



Fundamentos Engenharia de Software

UNIDADE 03

Estimativas de pontos de função

Nesta Unidade, conheceremos em detalhes os elementos da técnica de estimativa que utiliza pontos de função como base de contagem. Também aprenderemos como aplicar essa técnica por meio de exemplos a cada elemento apresentado. A Unidade tem início com uma breve introdução da técnica e na sequência é discutido cada elemento, como identificá-lo, como atribuir complexidade e, por fim, chegar ao ponto de função. Ao final da primeira parte, é apresentado um exemplo consolidado de um sistema para controle de vendas em uma loja de roupas, a fim de demonstrar como a contagem deve ser aplicada. Após a compreensão de como é feita a contagem de pontos não ajustados, são apresentados os fatores técnicos que ajustam a contagem de pontos a partir da complexidade técnica. Quanto maior a complexidade, maior será o total de

pontos de função ajustados. Por fim, a Unidade apresenta os cálculos necessários para estimar o tempo de desenvolvimento e o custo envolvido a partir da contagem de pontos de função.

Pontos de função

A Análise de Pontos de Função (APF) é uma técnica paramétrica de estimativa de esforço para o desenvolvimento de software e usa como parâmetro os conceitos de funções de dados e funções transacionais, que basicamente correspondem aos requisitos funcionais de um sistema.

A medida coletada pela APF independe da tecnologia utilizada e/ou da linguagem de programação, em que a(s) funcionalidade(s) foi(ram) implementada(s). A técnica foi proposta em 1979, por Alan J Albrecht, quando ele pesquisava sobre produtividade na IBM. Atualmente, ela é padronizada pelo International Function Point User Group (IFPUG). Para maior aprofundamento, o site pode ser consultado: <https://www.ifpug.org/>.

É importante ressaltar que essa técnica é bastante utilizada em contratos com o governo. Ela pode ser aplicada para medir o tamanho de um sistema antes de desenvolvê-lo, de forma que seu custo e tempo de desenvolvimento seja previsto adequadamente. Além disso, a APF pode ser utilizada para processos de manutenção, seja de melhorias ou inclusão de novas funcionalidades, e pode ser aplicada para estimar o tamanho funcional de uma aplicação já existente, também.

De maneira geral, a técnica é aplicada de acordo com os seguintes passos:

- Determinar o tipo de contagem (novo projeto, manutenção/melhoria ou aplicação existente).
- Determinar os limites ou fronteiras da aplicação (escopo).
- Identificar e atribuir valor em pontos de função para as funções de dados.
- Identificar e atribuir valor em pontos de função para as funções transacionais.

Identificação das funcionalidades

Um ponto importante na aplicação da técnica é considerar apenas as funcionalidades que são visíveis para o usuário, uma vez que a técnica tem como ponto de partida as funcionalidades sob o ponto de vista do usuário. Portanto, não é qualquer funcionalidade que conta.

Apenas as transferências de informação para dentro e para fora da fronteira do sistema são reconhecíveis pelo usuário.

A técnica APF avalia duas naturezas de funções:

+ **Funções de dados**

Podendo ser Arquivos Lógicos Internos (ALI) e Arquivos de Interface Externa (AIE). São elementos de informações reconhecíveis pelo usuário e que possam ser representados por classes, no modelo conceitual, ou tabela, no modelo relacional. Não se conta classes ou tabelas que são de apoio para a aplicação, tais como tabelas temporárias, classes de controle ou de interface etc.

+ **Funções transacionais**

Podendo ser Entradas Externas (EE), Consultas Externas (CE) e Saídas Externas (SE). Devem ser elementos elementares, isto é, de único passo. Uma transação é a menor unidade de atividade sob o ponto de vista do usuário.

O entendimento de cada um destes elementos é decisivo para a aplicação da técnica APF. A seguir, discutiremos em mais detalhes cada uma das funções.

