



Modelo de Análisis

InformAPI

INGENIERÍA DE SOFTWARE

Elaborado por:

- ♦ Chávez Cruz Adolfo
- ♦ Ramos Velasco Gabriel Antonio
- ♦ Sánchez Ortega Gabriel

Grupo: 6CV4

Semestre: 25-1

Introducción y visión general del sistema

El Sistema de Búsqueda y Recomendación de Artículos Científicos (InformAPI) es una aplicación web desarrollada con Spring Boot que permite a los usuarios buscar, visualizar y gestionar artículos científicos. Este sistema proporciona una interfaz para la búsqueda de información académica, con funcionalidades de autenticación, gestión de usuarios, y recomendaciones personalizadas.

La visión del sistema es facilitar el acceso a la literatura científica mediante una plataforma que centraliza las búsquedas, permite filtrar resultados según diversos criterios, y ofrece recomendaciones basadas en el historial y preferencias del usuario. El sistema está diseñado para atender las necesidades tanto de investigadores como de estudiantes y profesionales que requieran acceso a información académica actualizada.

InformAPI integra la API de Scopus para la búsqueda de artículos científicos, ofreciendo acceso a una amplia base de datos de publicaciones académicas. Además, el sistema mantiene una gestión robusta de usuarios con diferentes roles y permisos.

Glosario de términos

- **API:** Application Programming Interface. Conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar software.
- **Spring Boot:** Marco de trabajo (framework) para el desarrollo de aplicaciones Java basado en Spring.
- **Scopus API:** Interfaz de programación proporcionada por Elsevier que permite acceder programáticamente a la base de datos Scopus de artículos científicos.
- **CRUD:** Create, Read, Update, Delete. Operaciones básicas para la gestión de datos.
- **REST:** Representational State Transfer. Estilo de arquitectura de software para sistemas distribuidos.
- **PostgreSQL:** Sistema de gestión de bases de datos relacional.

- **Docker:** Plataforma de contenedores que permite empaquetar aplicaciones con todas sus dependencias.
- **Bootstrap:** Framework de diseño para desarrollar interfaces web responsivas.
- **Thymeleaf:** Motor de plantillas Java para aplicaciones web.
- **JPA:** Java Persistence API. API de persistencia para mapear objetos Java a bases de datos relacionales.
- **HTML:** HyperText Markup Language. Lenguaje de marcado para la elaboración de páginas web.
- **CSS:** Cascading Style Sheets. Lenguaje de diseño gráfico para definir la presentación de documentos.
- **JavaScript:** Lenguaje de programación interpretado que se ejecuta en el lado del cliente.
- **MVC:** Model-View-Controller. Patrón de arquitectura de software que separa los datos de la lógica y la interfaz.

Descripción de los usuarios del sistema

El sistema contempla dos tipos principales de usuarios:

1. Usuario Regular (ROLE_USER):

- Puede registrarse y crear una cuenta en el sistema.
- Puede iniciar sesión en el sistema.
- Puede buscar artículos científicos utilizando la interfaz de búsqueda simple o avanzada.
- Puede ver la información detallada de los artículos encontrados.
- Puede editar su perfil personal (nombre, correo, contraseña).
- Puede activar/desactivar el modo oscuro para la interfaz.
- Puede acceder a recomendaciones personalizadas basadas en su historial y favoritos.

2. Administrador (ROLE_ADMIN):

- Tiene todas las capacidades de un usuario regular.
- Puede ver la lista completa de usuarios registrados en el sistema.

- Puede editar la información de cualquier usuario (nombre, correo, contraseña).
- Puede eliminar usuarios del sistema.
- Puede acceder al panel de administración.

Restricciones y suposiciones

Restricciones técnicas:

- El sistema utiliza Java 21 como lenguaje de programación.
- La aplicación está desarrollada con Spring Boot 3.4.3.
- Se utiliza PostgreSQL como sistema de gestión de bases de datos.
- La interfaz de usuario está desarrollada con Bootstrap y Thymeleaf.
- El sistema debe ser desplegable mediante Docker para facilitar la instalación.

