

Engenharia de Requisitos

Prof. Thiago G. Traue

Sumário	Relevância/ Aplicabilidade
1. Engenharia de Requisitos	<ul style="list-style-type: none"> • Aprender o que é um requisito • Entender quais são as fases da engenharia de requisitos
2. Tipos de requisitos	<ul style="list-style-type: none"> • Descobrir quais são os tipos de requisitos mais comuns
3. A relação dos bancos de dados e requisitos	<ul style="list-style-type: none"> • Entender a relação da engenharia de requisitos com o projeto do banco de dados
4. Modelo FURPS	<ul style="list-style-type: none"> • Conhecer um pouquinho mais da engenharia de requisitos • Aprender sobre o famoso modelo “FURPS”

Bibliografia

- Pressman, R., Maxim B., Engenharia de Software, 8ª. Ed, Bookman., 2016
- *C. A. Heuser*, Projeto de Banco de Dados. (6ª ed). Ed. Bookman, 2009
- IEEE Std. **610.12** IEEE Standard Glossary of Software Engineering Terminology. The Institute of Electrical and Electronics Engineers, 1990

Engenharia de Requisitos

- A engenharia de requisitos é uma disciplina da engenharia de software
- Na década de 70 os desenvolvedores notaram que faltava algo nos processos de desenvolvimento de software
- É uma das atividades mais importantes no processo de desenvolvimento de um software

Engenharia de Requisitos



Dilbert by Scott Adams, 2006

(Dilbert, 2006)

Mas o que é um requisito?

E segundo nosso bom e velho amigo, o dicionário, é...

“condição necessária para a obtenção de certo objetivo, ou para o preenchimento de certo fim”

Mas o que é um requisito?

Segundo o IEEE:

1. Uma condição ou capacidade necessitada por um usuário para resolver um problema ou alcançar um objetivo.
2. Uma condição ou capacidade que deve ser satisfeita ou possuída por um sistema ou componente do sistema para satisfazer um contrato, um padrão ou uma especificação.
3. Uma representação documentada de uma condição ou capacidade como em (1) ou (2).

Mas o que é um requisito?

- É, basicamente, o que o sistema desenvolvido precisa fazer
- Exemplo:
 - O sistema precisa armazenar de forma segura os dados dos usuários
 - O sistema deve permitir que o usuário altere seus dados
 - Os usuários não podem alterar seu CPF
 - Usuários menores de idade precisam informar o CPF dos pais
 - ...

Mas o que é um requisito?

- Mas também, como o projeto oferece seus “serviços”
- Exemplo:
 - O sistema deve ficar indisponível aos finais de semana
 - O sistema deve ser desenvolvido com plataformas OpenSource
 - O sistema precisará ter um backup a cada 10 minutos
 - O tempo de resposta para o usuário deve ser inferior à 2 segundos
 - ...
- Veremos, em breve, que isso é uma categorização de requisitos

Engenharia de Requisitos

- Normalmente há um processo bem definido para essa tarefa, dentro do projeto
- De maneira geral, podemos dividir a Engenharia de Requisitos em três grandes fases:
 - Descoberta
 - Especificação
 - Validação e verificação

Engenharia de Requisitos - Descoberta

- O desafio para os designers é desenvolver uma **compreensão** do domínio da aplicação e das **necessidades** específicas
- O processo de descoberta dos requisitos é a **primeira** tarefa de um engenheiro de requisitos
- Normalmente, através de [muitas] consultas com o cliente
- Podem vir de outras fontes: Especialistas trabalhando no projeto, por exemplo
- Existem várias técnicas para a extração dos requisitos

Engenharia de Requisitos - Descoberta

- Na fase de descoberta os detalhes técnicos não podem influenciar o processo
- Levantar requisitos em áreas de conhecimento diferente da nossa (T.I.) é uma das tarefas mais **complexas** do projeto
- É muito comum novos requisitos surgirem com o projeto em andamento
 - Por isso precisamos definir bem, também, o processo usado para gerir o projeto

Engenharia de Requisitos - Especificação

Não confunda com a fase de validação!

