

FIA/P GRADUAÇÃO

DISCIPLINA: PROJETO DE SISTEMAS APLICADO AS MELHORES PRÁTICAS EM QUALIDADE DE SOFTWARE E GOVERNANÇA DE TI

**AULA:
12 -ESTIMATIVA DE SOFTWARE**

**PROFESSOR:
RENATO JARDIM PARDUCCI**

PROFRENATO.PARDUCCI@FIAP.COM.BR

[Renato Parducci - YouTube](#)

AGENDA DA AULA

- ✓ CMMi nível 2 de maturidade – MA
- ✓ MPS.br nível F - MDE
- ✓ Introdução à estimativa de SW
- ✓ Métricas de estimativa de esforço, tempo, custo de projetos com métricas - visão geral COCOMO, FPA, UCP

PRÁTICAS E NÍVEL 2 –PP

**Métricas de Estimativa de
Esforço, Custo e Prazo de Projetos
de Software**

ESTUDO DE CASO SIMULADO



O programa da qualidade da empresa de Dilan continua evoluindo no desenho de novos processos de trabalho.

Na última semana, três clientes muito importantes para Dilan reclamaram que receberam propostas de desenvolvimento de novos projetos com estimativas de projeto “chutadas”. Reclamaram da equipe de projetistas jogar poker para decidir a complexidade de cada item de desenvolvimento.

Consuelo foi acionada e apresentou a proposta de substituição da dinâmica de poker planning por uma que use métricas matemáticas desenvolvidas por estudiosos da área de engenharia e qualidade de software.

Consuelo sugeriu usar o FPA/APF (Function Point Analysis ou Análise de Pontos de Função) que é o modelo mais consagrado mundialmente para estimativas quantitativas de projetos de software. O FPA foi desenvolvido anos atrás, com a experiência da IBM e evoluiu ao longo dos anos com a sua aplicação em milhares de projetos pelo mundo.

Ela elaborou uma apresentação introdutória sobre o FPA e você, como membro da equipe de transformação da qualidade, vai assisti-la!

A estimativa de software serve para apontar qual será o comportamento de um projeto em termos do seu ESFORÇO (horas de trabalho), PRAZO e consequentemente CUSTO (custo da mão de obra, sem considerar compras de software, hardware e serviços que não tenham relação com o tamanho do código do software que será construído).

As estimativas devem ser feitas antes mesmo do detalhamento de um plano de projeto.

Elas servem para obter o patrocínio para o projeto de forma a liberar pessoas para trabalhar em suas atividades, inclusive as de planejamento.

ATIVIDADE PRÁTICA



Agrupados de 5 em 5 pessoas, determinem o prazo do seu grupo para realizar as seguintes entregas de um projeto de software:

- Criar um programa que possua uma tela onde o usuário digite informações de cadastro de um cliente, os quais serão gravados em um banco de dados;
- Criar um programa online, com tela de interface para usuário, que seja usada para informar produtos e valores vendidos a um cliente;
- Criar um programa de emissão de relatório em formato digital, que permita que o usuário escolha: Relatório 1 - valor total de vendas por cliente; Relatório 2 – valor detalhado de vendas por cliente; Relatório 3 – valor total de vendas mês a mês.

VAMOS COMPARAR OS RESULTADOS AO FINAL!

Um pouco mais sobre os principais modelos de estimativa...

Quando tentamos determinar alguma coisa futura em nossas vidas, assim como quanto tentamos avaliar os possíveis custos, esforço e tempo de um trabalho futuro, podemos adotar:

- **Previsão:** definir algo com base em probabilidades;
- **Projeção:** definir algo com base em um histórico passado (por analogia);
- **Predição:** adivinhar o futuro por sentimento.



https://www.google.com/search?biw=1242&bih=602&tbm=isch&sa=1&ei=b41oW57dMsypwgT77LnwDQ&q=proje%C3%A7%C3%A3o&oq=proje%C3%A7%C3%A3o&gs_l=img.3..0l10.53355.54512.0.54758.8.8.0.0.0.262.920.1j5j1.7.0....0...1c.1.64.img..1.7.918...35i39k1j0i67k1.0.mqh3ulj8iKU#imgsrc=3yIjzETERSV5M:

Um pouco mais sobre os principais modelos de estimativa...

Tanto para processos Clássicos quanto Ágeis de software, existem modelos de estimativa que possibilitam:

- A) Estimativa Direta:** atribui-se diretamente um valor estimado para fazer algo. Por exemplo: atribui-se um número de horas de trabalho para criar uma tabela no banco de dados.
- B) Estimativa Paramétrica:** utiliza-se um fator de referência, um item de avaliação de grau de complexidade, para determinar o esforço para fazer algo. Por exemplo: define-se o número de campos de dados que uma tabela de banco de dados vai ter e com base nessa informação, define-se o tempo para criar a tabela no banco de dados.

Um pouco mais sobre os principais modelos de estimativa...

Tanto para processos Clássicos quanto Ágeis de software, existem modelos de estimativa que possibilitam:

A) Estimativa Direta: POR EXEMPLO O PLANNING POKER!

B) Estimativa Paramétrica: ESTUDADOS A SEGUIR!

Um pouco mais sobre os principais modelos de estimativa...

- Ao longo dos anos de desenvolvimento de software, investiu-se no desenvolvimento de diversos modelos de estimativa de projeto.
- O primeiro modelo: Mito do Home/Mês (**The Mithical Man Month**), criado por Fred Brooks – associava-se um tempo para produzir uma linha de código por uma pessoa e multiplicava-se as horas de trabalho por linha de código pelo número total estimado de linhas de código a produzir, para calcular o prazo e custo do projeto.

Um pouco mais sobre os principais modelos de estimativa...

- O primeiro grande modelo que ganhou escala mundial e é adotado até hoje em várias empresas do mundo é o **COCOMO** que foi criado por Barry Boehm, com base no conceito original de Brooks (The Mythical Man Month) – **associava-se um tempo para produzir cada 1.000 linhas de código (KDSI) por uma pessoa, porém, acrescentou-se elementos de conversão de objetos funcionais a construir para linhas de código, através de tabelas de associação** (na versão 2.0 do COCOMO).
- O COCOMO ainda acrescenta a **possibilidades de ajustar as estimativas conforme fatores de influência do ambiente de produção do software, competência da equipe e aspectos não funcionais** esperados no produto final.

Um pouco mais sobre os principais modelos de estimativa...

- O **COCOMO** é o modelo de estimativas **mais complexo de ser aplicado**.
- Ele possui **submodelos Básico, Intermediário e Avançado** com possibilidades de ponderações e regras de cálculo diferentes, permitindo **adicionar julgamentos subjetivos** de variáveis condicionantes do projeto.
- Ainda possui **ponderações distintas de esforço, dependendo de como é o conhecimento e experiência** da equipe (concentrado em uma pessoa, distribuído entre poucas pessoas conhecidas ou difuso) – **modelo Orgânico, Semi-destacado ou Embutido**
- **Diferencia** ainda as situações de **manutenção** de software existente **e criação de software** completamente novo
- **Diferencia** a aplicação de **reuso de software** ou não.

Em todos modelos de estimativa de software:

- Estima-se a quantidade de itens de desenvolvimento ou **tamanho do projeto** (pode ser em número estimado de linhas de código, número de funcionalidades ou número de Classes a serem desenvolvidos);
- Pondera-se as quantidades de itens de desenvolvimento por **complexidade** (alta, média, baixa);
- Ajusta-se a estimativa conforme o **nível de reutilização** de componentes;
- Ajusta-se a estimativa conforme a **complexidade do ambiente** de desenvolvimento (existência de **ferramentas**, conhecimento disponível, **experiência** anterior com o mesmo tipo de projeto e **processo**);
- **Distribui-se** as estimativas **por fase do ciclo** de vida do projeto.

Um pouco mais sobre os principais modelos de estimativa...

