Introdução Arquitetura de Software: Projeto camada de dados

Disciplina : Projeto de Software

Profº Tadeu dos Reis Faria

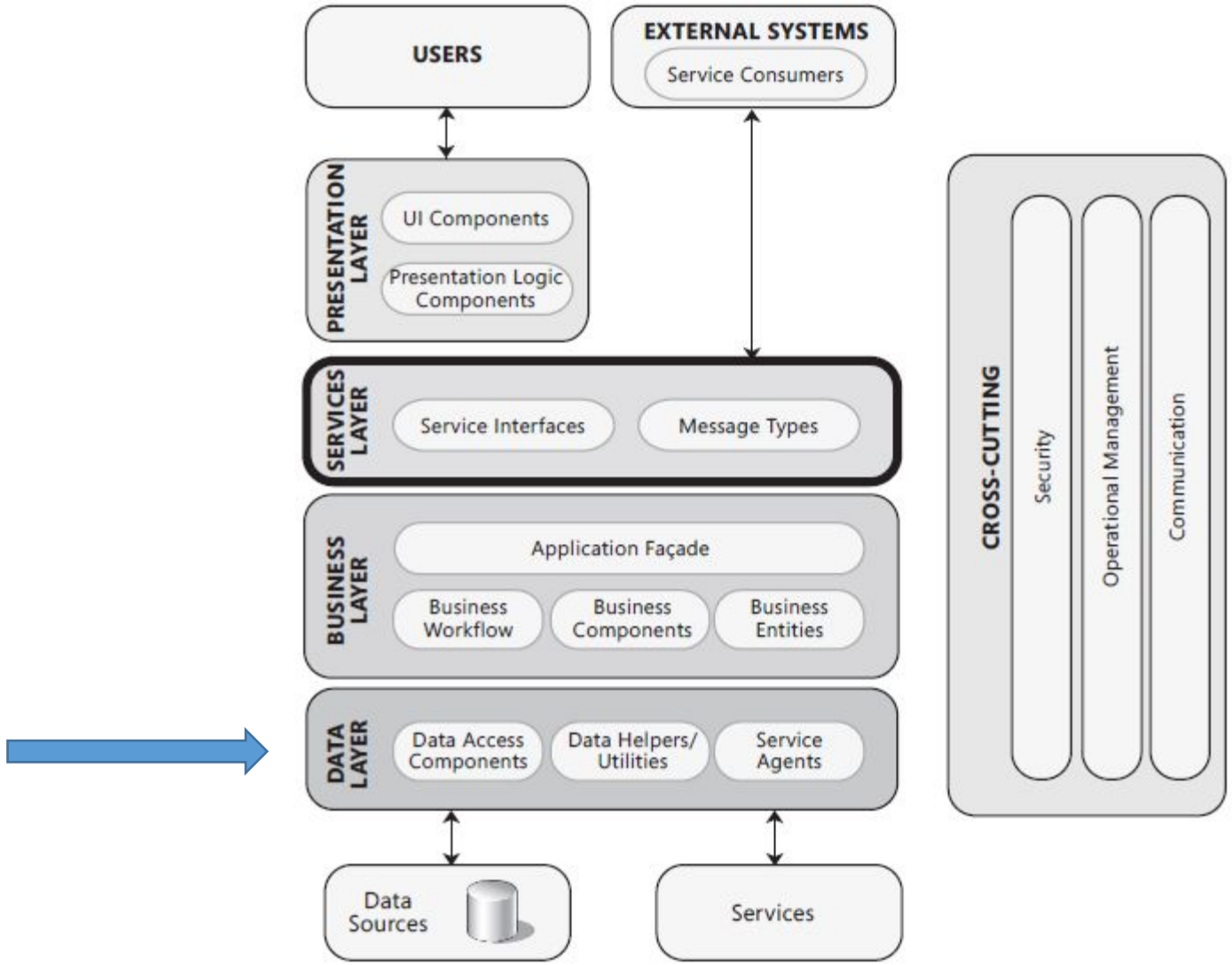
PUC Minas

INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA
UNIDADE EDUCACIONAL PRAÇA DA LIBERDADE
Bacharelado em Engenharia de Software

