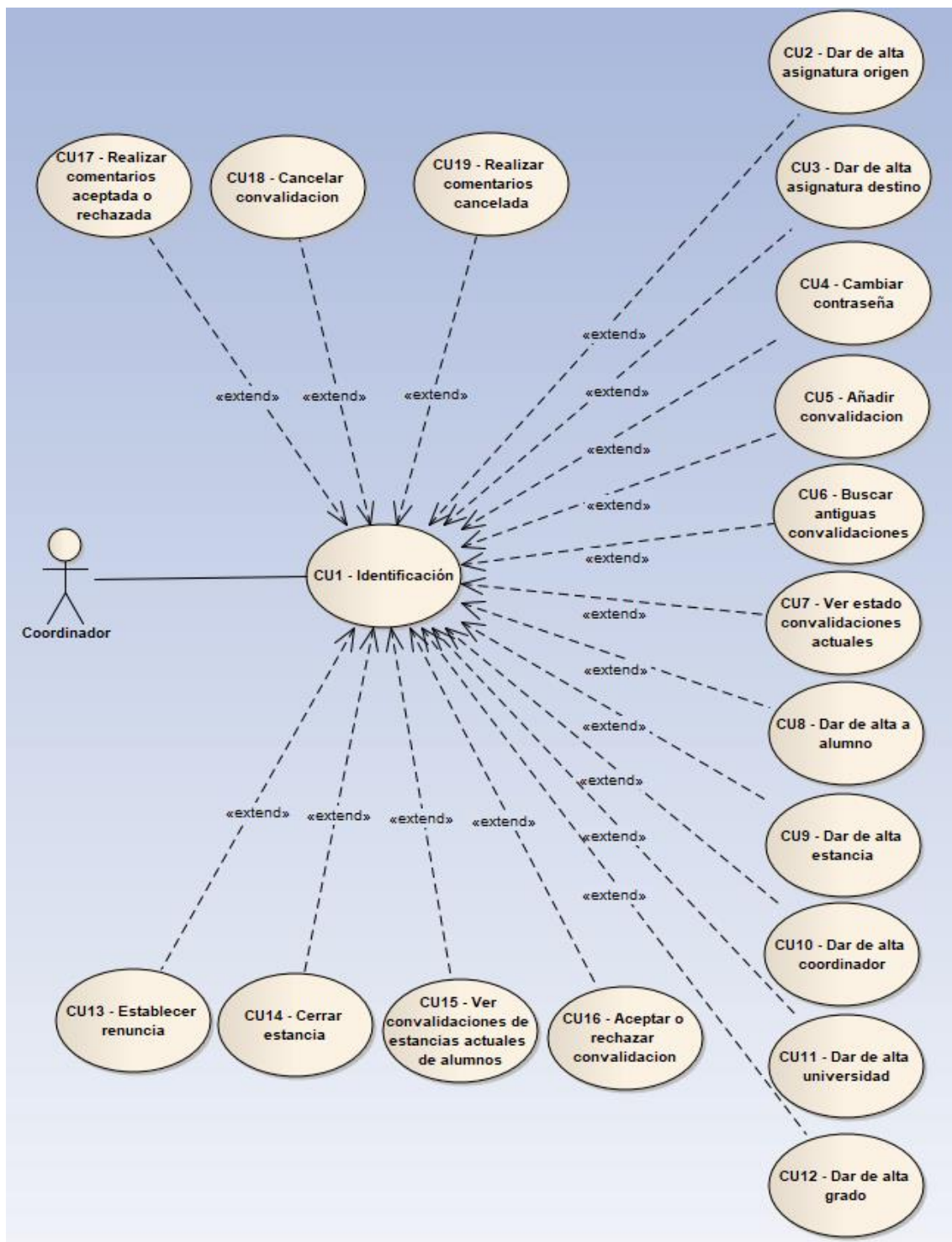


Fig.3 Diagrama de casos de uso del actor coordinador

CU1 – Identificación

Descripción: caso de uso que representa el acceso a las funcionalidades del sistema por los dos tipos de usuario: alumno y profesor.

Requisitos: RF01, RF02 y RF03

CU2 – Dar de alta asignatura origen

Descripción: caso de uso que representa la inserción por parte de un alumno o de un profesor de una asignatura que se cursa en la Universidad de Alcalá. Para ello, habrá que introducir el código, el nombre, el número de créditos, el tipo (básica, obligatoria, optativa o transversal) y la información (la página web de la asignatura).

Requisitos: RF-04 y RF-05

CU3 – Dar de alta asignatura destino

Descripción: caso de uso que representa la inserción por parte de un alumno o de un profesor de una asignatura que se desea cursar en la Universidad donde el alumno ha sido admitido. Para ello, habrá que introducir el código, el nombre, el número de créditos, la universidad donde se imparte y la información (la página web de la asignatura).

Requisitos: RF-04 Y RF-05

CU4 – Cambiar contraseña

Descripción: caso de uso que representa la posibilidad por parte de un alumno o un profesor de cambiar su contraseña. El usuario deberá introducir la contraseña antigua y la nueva dos veces para su correcta modificación.

Requisitos: RF-04 y RF-06

CU5 – Añadir convalidación

Descripción: caso de uso que representa la inserción por parte del alumno de una convalidación de una estancia determinada. El alumno deberá elegir la estancia en la que quiere realizar la convalidación (por si tiene más de una sin cerrar), la asignatura de origen (será un desplegable con todas las asignaturas registradas de la Universidad de Alcalá) y la asignatura de destino (será un desplegable con todas las asignaturas de las universidades de las estancias que tiene en curso).

Requisitos: RF-04 y RF-07

CU6 – Buscar antiguas convalidaciones

Descripción: caso de uso que representa la posibilidad por parte del alumno de buscar antiguas convalidaciones de la asignatura de origen que él desee. El alumno deberá introducir el código de la asignatura de origen y le aparecerá toda la información de las convalidaciones aceptadas realizadas con dicha asignatura. Esta información recoge, por parte de la asignatura de origen, el código, nombre, tipo, créditos y un link que redirige a la página de la asignatura y, en la parte derecha por parte de la asignatura de destino, el código, nombre, créditos, un link que redirige a la página de la asignatura, universidad, fecha de convalidación, curso académico, estado (siempre aceptada) y comentarios.

Requisitos: RF-08

CU7 – Ver estado convalidaciones actuales

Descripción: caso de uso que representa la posibilidad por parte del alumno de poder observar el estado de las convalidaciones actuales que desea realizar. El alumno elegirá la estancia que él desee comprobar y le aparecerá la información. Esta información recoge, por parte de la asignatura de origen, el código, nombre, tipo, créditos y un link que redirige a la página de la asignatura y, en la parte derecha por parte de la asignatura de destino, el código, nombre, créditos, un link que redirige a la página de la asignatura, universidad, fecha de convalidación, curso académico, estado (siempre aceptada) y comentarios. Además, aparecen al final los créditos totales aceptados en la Universidad de Alcalá y de la Universidad de destino.

Requisitos: RF-09

CU8 – Dar de alta a alumno

Descripción: caso de uso que representa la inserción por parte del profesor de un alumno que ha sido asignado a dicho profesor. Para darlo de alta en el sistema habrá que introducir su nombre, apellidos, contraseña, nombre de usuario, DNI y grado que cursa en la Universidad de Alcalá.

Requisitos: RF-04 y RF-10

CU9 – Dar de alta estancia

Descripción: caso de uso que representa la inserción por parte del profesor de una estancia de un alumno. El profesor tendrá que seleccionar el alumno, el tipo de estancia, curso académico, duración y la universidad de destino.

Requisitos: RF-04 y RF-11

CU10 – Dar de alta coordinador

Descripción: caso de uso que representa la inserción de un profesor por parte de otro profesor. Se tendrán que introducir los campos nombre, apellidos, contraseña, nombre de usuario, DNI y departamento.

Requisitos: RF-04 y RF-12

CU11 – Dar de alta universidad

Descripción: caso de uso que representa la inserción por parte de un profesor de una universidad posible de destino. Se tendrán que introducir los campos, código de universidad, nombre, país y ciudad.

Requisitos: RF-04 y RF-13

CU12 – Dar de alta grado

Descripción: caso de uso que representa la inserción por parte de un profesor de un grado en la Universidad de Alcalá. Se tendrán que introducir los campos, código de grado, nombre y facultad.

Requisitos: RF-04 y RF-14

CU13 – Establecer renuncia

Descripción: caso de uso que representa la renuncia de una estancia por parte del alumno. Dicho alumno deberá comunicar al profesor previamente su deseo de renuncia y, una vez confirmada su renuncia, el profesor lo introducirá en el sistema.

Requisitos: RF-01 Y RF-15

CU14 – Cerrar estancia

Descripción: caso de uso que representa el cierre de una estancia. La mayoría de veces, este cierre se confirmará cuando el alumno vuelva tras realizar su estancia con éxito. Una vez confirmado su cierre, el profesor lo introducirá en el sistema.

Requisitos: RF-04 Y RF-16

CU15 – Ver convalidaciones de estancias actuales de alumnos

Descripción: caso de uso que representa la posibilidad de que el profesor vea la información de las convalidaciones de una estancia que un alumno desea realizar. Esta información recoge, por parte de la asignatura de origen, el código, nombre, tipo, créditos y un link que redirige a la página de la asignatura y, en la parte derecha por parte de la asignatura de destino, el código, nombre, créditos, un link que redirige a la página de la asignatura, universidad, fecha de convalidación, curso académico, estado (siempre aceptada) y comentarios. Además, aparecen al final los créditos totales aceptados en la Universidad de Alcalá y de la Universidad de destino.

Requisitos: RF-17

CU16 – Aceptar o rechazar convalidación

Descripción: caso de uso que representa la posibilidad de que el profesor, tras revisar la información de la convalidación, acepte o rechace la convalidación. Si es aceptada, podrá formar parte del Learning Agreement del alumno.

Requisitos: RF-04 y RF-18

CU17 – Realizar comentarios aceptada o rechazada

Descripción: caso de uso que representa la posibilidad de que el profesor, tras aceptar o rechazar una convalidación, redacte el motivo de su decisión.

Requisitos: RF-04 y RF-19

CU18 – Cancelar convalidación

Descripción: caso de uso que representa la posibilidad de que el profesor pueda cancelar una convalidación que había sido previamente aceptada. Esto se realizará cuando un alumno lo desee y se lo comunique al profesor

Requisitos: RF-04 y RF-20

CU19 – Realizar comentarios cancelada

Descripción: caso de uso que representa la posibilidad de que el profesor, tras cancelar una convalidación, redacte el motivo de su decisión.

Requisitos: RF-04 y RF-21

Diseño

A continuación, se va a explicar la arquitectura software que va a tener el sistema, desde el diseño de la base de datos hasta el modelado de clases del proyecto.

Modelado de datos

Para el modelado de datos, se ha optado por la opción de una base de datos relacional, debido a que facilitará las relaciones entre las diferentes asignaciones, como por ejemplo los alumnos asignados a un profesor, las estancias a los alumnos, las convalidaciones a las estancias, etc. Esto permitirá que los datos almacenados sean más accesibles independientemente de la acción que quiera realizar el usuario que esté utilizando la aplicación.

Para el proceso de creación de la base de datos que se va a utilizar, primero hay que crear el modelo entidad relación, compuesto por las tablas necesarias con sus campos, claves, claves foráneas y relaciones entre ellas, teniendo en cuenta la cardinalidad. Para la creación del modelo entidad relación se ha utilizado la herramienta PgModeler, ya que permite, una vez creado el modelo, exportar el modelo a código PostgreSQL, el cual se utilizará para la crear la base de datos en el servidor.

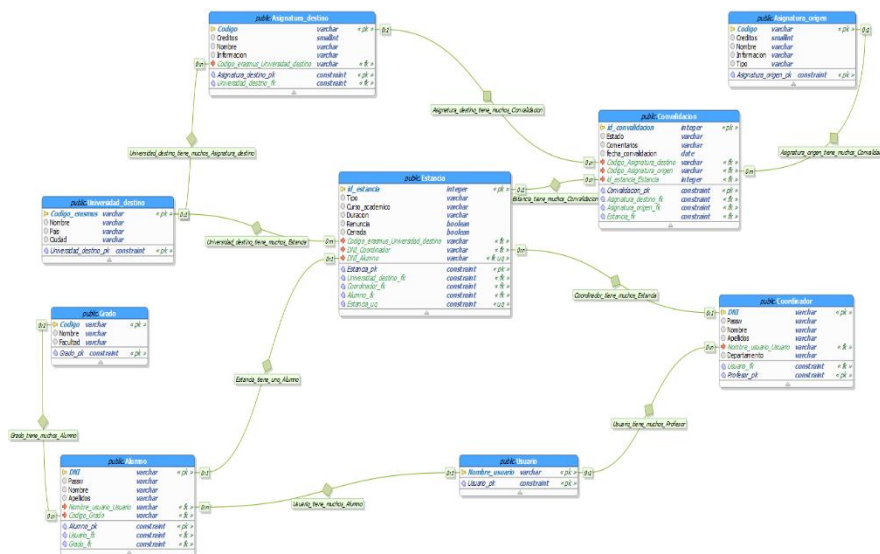


Fig.4 Diagrama entidad relación del sistema

Modelado de clases

La estructura de las clases de la aplicación estará basada en el patrón de arquitectura Modelo Vista Controlador para dividir la aplicación en módulos.

La arquitectura de la aplicación es la siguiente:

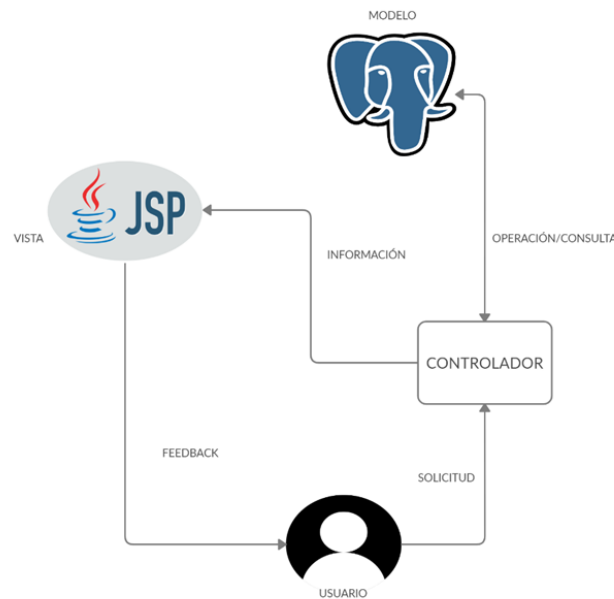


Fig.5 Esquema de la aplicación

El primer módulo, denominado modelo, será la lógica del programa donde se almacenará la base de datos. El modelo será el encargado de interactuar con la base de datos realizando operaciones con ella, tanto de lectura y escritura. Las clases que constituyen el modelo son las siguientes:

- Clases con el mismo nombre que las tablas de la base de datos (Alumno, Coordinador, Estancia, Convalidación...). Estas clases tendrán los mismos atributos que las de la base de datos con su correspondiente constructor y métodos getter y setter.
- Una clase adicional denominada RelacionAsignaturas, la cual se utiliza para mostrar en diferentes tablas información sobre asignaturas de origen y de destino de una misma convalidación.
- Las clases DAO de cada una de las anteriores (AlumnoDAO, CoordinadorDAO, RelacionAsignaturasDAO...), las cuales sirven para interactuar directamente con la base de datos utilizando los métodos de las clases anteriores.