- Existem vários outros modelos como MARK-II, SLIM, Simulação de Monte Carlo, APF/FPA, UCP
- Vamos nos dedicar a compreender a aplicar dois dos modelos de estimativa com base em métricas de software mais difundidos na atualidade: FPA-Function Point Analysis

APF / FPA
Estimativa de Projeto por Pontos
de Função

Vamos estudar como a APF (Análise de Pontos de Função) ou FPA (Function Point Analysis) trabalha e adaptar seus conceitos para estimar o esforço de produção na Sprint do SCRUM.

O FPA é desenvolvido pelo



com mais de 1.400 instituições associadas.

Ele está alinhado com as exigências da ISO 29881, de qualidade de software.

1º PASSO - Decomposição Funcional

Consiste em decompor o projeto ou a aplicação em elementos (componentes) funcionais básicos que já aprendemos a representar através dos Itens do Backlog de Produto do SCRUM.

Componente Funcional Básico (CFB), é uma unidade elementar dos Requisitos Funcionais do Usuário – captura o que tem que ser feito.

Ex. Um Requisito Funcional do Usuário poderia ser “Manter Clientes”, o qual poderia consistir dos seguintes CFBs: “Incluir um novo cliente”, “Reportar Compras do Cliente” e “Alterar Detalhes do Cliente”.

1º PASSO - Decomposição Funcional

Nas avaliações por pontos de função, é necessário distinguir claramente os

RF – Requisitos Funcionais



São computados em **pontos de função** (itens de desenvolvimento)

RNF – Requisitos Não Funcionais

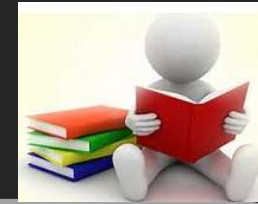


São computados como **fatores (de ajuste)** para ajustar a complexidade de desenvolvimento com um índice multiplicador de dificuldade

Não confundir RF com Requisito de Usuário!

RF podem conter requerimentos de desenvolvimento para implementar regras de negócio ou para resolver situações técnicas do projeto (requerimentos de engenharia e arquitetura).

ESTUDO DE CASO SIMULADO



Dilan e Consuelo querem usar um novo projeto que a software house precisa desenvolver, como estudo de caso para que a equipe aprenda a usar FPA.

A primeira tarefa da equipe é identificar os requisitos funcionais no escopo do projeto declarado a seguir em forma de história.

“Você foi contratado para desenvolver um sistema que permitirá o controle de vendas de uma loja de calçados.

Esse sistema precisa permitir cadastrar Produtos com detalhes de Código de fabricante, Nome do produto, Preço unitário, Unidade de medida. Exclusões de dados não serão permitidas.

Uma vez cadastrado, o produto pode ter seu preço alterado.

Todos produtos podem ser consultados por Código de Identificação ou Nome.

Uma função de exportação de dados deve existir, a qual poderá ser usada por um programa externo para obter dados do produto, através de um arquivo de extração.

A interface do sistema terá que ser compatível com WEB Browsers”

ESTUDO DE CASO SIMULADO



Agora...

Use a planilha FerramentaFPA-ProfRJP-Manual.xlsx

e preencha a coluna Componente Funcional Básico (CFB) com a lista de itens de backlog que representam o escopo do projeto.

“Você foi contratado para desenvolver um sistema que permitirá o controle de vendas de uma loja de calçados.

Esse sistema precisa permitir cadastrar Produtos com detalhes de Código de fabricante, Nome do produto, Preço unitário, Unidade de medida. Exclusões de dados não serão permitidas.

Uma vez cadastrado, o produto pode ter seu preço alterado.

Todos produtos podem ser consultados por Código de Identificação ou Nome.

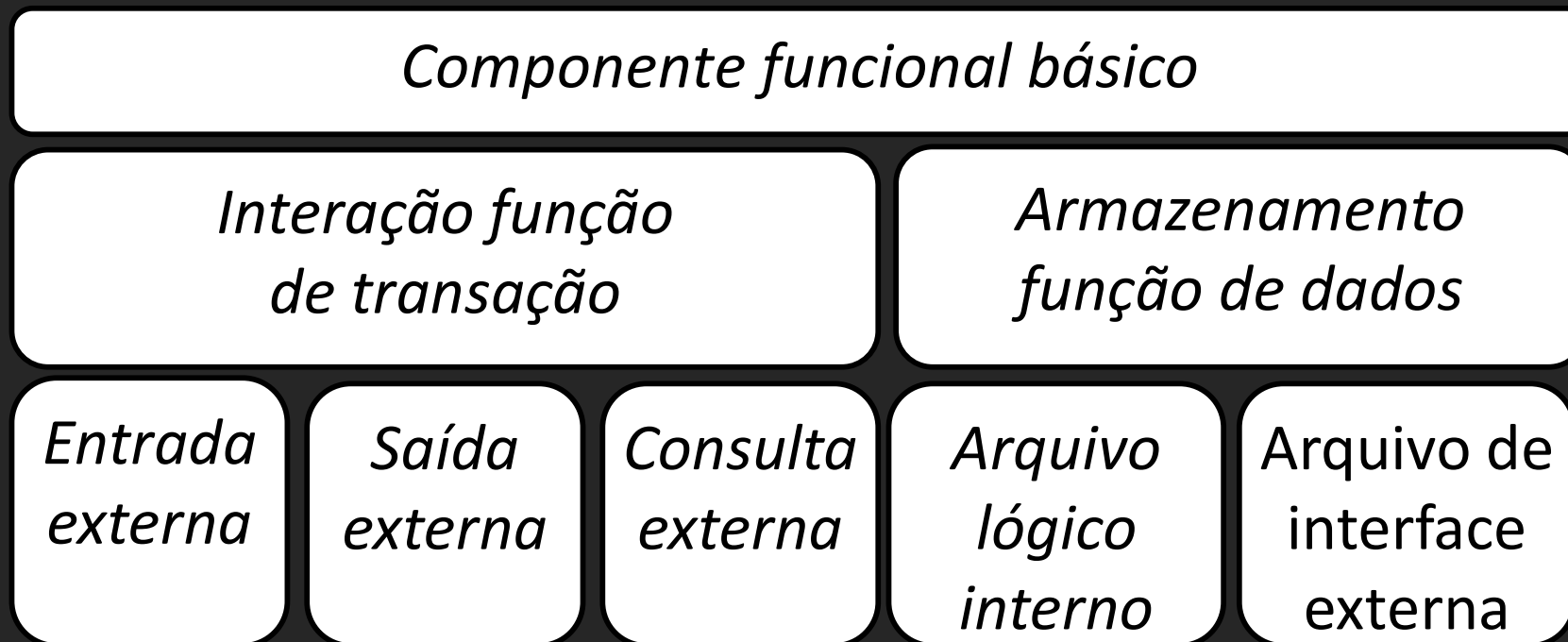
Uma função de exportação de dados deve existir, a qual poderá ser usada por um programa externo para obter dados do produto, através de um arquivo de extração.

