

## - MÓDULO 2.1 - ANÁLISE DE PONTO POR FUNÇÃO - APF

### 1. INTRODUÇÃO

Criada em 1979 por Allan J. Albrecht (IBM), a APF - ANÁLISE DE PONTOS POR FUNÇÃO é uma técnica para medição de projetos cujo objeto seja o desenvolvimento de um software. Este tipo de técnica tem como objetivo estabelecer uma unidade de medida chamada Ponto de Função (PF), que mensura o tamanho do software levando em consideração a funcionalidade implementada sob a perspectiva do usuário, desconsiderando a tecnologia que será utilizada na implementação.

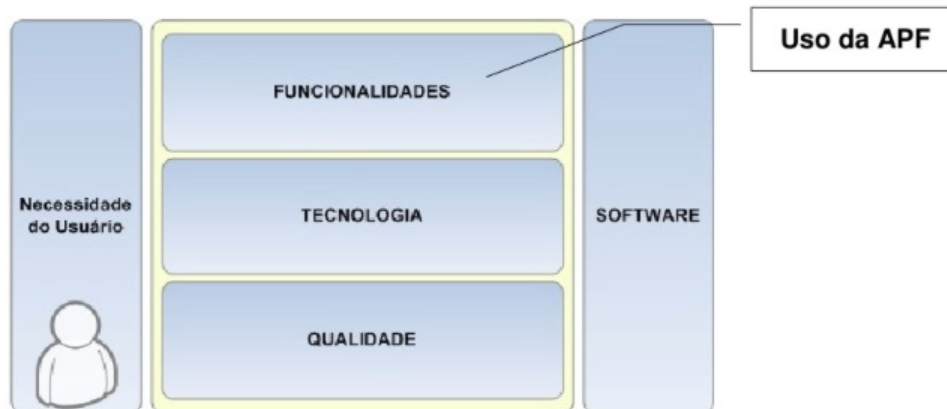


Figura 1 - Perspectiva da APF

Sobre funcionalidades entenda-se:

- Telas
- Arquivos de Controle
- Relatórios
- Arquivos de Referência
- Arquivos mestres
- Sinais

As organizações podem aplicar a Análise de Pontos por Função como:

- Uma ferramenta para determinar o tamanho de pacotes de software adquiridos, através da contagem de todos os Pontos por Função incluídos no pacote;
- Uma ferramenta para apoiar a análise da qualidade e da produtividade;
- Um mecanismo para estimar custos e recursos envolvidos em projetos de desenvolvimento e manutenção de software;
- Um fator de normalização para comparação de software.

#### 1.1. Métrica de Software

Antes de iniciar os estudos sobre o APF é importante responder uma pergunta elementar: Por que medir? Muitas são as respostas:

- Estimar custos e recursos do projeto;
- Avaliar a aquisição de pacotes;
- Suportar análises de produtividade e qualidade;
- Remunerar fornecedores;
- Apoiar a gerência de escopo do projeto;

- Tornar o gerenciamento de projetos baseados em fatos e não em “achismos”.

A APF em conjunto com outras métricas permite a indicação de vários indicadores, como:

- Horas/PF
- PF/Homem/Mês
- R\$/PF
- etc

Medir um sistema é essencial para tomadas de decisões quanto a sua viabilidade, exequibilidade e definição de custos. Se não é possível medir, então não é possível controlar e gerenciar o que deve ser feito.

## 1.2. Objetivos da APF

Sob esse contexto, os objetivos da APF são:

- Medir a funcionalidade solicitada pelo usuário, antes do projeto de software, de forma a estimar seu tamanho e seu custo;
- Medir projetos de desenvolvimento e manutenção de software, independentemente da tecnologia utilizada na implementação, de forma a acompanhar sua evolução;
- Medir a funcionalidade recebida pelo usuário, após o projeto de software, de forma a verificar seu tamanho e custo, comparando-os com o que foi originalmente estimado;

## 2. FUNDAMENTOS DA CONTAGEM DE PONTOS POR FUNÇÃO

O procedimento para contagem de Pontos por Função compreende sete passos (a - g), mostrados na figura 2 e descritos em seguida.