Funções de dados

Conforme visto anteriormente, as funções de dados se dividem em ALIs e AIEs. A seguir, confira o que são e como identificá-los.

ALI

Um ALI pode ser considerado um elemento do modelo conceitual percebido pelo usuário e mantido pelo sistema. Um conceito bem importante neste tipo de arquivo é o de Registros Lógicos (RL), que correspondem aos subconjuntos de informações reconhecidas pelo usuário que está dentro de um ALI. Por exemplo, se um automóvel é composto por motor, carroceria e pneus, apenas o conceito automóvel é considerado um ALI. E os subconjuntos de dados contidos neste ALI são considerados RL. Assim, um ALI é uma classe, que caso participe de uma composição ou agregação, ocupa o lugar mais alto da hierarquia. Um ALI pode ter componentes, mas não ser um componente. Este entendimento é importante para a identificação da complexidade das funções de dados, tanto para ALI quanto para AIE, que veremos a seguir.



EXEMPLO

ALI em um contexto de uma aplicação que gerencia as vendas em uma loja de roupas:

- Cliente (dados pessoais do cliente).
- Produto (informações do produto).
- Venda (dados da venda).

AIE

Um AIE pode ser considerado um elemento do modelo conceitual percebido pelo usuário e mantido externamente por outras aplicações. Os dados obtidos por meio de web service externo à aplicação são exemplos de AIE. O conceito de RL também se aplica a AIE, ou seja, um AIE pode ser componente, mas não ser um componente.



EXEMPLO

AIE em um contexto de uma aplicação que gerencia as vendas em uma loja de roupas:

- Dados de crédito (web service integrado com o Serasa para identificar a situação de crédito do cliente).

- Débitos municipais (integração com a prefeitura para obter certidão negativa de débitos).

Tabela de complexidade das funções de dados

Após a identificação de todos os ALI e dos AIE que fazem parte do escopo da aplicação, é necessário identificar a complexidade de cada arquivo. A complexidade é definida por meio da Tabela 1 (item a).

A tabela a seguir apresenta a **complexidade funcional** (baixa, média ou alta) para cada função de dados (ALI ou AIE) e função transacional (EE, SE e CE) de acordo com parâmetros. A tabela também apresenta para cada tipo de função (ALI, AIE, EE, SE e CE) e complexidade funcional (baixa, média ou alta) a quantidade de pontos de função não ajustados que devem ser considerados.

Tabela 1: Complexidade funcional e pontos de função não ajustados

Complexidade funcional											
ALI e AIE (a)			EE (b)		SE e CE (c)						
#TDE				#TDE				#TDE			
#RL	1 a 19	20 a 50	51 ou +	#AL	1 a 4	5 a 15	16 ou +	#A L	1 a 5	6 a 19	20 ou +
1	Baixa	Baixa	Média	0 a 1	Baixa	Baixa	Média	0 a 1	Baixa	Baixa	Média
2 a 5	Baixa	Média	Alta	2	Baixa	Média	Alta	2 a 3	Baixa	Média	Alta
6 ou +	Média	Alta	Alta	3 ou +	Média	Alta	Alta	4 ou +	Média	Alta	Alta

Pontos de função não ajustados por tipo de função e sua complexidade (d)

Complexidade funcional			
Tipo de função	Baixa	Média	Alta

ALI	7	10	15
AIE	5	7	10
EE	3	4	6
SE	4	5	7
CE	3	4	6

Fonte: O autor (2021).

Em que:

(#RL): corresponde a um subconjunto de dados reconhecível pelo usuário dentro de um ALI.

Tipo de Dado Elementar (#TDE): corresponde a uma unidade de informação (um campo) reconhecível pelo usuário. Normalmente, é um atributo em uma classe ou um campo em uma tabela.