A interface do sistema terá que ser compatível com WEB Browsers”

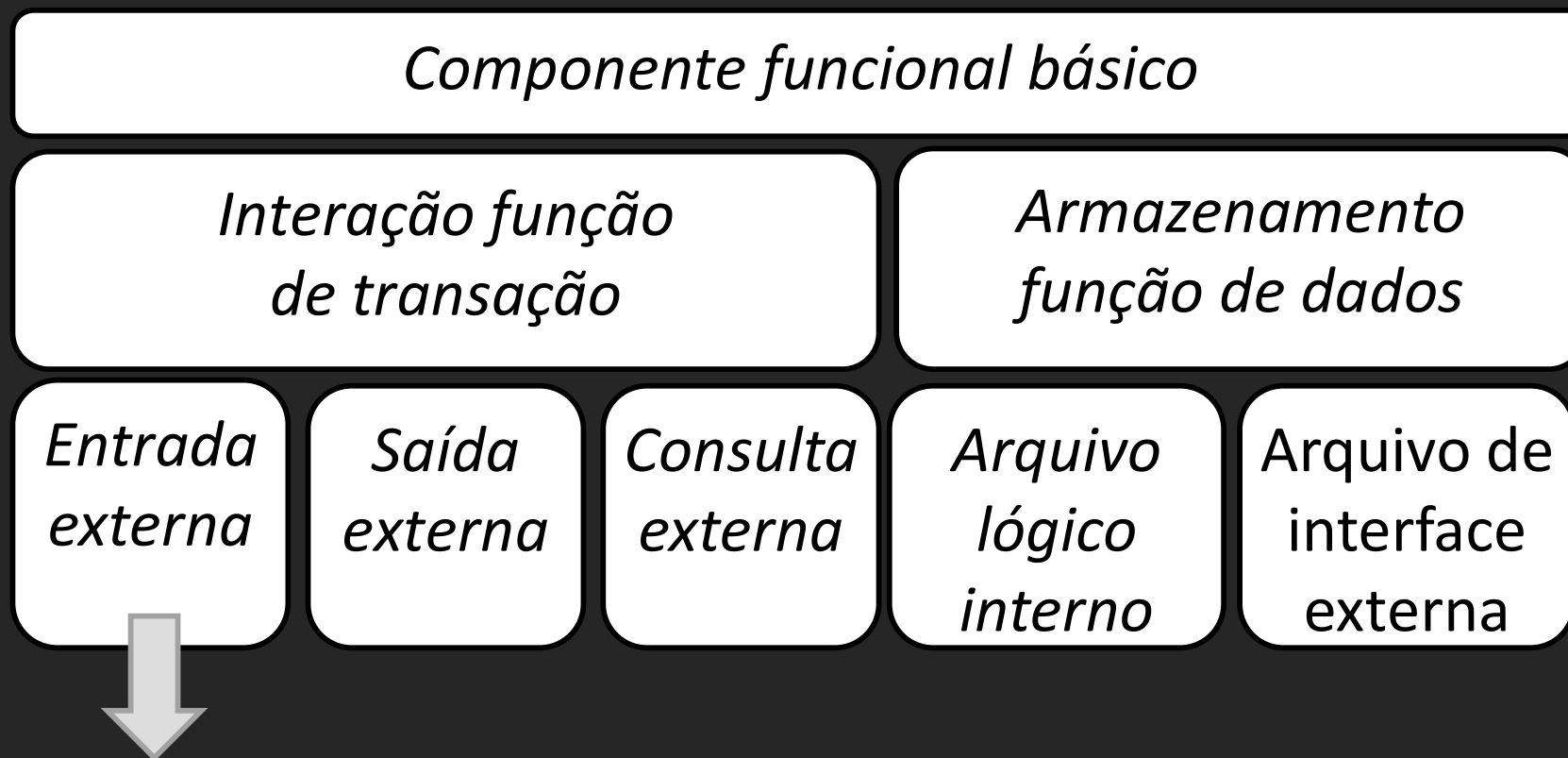
2º PASSO - Classificação Funcional

Os Componentes Funcionais Básicos (equivalentes aos Itens do Backlog de Produto), devem ser Classificados em categorias que demonstram qual o objetivo do componente.

O quadro a seguir representa como a APF classifica os CFBs.



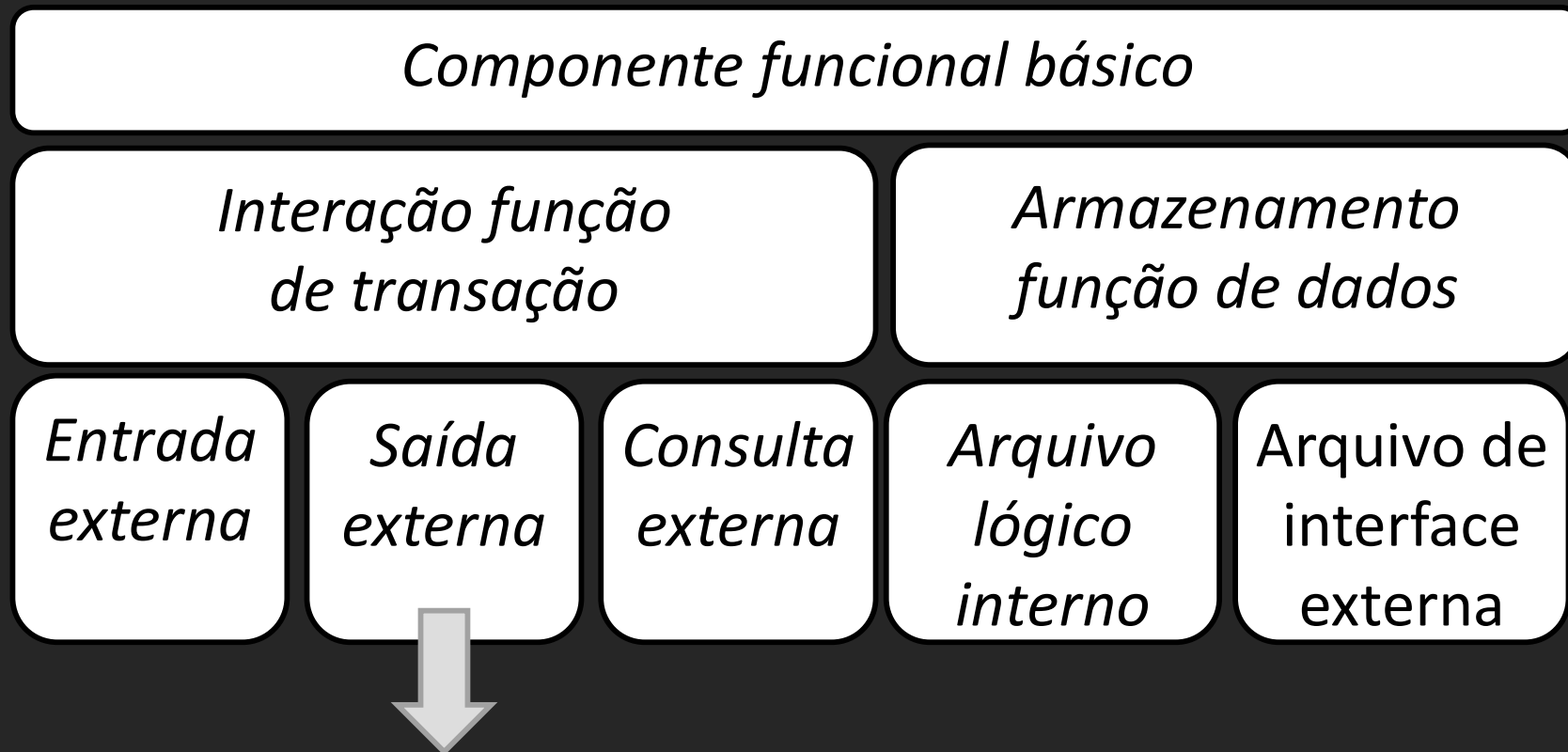
2º PASSO - Classificação Funcional



Corresponde a um componente que tem por objetivo capturar, transformar e armazenar dados que vêm de fora do sistema.

Exemplo: Tela de entrada de dados; Programa de carga de dados em massa à partir de um arquivo de entrada; ETL para carga de um Datawarehouse a partir de um banco de dados de uma área de transporte (Staging).