El siguiente módulo será la interfaz donde el usuario podrá interactuar con la aplicación, este módulo se denomina vista. La vista se encarga de recibir los datos del modelo y mostrárselos al usuario. Las clases que componen la vista son de tipo JSP, las cuales se comunican con el controlador para realizar las acciones necesarias que el usuario

requiere en cada momento. En estas clases se crean toda la estructura de la vista, como cuadros de texto necesarios, desplegados, tablas, botones, enlaces, etc, a través de etiquetas HTML y se dan formato a través de estilos CSS con formato Bootstrap.

El último será el módulo de las comunicaciones donde el usuario podrá comunicarse con la base de datos. Este módulo, llamado controlador, recibe los eventos de entrada, como puede ser una cadena de un campo de texto que inserte algún registro en la base de datos. La clase que representa al controlador se denomina PropuestaLA y va recibiendo continuamente peticiones de la vista a través de los métodos doGet y doPost. En el momento que recibe estas peticiones, realiza la acción requerida y responde a la vista con el resultado de aplicar la acción en ese preciso momento. Por ejemplo, en el caso de la vista login.jsp, recibe como entrada por parte del usuario, un nombre de usuario y su contraseña. Se envía al controlador y se ejecuta la acción “comprobarlogin” donde comprueba si los campos introducidos pertenecen a un usuario profesor, un alumno o son incorrectos. En los dos primeros casos abre la vista correspondiente a cada uno y, en el último, muestra una ventana de error que indica que las credenciales introducidas son incorrectas.

Por último, existirá una clase Conexión donde se realizará la conexión con la base de datos. Esta clase se instanciará en las clases DAO del modelo para que puedan interactuar con la base de datos una vez realizada la conexión.

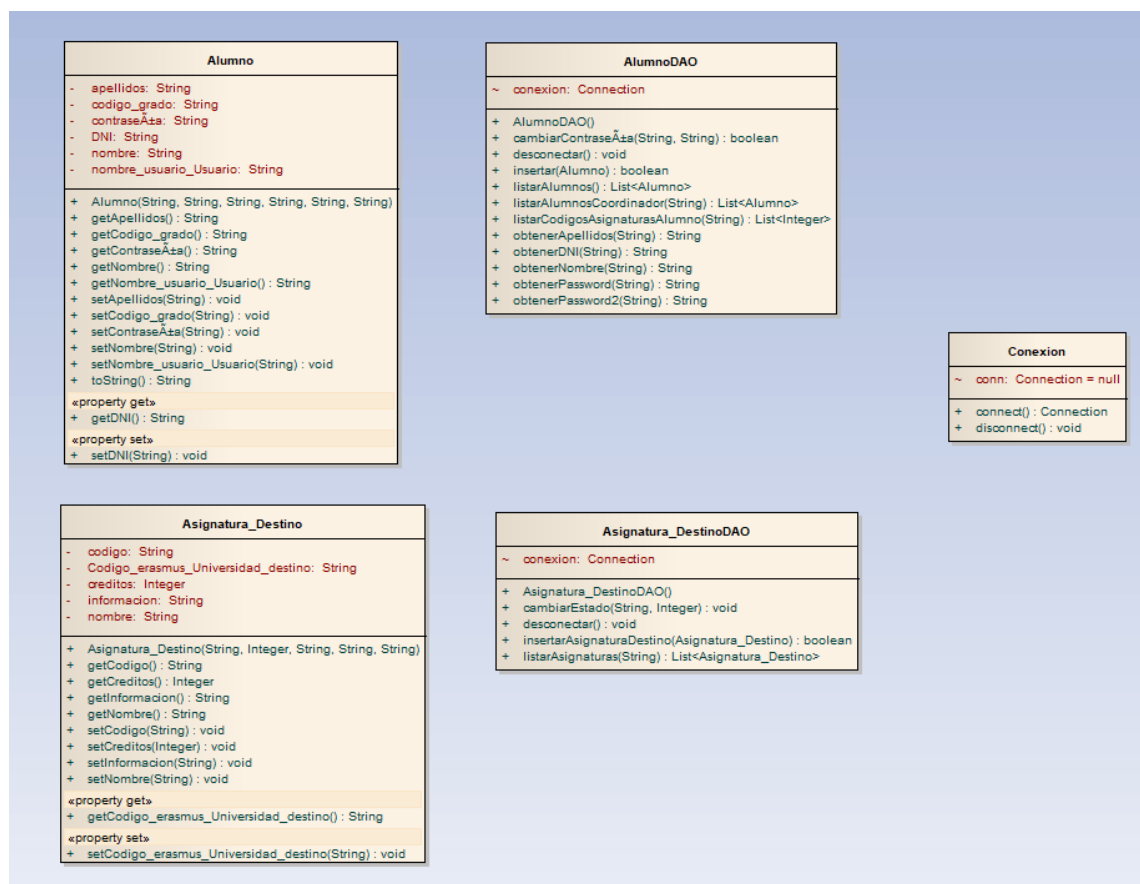


Fig.6 Algunas clases del modelo y la clase Conexión

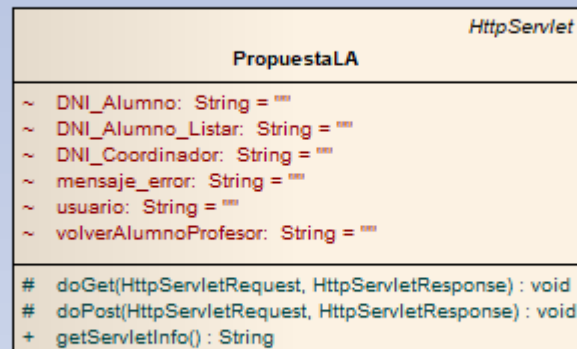


Fig.7 Clase controlador

Diagramas de secuencia

Login

Este diagrama de secuencia consiste en el acceso de un usuario al sistema. El usuario se encontrará un formulario para introducir sus credenciales (usuario y contraseña). Si estas existen en el sistema, comprueba si es un usuario Coordinador y, si lo es, accede al sistema como Coordinador. Si no, comprueba si es Alumno y accede al sistema como Alumno. En caso contrario, devuelve error.

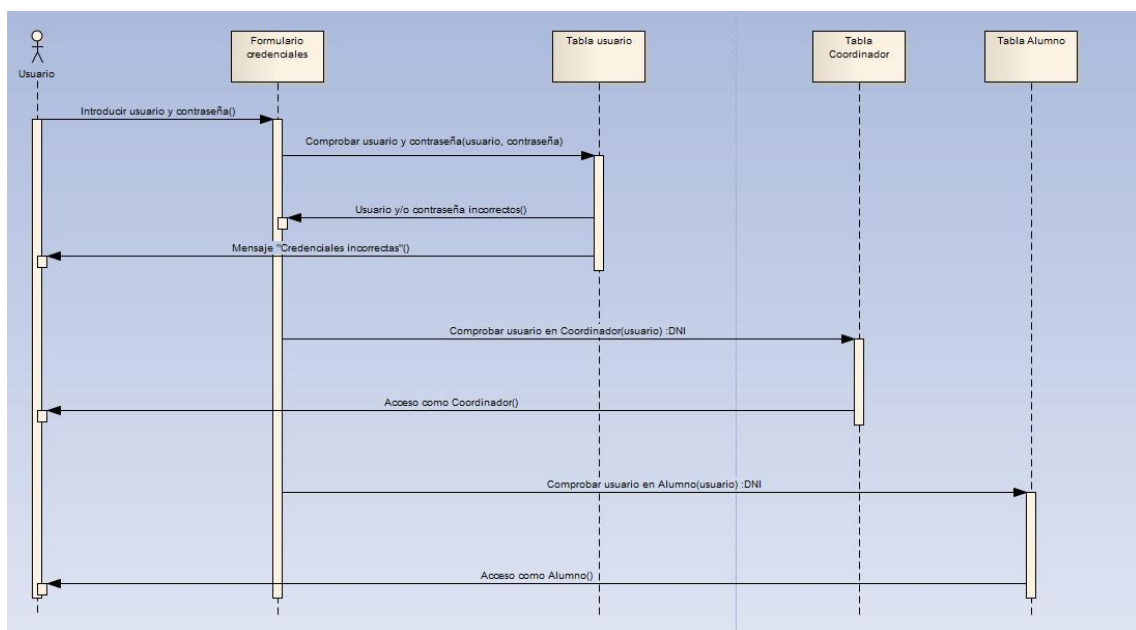


Fig.8 Diagrama de secuencia del login

Nuevo Alumno

Este diagrama de secuencia consiste en dar de alta a un alumno en el sistema. El Coordinador introduce los datos del alumno y, si son correctos, se inserta un nuevo Alumno y lo confirma con un mensaje. En caso contrario, muestra un mensaje de error.

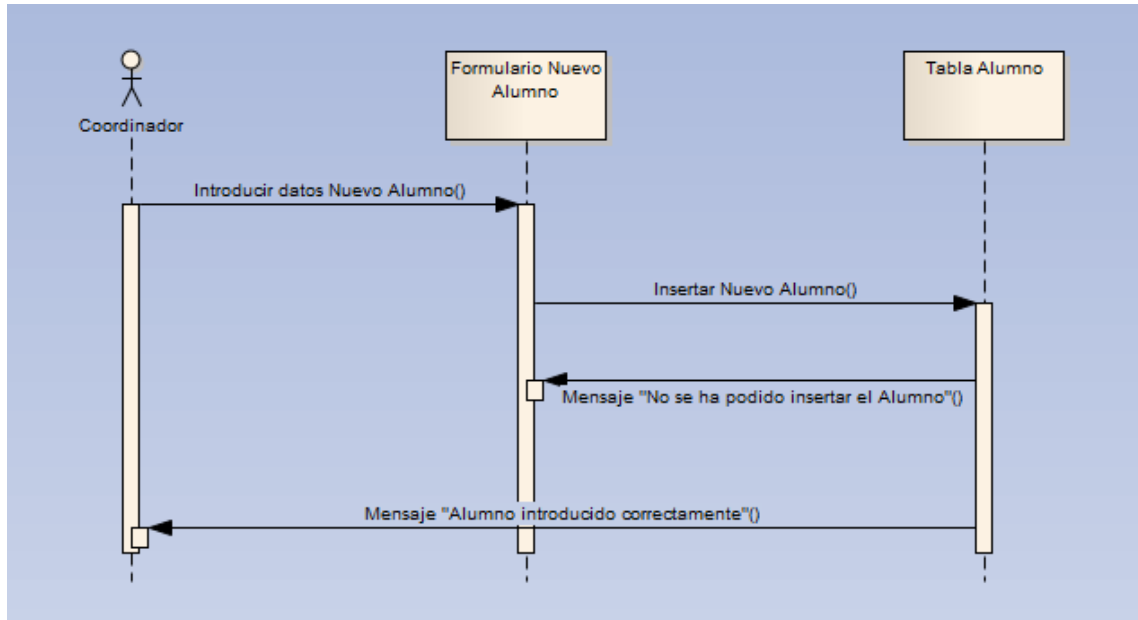


Fig.9 Diagrama de secuencia de dar de alta a alumno

Nueva Estancia

Este diagrama de secuencia consiste en dar de alta a una estancia en el sistema. El Coordinador introduce los datos de la estancia y, si son correctos, se inserta una nueva Estancia y lo confirma con un mensaje. En caso contrario, muestra un mensaje de error.

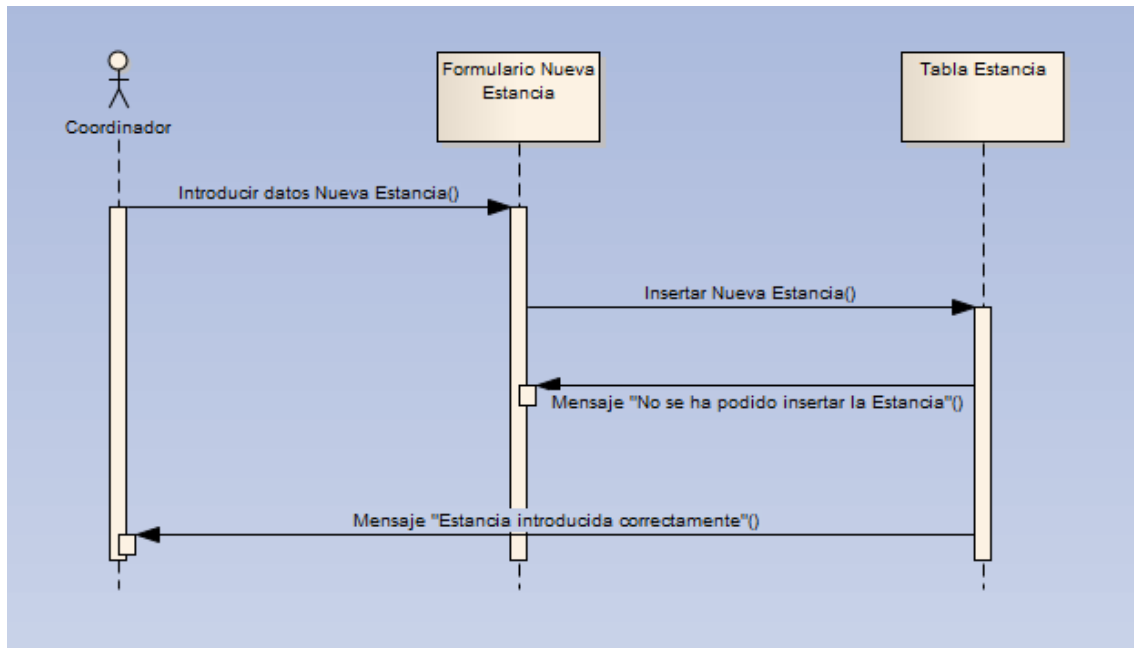


Fig.10 Diagrama de secuencia de dar de alta una estancia

Nueva Universidad

Este diagrama de secuencia consiste en dar de alta a una Universidad en el sistema. El Coordinador introduce los datos de la Universidad y, si son correctos, se inserta una nueva Universidad y lo confirma con un mensaje. En caso contrario, muestra un mensaje de error.

