** F\_AA\_234A**

**VICERRECTORADO DE DOCENCIA**

**PLAN DE TRABAJO DE INTEGRACIÓN CURRICULAR**

|  |  |  |
| --- | --- | --- |
| **I. INFORMACIÓN BÁSICA** | | |
| **Unidad Académica:** | Facultad de Ingeniería de Sistemas | |
| **Carrera:** | Ingeniería de Software | |
| **Proyecto:** | Implementación de una aplicación para gestión de un SLR | |
| **Componente:** | Módulo de Descarga de Metadatos y Documentos | |
| **Línea de investigación:** | Ingeniería de Software | |
| **Nombres y apellidos del estudiante:** | Alexis Jahir Lapo Cabrera | |
| **Nombres y apellidos del Profesor:** | Carlos Anchundia, PhD. | |
| **II. DESCRIPCIÓN DEL COMPONENTE** | | |
| El Módulo de Descarga es la respuesta técnica y centralizada a uno de los mayores cuellos de botella en las Revisiones Sistemáticas de Literatura (SLR): la recuperación de estudios primarios. Este proceso, tradicionalmente ejecutado de forma manual, descentralizada y desestructurada, conduce a una significativa pérdida de tiempo y a una baja calidad en la producción de resultados fiables. Nuestro componente ataca directamente este problema al funcionar como un motor de adquisición de datos automatizado, diseñado para interactuar de manera sistemática y eficiente con las principales bases de datos académicas.  Para lograr su objetivo, el módulo implementa una arquitectura de backend robusta con tres elementos clave. Primero, utiliza conectores de API especializados para interactuar directamente con motores de búsqueda fundamentales como IEEE Xplore y Scopus, asegurando un acceso fiable a la información. Segundo, incorpora un parser de consultas, capaz de traducir cadenas de búsqueda booleanas a la sintaxis nativa de cada motor, garantizando que las búsquedas sean precisas y consistentes. Finalmente, gestiona las descargas a través de un sistema de colas asíncronas, lo que le permite procesar grandes volúmenes de peticiones en paralelo sin comprometer el rendimiento del sistema y asegurando la interoperabilidad mediante la exportación de metadatos a formatos estándar como BibTeX y RIS.  El impacto de este componente es triple y fundamental para el rigor metodológico de una SLR. En primer lugar, aporta eficiencia, al automatizar una tarea que consume semanas de trabajo manual. En segundo lugar, garantiza la trazabilidad completa del proceso de búsqueda, creando un registro auditable que es esencial para la reproducibilidad del estudio. Y, lo más importante, eleva la calidad y el rigor científico de la revisión al asegurar que la fase de recuperación de literatura sea sistemática, exhaustiva y estandarizada, alineándose con las directrices metodológicas más exigentes del ámbito científico. | | |
| **III. OBJETIVOS** | | |
| **Objetivo General:**  **Desarrollar e integrar un servicio de backend robusto y** para la recuperación automatizada de literatura científica, que se integre directamente con las APIs de IEEE Xplore y Scopus, con el fin de optimizar la eficiencia, garantizar la trazabilidad y elevar el rigor metodológico del proceso de SLR.  **Objetivos Específicos:**   * Implementar conectores de API funcionales y tolerantes a fallos para los motores de búsqueda IEEE Xplore y Scopus. * Desarrollar un parser de consultas que traduzca una cadena de búsqueda booleana genérica a la sintaxis nativa de cada API soportada. * Construir un sistema de gestión de descargas asíncronas mediante una cola de trabajos, capaz de manejar peticiones masivas y optimizar el uso de las cuotas de las APIs. * Asegurar la exportación de metadatos en formatos estándar (BibTeX/RIS) para garantizar la interoperabilidad con herramientas externas. * **Validar la funcionalidad y fiabilidad del módulo** mediante pruebas de integración continuas y la simulación de escenarios de uso basados en Revisiones Sistemáticas de Literatura reales. | | |
| **IV. ALCANCE DEL COMPONENTE** | | |
