



**Regras de Negócios**  
**Não castigue o seu**  
**servidor de aplicações!**

**Compartilhar  
experiências como  
programador e DBA**

**E criar alguma polêmica!**

# Pesquisa

- **Quantas camadas?**
- **Regras na Aplicação?**
- **Regras numa Camada Intermediária?**
- **Regras no Banco de Dados?**
- **Explain / Plano de Consulta?**

# Mas o que é?

**Regras do Negócio são requisitos e necessidades que serão implementados em um sistema de software, através de rotinas lógicas e matemáticas, definindo a sua forma de funcionamento.**

**Ou seja...**

**Elas definem como o negócio funciona e devem ser refletidas nos requisitos do Software.**

# Mas isso é novo?

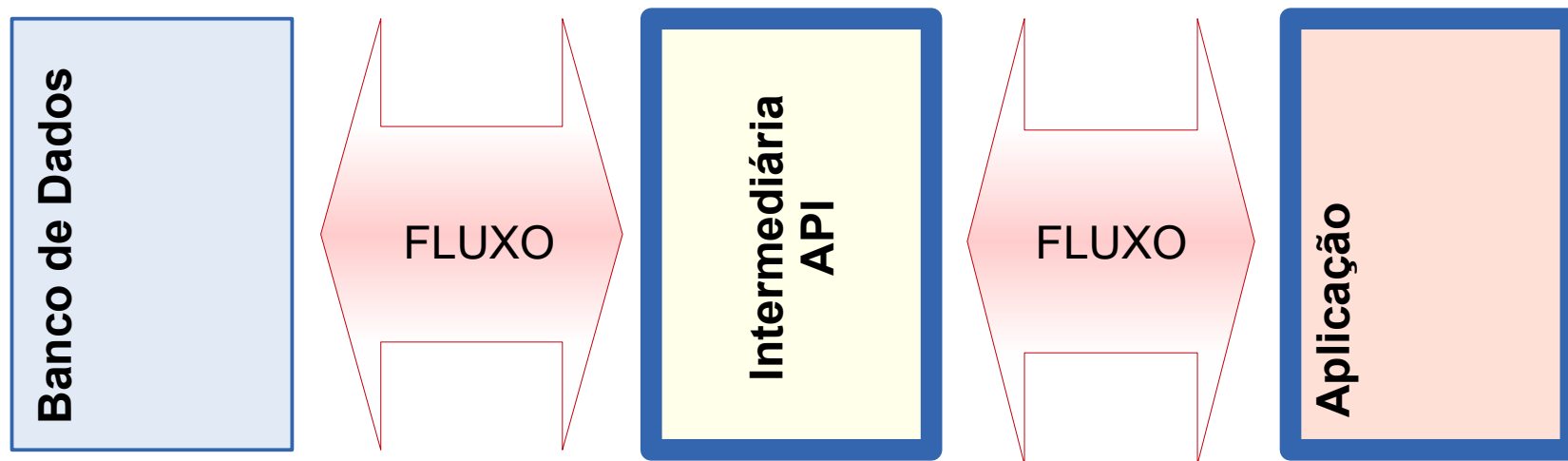
## Ou é antigo?

