

DEPARTAMENTO DE INFORMÁTICA - DPI

# Guia Prático em Análise de Ponto de Função

---

**Projeto: Jhoney da Silva Lopes**

**Orientador: José Luis Braga**

## Sumário

1. INTRODUÇÃO.....	3
1.1 Análise de Ponto de Função .....	3
1.2 Objetivo .....	3
1.3 Motivação e Benefícios .....	4
1.4 Pontos chave .....	4
2. COMO REALIZAR A CONTAGEM DE PONTO DE FUNÇÃO .....	6
2.1 Visão geral .....	6
2.1.1 Requisitos .....	6
2.2 Determinar o tipo de contagem .....	7
2.2.1 Projeto de desenvolvimento .....	7
2.2.2 Projeto de melhoria.....	7
2.2.3 Aplicação.....	8
2.2.4 Aplicando o conhecimento.....	8
2.3 Identificar o escopo da contagem .....	8
2.3.1 Aplicando o conhecimento.....	9
2.4 Contar funções do tipo dado .....	9
2.4.1 Arquivo Lógico Interno .....	10
2.4.2 Arquivo de Interface Externa.....	10
2.4.3 Determinação da complexidade e da contribuição .....	10
2.4.4 Aplicando o conhecimento.....	12
2.5 Contar funções do tipo transação .....	15
2.5.1 Entrada Externa .....	15
2.5.2 Saída Externa .....	16
2.5.3 Consulta Externa.....	16
2.5.4 Determinação da complexidade e da contribuição .....	16
2.5.5 Aplicando o conhecimento.....	18
2.6 Pontos de função não ajustados .....	22
2.6.1 Aplicando o conhecimento.....	23
2.7 Determinar o fator de ajuste .....	23
2.7.1 Aplicando o conhecimento.....	24
2.8 Realizar o cálculo dos pontos de função ajustados.....	24
2.8.1 Aplicando o conhecimento .....	25
3. DERIVAÇÕES .....	26
3.1 Esforço .....	26
3.1.1 Aplicando o conhecimento.....	27
3.2 Custo.....	27
3.2.1 Aplicando o conhecimento.....	27
3.3 Prazo .....	27
3.3.1 Aplicando o conhecimento.....	28
4. CONSIDERAÇÕES.....	29
5. BIBLIOGRAFIA .....	30

## 1. INTRODUÇÃO

A elaboração desse guia visa auxiliar micro e pequenas empresas na utilização de uma técnica para estimar os seus projetos em custo, prazo e esforço. Muitas empresas não utilizam de técnicas para estimar os seus projetos, a maioria possui um funcionário com experiência que avalia os projetos a partir do seu “feeling” sem utilizar nenhum padrão.

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 guia organiza de forma simples e introdutória conhecimentos sobre a análise de ponto de função.

O guia não tem a intenção de substituir o uso apropriado e completo da contagem de ponto função, mas mostrar que existem ferramentas usuais que solucionam problemas recorrentes de várias empresas.

### 1.1 Análise de Ponto de Função

Análise de Ponto de Função é uma técnica de medição do tamanho funcional de um software. Essas funções são operações extraídas dos requisitos funcionais gerados a partir da visão do usuário<sup>1</sup>. A partir dessa medição é possível estimar o esforço para implementação do sistema utilizando Ponto de Função que é a unidade de medida desta técnica.

APF tem por definição medir o que o software faz, e não como ele foi construído, portanto o processo de medição é fundamentado em uma avaliação padronizada dos requisitos lógicos do usuário.

Sobre o estudo desse método é importante destacar que pontos de função não medem diretamente o esforço, produtividade, custo ou outras informações específicas. É exclusivamente uma medida de tamanho funcional de software que aliado a estimação de outras variáveis, poderá ser usado para derivar produtividade, custo e estimar esforço.

Essa técnica surgiu no início da década de 70 na IBM, desenvolvida por Allan Albrecht (Vazquez,2009), como uma alternativa às métricas baseadas em linhas de código. O IFPUG (International Function Point Users Group) é uma entidade sem fins lucrativos, composta por pessoas e empresas de diversos países cuja finalidade é promover um melhor gerenciamento dos processos de desenvolvimento e manutenção de software com o uso de pontos de função e outros métodos ([www.ifpug.org](http://www.ifpug.org)).

### 1.2 Objetivo

Muitas micro e pequenas empresas passam pela dificuldade de orçar prazo, custo e esforço para os seus projetos. A elaboração desse guia não tem por objetivo tratar de todas as possíveis variações no processo de contagem, mas sim proporcionar uma visão geral sobre a metodologia e com isso auxiliar em uma aproximação do valor real da contagem, ou seja, uma estimativa desse valor.

O objetivo desse guia é auxiliar na estimativa em pontos de função na fase inicial do ciclo de vida de um projeto de desenvolvimento. Na fase inicial, você possui apenas a proposta para o projeto, por este motivo não é possível medir o tamanho

---

<sup>1</sup> Em APF usuário possui um conceito mais amplo. Qualquer entidade que se relacione com o sistema ou produza um ônus ao mesmo. Ex: Pessoa, aplicação, leis, restrições e etc.

funcional do software, pois os requisitos não estão maduros, mas é possível realizar uma estimativa em pontos de função para o mesmo.

Essa abordagem simples e direta tem por finalidade também difundir o uso da técnica de análise de ponto de função, realizando uma visão geral com o intuito de instigar os seus utilizadores a estudos mais aprofundados sobre a metodologia. Não é objetivo deste guia ofender nenhuma organização ou profissional certificado e experiente na utilização da metodologia nem tão pouco limitar o estudo e utilização da mesma.

### **1.3 Motivação e Benefícios**

É necessário saber qual é a sua verdadeira motivação para a utilização da técnica de análise de ponto de função. O que ganhamos medindo um software? Pense em um terreno, esse possui uma área, você o mediria para poder vender, comprar, construir. Fica fácil perceber motivos pelos quais você mediria um terreno, certo?

Você compraria um terreno sem saber o seu tamanho? Com softwares deveria ser a mesma situação.

Em uma obra você precisa saber a área a ser construída para poder comprar os materiais, contratar pessoas e com isso também avaliar o tempo de elaboração da mesma. Quando medimos softwares utilizando a técnica de análise de ponto de função, podemos realizar as mesmas derivações a partir do seu tamanho funcional, ou seja, estimar o esforço, custo e prazo.