Restricciones operativas:

- El sistema requiere conexión a internet para acceder a la API de Scopus.
- Se requiere una clave de API válida para Scopus.
- El sistema debe estar disponible 24/7 con mantenimientos programados fuera de horas pico.

Suposiciones:

- Los usuarios tienen conocimientos básicos de informática y búsqueda en internet.
- Los usuarios tienen acceso a un dispositivo con navegador web moderno.
- La API de Scopus mantendrá su estructura y disponibilidad actual.
- Los usuarios proporcionarán información válida durante el registro.
- El volumen inicial de usuarios será moderado, permitiendo escalabilidad gradual.

Diagrama de Casos de Uso:

El sistema contempla cinco casos de uso principales que representan las funcionalidades clave:

1. **Inicio de Sesión:** Permite a los usuarios autenticarse en el sistema.
2. **Gestión de Usuarios (Admin):** Permite a los administradores gestionar los usuarios del sistema.

3. **Gestión de Usuarios (Usuario):** Permite a los usuarios gestionar su propio perfil.
4. **Búsqueda de Artículos:** Permite a los usuarios buscar artículos científicos.
5. **Visualización de Artículos:** Permite a los usuarios ver información detallada de artículos.
6. **Recomendación Personalizada:** Ofrece recomendaciones basadas en el historial y favoritos del usuario.

Establecimiento de las relaciones entre casos de uso

Relaciones entre casos de uso:

1. **<<include>>:** Relación que indica que un caso de uso base incluye la funcionalidad de otro caso de uso incluido.
 - "Ingresar Usuario y contraseña" incluye "Validar datos" en el caso de uso de Inicio de Sesión.
 - "Gestión de Usuarios" incluye "Inicio de Sesión" tanto para administradores como para usuarios normales.
 - "Seleccionar Artículo" incluye "Mostrar lista de artículos" en el caso de uso de Visualización.
 - "Mostrar artículos que coincidan" incluye "Recuperar lista de favoritos" y "Recuperar historial" en el caso de Recomendación.
2. **<<extend>>:** Relación que indica que un caso de uso base puede ser extendido con la funcionalidad de otro caso de uso.
 - "Manejar error de validación" extiende "Validar datos" en el caso de uso de Inicio de Sesión.
 - "Creación/Edición/Eliminación de usuario" extiende "Gestión de Usuarios" tanto para administradores como para usuarios normales.
 - "Retornar lista de coincidencias" y "Manejar mensaje de no coincidencias" extienden "Consulta base de datos(API)" en el caso de Búsqueda.
 - "Guardar registro del artículo como favorito" extiende "Mostrar información del artículo" en el caso de Visualización.

Definición de los límites del sistema

Límites físicos:

- El sistema está diseñado para funcionar como una aplicación web accesible desde cualquier navegador moderno.
- La aplicación puede ser desplegada en servidores locales o en la nube.
- El sistema requiere una base de datos PostgreSQL para almacenar información de usuarios y gestión.

Límites funcionales:

- La búsqueda de artículos está limitada a la API de Scopus y sus capacidades.
- El sistema no almacena los artículos completos, solo referencias y metadatos.
- La aplicación no procesa pagos ni gestiona suscripciones a artículos de pago.
- No incluye funcionalidades de red social o mensajería entre usuarios.

Límites de seguridad:

- Acceso restringido mediante autenticación de usuarios.
- Separación clara de roles y permisos (usuario/administrador).
- Contraseñas almacenadas con encriptación.
- Protección CSRF para formularios y peticiones.

Documentación de Casos de Uso:

Caso de uso: Inicio de Sesión

ID y nombre: CU-01 - Inicio de Sesión

Descripción y propósito: Permite a los usuarios autenticarse en el sistema mediante credenciales (nombre de usuario y contraseña) para acceder a las funcionalidades correspondientes a su rol.