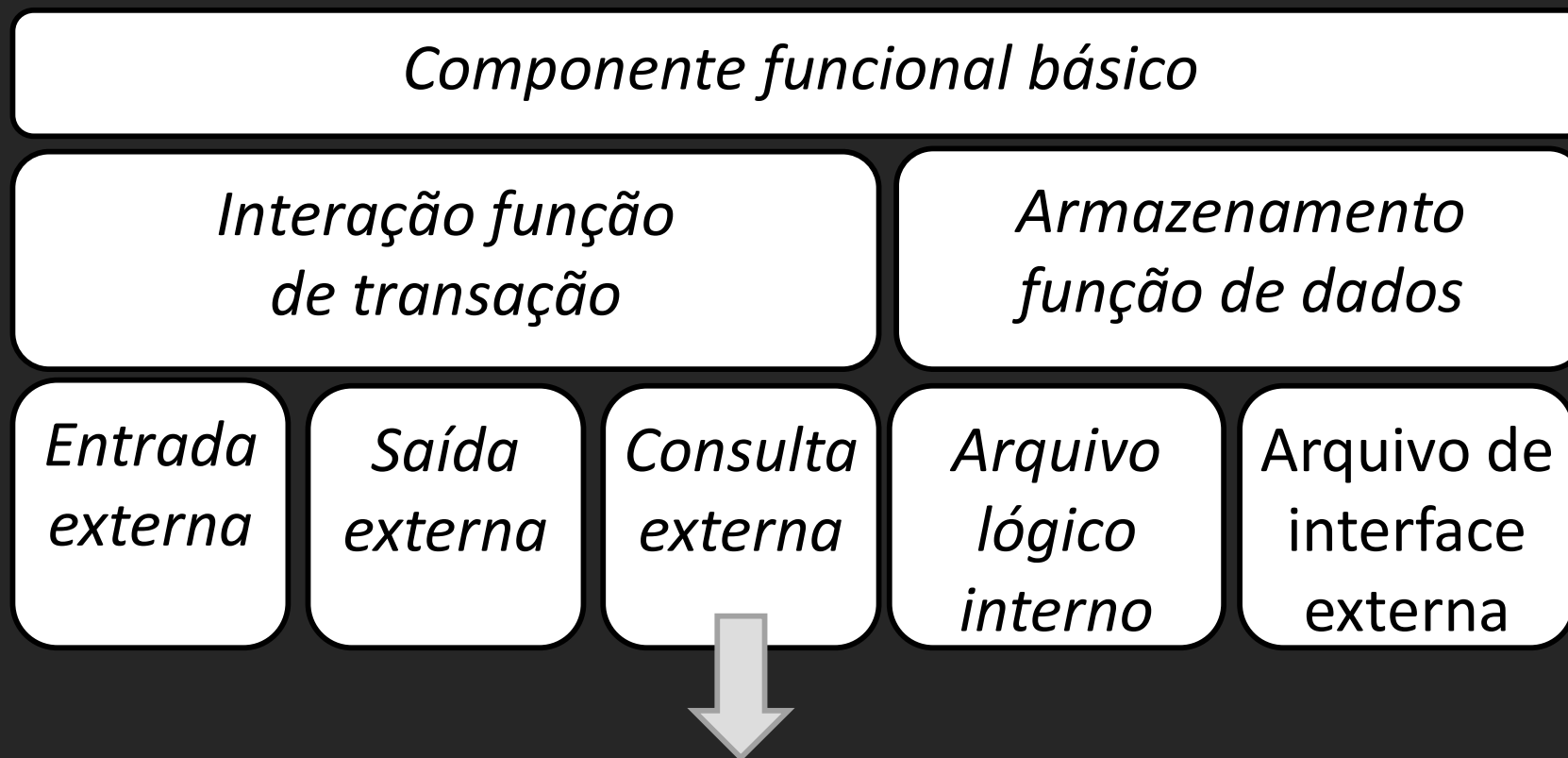
2º PASSO - Classificação Funcional



Corresponde a um componente de software que tem por objetivo gerar exportação de dados para fora da aplicação (interface) com base em dados armazenados. Envolve uma transformação dos dados na extração.

Exemplo: Geração de arquivo de carga para BI a partir de um banco relacional de ERP.

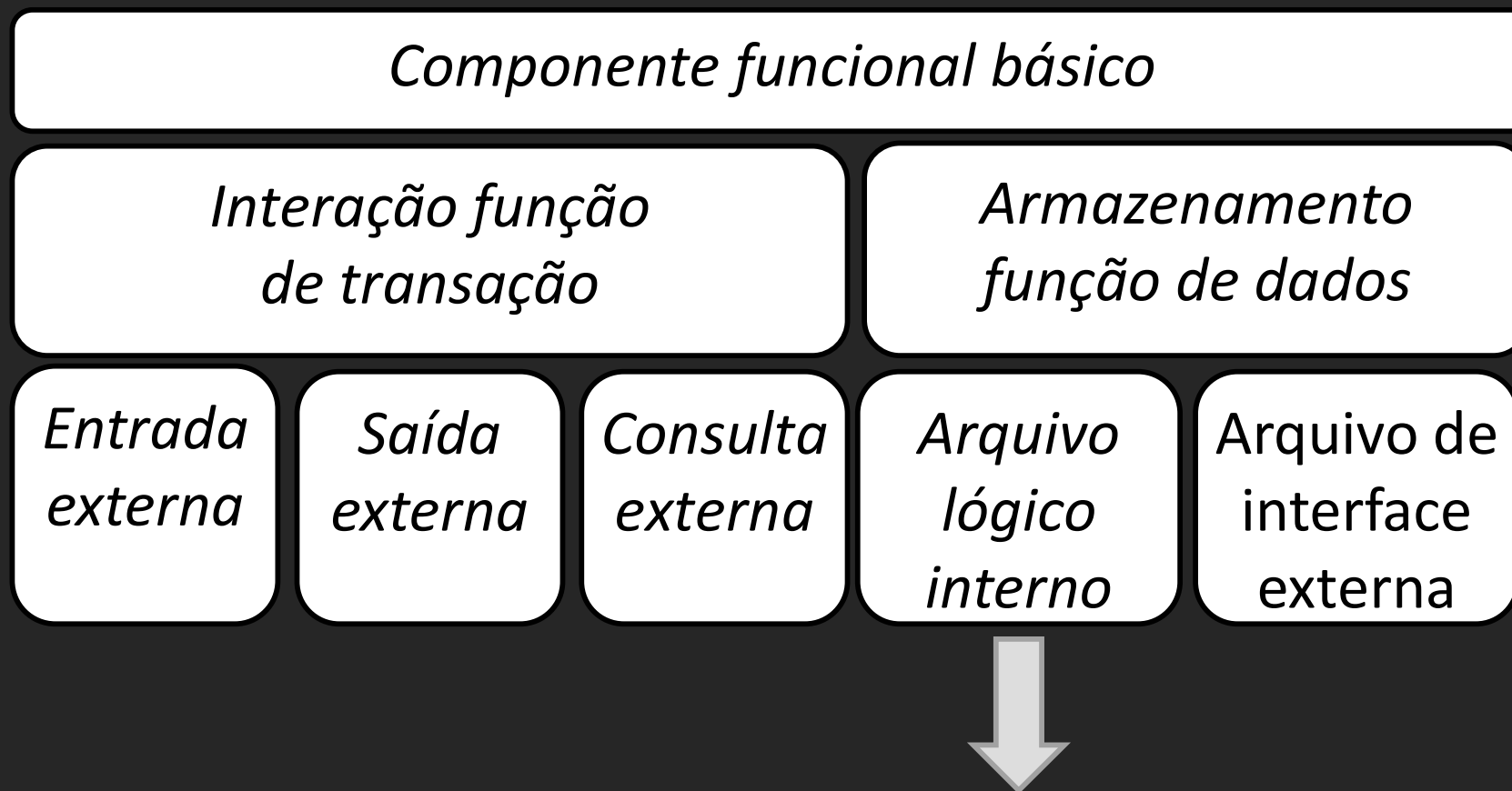
2º PASSO - Classificação Funcional



Corresponde a um componente de pesquisa de informação em formato impresso ou digital. Busca dados e os exibe sem transformá-los (diferente do que acontece na Saída Externa).

