

# Implementação de Navegador e Fila Hospitalar

Luís Filipe França

Março de 2025

## 1 Introdução

Este documento descreve a implementação de dois sistemas distintos: um navegador web com botões de avançar e voltar e um sistema de controle de fila para pacientes em hospitais. Ambos foram implementados utilizando a linguagem Java, seguindo os princípios da programação orientada a objetos.

## 2 Implementação

### 2.1 Sistema de Navegador

O sistema de navegação foi desenvolvido utilizando duas pilhas para armazenar os históricos de navegação, possibilitando o funcionamento dos botões de avançar e voltar.

#### 2.1.1 Estrutura de Dados Utilizada

Foram utilizadas as classes `Stack<String>` para armazenar os endereços de páginas acessadas. Quando o usuário visita uma nova página, a página anterior é armazenada na pilha de "voltar". Ao pressionar "voltar", a página atual vai para a pilha de "avançar".

#### 2.1.2 Código Fonte

```
1 import java.util.Stack;
2
3 class Navegador {
4     private Stack<String> pilhaVoltar = new Stack<>();
5     private Stack<String> pilhaAvancar = new Stack<>();
6     private String paginaAtual;
7
8     public Navegador() {
9         this.paginaAtual = null;
10    }
11
12    public void acessarPagina(String url) {
13        if (paginaAtual != null) {
```

```

14         pilhaVoltar.push(paginaAtual);
15     }
16     paginaAtual = url;
17     pilhaAvancar.clear();
18 }
19
20 public void voltar() {
21     if (!pilhaVoltar.isEmpty()) {
22         pilhaAvancar.push(paginaAtual);
23         paginaAtual = pilhaVoltar.pop();
24     }
25 }
26
27 public void avancar() {
28     if (!pilhaAvancar.isEmpty()) {
29         pilhaVoltar.push(paginaAtual);
30         paginaAtual = pilhaAvancar.pop();
31     }
32 }
33 }

```

## 2.2 Sistema de Fila de Pacientes

Para o gerenciamento de pacientes, foi utilizada uma estrutura de fila, garantindo a ordem de chegada para atendimento.

### 2.2.1 Estrutura de Dados Utilizada

Utilizamos a classe `Queue<String>` (implementada com `LinkedList`) para armazenar as senhas dos pacientes. O método `poll()` permite retirar o primeiro paciente da fila para atendimento.

### 2.2.2 Código Fonte

```

1 import java.util.LinkedList;
2 import java.util.Queue;
3
4 class Hospital {
5     private Queue<String> filaPacientes = new LinkedList<>();
6     private int contadorSenhas = 1;
7
8     public void gerarSenha(String nome) {
9         String senha = "p" + contadorSenhas + " - " + nome;
10        filaPacientes.add(senha);
11        contadorSenhas++;
12    }
13
14    public void chamarProximo() {
15        if (!filaPacientes.isEmpty()) {
16            System.out.println("Chamando: " + filaPacientes.poll());
17        }
18    }
19 }

```

### 3 Testes Executados

Foram realizados testes para validar a navegação entre páginas e o gerenciamento da fila hospitalar. Os resultados demonstraram que os sistemas funcionam corretamente, permitindo navegação e controle de fila de maneira eficiente.

### 4 Conclusão

A implementação dos sistemas atendeu aos requisitos propostos. O uso de pilhas para a navegação e filas para a organização dos pacientes demonstrou-se eficaz.

### 5 Bibliografia

- Documentação oficial Java: <https://docs.oracle.com/en/java/>
- Exemplos de estrutura de dados: <https://www.geeksforgeeks.org/>