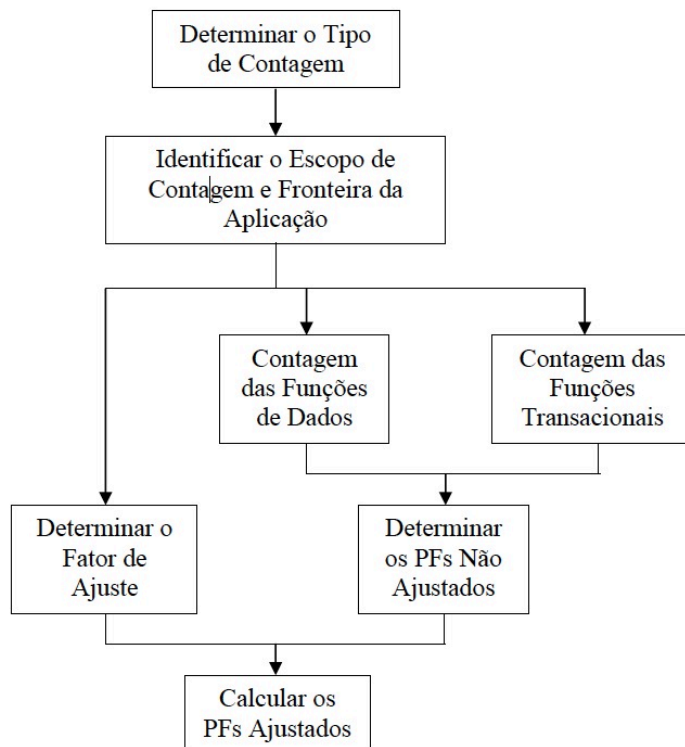


Figura 2 - Procedimento de Contagem de Pontos de Função

**a) Determinar o tipo de contagem de pontos de função:** este é o primeiro passo no processo de contagem, sendo que existem três tipos de contagem: **contagem de PF de projeto de desenvolvimento**, de **aplicações instaladas** e de **projetos de manutenção**.

**b) Identificar o escopo de contagem e a fronteira da aplicação:** neste passo, definem-se as funcionalidades que serão incluídas em uma contagem de PF's específica. A fronteira da aplicação é definida estabelecendo um limite lógico entre a aplicação que está sendo medida, o usuário e outras aplicações. **O escopo de contagem define a parte do sistema (funcionalidades) a ser contada.**

**c) Determinar a contagem de pontos de função não ajustados:** os pontos de função não ajustados (PFNA) refletem as **funcionalidades fornecidas pelo sistema para o usuário**. Essa contagem leva em conta dois tipos de função: de **dados e transacionais**, bem como sua **complexidade (simples, média ou complexa)**.

**d) Contagem das funções de dados:** as funções de dados representam as funcionalidades relativas aos requisitos de **dados internos e externos à aplicação**. São elas os arquivos lógicos internos e os arquivos de interface externa. Ambos são grupos de dados logicamente relacionados ou informações de controle que foram identificados pelo usuário. A diferença está no fato de um **Arquivo Lógico Interno (ALI)** ser mantido dentro da fronteira da aplicação, isto é, **armazenar os dados** mantidos através de um ou mais processos elementares da aplicação, enquanto que um **Arquivo de Interface Externa (AIE)** é apenas referenciado pela aplicação, ou seja, ele **é mantido dentro da fronteira de outra aplicação**. Assim, o objetivo de um AIE é armazenar os dados referenciados por um ou mais processos elementares da aplicação sendo contada, mas que são mantidos por outras aplicações.

**e) Contagem das funções transacionais:** as funções transacionais representam as funcionalidades de processamento de dados do sistema fornecidas para o usuário. São elas: as entradas externas, as saídas externas e as consultas externas. As **Entradas Externas (EE)** são **processos elementares** que processam dados (ou informações de controle) **que entram pela fronteira da aplicação**. O objetivo principal de uma EE é **manter um ou mais ALI's ou alterar o comportamento do sistema**. As **Saídas Externas (SE)** são **processos elementares que enviam dados (ou informações de controle) para fora da fronteira da aplicação**. Seu objetivo é **mostrar informações recuperadas através de um processamento lógico** (isto é, **que envolva cálculos** ou criação de dados derivados) e não apenas uma simples recuperação de dados. Uma SE pode, também, manter um ALI ou alterar o comportamento do sistema. Por fim, uma **Consulta Externa (CE)**, assim como uma SE, é um processo elementar que **envia dados (ou informações de controle) para fora da fronteira da aplicação, mas sem realização de nenhum cálculo nem a criação de dados derivados**. Seu objetivo é **apresentar informação para o usuário, por meio apenas de uma recuperação das informações**. Nenhum ALI é mantido durante sua realização, nem o comportamento do sistema é alterado.

**f) Determinar o valor do fator de ajuste:** o fator de ajuste é baseado em 14 características gerais de sistemas, que **avaliam a funcionalidade geral da aplicação que está sendo contada, e seus níveis de influência**. O nível de influência de uma característica é determinado com base em uma **escala de 0 (nenhuma influência) a 5 (forte influência)**.

**g) Calcular os pontos de função ajustados:** finalmente, os **PF's ajustados são calculados, considerando-se o tipo de contagem** definido no primeiro passo.

A figura 3 apresenta uma visão geral dos tipos de função que são considerados na contagem da APF.

