➢ Parte II.I

¿Cómo se podría modificar la implementación para minimizar el número de consultas en estos casos?

RTA: Se puede mejorar incorporando un mecanismo de parada temprana coordinada entre los hilos, en lugar de que cada hilo recorra completamente su segmento asignado, todos los hilos comparten un contador global de ocurrencias encontradas usando una variable atómica la AtomicInteger foundReports para que las operaciones sobre ella se ejecuten de forma indivisible.

Cuando este contador alcanza el número mínimo requerido de listas negras que son 5, entonces los hilos dejan de realizar nuevas consultas, aunque no hayan terminado de recorrer su segmento completo.

```java
import lombok.Getter;
import lombok.RequiredArgsConstructor;

import java.util.LinkedList;
import java.util.List;
import java.util.concurrent.atomic.AtomicInteger;

@Getter
@RequiredArgsConstructor
public class BlackListSearchThread  extends Thread{
    private final int start;
    private final int end;
    private final String ip;
    private final List<Integer> occurrences = new LinkedList<>();
    private int count = 0;

    private final AtomicInteger foundReports;
    private final int reportLimit;

    @Override
    public void run() {
        HostBlacklistsDataSourceFacade skds =
                HostBlacklistsDataSourceFacade.getInstance();

        for (int k = start; k < end && foundReports.get() < reportLimit; k++) {
            count++;
            if (skds.isInBlackListServer(k, ip)) {
                occurrences.add(k);
                foundReports.incrementAndGet();
            }
        }
    }

}
```

Nota: En este punto cambiamos stream por un buble for porque permite evaluar de forma inmediata una condición compartida entre múltiples hilos y detener la búsqueda tan pronto se alcanza el número mínimo de reportes, algo que stream con takeWhile no garantiza en escenarios concurrentes.

```java
public class HostBlackListsValidator {
    public List<Integer> checkHost(String ipaddress, int N){
        LinkedList<Integer> blackListOcurrences=new LinkedList<>();

        int ocurrencesCount=0;

        HostBlacklistsDataSourceFacade skds=HostBlacklistsDataSourceFacade.getInstance();

        int checkedListsCount=0;

        int totalServers = skds.getRegisteredServersCount();

        if (N <= 0) N = 1;
        if (N > totalServers) N = totalServers;

        int segmentSize = totalServers / N;

        AtomicInteger foundReports = new AtomicInteger(initialValue: 0);

        List<BlackListSearchThread> threads = new LinkedList<>();

        for (int i = 0; i < N; i++) {
            int start = i * segmentSize;
            int end = (i == N - 1)
                    ? totalServers
                    : start + segmentSize;

            threads.add(new BlackListSearchThread(start, end, ipaddress, foundReports, BLACK_LIST_ALARM_COUNT));
        }

        for (BlackListSearchThread thread : threads) {
            thread.start();
        }

        for (BlackListSearchThread t : threads) {
            try {
```

¿Qué elemento nuevo trae esto al problema?

RTA: El principal problema sería que todos los hilos dependen de un recurso común el foundReports e introduciría mayor complejidad del control de concurrencia.

Salida: La salida muestra que, usando 100 hilos que la búsqueda se detuvo anticipadamente al encontrar 5 reportes, revisando solo 58.746 de 80.000 listas, lo que confirma que el corte temprano reduce consultas y mejora el tiempo de ejecución 1176 ms.

```
INFO: Checked Black Lists:58.746 of 80.000
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
INFO: Found reports total: 5
Threads: 100 | Time: 1176 ms | Blacklists found: 5
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
INFO: Checked Black Lists:58.746 of 80.000
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
INFO: Found reports total: 5
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
INFO: Checked Black Lists:58.746 of 80.000
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
INFO: Found reports total: 5
Threads: 100 | Time: 1176 ms | Blacklists found: 5
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
INFO: Checked Black Lists:58.746 of 80.000
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
INFO: Found reports total: 5
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
INFO: Checked Black Lists:58.746 of 80.000
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
INFO: Found reports total: 5
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
INFO: Checked Black Lists:58.746 of 80.000
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
INFO: Checked Black Lists:58.746 of 80.000
ene. 29, 2026 6:35:05 P.ÃM. edu.eci.arsw.blacklistvalidator.HostBlackListsValidator checkHost
INFO: Found reports total: 5
Threads: 100 | Time: 1176 ms | Blacklists found: 5
PS C:\Users\robin\Downloads\Lab1_ARSW>
```