**Universidad de Las Américas**

Facultad de Ingenierías y Ciencias Agropecuarias

*Ingeniería de Softwate*

**Informe de laboratorio**

1. **DATOS DEL ALUMNO: Dylan Clerque**
2. **TEMA DE LA PRÁCTICA: Ejercicio de Caché**
3. **OBJETIVO DE LA PRÁCITCA**

Optimizar el acceso a la base de datos al implementar un mecanismo de caché que almacene los resultados de las consultas frecuentes, reduciendo así el número de accesos a la base de datos y mejorando el rendimiento general de la aplicación.

1. **OBJETIVOS ESPECÍFICOS**

1. Desarrollar e integrar una estrategia de caché que sea rápida y eficiente para almacenar los resultados de consultas a la base de datos que se utilizan con frecuencia, minimizando los tiempos de respuesta de la aplicación.

2. Asegurarse de que el acceso a la caché esté controlado y seguro, evitando inconsistencias en los datos y garantizando la coherencia entre la base de datos y el caché.

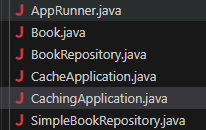
3. Diseñar un mecanismo para invalidar los datos del caché cuando estos se actualizan en la base de datos, asegurando que la aplicación trabaje siempre con datos precisos y actualizados. Esto puede incluir estrategias como el establecimiento de un tiempo de vida (TTL) para los datos en caché o la invalidación automática basada en eventos de cambios en la base de datos.

