

# Relatório de Testes - FECAPay+

## Introdução

Este relatório apresenta os testes automatizados desenvolvidos para o sistema FECAPay, uma aplicação de gerenciamento financeiro para alunos da FECAP. Os testes abrangem desde testes unitários até testes de integração, garantindo a qualidade e funcionalidade do sistema.

## Estrutura de Testes

O projeto possui 5 arquivos de teste principais:

1. `authController.test.js` - Testes de autenticação
2. `balanceController.test.js` - Testes de saldo
3. `boletosController.test.js` - Testes de boletos
4. `userController.test.js` - Testes de usuários
5. `api.test.js` - Teste de integração completo

## Detalhamento dos Testes

### 1. Testes de Autenticação (`authController.test.js`)

Cenários testados:

- **Login com credenciais válidas:** Verifica se o sistema autentica corretamente quando fornecido RA e senha válidos.
  - Esperado: Status 200 e mensagem de sucesso.
- **Login com senha incorreta:** Testa a resposta do sistema quando a senha está errada.
  - Esperado: Status 401 e mensagem de senha incorreta.
- **Login com RA inexistente:** Verifica o comportamento quando o RA não existe no banco.
  - Esperado: Status 400 e mensagem de conta não encontrada.
- **Erro no servidor:** Simula uma falha no banco de dados.
  - Esperado: Status 500 e mensagem de erro no servidor.

**Cobertura:** Testa todos os fluxos principais do sistema de autenticação, incluindo casos de sucesso e falha.

### 2. Testes de Saldo (`balanceController.test.js`)

#### Cenários testados:

- **Consulta de saldo válida:** Verifica se retorna o saldo correto para um RA existente.
  - Esperado: Status 200 e valor do saldo.
- **Consulta de saldo com RA inválido:** Testa a resposta para um RA sem registro.
  - Esperado: Status 404 e mensagem de dados não encontrados.
- **Adição de saldo válida:** Testa a adição de valor positivo ao saldo.
  - Esperado: Status 200 e novo saldo calculado.
- **Adição de saldo inválido:** Testa valores negativos, zero e strings.
  - Esperado: Status 400 para todos os casos.
- **Adição para RA inexistente:** Verifica o comportamento quando o RA não existe.
  - Esperado: Status 404 e mensagem de usuário não encontrado.

**Cobertura:** Garante que todas as operações relacionadas ao saldo funcionem corretamente e tratem adequadamente entradas inválidas.

### 3. Testes de Boletos (**boletosController.test.js**)

#### Cenários testados:

- **Listagem de boletos:** Verifica se retorna a lista de boletos para um RA.
  - Esperado: Status 200 e array de boletos.
- **Listagem para RA sem boletos:** Testa a resposta para RA sem boletos cadastrados.
  - Esperado: Status 404 e mensagem de dados não encontrados.
- **Pagamento de boleto com saldo:** Testa o pagamento bem-sucedido.
  - Esperado: Status 200 e mensagem de sucesso.
- **Pagamento de boleto já pago:** Verifica o comportamento ao tentar pagar boleto já pago.
  - Esperado: Status 400 e mensagem informando que já foi pago.
- **Pagamento com saldo insuficiente:** Testa quando o saldo é menor que o valor do boleto.
  - Esperado: Status 400 e mensagem de saldo insuficiente.

**Cobertura:** Abrange todos os fluxos principais do módulo de boletos, garantindo comportamentos adequados em diferentes situações.

### 4. Testes de Usuários (**UserController.test.js**)

#### Cenários testados:

- **Cadastro de novo usuário:** Verifica o cadastro bem-sucedido.
  - Esperado: Status 200 e dados do usuário.
- **Cadastro duplicado:** Testa tentativa de cadastro com RA ou email já existentes.
  - Esperado: Status 400 e mensagem de usuário já cadastrado.
- **Alteração de senha válida:** Testa a alteração com credenciais corretas.

- Esperado: Status 200 e mensagem de sucesso.
- **Alteração com senha atual incorreta:** Verifica o comportamento quando a senha atual está errada.
  - Esperado: Status 401 e mensagem de senha incorreta.
- **Consulta de usuário:** Testa a busca por RA válido.
  - Esperado: Status 200 e dados do usuário.
- **Exclusão de usuário:** Verifica a remoção bem-sucedida.
  - Esperado: Status 200 e dados do usuário excluído.
- **Listagem de usuários:** Testa a recuperação de todos os usuários.
  - Esperado: Status 200 e array de usuários.

**Cobertura:** Testa todas as operações CRUD do módulo de usuários, incluindo casos de sucesso e falha.

## 5. Teste de Integração Completa (**api.test.js**)

**Fluxo testado:**

1. Cadastro de novo usuário
2. Login com as credenciais cadastradas
3. Adição de saldo à conta
4. Consulta do saldo adicionado
5. Listagem de boletos gerados

**Objetivo:** Verificar se todos os módulos do sistema trabalham corretamente em conjunto, simulando um fluxo real de uso.

## Conclusão

Os testes desenvolvidos cobrem a maior parte das funcionalidades críticas do sistema FECAPay+, garantindo:

- Comportamento correto nos fluxos principais
- Tratamento adequado de erros e casos extremos
- Integração entre os diferentes módulos
- Segurança nas operações sensíveis (como autenticação e alteração de senha)