Com isso é possível observar uma série de benefícios enumerados por (VAZQUEZ,2009):

1. Controlar o andamento da produtividade de um determinado software. Um sistema pode ter mais de uma equipe envolvida em seu desenvolvimento, é possível avaliar a produtividade de diferentes equipes pela quantidade de Pontos de Função entregados.
2. Realizar a medição do tamanho funcional do software e com isso estimar, custo, esforço e prazo. Uma vez realizada a medição ou estimativa dos Pontos de Função totais do sistema é possível utilizar este número para realizar derivações.
3. Sabendo o tamanho funcional de um software é possível realizar comparações. Pode ser realizada uma avaliação entre dois ou mais sistemas.
4. Com a utilização da técnica é possível tomar decisões do tipo “Make or Buy”, seria a decisão de desenvolver um sistema ou comprar uma solução pronta no mercado.
5. Utilizar a medida para fundamentar contratos de compra e venda de softwares ou contratar serviços.

### **1.4 Pontos chave**

1. Análise de Ponto de função é uma técnica que mede o tamanho funcional de um software do ponto de vista do usuário;
2. IFPUG - International Function Point Users Group é o órgão internacional responsável pela manutenção e evolução da técnica;

3. Medir ou estimar? Para a utilização da técnica de APF com o intuito de medição, é necessário que os requisitos do sistema estejam maduros. Logo a efetividade de uma medida só é possível após a instalação da aplicação;
4. Análise de ponto de função não leva em consideração como o software é construído, mas sim o que ele faz.

## 2. COMO REALIZAR A CONTAGEM DE PONTO DE FUNÇÃO

### 2.1 Visão geral

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

A análise em ponto de função 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

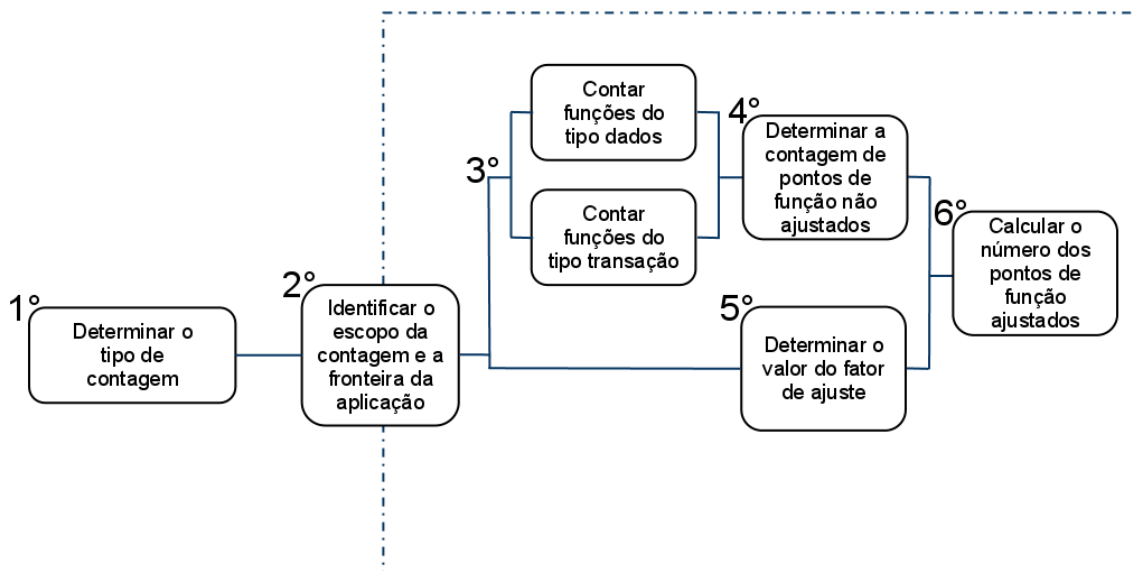


Figura 2.1: Passos para a contagem dos pontos de função

#### 2.1.1 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 está ligada diretamente ao sucesso ou ao fracasso de um projeto.

Na aplicação da análise de ponto de função a definição destes requisitos é tão importante quanto para qualquer outro fim, pois você pode subestimar ou superestimar sua contagem e com isso afetar todas as derivações possíveis da técnica.

Claro que é impossível extrair todos estes requisitos nesta fase inicial, logo uma melhor extração irá gerar uma melhor estimativa.

## 2.2 Determinar o tipo de contagem

O primeiro passo para a contagem:

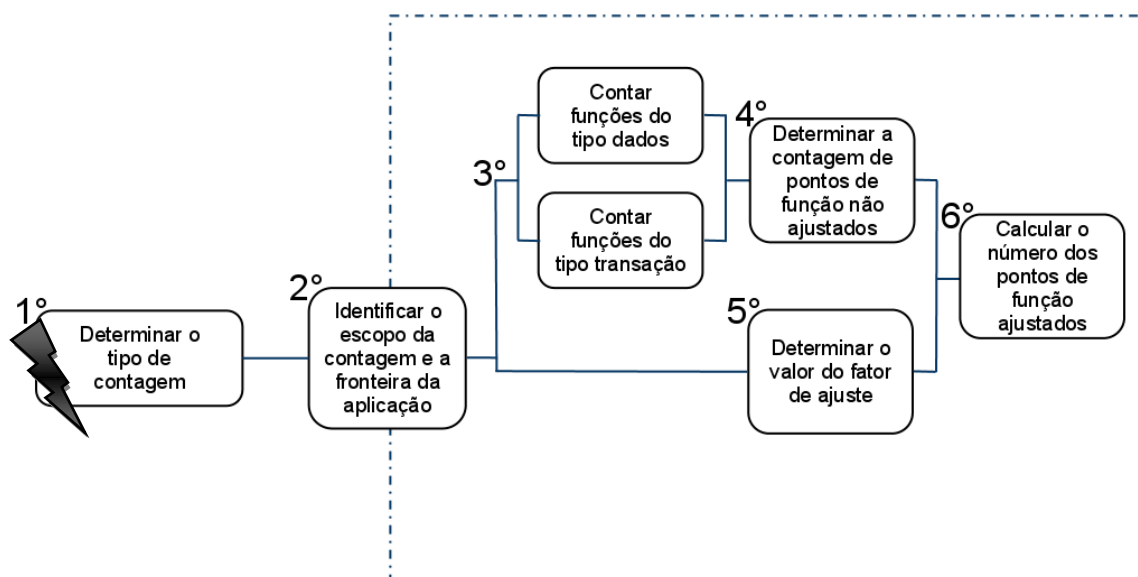


Figura 2.2: Determinar o tipo de contagem

Na análise de ponto de função existem três tipos de contagem:

1. Projeto de desenvolvimento;
2. Projeto de melhoria;
3. Aplicação.

O guia tem por objetivo apresentar a solução de contagem para projeto de desenvolvimento, mas os outros tipos também serão apresentados.

### 2.2.1 Projeto de desenvolvimento

É caracterizado como projeto de desenvolvimento, um novo projeto desde a fase de extração de requisitos até a instalação do mesmo.

Neste tipo de projeto é contado na análise de ponto de função todas as funcionalidades fornecidas aos usuários até a instalação do sistema, ou seja, funcionalidades de conversão também são contadas. Por 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.

