



Universidad Politécnica
de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos



Grado en Ingeniería Informática

Trabajo de Fin de Grado

**Desarrollo de una aplicación para el
aprendizaje de modelos conceptuales
usando SCORM**

Memoria Final

Autor: David Hernández Salor
Tutor: Óscar Dieste Tubio

Resumen

Cada día se crean proyectos nuevos e ideas futuras prometedoras a nivel de software, pero para poder llevarlos a cabo se necesitan de ciertas fases previas antes de llevarlos a cabo, aquí se encuentra la llamada actividad de Requisitos, que tiene como objetivo conocer el producto que se quiere realizar y que los clientes o usuarios necesitan. El problema comienza cuando se quieren utilizar métodos ya definidos para poder llevar a cabo dicha fase, pero que en su mayoría no son perfectos y pueden llevar a provocar errores o problemas de proyecto tras una mala planificación.

Para ello una de las técnicas de análisis más frecuentemente utilizadas es el uso de modelos conceptuales para la realización de dicha fase.

Este trabajo consiste en crear un sistema de actividades automatizado para que los usuarios aprendan a base de prueba y error a realizar dichos modelos mediante una aplicación SCORM en este caso enfocada a la plataforma de aprendizaje Moodle, para que permita a los profesores, que se encarguen de impartir asignaturas que conlleven el uso de modelos conceptuales, a proporcionar a sus alumnos un sistema de enseñanza que no tenga que ser continuamente supervisado a la hora de corregir los ejercicios.

Abstract

Every day new projects and promising future ideas are created at software level, but in order to carry them out, certain previous phases are needed before carrying them out, here is the so-called Requirements activity, which has the objective of knowing the product that is to be made and that the clients or users need. The problem begins when you want to use methods already defined to carry out this phase, but most of them are not perfect and can lead to errors or project problems after poor planning.

For this reason, one of the most frequently used analysis techniques is the use of conceptual models to carry out this phase.

This work consists of creating an automated system of activities so that users learn by trial and error to carry out these models by means of a SCORM application, in this case focused on the Moodle learning platform, to enable teachers, who are responsible for teaching subjects that involve the use of conceptual models, to provide their students with a teaching system that does not have to be continuously supervised when correcting the exercises.

Índice general

Resumen	I
Abstract	II
1. Introducción	1
2. Antecedentes	3
2.1. Moodle	3
2.2. SCORM	4
2.3. Mermaid	5
3. Objetivos	6
4. Plan de trabajo	7
4.1. Plan de trabajo inicial	7
4.2. Descripción	7
4.3. Objetivos	7
4.4. Lista de tareas	8
4.5. Copia de la propuesta de trabajo escrito por el tutor	8
4.6. Plan de trabajo intermedio	9
5. Desarrollo	10
5.1. Estudio de tecnologías	10
5.2. Definición de requisitos	10
5.3. Confección de prototipo y primer diseño de la aplicación	10
5.4. Implementación	11
5.5. Pruebas y confección de ejercicios	12
5.6. Escritura de la memoria del TFG	13
6. Diseño final	15
7. Conclusión	22
8. Análisis de impacto	23
9. Apéndice	25
9.1. Introducción	25
9.2. Propósito	25
9.2.1. Ámbito del sistema	25

9.2.2.	Definiciones, acrónimos y abreviaturas	25
9.2.3.	Referencias	26
9.3.	Descripción general	26
9.3.1.	Perspectiva del producto	26
9.3.2.	Funciones del sistema	26
9.3.3.	Características de los usuarios	27
9.3.4.	Restricciones:	27
9.3.5.	Suposiciones y dependencias	27
9.4.	Requisitos específicos	27
9.5.	Diseño de las interfaces externas	29
9.6.	Casos de uso	31
9.6.1.	Seleccionar atributos del texto según su tipo	31
9.6.2.	Eliminar atributo seleccionado anteriormente	32
9.6.3.	Paso a la pantalla de crear relaciones con los atributos seleccionados	33
9.6.4.	Crear una relación con los atributos seleccionados	34
9.6.5.	Eliminar relación creada anteriormente	35
9.6.6.	Paso a la pantalla de selección de atributos desde la pantalla de crear relaciones	35
9.6.7.	Paso a la pantalla de crear un modelo conceptual con las relaciones creadas	36
9.6.8.	Paso a la pantalla de crear relaciones con los atributos seleccionados desde la pantalla de modelo conceptual	37
9.6.9.	Ver calificaciones del ejercicio	37
9.6.10.	Ver solución oficial del ejercicio	38
9.6.11.	Pasar a la pantalla de calificaciones del ejercicio desde la pantalla de solución oficial del modelo conceptual	38
9.6.12.	Reintentar ejercicio	39

Índice de figuras

5.1. Guardar elementos seleccionados	12
5.2. Recuperar elementos guardados para poder actualizarlos	12
5.3. Prueba de ejercicio	12
5.4. Prueba de ejercicio	14
6.1. archivo xml con discurso y solucion	15
6.2. Boceto de solución	15
6.3. Pantalla principal	16
6.4. Pantalla principal con los atributos ya seleccionados	17
6.5. Pantalla de relación de atributos	18
6.6. Pantalla de relación de atributos con las relaciones ya creadas	19
6.7. Pantalla de modelo conceptual	20
6.8. Pantalla de calificaciones	20
6.9. Pantalla de solución de ejercicio	21
9.1. Pantalla principal y de selección de atributos del texto	29
9.2. Pantalla de relacionar atributos	29
9.3. Pantalla que muestra un modelo conceptual construido en base a las relaciones propuestas en la anterior pantalla	30
9.4. Selección de finalización de ejercicio	30
9.5. Pantalla de finalización de ejercicio con comparación de soluciones	31

Capítulo 1

Introducción

Con las tecnologías avanzadas, los dispositivos informáticos son cada vez más potentes, y son cada vez más los estudiantes que comienzan a realizar cursos de educación a distancia o basan una parte importante de su aprendizaje en cualquier materia mediante el uso de Internet, además son cada vez más los profesores que proponen gran cantidad de actividades, referentes a asignaturas que ellos mismos imparten, mediante el uso de plataformas de enseñanza online como Moodle, para el autoaprendizaje de sus alumnos. Estas modalidades de enseñanza online presentan mejoras y una mayor comodidad de enseñanza respecto a la típica enseñanza tradicional impartida en un aula en casos como la realización de cuestionarios o ejercicios de respuesta simple, tales como un número o una palabra en específico. Sin embargo, este sistema de enseñanza no es el más adecuado para todos los ámbitos de enseñanza, como la realización de ejercicios complejos que normalmente un profesor debe corregir, debido a la dificultad que puede tener un sistema en corregir ejercicios de respuesta compleja, como los que tienen que ver con la interpretación de un alumno a un ejercicio en particular mediante comentarios o redacciones.

Este proyecto se centra en promover el uso de este tipo de enseñanza para la actividad de Requisitos, correspondiente a la primera fase que todo programador debe realizar para crear o desarrollar software, que se encarga de extraer los requisitos de un determinado producto software y plasmarlos en un documento ERS. Por ello durante esta actividad se aplican diferentes técnicas para detectar y corregir problemas o errores que un desarrollo software pueda llegar a tener.

Una de las técnicas de análisis más frecuentemente usadas y en la que este TFG se va a centrar es el uso de modelos conceptuales para la actividad de Requisitos, que a diferencia de las con las matemáticas, los modelos conceptuales, tal y como se entienden en Ingeniería del software, son un tipo de modelos relativamente poco sofisticados y, por lo tanto, más simples de comprender. Existen varias formas de realizar un modelo conceptual, pero específicamente se van a utilizar los modelos orientados a procesos, los cuales consisten en comprender el dominio de un discurso, es decir, se fijan en las transformaciones que ocurren en dicho dominio abstrayéndolas mediante el concepto de procesos. Un proceso es el fenómeno que se encarga de originar algo nuevo a partir de algo ya preexistente, dicho de otra manera, un proceso es el fenómeno que transforma un conjunto de entradas en un conjunto de salidas que no existían anteriormente. En este tipo de actividades centradas en el autoaprendizaje de modelos conceptuales centrados en el modelo orientado a objetos, los alumnos deberán comprender los aspectos más importantes del dominio de un discurso

como las tareas o actividades que se llevan a cabo en dicho discurso, que en el ámbito de desarrollo software reside en comprender las tareas o actividades que se requieren para la implementación futura de un producto software.

Para poder plasmar un modelo orientado a procesos dentro de un documento se necesita del uso de diagramas de flujo de datos (DFD), que consisten en una representación gráfica para el trazado de cualquier flujo de información para cualquier proceso o sistema mediante el empleo de símbolos definidos. Los DFD centrados en los modelos orientados a procesos tienen cuatro constructores básicos:

- Procesos: Se representan mediante un círculo y se definen como cualquier acción que se desarrolle en el dominio del discurso. Los procesos reciben entradas y generan unos resultados o salidas.
- Flujos de datos: Se representan mediante una flecha, con o sin una etiqueta encima que las defina, y se definen como la entrada o salida de datos de un proceso.
- Entidades externas: Se representan mediante rectángulos y se definen como los elementos que consumen o generan datos en el dominio de un discurso.
- Almacenes: Se representan mediante cajas (Una línea debajo y otra encima del nombre de la etiqueta que contiene el nombre del almacén de datos), y se definen como reservorios temporales de los flujos de datos en su paso por distintos procesos.

Para la realización de este tipo de actividades y cumplimentar el objetivo de este Trabajo de Fin de Grado, se va a seguir un proceso de desarrollo de una aplicación SCORM (Sharable Content Object Reference Model), que se puede empaquetar y subir a la plataforma de enseñanza (Moodle), sistema el cual procesará dicha aplicación o sistema como un ejercicio.

