

Nesta seção você encontra artigos voltados para testes, processo, modelos, documentação, entre outros

## Estimativas de tamanho em projetos de software utilizando pontos de função

Um tutorial voltado para micro e pequenas empresas



### Jhoney da Silva Lopes

[jhoney.lopes@gmail.com](mailto:jhoney.lopes@gmail.com)

Graduando em Ciência da Computação pela Universidade Federal de Viçosa (UFV). Atualmente é Diretor Presidente da empresa júnior No Bugs ([www.nobugs.com.br](http://www.nobugs.com.br)), Assessor da presidência da Skep Design ([www.skepdesign.com](http://www.skepdesign.com)) e Diretor Vice-Presidente de O Primeiro Plano. Áreas de interesse: Engenharia de Software, Empreendedorismo, Gerenciamento de Projetos e Business Intelligence.



### José Luis Braga

[zeluis@dpi.ufv.br](mailto:zeluis@dpi.ufv.br)

Pós-doutoramento em Tecnologias da Informação na University of Florida. Doutor em Informática pelo Departamento de Informática da PUC-Rio. Mestre em Ciências da Computação pelo Departamento de Ciência da Computação da UFMG. Atualmente é Professor Titular do Departamento de Informática do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa-MG. Atua na área de Ciência da Computação, com ênfase em Engenharia de Software e Sistemas de Informação. Áreas de Interesse: Qualidade de Software com Foco em Processos, Engenharia de Software Experimental, Engenharia de Software Apoiada por Ontologias, Engenharia de Software Baseada em Agentes, Sistemas de Apoio à Decisão.

### De que se trata o artigo?

Estimativa de software com derivações em custo, prazo e esforço é uma necessidade comum entre as empresas de TI. Nesse contexto, o objetivo deste artigo é apresentar de forma simples, por meio de exemplos, o uso de um método poderoso para a solução destes problemas recorrentes, a APF (Análise de Ponto de Função).

Análise de Ponto de Função é um método de medição do tamanho funcional de um software, com base em operações extraídas dos requisitos funcionais. A partir dessa medição inicial de tamanho, derivam-se outras medidas como, por exemplo, o tempo necessário para desenvolvimento, e uma estimativa inicial de custos. APF tem por definição medir o que o software deve fazer, e não como ele deve ser construído, portanto o processo de medição é fundamentado em uma avaliação padronizada dos requisitos lógicos do usuário.

### Em que situação o tema é útil?

Na fase inicial de um projeto, existe a necessidade de obter o custo, prazo e o esforço de implementação para estabelecimento de contratos de desenvolvimento. Análise de Ponto de Função é um método de medição do tamanho funcional de software que permite fazer essas avaliações com base em informações disponíveis nas fases iniciais dos projetos. O uso do método profissionaliza a avaliação inicial de tamanho, permitindo repetição do processo e aumento do nível de maturidade no gerenciamento de projetos.

### Resumo

Na fase inicial de um projeto, a necessidade em obter o custo, prazo e o esforço é observado em todas as empresas, pois as mesmas precisam gerar um orçamento para os seus clientes e avaliar uma série de projeções. Este artigo organiza de forma simples e introdutória conhecimentos sobre a Análise de Ponto de Função.

Inicialmente não se tem conhecimento de todas as características do produto bem como da sua real dimensão, por esse motivo é necessário utilizar técnicas de extração de requisitos para realizar um levantamento inicial dos mesmos e compreender melhor o problema. Os requisitos são condições, características ou capacidades necessárias para atingir uma finalidade, categorizados na implementação de sistemas em funcionais e não funcionais como forma de estabelecer critérios de qualidade.

Uma vez definidos os requisitos, pode-se então avaliar o tamanho do sistema em termos de suas funcionalidades. Existem vários métodos de estimativa, a APF (Análise de Ponto de Função) é uma delas. Esse método não tem como objetivo realizar estimativas de prazo, custo e esforço para implementação de sistema, mas sim avaliar o tamanho de um sistema em termos de suas funcionalidades. Resulta desta avaliação o tamanho funcional do sistema expresso em Pontos de Função (unidade de medida do método). A partir do tamanho funcional, podem ser derivadas estimativas adicionais, como tempo e custo.

Este artigo apresentará uma visão geral sobre todos os passos necessários para utilização da análise de ponto de função, para a realização de estimativas na fase inicial de um projeto de desenvolvimento, proporcionando ao leitor uma visão restrita do método, mas suficiente para estimar um projeto em sua fase inicial e, com isso, realizar derivações de acordo com suas necessidades.

A análise por pontos de função surgiu em 1979 na IBM (*International Business Machines*) e foi desenvolvida por Alan Albrecht. A aceitação foi muito grande, e uma quantidade cada vez maior de pessoas começaram a utilizar o método, resultando na criação do IFPUG (*International Function Point Users Group*). Hoje o método é mantido e evoluído pelo IFPUG e fundamenta-se em seis passos:

1. Determinar o tipo de contagem;
2. Identificar o escopo da contagem e a fronteira da aplicação;
3. Contar funções:
  - a. Tipo dados;
  - b. Tipo transação;
4. Determinar a contagem de pontos de função não ajustados;
5. Determinar o valor do fator de ajuste;
6. Calcular o número dos pontos de função ajustados.

Como foi dito, a APF é um método que visa medir o tamanho funcional de um software do ponto de vista do usuário e possui como unidade de medida o ponto de função (PF) (ler **Nota do DevMan 1**).



## Nota do DevMan 1

**Usuário:** Em Análise de Ponto de Função, usuário possui um conceito mais amplo: qualquer entidade que se relacione com o sistema ou que produza um ônus ao mesmo. Ex: Pessoa, aplicação, leis, restrições, etc.

É importante perceber que para a APF o usuário não é somente a pessoa que interage com o sistema.

Possui como base os seguintes objetivos:

- Medir as funcionalidades (ler **Nota do DevMan 2**) implementadas no sistema;
- Medir esforço de implementação e de manutenção do software, independente de tecnologia, ou seja, a APF leva em consideração o que o software faz e não como ele é construído;
- Simplicidade, o trabalho para aplicar a APF deve ser o mínimo possível, ele não deve ser um ônus no desenvolvimento do software.

É importante ter em mente, durante todo o processo da utilização do método de análise de pontos de função, que o usuário é quem define o que faz parte ou não do projeto, diferentemente de alguns métodos que levam em consideração

a visão do desenvolvedor, ou seja, uma visão técnica. Na APF o que importa é a visão do negócio e quem possui essa visão é o usuário, lembrando que usuário na APF possui um conceito amplo e não é somente a pessoa que irá interagir com o software final.