EXEMPLO

No contexto de uma aplicação que gerencia as vendas em uma loja de roupas, a função de dados **cliente** (nome, data de nascimento, CPF, RG, estado civil, fone e e-mail) teria a classificação **baixa**, porque:

- Função de dados = ALI: é um elemento do modelo conceitual mantido pela aplicação.
- RL = 1: não apresenta nenhum subconjunto de dados.
- #TDE = 7: não ultrapassa de 19 campos.

Funções transacionais

As funções transacionais se dividem em EE, SE e CE. A seguir, será apresentado o que são e como identificá-los.

EE

A EE consiste em uma entrada de dados ou controle que tem como consequência a alteração do estado interno dos dados dos sistemas. Esta função tem o objetivo de incluir, alterar ou excluir dados. Caso a operação ocorra em dois passos, ou seja, a consulta de um registro e posteriormente a sua alteração, devem ser contadas duas funções, uma que será CE e outra, EE.



EXEMPLO

EE em um contexto de uma aplicação que gerencia as vendas em uma loja de roupas:

- Cadastrar cliente.
- Alterar cliente.
- Excluir cliente.
- Cadastrar produto.
- Alterar produto.
- Excluir produto.

Além disso, se uma entrada externa produzir um conjunto de mensagens de erros, cada mensagem conta como um argumento de entrada.

SE

Uma SE consiste em uma saída de dados que pode ser precedida ou não da entrada de parâmetros. Pelo menos um dos dados de saída deve ser derivado, ou seja, calculado. SE são funções que recuperam dados do sistema e apresentam ou usuário ou enviam à outras aplicações, sendo que pelo menos um valor calculado deve existir para ser considerado uma saída externa.



EXEMPLO

SE em um contexto de uma aplicação que gerencia as vendas em uma loja de roupas. Considera-se neste exemplo que é necessário informar o CPF do cliente e a partir dele, filtrar todas as vendas realizadas com um totalizador ao final.

- Relatório de vendas de um cliente com totalização.

CE

A CE é uma saída de dados que pode ser precedida ou não de entrada de parâmetros. Os dados armazenados no sistema não sofrem transformações ou cálculos. Normalmente, a consulta incluem parâmetros de entrada para localizar os objetos da consulta.



EXEMPLO

CE em um contexto de uma aplicação que gerencia as vendas em uma loja de roupas. Considera-se neste exemplo que é necessário informar parte da descrição de um produto e a partir do mesmo filtrar todos os produtos com descrição similar.

- Consulta de produtos por descrição.

Tabela de complexidade das funções transacionais

Após a identificação de todos os EE, SE e CE que fazem parte do escopo da aplicação, é necessário identificar a complexidade de cada transação. A complexidade é definida por meio da Tabela 1 (itens b e c).

A tabela a seguir apresenta a complexidade funcional (baixa, média ou alta) para cada função de dados (ALI ou AIE) e função transacional (EE, SE e CE), de acordo com parâmetros. A tabela também apresenta para cada tipo de função (ALI, AIE, EE, SE e CE) e complexidade funcional (baixa, média ou alta) a quantidade de pontos de função não ajustados que devem ser considerados.

Tabela 1: Complexidade funcional e pontos de função não ajustados

Complexidade funcional											
ALI e AIE (a)				EE (b)			SE e CE (c)				
#TDE				#TDE				#TDE			
#RL	1 a 19	20 a 50	51 ou +	#AL	1 a 4	5 a 15	16 ou +	#A L	1 a 5	6 a 19	20 ou +

1	Baixa	Baixa	Média
2 a 5	Baixa	Média	Alta
6 ou +	Média	Alta	Alta

0 a 1	Baixa	Baixa	Média
2	Baixa	Média	Alta
3 ou +	Média	Alta	Alta

0 a 1	Baixa	Baixa	Média
2 a 3	Baixa	Média	Alta
4 ou +	Média	Alta	Alta

Pontos de função não ajustados por tipo de função e sua complexidade (d)

		Complexidade funcional		
Tipo de função		Baixa	Média	Alta
ALI		7	10	15
AIE		5	7	10
EE		3	4	6
SE		4	5	7
CE		3	4	6

Fonte: O autor (2021).