Actores involucrados: Usuario (cualquier usuario registrado, incluyendo administradores)

Precondiciones:

- El usuario debe estar registrado en el sistema.
- El usuario debe tener sus credenciales (nombre de usuario y contraseña).

Postcondiciones:

- El usuario queda autenticado en el sistema.
- El sistema redirige al usuario a la página principal correspondiente a su rol.

Flujo principal:

1. El usuario accede a la página de inicio de sesión.
2. El usuario ingresa su nombre de usuario y contraseña.
3. El sistema valida las credenciales.
4. Si las credenciales son válidas, el sistema autentica al usuario.
5. El sistema redirige al usuario a la página principal según su rol.

Flujos alternativos:

- Si las credenciales son inválidas, el sistema muestra un mensaje de error y permite al usuario intentar nuevamente.
- Si el usuario olvida su contraseña, el sistema ofrece una opción para recuperarla.

Requisitos especiales:

- Las contraseñas deben almacenarse encriptadas en la base de datos.
- El sistema debe implementar medidas contra ataques de fuerza bruta.
- La sesión debe expirar después de un periodo de inactividad.

Caso de uso: Búsqueda de Artículos

ID y nombre: CU-02 - Búsqueda de Artículos

Descripción y propósito: Permite a los usuarios buscar artículos científicos en la base de datos de Scopus utilizando diversos criterios y filtros.

Actores involucrados: Usuario (cualquier usuario autenticado)

Precondiciones:

- El usuario debe estar autenticado en el sistema.
- El sistema debe tener conexión con la API de Scopus.

Postcondiciones:

- El sistema muestra los resultados de la búsqueda al usuario.
- La búsqueda se registra en el historial del usuario.

Flujo principal:

1. El usuario accede a la página de búsqueda.
2. El usuario ingresa parámetros de búsqueda (términos, filtros).

3. El sistema envía la consulta a la API de Scopus.
4. La API retorna los resultados de la búsqueda.
5. El sistema muestra los resultados al usuario de forma ordenada y paginada.

Flujos alternativos:

- Si no hay resultados, el sistema muestra un mensaje indicando que no se encontraron coincidencias.
- Si la conexión con la API falla, el sistema muestra un mensaje de error.
- El usuario puede refinar su búsqueda utilizando filtros adicionales.

Requisitos especiales:

- La interfaz debe permitir búsquedas simples y avanzadas.
- El sistema debe implementar paginación para manejar grandes cantidades de resultados.
- Las búsquedas deben ser rápidas y eficientes.

Caso de uso: Gestión de Usuarios (Admin)

ID y nombre: CU-03 - Gestión de Usuarios (Admin)

Descripción y propósito: Permite a los administradores gestionar los usuarios del sistema, incluyendo creación, edición y eliminación de cuentas.

Actores involucrados: Administrador

Precondiciones:

- El usuario debe estar autenticado con rol de administrador.

Postcondiciones:

- Los cambios realizados quedan registrados en la base de datos.

Flujo principal:

1. El administrador accede al panel de administración.
2. El sistema muestra la lista de usuarios registrados.
3. El administrador selecciona la acción a realizar (crear, editar o eliminar usuario).
4. El sistema ejecuta la acción solicitada.
5. El sistema muestra un mensaje de confirmación.

Flujos alternativos:

- Si hay errores al procesar la acción, el sistema muestra un mensaje de error.

- El administrador puede cancelar la acción en cualquier momento.

Requisitos especiales:

- El administrador no debe poder eliminar su propia cuenta.
- El sistema debe solicitar confirmación antes de eliminar un usuario.
- El sistema debe validar la información ingresada durante la creación o edición de usuarios.

Caso de uso: Visualización de Artículos

ID y nombre: CU-04 - Visualización de Artículos

Descripción y propósito: Permite a los usuarios ver información detallada de artículos científicos y acceder a recursos relacionados.

Actores involucrados: Usuario (cualquier usuario autenticado)

Precondiciones:

- El usuario debe estar autenticado en el sistema.
- El usuario debe haber seleccionado un artículo de la lista de resultados de búsqueda.