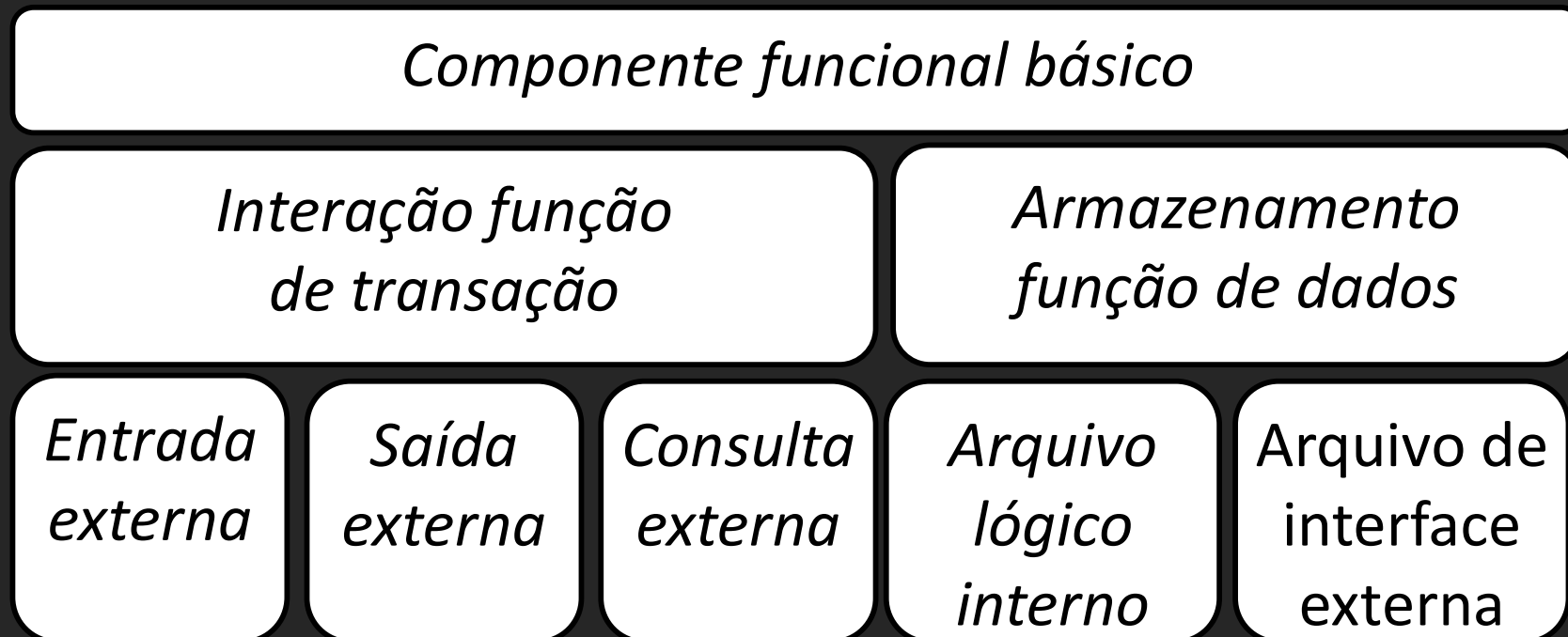
Exemplo: Página de consulta de dados na WEB; Tela de consulta de cadastro de um sistema; Script de consulta sobre bases de dados (SQL/Query).

2º PASSO - Classificação Funcional



Base de dados de uso do sistema que está sendo desenvolvido.
 Não é acessada diretamente por outros sistemas de informação.
 Não serve como arquivo de interface, portanto.

2º PASSO - Classificação Funcional



Base de dados usada na troca de dados entre o sistema em desenvolvimento e outros sistemas externos (legados).

É alimentada ou lida pelo sistema em desenvolvimento. Tem o objetivo de funcionar como elemento de interface de dados.

São essas classes que definirão a complexidade de produzir um componente.

Diversas empresas já adaptaram esse conceito de classificação, criando novas categorias de componentes para representar de forma mais fiel a forma como trabalham.

Alguns exemplos de categorias que empresas criam na hora de classificar seus componentes:

- Estrutura de Classe de reuso para a biblioteca de componentes:
- **Método de consulta/recuperação de dados simples de uma Classe;**
- **Método de atualização de múltiplos dados de uma Classe;**
- **Método de inclusão/instanciação de um Objeto em uma Classe.**

ESTUDO DE CASO SIMULADO



Considere que você e a equipe fecharam o seguinte escopo funcional para o projeto que estão estimando e classifique os CFBs (Componentes Funcionais Básicos). Use a planilha de estimativa onde você registrou os requisitos:

Item a desenvolver	Depende de	Seq		Complexidade
Base de dados de produtos	• -	1		
Função de Criação de Objetos/Registros	• Base de dados de produtos	2		
Função de Consulta de Objetos/Registros	• Função de Criação de Objetos/Registros	3		
Função de Atualização de dados	• Função de Criação de Objetos/Registros	4		
Função de Extração de dados cadastrais	• Função de Criação de Objetos/Registros	5		
Arquivo de dados extraídos	• Função de Extração de dados cadastrais	5		



<https://www.youtube.com/watch?v=tcf7xjnGoYY>

3º PASSO – Identificação de Dados Referenciados

Cada Componente Funcional Básico opera uma determinada quantidade de dados.

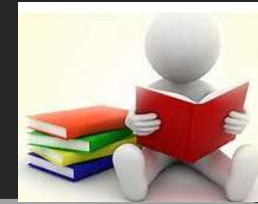
Essa quantidade de dados usados na execução da transação é fator preponderante da sua complexidade.

A APF chama esses dados de DER – Dado Elementar Referenciado.



DER

ESTUDO DE CASO SIMULADO



Agora, a tarefa é associar quais dados são acessados em cada uma das transações e arquivos mapeados como CFBs! No caso das Saídas Externas, você deve separar dados de entrada e dados de saída! Liste os dados e quantifique-os usando a planilha!

Item a desenvolver	Depende de	<u>Seq</u>	Classe	DER	Complexidade
Base de dados de produtos	• -	1			
Função de Criação de Objetos/Registros	• Base de dados de produtos	2			
Função de Consulta de Objetos/Registros	• Função de Criação de Objetos/Registros	3			
Função de Atualização de dados	• Função de Criação de Objetos/Registros	4			
Função de Extração de dados cadastrais	• Função de Criação de Objetos/Registros	5			
Arquivo de dados extraídos	• Função de Extração de dados cadastrais	5			

Para classificar a complexidade de Transações (as que não são de Armazenamento de dados), devemos considerar também o número de Arquivos Referenciados (ALR – Arquivo Lógico Referenciado) pela transação funcional.

Exemplo: se um Componente Funcional lê um arquivo do sistema e gera outro de exportação de dados para outro sistema, ele acessa 1 Arquivo Interno e 1 Arquivo Externo, portanto, referencia 2 ALR.

Obs: não computamos o número de arquivos para ALI e AIE.

Para classificar a complexidade de Arquivos (ALI ou AIE), devemos computar além do número de Dados Elementares (DER), o número de Tipos de Registros.

Um tipo de registro é um formato e TUPLA no arquivo, exemplo:

Id_Cliente	Nome-Cliente	Endereço-Cliente
C001	Joaresino	Rua Peloron 12
C002	Aurelina	Av. Brazuca, 111

1 único formato /
tipo de registro

Campo 1	Campo 2	Campo 3
C001	Joaresino	""
Rua Peloron	12	02332090
C002	Aurelina	""
Av. Brazuca	111	02333199

2 formatos / tipos de
registro diferentes
no mesmo arquivo
de dados

Registros variáveis em um mesmo arquivo são comuns em BIGDATA e COBOL.