Em que:

#AL: corresponde a uma ALI ou AIE, envolvido na transação em questão.

#TDE: corresponde a uma unidade de informação (um campo) reconhecível pelo usuário. Normalmente é um atributo em uma classe ou um campo em uma tabela.

Sobre a contagem do TDE, no caso de EE podem ser os campos de entrada das informações, nas SE podem ser os campos em um relatório e nas CE podem ser os campos usados para pesquisa e os mostrados como resposta. As mensagens de erros e os botões que podem ser pressionados também podem ser contabilizados.



EXEMPLO

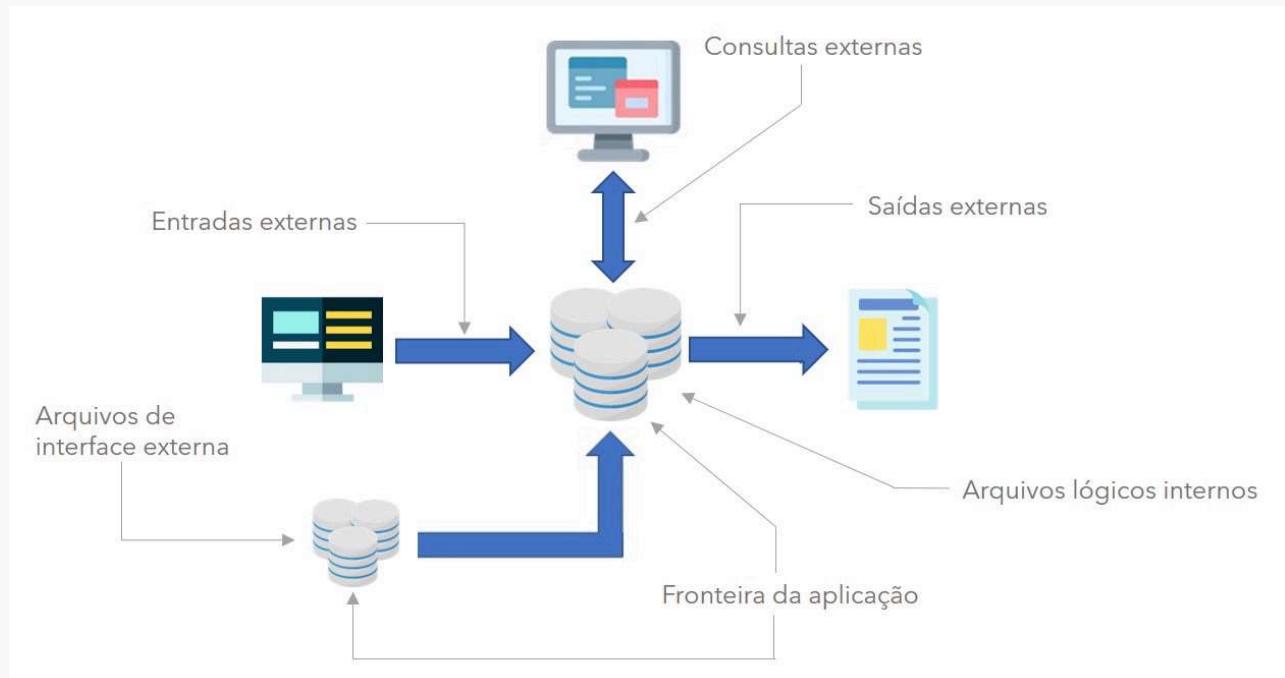
No contexto de uma aplicação que gerencia as vendas em uma

loja de roupas, a função de transação **relatório de vendas de um cliente com totalização** poderia ter as seguintes informações:

- $AL = 3$, se considerarmos que o relatório envolverá a recuperação de dados nas três funções de dados: cliente, produto e venda).
 - $\#TDE = 5$ a 15 , se considerarmos que no relatório serão impressas as seguintes informações: nome, CPF e fone (cliente); descrição (produto); data, qtd. de itens, valor unitário e valor total (venda).
 - Logo, a complexidade desta função de transação seria igual a média.