Nós só conseguimos 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.

### 2.2.2 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.

### 2.2.3 Aplicação

Entende-se por contagem do tipo aplicação<sup>2</sup> um software instalado, ou seja, a contagem após o término de um projeto de desenvolvimento. Neste caso não levamos em consideração as funções do tipo conversão.

### 2.2.4 Aplicando o conhecimento

Esta etapa está pronta, o foco deste guia 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.

Exemplo de caso: Tipo de contagem - Projeto de desenvolvimento.

## 2.3 Identificar o escopo da contagem

O segundo passo para a contagem:

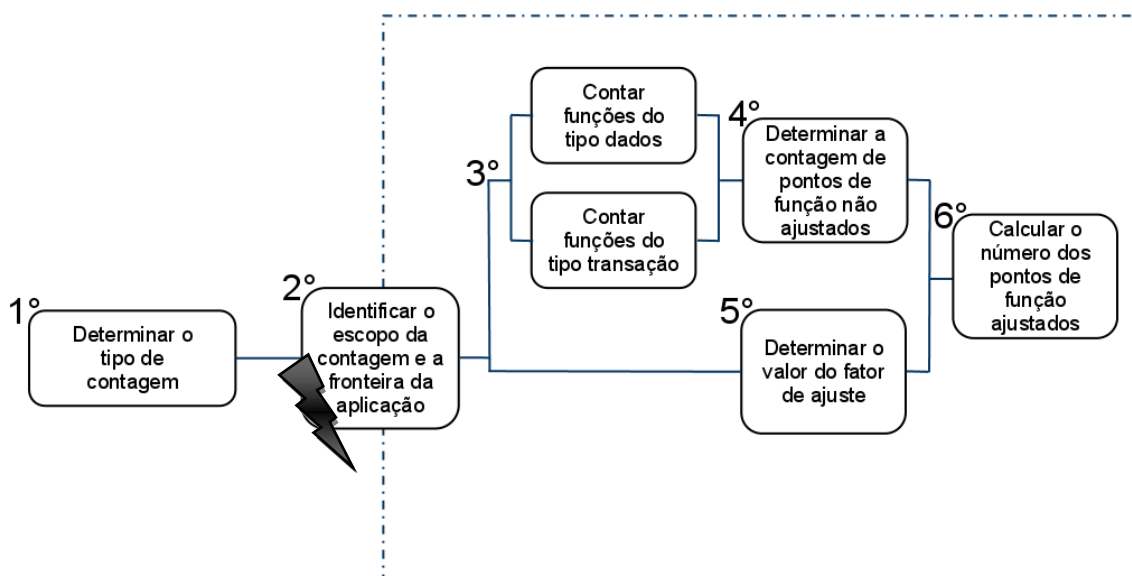


Figura 2.3: Identificar o escopo da contagem e a fronteira da aplicação

Muitas vezes a identificação do escopo e da fronteira da aplicação não são levados tão a sério, principalmente por empresas que não utilizam de gerência de projetos.

Esta é uma etapa crucial para o andamento do projeto, a definição de um escopo<sup>3</sup> errado pode acarretar em prejuízos incalculáveis 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 utilizadas ou específicas.

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

Uma tarefa 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.

<sup>2</sup> Aplicação neste caso pode ser interpretada como sistema.

<sup>3</sup> 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.



Exemplo: Neste caso são apresentadas três aplicações, AP01, AP02 e AP03, cada uma com arquivo lógico interno – ALI e referenciando arquivos de interface externa – AIE, serão apresentados detalhes sobre os arquivos lógicos no próximo tópico.

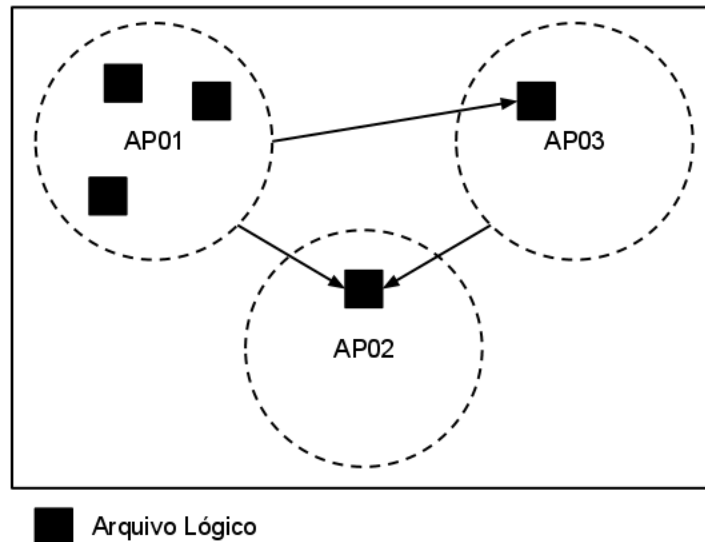


Figura 2.4: Arquivos lógicos e fronteiras das aplicações

### 2.3.1 Aplicando o conhecimento

Como foi visto, nesta etapa devemos definir o escopo da contagem e a fronteira da aplicação.

Exemplo de caso: Software destinado a uma empresa que realiza locação de automóveis, o sistema é simples e composto por uma única aplicação.

### 2.4 Contar funções do tipo dado

O terceiro passo primeira parte:

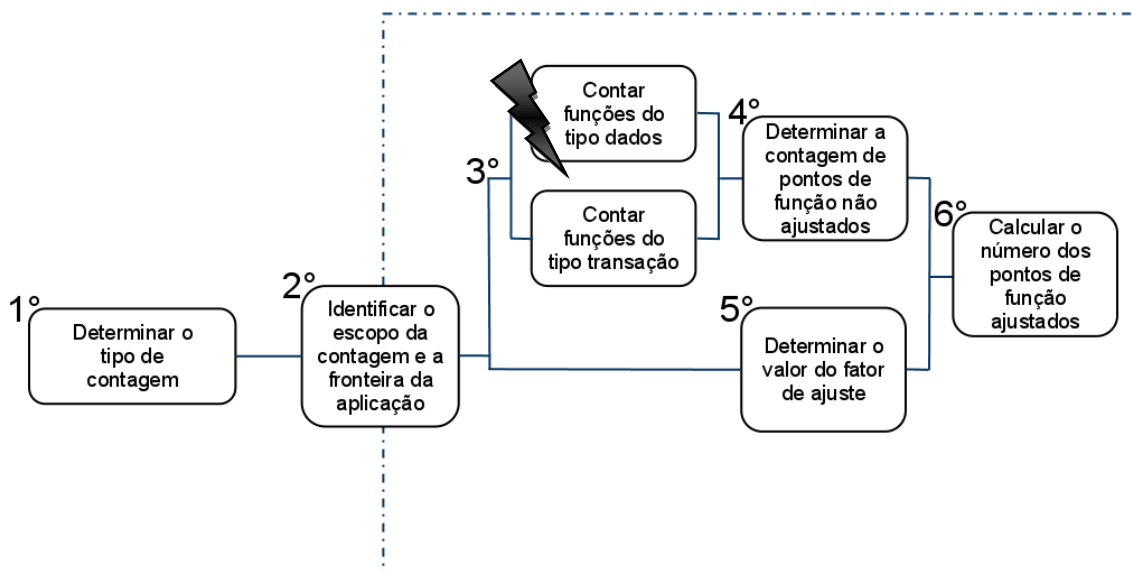


Figura 2.5: Contar funções do tipo dados

Nesta etapa iniciamos o processo de contagem, as funções do tipo dado são as funcionalidades fornecidas para o armazenamento de dados na aplicação sendo

contada, são caracterizados como arquivos lógicos e eles podem ser mantidos pela aplicação ou lida de outra, como no exemplo da (figura 2.4).

