

# 1. Introdução

Este documento descreve o desenvolvimento de um microserviço para consulta de CEP, utilizando:

- **Backend:** Python com FastAPI
- **Frontend:** Vue.js
- **Containerização:** Docker
- **API pública:** ViaCEP

O projeto atende aos requisitos das disciplinas de Sistemas Distribuídos, Qualidade de Software, Testes de Software, Gerência de Configuração e Implantação de Sistemas.

---

## 2. Como Foi Desenvolvido

### 2.1 Pesquisa Inicial

- Investigação sobre APIs de CEP (ViaCEP) e padrões de resposta.
- Análise de bibliotecas para criação de APIs em Python (FastAPI escolhido por simplicidade e performance).

### 2.2 Implementação

#### Backend (Python/FastAPI):

- Endpoint `/cep/{cep}` para consulta, com validação de formato (8 dígitos).
- Tratamento de erros para CEPs inválidos, inexistentes ou falhas na API ViaCEP.
- Integração com CORS para permitir requisições do frontend.

#### Frontend (Vue.js):

- Interface simples com campo de entrada e exibição de resultados.
- Formatação automática do CEP (ex: `00000-000`).
- Feedback visual para erros e carregamento.

#### Infraestrutura (Docker):

- Containerização separada para frontend e backend.
- Orquestração via `docker-compose`.

### 2.3 Testes

- **Testes unitários** no backend (pytest):
  - CEP válido (`01001000` → São Paulo/SP).
  - CEP inválido (`123` → erro 400).
  - CEP inexistente (`00000000` → erro 404).

- **Testes manuais** no frontend:
    - Validação de formatos de entrada.
    - Exibição de mensagens de erro.
- 

## 3. Como Instalar e Usar

### 3.1 Pré-requisitos

- Docker e Docker Compose instalados.

### 3.2 Passos para Execução

bash

# Clone o repositório

git clone <https://github.com/seu-usuario/cep-microservice.git>

# Inicie os containers

docker-compose up --build

**Acesse:**

- Frontend: <http://localhost:8080>
  - Backend (API): <http://localhost:8000/cep/01001000>
- 

## 4. Aprendizados

- **Integração de APIs:** Consumo eficiente do ViaCEP com tratamento de falhas.
- **Microserviços:** Separação clara entre frontend e backend.
- **Docker:** Simplificação do ambiente de desenvolvimento e deploy.
- **Testes:** Importância de validar entradas inválidas para robustez.

## 5. Repositório no GitHub

Disponível em:

text

<https://github.com/seu-usuario/cep-microservice>

**Estrutura:**

- Código organizado com [README.md](#).
- Testes automatizados na pasta [backend/tests](#).
- Dockerfiles e [docker-compose.yml](#) para fácil reprodução.

## Anexos:

- [https://drive.google.com/file/d/18ZsrNIIvku3\\_rfOhRv0NmCVx3nio1aJz/view?usp=drive\\_link](https://drive.google.com/file/d/18ZsrNIIvku3_rfOhRv0NmCVx3nio1aJz/view?usp=drive_link)

## 6. Referências Bibliográficas

Abaixo estão as referências utilizadas para o desenvolvimento deste trabalho, seguindo as normas ABNT:

- 
1. **FASTAPI.** *Documentação Oficial*. Disponível em: <https://fastapi.tiangolo.com/>. Acesso em: 30 jun. 2025.  
(*Documentação oficial do FastAPI, utilizada para implementação do backend em Python.*)
  2. **VIA CEP.** *API de Consulta de CEP*. Disponível em: <https://viacep.com.br/>. Acesso em: 30 jun. 2025.  
(*API pública gratuita para consulta de endereços por CEP, integrada ao microserviço.*)
  3. **VUE.JS.** *Documentação Oficial*. Disponível em: <https://vuejs.org/>. Acesso em: 29 jun. 2025.  
(*Referência para desenvolvimento da interface frontend com Vue.js.*)
  4. **DOCKER.** *Documentação Oficial*. Disponível em: <https://docs.docker.com/>. Acesso em: 28 jun. 2025.  
(*Guias para containerização da aplicação com Docker e Docker Compose.*)
  5. **PYTEST.** *Framework de Testes para Python*. Disponível em: <https://docs.pytest.org/>. Acesso em: 30 jun. 2025.  
(*Utilizado para implementar testes automatizados no backend.*)
  6. **GIT.** *Versionamento de Código*. Disponível em: <https://git-scm.com/doc>. Acesso em: 25 jun. 2025.  
(*Controle de versão do projeto com Git e GitHub.*)
  7. **RESTful APIs.** *Princípios de Design*. In: O'REILLY. *REST API Design Rulebook*. 1. ed. 2011.  
(*Padrões adotados para estrutura do endpoint /cep.*)
  8. **SENAI.** *Diretrizes para Trabalhos Acadêmicos*. 2025.  
(*Normas internas para entrega de trabalhos.*)