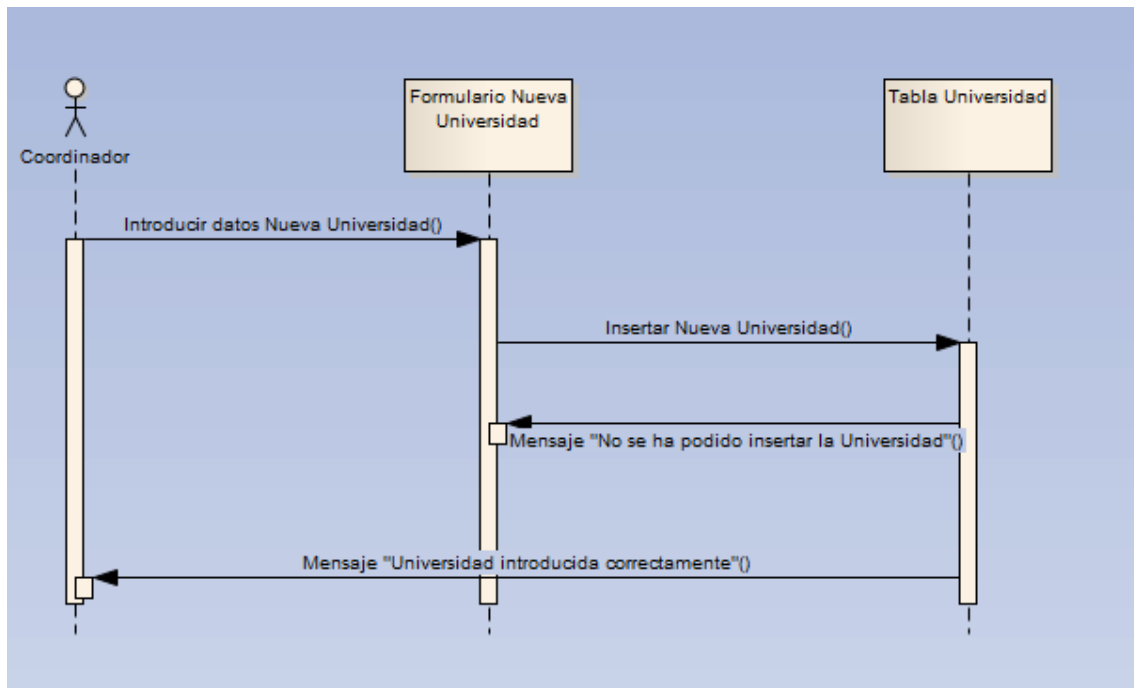


Fig.11 Diagrama de secuencia de dar de alta una universidad

Nueva Asignatura Origen

Este diagrama de secuencia consiste en dar de alta a una asignatura origen en el sistema. El usuario introduce los datos de la asignatura origen y, si son correctos, se inserta una nueva asignatura origen y lo confirma con un mensaje. En caso contrario, muestra un mensaje de error.

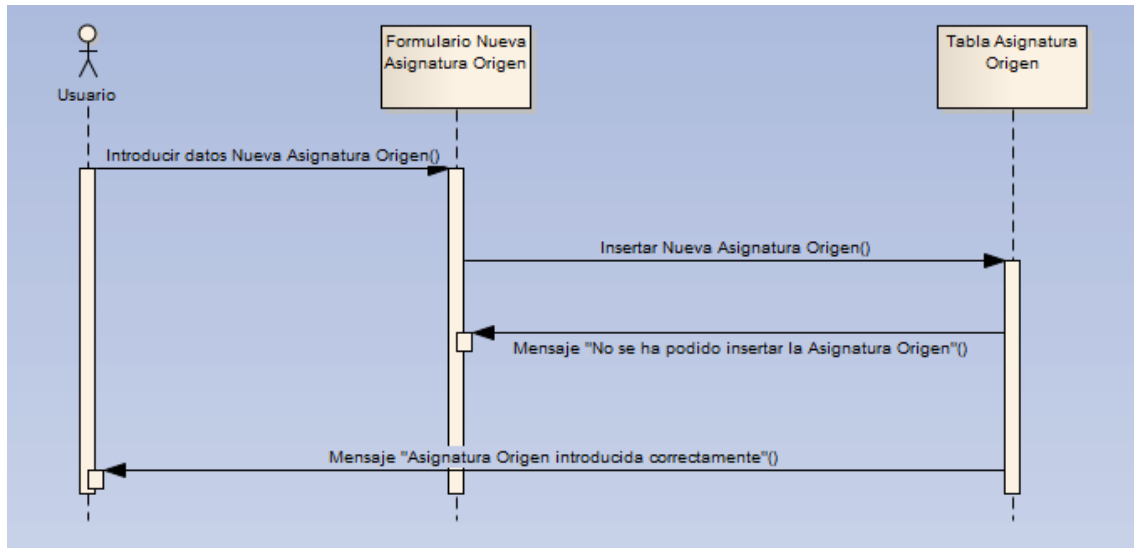


Fig.12 Diagrama de secuencia de dar de alta una nueva asignatura origen

Nueva Asignatura Destino

Este diagrama de secuencia consiste en dar de alta a una asignatura destino en el sistema. El usuario introduce los datos de la asignatura destino y, si son correctos, se inserta una nueva asignatura destino y lo confirma con un mensaje. En caso contrario, muestra un mensaje de error.

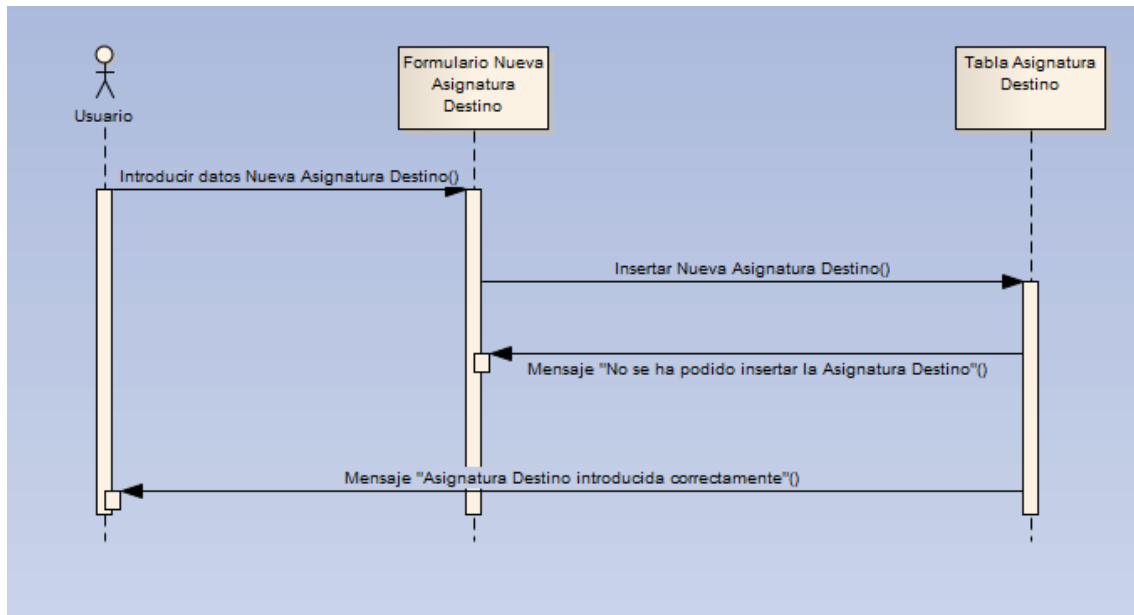


Fig.13 Diagrama de secuencia de dar de alta una nueva asignatura destino

Nuevo Grado

Este diagrama de secuencia consiste en dar de alta a un grado en el sistema. El Coordinador introduce los datos del grado y, si son correctos, se inserta un nuevo grado y lo confirma con un mensaje. En caso contrario, muestra un mensaje de error.

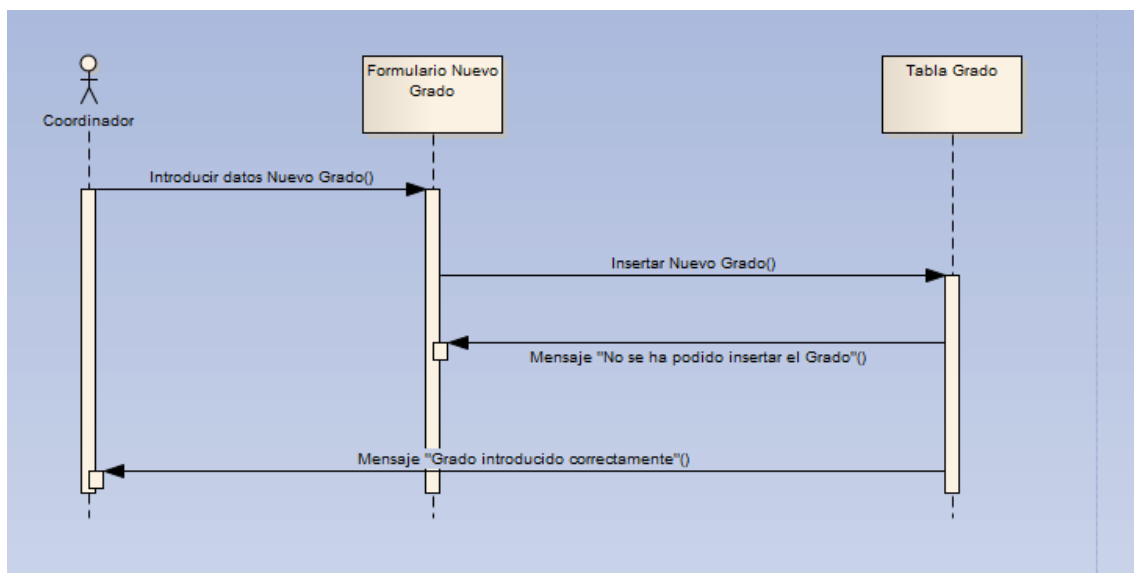


Fig.14 Diagrama de secuencia de dar de alta una un nuevo grado

Nuevo Profesor

Este diagrama de secuencia consiste en dar de alta a un profesor en el sistema. El Coordinador introduce los datos del profesor y, si son correctos, se inserta un nuevo profesor y lo confirma con un mensaje. En caso contrario, muestra un mensaje de error.

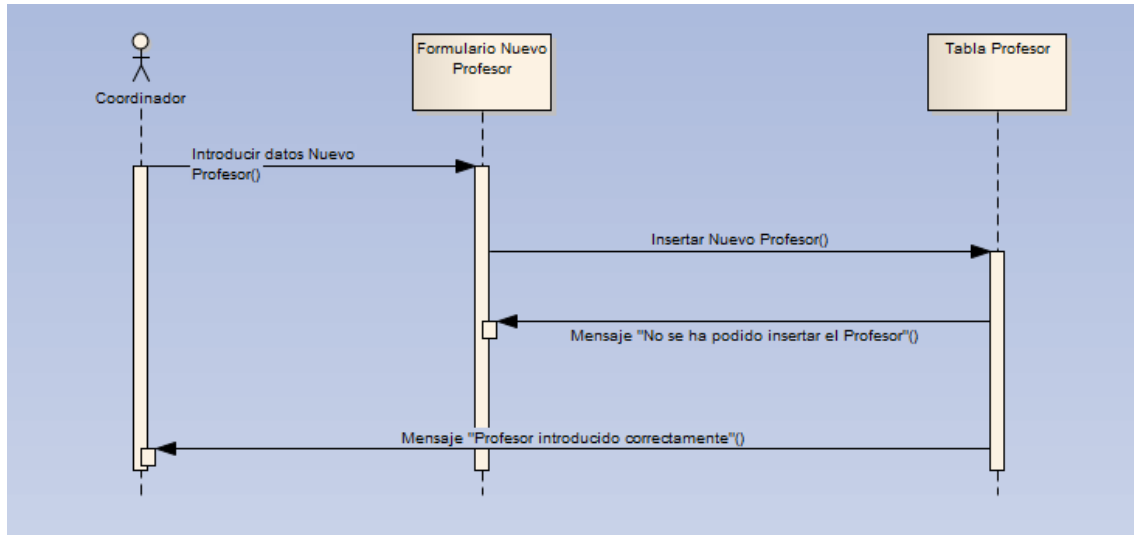


Fig.15 Diagrama de secuencia de dar de alta una nueva asignatura origen

Renuncia Estancia

Este diagrama de secuencia consiste en renunciar una estancia. El coordinador selecciona la estancia que va a ser renunciada por el alumno. Si todo es correcto muestra un mensaje de confirmación y si no lo es muestra un mensaje de error.

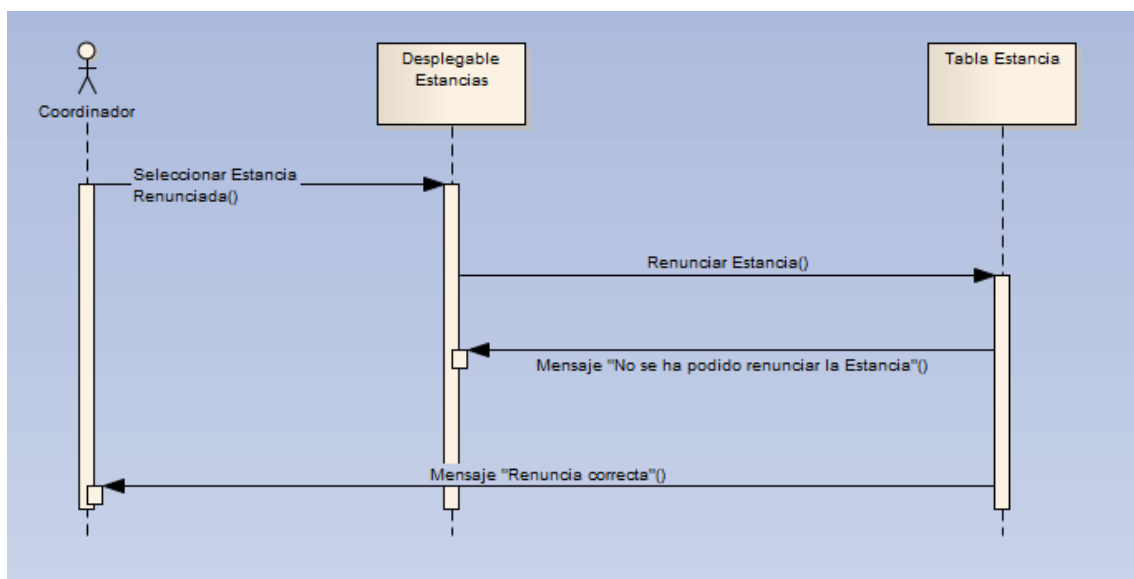


Fig.16 Diagrama de secuencia de renunciar una estancia

Cerrar Estancia

Este diagrama de secuencia consiste en cerrar una estancia. El coordinador selecciona la estancia que va a ser cerrada. Si todo es correcto muestra un mensaje de confirmación y si no lo es muestra un mensaje de error.