A Figura 1 a seguir ilustra os principais elementos mobilizados para a aplicação da técnica APF.

Figura 1: Elementos a serem identificados para a aplicação da APF



Para a aplicação da APF é necessário entender e identificar os elementos considerados pela técnica. Fonte: ABREU (2011).

Pontos de Função Não Ajustados (PFN)

Após identificadas todas as funções de dados (ALI e AIE), as funções transacionais (EE, SE e CE) e suas respectivas complexidades, é o momento da identificação dos pontos gerados para cada tipo de função. A Tabela 1 (item d) apresenta a respectiva pontuação.

A tabela apresenta a **complexidade funcional** (baixa, média ou alta) para cada função de dados (ALI ou AIE) e função transacional (EE, SE e CE) de acordo com parâmetros. A tabela também apresenta para cada tipo de função (ALI, AIE, EE, SE e CE) e complexidade funcional (baixa, média ou alta) a quantidade de pontos de função não ajustados que devem ser considerados.

Tabela 1: Complexidade funcional e pontos de função não ajustados

Complexidade funcional											
ALI e AIE (a)			EE (b)		SE e CE (c)						
#TDE				#TDE				#TDE			
#RL	1 a 19	20 a 50	51 ou +	#AL	1 a 4	5 a 15	16 ou +	#A L	1 a 5	6 a 19	20 ou +
1	Baixa	Baixa	Média	0 a 1	Baixa	Baixa	Média	0 a 1	Baixa	Baixa	Média
2 a 5	Baixa	Média	Alta	2	Baixa	Média	Alta	2 a 3	Baixa	Média	Alta
6 ou +	Média	Alta	Alta	3 ou +	Média	Alta	Alta	4 ou +	Média	Alta	Alta

Pontos de função não ajustados por tipo de função e sua complexidade (d)

Complexidade funcional				
Tipo de função		Baixa	Média	Alta
ALI		7	10	15
AIE		5	7	10
EE		3	4	6
SE		4	5	7

Fonte: O autor (2021).

No exemplo citado anteriormente, para a SE **relatório de vendas de um cliente com totalização**, cuja complexidade identificada foi **média**, a quantidade de pontos atribuídos a esta função seria de: **cinco pontos não ajustados**.



EXEMPLO

Sistema para gerenciar vendas em uma loja de roupas.

A seguir, vamos aplicar um exemplo consolidado a partir de um escopo definido para um sistema que gerencia vendas em uma loja de roupas.

Considere que neste exemplo, fazem parte do escopo os seguintes requisitos funcionais:

+ RF1

O sistema deve manter as informações do cliente considerando as informações: nome, data nascimento, CPF, RG, estado civil, fone e e-mail.

+ RF2

O sistema deve manter as informações dos produtos considerando as informações: código, descrição, valor e categoria.

+ RF3

O sistema deve registrar as vendas identificando cliente, data venda, produto, valor unitário do produto, qtd. itens, tipo pagamento, número cartão e dígito verificador.

+ RF4

O sistema deve validar com a operadora quando pagamento realizado com cartão de crédito. Para isso deve ser informado nome, CPF, valor venda, número cartão e dígito verificador. O retorno será o status da validação.

+ RF5

O sistema deve gerar relatório de vendas de um cliente com totalização, considerando as informações: nome, CPF e fone (cliente); descrição produto (produto); data venda, qtd. itens, valor unitário e valor total (venda).

O resultado dos pontos para cada função identificada é apresentado na Tabela 2.