- Na fase de especificação os requisitos são validados com o cliente
- É confeccionado um documento formal com as especificações
- Essa fase é importante para detectar inconsistências
- Auxilia também a comunicação entre grupos de distintos dos *stakeholders*
- Aqui também há a categorização e priorização dos requisitos

Engenharia de Requisitos – Validação e verificação

- Este processo é contínuo, ou seja, ocorre até o final do projeto
- Aqui os requisitos são checados constantemente se estão sendo atendidos
- É preciso ter uma pessoa responsável para isso durante o projeto todo
- Serve, também, para garantir o processo de atendimento às funcionalidades

Engenharia de Requisitos – Validação e verificação

- Validação: “Estou desenvolvendo o produto certo?”
- Verificação: “Estou desenvolvendo o produto corretamente?”

Engenharia de Requisitos

- Precisamos organizar o levantamento de requisitos
- Há muitas formas de especificar e levantar requisitos
- Essencialmente, um requisito pode ser FUNCIONAL ou NÃO-FUNCIONAL

Mas isso é assunto para próxima aula... =)

Tipos de Requisitos

Prof. Thiago G. Traue

Sumário	Relevância/ Aplicabilidade
1. Engenharia de Requisitos	<ul style="list-style-type: none"> • Aprender o que é um requisito • Entender quais são as fases da engenharia de requisitos
2. Tipos de requisitos	<ul style="list-style-type: none"> • Descobrir quais são os tipos de requisitos mais comuns
3. A relação dos bancos de dados e requisitos	<ul style="list-style-type: none"> • Entender a relação da engenharia de requisitos com o projeto do banco de dados
4. Modelo FURPS	<ul style="list-style-type: none"> • Conhecer um pouquinho mais da engenharia de requisitos • Aprender sobre o famoso modelo “FURPS”

Bibliografia

- Pressman, R., Maxim B., Engenharia de Software, 8ª. Ed, Bookman., 2016
- *C. A. Heuser*, Projeto de Banco de Dados. (6ª ed). Ed. Bookman, 2009
- IEEE Std. **610.12** IEEE Standard Glossary of Software Engineering Terminology. The Institute of Electrical and Electronics Engineers, 1990

Tipos de requisitos

- Podemos classificar os requisitos.
 - FUNCIONAIS
 - NÃO-FUNCIONAIS
- Isso nos ajuda a separa-los corretamente dentro do processo de análise

Tipos de requisitos - Funcionais

- Representam tudo que o sistema deve fazer → Metas e necessidades
- Não devem ser considerados aspectos tecnológicos aqui
 - Ex.: Banco de dados Oracle; Servidor Apache; ...
- Podem possuir prioridades diferentes
- Podem representar uma ordem de ação (ex.: Edição de campo só pode ser feita se o dado já estiver guardado)
- Requisitos funcionais ↔ Casos de Uso (UML)

Tipos de requisitos - Funcionais

- Normalmente possuem um identificador único (RF01, RF02 ...)
- Podem ser representados de diversas formas
 - Tabelas
 - Listas
 - Diagramas
 - ...