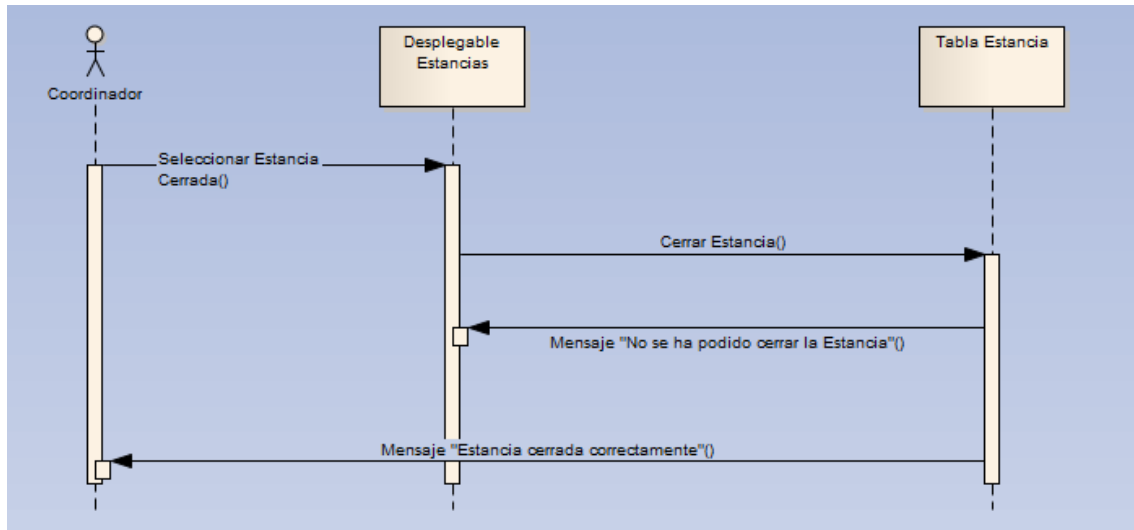


Fig.16 Diagrama de secuencia de cerrar una estancia

Cambiar contraseña

Este diagrama de secuencia consiste en cambiar la contraseña del usuario. El coordinador introduce la contraseña actual y la nueva contraseña. Si todo es correcto, la contraseña se cambia correctamente, si no, devuelve un error

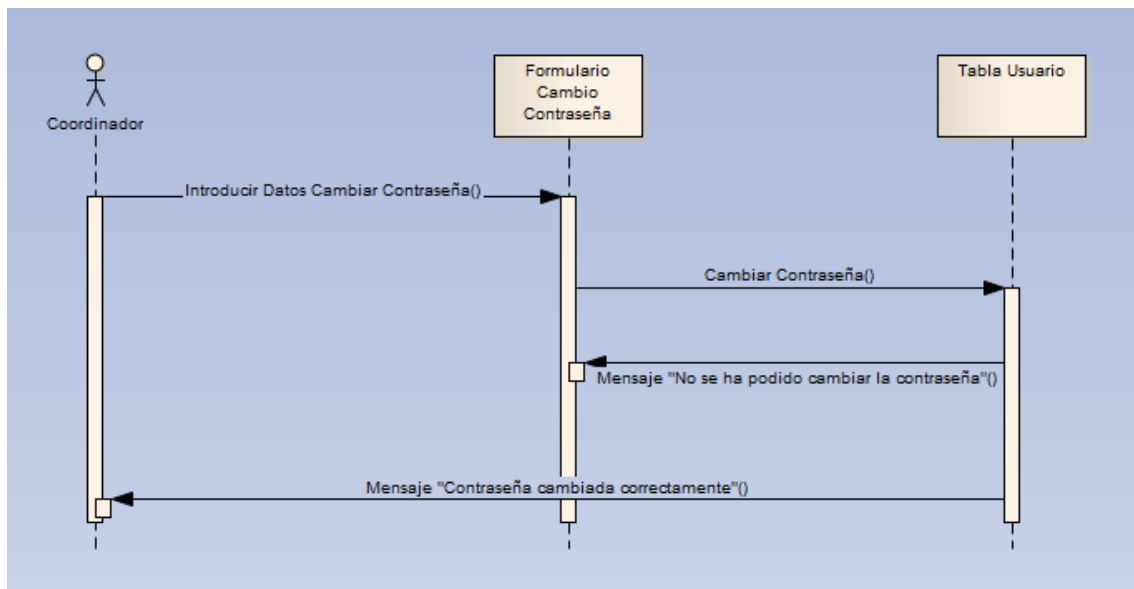


Fig.17 Diagrama de secuencia de cambiar una contraseña

Modificar convalidación

Este diagrama de secuencia consiste en modificar una convalidación, ya sea aceptarla, rechazarla o cancelarla. El coordinador selecciona la estancia que quiere contemplar. Una vez seleccionada, muestra una tabla con los datos de las convalidaciones y asignaturas de dicha estancia. A partir de ahora, el coordinador tiene toda la información necesaria para poder aceptar, rechazar o cancelar. Aparecerán dos desplegables, uno con las asignaturas pendientes, las cuales se podrán aceptar o rechazar y otro con la asignaturas aceptadas las cuales se podrán cancelar.

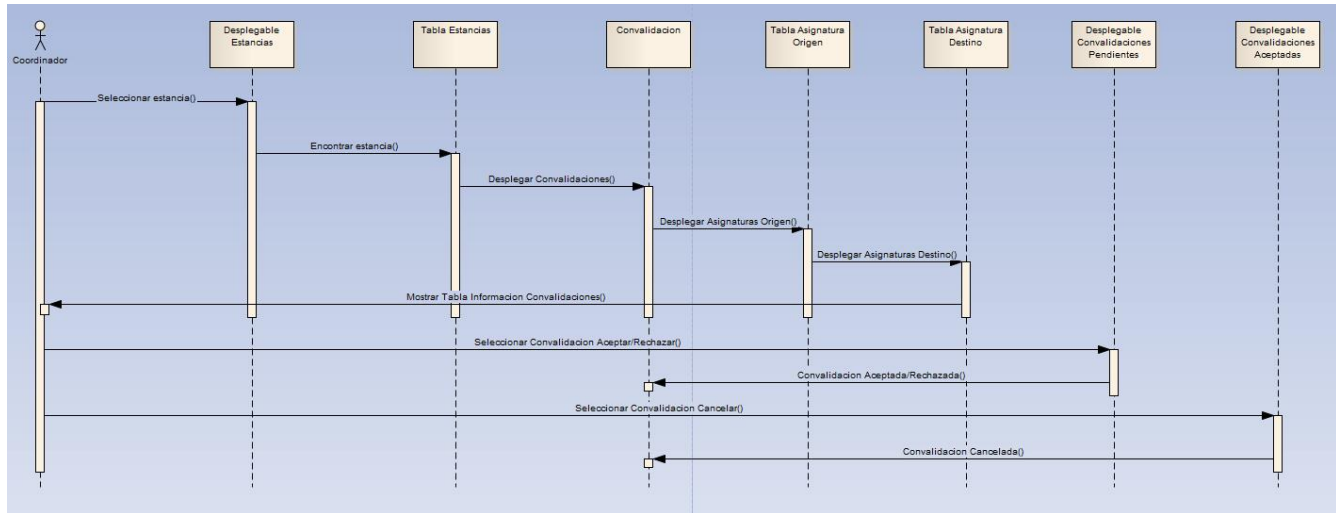


Fig.18 Diagrama de secuencia de modificar una convalidación

Nueva convalidación

Este diagrama de secuencia consiste en insertar una nueva convalidación. Aparecen tres desplegables, uno con las estancias abiertas donde el alumno puede realizar la convalidación, otro con las asignaturas de la universidad de origen y otro con las asignaturas en el sistema de las universidades de las estancias abiertas del alumno. Si todo se realiza correctamente, aparece un mensaje de confirmación.

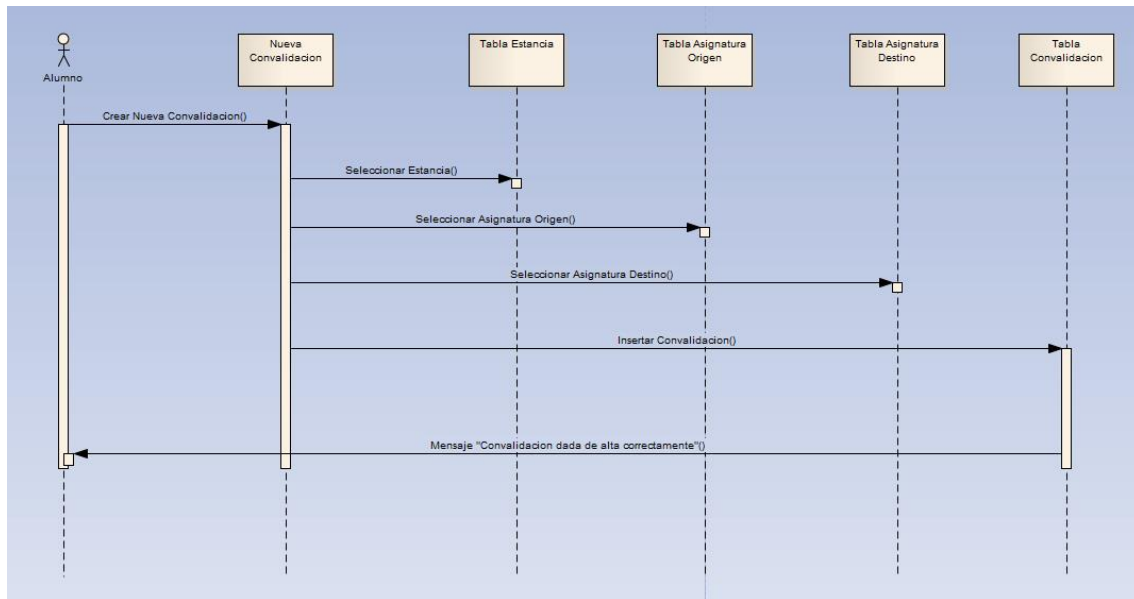


Fig.19 Diagrama de secuencia de cambiar una contraseña

Buscar antiguas convalidaciones

Este diagrama de secuencia consiste en buscar antiguas convalidaciones de otros alumnos. El alumno introduce el código de una asignatura de origen y se le muestra una tabla con todas las convalidaciones aceptadas con esa asignatura en todas las universidades.

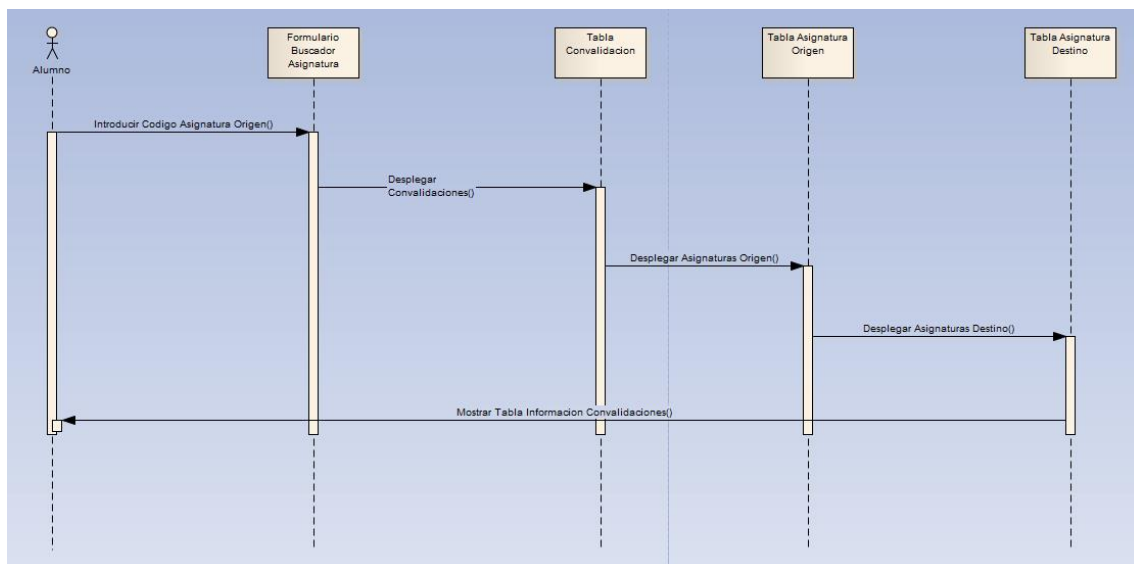


Fig.20 Diagrama de secuencia de modificar una convalidación

Ver estado convalidaciones

Este diagrama de secuencia consiste en modificar una convalidación, ya sea aceptarla, rechazarla o cancelarla. El alumno selecciona la estancia que quiere contemplar. Una vez seleccionada, muestra una tabla con los datos de las convalidaciones y asignaturas de dicha estancia.