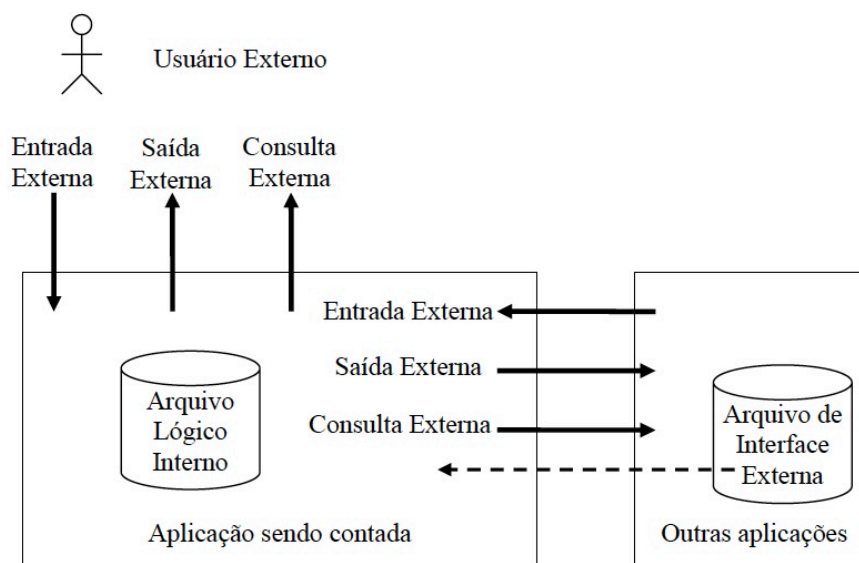


Figura 3 - Visão Geral das Funções de uma Aplicação segundo a APF

## 2.1. Contagem das Funções de Dados

Conforme discutido anteriormente, o primeiro passo para a contagem das funções de dados consiste em **identificar arquivos lógicos internos (ALI's)** e **arquivos de interface externa (AIE's)**. Cada uma dessas funções de dados deve ser classificada segundo sua complexidade funcional. Essa complexidade é definida com base em dois conceitos:

- Registros lógicos; e
- Itens de dados.

**Registros Lógicos** são **subconjuntos** de dados dentro de um ALI/AIE, que foram reconhecidos pelo usuário. **Se o usuário não reconhecer subconjuntos de dados em um ALI/AIE, então se deve contar o ALI/AIE como um registro lógico.**

Um **Item de Dados**, por sua vez, **é um campo** reconhecido pelo usuário como único e não repetido. Vale destacar que só devem ser contados os itens de dados utilizados pela aplicação em contagem.

Contando-se os registros lógicos e os itens de dados de um ALI/AIE, pode-se chegar à sua complexidade, utilizando a tabela 1.

Tabela 1 - Tabela de Identificação da Complexidade das **Funções de Dados**

Número de Registros Lógicos	Número de Itens de Dados Referenciados		
	De 1 a 19	De 20 a 50	51 ou mais
Apenas 1	Simple	Simple	Média
De 2 a 5	Simple	Média	Complexa
6 ou mais	Média	Complexa	Complexa

Exemplo:

Um DER, com três entidades e dois relacionamentos:

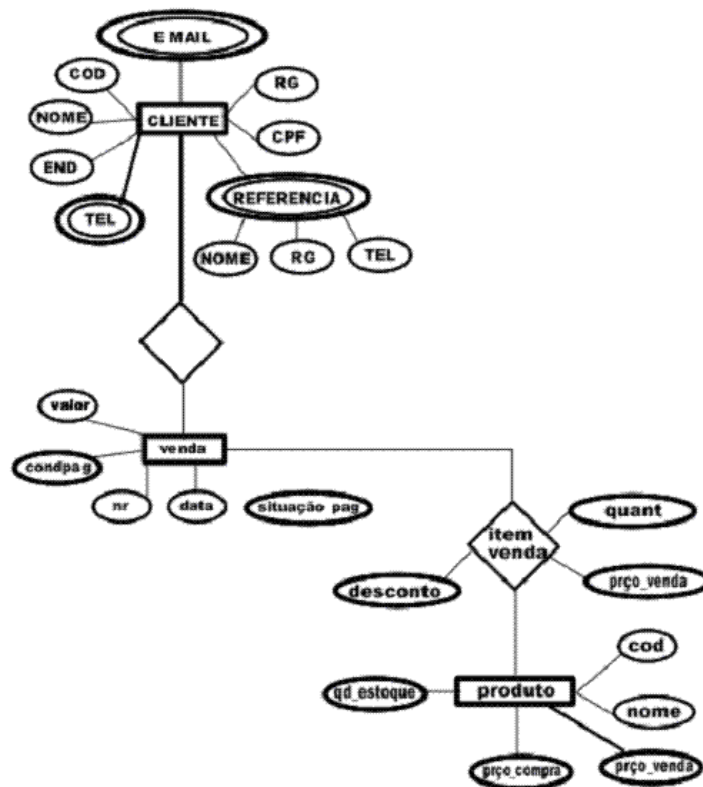


Figura 4 – DER Exemplo

**Arquivos:** Cliente, Venda e Produto (ALI), pois tratam-se de tabelas do próprio sistema e não externas ao sistema.

**Itens de dados:** 21. Todos os atributos, exceto os multivalorados, pois os atributos multivalorados serão contados como registros.

