

# UniversidadeVigo

Diseño y desarrollo de una funcionalidad de búsqueda y  
filtrado para un catálogo de cursos

Aarón Riveiro Vilar

Trabajo Fin de Grado  
Escuela de Ingeniería de Telecomunicación  
Grado en Ingeniería de Tecnologías de Telecomunicación

Tutores:  
Manuel Caeiro Rodríguez

# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Contexto del proyecto . . . . .	1
1.2. Motivación . . . . .	1
<b>2. Objetivos y requisitos</b>	<b>2</b>
2.1. Requisitos funcionales . . . . .	2
2.1.1. Sistema de búsqueda . . . . .	2
2.1.2. Sistema de filtrado . . . . .	3
2.1.3. Integración y rendimiento . . . . .	3
2.2. Requisitos no funcionales . . . . .	4
<b>3. Diseño</b>	<b>4</b>
<b>4. Desarrollo</b>	<b>4</b>
<b>5. Conclusiones</b>	<b>4</b>
<b>6. Referencias</b>	<b>4</b>
<b>A. Anexos</b>	<b>5</b>

# 1. Introducción

En los últimos años, el crecimiento sostenido de los sistemas de información digitales ha dado lugar a repositorios con volúmenes de datos cada vez mayores y más heterogéneos. En este contexto, la mera disponibilidad de la información ya no resulta suficiente: es necesario dotar a los sistemas de mecanismos eficaces que permitan localizar, filtrar y presentar los datos de forma relevante para el usuario. La disciplina de la recuperación de información (Information Retrieval) estudia precisamente los métodos y modelos que permiten acceder de manera eficiente a grandes colecciones de documentos, maximizando la relevancia de los resultados ofrecidos frente a una consulta determinada [1].

A medida que aumenta el tamaño y la complejidad de los repositorios, la experiencia de usuario pasa a depender en gran medida de la calidad del sistema de búsqueda y filtrado. No solo es importante la rapidez de respuesta, sino también la precisión y exhaustividad de los resultados, así como la capacidad del sistema para interpretar la intención de búsqueda del usuario y ofrecer mecanismos de refinamiento progresivo mediante filtros, facetas o criterios adicionales [2]. Un sistema de búsqueda limitado a coincidencias literales sobre un único campo de texto puede resultar insuficiente en entornos donde los datos presentan múltiples dimensiones y atributos relevantes.

Por ello, el diseño e implementación de estrategias avanzadas de búsqueda y filtrado constituye un elemento clave en cualquier sistema de información orientado al usuario final, especialmente cuando se persigue facilitar el acceso eficiente a contenidos estructurados y mejorar la usabilidad global de la plataforma.

## 1.1. Contexto del proyecto

El proyecto DACEM (Digitising Academic Catalogues for Enhanced Mobility) [3], aprobado en la convocatoria 2023 del programa Erasmus+ para Asociaciones Estratégicas en Educación Superior (código de proyecto 2023-1-ES01-KA220-HED-000160344), tiene como finalidad el desarrollo de una plataforma de catálogo de cursos dirigida a instituciones educativas europeas. Su objetivo principal es ofrecer un sistema que permita publicar y mantener información actualizada sobre titulaciones y asignaturas de forma accesible, estructurada e interoperable, facilitando así los procesos de movilidad académica internacional.

El sistema de información asociado a dicha plataforma se fundamenta en tecnologías de software libre y está concebido para poder desplegarse tanto en entornos locales como en infraestructuras en la nube. Más allá del almacenamiento estructurado de datos académicos, el correcto funcionamiento del sistema depende de su capacidad para gestionar, organizar y recuperar la información de forma eficiente. En este contexto, los mecanismos de búsqueda y filtrado desempeñan un papel esencial, ya que constituyen el principal medio de interacción entre el usuario y el repositorio de cursos. La calidad de estos mecanismos condiciona directamente la usabilidad del sistema, la precisión de los resultados obtenidos y la experiencia global de navegación.

## 1.2. Motivación

Uno de los retos fundamentales en los sistemas de información que gestionan grandes volúmenes de datos estructurados no reside únicamente en su almacenamiento, sino en la forma en que estos datos son recuperados y presentados al usuario. A medida que un repositorio crece en número de elementos y en complejidad de atributos, los mecanismos de búsqueda simples basados en coincidencias textuales directas pueden resultar insuficientes para ofrecer resultados relevantes y ajustados a las necesidades reales de consulta.