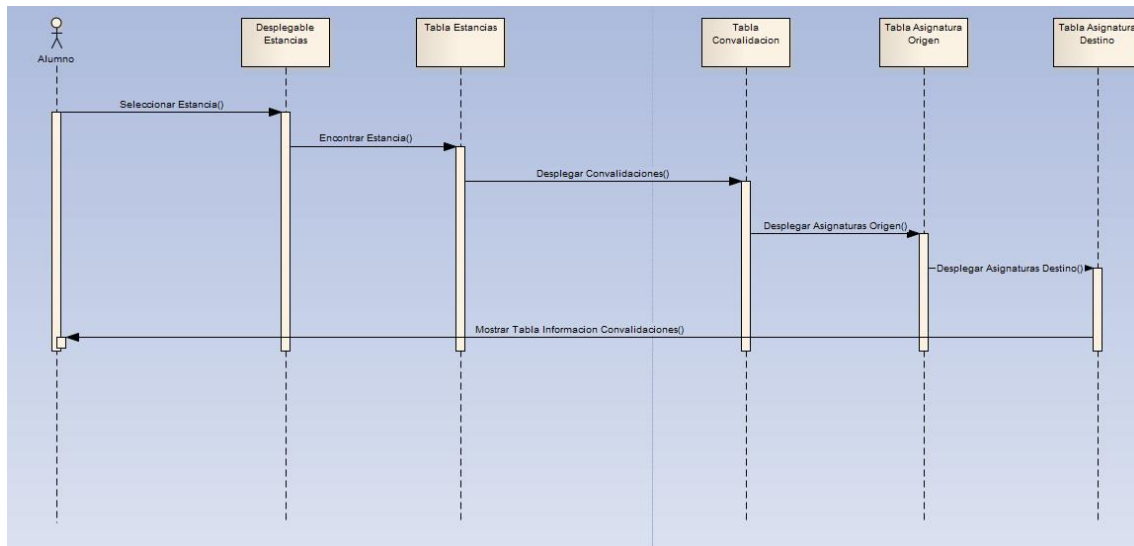


Fig.21 Diagrama de secuencia de ver el estado de convalidaciones

Implementación

Medios utilizados

Se han utilizado diversos medios para la realización del proyecto, los cuales algunos se han ido comentado y otros aún no. A continuación, vamos a describir cada uno de ellos indicando su versión si lo requiere. Los más relevantes son los siguientes:

- NetBeans (8.2): programa que se utiliza como IDE que permite programar en diferentes lenguajes (en este caso Java, JSP, HTML y CSS).
- Java (JDK 1.8): lenguaje de programación orientado a objetos.
- JSP: tecnología diseñada para crear páginas web utilizando etiquetas HTML y código Java.
- Apache Tomcat (9.0.39): servidor web usado en tradicionalmente para proyectos Java por su implementación de servlets o páginas JSP.
- PostgreSQL (10): sistema gestor de bases de datos relacionales que incluye técnicas de orientación a objetos. Es multiplataforma y OpenSource.
- PgModeler (0.9.1): herramienta creada para diseñar las estructuras de datos de nuestras bases de datos en PostgreSQL.
- PgAdmin (4): aplicación de diseño y administración de bases de datos para su uso con PostgreSQL.
- HTML: lenguaje de marcado utilizado para el desarrollo de páginas de Internet.
- JDBC: interfaz de acceso a bases de datos SQL que proporciona un acceso uniforme a una gran variedad de bases de datos relacionales.
- CSS: lenguaje para estilizar código HTML.
- Bootstrap: librería de código abierto utilizada para diseñar aplicaciones web.

Estructura de paquetes

Siguiendo la estructura del patrón modelo vista controlador, se han creado diferentes paquetes para hacer más legible la organización de las clases. Se ha creado un paquete denominado “modelo”, el cual contiene las clases pertenecientes al módulo modelo comentadas previamente, como Alumno, AlumnoDAO, Coordinador, CoordinadorDAO... También, se ha creado un paquete donde se encuentra el controlador y otro, denominado “ACUERDOS” donde se encuentran todas las vistas JSP. Además, se ha creado otro paquete adicional denominado “config” con la clase Conexión, donde se realiza la conexión con la base de datos.

Acceso y obtención de datos

Los datos estarán alojados en un servidor donde estarán seguros y serán accesibles. Cada vez que se ejecute la aplicación, el usuario debe poder trabajar con los datos con la mayor facilidad posible. Por lo tanto, se han implementado una serie de clases que facilitan esto.

Como hemos ido comentando, dentro del modelo se encuentran una serie de clases con la nomenclatura ClaseDAO, donde se realiza la interacción con la base de datos. Estas clases, al instanciarse, se conectan con la base de datos a través de la clase Conexión y, una vez establecida la conexión, comienzan a operar. Existen dos tipos de métodos: los de escritura y los que combinan lectura y escritura.

- Los de escritura consisten en enviar información a la base de datos para realizar una operación con ella sin recibir nada. Habitualmente son los métodos que insertan o modifican registros en la base de datos. Reciben como parámetro variables que contienen la información que se quiere enviar a la base de datos y no devuelven nada. Un caso de ejemplo sería “INSERT INTO TABLE VALUES (?, ?, ?, ?, ?)” donde las interrogaciones representan valores (recogidos de los parámetros) que se van a introducir en la base de datos.
- Los que combinan lectura y escritura son aquellos que envían información a la base de datos, para que realice una determinada operación y devuelva un resultado. Reciben como parámetro variables que contienen la información que se quiere enviar a la base de datos y devuelven el resultado de la acción. Un caso básico sería “SELECT * FROM TABLE WHERE = ?”.

Para crear un método, antes de nada, hay que tener clara qué acción ejecutará el método y a cuál de las dos anteriores pertenece. Los métodos que tengan parámetros, los recibirán del controlador, que a su vez los recibe de la vista. Al inicio del método, es necesario instanciar las clases PreparedStatement (necesaria para enviar la acción) y ResultSet (necesaria para ejecutar la acción). Si se quiere enviar algo a la base de datos, ya sea para escribir en ella o para una condición de una consulta, se debe utilizar el método setTIPO(POSICIÓN, VARIABLE), donde POSICIÓN es el lugar que ocupa la interrogación en la consulta, VARIABLE el contenido va a ir en esa interrogación y TIPO es el tipo de dato que se envía (puede ser setString, setInt, etcétera). Por ejemplo, si se desea realizar la consulta “select DNI from Alumno where Nombre_usuario_Usuario = ?” para obtener el DNI de un alumno a partir de su nombre de usuario, se debe escribir la línea “ps.setString(1, usuario);” donde el contenido de la variable usuario irá donde la interrogación.

Para obtener el resultado de la consulta, se debe utilizar el método getString(NOMBRE_CAMPO) donde NOMBRE_TABLA es el nombre del campo de la base de datos que se desea obtener. En el ejemplo anterior habría que escribir rs.getString("DNI"); para obtener el DNI. Cabe destacar que, si alguna consulta devuelve más de un registro, como es habitual, habría que introducir en un bucle while(rs.next()). El resultado de este método se guarda en una variable que retornará el método.

Por lo tanto, este es el proceso que hay que seguir para que la información llegue desde la base de datos a la vista, pasando por el controlador, quien decide qué acción se debe realizar.

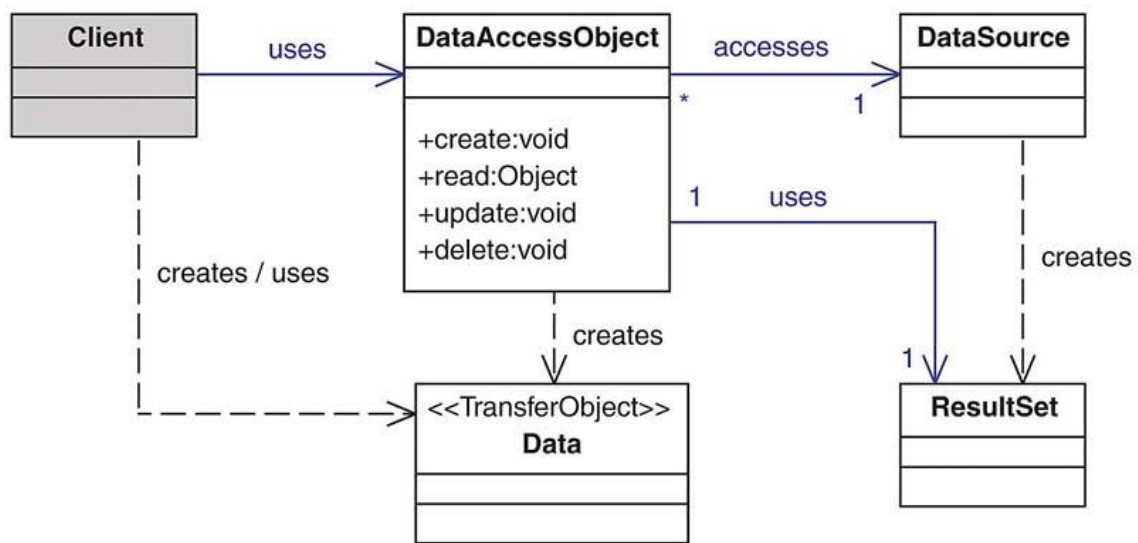


Fig.22 Diagrama de clases de DataAccesObject

Manejo de acciones del usuario

Toda la información se debe introducir y se debe leer a través de una vista. A través de JSP, CSS y Bootstrap se crean las vistas, las cuales consisten en unas interfaces sencillas para que los usuarios puedan trabajar con ellas sin dificultad. A través de las etiquetas `<input>`, para que el usuario introduzca unos datos de entrada, o `<select>`, para que el usuario seleccione una opción entre varias de un desplegable, el usuario puede interactuar con la herramienta y enviar información al sistema.

Sin embargo, esta información no se envía directamente a la base de datos, ya que existe un intermediario que realiza todas las conexiones: el controlador. Para ello, hemos creado una clase que extiende de un servlet, más concretamente de `HttpServlet`. Un servlet se utiliza para ejecutar un servidor web y construir páginas web. En esta clase existen dos métodos, el método `doGet` y el método `doPost`.

- `doGet(HttpServletRequest request, HttpServletResponse response)`: se instancian las clases DAO para acceder a la base de datos a través de ellas. Se crea una variable `RequestDispatcher`, la cual recibirá solicitudes y las enviará al recurso JSP. A través del método `getParameter()` del parámetro `request` se obtienen variables de la vista para poder realizar las operaciones requeridas en el modelo. Con el método `setAttribute()` enviamos información a la vista. También, hay una variable “acción” donde se almacena el valor de la acción que llega de la vista y el usuario quiere que se realice.
- `doPost(HttpServletRequest request, HttpServletResponse response)`: le llegan todas las peticiones del usuario y llama al método `doGet()` para que ejecute estas peticiones.

Cada vez que el usuario quiere realizar una acción en la aplicación, es enviada por la vista al controlador a través del método POST. Por ejemplo, para expresar al sistema que queremos iniciar sesión en el sistema, la línea necesaria es `<form method="POST"`

action="PropuestaLA?accion=comprobarLogin" autocomplete="off" >. El controlador recibirá la acción “comprobarLogin” y realizará las operaciones necesarias para comprobar que las credenciales introducidas son correctas.

Presupuesto

A continuación, se va a detallar el presupuesto aproximado que supone la elaboración de este proyecto. Dentro del presupuesto, es importante diferenciar las diferentes partes en las que se dividirán los costes del proyecto: costes de mano de obra, costes de hardware y materiales y coste de software.

Coste de mano de obra

Para que un proyecto se realice de forma eficiente y se haga una planificación óptima que cumpla los plazos previstos, es necesario dividirlo en tareas. A estas tareas se les asocia un determinado periodo de tiempo. En nuestro caso, vamos a establecer ese periodo de tiempo a un número de horas y, a su vez, vamos a determinar un coste por hora para establecer el coste total de la mano de obra. Según lo requerido en este proyecto las etapas que se han de realizar son las siguientes:

1. Estudio previo de los requerimientos del cliente: consiste en constantes reuniones con el cliente para entender de la manera más eficiente qué es la herramienta que hay que desarrollar. Aproximadamente, esta tarea puede durar unas 20 horas a un precio por hora de 10€, por lo que el coste estimado de esta tarea sería de 200€.
2. Análisis y especificación de requisitos: una vez hecho el estudio previo, se lleva a cabo la especificación de requisitos. Aproximadamente, esta tarea puede durar unas 40 horas a un precio por hora de 12€, por lo que el coste estimado de esta tarea sería de 480€.
3. Diseño y arquitectura software: tras la especificación de requisitos, comienza el diseño y la arquitectura del sistema, donde se organizan todas las herramientas que hay que usar, su estructura y cómo se van a usar. Aproximadamente, esta tarea puede durar unas 80 horas a un precio por hora de 20€, por lo que el coste estimado de esta tarea sería de 1600€.
4. Programación e implementación: llegamos a la implementación de la aplicación, donde se va a realizar toda la lógica que hará que el sistema funcione correctamente. Aproximadamente, esta tarea puede durar unas 120 horas a un precio por hora de 20€, por lo que el coste estimado de esta tarea sería de 2400€.
5. Pruebas y revisión: antes de cerrar el proyecto, hay que realizar una serie de pruebas que confirmen el correcto funcionamiento del sistema. Esto consiste en probar todas las decisiones posibles que puede tomar el usuario al usar la aplicación, por si alguna puede causar problemas. . Aproximadamente, esta tarea puede durar unas 30 horas a un precio por hora de 12€, por lo que el coste estimado de esta tarea sería de 360€.
6. Documentación: una vez terminado el sistema, llegamos a la etapa fundamental de documentación, donde se documenta el código y se realiza una memoria con la explicación del proyecto. . Aproximadamente, esta tarea puede durar unas 10 horas a un precio por hora de 10€, por lo que el coste estimado de esta tarea sería de 100€.

Tras el tiempo y coste estimando de cada etapa, el coste total de la mano de obra sería de 5140€ en un total de 300 horas. La media del precio por hora serían de 17.13€ la hora.

Coste de hardware y materiales

El coste de los materiales físicos para el correcto funcionamiento del sistema se dividen en dos: implementación y lanzamiento. Para la implementación se necesita el hardware habitual en todos los proyectos: un ordenador, un teclado, un ratón, webcam, micrófono, disco duro... Y para el lanzamiento se necesita un servidor donde se aloje el sistema.

Para la implementación del sistema, se ha estimado que un ordenador de sobremesa con excelente funcionamiento, sin sistema operativo instalado y que incluya todos los periféricos necesarios, al igual que un buen procesador, memoria RAM, tarjeta gráfica o disco duro, puede costar alrededor de 1200€.

Por otro lado, para el lanzamiento del sistema se necesita un servidor físico para proporcionar los servicios que ofrece el sistema a los usuarios que se conecten a la red privada de la Universidad de Alcalá para acceder a él. Este servidor se ha estimado que puede costar alrededor de 400€.

Por lo tanto, el coste total estimado de hardware y materiales sería de 1600€.

Coste de software

En cuanto al coste del software utilizado, será el coste más bajo de los tres tipos de coste debido a que los programas utilizados son gratuitos. En el apartado anterior, a la hora de establecer el precio estimado del ordenador no hemos incluido el sistema operativo. Es recomendable que el proyecto se realice en Windows 10, por lo que una licencia de este sistema operativo costaría alrededor de 150€.

El resto de herramientas como NetBeans, Java, PostgreSQL o JSP se pueden adquirir de manera gratuita, por lo que estas no requerirán un coste adicional.

Por último, el dominio .es requerido para lanzar finalmente la aplicación tendría un coste estimado de 15€ anuales.