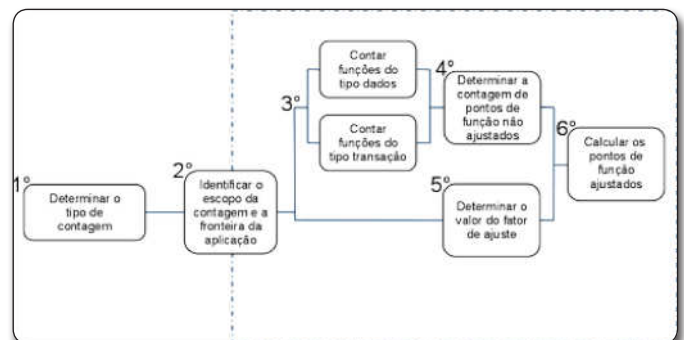
## Nota do DevMan 2

**Funcionalidade:** Funcionalidade é o conjunto de tarefas fornecidas pelo sistema para poder realizar transação, processamento e armazenamento dos dados dos usuários.

### Determinar o tipo de contagem

O primeiro passo para a contagem será determinar qual o tipo de contagem a ser realizada, como pode ser visto na **Figura 1**. Na APF existem três tipos de contagem:

- Projeto de desenvolvimento;
- Projeto de melhoria;
- Aplicação.



**Figura 1.** Passos para a contagem dos pontos de função (VAZQUEZ, 2009)

Este artigo tem por objetivo apresentar a solução de contagem para projeto de desenvolvimento, mas serão apresentadas as diferenças entre elas.

#### Projeto de desenvolvimento

É caracterizado como projeto de desenvolvimento, um novo projeto desde a fase de extração de requisitos (ler **Nota do DevMan 3**) até a instalação do mesmo. Neste tipo de projeto são contadas na APF todas as funcionalidades fornecidas aos usuários até a instalação do sistema, ou seja, funcionalidades de conversão (ler **Nota do DevMan 4**) também são contadas. Só se pode saber todos os requisitos de um sistema após o término do projeto, sendo assim, toda a contagem de um projeto de desenvolvimento pode ser entendida como estimativa e não medição.

#### Projeto de melhoria

O projeto de melhoria mede todas as funcionalidades novas, modificadas e excluídas de um determinado sistema. Ao término de um projeto de melhoria a aplicação deverá ser contada com o intuito de atualizar o valor em pontos de função da mesma.



### Nota do DevMan 3

**Requisitos:** São as necessidades e características que o sistema deve ter para atingir as expectativas do cliente. A extração dos requisitos consiste em uma parte crítica na elaboração de uma proposta, ela é parte determinante do sucesso ou fracasso de um projeto.



### Nota do DevMan 4

**Funções de Conversão:** Para um entendimento considere o exemplo: um sistema A possui uma lista de funcionários cadastrados, o sistema B sendo contado deverá incluir todos esses funcionários em sua base de dados, essa funcionalidade será disparada uma única vez que é durante a instalação do sistema, sendo caracterizada como função de conversão.

#### Aplicação

Entende-se por contagem do tipo aplicação um software instalado, ou seja, a contagem após o término de um projeto de desenvolvimento. Neste caso, não se leva em consideração as funções do tipo conversão.

### Aplicando o conhecimento – Determinar o tipo de contagem

Neste artigo será realizada a contagem de um projeto simples, extraído da base de projetos da No Bugs - Empresa Júnior do Bacharelado em Ciência da Computação da Universidade Federal de Viçosa.

O foco deste artigo são as derivações dos pontos de função para auxiliar na elaboração da proposta do projeto para o cliente. A sua contagem será de um projeto de desenvolvimento, um sistema simples de locação de veículos.

### Identificar o escopo da contagem

O segundo passo para a contagem é identificar o escopo da contagem e a fronteira da aplicação, como pode ser visto na **Figura 1**. Muitas vezes a identificação do escopo e da fronteira da aplicação não são levados tão a sério, principalmente por empresas que não utilizam gerência de projetos no gerenciamento do desenvolvimento de software.

Esta é uma etapa crucial para o andamento do projeto, a definição de um escopo (ler **Nota do DevMan 5**) errado pode acarretar em prejuízos para o projeto ou até a perda total dele. O escopo define quais funções serão incluídas na contagem, ele pode abranger todas as funcionalidades, apenas as funções utilizadas ou funções específicas.

A fronteira da aplicação é a linha que separa uma aplicação de outra. Dentro de um escopo de contagem pode existir mais de uma aplicação a ser contada, por isso é importante definir qual é a sua fronteira.

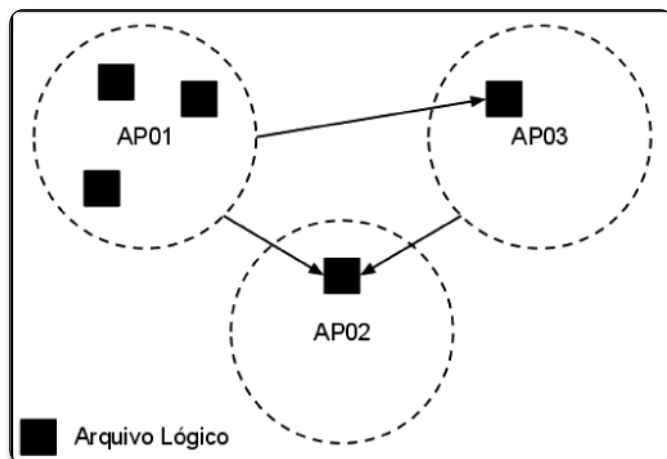


### Nota do DevMan 5

**Escopo do Projeto:** Escopo do projeto é o trabalho que precisa ser realizado para entregar um produto, serviço ou resultado com as características e funções especificadas (PMBOK, 2004), o escopo da contagem é tudo aquilo que deve ser contado.

Uma sugestão simples para não errar nesta etapa é seguir a regra do IFPUG que é determinar a fronteira da aplicação baseado no *Ponto de Vista do Usuário*. O usuário define o que ele entende sobre as atribuições do sistema e de cada aplicação.

Considera-se o exemplo apresentado na **Figura 2**, que mostra três aplicações, AP01, AP02 e AP03, separadas por fronteiras (círculos tracejados) e cada uma dessas aplicações interagem umas com as outras, esta interação é feita a partir do que é chamado de arquivos lógicos, que aparecem na figura como quadrados preenchidos de preto.



**Figura 2.** Arquivos lógicos e fronteiras das aplicações

### Aplicando o conhecimento – Identificar o escopo da contagem

Como foi visto, nesta etapa devemos definir o escopo da contagem e a fronteira da aplicação. No exemplo deste artigo, trata-se de software destinado a uma empresa que realiza locação de automóveis, o sistema é simples e composto por uma única aplicação.

### Contar funções do tipo dados

O terceiro passo para a contagem é dividido em duas partes, contar funções do tipo dados e contar funções do tipo transação, como pode ser visto na **Figura 1**. Funções do tipo dados é o nome designado às funcionalidades fornecidas para o armazenamento de dados na aplicação sendo contada (ler **Nota do DevMan 6**), são caracterizados como arquivos lógicos e eles podem ser mantidos pela aplicação ou lida de outra, como pode ser visto na **Figura 2**.

Arquivos lógicos que estão dentro da fronteira da aplicação e mantidos pela mesma são chamados de Arquivos Lógicos



Internos (ALI), já os arquivos lógicos lidos de outra aplicação são chamados de Arquivos de Interface Externa (AIE).