Esta problemática es especialmente significativa en entornos donde los datos presentan múltiples dimensiones descriptivas —como titulaciones, asignaturas, áreas de conocimiento, idiomas, créditos o modalidades— y donde el usuario puede tener criterios de búsqueda diversos y combinados. La ausencia de mecanismos avanzados de filtrado y refinamiento progresivo limita la capacidad del sistema para adaptarse a diferentes perfiles de usuario

y escenarios de consulta, reduciendo la eficiencia en la localización de información pertinente.

En el ámbito actual del desarrollo de aplicaciones web y sistemas de gestión de contenidos, las tendencias apuntan hacia la implementación de motores de búsqueda más sofisticados, capaces de indexar múltiples campos, ponderar resultados según relevancia, incorporar criterios de filtrado dinámico y mejorar la experiencia de usuario mediante interfaces intuitivas. Estas soluciones no solo optimizan el acceso a la información, sino que contribuyen a una mayor usabilidad y aprovechamiento del sistema en su conjunto.

El presente Trabajo Fin de Grado se enmarca en este contexto, proponiendo el diseño y desarrollo de una mejora en los mecanismos de búsqueda y filtrado del catálogo de cursos de la plataforma DACEM. El objetivo es evolucionar desde un sistema de coincidencia textual básica hacia una solución más flexible y precisa, alineada con las buenas prácticas actuales en recuperación de información y desarrollo web, empleando tecnologías abiertas e integradas en el entorno existente.

## **2. Objetivos y requisitos**

El presente trabajo se centra en la mejora de los mecanismos de búsqueda y filtrado del catálogo de cursos desarrollado en el marco del proyecto DACEM. A diferencia de otros desarrollos orientados a la recopilación o estructuración de datos, este proyecto aborda la fase de recuperación y presentación de la información, poniendo el foco en la interacción entre el usuario y el sistema.

El objetivo principal consiste en analizar las limitaciones del sistema actual de consulta e implementar una solución que permita mejorar la precisión, flexibilidad y usabilidad de los resultados obtenidos. Para ello, se definen a continuación los requisitos funcionales y no funcionales que deben guiar el diseño e implementación de la propuesta.

Se entiende por requisito funcional aquel que describe las capacidades concretas que debe ofrecer el sistema, mientras que los requisitos no funcionales establecen condiciones relativas a su comportamiento, rendimiento, integración y mantenibilidad, sin definir directamente nuevas funcionalidades.

### **2.1. Requisitos funcionales**

#### **2.1.1. Sistema de búsqueda**

1. El sistema debe permitir la realización de búsquedas textuales sobre el catálogo de cursos a partir de una cadena de consulta introducida por el usuario.
2. La búsqueda no debe limitarse exclusivamente al título del curso, sino que deberá contemplar múltiples campos relevantes del modelo de datos, tales como descripción, área de conocimiento, titulación asociada u otros atributos significativos que formen parte de la información estructurada del catálogo.
3. El sistema debe soportar coincidencias parciales y no estrictamente literales, permitiendo la localización de resultados aun cuando la consulta no coincida exactamente con el texto almacenado. En este sentido, se deberá contemplar la normalización de mayúsculas y minúsculas y, en su caso, la gestión de caracteres acentuados.
4. El sistema deberá ordenar los resultados de búsqueda según un criterio de relevancia, de manera que los elementos más pertinentes en relación con la consulta aparezcan en las primeras posiciones.
5. El sistema debe garantizar que las búsquedas con consultas vacías o ambiguas no provoquen errores ni comportamientos inesperados, definiendo un comportamiento consistente en estos casos (por ejemplo, mostrando todos los resultados o solicitando una consulta válida).

## **2. Objetivos y requisitos**

---

6. El sistema deberá integrarse con el modelo de datos existente sin alterar su estructura fundamental, reutilizando en la medida de lo posible los mecanismos de indexación y consulta proporcionados por la plataforma.
7. El sistema debe permitir la extensión futura del mecanismo de búsqueda, de forma que puedan incorporarse nuevos campos o criterios sin requerir una reestructuración completa de la solución implementada.