Sumando estos costes, el coste estimado del software sería alrededor de 165€, teniendo en cuenta que habría que abonar unos 15€ anuales para mantener el dominio.

Coste total

Tras los estudios anteriores, los cuales reflejan el coste estimado de cada una de las partes en las que el proyecto puede ser dividido, el coste total del proyecto sería de 6905€, teniendo en cuenta el abono anual de 15€ para el dominio.

Conclusiones y trabajo futuro

Conclusiones

En este proyecto se ha podido comprobar lo importante que son las tecnologías para cualquier ámbito de nuestra vida diaria. Al fin y al cabo, la herramienta creada ha sido elaborada para facilitar a las personas involucradas la elección de los acuerdos que se van a llevar a cabo en el Learning Agreement. Los tutores de Erasmus tendrán mucha más comodidad para observar los contenidos de las asignaturas que tienen que aceptar y también tendrán una mejor organización de los estudiantes a los que tutorizan. A los alumnos también les ayudará bastante la implementación de esta herramienta proporcionándoles un buscador de acuerdos de otros años, además de recibir respuestas más rápidas de sus tutores en cuanto a las convalidaciones que pueden realizar.

Esta herramienta será una más de las decenas de ellas que usamos en nuestra vida diaria. Muchas de ellas no sabíamos que las necesitaríamos en nuestras vidas hasta que las utilizamos por primera vez y, hoy en día, no podemos vivir sin ellas. Esta es la principal conclusión que he obtenido al realizar este proyecto.

Estamos rodeados de tecnología en nuestro día a día. Tenemos a nuestro alcance cualquier cosa a través un ordenador y una conexión a Internet. Un día alguien inventó Amazon, cuando nadie necesitaba hacer pedidos por Internet. Sin embargo, ahora llegan a nuestras casas numerosos paquetes a la semana.

Me encantaría facilitarles la vida a un pequeño número de personas que alguna vez utilicen esta herramienta que he creado aportando mi granito de arena. Ojalá algún día, gracias a la elaboración de esta herramienta, ningún estudiante recuerde lo tedioso que era el hecho de buscar asignaturas que convalidar para el Erasmus, ni ningún profesor recuerde lo complicado que era decidir si un acuerdo debía ser aceptado o no.

Trabajo futuro

En este apartado se van a proponer varias posibles mejoras que se podrían implementar para mejorar la herramienta. Algunas de ellas son:

- Integración con servidor Cloud: como se ha comentado, los datos se almacenarán en un servidor físico. Una posible mejora de futuro sería integrar la aplicación en un servidor Cloud, como Azure o AWS para poder disfrutar de todas las mejoras que nos proporciona, como seguridad de los datos, escalabilidad o capacidad ilimitada de almacenamiento.
- Mejora de la interfaz: pese a poseer todas las funcionalidades necesarias, la interfaz de la aplicación es muy básica, por lo que tiene mucho margen de mejora si se elabora una interfaz mucho más profesional.
- Expansión a otras universidades: en primera instancia, va a ser utilizada por personas de la Universidad de Alcalá. Otra posible mejora podría ser que más universidades puedan utilizarla, ya que les podría ser muy útil.

- Creación de aplicación móvil: la herramienta creada consiste en una página web. Esta puede ser la opción más útil para todos, ya que es fácil acceder a ella. Sin embargo, podría ser todavía más útil crear aplicaciones móvil donde los usuarios tengan la suya en su móvil y puedan acceder a ella todavía más fácilmente.

Manual de usuario

A continuación, se va a describir una guía para facilitar a los usuarios el uso de la aplicación. Como se ha ido comentando a lo largo de este documento, existirán dos tipos de usuarios: alumno y profesor. Se mostrará un caso de prueba en el que el usuario irá realizando diferentes funciones.

Al introducir en la URL la dirección de la aplicación web, aparecerá una ventana donde se pide al usuario que introduzca sus credenciales. Dependiendo de los registrado en la base de datos, la ventana será distinta, ya que para los alumnos es una interfaz y para los tutores otra. La ventana es la siguiente:

The image shows a login form for the 'PROPUESTA LEARNING AGREEMENT ESTUDIANTES UNIVERSIDAD DE ALCALA'. It features two input fields: 'Usuario' with the text 'javier.martin' and 'Password' with masked characters '*****'. Below the fields is a blue 'Login' button.

Fig.23 Vista de inicio de sesión

Si las credenciales son incorrectas, aparecerá un mensaje notificándolo:

Credenciales introducidas incorrectas

Aceptar

Fig.24 Ventana error inicio de sesión

En todas las interfaces, excepto en la del login, existe un botón en la esquina superior izquierda de volver para regresar a la interfaz anterior que el usuario estaba utilizando:

Volver

Fig.25 Enlace volver ventana anterior

Una vez que hemos introducido las credenciales correctamente, aparecerá la interfaz correspondiente a nuestro usuario.

Cada vez que se realiza una acción, aparece una ventana informando si se ha realizado correctamente o ha habido algún error.

Alumno dado de alta correctamente

Aceptar

Fig.26 Ventana confirmación alta alumno

No se ha podido dar de alta el alumno

Aceptar

Fig.27 Ventana alta alumno incorrecto

Manual del alumno

Al iniciar sesión un usuario profesor, le aparecerá la siguiente ventana:

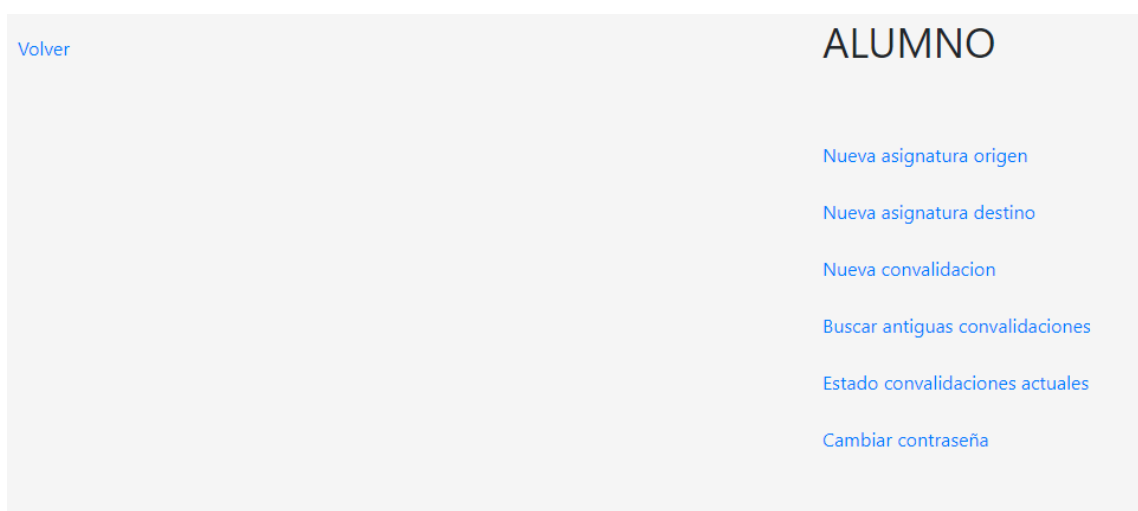
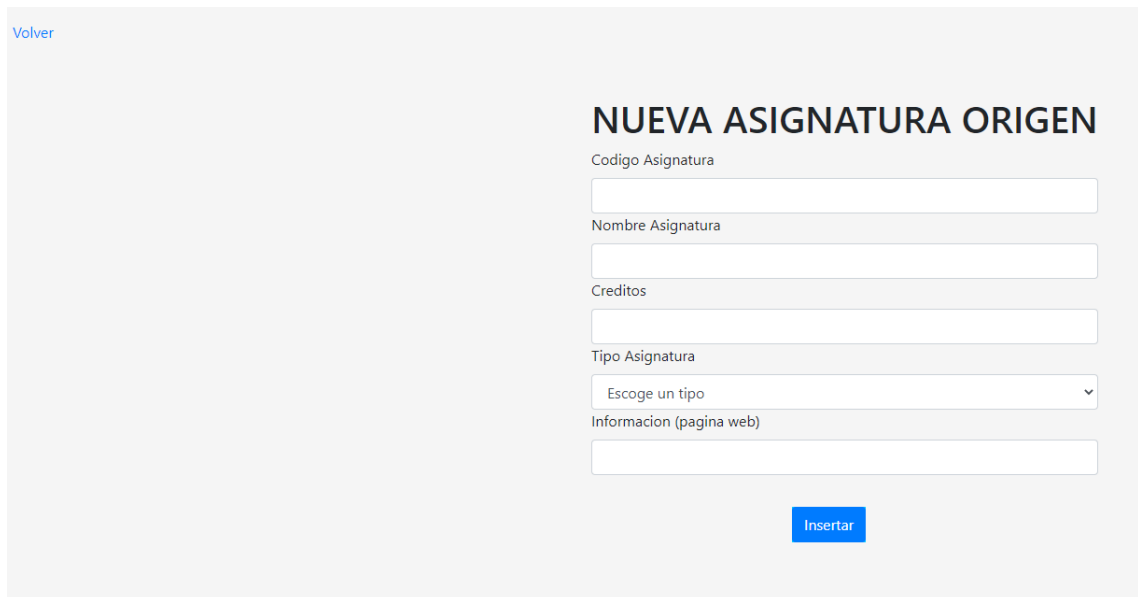


Fig.28 Ventana usuario alumno

Se encuentran 6 enlaces (7 si contamos el de volver) donde el alumno puede realizar las funciones que requiera.

Nueva asignatura origen

El enlace “Nueva asignatura origen” nos redirige a la siguiente interfaz, donde habrá que completar los campos requeridos:



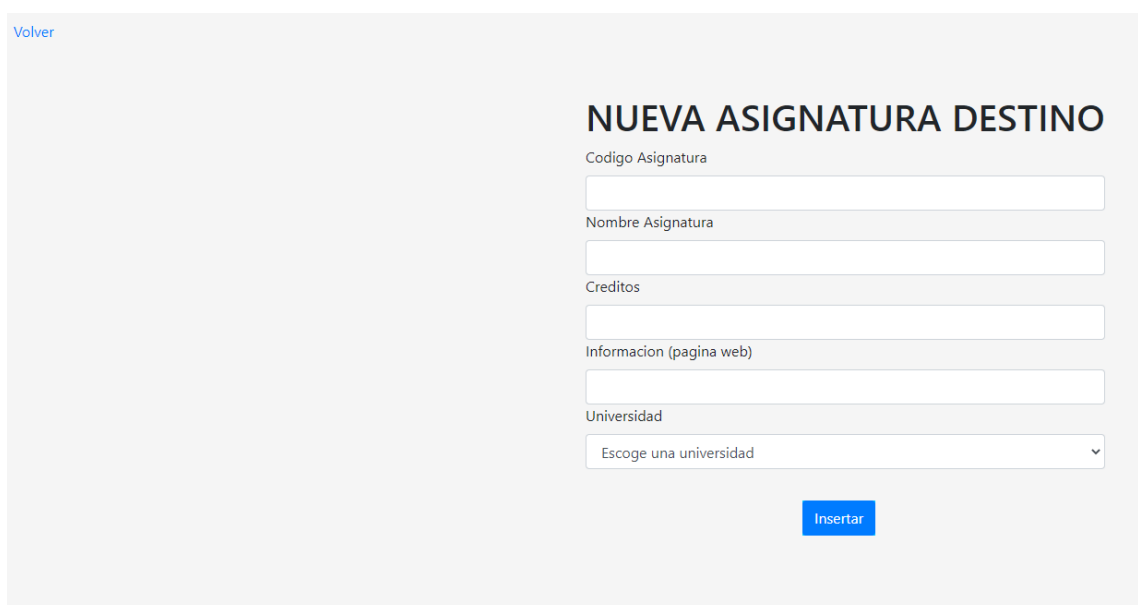
Formulario para crear una nueva asignatura origen. El formulario contiene los siguientes campos:

- Volver** (enlace en la esquina superior izquierda)
- NUEVA ASIGNATURA ORIGEN** (título principal)
- Código Asignatura** (campo de texto)
- Nombre Asignatura** (campo de texto)
- Créditos** (campo de texto)
- Tipo Asignatura** (menú desplegable con la opción "Escoge un tipo")
- Información (pagina web)** (campo de texto)
- Insertar** (botón de acción)

Fig.29 Formulario nueva asignatura origen

Nueva asignatura destino

El enlace “Nueva asignatura destino” nos redirige a la siguiente interfaz, donde habrá que completar los campos requeridos:



Formulario para crear una nueva asignatura destino. El formulario contiene los siguientes campos:

- Volver** (enlace en la esquina superior izquierda)
- NUEVA ASIGNATURA DESTINO** (título principal)
- Código Asignatura** (campo de texto)
- Nombre Asignatura** (campo de texto)
- Créditos** (campo de texto)
- Información (pagina web)** (campo de texto)
- Universidad** (menú desplegable con la opción "Escoge una universidad")
- Insertar** (botón de acción)

Fig.30 Formulario nueva asignatura destino

Nueva convalidación

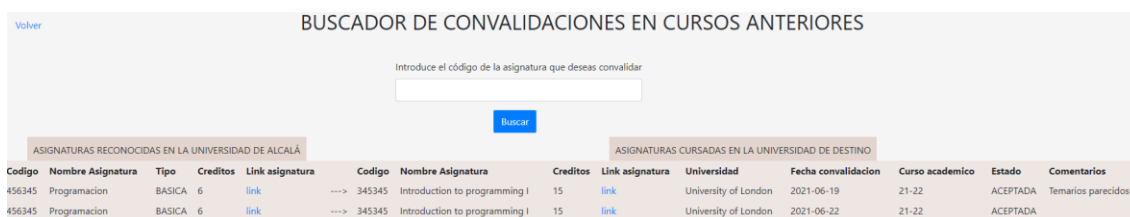
El enlace “Nueva convalidación” nos redirige a la siguiente interfaz, donde habrá tres desplegables. En el desplegable superior se encontrarán las estancias que el alumno tiene abiertas y donde el alumno desee realizar la nueva convalidación. En el desplegable inferior izquierdo aparecen todas las asignaturas de la Universidad de Alcalá registradas en la base de datos. En el desplegable inferior derecho aparecen todas las asignaturas de las universidades de destino asociadas a todas las estancias abiertas del alumno registradas en la base de datos:



Fig.31 Vista registrar nueva convalidación

Buscar antiguas convalidaciones

El enlace “Buscar antiguas convalidaciones” nos redirige a la siguiente interfaz, donde habrá tendremos que introducir el código de la asignatura de origen que queremos convalidar en la universidad de destino. Una vez introducido, aparece una tabla con todas las convalidaciones aceptadas con dicha asignatura:



Codigo	Nombre Asignatura	Tipo	Creditos	Link asignatura
456345	Programacion	BÁSICA	6	link
456345	Programacion	BÁSICA	6	link

Codigo	Nombre Asignatura	Creditos	Link asignatura	Universidad	Fecha convalidacion	Curso academico	Estado	Comentarios
345345	Introduction to programming I	15	link	University of London	2021-06-19	21-22	ACEPTADA	Temarios parecidos
345345	Introduction to programming I	15	link	University of London	2021-06-22	21-22	ACEPTADA	

Fig.32 Vista buscador antiguas convalidaciones

Ver convalidaciones actuales

El enlace “Estado convalidaciones actuales” nos redirige a la siguiente interfaz, donde habrá un desplegable con las estancias abiertas del alumno. Al seleccionar una estancia, aparece una tabla con las convalidaciones solicitadas por el alumno, donde se incluyen el id de la convalidación, el código, nombre, tipo, créditos y link a la página web de la asignatura de origen, código, nombre, créditos, universidad y link de la página web de la

asignatura de destino, fecha de la convalidación, curso académico, su estado (aceptada, rechazada, cancelada o pendiente) y comentarios sobre su estado (si está pendiente no hay comentarios, ya que el profesor no ha modificado el estado) y debajo de la tabla el total de créditos aceptados en ambas universidades.