**Registros Lógicos:** 6 (Email, Cliente, Referencia, Telefone, venda, Item\_venda e Produto). A própria entidade Cliente é um Registro lógico e o atributo multivalorado Telefone também é um Registro lógico, toda entidade e atributo multivalorado são registros lógicos.

**Complexidade:** Complexo. Vide a tabela 1.

## 2.2. Contagem das Funções Transacionais

De maneira análoga à contagem das funções de dados, a contagem das funções transacionais envolve a identificação de funções transacionais (entradas externas, saídas externas e consultas externas) e sua classificação de acordo com a complexidade funcional envolvida (simples, média ou complexa).

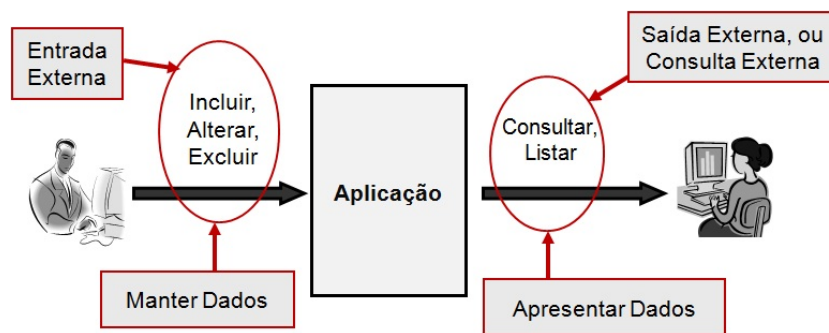


Figura 5 – Funções de Transação

- **Entrada Externa (EE):** Processo elementar da aplicação que processa dados ou informações de controle que vem de fora da fronteira da aplicação. A entrada externa tem a intenção de manter os ALI's ou alterar o comportamento da aplicação. Exemplo: Inserções, atualizações, validações, fórmulas, cujos parâmetros vem de fora.

- **Saída Externa (SE):** Processo elementar da aplicação que gera dados ou informações de controle que são enviados para fora da fronteira da aplicação. Exemplo: relatórios e gráficos.

- **Consulta Externa (CE):** Processo elementar da aplicação que representa um combinação de entrada e saída. Exemplo: consultas implícitas, verificação de senhas e recuperação de dados baseados em parâmetros.

A definição da complexidade funcional é feita com base no número de arquivos referenciados e dos itens de dados manipulados pela função, utilizando as tabelas 2 para entradas externas e 3 para saídas e consultas externas.

Nessas tabelas, um arquivo referenciado pode ser um ALI lido ou mantido pela função transacional, ou um AIE lido pela função transacional. Já o número de itens de dados referenciados é calculado considerando apenas os itens de dados efetivamente referenciados pela função transacional em questão.

Tabela 2 - Tabela de Identificação da Complexidade de Entradas Externas

Número de Arquivos Referenciados	Número de Itens de Dados Referenciados		
	De 1 a 4	De 5 a 15	16 ou mais
0 ou 1	Simple	Simple	Média
2	Simple	Média	Complexa
3 ou mais	Média	Complexa	Complexa

Tabela 3 - Tabela de Identificação da Complexidade de Saídas e Consultas Externas

Número de Arquivos Referenciados	Número de Itens de Dados Referenciados		
	De 1 a 5	De 6 a 19	20 ou mais
0 ou 1	Simple	Simple	Média
2 ou 3	Simple	Média	Complexa
4 ou mais	Média	Complexa	Complexa

Exemplo:

A seguir observa-se um componente do tipo *drop-down* que pode ser contado como consulta externa.