Capítulo 2

Antecedentes

El sistema de la aplicación sirve principalmente para ayudar en el aprendizaje de asignaturas que plantean el uso de modelos conceptuales como elemento necesario para entender el desarrollo de un proyecto software como son las asignaturas de Ingeniería de Software I y II como claros ejemplos.

En esta sección se describen las herramientas que han sido necesarias para el correcto desarrollo del proyecto, como son la plataforma Moodle(entorno educativo para el cual se ha desarrollado la aplicación), SCORM(estándar necesario para la implementación del proyecto) y Mermaid(librería necesaria la creación de diagramas y visualizaciones usando texto y código). Los principales aspectos que definen cada una de estas herramientas son:

2.1. Moodle

[3] Moodle es una plataforma educativa en línea que proporciona entornos de aprendizaje personalizados para los estudiantes. Los profesores encargados de gestionar la plataforma de Moodle según la enseñanza que quieran impartir podrán crear lecciones, administrar cursos, gestionar actividades o exámenes y poder realizar una interacción vía online con los alumnos y resto del profesorado rápida y sencilla. Es una plataforma de código abierto y gratuita, lo cual implica que cualquiera puede crear sus propios cursos y modificarlos a su antojo con las herramientas que la propia plataforma proporciona también además de ser continuamente revisado para adaptarse a las necesidades actuales de los usuarios. Según un estudio del año 2020 el número de usuarios de Moodle a nivel mundial ronda sobre los 200 millones de usuarios, tanto académicos como empresariales y que lo convierten en la plataforma de aprendizaje más ampliamente utilizada del mundo, habiéndose ganado la confianza de grandes y pequeñas instituciones.

En la realización de este proyecto la principal parte a tener en cuenta sobre Moodle es el apartado de actividades que proporciona la plataforma de Moodle UPM, permitiendo a sus usuarios crear actividades o recursos de tipo:

[1]Actividad de H5P: Para utilizar un contenido compatible con H5P Asistencia: Para que un profesor pueda tomar registro en clase de la asistencia de sus alumnos por ejemplo. Base de Datos: permite a los usuarios crear, mantener y buscar información en un repositorio de registros Chat: Permite una comunicación vía online de sus participantes. Consulta: Permite realizar una pregunta con varias respuestas posibles. Cuestionario: permite a los profesores diseñar y plantear cuestionarios hacia

el alumnado, mediante respuesta múltiple, preguntas de verdadero o falso y respuestas cortas o numéricas. Elección de grupo: permite a los usuarios matricularse en los grupos que un profesor haya propuesto. Encuesta: De tres tipos, para obtener la opinión de los participantes de la encuesta, para recopilar información acerca de los usuarios o para recopilar información entre sus alumnos que le ayude a conocer mejor su clase. Foro: Para poder plantear discusiones sobre cualquier tema. Geogebra: Permite la incorporación de actividades GeoGebra. Glosario: Para crear y mantener una lista de definiciones, de forma similar a un diccionario. Herramienta externa: Para interactuar con recursos o actividades en internet externas a la plataforma. HotPot: Sirve para distribuir materiales de aprendizaje interactivos a sus estudiantes y calificar o realizar un estudio de las respuestas. Lección: Sirve para presentar contenido o actividades de un tema en específico. Paquete SCORM: Actividad en la cual se ha basado el desarrollo de este Trabajo de Fin de Grado que sirve para empaquetar un conjunto de archivos según un estándar determinado para realización de actividades o visualización de contenido. Taller: Para la evaluación por pares de trabajos del alumnado. Tarea: Permite evaluar a los usuarios mediante cualquier tipo de tarea que el profesorado considere acorde. Wiki: Permite a los participantes añadir y editar una colección de páginas web. Wooclap: Proporciona una integración de plataforma interactiva de tipo Wooclap. Zoom meeting: Permite la visualización de clases o interacción de los usuarios de manera telemática.

2.2. SCORM

[5] [4] SCORM(Sharable Content Object Reference Model o Modelo Referenciado de Objetos de Contenido Compartible), es un estándar de productos eLearning(procesos de enseñanza-aprendizaje que se llevan a cabo a través de Internet), para ser utilizado en plataformas de enseñanza online, dicho de otra forma, SCORM es una lista de requisitos técnicos establecidos para que un curso o actividad funcione en cualquier plataforma compatible con SCORM.

SCORM funciona de forma que los archivos o materiales compatibles con su estándar son empaquetados mediante una jerarquía específica y crear de esta forma un curso o actividad eLearning.

Dentro de los beneficios que proporciona SCORM se encuentran:

- Compatibilidad: La mayoría de las LMS reconoce a SCORM para la creación de cursos.
- Guardar el progreso: Teniendo la opción de continuar el curso o actividad desde donde lo dejó el alumno.
- Durabilidad: Se puede adaptar a los cambios de necesidades o tecnológicos sin variar el diseño.
- Feedback: Dando la posibilidad al profesorado de evaluar el progreso de los estudiantes.
- Una estructura de curso clara.

- Modularidad: En un curso SCORM, los materiales de formación consisten en unidades individuales, o módulos y cada uno de ellos puede ser usado en cualquier otro curso dentro del LMS.
- Reusabilidad: Se pueden usar elementos en distintas aplicaciones de forma sencilla.

Para que una aplicación empaquetada en formato SCORM funcione correctamente es necesario que incluya:

- Un archivo `imsmanifest.xml`: describe los metadatos del proyecto, la información de la organización como el desarrollador y el título y los recursos usados en la aplicación, por ejemplo el primer fichero que se utiliza para que la aplicación funcione acorde a los deseos del usuario que la ha creado.
- Una serie de archivos de control `.XSD` para manejar los metadatos del paquete.
- Los archivos a utilizar que definirán el diseño y funcionamiento de la aplicación SCORM.

2.3. Mermaid

Mermaid es una herramienta de creación de gráficos y diagramas basado en Javascript. Mermaid permite crear diagramas, sencillos o complejos, fácilmente modificables. En el caso de este proyecto se ha utilizado Mermaid para la implementación de flowcharts, que son diagramas de secuencia que pueden mostrar el funcionamiento de un programa informático. Las características a tener en cuenta para crear flowcharts mediante mermaid son:

- Debe comenzar con el tipo de diagrama que se desea generar, en este caso se usaría la línea de código "graph TD".
- Para generar relaciones se usan los siguientes elementos (ejemplo de relacion: `idatributo1[(nombre del atributo1)] -- > |nombre del flujo| idatributo2((nombre del atributo2))`):
 - 1. Identificadores de los atributos
 - 2. Tipo de encasillamiento con el nombre del atributo:
 - Proceso ((nombre del atributo))
 - Entidad [nombre del atributo]
 - Flujo |nombre del flujo de datos|
 - Almacen [(nombre del atributo)]

Capítulo 3

Objetivos

El objetivo principal de este trabajo de Fin de Grado consiste en la realización de una aplicación de enseñanza acerca de la actividad de Requisitos mediante la creación de modelos conceptuales. Para poder lograr dicho objetivo es necesario que la aplicación permita:

- Facilitar la labor de corrección por parte del profesorado de la actividad de Requisitos mediante modelos conceptuales: la corrección de los ejercicios realizados por el alumnado para su enseñanza en la actividad de Requisitos con el uso de modelos conceptuales será mucho más rápida.
- Mejorar el proceso de aprendizaje de la actividad de Requisitos mediante el uso de modelos conceptuales: proporciona una base para que los alumnos puedan practicar e ir mejorando su nivel de conocimiento en cuanto a la realización de modelos conceptuales.
- Automatizar el proceso de aprendizaje de la actividad de Requisitos mediante modelos conceptuales: El sistema mostrará de forma automática las soluciones de los ejercicios que los alumnos hayan ido realizando.
- Obtener calificaciones y resultados de forma eficiente, sencilla y eficaz: acceder a las calificaciones de dicho ejercicio de manera eficaz y rápida.

Capítulo 4

Plan de trabajo

Esta sección se centra en los principales puntos a tener en cuenta del planteamiento del proyecto en cuestión, es decir, la descripción general del trabajo, la lista de objetivos concretos y un planteamiento de las fechas de inicio y final de dichos objetivos.

4.1. Plan de trabajo inicial

4.2. Descripción

Una de las técnicas más usadas para la actividad de requisitos, dentro de un proceso de desarrollo software, que tiene como objetivo conocer el tipo de producto que los usuarios necesitan es la creación de modelos conceptuales, pero como todo nuevo modelo de creación en este caso, es necesario un proceso de práctica para aprender sobre la utilización de dichos modelos.

El proyecto tiene como objetivo general dar soporte a este proceso de aprendizaje mediante la creación de una aplicación para el aprendizaje de modelos conceptuales usando SCORM (Sharable Content Object Reference Model), un formato que permite estandarizar la producción de contenidos en e-learning, (sistema de formación o aprendizaje que se realiza a través de internet) para poder crear contenidos fácilmente reutilizables usando conocimientos de programación en javascript y HTML+CSS.