Tabela 2: Exemplo de contagem em um contexto de um sistema para controle das vendas em uma loja de roupas

Função	Tipo	AL	RL	TDE	# AL	# RL	#T DE	Com pl.	P FN
Cliente	A LI		Client e	Nome, data nascimento, CPF, RG, estado civil, fone e e-mail.		1	7	Baixa	7

Produto	A LI		Produc to	Código, descrição, valor e categoria.		1	4	Baix a	7
Venda	A LI		Venda , itens da venda	Cliente, produto, valor unitário, qtd. Itens, data de venda, tipo do pagamento, número do cartão e dígito verificador.		2	8	Baix a	7
Operad ora	A I E		Opera dora	Nome, CPF, valor de venda, número do cartão, Dígito verificador e status.		1	6	Baix a	5
Incluir cliente	E E	Clien te		Nome, data nascimento, CPF, RG, estado civil, fone e e-mail.	1		7	Baix a	3
Excluir cliente	E E	Clien te		Cpf.	1		1	Baix a	3
Alterar cliente	E E	Clien te		Nome, data nascimento, CPF, RG, estado civil, fone e e-mail.	1		7	Baix a	3
Consult ar cliente	C E	Clien te		Nome, data nascimento, CPF, RG, estado civil, fone e e-mail.	1		7	Baix a	3
Incluir produto	E E	Prod uto		Código, descrição, valor e categoria.	1		4	Baix a	3
Excluir produto	E E	Prod uto		Código.	1		1	Baix a	3
Alterar produto	E E	Prod uto		Código, descrição, valor e categoria.	1		4	Baix a	3
Consult ar produto	C E	Prod uto		Código, descrição, valor e categoria.	1		4	Baix a	3
Registra rvendas	E E	Clien te, Prod uto, Vend a		Nome, produto, valor unitário, qtd. Itens, data venda, tipo pagamento, número do cartão e dígito verificador.	3		8	Alta	6

Validar operadora pagamento	SE	Cliente, venda, operador a		Nome, CPF, valor de venda, número do cartão, dígito verificador e <i>status</i> .	3	-	6	Média	5
Gerar relatório de Vendas	SE	Cliente, produto, vendas		Nome, CPF, fone, data venda, descrição do produto, valor unitário, qtd. Itens e valor de venda.	3		8	Média	5
								TOTAL	66

A tabela apresenta um exemplo completo de contagem de PFN considerando os tipos de função identificados, os seus parâmetros (AL, RL e TDE) e a complexidade. Fonte: O autor (2021).

A Tabela 2 contém todas as funções de dados (ALI e AIE) e as funções transacionais (EE, SE, CE), que são possíveis de serem identificadas a partir da lista de requisitos funcionais identificados.

Para as funções de dados é necessário identificar os Registros Lógicos (RL, subconjuntos de dados) e os TDEs.

Já para as funções transacionais é necessário identificar os Arquivos Lógicos (AL, sejam internos ou externos) envolvidos na transação e os TDEs.

Após a identificação das funções de dados e funções transacionais, bem como os AL, RL e TDE, é possível identificar a complexidade de cada função (baixa, média e alta). Por fim, com as complexidades identificadas, obtém-se a quantidade de pontos de função não ajustados. A partir da soma desta pontuação, chega-se ao total de ponto não ajustados do sistema.

Assista a videoaula desta Unidade para acompanhar o passo a passo o preenchimento da Tabela 2.

Técnica de estimativa de pontos de função e sua aplicação



Pontos de Função Ajustados (PFA)

A técnica de pontos de função sugere 14 fatores de ajuste técnico. Todos têm o mesmo peso, e a eles deve ser atribuída nota de zero a cinco, em que zero significa que o fator não tem nenhuma influência no projeto e cinco significa que o fator tem influência determinante no projeto.

As características do sistema podem influenciar na sua complexidade, variando em um intervalo de -35% a +35%, o que implica uma variação de 0,65 a 1,35. A Tabela 3 a seguir apresenta os 14 fatores e um exemplo de aplicação, em que se atribuiu o valor cinco para todos os itens, totalizando 70 pontos para os Fatores de Influência (FI).