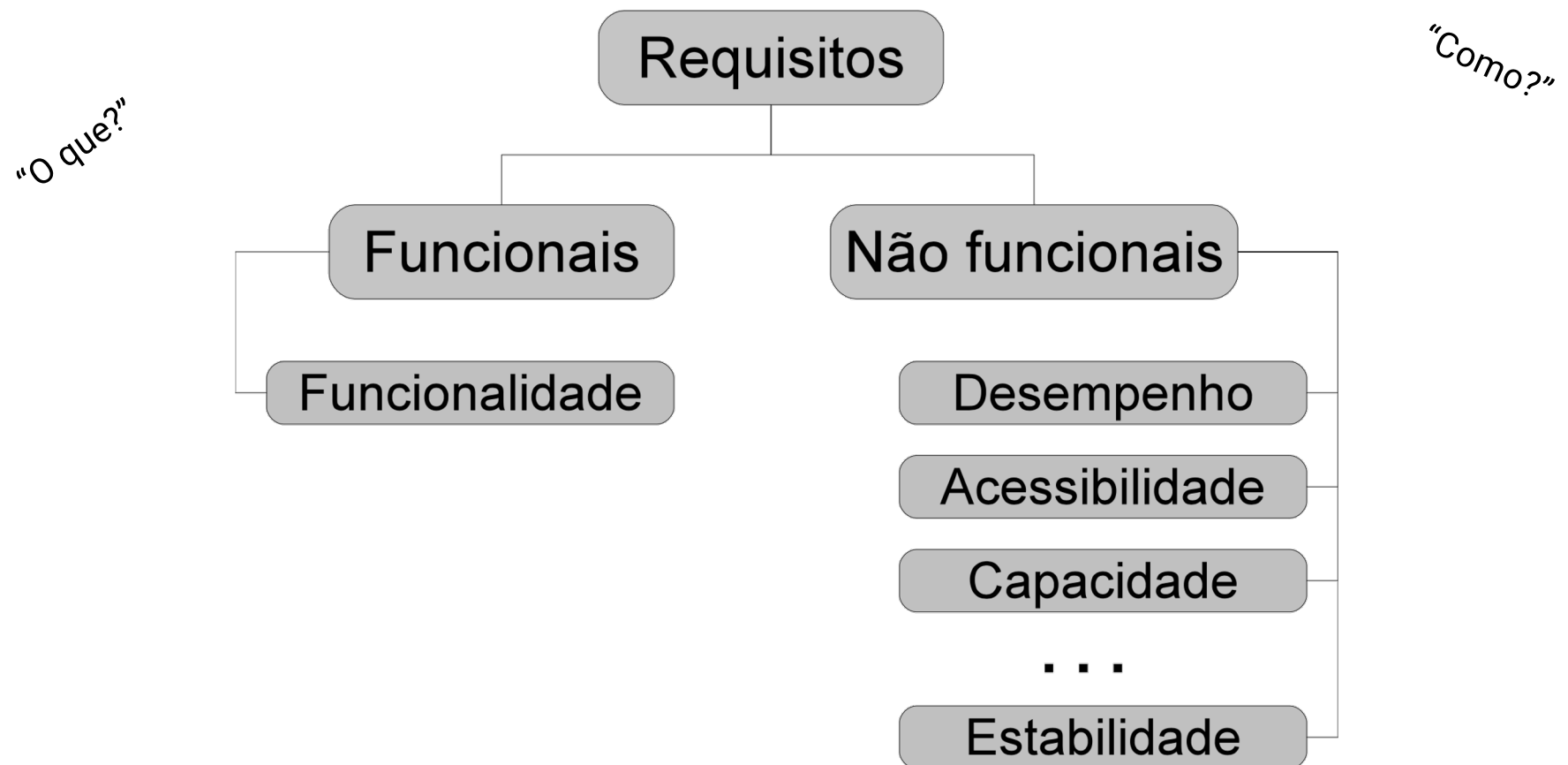
Tipos de requisitos – Não-Funcionais

- Normalmente categorizados como requisitos de
 - Desempenho
 - Acessibilidade
 - Capacidade
 - Segurança
 - ...
- Podem representar aspectos mais técnicos da aplicação

