

Métricas para Avaliação de Persistência de Dados num Ambiente Objeto-Relacional

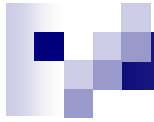
MAC499 - Trabalho de Formatura Supervisionado

Professor Orientador:
João Eduardo Ferreira

Autores:

Diogo Vernier dos Santos

Marcelo de Rezende Martins



Introdução - Motivação

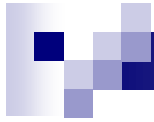
Nos projetos de software atuais:

- n O uso das **linguagens orientadas a objeto** cresce a cada dia.
- n O uso de **bancos de dados relacionais** consolida-se como a principal escolha.



Mais motivos...

- n Os **bancos de dados orientados a objetos** ainda não atendem as necessidades práticas totalmente.
- n Surge então a combinação de **persistir dados** em bd's relacionais e **construir a lógica** da aplicação utilizando linguagens orientadas a objeto.



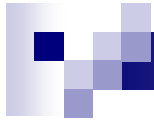
Persistência de Objetos

- n Objetos persistem além do tempo de execução da aplicação.
- n Objetos podem ser armazenados e recuperados em algum meio físico.



Modelo Relacional

- n Criado em 1969 por Edgar F. Codd.
- n Os dados são representados por **relações matemáticas**.
- n Uma relação é um subconjunto do produto cartesiano de vários conjuntos, ou seja, algo da forma $\langle s_1, s_2, \dots \rangle$ dados os conjuntos S_1, S_2, \dots e $s_i \in S_i$ para todo i .



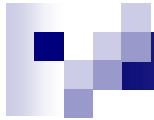
Mapeamento Objeto-Relacional

n Conhecido também como “Object Relational Mapping” (ORM).

É um conceito que define como fazer de forma consistente:

n A representação de objetos em tabelas de um BD relacional.

n A transformação do conteúdo das tabelas em objetos.



Qualidade da Persistência

Problemas comuns nas implementações de ORM:

- n Baixa qualidade da representação semântica.
- n Recuperação em larga escala ineficiente.

Como avaliar a qualidade da persistência?



Métricas e Testes

Missão do nosso trabalho:

- n Definir **métricas** para avaliar a qualidade da persistência de dados feita pela ORM.
- n **Testar** diferentes implementações de ORM utilizando essas métricas.
- n Elaborar uma **análise comparativa** das implementações a partir dos testes.



O que queremos avaliar?

- n Performance.
- n Eficácia da representação do modelo.
- n Capacidade de reuso.
- n Complexidade operacional.
- n Dependência gerada pela ferramenta.
- n Armazenamento físico.
- n Uso de memória.



Quem iremos avaliar?

Testaremos as seguintes implementações:

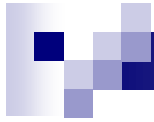
- n Toplink (JPA)
- n Hibernate (JPA)
- n Active Record (Ruby)



Métricas e Testes

Como está sendo feito:

1. Construir aplicações modelo.
2. Configurar ambiente de testes.
3. Elaborar métricas.
4. Testar aplicações.
5. Avaliar resultados perante as métricas.

