

# SEMANA 02 (20/10) - Regras de Negócio e Funcionalidades Internas

## Objetivo

Implementar as regras de negócio e os fluxos principais de cada módulo, criando a lógica central da plataforma.

## Atividades Concluídas

### 1. Camada Service (Lógica de Negócio)

#### 1.1 AlunoService

##### Regras de Negócio Implementadas:

- CPF único por aluno
- Email único e validado
- CPF com 11 dígitos
- Aluno pode estar ativo ou inativo
- Apenas alunos ativos podem se inscrever

#### 1.2 InstrutorService

##### Regras de Negócio Implementadas:

- Email único para instrutor
- Senha com até 8 caracteres
- Instrutor pode ter múltiplos cursos
- Apenas instrutores ativos podem criar cursos

#### 1.3 CursoService

### **Regras de Negócio Implementadas:**

- Preço mínimo de R\$ 0,01
- Nome e descrição obrigatórios
- Instrutor deve existir
- Apenas cursos ativos aparecem no catálogo
- Cada curso pertence a um instrutor

### **1.4 InscricaoService**

#### **Regras de Negócio Implementadas:**

- Aluno não pode se inscrever duas vezes no mesmo curso
- Apenas alunos ativos podem se inscrever
- Apenas cursos ativos aceitam inscrições
- Data de inscrição registrada automaticamente
- Status inicial: PENDENTE

### **1.5 PagamentoService**

#### **Regras de Negócio Implementadas:**

- Valor deve ser positivo
- Métodos: PIX, CARTAO, BOLETO
- Status: PENDENTE → CONFIRMADO ou REJEITADO
- Reembolso permitido
- Transações imutáveis após confirmação

## **2. Validações Implementadas**

**2.1 Validador de CPF**

**2.2 Validador de Email**

**2.3 Validador de Dados**

### **3. Autenticação e Autorização**

**3.1 Autenticação Aluno**

**3.2 Autenticação Instrutor**

**3.3 Autenticação Admin**

### **4. DTOs Implementados**

**AlunoDTO**

**CursoDTO**

**InscricaoDTO**

**PagamentoDTO**

**ReembolsoDTO**

### **5. Exceções Customizadas**

**NegocioException**

**RecursoNaoEncontradoException**

**ValidacaoException**

**GlobalExceptionHandler**

### **6. Controladores Atualizados**

**AlunoController**

**InstrutorController**

**CursoController**

**AlunoFluxoController**

**PagamentoController**

## **Fluxos de Negócio Implementados**

## **Fluxo de Inscrição em Curso**

Aluno seleciona curso

Validações

Criar inscrição

Redirecionar para pagamento

Processar pagamento

## **Fluxo de Pagamento**

Validar dados

Validar método

Criar pagamento

Confirmar ou rejeitar

Atualizar inscrição

## **Fluxo de Reembolso**

Validar pagamento

Validar status

Criar reembolso

Atualizar inscrição

## **Testes de Negócio**

### **Cenário 1: Inscrição com Sucesso**

Aluno ativo, curso ativo, resultado: inscrição criada.

### **Cenário 2: Inscrição Duplicada**

Aluno tenta se inscrever duas vezes; sistema retorna erro.

### **Cenário 3: Pagamento com Cartão**

Pagamento confirmado com sucesso.

## **Checklist de Implementação**

### **Services**

- AlunoService
- InstrutorService
- CursoService
- InscricaoService
- PagamentoService

## **Validações**

- CPFValidator
- EmailValidator
- ValidadorDados

## **Autenticação**

- Login de aluno
- Login de instrutor
- Login de admin
- Geração de sessão

## **DTOs**

- AlunoDTO
- InstrutorDTO
- CursoDTO
- InscricaoDTO
- PagamentoDTO
- ReembolsoDTO

## **Controladores**

- Todos os endpoints concluídos
- Tratamento de exceções
- Validação de entrada