4.3. Objetivos

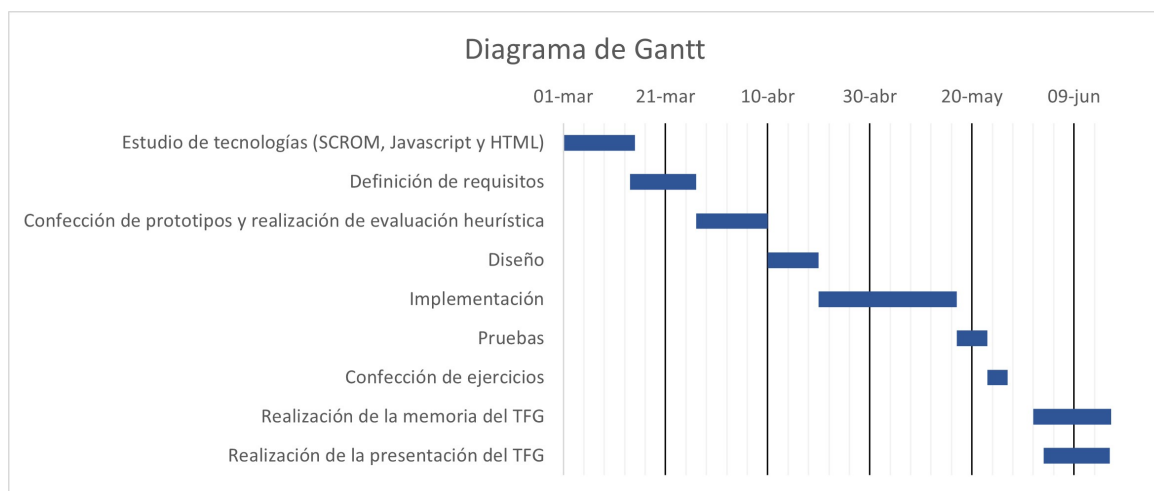
1. Definir los requisitos de la aplicación.
2. Realizar una evaluación de la interacción usando prototipos.
3. Implementar, probar e implantar la aplicación en Moodle.
4. Poblar el repositorio de la aplicación con ejercicios.

4.4. Lista de tareas

En esta tabla se van a reflejar todas las tareas a completar durante el desarrollo del TFG:

Tareas	Fechas de inicio y fin
Estudio de tecnologías (SCROM, Javascript y HTML)	01/03/2022 - 15/03/2022
Definición de requisitos	14/03/2022 - 27/03/2022
Confección de prototipos y realización de evaluación heurística	27/03/2022 - 10/04/2022
Diseño	10/04/2022 - 20/04/2022
Implementación	20/04/2022 - 17/05/2022
Pruebas	20/04/2022 - 17/05/2022
Confección de ejercicios	17/05/2022 - 24/05/2022
Realización de la memoria del TFG	01/03/2022 - 02/06/2022
Realización de la presentación del TFG	03/06/2022 - 16/06/2022

En este diagrama se van a reflejar los datos de la tabla en un diagrama de Gantt:



4.5. Copia de la propuesta de trabajo escrito por el tutor

Planes para los que la propuesta es válida:

10II, 10ID/II.

ID de la oferta:

6169.

Título del trabajo:

Aplicación para el aprendizaje de modelos conceptuales usando SCORM.

Resumen general del trabajo:

4.6. PLAN DE TRABAJO INTERMEDIO

La actividad de Requisitos, dentro del proceso de desarrollo de software, tiene como objetivo conocer el producto que los clientes/usuarios necesitan. Sin embargo, los métodos y técnicas utilizados no son perfectos y llevan con frecuencia a errores y omisiones. Durante la actividad de requisitos se aplican técnicas de análisis para detectar y corregir dichos problemas.

Una de las técnicas de análisis más frecuentemente usadas son los modelos conceptuales. El problema de enseñar a construir modelos conceptuales es que son como las integrales: se aprenden practicando. Es conveniente, por lo tanto, proporcionar algún tipo de soporte automatizado que permita realizar repetidamente ejercicios.

La plataforma educativa usada por la UPM es Moodle. Moodle permite crear tareas usando SCORM. SCORM permite programar la interacción alumno-Moodle que desea el profesor para lograr el objetivo de enseñanza-aprendizaje.

El objetivo general de este TFG es crear una aplicación para el aprendizaje de modelos conceptuales usando SCORM.

Lista de objetivos concretos:

1. Definir los requisitos de la aplicación.
2. Realizar una evaluación de la interacción usando prototipos.
3. Implementar, probar e implantar la aplicación en Moodle.
4. Poblar el repositorio de la aplicación con ejercicios.

Desglose de la dedicación total del trabajo en horas (324 horas en los Grados):

- Estudio de tecnologías (30 horas).
- Definición de requisitos (20 horas).
- Confección de prototipos y realización de evaluación heurística (40 horas).
- Diseño (20 horas).
- Implementación (100 horas).
- Pruebas (30 horas).
- Documentación y posible migración a Open Source (10 horas).
- Confección de ejercicios (24 horas).
- Escritura del TFG (80).

Conocimientos previos recomendados para hacer el trabajo:

- Se necesitará programar en Javascript, así como escribir código HTML+CSS.
- Los conocimientos de SCORM se obtendrán a lo largo de la realización del proyecto.

Tutor 1:

OSCAR DIESTE TUBIO - odieste@fi.upm.es

4.6. Plan de trabajo intermedio

No ha habido cambios en el plan de trabajo respecto al plan de trabajo inicial y en cuanto al progreso realizado me he centrado en la especificación de requisitos y los diferentes puntos que ésta conyeva, para realizar un primer diseño mediante bocetos de las interfaces de la aplicación con sus requisitos, limitaciones y funcionamiento. Además he seguido avanzando en el proceso de aumento de conocimiento de la tecnología SCORM y los lenguajes de programación JavaScript, HTML y CSS.

Capítulo 5

Desarrollo

En esta sección se describen las actividades realizadas para el desarrollo de la aplicación:

5.1. Estudio de tecnologías

La primera actividad que realicé para el desarrollo de la aplicación fue documentarme sobre SCORM en internet, cómo funciona, en qué consiste, sus principales funciones y cómo había que organizar un paquete zip SCORM para su correcto funcionamiento en la plataforma de Moodle. Tutoriales variados que encontré en la red, foros y la página oficial de SCORM fueron mis principales herramientas para poder aprender sobre SCORM y cómo debía organizar mi proyecto.

[2]Otros estudios que tuve que realizar para el desarrollo del proyecto de fin de grado fue la documentación sobre los lenguajes de Javascript, HTML y CSS para realizar tanto el diseño como el funcionamiento de distintos elementos de la aplicación. La visualización de video tutoriales acerca de los tres lenguajes, poner en práctica conocimientos adquiridos de ellos y realizar ejercicios de páginas de aprendizaje como HackerRank fueron fundamentales para mi desarrollo personal en conocimiento con estos tres lenguajes de programación.

5.2. Definición de requisitos

Esta parte fue el primer acercamiento hacia el desarrollo de la aplicación, en cuanto a generar un producto software. Para el desarrollo de esta parte se tuvieron en cuenta las principales funciones que el programa debía tener más las que se desearan aplicar para el correcto funcionamiento de la aplicación, y de esta forma tenerlas en cuenta en las fases posteriores de desarrollo del sistema. Esta especificación viene detallada en el apéndice junto con una sección de diseño de la aplicación y otra de casos de uso.

5.3. Confección de prototipo y primer diseño de la aplicación

Esta sección fue realizada mediante wireframes representados en el apéndice, que presentan una idea del diseño que se quería disponer para la aplicación y, por lo tanto, mostrar una idea general de cómo se quiere realizar.

Con respecto al proyecto final se realizaron cambios del diseño inicial, como lo son la no

utilización de colores para la selección de los atributos necesarios para realizar el modelo conceptual y por lo tanto cambiar el diseño de las cajas de atributos de la pantalla principal y un cambio más grande al acabar el ejercicio con una pantalla más elaborada a la hora de mostrar la puntuación y el modelo oficial del ejercicio.

En cuanto al desarrollo de un primer prototipo, se realizó al mismo tiempo que se iban realizando los wireframes para poder tener una visión más precisa en cuanto a lo que se quería conseguir, pero el primer prototipo tan sólo consistía en elementos colocados en ciertas localizaciones de la pantalla y cómo relacionar los archivos y materiales a utilizar para el desarrollo de la aplicación.

5.4. Implementación

Esta sección fue la más larga en cuanto a tiempo de realización, y se realizó de forma gradual, basándome en los casos de uso y añadiendo las funciones que considerara necesarias en cada momento. En primer lugar organicé como quería que la aplicación gestionara las pantallas, es decir, las funciones que mi aplicación debía proporcionar. El programa funciona de tal manera que mediante el archivo `imsmanifest.xml`, declaro un archivo `html` como primer archivo a tener en cuenta de la aplicación y por lo tanto la pantalla desde la cual se quiere llegar a las demás a lo largo del desarrollo de la actividad que proporciona el servicio. Los principales factores a tener en cuenta en el desarrollo de la implementación son:

- Comunicación con un archivo `xml` mediante una función Javascript desde el cual se quiere sacar el discurso con los atributos que el usuario quiere señalar para la creación del modelo conceptual y la solución oficial a comparar con la solución del usuario una vez haya acabado el ejercicio. El archivo `xml` viene definido de la siguiente manera:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <text>
    discurso del cual se quieren sacar los atributos , para
    crear las relaciones del modelo conceptual.
  </text>
  <attribute type="tipo del atributo (proceso , entidad ,
    flujo , almacen o relacion)"> nombre del atributo o
    relacion en formato flowchart de Mermaid en el caso de
    ser de tipo relacion </attribute>
</document>
```

- Realizar un diseño adecuado de la aplicación final mediante `html+css` para la organización de datos a utilizar o datos generados tras acciones del usuario y a su vez crear un entorno de aplicación agradable y lógico en cuanto a las acciones que tiene que realizar el usuario para el progreso de la actividad, usando también ficheros Javascript para las funciones que se desean tener en el ejercicio en cuanto a nivel de interacción, todo ello desarrollado en el entorno de desarrollo Visual Studio Code.
- Gestionar el guardado de datos para el correcto funcionamiento del servicio mediante:

5.5. PRUEBAS Y CONFECCIÓN DE EJERCICIOS

