

# 1. Introdução

Este relatório apresenta o desenvolvimento, a arquitetura, e os testes automatizados realizados no backend do projeto Fecapass, um sistema para gerenciamento de usuários com funcionalidades de cadastro, login, atualização, e recuperação de senha. O objetivo principal foi construir um backend robusto, testável e escalável, utilizando Node.js, e aplicar estratégias de testes unitários, de integração e de carga para garantir a qualidade do sistema.

---

## 2. Objetivos

- Desenvolver as funcionalidades principais do backend para gerenciamento de usuários.
  - Garantir a qualidade do software por meio da implementação de testes automatizados.
  - Avaliar a performance e a estabilidade do sistema através de testes de carga.
  - Documentar o processo e os resultados para apresentação no projeto interdisciplinar.
- 

## 3. Metodologia

### 3.1 Ferramentas Utilizadas

- **Linguagem:** JavaScript (Node.js)
- **Testes:** Jest (framework de testes)
- **Banco de Dados:** (não utilizado diretamente nos testes unitários)
- **Ambiente de desenvolvimento:** Visual Studio Code, terminal do Windows.

### 3.2 Estrutura do Projeto

- `/Services`: contém as funções que implementam a lógica de negócio, como cadastro, login, atualização, validação de senha, etc.
- `/controllers`: controladores para receber as requisições (não detalhado neste relatório).
- `/tests/unit`: testes unitários para funções isoladas.
- `/tests/integration`: testes que verificam a integração entre componentes do sistema.
- `/tests`: testes de carga para simular alta demanda.

### 3.3 Estratégia de Testes

- **Testes unitários:** validar cada função isoladamente para garantir o correto comportamento diante de entradas válidas e inválidas.

- **Testes de integração:** garantir que componentes relacionados funcionem em conjunto, especialmente para operações críticas como login.
  - **Testes de carga:** simular múltiplos acessos e operações simultâneas para medir a estabilidade e desempenho do backend.
- 

## 4. Desenvolvimento

Durante o desenvolvimento, as funções foram construídas utilizando o padrão CommonJS (`require` e `module.exports`), favorecendo a compatibilidade com Jest e simplificando o ambiente de testes.

Foram implementadas funções para:

- Cadastro de usuário (`cadaststrarUsuario`), com verificação de e-mail duplicado.
  - Login de usuário (`loginUsuario`), que valida o e-mail e a senha e retorna um token JWT.
  - Atualização de dados do usuário (`updateUser`), para alterar nome e senha.
  - Validação de senha (`validarSenha`), com regras para aceitar ou rejeitar uma senha.
  - Envio de código para recuperação e redefinição de senha (não detalhado nos testes unitários).
- 

## 5. Testes Automatizados

### 5.1 Testes Unitários

Os testes unitários foram desenvolvidos para garantir a confiabilidade das funções críticas isoladamente. Eles validam os casos de sucesso e de erro esperado.

**Principais casos testados:**

- Validação de senha:
  - Deve lançar erro se a senha for inválida.
  - Deve retornar sucesso para senha válida.
- Login com email inválido:
  - Deve lançar erro para e-mail não cadastrado.
- Atualização de usuário:
  - Deve atualizar nome e senha corretamente.
- Cadastro de usuário:
  - Deve lançar erro se o e-mail já estiver em uso.

**Resultados obtidos:**

Arquivo	Testes Realizados	Testes Aprovados	Descrição
validarSenha.test.js	2	2	Testa validação de senha
login.test.js	1	1	Login com e-mail inválido
updateUser.test.js	1	1	Atualização de usuário
cadastrar.test.js	1	1	Cadastro com e-mail duplicado

**Total:** 5 testes unitários aprovados com sucesso.

```

PS C:\Users\PC\OneDrive\Desktop\Testes Software\fecapass-backend> npx jest tests/units/updateUser.test.js
PASS tests/units/updateUser.test.js
  updateUser - atualização simples
    ✓ deve atualizar nome e senha de um usuário existente (62 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 0.813 s, estimated 1 s
Ran all test suites matching /tests\units\updateUser.test.js/i.
PS C:\Users\PC\OneDrive\Desktop\Testes Software\fecapass-backend> npx jest tests/units/validarSenha.test.js
PASS tests/units/validarSenha.test.js
  validação de senha com lançamento de erro
    ✓ deve lançar erro se a senha for inválida (6 ms)
    ✓ deve retornar true se a senha for válida

Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 0.753 s, estimated 1 s
Ran all test suites matching /tests\units\validarSenha.test.js/i.
PS C:\Users\PC\OneDrive\Desktop\Testes Software\fecapass-backend> npx jest tests/units/login.test.js
PASS tests/units/login.test.js
  loginUsuario (email inválido)
    ✓ deve lançar erro ao tentar login com e-mail inexistente (15 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 0.759 s, estimated 1 s
Ran all test suites matching /tests\units\login.test.js/i.
PS C:\Users\PC\OneDrive\Desktop\Testes Software\fecapass-backend> npx jest tests/units/cadastrar.test.js
PASS tests/units/cadastrar.test.js
  cadastrarUsuario - email em uso
    ✓ deve lançar erro se o e-mail já estiver em uso (15 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 0.935 s
Ran all test suites matching /tests\units\cadastrar.test.js/i.