| El alcance de este componente se define por la implementación de un servicio de backend especializado en la adquisición de datos académicos. Su responsabilidad se inicia al recibir una cadena de búsqueda compleja desde el Módulo de Diseño y concluye con la entrega de metadatos estructurados al Módulo de Selección. Para ello, se desarrollarán conectores de API específicos para interactuar con las bases de datos de IEEE Xplore y Scopus. El núcleo funcional del módulo residirá en un parser capaz de interpretar operadores booleanos y filtros de búsqueda para traducirlos a la sintaxis nativa de cada API, asegurando la precisión en la ejecución de las consultas.  La funcionalidad del componente garantiza la extracción de un conjunto definido de metadatos, incluyendo DOI, Título, Resumen, y Palabras Clave. Toda la información recolectada será normalizada y estructurada para permitir su exportación en formatos estándar de la industria, específicamente BibTeX y RIS, asegurando la plena compatibilidad con los gestores de referencias más comunes en el ámbito de la investigación. La comunicación con otros módulos del sistema se realizará a través de una API REST interna que expondrá los servicios del componente.  Y finalmente para gestionar las solicitudes de manera eficiente y escalable, el alcance incluye la implementación de un sistema asíncrono basado en una cola de trabajos. Esta arquitectura permitirá el procesamiento en paralelo de las descargas de documentos de acceso abierto, optimizando los tiempos de respuesta y manejando grandes volúmenes de peticiones sin afectar el rendimiento general de la aplicación. | | |
| **V. TAREAS ESPECÍFICAS** | | |
| |  |  |  |  | | --- | --- | --- | --- | | **Semana referencial /**  **Fases** | **Fecha inicio-fin (si aplica)** | **Tareas específicas** | **Resultado esperado (si aplica)** | | **1-2** | 23/09/2025 - 06/10/2025 | Setup del Entorno y Definición de Escenarios: Configurar el repositorio, dependencias y pipeline de CI/CD. Escribir los escenarios BDD (.feature) para las funcionalidades críticas de búsqueda y exportación. | Entorno de desarrollo 100% funcional. Archivo search.feature con sus respectivos escenarios clave (Given-When-Then). Pipeline de pruebas automatizadas que se ejecuta exitosamente. | | **3-6** | 07/10/2025 - 03/11/2025 | Implementar y Probar los Conectores API: Desarrollar los clientes para Scopus e IEEE Xplore. Implementar manejo robusto de errores (4xx, 5xx) y de límites de cuota (throttling). | Módulos de los conectores funcionales y probados con BDD, con una tasa de éxito en peticiones de prueba y manejo explícito de errores. | | **7** | 04/11/2025 - 10/11/2025 | Implementar el Traductor de Cadenas Booleanas: Desarrollar la lógica que convierte las consultas genéricas a la sintaxis nativa de cada API. | Parser funcional con su documentación técnica interna y un set de pruebas unitarias que cubran casos lógicos (operadores AND/OR/NOT). | | **8-9** | 11/11/2025 - 24/11/2025 | Integrar Celery y Redis para Descargas Paralelas: Configurar el sistema de colas y desarrollar los "workers" que procesarán las tareas de descarga de documentos de acceso abierto. | Sistema de colas funcional, capaz de recibir, procesar y reportar el estado de las tareas, con una capacidad de procesar un lote de 100 tareas de descarga. | | 10 | 25/11/2025 - 01/12/2025 | Generar Archivos BibTeX/RIS: Implementar el módulo que toma los metadatos normalizados y los convierte a los formatos estándar. | Módulo de exportación validado, cuyos archivos generados se importan exitosamente sin errores. | | 11-13 | 02/12/2025 - 22/12/2025 | Ejecutar Pruebas End-to-End (E2E): Probar el flujo completo desde la recepción de una consulta hasta la exportación de resultados, usando datos de una SLR real o simulada. |  | | 14-15 | 23/12/2025 - 05/01/2026 | Redactar Guía Técnica y Refactorizar Código: Documentar la API interna del módulo con Swagger/OpenAPI. Realizar un code review completo y aplicar mejoras de calidad y rendimiento. | Documentación final de la API. Código fuente con una buena calificación de calidad en herramientas de análisis estático (ej. SonarQube, pylint). | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | | | |
| **VI. BIBLIOGRAFÍA** | | |
| **Kitchenham, B. (2007). *Guidelines for SLR in Software Engineering*. EBSE.** | | |
| **VII. FIRMAS DE RESPONSABILIDAD** | | |
| **Desarrollado por:** | | Alexis Jahir Lapo Cabrera |
| **Aprobado por:** | |  |