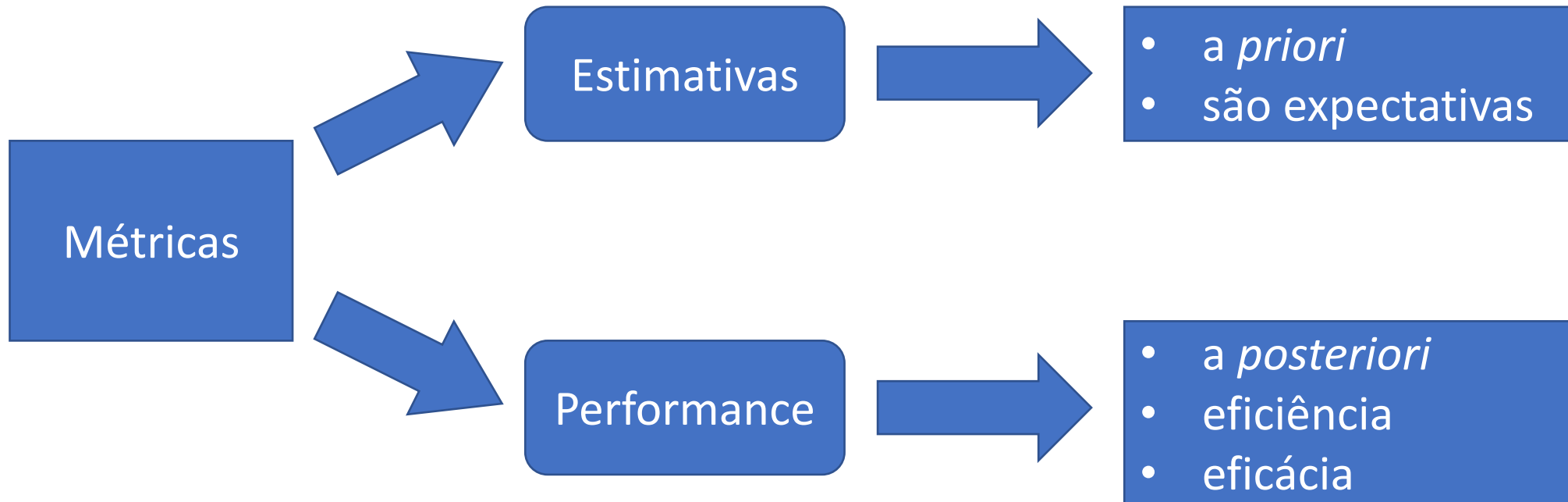


Métricas de Software

METRICAS – Servem para definir precisamente uma característica de forma numérica (quantitativa). Inferências sobre os processos de trabalho ou produtos que traduzem:



Fonte: Própria

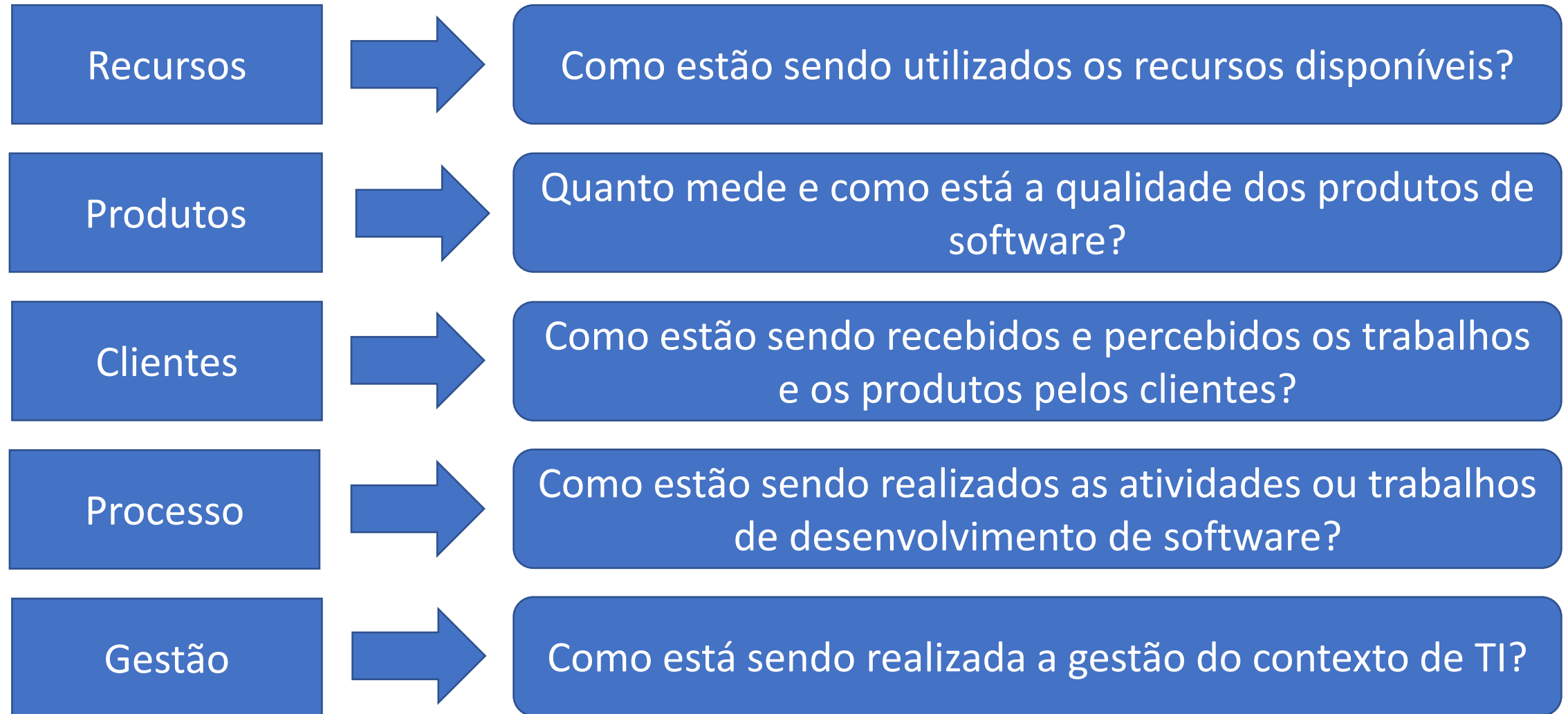
Qual a motivação que nos leva a utilizar métricas?

- “Não se pode gerenciar o que não se pode medir.” - Tom De Marco;
- “Se você não sabe para onde quer ir, pode seguir qualquer caminho. Se você não sabe onde está, não adianta ter um mapa, que ele não vai ajudar!” - Roger Pressman;
- “Quando você pode medir o que você está falando, e expressá-lo em números, você sabe algo sobre isso; mas quando você não consegue medir, quando você não pode expressá-lo em números, o seu conhecimento é algo frágil e insatisfatório: é talvez o princípio do conhecimento.” - Lord Kelvin.

Por que medir software?

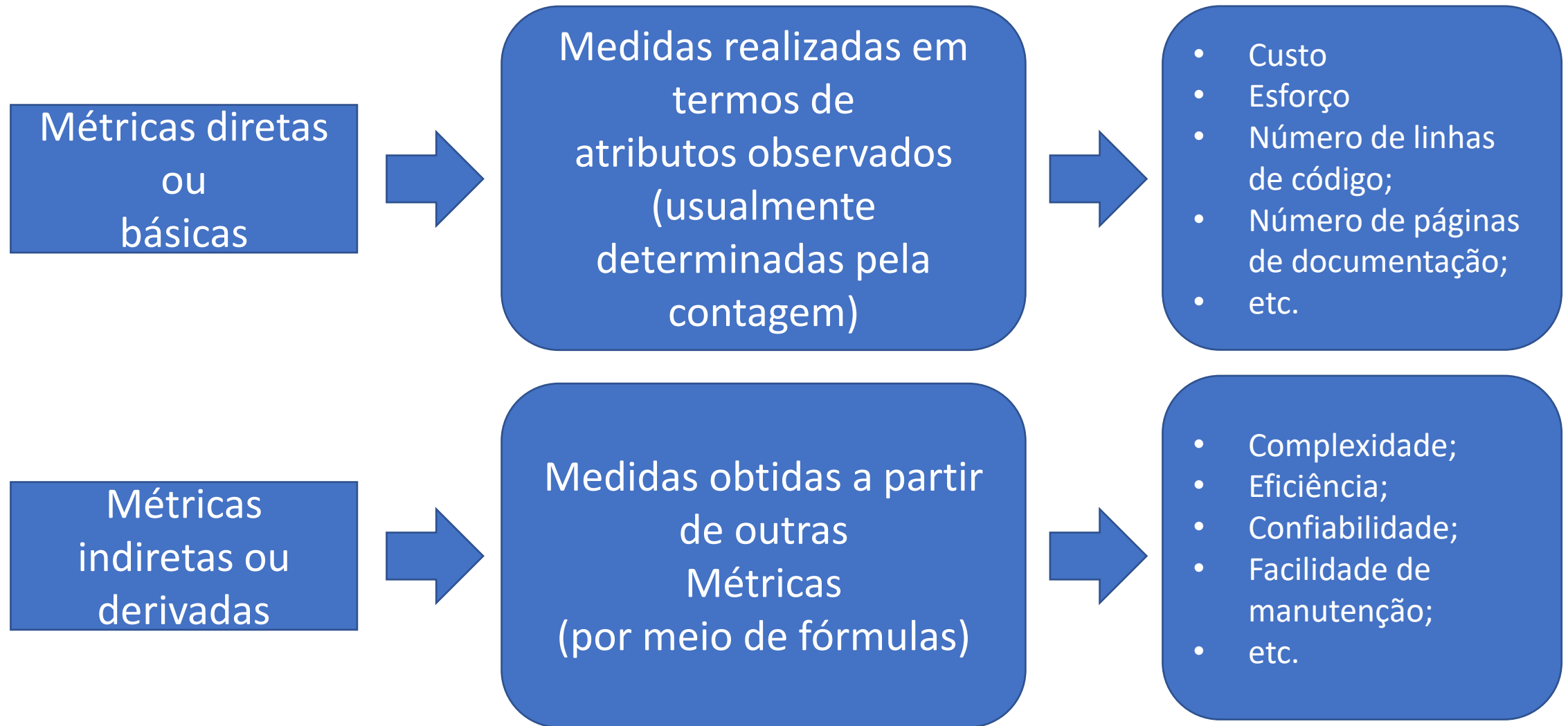
- Entender e melhorar o processo de desenvolvimento de software, indicando a qualidade de um produto de software e suportando análise de produtividade x qualidade;
- Melhorar a gerência de projetos, principalmente a estimativa de custos, cronograma e recursos;
- Melhorar o relacionamento com clientes, mitigar, por exemplo, frustrações e pressões de cronograma;
- Avaliar a aquisição de pacotes, gerenciar contratos de software e remunerar fornecedores;
- Apoiar a gerência de escopo do projeto;
- Benchmarking;

O que podemos medir?