## Nota do DevMan 6

**Aplicação Contada:** Como visto, um projeto pode ser composto por várias aplicações internas que interagem umas com as outras, conforme a Figura 2. Quando um projeto possui mais de uma aplicação, diz-se da aplicação sendo contada a que no momento está sendo analisada, já projetos que possuem apenas uma aplicação, o termo aplicação sendo contada refere-se a todo o projeto.

### Arquivo Lógico Interno

Grupo lógico de dados persistentes e que são mantidos dentro da fronteira da aplicação e alterados por meio de processos elementares. Um processo elementar é a menor unidade de atividade significativa para o usuário final (VAZQUEZ,2009). É a menor funcionalidade disponibilizada ao usuário.

Considerando a **Figura 2**, a AP01 possui três arquivos lógicos internos (ALI). À primeira vista, parecerá que cada tabela do banco de dados da aplicação será um ALI. Essa é uma maneira simples de entender o que seria um arquivo lógico, mas serão mostrados alguns exemplos de que é um erro pensar dessa forma, pois um grupo de tabelas pode ser considerado como um único arquivo lógico, dependendo da forma como é utilizado pela aplicação.

Exemplos de ALI:

- Arquivo de configuração, conexão, segurança (senhas) mantidos pela aplicação;
- Tabelas ou grupos de tabelas do banco de dados mantidas pela aplicação.

Não são exemplos:

- Arquivos temporários ou de backup;
- Tabelas temporárias ou *views*.

### Arquivo de Interface Externa

Grupo lógico de dados persistentes mantidos dentro da fronteira de outra aplicação, mas requerido ou referenciado pela aplicação que está sendo contada, ou seja, a aplicação sendo contada acessa os dados presentes neste arquivo lógico, mas o mesmo não está dentro da aplicação que está sendo contada e sim em outra. Nesse caso denomina-se esse arquivo lógico de arquivo de interface externa (AIE).

Considerando a **Figura 2**, a AP01 referencia arquivos lógicos da AP02 e AP03, estes são os arquivos denominados de AIE.

Exemplos de AIE:

- Dados de segurança armazenados em arquivos lógicos e mantidos por aplicações específicas a este fim;
- Dados salariais armazenados na aplicação financeira, mas utilizados pela aplicação contada.

Não é exemplo:

- Dados armazenados na aplicação sendo contada e utilizados por uma aplicação externa. Nesse caso a sua aplicação possui um ALI e outra aplicação reconhece esses dados vindos de um AIE.

### Determinação da complexidade e da contribuição

Complexidade é o grau de influência que um arquivo lógico tem para o tamanho funcional do sistema. A contribuição é a conversão do grau de complexidade em pontos de função. A complexidade é calculada a partir da contagem dos tipos de dados e dos tipos de registro.

### Tipos de dados (TD)

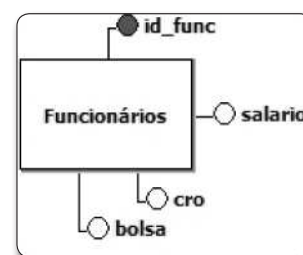
É um campo não recursivo de dado, único e reconhecido pelo usuário, em uma visão geral e limitada. Por exemplo, poderiam ser os atributos de uma tabela.

### Tipos de Registro (TR)

É caracterizado por ser um subgrupo de dados.

Em uma análise míope, quando um agrupamento de tabelas é reconhecido pelo usuário como um único arquivo lógico, ALI ou AIE, a tabela reconhecida pelo usuário é contada e as demais se tornam simplesmente tipos de registro, ou seja, essas tabelas extras não pertencem à visão do negócio, pois o usuário não as reconhece. Mas seus atributos são reconhecidos e, por este motivo, devem gerar um impacto na contagem, sendo denominado subgrupo de dados. Esses campos pertencentes à visão do negócio e reconhecidos pelo usuário são atribuídos aos demais arquivos lógicos que estão relacionados com esses tipos de registro.

Considera-se como exemplo uma especialização, onde a **Figura 3** representa essa especialização na visão do usuário, com os seguintes campos: *id\_func* (código de identificação do funcionário), *salário*, *cro* (número do registro para dentistas), *bolsa* (bonificação no salário dos auxiliares).



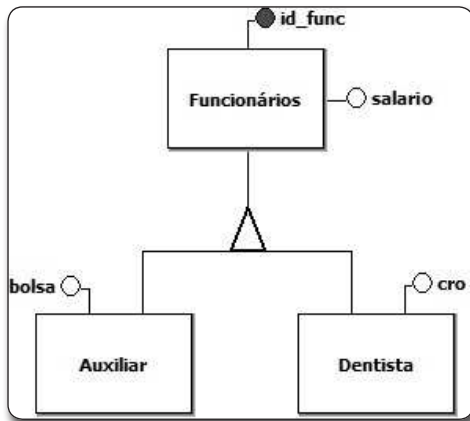
**Figura 3.** Especialização na visão do usuário

Na modelagem, foram separados os diferentes tipos de funcionários, como pode ser visto na **Figura 4**.

Neste exemplo contam-se os funcionários como uma ALI ou AIE, pois como foi apresentado, o usuário não reconhece funcionários como entidades diferentes, mas para a modelagem o desenvolvedor optou por separar estes funcionários devido aos atributos que os especializam. Como estas entidades criadas na visão do desenvolvedor possuem atributos reconhecidos pelo

usuário, esse grupo de tabelas se constitui em uma ALI ou AIE com três tipos de registro e esses atributos reconhecidos devem ser contados como atributo de funcionários. São contados três tipos de registro, pois todo arquivo lógico é um tipo de registro dele mesmo e os atributos são somados a funcionários, pois somente ele é reconhecido pelo usuário.

É importante perceber que esta solução é adotada uma vez que o usuário enxerga Auxiliar e Dentista como Funcionário, e não como entidades separadas, como pode ser visto na **Figura 5**. É imprescindível entender que um mesmo problema pode ter uma contagem diferente com visões de negócios diferentes, ou seja, o importante é o ponto de vista do usuário para a APF.



**Figura 4.** Especialização é um tipo de registro



**Figura 5.** Visão do Negócio

### Tabela de complexidade

A tabela de complexidade é padronizada pelo IFPUG, todos os usuários do método de análise de pontos de função utilizam os mesmos valores.

Após o entendimento sobre tipo de dados e tipo de registro, os mesmos serão utilizados para definir a complexidade do arquivo lógico. Realiza-se a contagem dos tipos de registro e tipos de dados de cada arquivo lógico, depois se verifica na **Tabela 1** o valor de cada um e o intervalo que pertence. Com isso, define-se a ALI ou AIE como sendo de complexidade baixa, média ou alta.

Tipos de Registro	Tipos de Dados		
	< 20	20 – 50	> 50
1	Baixa	Baixa	Média
2 – 5	Baixa	Média	Alta
> 5	Média	Alta	Alta

**Tabela 1.** Complexidade ALI e AIE

### Tabela de contribuição

A tabela de contribuição é padronizada pelo IFPUG, e todos os usuários do método de análise de pontos de função utilizam os mesmos valores.

