



Disciplina	Prof. Dacio Machado	
PROJETO IMPLEMENTAÇÃO E TESTE DE SOFTWARE	Valor	+01 ATV
ATIVIDADE : Teste Integração	Aluno:	
ESOFT - 6 - N		

Atividade prática de Teste de Integração:

Descrição da Aplicação

A Universidade Tabajara “TecLearn TABAJARA” decidiu informatizar o controle de sua biblioteca. O sistema deve permitir o **empréstimo de livros, cadastro de usuários, controle de acervo e relatórios de uso**.

A universidade contratou uma empresa de desenvolvimento júnior (os alunos) para criar o sistema em módulos, cada equipe ficando responsável por uma parte do sistema. A integração e os testes finais definirão se o sistema será aprovado.

Visão geral do sistema O SGBU faz o controle de usuários, acervo (livros), empréstimos e relatórios. O sistema final será composto pela integração dos módulos desenvolvidos por cada equipe.

Equipe	Módulo / Responsabilidade	Principais Funcionalidades	Exemplo de Integração
Equipe 1 Cadastro de Usuários	– CRUD de usuários	Cadastrar, editar, remover e listar usuários (nome, matrícula, tipo de usuário)	Se integra com o módulo de Empréstimo
Equipe 2 Catálogo de Livros	– CRUD de livros	Cadastrar livros, autores, estoque, status (disponível/emprestado)	Se integra com o módulo de Empréstimo
Equipe 3 Empréstimo e Devolução	– Controle de empréstimos	Registrar empréstimos e devoluções, verificar disponibilidade	Depende dos módulos de Usuários e Catálogo
Equipe 4 Relatórios	– Geração de relatórios simples	Relatórios de livros mais emprestados, usuários mais ativos	Depende dos dados das outras equipes



Informações Gerais :

Cada equipe deve seguir as regras e interfaces e requisitos aqui definidas para desenvolver o seu módulo com a metodologia TDD (Test Driven Development).

O foco principal é a integração entre módulos e a criação de testes de integração que provem o bom funcionamento do sistema composto.

Código-fonte: repositório Git (GitHub/) – cada equipe cria um repositório privado com a colaboração do professor.

Use como base de tecnologia, o projeto implementado pelo professor e testado por vocês nas primeiras aulas disponível em :

<https://replit.com/@daciofrancisco/PersonalExpenseOrganizer>

Formato dos commits (*obrigatório para evidência de TDD*): Cada teste deve ter commit que o cria (falha), commit que faz o teste passar, e commit de refactor. Mensagem padronizada: [TDD red] <descrição do teste> , [TDD green] <descrição do teste> , [TDD refactor] <descrição>.

Testes automatizados: cada equipe deve prover pelo menos:

- 10 testes unitários cobrindo casos de borda (validações, erros, sucesso).
- 5 testes de contrato/integridade (ex.: serialização correta, contratos de campo obrigatórios).