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).

#### 2.4.1 Arquivo Lógico Interno

Grupo lógico de dados e persistentes mantidos dentro da fronteira da aplicação e alterado por meio de processos elementares<sup>4</sup>.

Considere a (figura 2.4), a AP01 possui três arquivos lógicos internos (ALI), a primeira vista parecerá que cada tabela do banco de dados da sua aplicação será um ALI, mas é um erro realizar essa premissa, pois um grupo de tabelas pode ser considerado como um único arquivo lógico.

Exemplos:

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

Não são exemplos:

1. Arquivos temporários ou de backup.
2. Tabelas temporárias ou views.

#### 2.4.2 Arquivo de Interface Externa

Grupo lógico de dados e persistentes mantidos dentro da fronteira de outra aplicação, mas requerido ou referenciado pela aplicação que está sendo contada.

Considere a (figura 2.4), a AP01 referencia arquivos lógicos da AP02 e AP03, estes arquivos são denominados arquivos de interface externa (AIE).

Exemplos:

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

Não são exemplos:

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

#### 2.4.3 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.

Essa 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, seria cada atributo de uma tabela.

---

<sup>4</sup> Um processo elementar é a menor unidade de atividade significativa para o usuário final (VAZQUEZ,2009). É a menor funcionalidade disponibilizada ao usuário.

### Tipos de Registro (TR):

É um subgrupo de dados.

Em uma análise míope, quando um agrupamento de tabelas são caracterizadas como um único arquivo lógico, ALI ou AIE, a tabela reconhecida pelo usuário é contada e as demais se tornam tipos de registro. Os campos de dados dos tipos de registro são atribuídos a todos os arquivos lógicos relacionados a estes tipos de registro.

Exemplo:

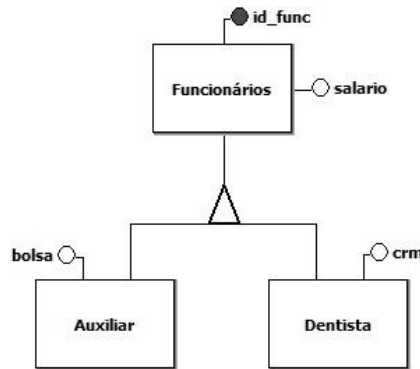


Figura 2.6: Especialização é um tipo de registro

Neste exemplo contamos funcionários como uma ALI ou AIE e incluímos as demais tabelas como tipo de registro e os seus tipos de dados são somados a Funcionários.

Exemplo:

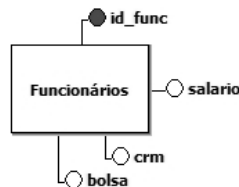


Figura 2.7: Especialização na visão do usuário

Temos a seguinte definição:

Descrição	Tipo	TD	TR
Funcionários	ALI ou AIE	4	3

Tabela 2.1: Descrição do Tipo de Registro (TR) e Tipo de Dado (TD)

São contados três tipos de registro, pois todo arquivo lógico é um tipo de registro dele mesmo.

É importante perceber que essa solução é tomada, uma vez que o usuário enxerga auxiliar e dentista como funcionário e não entidades separadas, ou seja, o importante é a visão do negócio.



Figura 2.7: Visão do Negócio

**Tabela de complexidade:**

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

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

Tabela 2.2: Complexidade ALI e AIE

**Tabela de contribuição:**

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

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

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 2.3: Tabela de contribuição

**2.4.4 Aplicando o conhecimento**

Para facilitar a identificação dos tipos de arquivos, deve-se elaborar um modelo lógico.

Uma dica geral e objetiva, mas passível de erro, 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. Se o usuário não reconhece a tabela, mas reconhece os tipos de dados presentes na mesma, provavelmente essa tabela será um tipo de registro.

Dica para classificar um arquivo lógico:

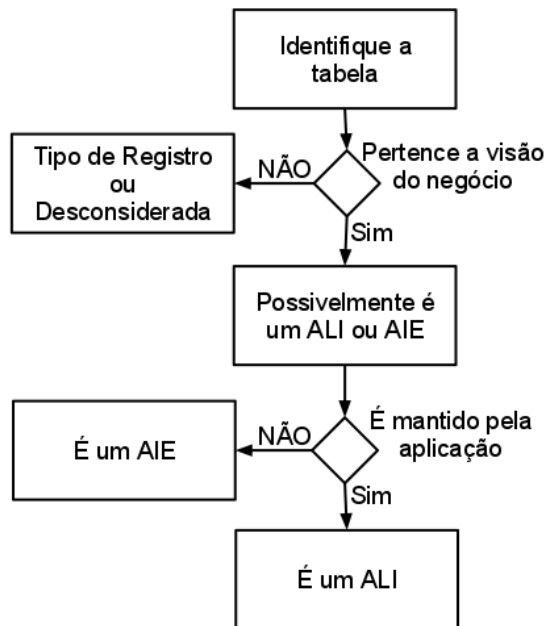


Figura 2.8: Fluxo para classificação do tipo lógico

Passos para uma estimativa da contagem desta etapa:

1. Elabore um modelo lógico do seu projeto  
Exemplo:

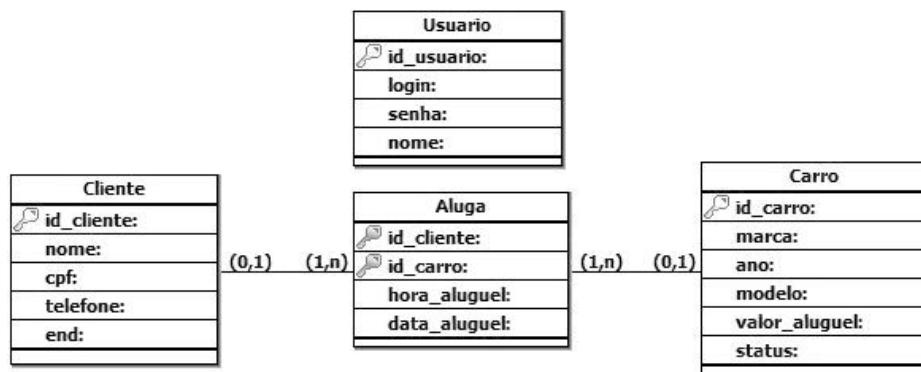


Figura 2.9: Modelo lógico

2. Identifique todas as tabelas reconhecidas pelo usuário, ou seja, as que fazem parte da visão do negócio e classifique-as como ALI ou AIE.