# HARDWARE

# SOFTWARE

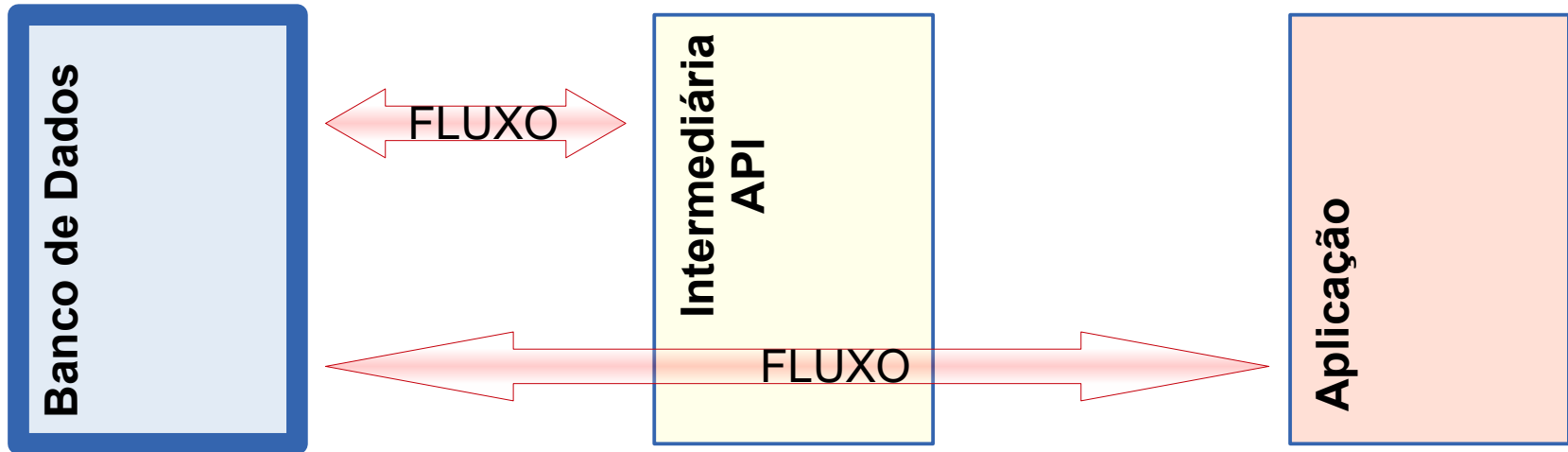


# Cenário Ruim



- Grande fluxo de informações entre as camadas;
- Alto consumo de memória;
- Alto consumo de CPU desnecessário;
- Perda de desempenho;
- Custos de infra mais altos;

# Cenário Bom



- Pequeno fluxo de Informações entre as camadas;
- Consumo correto de memória;
- Consumo correto de CPU;
- Bom desempenho e consistência transacional;
- Custos de infra menores;



**E as minhas regras  
de negócio?**

**E isso vai influenciar no  
desempenho?**

# Requisitos

- **Um bom projeto e dimensionamento da sua estrutura de dados, respeitando as Normas Formais;**
- **Costume de Concentrar regras e processamento de informações no banco de dados;**
- **Dimensionamento correto das Chaves de Referência (FK) e dos Índices;**
- **SGBD / Instância bem dimensionados e configurados;**

# Requisitos

- **Conhecimento mais profundo da linguagem e recursos de consultas;**
- **Conhecimento de Explain e Tuning de query;**
- **Conhecimento de Linguagem Procedural;**
  - ✓ PL/pgSQL
  - ✓ PL/PHP
  - ✓ PL/...
  - ✓ C

# DESVANTAGENS

# VANTAGENS

# DESVANTAGENS

- **Necessita de uma estrutura mais potente, podendo gerar mais custos;**
- **Não utiliza todos e melhores recursos do SGBD;**
- **Conhecimento mais profundo dos recursos do SGBD e do hospedador; (?)**
- **Pode gerar uma dependência maior do SGBD utilizado, dificultando migrações;**

# VANTAGENS

- **Melhora no desempenho geral;**
- **Melhora no custo de estrutura/hosting;**
- **Solução para os problemas de integridade referencial dos dados;**
- **Praticamente elimina a necessidade de migração de SGBD;**

# Fidelidade

- Transações explícitas curtas (quando usadas);  
begin;  
< comandos >  
commit;
- Regras bem definidas e distribuídas;
- Tentar concentrar regras lógicas e matemáticas pesadas, no banco de dados;
- Saber exatamente o que está sendo feito no banco de dados;

# Infidelidade

- Transações tem um custo, apesar de úteis;
- Transações longas (quando usadas);
- Regras que deveriam estar juntas, #sqn;
- Regras lógicas e matemáticas que causam um grande fluxo de dados entre as camadas;
- Não saber exatamente o que está sendo feito no banco de dados;



# Mais identificados

- ORM que não montam as melhores consultas;
- Abertura de transações para leituras;
- Regras lógicas e matemáticas que causam um grande fluxo de dados entre as camadas;
- Não saber exatamente o que está sendo feito no banco de dados;

# Linguagens Procedurais?

- Fácil aprendizado;
- Muitos recursos;
- Fácil programação;
- Captura e Definição de Erros Pessoais;
- Poder de decisão nas Triggers / Rules;

**# Mas ai o Banco de Dados vai sofrer!!**

# Cuidado com Locks

- Como funcionam;
- Podem gerar perda de desempenho;
- Podem gerar espera;
- São bons para a manutenção da integridade referencial dos dados;
- Evite os LOCK TABLES;

# O que mais pode ajudar?

- Evitar o overhead das conexões. Pools ajudam;
- No caso de inserts, agrupar em blocos;
- Usar sempre bind nas queries (query plan);
- Evitar deadlocks no BD;
- Em bancos grandes, particionamento;
- Usar constraints no banco de dados;

# CONCLUSÃO

- ▶ Os bancos de dados não são somente repositórios de dados;
- ▶ É perfeitamente possível e viável, concentrar mais regras de negócios e processamento na camada de dados, com melhor desempenho, trazendo tranquilidade para a sua aplicação;

# OBRIGADO!

e-Mail: [lucio@vorio.com.br](mailto:lucio@vorio.com.br)

Telegram: [@chiessilmb](https://t.me/@chiessilmb)

LinkedIN:

<https://www.linkedin.com/in/lucio-chiessi/>