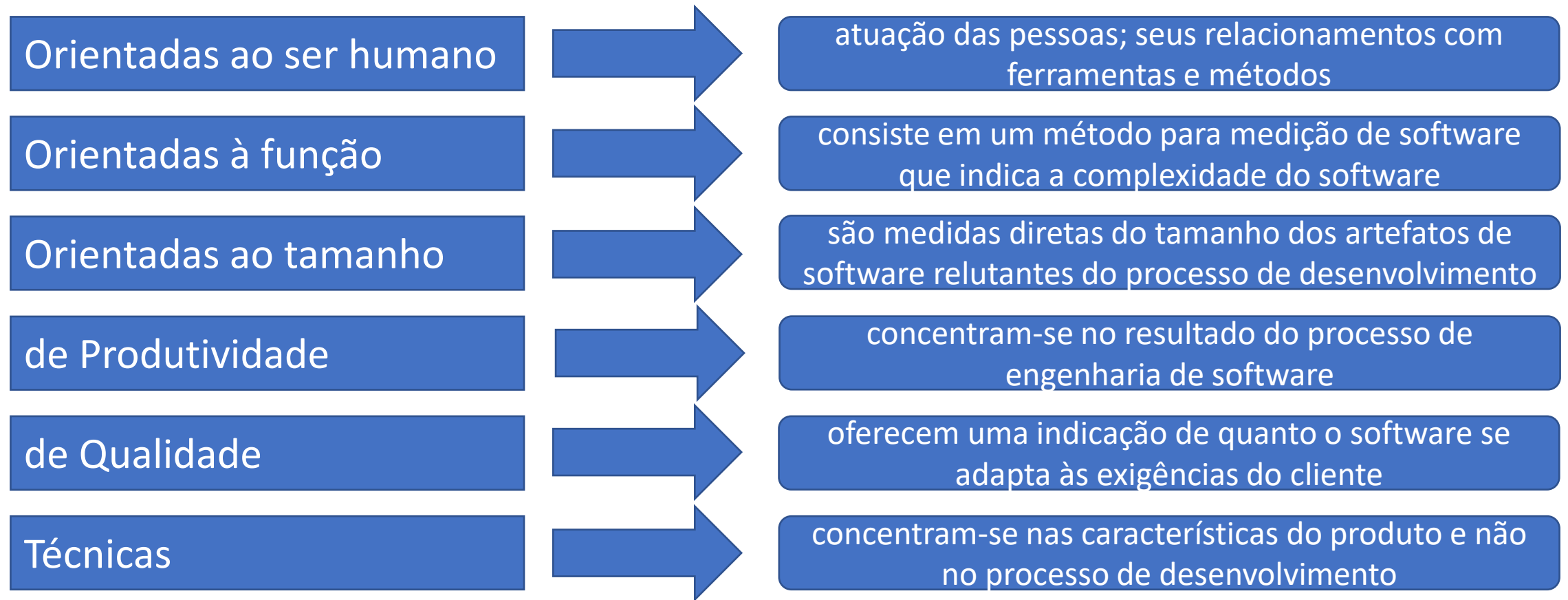
Fonte: Própria

Tipos de Métricas



Fonte: Própria

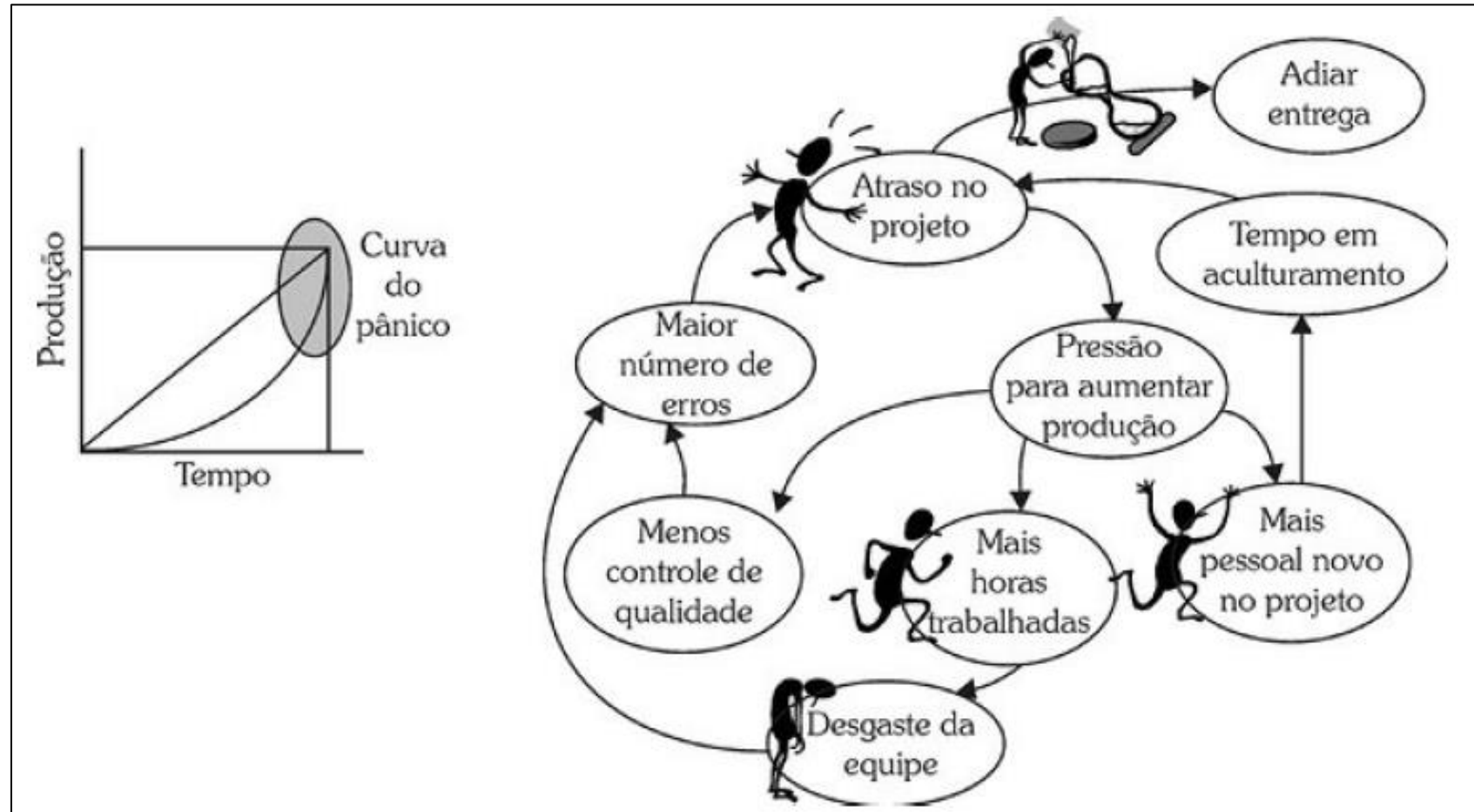
Classificação das Métricas



Fonte: Própria

A Curva do Pânico

- Segundo [1], o resultado da medição tem papel de permitir a comunicação efetiva não só entre os vários indivíduos e organizações envolvidas no projeto, mas também com os que possam ser afetados por ele;
- A penalidade pela falta desses mecanismos de apuração ou pela sua não-utilização frequente é enfrentar a dinâmica representada na figura ao lado, dinâmica dos 99% concluídos.

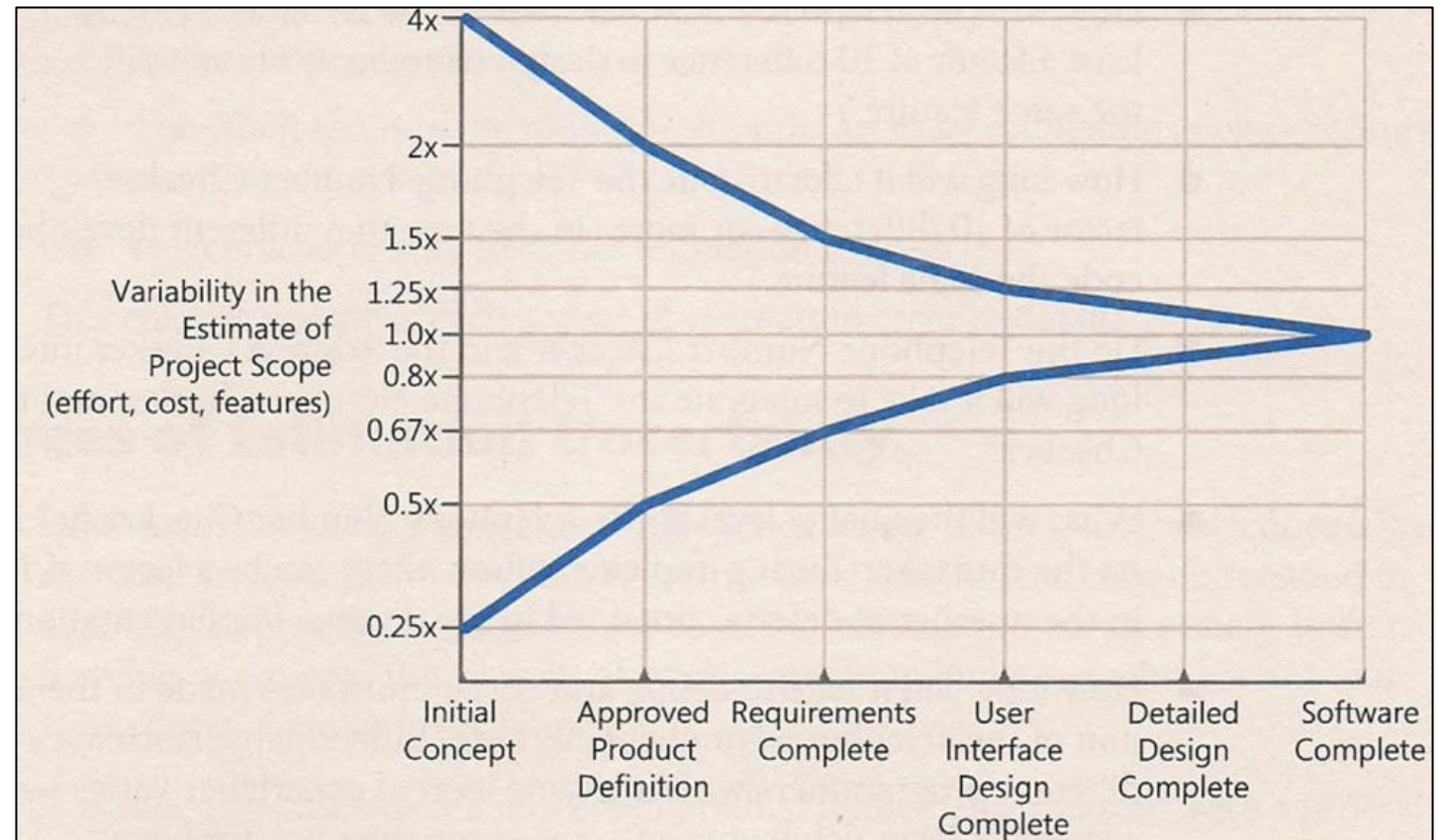


A Dinâmica do Pânico

Fonte [1]

O Cone da Incerteza

- De acordo com Steve McConnell [2], o Cone de Incerteza mostra que durante a fase inicial de um projeto a estimativa de tempo varia de 60% a 160%;
- Isto é, um projeto com expectativa de duração de 10 semanas pode durar de 6 a 26 semanas;
- Numa fase mais avançada do projeto, as estimativas poderão variar para + ou para - 15%;
- Desta forma uma estimativa de 10 semanas poderá se concretizar entre 7 a 13 semanas.

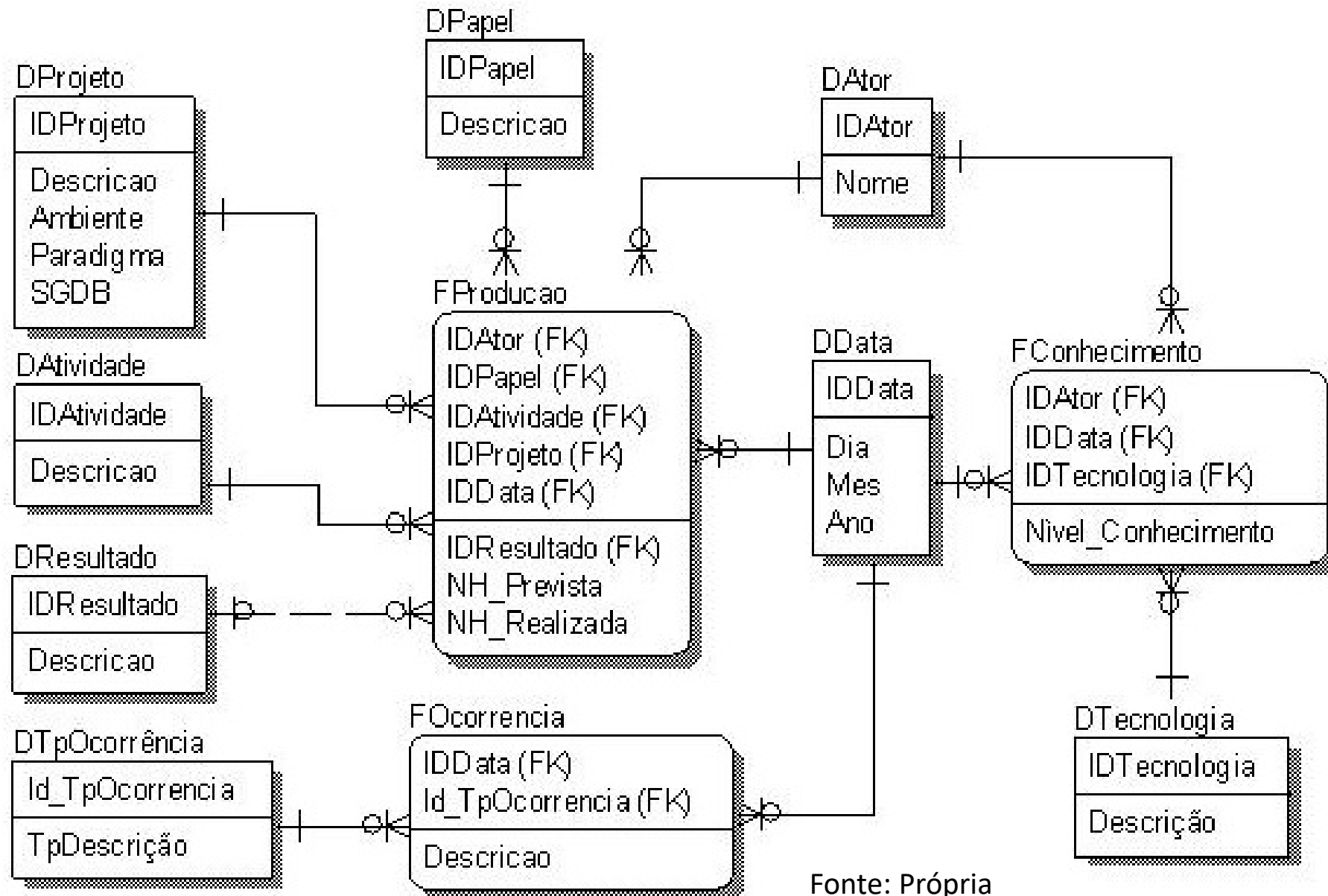


O Cone da Incerteza
Fonte [2]

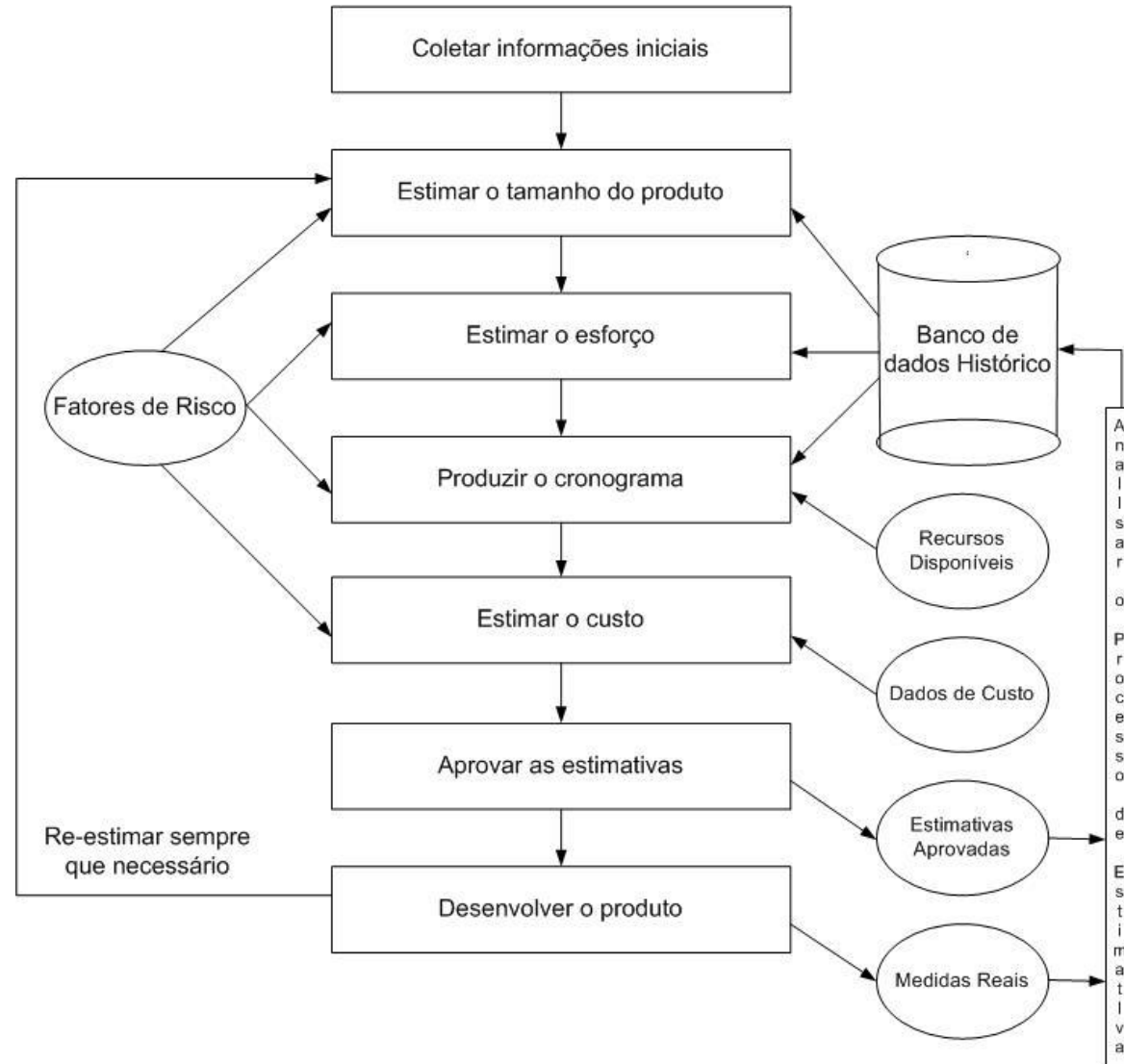
Banco de Dados Histórico

- “Aqueles que não aprendem com o passado estão condenados a repeti-lo.” - Stephen King - Escritor Americano
- O Banco de Dados Histórico é um fator que reduz o risco das estimativas;
 - Estimativas podem ser feitas com maior segurança;
 - Prazos podem ser estabelecidos para se evitar dificuldades passadas;
 - Riscos globais podem ser reduzidos.