Postcondiciones:

- El sistema registra el artículo en el historial del usuario.
- El usuario puede añadir el artículo a favoritos.

Flujo principal:

1. El usuario selecciona un artículo de la lista de resultados.
2. El sistema muestra la información detallada del artículo.
3. El sistema registra automáticamente el artículo en el historial del usuario.
4. El usuario tiene la opción de añadir el artículo a favoritos.

Flujos alternativos:

- Si el artículo ya está en favoritos, el usuario puede removerlo.
- El usuario puede acceder al artículo completo si está disponible mediante un enlace externo.

Requisitos especiales:

- La interfaz debe ser clara y mostrar toda la información relevante del artículo.
- El sistema debe indicar si el artículo es de acceso abierto o requiere suscripción.

- El sistema debe mostrar métricas del artículo (número de citas, factor de impacto).

Caso de uso: Recomendación Personalizada

ID y nombre: CU-05 - Recomendación Personalizada

Descripción y propósito: Ofrece al usuario recomendaciones de artículos basadas en su historial de visualización y artículos favoritos.

Actores involucrados: Usuario (cualquier usuario autenticado)

Precondiciones:

- El usuario debe estar autenticado en el sistema.
- El usuario debe tener historial de búsqueda o artículos favoritos.

Postcondiciones:

- El sistema muestra una lista de artículos recomendados al usuario.

Flujo principal:

1. El usuario accede a la sección de recomendaciones.
2. El sistema recupera el historial y favoritos del usuario.
3. El sistema analiza los datos para identificar patrones y preferencias.
4. El sistema muestra artículos que coinciden con los intereses del usuario.

Flujos alternativos:

- Si el usuario no tiene suficiente historial, el sistema muestra recomendaciones basadas en tendencias generales.
- El usuario puede refinar las recomendaciones indicando temas de interés adicionales.

Requisitos especiales:

- El algoritmo de recomendación debe ser efectivo y relevante.
- El sistema debe actualizar las recomendaciones periódicamente.
- Las recomendaciones deben incluir artículos recientes y relevantes.

Priorización:

Caso de uso principal a implementar:

CU-02 - Búsqueda de Artículos ha sido seleccionado como el caso de uso principal a implementar en esta práctica.

Justificación de la selección:

La funcionalidad de Búsqueda de Artículos ha sido seleccionada como caso de uso principal para implementación prioritaria basándose en un análisis integral de su impacto en el sistema. Constituye el núcleo funcional de la plataforma, ofreciendo beneficio inmediato y tangible al permitir a los usuarios localizar información científica específica y relevante, representando así la propuesta de valor esencial del sistema. Su implementación requiere integración con servicios externos como la API de Scopus, diseño de algoritmos de consulta sofisticados y desarrollo de interfaces intuitivas; este nivel de complejidad representa un desafío técnico significativo que, sin embargo, resulta abordable con los recursos disponibles y establece un marco tecnológico robusto para el proyecto.

La búsqueda funciona como componente habilitador para otros casos de uso críticos, particularmente la visualización detallada de artículos y el sistema de recomendaciones personalizadas. Su implementación temprana establece los cimientos necesarios para el desarrollo coherente de estas funcionalidades dependientes, creando una secuencia lógica en el proceso de desarrollo. Además, define patrones arquitectónicos cruciales para el manejo de datos externos, presentación de información jerarquizada e interacción con el usuario, que serán reutilizados consistentemente en otros módulos del sistema, garantizando coherencia y mantenibilidad en toda la aplicación.

Finalmente, la búsqueda de artículos ofrece resultados inmediatos y altamente visuales que permiten demostrar eficazmente el valor del sistema a usuarios y stakeholders desde etapas iniciales del desarrollo. Esta característica facilita ciclos de retroalimentación tempranos y posibilita realizar ajustes estratégicos antes de avanzar con funcionalidades más complejas, optimizando así los recursos de desarrollo y aumentando las probabilidades de satisfacción del usuario final.