1. **MATERIALES Y MÉTODOS**

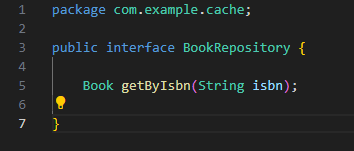
* Visual Studio Code
* Maven
* Spring

1. **DESARROLLO DE LA PRÁCTICA Y RESULTADOS**

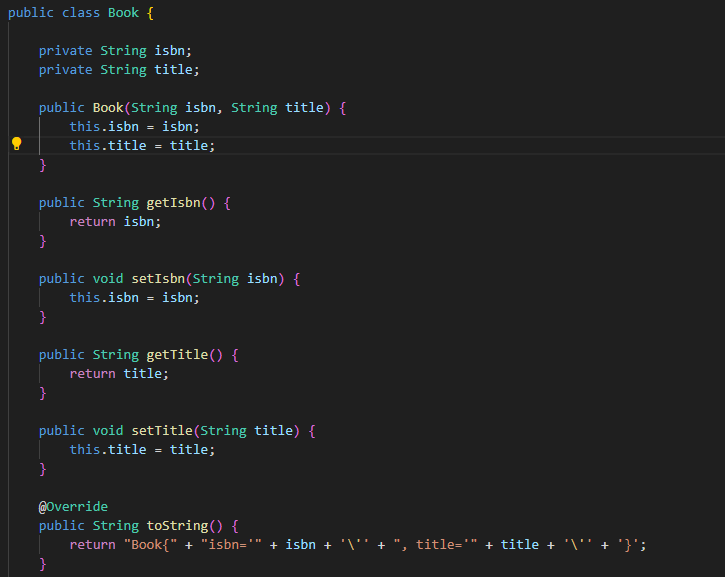
**Crear estas clases:**

****

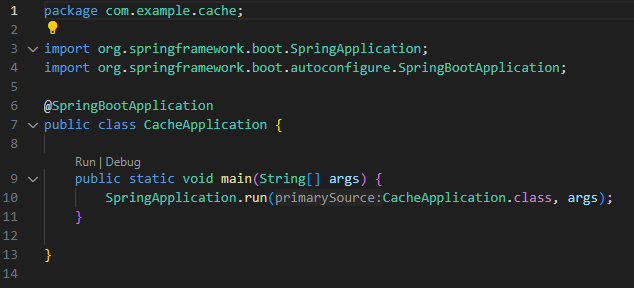
**Clase BookRepository:**

****

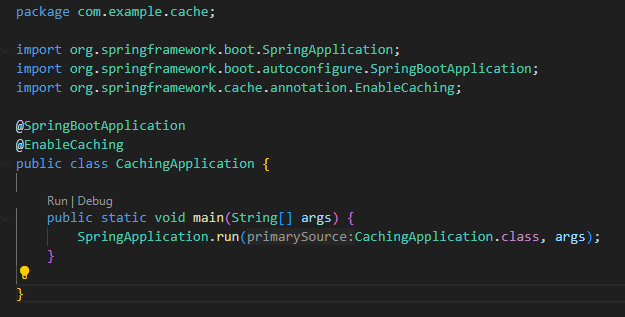
**Clase BookJava:**

****

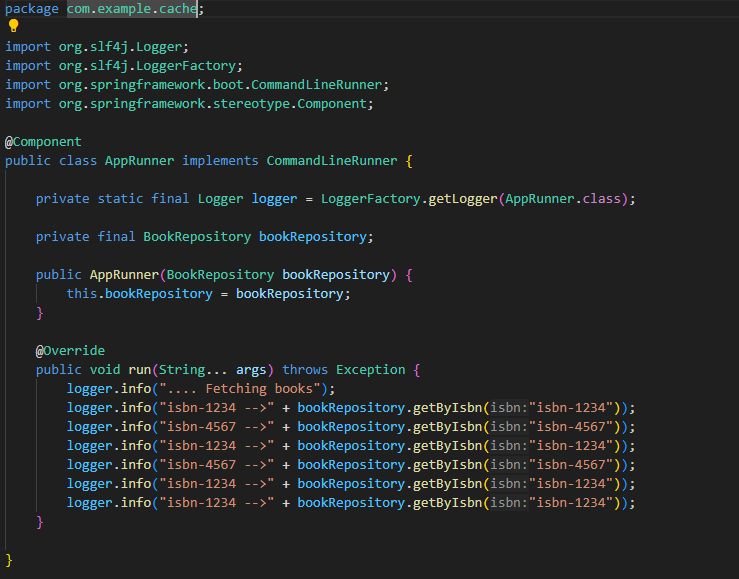
**Clase CacheApplication:**

****

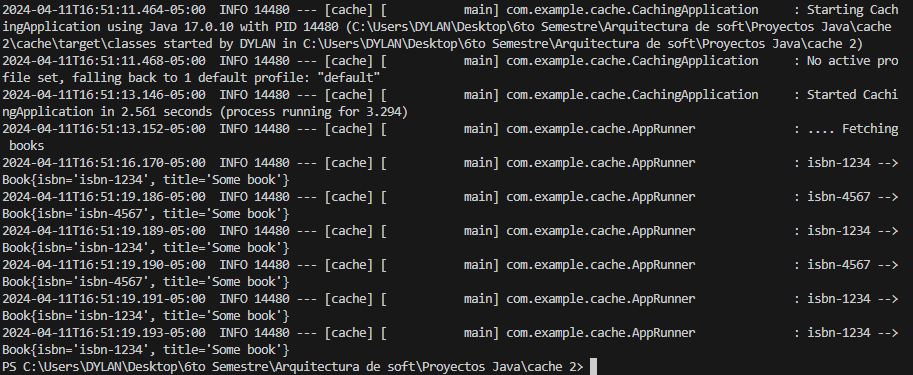
**Caching Application:**

****

**Clase App Runner:**

****

**Resultado**

****

1. **OPINIÓN PERSONAL**

1. Guardar en caché los resultados de las consultas a la base de datos puede mejorar considerablemente el rendimiento de la aplicación. Al reducir la cantidad de consultas a la base de datos, los tiempos de respuesta de la aplicación se reducen y los usuarios experimentan un servicio más rápido y fluido.

2. Si bien el uso del caché puede ser beneficioso, también puede añadir complejidad en la gestión de la coherencia de los datos. Mantener la sincronización entre la base de datos y el caché puede ser un desafío, especialmente cuando hay cambios frecuentes en los datos subyacentes.

3. Elegir la estrategia de caché adecuada es fundamental para el éxito de esta actividad. Deben tenerse en cuenta aspectos como el tamaño del caché, la política de reemplazo, la duración de los datos en caché (TTL), entre otros, para optimizar el uso de la memoria y evitar un rendimiento ineficiente.

4. Una mala gestión del caché puede llevar a una sobrecarga de la memoria si se almacenan demasiados datos o si los datos permanecen en caché durante demasiado tiempo. Es importante establecer límites claros sobre qué y cuánto se guarda en caché para evitar problemas de rendimiento o posibles caídas de la aplicación.

1. **ANEXOS (Ai se requiere)**

Git hub



<https://github.com/DDCT2003/Cache/tree/main/cache%202/cache/src/main/java/com/example/cache>

1. **BIBLIOGRAFÍA (Si se requiere)**

*Getting Started | Caching Data with Spring*. (n.d.). Getting Started | Caching Data With Spring. https://spring.io/guides/gs/caching