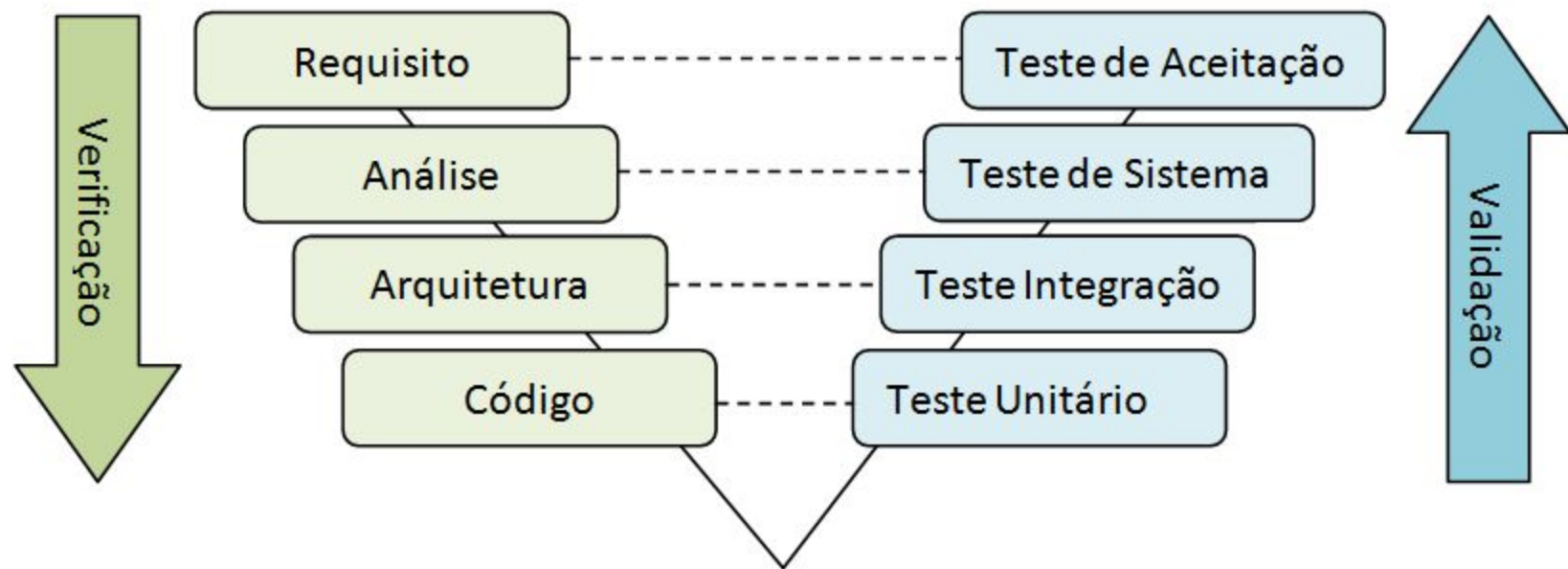


# *Integration Testing*



Projeto, implementação e Teste de Software

## Teste de Integração





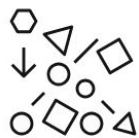
## Integração

O teste de integração é uma técnica sistemática para construir a arquitetura de software ao mesmo tempo que conduz testes para descobrir erros associados com as interfaces.

O objetivo é construir uma estrutura de programa determinada pelo projeto a partir de componentes testados em unidade.

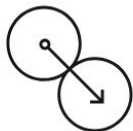


Os casos de teste utilizados nos testes de integração ajudam os desenvolvedores a focar em áreas específicas da operação:



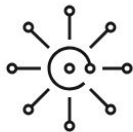
### **Fluxo de dados**

Os dados que percorrem um sistema seguem uma trajetória, indo da origem ao destino. Essa informação passa por processamento à medida que atravessa diferentes etapas e componentes do sistema. Esse processo de movimentação é conhecido como **fluxo de dados**.



### **Coordenação de interface**

Assim como equipes eficazes precisam de liderança, há uma “inteligência superior” que orienta a operação e a interação fluida entre os componentes do software. Chamamos esse processo de gerenciamento de coordenação de interfaces.



### **Protocolos de comunicação**

São os protocolos de comunicação que determinam o modo como os dispositivos trocam informações. Esses protocolos definem as regras para a transferência de dados e determinam como as mensagens devem ser estruturadas. Os protocolos de comunicação também definem como os sistemas devem corrigir falhas quando ocorrem erros.



## Integração

Muitas vezes há uma tendência de tentar integração não incremental; isto é, construir o programa usando uma abordagem **big bang**.

Todos os componentes são combinados com antecedência.

O programa inteiro é testado como um todo.



## Big Bang Theory

Outra forma importante de realizar testes de integração é por meio da integração tipo big bang. Nesse caso, todas as unidades, componentes e módulos do sistema são integrados e testados de uma só vez, como se formassem uma única unidade.





## Big Bang Theory

No entanto, essa forma de teste é limitada. Se o processo mostrar que o sistema não funciona como deveria, o teste big bang não indica quais partes estão falhando na integração.

O teste big bang oferece uma resposta rápida quando o sistema funciona corretamente com todos os seus elementos.