[Volver](#)

VER CONVALIDACIONES ACTUALES

Escoge la estancia que deseas comprobar

Seleccionar

ASIGNATURAS RECONOCIDAS EN LA UNIVERSIDAD DE ALCALÁ

Convalidación	Código	Nombre Asignatura	Tipo	Créditos	Link asignatura
4	12312434	Base de datos	BÁSICA	6	link
5	12312434	Base de datos	BÁSICA	6	link
6	456345	Programacion	BÁSICA	6	link

ASIGNATURAS CURSADAS EN LA UNIVERSIDAD DE DESTINO

Código	Nombre Asignatura	Créditos	Link asignatura	Universidad	Fecha convalidación	Curso académico	Estado	Comentarios
---> 3	Database1	3	link	University of London	2021-06-22	21-22	PENDIENTE	
---> 6778890	Database 2	3	link	University of London	2021-06-22	21-22	PENDIENTE	
---> 345345	Introduction to programming I	15	link	University of London	2021-06-22	21-22	ACEPTADA	

Total créditos aceptados en la Universidad de Alcalá: 6

Total créditos aceptados en la Universidad de Destino: 15

Fig.33 Vista ver convalidaciones actuales

Cambiar contraseña

El enlace “Cambiar contraseña” nos redirige a la siguiente interfaz, donde habrá que introducir la contraseña actual y la nueva dos veces para hacer efectivo el cambio de ella:

[Volver](#)

CAMBIAR CONTRASEÑA

Contraseña antigua

Contraseña nueva

Repetir contraseña nueva

Cambiar

Fig.34 Formulario cambiar contraseña

Manual del profesor

Al iniciar sesión un usuario profesor, le aparecerá la siguiente ventana:

The screenshot displays the professor's dashboard. At the top left, there is a 'Volver' link. A list of navigation links includes: 'Dar de alta a alumno', 'Nueva estancia de alumno', 'Dar de alta a profesor', 'Nueva universidad de destino', 'Nuevo grado', 'Renuncia alumno', 'Cerrar estancia', 'Nueva asignatura origen', 'Nueva asignatura destino', and 'Cambiar contraseña'. A dropdown menu is set to 'Escoge una estancia y alumno' with an 'Aceptar' button. On the top right, user information is shown: 'Nombre:', 'Apellidos:', 'Codigo Grado:', 'Nombre Grado:', and 'Facultad:'. Below this, two tabs are visible: 'ASIGNATURAS A RECONOCER EN LA UNIVERSIDAD DE ALCALÁ' and 'ASIGNATURAS CURSADAS EN LA UNIVERSIDAD DE DESTINO'. A table with 12 columns (Convalidacion, Codigo, Nombre Asignatura, Tipo, Creditos, Link asignatura, Codigo, Nombre Asignatura, Creditos, Link asignatura, Universidad, Fecha convalidacion, Curso academico, Estado, Comentarios) is present. Below the table, two sections are shown: 'MODIFICAR CONVALIDACION' and 'CANCELAR CONVALIDACION'. Each section has a dropdown for 'Escoge una convalidacion', a dropdown for 'Escoge un estado', a 'Modificar' or 'Cancelar' button, and a 'Comentarios' text area.

Fig.35 Vista usuario profesor

Enlaces

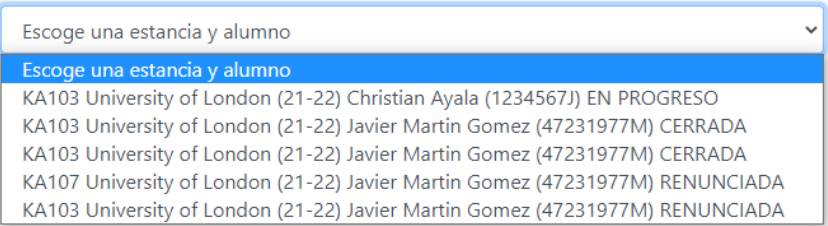
Como vemos, la interfaz del profesor se divide en varias partes. La primera consiste en una serie de enlaces que redirige a otra ventana donde hacer las operaciones que marca cada enlace:

- [Dar de alta a alumno](#)
- [Nueva estancia de alumno](#)
- [Dar de alta a profesor](#)
- [Nueva universidad de destino](#)
- [Nuevo grado](#)
- [Renuncia alumno](#)
- [Cerrar estancia](#)
- [Nueva asignatura origen](#)
- [Nueva asignatura destino](#)
- [Cambiar contraseña](#)

Fig.36 Enlaces usuario profesor

Ver estado convalidaciones

La siguiente parte consiste en un desplegable donde aparecen las estancias de todos los alumnos a los que se está tutorizando y el estado de la estancia (en progreso, cerrada o renunciada):



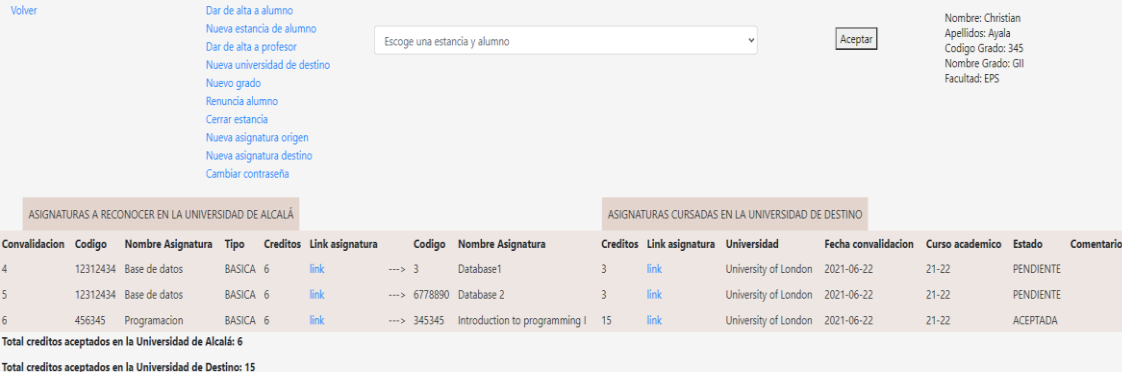
Escoge una estancia y alumno

- Escoge una estancia y alumno
- KA103 University of London (21-22) Christian Ayala (1234567J) EN PROGRESO
- KA103 University of London (21-22) Javier Martin Gomez (47231977M) CERRADA
- KA103 University of London (21-22) Javier Martin Gomez (47231977M) CERRADA
- KA107 University of London (21-22) Javier Martin Gomez (47231977M) RENUNCIADA
- KA103 University of London (21-22) Javier Martin Gomez (47231977M) RENUNCIADA

Aceptar

Fig.37 Desplegable estancias asignadas a profesor

Al seleccionar una estancia, aparecen los datos del alumno (parte derecha), una tabla con las convalidaciones solicitadas por el alumno, donde se incluyen el id de la convalidación, el código, nombre, tipo, créditos y link a la página web de la asignatura de origen, código, nombre, créditos, universidad y link de la página web de la asignatura de destino, fecha de la convalidación, curso académico, su estado (aceptada, rechazada, cancelada o pendiente) y comentarios sobre su estado (si está pendiente no hay comentarios, ya que el profesor no ha modificado el estado) y debajo de la tabla el total de créditos aceptados en ambas universidades.



Volver

Dar de alta a alumno
Nueva estancia de alumno
Dar de alta a profesor
Nueva universidad de destino
Nuevo grado
Renuncia alumno
Cerrar estancia
Nueva asignatura origen
Nueva asignatura destino
Cambiar contraseña

Escoge una estancia y alumno

Aceptar

Nombre: Christian
Apellidos: Ayala
Codigo Grado: 345
Nombre Grado: Gil
Facultad: EPS

ASIGNATURAS A RECONOCER EN LA UNIVERSIDAD DE ALCALÁ						ASIGNATURAS CURSADAS EN LA UNIVERSIDAD DE DESTINO								
Convalidacion	Codigo	Nombre Asignatura	Tipo	Creditos	Link asignatura	Codigo	Nombre Asignatura	Creditos	Link asignatura	Universidad	Fecha convalidacion	Curso academico	Estado	Comentarios
4	12312434	Base de datos	BASICA	6	link	---	3 Database1	3	link	University of London	2021-06-22	21-22	PENDIENTE	
5	12312434	Base de datos	BASICA	6	link	---	6778890 Database 2	3	link	University of London	2021-06-22	21-22	PENDIENTE	
6	456345	Programacion	BASICA	6	link	---	345345 Introduction to programming I	15	link	University of London	2021-06-22	21-22	ACEPTADA	

Total creditos aceptados en la Universidad de Alcalá: 6

Total creditos aceptados en la Universidad de Destino: 15

Fig.38 Vista estado convalidaciones estancia seleccionada

Modificar convalidación

La siguiente parte consiste en un desplegable con las convalidaciones pendientes donde se podrán aceptar o rechazar estas convalidaciones y escribir comentarios sobre la decisión:

MODIFICAR CONVALIDACION

Base de datos --> Database1 (4) ▼

RECHAZAR ▼

Modificar

Comentarios

Las asignaturas no tienen el mismo temario

Fig.39 Desplegables y cuadros de texto para aceptar o rechazar una convalidación

Cancelar convalidación

La última parte consiste en lo mismo que la anterior, pero con la función de cancelar las convalidaciones a petición del alumno:

CANCELAR CONVALIDACION

Programacion --> Introduction to programming I (6) ▼

Cancelar

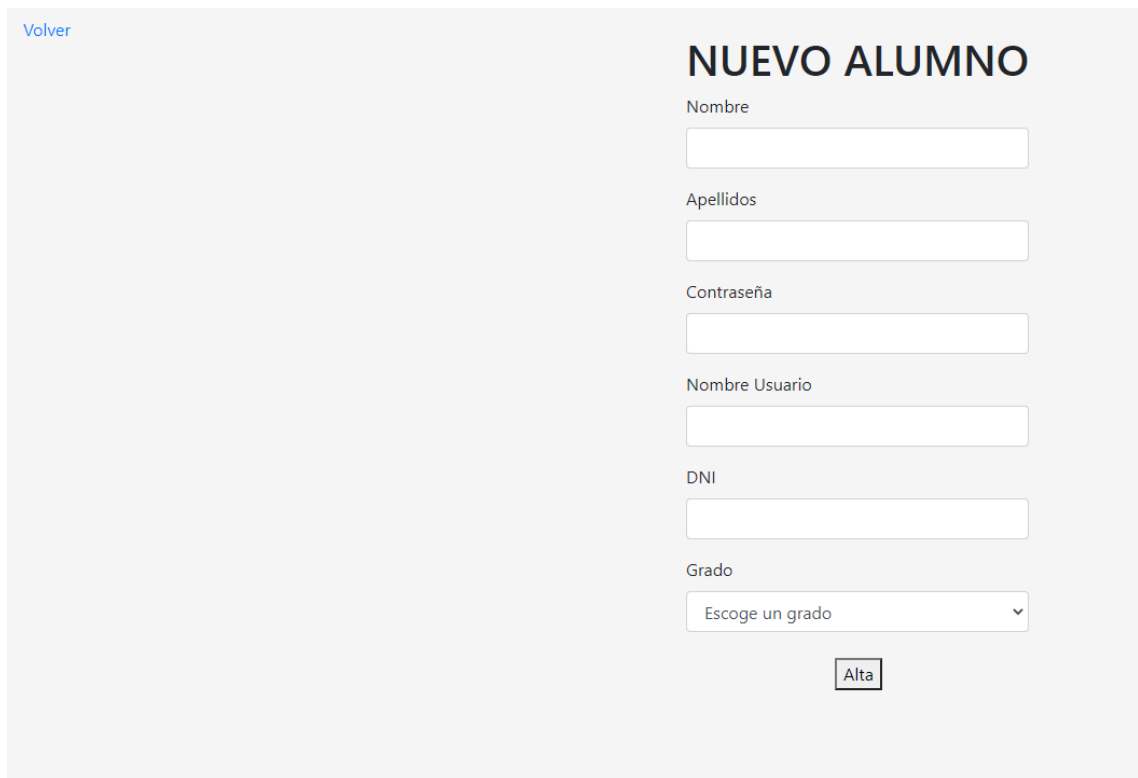
Comentarios

Cancelada por peticion del alumno

Fig.40 Desplegables y cuadros de texto para cancelar una convalidación

Nuevo alumno

El enlace “Dar de alta a alumno” nos redirige a la siguiente interfaz, donde habrá que completar los campos requeridos:



Formulario para crear un nuevo alumno. Incluye campos para Nombre, Apellidos, Contraseña, Nombre Usuario, DNI, Grado, y un botón de Alta.

Volver

NUEVO ALUMNO

Nombre

Apellidos

Contraseña

Nombre Usuario

DNI

Grado

Fig.41 Formulario nuevo alumno

Nueva estancia

El enlace “Nueva estancia de alumno” nos redirige a la siguiente interfaz, donde habrá que completar los campos requeridos. Habrá un desplegable con las alumnos asignados al profesor que ha iniciado sesión:



Formulario para crear una nueva estancia. Incluye campos para Escoge un alumno, Tipo de estancia, Curso Academico, Duracion, Universidad, y un botón de Alta.