Banco de Dados Histórico - Exemplo



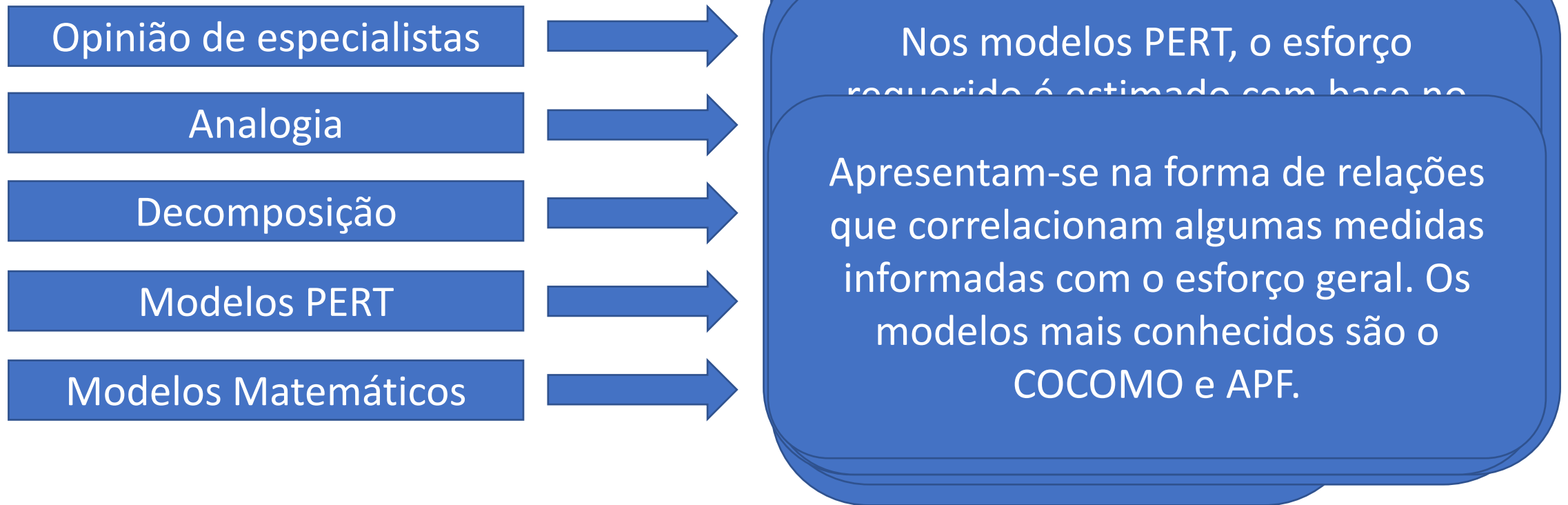
Processo de Estimativa de Software



Fonte: Adaptado de [6]

Classificação das Técnicas de Estimativa

Existem muitas técnicas de estimativa de software disponíveis, sendo classificadas em técnicas:



Técnicas de Estimativa

- Delphi
- Análise de Pontos de Função (APF)
- COCOMO
- Wideband-Delphi
- Use Case Point (UCP)
- COCOMO II
- Planning Poker
- Agile COCOMO II

Delphi

- A técnica Delphi é usada para obter consenso entre especialistas preservando o anonimato dos mesmos. Passos:
 1. Define-se o facilitador que fica responsável por compilar as informações dos questionários respondidos pelos especialistas;
 2. Define-se critérios de consenso ou de término das rodadas;
 3. Selecione os especialistas com base no tema sendo analisado;
 4. Cria-se o questionário para enviar para os especialistas. Por exemplo: Questionário para levantar o custo e o esforço de um projeto.
 5. O facilitador organiza as respostas recebidas, e cria um sumário das respostas para submeter aos especialistas de modo a buscar uma revisão da resposta inicial e assim conseguir um consenso entre eles.
 6. Caso não seja obtido um consenso, repete-se o passo 2 até que seja obtido o consenso ou que algum critério pré-definido seja alcançado, como por exemplo, o número de rodadas.
- Desvantagem do método: o tempo dispendido e o custo associado, já que podem ser necessárias várias rodadas para se obter consenso.

Wideband-Delphi

- Wideband-Delphi é uma variação da técnica Delphi. O objetivo desta técnica é produzir uma estimativa precisa e imparcial. Trata-se de uma técnica estruturada de Julgamento Especialista e é essencialmente baseada em procedimentos com múltiplos passos, a saber:
 1. O coordenador apresenta aos especialistas participantes as especificações do projeto proposto, como também qualquer outra informação pertinente (como dados em projetos semelhantes que foram completados no passado);
 2. O coordenador convoca os especialistas para uma reunião de discussão do projeto;
 3. Cada especialista anota sua estimativa inicial num formulário previamente distribuído, de forma individual e anônima;
 4. O coordenador e sua equipe compilam os dados e preparam um resumo das respostas e solicitam nova interação, sem solicitar as razões das estimativas;
 5. O coordenador convoca os especialistas para uma reunião de discussão sobre os pontos onde as estimativas variam de forma considerável;
 6. Cada especialista, de forma individual e anônima, anota sua nova estimativa num formulário previamente distribuído para interações. Esta etapa, assim como as duas anteriores, acontece quantas vezes for necessário, por exemplo, até que um determinado critério de parada seja atingido.

Cartão Wideband-Delphi

Name <i>Mike</i>		Date <i>4/3/2004</i>		Estimation form <i>1/1</i>				
Goal statement <i>To estimate the time to develop prototype for customers A & B</i>		Units <i>days</i>						
Category		<input checked="" type="checkbox"/> goal tasks <input checked="" type="checkbox"/> quality tasks <input type="checkbox"/> waiting time <input type="checkbox"/> project overhead						
WBS# or priority	Task name	Est.	Delta 1	Delta 2	Delta 3	Delta 4	Total	Assumptions
<i>1</i>	<i>Interview customers (A+B)</i>	<i>3</i>	<i>+2</i>	<i>+1</i>				<i>Needs off-site trip</i>
<i>2</i>	<i>Develop requirements docs</i>	<i>6</i>	<i>+5</i>	<i>-2</i>	<i>+1</i>			<i>Start from scratch</i>
<i>3</i>	<i>Inspect requirements docs</i>	<i>1</i>	<i>+2</i>	<i>+2</i>	<i>-2</i>			<i>Team of 4 BSAs</i>
<i>4</i>	<i>Do rework</i>	<i>1</i>	<i>+4</i>					
<i>5</i>	<i>Prototype design</i>	<i>20</i>	<i>-3</i>	<i>4</i>	<i>-2</i>			<i>Includes DB</i>
<i>6</i>	<i>Test design</i>	<i>5</i>	<i>+3</i>					<i>20% exists now</i>
	Delta		<i>+3</i>	<i>+5</i>	<i>-3</i>			
	Total	<i>36</i>	<i>49</i>	<i>54</i>	<i>51</i>			

Cartão estimativas Wideband Delphi Fonte:

<https://flylib.com/books/en/1.32.1.28/1/>

Planning Poker

- O Planning Poker é uma técnica muito aplicada em projetos que utilizam metodologias ágeis, cujo objetivo é estimar o esforço necessário para o desenvolvimento das funcionalidades de uma aplicação;
- o Planning Poker é uma espécie de jogo de cartas usado para estimar o tempo que cada tarefa precisará para ser cumprida. Para isso, cada carta do jogo recebe o número de uma sequência de valores.
- Em equipes Scrum, normalmente a sequência é baseada numa parte da sequência de Fibonacci modificada, que é a seguinte:
também apresenta os seguintes símbolos:
 - **0**: é utilizado para itens que já foram estimados;
 - **½**: usado quando a *user story* é muito pequena, mas não se encaixa nos valores anteriores;
 - **1, 2 e 3**: se referem a itens ainda não estimados;
 - **5, 8 e 13**: usados para estimar itens de tamanho médio;
 - **20, 40 e 100**: são usados quando o item é grande e requer refinamento da *user story*;
 - **∞**: é usado quando a complexidade é muito alta e não há uma estimativa para ela;
 - **?**: Dúvidas;
 - **xícara de café**: esse símbolo é usado quando o item é muito complexo e pode levar horas de trabalho que serão gastas em cada item. Eles são usados para itens do *backlog*.



Fonte: Própria

Planning Poker - Etapas

1. Estimativa individual

- Iniciada a reunião do *Team Scrum*, a pessoa que atua como *Product Owner* seleciona um item descrito no *product backlog*, descrevendo-o para toda a equipe;
- O item é discutido entre as partes e cada integrante do time de desenvolvimento faz a sua estimativa individual, escolhendo a carta do baralho que julga ser a mais adequada;
- As cartas são reveladas. Se todas as pessoas tiverem feito a mesma escolha, a estimativa do item é finalizada. Caso contrário, segue-se para o próximo passo;

2. Exposição de opiniões

- Integrantes que indicaram o maior e o menor valor devem defender seus pontos de vista, explicando o motivo pelo qual escolheram as cartas em questão;
- Com essa nova reflexão, cada pessoas faça uma nova escolha de estimativa. As cartas são mostradas novamente e, se foram todas iguais, a estimativa é encerrada. Se não, a equipe segue para a próxima etapa;

Planning Poker - Etapas

3. Decisão baseada no consenso

- Em geral, após a primeira rodada, a tendência é que as escolhas dos valores das cartas sejam mais próximos uns dos outros, o que facilita a estimativa. Por isso, nesse momento, a pessoa que assume a função de *Scrum Master* pode fazer uma escolha entre:
 - escolher o maior valor, caso sejam próximos o suficiente;
 - calcular a média entre as estimativas de cada carta mostrada;
 - ou fazer novas rodadas de estimativas individuais até que a equipe toda escolha a mesma carta;
- É importante ressaltar que o *Planning Poker* valoriza o consenso, sendo interessante envolver e analisar a opinião da equipe para fazer essa escolha.

COCOMO

- Constructive Cost Model, também conhecido como COCOMO 81, é um modelo de estimativa do tempo de desenvolvimento de um software;
- Está obsoleto, e foi substituído por COCOMO II e pelo Agile COCOMO;
- O modelo COCOMO foi criado por Boehm (1981) a partir de um estudo empírico sobre sessenta e três projetos na empresa TRW Aerospace;
- Os programas examinados tinham de 2 a 100.000 LOC e eram escritos em linguagens tão diversas quanto Assembly e PL/I;
- COCOMO consiste em três implementações:
 - Básico: É um modelo estático que calcula o esforço de desenvolvimento de software e seu tempo de desenvolvimento, em função do tamanho de linhas de códigos desenvolvidas.
 - Intermediário: Calcula o esforço de desenvolvimento de software em função do tamanho do programa, que inclui custo, avaliação subjetiva do produto, hardware, pessoal e atributos de projeto.
 - Avançado: São incorporadas características da versão intermediária com uma avaliação de impacto de custo em cada passo de todo o projeto.