BIBLIOGRAFIA

- LARMAN, Graig. Utilizando UML e Padroes:Uma introdução a análise e ao projeto orientados a objetos. Porto Alegre: Bookman, 3ª Edição, 2007. capítulo 39
- BEZERRA, Eduardo. Princípios de analise e projeto de sistemas com UML.Rio de Janeiro: Campus, 2a Edição, 2007. capítulo 11
- GUEDES, Gilleanes T. A. UML 2 : uma abordagem prática. São Paulo: Novatec, 1a Edição, 2009.
- Microsoft Application Architecture Guide

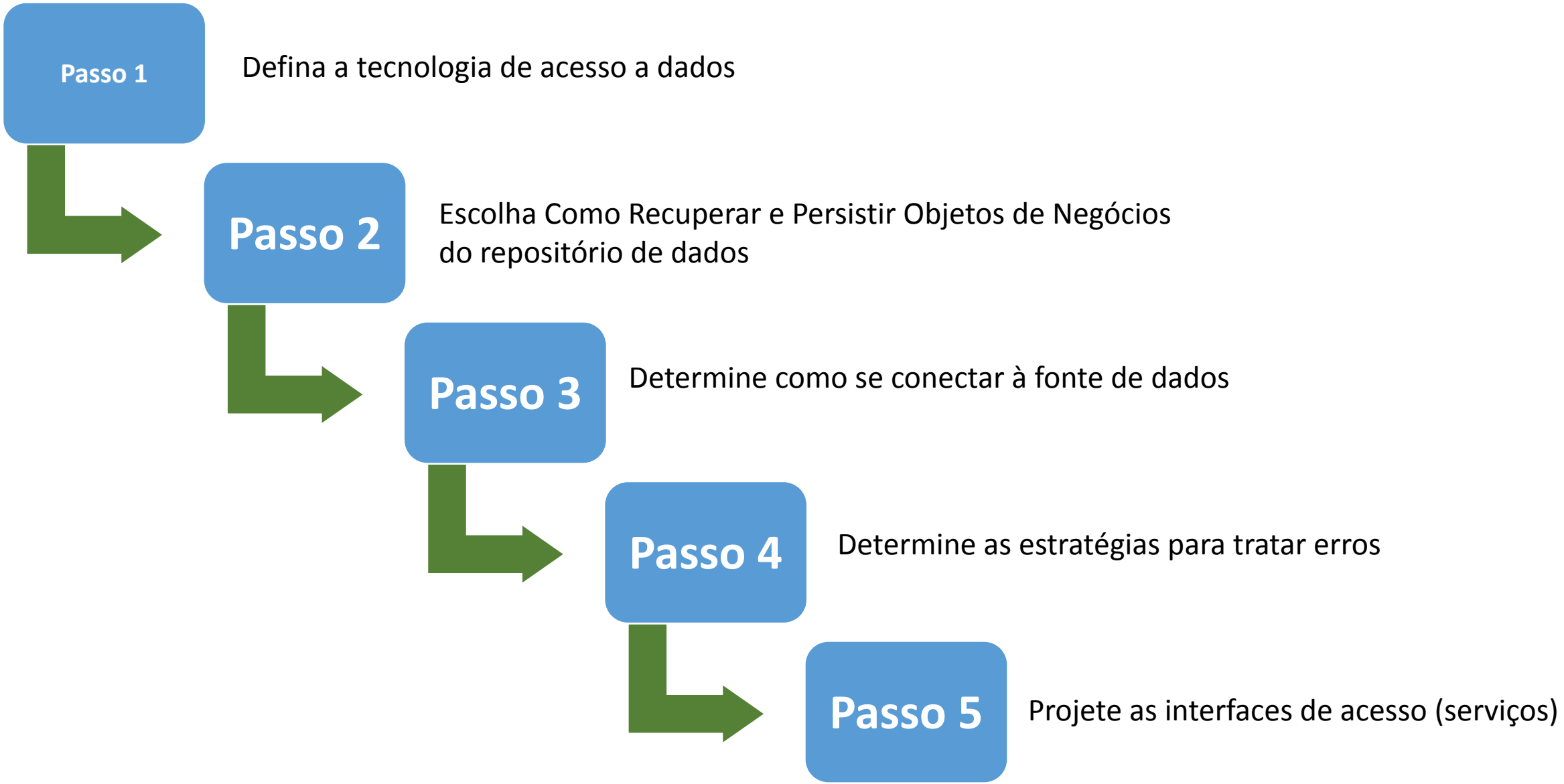
Projeto camada de dados



Projeto camada de acesso aos dados

- A Camada de dados agrupa classes que têm por finalidade prover criação, remoção, alteração e recuperação de dados persistentes.
- Não é o próprio mecanismo de persistência (banco de dados ou arquivo), mas um front-end que empacota o acesso a ele.