Exemplo:

Descrição	Tipo
Usuário	ALI
Cliente	ALI
Carro	ALI

Tabela 2.4: Classificação dos arquivos lógicos

Todas as tabelas foram caracterizadas como arquivo lógico interno, pois elas são mantidas pelo sistema sendo contado.

3. Faça uma análise de todas as tabelas que não estão na visão do negócio:

- a. Se a tabela não pertence à visão do negócio, mas os seus tipos de dados pertencem, conte-a como um tipo de registro para cada arquivo lógico relacionado a ela e atribua os seus tipos de dados a cada um deles.
- b. Se nem a tabela nem os seus tipos de dados pertencem à visão do negócio, descarte-a da contagem.

Exemplo:

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.

Descrição	Tipo	TD	TR
Usuário	ALI	4	1
Cliente	ALI	7	2
Carro	ALI	8	2

**Tabela 2.5: Tipo de Dado (TD) e Tipo de Registro (TR)**

4. Determine a complexidade de cada arquivo lógico.

Exemplo:

Para definir a complexidade basta analisar a quantidade de tipos de dados mais os tipos de registro e conferir (tabela 2.2):

Descrição	Tipo	TD	TR	Complexidade
Usuário	ALI	4	1	Baixa
Cliente	ALI	7	2	Baixa
Carro	ALI	8	2	Baixa

**Tabela 2.6: Complexidade**

5. Determine a contribuição de cada arquivo lógico.

Exemplo:

Para determinar a contribuição basta verificar na (tabela 2.3) o ponto de função referente a cada complexidade.

Descrição	Tipo	TD	TR	Complexidade	Contribuição
Usuário	ALI	4	1	Baixa	7
Cliente	ALI	7	2	Baixa	7
Carro	ALI	8	2	Baixa	7

**Tabela 2.7: Contribuição**

6. Realize a soma de todas as contribuições.

Exemplo:

Para finalizar a contagem das funções do tipo dados, some as contribuições de todos os arquivos lógicos:

Descrição	Tipo	TD	TR	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 2.8: Contagem das funções do tipo dados

## 2.5 Contar funções do tipo transação

Terceiro passo segunda parte:

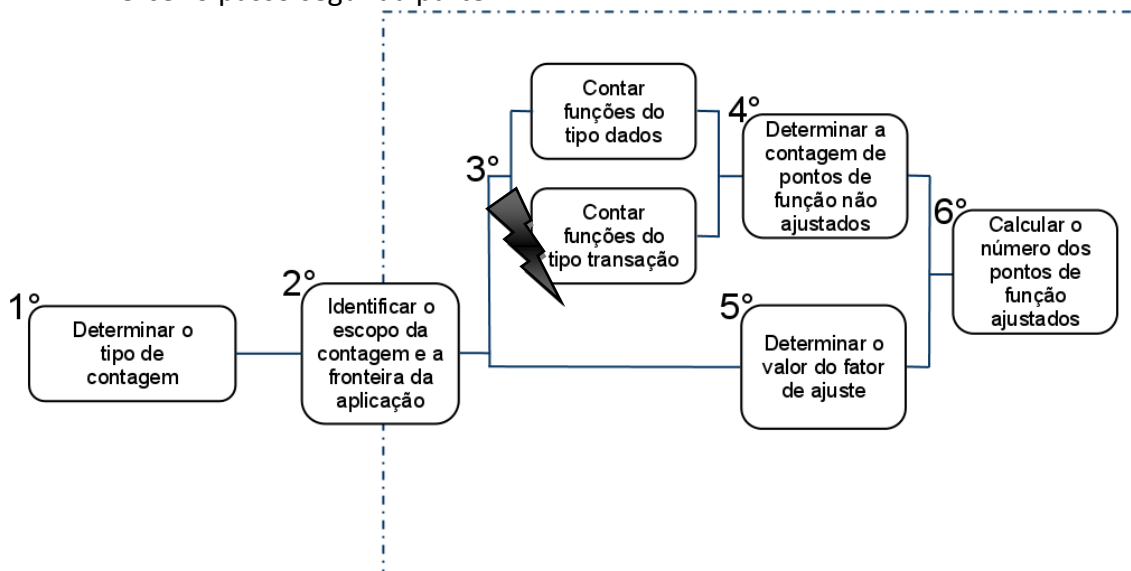


Figura 2.10: Contar funções do tipo transação

Agora que aprendemos a contar funções do tipo dados, podemos dar continuidade a contagem da aplicação. As funções do tipo transação são as funcionalidades base para o funcionamento do sistema, estas funções são chamadas de processos elementares e são classificadas em Entradas Externas, Saídas Externas, 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, podemos 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.

Um processo elementar deve ser único. Por exemplo, consultas que diferem uma da outra pela organização dos dados gerados, não podem ser consideradas diferentes.

### 2.5.1 Entrada Externa

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.

Definida por (VAZQUEZ,2009) como sua principal intenção manter (incluir, alterar ou excluir dados) um ou mais Arquivos Lógicos Internos e/ou alterar a forma como o sistema se comporta.

Exemplos:

1. Transações destinadas a manter Arquivos Lógicos Internos.
2. Processos destinados a realizar registros.

Não são exemplos:

1. Telas de filtro.
2. Preenchimento de campos de dados.
3. Telas de login.
4. Gerar relatórios.

### **2.5.2 Saída Externa**

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

Definida por (VAZQUEZ,2009) como sua principal intenção apresentar informação a partir de lógica de processamento que não seja uma simples recuperação de dado ou informação de controle, podendo manter Arquivos Lógicos Internos e alterar o comportamento do sistema.

Exemplos:

1. Tela de login (com criptografia).
2. Relatórios financeiros, supondo estes gerados por cálculos.
3. Consultas complexas com processamento de dados a partir de cálculos.
4. Apresentação de gráficos com dados processados a partir de cálculos.

Não são exemplos:

1. Telas de filtro.
2. Consultas simples, sem processamento de dados utilizando cálculos.

### **2.5.3 Consulta Externa**

Processo elementar que apresenta informação ao usuário ou a outra aplicação externa por meio de recuperação simples.