```
function save(){
    window.location = "relacionarAtributos.html";
    localStorage.setItem("procesos_sel", JSON.stringify(procesos_sel));
    localStorage.setItem("entidades_sel", JSON.stringify(entidades_sel));
    localStorage.setItem("flujos_sel", JSON.stringify(flujos_sel));
    localStorage.setItem("almacenes_sel", JSON.stringify(almacenes_sel));
}
```

Figura 5.1: Guardar elementos seleccionados

```
var procesos = localStorage.getItem("procesos_sel");
var entidades = localStorage.getItem("entidades_sel");
var flujos = localStorage.getItem("flujos_sel");
var almacenes = localStorage.getItem("almacenes_sel");

procesos_sel = JSON.parse(procesos);
entidades_sel = JSON.parse(entidades);
flujos_sel = [];
flujos_sel = JSON.parse(flujos);
almacenes_sel = JSON.parse(almacenes);
```

Figura 5.2: Recuperar elementos guardados para poder actualizarlos

5.5. Pruebas y confección de ejercicios

Para la realización de pruebas se me proporcionó acceso a la asignatura de "Profundización en ingeniería del software" con permisos de edición, para de esta manera añadir actividades con paquetes SCORM y poder probar el funcionamiento de la aplicación. Se realizaron pruebas gradualmente a medida que iba avanzando en la implementación de la aplicación y prevenir futuros problemas con el funcionamiento de la aplicación, para ello se desarrolló un primer ejemplo de ejercicio mediante un archivo xml del que se tienen que extraer el texto que representa el discurso y las soluciones:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <document>
3      <text>
4          xxxxx entidad1 xxxxxx proceso1 xxxxxx flujo1 xxxxxx entidad2 xxxxxx proceso2 xxxxxx almacen1 xxxxxx proceso3 xxxxxx entidad3.
5      </text>
6      <attribute type="proceso">proceso1</attribute>
7      <attribute type="proceso">proceso2</attribute>
8      <attribute type="proceso">proceso3</attribute>
9      <attribute type="entidad">entidad1</attribute>
10     <attribute type="entidad">entidad2</attribute>
11     <attribute type="entidad">entidad3</attribute>
12     <attribute type="flujo">flujo1</attribute>
13     <attribute type="almacen">almacen1</attribute>
14     <attribute type="relacion">entidad1[entidad1]-->|flujo1|proceso1((proceso1));</attribute>
15     <attribute type="relacion">proceso1((proceso1))-->|sin_definir|almacen1[(almacen1)];</attribute>
16     <attribute type="relacion">proceso2((proceso2))-->|flujo1|entidad3[entidad3];</attribute>
17     <attribute type="relacion">entidad3[entidad3]-->|sin_definir|almacen1[(almacen1)];</attribute>
18 </document>
```

Figura 5.3: Prueba de ejercicio

Una vez realizada la aplicación y probado que todos los casos de uso se cumplen perfectamente, se pasó a comprobar el funcionamiento mediante un ejemplo definido, en el cual se encontraron problemas relacionados con cómo se mostraban los nombres de los atributos en la pantalla o problemas con los botones que se generaban cada vez que se seleccionaba un atributo o se creaba una relación y fueron solucionados para asegurar una vez más el correcto funcionamiento de la aplicación.

El papel del tutor fue también fundamental para recibir una aprobación más profesional acerca del servicio y que cumpla con los principales requisitos que se declararon al principio del desarrollo de la aplicación.

5.6. Escritura de la memoria del TFG

La memoria del TFG recoge la información del Trabajo de Fin de Grado en su totalidad y las principales características que definen al producto software, tanto en su realización de principio a fin, como la información y resultados sacados de la aplicación.

Para confeccionarla se ha hecho uso del lenguaje Latex mediante un editor online llamado Overleaf y asegurarse que ante cualquier posible coontratiempo de pérdida de ficheros, la memoria estaría segura en la nube.

La forma en la que se ha estructurado el proyecto Overleaf es la siguiente:

- Carpetas que contienen las imágenes o ficheros.tex que confeccionan la memoria.
- Un archivo main.tex en el que se organizan los apartados de la memoria según aparecen en la memoria.