- O fluxo de mensagem é da Camada de Negócio para a Camada de dados.
- Benefício:
  - torna possível realizar alterações na forma de persistência de dados sem impacto para o restante do sistema.

Passos no design de acesso dados



Passos no design de acesso dados

Defina a tecnologia de acesso a dados

- Uso de um SGBDOO ou de um SGBDOR
- Acesso direto ao banco de dados
- Uso do padrão DAO (Data Access Object)
- Uso padrão Repository
- Uso de um framework ORM
- Variações ORM (Dapper)
- Uso do padrão Active Record
- Acesso NoSQL

Passos no design de acesso dados

Escolha como recuperar e persistir objetos de negócios do repositório de dados

- ORM
- Json
- XML

Passos no design de acesso dados

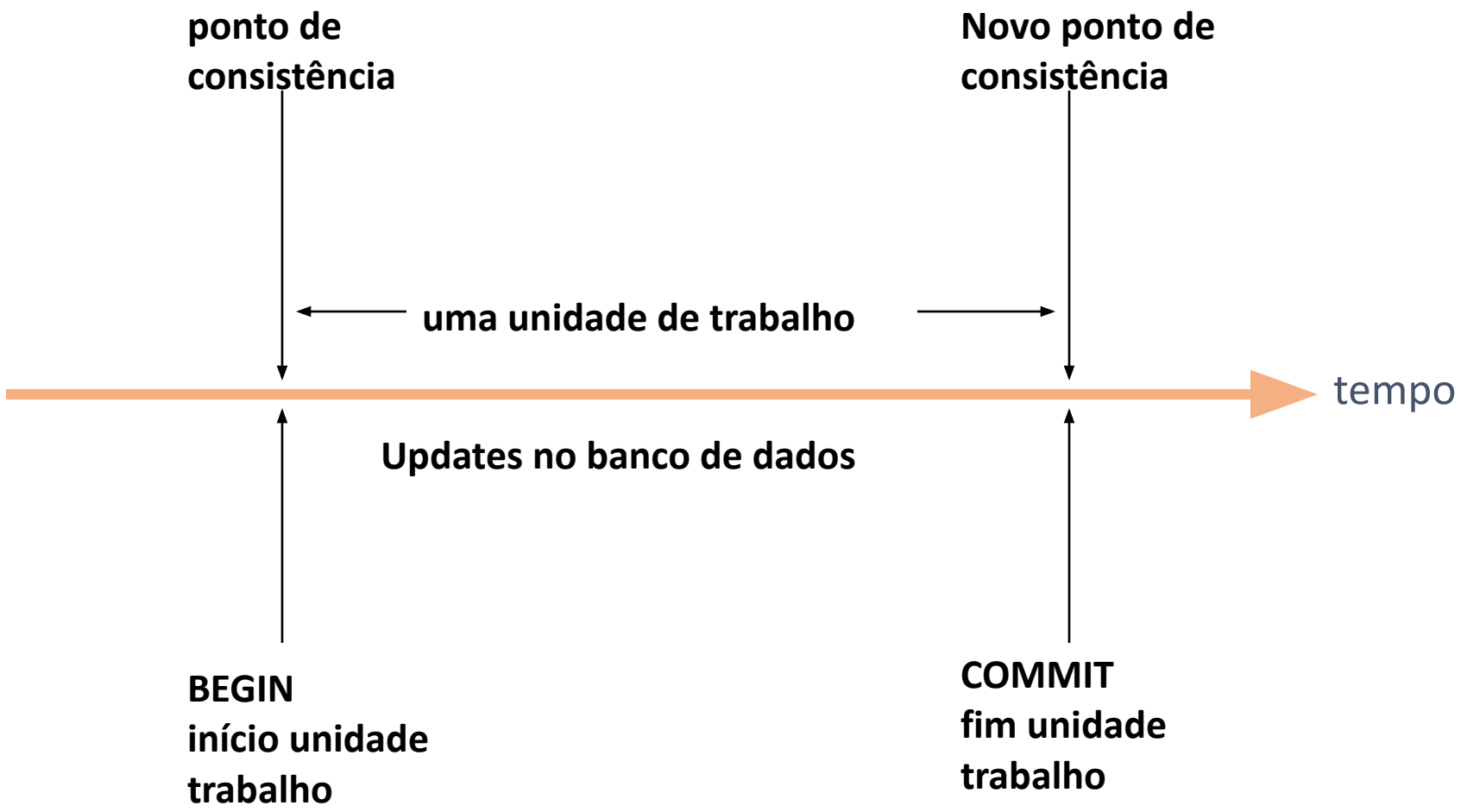
Determine como se conectar à fonte de dados

