INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE COMPUTO

INGENIERÍA DE SOFTWARE

TAREA 1

FUNDAMENTOS DE INGENIERÍA DE SOFTWARE

NOMBRE DEL ALUMNO: GARCÍA QUIROZ GUSTAVO IVAN

BOLETA: 2022630278

GRUPO: 6CV3

NOMBRE DEL PROFESOR: HURTADO AVILÉS GABRIEL

FECHA DE ENTREGA: 29/09/2024

Índice

Introducción

Este reporte se explica la implementación de una API REST utilizando Spring Boot para calcular la cerradura de Kleene y la cerradura positiva de conjuntos de cadenas binarias. El proyecto se desarrolló como parte de una práctica para familiarizarse con herramientas comunes en el desarrollo de software y aplicar conceptos de teoría de lenguajes formales.

El proyecto se estructura en tres componentes principales: un servicio, un controlador y una interfaz web.

1. Servicio (CerraduraService): Implementa dos métodos: calcularCerraduraEstrella() y calcularCerraduraPositiva(). Utiliza un algoritmo recursivo para generar todas las cadenas binarias hasta una longitud máxima n. La cerradura de Kleene incluye la cadena vacía, mientras que la positiva no.
2. Controlador (CerraduraController): Define dos endpoints REST que aceptan solicitudes PUT. Recibe un parámetro 'n' que determina la longitud máxima de las cadenas binarias. Invoca los métodos correspondientes del servicio y devuelve los resultados.
3. Interfaz web (index.html): Proporciona un formulario simple para ingresar el valor de 'n'. Utiliza JavaScript para realizar solicitudes a la API y mostrar los resultados.

La lógica central reside en el método generarCadenasBinarias, que construye recursivamente todas las combinaciones posibles de '0' y '1' hasta la longitud especificada. Este enfoque garantiza la generación de todas las cadenas binarias requeridas para las cerraduras de Kleene y positiva.

Desarrollo

CerraduraController.java

```java
package com.cerraduras.cerraduras.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import com.cerraduras.cerraduras.service.CerraduraService;
import java.util.Set;

@RestController
@RequestMapping("/api/cerradura")
@CrossOrigin(origins = "*")  // Permite solicitudes de cualquier origen
public class CerraduraController {

    @Autowired
    private CerraduraService cerraduraService;

    @PutMapping("/estrella")
    public Set<String> cerraduraEstrella(@RequestParam int n) {
        return cerraduraService.calcularCerraduraEstrella(n);
    }

    @PutMapping("/positiva")
    public Set<String> cerraduraPositiva(@RequestParam int n) {
        return cerraduraService.calcularCerraduraPositiva(n);
    }
}
```

Figura 1 CerraduraController.java

Este controlador maneja las solicitudes HTTP para la API REST. Define dos endpoints que responden a solicitudes PUT: uno para la cerradura de Kleene (/api/cerradura/estrella) y otro para la cerradura positiva (/api/cerradura/positiva). Ambos endpoints aceptan un parámetro n que especifica la longitud máxima de las cadenas binarias a generar. El controlador utiliza la anotación @Autowired para inyectar el CerraduraService y delegar el cálculo de las cerraduras. La anotación @CrossOrigin permite solicitudes desde cualquier origen, facilitando la interacción con la interfaz web.

CerraduraService.java

```java
import org.springframework.stereotype.Service;
import java.util.HashSet;
import java.util.Set;

@Service
public class CerraduraService {

    public Set<String> calcularCerraduraEstrella(int n) {
        Set<String> resultado = new HashSet<>();
        resultado.add(e:"");  // λ (cadena vacía)

        for (int i = 1; i <= n; i++) {
            generarCadenasBinarias(prefijo:"", i, resultado);
        }

        return resultado;
    }

    public Set<String> calcularCerraduraPositiva(int n) {
        Set<String> resultado = new HashSet<>();

        for (int i = 1; i <= n; i++) {
            generarCadenasBinarias(prefijo:"", i, resultado);
        }

        return resultado;
    }

    private void generarCadenasBinarias(String prefijo, int longitud, Set<String> resultado) {
        if (longitud == 0) {
            resultado.add(prefijo);
            return;
        }

        generarCadenasBinarias(prefijo + "0", longitud - 1, resultado);
        generarCadenasBinarias(prefijo + "1", longitud - 1, resultado);
    }
```

Figura 2 CerraduraService.java

Este archivo contiene la lógica principal para calcular las cerraduras. Define dos métodos públicos: calcularCerraduraEstrella y calcularCerraduraPositiva. Ambos utilizan un método privado generarCadenasBinarias, que implementa un algoritmo recursivo para generar todas las cadenas binarias hasta una longitud máxima n. La diferencia clave entre los dos métodos públicos es que la cerradura de Kleene incluye la cadena vacía, mientras que la positiva no. El servicio utiliza un Set para almacenar las cadenas únicas generadas, evitando duplicados eficientemente.