```

## 5.2 Testes de Integração

Os testes de integração verificaram a interação entre as funções de login e autenticação, confirmando a geração do token JWT.

### Caso testado:

- Login com sucesso retorna token JWT válido.

### Resultado:

Arquivo	Testes Realizados	Testes Aprovados	Descrição
login.test.js (integração)	1	1	Login bem-sucedido com token JWT

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\PC\OneDrive\Desktop\Testes Software\fecapass-backend> npx jest tests/integration/login.test.js
PASS tests/integration/login.test.js
  Integração: Login com sucesso
    ✓ deve fazer login e retornar um token JWT (37 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.539 s
Ran all test suites matching /tests\\integration\\login.test.js/i.

```

### 5.3 Testes de Carga

Foram realizados testes para avaliar a performance do backend sob alta demanda, simulando execuções em larga escala.

#### Casos testados:

- 1.000 cadastros de usuários sequenciais.
- 10.000 validações de senha em paralelo.
- 2.000 cadastros simultâneos.
- 2.000 logins simultâneos.

#### Resultados:

Teste	Execuções	Tempo Total	Descrição
Cadastro sequencial	1.000	14.218 ms	Execução de cadastros em sequência
Validação paralela de senha	10.000	2 ms	Validações de senha em alta concorrência
Cadastros simultâneos	2.000	28.422 ms	Execução paralela de cadastros
Logins simultâneos	2.000	28.490 ms	Logins executados simultaneamente

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\PC\OneDrive\Desktop\Testes Software\fecapass-backend> npx jest tests/testesDeCarga.test.js
console.log
  Iniciando 1.000 cadastros de usuários...

    at log (tests/testesDeCarga.test.js:32:13)

console.log
  ✓ 1.000 cadastros executados em 14218 ms

    at log (tests/testesDeCarga.test.js:38:13)

console.log
  Iniciando 10.000 validações de senha...

    at log (tests/testesDeCarga.test.js:53:13)

console.log
  🟡 10000 validações de senha executadas em 2 ms

    at log (tests/testesDeCarga.test.js:63:13)

console.log
  Iniciando 2000 cadastros de usuários...

    at log (tests/testesDeCarga.test.js:76:13)

console.log
  🟡 2000 cadastros realizados em 28422 ms

    at log (tests/testesDeCarga.test.js:83:13)

console.log
  Iniciando 2000 logins simultâneos...

    at log (tests/testesDeCarga.test.js:105:13)

console.log
  🟡 2000 logins bem-sucedidos em 28490 ms

    at log (tests/testesDeCarga.test.js:110:13)

PASS tests/testesDeCarga.test.js (79.441 s)
Testes de carga e integração para userServices
  ✓ Deve cadastrar usuário com dados válidos (teste de carga leve - 1000 execuções) (21343 ms)
  ✓ Executa validarSenha 10.000 vezes em paralelo (bloco 1) (5 ms)
  ✓ Executa 2.000 cadastros de usuários em paralelo (28555 ms)
  ✓ Executa 2.000 logins simultâneos com usuários válidos (28686 ms)

Test Suites: 1 passed, 1 total
Tests: 4 passed, 4 total
Snapshots: 0 total
Time: 79.491 s, estimated 263 s
Ran all test suites matching /tests\\testesDeCarga.test.js/i.
```

## 6. Conclusão

O backend do sistema Fecapass foi implementado com sucesso, cumprindo os requisitos de funcionalidade e desempenho. Os testes unitários confirmaram a correta implementação das funções isoladas, os testes de integração garantiram a correta interação entre módulos, e os testes de carga demonstraram que o sistema suporta alto volume de requisições simultâneas dentro de tempos aceitáveis.

O uso do Jest para automação dos testes contribuiu para a qualidade do software e para a confiança na estabilidade do sistema, importante para a entrega final e possíveis ampliações futuras.