Após identificar a complexidade de cada ALI e AIE do sistema, é possível determinar a contribuição desses para a contagem dos pontos de função.

Após verificar na **Tabela 1** a complexidade do ALI ou AIE, a tarefa para estabelecer a contribuição é muito simples, basta saber a complexidade do seu arquivo lógico e se o mesmo é um ALI ou AIE e verificar o valor na **Tabela 2**.

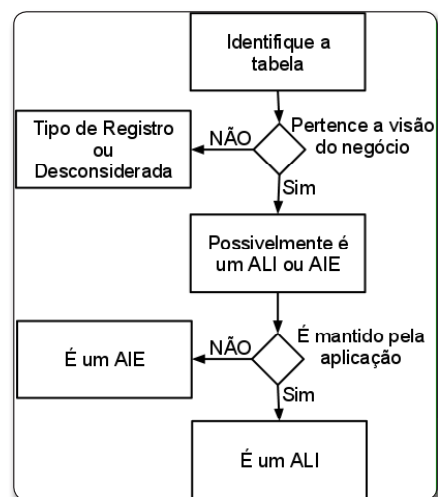
Tipo de Função	Baixa	Média	Alta
Arquivo Lógico Interno	7 PF	10 PF	15 PF
Arquivo de Interface Externa	5 PF	7 PF	10 PF

**Tabela 2.** Tabela de contribuição

### Aplicando o conhecimento – Contar Funções do tipo dados

Para facilitar a identificação dos tipos de arquivos, deve-se elaborar um modelo lógico. Uma dica geral e objetiva é contar um arquivo lógico ALI ou AIE para cada tabela reconhecida pelo usuário, ou seja, se a tabela existe no ponto de vista do usuário ela deve ser contada, caso contrário não deve ser contada. Se o usuário não reconhece a tabela, mas reconhece os tipos de dados presentes na mesma, provavelmente essa tabela será um tipo de registro.

A **Figura 6** apresenta um esquema para classificar um arquivo lógico.



**Figura 6.** Fluxo para classificação do tipo lógico

### Passos para uma estimativa da contagem desta etapa

a) Elabora-se um modelo lógico seu projeto, como exemplificado na **Figura 7**.

Identificam-se todas as tabelas reconhecidas pelo usuário, ou seja, as que fazem parte da visão do negócio, classificando-as como ALI ou AIE (vide **Tabela 3**).

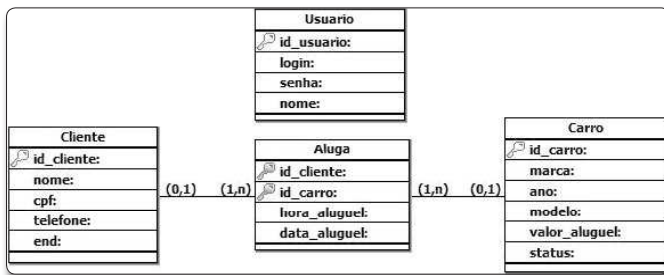


Figura 7. Modelo lógico

Descrição	Tipo
Usuário	ALI
Cliente	ALI
Carro	ALI

Tabela 3. Classificação dos arquivos lógicos

Na Figura 7 as tabelas Usuário, Cliente e Carro foram caracterizadas como arquivo lógico interno e Aluga foi definida como tipo de registro de Cliente e Carro, pois a mesma não é reconhecida pelo usuário, mas os seus atributos são.

b) Faz-se uma análise de todas as tabelas que não estão na visão do negócio:

- Se a tabela não pertence à visão do negócio, mas os seus tipos de dados pertencem, deve-se contá-la como um tipo de registro para cada arquivo lógico relacionado a ela, e atribuir os seus tipos de dados a cada um deles;
- Se nem a tabela nem os seus tipos de dados pertencem à visão do negócio, deve ser descartada da contagem.

A tabela Aluga foi considerada um tipo de registro, pois na visão do negócio, os campos hora\_aluguel e data\_aluguel são reconhecidos pelo usuário e por este motivo eles foram somados aos tipos de dados de Cliente e Carro, conforme a Tabela 4.

Descrição	Tipo	Qtd. TDs	Qtd. TRs
Usuário	ALI	4	1
Cliente	ALI	7	2
Carro	ALI	8	2

Tabela 4. Tipo de Dado (TD) e Tipo de Registro (TR)

Quando se tem uma relação entre mais de um arquivo lógico com uma entidade definida como tipo de registro, devem-se identificar todos os atributos reconhecidos pelo usuário presentes nesta entidade tipo de registro e somá-los a todos os arquivos lógicos que se relacionam com esta entidade. Isso é necessário, pois estes atributos influenciam aos demais arquivos lógicos e geram impacto no desenvolvimento do projeto.

- c) Determinação da complexidade de cada arquivo lógico.
- Para definir a complexidade basta analisar a quantidade de tipos de dados mais as quantidades de tipos de registro e conferir na Tabela 1. O resultado é apresentado na Tabela 5.

Descrição	Tipo	Qtd. TDs	Qtd. TRs	Complexidade
Usuário	ALI	4	1	Baixa
Cliente	ALI	7	2	Baixa
Carro	ALI	8	2	Baixa

Tabela 5. Complexidade

d) Determinação da contribuição de cada arquivo lógico e realização a soma de todos.

Para determinar a contribuição basta verificar na Tabela 2 o ponto de função referente a cada complexidade, o resultado é apresentado na Tabela 6.

Descrição	Tipo	Qtd. TDs	Qtd. TRs	Complexidade	Contribuição
Usuário	ALI	4	1	Baixa	7
Cliente	ALI	7	2	Baixa	7
Carro	ALI	8	2	Baixa	7
Total de Pontos de Função =					21

Tabela 6. Contagem das funções do tipo dados

### Contar funções do tipo transação

O terceiro passo para a contagem é dividido em duas partes, contar funções do tipo dados e contar funções do tipo transação, como pode ser visto na Figura 1. Agora que foi aprendido como contar funções do tipo dados, pode-se dar continuidade à contagem da aplicação. As funções do tipo transação são as funcionalidades base para o funcionamento do sistema, essas funções são chamadas de processos elementares e são classificadas em Entradas Externas, Saídas Externas e Consultas Externas.

Um processo elementar é a menor unidade de uma função disponível ao usuário. Por exemplo, consultar clientes pode ser entendido como uma função, mas o mesmo não pode ser entendido como um processo elementar, uma vez que podem ser realizadas inúmeras consultas diferentes aos clientes: consultar clientes pelo nome, consultar clientes em débito, consultar registro de clientes, e outras mais. Pode-se perceber que cada consulta é uma funcionalidade única e independente. Desta forma, para determinar um processo elementar é necessário identificar todas as funcionalidades únicas e independentes de uma função.