Index.html

```html
<html lang="es">
<body>
    <h1>Calculadora de Cerraduras</h1>
    <form id="cerraduraForm">
        <label for="n">Longitud máxima (n):</label>
        <input type="number" id="n" name="n" required min="1" max="10">
        <br>
        <button type="button" onclick="calcularCerradura('estrella')">Calcular Cerradura de Kleene</button>
        <button type="button" onclick="calcularCerradura('positiva')">Calcular Cerradura Positiva</button>
    </form>
    <div id="resultado"></div>

    <script>
        async function calcularCerradura(tipo) {
            const n = document.getElementById('n').value;
            const resultadoDiv = document.getElementById('resultado');
            resultadoDiv.innerHTML = 'Calculando...';

            try {
                const response = await axios.put(`/api/cerradura/${tipo}?n=${n}`);
                const conjuntoCadenas = response.data.join(', ');
                resultadoDiv.innerHTML = `Cerradura ${tipo === 'estrella' ? 'de Kleene' : 'Positiva'} (n=${n}):\n${conjuntoCadenas}`;
            } catch (error) {
                resultadoDiv.innerHTML = `Error: ${error.message}`;
            }
        }
    </script>
</body>
</html>
```

Figura 3 Index.html

Este archivo proporciona una interfaz de usuario simple para interactuar con la API. Contiene un formulario HTML con un campo de entrada para el valor de n y dos botones para calcular las cerraduras de Kleene y positiva. Utiliza JavaScript para manejar las interacciones del usuario y realizar solicitudes asíncronas a la API mediante la biblioteca Axios. Los resultados de las solicitudes se muestran dinámicamente en la página.

Pruebas de funcionamiento

Cerradura de Kleene

Figura 4 Cerradura de Kleene.

Cerradura Positiva

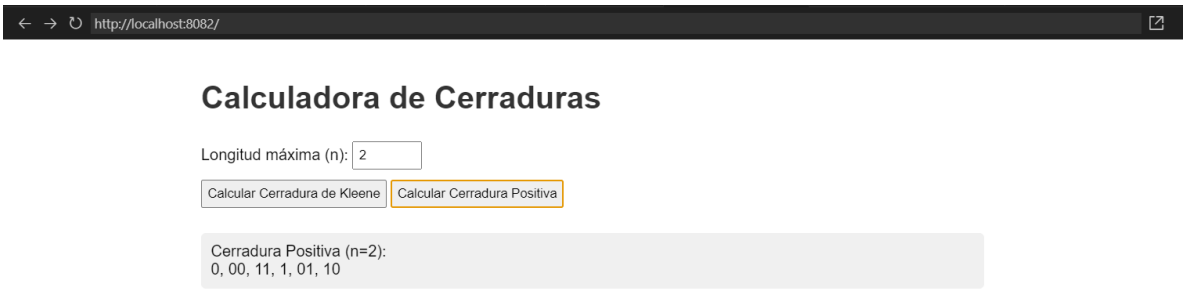

Figura 5 Cerradura Positiva.

Conclusiones

El desarrollo de esta práctica tuvo varios retos. Se implementó el algoritmo recursivo para generar todas las cadenas binarias hasta una longitud máxima n. En la práctica también se realizó la integración de Spring Boot y el manejó conjuntos en Java. Además, se desarrolló de la interfaz web interactiva que se comunique con la API REST.

Logramos la creación de una API REST funcional y la interfaz de usuario intuitiva. Se consiguió una página web simple pero efectiva que permite a los usuarios interactuar fácilmente con la API. Después se implementaron con éxito los conceptos de cerradura de Kleene y cerradura positiva en un contexto práctico de desarrollo de software.

La integración de herramientas de desarrollo se logró utilizar como pueden ser Spring Boot, Maven, y GitHub.

Esta práctica ha proporcionado una experiencia en el desarrollo de aplicaciones web basadas en API, reforzando la comprensión de conceptos fundamentales de lenguajes formales y su aplicación en el desarrollo de software.

Bibliografía APPA

- Documentation overview. (s/f). Recuperado el 30 de septiembre de 2024, de Spring.io website: https://docs.spring.io/spring-boot/documentation.html
- Extensions, L. M. A. (s/f). Visual Studio Code - code editing. Redefined. Recuperado el 30 de septiembre de 2024, de Visualstudio.com website: https://code.visualstudio.com/
- Redmond, E. (2008, enero 1). Maven in 5 minutes. Recuperado el 30 de septiembre de 2024, de Apache.org website: https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html
- (S/f). Recuperado el 30 de septiembre de 2024, de Amazon.com website: https://aws.amazon.com/corretto/