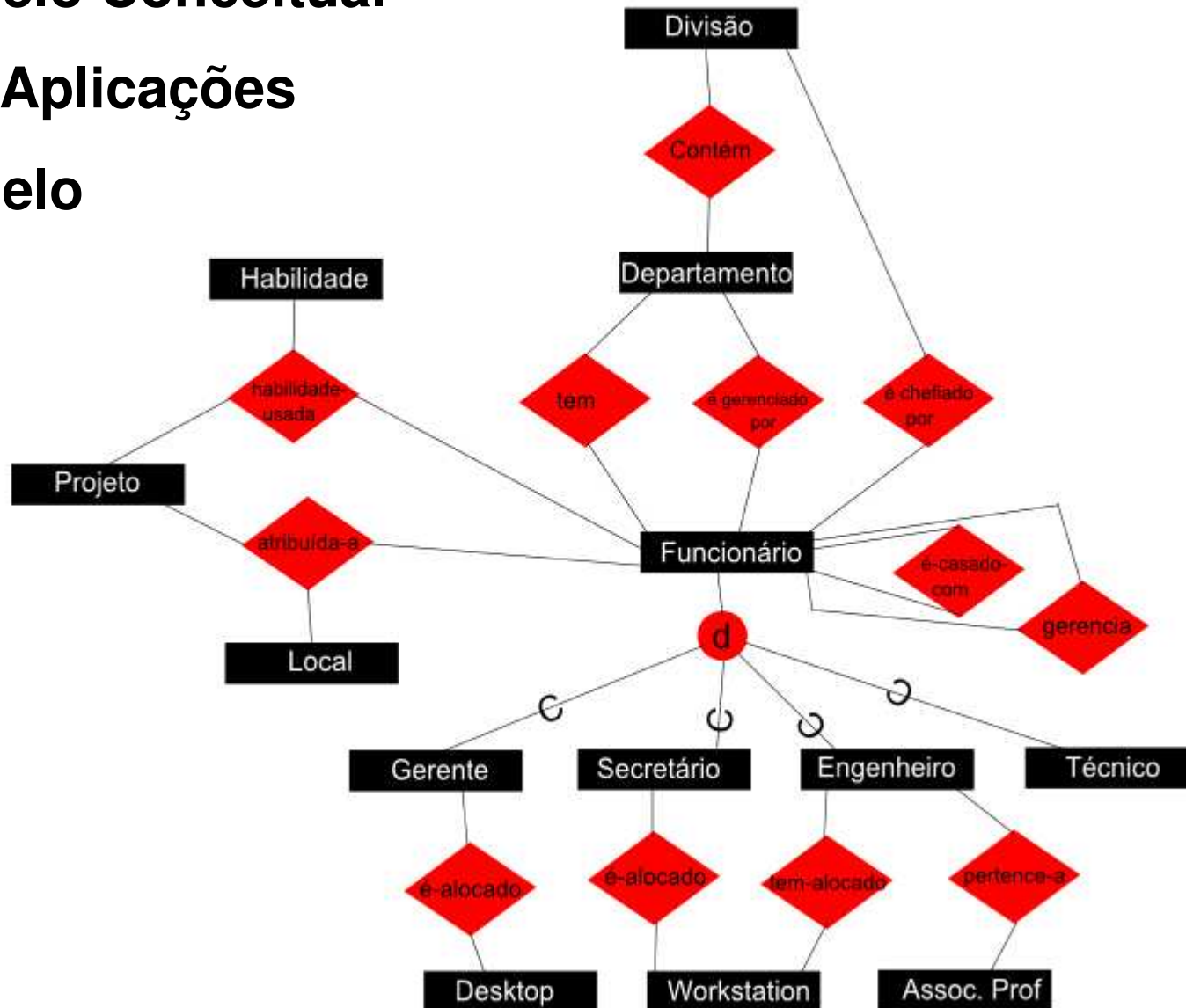


Fase 1 – Aplicações Modelo

- n Construção de aplicações modelo com um esquema comum de BD cada qual utilizando uma ferramenta específica para realizar o ORM.
- n Serão utilizadas dentro de uma plataforma comum de testes.

Modelo Conceitual das Aplicações

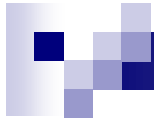
Modelo





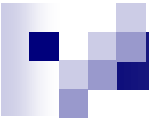
Fase 2 – Ambiente de Testes

- n **Poleposition**: é um **framework de testes** (GPL) para comparar bancos de dados relacionais e tecnologias de mapeamento objeto-relacional.
- n O Poleposition aplica o conceito de **circuito**.
- n Cada aplicação modelo percorre uma **volta** no circuito e o framework coordena a **corrida**.



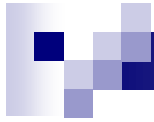
Fase 3 – Elaborar as Métricas

- n Definir maneiras seguras e eficazes de avaliar a **qualidade da persistência** de dados produzida num ambiente objeto-relacional.



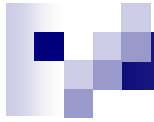
Fase 4 - Testes

- n A partir das métricas e do ambiente prontos, as aplicações irão desempenhar uma mesma sequência de operações (**voltas**) dentro do circuito.



Fase 5 - Análise Comparativa

- n Utilizar as métricas criadas para avaliar os resultados obtidos com os testes.
- n Gerar gráficos, relatórios, análises e conclusões.



Maiores informações em:

[n http://www.linux.ime.usp.br/~dvsantos/mac499/](http://www.linux.ime.usp.br/~dvsantos/mac499/)