ESTUDO DE CASO SIMULADO

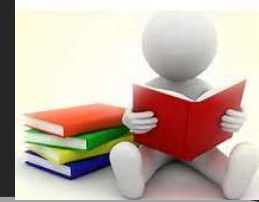


Para completar os parâmetros de avaliação de complexidade por CFB, você precisa associar quantidade de arquivos usados por transação e a quantidade de tipos de registro por arquivo! Lembre-se que no caso das Saídas Externas, é necessário separar dados e arquivos de entrada e de saída! Liste os arquivos e registros e quantifique-os usando a planilha de estimativa !

Item a desenvolver	Depende de	Seq	Classe	DER	Arquivos Registros	Complexidade
Base de dados de produtos	• -	1			?	
Função de Criação de Registros	• Base de dados de produtos	2			?	
Função de Consulta de Registros	• Função de Criação	3			?	
Função de Atualização de dados	• Função de Criação	4			?	
Função de Extração de dados	• Função de Criação	5			?	
Arquivo de dados extraídos	• Função de Extração	5			?	

Cálculos da dos Pontos de Função

ESTUDO DE CASO SIMULADO



Consuelo vai continuar o treinamento prático de FPA, usando como base um projeto real da empresa de Dilan.

É hora de calcular a complexidade de cada componente funcional a ser produzido no projeto. Para isso, você não vai mais jogar poker!

O FPA possui um modelo matemático desenvolvido à partir dos projetos que o grupo internacional de usuários (IFPUG) trabalhou.

Veja as instruções de Consuelo para que você realize a estimativa inicial do projeto, lembrando que até esse momento, você só considerou aspectos funcionais do escopo do projeto de software.

4º PASSO – Classificação da Complexidade Funcional

Cada Componente Funcional Básico opera uma determinada quantidade de dados e arquivos.

Cada arquivo opera sobre uma quantidade de dados e registros.

Vamos continuar aprendendo APF com base no estudo de caso da última aula...

Agora, vamos atribuir os PESOS a cada item do Backlog de Produto.

Os PESOS representarão o Esforço em Pontos de Função a desenvolver.

A APF possui um conjunto de tabelas de métricas para definir a complexidade de cada item de desenvolvimento, apresentadas a seguir.

PARA ARQUIVOS LÓGICO

Arquivos (ALI/AIE)

Quantidade de tipos/formatos de registros	Número de Dados elementares por registro		
	1 a 19	20 a 50	Mais de 50
1	Simples	Simples	Médio
2 a 5	Simples	Médio	Complexo
6 ou mais	Médio	Complexo	Complexo

Trabalharemos somente com 1 tipo de registro.
Múltiplos tipos de registros implicam em mais de um formato de guarda de dados em um mesmo arquivo, o que não é comum em Bancos de Dados estruturados.

PARA ENTRADAS EXTERNAS

Entradas Externas (EE)

Quantidade de arquivos referenciados	Número de Dados elementares referenciados		
	1 a 4	5 a 15	16 ou mais
0 ou 1	Simples	Simples	Médio
2	Simples	Médio	Complexo
3 ou mais	Médio	Complexo	Complexo

Perceba que no caso de transações de Entrada, assim como nas de Saída e Consulta, o número de arquivos operados também é fator de complexidade!

PARA CONSULTAS EXTERNAS

Consultas Externas (CE)

Quantidade de arquivos referenciados	Número de Dados elementares referenciados		
	1 a 5	6 a 19	20 ou mais
0 ou 1	Simples	Simples	Médio
2 a 3	Simples	Médio	Complexo
4 ou mais	Médio	Complexo	Complexo

PARA SAÍDAS EXTERNAS

Ponderação das saídas (transformação de dados)

Quantidade de arquivos referenciados	Número de Dados elementares referenciados		
	1 a 5	6 a 19	20 ou mais
0 ou 1	Simples	Simples	Médio
2 a 3	Simples	Médio	Complexo
4 ou mais	Médio	Complexo	Complexo

Ponderação das entradas (consulta para transformar)

Quantidade de arquivos referenciados	Número de Dados elementares referenciados		
	1 a 4	5 a 15	16 ou mais
0 ou 1	Simples	Simples	Médio
2	Simples	Médio	Complexo
3 ou mais	Médio	Complexo	Complexo

Prevalece a ponderação mais alta entre os dois quadros

ESTUDO DE CASO SIMULADO



Mãos à obra!

Você deve agora, calcular qual a complexidade de cada componente funcional básico, usando as matrizes do FPA que constam na planilha de estimativas!

Item a desenvolver	Depende de	Seq	Classe	DER	Arquivos Registros	Complexidade
Base de dados de produtos	• -	1	ALI	4	1	
Função de Criação de Registros	• Base de dados de produtos	2	EE	4	1	
Função de Consulta de Registros	• Função de Criação	3	CE	4	1	
Função de Atualização de dados	• Função de Criação	4	EE	3	1	
Função de Extração de dados	• Função de Criação	5	SE	6 O 4 I	1 – Out 1 – In	
Arquivo de dados extraídos	• Função de Extração	5	AIE	6	1	

ESTUDO DE CASO SIMULADO



Mãos à obra!

Faça por fim, uma matriz, mostrando o total de componentes funcionais básicos por tipo e complexidade!

	SIMPLES	MÉDIO	COMPLEXO
ALI			
AIE			
EE			
CE			
SE			

ESTUDO DE CASO SIMULADO



Ainda não chegamos a associar pontos de esforço de projeto.

Consuelo vai lhe mostrar agora, como fazer isso!

5º PASSO – Ponderação da Complexidade Funcional

Agora, vamos atribuir os PESOS representarão o Esforço em Pontos de Função a desenvolver.

A APF possui uma ponderação padrão de peso para cada Tipo de Componente Funcional e Complexidade.

A multiplicação do número de componentes por tipo/complexidade pelo peso determinará o que a Análise de Pontos de Função chama de PONTOS DE FUNÇÃO NÃO AJUSTADOS.

O termo não ajustado é usado para deixar claro que não foram consideradas ainda as questões sobre o ambiente de produção e os requisitos não funcionais do produto.

Algumas empresas criam suas próprias matrizes de ponderação de pontos de função mas a qualidade desse trabalho depende de criar uma base histórica de projetos que seja suficiente para testar a credibilidade dos cálculos!

A tabela de pesos por Tipo de Componente Funcional e Complexidade da APF é:

Os pesos da tabela equivalem àquilo que produzimos com o Planning Poker no SCRUM – Na verdade esse processo de estimativa de pesos substitui a prática do Planning Poker!

Fatores de ponderação

	Multiplicador por complexidade		
	Simples	Médio	Complexo
ALR	x7	x10	x15
CE	x4	x5	x7
EE	x3	x4	x6
SE	x4	x5	x7

ALR é um ALI ou AIE

A tabela de pesos por Tipo de Componente Funcional e Complexidade da APF é:

Empresas que praticam esse tipo de estimativa acabam montando tabelas de ponderação onde os Multiplicadores correspondem ao Número de Dias ou Horas para produzir um Componente Funcional de determinado tipo e complexidade

Fatores de ponderação

	Multiplicador por complexidade		
	Simples	Médio	Complexo
ALR	x7	x10	x15
CE	x4	x5	x7
EE	x3	x4	x6
SE	x4	x5	x7

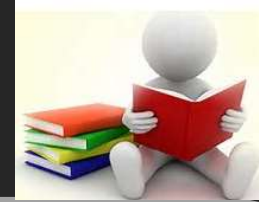
ALR é um ALI ou AIE

De posse das pontuações, aplicamos a seguinte equação padrão de ajuste, criada pela IFPUG/FPA:

Fator de Ajuste Final = $0,65 + (0,01 \times \text{Fator de Influência Total})$

PF Ajustados Finais = Total de PF não ajustados x Fator de Ajuste Final

ESTUDO DE CASO SIMULADO

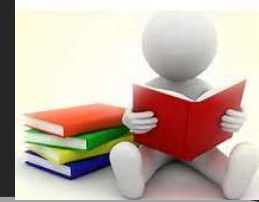


Calcule o número de pontos de função (equivalentes aos pontos que antes você atribuía com poker no SCRUM)!