A tabela apresenta os 14 fatores técnicos a serem considerados para ajustar a contagem de pontos de função. A tabela exemplifica uma situação em que todos os fatores foram atribuídos com valor 5, somando 70 pontos de influência técnica.

Tabela 3: Fatores técnicos e um exemplo de aplicação

Fator técnico	Ajuste
Comunicação de dados	5
Processamento distribuído	5
Performance	5

Configuração do equipamento	5
Volume de transações	5
Entrada de dados <i>on-line</i>	5
Interface com o usuário	5
Atualização <i>on-line</i>	5
Processamento complexo	5
Reusabilidade	5
Facilidade de implantação	5
Facilidade operacional	5
Múltiplos locais	5
Facilidade de mudanças (flexibilidade)	5
ΣFI	70

Fonte: O autor (2021).

Após a somatória dos fatores de influência é aplicada a fórmula para se chegar ao Fator de Complexidade Técnica (FCT).

$$FCT = 0,65 + (0,01 \times \Sigma FI).$$

No exemplo demonstrado na Tabela 2, o FCT seria igual a:

$$FCT = 0,65 + (0,01 \times 70).$$

$$FCT = 1,35.$$

Ou seja, neste caso, o ajuste dos pontos de função aumentará em 35% os pontos de PFN. Isso ocorre porque no exemplo da Tabela 3, atribuímos o valor máximo (cinco) para todos os fatores de influência.

Tendo como exemplo o caso de contagem da Tabela 2, em que o total de pontos de PFN foi de 66 pontos de função, se aplicado o FCT, o valor dos PFAs seria 89,1 pontos, em que:

$$PFA = PFN \times FCT.$$

$$PFA = 66 \times 1,35.$$

PFA = 89,1.

Até este passo, aprendemos a calcular os pontos de função e posteriormente aplicar o fator de complexidade técnica para chegar aos PFA. Na próxima seção, aprenderemos como a partir de um PFA calcular a duração e o custo de um projeto.

Duração e custo de um projeto

Uma vez calculado o PFA, basta multiplicá-lo pelo IP da equipe, que depende do contexto do projeto e da equipe envolvida e deve ser coletado a partir de bases históricas. Por exemplo, em um projeto em que quatro pessoas trabalharam em tempo integral em uma semana de 40 horas, o valor do esforço total da equipe é 160 horas semanais. Se naquela semana foram desenvolvidas funcionalidades referentes a 15 pontos de função, a equipe teve uma produtividade de 10,7 horas ($160/15$) por pontos de função. Se houver mais projetos, pode-se somar os APF do projeto e dividir pelo esforço despendido.

A coleta e a análise de uma base histórica seria o ideal para converter os APF em horas de trabalho e consequentemente o custo de desenvolvimento. Como nem sempre as empresas mantêm bases históricas de seus projetos, há algumas referências de produtividade, a partir das quais os projetos podem converter seu APF em tempo e custo.

A Tabela 4 a seguir apresenta uma destas referências para as linguagens mais populares em 2018.

Tabela 4: Índices de produtividade por linguagem de programação

Linguagem	Índice de produtividade em horas por pontos de função		
	Baixa	Média	Alta
Java	14	10	6
C	24	18	12
C++	18	12	6
Python	18	14	8
Visual Basic	12	8	6
C#	17	12	7

PHP	15	10	5
JavaScript	16	12	8

A tabela apresenta valores de referência para índices de produtividade, de acordo com a linguagem de programação utilizada no projeto. Fonte: O autor (2021).

Exemplo de cálculo do tempo considerando a Tabela 4 como referência.

Considere um projeto em que foram contados **10.000 pontos de função**, trata-se de uma aplicação a ser desenvolvida em Java e espera-se uma alta produtividade da equipe. Segundo a Tabela 4, neste caso, 1 ponto de função equivale a seis horas de trabalho. Logo, seriam necessárias **60.000 horas de projeto**.