Tipos de requisitos – Não-Funcionais

- Podem influenciar na arquitetura da aplicação

Tipos de requisitos



O modelo FURPS para requisitos

Prof. Thiago G. Traue

Sumário	Relevância/ Aplicabilidade
1. Engenharia de Requisitos	<ul style="list-style-type: none"> • Aprender o que é um requisito • Entender quais são as fases da engenharia de requisitos
2. Tipos de requisitos	<ul style="list-style-type: none"> • Descobrir quais são os tipos de requisitos mais comuns
3. Modelo FURPS	<ul style="list-style-type: none"> • Conhecer um pouquinho mais da engenharia de requisitos • Aprender sobre o famoso modelo “FURPS”
4. A relação dos bancos de dados e requisitos	<ul style="list-style-type: none"> • Entender a relação da engenharia de requisitos com o projeto do banco de dados

Bibliografia

- Pressman, R., Maxim B., Engenharia de Software, 8ª. Ed, Bookman., 2016
- *C. A. Heuser*, Projeto de Banco de Dados. (6ª ed). Ed. Bookman, 2009
- IEEE Std. **610.12** IEEE Standard Glossary of Software Engineering Terminology. The Institute of Electrical and Electronics Engineers, 1990

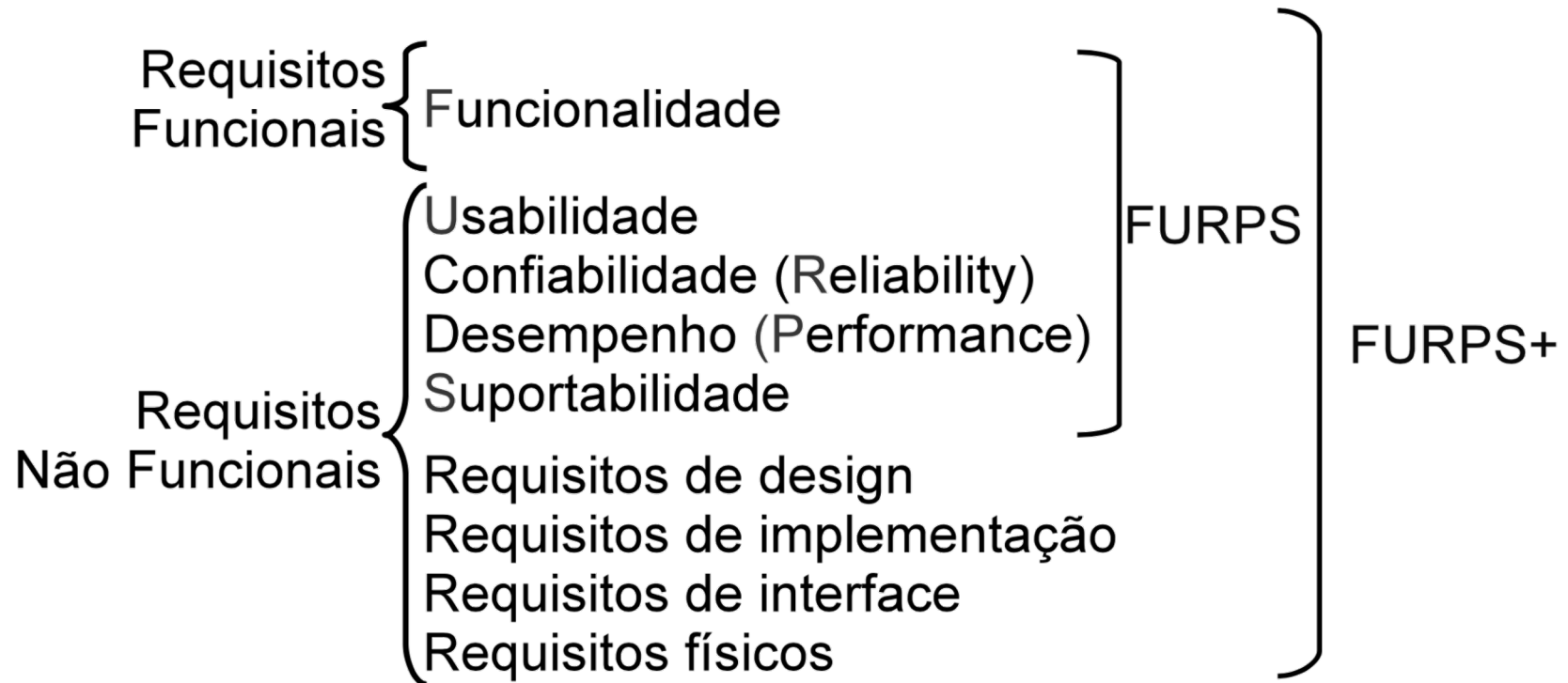
FURPS

- Mas o significa FURPS? → **F**uncionalidade, **U**sabilidade, **R**eliabilidade (confiabilidade), **P**erformance, **S**uportabilidade
- É uma forma de categorizar melhor os requisitos, especialmente os não funcionais
- Existe, também o FURPS+

FURPS

- Possuem uma grande flexibilidade
- Podem ser modificados para melhor atenderem os requisitos de uma aplicação.
- Cada projeto é único e deve usar modelos que se adaptem à ele e não o oposto!

FURPS e FURPS+



FURPS

- **Funcionalidade:** Especifica as funcionalidades do sistema
- **Usabilidade:** Normalmente associados à interface de usuário (UI) e experiência do usuário (UX). Pode ter subcategorias, por exemplo: prevenção de erros, estética e design, ajudas, padrões

FURPS

- **Confiabilidade (Reliability):** Referem-se à integridade do software. Os requisitos a serem considerados normalmente são: frequência e gravidade de falhas, possibilidade de recuperação, extensão de uma falha (valorização/sobrevivência), previsibilidade etc.