Lembre-se que você está calculando ponderações chamadas Pontos de Função Brutos, por que representam somente a parte funcional do projeto.

Use as matrizes de pontuação do FPA que estão na sua planilha de estimativas!

ESTUDO DE CASO SIMULADO



Consuelo vai apresentar a você como calcular o esforço total do projeto, considerando inclusive os requisitos não funcionais para o software.

Preste atenção na explicação dela para finalizar suas estimativas!

5º PASSO – Cálculo do Esforço de Projeto

Tendo em mãos os pontos de função não ajustados, aplicaremos fatores de ajuste que influenciam a produtividade da equipe no desenvolvimento do software.

Os fatores de ajuste são ligados às ferramentas e técnicas empregadas e aos requisitos não funcionais do software.

A APF possui uma tabela de fatores de ajustes que estudaremos a seguir.

Algumas empresas criam sua própria lista de fatores de ajuste e fórmulas de aplicação mas a qualidade desse trabalho depende de criar uma base histórica de projetos que seja suficiente para testar a credibilidade dos cálculos!

A lista de Fatores de Ajuste do APF é dada pelos 14 critérios a seguir, que devem receber notas de 0 a 5, sendo 5 para coisas que dificultam muito o projeto e 0, para coisas que não influenciam o projeto.

1. TELEPROCESSAMENTO
2. PROCESSAMENTO DISTRIBUÍDO
3. PERFORMANCE
4. CARGA DE MÁQUINA
5. VOLUME DE TRANSAÇÕES
6. ENTRADA DE DADOS ON-LINE
7. ATUALIZAÇÕES ON-LINE
8. EFICIÊNCIA DO USUÁRIO FINAL
9. COMPLEXIDADE DE PROCESSAMENTO
10. REUTILIZAÇÃO DE CÓDIGO
11. FACILIDADE DE IMPLANTAÇÃO
12. FACILIDADE DE OPERAÇÃO
13. FACILIDADE DE MANUTENÇÃO / ALTERAÇÕES
14. OPERAÇÃO EM MÚLTIPLOS LOCAIS

O FPA descreve detalhadamente quando atribuir cada peso do range de 0 a 5 em seu documento original.

ESTUDO DE CASO SIMULADO



Mãos à obra!

Faça a apuração dos pontos de função ajustados!

Leve em conta que o software será implantado em 15 filiais que o usarão via WEB/Internet, e a base de dados e backend ficarão instalados em um servidor corporativo!

- | | | |
|-----|----------------------|---------------------------------------|
| 1. | <input type="text"/> | TELEPROCESSAMENTO |
| 2. | <input type="text"/> | PROCESSAMENTO DISTRIBUÍDO |
| 3. | <input type="text"/> | PERFORMANCE |
| 4. | <input type="text"/> | CARGA DE MÁQUINA |
| 5. | <input type="text"/> | VOLUME DE TRANSAÇÕES |
| 6. | <input type="text"/> | ENTRADA DE DADOS ON-LINE |
| 7. | <input type="text"/> | ATUALIZAÇÕES ON-LINE |
| 8. | <input type="text"/> | EFICIÊNCIA DO USUÁRIO FINAL |
| 9. | <input type="text"/> | COMPLEXIDADE DE PROCESSAMENTO |
| 10. | <input type="text"/> | REUTILIZAÇÃO DE CÓDIGO |
| 11. | <input type="text"/> | FACILIDADE DE IMPLANTAÇÃO |
| 12. | <input type="text"/> | FACILIDADE DE OPERAÇÃO |
| 13. | <input type="text"/> | FACILIDADE DE MANUTENÇÃO / ALTERAÇÕES |
| 14. | <input type="text"/> | OPERAÇÃO EM MÚLTIPLOS LOCAIS |

Atribua pontos de 0 a 5
para cada fator,
considerando o projeto

ESTUDO DE CASO SIMULADO



Após as atividades de estimativas, Dilan questionou a equipe que não será possível apresentar uma estimativa de esforço para o cliente em PONTOS. Ele quer que a estimativa seja feita em horas!

Para atender a esse objetivo, como é a primeira vez que a empresa trabalha com FPA, Consuelo propôs que realizem uma SPRINT 0 no projeto, para desenvolver um primeiro conjunto de atividades e entregas e computar até o final da SPRINT, quantas horas foram gastas para realizar os pontos realmente cumpridos nessa SPRINT, gerando uma matriz “de para” de pontos para horas da SPRINT 0. Essa dinâmica deve ser repetida em todas as SPRINTS de forma a criar uma matriz “de para” de onde se possa apurar a média geral de horas por ponto em um projeto, considerando a performance da equipe que Dilan possui.

Para esse primeiro projeto, ajustes de estimativas e custos terão que ser negociados a cada planejamento de SPRINT como parte de um processo de ajuste fino. Isso terá que ser negociado com o cliente.

Consuelo vai explicar agora, como calcular o custo com FPA...

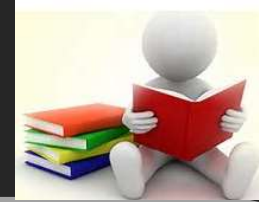
IMPORTANTE: a estimativa por ponto de função prevê que no total de esforço calculado estão todas as tarefas de desenvolvimento (que serão abertas no Backlog de Sprint que estudamos no SCRUM, por exemplo).

O esforço calculado contempla portanto as horas para:

- **Modelar o software**
- **Prototipar**
- **Criar casos de teste**
- **Construir**
- **Aplicar testes**
- **Liberar o pacote produzido**

Para calcular o custo do projeto, basta multiplicar o custo médio hora do desenvolvedor pelo número de horas do projeto. Você pode trabalhar também com custo por ponto de função ajustado.

ESTUDO DE CASO SIMULADO



Imagine que sua equipe tenha um salário médio de R\$ 50,00 por hora e calcule o custo do projeto para o nosso estudo de caso que tem um total de 27,16 pontos de função ajustados.

Considere que cada ponto de função ajustado corresponda a 1 hora de esforço de trabalho.

Imagine que sua equipe tenha um salário médio de R\$ 50,00 por hora e calcule o custo do projeto para o nosso estudo de caso que tem um total de **27,16 pontos de função ajustados**.

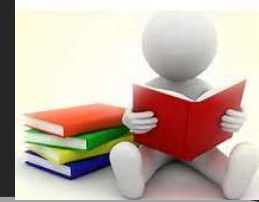


Considere que as tabelas de métricas de pontos de função foram montadas de forma que o número de pontos de função corresponda ao número de horas de trabalho.

