Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ciencias de la Computación e Informática
Bachillerato en Ingeniería en Computación

Bases de Datos Avanzadas

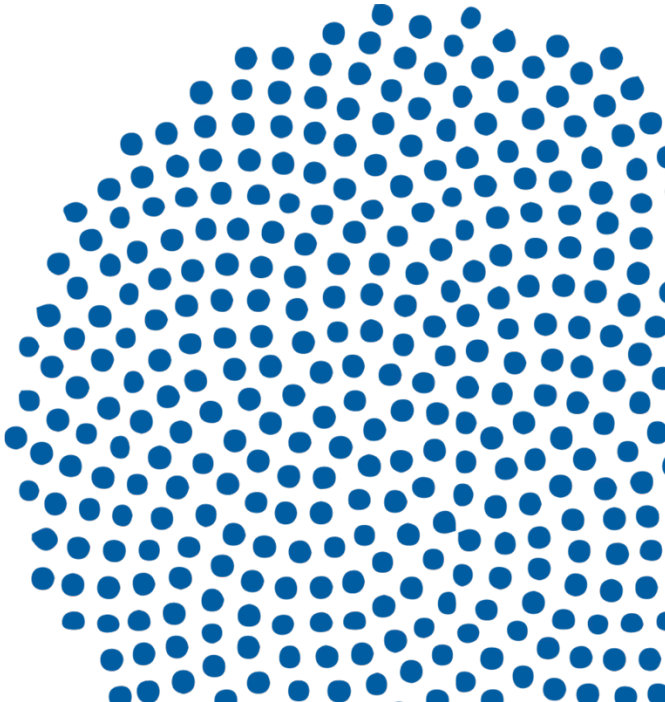
CI-0141

Especificación del Proyecto Programado

Profesor:

Ing. Sleyter Angulo Chavarría, M.Sc.

II Ciclo 2025



Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ciencias de la Computación e Informática
Bachillerato en Ingeniería en Computación

Sistema de Búsqueda Distribuido: Motor de Indexación y Consulta de Información en la Web.

Generalidades

- Se trabajará en los grupos definidos para tal fin.
- El fraude académico será severamente castigado. Por ejemplo, un trabajo de copy and paste de internet o utilizando alguna herramienta de inteligencia artificial (ejemplo: ChatGPT, entre otras), sin su respectiva referencia, se considera plagio y por ende fraude.

Objetivo General

Diseñar e implementar un **sistema distribuido de bases de datos** que permita **almacenar**, **procesar** y **consultar información obtenida desde la web**, aplicando los principios de **fragmentación, replicación, distribución de consultas y tolerancia a fallos.**

Objetivos Específicos

1. Implementar un modelo de base de datos distribuida entre múltiples nodos (motores distintos o instancias separadas).
2. Diseñar un módulo de extracción de información (crawler) para recolectar datos desde páginas web.
3. Procesar y almacenar los datos en distintas bases de datos distribuidas (por ejemplo, MySQL, MongoDB, PostgreSQL).
4. Desarrollar una interfaz gráfica de usuario (GUI) que permita realizar consultas de búsqueda.
5. Evaluar la eficiencia y consistencia de las consultas distribuidas.
6. Implementar estrategias de replicación y partición de datos para garantizar disponibilidad.

Descripción General del Proyecto

El sistema funcionará como un buscador académico o temático, que permite realizar consultas sobre un conjunto de páginas web indexadas (por ejemplo, sitios de noticias, universidades o blogs técnicos).
La arquitectura se basará en múltiples bases de datos distribuidas en diferentes nodos o motores.

Componentes principales:

1. **Módulo de Crawler (Extracción de datos):**
   - Recorre un conjunto de URLs predefinidas.
   - Extrae título, descripción, palabras clave y enlace.
   - Almacena los resultados temporalmente en una cola (por ejemplo, RabbitMQ o Kafka).
2. **Módulo de Almacenamiento Distribuido:**
   - Los datos se fragmentan o replican entre distintos motores:
     - **Nodo 1:** MySQL (metadatos).
     - **Nodo 2:** MongoDB (texto y contenido).
     - **Nodo 3:** PostgreSQL (índices de búsqueda).
3. **Módulo de Consulta:**
   - Recibe una palabra clave del usuario.
   - Distribuye la consulta a los diferentes nodos.
   - Combina los resultados y los presenta al usuario en la interfaz.
4. **Interfaz Gráfica (GUI o Web App):**
   - Permite al usuario realizar búsquedas.
   - Muestra los resultados en forma de lista con título, enlace y descripción.
   - Puede desarrollarse con **Flask, Django o React + FastAPI**.

Tecnologías recomendadas

La tabla 1 muestra información de tecnologías sugeridas para realizar cada uno de los componentes para el proyecto.

Categoría	Opciones sugeridas
Lenguaje principal	Python / Java / Node.js
Motores de bases de datos	MySQL, PostgreSQL, MongoDB
Middleware / Mensajería	RabbitMQ / Kafka (opcional)
Framework para interfaz	Flask, Django, o React
APIs de búsqueda	BeautifulSoup, Requests, Scrapy
Orquestación	Docker Compose (para simular nodos distribuidos)

Tabla 1: Opciones de Tecnologías

Requisitos Funcionales

- El sistema debe permitir registrar y consultar información de al menos **500 páginas web**.
- El usuario podrá realizar búsquedas mediante palabras clave.
- Las consultas deben ejecutarse sobre **distintos nodos de bases de datos**.
- Los resultados deben combinarse y mostrarse en **tiempo real** o bajo demanda.
- El sistema debe tolerar la caída de al menos un nodo sin perder la disponibilidad completa.

Requisitos No Funcionales

- **Disponibilidad:** alta mediante replicación de datos.
- **Escalabilidad:** debe poder añadir nodos de base de datos sin modificar el sistema central.
- **Consistencia:** los datos deben mantenerse sincronizados según la estrategia elegida.
- **Seguridad:** proteger las consultas y accesos mediante autenticación básica.

Criterio de evaluación

La tabla 2 muestra los criterios de evaluación para la realización de este proyecto.

Criterio	Ponderación
Diseño de arquitectura distribuida	20%
Implementación de fragmentación / replicación	20%
Eficiencia de consultas distribuidas	15%
Interfaz de usuario funcional	15%
Documentación y diagrama ER / topología	15%
Presentación y defensa del proyecto	15%

Tabla 2: Evaluación

Notas importantes

- Fecha de entrega: 10 de noviembre.

Herramientas

- Repositorio de Git: Bitbucket, GitHub, Gitlab, Git ECCI.
- Manejador de tareas: Jira, GitHub Project, etc.
- IDE: PyCharm, Visual Studio Code.
- Prototipado: Figma.

Enunciado de honor

El trabajo realizado en este, será el resultado de mi esfuerzo. No usaré, recibiré, ni ofreceré ayuda no autorizada. No copiare de otros proyectos, no utilizaré código publicado en internet ni permitiré que nadie copie parte alguna de este proyecto. No realizaré ninguna trampa ni procedimiento deshonesto en la realización de este trabajo.
Al hacer entrega de cualquier parte de este proyecto lo hago bajo fe de juramento de cumplir con el presente enunciado de honor.