Definida por (VAZQUEZ,2009) como sua principal intenção apresentar informações ao usuário por meio de uma simples recuperação de dados ou informações de controle de ALIs e/ou AIEs, sendo que a lógica de processamento não deve conter cálculos ou fórmulas matemáticas e não deve alterar o comportamento do sistema.

Exemplos:

1. Consultar clientes pelo nome.
2. Apresentar dados em formato gráfico a partir de recuperação simples.

Não são exemplos:

1. Relatórios financeiros, gerados a partir de cálculos.
2. Telas de filtro.

### **2.5.4 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:**



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 entre outros são tipos de dados, já em um relatório, o código do produto, o nome, a descrição, o valor, em um gráfico o raciocínio é o mesmo:

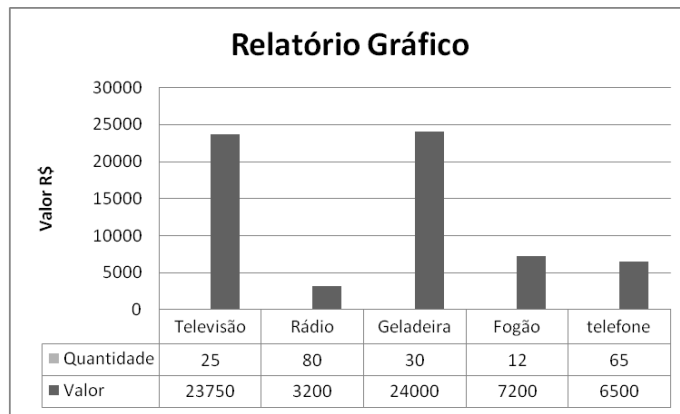


Figura 2.11: Apresentação de relatório gráfico

Contamos um tipo de dado para o nome do produto, um para a quantidade e um para o valor. No total temos três tipos de dados neste relatório.

**Arquivo Referenciado:**

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

**Tabela de complexidade:**

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

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

Tabela 2.9: 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 2.10: Complexidade Saída Externa (SE) e Consulta Externa (CE)

**Tabela de contribuição:**

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

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

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 2.11: Tabela de Contribuição

### 2.5.5 Aplicando o conhecimento

Para finalizar o terceiro passo nós devemos determinar a contagem das funções do tipo transação.

O fluxo a seguir auxilia na determinação do tipo do processo elementar:

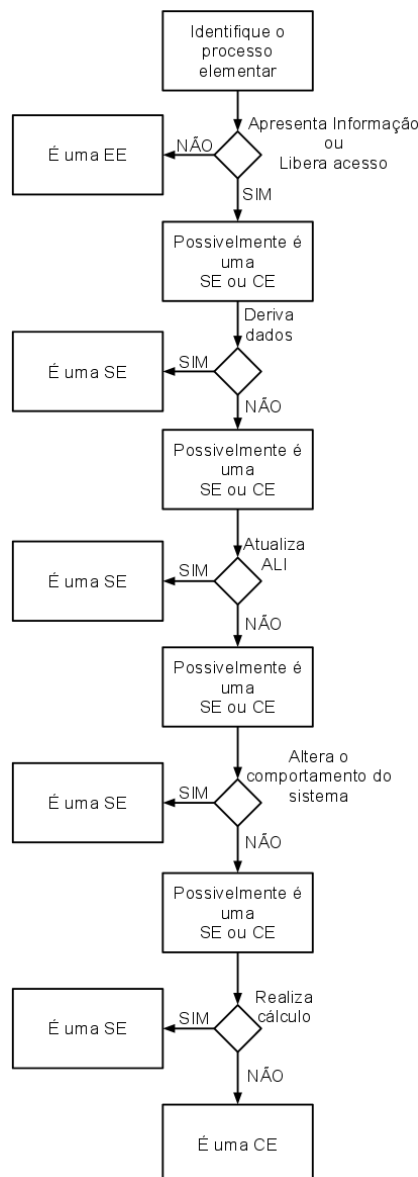


Figura 2.12: Fluxo para classificação do processo elementar

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

Outra dica, quando você não reconhece a classificação de uma função de transação, pode ser que esta ainda não é 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).

1. Identifique todos os processos elementares

Exemplo:

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

**Tabela 2.12: Identificação dos processos elementares**

2. Classifique o processo elementar quanto ao seu tipo  
Para facilitar a identificação utilize o fluxo (figura 2.12).

Exemplo:

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 2.13: Tipos dos processos elementares**

3. Determine os tipos de dados e os arquivos referenciados

Neste passo é necessário analisar cada processo elementar e definir seus tipos de dados e os arquivos aos quais 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.

Exemplo:

Descrição	Tipo	TD	AR
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 2.14: Tipos de Dados (TD) e Arquivos Referenciados (AR)**

4. Verifique a complexidade

Após definir os tipos de dados e os arquivos referenciados, determine a complexidade de cada processo elementar consultando a (tabela 2.9 ou tabela 2.10).

Exemplo:

Descrição	Tipo	TD	AR	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 2.15: Complexidade**

5. Determine a contribuição de cada processo elementar

Para determinar a contribuição basta verificar na (tabela 2.11) o ponto de função referente a cada complexidade.

Exemplo:

Descrição	Tipo	TD	AR	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

**Tabela 2.15: Contribuição**

6. Determine a contribuição total

Para finalizar a contagem das funções do tipo dados, some as contribuições de todos os processos elementares.

Exemplo:

Descrição	Tipo	TD	AR	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 2.16: Contagem das funções do tipo transação

## 2.6 Pontos de função não ajustados

O quarto passo é determinar a contagem dos pontos de função não ajustados:

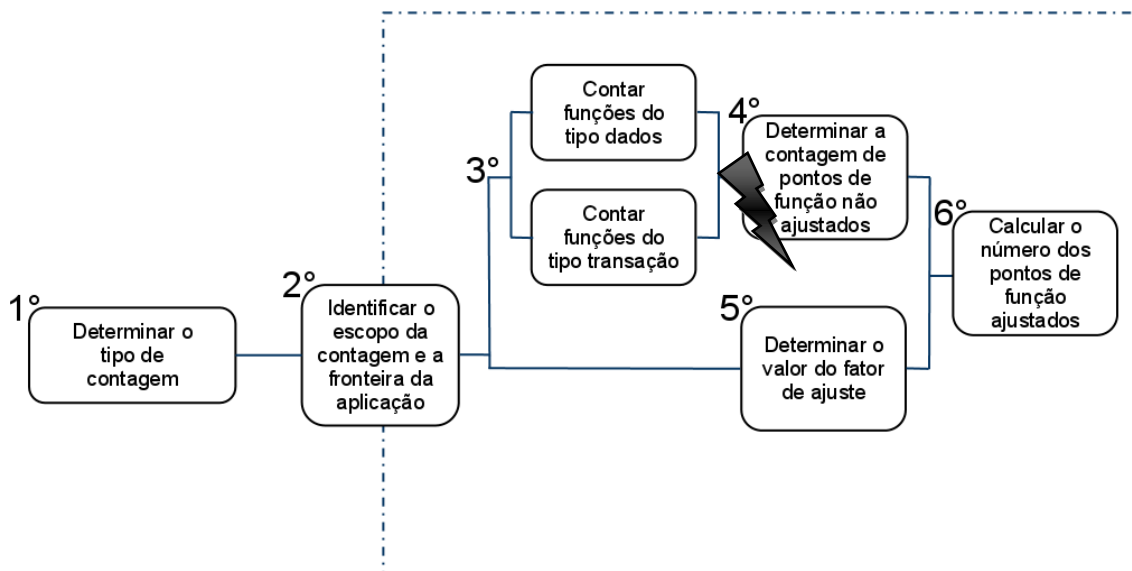
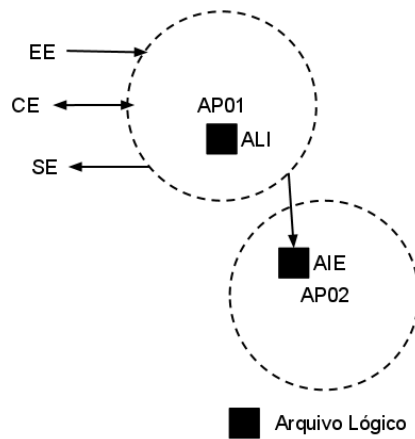


Figura 2.13: Determinar pontos de função não ajustados

Neste ponto nós entendemos a relação de um arquivo lógico com um processo elementar:



**Figura 2.14: Relação arquivo lógico e processo elementar**

Neste exemplo temos uma aplicação AP01 com um arquivo lógico interno e uma série de processos elementares, a mesma realiza uma leitura de um arquivo lógico da aplicação AP02, este arquivo lógico localiza-se fora da fronteira da aplicação AP01 e deve ser classificado como um arquivo de interface externa.

Agora nós devemos realizar a contagem dos pontos de função não ajustados, esta análise é simples. Devemos apenas somar as contribuições das funções do tipo dado com as contribuições das funções do tipo transação.

### 2.6.1 Aplicando o conhecimento

Devemos somar as contribuições de todas as funções do tipo dado e do tipo transação.

Exemplo:

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

**Tabela 2.17: Pontos de função não ajustados**

### 2.7 Determinar o fator de ajuste

O quinto passo é determinar o fator de ajuste:

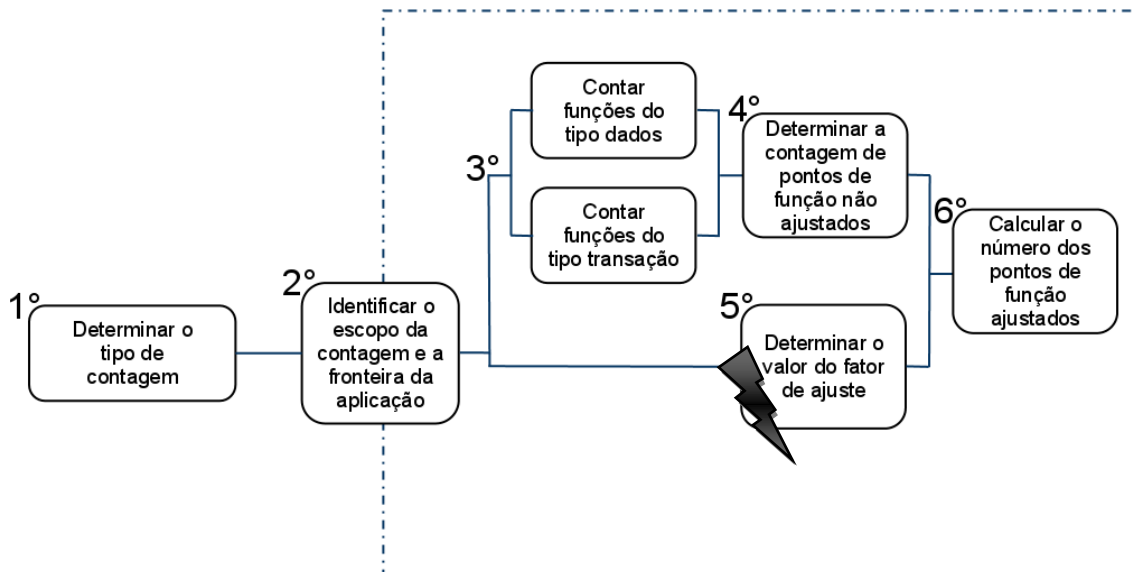


Figura 2.15: Determinar o fator de ajuste

Para o quinto passo devemos determinar o fator de ajuste, mas nós não iremos realizar esta análise e atribuiremos o valor do fator de ajuste como um.

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 da técnica de análise de ponto de função ignorassem esta etapa antes mesmo do IFPUG adotá-la como opcional em 2002.

Este guia tem por objetivo estimar pontos de função, a inclusão de análises subjetivas afetará a contagem e aumentará o erro.

### 2.7.1 Aplicando o conhecimento

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

Exemplo:

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

O VAF sendo um não irá interferir na contagem.

### 2.8 Realizar o cálculo dos pontos de função ajustados

Este é o sexto e último passo para a contagem:



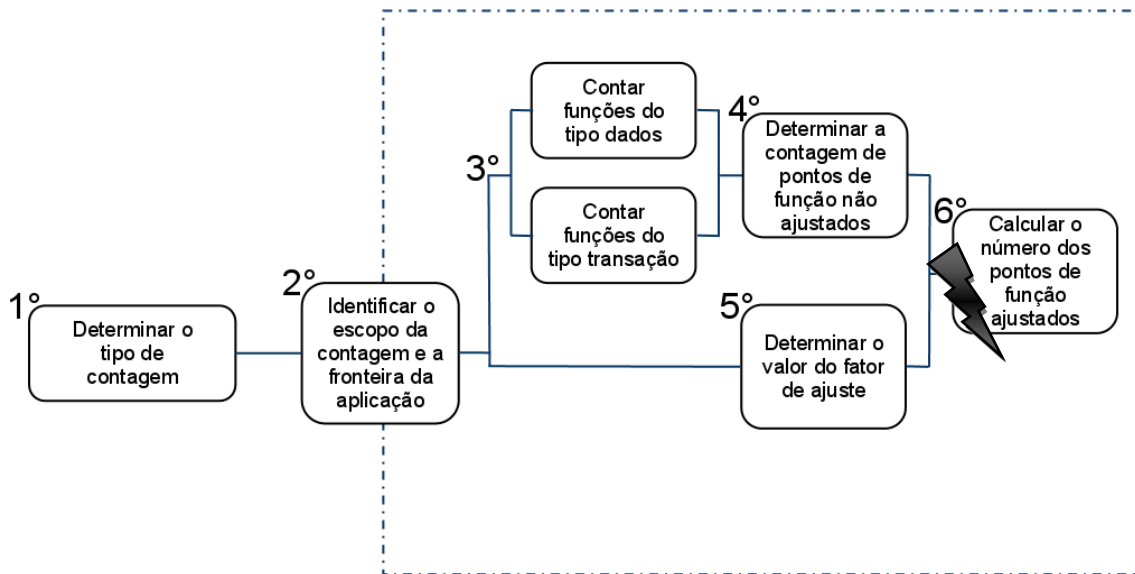


Figura 2.15: Calcular os pontos de função ajustados

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

1. Projeto de Desenvolvimento
2. Projeto de Melhoria
3. Aplicação

Como este guia visa à contagem de projeto de desenvolvimento não entraremos em detalhes dos demais tipos de contagem.

Para determinar os pontos de função ajustados para projeto de desenvolvimento é necessário aplicar a seguinte fórmula:

$$DFP = (UFP + CFP) \times VAF$$

Sendo:

- **DFP:** O 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.

### 2.8.1 Aplicando o conhecimento

Todos os valores estimados até este ponto serão utilizados para determinar os pontos de função ajustados.

Exemplo:

Para terminar a contagem do projeto de desenvolvimento, substitua os valores estimados até aqui na fórmula.

$$DFP = (78 + 0) \times 1$$

A minha aplicação não possui funções de conversão, por este motivo somei zero as funções disponíveis após a instalação.

A minha aplicação contada possui um tamanho funcional estimado em 78 pontos de função.

### 3. DERIVAÇÕES

Neste ponto já possuímos o tamanho funcional da nossa aplicação, agora será apresentado as derivações que podem ser realizadas com ele.

Até aqui utilizamos a análise de pontos de função na perspectiva de produto, agora iremos 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. Na primeira vez que aplicar estas estimativas o erro será grande, mas conforme for ampliando a sua base de históricos de projeto tenderá a diminuir este erro.

#### 3.1 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 sua empresa.

A estimativa de esforço pode ser:

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

Temos por base que a taxa de produtividade é media 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 que incluem tabelas de produtividade mínima no desenvolvimento de projetos.

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 3.1: Tabela de produtividade mínima ACINE

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

### 3.1.1 Aplicando o conhecimento

Exemplo:

A nossa aplicação foi estimada em 78 pontos de função. Considere 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:

$$\text{Esforço} = (5 \times 78)$$

Esta empresa gastaria 390 horas pra produzir o sistema ou três meses.

### 3.2 Custo

A estimativa do custo de um projeto é a informação primordial na hora de elaborar uma proposta, este não pode exceder as expectativas do cliente e nem tão pouco 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 é dado por:

- Custo por hora vezes hora por ponto de função (C/H x H/PF).

Assim nós obtemos o custo por ponto de função.

#### 3.2.1 Aplicando o conhecimento

Exemplo:

Suponha que a hora de trabalho custa R\$ 21,00 e como é produzido um ponto de função a cada cinco horas o valor do ponto de função é de R\$ 105,00.

Estimamos que os esforços necessários para produzir nossa aplicação são de 390 horas e a mesma possui 78 pontos de função.

$$\text{Custo} = (78 \times 100,00)$$

Podemos assim inferir que a aplicação tem um custo de aproximadamente R\$ 7800,00.

### 3.3 Prazo

O prazo é um fator crítico a ser determinado, pois para estimativas nós supomos ele sendo uma função linear com o recurso, o que é uma suposição muito 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 voltamos ao ponto de que a melhor maneira de evitar estes erros é possuindo uma base histórica dos projetos desenvolvidos.

Implicamos o prazo da seguinte forma:

- Prazo é a relação de esforço por recurso.

$$\text{Prazo} = \frac{\text{Esforço}}{\text{Recurso}}$$

### 3.3.1 Aplicando o conhecimento

Exemplo:

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.

$$\text{Prazo} = (3 / 2)$$

Utilizando dessas informações concluímos que o prazo para a entrega do sistema será de um mês e meio.

## 4. CONSIDERAÇÕES

Durante a elaboração deste guia tive contato com outras técnicas de estimativas de software, contagem por linha de código, contagem de telas, pontos por caso de uso e outras, foi possível perceber que das soluções usuais encontradas a Análise de Ponto de Função – APF é uma ótima solução e consegue atender de forma satisfatória as necessidades do mercado.

Para elaboração deste guia foi utilizado artigos, livros e vídeos-aula. Atribuo grande parte do meu conhecimento ao livro Análise de Pontos de Função (VAZQUEZ,2009) que sem o qual eu teria grande dificuldade em terminar o meu projeto. Outra fonte de conhecimento que me foi de grande ajuda, não pela estrutura formal, mas pelos conhecimentos gerados diariamente a partir de dúvidas dos usuários da técnica, foi o grupo de leitores de APF, disponível em: <<http://groups.yahoo.com/group/livro-apf/>>.

Finalizando as considerações, agradeço ao José Luis Braga meu orientador por me possibilitar o conhecimento nesta área e estar sempre à disposição para ajudar e indicar materiais surpreendentes sobre os mais variados conhecimentos desde engenharia de software, gerência de projetos a conhecimentos do mundo e fora dele.

Não se limite a este guia, pois ele apresenta uma visão superficial da técnica de análise de pontos de função e que ele te instigue a buscar mais conhecimentos sobre essa área.

“Um conhecimento nunca é mantido constante, ou ele é perdido ou é enriquecido.”

## 5. BIBLIOGRAFIA

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.

Softex. MPS.BR - Melhoria de processo do software brasileiro - Guia geral, 2009.

IFPUG(International Function Point Users Group). Disponível em: <<http://www.ifpug.org>>. Acesso em: 01 nov 2010.

BFPUG(Brazilian Function Point Users Group). Disponível em: <<http://www.bfpug.com.br>>. Acesso em: 01 nov 2010.

PMI (Project Management Institute). *Um Guia do Conjunto de Conhecimentos em Gerenciamentos de Projetos (PMBOK)*. Estados Unidos: PMI Publications, 2004.

DEKKERS, C. Pontos de Função e Medidas - O Que é um Ponto de Função?. QAI Journal, dez. 1998

DEKKERS, C. Desmistificando Pontos de Função: Entendendo a Terminologia. IT Metrics Strategies, out. 1998.

ACINE. Anexo XVIII – Tabelas de produtividade mínima, 2008. Disponível em: <<http://www.ancine.gov.br/media/concorrencia0012008/AnexoXVIII.pdf>>. Acesso em: 19 jun 2011.