COCOMO Básico

$$E = {}_{ab}(KLOC)^{bb}$$

$$D = {}_{cb}(E)^{db}$$

$$P = E / D$$

Onde E é o esforço aplicado pela pessoa no mês, D é o tempo de desenvolvimento em meses cronológicos, KLOC é o número calculado de linhas de código para o projeto (expressado em milhares), e P é o número das pessoas necessário. Os coeficientes ab, bb, cb e db são dados na seguinte tabela:

Projeto de Software	ab	bb	cb	db
Orgânico	2.4	1.05	2.5	0.38
Semi-Destacado	3.0	1.12	2.5	0.35
Embutido	3.6	1.20	2.5	0.32

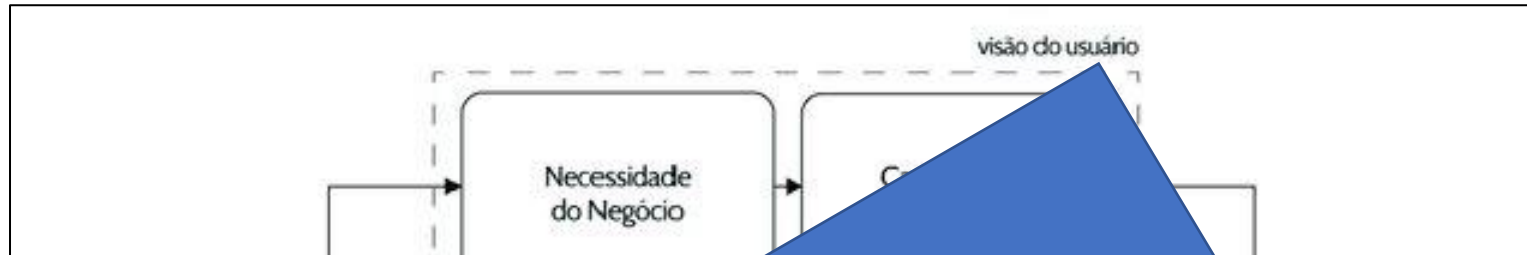
Análise de Pontos de Função - APF

- Ponto de função foi definido em 1977 por Allan Albrecht [5] na IBM, e é uma unidade de medida de software reconhecida pela ISO/IEC 20926 para estimar o tamanho de um sistema de informação baseando-se na funcionalidade percebida pelo usuário do sistema, independentemente da tecnologia usada para implementá-lo [1];
- A técnica da análise de pontos de função surgiu como alternativa às métricas baseadas em linhas de código (LOC);
- Allan Albrecht foi encarregado de estudar a produtividade dos projetos de software desenvolvidos por uma unidade da IBM, sendo que esses projetos tinham sido desenvolvidos em linguagens de programação distintas, tornando inviável uma análise conjunta de produtividade usando a métrica de linhas de código;
- O método é mantido atualizado por grupos de usuários da APF que cooperam internacionalmente:
 - IFPUG (*International Function Point Users Group*) e BFPUG (*Brazilian Function Point Users Group*);
 - NESMA (*Netherlands Software Metrics users Association*);
 - COSMIC-FFP (*COmmon Software Measurement International Consortium*).
 - MARK II (*UK Software Metrics Association*);

Objetivos da APF

- Medir a funcionalidade que o usuário solicita e recebe (**visão do usuário**);
- Medir o desenvolvimento e manutenção de software de forma independente da tecnologia utilizada para sua implementação;
- Conceito de USUÁRIO: É QUALQUER PESSOA OU COISA QUE INTERAJA (ENVIA OU RECEBA DADOS) COM A APLICAÇÃO OU ESPECIFIQUE SEUS REQUISITOS.

Visão do Usuário x Visão do Desenvolvedor



Visão do Usuário

- Representa uma descrição formal das necessidades de negócio do usuário em sua própria linguagem;
- Desenvolvedores traduzem a informação do usuário em linguagem de tecnologia da informação para fornecer uma solução;
- Características da descrição:
 - É uma descrição das funções do negócio;
 - É aprovada pelo usuário;
 - Pode ser usada para contar pontos de função;
 - Pode variar em sua forma física (protótipos, atas de reunião, modelos, documento de visão, casos de uso, etc.).

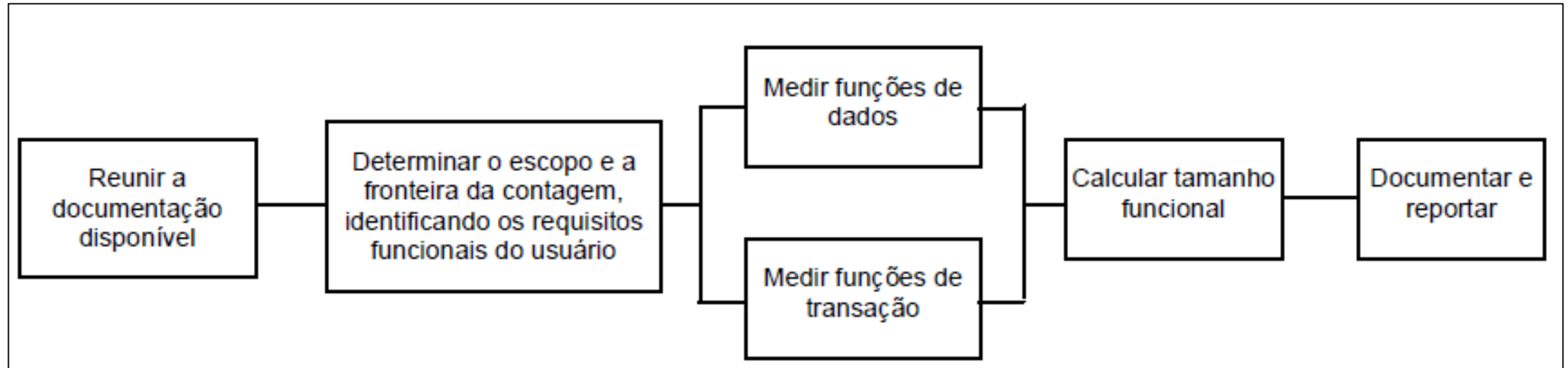
APF – Características

- A APF mede **o que** o software faz, independentemente de **como** ele foi construído;
- O processo de medição ou de contagem de pontos de função é baseado na avaliação dos **requisitos funcionais do usuário**, como descrito nos artefatos do desenvolvimento;
- O procedimento de medição está descrito pelo *IFPUG* em seu Manual de Práticas de Contagem (*Counting Practices Manual*).
- PF **não** medem diretamente esforço, produtividade ou custo, porque ela produz uma medida de tamanho funcional do software;
- Com o tamanho funcional em mãos é possível correlacioná-lo com outras variáveis, tornando possível identificar produtividade, estimar esforço e/ou custo de projetos de software.

Benefícios da APF

- Uma ferramenta para determinar o tamanho de um pacote adquirido através da contagem de todas as funções incluídas;
- Contagem das funções que especificamente correspondem aos seus requisitos (*make ou buy*);
- Análise de produtividade e qualidade;
- Gerenciamento de escopo (controlar o aumento do escopo);
- Gerenciamento dos requisitos (auxiliar na verificação da solidez e completeza dos requisitos especificados);
- Estimar custo e recursos: essa medida pode ser utilizada como entrada para diversos modelos de estimativa de esforço, prazo e custo, por exemplo COCOMO II;
- Normalização para comparação de software (*benchmarking*).

Processo de Contagem de Pontos de Função



Fonte: IFPUG [8]

Reunir a Documentação Disponível

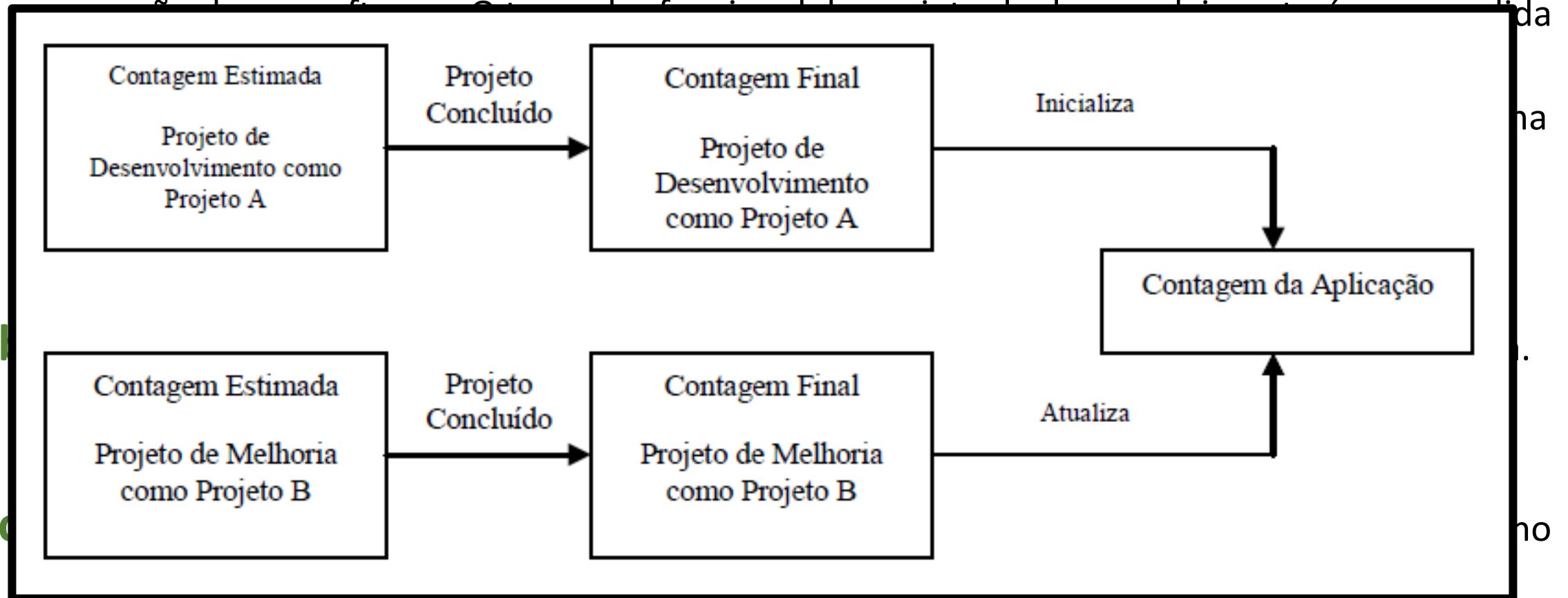
- Segundo IFPUG [8]:
 - A documentação de suporte a uma contagem de pontos de função deve descrever a funcionalidade entregue pelo software ou a funcionalidade impactada pelo projeto de software medido;
 - Deve ser obtida documentação suficiente para conduzir a contagem de pontos de função, ou acesso a especialistas no assunto capazes de fornecer informações adicionais para suprir quaisquer falhas na documentação;
 - A documentação adequada pode incluir requisitos, modelos de dados/objetos, diagramas de classes, diagramas de fluxos de dados, casos de uso, descrições de procedimentos, formatos de relatórios, manuais de usuário e outros artefatos do desenvolvimento de software.

Determinar o Escopo, Fronteira e os Requisitos Funcionais

- Para determinar o escopo e fronteira da contagem e identificar os Requisitos Funcionais do Usuário, devem ser executadas as seguintes atividades segundo [8]:
 1. Identificar o propósito da contagem, por exemplo: O propósito da contagem poderia ser a determinação do tamanho de uma release de software específica; O propósito da contagem poderia ser a determinação do tamanho de uma aplicação, como parte do esforço da organização para determinar o tamanho de seu portfólio de software;
 2. Identificar o tipo de contagem, com base no propósito, como um dos seguintes:
 - a. uma contagem de pontos de função de projeto de desenvolvimento;
 - b. uma contagem de pontos de função de aplicação;
 - c. uma contagem de pontos de função de um projeto de melhoria,
 3. Determinar o escopo da contagem, com base no propósito e tipo de contagem;
 4. Determinar a fronteira de cada aplicação contida no escopo da contagem com base na visão do usuário e não em considerações técnicas;
 5. Os requisitos do usuário podem incluir uma mistura de requisitos funcionais e não-funcionais; identificar quais requisitos são funcionais e excluir os não-funcionais.

Tipo de Contagem

- a. **Projeto de desenvolvimento**: É um projeto para desenvolver e fornecer a primeira



Fonte: IFPUG [8]

Escopo da Contagem

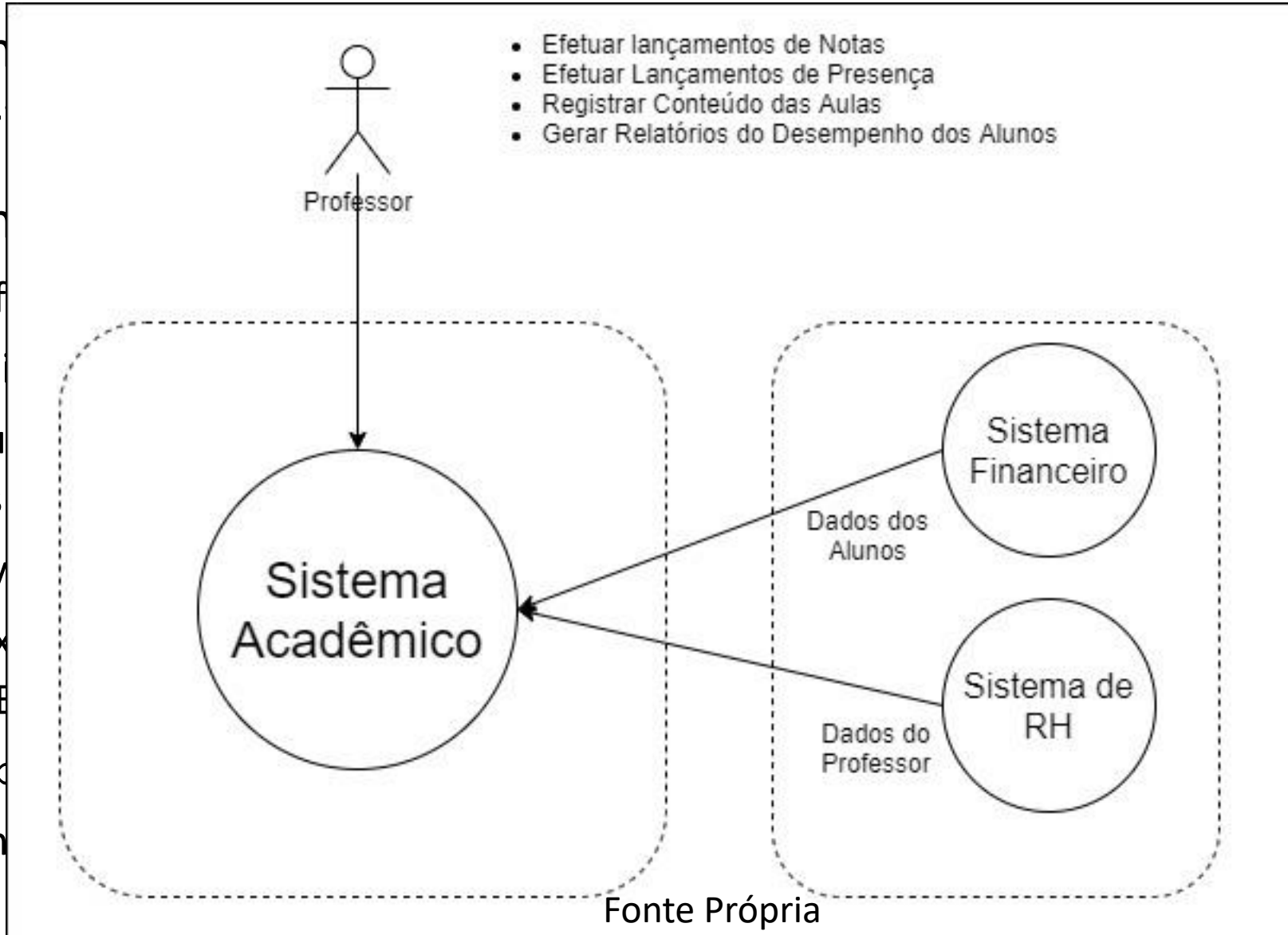
- O escopo da contagem define o conjunto de Requisitos Funcionais para ser incluído na contagem de pontos de função;
- O escopo:
 - Define o (sub)conjunto do software que está sendo medido;
 - É determinado pelo propósito para a realização da contagem de pontos de função;
 - Identifica quais funções serão incluídas na medida de tamanho funcional assim como fornecer respostas relevantes para o propósito da contagem;
 - Pode incluir mais de uma aplicação.