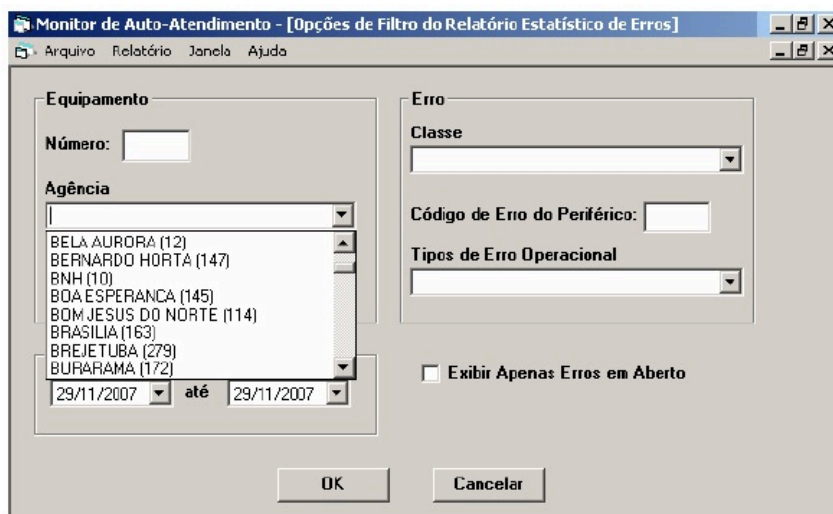


Figura 6 – Drop-down

Neste exemplo, há um processo elementar de consulta à relação de agências com seus respectivos códigos. Este processo é classificado como uma consulta externa, cujo nome poderia ser Agência – Consultar Lista (*drop-down*). Há três tipos de dados: *comando*, *nome da agência* e *código da agência*. Somente o arquivo de agência é referenciado.

**Arquivos referenciados:** 1 (Agência)

**Itens de dados referenciados:** 3 (comando, nome da agência e código da agência)

**Complexidade:** Simples. Vide a tabela 3.

### 2.3. Cálculo dos Pontos de Função Não Ajustados (PFNA)

Uma vez determinadas a complexidade das funções de dados e as funções transacionais, é possível calcular os PF's não ajustados de uma aplicação. Esse cálculo é feito computando os totais de pontos de função para cada um dos cinco tipos de função (ALI, AIE, EE, SE e CE), considerando a tabela 4.

Tabela 4 - Contribuição das Funções na Contagem de PF's Não Ajustados

FUNÇÃO	COMPLEXIDADE		
	Simples	Média	Complexa
ALI	7	10	15
AIE	5	7	10
EE	3	4	6
SE	4	5	7
CE	3	4	6

Para calcular os **pontos de função não ajustados**, multiplica-se o número de funções identificadas para uma determinada complexidade por sua contribuição. Ao final, soma-se todos os pontos de função encontrados.

A seguir é apresentado um exemplo para o cálculo dos pontos de função não ajustados (PFNA) gerados pelos ALI de um sistema hipotético. O mesmo deve ser feito para a outras funções do sistema (AIE, EE, SE e CE).

Tabela 5 - Contribuição das Funções na Contagem de PF's Não Ajustados

Função	Contagem de pontos por complexidade	Contribuição	Total por Complexidade	Total de PDNA
ALI	1 Baixa	x 7	7	42
	2 Média	x 10	20	
	1 Alta	x 15	15	

## 2.4. Cálculo dos Pontos de Função Ajustados (PFA)

O número de pontos de função não ajustados de um sistema reflete a funcionalidade que o sistema fornecerá ao usuário, **sem considerar as especificidades do sistema**. Por exemplo, um mesmo sistema pode ser implementado para operar *stand alone* para um cliente e em arquitetura cliente-servidor para outro. As funcionalidades seriam as mesmas, o que resultaria na mesma contagem de pontos de função não ajustados, mas quando considera-se as características do sistema para cada cliente, observa-se que os pontos de função devem ser ajustados para refletir a maior complexidade do sistema na arquitetura cliente servidor.

Para ajustar os pontos de função encontrados na etapa anterior devem ser levadas em consideração **14 (quatorze) características do sistema** que serão analisadas e fornecerão o valor do fator de ajuste.

- a. Comunicação de Dados
- b. Processamento de Dados Distribuído
- c. Desempenho
- d. Utilização do Equipamento (Restrições de Recursos Computacionais)
- e. Volume de Transações
- f. Entrada de Dados On-line
- g. Eficiência do Usuário Final (Usabilidade)
- h. Atualização On-line
- i. Processamento Complexo
- j. Reusabilidade
- k. Facilidade de Implantação
- l. Facilidade Operacional (Processos Operacionais, tais como Inicialização, Cópia de Segurança, Recuperação etc)
- m. Múltiplos Locais e Organizações do Usuário
- n. Facilidade de Mudanças (Manutenibilidade)

### 2.4.1. COMUNICAÇÃO DE DADOS

Os aspectos relacionados aos recursos utilizados para a comunicação de dados do sistema deverão ser descritos de forma global. Descrever se a aplicação utiliza protocolos diferentes para recebimento/envio das informações do sistema.