- **Performance (Desempenho):** Referem-se ao desempenho do sistema, nomeadamente, como por exemplo tempo de resposta de uma determinada funcionalidade, consumo de recursos (energia, RAM, CPU, cache, etc.), capacidade e escalabilidade.

FURPS

- **Suportabilidade** : Os requisitos de suportabilidade agrupam características, como: testabilidade, adaptabilidade, manutenibilidade, compatibilidade, configurabilidade, escalabilidade entre outros.

FURPS

- Essa forma de classificar os requisitos ajuda na documentação
- O processo de desenvolvimento de software precisa ser documentado
- Uma forma de documentar os requisitos, aqui é fazendo com que cada requisito seja “rastreável”. Por exemplo...

FURPS

- **Identificador único:** Cada requisito precisa possuir um código que o identifique que seja único;
- **Título:** Um título que resuma sua essência;
- **Categoria:** Cada requisito deve possuir uma categoria dentre as categorias aplicáveis de requisitos no projeto, visando sua rastreabilidade;

FURPS

- **Casos de uso associados:** Cada requisito pode conter uma lista de casos de uso relacionados. Cada caso de uso deve ser identificado com um identificador único;
- **Prioridade:** Uma escala de prioridade ou importância deste requisito para o sistema (ex.: Essencial, desejável, entre outros);
- **Descrição:** Descrição textual detalhada do requisito
- **Requisitos não funcionais relacionados:** Se o requisito for funcional, ele pode conter uma lista requisitos não funcionais relacionados

A relação com os Bancos de Dados

Prof. Thiago G. Traue

Sumário	Relevância/ Aplicabilidade
1. Engenharia de Requisitos	<ul style="list-style-type: none"> • Aprender o que é um requisito • Entender quais são as fases da engenharia de requisitos
2. Tipos de requisitos	<ul style="list-style-type: none"> • Descobrir quais são os tipos de requisitos mais comuns
3. A relação dos bancos de dados e requisitos	<ul style="list-style-type: none"> • Entender a relação da engenharia de requisitos com o projeto do banco de dados
4. Modelo FURPS	<ul style="list-style-type: none"> • Conhecer um pouquinho mais da engenharia de requisitos • Aprender sobre o famoso modelo “FURPS”