Uma observação importante é que um processo elementar deve ser único. Por exemplo, consultas que diferem umas das outras pela organização dos dados gerados, não podem ser consideradas diferentes.

#### Entrada Externa (EE)

Uma entrada externa é um processo de controle, ela também realiza o processamento de dados do sistema e direciona o mesmo para atender os requisitos da aplicação. A principal intenção da EE é realizar operações que visam manter (incluir, alterar ou excluir) dados de um ou mais arquivos lógicos internos, realizando processamento, transação e armazenamento de dados dos usuários do software.

Uma EE deve ser única e não necessitar de ações anteriores e nem sucessores. Por exemplo, no envio de um e-mail, escrever o destinatário, o assunto e o corpo do e-mail não podem ser entendidos como EE, pois não se constituem uma operação completa, mas o conjunto de todas estas, e clicar no botão de envio é que se caracteriza como uma EE.

Exemplos de Entrada Externa:

- Transações destinadas a manter ALI, incluir funcionário, excluir funcionário, alterar funcionário e etc.;
- Operações;
- Processos destinados a realizar registros, por exemplo, em um sistema de ponto eletrônico o usuário registra entrada e saída.

Não são exemplos de Entrada Externa:

- Telas de filtro;
- Preenchimento de campos de dados;
- Telas de login;
- Gerar relatórios.

De modo geral, EEs possuem nomes característicos, como incluir, alterar, salvar, excluir, editar, encaminhar, submeter e registrar, dentre outros.

### Saída Externa (SE)

Processo elementar destinado à apresentação de informação ao usuário ou a outra aplicação externa que utilize cálculos para processar essas informações.

A principal intenção de uma SE é apresentar informação para o usuário a partir de lógica de processamento, ou seja, as informações apresentadas devem passar por um processamento, elas não devem ser simplesmente uma consulta simples. Este processamento pode ter como objetivo manter ALIs ou alterar o comportamento do sistema.

Exemplos de Saída Externa:

- Tela para liberar acesso (tela de *login*), sendo os dados da transação criptografados;
- Relatórios financeiros, supondo esses gerados por cálculos;
- Consultas complexas com processamento de dados a partir de cálculos;
- Apresentação de gráficos com dados processados a partir de cálculos.

Não são exemplos de Saída Externa:

- Telas de filtro;
- Consultas simples, sem processamento de dados utilizando cálculos.

### Consulta Externa (CE)

Processo elementar que apresenta informação ao usuário ou a outra aplicação externa por meio de recuperação simples. Uma CE tem como principal intenção apresentar informações ao usuário por meio de uma simples recuperação de dados

de uma ALI e/ou AIE. Esse processamento não deve possuir lógica matemática, nem qualquer tipo de cálculo, não deve gerar dados derivados e nem alterar o comportamento do sistema. A CE é uma consulta simples e direta, com o retorno da informação requisitada pelo usuário.

Exemplos de Consulta Externa:

- Consultar clientes pelo nome;
- Apresentar dados em formato gráfico a partir de recuperação simples.

Não são exemplos de Consulta Externa:

- Relatórios financeiros, gerados a partir de cálculos;
- Telas de filtro.

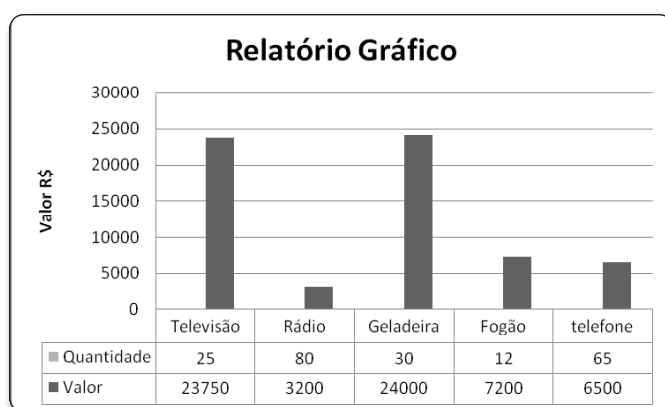
### Determinação da complexidade e da contribuição

Complexidade é o grau de influência que um processo elementar tem para o tamanho funcional do sistema. A contribuição é a conversão do grau de complexidade em pontos de função.

Essa complexidade é calculada a partir da contagem dos tipos de dados e dos arquivos referenciados.

### Tipos de dados (TD)

É um campo não recursivo de dado, único e reconhecido pelo usuário, ou seja, é cada campo preenchido ou apresentado ao usuário. Por exemplo, em um formulário os campos nome, CPF, endereço, o botão de confirmação, uma janela de mensagem de erro, dentre outros, são tipos de dados. Já em um relatório, o código do produto, o nome, a descrição, o valor, são tipos de dados. Em um gráfico, o raciocínio é o mesmo, conta-se um tipo de dado para o nome do produto, um para a quantidade e um para o valor, no total tem-se três tipos de dados neste relatório, como pode ser visto na **Figura 8**.



**Figura 8.** Apresentação de relatório gráfico

### Arquivo Referenciado (AR)

Um arquivo referenciado é todo arquivo lógico lido, pode ser um ALI ou AIE, ou todo arquivo lógico mantido, nesse caso só pode ser um ALI. Um tipo de registro não é um arquivo lógico, ele pertence a um arquivo lógico. Não se deve contar tipos de registro e arquivos lógicos lidos várias vezes. Esses são contados apenas uma única vez.



**Tabela de complexidade**

A tabela de complexidade é padronizada pelo IFPUG, todos os usuários do método de análise de pontos de função utilizam os mesmos valores.

Após o entendimento sobre tipo de dado e arquivo referenciado, pode-se utilizá-lo para definir a complexidade do processo elementar. Realiza-se a contagem dos tipos de dados e dos arquivos referenciados de cada processo elementar, depois verifica-se na **Tabela 7** ou na **Tabela 8** o valor de cada um e o intervalo a que ele pertence, com isso definindo a EE, SE ou CE como sendo de complexidade baixa, média ou alta.

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

**Tabela 7.** Complexidade Entrada Externa (EE)

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

**Tabela 8.** Complexidade Saída Externa (SE) e Consulta Externa (CE)

É importante perceber que a **Tabela 7** é destinada a EE e a **Tabela 8** a SE e CE. Essa diferença é importante uma vez que processos elementares destinados a apresentar informação, possuem normalmente mais tipos de dados e referenciam mais arquivos.

**Tabela de contribuição**

A tabela de contribuição é padronizada pelo IFPUG, todos os usuários do método de análise de pontos de função utilizam os mesmos valores.

Após identificar a complexidade de cada processo elementar do sistema, é possível determinar a contribuição desses para a contagem dos pontos de função.

