



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Simulação de Controle de Acesso para Restaurante Universitário

Análise e Projeto de Sistemas OO (14108)

**Professor:
Lucas Albertins**

José Edson Amorim SEBASTIÃO¹

¹Graduandos em Licenciatura em Computação. E-mail para correspondência:
joseedson.sebastiao@ufrpe.br



Introdução

Este documento apresenta a arquitetura e a implementação de um sistema de controle de acesso para um restaurante universitário, desenvolvido utilizando uma arquitetura baseada em microsserviços. O objetivo deste sistema é simular o processo de autenticação e validação de usuários (alunos) por meio de uma senha e de uma verificação biométrica, permitindo o acesso controlado à catraca do restaurante.

A estrutura do documento inclui a descrição dos serviços implementados, o padrão de arquitetura utilizado e detalhes das tecnologias envolvidas. O foco principal será nos microsserviços desenvolvidos e na configuração da infraestrutura necessária para o seu funcionamento.

Histórico de Revisões

Data	Descrição	Responsável
09/09/24	Criação do repositório e definição dos microservicos	José Edson Amorim
11/09/24	Conexão com o banco de dados, estruturação do Eureka e Docker.	José Edson Amorim
18/09/24	Criação do front-end	José Edson Amorim

Arquitetura de Serviços

Serviços Implementados

O sistema é composto por três microsserviços principais, que juntos simulam o controle de acesso ao restaurante universitário:

1. **Auth-Service (Serviço de Autenticação):**
 - Responsável por gerenciar o registro e o login dos usuários.



- Utiliza JPA para persistência e o banco de dados H2 para armazenamento local.
- O aluno insere sua senha para validar a entrada.
- 2. **Biometrics-Service (Serviço de Biometria):**
 - Valida a biometria do aluno.
 - Simula o processo de verificação biométrica.
- 3. **Turnstile-Service (Serviço de Catraca):**
 - Controla a liberação da catraca após a autenticação e validação biométrica.

Além dos microserviços principais, o sistema conta com os seguintes componentes auxiliares:

- **Discovery Service (Eureka):**
 - Responsável pelo registro dos microserviços e gerenciamento dinâmico dos endereços de rede.
- **API Gateway:**
 - Utilizado para rotear as requisições entre o cliente (interfaces HTML) e os microserviços.

Padrão de Arquitetura e Tecnologias Utilizadas

O sistema segue uma **arquitetura de microserviços**, onde cada serviço é independente e realiza uma função específica. A comunicação entre os microserviços é facilitada pelo Discovery Service e pelo API Gateway.

Front-End

O front-end é composto por páginas HTML simples, servidas localmente por um servidor HTTP (**http-server**). Essas páginas permitem a interação com o sistema de autenticação, validação biométrica e controle da catraca.

Framework MVC (Opcional)

Embora não tenha sido utilizado um framework completo para o front-end, a separação de responsabilidades entre os serviços segue o padrão de design MVC (Model-View-Controller) no back-end.

Back-End

O back-end foi desenvolvido utilizando o **Spring Boot**, um framework Java que facilita a criação de aplicações web com arquitetura em camadas.

- **Banco de Dados:**
 - O sistema utiliza o **H2**, um banco de dados local em memória, para armazenar informações dos usuários, como suas credenciais e dados biométricos.
- **Comunicação entre Microserviços:**
 - A comunicação entre os serviços é gerenciada pelo **Eureka** e roteada pelo **API Gateway**, facilitando a escalabilidade e manutenção do sistema.



Docker

A estrutura para a execução dos microsserviços em containers **Docker** já foi configurada, mas ainda precisa de ajustes para funcionar corretamente. O objetivo é que cada microsserviço possa ser executado de forma independente e isolada, garantindo a flexibilidade e escalabilidade do sistema.