Bibliografia

- Pressman, R., Maxim B., Engenharia de Software, 8ª. Ed, Bookman., 2016
- *C. A. Heuser*, Projeto de Banco de Dados. (6ª ed). Ed. Bookman, 2009
- IEEE Std. **610.12** IEEE Standard Glossary of Software Engineering Terminology. The Institute of Electrical and Electronics Engineers, 1990

Requisitos X Bancos de Dados

- Já sabemos que a especificação de requisitos pode influenciar diretamente no resultado final do projeto
- Requisitos errados → Software errado
- Software errado → Cliente ☹️

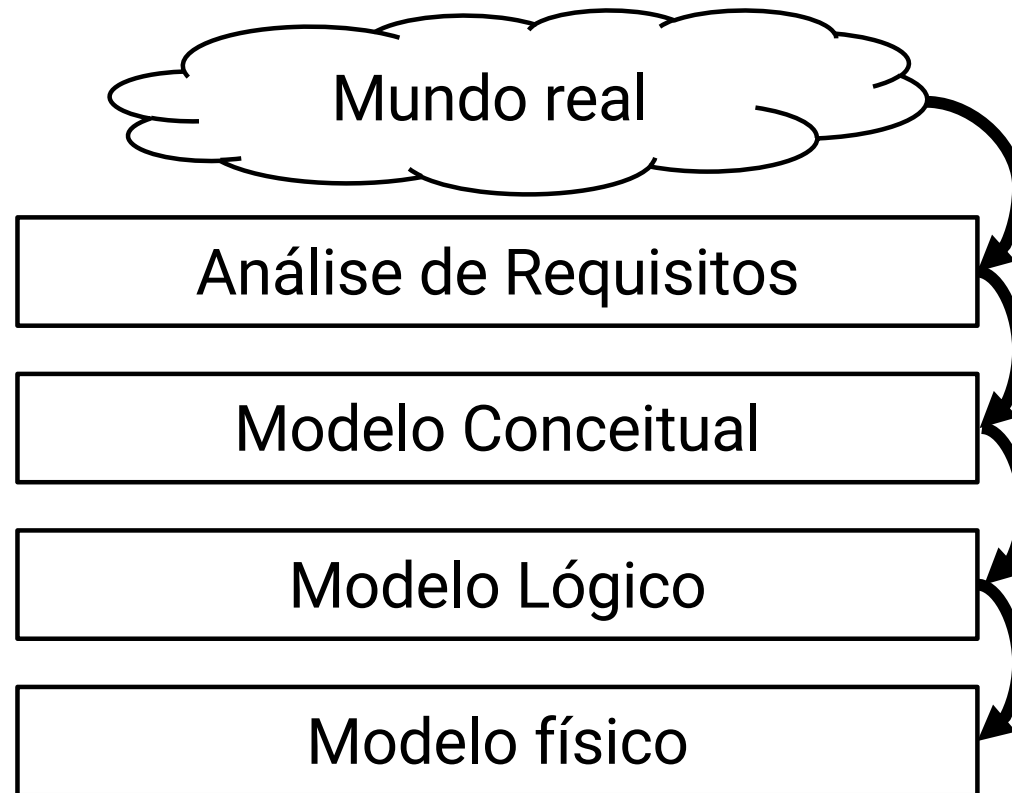
Requisitos X Bancos de Dados

- Ok, mas qual a relação dos requisitos com os bancos de dados?
- Os bancos de dados também podem ser modelados de forma incorreta se os requisitos estiverem incorretos!
- Lembre-se: “Dado é o novo petróleo!” (Clive Humby, 2006)
- Requisito errado → Dado gradado de forma incorreta
- Dado guardado de forma incorreta → Cliente ☹