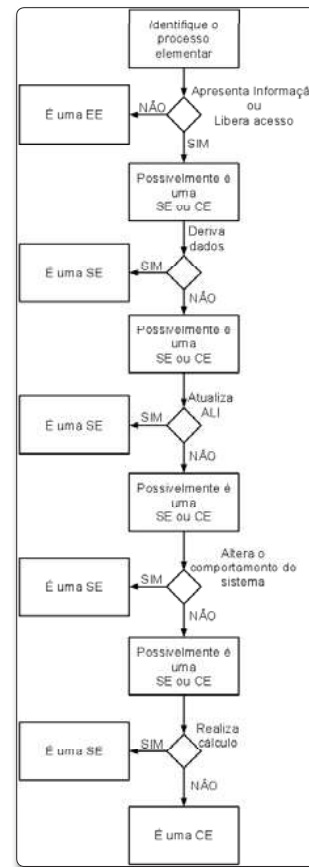
Após a definição da complexidade do seu processo elementar, definir a sua contribuição é muito simples, basta saber a complexidade do seu processo elementar e se o mesmo é uma EE, SE ou CE e utilizar o valor correspondente na **Tabela 9**.

Tipo de Função	Baixa	Média	Alta
Entrada Externa	3 PF	4 PF	6 PF
Saída Externa	4 PF	5 PF	7 PF
Consulta Externa	3 PF	4 PF	6 PF

**Tabela 9.** Tabela de Contribuição

**Aplicando o conhecimento – Contar funções do tipo transação**

Para finalizar o terceiro passo, deve-se determinar a contagem das funções do tipo transação.



**Figura 9.** Fluxo para classificação do processo elementar

O fluxo da **Figura 9** foi idealizado para facilitar o entendimento do processo de determinação dos tipos de processo elementar.

Esta é uma visão geral, o importante é saber qual a finalidade do seu processo elementar. Por exemplo, um cadastro é uma EE que pode apresentar informações ao final do processamento que não o torna uma CE ou SE, pois sua finalidade poderia ser cadastrar.

Outra dica, quando não se reconhece a classificação de uma função de transação, pode ser que esta ainda não seja um processo elementar, cabe então reconhecer todos os processos elementares no interior desta função antes de verificar a classificação em Entrada Externa (EE), Consulta Externa (CE) ou Saída Externa (SE).

Como primeiro passo para a contagem, deve-se identificar todos os processos elementares e classificá-los quanto ao seu tipo, conforme a **Tabela 10**.

Determinam-se então os tipos de dados e os arquivos referenciados (vide **Tabela 11**). Neste passo é necessário analisar cada processo elementar e definir seus tipos de dados e os arquivos que referencia.

Este passo é mais relevante quando os tipos de dados ou os arquivos referenciados estão na fronteira da mudança da complexidade. Longe da fronteira, erros neste ponto não irão influenciar na contagem.

Verifica-se a complexidade, após definir os tipos de dados e os arquivos referenciados, determinando a complexidade de



Descrição	Tipo
Incluir Cliente	EE
Excluir Cliente	EE
Alterar Cliente	EE
Incluir Usuário	EE
Excluir Usuário	EE
Alterar Usuário	EE
Incluir Automóveis	EE
Excluir Automóveis	EE
Alterar Automóveis	EE
Registrar Locação	EE
Finalizar Locação	EE
Login (com criptografia)	SE
Consulta clientes por nome	CE
Consulta carros alugados	CE
Consulta data do aluguel	CE
Consulta clientes com carro alugado	CE
Consulta carro mais alugado	CE
Consulta cliente que mais aluga	CE

**Tabela 10.** Tipos dos processos elementares

Descrição	Tipo	Qtd. TDs	Qtd. ARs
Incluir Cliente	EE	6	1
Excluir Cliente	EE	3	1
Alterar Cliente	EE	6	1
Incluir Usuário	EE	3	2
Excluir Usuário	EE	3	2
Alterar Usuário	EE	3	1
Incluir Automóveis	EE	7	2
Excluir Automóveis	EE	3	2
Alterar Automóveis	EE	7	1
Registrar Locação	EE	3	2
Finalizar Locação	EE	4	2
Login (com criptografia)	SE	4	1
Consulta clientes por nome	CE	3	2
Consulta carros alugados	CE	3	2
Consulta data do aluguel	CE	3	2
Consulta clientes com carro alugado	CE	3	3
Consulta carro mais alugado	CE	3	3
Consulta cliente que mais aluga	CE	3	2

**Tabela 11.** Tipos de Dados (TD) e Arquivos Referenciados (AR)

cada processo elementar consultando a **Tabela 7** ou **Tabela 8**. O resultado pode ser visto na **Tabela 12**.

A seguir, determina-se a contribuição de cada processo elementar e realiza-se a soma de todos, conforme **Tabela 13**.

Para determinar a contribuição basta verificar na **Tabela 9** o ponto de função referente a cada complexidade.

Descrição	Tipo	Qtd. TDs	Qtd. ARs	Complexidade
Incluir Cliente	EE	6	1	Baixa
Excluir Cliente	EE	3	1	Baixa
Alterar Cliente	EE	6	1	Baixa
Incluir Usuário	EE	3	2	Baixa
Excluir Usuário	EE	3	2	Baixa
Alterar Usuário	EE	3	1	Baixa
Incluir Automóveis	EE	7	2	Média
Excluir Automóveis	EE	3	2	Baixa
Alterar Automóveis	EE	7	1	Baixa
Registrar Locação	EE	3	2	Baixa
Finalizar Locação	EE	4	2	Baixa
Login (com criptografia)	SE	4	1	Baixa
Consulta clientes por nome	CE	3	2	Baixa
Consulta carros alugados	CE	3	2	Baixa
Consulta data do aluguel	CE	3	2	Baixa
Consulta clientes com carro alugado	CE	6	3	Média
Consulta carro mais alugado	CE	3	3	Baixa
Consulta cliente que mais aluga	CE	3	2	Baixa

**Tabela 12.** Complexidade

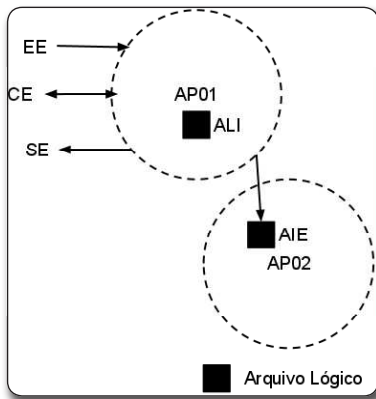
Descrição	Tipo	Qtd. TDs	Qtd. ARs	Complexidade	Contribuição
Incluir Cliente	EE	6	1	Baixa	3
Excluir Cliente	EE	3	1	Baixa	3
Alterar Cliente	EE	6	1	Baixa	3
Incluir Usuário	EE	3	2	Baixa	3
Excluir Usuário	EE	3	2	Baixa	3
Alterar Usuário	EE	3	1	Baixa	3
Incluir Automóveis	EE	7	2	Média	4
Excluir Automóveis	EE	3	2	Baixa	3
Alterar Automóveis	EE	7	1	Baixa	3
Registrar Locação	EE	3	2	Baixa	3
Finalizar Locação	EE	4	2	Baixa	3
Login (com criptografia)	SE	4	1	Baixa	4
Consulta clientes por nome	CE	3	2	Baixa	3
Consulta carros alugados	CE	3	2	Baixa	3
Consulta data do aluguel	CE	3	2	Baixa	3
Consulta clientes com carro alugado	CE	6	3	Média	4
Consulta carro mais alugado	CE	3	3	Baixa	3
Consulta cliente que mais aluga	CE	3	2	Baixa	3
Total de Pontos de Função =					57

**Tabela 13.** Contagem das funções do tipo transação

**Pontos de função não ajustados**

O quarto passo para a contagem é determinar a contagem dos pontos de função não ajustados, como pode ser visto na **Figura 1**. Neste ponto entende-se a relação de um arquivo lógico com um processo elementar.