Volver

NUEVA ESTANCIA

Escoge un alumno

Tipo de estancia

Curso Academico

Duracion

Universidad

Fig.42 Formulario nueva estancia

Nuevo profesor

El enlace “Dar de alta a profesor” nos redirige a la siguiente interfaz, donde habrá que completar los campos requeridos:



Formulario para dar de alta a un nuevo profesor. El formulario contiene los siguientes campos de entrada:

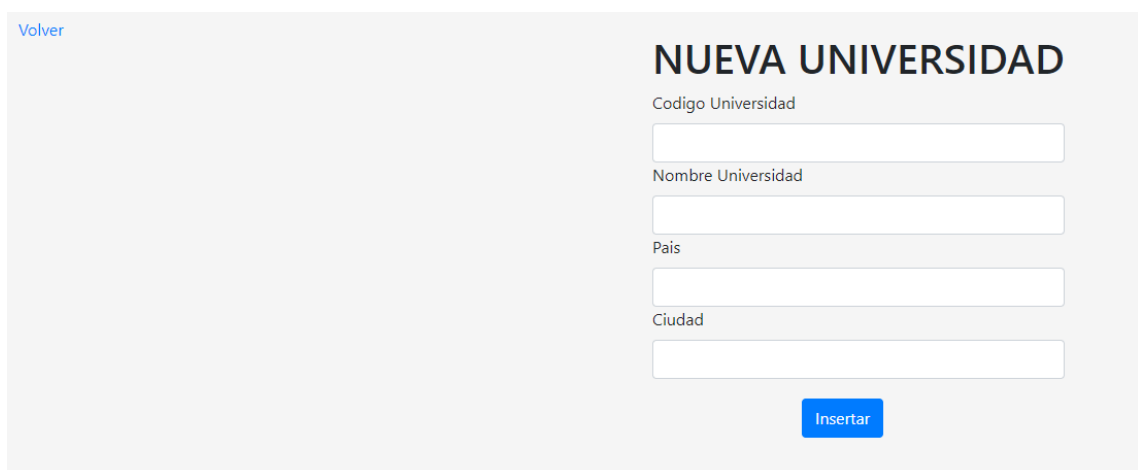
- Nombre
- Apellidos
- Contraseña
- Nombre Usuario
- DNI
- Departamento

El botón "Alta" se encuentra al final del formulario.

Fig.43 Formulario nuevo profesor

Nueva universidad

El enlace “Nueva universidad de destino” nos redirige a la siguiente interfaz, donde habrá que completar los campos requeridos:



Formulario para crear una nueva universidad. El formulario contiene los siguientes campos de entrada:

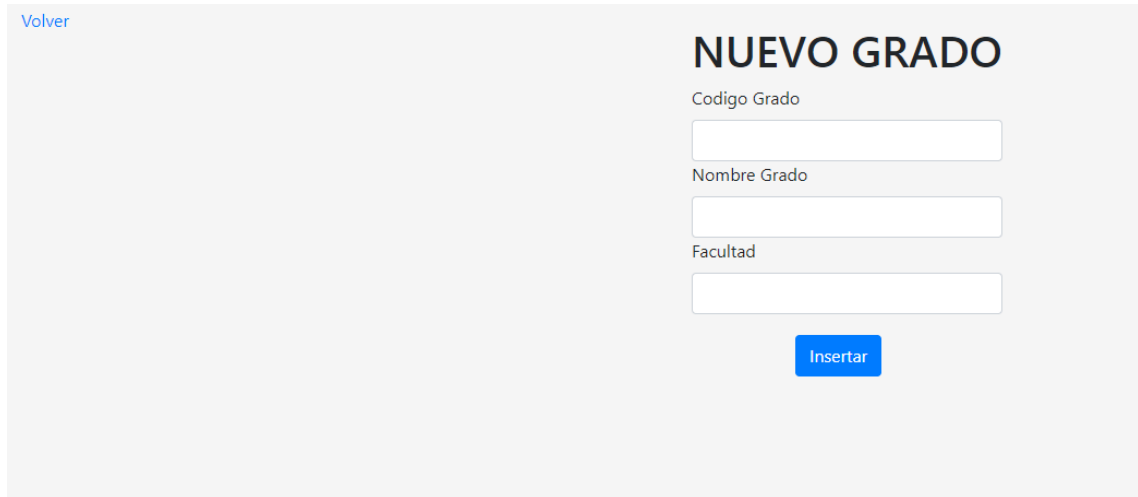
- Código Universidad
- Nombre Universidad
- País
- Ciudad

El botón "Insertar" se encuentra al final del formulario.

Fig.44 Formulario nueva universidad

Nuevo grado

El enlace “Nuevo grado” nos redirige a la siguiente interfaz, donde habrá que completar los campos requeridos:

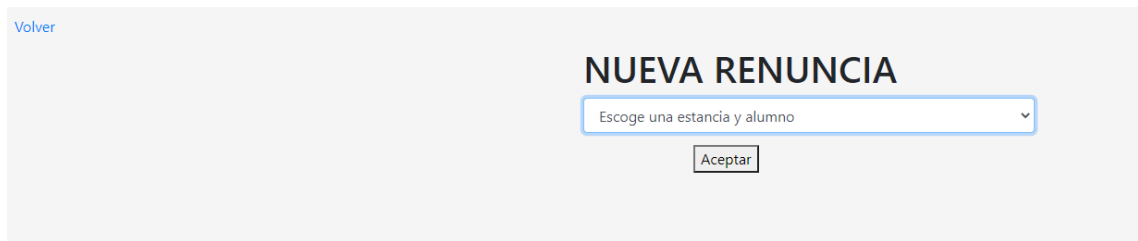


Formulario para crear un nuevo grado. El formulario está ubicado en la parte superior derecha de la interfaz, con un enlace "Volver" en la esquina superior izquierda. El título del formulario es "NUEVO GRADO". Hay tres campos de entrada de texto: "Codigo Grado", "Nombre Grado" y "Facultad". Debajo de los campos hay un botón azul con el texto "Insertar".

Fig.45 Formulario nuevo grado

Renuncia alumno

El enlace “Renuncia alumno” nos redirige a la siguiente interfaz, habrá un desplegable con los alumnos con estancias en progreso, los cuales se podrán seleccionar los que deseen renunciar:



Formulario para registrar una nueva renuncia. El formulario está ubicado en la parte superior derecha de la interfaz, con un enlace "Volver" en la esquina superior izquierda. El título del formulario es "NUEVA RENUNCIA". Hay un campo de entrada de texto con un menú desplegable que muestra "Escoge una estancia y alumno". Debajo del campo hay un botón con el texto "Aceptar".

Fig.46 Vista para nueva renuncia de estancia

Cierre estancia

El enlace “Cerrar estancia” nos redirige a la siguiente interfaz, habrá un desplegable con los alumnos con estancias en progreso a los que se les cerrará la estancia una vez la terminen:

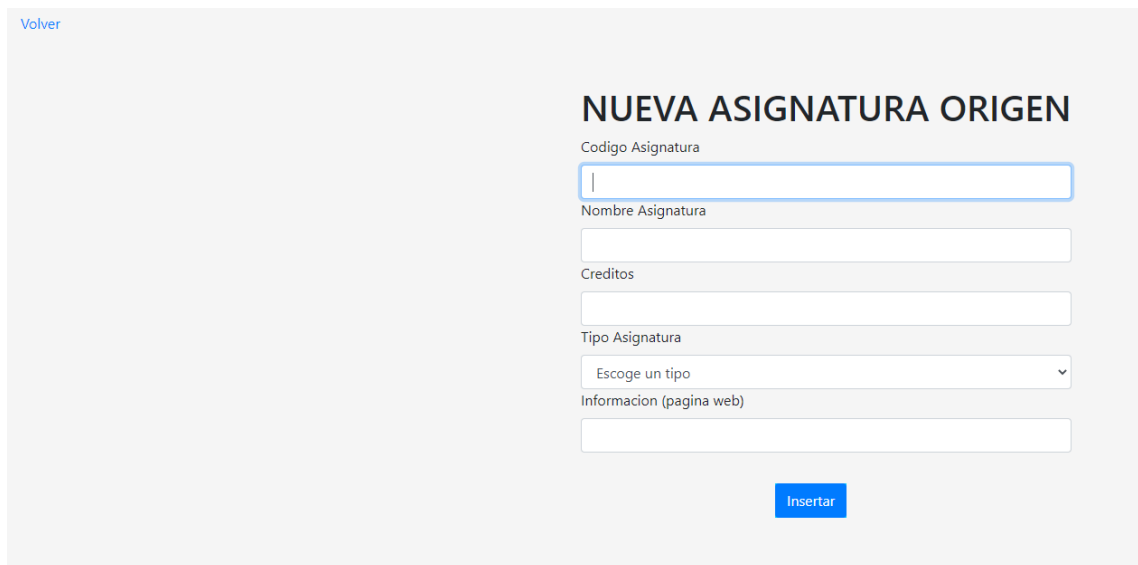


The screenshot shows a web interface titled "NUEVA CIERRE DE ESTANCIA". In the top left corner, there is a blue link labeled "Volver". Below the title, there is a dropdown menu with the text "Escoge una estancia y alumno" and a downward arrow. Below the dropdown menu is a button labeled "Aceptar".

Fig.47 Vista para nuevo cierre de estancia

Nueva asignatura origen

El enlace “Nueva asignatura origen” nos redirige a la siguiente interfaz, donde habrá que completar los campos requeridos:

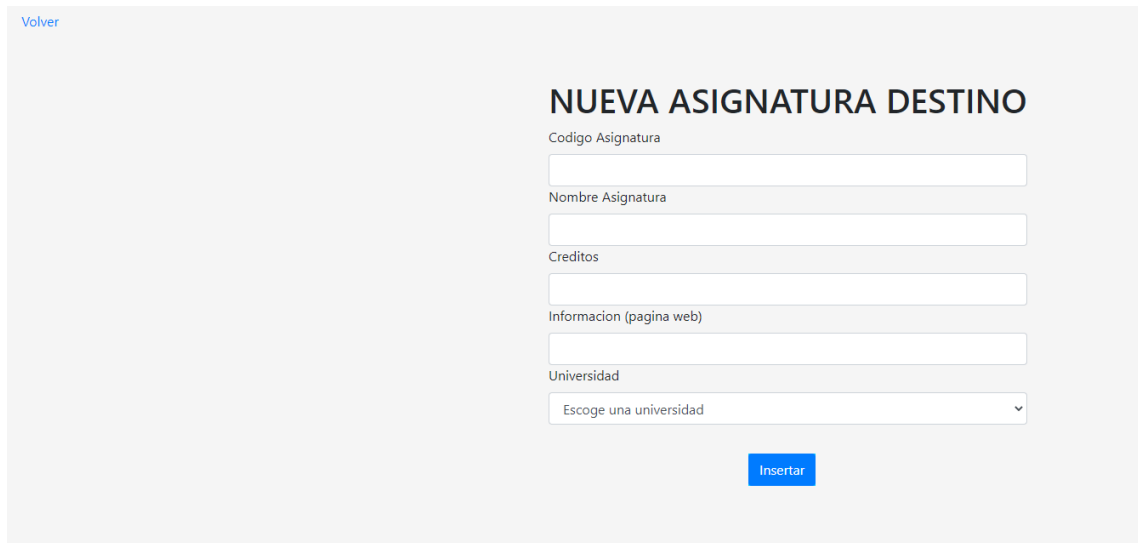


The screenshot shows a web interface titled "NUEVA ASIGNATURA ORIGEN". In the top left corner, there is a blue link labeled "Volver". Below the title, there are several input fields: "Codigo Asignatura" (a text box), "Nombre Asignatura" (a text box), "Creditos" (a text box), "Tipo Asignatura" (a dropdown menu with "Escoge un tipo" and a downward arrow), and "Informacion (pagina web)" (a text box). At the bottom right of the form is a blue button labeled "Insertar".

Fig.48 Formulario nueva asignatura origen

Nueva asignatura destino

El enlace “Nueva asignatura destino” nos redirige a la siguiente interfaz, donde habrá que completar los campos requeridos:



Formulario para crear una nueva asignatura destino. El formulario está ubicado en la parte superior derecha de la interfaz, con un botón "Volver" en la esquina superior izquierda. El título del formulario es "NUEVA ASIGNATURA DESTINO". Los campos de entrada son:

- Codigo Asignatura
- Nombre Asignatura
- Creditos
- Informacion (pagina web)
- Universidad
- Escoge una universidad (lista desplegable)

Un botón "Insertar" azul está ubicado debajo de los campos de entrada.

Fig.49 Formulario nueva asignatura destino

Cambiar contraseña

El enlace “Cambiar contraseña” nos redirige a la siguiente interfaz, donde habrá que introducir la contraseña actual y la nueva dos veces para hacer efectivo el cambio de ella:



Formulario para cambiar la contraseña. El formulario está ubicado en la parte superior derecha de la interfaz, con un botón "Volver" en la esquina superior izquierda. El título del formulario es "CAMBIAR CONTRASEÑA". Los campos de entrada son:

- Contraseña antigua
- Contraseña nueva
- Repetir contraseña nueva

Un botón "Cambiar" azul está ubicado debajo de los campos de entrada.

Fig.50 Formulario cambiar contraseña

Bibliografía

<http://www.todoerasmus.es/learning-agreement/>

https://www.deusto.es/cs/Satellite?blobcol=urldata&blobheader=application%2Fpdf&blobheadername1=Expires&blobheadername2=content-type&blobheadername3=MDT-Type&blobheadername4=Content-Disposition&blobheadervalue1=Thu%2C+10+Dec+2020+16%3A00%3A00+GMT&blobheadervalue2=application%2Fpdf&blobheadervalue3=abinary%3Bcharset%3DUTF-8&blobheadervalue4=inline%3Bfilename%3D%22GUIAcumplimentarLA_1516_BILATERAL.pdf%22&blobkey=id&blobtable=MungoBlobs&blobwhere=1344380071820&ssbinary=true

<https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

<https://sites.google.com/site/sistemadeacademiaelohim/especificacion-de-casos-de-uso>

http://www.jtech.ua.es/j2ee/2002-2003/modulos/servlets/apuntes/apuntes1_1.htm

<https://docs.oracle.com/javaee/7/api/javax/servlet/RequestDispatcher.html>

<https://www.occamagenciadigital.com/blog/conoce-las-fases-de-un-proyecto-de-desarrollo-de-software>

http://www.cua.uam.mx/pdfs/revistas_electronicas/libros-electronicos/2017/java/Java.pdf

<https://empresas.blogthinkbig.com/10-ventajas-del-cloud-computing/>

<https://www.informit.com/articles/article.aspx?p=1398621&seqNum=3> (fig.22)

<https://www.hostinger.es/tutoriales/que-es-apache/>