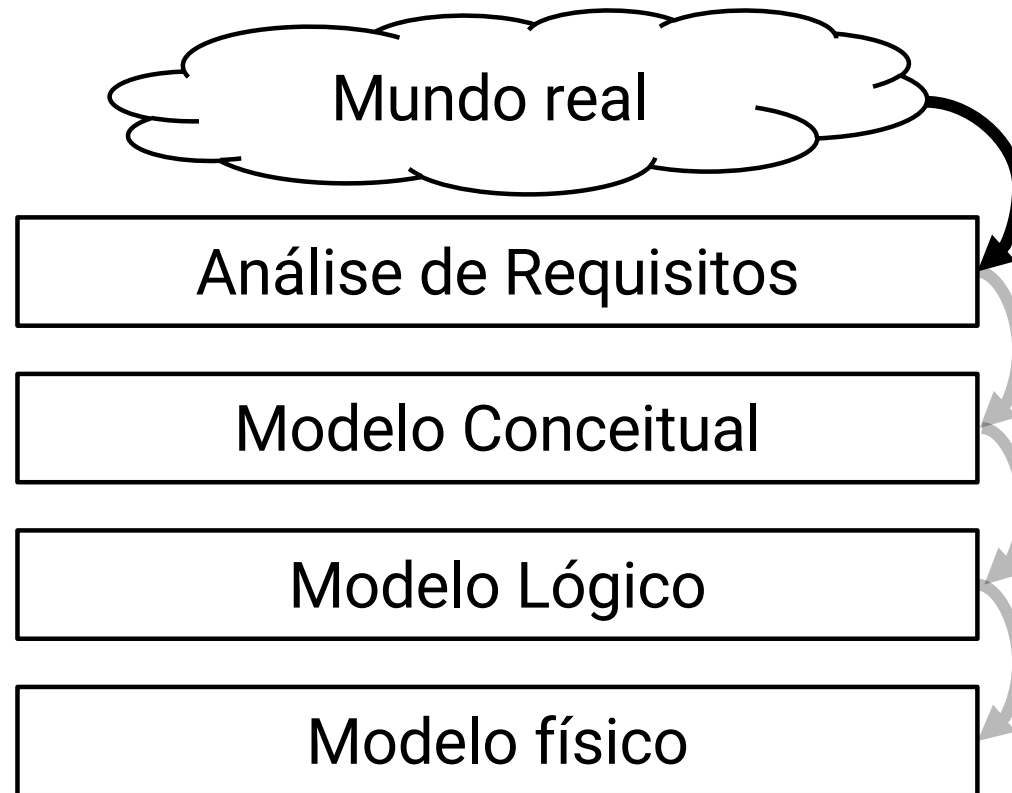
Requisitos X Bancos de Dados

Vamos relembrar um pouquinho do “Projeto de Banco de Dados”...

Requisitos X Bancos de Dados



Requisitos X Bancos de Dados



Requisitos X Bancos de Dados

- Os engenheiros de requisitos devem trabalhar juntos com os engenheiros de dados
- O processo deve ocorrer junto
- Para cada requisito funcional extraído, deve-se considerar quais dados estão envolvidos
- O processo de validação deve ser contínuo

Requisitos X Bancos de Dados

Exemplo:

“RF0643 – Os usuários são identificados por e-mail ou CPF

Prioridade: Alta”

Dados envolvidos:

- Dados do usuário em geral
- Sabemos, então que e-mail e CPF formam uma dupla que não podem se repetir na estrutura do banco!

Requisitos X Bancos de Dados

Exemplo:

“RF0644 – Os usuários podem gerar extratos detalhados de consumo

Prioridade: Média”

Dados envolvidos:

- Agora sabemos que a geração de relatórios tipo “extrato” possuem um relacionamento direto com o CPF (ou e-mail) do usuário.

Requisitos X Bancos de Dados

Exemplo:

“RF0645 – Usuários administradores podem gerar qualquer tipo de extrato”

Dados envolvidos:

- Agora sabemos que há níveis de usuários diferentes, tipos de extratos diferentes e há uma relação de permissão entre essas entidades