Fronteira ou Fronteira da Aplicação

- A fronteira do usuário

- A fronteira da aplicação

- Definição de requisitos
- Identificação de atores
- Atualização de requisitos
- Seleção de requisitos
- Envio de requisitos
- Auxílio de requisitos (AIE)
- Depuração de requisitos
- É in



todo e seus

transações (EEs,

s pela aplicação

Medir Funções de Dados

- **Arquivos Lógicos Internos**
 - Um arquivo lógico interno (ALI) é um grupo de dados ou de informações de controle logicamente relacionados, reconhecido pelo usuário, mantido dentro da fronteira da aplicação que está sendo contada. A intenção primária de um ALI é armazenar dados mantidos através de um ou mais processos elementares da aplicação que está sendo contada [8];
- **Arquivo de Interface Externa**
 - Um arquivo de interface externa (AIE) é um grupo de dados ou de informações de controle logicamente relacionados, reconhecido pelo usuário, referenciado pela aplicação que está sendo contada, porém, mantido dentro da fronteira de uma outra aplicação. A intenção primária de um AIE é armazenar dados referenciados através de um ou mais processos elementares dentro da fronteira da aplicação que está sendo contada. Isto significa que um AIE contado para uma aplicação deve ser um ALI em outra aplicação [8].

Medir Funções de Dados

- **Informações de controle**: são dados que influenciam um processo elementar. Especificam o que, quando ou como os dados serão processados;
- **Reconhecido pelo Usuário**: refere-se a requisitos definidos para processos e/ou grupos de dados que foram acordados e entendidos tanto pelo(s) usuário(s) quanto pelos desenvolvedor(es) de software;
- **Mantido**: refere-se à habilidade de incluir, modificar ou excluir dados a partir de um processo elementar;

Medir Funções de Dados

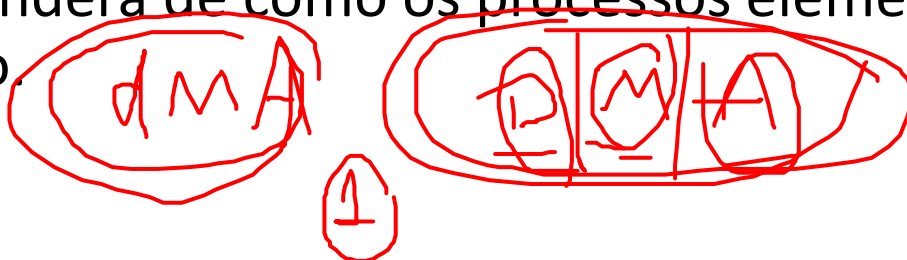
- **Processo elementar**: é a menor unidade de atividade que tem significado para o usuário. Deve-se compor e/ou decompor os Requisitos Funcionais do Usuário até a menor unidade de atividade, a qual satisfaz os itens a seguir:
 - é significativo para o usuário;
 - constitui uma transação completa;
 - é auto contida;
 - deixa o negócio da aplicação contada em um estado consistente.

Contando Funções do Tipo Dados

- O número de ALIs, AIEs, e suas respectivas **complexidades funcionais** determinam a contribuição das funções de dados para o tamanho funcional;
- Atribua a cada ALI e AIE identificado uma complexidade funcional com base na quantidade de **tipos de dados elementares** (TDs) e de **tipo de registros elementares ou lógicos** (TRLs) associados ao ALI ou AIE;
- **Observação importante**: Funções de dados são mais facilmente identificadas quando se utiliza um modelo lógico de dados; entretanto, isto não impede a medição em ambientes onde técnicas alternativas de modelagem de dados ou objetos são empregadas. A terminologia de modelagem de dados é utilizada para documentar as regras de funções de dados, porém a mesma abordagem pode ser aplicada com outras técnicas.

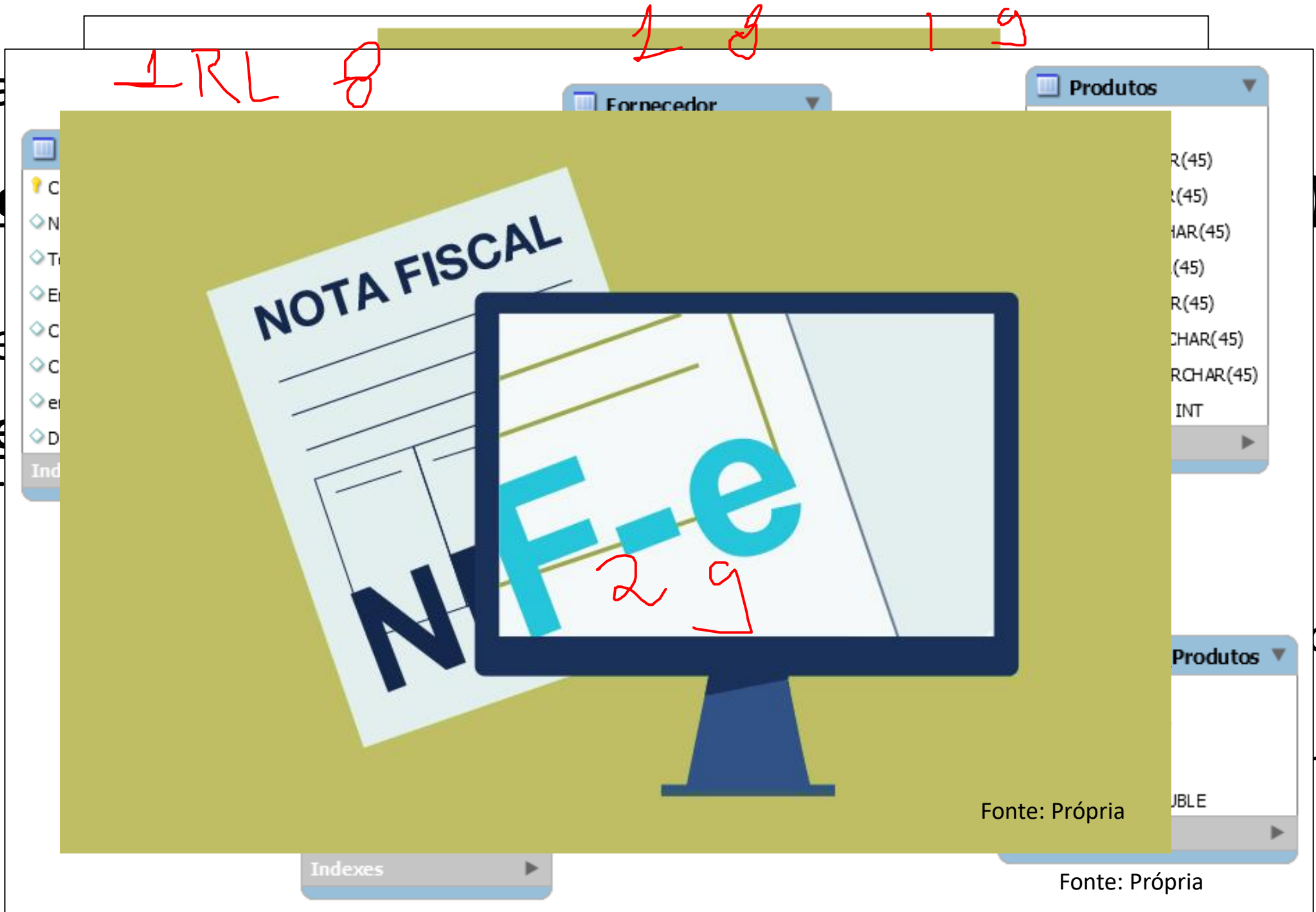
Contando Funções do Tipo Dados

- **Definição de tipos de dados elementares (TD)**: Um tipo de dado elementar é um campo único, reconhecido pelo usuário e não repetido;
- **Regras** para a contagem de Tipos de Dados Elementares (TDs) para uma função de dados:
 - Conte um TD para cada campo único, reconhecido pelo usuário e não repetido, mantido ou recuperado pela função de dados durante a execução de todos os processos elementares no escopo da contagem;
 - Conte um TD para cada parte de dado requisitada pelo usuário para estabelecer um relacionamento com outra função de dado;
 - Revisar os atributos relacionados para determinar se eles estão agrupados e contados como um simples TD ou se são contados como TD múltiplos; o agrupamento dependerá de como os processos elementares usam os atributos dentro da aplicação.



Conta

- Dé de
- Re (T



Determinando a Complexidade das Funções do Tipo Dados

- Cada Arquivo Lógico Interno e cada Arquivo de Interface Externa deve ser classificado com relação à sua complexidade funcional (baixa, média ou alta) baseado em:
- Número de Tipos de Dados Elementar (TD);
- Número de Tipos de Registros Lógico (TRL).

Tipos de Registros	Tipos de Dados			
		< 20	20 - 50	> 50
	1	Baixa	Baixa	Média
	2 - 5	Baixa	Média	Alta
	> 5	Média	Alta	Alta

Arquivo Lógico Interno

Grau de Complexidade Funcional	Pontos de Função
Baixo	7
Médio	10
Alto	15

Arquivo de Interface Externa

Grau de Complexidade Funcional	Pontos de Função
Baixo	5
Médio	7
Alto	10

Um Pequeno Exemplo

Codigo INT

Nome VARCHAR(45)

Telefone VARCHAR(45)

Endereço VARCHAR(45)

Cidade VARCHAR(45)

Celular VARCHAR(45)

email VARCHAR(45)

Data Nascimento VARCHAR(45)

Indexes

Observações VARCHAR(45)

Vendascol VARCHAR(45)

Cientes_Codigo INT

Indexes

Quantidade INT

Preço Unitário DOUBLE

Indexes

ALI

Co

Total de Pontos de Função do Tipo Dados

Cientes = 5 PF

Fornecedores = 5 PF

Produtos = 7 PF

Vendas = 7 PF

===

Total = 24 PF

AIE

1 TRL

8 TD

lexidade = Baixa

PF = 5

Medir Funções de Transações

- **Entrada Externa**

- Uma entrada externa (EE) é um processo elementar que processa dados ou informações de controle que vêm de fora da fronteira da aplicação. A intenção primária de uma EE é manter um ou mais ALIs e/ou alterar o comportamento do sistema;

- **Consulta Externa**

- Uma consulta externa (CE) é um processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação. A intenção primária de uma consulta externa é apresentar informações ao usuário através da recuperação de dados ou informações de controle. A lógica de processamento não contém fórmula matemática ou cálculo, e nem cria dados derivados. Nenhum ALI é mantido durante o processamento, nem o comportamento do sistema é alterado.

- **Saída Externa**

- Uma saída externa (SE) é um processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação e que inclui um processamento adicional ao de uma consulta externa. A intenção primária de uma SE é apresentar informações ao usuário através de lógica de processamento que não seja apenas a recuperação de dados ou informações de controle. A lógica de processamento deve conter pelo menos uma fórmula matemática ou cálculo, criar dados derivados, manter um ou mais ALIs ou alterar o comportamento do sistema.

Medir Funções de Transações

- Segundo IFPUG [8], a principal diferença entre os tipos de função de transação é sua intenção primária. A tabela abaixo mostra resumidamente as funções que podem ser executadas por cada tipo de função de transação, especificando a intenção primária de cada uma.

Função	Tipo de Função de Transação		
	EE	SE	CE
Alterar o comportamento do sistema	IP	F	N/A
Manter um ou mais ALI's	IP	F	N/A
Apresentar a informação ao usuário	F	IP	IP

- IP a intenção primária do tipo de função de transação;
- F é uma função do tipo de função de transação, mas não é a intenção primária e está presente algumas vezes;
- N/A não é permitida para o tipo de função de transação.

Medir Funções de Transação

- **Processo elementar**: é a menor unidade de atividade que tem significado para o usuário. Deve-se compor e/ou decompor os Requisitos Funcionais do Usuário até a menor unidade de atividade, a qual satisfaz os itens a seguir:
 - é significativo para o usuário;
 - constitui uma transação completa;
 - é auto contida;
 - deixa o negócio da aplicação contada em um estado consistente.

Medir Funções de Transação

- **Informações de controle**: são dados que influenciam um processo elementar. Especificam o que, quando ou como os dados serão processados;
- **Reconhecido pelo Usuário**: refere-se a requisitos definidos para processos e/ou grupos de dados que foram acordados e entendidos tanto pelo(s) usuário(s) quanto pelos desenvolvedor(es) de software;
- **Mantido**: refere-se à habilidade de incluir, modificar ou excluir dados a partir de um processo elementar;

Medir Funções de Transação

- **Significativo**: É reconhecido pelo usuário e satisfaz um Requisito Funcional do Usuário;
- **Lógica de Processamento**: Lógica de Processamento é definida como qualquer um dos requisitos especificamente solicitados pelo usuário para completar um processo elementar como validações, algoritmos ou cálculos e leitura ou manutenção de uma função de dados.