### **2.1.2. Sistema de filtrado**

1. El sistema debe permitir la aplicación de filtros sobre el conjunto de resultados obtenidos, posibilitando la restricción de los cursos mostrados en función de distintos atributos del modelo de datos.
2. Los filtros deberán contemplar, al menos, aquellos campos estructurados que resulten relevantes para la consulta de los usuarios, tales como titulación, área de conocimiento, idioma de impartición, número de créditos, modalidad u otros atributos disponibles en el catálogo.
3. El sistema debe permitir la combinación simultánea de múltiples filtros, de manera que el usuario pueda refinar progresivamente los resultados mediante criterios acumulativos.
4. La aplicación o modificación de filtros no deberá generar inconsistencias en los resultados mostrados, garantizando que el conjunto final de cursos cumpla todos los criterios activos en cada momento.
5. El sistema debe actualizar los resultados de forma coherente tras la activación o desactivación de filtros, manteniendo una experiencia de usuario fluida y evitando recargas innecesarias que afecten negativamente a la interacción.
6. El sistema debe permitir que los filtros sean dinámicos y adaptativos al conjunto de datos disponible, evitando mostrar opciones de filtrado que no correspondan con resultados existentes.
7. El mecanismo de filtrado deberá integrarse de forma consistente con el sistema de búsqueda, permitiendo que ambos procesos actúen de manera complementaria, de forma que la consulta textual y los criterios estructurados puedan aplicarse conjuntamente.
8. El diseño del sistema de filtrado debe contemplar la posibilidad de ampliación futura, permitiendo la incorporación de nuevos criterios sin necesidad de rediseñar la arquitectura principal.

### **2.1.3. Integración y rendimiento**

1. La solución propuesta debe integrarse con la plataforma existente sin alterar la estructura fundamental del modelo de datos ni comprometer el funcionamiento del resto de funcionalidades del sistema.
2. El sistema debe ser compatible con la arquitectura actual basada en Drupal y con los módulos utilizados para la gestión del contenido, garantizando que la mejora del mecanismo de búsqueda y filtrado no requiera una reimplementación completa del entorno.
3. La implementación debe reutilizar, en la medida de lo posible, los mecanismos de indexación y consulta disponibles en la plataforma, o bien extenderlos de forma controlada cuando resulte necesario.
4. El sistema debe garantizar tiempos de respuesta adecuados en las consultas realizadas por el usuario, de modo que la experiencia de uso no se vea degradada por demoras excesivas en la generación de resultados.

5. El rendimiento del sistema debe mantenerse estable ante un incremento razonable del volumen de datos almacenados en el catálogo, asegurando que la escalabilidad de la solución permita su evolución futura.
6. El sistema debe soportar la ejecución concurrente de múltiples consultas sin producir errores, bloqueos o degradaciones significativas del rendimiento.
7. La solución implementada deberá minimizar el impacto en los recursos del servidor, evitando sobrecargas innecesarias y optimizando las consultas realizadas a la base de datos.

## **2.2. Requisitos no funcionales**

1. La solución debe estar fundamentada en tecnologías de uso libre y código abierto, manteniendo coherencia con los principios y la arquitectura general del proyecto DACEM.
2. El desarrollo debe garantizar la mantenibilidad del código, estructurando la implementación de forma modular y documentada para facilitar futuras modificaciones o ampliaciones.
3. La solución debe ser compatible con futuras actualizaciones de la plataforma y de los módulos utilizados, evitando dependencias que comprometan la evolución del sistema.
4. El sistema debe garantizar un comportamiento estable y predecible ante entradas incorrectas, consultas mal formadas o combinaciones de filtros no válidas, evitando errores visibles para el usuario final.
5. La interfaz de búsqueda y filtrado debe cumplir criterios básicos de usabilidad, asegurando que los mecanismos de interacción resulten intuitivos y coherentes con el diseño general de la plataforma.
6. La solución debe preservar la integridad de los datos existentes, garantizando que las mejoras introducidas en la capa de consulta no alteren la información almacenada ni su consistencia.
7. El diseño del sistema deberá contemplar su posible reutilización o adaptación en otros contextos similares, favoreciendo la portabilidad y la extensibilidad de la solución implementada.

## **3. Diseño**

## **4. Desarrollo**

## **5. Conclusiones**

## **6. Referencias**

- [1] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [2] “Retrieval accuracy,” <https://www.sciencedirect.com/topics/computer-science/retrieval-accuracy>.
- [3] “Digitising academic catalogues for enhanced mobility,” <https://projects.uni-foundation.eu/dacem/>, European University Foundation, 07/2024.

## **A. Anexos**