

# Requerimientos FURPS+ para un Sistema de Búsqueda y Recomendación de Artículos Científicos Basado en APIs

---

## Propósito:

El propósito de este sistema es proporcionar una plataforma que permita a los usuarios buscar, visualizar y recibir recomendaciones de artículos científicos. Esto se logrará integrando APIs abiertas de acceso a datos académicos y desarrollando servicios REST con formatos JSON para ofrecer endpoints que permitan consumir y compartir la información.

---

## Requerimientos FURPS+ Funcionales (F)

### 1. Búsqueda de artículos:

- Permitir a los usuarios buscar artículos por palabras clave, autor, año de publicación o temas específicos.
- Implementar filtros avanzados, como idioma, revista o tipo de artículo.

### 2. Visualización de artículos:

- Mostrar información relevante de los artículos:
  - Título, autores, resumen, año, y enlace al documento completo si está disponible.
- Proveer un visor embebido para documentos en acceso abierto (PDF).

### 3. Recomendación personalizada:

- Ofrecer recomendaciones basadas en:
  - Historial de búsquedas.
  - Artículos marcados como favoritos.
  - Tendencias generales en el sistema.

### 4. Gestión de usuarios:

- Registro y autenticación con soporte para inicio de sesión con Google y otros servicios.
- Gestión de perfiles: permitir a los usuarios personalizar preferencias y guardar búsquedas.

### 5. Historial y favoritos:

- Registrar el historial de búsquedas de cada usuario.
- Permitir marcar artículos como favoritos para una fácil consulta posterior.

### 6. Consumo de APIs públicas:

- Integrar APIs como CrossRef o Semantic Scholar para obtener metadatos de artículos académicos.
- Procesar respuestas en formato JSON y almacenarlas para uso posterior.

### 7. Sistema de endpoints REST:

- Exponer información del sistema mediante endpoints que:
  - Devuelvan artículos filtrados según criterios de búsqueda.
  - Provean el historial y favoritos del usuario.
  - Permitan a terceros integrar funcionalidades mediante la API.

### 8. Cumplimiento ético y legal:

- Respetar las licencias y derechos de autor de los artículos procesados.
- Asegurar que los datos compartidos cumplan con normativas como GDPR y leyes locales de protección de datos.

---

## **Usabilidad (U)**

### **1. Interfaz amigable:**

- Diseño limpio y minimalista, centrado en facilitar el acceso rápido a búsquedas y recomendaciones.
- Implementación responsiva para que el sistema funcione en dispositivos móviles y escritorio.

### **2. Accesibilidad:**

- Compatibilidad con lectores de pantalla y navegación por teclado.
- Opciones de personalización de texto (tamaño y contraste).

### **3. Experiencia de usuario:**

- Incluir tutoriales interactivos para usuarios nuevos.
- Proveer mensajes de error claros y específicos.

### **4. Prototipos de interfaces:**

- Crear prototipos funcionales que muestren las principales funcionalidades del sistema, siguiendo un enfoque de diseño centrado en el usuario.

---

## **Confiabilidad (R)**

### **1. Disponibilidad:**

- Garantizar un tiempo de actividad del 99.9% mediante infraestructura en la nube.

### **2. Gestión de errores:**

- Manejo robusto de fallos en las solicitudes a APIs externas y recuperación automática.
- Validación exhaustiva de datos de entrada para evitar errores en el sistema.

### **3. Seguridad:**

- Implementar autenticación segura con tokens y cifrado de datos sensibles.
- Proteger los endpoints contra ataques como inyección SQL y ataques de fuerza bruta.

---

## **Rendimiento (P)**

### **1. Tiempos de respuesta:**

- Consultas a la base de datos y generación de recomendaciones en menos de 3 segundos.
- Búsquedas en APIs externas procesadas en menos de 5 segundos.

### **2. Escalabilidad:**

- Soportar al menos 10,000 usuarios concurrentes, con capacidad de expansión según la demanda.

### **3. Eficiencia:**

- Optimizar el consumo de recursos mediante técnicas de caché y procesamiento distribuido.

---

## **Soporte (S)**

### **1. Mantenibilidad:**

- Documentación técnica completa para desarrolladores, con ejemplos de uso de los endpoints REST.
- Código modular y comentado para facilitar futuras actualizaciones.

## **2. Compatibilidad:**

- Integración con APIs estándar de datos académicos (CrossRef, CORE, Semantic Scholar).
- Exportación de datos a formatos JSON y CSV para análisis externo.

## **3. Actualizaciones y soporte técnico:**

- Ciclos regulares de mantenimiento y mejoras basadas en el feedback del usuario.

---

### **Extensiones (+)**

#### **1. Cumplimiento ético y legal:**

- Implementar mecanismos para obtener y respetar las licencias de uso de las fuentes de datos.
- Asegurar que los datos compartidos estén alineados con normativas internacionales de privacidad y derechos de autor.

#### **2. Análisis avanzado:**

- Implementar algoritmos de aprendizaje automático para mejorar las recomendaciones.
- Analizar tendencias en el uso del sistema para adaptar las funcionalidades.

#### **3. Exportación de datos:**

- Permitir a los usuarios exportar resultados de búsqueda y listas de favoritos para integrarlas en otros sistemas.

#### **4. Integración con ontologías:**

- Usar bases de conocimiento como DBpedia o Wikidata para enriquecer los datos mostrados.