Medir Funções de Transação

Formas de lógica de processamento:	Tipo da Função de Transação		
	EE	SE	CE
1. Validações são efetuadas	p	p	p
2. Cálculos matemáticos são efetuados	p	d*	n
3. Valores equivalentes são convertidos	p	p	p
4. Dados são filtrados e selecionados por critérios específicos para comparar vários grupos de dados	p	p	p
5. Condições são analisadas para determinar quais se aplicam	p	p	p
6. Pelo menos um ALI é atualizado	d*	d*	n
7. Pelo menos um ALI ou AIE é referenciado	p	p	d
8. Dados ou informações de controle são recuperados	p	p	d
9. Dados derivados são criados	p	d*	n
10. O comportamento do sistema é alterado	d*	d*	n
11. Preparar e apresentar informações para fora da fronteira	p	d	d
12. Dados ou informações de controle entrando pela fronteira da aplicação são aceitos	d	p	p
13. Os dados são reclassificados ou reorganizados	p	p	p

- **d** - o tipo de função deve executar esta forma de lógica de processamento;
- **d*** - o tipo de função deve executar pelo menos uma destas formas de lógica de processamento;
- **p** - o tipo de função pode executar esta forma de lógica de processamento, mas a mesma não é obrigatória;
- **n** - o tipo de função não pode executar esta forma de lógica de processamento.

Contando Funções do Tipo Transações

- O número de **EEs**, **SEs** e **CEs** e suas **complexidades funcionais** determinam a contribuição das funções de transação para o tamanho funcional;
- Atribua uma complexidade funcional a cada **EE**, **SE** e **CE** identificada, com base no **número de tipos de arquivos referenciados** (TARs) e **número de tipos de dados elementares** (TDs).

Contando Funções do Tipo Transações

- **Definição de Tipo de Arquivo Referenciado (TAR)**: Um tipo de arquivo referenciado é uma função de dados lida e/ou mantida pela função de transação. Um tipo de arquivo referenciado inclui:
 - Um arquivo lógico interno lido ou mantido por uma função de transação;
 - Um arquivo de interface externa lido por uma função de transação;
- **Regra** para a contagem de Tipos de Arquivo Referenciado para uma função de transação:
 - Um TAR deve ser contado para cada função de dados acessada (lida e/ou escrita).

Contando Funções do Tipo Transações

- **Definição de Tipo de Dado Elementar (TD)**: Um tipo de dado elementar é um campo único, reconhecido pelo usuário e não repetido;
- **Regras** para a contagem de Tipos de Dados Elementares como uma função de transação:
 - Conte um TD para cada campo único reconhecido pelo usuário, atributo não repetido que cruza (entra e/ou sai) a fronteira durante o processamento da função de transação;
 - Conte apenas um TD por função de transação para a habilidade de enviar uma mensagem de resposta de aplicação mesmo que sejam mensagens múltiplas;

Contando Funções do Tipo Transações

- Conte apenas um TD por função de transação para a habilidade de iniciar uma ação ou ações mesmo que haja múltiplos meios para realizá-la;
- **Não conte os itens a seguir como TDs:**
 - Literais como títulos de relatório, tela ou identificador do painel, títulos de coluna e títulos de atributos ou selos gerados automaticamente pelo sistema como atributos de data e hora;
 - Variável de paginação, número de páginas e informação de posicionamento;
 - Ajudas de navegação como a habilidade de navegar com uma lista utilizando “anterior”, “próximo”, “primeiro”, “último” e seus gráficos equivalentes;
 - Atributos gerados dentro da fronteira por uma função de transação e armazenado em um ALI sem sair da fronteira;
 - Atributos obtidos ou referenciados de um ALI ou AIE para a participação em processamento sem sair da fronteira.

Determinando a Complexidade das Funções do Tipo Transações

- Cada EE, SE e CE deve ser classificada com relação à sua complexidade funcional (baixa, média ou alta) baseado em:
 - Número de Tipos de Dados Elementar (TD);
 - Número de Tipos de Arquivos Referenciados (TAR).

EE

Arquivos Referenciados	Tipos de Dados			
		< 5	5 - 15	> 15
	< 2	Baixa	Baixa	Média
	2	Baixa	Média	Alta
	> 2	Média	Alta	Alta

SE e CE

Arquivos Referenciados	Tipos de Dados			
		< 6	6 - 19	> 19
	< 2	Baixa	Baixa	Média
	2 - 3	Baixa	Média	Alta
	> 3	Média	Alta	Alta

Determinando a Complexidade das Funções do Tipo Transações

Entrada Externa e Consulta Externa

Classificação da Complexidade Funcional	Pontos de Função
Baixa	3
Média	4
Alta	6

Saída Externa

Classificação da Complexidade Funcional	Pontos de Função
Baixa	4
Média	5
Alta	7

Estimando em PF – Exercício: “Hotel 10 Estrelas”

Exercício: “Hotel 10 Estrelas”

1. Diagrama de Contexto e Diagrama Entidade Relacionamento:

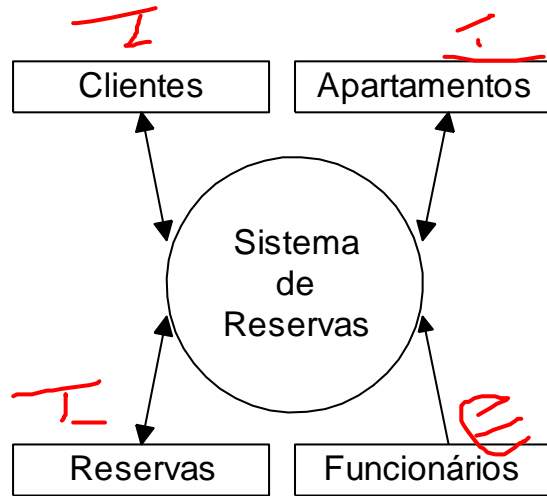
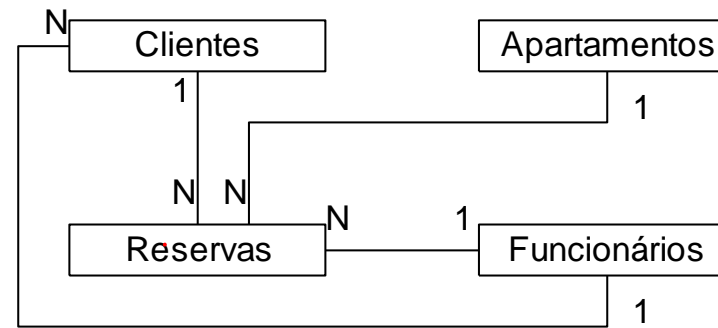


Diagrama de Contexto

Fonte: Própria



DER

Fonte: Própria

Estimando em PF – Exercício: “Hotel 10 Estrelas”

Exercício: “Hotel 10 Estrelas”

2. Tabelas e respectivos Campos:

- **Arquivo de reservas:**

- número_da_reserva (PK)
- matrícula_funcionário (FK)
- código_do_cliente (FK)
- data_da_reserva
- período_inicial
- período_final
- número_apartamento (FK)
- nome_da_pessoa

- **Arquivo de apartamentos:**

- número_do_apartamento (PK)
- tipo_do_apartamento

- **Arquivo de clientes:**

- código_do_cliente (PK)
- tipo_classe
- nome_cliente
- tipo_pessoa
- CGC/CPF
- Endereço
- pessoa_contato
- telefone
- matrícula_funcionário (FK)

- **Arquivo de funcionários:**

- matrícula_funcionário (PK)
- nome_funcionário
- função
- salário
- cpf
- endereço
- telefone

Estimando em PF – Exercício: “Hotel 10 Estrelas”

Exercício: “Hotel 10 Estrelas”

3. Calculando a Complexidade das Funções do Tipo Dados:

#	Processo Elementar ou Grupo de Dados	Tipo	Cálculo do Número de PF			
			TD	AR/TR	Complex.	PF
1	Arquivo de Reservas	ALI	8	1	Baixa	7
2	Arquivo de Apartamentos	ALI	2	1	Baixa	7
3	Arquivo de Clientes	ALI	9	1	Baixa	7
4	Arquivo de Funcionários	AIE	7	1	Baixa	5

Estimando em PF – Exercício: “Hotel 10 Estrelas”

Exercício: “Hotel 10 Estrelas”

4. Interfaces, Observações e Regras de Negócio:

- Registrar Reservas (Incluir Reservas);
- Excluir Reservas;
- Inserir Apartamentos;
- Alterar Apartamentos;
- Excluir Apartamentos;
- Consultar Apartamentos;
- Inserir Clientes;
- Alterar Clientes;
- Excluir Clientes;
- Consultar Clientes;
- Relatório das Reservas.

Estimando em PF – Exercício: “Hotel 100”

Entrada Externa
TD: 3
AR: 1

Entrada Externa
TD: 10
AR: 4

ações possuem comando e mensagem de confirmação.

Registro de Reservas	
Número da Reserva: []	Matrícula do Funcionário: []
Código do Cliente: []	Data de [/ /]
Período: de [/ /] a [/ /]	Motivo: []

Excluir Reserva

Número da Reserva: []

Consulta de Apartamentos

Informe Operação: [] I - Incluir, A - Alterar, E - Excluir, C - Consultar

Número: []

Tipo: [] S - Simples, D - Duplo, T - Triplo

Consulta Externa
TD: 5
AR: 1

- **RN1:** Para excluir Apartamentos fornecer operação e número do apartamento;
- **RN2:** Se o apartamento tiver uma reserva ele não deverá ser excluído;
- **RN3:** Para consultar Apartamentos informar operação e número do apartamento. O sistema deverá informar o tipo do apartamento.

Estimando em PF – Exercício

Entrada

Entrada

Entrada

Consulta Externa

TD: 12
AR: 1

Informe Operação: [] I - Incluir, A - Alterar, E - Excluir, C - Consultar
Código do Cliente: [] Tipo (Vip/Std): [] Nome: []
Tipo (Fis/Jur): [] CGC/ CPF: []
Endereço: [] Pessoa Contato: []
Telefone: [] Matrícula Funcionário: []

- **RN4:** Para excluir Clientes fornecer operação e nome do cliente;
- **RN5:** Se o cliente tiver uma reserva, e o sistema estiver com o cliente em reserva, o sistema deverá informar as demais informações.
- **RN6:** Para consultar Clientes informar o código do cliente. O sistema deverá informar as demais informações.

Saída Externa

TD: 5
AR: 2

Total de Reservas por Cliente		
Código do Cliente	Nome do Cliente	Total de Reservas
999999	XXXXXXXXXXXXXXXXXXXXXXX	999999999

Estimando em PF – Exercício: “Hotel 10 Estrelas”

Exercício: “Hotel 10 Estrelas”

5. Calculando a Complexidade das Funções do Tipo Transações:

#	Processo Elementar ou Grupo de Dados	Tipo	Cálculo do Número de PF			
			TD	AR/TR	Complex.	PF
5	Registrar Reservas	EE	10	4	Alta	6
6	Excluir Reservas	EE	3	1	Baixa	3
7	Incluir Apartamentos	EE	5	1	Baixa	3
8	Alterar Apartamentos	EE	5	1	Baixa	3
9	Excluir Apartamentos	EE	4	2	Baixa	3
10	Consultar Apartamentos	CE	5	1	Baixa	3
11	Incluir Clientes	EE	12	2	Média	4
12	Alterar Clientes	EE	12	2	Média	4
13	Excluir Clientes	EE	4	2	Baixa	3
14	Consultar Clientes	CE	12	1	Baixa	3
15	Total de Reservas por Cliente	SE	5	2	Baixa	4

Estimando em PF – Exercício: “Hotel 10 Estrelas”

Exercício: “Hotel 10 Estrelas”

6. Interpretando o número total de PF:

- **TOTAL** = 65 PF
- **Suposição**: Inferindo no Banco de Dados Histórico da Organização, verificou-se:
 - Projeto Java: 1 PF = 3,5 horas;
 - Projeto C++: 1 PF = 7,2 horas;

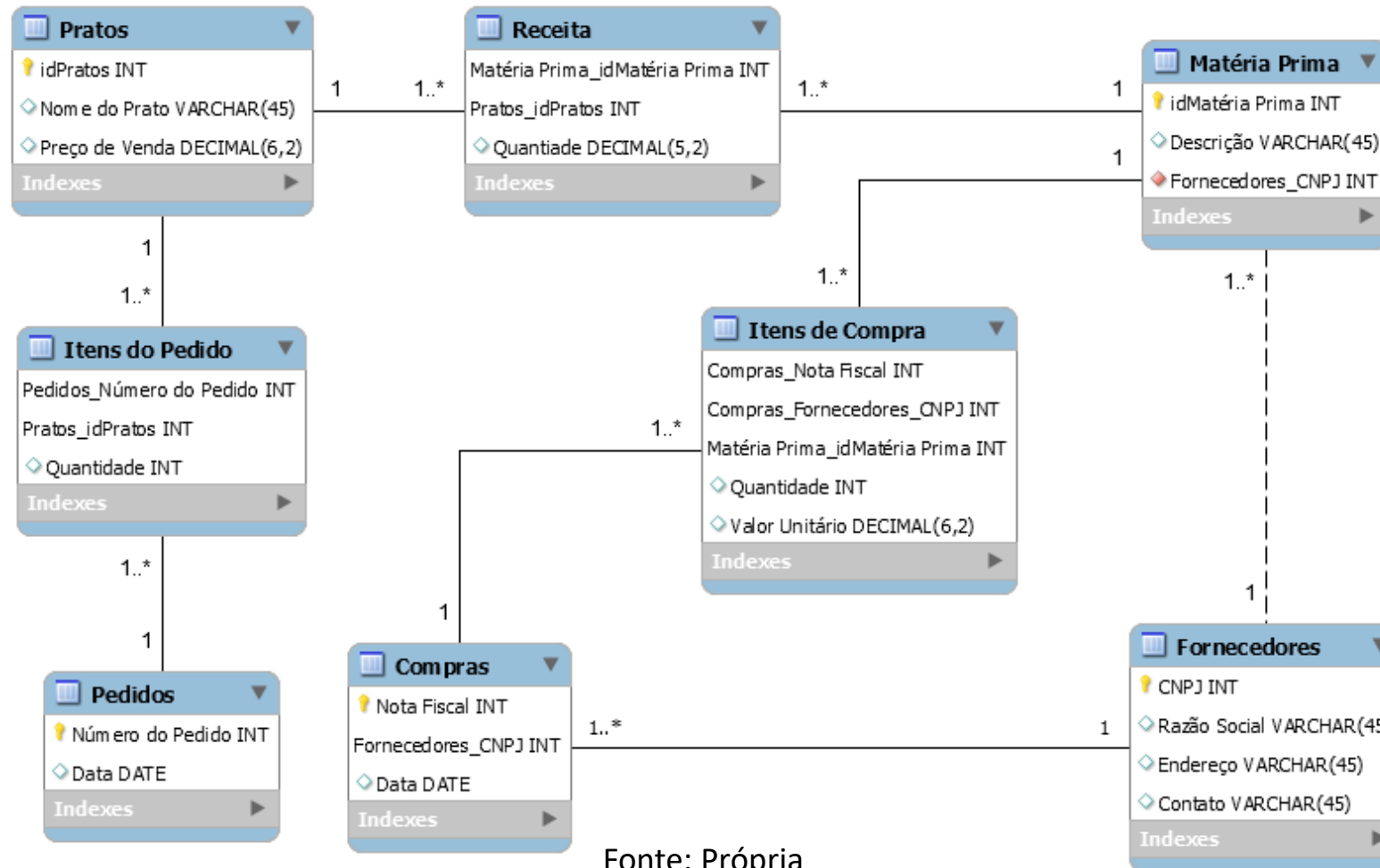
Tabela da quantidade média de linhas de código-fonte (SLOC) por Ponto de Função