- Connections
- Connection Pooling
- Transactions and Concurrency

Passos no design de acesso dados:Transações

- Delimitada pelas instruções:
  - begin transaction e
  - end transaction
- A transação consiste em todas as operações executadas entre begin e end transaction
- Toda a comunicação com um banco de dados tem que ocorrer dentro de uma transação, não importa se você vai ler ou escrever dados

Passos no design de acesso dados:Transações



Passos no design de acesso dados: concorrência

Técnicas de controle concorrência: Técnicas de Travamento em Duas Fases : bloqueios compartilhados/ Exclusivos

- A trava tem três estados possíveis
  - Read-locked (compartilhado), travado para leitura
    - ✔ Outras transações podem ler o item
  - Write-locked (exclusivo), travado para gravação
    - ✔ Apenas o dono do travamento pode ler ou gravar o item
  - Unlocked, destravado
- Operações: read_lock(X), write_lock(X), unlock(X)
- A tabela de travamento (locks) terá quatro entradas
  - Nome do item de dados
  - Valor da trava
  - Número de leituras simultâneas
  - Transações responsáveis pelo travamento

Passos no design de acesso dados: concorrência

- Na tabela, o valor da trava é sempre read_locked ou write_locked (itens destravados não são mantidos na tabela)
- Se o valor for write_locked, então o número de transações envolvidas terá que ser 1 e a lista terá apenas a transação responsável pelo travamento
- Se o valor for read_locked, haverá um número qualquer >0 de transações e suas identificações formarão uma lista

Passos no design de acesso dados

Determine as estratégias para tratar erros

- Todas as exceções devem ser capturadas e repassadas para outras camadas somente se as falhas afetarem responsividade ou funcionalidade da aplicação
- Trate:
  - Exceptions
  - Retry Logic
  - Timeouts

Passos no design de acesso dados

Projete as interfaces de acesso (serviços)

- São os serviços de acesso aos dados
- Use ferramenta apropriada para adicionar uma referência de serviço
- Determine como o serviço será usada na aplicação ou por agentes externos