5.6. ESCRITURA DE LA MEMORIA DEL TFG

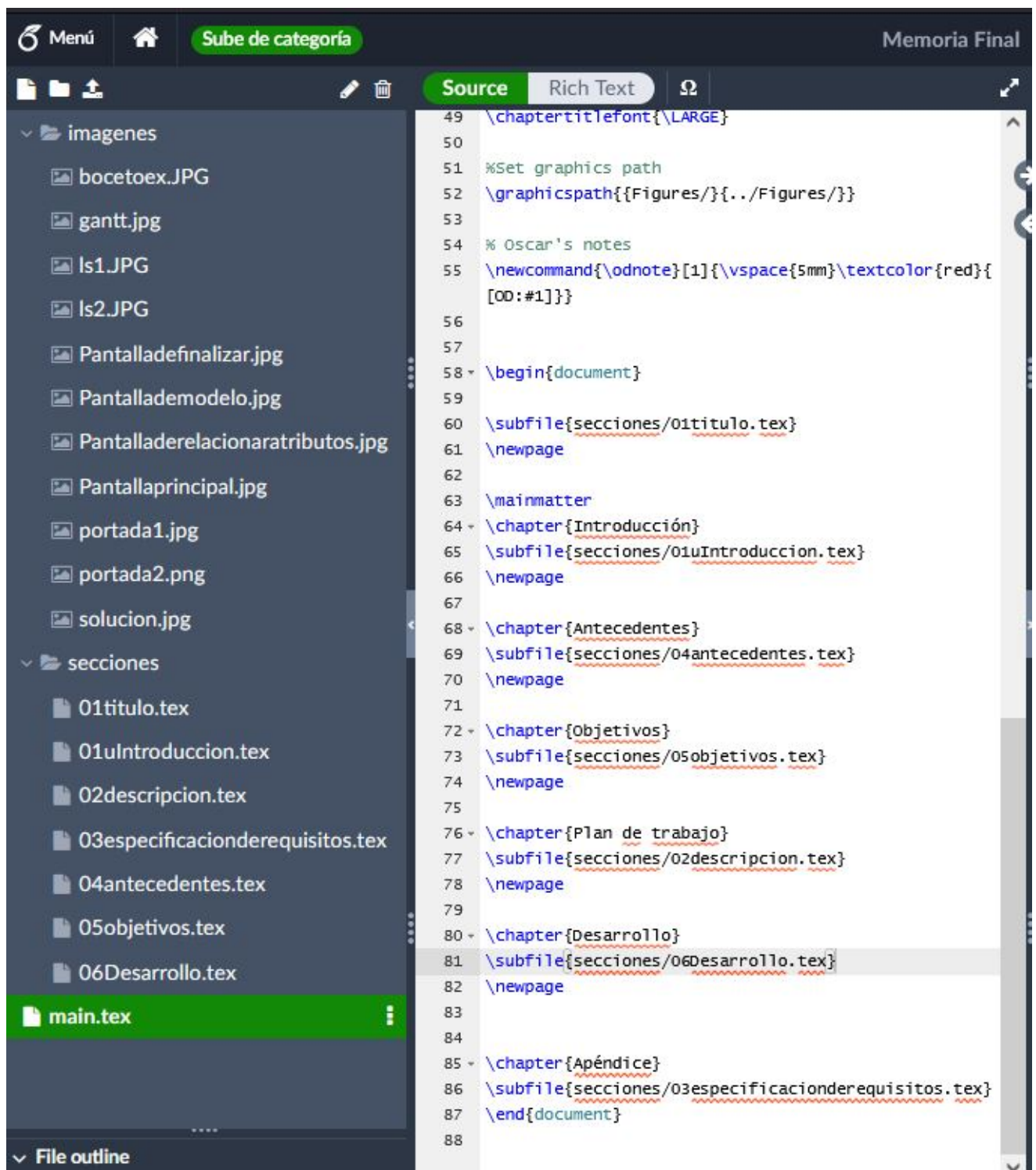


Figura 5.4: Prueba de ejercicio

Capítulo 6

Diseño final

En este apartado se va a mostrar el diseño final de la aplicación mediante una prueba real realizada desde la plataforma Moodle:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <text>
    El sistema deberá generar un listado indicando los pedidos pendientes de entregar a los clientes
    Deben considerarse únicamente los pedidos de las últimas 24 horas
    Junto con cada pedido, debe indicarse si el cliente se encuentra a corriente de pagos
    El listado es enviado al director financiero
  </text>
  <attribute type="proceso">generar un listado</attribute>
  <attribute type="entidad">director financiero</attribute>
  <attribute type="flujo">pedidos de las últimas 24 horas</attribute>
  <attribute type="flujo">listado</attribute>
  <attribute type="flujo">corriente de pagos</attribute>
  <attribute type="almacen">pedidos</attribute>
  <attribute type="almacen">clientes</attribute>
  <attribute type="relacion">clientes[clientes]-->[corriente de pagos]generarunlistado((generar un listado));</attribute>
  <attribute type="relacion">pedidos[pedidos]-->[pedidos de las últimas 24 horas]generarunlistado((generar un listado));</attribute>
  <attribute type="relacion">generarunlistado((generar un listado))-->[listado]directorfinanciero[director financiero];</attribute>
</document>
```

Figura 6.1: archivo xml con discurso y solución

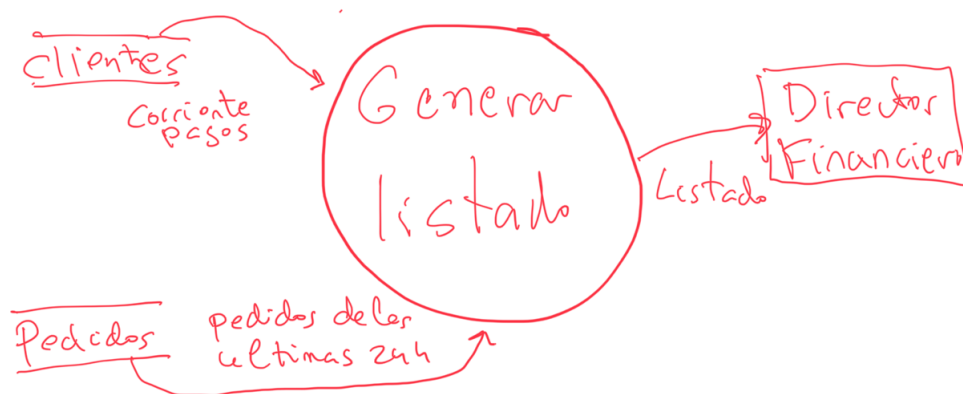


Figura 6.2: Boceto de solución

En la figura 6.1 se muestra el xml referente al discurso del que se plantean seleccionar los atributos referentes a las relaciones que se desean crear, con las soluciones oficiales del

ejercicio, y el modelo conceptual que se prevee como boceto del resultado que debería dar el ejercicio mostrado en la figura 6.2.

Aplicación para el aprendizaje de modelos conceptuales usando SCORM

Modo Revisión

El sistema deberá generar un listado indicando los pedidos pendientes de entregar a los clientes

- Deben considerarse únicamente los pedidos de las últimas 24 horas
- Junto con cada pedido, debe indicarse si el cliente se encuentra a corriente de pagos
- El listado es enviado al director financiero

Procesos

Entidades

Flujos

Almacenes

Relacionar atributos

Figura 6.3: Pantalla principal

En la figura 6.2 se muestra la pantalla principal del programa. Para poder seleccionar los atributos que el usuario considere importantes para la realización del modelo conceptual se debe seleccionar el rango de texto que se quiere tomar como atributo y seleccionar el título del tipo de atributo que desea darle, que a su vez funciona como botón para realizar esta función.

Aplicación para el aprendizaje de modelos conceptuales usando SCORM

Modo Revisión

El sistema deberá generar un listado indicando los pedidos pendientes de entregar a los clientes

- Deben considerarse únicamente los pedidos de las últimas 24 horas
- Junto con cada pedido, debe indicarse si el cliente se encuentra a corriente de pagos
- El listado es enviado al director financiero

El diagrama muestra la interfaz de usuario en modo revisión, organizada en cuatro paneles verticales, cada uno con un título y un campo de texto para atributos seleccionados:

- Procesos:** El atributo seleccionado es "generar un listado".
- Entidades:** El atributo seleccionado es "director financiero".
- Flujos:** Los atributos seleccionados son "pedidos de las últimas 24 horas", "corriente de pagos" y "listado".
- Almacenes:** Los atributos seleccionados son "pedidos" y "clientes".

Cada panel tiene una barra de desplazamiento vertical a la derecha y una barra de desplazamiento horizontal en la parte inferior.

Relacionar atributos

Figura 6.4: Pantalla principal con los atributos ya seleccionados

En la figura 6.4 se recoge la misma pantalla que la anterior pero con los atributos ya seleccionados.

Aplicación para el aprendizaje de modelos conceptuales usando SCORM

Modo Revisión

The interface is divided into two main panels. The left panel contains four stacked scrollable lists: 'Procesos' (with 'generar un listado'), 'Entidades' (with 'director financiero'), 'Flujos' (with 'sin definir', 'pedidos de las últimas 24 horas', 'corriente de pagos', and 'listado'), and 'Almacenes' (with 'pedidos' and 'clientes'). Below these lists is a 'pág anterior' button. The right panel is a large empty area titled 'Relaciones' with a 'pág siguiente' button at the bottom.

Figura 6.5: Pantalla de relación de atributos

En la figura 6.5 se muestra la pantalla de relación de atributos que funciona de forma que se debe seleccionar un primer atributo de la relación, su flujo definido o sin definir y el atributo a que se lo desea comunicar.

Aplicación para el aprendizaje de modelos conceptuales usando SCORM

Modo Revisión

Procesos
generar un listado

Entidades
director financiero

Flujos
sin_definir pedidos de las últimas 24 horas
corriente de pagos listado

Almacenes
pedidos clientes

pág anterior

Relaciones
clientes---corriente de pagos--->generar un listado
pedidos---pedidos de las últimas 24 horas--->generar un listado
generar un listado---listado--->director financiero

pág siguiente

Figura 6.6: Pantalla de relación de atributos con las relaciones ya creadas

En la figura 6.6 se recoge la misma pantalla que la anterior pero con las relaciones ya creadas.

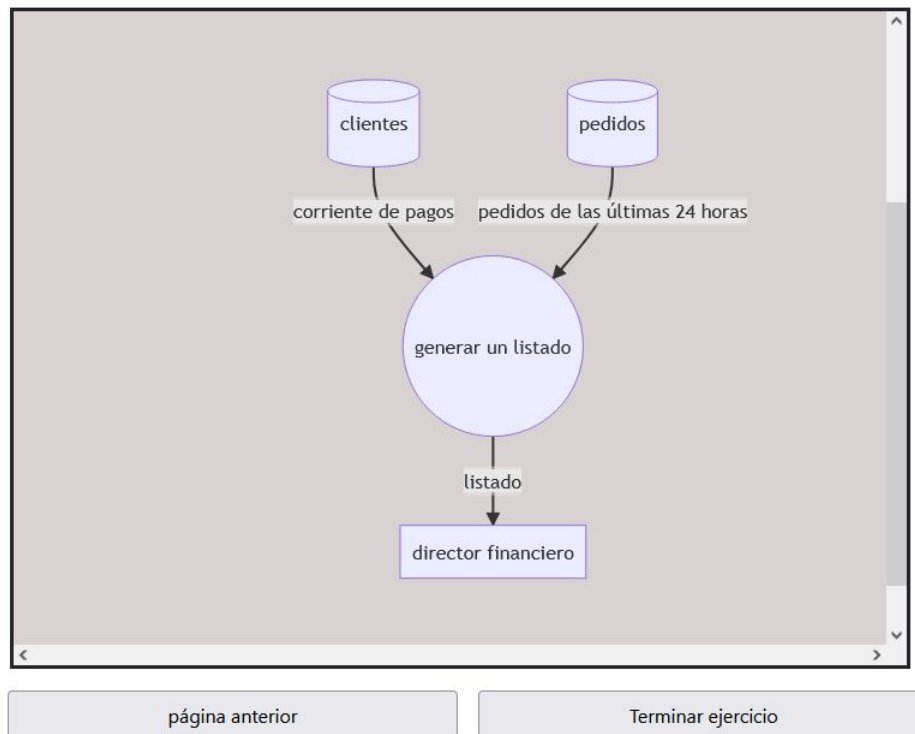


Figura 6.7: Pantalla de modelo conceptual

En la figura 6.7 se muestra la pantalla con el modelo conceptual que se ha creado con las relaciones que el usuario ha creado y por lo tanto la solución del usuario.

Atributo	Aciertos	Total	Peso en la nota
Procesos	1	1	1/10
Entidades	1	1	1/10
Flujos	3	3	1/10
Almacenes	2	2	1/10
Relaciones	3	3	5/10
Resultado	10 (10)	10 (10)	

Solución del modelo

Reintentar ejercicio

Figura 6.8: Pantalla de calificaciones

En la figura 6.8 se muestran las calificaciones tras haber realizado el ejercicio, con una tabla en la que se recogen los aciertos de los atributos seleccionados y de las relaciones creadas, cada uno con un peso determinado en la nota final. También se aprecian dos botones que sirven para pasar a la pantalla que muestra la solución oficial de la actividad y otro para reintentar el ejercicio con las últimas respuestas que el usuario seleccionó guardadas para su posible actualización.

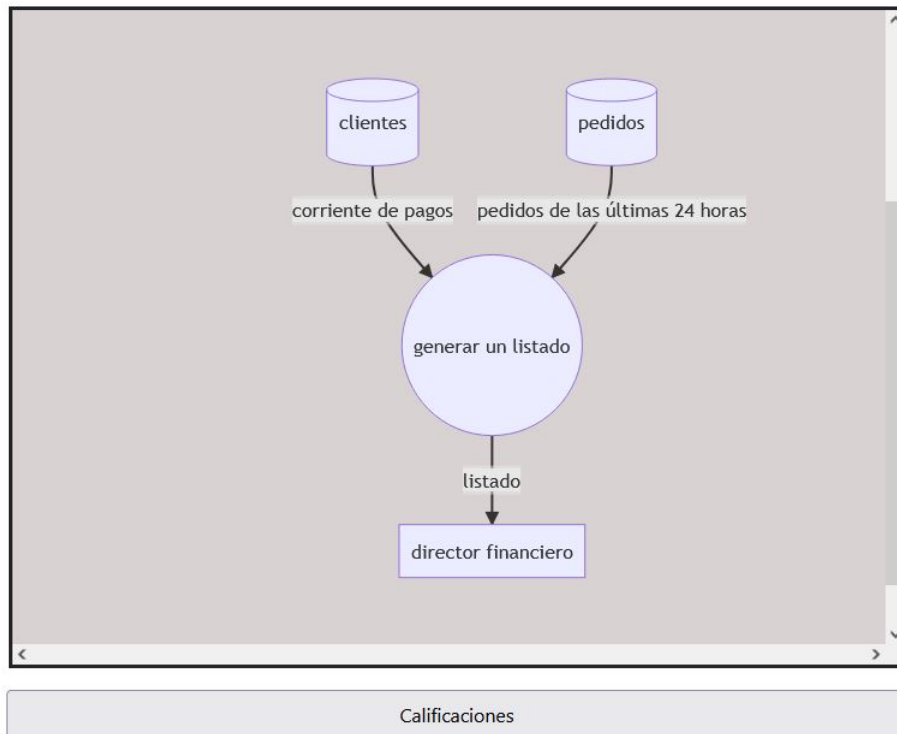


Figura 6.9: Pantalla de solución de ejercicio

En la figura 6.9 se muestra la solución oficial del ejercicio.

Capítulo 7

Conclusión

En conclusión, me siento bastante satisfecho con el desarrollo de este proyecto, superando mis expectativas personales a nivel de desarrollo e investigación, teniendo en cuenta que no partía con conocimientos previos de HTML+CSS, Javascript ni SCORM para la realización del Trabajo de Fin de Grado y tras haber finalizado con éxito la aplicación considero tener un dominio consistente de dichos lenguajes.