Link: <http://www.qsm.com/FPGearing.html>

Estimando em PF – Exercício: “Restaurante 10 Estrelas”

Exercício: “Restaurante 10 Estrelas”

1. DER



Fonte: Própria

Estimando em PF – Exercício: “Restaurante 10 Estrelas”

Exercício: “Restaurante 10 Estrelas”

2. Interfaces

MENU MENU

Incluir Pedido

Número do Pedido:

Data:

Código do Prato	Descrição	Quantidade	Valor Total
<div>Mensagem de Erro Emitida: Número do Pedido não existe</div>			

Confirmar

Cancelar

Fonte: Própria

Estimando em PF – Exercício: “Restaurante 10 Estrelas”

Exercício: “Restaurante 10 Estrelas”

MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU

Consultar Último Preço de Custo da Matéria-Prima

Código da Matéria Prima:

Consultar

Nome da Matéria Prima:

Nome do Fornecedor:

Data da Última Compra:

Preço de Custo:

Mensagem de Erro Emitida: Código da Matéria Prima não Existe

Estimando em PF – Exercício: “Restaurante 10 Estrelas”

Exercício: “Restaurante 10 Estrelas”

Matéria-Prima

Operação: [] (alterar, incluir, excluir)

Código: [] Descrição: []

CNPJ Fornecedor: []

[CONFIRMAR]

[CANCELAR]

RN1: Uma matéria-prima somente poderá ser excluída se ela não estiver relacionada com nenhum prato e com nenhuma Compra.

Incluir Prato

Código: []

Nome: []

Preço de Venda: []

Código_matéria_prima	Nome_matéria_prima	Quantidade
[]	[]	[]
[]	[]	[]
[]	[]	[]

[CONFIRMAR]

[CANCELAR]

Exercício: “Restaurante 10 Estrelas”

Excluir Matéria-Prima do Prato

Código: []
Nome: []

Código_matéria_prima	Nome_matéria_prima	Quantidade	Excluir
[]	[]	[]	[]
[]	[]	[]	[]
[]	[]	[]	[]

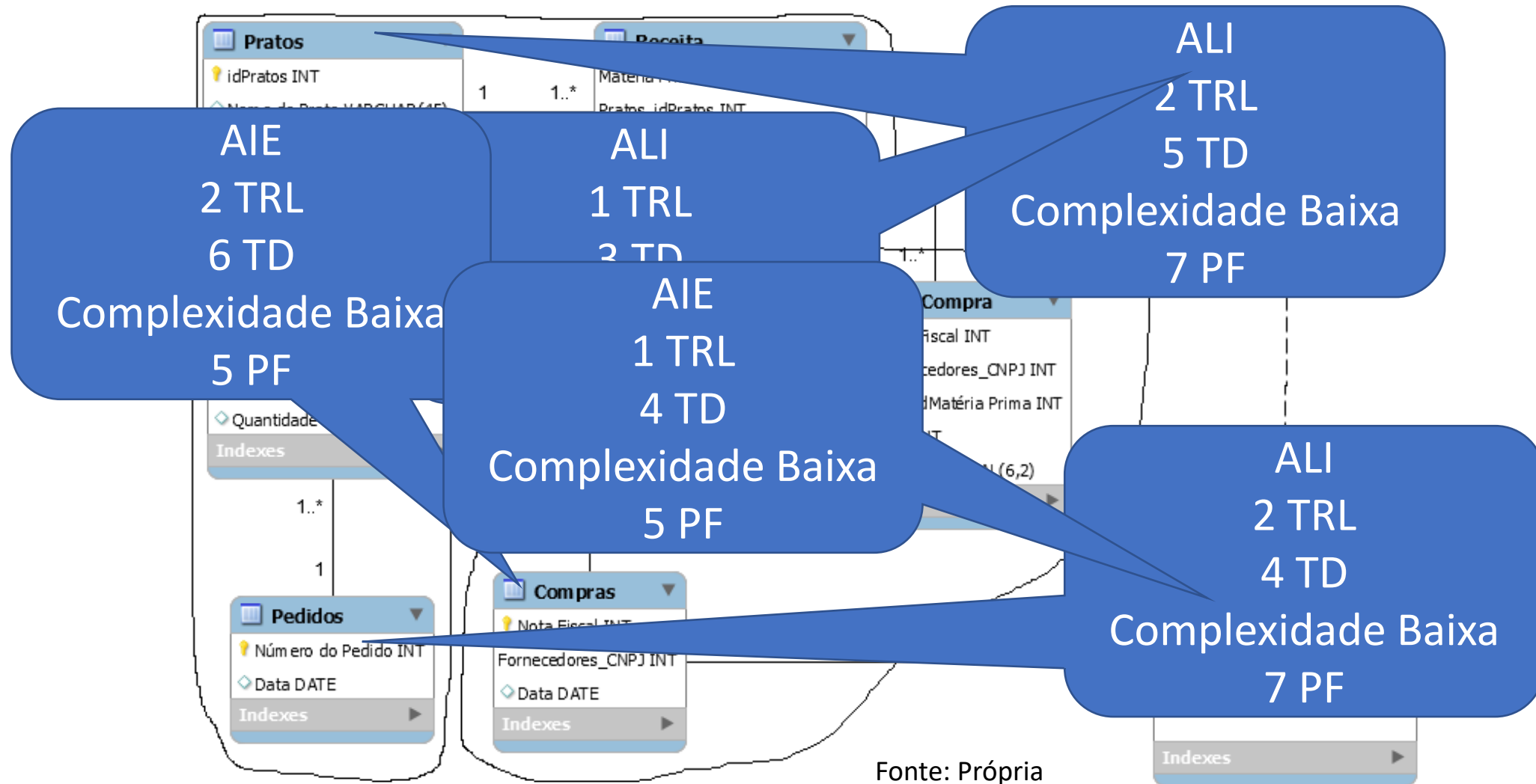
[CONFIRMAR] [CANCELAR]

Exercício: “Restaurante 10 Estrelas”

Fonte: Própria

Estimando em PF – Exercício: “Restaurante 10 Estrelas”

3. Calculando a Complexidade das Funções do Tipo Dados



Estimando em PF – Exercício: “Restaurante 10 Estrelas”

Exercício: “Restaurante 10 Estrelas”

3. Calculando a Complexidade das Funções do Tipo Dados

#	Processo Elementar ou Grupo de Dados	Tipo	Cálculo do Número de PF			
			TD	AR/TR	Complex.	PF
1	arquivo materia prima	ALI	3	1	Baixa	7
2	arquivo pratos	ALI	5	2	Baixa	7
3	arquivo pedido	ALI	4	2	Baixa	7
4	arquivo fornecedores	AIE	4	1	Baixa	5
5	arquivo compras	AIE	6	2	Baixa	5

Estimando em PF – Exercício: “Restaurante 10 Estrelas”

4. Calculando a Complexidade das Funções do Tipo Transações

MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU

Incluir Pedido

Número do Pedido:

Data:

Código do Prato	Descrição	Quantidade	Valor Total
Mensagem de Erro Emitida: Número do Pedido não existe			

Confirmar

Cancelar

Entrada Externa
TD: 8
AR: 2

Fonte: Própria

Estimando em PF – Exercício: “Restaurante 10 Estrelas”

4. Calculando a Complexidade das Funções do Tipo Transações

MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU MENU

Consultar Último Preço de Custo da Matéria-Prima

Código da Matéria Prima:

Nome da Matéria Prima:

Nome do Fornecedor:

Data da Última Compra:

Preço de Custo:

Mensagem de Erro Emitida: Código da Matéria Prima não Existe

Consulta Externa
TD: 7
AR: 3

Estimando em PF – Exercício: “Restaurante 10 Estrelas”

4. Calculando a Complexidade das Funções do Tipo Transações

Matéria-Prima

Operação: [] (alterar, incluir, excluir)

Código: [] Descrição: []

CNPJ Fornecedor: []

[CONFIRMAR] [CANCELAR]

RN1: Uma matéria-prima somente poderá ser excluída se ela não estiver relacionada com nenhum prato e com nenhuma Compra.

Entrada Externa
TD: 4
AR: 3

Incluir Prato

Código: []

Nome: []

Preço de Venda: []

Código_matéria_prima	Nome_matéria_prima	Quantidade
[]	[]	[]
[]	[]	[]
[]	[]	[]

[CONFIRMAR] [CANCELAR]

Entrada Externa
TD: 8
AR: 2

Estimando em PF – Exercício: “Restaurante 10 Estrelas”

4. Calculando a Complexidade das Funções do Tipo Transações

Código: []

Nome: []

Preço de Venda: []

[CONFIRMAR] [CANCELAR]

Entrada Externa

TD: 5

AR: 1

Alterar Itens do Pedido

Número do Pedido: []

Código do Prato

Nome do Prato

Quantidade

[]

[]

[]

[CONFIRMAR]

[CANCELAR]

RN2: Na alteração, somente a quantidade pode ser alterada.

Entrada Externa

TD: 6

AR: 2

Estimando em PF – Exercício: “Restaurante 10 Estrelas”

Exercício: “Restaurante 10 Estrelas”

4. Calculando a Complexidade das Funções do Tipo Transações

#	Processo Elementar ou Grupo de Dados	Tipo	Cálculo do Número de PF			
			TD	AR/TR	Complex.	PF
6	incluir pedido	EE	8	2	Média	4
7	consultar último preço de custo de MP	CE	7	3	Média	4
8	incluir mp	EE	6	2	Média	4
9	alterar mp	EE	6	2	Média	4
10	excluir mp	EE	4	3	Média	4
11	incluir prato	EE	8	2	Média	4
12	alterar preço de venda de prato	EE	5	1	Baixa	3
13	alterar itens do pedido	EE	6	2	Média	4
14	excluir itens do pedido	EE	7	2	Média	4
15	excluir mp do prato	EE	8	2	Média	4
16	alterar mp do prato (somente qtd)	EE	7	2	Média	4

Estimando em PF – Exercício: “Restaurante 10 Estrelas”

Exercício: “Restaurante 10 Estrelas”

5. Interpretando o número total de PF:

- **TOTAL** = 74 PF
- **Suposição**: Inferindo no Banco de Dados Histórico da Organização, verificou-se:
 - Projeto Java: 1 PF = 3,5 horas;
 - Projeto C++: 1 PF = 7,2 horas;

Pontos por Caso de Uso

- Em 1993 Gustav Karner [9], propôs a metodologia dos Pontos por Caso de Uso (baseado na Análise de Pontos de Função);
- O seu objetivo foi estimar recursos para projetos de software orientados a objeto desenvolvidos utilizando o processo *Objectory*, um dos embriões do RUP;
- O método trata de estimar o tamanho de um sistema de acordo com o modo como os usuários o utilizarão, a complexidade de ações requerida por cada tipo de usuário e uma análise em alto nível dos passos necessários para a realização de cada tarefa, em um nível muito mais abstrato que a técnica de Pontos de Função;
- Uma vez que os casos de uso principais do sistema sejam levantados, é possível estimar o tamanho do software como um todo baseando-se em um conjunto simples de métricas e modificadores, similar à técnica de Pontos de Função.

Pontos por Caso de Uso

- O processo de medição do Ponto de Caso de Uso consiste resumidamente em:
 1. Contar os atores e identificar sua complexidade;
 2. Contar os casos de uso e identificar sua complexidade;
 3. Calcular os Pontos por Caso de Uso não ajustados;
 4. Determinar o fator de complexidade técnica;
 5. Determinar o fator de complexidade ambiental;
 6. Calcular os Pontos por Caso de Uso ajustados.

Pontos por Caso de Uso - Passo 1: Contar os atores e identificar sua complexidade

- Classificar os atores envolvidos em cada caso de uso, de forma a obter um somatório de pontos não-ajustado;
- O peso total dos atores do sistema é calculado pela soma dos produtos do número de atores de cada tipo pelo respectivo peso;
- Por exemplo, um sistema projetado para dois tipos de usuários e que fosse acessado por um outro sistema utilizando-se de um protocolo de comunicação, teria um valor de 8 (veja tabela abaixo);

Tipo de Ator	Peso	Descrição
Ator Simples	1	Outro sistema acessado através de uma API de programação
Ator Médio	2	Outro sistema interagindo através de um protocolo de comunicação, como TCP/IP ou FTP
Ator Complexo	3	Um usuário interagindo através de uma interface gráfica (stand-alone ou Web)

Pontos por Caso de Uso - Passo 2: Contar os casos de uso e identificar sua complexidade

- Uma vez calculado o peso dos atores do sistema, partimos para o cálculo inicial do peso bruto dos casos de uso;
- Os casos de uso são divididos em três níveis de complexidade, de acordo com o número de transações envolvidas em seu processamento;
- Por transação, entende-se como uma série de processos que devem, garantidamente, ser realizados em conjunto - ou cancelados em sua totalidade, caso não seja possível concluir o processamento;
- Segundo Karner [9], uma transação é um evento que ocorre entre a requisição do usuário e a resposta do sistema.

Tipo de Caso de Uso	Número de Transações	Peso
Simple	Até 3	1
Médio	4 a 7	2
Complexo	7 ou mais	3

Pontos por Caso de Uso - Passo 2: Contar os casos de uso e identificar sua complexidade

- Uma outra maneira de se calcular o peso dos casos de uso do sistema é levar em consideração o número de classes envolvidas no processo;
- O cálculo, neste caso, é realizado da mesma forma que na abordagem anterior, e pode ser aplicado quando já for possível antever as entidades envolvidas em um dado processo.