Na **Figura 10** tem-se um exemplo de uma aplicação AP01 com um ALI e uma série de processos elementares. Ela realiza uma leitura de um arquivo lógico da aplicação AP02. Esse arquivo lógico localiza-se fora da fronteira da aplicação AP01 e deve ser classificado como um AIE.



**Figura 10.** Relação arquivo lógico e processo elementar

Agora se deve realizar a contagem dos pontos de função não ajustados. Esta análise é simples. Deve-se apenas somar as contribuições das funções do tipo dado com as contribuições das funções do tipo transação.

**Aplicando o conhecimento – Pontos de função não ajustados**

Para realizar a determinação dos pontos de função não ajustados, deve-se somar as contribuições de todas as funções do tipo dado e do tipo transação (ver **Tabela 14**).

Descrição	Contribuição
Funções do tipo dado	21 PF
Funções do tipo transação	57 PF
Total de Pontos de Função Não Ajustados =	78 PF

**Tabela 14.** Pontos de função não ajustados

**Determinar o fator de ajuste**

O quinto passo para a contagem é determinar o valor do fator de ajuste, como pode ser visto na **Figura 1**.

Para o quinto passo deve-se determinar o fator de ajuste, mas essa análise não será feita e será atribuído o valor do fator de ajuste como um. A adoção desse valor será melhor compreendida no exemplo apresentado.

O fator de ajuste, pelo seu caráter subjetivo e o impacto gerado na contagem, podendo ser de +35% a -35%, fez com que vários utilizadores do método de análise de ponto de função ignorassem essa etapa antes mesmo do IFPUG classificá-la como opcional em 2002.

**Aplicando o conhecimento – Fator de ajuste**

Não será feita análise para esta etapa, uma vez que a mesma é instituída opcional pelo IFPUG e pode aumentar o erro na estimativa.

Para o nosso exemplo, deve-se considerar:

- Valor de ponto de função não ajustado (VAF) = 1;

**Realizar o cálculo dos pontos de função ajustados**

O sexto e último passo para a contagem é calcular os pontos de função ajustados, como pode ser visto na **Figura 1**.

Esta é a etapa final para obter o tamanho funcional do seu projeto. Existem três tipos de contagem, como já foi dito: Projeto de Desenvolvimento, Projeto de Melhoria e Aplicação.

Como este artigo visa à contagem de projeto de desenvolvimento, não serão tecidos detalhes dos demais tipos de contagem.

Para determinar os pontos de função ajustados para projeto de desenvolvimento é necessário aplicar a seguinte fórmula apresentada na **Tabela 15**.

$DFP = (UFP + CFP) \times VAF$
Onde:
DFP: Número de pontos de função do projeto de desenvolvimento;
UFP: Número de pontos de função não ajustados das funções disponíveis aos usuários após a instalação;
CFP: Número de pontos de função não ajustados das funções de conversão, ou seja, as funções transitórias que são inutilizadas após a instalação;
VAF: Valor do fator de ajuste.

**Tabela 15.** Tabela da fórmula dos pontos de função de projeto de desenvolvimento

**Aplicando o conhecimento – Pontos de função ajustados**

Todos os valores estimados até este ponto serão utilizados para determinar os pontos de função ajustados. Para terminar a contagem do projeto de desenvolvimento, deve-se substituir os valores estimados até aqui na fórmula apresentada na **Tabela 16**.

A aplicação contada não possui funções de conversão, por este motivo foi somado zero às funções disponíveis após a instalação.

Assim, a aplicação contada possui um tamanho funcional estimado em 78 pontos de função.

$DFP = (UFP + CFP) \times VAF$
Onde:
DFP: Número de pontos de função do projeto de desenvolvimento;
UFP: Número de pontos de função não ajustados das funções disponíveis após a instalação;
CFP: Número de pontos de função não ajustados das funções de conversão;
VAF: Valor do fator de ajuste.
Resultado:
$DFP = (78 + 0) \times 1$
$DFP = 78$ Pontos de Função

**Tabela 16.** Valor total dos Pontos de Função ajustados

**Derivações**

Neste ponto já se possui o tamanho funcional da aplicação, agora serão apresentadas as derivações que podem ser realizadas com ele.

Até aqui se utilizou a APF na perspectiva de produto, agora pode-se fazer uma análise na perspectiva de processo (esforço, custo e prazo). Independente da derivação, o importante é possuir um histórico de projeto, só assim será possível estimar esforço, custo e prazo de forma satisfatória. Na primeira vez que aplicar estas estimativas o erro será grande, mas conforme for ampliando a base de históricos de projetos, o erro tenderá a diminuir.

### Esforço

Para calcular o esforço é necessário conhecer quantos pontos de função são produzidos em uma hora e saber quantas horas de trabalho são consideradas em um mês na empresa.

A estimativa de esforço pode ser:

- Pontos de Função por Homem Mês (PF/HM);
- Pontos de Função por Hora (PF/H).

Tem-se por base que a taxa de produtividade é medida em hora por ponto de função (H/PF). Cada linguagem ou tecnologia demandam um esforço diferente, essas características não influenciam nos pontos de função, mas sim no esforço que demanda produzir cada ponto de função.

Existem vários editais para licitação de desenvolvimento de sistemas, que incluem tabelas de produtividade mínima no desenvolvimento de projetos.

A **Tabela 17** foi retirada do edital da ACINE (Agência Nacional do Cinema), que define a produtividade mínima de PF que a empresa deverá produzir por linguagem no tempo definido. Por exemplo, uma empresa que concorrer a esse edital, deverá ter competência de produzir no mínimo um PF a cada quinze horas na linguagem Java.

Desenvolvimento e manutenção de sistemas	
Tecnologia	Produtividade Mínima
Java	15 h/PF
ASP (Vbscript e Javascript)	10 h/PF
PHP	11 h/PF
JSP	13 h/PF
HTML	7 h/PF
Cold Fusion	11 h/PF
Delphi	9 h/PF
Crystal reports	9 h/PF
PL/SQL	9 h/PF
Visual Basic	9 h/PF

**Tabela 17.** Tabela de produtividade mínima ACINE

Utilizar bases de editais (sem o conhecimento sobre o projeto) ou de outras empresas, constitui um risco muito grande, pois a produtividade é intrínseca de cada empresa, pois essas possuem funcionários e processos diferenciados.

