

## **SEMANA 05 (10/11) - Finalização e Relatórios**

### **Objetivo**

Testar, documentar e apresentar o sistema final com qualidade de produção.

### **Atividades Concluídas**

#### **Testes Automatizados**

Foram desenvolvidos testes unitários utilizando JUnit 5 para todos os serviços principais, incluindo AlunoService, CursoService, InscricaoService e PagamentoService. Os testes validam criação, atualização, autenticação, tratamento de erros, validações e operações de negócio. Também foram implementados testes de integração para os controllers, utilizando TestRestTemplate, cobrindo operações de criação, listagem, consulta, atualização, exclusão e autenticação via API.

#### **Cobertura de Testes**

A cobertura geral ultrapassou 80 por cento. Os serviços atingiram cerca de 90 por cento de cobertura total, variando entre 85 e 95 por cento conforme o módulo. Os controllers ficaram com aproximadamente 82 por cento de cobertura. Modelos e repositórios ficaram entre 70 e 75 por cento. O total geral de cobertura foi aproximadamente 85 por cento.

#### **Guia de Uso**

O arquivo COMO\_USAR foi atualizado com todas as instruções necessárias para iniciar o sistema, executar o Maven Wrapper, rodar o arquivo JAR, orientar o login de cada perfil, descrever as funcionalidades disponíveis para alunos, instrutores e administradores, explicar métodos de pagamento e ensinar como resetar o banco de dados.

#### **Manual Técnico**

O manual técnico inclui a arquitetura do sistema, explicação das camadas Controller, Service, Repository e Model, detalhamento do fluxo de requisição, entidades principais do banco de dados, relacionamentos das tabelas, padrões de URLs da API, códigos de resposta HTTP utilizados, validações de segurança, otimizações de desempenho implementadas e uma seção de troubleshooting para resolução de erros comuns.

#### **Guia de Testes**

Foi criado um guia contendo comandos para executar testes completos, testes específicos e relatórios de cobertura. O guia define cenários de teste para login, inscrição, criação de curso, relatórios e casos de erro, como email duplicado, CPF inválido, inscrição repetida e valores negativos em pagamentos.

#### **Relatórios**

##### **Relatório de Receita**

Foi gerado um relatório detalhando o período analisado, total arrecadado, distribuição por forma de pagamento, receita por instrutor, cursos mais vendidos, ticket médio e taxa de conversão.

## **Relatório de Usuários**

O relatório inclui quantidade de alunos ativos e inativos, instrutores ativos, total de cursos, distribuição por status e taxa de crescimento mensal.

## **Relatório de Performance**

Foram analisados tempo médio de resposta, taxa de erro, disponibilidade, número de usuários simultâneos e consultas mais pesadas no banco de dados.

## **Apresentação do Sistema**

### **Estrutura da Apresentação**

A apresentação foi organizada com introdução, demonstração completa do sistema, explicação da arquitetura, seção de testes e conclusão. Cada etapa tem tempo definido, descrição e sequência lógica.

### **Roteiro de Demonstração**

O roteiro segue os acessos realizados com cada perfil. Inclui login do aluno, inscrição em curso, pagamento, navegação do instrutor e visualização de relatórios administrativos, finalizando com apresentação de estatísticas e dados.

## **Checklist de Implementação**

### **Testes**

Testes unitários, testes de integração, cobertura superior a 80 por cento e testes da API foram concluídos.

### **Documentação**

README, COMO\_USAR, Manual Técnico, Guia de Testes e Swagger finalizados.

### **Relatórios**

Relatórios de receita, usuários, performance e análise de problemas foram concluídos.

### **Apresentação**

Slides concluídos, demonstração organizada, roteiro definido e tempo de apresentação configurado.

### **Qualidade**

Revisão de código feita, refatorações aplicadas, otimizações de performance realizadas e verificações de segurança concluídas.