0. Aplicação batch ou funciona *stand alone*;
1. Aplicação batch, mas utiliza entrada de dados ou impressão remota;
2. Aplicação batch, mas utiliza entrada de dados e impressão remota;



3. Aplicação com entrada de dados on-line para alimentar processamento batch ou sistema de consulta;

4. Aplicação com entrada de dados on-line, mas suporta apenas um tipo de protocolo de comunicação;

5. Aplicação com entrada de dados on-line e suporta mais de um tipo de protocolo de comunicação.

#### 2.4.2. PROCESSAMENTO DE DADOS DISTRIBUÍDO:

Esta característica refere-se a sistemas que utilizam dados ou processamento distribuído, valendo-se de diversas CPU's.

0. Aplicação não auxilia na transferência de dados ou funções entre os processadores da empresa;

1. Aplicação prepara dados para o usuário final utilizar em outro processador (do usuário final), tal como planilhas;

2. Aplicação prepara dados para transferência, transfere-os para serem processados em outro equipamento da empresa (não pelo usuário final);

3. Processamento é distribuído e a transferência de dados é *on-line* e apenas em uma direção;

4. Processamento é distribuído e a transferência de dados é *on-line* e em ambas as direções;

5. As funções de processamento são dinamicamente executadas no equipamento (CPU) mais apropriada;

#### 2.4.3. DESEMPENHO

Trata-se de parâmetros estabelecidos pelo usuário como aceitáveis, relativos a tempo de resposta.

0. Nenhum requisito especial de desempenho foi solicitado pelo usuário;

1. Requisitos de desempenho foram estabelecidos e revistos, mas nenhuma ação especial foi requerida;

2. Tempo de resposta e volume de processamento são itens críticos durante horários de pico de processamento. Nenhuma determinação especial para a utilização do processador foi estabelecida. A data limite para a disponibilidade de processamento é sempre o próximo dia útil;

3. Tempo de resposta e volume de processamento são itens críticos durante todo o horário comercial. Nenhuma determinação especial para a utilização do processador foi estabelecida. A data-limite necessária para a comunicação com outros sistemas é limitante.

4. Os requisitos de desempenho estabelecidos requerem tarefas de análise de desempenho na fase de planejamento e análise da aplicação.

5. Além do descrito no item anterior, ferramentas de análise de desempenho foram usadas nas fases de planejamento, desenvolvimento e/ou implementação para atingir os requisitos de desempenho estabelecidos pelos usuários.

#### 2.4.4. UTILIZAÇÃO DO EQUIPAMENTO

Trata-se de observações quanto ao nível de utilização de equipamentos requerido para a execução do sistema. Este aspecto é observado com vista a planejamento de capacidades e custos.

0. Nenhuma restrição operacional explícita ou mesmo implícita foi incluída.

1. Existem restrições operacionais leves. Não é necessário esforço especial para atender às restrições.

2. Algumas considerações de ajuste de desempenho e segurança são necessárias.

3. São necessárias especificações especiais de processador para um módulo específico da aplicação.

4. Restrições operacionais requerem cuidados especiais no processador central ou no processador dedicado para executar a aplicação.

5. Além das características do item anterior, há considerações especiais que exigem utilização de ferramentas de análise de desempenho, para a distribuição do sistema e seus componentes, nas unidades processadoras.

#### 2.4.5. VOLUME DE TRANSAÇÕES

Consiste na avaliação do nível de influência do volume de transações no projeto, desenvolvimento, implantação e manutenção do sistema.

#### 2.4.6. ENTRADA DE DADOS ON-LINE

A análise desta característica permite quantificar o nível de influência exercida pela utilização de entrada de dados no modo on-line no sistema.

0. Todas as transações são processadas em modo batch.

1. De 1% a 7% das transações são entradas de dados on-line.

2. De 8% a 15% das transações são entradas de dados on-line.

3. De 16% a 23% das transações são entradas de dados on-line.

4. De 24% a 30% das transações são entradas de dados on-line.

5. Mais de 30% das transações são entradas de dados on-line.

#### 2.4.7. USABILIDADE

A análise desta característica permite quantificar o grau de influência relativo aos recursos implementados com vista a tornar o sistema amigável, permitindo incrementos na eficiência e satisfação do usuário final, tais como:

- Auxílio à navegação (teclas de função, acesso direto e menus dinâmicos)

- Menus Documentação e help *on-line*

- Movimento automático do cursor.

- Movimento horizontal e vertical de tela.

- Impressão remota (via transações *on-line*)

- Teclas de função preestabelecidas.

- Processos batch submetidos a partir de transações *on-line*

- Utilização intensa de campos com vídeo reverso, intensificados, sublinhados, coloridos e outros indicadores.

- Impressão da documentação das transações *on-line* através de *hard copy*.

- Utilização de mouse.

- Menus pop-up.

- O menor número possível de telas para executar as funções de negócio.