Tipo de Caso de Uso	Número de Entidades	Peso
Simples	5 ou menos	1
Médio	5 a 10	2
Complexo	Mais de 10	3

Pontos por Caso de Uso - Passo 3: Calcular os Pontos por Caso de Uso não ajustados

- Depois de Calcular a Complexidade dos Atores e Calcular a Complexidade dos Use Cases, deve-se calcular os Pontos de Casos de Uso não Ajustados (PCUNA);
- Para tanto, basta somar as complexidades calculadas nos dois passos anteriores:

$$\text{PCUNA} = \text{Peso dos Atores} + \text{Peso dos Casos de Uso}$$

Pontos por Caso de Uso - Passo 4: Determinar o fator de complexidade técnica

Fator	Requisito	Peso	Influência	TFactor
T1	Sistema distribuído	2		
T2	Tempo de Resposta	2		
T3	Eficiência	1		
T4	Processamento complexo	1		
T5	Código reusável	1		
T6	Facilidade de instalação	0.5		
T7	Facilidade de uso	0.5		
T8	Portabilidade	2		
T9	Facilidade de mudança	1		
T10	Concorrência	1		
T11	Recursos de segurança	1		
T12	Acessível por terceiros	1		
T13	Requer treinamento especial	1		
Σ TFactor				

Influência: entre 0 e 5 - sendo que o valor 0 indica nenhuma influência, 3 indica influência moderada e 5 indica forte influência.

Complexidade Técnica (CT)

=

$0,6 + (0,01 * \text{TFactor})$

Pontos por Caso de Uso - Passo 5: Determinar o fator de complexidade ambiental

O nível de influência indica o nível de disponibilidade de cada recurso no decorrer do projeto, por exemplo, um grau de influência mínimo (0) atribuído ao fator E3 indica uma equipe com total desconhecimento de Orientação a Objetos, enquanto que o grau máximo (5) indica a disponibilidade de uma equipe experiente neste paradigma de desenvolvimento.

Fator	Requisito	Peso	Influência	EFactor
E1	Familiaridade com RUP ou outro processo formal	1.5		
E2	Experiência com a Aplicação em desenvolvimento	0.5		
E3	Experiência em Orientação a Objetos	1		
E4	Presença de analista experiente	0.5		
E5	Motivação	1		
E6	Requisitos estáveis	2		
E7	Desenvolvedores em meio-expediente	-1		
E8	Linguagem de programação difícil	-1		
Σ EFactor				

Complexidade Ambiental (CA)

=

$1,4 + (-0,03 * TFACTOR)$

Influência: entre 0 e 5 - sendo que o valor 0 indica nenhuma influência, 3 indica influência moderada e 5 indica forte influência.

Pontos por Caso de Uso - Passo 5: Calcular os Pontos por Caso de Uso ajustados

$$PCU = PCUNA * CT * CA$$

Segundo Karner [9], podemos estimar o tempo necessário para o desenvolvimento do projeto calculando-se uma média de 20 horas de trabalho por Ponto de Caso de Uso (PCU), sendo que experiências demonstram uma variação entre 15 e 30 horas por ponto.

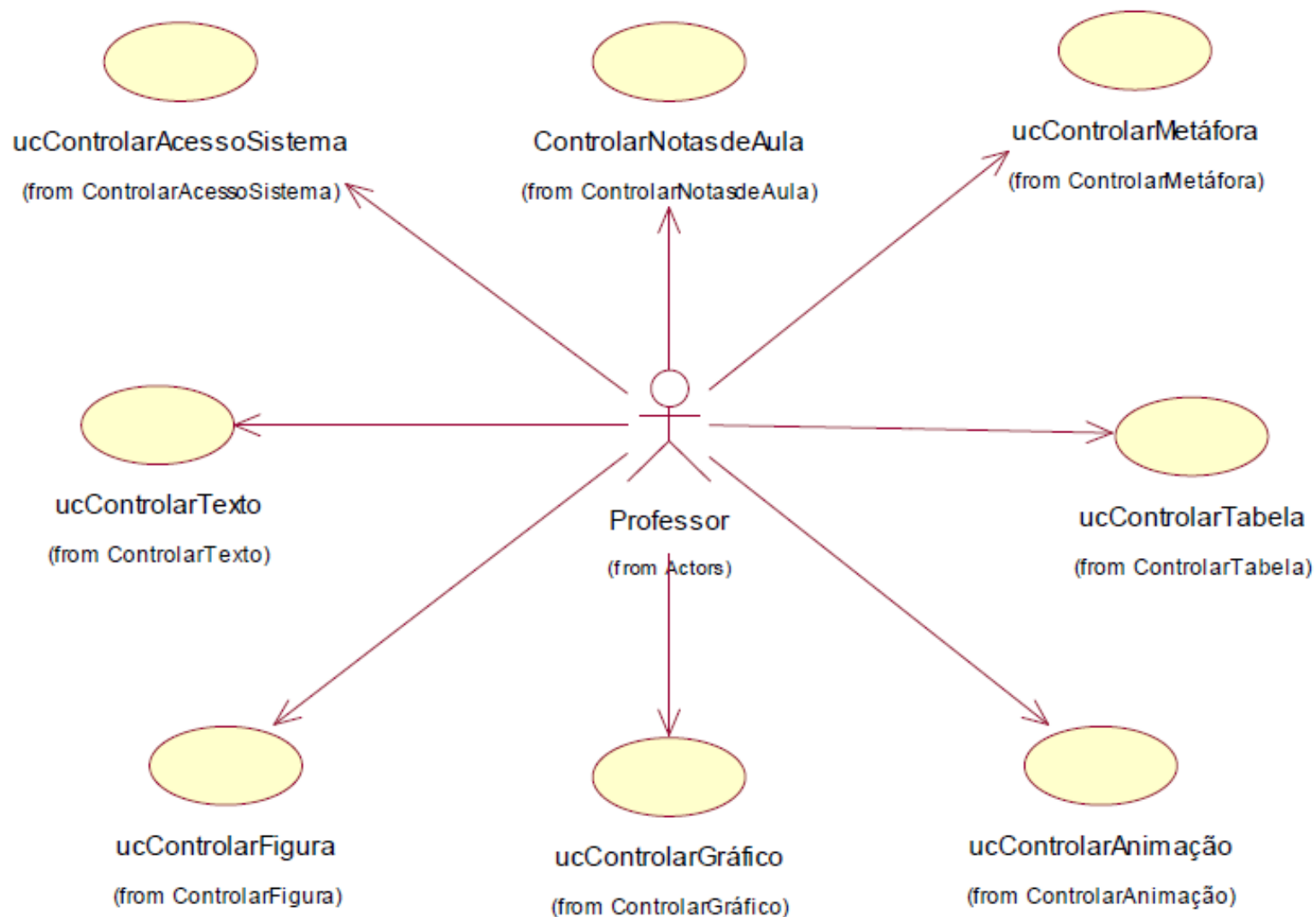
$$\text{Estimativa em Pessoa-Hora} = PCU * 20 \text{ horas}$$

$$\text{Tamanho da Equipe} = 4 \text{ pessoas, por exemplo}$$

$$\text{Estimativa em Horas} = (PCU * 20) / \text{Tamanho da Equipe}$$

$$\text{Estimativa em Meses} = \text{Estimativa em Horas} / 160$$

Pontos por Caso de Uso - Exercício



Fonte: Própria

Pontos por Caso de Uso - Exercício

Especificação do Caso de Uso Controlar Figura

Este caso de uso permitirá ao professor cadastrar, consultar, excluir e alterar Figuras.

Fluxo de Eventos

Este caso de uso inicia-se depois que o usuário acessar o site do Repositório de Metáforas e selecionar o menu Figura.

Fluxo Básico

- 1) O sistema o direciona para uma interface gráfica referente às Figuras.
- 2) O sistema exibirá uma interface listando todas as Figuras já registradas permitindo consultá-las, alterá-las ou excluí-las.
- 3) O usuário clica no link Cadastrar nova Figura.
- 4) O sistema gera automaticamente o código da Figura.
- 5) O usuário preenche os campos e clica no botão Cadastrar.

Fluxos Alternativos

Consultar Figura

- 1) Após o ponto 2 do Fluxo Básico, o usuário clica no link editar relacionado à Figura desejada.
- 2) O sistema conduz o usuário para uma interface, na qual todos os dados referentes à Figura poderão ser consultados.

Alterar Figura

- 1) Após o ponto 1 do Fluxo Alternativo Consultar Figura, o usuário clica no link Editar relacionado à Figura desejada.
- 2) O sistema conduz o usuário para uma interface de alteração, na qual todos os dados referentes à Figura estarão preenchidos.
- 3) O Usuário modifica os dados que deseja e clica no botão Editar.

Excluir Figura

- 1) Após o ponto 2 do Fluxo Básico, o usuário clica no link remover relacionado à Figura desejada.
- 2) Caso o sistema verifique que é possível a exclusão, a Figura é excluída e a lista é atualizada. Do contrário, uma mensagem é enviada para o usuário informando que a exclusão não foi possível.

Requisitos Especiais

Não existem.

Pré-condições

O usuário deve estar logado no sistema.

Pós-condições

Não existem.

Diagrama de Caso de Uso



Fonte: Própria

Pontos por Caso de Uso - Exercício

1.0. Mensurando Complexidade dos Atores				
Ator	Interface	Peso		
Simple	Interface de programa (API)	1		
Médio	Protocolo (Ex.:TCP/IP) ou interface em modo texto	2		
Complexo	Interface gráfica	3		
			Total	
2.0. Mensurando Complexidade dos Use Cases				
Caso de U	Descrição	Peso		
Simple	< 3 transações ou < 5 classes de análise	5		
Médio	4-7 transações ou 5 a 10 classes de análise	10		
Complexo	> 7 transações ou > 10 classes de análise	15		
			Total	
			PCUNA	
			PCUNA = Pontos de Casos de Uso Não Ajustados	

Fonte: Própria

Pontos por Caso de Uso - Exercício

3.0. Considerando Fatores Técnicos do Projeto					
Fator	Descrição	Peso		Atribuído	Valor
T1	Sistema distribuído	2		3	6
T2	Objetivos de performance	1		3	3
T3	Eficiência on-line	1 *		3	3
T4	Complexidade de processamento	1		3	3
T5	Código reusável em outras aplicações	1		3	3
T6	Facilidade de instalação	0,5		3	1,5
T7	Facilidade de uso	0,5		3	1,5
T8	Portabilidade	2		3	6
T9	Facilidade de alterações (<i>changeability</i>)	1		3	3
T10	Concorrência	1		3	3
T11	Segurança	1		3	3
T12	Acesso direto a terceiros	1		3	3
T13	Necessidade de facilidades especiais de treinamento para usuários	1		3	3
				FatorT	42
				FCT	1,02
				FCT = Fator de Complexidade Técnica	

Fonte: Própria

Pontos por Caso de Uso - Exercício

4.0. Considerando Fatores Ambientais				
Fator	Descrição	Peso	Atribuído	Valor
F1	Familiaridade da equipe com RUP	1,5	3	4,5
F2	Experiência da equipe	0,5	3	1,5
F3	Experiência da equipe em OO	1	3	3
F4	Capacidade dos analistas da equipe	0,5	3	1,5
F5	Motivação	1	3	3
F6	Estabilidade dos requisitos	2	3	6
F7	Estagiários ou funcionários em tempo parcial	-1	3	-3
F8	Domínio da tecnologia e configuração do ambiente	-1,5	3	-4,5
			FatorA	12
			FA	1,04
			FA = Fator Ambiental	

Fonte: Própria

Pontos por Caso de Uso - Exercício

5.0. Pontos de Caso de Uso			
PCU	$PCUNA * FCT * FA$	88,05	
Pessoa-hora por unidade de PCU		20	pessoa-hora/PCU
Estimativa em pessoa-hora		1760,93	
Tamanho da equipe		4	pessoas
Estimativa em horas		440,23	horas
Estimativa em meses		2,75	meses

Fonte: Própria

Métricas de Software

Medir, seja a performance ou uma estimativa, é uma atividade importante para o desenvolvimento de um software e a busca constante da melhoria contínua do processo de desenvolvimento. A importância ainda é maior pelo fato do software fazer parte do nosso dia a dia, nas nossas relações sociais, no nosso trabalho, no nosso lazer. Neste sentido, a qualidade do software e o seu processo de desenvolvimento devem ser medidas para alcançarmos padrões altos de qualidade, garantindo que o produto software seja entregue de maneira a garantir o que realmente foi projetado para fazer. Estimar e medir este processo, bem como o produto, e as pessoas envolvidas no desenvolvimento foi tratado nesta disciplina. As técnicas apresentadas, APF e UCP, certamente irão ajudá-los a estimar e medir. Vamos lá!!

Referências

- [1] VASQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira; ALBERT, Renato Machado; Análise de ponto de função: Medição, Estimativas e Gerenciamento de Projetos de Software. 13. ed. SP: Érica-Saraiva, 2016.
- [2] MCCONNEL, Steve. Software Project Survival Guide: How to Be Sure Your First Important Project Isn't Your Last. Redmond, Wa.: Microsoft Press, 288 pages, 1997.
- [3] BOEHM, B.W. Software cost estimation with COCOMO II. Prentice Hall, 2000.
- [4] FENTON, N. E.; PFLEEGER, S. L. Software metrics: a rigorous & practical approach. 2ª Edição. Boston: PWS Publishing, 1999.
- [5] Albrecht, A.J. (1979) Measuring Application Development Productivity. Proceedings of IBM Applications Development Symposium, Monterey, 14-17 October 1979, 83.

Referências

- [6] HAZAN C.; STAA, A.v. Análise e Melhoria de um Processo de Estimativas de Tamanho de Projetos de Software. Monografias em Ciências da Computação nº 04/05, Departamento de Informática PUC-Rio, ISSN 0103-9741, Fevereiro 2005.
- [7] HAZAN, C. Análise de Pontos de Função: Uma Aplicação nas Estimativas de Tamanho de Projetos de Software. Engenharia de Software Magazine, Edição 2, Devmedia, pp.25-31.
- [8] IFPUG. Counting Practices Manual. Version 4.3.1, Janeiro, 2010.
- [9] KARNER, G.. Resource Estimation for Objectory Projects. Objectory Systems. September, 1993.