

# Simulação de Navegador Web e Sistema de Senhas Hospitalares

Trabalho de Estrutura de Dados

Andre Luiz Braga

20 de junho de 2025

## 1 Introdução

Este trabalho propõe a implementação de dois sistemas distintos, utilizando estruturas de dados fundamentais em Java:

- Um simulador do funcionamento dos botões **voltar** e **avancar** de um navegador web, utilizando pilhas.
- Um sistema gerador e controlador de senhas para atendimento hospitalar, utilizando filas e histórico de atendimento.

A proposta visa reforçar a compreensão prática do uso de pilhas e filas, bem como aplicar boas práticas de desenvolvimento orientado a objetos.

## 2 Implementação

### 2.1 Simulador de Navegador

O comportamento do navegador é controlado com duas pilhas:

- **Pilha de voltar:** armazena as páginas anteriores.
- **Pilha de avançar:** armazena as páginas retornadas.

Sempre que uma nova página é acessada, o histórico de avançar é limpo. As pilhas são manipuladas conforme as operações do usuário (acessar, voltar, avançar).

**Diagrama ilustrativo**

### 2.2 Sistema de Senhas Hospitalares

O sistema utiliza uma **fila FIFO** para armazenar senhas, e uma **lista de histórico** para registrar os atendimentos. Novas senhas são geradas sequencialmente. Quando um paciente é chamado, a senha é removida da fila e adicionada ao histórico.

## Diagrama ilustrativo

### 2.3 Formato de Entrada/Saída

As simulações são feitas no método `main` da aplicação. As saídas são exibidas diretamente no terminal, simulando o comportamento dos sistemas. As entradas são fixas para efeito de teste.

### 2.4 Decisões de Implementação

Foi decidido:

- Não utilizar interface gráfica para manter o foco nas estruturas.
- Usar classes separadas para cada estrutura (navegador e gerador de senhas).
- Não utilizar bibliotecas externas.

## 3 Testes Executados

Os seguintes testes foram realizados com base no método `main()`:

- Acessar múltiplas páginas e verificar o estado das pilhas.
- Testar comportamento ao tentar avançar ou voltar sem elementos.
- Gerar múltiplas senhas e chamar pacientes até a fila esvaziar.
- Verificar histórico após atendimento.

Todos os testes foram bem-sucedidos, com comportamento condizente ao esperado.

## 4 Conclusão

O trabalho permitiu reforçar conceitos fundamentais de estruturas de dados (pilhas e filas) com aplicações reais. A principal dificuldade foi a organização das operações de navegação de forma a refletir fielmente o funcionamento de um navegador, bem como o controle de histórico do sistema de senhas.

## 5 Bibliografia

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Algoritmos: Teoria e Prática*. Bookman.
- <https://docs.oracle.com/javase/8/docs/api/java/util/Stack.html>
- <https://docs.oracle.com/javase/8/docs/api/java/util/Queue.html>
- Slides e material da disciplina de Estrutura de Dados - UniBH

# Anexos

## Main.java

```
1 public class Main {
2     public static void main(String[] args) {
3         System.out.println("=== SIMULANDO NAVEGADOR ===");
4         Navegador nav = new Navegador();
5         nav.acessarPagina("google.com");
6         nav.acessarPagina("mozilla.org");
7         nav.acessarPagina("openai.com");
8         nav.voltar();
9         nav.voltar();
10        nav.avancar();
11        nav.acessarPagina("github.com");
12
13        System.out.println("\n=== GERADOR DE SENHAS ===");
14        GeradorDeSenhas gerador = new GeradorDeSenhas();
15
16        gerador.gerarSenha();
17        gerador.gerarSenha();
18        gerador.gerarSenha();
19
20        gerador.chamarProximo();
21        gerador.chamarProximo();
22        gerador.chamarProximo();
23        gerador.chamarProximo(); // sem senha
24
25        System.out.println("Histórico de atendimentos: " + gerador.
26                               getHistorico());
27    }
```

## Navegador.java

```
1 import java.util.Stack;
2
3 public class Navegador {
4     private Stack<String> voltar;
5     private Stack<String> avancar;
6     private String paginaAtual;
7
8     public Navegador() {
9         voltar = new Stack<>();
10        avancar = new Stack<>();
11        paginaAtual = null;
12    }
13
14    public void acessarPagina(String url) {
15        if (paginaAtual != null) {
16            voltar.push(paginaAtual);
17        }
18        paginaAtual = url;
19        avancar.clear(); // ao acessar uma nova página, a pilha de
20                          avanço é descartada
21        printStatus("Acessando nova página");
22    }
```

```

23     public void voltar() {
24         if (!voltar.isEmpty()) {
25             avancar.push(paginaAtual);
26             paginaAtual = voltar.pop();
27         }
28         printStatus("Voltar");
29     }
30
31     public void avancar() {
32         if (!avancar.isEmpty()) {
33             voltar.push(paginaAtual);
34             paginaAtual = avancar.pop();
35         }
36         printStatus("Avançar");
37     }
38
39     public boolean podeVoltar() {
40         return !voltar.isEmpty();
41     }
42
43     public boolean podeAvancar() {
44         return !avancar.isEmpty();
45     }
46
47     private void printStatus(String acao) {
48         System.out.println("\n--- " + acao + " ---");
49         System.out.println("Página atual: " + paginaAtual);
50         System.out.println("Pode voltar? " + podeVoltar());
51         System.out.println("Pode avançar? " + podeAvancar());
52     }
53 }

```

## GeradorDeSenhas.java

```

1 import java.util.ArrayList;
2 import java.util.LinkedList;
3 import java.util.List;
4 import java.util.Queue;
5
6 public class GeradorDeSenhas {
7     private Queue<Integer> filaSenhas;
8     private List<Integer> historico;
9     private int proximaSenha;
10
11     public GeradorDeSenhas() {
12         filaSenhas = new LinkedList<>();
13         historico = new ArrayList<>();
14         proximaSenha = 1;
15     }
16
17     public void gerarSenha() {
18         filaSenhas.add(proximaSenha++);
19         System.out.println("Senha gerada: " + (proximaSenha - 1));
20     }
21
22     public void chamarProximo() {
23         if (!filaSenhas.isEmpty()) {
24             int senha = filaSenhas.poll();

```

```
25         historico.add(senha);
26         System.out.println("Chamando senha: " + senha);
27     } else {
28         System.out.println("Nenhuma senha na fila.");
29     }
30 }
31
32 public boolean filaVazia() {
33     return filaSenhas.isEmpty();
34 }
35
36 public List<Integer> getHistorico() {
37     return historico;
38 }
39 }
```