- Suporte bilíngue. (contar como 4 itens)

- Suporte multilíngue. (contar como 6 itens)

Pontuação:

0. Nenhum dos itens descritos.

1. De um a três itens descritos.

2. De quatro a cinco dos itens descritos.

3. Mais de cinco dos itens descritos, mas não há requisitos específicos do usuário quanto a usabilidade do sistema.

4. Mais de cinco dos itens descritos e foram estabelecidos requisitos quanto à usabilidade fortes o suficiente para gerarem atividades específicas envolvendo fatores, tais como minimização da digitação, para mostrar inicialmente os valores utilizados com mais frequência.

5. Mais de cinco dos itens descritos e foram estabelecidos requisitos quanto à usabilidade fortes o suficiente para requerer ferramentas e processos especiais para demonstrar antecipadamente que os objetivos foram alcançados.

#### 2.4.8. ATUALIZAÇÕES ON-LINE

Mede a influência no desenvolvimento do sistema face à utilização de recursos que visem a atualização dos Arquivos Lógicos Internos, no modo *on-line*.

0. Nenhuma.

1. Atualização *on-line* de um a três arquivos lógicos internos. O volume de atualização é baixo e a recuperação de dados é simples.

2. Atualização *on-line* de mais de três arquivos lógicos internos. O volume de atualização é baixo e a recuperação dos dados é simples.

3. Atualização *on-line* da maioria dos arquivos lógicos internos.

4. Em adição ao item anterior, é necessária proteção contra perdas de dados que foi projetada e programada no sistema.

5. Além do item anterior, altos volumes trazem considerações de custo no processo de recuperação. Processos para automatizar a recuperação foram incluídos minimizando a intervenção do operador.

#### 2.4.9. PROCESSAMENTO COMPLEXO

A complexidade de processamento influencia no dimensionamento do sistema, e, portanto, deve ser quantificado o seu grau de influência, com base nas seguintes categorias:

- Processamento especial de auditoria e/ou processamento especial de segurança foram considerados na aplicação;

- Processamento lógico extensivo;

- Processamento matemático extensivo;
- Processamento gerando muitas exceções, resultando em transações incompletas que devem ser processadas novamente. Exemplo: transações de autoatendimento bancário interrompidas por problemas de comunicação ou com dados incompletos;
- Processamento complexo para manusear múltiplas possibilidades de entrada/saída.

Exemplo: multimídia.

Pontuação

0. Nenhum dos itens descritos.
1. Apenas um dos itens descritos.
2. Dois dos itens descritos.
3. Três dos itens descritos.
4. Quatro dos itens descritos.
5. Todos os cinco itens descritos.

### 2.4.10. REUSABILIDADE

A preocupação com o reaproveitamento de parte dos programas de uma aplicação em outras aplicações implica em cuidados com padronização. O grau de influência no dimensionamento do sistema é quantificado observando-se os seguintes aspectos:

0. Nenhuma preocupação com reutilização de código.
1. Código reutilizado foi usado somente dentro da aplicação.
2. Menos de 10% da aplicação foi projetada prevendo utilização posterior do código por outra aplicação.
3. 10% ou mais da aplicação foi projetada prevendo utilização posterior do código por outra aplicação.
4. A aplicação foi especificamente projetada e/ou documentada para ter seu código reutilizado por outra aplicação e a aplicação é customizada pelo usuário em nível de código-fonte.
5. A aplicação foi especificamente projetada e/ou documentada para ter seu código facilmente reutilizado por outra aplicação e a aplicação é customizada para uso através de parâmetros que podem ser alterados pelo usuário.

### 2.4.11. FACILIDADE DE IMPLANTAÇÃO

A quantificação do grau de influência desta característica é feita, observando-se o plano de conversão e implantação e/ou ferramentas utilizadas durante a fase de testes do sistema.

0. Nenhuma consideração especial foi estabelecida pelo usuário e nenhum procedimento especial é requerido na implantação.
1. Nenhuma consideração especial foi estabelecida pelo usuário, mas procedimentos especiais são necessários na implementação.
2. Requisitos de conversão e implantação foram estabelecidos pelo usuário e roteiro de conversão e implantação foram providos e testados. O impacto da conversão no projeto não é considerado importante.
3. Requisitos de conversão e implantação foram estabelecidos pelo usuário e roteiro de conversão e implantação foram providos e testados. O impacto da conversão no projeto é considerado importante.

4. Além do item 2, conversão automática e ferramentas de implantação foram providas e testadas.

5. Além do item 3, conversão automática e ferramentas de implantação foram providas e testadas.

#### 2.4.12. FACILIDADE OPERACIONAL

A análise desta característica permite quantificar o nível de influência na aplicação, com relação a procedimentos operacionais automáticos que reduzem os procedimentos manuais, bem como mecanismos de inicialização, salvamento e recuperação, verificados durante os testes do sistema.