He comprendido mejor como gestionar la realización de un proyecto software al haber realizado investigaciones acerca de la actividad de Requisitos, a la vez que iba realizando el proyecto porque debía implementar dicha fase para mi proyecto también, mediante planes de trabajo, especificar los requisitos y realizar un primer diseño de lo que planeé realizar para mi aplicación y ver su proceso de evolución, tanto de como está siendo creada como de cambio de características que no me acababan de convencer.

Esta aplicación es una primera versión realizada con éxito a la que se le pueden aplicar futuras modificaciones para seguir mejorando la aplicación y que su nivel de enseñanza y automatización sea mejor aún.

Para terminar quiero agradecer el papel de mi tutor Óscar, al estar siempre atento a cada duda que me surgiera, las recomendaciones que me proporcionaba en forma de ideas y las reuniones que realizábamos cada semana para poder realizar un control de como avanzaba mi proyecto de Fin de Grado y de esta manera disminuir los problemas al mínimo y no quedarme atascado.

Capítulo 8

Análisis de impacto

Esta sección se centrará en un análisis de impacto que la aplicación puede suponer tras los resultados obtenidos en distintos aspectos:

- Personal: A nivel personal y académico supone un gran impacto, debido al conocimiento que he logrado adquirir sobre la creación y documentación de un proyecto software y los problemas que he logrado superar por mí sólo.
- A nivel de comercialización y social, pretendo que sea de código abierto para que pueda ser utilizado por cualquier usuario e institución que necesite implantar un proceso de aprendizaje más dinámico sobre la actividad de Requisitos para que cada vez más alumnos puedan aprender acerca de ello y de esta forma estar más concienciados de lo que supone crear un proyecto software y dar rienda suelta a sus ideas, en caso de querer realizar un futuro desarrollo de algún producto que pueda llegar a ser promotor y posiblemente comercializado.
- Medioambiental: No supone ningún impacto medioambiental del que se tenga constancia a parte del poco consumo energético que conlleva realizar dicha actividad.
- Cultural: Podría suponer un gran impacto cultural, porque tal y como he explicado antes, toda persona con un conocimiento general acerca de la informática que no sepa acerca de las fases de creación de un proyecto software puede partir con el aprendizaje de la actividad de Requisitos mediante el uso de modelos conceptuales para poder ver con más claridad y organización sus ideas.

Bibliografía

- [1] *Actividades*. URL: <https://docs.moodle.org/all/es/Actividades>.
- [2] Chris Love. "9 Ways to Remove Elements From A JavaScript Array - Plus How to Safely Clear JavaScript Arrays". URL: <https://love2dev.com/blog/javascript-remove-from-array/>.
- [3] *MOOC: ¿Qué es un Mooc?* URL: <https://mooc.es/que-es-un-mooc/>.
- [4] *XML Schema Definition Files*. URL: <https://scorm.com/scorm-explained/technical-scorm/content-packaging/xml-schema-definition-files/>.
- [5] Ronald J. Mora Z. *Un comentario sobre "SCORM: Qué es y cómo usarlo"*. URL: <https://www.ispring.es/blog/que-es-scorm>.

Capítulo 9

Apéndice

9.1. Introducción

Esta parte del documento se corresponde a la especificación de requisitos de software (ERS) referente al proyecto de una aplicación para el aprendizaje de modelos conceptuales usando paquetes SCORM.

9.2. Propósito

9.2.1. Ámbito del sistema

Este servicio consiste en proporcionar un soporte automatizado mediante el uso de la paquetería SCORM a base de JavaScript y HTML+ CSS, que permita realizar ejercicios de creación de modelos conceptuales para que los alumnos practiquen técnicas de la actividad de Requisitos, dentro del proceso de desarrollo de software. Tras la realización de dicho sistema, se implantará dentro del aula virtual de Moodle en el apartado que el profesor considere correspondiente, facilitando así la labor de corrección del profesorado con este tipo de ejercicios de forma repetitiva que tiene los siguientes objetivos:

- Facilitar la labor de corrección y enseñanza por parte del profesorado de la actividad de Requisitos mediante el uso de modelos conceptuales hacia el alumnado.
- Mejorar el proceso de aprendizaje de la actividad de Requisitos mediante modelos conceptuales por parte del alumnado.
- Automatizar el proceso de aprendizaje de la actividad de Requisitos mediante modelos conceptuales por parte del alumnado.
- Obtener calificaciones y resultados de forma eficiente, sencilla y eficaz.

9.2.2. Definiciones, acrónimos y abreviaturas

SCORM (Sharable Content Object Reference Model): Modelo Referenciado de Objetos de Contenido Compartible, es un estándar de productos eLearning, para ser utilizado en plataformas de enseñanza online.

HTML (HyperText Markup Language): Lenguaje de Marcas de Hipertexto, se encarga de definir el significado y la estructura de cualquier contenido web.

CSS (Cascading Style Sheets): Hojas de estilo en cascada, se encarga de especificar el diseño y la compaginación del código HTML.

9.2.3. Referencias

Ninguna.

9.3. Descripción general

En esta sección se van a especificar las principales funciones y restricciones que posee el sistema.

9.3.1. Perspectiva del producto

El producto final consistirá en un sistema software compatible con plataformas de enseñanza online, en este caso Moodle, mediante el uso de un bloque de material web empaquetado que sigue el estándar SCORM, para que los profesores de asignaturas que impartan la actividad de Requisitos como parte de su asignatura puedan proporcionar un sistema de autoaprendizaje automatizado a sus alumnos que les permita practicar la técnica de análisis que consiste en la creación de modelos conceptuales, para dicha actividad.

9.3.2. Funciones del sistema

En este apartado se describirán las funciones que nuestro sistema debe estar capacitado para realizar:

Facilitar la labor de corrección por parte del profesorado de la actividad de Requisitos mediante modelos conceptuales:

Mediante el uso del sistema por parte del profesorado, la corrección de los ejercicios realizados por el alumnado para su enseñanza en la actividad de Requisitos con el uso de modelos conceptuales será mucho más rápida, anulando la necesidad de ir visualizando estos ejercicios uno a uno.

Mejorar el proceso de aprendizaje de la actividad de Requisitos mediante el uso de modelos conceptuales:

El problema de enseñar a construir modelos conceptuales es que se necesita una práctica continua de realización de este tipo de modelos para poder hacerlos de manera más eficiente a medida que se van realizando más, por lo que el uso de este sistema en el aula virtual proporciona una base para que los alumnos puedan practicar e ir mejorando su nivel de conocimiento en cuanto a la realización de modelos conceptuales y todo lo que ello conlleva.

Automatizar el proceso de aprendizaje de la actividad de Requisitos mediante modelos conceptuales:

El sistema mostrará de forma automática las soluciones de los ejercicios que los alumnos hayan ido realizando, anulando así los tiempos de espera que se necesitarían en caso de que sea un profesor el que tenga que realizar dicha labor de corrección de los ejercicios.

Obtener calificaciones y resultados de forma eficiente, sencilla y eficaz:

El momento en el que un alumno haya realizado un ejercicio de modelos conceptuales, tanto el alumno como los profesores podrán acceder a las calificaciones de dicho ejercicio de manera eficaz y rápida, sin necesidad de tiempos de espera para corregirlo.

9.3.3. Características de los usuarios

Existen dos tipos de usuarios para este sistema: Profesores: serán los encargados de realizar la subida o eliminación de los paquetes SCORM que contienen el sistema de realización de ejercicios. Alumnos: realizarán el proceso de resolución de los ejercicios que hayan sido subidos en el aula virtual por parte del profesorado correspondiente.

9.3.4. Restricciones:

El sistema partirá de un ejercicio ya solucionado y el software necesario para su implementación en el aula virtual tiene que estar desarrollado con el estándar SCORM.

9.3.5. Suposiciones y dependencias

Serán necesarias una conexión a Internet y el uso de la plataforma Moodle para poder usar el sistema de ejercicios.

9.4. Requisitos específicos

- Requisito 1: El profesor podrá subir paquetes SCORM con el sistema del ejercicio.
- Requisito 2: El profesor podrá eliminar paquetes SCORM con el sistema del ejercicio.
- Requisito 3: El sistema debe corregir que el modelo contiene los procesos, flujos de datos, entidades externas, almacenes de datos y relaciones entre atributos correctos para la realización del modelo conceptual.
- Requisito 4: El sistema debe permitir pasar entre pestañas del ejercicio cuando el usuario lo crea necesario antes de finalizar el ejercicio.
- Requisito 5: El sistema debe permitir salir del ejercicio sin haberlo completado.
- Requisito 6: El sistema debe estar integrado en Moodle.

9.4. REQUISITOS ESPECÍFICOS

- Requisito 7: El sistema deberá proporcionar al módulo Moodle la calificación total del ejercicio una vez terminado.
- Requisito 8: El sistema deberá mostrar la solución oficial del ejercicio una vez haya sido realizado.
- Requisito 9: El sistema debe permitir al alumno visualizar un texto para la realización posterior de un modelo conceptual mediante la señalización en el texto de los atributos que este contiene y posteriormente relacionarlos para generar un modelo conceptual.
- Requisito 10: El sistema debe permitir al alumno el ejercicio cuando este lo crea conveniente.
- Requisito 11: El sistema debe permitir al alumno eliminar cualquier atributo o relación que haya señalado antes de dar por finalizado el ejercicio.
- Requisito 12: El sistema debe mostrar en todo momento los atributos y relaciones que el alumno considere necesarias para la realización del modelo conceptual.
- Requisito 13: El sistema debe mostrar por pantalla un modelo conceptual generado con la solución que el alumno ha proporcionado por pantalla.
- Requisito 14: El sistema debe dejar reintentar el ejercicio en caso de que quiera volver a realizarlo con los elementos del último intento guardados.
- Requisito 15: El sistema debe guardar cada elemento o respuesta que el alumno haya señalado en la realización del ejercicio.

9.5. Diseño de las interfaces externas

En esta sección se muestran los bocetos o wireframes que representan el diseño inicial propuesto para la aplicación:

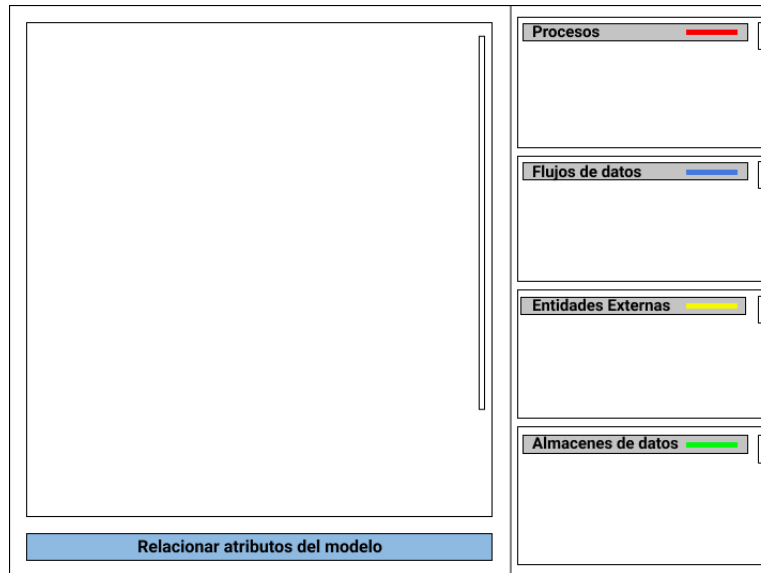


Figura 9.1: Pantalla principal y de selección de atributos del texto

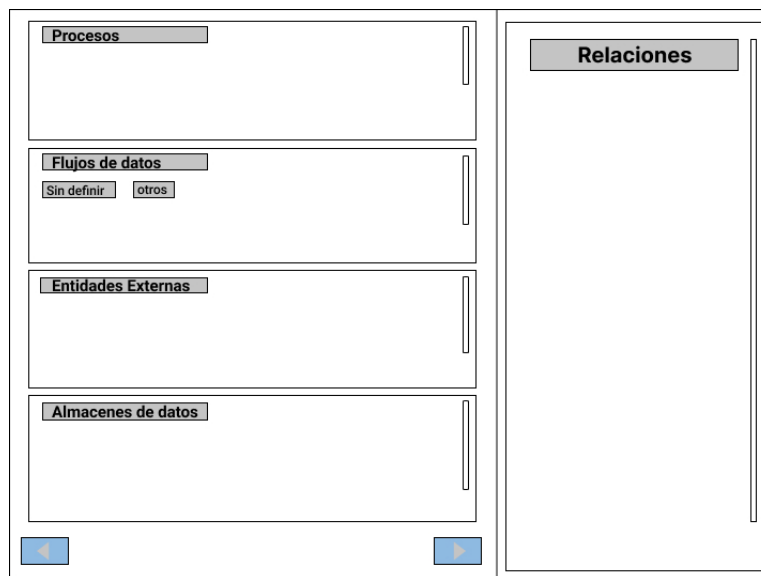


Figura 9.2: Pantalla de relacionar atributos

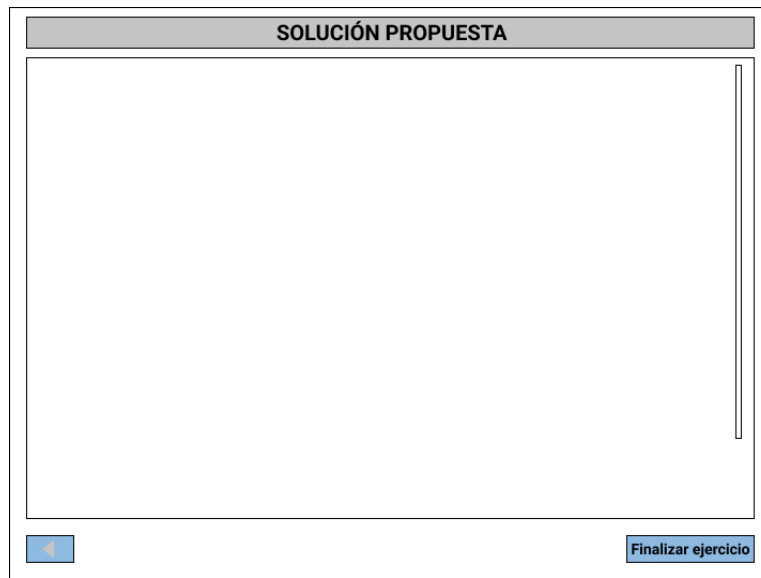


Figura 9.3: Pantalla que muestra un modelo conceptual construido en base a las relaciones propuestas en la anterior pantalla

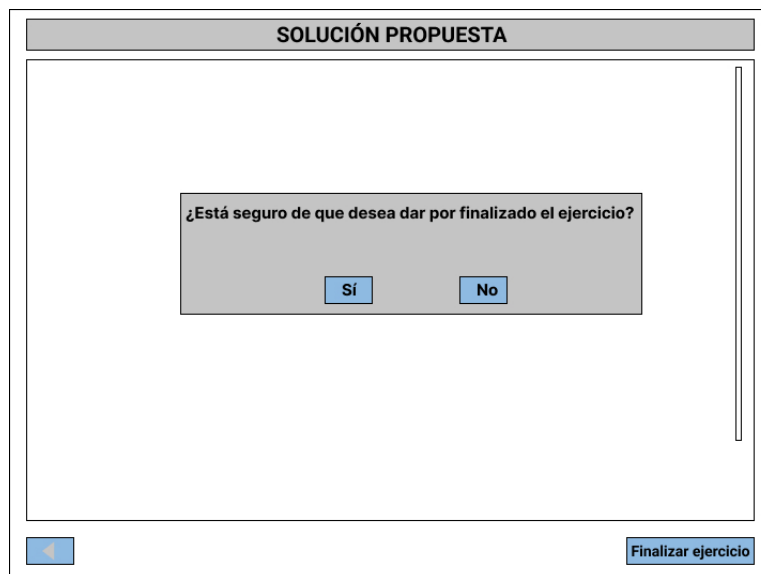


Figura 9.4: Selección de finalización de ejercicio

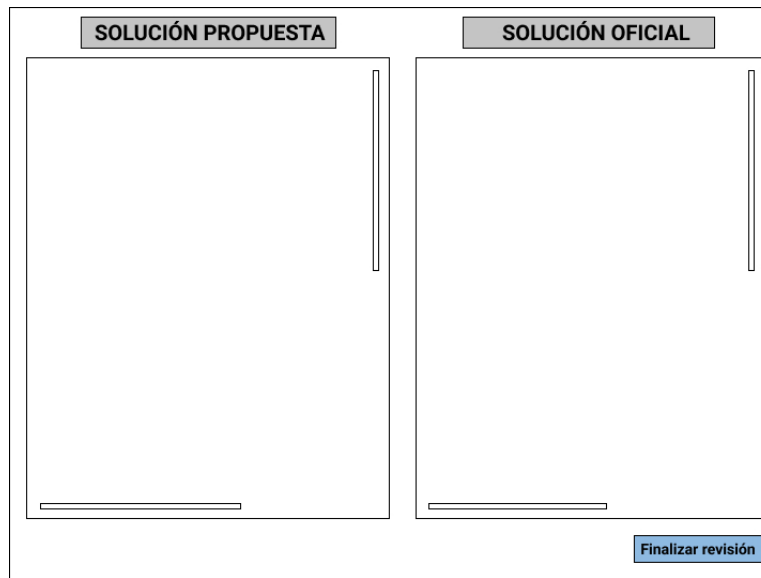


Figura 9.5: Pantalla de finalización de ejercicio con comparación de soluciones

9.6. Casos de uso

En este apartado se van a explicar los casos de uso del sistema:

9.6.1. Seleccionar atributos del texto según su tipo

Nombre	Seleccionar atributos del texto según su tipo
Autor	David Hernández Salor
Fecha	29/04/2022
Descripción	Permite al usuario seleccionar los atributos que considere necesarios para la realización del modelo conceptual
Actores	Usuario registrado en Moodle, en la asignatura en la cual se encuentre la actividad y deberá estar realizando el ejercicio
Precondiciones	El usuario debe haber iniciado sesión en la plataforma de Moodle y estar realizando el ejercicio propuesto
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la parte del texto que quiere considerar como un atributo 2. El usuario da click en el botón correspondiente al tipo de atributo que quiere darle al texto seleccionado
Postcondiciones	Los atributos seleccionados por el alumno que no hayan sido eliminados posteriormente se mantendrán guardados para seguir adelante con el ejercicio y guardar el progreso

9.6.2. Eliminar atributo seleccionado anteriormente

Nombre	Eliminar atributo seleccionado anteriormente
Autor	David Hernández Salor
Fecha	29/04/2022
Descripción	Permite al usuario eliminar los atributos que ha seleccionado previamente y ya no considere necesarios para la realización del modelo conceptual
Actores	Usuario registrado en Moodle, en la asignatura en la cual se encuentre la actividad y deberá estar realizando el ejercicio
Precondiciones	El usuario debe haber iniciado sesión en la plataforma de Moodle y estar realizando el ejercicio propuesto
Flujo Normal	1. El usuario selecciona el botón correspondiente al atributo que desea eliminar de su respuesta y se elimina automáticamente
Postcondiciones	El atributo eliminado se elimina también de los elementos guardados para el progreso del ejercicio

9.6.3. Paso a la pantalla de crear relaciones con los atributos seleccionados

Nombre	Paso a la pantalla de crear relaciones con los atributos seleccionados
Autor	David Hernández Salor
Fecha	29/04/2022
Descripción	Permite al usuario pasar de la pantalla de seleccionar los atributos que considere necesarios para la creación del modelo conceptual a la pantalla de crear relaciones
Actores	Usuario registrado en Moodle, en la asignatura en la cual se encuentre la actividad y deberá estar realizando el ejercicio
Precondiciones	El usuario debe haber iniciado sesión en la plataforma de Moodle y estar realizando el ejercicio propuesto
Flujo Normal	1. El usuario selecciona el botón llamado relacionar atributos” para pasar a la pantalla de relación de atributos
Postcondiciones	Los atributos seleccionados previamente han sido guardados y se muestran también en la pantalla de relación de atributos

9.6.4. Crear una relación con los atributos seleccionados

Nombre	Crear una relación con los atributos seleccionados
Autor	David Hernández Salor
Fecha	29/04/2022
Descripción	Permite al usuario crear relaciones a partir de los atributos seleccionados y guardados previamente
Actores	Usuario registrado en Moodle, en la asignatura en la cual se encuentre la actividad y deberá estar realizando el ejercicio
Precondiciones	El usuario debe haber iniciado sesión en la plataforma de Moodle y estar realizando el ejercicio propuesto