## Por que não ?

Um novato no mundo do software pode levantar uma questão aparentemente legítima quando todos os módulos tiverem passado pelo teste de unidade:

“ Se todos funcionam individualmente, porque você duvida que funcionem quando estiverem juntos? ”

**Novamente,  
Por que não ?**





## ATIVIDADE

Atividade consiste em simular o desenvolvimento da aplicação através de módulos, cada módulo é de responsabilidade de um time(equipe) aplicando o método estudado anteriormente(TDD), que posteriormente serão integrados como um único sistema.

Assim podemos validar o questionamento levantado no slide anterior se o funcionamento unitário garante o funcionamento integral

Após isso podemos utilizar como estudo de caso para outras técnicas como CI/CD



## Descrição da Aplicação

A Universidade Tabajara “TecLearn TABAJARA” uma instituição das Organizações Tabajara, decidiu informatizar o controle de sua biblioteca. O sistema é para 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	Responsabilidade
Equipe 1 — Cadastro de Usuários	CRUD de usuários (criar, ler, atualizar, excluir), validações e autenticação básica de exemplo (opcional: token pré-simulado para chamadas internas).
Equipe 2 — Catálogo de Livros	CRUD de livros, controle de cópia/estoque, metadados (título, autor, ISBN, ano, edição), status de disponibilidade.
Equipe 3 — Empréstimo e Devoluções	Regras de negócio para empréstimos, verificação de disponibilidade, bloqueios por usuário, datas de devolução, multas (simuladas).
Equipe 4 — Relatórios	Geração de relatórios agregados (livros mais emprestados, usuários com mais empréstimos, empréstimos por período).



## 11 Parte 1 — Equipe 1: Cadastro de Usuários

### Visão Geral

Responsável por todo o gerenciamento dos usuários que utilizarão a biblioteca: alunos, professores e funcionários. Fornece dados para autenticação básica (simulada) e para o módulo de Empréstimo.

### Modelos de Dados (contrato) — Usuário

Campo	Tipo e restrições
id	inteiro (auto-incremento no banco) — obrigatório.
nome	string, 1..100 caracteres — obrigatório.
matricula	string alfanumérica, 5..20 caracteres, única — obrigatório.
tipo	string enum: {"ALUNO","PROFESSOR","FUNCIONARIO"} — obrigatório.
email	string, formato email válido, único — opcional, recomendado.
ativoDeRegistro	string ISO 8601 (data de registro) — obrigatório.
status	string enum: {"ATIVO","INATIVO","SUSPENSO"} — padrão "ATIVO".



## 12 Parte 2 — Equipe 2: Catálogo de Livros

### Visão Geral

Gerencia todo o acervo de livros e cópias (exemplares). Deve prover informação suficiente para que o módulo de Empréstimo saiba se um livro está disponível.

### Modelos de Dados

Campo	Tipo e restrições
bookId	inteiro (auto-incremento).
titulo	string, 1..200 caracteres
autores	array de strings (cada autor 1..100 caracteres)
ISBN	string 10 ou 13 caracteres, único quando presente
edicao	string, opcional
ano	inteiro (por exemplo 2020)
copiasTotal	inteiro $\geq 0$
copiasDisponiveis	inteiro $\geq 0$ (deve ser consistente com empréstimos)
status	nao obrigatório, pode ser: {"DISPONIVEL", "INDISPONIVEL"}



## 13 Parte 3 — Equipe 3: Empréstimo e Devoluções

### Visão Geral

Responsável pelas regras de negócio: reservar/executar empréstimos, verificar disponibilidade e status do usuário, calcular data prevista de devolução e multas (simples). Esse módulo consome os serviços de Usuários e Catálogo.

### Modelos de Dados

Campo	Tipo e restrições
loanId	inteiro (auto-incremento).
userId	inteiro (id do usuário, referência externa).
bookId	inteiro (id do livro).
copyId	opcional (se controlarem cópias individuais).
loanDate	ISO 8601 data/hora de empréstimo.
dueDate	ISO 8601 data/hora de devolução prevista (regras: padrão 14 dias para ALUNO, 30 dias para PROFESSOR, 14 para FUNCIONARIO).
returnDate	ISO 8601, se devolvido.
status	enum: {"ACTIVE", "RETURNED", "OVERDUE", "CANCEL
fine	numero (valor monetario simulado), default 0.



## 14 Parte 4 — Equipe 4: Relatórios

### Visão Geral

Gerar relatórios agregados usando dados dos outros módulos. Relatórios podem ser exportados em JSON e CSV. Importante: não reter dados privados; usar apenas os campos essenciais.

### Requisitos funcionais

- Cada relatório deve aceitar períodos (start/end) em ISO 8601; se não informado, usar últimos 30 dias.
- Limite máximo por query: 1000 registros; forçar paginação se exceder.
- Performance (simulada): responder relatório simples em até 5s na infra local; se mais lento, fazer paginação por padrão.
- Privacidade: não retornar dados sensíveis (e.g., emails completos) nos relatórios públicos — retornar apenas matrícula e nome.



# Integração







**OBRIGADO**

[dacio.francisco@unicesumar.edu.br](mailto:dacio.francisco@unicesumar.edu.br)