0. Nenhuma consideração especial de operação, além do processo normal de salvamento foi estabelecida pelo usuário.

1-4. Verifique quais das seguintes afirmativas podem ser identificadas na aplicação. Selecione as que forem aplicadas. Cada item vale um ponto, exceto se definido explicitamente:

- Foram desenvolvidos processos de inicialização, salvamento e recuperação, mas a intervenção do operador é necessária.

- Foram estabelecidos processos de inicialização, salvamento e recuperação, e nenhuma intervenção do operador é necessária (conte como dois itens).

- A aplicação minimiza a necessidade de montar fitas magnéticas.

- A aplicação minimiza a necessidade de manuseio de papel.

5. A aplicação foi desenhada para trabalhar sem operador, nenhuma intervenção do operador é necessária para operar o sistema além de executar e encerrar a aplicação. A aplicação possui rotinas automáticas para recuperação em caso de erro.

#### 2.4.13. MÚLTIPLOS LOCAIS E ORGANIZAÇÕES DO USUÁRIO

Consiste na análise da arquitetura do projeto, observando-se a necessidade de instalação do sistema em diversos lugares.

0. Os requisitos do usuário não consideraram a necessidade de instalação em mais de um local.

1. A necessidade de múltiplos locais foi considerada no projeto e a aplicação foi desenhada para operar apenas em ambientes de software e hardware idênticos.

2. A necessidade de múltiplos locais foi considerada no projeto e a aplicação está preparada para trabalhar apenas em ambientes similares de software e hardware.

3. A necessidade de múltiplos locais foi considerada no projeto e a aplicação está preparada para trabalhar em diferentes ambientes de hardware e/ou software.

4. Plano de documentação e manutenção foram providos e testados para suportar a aplicação em múltiplos locais, além disso, os itens 1 ou 2 caracterizam a aplicação.

5. Plano de documentação e manutenção foram providos e testados para suportar a aplicação em múltiplos locais, além disso, o item 3 caracteriza a aplicação.

#### 2.4.14. FACILIDADE DE MUDANÇAS

Focaliza a preocupação com a influencia da manutenção no desenvolvimento do sistema. Esta influência deve ser quantificada baseando na observação de atributos, tais como:

- disponibilidade de facilidades como consultas e relatórios flexíveis para atender necessidades simples (conte como 1 item);
- disponibilidade de facilidades como consultas e relatórios flexíveis para atender necessidades de complexidade média (conte como 2 itens);
- disponibilidade de facilidades como consultas e relatórios flexíveis para atender necessidades complexas (conte 3 itens);
- se os dados de controle são armazenados em tabelas que são mantidas pelo usuário através de processos on-line, mas mudanças têm efeitos somente no dia seguinte;
- se os dados de controle são armazenados em tabelas que são mantidas pelo usuário através de processos on-line, as mudanças têm efeito imediatamente (conte como 2 itens).

Para cada uma dessas 14 características deve-se atribuir um valor de 0 (nenhuma influência) a 5 (forte influência), dito grau ou nível de influência, que indica o quanto determinada característica tem influência no sistema.

O fator de ajuste influencia os pontos de função não ajustados em +/- 35%, obtendo-se o número de PF's ajustados (PFA). Para cada característica deve ser atribuído um nível de influência, conforme tabela 6.

**Tabela 6 – Graus de Influência**

<b>Grau</b>	<b>Descrição</b>
0	Nenhum influência
1	Influência mínima
2	Influência moderada
3	Influência média
4	Influência significativa
5	Influência forte

Para calcular o valor do fator de ajuste deve seguir a relação:

$$\text{FATOR DE AJUSTE} = (\text{GRAU DE INFLUÊNCIA} * 0,01) + 0,65$$

Depois de calculado o valor do fator de ajuste, os pontos por função não ajustados podem ser ajustados multiplicando-se o valor do fator de ajuste pelos pontos por função não ajustados.

$$\text{PFA} = \text{PFNA} * \text{FATOR DE AJUSTE}$$

O número de pontos por função encontrado representa o tamanho da aplicação de acordo com sua funcionalidade.

### 3. REFERÊNCIA BIBLIOGRÁFICA

R. Dias, "Análise por Pontos de Função: Uma Técnica para Dimensionamento de Sistemas de Informação", on-line. Disponível em: [www.presidentekennedy.br/resi/edicao03/artigo02.pdf](http://www.presidentekennedy.br/resi/edicao03/artigo02.pdf).

HAZAN, Cláudia. Análise de Pontos por Função: Uma Abordagem Gerencial. Rio de Janeiro: Congresso Nacional da SBC, Jornada de Atualização em Informática (JAI); 2000.

HAZAN, Cláudia. Análise de pontos por função: Uma ferramenta na implantação o modelo CMM. Rio de Janeiro: BFPUG, 2001.