### Custo

A estimativa do custo de um projeto é a informação primordial na hora de elaborar uma proposta, este não pode exceder

às expectativas do cliente e nem ter um valor inferior ao necessário para o funcionamento da empresa.

Como na determinação do esforço, o custo também é estimado a partir de dados da empresa, neste caso é necessário ter o conhecimento do custo da hora da equipe de desenvolvimento ou o valor de um ponto de função para sua empresa.

O custo de um Ponto de Função é dado por:

- Custo por hora vezes a quantidade de horas necessárias para produzir um Ponto de Função ( $C/H \times H/PF$ ).

### Prazo

O prazo é um fator crítico a ser determinado pois, para estimativas, pode-se supor que o prazo é uma função linear com o recurso, o que é uma suposição falha. Por exemplo, se um projeto desenvolvido por dois desenvolvedores gasta um prazo de dois meses, alocar mais dois desenvolvedores para o projeto não necessariamente implica que o mesmo irá durar apenas um mês.

A análise empírica mostra que essa linearidade não existe, uma mulher demora nove meses para gerar um bebê, nove mulheres não geram um bebê em um mês (VAZQUEZ, 2009).

Quanto maior o tamanho funcional de um projeto, maior será o prazo e maior será o erro. Para projetos pequenos o erro é aceitável, mas novamente volta-se ao ponto de que a melhor maneira de evitar esses erros é possuir uma base histórica dos projetos desenvolvidos.

O prazo é derivado da seguinte forma:

- Prazo é a relação de esforço por recurso ( $\text{Prazo} = \text{Esforço} / \text{Recurso}$ ).

## Aplicando o conhecimento – Derivações

Até aqui definiu-se o tamanho funcional do sistema exemplo, agora utilizando de um dos benefícios do método de análise de ponto de função que é a realização de derivações a partir da quantidade de pontos de função estimados, serão realizadas derivações quanto ao esforço, custo e prazo necessários para o desenvolvimento do software.

### Esforço

A aplicação de exemplo foi estimada em 78 pontos de função. Será considerada uma empresa que possui uma taxa de produtividade mínima em Java de 5 H/PF e com uma carga de trabalho de 130 horas por homem-mês. Essa empresa gastaria 390 horas para produzir o sistema, ou três meses, conforme a **Tabela 18**.

### Custo

Suponha que a hora de trabalho custe R\$ 20,00 e, como é produzido um ponto de função a cada cinco horas, o valor do ponto de função é de R\$ 100,00.

Estima-se que o esforço necessário para produzir a aplicação de exemplo é de 390 horas, e a mesma possui 78 pontos de função.

Pode-se assim inferir que a aplicação tem um custo de aproximadamente R\$ 7.800,00, conforme a **Tabela 19**.

Esforço = H/PF x Tamanho do projeto em PF
Onde:
H: Hora;
PF: Pontos de função.
Resultado:
Esforço = 5 x 78
Esforço = 390 horas

**Tabela 18.** Valor total em horas do esforço para desenvolver o projeto

<b>Custo = Tamanho do projeto em PF x Custo do Ponto de Função</b>
Resultado:
Custo = 78 x R\$ 100,00
Custo = R\$ 7.800,00

**Tabela 19.** Custo total para elaboração do projeto

### Prazo

Foi definido que o esforço necessário para produzir a aplicação é de 390 horas ou três meses. Suponha que esta empresa possua dois funcionários habilitados a desenvolver o projeto na tecnologia estabelecida.

Utilizando dessas informações conclui-se que o prazo para a entrega do sistema será de um mês e meio, conforme a **Tabela 20**.

<b>Prazo = Esforço / Recurso</b>
Prazo = 3 / 2
Prazo = 1 mês e 15 dias

**Tabela 20.** Prazo total para elaboração do projeto

### Considerações finais

A APF pode ser usada para estimar desenvolvimento, melhoria e aplicação. Todas as formas de estimativa e derivações são referentes à fase de construção, mas é de conhecimento que na produção de um sistema, têm-se ainda as fases de concepção, elaboração e transição. É importante dizer que as dicas que serão passadas não possuem nenhuma ligação com o método de APF, mas sim uma observação de como extrapolar os valores derivados de esforço, prazo e custo em todo o processo de produção do sistema.

Embora o ciclo de vida varie muito por empresas e projetos diferentes, um projeto médio, possui a distribuição de esforço e programação como apresentado na **Tabela 21**.

	Concepção	Elaboração	Construção	Transição
Esforço	5%	20%	65%	10%
Programação	10%	30%	50%	10%

**Tabela 21.** Distribuição de esforço e programação em projetos de médio porte (Philippe Kruchten, 2003)

Para extrapolar os resultados obtidos nas derivações da APF, basta fazer uma regra de três simples. Se o esforço destinado à

fase de construção é de 65% e você possui o esforço e custo para o desenvolvimento desta fase, pode-se extrapolar um de cada vez para todas as fases, depois são feitos ajustes necessários de acordo com cada empresa.

Utilizando dos valores obtidos neste artigo, segue o exemplo para extrapolar o esforço para a construção do sistema, no esforço para a realização da fase de concepção do projeto, conforme a **Tabela 22**.

<b>Extrapolando Esforço Construção em Esforço Iniciação</b>
65% - 390 (horas)
05% - X (horas)
X = 30 horas

**Tabela 22.** Esforço total extrapolado para realização da fase de concepção

O prazo pode ser obtido a partir do esforço necessário para produzir e dos recursos disponíveis.

Assim, deve-se realizar análise histórica dos projetos e construir uma tabela específica para cada organização, mas observando cada projeto, pois ajustes serão sempre necessários.

### Referências

1. VAZQUEZ, C.E., SIMÕES, G.S., ALBERT, R.M. Análise de ponto de função medição, estimativa e gerenciamento de projetos de software. São Paulo, Editora Érica, 2009
2. Softex. MPS.BR - Melhoria de processo do software brasileiro - Guia geral, 2009
3. IFPUG (International Function Point Users Group). Disponível em: <<http://www.ifpug.org>>.
4. BFPUG (Brazilian Function Point Users Group). Disponível em: <<http://www.bfpug.com.br>>.
5. PMI (Project Management Institute). Um Guia do Conjunto de Conhecimentos em Gerenciamentos de Projetos (PMBOK). Estados Unidos: PMI Publications, 2004
6. DEKKERS, C. Pontos de Função e Medidas - O Que é um Ponto de Função?. QAI Journal, dez. 1998
7. DEKKERS, C. Desmistificando Pontos de Função: Entendendo a Terminologia. IT Metrics Strategies, out. 1998
8. HAZAN, Cláudia - Análise de Pontos por Função - agosto, 2001. disponível em <http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/APF.pdf>.
9. ACINE. Anexo XVIII - Tabelas de produtividade mínima, 2008. Disponível em: <<http://www.ancine.gov.br/media/concorrenca0012008/AnexoXVIII.pdf>>.
10. Philippe Kruchten. The Rational Unified Process: An Introduction. Boston, Editora Pearson Education, 2003

### Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista! Dê seu voto sobre este artigo, através do link:

[www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)