Solução:

$$\text{Custo do projeto} = 27,16 \times 50 = \text{R\$ } 1.358,00$$

ESTUDO DE CASO SIMULADO



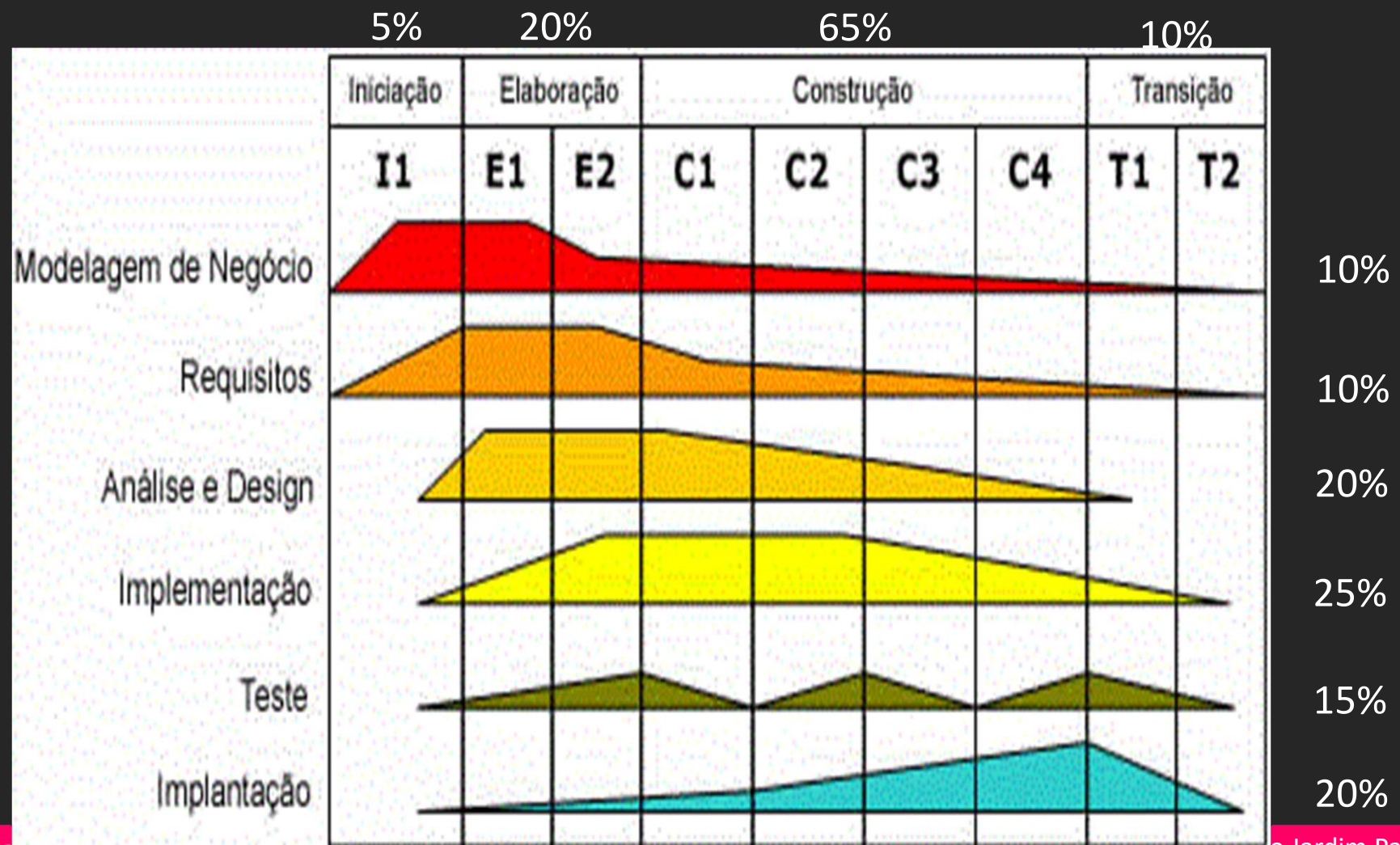
Para finalizar os estudos de métricas de estimativa de projeto de software, Consuelo tem algumas considerações a fazer...

Agora você já descobriu como estimar rapidamente o esforço e custo de um projeto a partir de uma lista de requisitos de software!

Essa é a forma ideal de prever o preço e número de horas de trabalho que são colocados no Termo de Abertura de Projeto, estudado no PMBoK/PMI, quando buscamos a liberação de recursos para iniciar um novo projeto de software.

Em métodos Ágeis como SCRUM, essa estimativa serve para dar uma previsão negociável para que o cliente consiga administrar seu orçamento e expectativas de resultados.

Se você estiver usando APF para estimar um projeto que seguirá um processo tradicional de desenvolvimento como o RUP, você pode distribuir o esforço total por fase ou competência aplicada, usando a tabela do RUP:

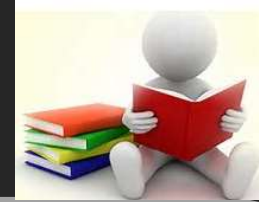


Se você estiver usando o SCRUM (processo ágil), o desenvolvimento acontece em Sprints de duração fixa!

Se a sua empresa trabalha com Sprints de 15 dias e os desenvolvedores vão trabalhar os itens do backlog em sequência, então você deve dividir o esforço estimado pelo prazo de cada Sprint, descobrindo assim o número de Sprints e o tempo total e projeto, afinal, as Sprints tem capacidade em horas ou pontos, sempre fixas.

Para criar frentes paralelas de projeto no SCRUM, você pode selecionar quais itens de produto vão para cada Sprint que correrá em paralelo e calcular os pontos de função ajustados de cada frente de trabalho (cada Sprint tratará um número de pontos de função ajustados que corresponde aos itens do backlog de produto selecionados para a Sprint).

ESTUDO DE CASO SIMULADO



Calcule o número de Sprints para concluir o projeto do estudo de caso da empresa de Dilan, considerando uma quantidade de pontos proporcional por Sprint.

Assim, você vai determinar uma expectativa de prazo.

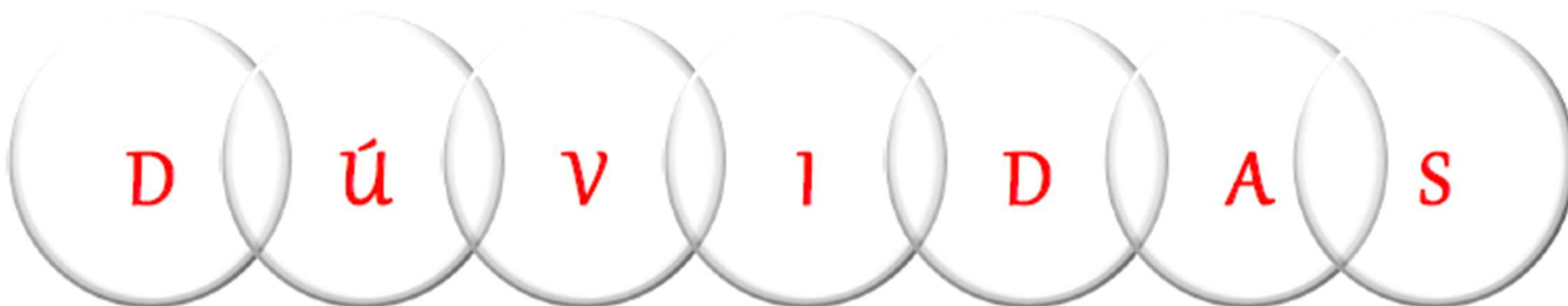
ATENÇÃO: O MODELO DE ESTIMATIVAS SÓ ALCANÇARÁ A MATURIDADE QUANDO A EMPRESA FOR CAPAZ DE AJUSTAR AS PONDERAÇÕES DE ESFORÇO E FATORES DE AJUSTE EM RAZÃO DA REALIDADE DOS SEUS PROJETOS!

NÃO EXISTE BOA ESTIMATIVA SEM UMA BASE HISTÓRICA DE PROJETOS!

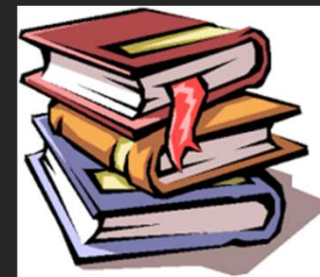
O MODELO ORIGINAL DEVE RECEBER UM *TUNNING* CONFORME AS CARACTERÍSTICAS DOS PROJETOS DA INSTITUIÇÃO E O APLICA!



<https://youtu.be/6lqMSUoA5Bk>



Referência bibliográficas



BIBLIOGRAFIA :

- PRESSMAN, Roger S.. Engenharia de software. - Uma abordagem profissional, 7ª edição. São Paulo, AMGH, 2011.

ESTIMATIVA POR PONTOS DE FUNÇÃO

Continua na próxima aula...

PROFESSOR:
RENATO JARDIM PARDUCCI

PROFRENATO.PARDUCCI@FIAP.COM.BR