Em média, um mês corresponde a 160 horas de trabalho, logo, seriam necessários **375 desenvolvedores** para realizar o trabalho em **um mês**.

Mas se, por exemplo, o projeto contar com 20 desenvolvedores ($160 \times 20 = 3.200$ horas mensais), logo são necessários $(60.000 / 3.200) = 18,75$ meses para a entrega do projeto.

A variação do tempo dependerá da quantidade de pessoas atuando no projeto ou o tempo necessário para a entrega. No exemplo citado anteriormente, se houvesse a necessidade de entregar o projeto em 12 meses, o cálculo para a quantidade de desenvolvedores poderia ser este:

$$60.000 / (12 \text{ meses} * 160 \text{ horas mensais}).$$

$$60.000 / 1.920 = 31,25.$$

Ou seja, seriam necessárias quase 32 pessoas dedicadas ao projeto para que fosse entregue em 12 meses.

Por fim, o custo médio do projeto pode ser obtido por meio da multiplicação do total de horas do projeto pelo custo médio da hora do desenvolvedor.

Supondo, no caso do exemplo do projeto a ser desenvolvido em Java, que o custo médio da hora de um desenvolvedor é **R\$ 50,00** e o total de horas do projeto é **60.000 horas**, o custo total do projeto será de aproximadamente **R\$ 3.000.000,00**.

É importante ressaltar que estas estimativas devem ser testadas e ajustadas ao longo do tempo. Quanto maior a base histórica dos projetos desenvolvidos pela equipe, maior a assertividade dos índices de produtividade.

| Conclusão

Nesta Unidade, aprendemos detalhadamente como aplicar a técnica de estimativas que utiliza como base pontos por função. Entendemos todos os elementos e os parâmetros que devem ser considerados para identificar a quantidade de pontos. Também aprendemos que esta primeira contagem não considera fatores técnicos e, portanto, em um segundo momento, estes fatores devem ser considerados para realizar o ajuste do total de pontos de função. Ao total, são 14 fatores técnicos, cujos valores atribuídos podem variar de 0 a 70, o que pode fazer com que o total de pontos finais variem de 0,65 a 1,35.

Por fim, entendemos que para calcular o tempo e o custo de desenvolvimento, deve-se multiplicar o total de pontos de função pela produtividade em horas da equipe e o custo médio da hora do desenvolvedor. A forma mais efetiva de obter a produtividade de uma equipe é manter registros históricos dos projetos desenvolvidos pela equipe. Caso a empresa não tenha esta base histórica, é possível utilizar referências que consideram a produtividade de acordo com o nível esperado de produtividade da equipe e a linguagem de programação utilizada no projeto.

Também é importante ressaltar que a contagem de pontos pode ser impactada pelo nível de detalhamento dos requisitos. No início do projeto, os requisitos podem não estar muito claros e muito menos as informações envolvidas. À medida que o projeto avança e os requisitos se tornam mais claros, é possível que a contagem precise ser ajustada.

| Referências bibliográficas

ABREU, A. Análise de pontos de função na prática. **Devmedia**, 2011. Disponível em: <https://www.devmedia.com.br/analise-de-pontos-de-funcao-na-pratica/22792>. Acesso em: 15 fev. 2021.

IFPUG. **IFPUG – International Function Point Group**, 2021. Página inicial. Disponível em: <https://www.ifpug.org/?lang=pt>. Acesso em: 15 fev. 2021.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software: uma abordagem profissional.** 8. ed. Porto Alegre: AMGH, 2016. Disponível em:
<https://integrada.minhabiblioteca.com.br/#/books/9788580555349/>. Acesso em: 15 fev. 2021.

WAZLAWICK, R. S. **Engenharia de software: conceitos e práticas.** 2. ed. Rio de Janeiro: Elsevier, 2019.



© PUCPR - Todos os direitos reservados.