Flujo Normal	<ol style="list-style-type: none"> 1. Para el primer atributo de la relación el usuario selecciona un botón de atributo que no sea de tipo flujo 2. El usuario selecciona el atributo de tipo flujo que considere necesario para la relación que quiera crear 3. El usuario selecciona el último atributo de la relación de tipo distinto de flujo para crearla
Postcondiciones	Las relaciones seleccionadas por el alumno que no hayan sido eliminados posteriormente se mantendrán guardados para seguir adelante con el ejercicio y guardar el progreso

9.6.5. Eliminar relación creada anteriormente

Nombre	Eliminar relación creada anteriormente
Autor	David Hernández Salor
Fecha	29/04/2022
Descripción	Permite al usuario eliminar las relaciones que ha creado previamente y ya no considere necesarias para la realización del modelo conceptual
Actores	Usuario registrado en Moodle, en la asignatura en la cual se encuentre la actividad y deberá estar realizando el ejercicio
Precondiciones	El usuario debe haber iniciado sesión en la plataforma de Moodle y estar realizando el ejercicio propuesto
Flujo Normal	1. El usuario selecciona el botón correspondiente a la relación que desea eliminar de su respuesta y se elimina automáticamente
Postcondiciones	La relación eliminada se elimina también de los elementos guardados para el progreso del ejercicio

9.6.6. Paso a la pantalla de selección de atributos desde la pantalla de crear relaciones

Nombre	Paso a la pantalla de selección de atributos desde la pantalla de crear relaciones
Autor	David Hernández Salor
Fecha	29/04/2022
Descripción	Permite al usuario pasar de la pantalla de crear relaciones para el modelo conceptual que considere necesarias a la pantalla de selección de atributos
Actores	Usuario registrado en Moodle, en la asignatura en la cual se encuentre la actividad y deberá estar realizando el ejercicio
Precondiciones	El usuario debe haber iniciado sesión en la plataforma de Moodle y estar realizando el ejercicio propuesto
Flujo Normal	1. El usuario selecciona el botón llamado "pág anterior" para pasar a la pantalla de selección de atributos
Postcondiciones	Las relaciones seleccionadas por el alumno que no hayan sido eliminados posteriormente se mantendrán guardados para seguir adelante con el ejercicio y guardar el progreso

9.6.7. Paso a la pantalla de crear un modelo conceptual con las relaciones creadas

Nombre	Paso a la pantalla de crear un modelo conceptual con las relaciones creadas
Autor	David Hernández Salor
Fecha	29/04/2022
Descripción	Permite al usuario pasar de la pantalla de crear relaciones para el modelo conceptual que considere necesarias a la pantalla que muestra el modelo conceptual creado con las relaciones seleccionadas
Actores	Usuario registrado en Moodle, en la asignatura en la cual se encuentre la actividad y deberá estar realizando el ejercicio
Precondiciones	El usuario debe haber iniciado sesión en la plataforma de Moodle y estar realizando el ejercicio propuesto
Flujo Normal	1. El usuario selecciona el botón llamado "pág siguiente" para pasar a la pantalla de modelo conceptual
Postcondiciones	Se crea un modelo conceptual con las relaciones creadas previamente en la pantalla de crear relaciones

9.6.8. Paso a la pantalla de crear relaciones con los atributos seleccionados desde la pantalla de modelo conceptual

Nombre	Paso a la pantalla de crear relaciones con los atributos seleccionados desde la pantalla de modelo conceptual
Autor	David Hernández Salor
Fecha	29/04/2022
Descripción	Permite al usuario pasar de la pantalla de modelo conceptual que a la pantalla de crear relaciones
Actores	Usuario registrado en Moodle, en la asignatura en la cual se encuentre la actividad y deberá estar realizando el ejercicio
Precondiciones	El usuario debe haber iniciado sesión en la plataforma de Moodle y estar realizando el ejercicio propuesto
Flujo Normal	1. El usuario selecciona el botón llamado "página anterior" para pasar a la pantalla de relación de atributos
Postcondiciones	

9.6.9. Ver calificaciones del ejercicio

Nombre	Ver calificaciones del ejercicio
Autor	David Hernández Salor
Fecha	29/04/2022
Descripción	Permite al usuario pasar a la pantalla de calificaciones
Actores	Usuario registrado en Moodle, en la asignatura en la cual se encuentre la actividad y deberá haber acabado el ejercicio
Precondiciones	El usuario debe haber iniciado sesión en la plataforma de Moodle y haber realizado el ejercicio propuesto
Flujo Normal	1. El usuario selecciona el botón llamado "Terminar ejercicio" desde la pantalla de modelo conceptual para pasar a la pantalla de calificaciones
Postcondiciones	Todas las respuestas que el usuario ha seleccionado se guardan en el sistema, y aparecen las calificaciones correspondientes al número de aciertos de dichas respuestas, restando puntuación en los casos necesarios como tener respuestas de más en algún tipo de atributo o relación

9.6.10. Ver solución oficial del ejercicio

Nombre	Ver solución oficial del ejercicio
Autor	David Hernández Salor
Fecha	29/04/2022
Descripción	Permite al usuario pasar de la pantalla de solución del modelo conceptual
Actores	Usuario registrado en Moodle, en la asignatura en la cual se encuentre la actividad y deberá haber acabado el ejercicio
Precondiciones	El usuario debe haber iniciado sesión en la plataforma de Moodle y haber realizado el ejercicio propuesto
Flujo Normal	1. El usuario selecciona el botón llamado "Solución del modelo" desde la pantalla de calificaciones para pasar a la pantalla de solución del modelo conceptual
Postcondiciones	

9.6.11. Pasar a la pantalla de calificaciones del ejercicio desde la pantalla de solución oficial del modelo conceptual

Nombre	Pasar a la pantalla de calificaciones del ejercicio desde la pantalla de solución oficial del modelo conceptual
Autor	David Hernández Salor
Fecha	29/04/2022
Descripción	Permite al usuario pasar de la pantalla de calificaciones
Actores	Usuario registrado en Moodle, en la asignatura en la cual se encuentre la actividad y deberá haber acabado el ejercicio
Precondiciones	El usuario debe haber iniciado sesión en la plataforma de Moodle y haber realizado el ejercicio propuesto
Flujo Normal	1. El usuario selecciona el botón llamado "Calificaciones" desde la pantalla de solución oficial del modelo conceptual para pasar a la pantalla de calificaciones
Postcondiciones	

9.6.12. Reintentar ejercicio

Nombre	Reintentar ejercicio
Autor	David Hernández Salor
Fecha	29/04/2022
Descripción	Permite al usuario reintentar el ejercicio
Actores	Usuario registrado en Moodle, en la asignatura en la cual se encuentre la actividad y deberá haber acabado el ejercicio
Precondiciones	El usuario debe haber iniciado sesión en la plataforma de Moodle y haber realizado el ejercicio propuesto
Flujo Normal	1. El usuario selecciona el botón llamado "Reintentar ejercicio" desde la pantalla de calificaciones
Postcondiciones	Las respuestas seleccionadas anteriormente seguirán estando disponibles por pantalla para su posterior actualización

Este documento esta firmado por



Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
Fecha/Hora	Thu Jun 30 23:54:46 CEST 2022
Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
Numero de Serie	561
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)